UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE INFORMATICA

TESIS DOCTORAL

Conditional narrowing modulo membership equational logic theories with SMT solvers

Estrechamiento condicional módulo lógicas ecuacionales de pertenencia con resolutores SMT

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Luis Manuel Aguirre García

Directores

Narciso Martí Oliet
Miguel Palomino Tarjuelo
María Isabel Pita Andreu

Madrid

UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMÁTICA

TESIS DOCTORAL

Conditional narrowing modulo membership equational logic theories with SMT solvers

Estrechamiento condicional módulo lógicas ecuacionales de pertenencia con resolutores SMT

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Luis Manuel Aguirre García

DIRECTORES

Narciso Martí Oliet
Miguel Palomino Tarjuelo
María Isabel Pita Andreu

Universidad Complutense de Madrid
Facultad de Informática

Tesis doctoral

Conditional narrowing modulo membership equational logic theories with SMT solvers

Estrechamiento condicional módulo lógicas ecuacionales de pertenencia con resolutores SMT

Memoria para optar al grado de Doctor en Ingeniería Informática presentada por

Luis Manuel Aguirre García

Directores

Narciso Martí Oliet
Miguel Palomino Tarjuelo
María Isabel Pita Andreu

Madrid, noviembre de 2023

Agradecimientos

Han sido muchas las personas que han colaborado al buen fin de esta tesis, aunque solo fuera dándome ánimo cuando este flaqueaba. Inevitablemente, en este reconocimiento no se menciona a alguna de ellas, pero no por voluntad propia sino porque mi memoria es como mi ánimo: a veces flaquea. Sirva este párrafo para solicitar disculpas a los omitidos.

En primer lugar, mi agradecimiento a mi familia: a mi esposa Ana y a mis tres hijos Alejandro, Alberto y Luis. Sin su apoyo constante, entreverado con alguna "bienintencionada" puya, no habría tenido la perseverancia requerida para terminar esta tesis.

A mis directores de tesis, Narciso, Miguel e Isabel, por toda su ayuda a lo largo de estos años leyendo, corrigiendo y puliendo los trabajos resultantes de las innumerables tardes de reunión en las que la pizarra terminaba casi siempre llena de fórmulas y reglas de derivación ininteligibles. Narciso, además, ha sido el cronista de la línea del tiempo de la lógica de reescritura que me ha permitido entender el origen y el porqué del estado pasado y actual tanto de la lógica de reescritura como del lenguaje Maude y sus distintas versiones implementadas.

A Santiago Escobar, por su total disponibilidad. Al ser sus temas de investigación tan afines a los de esta tesis, siempre ha sido el primer consultado ante cualquier disyuntiva.

A David de Frutos, por nuestra larga relación de años. Inicialmente tuvo que lidiar con mi yo "prelicenciado" durante tres cursos. Años después me convenció para que siguiera ligado a la universidad otros tantos cursos como profesor asociado. Cuando le pedí consejo, previo a iniciar mis estudios de posgrado, me recomendó encarecidamente a Narciso como la persona idónea para guiarme en ese nuevo camino, y no se equivocaba.

A Rubén Rubio. Su rauda y efectiva respuesta siempre que surgía alguna duda teórica sobre estrategias o algún problema con las versiones alfa más recientes de Maude, ha facilitado enormemente el desarrollo del prototipo para el último cálculo de esta tesis.

A Óscar Martín. Aunque los temas de nuestras tesis eran disimilares, siempre había algo que aprender de sus trabajos, así como de las charlas que mantuvimos en los descansos del congreso y la escuela de verano a los que asistimos conjuntamente.

Agradezco tanto su disposición a participar en esta tesis como sus comentarios a los miembros del tribunal y a los evaluadores externos, Santiago Escobar y Camilo Rocha. También quiero expresar mi agradecimiento a los revisores anónimos que han accedido a leer nuestros artículos y cuyos comentarios han resultado tan útiles.

Finalmente, a los distintos proyectos que durante este tiempo han apoyado económicamente la realización de esta tesis.

Abstract

This dissertation is an exploration on how to improve the efficiency of conditional narrowing modulo membership equational logic theories for symbolic reachability problems in rewriting logic, using the Maude language. To this end, different approaches have been studied in the dissertation. For each of these approaches a reachability calculus has been defined, and a proof of the soundness and weak completeness of the calculus has been given. Also, several prototypes have been developed for some of the calculi.

Rewriting logic is an over thirty years old computational logic, that focuses on the specification of concurrent systems. This logic has been designed with a precise mathematical semantics that allows to prove properties of the specified systems. Reachability in rewriting logic, which is usually related to checking the safety properties of specifications, is of the utmost interest

Although rewriting can only solve reachability problems having an initial ground state, except for very special cases of initial symbolic states, at some point it was proved that narrowing, that was just a unification method, could also be used to solve reachability problems for any initial symbolic state.

Some prototypes for narrowing have been developed in Maude using its reflective capabilities, and later included as new features of the Maude engine, achieving faster performance through their implementation in C++.

This dissertation takes narrowing one step ahead in several directions, extending both the types of rewrite theories suitable for narrowing and the form of the reachability problems that can be addressed.

All the calculi shown in this dissertation are based on the unification of a term and the head of a rule, modulo a subset of the equational theory within the given rewrite theory: its axioms. The first calculus that will be presented focuses on using the information of the sorts of the two terms that are being unified, to avoid the application of rules when their unification does not generate a term with the required sort. The second calculus uses normalization of the current reachability term and non reducibility of the composition of the unifiers that have been applied in the computation to prune the search space. The third calculus focuses on the use of an SMT solver as an oracle for some of the subtheories within the equational theory included in the given rewrite theory. Finally, the fourth calculus uses a strategy language, together with the SMT solver, to further prune the search space, and it also allows for the parameterization of the rewrite theories and the strategies defined in a given reachability problem.

Resumen

Esta tesis es un estudio sobre cómo mejorar la eficiencia del estrechamiento condicional módulo lógicas ecuacionales de pertenencia para problemas simbólicos de alcanzabilidad en lógica de reescritura, usando el lenguaje Maude. Con este fin, se han analizado distintos enfoques en la tesis. Para cada uno de ellos se ha definido un cálculo de alcanzabilidad y se ha demostrado su corrección y completitud débil. También se han desarrollado varios prototipos para algunos de estos cálculos.

La lógica de reescritura es una lógica computacional orientada a la especificación de sistemas concurrentes. Esta lógica se ha diseñado con una semántica matemática precisa que permite probar propiedades de los sistemas especificados. La resolución de problemas de alcanzabilidad en lógica de reescritura es del mayor interés, ya que normalmente está asociada a la comprobación de propiedades de seguridad de las especificaciones.

Aunque la reescritura sólo permite resolver problemas de alcanzabilidad con un estado inicial carente de variables, excepto para casos muy especiales de estados iniciales simbólicos, en un momento dado se demostró que el estrechamiento, que hasta entonces se utilizaba básicamente como método de unificación, se podía usar también para resolver problemas de alcanzabilidad con un estado simbólico inicial cualquiera.

Algunos prototipos de alcanzabilidad han sido desarrollados en Maude usando sus capacidades reflexivas, y posteriormente se han añadido al motor de Maude como nuevas opciones, alcanzando un mejor rendimiento a través de su implementación en C++.

Esta tesis lleva la alcanzabilidad un paso más allá, extendiendo tanto las clases de teorías de reescritura aceptables para el estrechamiento como los tipos de problemas de alcanzabilidad que se pueden formular.

Todos los cálculos mostrados en esta tesis están basados en la unificación de un término con la cabeza de una regla, módulo un subconjunto de la teoría ecuacional subyacente a la teoría de reescritura considerada: sus axiomas. El primer cálculo mostrado en esta tesis se centra en el uso de la información de los tipos de los dos términos que se están unificando, para evitar aplicar reglas cuando la unificación no genera un término con el tipo requerido. El segundo cálculo usa la normalización del término sobre el que se intenta la alcanzabilidad así como la no reducibilidad de la composición de los unificadores aplicados en el cálculo para podar el espacio de estados. El tercer cálculo se centra en el uso de resolutores SMT como oráculo para algunas de las subteorías de la teoría ecuacional subyacente a la teoría de reescritura considerada. Finalmente, el cuarto cálculo usa un lenguaje de estrategias, junto con resolutores SMT, para podar más el espacio de estados, además de permitir la parametrización tanto de las teorías de reescritura como de las estrategias definidas para un problema de alcanzabilidad dado.

Contents

List of Figures

Chapter 1

Introduction

The task proposed at first by my Ph.D. advisors for this dissertation was to investigate conditional narrowing in the widest possible way. The only limitation on this imposed goal was that although the investigation would be mainly theoretical, we would use the resources provided by version 2 of the Maude engine for our prototypes, where reflection for functional and system modules, and metaunification where the main tools at our disposal. The years that have passed since this task was accepted until these lines are finally being written, allowed the development of Maude 3 including, among many others, two features that attracted our interest: the strategies language and the support for SMT solvers, very important additions for our goal since narrowing cannot perform inductive proving. Thus, this dissertation has two very different parts: while the first two calculi explore different approaches to narrowing, based on the features available in Maude 2, the other two calculi focus on the aforementioned two new features added in Maude 3 and how to improve narrowing with them.

Due to their increasing complexity, the amount of time devoted to develop the framework needed for each of the last three calculi grew significantly, with the Appendix of the last published technical report taking more than 50% of the pages in that report, mainly due to the proofs of the results required for the new framework.

We review in the next sections the main characteristics of the calculi presented in this dissertation.

1.1 Conditional rewriting modulo

A rewrite theory [Mes92] has the form $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, with Σ a signature that tells us how to construct valid terms and $\mathcal{E}$ a set of equations. We call the pair $(\Sigma, \mathcal{E})$ an equational theory, that defines an equivalence relation between valid terms (an abstract data type). R is a set of rules, each one of the form $l \to r$ *if* C (C being some condition), that defines when a term t_1 may evolve to another term t_2 (written $t_1 \to^1_R t_2$ and called a rewrite step), which is the simplest case of rewrite relation. We can also define a rewrite relation modulo $\mathcal{E}$ where t_1 may evolve to t_2 (written $t_1 \to^1_{R/\mathcal{E}} t_2$) if there are terms t'_1 and t'_2 in the equivalence classes modulo $\mathcal{E}$ of t_1 and t_2, respectively, such that $t'_1 \to^1_R t'_2$ (in case that the rule either has no conditions or all of its conditions are related to $\mathcal{E}$, i.e., equational. The general case requires a more complex rewrite relation that will be defined).

While conditional rewriting is based on matching as the way to generate instances of the rules to apply, conditional rewriting modulo is based on matching modulo $\mathcal{E}$, also called $\mathcal{E}$-matching, which requires a specific algorithm for each equational theory.

1.2 Conditional narrowing modulo

Conditional rewriting modulo can solve reachability problems, see Section 1.3, when the initial term t is not going to be instantiated, because it has no variables (a ground term) or because although it has variables, we consider that t is a *pattern* that represents all the ground instances of t.

Narrowing [Fay79], that was first proposed as a method for solving equational goals (*unification*), also allows us to know if there are any instances of the initial and the final symbolic terms of a reachability problem, say t and t', such that t' is *reachable* from t, see Section 2.3.10. While in conditional rewriting modulo we only instantiate a rule with $\mathcal{E}$-matching in order to apply it to t, in conditional narrowing modulo (CNM in short) we instantiate both t and the rule, using a so called $\mathcal{E}$-unification algorithm which, in general, is far more complex that the corresponding $\mathcal{E}$-matching algorithm for any given equational theory $\mathcal{E}$.

1.3 Reachability problems

A reachability problem can have the form $\exists \bar{x}(t(\bar{x}) \rightarrow^* t'(\bar{x}))$, with t, t' terms with variables in $\bar{x}$, or be a conjunction $\exists \bar{x} \bigwedge_i (t_i(\bar{x}) \rightarrow^* t'_i(\bar{x}))$. Reachability problems can be solved by model-checking methods for finite state spaces. As already stated, when the initial term t has no variables, i.e., it is a ground term, and under certain admissibility conditions, rewriting can be used in a breadth-first way to traverse the state space, trying to find a suitable matching of $t'(\bar{x})$ in each traversed node. In the general case where $t(\bar{x})$ is not a ground term, narrowing has been extended to cover also reachability problems [MT07], leaving equational goals as a special case. Nowadays, narrowing is used to inspect complex concurrent and deductive systems [Mes12].

When narrowing is used to solve reachability problems, it can also be used at another level for finding the solution to the equational goals that may appear in the conditions of rules and equations. Specific unification algorithms exist for a small number of equational theories, but if $(\Sigma, \mathcal{E})$ is an equational theory, $\mathcal{E}$ has the form $E \cup B$, where B is a set of axioms having a complete unification algorithm, and the sentences in E can be turned into a set of rules $\overrightarrow{E}$, by orienting them, such that the rewrite theory $\mathcal{R} = (\Sigma, B, \overrightarrow{E})$ complies with certain restrictions, then narrowing can be used on $\mathcal{R}$ to solve $\mathcal{E}$-unification goals.

One of the weaknesses of narrowing is the state space explosion associated to any reachability problem where arithmetic equational theories are involved, because narrowing cannot perform inductive proving. The inclusion in the last two narrowing calculi of this dissertation of *Satisfiability modulo theories* (SMT) solvers [dMB08], an extension of *Boolean satisfiability* (SAT) solvers that can handle a wide variety of equational theories, including integer and real numbers, helped mitigating the aforementioned state space explosion. Other known sources of state space explosion are: (i) the order of application of the rules and (ii) the application of unneeded rules. These two problems have been addressed in the last narrowing calculi of this dissertation with the use of strategies.

1.4 Seminal and related work

In this section we discuss the previous work that set the foundations for our research, together with some related work that has been developed while this research took place.

In the beginning, there were the Knuth-Bendix completion algorithm [KB70], which were capable of solving certain word problems, and the paramodulation rule [RW69], which was used to generate all "equal" versions of clauses, modulo conditions on the equality information. After the advent of canonical term rewriting systems, Slagle [Sla74] formulated narrowing as an adaptation of the paramodulation rule that could be used for showing equality in theorem proving.

Then came Fay [Fay79], who proved that narrowing could also be used as a universal unification procedure to solve equations inside the theory defined by any canonical term rewriting system. As the proposed algorithm was too costly in time and space, the search for strategies that could be used to make the use of narrowing feasible began.

Fay's work was extended by Hullot [Hul80], where he proved that any normalized solution to a reachability problem could be lifted to a narrowing derivation that computed a more general solution, using a narrowing strategy that he called *basic narrowing*, and showing the intimate relationship between rewriting and reachability problems. Later on, Jouannaud et al. [JKK83] generalized these previous results to the case of term rewriting systems where the set of axioms could be split into a set of rules R and a set of equations E, such that R was E-confluent and E-coherent, as defined in [Jou83], without any linearity hypothesis.

The specification of algebraic data types was one of the causes of the shift towards the study of conditional rewriting systems [Kap85]. When Goguen and Meseguer [GM86b] showed that conditional rewriting systems provide a natural computational paradigm combining logic and functional programming, this shift was enforced and there was also an increase in the investigation done about E-unification, e.g., Gallier and Snyder [GS87], and narrowing. Bosco et al. [BGM87], present a refined strategy to obtain a complete E-unification algorithm for a certain class of canonical theories. Also, Dershowitz et al. [DOS88], study the use of a "decreasing" ordering that makes decidable rewriting and the computation of normal forms.

Meseguer and Winkler [MW91] present the Maude system, which is based on a simple logic of action called rewriting logic [Mes90], as a framework not only for programming, but also for specification.

The use of narrowing strategies caused some concern about when it was secure to use some narrowing strategy. Echahed [Ech90] addresses the problem of unification modulo a set of equations using the narrowing relation, and proposes some syntactical criteria on algebraic specifications that ensure the completeness of narrowing strategies

Another classic reference is the work of Bockmayr [Boc93], where he develops equational conditional narrowing modulo a set of conditional equations and proves its correctness and completeness for equational conditional rewrite systems R, E without extra variables, where E is regular and R, E is Church-Rosser modulo E and decreasing modulo E. A result that can be seen as the theoretical foundation of a special form of constraint logic and functional programming.

The grounds upon where effective narrowing was established were deeply shaken when Middeldorp and Hamoen [MH94] proved that basic narrowing was not complete with respect to normalizable solutions for equational theories defined by confluent term rewriting systems, contrary to what had been conjectured, imposed syntactic restrictions on the

rewrite rules that allowed the recovery of completeness, and refuted a result of Höll-dobler [Höl89] which states the completeness of basic conditional narrowing for complete (i.e., confluent and terminating) conditional term rewriting systems without extra variables in the conditions of the rewrite rules. A gear had to be stepped up in the search for narrowing strategies.

The idea of constraint solving by narrowing in combined algebraic domains was presented by Kirchner and Ringeissen [KR94], where the supported theories had unconstrained equalities and the rewrite rules had constraints from an algebraic built-in structure, but they did not allow for reachability problems.

Antoy et al. [AEH94] present a new *needed narrowing* strategy, which is the base of the *Curry* language [HKMN95,Han97], that is optimal in several respects, using a notion of a needed narrowing step that, for inductively sequential rewrite systems, extends the Huet and Lévy notion of a needed reduction step [HL91]. They define a strategy, based on this notion, that computes only needed narrowing steps. The strategy is sound and complete for a large class of rewrite systems, is optimal with respect to the cost measure that counts the number of distinct steps of a derivation, computes only incomparable and disjoint unifiers, and is efficiently implemented by pattern matching.

In [MOI96], Middeldorp et al. show that one narrowing calculus, which they call *lazy narrowing* (LNC for short), lacks strong completeness, contrary to what had been stated in the literature, so selection functions to cut down the search space are not applicable, prove completeness of the calculus, and also prove that LNC is strongly complete whenever basic narrowing is complete. They also present another strategy called *eager variable elimination* and prove its completeness in the case of *orthogonal* term rewriting systems.

Denker, Meseguer, and Talcott [DMT98] propose rewriting logic as an executable specification formalism for security protocols and discuss, for the first time, the possibility of implementing narrowing in Maude as a general mechanism for solving reachability goals using symbolic execution techniques. Some years later, Meseguer and Thati [MT04, MT07] present a generalization of narrowing which can be used to solve reachability goals in initial and free models of a rewrite theory.

In [Ham00], Hamada proves that conditional LNC (LCNC for short) is strong complete for terminating and level-confluent conditional term rewriting systems and LCNC is complete for level-complete conditional rewrite systems, in both results without assuming any restrictions on the extra variables in the conditional rewrite systems.

At the same time, Antoy et al. [AEH00] present an improved version of their needed narrowing strategy that uses unification instead of pattern matching. Santiago Escobar proposes in his Ph.D. dissertation [Esc04] two narrowing strategies: incremental needed narrowing and natural narrowing.

Conditional narrowing without axioms for equational theories with an order-sorted type structure has been thoroughly studied for increasingly complex categories of term rewriting systems. A wide survey can be found in [MH94]. The literature is scarce when we allow for extra variables in conditions (e.g., [GM86a], [Ham00]), or conditional narrowing modulo a set of equations or axioms (e.g., [Boc93], [CEM15]).

For equational goals the idea of *variants of a term* has been applied in recent years to narrowing [EMS08]. A strategy known as *folding variant narrowing* [ESM12], which computes a complete set of variants of any term, has been developed by Escobar, Sasse, and Meseguer, allowing unification modulo a set of unconditional equations plus axioms. The strategy terminates on any input term on those systems enjoying the *finite variant property* (FVP), and it is optimally terminating. It is being used for cryptographic

protocol analysis [MT07] with tools like Maude-NPA [EMM09], termination algorithms modulo axioms [DLM+08], algorithms for checking confluence and coherence of rewrite theories modulo axioms [DM12], and infinite-state model checking [BM14].

Foundations for order-sorted conditional rewriting have been published by Meseguer [Mes17]. Cholewa, Escobar, and Meseguer [CEM15] have defined a new hierarchical method, called layered constraint narrowing, to solve narrowing problems in order-sorted conditional equational theories, an approach similar to ours, and given new theoretical results on that matter, including the definition of constrained variants for order-sorted conditional rewrite theories, but with no specific support for SMT solvers.

Recent development in conditional narrowing has been made for order-sorted equational theories [CEM15] and also for rewriting with constraint solvers [RMM17].

In [EM19], Escobar and Meseguer present canonical constrained narrowing, a symbolic reachability analysis technique applicable to topmost rewrite theories where the equational theory has the finite variant property. This work has been recently extended by López-Rueda and Escobar [LE22] to handle conditional rules with SMT constraints.

In [Mes20], Meseguer studies reachability in Generalized Rewrite Theories, that include constructors and variants, where frozenness is used as a type of strategy.

In [Mes23], Meseguer investigates the notions of FVP, variant unification, and variant satisfiability when the unification algorithm is infinitary, and generalizes FVP theories to a bigger class of *boundedness property* (BP) theories under several assumptions.

The strategy language that we have proved suitable for our narrowing calculus in this dissertation is a subset of the Maude strategy language [MOMV04,EMOMV07,RMPV21]. This strategy language and a connection with SMT solvers that have been incorporated into the Maude 3 engine [DEE+20] were used to develop our prototype for the calculus with strategies.

1.5 Contributions of this work

The goal of this work is to study the relationship between verifiable and computable answers to reachability problems in rewrite theories with an underlying membership equational logic. The main contributions to that end of this dissertation are:

(i) A narrowing calculus [AMPP14], basically developed as a proof that conditional narrowing modulo membership equational theories is feasible, where the space state is pruned by checking the sort of the terms involved in each narrowing step.

(ii) The definition of two new concepts, *fresh pattern property* and *narrowable rewrite theory*, and the development of a narrowing calculus for these definitions, that accepts a larger class of rewrite theories, with respect to (i), and adds more pruning solving reachability problems.

Two versions of this calculus have been published [AMPP15, AMPP18].

(iii) The development of a narrowing calculus [AMPP17] for conditional narrowing modulo axioms plus SMT equational theories, i.e, it allows not only properties like associativity, commutativity, etc., but also equational theories that can be handled by an SMT solver, like the ones for integers or reals.

The conditions that may appear in the rules of the admitted rewrite theories for this calculus are either rewrite conditions or quantifier-free SMT formulas, with no

restriction regarding the variables that appear in these rules or in the reachability problems.

(iv) The development of a narrowing calculus [AMPP23] that extends the previous one by adding (i) strategies, for explicit control of the rewrite steps that can be applied, and (ii) parameters, for an enhanced expressivity of the admitted reachability problems.

All the calculi presented in this dissertation have been proved sound and weakly complete, i.e., complete with respect to idempotent $R/\mathcal{E}$-normalized answers.

1.6 Structure of the thesis

As already explained at the beginning of this chapter, this dissertation has two main parts. The first part has two chapters, each one showing a narrowing calculus *without* SMT solvers; the second part also has two chapters, each one showing a narrowing calculus *with* SMT solvers. The related proofs are included at the end of each of these chapters. Two leading chapters precede the main ones:

- this chapter, where it is shown the motivation for this work, followed by a brief introduction of the main elements at play through all the dissertation, a section discussing seminal and related work, and this section;

- a chapter with all the needed background for the rest of the dissertation. The chapter mainly contains already existing concepts and results, citations are used to acknowledge where they have been taken from, but some definitions and results for narrowing with SMT solvers developed for this dissertation, have been moved to this chapter because they are shared by both chapters on narrowing with SMT solvers.

The dissertation ends with a chapter that contains some reflections that I wanted to put on paper as my takeout from all the years that I have been working on it, and the bibliography.

Chapter 2 presents all the topics required to understand the contributions of this dissertation: membership equational logic, rewriting logic, narrowing, and strategies. They are discussed in great detail, since these details are pivotal in the development of the narrowing calculi of the dissertation. The last section of the chapter introduces the Maude language and the three types of Maude modules that are used in this work: functional modules, system modules, and strategy modules. Maude is used on most of the examples in this work and also on all the prototypes that have been developed to test the different calculi.

Chapter 3 presents a narrowing calculus that proves the feasibility of conditional narrowing modulo. First, the part of the calculus that can be used for equational unification is presented and its correctness properties with respect to equational unification are stated and proved. Then, the full calculus for reachability is shown and its correctness properties with respect to narrowing for reachability are stated and proved.

Chapter 4 presents a narrowing calculus that addresses one of the sources of state explosion in the first calculus: the fact that unification and reachability steps can be interleaved in many ways in the narrowing paths generated by the calculus. Larger classes

of rewrite theories and reachability goals are accepted by the calculus with respect to the first calculus. Again, the narrowing calculus is shown in two phases, the part related with equational unification and the full calculus for reachability, and correctness properties for each phase are stated and proved.

Chapter 5 presents the first narrowing calculus with SMT solvers of the dissertation. It has been developed for order-sorted rewrite theories where the conditions that appear in their rules are either rewrite conditions or quantifier-free SMT formulas. Again, its correctness with respect to narrowing for reachability problems are stated and proved. Finally, there is a section discussing two different approaches taken to develop prototypes of the calculus and a comparison of their performance when different improvements were added to them.

Chapter 6 extends the previous one with the addition of two elements: *strategies* for explicit control of the rewrite steps that can be applied and *parameters* for an enhanced expressivity of the reachability problems. A narrowing calculus that uses both elements is shown in this chapter and its correctness properties with respect to narrowing for reachability are stated and proved. A prototype for narrowing with strategies using this calculus has also been developed.

This dissertation ends with Chapter 7 where I have written down not only the technical conclusions of this work and some guidelines for future work, but also a few personal reflections on what it takes to obtain a Ph.D.

The prototypes developed for this dissertation, with examples of their use, and other related material can be accessed through the homepage of this thesis, https://maude.ucm.es/cnarrowing.

Chapter 2

Background

This chapter is the densest one of this dissertation. It explains, with great detail, the main frameworks upon which the work of all these years stands. As a suggestion to the reader, it is possible to flick through the included sections, and come back to any of them for a deeper reading when any of the contents in the rest of this dissertation demands so.

To make this upcoming avalanche of concepts easier to understand, a failing vending machine will be used as a motivating example and to explain some of the definitions in a less abstract way. This machine accepts a `Coin` (either a quarter `q` or a dollar `$`) that may be inserted at any time and serves one `Item` if there is enough credit: an apple `a` at a price of one dollar or a coffee `c` at a price of three quarters. The vending machine is misbehaving: in order to serve anything, there must be a credit of at least one dollar; then the machine may serve either a coffee or an apple, nondeterministically, and the remaining credit is updated accordingly. The vending machine knows that four quarters make a dollar. The vending machine has a `State` which is a nonempty multiset of `Coins` and `Items` (the initial `State` may not be empty). The `State` tells us the credit and the `Items` that have already been served. A single `Coin` or `Item` is a `State`. `States` are written as a mere juxtaposition of `Coins` and `Items`, i.e., we admit the use of empty notation for union.

2.1 Specifications and strategies

A system is specified in rewriting logic as a rewrite theory with an underlying equational theory where terms, that usually represent the state of the system, are given an algebraic data type. The rewrite theory has a set of rules that specify in which ways the system can evolve from the current state to the next one.

Strategies allow modular separation between the rules that specify a system and the way that these rules are applied, restricting the reachable states from an initial state. They can be used both to implement and test different algorithms over a given specification or to drive the search of solutions to reachability problems.

2.2 Membership equational logic

Membership equational logic is presented through a sugared version, similar to the one that is used for defining Maude specifications. Let $(S, \leq)$ be a partially ordered set of

sorts, whose *connected components* are the equivalence classes corresponding to the least equivalence relation $\equiv_{\leq}$ containing $\leq$.

2.2.1 Membership equational logic signature

A *membership equational logic* (MEL) *signature* [BM06] is defined by a *kind-complete* tuple $\Sigma = (K, S, \leq, F)$ meaning that:

- K is a set of *kinds*, where $K \cap S = \emptyset$.
- S is split into a K-kinded family of disjoint sets of sorts S_k, i.e., $S = \bigcup_{k \in K} S_k$, such that if $s_i \leq s_j$ and $s_i \in S_k$ then $s_j \in S_k$. We write $[s_i] = k$ and say that the kind of s_i is k, i.e., each sort in a connected component of $(S, \leq)$ has the same kind. $\leq$ is extended so that $s_i \leq k$ iff $s_i \in S_k$, i.e., k is the top sort of its connected component (we also define $[k] = k$ if $k \in K$ for simplicity of notation).
- $F = \{\Sigma_{\bar{\kappa},\kappa}\}_{(\bar{\kappa},\kappa) \in (K \cup S)^* \times (K \cup S)}$ is an algebraic signature of *function symbols* where for each symbol $f \in \Sigma_{\kappa_1 \ldots \kappa_n, \kappa}$ if $n \geq 1$ and at least one of the subscripts is not a kind, implicitly there exists another function symbol at the kind level $f \in \Sigma_{[\kappa_1] \ldots [\kappa_n], [\kappa]}$. When $f \in \Sigma_{\epsilon,\kappa}$ (ϵ being the empty word), we say that f is a *constant* with *type* (meaning sort or kind) κ. We write $f \in \Sigma_\kappa$ instead of $f \in \Sigma_{\epsilon,\kappa}$.

The MEL *signature* for our vending machine has only one kind, $K = \{[\texttt{State}]\}$ (we write `St` as a shortcut), with three sorts, $S_{[St]} = \{\texttt{State}, \texttt{Coin}, \texttt{Item}\}$; $S = \{S_{[\texttt{St}]}\}$; $F = \{\Sigma_{\texttt{Coin}}, \Sigma_{\texttt{Item}}, \Sigma_{\texttt{St St,St}}\}$ with $\Sigma_{\texttt{Coin}} = \{q, \$\}$, $\Sigma_{\texttt{Item}} = \{a, c\}$, and $\Sigma_{\texttt{St St,St}} = \{\cdot_{\texttt{St St,St}}\}$. The functions `q`, `$`, `a`, and `c` are the constants of F. The function $\cdot$ (understood as juxtaposition) returns an element with sort `State` given any pair of elements with sort `State`.

If $f \in \Sigma_{\kappa_1 \ldots \kappa_n, \kappa}$, then we write $f : \kappa_1 \ldots \kappa_n \to \kappa$, and say that f has *arity* n and *codomain* κ. We call this a *rank* declaration for symbol f. Constant symbols have only one rank declaration $f : \to \kappa$ (plus the mandatory $f : \to [\kappa]$ if κ is not a kind). We extend the order $\leq$ on $K \cup S$ to $(K \cup S)^*$, component-wise, where we use the letters w, w' as synonyms for the elements $\kappa_1 \ldots \kappa_n, \kappa'_1 \ldots \kappa'_n \in (K \cup S)^*$ respectively. Then F must also satisfy a *monotonicity condition*: $f \in \Sigma_{w,\kappa} \bigcap \Sigma_{w',\kappa'}$ and $w \leq w'$ imply $\kappa \leq \kappa'$. If $f \in \Sigma_{w,\kappa}$ and $t_1, \ldots, t_n$ have types $\kappa_1, \ldots, \kappa_n$ respectively, then the term $f(t_1, \ldots, t_n)$ (we will also write $f(\bar{t})$) has type κ. If $\kappa \leq \kappa'$ and the term t has type κ, then t has also type κ'. This means that a term may have several types. In fact, as for every sort s we have that $s \leq [s]$, if a term has only one type then it must be a kind. For simplicity, we will only allow overloading of functions when their codomains have the same kind.

Σ-algebras

A MEL Σ-*algebra* $\mathcal{A}$ contains a set $\mathcal{A}_k$ for each kind $k \in K$, an n-ary function $\mathcal{A}_f : \mathcal{A}_{\kappa_1} \ldots \mathcal{A}_{\kappa_n} \to \mathcal{A}_\kappa$ for each function $f \in \Sigma_{w,\kappa}$, and a subset $\mathcal{A}_s \subseteq \mathcal{A}_k$ for each sort $s \in S_k$ such that if $s_i \leq s_j$ then $\mathcal{A}_{s_i} \subseteq \mathcal{A}_{s_j}$, and if $f \in \Sigma_{w,\kappa} \bigcap \Sigma_{w',\kappa'}$ and $w \leq w'$, where $w = \kappa_1 \ldots \kappa_n$ and $w' = \kappa'_1 \ldots \kappa'_n$, then $\mathcal{A}_f : \mathcal{A}_{\kappa_1} \ldots \mathcal{A}_{\kappa_n} \to \mathcal{A}_\kappa$ equals $\mathcal{A}_f : \mathcal{A}_{\kappa'_1} \ldots \mathcal{A}_{\kappa'_n} \to \mathcal{A}_{\kappa'}$ on $\mathcal{A}_{\kappa_1} \ldots \mathcal{A}_{\kappa_n}$.

In membership equational logic the elements in a sort are well-defined, while the elements in a kind that don't belong to any sort are usually meant to refer to error or undefined elements. Kinds also provide a general way of dealing with partial functions in

equational specifications: a partial function f with codomain s can be defined as a total function with codomain $[s]$. If $f(\bar{t})$ is undefined when f is seen as a partial function, then $f(\bar{t})$ has type $[s]$ when f is seen as a total function.

We allow *mix-fix* notation in F, where the symbol _ is used to identify the position of each $\kappa_i \in \bar{\kappa}$. If omitted, we assume the usual functional notation $f(\kappa_1, \ldots, \kappa_n)$, which is an alternative notation admitted for all functions. Then, if we have a sort s with constants a and b, and a rank declaration $\_f\_ : s\ s \to s$, we can write either $a\ f\ b$ or $f(a, b)$.

Variables

We assume a family $\mathcal{X} = \{\mathcal{X}_\kappa\}_{\kappa \in (K \cup S)}$ of infinite sets of *variables*, such that $\kappa \neq \kappa'$ implies $\mathcal{X}_\kappa \cap \mathcal{X}'_\kappa = \emptyset$. If κ is a sort then x_κ has sort κ (and kind $[\kappa]$), otherwise x_κ has kind κ but no sort. The type of a variable can be omitted if it can be inferred or it is not relevant. Each set of variables is infinite, but any computation will only require a finite number of variables. The type of a variable can be omitted when it is not relevant. A term that has no variables in it is said to be *ground*. A term where each variable occurs only once is said to be *linear* (ground terms are linear).

The sets $T_{\Sigma,\kappa}$, $T_\Sigma(\mathcal{X})_\kappa$ denote, respectively, the set of ground Σ-terms with sort or kind κ and the set of Σ-terms with sort or kind κ over $\mathcal{X}$. We ambiguously use the notation T_Σ to refer to the initial Σ-algebra and as a shortcut for $\bigcup_{\kappa \in (K \cup S)} T_{\Sigma,\kappa}$. We also ambiguously use the notation $T_\Sigma(\mathcal{X})$ to refer to the free Σ-algebra on $\mathcal{X}$ and as a shortcut for $\bigcup_{\kappa \in (K \cup S)} T_\Sigma(\mathcal{X})_\kappa$. We write $vars(t)$, or V_t, to denote the set of variables in a term t from $T_\Sigma(\mathcal{X})$. Σ is assumed to be *sensible* meaning that if $f \in \Sigma_{\kappa_1 \ldots \kappa_n, \kappa}$, $f \in \Sigma_{\kappa'_1 \ldots \kappa'_n, \kappa'}$ and $[\kappa_i] = [\kappa'_i]$ for $i = 1, \ldots, n$ then $[\kappa] = [\kappa']$. We also assume that Σ has non-empty sorts, i.e., $T_{\Sigma,s} \neq \emptyset$ for all $s \in S$.

In our vending machine, $T_{\Sigma,\texttt{Coin}} = \{\texttt{q}, \$\}$, $T_{\Sigma,\texttt{Item}} = \{\texttt{a}, \texttt{c}\}$, and $T_{\Sigma,\texttt{State}}$ contains, aside from all the constants (q, $, a, and c), any finite concatenation (the only non-constant function) of these constants, e.g., $\texttt{q}\ \texttt{q} \in T_{\Sigma,\texttt{State}}$.

Positions

When a term t is expressed in functional notation as $f(t_1, \ldots, t_n)$, it can be recursively pictured as a tree with *root* f and *tree children* t_i at *position* i, for $1 \leq i \leq n$. Then the root position of t is referred as ϵ and the inner positions of t are referred as lists of nonzero natural numbers separated by dots, $i_1.i_2.\cdots.i_m$, meaning the position $i_2.\cdots.i_m$ of t_{i_1}, where $1 \leq i_1 \leq n$. The set of positions of a term is written $pos(t)$. The set of non-variable positions of a term whose root is a function symbol in Σ is written $pos_\Sigma(t)$. The set of positions of variables from $\mathcal{X}$ in a term is written $pos_\mathcal{X}(t)$. $t|_p$ is the subtree of t below position p. $t[u]_p$ is the replacement in t of the subterm at position p with a term u. $t[\,]_p$ is a *term with hole* that is equal to t except that in the position p there is a special symbol $[\,]$, the hole. As an example, if t is $f(g(a, b), c)$, then $t|_1$ is $g(a, b)$, $t|_{1.2}$ is b, $t[\,]_{1.2}$ is $f(g(a, [\,]), c)$, and $t[d]_{1.2}$ is $f(g(a, d), c)$. For any position p define $p.\epsilon = p$. For positions p and q, we write $p \leq q$ if there is a position r such that $q = p.r$, and write $p < q$ if $q = p.r$ and $r \neq \epsilon$. Trivially $p \leq p$ because $p = p.\epsilon$. $t[u_1, \ldots, u_n]_{p_1 \ldots p_n}$ is the replacement in t of the subterms at the *unique* positions $p_1, \ldots, p_n$ with the terms $u_1, \ldots, u_n$, respectively, where for all $1 \leq i, j \leq n$ if $i \neq j$ then $p_i \not\leq p_j$. We also write $t[\bar{u}]_{\bar{p}}$ if the *ordered lists* $\bar{u} = u_1, \ldots, u_n$ and $\bar{p} = p_1, \ldots, p_n$ are known from the context. We extend the definition of hole to ordered lists and the definition of replacement to pairs

of ordered lists having the same number of elements: $t[]_{\bar{p}} = t[]_{p_1} \ldots []_{p_n}$ is the same term as t but with holes in the positions in $\bar{p}$; if $\bar{v} = v_1, \ldots, v_n$ and $\bar{q} = q_1, \ldots, q_n$, then $\bar{u}[\bar{v}]_{\bar{q}}$ is $u_1[v_1]_{q_1}, \ldots, u_n[v_n]_{q_n}$, i.e., the same ordered list except that the subterm at position q_i of each term u_i is replaced with v_i. Then, $t[\bar{u}[\bar{v}]_{\bar{q}}]_{\bar{p}} = t[u_1[v_1]_{q_1}]_{p_1} \ldots [u_n[v_n]_{q_n}]_{p_n}$. Given any ordered list $\bar{u}$, which may have repetitions, we denote by $\hat{u}$ the set of elements of $\bar{u}$. If $\bar{p} = p_1, \ldots, p_n$ and $\hat{p} \subseteq pos(t)$ then $t|_{\bar{p}} = t|_{p_1}, \ldots, t|_{p_n}$ and $t|_{\hat{p}} = \{t|_{p_1}, \ldots, t|_{p_n}\}$. $vars(t[]_{\bar{p}})$ is the set of variables appearing in the term with holes $t[]_{\bar{p}}$. We also allow the use of holes and replacement in tuples, if $T = (t_1, \ldots, t_n)$ then $T|_1 = t_1$, $T[x]_1 = (x, t_2, \ldots, t_n)$, etc.

Preregularity

Given an order-sorted signature Σ, for each natural number n, for every function symbol f in Σ with arity n, and for every tuple $(s_1, \ldots, s_n)$ in S^n, let $S_{f,s_1\ldots,s_n}$ be the set containing all the sorts s' that appear in rank declarations in Σ of the form $f : s'_1 \ldots s'_n \to s'$ such that $s_i \leq s'_i$, for $1 \leq i \leq n$. If whenever $S_{f,s_1,\ldots,s_n}$ is not empty (so a term $f(t_1, \ldots, t_n)$ where t_i has type s_i for $1 \leq i \leq n$ would be a Σ-term), it is the case that $S_{f,s_1,\ldots,s_n}$ has a least sort, then Σ is said to be *preregular*.

Preregularity guarantees that every Σ-term t has a *least sort*, denoted $ls(t)$, among all the sorts that t has because of the different rank declarations that can be applied to t, which is the most accurate classification for t, i.e., for any rank declaration $f : s_1 \ldots s_n \to s$ that can be applied to t it is true that $ls(t) \leq s$.

Substitutions

A *substitution* $\sigma : \mathcal{X} \to \mathcal{B}$, where $\mathcal{B} \subseteq \mathcal{T}_\Sigma(\mathcal{X})$, is a function that matches the identity function in all $\mathcal{X}$ except for a finite set of variables called its *domain*, denoted $dom(\sigma)$. If $\mathcal{B} \subseteq \mathcal{T}_\Sigma$ then the substitution is *ground*. Substitutions are written as $\sigma = \{y^1_{s_1} \mapsto t_1, \cdots, y^n_{s_n} \mapsto t_n\}$, where $dom(\sigma)$ is $\{y^1_{s_1}, \ldots, y^n_{s_n}\}$ and the *range* of σ is $ran(\sigma) = \bigcup_{i=1}^n vars(t_i)$. We will write $\sigma = \{\bar{y} \mapsto \bar{t}\}$ as a shorthand if both $\bar{y}$ and $\bar{t}$ are known. We write $\sigma : \mathcal{D} \to \mathcal{B}$, where $\mathcal{D}$ is a finite set of variables, to imply that $dom(\sigma) = \mathcal{D}$. We represent the application of a substitution σ to a variable x in $\mathcal{X}$ as $x\sigma$. The substitution instance $t\sigma$ of a term t is a term obtained from t by *simultaneously* replacing any occurrence of each variable $x \in Dom(\sigma)$ that appears in t with $x\sigma$. In our vending machine, if $t = \mathtt{q}\ x_{\mathtt{Item}}$ and $\sigma = \{x_{\mathtt{Item}} \mapsto \mathtt{c}\}$ then $t\sigma = \mathtt{q\ c}$. A substitution σ is *well-formed* if $ls(y_s\sigma) \leq s$ for each variable y_s in $dom(\sigma)$. It is assumed throughout that all substitutions are well-formed. The identity substitution is displayed as *none*. A substitution σ where $dom(\sigma) = \{x^1_{s_1}, \ldots, x^n_{s_n}\}$ $(n \geq 0)$, $x^i_{s_i}\sigma = y^i_{s_i} \in \mathcal{X}$, for $1 \leq i \leq n$, and $y^i_{s_i} \neq y^j_{s_j}$ for $1 \leq i < j \leq n$ is called a *renaming*, with *inverse* $\sigma^{-1} = \{y^i_{s_i} \mapsto x^i_{s_i}\}_{i=1}^n$, being *none* the trivial renaming. The restriction $\sigma_\mathcal{V}$ of σ to a set of variables $\mathcal{V}$ is defined as $x\sigma_\mathcal{V} = x\sigma$ if $x \in \mathcal{V}$ and $x\sigma_\mathcal{V} = x$ otherwise. The deletion $\sigma_{\setminus\mathcal{V}}$ of a set of variables $\mathcal{V}$ from σ is defined as $x\sigma_{\setminus\mathcal{V}} = x\sigma$ if $x \in dom(\sigma) \setminus \mathcal{V}$ and $x\sigma_{\setminus\mathcal{V}} = x$ otherwise. Substitutions are homomorphically extended to terms in $\mathcal{T}_\Sigma(\mathcal{X})$ and also to any other syntactic structures unless explicitly stated. The *composition* of two substitutions σ and σ' is denoted by $\sigma\sigma'$, with $x(\sigma\sigma') = (x\sigma)\sigma'$ (left associativity). Their *closed composition*, denoted by $\sigma{\cdot}\sigma'$, is defined as $\sigma{\cdot}\sigma' = (\sigma\sigma')_{\setminus ran(\sigma)}$ (then, if $\sigma = \{x \mapsto y\}$ and $\sigma' = \{y \mapsto z\}$, $\sigma\sigma' = \{x \mapsto z, y \mapsto z\}$ and $\sigma{\cdot}\sigma' = \{x \mapsto z\}$). For a substitution σ, if $\sigma\sigma = \sigma$ we say that σ is *idempotent*. It is assumed throughout that all substitutions are idempotent, usually because $dom(\sigma) \cap ran(\sigma) = \emptyset$. For substitutions σ and σ', where $dom(\sigma) \cap dom(\sigma') = \emptyset$

and $(dom(\sigma) \cup dom(\sigma')) \cap (ran(\sigma) \cup ran(\sigma')) = \emptyset$, we denote their union by $\sigma \cup \sigma'$. A *context* $\mathcal{C}$ is a λ-term of the form $\lambda x^1_{s_1} \cdots x^n_{s_n}.t$, with $t \in \mathcal{T}_\Sigma(\mathcal{X})$ and $\{x^1_{s_1}, \ldots, x^n_{s_n}\} \subseteq vars(t)$.

2.2.2 MEL theory

A Σ-*equation* is an expression of the form $t = t'$. A Σ-*equation* $t = t'$ is said to be:

- *Regular* iff $vars(t) = vars(t')$.
- *Sort-preserving* iff for each substitution σ, we have $t\sigma \in T_\Sigma(\mathcal{X})_\kappa$ ($\kappa \in K \cup S$) implies $t'\sigma \in T_\Sigma(\mathcal{X})_\kappa$ and vice versa.
- *Left (or right) linear* iff t (resp., t') is linear.
- *Linear* iff it is both left and right linear.

A set of Σ-equations is said to be regular, or sort-preserving, or (left or right) linear, if each equation in it is so.

A MEL theory [BM06] is a pair $(\Sigma, \mathcal{E})$, where Σ is a MEL signature and $\mathcal{E}$ is a finite set of MEL sentences (Horn clauses), either conditional equations or conditional memberships of the forms:

$$t = t' \text{ if } A_1 \wedge \ldots \wedge A_n, \qquad t : s \text{ if } A_1 \wedge \ldots \wedge A_n,$$

where $t = t'$ is a Σ-equation, $t : s$, $s \in S$, is a unary membership predicate stating that t is a term with sort s, provided that the condition holds, and each A_i can be of the form $t = t'$, $t : s$, or $t := t'$ (a *matching* equation).

All the variables appearing in a MEL sentence are interpreted as universally quantified. Matching equations are treated as ordinary Σ-equations. They are a warning that new *extra* variables appear in t, in a concrete way, imposing a limitation in the syntax of the equation. We also admit unconditional sentences in $\mathcal{E}$. $x_{\kappa_1} : \kappa_2$ is an unconditional membership expressing $\kappa_1 \leq \kappa_2$. For each variable $x_\kappa \in \mathcal{X}_\kappa$, where $\kappa \in S \cup K$, we have that $x_\kappa : \kappa \in \mathcal{E}$.

As an exception, there are two types of unconditional memberships over kinds, instead of sorts, that are implied by a MEL signature: if $f \in \Sigma_{\kappa_1 \ldots \kappa_n, k}$, $k \in K$ then $f(x_{\kappa_1}, \ldots, x_{\kappa_n}) : k \in \mathcal{E}$; also, for each variable $x_\kappa \in \mathcal{X}_\kappa$ such that $[\kappa] = k$, $x_\kappa : k \in \mathcal{E}$.

Throughout this dissertation we assume that all signatures are preregular and all their equations and memberships $t = t'$, $t := t'$, and $t : s$, respectively satisfy the conditions $[ls(t)] = [ls(t')]$ and $[ls(t)] = [s]$, that is, they are *well-formed*.

As $\leq$ can be derived from the memberships in $\mathcal{E}$, it is also usual to write $\Sigma = (K, S, F)$ instead of $\Sigma = (K, S, \leq, F)$ when talking about MEL theories.

A MEL signature Σ imposes an associated set of memberships to any MEL theory $(\Sigma, \mathcal{E})$:

- for each $\kappa_1, \kappa_2 \in S \cup K$ such that $\kappa_1 < \kappa_2$, there is an associated unconditional membership $x_{\kappa_1} : \kappa_2$ in $\mathcal{E}$;
- each constant definition $c \in \Sigma_\kappa$ has an associated unconditional membership $c : \kappa$ in $\mathcal{E}$;
- each non constant definition $f \in \Sigma_{\kappa_1 \ldots \kappa_n, s}$, so $n \geq 1$, has an associated conditional membership $f(x_{[\kappa_1]}, \ldots, x_{[\kappa_n]}) : s \text{ if } x_{[\kappa_1]} : \kappa_1 \wedge \ldots \wedge x_{[\kappa_n]} : \kappa_n$ in $\mathcal{E}$; and

$$\frac{t \in T_\Sigma(\mathcal{X})}{t = t}\ \texttt{Reflexivity} \qquad \frac{t = t'}{t' = t}\ \texttt{Symmetry}$$

$$\frac{t' : s \quad t = t'}{t : s}\ \texttt{Membership} \qquad \frac{t_1 = t_2 \quad t_2 = t_3}{t_1 = t_3}\ \texttt{Transitivity}$$

$$\frac{f \in \Sigma_{k_1 \cdots k_n, k} \quad t_i = t_i' \quad t_i, t_i' \in T_\Sigma(X)_{k_i}, 1 \leq i \leq n}{f(t_1, \ldots, t_n) = f(t_1', \ldots, t_n')}\ \texttt{Congruence}$$

$$\frac{(\, A_0 \ \textit{if}\ \bigwedge_{i=1}^{n} A_i) \in \mathcal{E} \quad \sigma : X \to T_\Sigma(Y) \quad A_1\sigma \ldots A_n\sigma}{A_0\sigma}\ \texttt{Replacement}$$

Figure 2.1: Deduction rules for membership equational logic.

- each definition $f \in \Sigma_{k_1 \ldots k_n, k}$, with $n \geq 0$, has an associated unconditional membership $f(x_{k_1}, \ldots, x_{k_n}) : k$ in $\mathcal{E}$.

A MEL theory whose only memberships are the associated ones is an *order-sorted (OS)* theory, or a *many-sorted* theory if $<$ is the empty relation, where we can use all known results for these types of equational theories.

The MEL theory for our vending machine consists of the MEL signature Σ defined before and the following set $\mathcal{E}$ of MEL sentences:

- $x_{\texttt{Item}} : \texttt{St}$ $(\texttt{Item} < \texttt{State})$, $x_{\texttt{Coin}} : \texttt{St}$ $(\texttt{Coin} < \texttt{State})$
- $\texttt{q} : \texttt{Coin}$, $\$: \texttt{Coin}$, $\texttt{c} : \texttt{Item}$, $\texttt{a} : \texttt{Item}$
- $x_{\texttt{St}} y_{\texttt{St}} : \texttt{St}$ (juxtaposition of `States` is a `State`)
- $x_{[\texttt{St}]} y_{[\texttt{St}]} = y_{[\texttt{St}]} x_{[\texttt{St}]}$ (juxtaposition is commutative)
- $(x_{[\texttt{St}]} y_{[\texttt{St}]}) z_{[\texttt{St}]} = x_{[\texttt{St}]} (y_{[\texttt{St}]} z_{[\texttt{St}]})$ (juxtaposition is associative)
- $\texttt{qqqq} = \$$ (four quarters make a dollar)

Definition 1 (B-preregularity)**.** *Given a preregular OS signature Σ and a set of Σ-equations B, Σ is called* B-preregular *iff for each Σ-equation $u = v$ in B and substitution σ, $ls(u\sigma) = ls(v\sigma)$.*

Given a MEL sentence MS, we denote by $\mathcal{E} \vdash MS$ the fact that MS can be deduced from $\mathcal{E}$ using the rules in Figure 2.1 [BM06]; for an equation $t = t'$, $\mathcal{E} \vdash t = t'$ is also written $t =_\mathcal{E} t'$, for a membership $t : s$, $\mathcal{E} \vdash t : s$ is also written $t :_\mathcal{E} s$. These rules, where the symbol $=$ stands for $=$ or $:=$ indistinctly, specify a sound and complete calculus.

It is immediate to prove by induction that for each MEL sentence MS and substitution σ, if $\mathcal{E} \vdash MS$ then $\mathcal{E} \vdash MS\sigma$ using the same number of deduction steps.

Given two substitutions γ and δ, we write $\gamma =_\mathcal{E} \delta$ iff (i) $dom(\gamma) = dom(\delta)$ and (ii) for each variable $x \in dom(\gamma)$, $x\gamma =_\mathcal{E} x\delta$ and $vars(x\gamma) = vars(x\delta)$.

A MEL theory $(\Sigma, \mathcal{E})$ has an *initial algebra* $T_{\Sigma/\mathcal{E}}$, whose elements are the equivalence classes $[t]_\mathcal{E} \subseteq T_\Sigma$ of ground terms identified by the equations in $\mathcal{E}$.

The initial algebra for the vending machine is the set of all non-empty finite multisets, represented with the juxtaposition function, that can be made up with the four atoms q, \$, c, and a, with the exception that we consider four q's to be the same as one \$. For instance, `a q q q q`, `q q q q a`, and `a $` are the same multiset.

As $K = \bigcup_{s_i \in S}[s_i]$, we will also write $\Sigma = (S, \leq, F)$ when K is not explicitly needed, for instance when $(\Sigma, \mathcal{E})$ is either an order-sorted or a many-sorted equational theory. In fact, the Maude engines derive K from S, $\leq$, and $\mathcal{E}$, using a specific nomenclature.

2.2.3 OS signature with built-in subsignature

An OS theory $(\Sigma, \mathcal{E})$, with signature $\Sigma = (S, \leq, F)$, has *built-in subsignature* $\Sigma_0 = (S_0, \leq, F_0)$ [RMM17] iff:

- $\Sigma_0 \subseteq \Sigma$,
- Σ_0 is many-sorted,
- S_0 is a set of minimal elements in $(S, \leq)$, and
- if $f : w \to s \in F_1$, where $F_1 = F \setminus F_0$, then s is a sort not in S_0 and f has no other typing in Σ_0.

We let $\mathcal{X}_0 = \{\mathcal{X}_s\}_{s \in S_0}$, $\mathcal{X}_1 = \mathcal{X} \setminus \mathcal{X}_0$, $S_1 = S \setminus S_0$, $\Sigma_1 = (S, \leq, F_1)$, $\mathcal{H}_\Sigma(\mathcal{X}) = \mathcal{T}_\Sigma(\mathcal{X}) \setminus \mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$, and $\mathcal{H}_\Sigma = \mathcal{T}_\Sigma \setminus \mathcal{T}_{\Sigma_0}$.

2.2.4 Abstraction of built-in

This section is the first one to consider the use of SMT solvers for narrowing. All the definitions and results are always presented in the most generic way. Throughout the rest of this dissertation the symbol Σ_0 will represent the signature corresponding to the theories supported by any SMT solver.

If $\Sigma \supseteq \Sigma_0$ is a signature with built-in subsignature, then an *abstraction of built-in* is a context $\mathcal{C} = \lambda x^1_{s_1} \cdots x^n_{s_n}.t^\circ$, with $n \geq 0$, such that $t^\circ \in \mathcal{H}_\Sigma(\mathcal{X})$ and $\{x^1_{s_1}, \ldots, x^n_{s_n}\} = \mathit{vars}(t^\circ) \cap \mathcal{X}_0$.

Lemma 1 shows that there exists an abstraction that provides a canonical decomposition of any term in $\mathcal{H}_\Sigma(\mathcal{X})$.

Lemma 1 (Existence of a canonical abstraction [RMM17])**.** *Let Σ be a signature with built-in subsignature Σ_0. For each term t in $\mathcal{H}_\Sigma(\mathcal{X})$ there exists an abstraction of built-in $\lambda x^1_{s_1} \cdots x^n_{s_n}.t^\circ$ and a substitution $\theta^\circ : \mathcal{X}_0 \to \mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$ such that (i) $t = t^\circ\theta^\circ$ and (ii) $dom(\theta^\circ) = \{x^1_{s_1}, \ldots, x^n_{s_n}\}$ are pairwise distinct and disjoint from $vars(t)$; moreover, (iii) t° can always be selected to be S_0-linear and with $\{x^1_{s_1}, \ldots, x^n_{s_n}\}$ disjoint from an arbitrarily chosen finite subset $\mathcal{Y}$ of $\mathcal{X}_0$.*

Abstract function

Given a term t in $\mathcal{T}_\Sigma(\mathcal{X})$ and a finite subset $\mathcal{Y}$ of $\mathcal{X}_0$, define $\mathit{abstract}_{\Sigma_1}(t, \mathcal{Y})$ as the expression $\langle \lambda x^1_{s_1} \cdots x^n_{s_n}.t^\circ; \theta^\circ; \phi^\circ \rangle$ where the context $\lambda x^1_{s_1} \cdots x^n_{s_n}.t^\circ$ and the substitution θ° satisfy properties (i)-(iii) in Lemma 1 and $\phi^\circ = \bigwedge_{i=1}^n (x^i_{s_i} = x^i_{s_i}\theta^\circ)$. If $t \in \mathcal{T}_{\Sigma_1}(\mathcal{X} \setminus \mathcal{X}_0)$, then $\mathit{abstract}_{\Sigma_1}(t, \mathcal{Y}) = \langle \lambda.t; \mathit{none}; \mathit{true} \rangle$.

We write $\mathit{abstract}_{\Sigma_1}(t)$ when $\mathcal{Y}$ is the set of all the variables that have already appeared in the current calculation, so each $x^i_{s_i}$ is a *fresh* variable. For pairs of terms we use the compact notation $\mathit{abstract}_{\Sigma_1}((u, v)) = \langle \lambda(\bar{x}, \bar{y}).(u^\circ, v^\circ); (\theta^\circ_u, \theta^\circ_v); (\phi^\circ_u, \phi^\circ_v) \rangle$.

2.2.5 Unification

Given a MEL theory $(\Sigma, \mathcal{E})$, the *$\mathcal{E}$-subsumption* preorder $\ll_{\mathcal{E}}$ on $T_\Sigma(\mathcal{X})_k$ is defined by $t \ll_{\mathcal{E}} t'$ if there is a substitution σ such that $t =_{\mathcal{E}} t'\sigma$. For substitutions σ, ρ and a set of variables $\mathcal{V}$ we define $\sigma|_{\mathcal{V}} \ll_{\mathcal{E}} \rho|_{\mathcal{V}}$ if there is a substitution η such that $\sigma|_{\mathcal{V}} =_{\mathcal{E}} (\rho\eta)|_{\mathcal{V}}$. Then we say that ρ is more general than σ with respect to $\mathcal{V}$. When $\mathcal{V}$ is not specified, we assume that $dom(\rho) \subseteq dom(\sigma)$ and say that ρ is more general than σ.

Given a MEL theory $(\Sigma, \mathcal{E})$, an *$\mathcal{E}$-unifier* for a Σ-equation $u = v$ is a substitution σ such that $u\sigma =_{\mathcal{E}} v\sigma$. A set of substitutions $CSU_{\mathcal{E}}^{\mathcal{W}}(u = v)$ is said to be a *complete set of $\mathcal{E}$-unifiers* of $u = v$ away from $\mathcal{W}$ iff:

- each substitution σ in $CSU_{\mathcal{E}}^{\mathcal{W}}(u = v)$ is an $\mathcal{E}$-unifier of $u = v$;
- for any $\mathcal{E}$-unifier ρ of $u = v$ there is a substitution σ in $CSU_{\mathcal{E}}^{\mathcal{W}}(u = v)$ such that $\rho|_{\mathcal{W}} \ll_{\mathcal{E}} \sigma|_{\mathcal{W}}$;
- for each substitution σ in $CSU_{\mathcal{E}}^{\mathcal{W}}(u = v)$, $dom(\sigma) \subseteq vars(u) \cup vars(v)$ and $ran(\sigma) \cap \mathcal{W} = \emptyset$.

We will usually write $CSU_{\mathcal{E}}$ with the understanding that $\mathcal{W}$ is the set of all the variables that have already appeared in the current calculation.

This notion was introduced by Plotkin [Plo72]. An $\mathcal{E}$-unification algorithm is *complete* if for any given Σ-equation it generates a complete set of $\mathcal{E}$-unifiers, which may not be finite. An $\mathcal{E}$-unification algorithm is said to be *finitary* and complete if it terminates after generating a finite and complete set of solutions.

For instance, in our vending machine $CSU_{\mathcal{E}}(x_{\mathtt{St}}\ \mathtt{q}\ \mathtt{q}\ \mathtt{q} = \$) = \{\{x_{\mathtt{St}} \mapsto \mathtt{q}\}\}$.

2.3 Rewriting logic

2.3.1 Rewrite theory

A rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ consists of a MEL theory $(\Sigma, \mathcal{E})$ together with a finite set R of possibly labeled *conditional rewrite rules* each of which has the form

$$(c :)\ l \to r\ \textit{if}\ \bigwedge_h p_h = q_h \wedge \bigwedge_i u_i := v_i \wedge \bigwedge_j w_j : s_j \wedge \bigwedge_k l_k \to r_k,$$

where c is the label of the rule, l, r, and also each pair l_k, r_k, are Σ-terms of the same kind, and the rest of conditions fulfill the same requirements pointed out for MEL sentences. We will sometimes write $c : l \to r\ \textit{if}\ C$ as a shortcut. Rewrite rules can also be unconditional.

In our vending machine, R is the following set of labeled rewrite rules:

- `add-quarter`: $x_{\mathtt{St}} \to x_{\mathtt{St}}\ \mathtt{q}$ (quarter inserted)
- `add-dollar`: $x_{\mathtt{St}} \to x_{\mathtt{St}}\ \$$ (dollar inserted)
- `buy-coffee`: $\$ \to \mathtt{c}$ (coffee served, credit updated)
- `buy-apple`: $\$ \to \mathtt{a}\ \mathtt{q}$ (apple served, credit updated)

A rewrite rule $c : l \to r$ *if* C is *sort-decreasing* if for each substitution σ we have that $r\sigma \in T_\Sigma(\mathcal{X})_\kappa$ ($\kappa \in K \cup S$) and $C\sigma$ is verified imply $l\sigma \in T_\Sigma(\mathcal{X})_\kappa$.

A rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ allows the definition of different *rewrite relations* over $\mathcal{T}_\Sigma(\mathcal{X})$, displayed as $\to^1_{Rel}$, that can be extended to other structures. The transitive (resp. transitive and reflexive closure) of a rewrite relation $\to^1_{Rel}$ will be displayed as $\to^+_{Rel}$ (resp. $\to^*_{Rel}$). $(a, b) \in \to^1_{Rel}$ will be usually displayed as $a \to^1_{Rel} b$.

Equational and membership conditions in a rewrite rule are intended to be solved within the MEL theory $(\Sigma, \mathcal{E})$, i.e., no rewriting is allowed on those conditions, whereas a *reachability condition* $l_k \to r_k$ is satisfied in $\to^1_{Rel}$ if $(l_k, r_k) \in \to_{Rel}$, we will write $l_k \to_{Rel} r_k$ and say that r_k is *reachable* from l_k in $\to_{Rel}$, where $\to_{Rel}$ is a relation, usually parametric to $\to^1_{Rel}$, that has to be defined explicitly for each rewrite relation $\to^1_{Rel}$.

For any rewrite relation $\to^1_{Rel}$ we say that a term t is $\to^1_{Rel}$-*irreducible* (or just *Rel*-irreducible) if there is no term t' such that $t \to^1_{Rel} t'$. We say that a substitution is *Rel-normalized* (or normalized if $\to^1_{Rel}$ can be deduced from the context) if $x\sigma$ is *Rel*-irreducible for all $x \in dom(\sigma)$. We also say that a term t is *strongly Rel*-irreducible if for every *Rel*-normalized substitution σ the term $t\sigma$ is *Rel*-irreducible.

The rewrite relation $\to^1_{Rel}$ is *terminating* if there are no infinite rewriting sequences in $\to^1_{Rel}$. The rewrite relation $\to^1_{Rel}$ is *confluent* if whenever $t \to^*_{Rel} t_1$ and $t \to^*_{Rel} t_2$, there exists a term t_3 such that $t_1 \to^*_{Rel} t_3$ and $t_2 \to^*_{Rel} t_3$. For any terminating rewrite relation $\to^1_{Rel}$ we call $t{\downarrow_{Rel}}$ (a *canonical* form of t) any *Rel*-irreducible term t' such that $t \to^*_{Rel} t'$. We write $t \to^!_{Rel} t'$ to signal rewriting to canonical form and we write $t{\downarrow}$ instead of $t{\downarrow_{Rel}}$ when $\to^1_{Rel}$ can be deduced from the context. When a rewrite relation is sort-decreasing, terminating, and confluent, any term has a unique canonical form (sometimes up to some equational theory equivalence).

We assume that whenever a rewrite rule c is applied in any rewrite relation, what it is really used is an equivalent *fresh* version of c, i.e., a renaming of c such that $vars(c)$ has no variable in common with the set of all the variables that have already appeared in the current calculation, unless otherwise stated.

2.3.2 Conditional rewriting

Given a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, the relation $\to_R$ on $T_\Sigma(\mathcal{X})$ is defined as $\to^*_R$, where for a term $t \in T_\Sigma(\mathcal{X})$, a position $p \in Pos(t)$, and a substitution σ, a rewrite rule $c : l \to r$ *if* $C \in R$ specifies a *rewrite step* $t \to^1_R t[r\sigma]_p$ iff $t|_p = l\sigma$ and the instantiated condition $C\sigma$ holds, i.e., if $C = \bigwedge_h p_h = q_h \wedge \bigwedge_i u_i := v_i \wedge \bigwedge_j w_j : s_j \wedge \bigwedge_k l_k \to r_k$, then $\bigwedge_h p_h\sigma =_\mathcal{E} q_h\sigma$, $\bigwedge_i u_i\sigma =_\mathcal{E} v_i\sigma$, $\bigwedge_j w_j\sigma :_\mathcal{E} s_j$, and $\bigwedge_k l_k\sigma \to_R r_k\sigma$.

We write $t \xrightarrow[c,p,\sigma]{}{}^1_R t[r\sigma]_p$ when we need to make explicit the rule, position, and substitution. Any of these items can be omitted when it is irrelevant. We write $t \xrightarrow[c\sigma]{}{}^1_R t'$ to express that there exists a substitution δ such that $t \xrightarrow[c,\sigma\cdot\delta]{}{}^1_R t'$. For every rewrite step $t \to^1_R t'$ there exists a closed proof tree witnessing it, in the sense of [LMM05].

We write $\mathcal{R} \vdash u \to v$ if we can prove $u \to v$ using the inference rules for rewrite theories in Figure 2.2. These inference rules can infer all possible computations in the system specified by $\mathcal{R}$ [BM12], i.e., we can reach a state v from a state u in $\to_R$, denoted $u \to_R v$, iff we can prove $\mathcal{R} \vdash u \to v$.

Our vending machine is, as most reactive systems are [AILS07], non terminating. From any initial `State` we can always apply rules `add-quarter` and `add-dollar`. Rule `buy-coffee` is not sort-decreasing because it can turn a term with sort `Coin` into a term

$$\frac{t \in T_\Sigma(X)}{t \to t}\ \texttt{Reflexivity} \qquad \frac{t_1 \to t_2,\ t_2 \to t_3}{t_1 \to t_3}\ \texttt{Transitivity}$$

$$\frac{f \in \Sigma_{k_1 \cdots k_n, k}\ t_i \to t_i'\ \ t_i, t_i' \in T_\Sigma(X)_{k_i}, 1 \leq i \leq n}{f(t_1, \ldots, t_n) \to f(t_1', \ldots, t_n')}\ \texttt{Congruence}$$

$$\frac{\begin{array}{c}(\, c : l \to r\ \texttt{if}\ \bigwedge_i p_i = q_i \wedge \bigwedge_j w_j : s_j \wedge \bigwedge_k l_k \to r_k) \in R \\ \theta : X \to T_\Sigma(Y)\quad \bigwedge_i E \vdash p_i\theta = q_i\theta\quad \bigwedge_j E \vdash w_j\theta : s_j\quad \bigwedge_k l_k\theta \to r_k\theta\end{array}}{l\theta \to r\theta}\ \texttt{Replace}$$

Figure 2.2: Deduction rules for rewrite theories.

with sort `Item`, and it is not true that `Item` $\leq$ `Coin`. Also rule `buy-apple` is not sort-decreasing because it can turn a term with sort `Coin` into a term with sort `State`, which is strictly bigger than sort `Coin`.

2.3.3 Rewriting with built-ins plus axioms

As Section 2.2.4 was, this section is also related to the use of SMT solvers for narrowing. In a similar way to Σ_0, throughout the rest of this dissertation, the E_0 symbol will represent the set of equations corresponding to the theories supported by any SMT solver. The SMT solver used in practice to solve a reachability problem will determine the actual values of Σ_0 and E_0 in that case, but this will not change the validity of the proved results for the general theoretical case.

A theory inclusion $(\Sigma_0, E_0) \subseteq (\Sigma, \mathcal{E})$ is called *protecting* iff the Σ_0-homomorphism $\mathcal{T}_{\Sigma_0/E_0} \longrightarrow \mathcal{T}_{\Sigma/\mathcal{E}}|_{\Sigma_0}$ to the Σ_0-*reduct of the initial algebra* $\mathcal{T}_{\Sigma/\mathcal{E}}$, i.e., the elements of $\mathcal{T}_{\Sigma/\mathcal{E}}$ that consist only in function symbols from Σ_0, which is unique, is a Σ_0-isomorphism, written $\mathcal{T}_{\Sigma_0/E_0} \simeq \mathcal{T}_{\Sigma/\mathcal{E}}|_{\Sigma_0}$.

Rewrite theory with built-in

A rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ with built-in subtheory plus axioms (Σ_0, E_0) consists of:

1. an OS equational theory $(\Sigma, \mathcal{E})$ where:
   - $\Sigma = (S, \leq, F)$ is an OS signature with built-in subsignature $\Sigma_0 = (S_0, \leq, F_0)$,
   - $\mathcal{E} = E_0 \cup B$, where E_0 is the set of Σ_0-equations in $\mathcal{E}$, the theory inclusion $(\Sigma_0, E_0) \subseteq (\Sigma, \mathcal{E})$ is protecting, B is a set of regular and linear equations, called *axioms*, each equation having only function symbols from F_1 and kinded variables,
   - there is a procedure that can compute $CSU_B(l = r)$ for any Σ-equation $l = r$,
   - Σ is B-preregular, and
2. a finite set of uniquely labeled rules R, i.e., expressions with the form $c : l \to r\ \mathit{if}\ \bigwedge_{i=1}^n l_i \to r_i \mid \phi$, written $c : l \to r\ \mathit{if}\ \bar{l} \to \bar{r} \mid \phi$ or $c : l \to r\ \mathit{if}\ C$ as a shortcut, where:

- c is the *label* of the rule,
- l, the *head* of the rule, and r are terms in $\mathcal{H}_\Sigma(\mathcal{X})$, with $ls(l) \equiv_\leq ls(r)$,
- for each pair l_i, r_i, $1 \leq i \leq n$, l_i is a term in $\mathcal{H}_\Sigma(\mathcal{X}) \setminus \mathcal{X}$ and r_i is a term in $\mathcal{H}_\Sigma(\mathcal{X})$, with $ls(l_i) \equiv_\leq ls(r_i)$, and
- $\phi \in QF(\mathcal{X}_0)$, the set of *quantifier free* formulas made up with terms in $\mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$, the comparison function symbols $=$ and $\neq$, and the connectives $\vee$ and $\wedge$.

We will write the rewrite theory with built-in $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ when there is no need to be more specific.

The symbol $\neg$ (that can be defined with respect to $=$, $\neq$, $\vee$, and $\wedge$) will also appear in this dissertation. The (unique) label of a rule will be used as a reference for the whole rule, when there is no need to make it explicit. All the variables appearing in a rule c, $vars(c)$, are interpreted as universally quantified. Three particular cases of the general form are admitted: $c : l \to r$ *if* $\bigwedge_{i=1}^n l_i \to r_i$, $c : l \to r$ *if* ϕ, and the unconditional case $c : l \to r$.

Normal form of rule in a rewrite theory with built-in

Let $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ be a rewrite theory with built-in. For every rewrite rule $c : l \to r$ *if* $\bigwedge_{i=1}^n l_i \to r_i \mid \phi$ in R, if $abstract_{\Sigma_1}(l) = \langle \lambda \bar{x}.l^\circ; \theta^\circ; \phi^\circ \rangle$ (see Section 2.2.4), then its *normal form* is:

$$c^\circ : l^\circ \to r \text{ if } \bigwedge_{i=1}^n l_i \to r_i \mid \phi \wedge \phi^\circ,$$

where c° is the same label as c. We will always write c° to refer to the normal form of the rule c.

Narrowing for reachability with built-in will be based in rewriting with normal forms of rules.

Subterms, holes, and replacement in a formula

We extend the use of subterms and holes to formulas. If ϕ is a formula from $QF(\mathcal{X}_0)$, i is a positive integer, p is a position, and t is a term, then $\phi|_{i.p}$ is the subterm that appears at position p in the term i of $\bar{\phi}$, the tuple formed by all terms that appear in ϕ, taken from left to right, $\phi[]_{i.p}$ consists in the replacement in $\phi|_i$ of its subterm at position p with $[]$, and $\phi[t]_{i.p}$ consists in the replacement in $\phi|_i$ of its subterm at position p with t.

Rewrite relation for rewrite theories with built-in

Given a rewrite theory with built-in $\mathcal{R} = (\Sigma, E_0 \cup B, R)$, a term t in $\mathcal{H}_\Sigma$, a position p in $pos(t)$, a rule $c : l \to r$ *if* $\bigwedge_{i=1}^n l_i \to r_i \mid \phi$ in R, and a substitution $\sigma : vars(c) \to \mathcal{T}_\Sigma$, the rewrite step $t \to_R^1 t[r\sigma]_p$ holds iff $t = t[l\sigma]_p$, $l_i\sigma \to_R r_i\sigma$, for $1 \leq i \leq n$, and $E_0 \vdash \phi\sigma$, where $\to_R$ is $\to_R^*$.

Topmost rewrite theory

A rewrite theory with built-in $\mathcal{R} = (\Sigma, E \cup B, R)$ such that for some top sort `state`, no operator in Σ has `state` as argument sort and each rule $c : l \to r$ *if* $\bigwedge_{i=1}^n l_i \to r_i \mid \phi$

in R satisfies $l, r \in \mathcal{T}_\Sigma(\mathcal{X})_{\texttt{state}}$ and $l_i, r_i \in \mathcal{T}_\Sigma(\mathcal{X})_{\texttt{state}}$, for $1 \leq i \leq n$, is called a *topmost* rewrite theory.

In this dissertation, strong completeness results are proved for topmost rewrite theories.

2.3.4 Rewriting modulo

Consider a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ and a set $S \subseteq \mathcal{E}$[1]. The relation $\rightarrow_{R/S}$ on $T_\Sigma(\mathcal{X})$ is defined as $\rightarrow^*_{R/S} \cup =_S$ (see Section 2.3.1) where, for any two terms $u, v \in T_\Sigma(\mathcal{X})$, a rewrite rule $c : l \rightarrow r$ *if* $C \in R$ specifies a rewrite step $u \rightarrow^1_{R/S} v$ iff there exist a term $t \in T_\Sigma(\mathcal{X})$, a position $p \in Pos(t)$, and a substitution σ such that $u =_S t$, $t|_p = l\sigma$, $t[r\sigma]_p =_S v$ and the instantiated condition $C\sigma$ holds in $\rightarrow_{R/S}$, i.e., if $C = \bigwedge_h p_h = q_h \wedge \bigwedge_i u_i := v_i \wedge \bigwedge_j w_j : s_j \wedge \bigwedge_k l_k \rightarrow r_k$, then $\bigwedge_h p_h\sigma =_\mathcal{E} q_h\sigma$, $\bigwedge_i u_i\sigma =_\mathcal{E} v_i\sigma$, $\bigwedge_j w_j\sigma :_\mathcal{E} s_j$, and $\bigwedge_k l_k\sigma \rightarrow_{R/S} r_k\sigma$. The rule, position, and substitution applied can be indicated under the arrow, if needed, as in conditional rewriting.

The relation $\rightarrow^1_{R/\mathcal{E}}$ on $T_\Sigma(\mathcal{X})$ induces a relation $\rightarrow^1_{R/\mathcal{E}}$ on $T_{\Sigma/\mathcal{E}}(\mathcal{X})$, the equivalence relation modulo $\mathcal{E}$, by $[t]_\mathcal{E} \rightarrow^1_{R/\mathcal{E}} [t']_\mathcal{E}$ iff $t \rightarrow^1_{R/\mathcal{E}} t'$.

In a confluent, terminating, sort-decreasing, rewrite modulo relation $\rightarrow^1_{R/\mathcal{E}}$, for each term $t \in T_\Sigma(\mathcal{X})$, there is a unique (up to $\mathcal{E}$-equivalence) $R/\mathcal{E}$-irreducible term t' obtained from t by rewriting to canonical form, which is denoted by $t \rightarrow^!_{R/\mathcal{E}} t'$, or $t{\downarrow_{R/\mathcal{E}}}$ when t' is not relevant.

Equivalent definition of rewriting modulo

We can define $\rightarrow^1_{R/S}$ using an auxiliary rewrite relation called $\rightarrow^1_{R(S)}$. This alternative characterization has been extremely helpful in the proof of Theorem 4 in Chapter 4.

The intent of using this new rewrite relation is to split any rewrite step $u \rightarrow^1_{R/S} v$ into three parts: $u =_S t$, $t \rightarrow^1_{R(S)} t'$, and $t' =_S v$, so that we can reason in the proof of the theorem only about the intermediate part, $t \rightarrow^1_{R(S)} t'$. The relation $\rightarrow_{R(S)}$ associated to $\rightarrow^1_{R(S)}$ is an exception to the general rule, since it is not parametric to $\rightarrow^1_{R(S)}$.

Given a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, any set $S \subseteq \mathcal{E}$, a term t in $\mathcal{H}_\Sigma$, a position p in $pos(t)$, a rule $c : l \rightarrow r$ *if* $C \in R$, and a substitution σ, the rewrite step $t \rightarrow^1_{R(S)} t[r\sigma]_p$ holds iff $t|_p = l\sigma$ and the instantiated condition $C\sigma$ holds in $\rightarrow_{R/S}$.

According to this definition and the previous definition of $\rightarrow^1_{R/S}$, $u \rightarrow^1_{R/S} v$ if and only if $u =_S t \rightarrow^1_{R(S)} t[r\sigma]_p =_S v$, i.e., $\rightarrow^1_{R/S}$ is equivalent to $=_S; \rightarrow^1_{R(S)}; =_S$.

Obviously, $\rightarrow^1_R \subseteq \rightarrow^1_{R(S)} \subseteq \rightarrow^1_{R/S} \subseteq \rightarrow^1_{R/\mathcal{E}}$.

Remark 1. *(i) If R has no reachability conditions on its rules then $\rightarrow^1_R = \rightarrow^1_{R(S)}$, since both relations only differ in the definitions of $\rightarrow_R$ and $\rightarrow_{R(S)}$.*

(ii) If $t \rightarrow^1_{R(\mathcal{E})} t'$ using a rule with no reachability conditions then $t \rightarrow^1_R t'$ using the same rule, and vice versa.

Remark 2. *If $t =_\mathcal{E} t'$ then $t \rightarrow_{R/\mathcal{E}} t'$, without applying any $\rightarrow^1_{R/\mathcal{E}}$ step.*

[1] In this dissertation S will always be either $\mathcal{E}$ or the set of *axioms* in $\mathcal{E}$ (usually called B), defined in Section 2.3.5

For a rewrite theory without built-in $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ whether a one step rewrite $t \to^1_{R/\mathcal{E}} t'$ holds is undecidable in general, since the elements in the equivalence class $[t]_\mathcal{E}$ may be infinite. Instead, we will use a new relation named $\to^1_{ER,B}$ that, under several conditions, can be used to imitate $\to^1_{R/\mathcal{E}}$. The first step is to associate a rewrite theory to each MEL theory. For appropriate MEL theories equality modulo $\mathcal{E}$ will be solved via rewriting using these rewrite theories.

2.3.5 Associated rewrite theory

For any MEL theory $(\Sigma, \mathcal{E})$, if $\mathcal{E}$ can be decomposed as $E \cup B$, with B a set of axioms, i.e., regular and linear equations having only function symbols and kinded variables, for which there exist a procedure that can compute $CSU_B(l = r)$ for any Σ-equation $l = r$, then $(\Sigma, \mathcal{E})$ has a corresponding rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$ associated to it [DLM+08]. The associated rewrite theory is constructed in the following way: we add a new connected component with sort *Truth*, a new constant `tt` of this sort to Σ, for each sort $s \in S$ a new function symbol $\_{:}s \ : \ [s] \to \mathit{Truth}$, and for each kind $k \in K$ a new function symbol $eq : \ k \ k \to \mathit{Truth}$. There are rules $eq(x_k, x_k) \to \mathtt{tt}$ in R_E for each kind $k \in K$. For each equation or membership in E

$$t = t' \ \mathit{if} \ A_1 \wedge \ldots \wedge A_n \qquad t : s \ \mathit{if} \ A_1 \wedge \ldots \wedge A_n,$$

R_E has a conditional rule of the form

$$t \to t' \ \mathit{if} \ A'_1 \wedge \ldots \wedge A'_n \qquad t{:}s \to \mathtt{tt} \ \mathit{if} \ A'_1 \wedge \ldots \wedge A'_n$$

where if A_i is $t_i : s_i$ then A'_i is $t_i{:}s_i \to \mathtt{tt}$, if A_i is $t_i := t'_i$ then A'_i is $t'_i \to t_i$, and if A_i is $t_i = t'_i$ then A'_i is $eq(t_i, t'_i) \to \mathtt{tt}$.

We define the relation $\to^1_{E/B}$ as $\to^1_{R_E/B}$, so $\to_{E/B}$ is $\to^*_{E/B} \cup =_B$ (or, equivalently, $\to^*_{E/B}; =_B$) by Section 2.3.4.

2.3.6 E, B-rewriting and $R(E), B$-rewriting

Given a rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$, if $\mathcal{R}_E = (\Sigma', B, R_E)$ is the rewrite theory associated to $(\Sigma, E \cup B)$ then we define the relations $\to^1_{E,B}$ and $\to^1_{R(E),B}$, where the relation $\to_{E,B}$ is $(\to^*_{E,B}; =_B)$, $\to^1_{ER,B}$ is $(\to^*_{E,B}; \to^1_{R(E),B})$, $\to_{ER,B}$ is $(\to^*_{ER,B}; =_E)$, and for terms $t, t' \in T_\Sigma(\mathcal{X})$:

- $t \to^1_{E,B} t'$ if there is a rule $l \to r \ \mathit{if} \ \bigwedge_{i \in I} A'_i$ in R_E, a position $p \in Pos(t)$, and a substitution σ such that $t|_p =_B l\sigma$ (B-matching), $t' = t[r\sigma]_p$, and for all $i \in I$ $t_i\sigma \to_{E,B} t'_i\sigma$, and
- $t \to^1_{R(E),B} t'$ if there is a position $p \in Pos(t)$, a rule $c : l \to r \ \mathtt{if} \ C \in R$, and a substitution σ such that $t|_p =_B l\sigma$ (B-matching), $t' = t[r\sigma]_p$, and $C\sigma$ holds under $\to_{ER,B}$, i.e., if $C = \bigwedge_h p_h = q_h \wedge \bigwedge_i u_i := v_i \wedge \bigwedge_j w_j : s_j \wedge \bigwedge_k l_k \to r_k$, then $\bigwedge_h eq(p_h\sigma, q_h\sigma) \to_{E,B} \mathtt{tt} \wedge \bigwedge_i v_i\sigma \to_{E,B} u_i\sigma \wedge \bigwedge_j w_j\sigma{:}s_j \to_{E,B} \mathtt{tt} \wedge \bigwedge_k l_k\sigma \to_{ER,B} r_k\sigma$.

Remark 3. *For all $t, t' \in T_\Sigma(\mathcal{X})$, $t \to_{E,B} t$ and if $t =_B t'$ then $t \to_{E,B} t'$, in both cases without applying any rewrite rule from R_E.*

$$\frac{t_1 \to^1 t_2,\ t_2 \to t_3}{t_1 \to t_3}\ \texttt{Transitivity} \qquad \frac{t \to^1 t', t' : s}{t : s}\ \texttt{Subject Reduction}$$

$$\frac{t =_B t'}{t \to t'}\ \texttt{Reflexivity} \qquad \frac{t_i \to^1 t_i'}{f(t_1, \ldots, t_i, \ldots, t_n) \to^1 f(t_1, \ldots, t_i', \ldots, t_n)}\ \texttt{Congruence}$$

$$\frac{t \to t' \text{ if } A_1' \ldots A_n' \in R_E \text{ and } u =_B t\sigma \quad A_1'\sigma \ldots A_n'\sigma}{u \to^1 t'\sigma}\ \texttt{Replacement}$$

$$\frac{t : s \text{ if } A_1' \ldots A_n' \in R_E \text{ and } u =_B t\sigma \quad A_1'\sigma \ldots A_n'\sigma}{u : s}\ \texttt{Membership}$$

Figure 2.3: Inference rules for membership rewriting.

Under several requirements, $\to^1_{R/\mathcal{E}}$, which is undecidable in general, can be imitated using $\to^1_{ER,B}; =_{\mathcal{E}}$, which will be decidable. In fact, it is enough to have a finite B-matching algorithm, but narrowing requires a complete B-unification algorithm, so we will stick to this requirement, which is stricter since a B-unification algorithm serves as B-matching algorithm when we consider the variables in one term, in this case $t|_p$, as new constants of Σ.

The *inference rules for membership rewriting in* $\mathcal{R}_E$ are the ones in Figure 2.3, adapted from [DLM+08, Fig. 4], where (i) the rules are defined for context-sensitive membership rewriting and (ii) for *executable* MEL theories (see Section 2.3.8), where it is always the case that $\to^1_{E/B} = (\to^1_{E,B}; =_B)$ and $\to_{E/B} = \to_{E,B}$, it is proved that:

1. $\mathcal{R}_E \vdash t \to^1 t'$ and $t \to^1_{E,B} t'$ are equivalent,
2. $\mathcal{R}_E \vdash t \to t'$, $t \to_{E,B} t'$ and $t =_{\mathcal{E}} t'$ are equivalent,
3. $\mathcal{R}_E \vdash t : s$ and $t :_{\mathcal{E}} s$ are equivalent, and
4. $\mathcal{R}_E \vdash t \to t'$ and $t :_{\mathcal{E}} s$ imply $t' :_{\mathcal{E}} s$ or, equivalently, $t =_{\mathcal{E}} t'$ and $t :_{\mathcal{E}} s$ imply $t' :_{\mathcal{E}} s$.

As direct consequences, $t{\downarrow} =_B t'{\downarrow}$ implies $t =_{\mathcal{E}} t'$, and $t{\downarrow} :_{\mathcal{E}} s$ implies $t :_{\mathcal{E}} s$.

2.3.7 R, B-rewriting for rewrite theories with built-in

Set of topmost Σ_0-positions

This is a concept that we did not find in the existing literature, which is *essential* to the definition of R, B-rewriting for rewrite theories with built-in.

Let $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ be a rewrite theory with built-in subtheory (Σ_0, E_0), and t a term in $\mathcal{H}_\Sigma(\mathcal{X})$. The set of topmost Σ_0 positions of t, $top_{\Sigma_0}(t)$, is $top_{\Sigma_0}(t) = \{p \mid p \in Pos(t) \wedge \exists i \in \mathbb{N}(p = q.i \wedge t|_q \in \mathcal{H}_\Sigma(\mathcal{X}) \wedge t|_p \in \mathcal{T}_{\Sigma_0}(\mathcal{X}_0))\}$.

$top_{\Sigma_0}(t)$ characterizes the largest $\mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$-subterms of t. Obviously, if $p, q \in top_{\Sigma_0}(t)$ and $p \neq q$ then neither $p \leq q$ nor $q \leq p$.

The definition of the relation $\to^1_{R,B}$ will require the use of a single *representative* for all the instances of each E_0-equivalence class that may appear in the top_{Σ_0} positions of the subterm that we are rewriting.

Representative of a Σ_0-term over a set of Σ_0 terms

Let t be a term in $\mathcal{T}_{\Sigma_0}$ and let $\hat{u} = \{u_1, \ldots, u_n\} \subseteq \mathcal{T}_{\Sigma_0}$ such that $t \in \hat{u}$. We define the Σ_0-representative of t over $\hat{u}$ as $rep^\circ_{\hat{u}}(t) = u_{min(\{i|u_i =_{E_0} t)\})}$. We homomorphically extend the definition to lists and sets of terms.

Then $rep^\circ_{\hat{u}}(\hat{u})$ will be a set containing one element for each E_0-equivalence class that appears in $\hat{u}$, the representative of the class over $\hat{u}$.

Example 1. *If (Σ_0, E_0) is integer arithmetic and $\hat{u} = \{3, 2{+}1, 1{+}1, 1{+}2, 2\}$ then $rep^\circ_{\hat{u}}(3) = 3$, $rep^\circ_{\hat{u}}(2{+}1) = 3$, $rep^\circ_{\hat{u}}(1{+}1) = 1{+}1$, $rep^\circ_{\hat{u}}(1{+}2) = 3$, $rep^\circ_{\hat{u}}(2) = 1{+}1$, and $rep^\circ_{\hat{u}}(\hat{u}) = \{3, 1{+}1\}$.*

Representative of a term over a set of Σ_0 terms

Let t be a term in $\mathcal{T}_\Sigma$, where $top_{\Sigma_0}(t) = \hat{p}$, and let $\hat{u} \subseteq \mathcal{T}_{\Sigma_0}$ such that $t|_{\hat{p}} \subseteq \hat{u}$. We define the *representative* of t over $\hat{u}$, as $rep_{\hat{u}}(t) = t[rep^\circ_{\hat{u}}(t|_{\bar{p}})]_{\bar{p}}$. We homomorphically extend the definition to lists and sets of terms.

Then $rep_{\hat{u}}(\hat{u})$ will be a set containing one element for each E_0-equivalence class that appears in $\hat{u}$, the *representative* of the class over $\hat{u}$.

Example 2. *If (Σ_0, E_0) is integer arithmetic, where we represent its only sort with i, $\hat{u} = \{3, 2{+}1, 1{+}1, 1{+}2, 2\}$, Σ has one sort s and one rank declaration $f : i\ i\ i \to s$, and $t = f(2, 2{+}1, 1{+}2)$, so $top_{\Sigma_0}(t) = \{1, 2, 3\}$, let $\bar{p} = 1, 2, 3$, then*

1. *$rep_{\hat{u}}(\hat{u})$ will be $\{3, 1{+}1\}$, i.e., $rep^\circ_{\hat{u}}(\hat{u})$, and*
2. *$rep_{\hat{u}}(t) = t[rep^\circ_{\hat{u}}(t|_{\bar{p}})]_{\bar{p}} = f(rep^\circ_{\hat{u}}(2), rep^\circ_{\hat{u}}(2{+}1), rep^\circ_{\hat{u}}(1{+}2)) = f(1{+}1, 3, 3)$.*

Remark 4. *From the previous definitions it is immediate that:*

- *if t is a term in $\mathcal{T}_\Sigma$ then $t =_{E_0} rep_{\hat{u}}(t)$,*
- *if t is a term in $\mathcal{T}_{\Sigma_0}$ then $rep^\circ_{\hat{u}}(t) = rep_{\hat{u}}(t)$,*
- *if $top_{\Sigma_0}(t) = \hat{p}$ and $t|_{\hat{p}} \subseteq \hat{u}$ then $rep^\circ_{\hat{u}}(t|_{\hat{p}}) = rep_{\hat{u}}(t|_{\hat{p}}) \subseteq rep_{\hat{u}}(\hat{u})$,*
- *if t is a term in $rep_{\hat{u}}(\hat{u})$ then $rep_{\hat{u}}(t) = t$, and*
- *if u_1 and u_2 are two elements of $rep_{\hat{u}}(\hat{u})$ and $u_1 =_{E_0} u_2$ then $u_1 = u_2$.*

Representative of a substitution over a set of Σ_0-terms

Let σ be a ground substitution and let $\hat{u} \subseteq \mathcal{T}_{\Sigma_0}$ such that $\bigcup_{z \in dom(\sigma)} \{(z\sigma)|_{top_{\Sigma_0}(z\sigma)}\} \subseteq \hat{u}$. We define the representative of σ as $rep_{\hat{u}}(\sigma) = \{z \mapsto rep_{\hat{u}}(z\sigma) \mid z \in dom(\sigma)\}$, i.e., each top_{Σ_0}-term in σ is replaced by its representative with respect to $\hat{u}$, so $\sigma =_{E_0} rep_{\hat{u}}(\sigma)$.

Representative of a term

Let t be a term in $\mathcal{T}_\Sigma$, where $top_{\Sigma_0}(t) = \hat{p}$. We define the *representative* of t as $rep(t) = rep_{t|_{\hat{p}}}(t)$.

Example 3. *In the previous example, as $t = f(2, 2{+}1, 1{+}2)$ and $\hat{p} = \{1, 2, 3\}$, then $t|_{\hat{p}} = \{2, 2{+}1, 1{+}2\}$ and $rep(t) = f(2, 2{+}1, 2{+}1)$.*

R, B-rewriting

The relation $\to_{R,B}$ is defined as $\to^*_{R,B}; =_{\mathcal{E}}$, where the relation $\to^1_{R,B}$ is inductively defined below.

Definition 2. *Given a rewrite theory with built-in $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, where $\mathcal{E} = E_0 \cup B$, terms t, t' in $\mathcal{H}_\Sigma$, and a rule $c : l \to r$ if $\bigwedge_{i=1}^n l_i \to r_i \mid \phi$ in R, if c° has the form $c^\circ : l^\circ \to r$ if $\bigwedge_{i=1}^n l_i \to r_i \mid \phi \wedge \phi^\circ$, and there exist a position p in $pos_{\Sigma_1}(t)$ and a substitution $\sigma : vars(c^\circ) \to \mathcal{T}_\Sigma$ such that $rep(t|_p) =_B l^\circ\sigma$, $t' = t[r\sigma]_p$, $l_i\sigma \to_{R,B} r_i\sigma$, for $1 \le i \le n$, and $E_0 \vdash (\phi \wedge \phi^\circ)\sigma$, then we say there is a one-step transition $t \to^1_{R,B} t'$.*

We write $t \xrightarrow[c,p,\sigma]{}{}^1_{R,B} t'$, when we need to make explicit the rule, position, and substitution. Any of these items can be omitted when it is irrelevant.

The following example shows the motivation behind the use of representatives in $\to^1_{R,B}$.

Example 4. *Let $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ be a rewrite theory, where E_0 is integer arithmetic, there is one non-E_0 sort s, with three function symbols $a :\to s$ (constant), $f : s\ s \to s$ (associative), and $g : i \to s$, so B is associativity for f, and $R = \{c : f(x_s, f(y_s, y_s)) \to y_s\}$, let $l = f(x_s, f(y_s, y_s))$, hence $abstract_{\Sigma_1}(l) = \langle \lambda.l; none; true\rangle$ and $l^\circ = l$.*

The term $t = f(f(a, g(3)), g(1 + 2))$ does not match $f(x_s, f(y_s, y_s))$ modulo B, but $rep(t) = f(f(a, g(3)), g(3))$ does, with $\sigma = \{x_s \mapsto a, y_s \mapsto g(3)\}$, because $rep(t) = f(f(a, g(3)), g(3)) =_B l^\circ\sigma (= l\sigma = f(a, f(g(3), g(3))))$, so $t \to^1_{R,B} g(3)$.

As $t' \xrightarrow[c,\epsilon,\sigma]{}{}^1_{R(\mathcal{E})} g(3)$, because $t' = l\sigma$, and $t =_{E_0} rep(t) =_B t'$, so $t =_{\mathcal{E}} t'$, then $t \to^1_{R/\mathcal{E}} g(3)$, i.e., the use of representatives allows us to imitate $\to^1_{R/\mathcal{E}}$ with $\to^1_{R,B}$.

2.3.8 Executable rewrite theory

For a rewrite theory without built-in $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, where $\mathcal{E} = E \cup B$, whether a one step rewrite $t \to^1_{R/\mathcal{E}} t'$ holds is undecidable in general. We impose additional conditions, under which $\to^1_{R/\mathcal{E}}$ can be imitated with $\to^1_{ER,B}; =_{\mathcal{E}}$. If $\mathcal{R}$ satisfies these conditions, we will say that $\mathcal{R}$ is *executable*.

Operational termination

One problem that can arise when trying to decide $t \to^1_R t'$ in $\mathcal{R}$ is that although $\to^1_R$ is terminating, an attempt to prove a condition in a rule, building a so-called *well-formed proof tree* [LM09], may generate a recursive infinite check of conditions, and a corresponding infinite well-formed proof tree. This leads us to the notion of *operational termination*:

The relation $\to^1_R$ is operationally terminating if there are no infinite well-formed proof trees.

This notion of operational termination was presented by Lucas, Marché and Meseguer [LMM05] in an attempt to exclude those conditional term rewriting systems like the one consisting of two constants (a and b), one binary function (f), and the single conditional rule:

$$a \to b \quad \textit{if} \quad f(a) \to b$$

The absence of unconditional rules makes the relation $\rightarrow_R^1$ trivially empty, hence terminating. Nevertheless, when trying to reduce the term a, most implementations will loop because of the following infinite derivation tree:

$$\dfrac{\dfrac{\dfrac{\dfrac{\cdots}{a \rightarrow b}}{f(a) \rightarrow b}}{a \rightarrow b}}{}$$

The condition of operational termination states that such derivation trees don't exist.

Σ-pattern. Admissible MEL theory

Another problem that may arise when trying to decide $t \rightarrow_R^1 t'$ in $\mathcal{R}$ is the handling of new variables that may appear in the MEL sentences of $\mathcal{E}$. We limit these sentences so that these new variables can always be instantiated by B-matching against the old ones.

Given a MEL theory $(\Sigma, E \cup B)$ we call a term $t \in T_\Sigma(\mathcal{X}) \setminus \mathcal{X}$ a Σ-*pattern* if for any E, B-normalized substitution σ, $t\sigma$ is E, B-irreducible.

A sufficient condition for t to be a Σ-pattern is the absence of B-unifiers between nonvariable subterms of t and lefthand sides of equations in E.

A MEL theory $(\Sigma, E \cup B)$ is *admissible* [CDE+07] if:

- For each equation $l = r$ *if* $\bigwedge_{i=1}^n A_i$ in E, $n \geq 0$, the following requirements are satisfied:

  1. $$vars(r) \subseteq vars(l) \cup \bigcup_{j=1}^{n} vars(A_j).$$

  2. If A_i is an equation $u_i = v_i$ or a membership $u_i : s_i$, then
  $$vars(A_i) \subseteq vars(l) \cup \bigcup_{j=1}^{i-1} vars(A_j).$$

  3. If A_i is a matching equation $u_i := v_i$, then u_i is a Σ-pattern and
  $$vars(v_i) \subseteq vars(l) \cup \bigcup_{j=1}^{i-1} vars(A_j).$$

- For each conditional membership $l : s$ *if* $\bigwedge_{i=1}^n A_i$ in E conditions 2 and 3 above are satisfied.

Admissible rewrite theory

A rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ is *admissible* if:

1. The MEL theory $(\Sigma, E \cup B)$ is admissible.
2. For each rule $l \rightarrow r$ *if* $\bigwedge_{i=1}^n A_i$ in R, $n \geq 0$:
   - conditions (1)-(3) above hold, and
   - if A_i is a reachability condition $u_i \rightarrow v_i$, then v_i is a Σ-pattern and
   $$vars(u_i) \subseteq vars(l) \cup \bigcup_{j=1}^{i-1} vars(A_j).$$

Executable rewrite theory

A rewrite theory without built-in $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, where $\mathcal{E} = E \cup B$, is *executable* if each kind in Σ is nonempty, E, B, and R are finite and the following conditions hold:

1. $\rightarrow^1_{E,B}$ and $\rightarrow^1_{R(E),B}$ are operationally terminating and admissible. Then we have a *deterministic 3-CTRS* [Ohl02]. Any new variable in the conditions will be instantiated by B-matching, in left to right order.

2. $\rightarrow^1_{E/B}$ is sort-decreasing, terminating, and confluent.

3. $\rightarrow^1_{E,B}$ is *coherent with* B, i.e., $\forall t_1, t_2, t_3$ we have $t_1 \rightarrow^+_{E,B} t_2$ and $t_1 =_B t_3$ implies $\exists t_4, t_5$ such that $t_2 \rightarrow^*_{E,B} t_4, t_3 \rightarrow^+_{E,B} t_5$ and $t_4 =_B t_5$ [MT07].

$$\begin{array}{ccc} t_1 & \rightarrow^+_{E,B} & t_2 \\ \|_B & & \\ t_3 & \longrightarrow^+_{E,B} & \end{array} \qquad \begin{array}{ccc} & \rightarrow^*_{E,B} & t_4 \\ & & \|_B \\ & & t_5 \end{array}$$

4. $\rightarrow^1_{R(E),B}$ is *$\mathcal{E}$-consistent with* B, i.e., for all t_1, t_2, t_3 we have that $t_1 \rightarrow^1_{R(E),B} t_2$ and $t_1 =_B t_3$ implies that there exists t_4 such that $t_3 \rightarrow^1_{R(E),B} t_4$ and $t_2 =_{\mathcal{E}} t_4$. Also $\rightarrow^1_{R(E),B}$ is *$\mathcal{E}$-consistent with* $\rightarrow^1_{E,B}$, i.e., for all t_1, t_2, t_3 we have that $t_1 \rightarrow^1_{R(E),B} t_2$ and $t_1 \rightarrow^*_{E,B} t_3$ implies that there exist t_4, t_5 such that $t_3 \rightarrow^*_{E,B} t_4$ and $t_4 \rightarrow^1_{R(E),B} t_5$ and $t_2 =_{\mathcal{E}} t_5$. In either case, the $\rightarrow^1_{R(E),B}$ rewriting step from t_3 and t_4, respectively, must be performed with the *same* rule that was applied to t_1 [MT07].

$$\begin{array}{ccc} t_1 & \rightarrow^1_{R(E),B} & t_2 \\ \|_B & & \|_{\mathcal{E}} \\ t_3 & \rightarrow^1_{R(E),B} & t_4 \end{array}$$

(a) $\mathcal{E}$-consistency of $\rightarrow^1_{R(E),B}$ with B

$$\begin{array}{ccccc} t_1 & & \longrightarrow^1_{R(E),B} & & t_2 \\ \downarrow^*_{E,B} & & & & \|_{\mathcal{E}} \\ t_3 & \rightarrow^*_{E,B} & t_4 & \rightarrow^1_{R(E),B} & t_5 \end{array}$$

(b) $\mathcal{E}$-consistency of $\rightarrow^1_{R(E),B}$ with $\rightarrow^1_{E,B}$

The rewrite theory for our vending machine is executable if we decompose $\mathcal{E}$ in the following way: the set B contains the equations for the commutative and the associative properties for function $\cdot$ (juxtaposition); the set E contains the other equations and all the membership predicates. E and R are admissible because they are regular and don't introduce new variables.

Theorem 1 (Relation between $\rightarrow^1_{R/\mathcal{E}}$ and $\rightarrow^1_{ER,B}$ for executable rewrite theories [Vir02])**.** *Let $\mathcal{R} = (\Sigma, E \cup B, R)$ be an executable rewrite theory. For all t_1, t_2, t_3, if $t_1 \rightarrow^1_{R/\mathcal{E}} t_2$ and $t_1 \rightarrow^*_{E,B} t_3$ then there exists t_4 such that $t_3 \rightarrow^1_{R(E),B} t_4$, i.e., $t_1 \rightarrow^1_{ER,B} t_4$, and $t_4 =_{\mathcal{E}} t_2$.*

$$\begin{array}{ccc} t_1 & \longrightarrow^1_{R/\mathcal{E}} & t_2 \\ \downarrow^*_{E,B} & & \|_{\mathcal{E}} \\ t_3 & \longrightarrow^1_{R(E),B} & t_4 \end{array}$$

As a direct consequence, due to the definitions of $\rightarrow_{R/\mathcal{E}}$ and $\rightarrow_{ER,B}$, we get the following corollary.

Corollary 1 (Equality of $\rightarrow_{R/\mathcal{E}}$ and $\rightarrow_{ER,B}$ for executable rewrite theories)**.** *If $\mathcal{R} = (\Sigma, E \cup B, R)$ is an executable rewrite theory then $\rightarrow_{R/\mathcal{E}} = \rightarrow_{ER,B}$.*

Executable MEL theory

If $\mathcal{R} = (\Sigma, E \cup B, R)$ is an executable rewrite theory then we call $(\Sigma, E \cup B)$ an *executable* MEL *theory*. Let $\mathcal{R}_E = (\Sigma', B, R_E)$ be the associated rewrite theory of $(\Sigma, E \cup B)$. As $B = \emptyset \cup B$:

- in the diagram for Theorem 1, as the relation $\rightarrow^1_{\emptyset,B}$ is empty, then $t_1 = t_3$ and $t_2 =_B t_4$, so $t_1 \rightarrow^1_{E/B} t_2$ if and only if $t_1 \rightarrow^1_{E,B} t_4 =_B t_2$, hence $\rightarrow^1_{E/B} = \rightarrow^1_{E,B}$ up to B-equivalence, which is decidable,
- from Corollary 1, $\rightarrow_{E/B} = \rightarrow_{E,B}$, and
- as $\rightarrow_{E/B}$ is confluent and terminating then, for any $t, t' \in T_\Sigma(\mathcal{X})$, $t =_\mathcal{E} t'$ if and only if $t{\downarrow} =_B t'{\downarrow}$.

2.3.9 System of sentences. Unification goal. $\mathcal{E}$-solution

A *system of sentences* F in a MEL theory $(\Sigma, \mathcal{E})$ has the form

$$\bigwedge_{i=1}^{n} u_i = u'_i \wedge \bigwedge_{j=1}^{m} v_j := v'_j \wedge \bigwedge_{k=1}^{l} t_k : s_k,$$

where v_j is a Σ-pattern, for $1 \leq j \leq m$. A substitution σ is an *$\mathcal{E}$-solution* for F if $u_i\sigma =_\mathcal{E} u'_i\sigma$ $(1 \leq i \leq n)$, $v_j\sigma =_\mathcal{E} v'_j\sigma$ $(1 \leq j \leq m)$, and $t_k\sigma :_\mathcal{E} s_k$ $(1 \leq k \leq l)$.

F has an associated *unification* goal G in $\mathcal{R}_E$ of the form

$$\bigwedge_{i=1}^{n} eq(u_i, u'_i) \rightarrow \mathtt{tt} \wedge \bigwedge_{j=1}^{m} v'_j \rightarrow v_j \wedge \bigwedge_{k=1}^{l} t_k{:}s_k \rightarrow \mathtt{tt}.$$

A substitution σ is an *$\mathcal{E}$-solution* for G if $eq(u_i\sigma, u'_i\sigma) \rightarrow_{E/B} \mathtt{tt}$, for $1 \leq i \leq n$, $v'_j\sigma \rightarrow_{E/B} v_j\sigma$, for $1 \leq j \leq m$, and $t_k\sigma{:}s_k \rightarrow_{E/B} \mathtt{tt}$, for $1 \leq k \leq l$. We will write $\top$ to represent a conjunction of any number of $\mathtt{tt}$'s.

For executable MEL theories, as $\rightarrow_{E/B} = \rightarrow_{E,B}$, we can check $\mathcal{E}$-solutions for systems of sentences using E, B-rewriting. As $\rightarrow^1_{E,B}$ is sort-decreasing, terminating, and confluent, we call the *normal form* of G, written $G{\downarrow}$, to the unification goal that results from replacing every term t in G with $t{\downarrow}$.

2.3.10 Reachability goals and problems

Rewrite theories without built-in

Given a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, a *reachability goal* G is a conjunction of the form $t_1 \rightarrow t'_1 \wedge \ldots \wedge t_n \rightarrow t'_n$ where for $1 \leq i \leq n$, $t_i, t'_i \in T_\Sigma(X)_{s_i}$ for appropriate s_i. A substitution σ is a *solution* of G if $t_i\sigma \rightarrow_{R/\mathcal{E}} t'_i\sigma$ for $1 \leq i \leq n$. A substitution σ is a *trivial* solution of G if $t_i\sigma =_\mathcal{E} t'_i\sigma$ for $1 \leq i \leq n$. G is trivial if the identity substitution id is a trivial solution of G.

For instance, in the rewrite theory for the vending machine the reachability goal $x_{\mathtt{St}}\ \$ \rightarrow x_{\mathtt{St}}\ \mathtt{c}$ is trivial. Also, $\{x_{\mathtt{St}} \mapsto q\}$ is a trivial solution of the reachability goal $x_{\mathtt{St}}\ \mathtt{q}\ \mathtt{q}\ \mathtt{q} \rightarrow \$\ (\mathtt{q}\ \mathtt{q}\ \mathtt{q}\ \mathtt{q} =_\mathcal{E} \$)$, but it is a non-trivial solution of the reachability goal $x_{\mathtt{St}}\ \mathtt{q}\ \mathtt{q} \rightarrow \$\ (\mathtt{q}\ \mathtt{q}\ \mathtt{q} \xrightarrow[\mathtt{add-quarter}]{}{}^{1}_{R/\mathcal{E}}\ \$)$.

For reachability goals $G : t_1 \to t_2 \wedge \ldots \wedge t_{2n-1} \to t_{2n}$ and $G' : t'_1 \to t'_2 \wedge \ldots \wedge t'_{2n-1} \to t'_{2n}$, we say $G =_{\mathcal{E}} G'$ if $t_i =_{\mathcal{E}} t'_i$ for $1 \leq i \leq 2n$. Any relation $\to^1_{Rel}$ over $\mathcal{T}_\Sigma(\mathcal{X})$ is extended to reachability goals in the following way: $G \to^1_{Rel} G'$ if there is an odd i such that $t_i \to^1_{Rel} t'_i$ and for all $j \neq i$ we have $t_j = t'_j$. That is, G and G' differ only in one *subgoal* ($t_i \to t_{i+1}$ vs $t'_i \to t_{i+1}$), but $t_i \to^1_{Rel} t'_i$, so when we rewrite t_i in G to t'_i, we get G'.

Rewrite theories with built-in

In the case of a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ with built-in subtheory (Σ_0, E_0) plus axioms B, and terms t, t' in $\mathcal{H}_\Sigma$, a *reachability problem* is an expression P with form $\bigwedge_{i=1}^n t_i \to t'_i \mid \phi$, with t_i and t'_i in $\mathcal{H}_\Sigma(\mathcal{X})$, for $1 \leq i \leq n$, and $\phi \in QF(\mathcal{X}_0)$. Each expression $t_i \to t'_i$, $1 \leq i \leq n$, is a *subgoal* of P and ϕ is the *reachability formula* of P.

A substitution σ is a solution of a reachability problem $P = \bigwedge_{i=1}^n t_i \to t'_i \mid \phi$ iff $t_i\sigma \to_{R/\mathcal{E}} t'_i\sigma$, for $1 \leq i \leq n$, and $E_0 \vdash \phi\sigma$.

2.3.11 Narrowing

Rewrite theories without built-in

In Section 2.3.5, for a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, where $\mathcal{E} = E \cup B$ with B a set of axioms, we have defined the rewrite theory associated to $(\Sigma, \mathcal{E})$, $\mathcal{R}_E = (\Sigma', B, R_E)$, where for each equation or membership in E

$$t = t' \text{ if } A_1 \wedge \ldots \wedge A_n \qquad t : s \text{ if } A_1 \wedge \ldots \wedge A_n,$$

R_E has a conditional rule of the form

$$t \to t' \text{ if } A'_1 \wedge \ldots \wedge A'_n \qquad t{:}s \to \texttt{tt} \text{ if } A'_1 \wedge \ldots \wedge A'_n$$

where if A_i is $t_i : s_i$ then A'_i is $t_i{:}s_i \to \texttt{tt}$, if A_i is $t_i := t'_i$ then A'_i is $t'_i \to t_i$, and if A_i is $t_i = t'_i$ then A'_i is $eq(t_i, t'_i) \to \texttt{tt}$.

Now, we define two *narrowing* relations that use B-unification instead of B-matching, where we use the symbol $\rightsquigarrow$ instead of $\to$ to distinguish between narrowing and rewrite relations. Given two terms $t, t' \in T_\Sigma(\mathcal{X})$:

- E, B-narrowing

  $t \rightsquigarrow^1_{E,B} t'$ if there is a position $p \in Pos(t)$, an equation $l = r \text{ if } \bigwedge_{i \in I} l_i \to r_i$ in R_E, and a substitution σ such that $t|_p\sigma =_B l\sigma$, $t' = (t[r]_p)\sigma$, and $l_i\sigma \to_{E,B} r_i\sigma$, for all $i \in I$.

- $R(E), B$-narrowing

  $t \rightsquigarrow^1_{R(E),B} t'$ if there is a position $p \in Pos(t)$, a rule $c : l \to r \texttt{ if } C \in R$, and a substitution σ such that $t|_p\sigma =_B l\sigma$, $t' = (t[r]_p)\sigma$, and $C\sigma$ holds under $\to^1_{R(E),B}$, as defined in Section 2.3.6.

We may write the substitution, position, and label of the rule under the $\rightsquigarrow$ symbol if needed. Finding the substitution σ such that $l_i\sigma \to_{E,B} r_i\sigma$, for all $i \in I$, may be accomplished by recursive application of $\rightsquigarrow^1_{E,B}$. Finding the substitution σ such that $C\sigma$ holds under $\to^1_{R(E),B}$ may require a recursive application of both $\rightsquigarrow^1_{E,B}$ and $\rightsquigarrow^1_{R(E),B}$.

When $\rightarrow^1_{R/\mathcal{E}}$ can be imitated with $\rightarrow^1_{ER,B};=_{\mathcal{E}}$, we will use two calculi that implement $\rightsquigarrow^1_{E,B}$ and $\rightsquigarrow^1_{R(E),B}$ to solve any given reachability goal.

In the vending machine example we can prove by rewriting that $x_{\mathtt{St}}\ \$ \rightarrow^1_{R/\mathcal{E}} x_{\mathtt{St}}\ \mathtt{c}$ whatever value $x_{\mathtt{St}}$ is given. Finding out the substitution $\sigma = \{x_{\mathtt{St}} \mapsto q\}$ as a solution of the reachability goal $x_{\mathtt{St}}\ \mathtt{q}\ \mathtt{q}\ \mathtt{q} \rightarrow \mathtt{c}$ requires narrowing with $\rightsquigarrow^1_{R(E),B}$ and $\rightsquigarrow^1_{E,B}$:

$$x_{\mathtt{St}}\ \mathtt{q}\ \mathtt{q}\ \mathtt{q} \underset{\sigma}{\rightsquigarrow^1_{E,B}}\ \$ \rightsquigarrow^1_{R(E),B}\ \mathtt{c}$$

but checking that $\mathtt{q}\ \mathtt{q}\ \mathtt{q}\ \mathtt{q} \rightarrow_{R/\mathcal{E}} \mathtt{c}$ is done using rewriting:

$$\mathtt{q}\ \mathtt{q}\ \mathtt{q}\ \mathtt{q} =_{\mathcal{E}} \$ \rightarrow^1_{R(\mathcal{E}),B}\ \mathtt{c}$$

Rewrite theories with built-in

Given a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ with built-in subtheory (Σ_0, E_0) and axioms B, two terms $t, t' \in T_\Sigma(\mathcal{X})$, and a rule $c : l \rightarrow r \ \textit{if}\ \bigwedge_{j=1}^m l_j \rightarrow r_j \mid \phi$ in R, with normal form $c^\circ : l^\circ \rightarrow r \ \textit{if}\ \bigwedge_{j=1}^m l_j \rightarrow r_j \mid \phi \wedge \phi'$ properly renamed so that $vars(c^\circ) \cap vars(t) = \emptyset$, if there exists a non-variable position p in $Pos_{\Sigma_1}(t)$, and a substitution σ such that $rep(t|_p)\sigma =_B l^\circ\sigma$, $t' = (t[r]_p)\sigma$, $l_j\sigma \rightarrow_{R,B} r_j\sigma$, for $1 \leq j \leq m$, and $E_0 \vdash (\phi \wedge \phi')\sigma$, then we write $t \rightsquigarrow^1_{p,c,\sigma} t'$ and say that there is a *narrowing step* from t to t'.

In the expression $P \rightsquigarrow^1_{p,c,\sigma} P'$, it is admitted to omit any part of the subscript when it is not relevant to the discussed matter.

2.4 Strategies

In this section we present the combinators of a strategy language suitable for narrowing, which is a subset of the Maude strategy language [MOMV04,EMOMV07,RMPV21], and a set-theoretic semantics for the language which is based in the construction of a *closed proof tree* for a given *goal*.

A *call strategy* is a name given to a strategy to simplify the development of more complex strategies. A *call strategy definition* is a user-defined association of a strategy to one call strategy.

For any rewrite theory with built-in $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, with $\mathcal{E} = E_0 \cup B$, and set of call strategy definitions for $\mathcal{R}$, written $Call_{\mathcal{R}}$, there exist an associated set of *derivation rules*, written $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$, and an associated set of strategies, written $Strat_{\mathcal{R},Call_{\mathcal{R}}}$, that are defined in Section 2.4.2 and will be used in the following.

2.4.1 Open goal. Closed goal. Derivation rule. Proof tree

An *open goal* has the form $t \rightarrow v/ST$, where t, its *head*, and v are terms in $\mathcal{H}_\Sigma$, and ST is a strategy; a *closed goal* has the form $\frac{}{G}$, with G an open goal.

A derivation rule has the form $\frac{}{G}$ or $\frac{G_1 \cdots G_n}{G}$, where G and each G_i, $1 \leq i \leq n$, are open goals. In either case the *head* of the rule is G.

Given a rewrite theory with built-in $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ and a set of call strategy definitions $Call_{\mathcal{R}}$, *a proof tree* T is inductively defined as either:

- an open or closed goal, G or $\frac{}{G}$, or

- a derivation tree $\frac{T_1 \cdots T_n}{G}$, constructed by application of the derivation rules in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$, where each T_i, $1 \leq i \leq n$, is a proof tree.

The *head* of T is G in all cases. T is said to be *closed* if it has no open goals.

Given any open goal $t \to v/ST$ in a proof tree and a derivation rule with head $t' \to v'/ST$ such that $t =_{\mathcal{E}} t'$ and $v =_{\mathcal{E}} v'$, the application of the rule to the open goal consists in putting the derivation rule in place of the open goal, but replacing t' with t and v' with v anywhere in the derivation rule.

2.4.2 Strategies and their semantics

We present now a general notion of the semantics that defines the result of the application of a strategy to the equivalence class of a term, which is based on the construction of closed proof trees. The full semantics, with all the details, that are cumbersome for an introduction, will be shown in Section 6.4. This semantics is given by a function

$$\_@\_ \ : \ Strat_{\mathcal{R},Call_{\mathcal{R}}} \times \mathcal{T}_{\Sigma_1/\mathcal{E}} \longrightarrow \mathcal{P}(\mathcal{T}_{\Sigma_1/\mathcal{E}}),$$

with $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ and $\mathcal{E} = E_0 \cup B$, where $[v]_{\mathcal{E}}$ is an element of $ST \,@\, [t]_{\mathcal{E}}$ if and only if a closed proof tree (c.p.t. from now on) with head $t \to v/ST$ can be constructed using the derivation rules in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$, also defined below.

`Idle` and `fail`

These are constant strategies that always belong to $Strat_{\mathcal{R},Call_{\mathcal{R}}}$. While the first always succeeds, the second always fails. For each equivalence class $[t]_{\mathcal{E}} \in \mathcal{T}_{\Sigma_1/\mathcal{E}}$, $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$ has a derivation rule

$$\frac{}{t \to t/\texttt{idle}}$$

There are no derivation rules for `fail`. Then, $\texttt{idle} \,@\, [t]_{\mathcal{E}} = \{[t]_{\mathcal{E}}\}$ and $\texttt{fail} \,@\, [t]_{\mathcal{E}} = \emptyset$.

Example 5. *Suppose that $t =_{\mathcal{E}} v$ and we have the open goal $t \to v/\texttt{idle}$ in a derivation tree $\frac{t \to v/\texttt{idle} \ T_2 \cdots T_n}{G}$. There is a term t' and a derivation rule $\frac{}{t' \to t'/\texttt{idle}}$ in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$ such that $t =_{\mathcal{E}} t'$. As $t =_{\mathcal{E}} v$ then also $v =_{\mathcal{E}} t'$, so we can apply this rule to the open goal. Thus, we replace the first t' in the rule with t and the second one with v, yielding $\frac{}{t \to v/\texttt{idle}}$, a closed goal that we put in place of the open goal, so the derivation tree becomes $\frac{\frac{}{t \to v/\texttt{idle}} \ T_2 \cdots T_n}{G}$.*

In particular, if the derivation tree is just $t \to v/\texttt{idle}$ then we get the c.p.t. $\frac{}{t \to v/\texttt{idle}}$, so $[v]_{\mathcal{E}} \in \texttt{idle} \,@\, [t]_{\mathcal{E}}$. The result $[v]_{\mathcal{E}} \in \texttt{idle} \,@\, [t]_{\mathcal{E}}$ was expected, since $\texttt{idle} \,@\, [t]_{\mathcal{E}} = \{[t]_{\mathcal{E}}\}$ and $t =_{\mathcal{E}} v$ implies $[v]_{\mathcal{E}} = [t]_{\mathcal{E}}$.

Rule application

A rule of R that has no rewrite conditions and a substitution form a *rule application*. If $c : l \to r \ \mathit{if} \ \psi$ is a rule in R, and γ is a substitution, then $c[\gamma]$ is a rule application in $Strat_{\mathcal{R},Call_{\mathcal{R}}}$. For each pair of terms t, v in $\mathcal{H}_\Sigma$ such that $ls(t) \equiv_{\leq} ls(v)$, if $t \underset{c\gamma \ R}{\longrightarrow}^{1} v$ then $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$ has a derivation rule

$$\frac{}{t \to v/c[\gamma]}$$

For rules with rewrite conditions, a strategy must be supplied for each rewrite condition. If $c : l \to r \ \mathit{if} \ \bigwedge_{j=1}^{m} l_j \to r_j \mid \psi$ is a rule in R, γ is a substitution, and $\overline{ST} = ST_1, \ldots, ST_m$ is a list of strategies, then $c[\gamma]\{\overline{ST}\}$ is a rule application in $\mathit{Strat}_{\mathcal{R},\mathit{Call}_\mathcal{R}}$.

For each substitution δ such that $E_0 \vdash \psi\gamma\delta$, each term u in $\mathcal{H}_\Sigma$, and each position p in $\mathit{pos}(u)$ such that $u|_p = l\gamma\delta$, $\mathcal{D}_{\mathcal{R},\mathit{Call}_\mathcal{R}}$ has a derivation rule

$$\frac{l_1\gamma\delta \to r_1\gamma\delta/ST_1\delta \cdots l_m\gamma\delta \to r_m\gamma\delta/ST_m\delta}{u \to u[r\gamma\delta]_p/c[\gamma]\{\overline{ST}\}}$$

Top

It is possible to restrict the application of a rule in R only to the top of the term. This is useful for structural rules, that are applied to the whole state.

If $c : l \to r \ \mathit{if} \ \psi$ is a rule in R and γ is a substitution, then $\mathtt{top}(c[\gamma])$ is a strategy in $\mathit{Strat}_{\mathcal{R},\mathit{Call}_\mathcal{R}}$. For each substitution δ such that $E_0 \vdash \psi\gamma\delta$, $\mathcal{D}_{\mathcal{R},\mathit{Call}_\mathcal{R}}$ has a derivation rule

$$\frac{}{l\gamma\delta \to r\gamma\delta/\mathtt{top}(c[\gamma])}$$

If $c : l \to r \ \mathit{if} \ \bigwedge_{j=1}^{m} l_j \to r_j \mid \psi$ is a rule in R, γ is a substitution, and $\overline{ST} = ST_1, \ldots, ST_m$ is a list of strategies, then $\mathtt{top}(c[\gamma]\{\overline{ST}\})$ is a strategy in $\mathit{Strat}_{\mathcal{R},\mathit{Call}_\mathcal{R}}$. For each substitution δ such that $E_0 \vdash \psi\gamma\delta$, $\mathcal{D}_{\mathcal{R},\mathit{Call}_\mathcal{R}}$ has a derivation rule

$$\frac{l_1\gamma\delta \to r_1\gamma\delta/ST_1\delta \cdots l_m\gamma\delta \to r_m\gamma\delta/ST_m\delta}{l\gamma\delta \to r\gamma\delta/\mathtt{top}(c[\gamma]\{\overline{ST}\})}$$

Call strategy

Call strategy definitions allow the use of parameters and the implementation of recursive strategies. A call strategy definition can be either unconditional or conditional, with the forms $\mathtt{sd}\ CS := ST$, $\mathtt{sd}\ CS(\bar{x}) := ST$, or $\mathtt{csd}\ CS(\bar{x}) := ST \ \mathit{if} \ \bigwedge_{j=1}^{m}(l_j = r_j) \wedge \phi$, with CS an alphanumerical name.

The semantics for *call strategy invocations*, given a pair of terms t and v in $\mathcal{H}_\Sigma$ such that $\mathit{ls}(t) \equiv_\leq \mathit{ls}(v)$ is:

- If $\mathtt{sd}\ CS := ST \in \mathit{Call}_\mathcal{R}$ then the call strategy invocation CS is a strategy in $\mathit{Strat}_{\mathcal{R},\mathit{Call}_\mathcal{R}}$, and $\mathcal{D}_{\mathcal{R},\mathit{Call}_\mathcal{R}}$ has a derivation rule

$$\frac{t \to v/ST}{t \to v/CS}$$

- If $\mathtt{sd}\ CS(\bar{x}) := ST \in \mathit{Call}_\mathcal{R}$, where $\bar{x} = x^1_{s_1}, \ldots, x^n_{s_n}$, $t_1, \ldots, t_n$ are terms in $\mathcal{T}_\Sigma(\mathcal{X} \setminus V_{\mathcal{R},\mathit{Call}_\mathcal{R}})$, with sorts $s_1, \ldots, s_n$ respectively, and we denote $\bar{t} = t_1, \ldots, t_n$, then the call strategy invocation $CS(\bar{t})$ is a strategy in $\mathit{Strat}_{\mathcal{R},\mathit{Call}_\mathcal{R}}$. If we denote $\rho = \{\bar{x} \mapsto \bar{t}\}$ then $\mathcal{D}_{\mathcal{R},\mathit{Call}_\mathcal{R}}$ has a derivation rule

$$\frac{t \to v/ST\rho}{t \to v/CS(\bar{t})}$$

- If $\mathtt{csd}\ CS(\bar{x}) := ST \ \mathit{if} \ \bigwedge_{j=1}^{m}(l_j = r_j) \wedge \phi \in \mathit{Call}_\mathcal{R}$, with $\bar{x}$, $\bar{t}$, and ρ as before, then the call strategy invocation $CS(\bar{t})$ is a strategy in $\mathit{Strat}_{\mathcal{R},\mathit{Call}_\mathcal{R}}$. If δ is a substitution such that $l_j\rho\delta =_\mathcal{E} r_j\rho\delta$, for $1 \leq j \leq n$, and $E_0 \vdash \phi\rho\delta$, $\mathcal{D}_{\mathcal{R},\mathit{Call}_\mathcal{R}}$ has a derivation rule

$$\frac{t \to v/ST\rho\delta}{t \to v/CS(\bar{t})}$$

Tests

Tests are strategies that check a property on an equivalence class $[t]_{\mathcal{E}}$ in $\mathcal{T}_{\Sigma_1/\mathcal{E}}$. If the property holds then the test returns a set containing $[t]_{\mathcal{E}}$ as its only element. Otherwise, the test returns the empty set.

For each test strategy $\texttt{match}\, u\, \texttt{s.t.}\, \bigwedge_{j=1}^{m}(l_j = r_j) \wedge \phi$, equivalence class $[t]_{\mathcal{E}}$ in $\mathcal{T}_{\Sigma_1/\mathcal{E}}$, and substitution δ such that $t =_{\mathcal{E}} u\delta$, $l_j\delta =_{\mathcal{E}} r_j\delta$, for $1 \leq j \leq m$, and $E_0 \vdash \phi\delta$, $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$ has a rule

$$\frac{}{t \to t/\texttt{match}\, u\, \texttt{s.t.}\, \bigwedge_{j=1}^{m}(l_j = r_j) \wedge \phi}$$

If-then-else

An if-then-else strategy has the form $\texttt{match}\, u\, \texttt{s.t.}\, \phi\, ?\, ST_1\ :\ ST_2$. For each pair of equivalence classes $[t]_{\mathcal{E}}$ and $[v]_{\mathcal{E}}$ in $\mathcal{T}_{\Sigma_1/\mathcal{E}}$ such that $ls(t) \equiv_{\leq} ls(v)$, and each substitution δ such that $t =_{\mathcal{E}} u\delta$, if $E_0 \vdash \phi\delta$, then $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$ has a rule

$$\frac{t \to v/ST_1\delta}{t \to v/\texttt{match}\, u\, \texttt{s.t.}\, \phi\, ?\, ST_1 : ST_2}$$

and if $E_0 \vdash \neg\phi\delta$ then $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$ has a rule

$$\frac{t \to v/ST_2\delta}{t \to v/\texttt{match}\, u\, \texttt{s.t.}\, \phi\, ?\, ST_1 : ST_2}$$

Regular expressions

Now, we define the concatenation, union, and iteration of strategies. Let ST_1 and ST_2 be strategies, and let t, v, and u be terms in $\mathcal{H}_{\Sigma}$ such that $ls(t) \equiv_{\leq} ls(u) \equiv_{\leq} ls(v)$. Then $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$ has rules

$$\frac{t \to u/ST_1 \quad u \to v/ST_2}{t \to v/ST_1\, ;\, ST_2}$$

$$\frac{t \to v/ST_1}{t \to v/ST_1 \mid ST_2}$$

$$\frac{t \to v/ST_2}{t \to v/ST_1 \mid ST_2}$$

$$\frac{t \to v/ST}{t \to v/ST+}$$

$$\frac{t \to v/ST\, ;\, ST+}{t \to v/ST+}$$

Of course, $ST*$ can be defined as $\texttt{idle} \mid ST+$.

Rewriting of subterms

The `matchrew` combinator allows the selection of a subterm to apply a rule and extends the scope of the substitution that validates a test strategy to subsequent steps of the execution path.

`Matchrew` strategies have the form

$$MS = \texttt{matchrew}\ u\ \texttt{s.t.}\ \bigwedge_{j=1}^{m}(l_j = r_j) \wedge \phi\ \texttt{by}\ x^1_{s_1}\ \texttt{using}\ ST_1, \ldots, x^n_{s_n}\ \texttt{using}\ ST_n$$

where $u = u[x^1_{s_1}, \ldots, x^n_{s_n}]_{p_1,\ldots,p_n}$, for appropriate $p_1, \ldots, p_n$. For each n-tuple $(t_1, \ldots, t_n)$ of terms in $\mathcal{T}^n_{\Sigma_1}$ such that $ls(\bar{t}) \leq \bar{s}$, and each substitution δ such that $u\delta \in \mathcal{T}_\Sigma$, $\{l_j\delta, r_j\delta\}_{j=1}^m \subset \mathcal{T}_\Sigma$, $\bar{l}\delta =_\mathcal{E} \bar{r}\delta$, $\phi\delta$ is ground, and $E_0 \vdash \phi\delta$, $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$ has a derivation rule

$$\frac{x^1_{s_1}\delta \to t_1/ST_1\delta \cdots x^n_{s_n}\delta \to t_n/ST_n\delta}{u\delta \to u\delta[t_1, \ldots, t_n]_{p_1,\ldots,p_n}/MS}$$

2.5 Maude

Maude is a high-level language and high-performance system supporting both equational and rewriting computation [CDE+02]. Maude's underlying equational logic is membership equational logic, which is an improvement over order-sorted algebra, allowing the faithful specification of types (like sorted lists or search trees) whose data are defined not only by means of constructors, but also by the satisfaction of additional properties [BM06]. Maude has three kinds of *modules* that are of interest for our purpose:

- *Functional modules* provide support for functional programming in membership equational logic.
- *System modules* allow the specification of concurrent systems, when used as a semantic framework, or deductive systems, when used as a logical framework, using rewriting logic.
- *Strategy modules* allow the separation between the rules that specify a system and the way that these rules are applied, restricting the reachable states from an initial state. They can be used both to implement and test different algorithms over a given specification or to drive the search of solutions to reachability problems.

Moreover, Maude makes a systematic and efficient use of reflection, where programs are represented as data, allowing metaprogramming and metalanguage applications, as well as extensions to the language itself.

The site https://maude.cs.illinois.edu/w/index.php/The_Maude_System holds the most up-to-date information of all the existing releases. The most in-depth coverage of the Maude system can be found in the book *All about Maude* [CDE+07].

2.5.1 Functional modules

Maude's functional modules allow the specification and execution of MEL theories $(\Sigma, E \cup B)$, where B is a set of equational axioms (usually commutativity, associativity and/or identity) for some of the operators in the signature, and E is a set of equations that are valid modulo B, as long as they are executable in the sense defined in Section 2.3.8, that is, we can always rewrite a term t to its canonical form $t{\downarrow_{E/B}}$ using the associated rewrite theory of the MEL theory.

We show the syntax of functional modules through the vending machine example:

```
fmod VENDING-MACHINE-EQ is
  sorts Coin Item State .
  subsorts Coin Item < State .
  op __ : State State -> State [assoc comm] .
  op $ : -> Coin .
  op q : -> Coin .
  op a : -> Item .
  op c : -> Item .
  eq q q q q = $ .
endfm
```

A functional module begins with the reserved word `fmod` and ends with the reserved word `endfm`. We declare sorts using the reserved word `sort`. Kinds are not defined in an explicit way. We refer to the kind of a sort s as $[s]$. Sort ordering, which as we saw in Section 2.2.2 is a shortcut for certain membership axioms, is defined using the reserved word `subsort`.

Functions are declared using the reserved word `op` followed by the name of the function (which can be empty), the sort of the arguments and the sort of the result. The position of the arguments is determined by the underscore "`_`" symbol that appears in the definition. The symbol `->` separates the input arguments from the result. If no sort is found to the left of `->` then the function is a constant. If no underscore symbol appears then the standard syntax for functions, with the arguments surrounded by brackets, is used. Axioms from B and other properties of the function are declared writing them between square brackets. In our example, the `State` constructor definition, which has empty name, has associative (`assoc`) and commutative (`comm`) properties, as expected for a non-empty multiset.

Equations are declared using the reserved words `eq`, or `ceq` when declaring conditional equations. Similarly, membership axioms are declared using the reserved words `mb` and `cmb`.

We can use Maude's command `reduce` to compute the canonical form of any term. For instance:

```
Maude> reduce q q q q .
result Coin: $
```

Maude does not check confluence and termination properties for functional modules: the user is responsible for providing them. However, in some cases it is possible to check these properties with Maude's Church-Rosser checker and termination tools [DM10, DLM+08].

2.5.2 System modules

System modules allow the specification of executable rewrite theories in the sense defined in [CDE+07]. The syntax is shown with our vending machine example:

```
mod VENDING-MACHINE is
  protecting VENDING-MACHINE-EQ .
  var M : State .
  rl [buy-coffee] : $ => c .
  rl [buy-apple] : $ => a q .
```

```
  rl [add-quarter] : M => M q .
  rl [add-dollar] : M => M $ .
endm
```

Reserved words: the module begins with `mod` and ends with `endm`. The reserved word `including` indicates that we are going to use another module within this one, the functional module `VENDING-MACHINE-EQ` in this case. Variables are declared using the reserved word `var`. They can also be used without previous declaration by writing their name, a colon and its sort (for instance `M:State`). (`crl`) `rl` declares (conditional) rewrite rules. Rules and equations can be labeled by writing the label between square brackets.

Maude's command `rewrite` rewrites any term with the existing rules (`[1]` means one rewrite step):

```
Maude> rewrite [1] q q q q .
result Item: c
```

Prior to rewriting, Maude always reduces terms to canonical form, in this case `$`. Then Maude applies the rule `buy-coffee` and returns the answer `c`. Although the answer is not unique, Maude always returns only one. We can choose the rule to apply using Maude's command `srewrite`, that is explained in the next Section, with the label of the desired rule:

```
Maude> srewrite q q q q using buy-apple .
result State: q a
```

2.5.3 Strategy modules

Maude's strategy handling, and its related strategy modules, are available since version 3 of the Maude system. The strategy modules allow the definition of call strategies. A strategy module begins with the reserved word `smod` and ends with the reserved word `endsm`.

```
smod EXAMPLE-STRAT is
  protecting INT .
  sort State .

  vars N Y OK : Int .

  op _/_/_ : Int Int Int -> State .

  strat testBelow30 : @ State .
  sd testBelow30 := match N / Y / OK s.t. Y < 30 .
endsm
```

Here, a call strategy named `testBelow30`, with no parameters, that applies to terms with sort `State` is defined. The strategy is a test that only succeeds for those terms with sort `State` whose second value is lower than 30. The sort `Int` is defined in Maude in the functional module `INT` included in the `prelude.maude` file, and imported to the `EXAMPLE-STRAT` strategy module through the `protecting INT` declaration.

Maude's command `srewrite` attempts to apply the strategy provided in the command to the given term:

```
Maude> srewrite 4 / 30 / 1 using testBelow30 .
No solution.

Maude> srewrite 4 / 20 / 1 using testBelow30 .
result State: 4 / 20 / 1

Maude> srewrite 4 / 30 / 1 using idle .
result State: 4 / 30 / 1
```

In the first case, Maude fails to rewrite the term, since its second value, 30, is not lower than 30, so `testBelow30` fails, which it does not in the second case, where the second value of the term is 20. In the last case, the `idle` strategy applies to any term, even with second value 30, returning the same term.

2.5.4 The metalevel

Maude's reflective capabilities are supported through the functional module `META-LEVEL` where each of Maude's reserved words has a corresponding sort (`Fmodule`, `Term`, ...). The module has several functions (`upModule`, `metaUnify`, `leastSort`, ...) that are used in the implementation of the different calculi in this dissertation. One of the most important uses of the metalevel are the unification algorithms, which are *theory-dependent*, since a different order-sorted unification algorithm is derived for each signature Σ and combination of axioms B, so the `metaUnify` command needs both as parameters (see [CDE+07, Chapter 15] and [CDE+23, Chapter 17]).

Chapter 3

First calculus for conditional narrowing modulo

The first calculus [AMPP14], developed at the beginning of the investigation for this dissertation, is the simplest one; we needed to know if conditional narrowing modulo is indeed feasible. The calculus requires the treatment of the equational and membership conditions in both the equational and the rewrite theory as reachability conditions to be solved by using the equations and membership predicates in the equational theory as oriented rules. It includes one minor attempt to reduce the space state by looking into the least sort of each term and stopping when that sort is bigger than the desired one in the computation.

3.1 Tower of Hanoi specification

The Tower of Hanoi puzzle is used as a running example for this calculus. In Maude, we declare it as follows:

```
mod HANOI is
  sorts Boolean Rod Disk Tower ValidTower Pair State .
  subsort Rod < ValidTower < Tower State .

  ops 1 2 3 4 : -> Disk .
  op t : -> Boolean .
  op _<_ : Disk Disk -> Boolean .
  ops a b c : -> Rod .
  op __ : Disk Tower -> Tower .
  op _-_ : Tower Tower -> Pair [comm] .
  op move(_) : Pair -> Pair .
  op _,_ : State State -> State [assoc comm] .

  var R : Rod .
  vars X Y : Disk .
  vars T T' : Tower .
  vars D E F G : ValidTower .

  mb X R : ValidTower .
```

```
cmb (X Y T) : ValidTower if (X < Y) = t /\ (Y T) : ValidTower .

eq 1 < 2 = t . eq 1 < 3 = t . eq 1 < 4 = t .
eq 2 < 3 = t . eq 2 < 4 = t . eq 3 < 4 = t .

eq move (X T - R) = (T - X R) .
ceq move(X T - Y T') = (T - X Y T') if (X < Y) = t .

crl D, E => F, G if (F - G) := move(D - E) .
```

endm

The way that Tower of Hanoi is played is defined in the memberships, conditional equation, and conditional rule:

- `subsort Rod < ValidTower`

  states that an empty `Tower`, i.e., one containing only a `Rod`, is a `ValidTower`,

- `mb X R : ValidTower`

  states that a single `Disk` in a `Tower` is a `ValidTower`,

- `cmb (X Y T) : ValidTower if (X < Y) = t /\ (Y T) : ValidTower`

  recursively defines a `ValidTower` as any `Tower` with two or more `Disks` on it where the first two `Disks` are in the right order and if we take off the upper `Disk`, the resulting `Tower` is also a `ValidTower`,

- `eq move (X T - R) = (T - X R)`

  allows the movement of one `Disk` from one `Tower` to a `Rod` (empty `Tower`),

- `ceq move(X T - Y T') = (T - X Y T') if (X < Y) = t`

  allows the movement of one `Disk X` from one `Tower` to another if the top `Disk`, `Y`, of the `Tower` where the `Disk` is placed is bigger than `X`,

- `crl D, E => F, G if (F - G) := move(D - E)`

  is the only rule in the module. We can evolve from one `State` to another `State` only by the application of the function `move` to any pair of `ValidTowers` in the initial `State` and the result is another pair of `ValidTowers`.

3.1.1 Signature

In the Tower of Hanoi specification, $\Sigma = (K, S, F)$ is:

- $K = \{[\texttt{TowerState}], [\texttt{Pair}], [\texttt{Disk}], [\texttt{Boolean}]\}$,
- $S = \{S_{[\texttt{TowerState}]}, S_{[\texttt{Pair}]}, S_{[\texttt{Disk}]}, S_{[\texttt{Boolean}]}\}$, where
  $S_{[\texttt{TowerState}]} = \{\texttt{Rod}, \texttt{ValidTower}, \texttt{Tower}, \texttt{State}\}$, $S_{[\texttt{Pair}]} = \{\texttt{Pair}\}$, $S_{[\texttt{Disk}]} = \{\texttt{Disk}\}$, $S_{[\texttt{Boolean}]} = \{\texttt{Boolean}\}$.
- $F = \{\{\cdot\}_{\texttt{Disk TowerState,TowerState}}, \{,\}_{\texttt{TowerState TowerState,TowerState}}, \{-\}_{\texttt{TowerState TowerState,Pair}}, \{\texttt{move}\}_{\texttt{Pair,Pair}}, \{<\}_{\texttt{Disk Disk,Boolean}}, \{\texttt{a}, \texttt{b}, \texttt{c}\}_{\texttt{Rod}}, \{1, 2, 3, 4\}_{\texttt{Disk}}, \{\texttt{t}\}_{\texttt{Boolean}}\}$,

3.1.2 MEL theory

We use some shortcuts for the running example: we write `V`, `D`, `R`, `T`, `P`, and `S` instead of `ValidTower`, `Disk`, `Rod`, `Tower`, `Pair`, and `State`, respectively, and [`TS`] for the kind of `Rod`, `ValidTower`, `Tower`, and `State`.

The MEL theory $(\Sigma, \mathcal{E})$ for the Tower of Hanoi puzzle consists of $\Sigma = (K, S, F)$ and $\mathcal{E}$ is the following set of MEL sentences, where it is shown the relation between the Maude code (each one of the bullets that follow) and the MEL sentences (the ones below each bullet):

- `subsort Rod < ValidTower < Tower State`
  $x_{[\mathtt{TS}]} : \mathtt{S}$ *if* $x_{[\mathtt{TS}]} : \mathtt{V}$ $\quad x_{[\mathtt{TS}]} : \mathtt{T}$ *if* $x_{[\mathtt{TS}]} : \mathtt{V}$
  $x_{[\mathtt{TS}]} : \mathtt{S}$ *if* $x_{[\mathtt{TS}]} : \mathtt{R}$ $\quad x_{[\mathtt{TS}]} : \mathtt{T}$ *if* $x_{[\mathtt{TS}]} : \mathtt{R}$
  $x_{[\mathtt{TS}]} : \mathtt{V}$ *if* $x_{[\mathtt{TS}]} : \mathtt{R}$

- `ops 1 2 3 4 : -> Disk`
  $1 : \mathtt{D}$
  $2 : \mathtt{D}$
  $3 : \mathtt{D}$
  $4 : \mathtt{D}$

- `op t : -> Boolean`
  $\mathtt{t} : \mathtt{Boolean}$

- `ops a b c : -> Rod`
  $\mathtt{a} : \mathtt{R}$
  $\mathtt{b} : \mathtt{R}$
  $\mathtt{c} : \mathtt{R}$

- `op __ : Disk Tower -> Tower`
  $x_{[\mathtt{D}]} Y_{[\mathtt{TS}]} : \mathtt{T}$ *if* $X_{[\mathtt{D}]} : \mathtt{D} \wedge Y_{[\mathtt{TS}]} : \mathtt{T}$

- `op _,_ : State State -> State [assoc comm]`
  $x_{[\mathtt{TS}]}, Y_{[\mathtt{TS}]} : \mathtt{S}$ *if* $X_{[\mathtt{TS}]} : \mathtt{S} \wedge Y_{[\mathtt{TS}]} : \mathtt{S}$
  $x_{[\mathtt{TS}]}, Y_{[\mathtt{TS}]} = Y_{[\mathtt{TS}]}, X_{[\mathtt{TS}]}$ (commutativity)
  $(X_{[\mathtt{TS}]}, Y_{[\mathtt{TS}]}), Z_{[\mathtt{TS}]} = X_{[\mathtt{TS}]}, (Y_{[\mathtt{TS}]}, Z_{[\mathtt{TS}]})$ (associativity)

- `op _-_ : Tower Tower -> Pair [comm]`
  $x_{[\mathtt{TS}]} - Y_{[\mathtt{TS}]} : \mathtt{P}$ *if* $X_{[\mathtt{TS}]} : \mathtt{T} \wedge Y_{[\mathtt{TS}]} : \mathtt{T}$
  $x_{[\mathtt{TS}]} - Y_{[\mathtt{TS}]} = Y_{[\mathtt{TS}]} - X_{[\mathtt{TS}]}$ (commutativity)

- `op _<_ : Disk Disk -> Boolean`
  $x_{[\mathtt{D}]} < Y_{[\mathtt{D}]} : \mathtt{Boolean}$ *if* $X_{[\mathtt{D}]} : \mathtt{D} \wedge Y_{[\mathtt{D}]} : \mathtt{D}$

- `op move(_) : Pair -> Pair`
  $\mathtt{move}(x_{[\mathtt{P}]}) : \mathtt{P}$ *if* $x_{[\mathtt{P}]} : \mathtt{P}$

- `mb X R : ValidTower`
  $x_{[\mathtt{D}]} R_{[\mathtt{TS}]} : \mathtt{V}$ *if* $X_{[\mathtt{D}]} : \mathtt{D} \wedge R_{[\mathtt{TS}]} : \mathtt{R}$

- `cmb (X Y T) : ValidTower if (X < Y) = t /\ (Y T) : ValidTower`
  $x_{[\mathtt{D}]} Y_{[\mathtt{D}]} T_{[\mathtt{TS}]} : \mathtt{V}$ *if* $X_{[\mathtt{D}]} < Y_{[\mathtt{D}]} = \mathtt{t} \wedge Y_{[\mathtt{D}]} T_{[\mathtt{TS}]} : \mathtt{V}$

- `eq 1 < 2 = t`
  $\texttt{1} < \texttt{2} = \texttt{t}$

  ...

- `eq 3 < 4 = t`
  $\texttt{3} < \texttt{4} = \texttt{t}$

- `eq move (X T - R) = (T - X R)`
  $\texttt{move}(X_{[\texttt{D}]}T_{[\texttt{TS}]} - R_{[\texttt{TS}]}) = T_{[\texttt{TS}]} - X_{[\texttt{D}]}R_{[\texttt{TS}]}$ *if* $X_{[\texttt{D}]} : \texttt{D} \wedge T_{[\texttt{TS}]} : \texttt{T} \wedge R_{[\texttt{TS}]} : \texttt{R}$

- `ceq move(X T - Y T') = (T - X Y T') if (X < Y) = t`
  $\texttt{move}(X_{[\texttt{D}]}T_{[\texttt{TS}]} - Y_{[\texttt{D}]}T'_{[\texttt{TS}]}) = T_{[\texttt{TS}]} - X_{[\texttt{D}]}Y_{[\texttt{D}]}T'_{[\texttt{TS}]}$ *if* $X_{[\texttt{D}]} < Y_{[\texttt{D}]} = \texttt{t} \wedge T_{[\texttt{TS}]} : \texttt{T} \wedge T'_{[\texttt{TS}]} : \texttt{T}$

3.1.3 Associated rewrite theory

The associated rewrite theory for the Tower of Hanoi puzzle is $\mathcal{R}_E = (\Sigma', B, R_E)$, where:

- Σ' is Σ with the addition of the new connected component with sort `Truth` and the new function symbols:
  $\texttt{tt} : \rightarrow \texttt{Truth}$,
  $\texttt{\_:B} : [\texttt{B}] \rightarrow \texttt{Truth}$,
  $\texttt{\_:R} : [\texttt{TS}] \rightarrow \texttt{Truth}$,
  $\texttt{\_:D} : [\texttt{D}] \rightarrow \texttt{Truth}$,
  $\texttt{\_:T} : [\texttt{TS}] \rightarrow \texttt{Truth}$,
  $\texttt{\_:V} : [\texttt{TS}] \rightarrow \texttt{Truth}$,
  $\texttt{\_:P} : [\texttt{P}] \rightarrow \texttt{Truth}$,
  $\texttt{\_:S} : [\texttt{TS}] \rightarrow \texttt{Truth}$,
  $\texttt{eq} : [\texttt{TS}]\ [\texttt{TS}] \rightarrow \texttt{Truth}$,
  $\texttt{eq} : [\texttt{P}]\ [\texttt{P}] \rightarrow \texttt{Truth}$,
  $\texttt{eq} : [\texttt{D}]\ [\texttt{D}] \rightarrow \texttt{Truth}$,
  $\texttt{eq} : [\texttt{B}]\ [\texttt{B}] \rightarrow \texttt{Truth}$.

- We show here only an excerpt of the rules in R_E:
  $\texttt{eq}(x_{[\texttt{TS}]}, x_{[\texttt{TS}]}) \rightarrow \texttt{tt}$
  $x_{[\texttt{TS}]}\texttt{:S} \rightarrow \texttt{tt}$ *if* $x_{[\texttt{TS}]}\texttt{:V} \rightarrow \texttt{tt}$
  $\texttt{1:D} \rightarrow \texttt{tt}$
  $\texttt{a:R} \rightarrow \texttt{tt}$
  $\texttt{t:B} \rightarrow \texttt{tt}$
  $X_{[\texttt{D}]}Y_{[\texttt{TS}]}\texttt{:T} \rightarrow \texttt{tt}$ *if* $X_{[\texttt{D}]}\texttt{:D} \rightarrow \texttt{tt} \wedge Y_{[\texttt{TS}]}\texttt{:T} \rightarrow \texttt{tt}$
  $x_{[\texttt{TS}]}, Y_{[\texttt{TS}]}\texttt{:S} \rightarrow \texttt{tt}$ *if* $X_{[\texttt{TS}]}\texttt{:S} \rightarrow \texttt{tt} \wedge Y_{[\texttt{TS}]}\texttt{:S} \rightarrow \texttt{tt}$
  $x_{[\texttt{TS}]}, Y_{[\texttt{TS}]} \rightarrow Y_{[\texttt{TS}]}, X_{[\texttt{TS}]}$
  $(X_{[\texttt{TS}]}, Y_{[\texttt{TS}]}), Z_{[\texttt{TS}]} \rightarrow X_{[\texttt{TS}]}, (Y_{[\texttt{TS}]}, Z_{[\texttt{TS}]})$
  $X_{[\texttt{D}]}Y_{[\texttt{D}]}T_{[\texttt{TS}]}\texttt{:V} \rightarrow \texttt{tt}$ *if* $\texttt{eq}(X_{[\texttt{D}]} < Y_{[\texttt{D}]}, \texttt{t}) \rightarrow \texttt{tt} \wedge Y_{[\texttt{D}]}T_{[\texttt{TS}]}\texttt{:V} \rightarrow \texttt{tt}$
  $\texttt{1} < \texttt{2} \rightarrow \texttt{t}$.

3.1.4 Rewrite theory

The rewrite theory without built-in $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ for the Tower of Hanoi puzzle has only one rule in R that translates the rule `crl D, E => F, G if (F - G) := move(D - E)` in the Maude module:

$D_{[\mathtt{TS}]}, E_{[\mathtt{TS}]} => F_{[\mathtt{TS}]}, G_{[\mathtt{TS}]}$ *if*
$F_{[\mathtt{TS}]} - G_{[\mathtt{TS}]} := \mathtt{move}(D_{[\mathtt{TS}]} - E_{[\mathtt{TS}]}) \wedge D_{[\mathtt{TS}]} : \mathtt{V} \wedge E_{[\mathtt{TS}]} : \mathtt{V} \wedge F_{[\mathtt{TS}]} : \mathtt{V} \wedge G_{[\mathtt{TS}]} : \mathtt{V}$

The rewrite theory for the Tower of Hanoi puzzle is executable if we define the set B to contain the associative and the commutative equations in the equational theory, the set E to contain the rest of equations and all memberships in the equational theory, and we add to R the following rule, that ensures $\mathcal{E}$-consistency:
$D_{[\mathtt{TS}]}, E_{[\mathtt{TS}]}, S_{[\mathtt{TS}]} => F_{[\mathtt{TS}]}, G_{[\mathtt{TS}]}, S_{[\mathtt{TS}]}$ *if*
$F_{[\mathtt{TS}]} - G_{[\mathtt{TS}]} := \mathtt{move}(D_{[\mathtt{TS}]} - E_{[\mathtt{TS}]}) \wedge D_{[\mathtt{TS}]} : \mathtt{V} \wedge E_{[\mathtt{TS}]} : \mathtt{V} \wedge S_{[\mathtt{TS}]} : \mathtt{S} \wedge F_{[\mathtt{TS}]} : \mathtt{V} \wedge G_{[\mathtt{TS}]} : \mathtt{V}$

Example 6. *In the Tower of Hanoi example, without the rule added for $\mathcal{E}$-consistency, from the* State $S = (\mathtt{1A}, \mathtt{B}), \mathtt{2C}$ *we can only reach in* $\rightarrow^1_{R(E),B}$ *the* State $(\mathtt{A}, \mathtt{1B}), \mathtt{2C}$ *and vice versa, since the only match with the head $D_{[\mathtt{TS}]}, E_{[\mathtt{TS}]}$ of the only rule in R is the subterm between parentheses in both cases, so we can only move* Disk 1 *from* Rod A *to* Rod B.

By adding the new rule to R, its head $D_{[\mathtt{TS}]}, E_{[\mathtt{TS}]}, S_{[\mathtt{TS}]}$ now matches S in other ways. For instance, $S \rightarrow^1_{R(E),B} (\mathtt{A}, \mathtt{12C}), \mathtt{B}$, *with* $\sigma = \{D_{[\mathtt{TS}]} \mapsto \mathtt{1A}, E_{[\mathtt{TS}]} \mapsto \mathtt{2C}, S_{[\mathtt{TS}]} \mapsto \mathtt{B}\}$, *since* $S =_B (\mathtt{1A}, \mathtt{2C}), \mathtt{B}$ *and* $\mathtt{move}(\mathtt{1A} - \mathtt{2C}) = \mathtt{A} - \mathtt{12C}$.

The introduction of this rule is explained in detail in Section 4.2.1 with the concept of *closure under B-extensions*. In general we will assume, unless otherwise stated, that the required rules for $\mathcal{E}$-consistency are already in R.

3.2 Unification by conditional narrowing modulo

Narrowing allows us to assign values to variables in such a way that a reachability goal holds. We implement narrowing using a calculus that given a reachability goal G computes a set (possibly infinite) of *answers* for G, with the following properties:

1. If σ is an $R/\mathcal{E}$-normalized idempotent solution for a reachability goal G, the calculus will compute a more general answer σ' for G, i.e., $\sigma \ll_{\mathcal{E}} \sigma'$.

2. If the calculus computes an answer σ for G, then σ is a solution for G.

That is, we want to compute a complete set of answers for G, a set that includes a generalization of any possible solution for G, with respect to $R/\mathcal{E}$-normalized substitutions.

We are going to split this task into two subtasks: first we will solve the part of the calculus that deals with unification; second, we will solve the part that deals with reachability.

3.2.1 Calculus rules for unification

From now on in this chapter, we assume $(\Sigma, \mathcal{E})$ with $\mathcal{E} = E \cup B$ to be an executable Mel theory, where we have a complete B-unification algorithm that returns a (possibly infinite) *CSU* for any pair of terms.

A *system of equations* F is a conjunction of the form $u_1 = v_1 \wedge \ldots \wedge u_n = v_n$ where for $1 \leq i \leq n$, $u_i = v_i$ is a Σ-equation. A unification equation is an expression $u{:}s = v{:}t$, with $[ls(u)] = [s] = [ls(v)] = [t]$, which is a shorthand for the system of equations $u = v \wedge u = x_s \wedge v = y_t$ (we will also write $u = v$, $u{:}s$, $v{:}t$). This means that we intend to unify u and v, with resulting types s and t respectively. A unification goal is a sequence

(understood as conjunction) of unification equations, i.e., unification goals are systems of equations with a specific format. A substitution σ is a solution of a system of equations F if $\mathcal{E} \vdash A\sigma$ for each Σ-equation A in F.

Admissible goals, or simply goals, are any sequence of $u{:}s{=}v{:}t$, $u{:}s{:=}v{:}t$, $u{:}s{\to}v{:}t$, $u{:}s{\to}^1 v{:}t$ and $v{:}t$.

Given two types κ and κ', we call $glbSorts(\kappa, \kappa')$ to the set of maximal types $\{\kappa_i\}_{i\in I}$ such that $\kappa_i \leq \kappa$ and $\kappa_i \leq \kappa'$, for $i \in I$.

Our calculus is defined by the set of inference rules in Figures 3.1 and 3.2, where given a MEL sentence S *if* c, we call $\hat{c}$ to the unification goal whose elements are obtained from c by keeping the membership conditions and turning any equational condition of the form $u{=}v$ or $u{:=}v$ into a unification goal $u{:}ls(u){=}v{:}ls(v)$ or $u{:}ls(u){:=}v{:}ls(v)$.

The first two rules, $[u]$ and $[x]$, transform *equational* problems into *rewriting* problems modulo axioms; rule $[u]$ tries to solve a single goal by narrowing both terms in the goal to unifiable terms modulo B, using the inference rules; rule $[n]$ describes one step of narrowing for unification where the conditions on the applied rule are turned into subgoals and the instantiated right side of the rule ($r\theta$) is required to have a sort which is a common subsort of S and T; rule $[t]$ allows us to apply several steps of narrowing for unification; rule $[i]$ decomposes a term allowing rule $[n]$ to be applied to any subterm of it; rule $[r]$ allows instantiation of variables on unifiable terms; rule $[m1]$ solves the membership problem for variables, and rules $[s]$ and $[m2]$ for the rest of terms, using the membership conditions in E.

When we apply one of the calculus rules for unification to a unification problem G_i with some inference rule $[z]$ and substitution σ_i, yielding another unification problem G_{i+1}, we display it as $G_i \rightsquigarrow_{[z],\sigma_i} G_{i+1}$ and say that there exists a *narrowing step* from G_i to G_{i+1} using the substitution σ_i and the inference rule $[z]$. $[z]$ and σ_i may be omitted when their actual values are irrelevant or can be inferred.

From a unification goal G a derivation is made applying rules of the calculus, generating a *narrowing path*. If the narrowing path ends in the empty goal, denoted by $\square$ and written $G \rightsquigarrow_{\sigma_1} G_1 \ldots \rightsquigarrow_{\sigma_n} \square$, or $G \rightsquigarrow^*_{\sigma} \square$, with $\sigma = \sigma_1 \ldots u\sigma_n$, then $\sigma_{vars(G)}$ is a *computed answer* for G.

The main result for this calculus is:

Theorem 2 (Correctness of the calculus for unification)**.** *The calculus for unification is sound and weakly complete, i.e., given a unification goal G, if $G \rightsquigarrow^*_{\sigma} \square$, then $G\sigma$ can be derived in $\to_{E/B}$ (equal to $\to_{RE,B}$) using the derivation rules in Figure 2.3, and if ρ is an E/B-normalized idempotent solution of G, so $G\rho \to^*_{E/B} \top$, then there is an idempotent substitution ρ', such that $\rho \ll_B \rho'$ and $G \rightsquigarrow^*_{\rho'} \square$.*

See proof on page 48.

3.3 Reachability by conditional narrowing modulo

Conditional narrowing relies on conditional unification. As we have used the symbol $\to$ in the calculus rules for unification, we will use a different symbol $\Rightarrow$ in the calculus rules for reachability. These new calculus rules deal with the $\rightsquigarrow_{ER,B}$ relation in the reachability problems. Narrowing, we call it *replacement* here, takes place only at position ϵ of terms, thanks to new transitivity and imitation calculus rules.

- $[u]$ *unification*

$$\frac{u{:}s = v{:}t, G'}{u{:}s' \to x_{[s']}{:}s', v{:}s' \to x_{[s']}{:}s', G'}$$

where $x_{[s']}$ fresh variable, $s' \in \mathit{glbSorts}(s,t)$.

- $[x]$ *matching*

$$\frac{u{:}s := v{:}t, G'}{v{:}s' \to u{:}s', G'}$$

where $s' \in \mathit{glbSorts}(s,t)$.

- $[n]$ *narrowing*

$$\frac{u{:}s \to^1 x_{[s]}{:}s, G'}{((\hat{c},)G')\rho\theta}$$

where u is not a variable, $(c)eq\ l{=}r\ (if\ c) \in E$ has fresh variables, $\theta \in CSU_B(u = l)$, and $\rho{=}\{x_{[s]} \mapsto r\}$.

- $[t]$ *transitivity*

$$\frac{u{:}s \to v{:}t, G'}{u{:}s' \to^1 x_{[s']}{:}s', x_{[s']}{:}s' \to v{:}s', u{:}s', G'}$$

where $x_{[s']}$ fresh variable, $s' \in \mathit{glbSorts}(s,t)$.

- $[i]$ *imitation*

$$\frac{f(u_1,\ldots,u_n){:}s \to^1 x_{[s]}{:}s, G'}{u_i{:}s_i \to^1 x'_{[s_i]}{:}s_i, G'\theta}$$

where $u_i \notin \mathcal{X}$, $s_i = ls(u_i)$,
$\theta = \{x_{[s]} \mapsto f(u_1,\ldots,u_{i-1},x'_{[s_i]},u_{i+1},\ldots,u_n)\}$,
and $x'_{[s_i]}$ fresh variable.

Figure 3.1: Calculus rules for unification I

- $[r]$ *removal of equations*

$$\frac{u{:}s \to v{:}t, G'}{(u{:}s', G')\theta}$$

with $\theta \in CSU_B(u = v)$ and $s' \in glbSorts(s, t)$

- $[s]$ *subject reduction*

$$\frac{u{:}s, G'}{u{:}s \to^1 x_{[s]}{:}s, x_{[s]}{:}s, G'}$$

with x_s fresh variable.

- $[m1]$ *membership*

$$\frac{x_{[s]}{:}t, G'}{(G')\theta}$$

where $\theta = \{x_{[s]} \mapsto x'_t\}$ with x'_t fresh variable.

- $[m2]$ *membership*

$$\frac{u{:}s, G'}{((\hat{c},)\ G')\theta}$$

where $(c)mb\ v{:}t\ (if\ c)$ is a fresh variant, with $t \leq s$, of a (conditional) membership in E and $\theta \in CSU_B(u = v)$.

Figure 3.2: Calculus rules for unification II

Reachability goals are any sequence (understood as conjunction) of subgoals of the form $u{:}s \Rightarrow v{:}t$. Admissible goals, or simply goals, are now extended to be any sequence of $u{:}s \Rightarrow v{:}t$, $u{:}s \Rightarrow^1 v{:}t$, $u{:}s = v{:}t$, $u{:}s \rightarrow v{:}t$, $u{:}s \rightarrow^1 v{:}t$, $u{:}s := v{:}t$ and $v{:}t$.

Given a reachability goal $G = u_1{:}s_1 \Rightarrow v_1{:}t_1, \ldots, u_n{:}s_n \Rightarrow v_n{:}t_n$, a *solution* of G is a substitution σ (ground or not) such that, for $1 \leq i \leq n$, $u_i\sigma{:}s_i$, $v_i\sigma{:}t_i$, and $u_i\sigma \rightarrow_{R/\mathcal{E}} v_i\sigma$ (we will write $u_i\sigma{:}s_i \rightarrow_{R/\mathcal{E}} v_i\sigma{:}t_i$ as a shortcut). For executable rewrite theories, the last requirement is equivalent to $u_i\sigma \rightarrow_{ER,B} v_i\sigma$.

As for unification, in our calculus any reachability subgoal of the form $u{:}s \Rightarrow v{:}t$ is equivalent to the reachability goal $u \rightarrow v$, where for any solution σ of $u \rightarrow v$ in $\rightarrow^1_{R/\mathcal{E}}$ we require $u\sigma{:}s$ and $v\sigma{:}t$.

We extend the definition of $\hat{c}$ to conditions with rules: for any conditional rule $l \rightarrow r$ *if* $c \in R$, any rewrite condition in c of the form $s \rightarrow t$ is turned into a reachability goal in $\hat{c}$ of the form $u{:}ls(u) \Rightarrow v{:}ls(v)$.

Reachability by conditional narrowing is achieved using the calculus rules in Figures 3.1 and 3.2, extended with the calculus rules in Figure 3.3 which are now briefly explained.

- Rule $[X]$ solves reachability problems by unification.
- Rule $[R]$ applies one step of reachability narrowing.
- Rule $[I]$ allows us to imitate narrowing at non root term positions.
- Rule $[T]$ enables reachability narrowing modulo and multiple steps of reachability narrowing. The use of the $\Rightarrow^1$ symbol in this rule disables continuous application of the rule, forcing the generation of an actual narrowing step through rule $[R]$, maybe with several applications of rule $[I]$ in-between, since rule $[R]$ is the only one that gets rid of the $\Rightarrow^1$ symbols.

The narrowing steps for reachability ($\Rightarrow^1$), which are generated by rule $[T]$, impose no sort within the given kind on the right side of the step, since rewriting rules do not need to be sort decreasing. From a reachability goal G, a narrowing path is constructed by applying rules of the calculus. Each application of the *reflexivity* rule generates a unification equation. These unification equations as well as any generated membership goals must be solved using the calculus rules for unification. Again, if $G \rightsquigarrow^*_\sigma \square$ then $\sigma_{vars(G)}$ is a computed answer for G.

Theorem 3 (Correctness of the calculus for reachability)**.** *The calculus for reachability is sound and weakly complete, i.e., given a reachability goal G, if $G \rightsquigarrow^*_\sigma \square$, then σ is a solution for G, and if θ is an $R/\mathcal{E}$-normalized idempotent answer for G, then there is σ idempotent, with $\theta \ll_{\mathcal{E}} \sigma_{vars(G)}$, such that $G \rightsquigarrow^*_\sigma \square$.*

See proof on page 53.

3.4 Narrowing example: Tower of Hanoi

As an example of our calculus we use the specification of the Tower of Hanoi puzzle and the reachability goal

$$(3T^0_T, b, c){:}S \Rightarrow (a, b, T^1_T){:}S$$

where from a `State` composed of one `Tower` with `Disk` 3 on top of it and two empty `Towers`, `Rods` b and c, we want to reach a `State` composed of two empty `Towers`, `Rods` a

- $[X]$ *reflexivity*

$$\frac{u{:}s \Rightarrow v{:}t, G'}{u{:}s = v{:}t, G'}$$

- $[R]$ *replacement*

$$\frac{u{:}s \Rightarrow^1 x_{[s]}{:}[s], G'}{((\hat{c},), G')\rho\theta}$$

where $(c)rl\ l \Rightarrow r\ (if\ c)$ is a fresh variant of a (conditional) rule in R,
$u \notin \mathcal{X}$, $\rho = \{x_{[s]} \mapsto r\}$, and $\theta \in CSU_B(u = l)$.

- $[T]$ *transitivity*

$$\frac{u{:}s \Rightarrow v{:}t, G'}{u{:}s \rightarrow x_{[s]}{:}s, x_{[s]}{:}s \Rightarrow^1 x'_{[s]}{:}[s], x'_{[s]}{:}[s] \Rightarrow v{:}t, G'}$$

where $x_{[s]}$ and $x'_{[s]}$ are fresh variables.

- $[I]$ *imitation*

$$\frac{f(u_1, \ldots, u_n){:}s \Rightarrow^1 x_{[s]}{:}[s], G'}{u_i{:}s_i \Rightarrow^1 x'_{[s_i]}{:}[s_i], G'\theta}$$

where $u_i \notin \mathcal{X}$, $s_i = ls(u_i)$,
$\theta = \{x_{[s]} \mapsto f(u_1, \ldots, x'_{[s_i]}, \ldots, u_n)\}$, and $x'_{[s_i]}$ fresh variable.

Figure 3.3: Calculus rules for reachability

and b, and another `Tower`. The subscript of each variable means its type (sort or kind) and we again write `V`, `D`, `R`, `T`, `P`, and `S` instead of `ValidTower`, `Disk`, `Rod`, `Tower`, `Pair`, and `State` for readability. In this example one narrowing path is shown. A different selection of B-unifiers would lead to other paths. Some steps of the path are omitted since they are similar to previous ones:

1. $\underline{(3T^0_T, b, c){:}S \Rightarrow (a, b, T^1_T){:}S} \rightsquigarrow_{[T]}$

   Transitivity decomposes reachability into several rewriting narrowing steps.

2. $\underline{(3T^0_T, b, c){:}S \rightarrow X^1_S{:}S}, X^1_S{:}S \Rightarrow^1 X^2_{[S]}{:}[S], X^2_{[S]}{:}[S] \Rightarrow (a, b, T^1_T){:}S$

   $\rightsquigarrow_{[r],\{T^0_T \mapsto a, X^1_S \mapsto (3a,b,c)\}}$ T^0_T is instantiated through rule $[r]$.

3. $\underline{(3a, b, c){:}S}, (3a, b, c){:}S \Rightarrow^1 X^2_{[S]}{:}[S], X^2_{[S]}{:}[S] \Rightarrow (a, b, T^1_T){:}S$

   We focus on the first subgoal.

4. $\underline{(3a, b, c){:}S} \rightsquigarrow_{[m2],S^1_{[S]},S^2_{[S]}:S \text{ if } S^1_{[S]}:S \wedge S^2_{[S]}:S,\{S^1_{[S]} \mapsto (3a,b),S^2_{[S]} \mapsto c\}}$

5. $\underline{c{:}S}, (3a, b){:}S \rightsquigarrow_{[m2],c:R}$. OK because $R \leq S$.

6. $\underline{(3a, b)}{:}S \rightsquigarrow \ldots$ similar to the two previous steps.

7. $\underline{3a{:}S} \rightsquigarrow_{[m2],X_{[D]}R_{[R]}:V \text{ if } X_{[D]}:D \wedge R_{[R]}:R,\{X_{[D]} \mapsto 3,R_{[R]} \mapsto a\}}$. OK because $V \leq S$.

8. $\underline{3{:}D, a{:}R} \rightsquigarrow \ldots$ similar to previous steps. First subgoal finished.

9. $\underline{(3a, b, c){:}S \Rightarrow^1 X^2_{[S]}{:}[S]}, X^2_{[S]}{:}[S] \Rightarrow (a, b, T^1_T){:}S$. We focus on the first subgoal.

10. $\underline{(3a, b, c){:}S \Rightarrow^1 X^2_{[S]}{:}[S]} \rightsquigarrow_{[R],D_{[T]},E_{[T]},X_{[S]} \rightarrow F_{[T]},G_{[T]},X_{[S]} \text{ if}}$

    $D_{[T]}:T \wedge E_{[T]}:T \wedge X_{[S]}:S \wedge F_{[T]}:T \wedge G_{[T]}:T \wedge F_{[T]} - G_{[T]} := move(D_{[T]} - E_{[T]}),$

    $\theta = \{D_{[T]} \mapsto 3a, E_{[T]} \mapsto c, X_{[S]} \mapsto b\}, \rho = \{X^2_{[S]}{:}[S] \mapsto F_{[T]}, G_{[T]}, X_{[S]}\}$ Narrowing step.

11. $\underline{(3a, b, c){:}S, 3a{:}T, c{:}T, b{:}S}, (F_{[T]} - G_{[T]}){:}[P] := move(3a - c){:}[P] \rightsquigarrow \ldots$

12. $\underline{F_{[T]} - G_{[T]}{:}[P] := move(3a - c){:}[P]} \rightsquigarrow_{[x]}$

13. $\underline{move(3a - c)}{:}[P] \rightarrow F_{[T]} - G_{[T]}{:}[P] \rightsquigarrow_{[t]}$

    Transitivity decomposes unification into several narrowing for unification steps.

14. $\underline{move(3a - c)}{:}[P] \rightarrow^1 Y_{[P]}{:}[P], Y_{[P]}{:}[P] \rightarrow F_{[T]} - G_{[T]}{:}[P] \rightsquigarrow_{[n],}$

    $move(X_{[D]}T_{[T]} - R_{[R]}) = T_{[T]} - X_{[D]}R_{[R]} \text{ if } X_{[D]}:D \wedge T_{[T]}:T \wedge R_{[R]}:R,$

    $\theta = \{X_{[D]} \mapsto 3, T_{[T]} \mapsto a, R_{[R]} \mapsto c\}, \rho = \{Y_{[P]} \mapsto T_{[T]} - X_{[D]}R_{[R]}\}$

    Narrowing for unification step. $Y_{[P]}$ is instantiated to a ground term.

15. $\underline{a - 3c{:}[P], 3{:}[D], a{:}[T], c{:}[R]}, a - 3c{:}[P] \rightarrow F_{[T]} - G_{[T]}{:}[P] \rightsquigarrow \ldots$

16. $\underline{a - 3c{:}[P] \to F_{[T]} - G_{[T]}{:}[P]} \rightsquigarrow_{[r],\theta_1=\{F_{[T]}\mapsto a, G_{[T]}\mapsto 3c\}}$ Removal of equations.

17. $\underline{a - 3c{:}[P]} \rightsquigarrow \ldots$ We omit this and go back to the second subgoal on step 9.

18. $\underline{(a, 3c, b) : [S] \Rightarrow (a, b, T_T^1){:}S} \rightsquigarrow_{[X]} \ldots$

19. $\underline{(a, 3c, b) : S \to X_S{:}S}, (a, b, T_T^1){:}S \to X_S{:}S \rightsquigarrow_{[r],\{X_S\mapsto(a,3c,b)\}}$

20. $\underline{(a, 3c, b) : S}, (a, b, T_T^1){:}S \to (a, 3c, b){:}S \rightsquigarrow \ldots$

21. $\underline{(a, b, T_T^1){:}S \to (a, 3c, b){:}S} \rightsquigarrow_{[r],\{T_T^1\mapsto 3c\}} T_T^1$ is instantiated through rule $[r]$.

22. $\underline{(a, b, 3c) : S} \rightsquigarrow \ldots \square$

From the substitutions in steps 2 and 21 the answer $\{T_T^1 \mapsto 3c, T_T^0 \mapsto a\}$ is computed. The calculus has found that $(3a, b, c){:}S \Rightarrow (a, b, 3c){:}S$, which is an instance of the given reachability goal $(3T_T^0, b, c){:}S \Rightarrow (a, b, T_T^1){:}S$, meaning $(3a, b, c){:}S$, $(a, b, 3c){:}S$, and $(3a, b, c) \to_{R/\mathcal{E}}^* (a, b, 3c)$.

3.5 Results and proofs

Theorem 2 (Correctness of the calculus for unification)**.** *The calculus for unification is sound and weakly complete, i.e., given a unification goal G, if $G \rightsquigarrow_\sigma^* \square$, then $G\sigma$ can be derived in $\to_{E/B}$ (equal to $\to_{RE,B}$) using the derivation rules in Figure 2.3, and if ρ is an E/B-normalized idempotent solution of G, so $G\rho \to_{E/B}^* \top$, then there is an idempotent substitution ρ', such that $\rho \ll_B \rho'$ and $G \rightsquigarrow_{\rho'}^* \square$.*

Proof. We prove that given a unification goal G, if $G \rightsquigarrow_\sigma^* \square$ then $G\sigma$ can be derived, so σ is a solution for G in $\to_{E/B}$, and if ρ is an E/B-normalized idempotent answer of G ($G\rho \to_{E/B}^* \top$), then there is ρ' idempotent, with $\rho \ll_B \rho'$, such that $G \rightsquigarrow_{\rho'} \square$. For clarity, we write $u \to_{\mathcal{R}_E} v$ when $\mathcal{R}_E \vdash u \to v$ and $u \to_{\mathcal{R}_E}^1 v$ when $\mathcal{R}_E \vdash u \to^1 v$.

1. Soundness: Soundness of the calculus is proved by induction on the length of the derivation. We transform any goal ($u{:}s$ *op* $v{:}t$) into (u *op* v, $u{:}s$, $v{:}t$), as explained in Section 3.2.

   Base step: proofs with length one. The only inference rules that delete goals without creating new ones are $[m1]$, and $[m2]$ in the case of constants ($i = 0$) and non conditional memberships:

   $[m1]$ *membership*

   $$\frac{x_{[s]}{:}t}{\square}$$

   where $\theta = \{x_{[s]} \mapsto x'_t\}$ with x'_t fresh variable.

Recall that $x_{[s]}{:}t$ is a shortcut for $x_{[s]} = y_t$ with y_t fresh variable. The substitution $\sigma = \{x_{[s]} \mapsto x'_t, y_t \mapsto x'_t\}$ is valid and $x'_t = x'_t$ is derivable by reflexitivity. When restricted to the variable in the problem (y_t does not appear elsewhere), we get the original θ.

[$m2$] *membership, $i = 0$, no conditions*

$$\frac{u{:}s}{\square}$$

where $mb\ u : t$ is a membership in E, with $t \leq s$.

Again, $u{:}s$ is a shortcut for $u = y_s$. As $t \leq s$, $\sigma = \{y_s \mapsto u\}$ is a valid substitution and $u = u$ is derivable by reflexitivity. When restricted to the variable in the problem we get the *id* substitution.

Induction step: We assume that if a derivation from a goal G, with length n or less, provides a substitution σ, then $G\sigma$ is derivable and the associated unification goal rewrites to $\top$, that is, σ is an answer of G. We have to prove that this property holds for derivations with length $n + 1$. In the following we write $u \to_{\mathcal{R}_E} v$ and $u \to^1_{\mathcal{R}_E} v$) as convenient shortcuts for $\mathcal{R}_E \vdash u \to v$ and $\mathcal{R}_E \vdash u \to^1 v$, respectively (see Fig. 2.3). We assume that G has the form g, G', where G' may be empty, and check all possible calculus rules applied to g:

[u] *unification*

$$\frac{u{:}s = v{:}t, G'}{u{:}s' \to x_{[s']}{:}s', v{:}s' \to x_{[s']}{:}s', G'}$$

where $x_{[s']}$ fresh variable and $s' \in \mathit{glbSorts}(s, t)$.

By induction hypothesis if there substitution σ, computed answer for $u{:}s' \to x_{[s']}{:}s'$, $v{:}t \to x_{[s']}{:}s'$, and G' then we can derive $u\sigma \to_{\mathcal{R}_E} x_{[s']}\sigma$, $v\sigma \to_{\mathcal{R}_E} x_{[s']}\sigma$, $u\sigma :_{\mathcal{E}} s'$ $v\sigma :_{\mathcal{E}} s'$, and $G'\sigma$. Since $s' \leq s$ and $s' \leq t$, we can derive $u\sigma :_{\mathcal{E}} s$ and $v\sigma :_{\mathcal{E}} t$.

Due to the derivation rules in Figure 2.3, $u\sigma \to_{\mathcal{R}_E} x_{[s']}\sigma$ can only be derived by a chain of derivations $u\sigma \to^1_{\mathcal{R}_E} s_1 \ldots \to^1_{\mathcal{R}_E} u_i \to_{\mathcal{R}_E} x_{[s']}\sigma$ where i can be 0, the last $\to_{\mathcal{R}_E}$ is a derivation using reflexivity ($u_i =_B x_{[s']}\sigma$) and everything joins by transitivity. The same goes for $v\sigma \to_{\mathcal{R}_E} x_{[s']}\sigma$ ($v_j =_B x_{[s']}\sigma$) and any other derivation $t \to t'$.

$g\sigma \equiv u\sigma{:}s = v\sigma{:}t$. We are going to show that the equivalent problem in R_E: $eq(u\sigma, v\sigma) \wedge u\sigma{:}s \wedge v\sigma{:}t$. As $u\sigma :_{\mathcal{E}} s$ and $v\sigma :_{\mathcal{E}} t$ are derivable all that is left to do is proving that $eq(u\sigma, v\sigma) \to_{\mathcal{R}_E} tt$. Applying congruence several times we get: $eq(u\sigma, v\sigma) \to^1_{\mathcal{R}_E} \ldots \to^1_{\mathcal{R}_E} eq(u_i, v_j)$. As $u_i =_B x_{[s']}\sigma =_B v_j$, from $eq(x_{[s']}, x_{[s']}) \to tt$ we derive $eq(u_i, v_j) \to^1_{\mathcal{R}_E} tt$ (then also $eq(u_i, v_j) \to_{\mathcal{R}_E} tt$). Now, by transitivity, $eq(u\sigma, v\sigma) \to_{\mathcal{R}_E} tt$.
σ is a solution of g and also of G', so σ is a solution of G.

[x] *matching*

$$\frac{u{:}s := v{:}t, G'}{v{:}s' \to u{:}s', G'}$$

where $s' \in \mathit{glbSorts}(s, t)$.

By I.H. if σ is a computed answer of $v{:}s' \to u{:}s'$ and G', we derive $u\sigma :_{\mathcal{E}} s$ and $v\sigma :_{\mathcal{E}} t$, as before, and $v\sigma \to_{\mathcal{R}_E} u\sigma$, so $v\sigma \to^1_{\mathcal{R}_E} v_1 \ldots \to^1_{\mathcal{R}_E} v_i \to_{\mathcal{R}_E} u\sigma$, with $u\sigma =_B v_i$. Recall that in a MEL theory we treat matching logically as equality, the difference between them is computational, so $g\sigma \equiv u\sigma{:}s = v\sigma{:}t$. Again, we show that we can solve the equivalent problem in R_E: $eq(u\sigma{:}s, v\sigma{:}t) \to_{\mathcal{R}_E} \top$. As we can derive $u\sigma :_{\mathcal{E}} s$ and $v\sigma :_{\mathcal{E}} t$, we have to prove $eq(u\sigma, v\sigma) \to_{\mathcal{R}_E} tt$. By congruence, $eq(u\sigma, v\sigma) \to^1_{\mathcal{R}_E} \ldots \to^1_{\mathcal{R}_E} eq(u\sigma, v_i)$. From $eq(x_{[s']}, x_{[s']}) \to tt$ by replacement with $\theta = \{x_{[s']} \mapsto u\sigma\}$ (as $eq(u\sigma, v_i) =_B eq(u\sigma, u\sigma)$) we have $eq(u\sigma, u\sigma) \to^1_{\mathcal{R}_E} tt$ (and $eq(u\sigma, u\sigma) \to_{\mathcal{R}_E} tt$). Then, by transitivity, $eq(u\sigma, v\sigma) \to_{\mathcal{R}_E} tt$.
σ is a solution of $u{:}s{:=}v{:}t$ and also of G', so σ is a solution of G.

[n] *narrowing*

$$\frac{u{:}s \to^1 x_{[s]}{:}s, G'}{((\hat{c},)G')\rho\theta}$$

where u is not a variable, $(c)eq\ l{=}r\ (if\ c) \in E$ has fresh variables,
$\theta \in CSU_B(u = l)$, and $\rho{=}\{x_{[s]} \mapsto r\}$.

As we are solving unification goals, either from the original problem or from a condition in a conditional equation or membership, the goal $u{:}s \to^1 x_{[t]}{:}t$ may only have appeared:

- by application of rules [t] or [s], so G' has to include the subgoal $u{:}s$.
- by application of rule [i], so $s = ls(u)$. Then G' does not include any unneeded check since $ls(u\sigma) \leq ls(u)$ for any substitution σ.

$\sigma = \rho\theta\sigma'$ is a computed answer for c and G', hence for $u{:}s$ in either of the cases above. By I.H. we can derive $u\sigma :_{\mathcal{E}} s$ (trivially in the second case above), $c\sigma$ and $G'\sigma$. $u\rho\theta = u\theta =_B l\theta = l\rho\theta$ implies $u\sigma =_B l\sigma$. We have the equation $(c)eq\ l = r\ (if\ c)$, and we have derived $c\sigma$, so we derive $u\sigma \to^1_{\mathcal{R}_E} r\sigma$, i.e. $u\sigma =_{\mathcal{E}} r\sigma$. As ρ is idempotent, we have $r\sigma = (x\rho)\rho\theta\sigma' = (x\rho\rho)\theta\sigma' = x\rho\theta\sigma' = x_{[s]}\sigma$, so $u\sigma =_{\mathcal{E}} x_{[s]}\sigma$. Then we have $u\sigma \to^1 x_{[s]}\sigma$ and , as $u\sigma :_{\mathcal{E}} s$ and $u\sigma =_{\mathcal{E}} x_{[s]}\sigma$, also $x_{[s]}\sigma :_{\mathcal{E}} s$.

[t] *transitivity*

$$\frac{u{:}s \to v{:}t, G'}{u{:}s' \to^1 x_{[s']}{:}s', x_{[s']}{:}s' \to t{:}s', u{:}s', G'}$$

where $x_{[s']}$ fresh variable and $s' \in glbSorts(s, t)$.

If σ is the computed answer, by I.H. we can derive $u\sigma \to^1_{\mathcal{R}_E} x_{[s']}\sigma$ and $x_{[s']}\sigma \to_{\mathcal{R}_E} v\sigma$, and $u\sigma :_{\mathcal{E}} s'$. Then, by transitivity, we can derive $u\sigma \to_{\mathcal{R}_E} v\sigma$, so $u\sigma =_{\mathcal{E}} v\sigma$, hence $v\sigma :_{\mathcal{E}} s'$. Since $s' \leq s$ and $s' \leq t$, also $u\sigma :_{\mathcal{E}} s$ and $v\sigma :_{\mathcal{E}} t$.

[i] *imitation*

$$\frac{f(\bar{u}){:}s \to^1 x_{[s]}{:}s, G'}{u_i{:}s_i \to^1 x'_{[s_i]}{:}s_i, G'\theta}$$

where u is not a variable, $s_i = ls(u_i)$,
$\theta = \{x_{[s]} \mapsto f(u_1, \ldots, u_{i-1}, x'_{[s_i]}, u_{i+1}, \ldots, u_n)\}$,
and $x'_{[s_i]}$ fresh variable.

As for rule $[n]$, either $s = ls(f(\bar{u}))$, so there is nothing to check, or G' contains the subgoal $f(\bar{u}){:}s$.

$\sigma = \theta\sigma'$, σ' computed answer for $u_i{:}s_i \to^1 x'_{[s_i]}{:}s_i, x\theta{:}s'$ and $G'\theta$ as before. By I.H. we can derive $u_i\sigma \to^1_{\mathcal{R}_E} x'_{[s_i]}\sigma$ and $G'\sigma$, hence $f(\bar{u})\sigma :_{\mathcal{E}} s$. As $u_i\sigma \to^1_{\mathcal{R}_E} x'_{[s_i]}\sigma$, by congruence, $f(\bar{u}\sigma) \to^1_{\mathcal{R}_E} x_{[s]}\sigma$, so $f(\bar{u}\sigma) =_{\mathcal{E}} x_{[s]}\sigma$, hence $x_{[s]}\sigma :_{\mathcal{E}} s$.

$[r]$ *removal of equations*

$$\frac{u{:}s \to v{:}t, G'}{(u{:}s', G')\theta}$$

with $\theta \in CSU_B(u = v)$, and $s' \in glbSorts(s, t)$

$\theta\sigma' = \sigma$ is a computed answer for G', c', $u{:}s'$ and $t{:}s'$. By I.H. we can derive $c'\sigma$, $u\sigma :_{\mathcal{E}} s'$, $v\sigma :_{\mathcal{E}} s'$ and $G'\sigma$. $u'\theta =_B v'\theta$ implies $u'\sigma =_B v'\sigma$ and $c'\sigma$ is derivable, so each instantiated variable has correct sort and then $u\sigma =_B v\sigma$. By reflexivity we derive $u\sigma \to_{\mathcal{R}_E} v\sigma$. Again, as $s' \leq s$ and $s' \leq t$, we get $u\sigma :_{\mathcal{E}} s$ and $v\sigma :_{\mathcal{E}} t$.

$[s]$ *subject reduction*

$$\frac{u{:}s, G'}{u{:}s \to^1 x_{[s]}{:}s, x_{[s]}{:}s, G'}$$

with $x_{[s]}$ fresh variable.

If σ is a computed answer then, by I.H., we can derive $u\sigma \to^1_{\mathcal{R}_E} x_{[s]}\sigma$, so $u\sigma =_{\mathcal{E}} x_{[s]}\sigma$, $x_{[s]}\sigma :_{\mathcal{E}} s$, hence $u\sigma :_{\mathcal{E}} s$, and $G'\sigma$.

$[m1]$ *membership*

$$\frac{x_{[s]}{:}t, G'}{(G')\theta}$$

where $\theta = \{x_{[s]} \mapsto x'_t\}$ with x'_t fresh variable.

By I.H., if $\theta\sigma' = \sigma$ is a computed answer for G', then we can derive $G'\sigma$.

As seen in the base case, $x'_t{:}t$ is trivially derivable by reflexivity, so any instance of x'_t has also sort t, hence $x_{[s]}\sigma :_{\mathcal{E}} t$.

$[m2]$ *membership*

$$\frac{u{:}s, G'}{((\hat{c},)\ G')\theta}$$

where $(c)mb\ v{:}t\ (if\ c)$ is a fresh variant, with $t \leq s$, of a (conditional) membership in E and $\theta \in CSU_B(u = v)$.

$\theta\sigma' = \sigma$ is a computed answer of c and G'. By I.H. $c\sigma$ and $G'\sigma$ are derivable. $u\theta =_B v\theta$ implies $u\sigma =_B v\sigma$. Then by membership, as $c\sigma$ is derivable, we derive $u\sigma :_{\mathcal{E}} t$ and, as $t \leq s$, again by membership we derive $u\sigma :_{\mathcal{E}} s$.

2. Completeness: As previously stated, we assume that we have a complete B-unification algorithm that returns a *CSU* for any pair of terms, so our substitutions may be more general than the ones we are imitating, therefore improving the answer.

We prove that if ρ is an E/B-normalized idempotent answer of G ($G\rho \to^* \top$), then there is ρ' idempotent, with $\rho \ll_B \rho'_{vars(G)}$, such that $G \rightsquigarrow_{\rho'} \Box$. Completeness of the calculus is proved by induction on the length of inferences in $\mathcal{R} = (\Sigma', B, R_E)$, using the derivation rules in Figure 2.3, looking at the last inference rule used:
Base step:

(Reflexivity)

$$\frac{}{u \to v}$$

if $u =_B v$

$u\rho =_B v\rho$ allows the inference $u\rho \to v\rho$. By hypothesis, the complete B-unification algorithm can compute some ρ', with $\rho \ll_B \rho'$, answer of $u =_B v$. Let $s = ls(u)$ and $t = ls(v)$. By definition both s and t are upper bounds on the type of u and v that are satisfied by any substitution. Then, this memberships are always computable in our calculus without instantiating any variables, except renamings of variables that don't affect $\ll_B$.
$u{:}s \to v{:}t \rightsquigarrow_{[r],\rho'} u\rho'{:}s, v\rho'{:}t \rightsquigarrow^* \Box$.

On the rest of rules when no membership gets involved we omit the part on sorts.

Induction step:

(Transitivity)

$$\frac{u_1 \to^1 u_2,\ u_2 \to u_3}{u_1 \to u_3}$$

$u_1\rho \to^1_{\mathcal{R}_E} u_2$ and $u_2 \to_{\mathcal{R}_E} u_3\rho$ allow the inference $u_1\rho \to_{\mathcal{R}_E} u_3\rho$, let $s = ls(u_2)$. Then $\sigma{=}\rho \cup \{x_{[s]}{:}s \mapsto u_2\}$ is an idempotent solution for $u_1{\to^1}x_{[s]}, x_{[s]}{\to}u_3$, so $u_1\sigma =_{\mathcal{E}} u_2$ and $u_1\sigma :_{\mathcal{E}} s$, with $x_{[s]}$ fresh variable. By I.H. we can compute idempotent answers σ_1, σ_2, and σ_3, with $\sigma \ll_B (\sigma_1 \cdot \sigma_2) \cdot \sigma_3 \ll_B \sigma_1 \cdot \sigma_2 \ll_B \sigma_1$, for $u_1{:}s \to^1 x_{[s]}{:}s$, $x_{[s]}\sigma_1{:}s \to u_3\sigma_1{:}s$, and $u_1\sigma_1\sigma_2{:}s$. So $(\sigma_1 \cdot \sigma_2) \cdot \sigma_3 = \rho' \cup \{x_{[s]}{:}s \mapsto u'_2\}$ with $\rho \ll_B \rho'$, so $\rho \ll_B (\sigma_1\sigma_2\sigma_3)_{vars(u_1 \to u_3)}$, and $u_1{:}s{\to}u_3{:}s \rightsquigarrow_{[t]} u_1{:}s{\to^1}x_{[s]}{:}s, x_{[s]}{:}s{\to}u_3{:}s, u_1{:}s \rightsquigarrow^*_{\sigma_1}$ $x_{[s]}\sigma_1{:}s{\to}u_3\sigma_1{:}s, u_1\sigma_1{:}s \rightsquigarrow^*_{\sigma_2} u_1\sigma_1\sigma_2{:}s \rightsquigarrow^*_{\sigma_3} \Box$.

(Congruence)

$$\frac{u_i \to^1 u'_i}{f(u_1, \ldots, u_i, \ldots, u_n) \to^1 f(u_1, \ldots, u'_i, \ldots, u_n)}$$

Let $\bar{u} = u_1, \cdots, u_n$, $\bar{u}' = u_1, \cdots, u'_i, \cdots, u_n$, $s = ls(f(\bar{u}))$ and $t = ls(u'_i)$. If ρ' is an idempotent answer for $f(\bar{u}){\to^1}f(\bar{u}')$, because $u_i\rho'{\to^1_{\mathcal{R}_E}}u'_i\rho'$, then $\rho{=}\rho'{\cup}\{x'_{[t]}{\mapsto}u'_i\rho'\}$ is an idempotent answer for $u_i{\to^1}x'_{[t]}$, with $x'_{[t]}$ fresh variable. By I.H. there is σ idempotent, with $\rho \ll_B \sigma$, such that σ is a computed solution for $u_i{\to^1}x'_{[t]}$, so there exists θ such that $\sigma\theta =_B \rho$. Without loss of generality we assume $i{=}1$, let $\bar{u}_2 = u_2, \cdots, u_n$.

$f(\bar{u}){:}s \to f(\bar{u}'){:}s \rightsquigarrow_{[t]} f(\bar{u}){:}s \to^1 x_{[s]}{:}s, f(\bar{u}){:}s \to f(\bar{u}'){:}s \rightsquigarrow_{[i],\{x_{[s]} \mapsto f(x'_{[t]}, \bar{u}_2)\}}$

$u_1{:}t \to x'_{[t]}{:}t, f(x'_{[t]}, \bar{u}_2){:}s \to f(u'_1, \bar{u}_2), f(\bar{u}){:}s \overset{I.H.}{\rightsquigarrow}_{\sigma}$

$f(x'_{[t]}\sigma, \bar{u}_2\sigma){\to}f(u'_1\sigma, \bar{u}_2\sigma), f(\bar{u})\sigma{:}s \rightsquigarrow_{[r],\theta'} f(\bar{u})\sigma\theta'{:}s \rightsquigarrow^* \Box$
where $\theta' \in CSU_B(x'_{[t]}\sigma = u'_1\sigma)$, with $\theta \ll_B \theta'$ because $(x'_{[t]}\sigma)\theta{=}_B(u'_1\sigma)\theta$, and

$f(\bar{u})\sigma\theta' :_{\mathcal{E}} s$ because $s = ls(f(\bar{u}))$. Then, $\sigma\theta'$ is an idempotent computed solution for $f(\bar{u}) \to f(\bar{u}')$ and $\rho =_B \sigma\theta \ll_B \sigma\theta'$.

(Subject Reduction)

$$\frac{u \to^1 v, v : s}{v : s}$$

If ρ is a solution, by I.H. there is σ, with $\rho \ll_B \sigma$, such that σ is a computed answer for $u \to^1 v$ and $v : s$. Then, by subject reduction, σ is a computed answer for $u{:}s$.

(Membership)

$$\frac{A_1^\bullet\sigma \ldots A_n^\bullet\sigma}{u : s}$$

if $v : s$ *if* $A_1 \ldots A_n$ in R_E and $u =_B v\sigma$

This is a two step process. First, we find a rule that matches our term u and we get σ. Then we find in instantion ρ of the conditions that allows us to derive these conditions. We call $c{=}A_1 \ldots A_n$ and $c^\bullet{=}A_1^\bullet \ldots A_n^\bullet$. As $c^\bullet\sigma\rho$ is derivable so is $c\sigma\rho$. If $\sigma\rho$ is a solution then, as $vars(u) \cap Dom(\sigma){=}\emptyset$, $u\rho{=}u\sigma\rho{=}_B v\sigma\rho$, so there are $\rho' \in CSU_B(u = v)$, with $\rho \ll_B \rho'$, and θ' such that $\sigma\rho{=}_B\rho'\theta'$. Then:

$u{:}s \rightsquigarrow_{[m2],\ t:s\ \mathit{if}\ c,\rho'} c\rho'$.

The variables in $c\sigma\rho$ are the same as the ones in $v\sigma\rho$, once the matching variables in $c\sigma\rho$ are instantiated. As $c\sigma\rho$ is derivable and $\sigma\rho{=}_B\rho'\theta'$, $(c\rho')\theta'$ is also derivable *in the same number of derivation steps*. By I.H. there is θ'' such that $\hat{c}\rho' \rightsquigarrow^*_{\theta''} \square$ and $\theta' \ll_B \theta''$, so there is σ'' such that $\theta''\sigma''{=}_B\theta'$. Then $\rho'\theta''\sigma''{=}_B\rho'\theta'{=}_B\sigma\rho$, so $\sigma\rho \ll_B \rho'\theta''$. Now:

$u{:}s \rightsquigarrow_{[m2],\ v:s\ \mathit{if}\ c,\rho'} \hat{c}\rho' \overset{I.H.^*}{\rightsquigarrow}_{\theta''} \square$,

and the computed answer $\rho'\theta''$ is more general than $\sigma\rho$.

□

Theorem 3 (Correctness of the calculus for reachability)**.** *The calculus for reachability is sound and weakly complete, i.e., given a reachability goal G, if $G \rightsquigarrow^*_\sigma \square$, then σ is a solution for G, and if θ is an $R/\mathcal{E}$-normalized idempotent answer for G, then there is σ idempotent, with $\theta \ll_{\mathcal{E}} \sigma_{vars(G)}$, such that $G \rightsquigarrow^*_\sigma \square$.*

Proof. We prove correctness of the calculus for reachability with respect to $R/\mathcal{E}$-normalized (equivalently ER, B) idempotent substitutions for the executable rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ in $\to_{R/\mathcal{E}}$.

1. Soundness: We prove that given a reachability goal G, if $G \rightsquigarrow^*_\sigma \square$ then $G\sigma$ can be derived, so σ is a solution for G in $\to_{R/\mathcal{E}}$. In the proof of completeness we will see that the set of computed answers is the same with respect to $R/\mathcal{E}$-normalized idempotent substitutions. Soundness of the reachability calculus is proved by induction on the length of the derivation. Recall that all calculus rules always check correct typings on the premises. We transform any goal $(u{:}s \Rightarrow v{:}t)$ into $(s \Rightarrow t,\ u{:}s,\ v{:}t)$, but we may use both writings for simplicity. By our previous proof of soundness, we know that if we compute a solution σ for $u{:}s{=}v{:}t$ we can derive $u\sigma =_{\mathcal{E}} v\sigma, u\sigma :_{\mathcal{E}} s, v\sigma :_{\mathcal{E}} t$ using the deduction rules for Mel. The proof is by induction on the number of calculus rules from Figure 3.3 applied in the narrowing path, where we only check

narrowing paths where the first rule applied is from Figure 3.3, since the rest of the rules have already been considered in the proof of soundness for Theorem 2 and $=_{\mathcal{E}} \subseteq \rightarrow_{R/\mathcal{E}}$.

Base step: narrowing paths with one rule from Figure 3.3. We have a goal with one element. The only new inference rule that deletes rewritings without creating new ones is rule $[X]$:

$[X]$ *reflexivity*

$$\frac{u{:}s \Rightarrow v{:}t}{u{:}s = v{:}t}$$

If σ is an answer computed by our unification calculus for $u{:}s{=}v{:}t$, then $u\sigma :_{\mathcal{E}} s, v\sigma :_{\mathcal{E}} t$, and $u\sigma =_{\mathcal{E}} v\sigma$ so, by definition of $\rightarrow_{R/\mathcal{E}}$, $u\sigma \rightarrow_{R/\mathcal{E}} v\sigma$, hence σ is a solution for $u{:}s \Rightarrow v{:}t$ in $\rightarrow_{R/\mathcal{E}}$.

Induction step: We assume that if a narrowing path $G \rightsquigarrow^*_\sigma \square$ applies n or less calculus rules from Figure 3.3, then $G\sigma$ is derivable, that is, σ is an answer of G. We have to prove that this property holds for narrowing paths that use $n+1$ rules from Figure 3.3. We assume that G has the form g, G', where G' may be empty, and that the first calculus rule from Figure 3.3 has been applied on g:

$[X]$ *reflexivity*

$$\frac{u{:}s \Rightarrow v{:}t, G'}{u{:}s = v{:}t, G'}$$

As in the base step, if σ is the computed answer, then σ is a solution for $u{:}s \Rightarrow v{:}t$ in $\rightarrow_{R/\mathcal{E}}$. By I.H., σ is also a solution for G', so σ is a solution for G in $\rightarrow_{R/\mathcal{E}}$.

$[R]$ *replacement*

$$\frac{u{:}s \Rightarrow^1 x_{[s]}{:}[s], G'}{((\hat{c},)G')\rho\theta}$$

where $(c)rl\ l \Rightarrow r\ (if\ c)$ is a fresh variant of a (conditional) rule in R,
$u \notin \mathcal{X}$, $\rho = \{x_{[s]} \mapsto r\}$, and $\theta \in CSU_B(u = l)$.

As $u\theta =_B l\theta$ then $u\theta =_{\mathcal{E}} l\theta$. If σ' is an idempotent computed answer for $c\rho\theta$ and $G'\rho\theta$ then, calling $\sigma{=}\rho\theta\sigma'$, $u\sigma =_{\mathcal{E}} l\sigma$ and, by I.H., $c\sigma$ is derivable in $\rightarrow_{R/\mathcal{E}}$, so $u\sigma \rightarrow^1_{R/\mathcal{E}} r\sigma$, hence $u\sigma \rightarrow_{R/\mathcal{E}} r\sigma (= x_{[S]}\sigma\sigma = x_{[S]}\sigma)$. By I.H. $u\sigma :_{\mathcal{E}} s$ and $G'\sigma$ are also derivable so σ is a solution for G in $\rightarrow_{R/\mathcal{E}}$.

$[T]$ *transitivity*

$$\frac{u{:}s \Rightarrow v{:}t, G'}{u{:}s \rightarrow x_{[s]}{:}s, x_{[s]}{:}s \Rightarrow^1 x'_{[s]}{:}[s], x'_{[s]}{:}[s] \Rightarrow v{:}t, G'}$$

where $x_{[s]}$ and $x'_{[s]}$ are fresh variables.

If σ is the computed answer, by I.H. we can derive $u\sigma =_{\mathcal{E}} x_{[s]}\sigma$, $x_{[s]}\sigma \rightarrow_{R/\mathcal{E}} x'_{[s]}\sigma$, $x'_{[s]}\sigma \rightarrow_{R/\mathcal{E}} v\sigma$, $u\sigma :_{\mathcal{E}} s$, $v\sigma :_{\mathcal{E}} t$, and $G'\sigma$, as before. Then, by transitivity, we can derive $u\sigma \rightarrow_{R/\mathcal{E}} v\sigma$, so σ is a solution for G in $\rightarrow_{R/\mathcal{E}}$.

$[I]$ *imitation*

$$\frac{f(u_1, \ldots, u_n){:}s \Rightarrow^1 x_{[s]}{:}[s], G'}{u_i{:}s_i \Rightarrow^1 x'_{[s_i]}{:}[s_i], f(\bar{u}){:}s, G'\theta}$$

$$\text{where } u_i \notin \mathcal{X},\ s_i = ls(u_i),$$
$$\theta = \{x_{[s]} \mapsto f(u_1, \ldots, x'_{[s_i]}, \ldots, u_n)\}, \text{ and } x'_{[s_i]} \text{ fresh variable.}$$

Let $\sigma = \theta\sigma'$, with σ' computed answer for $u_i{:}s_i \Rightarrow^1 x'_{[s_i]}{:}s_i, f(\bar{u}){:}s$, and G'. As $f(\bar{u}\sigma) = f(\bar{u}\theta\sigma') = f(\bar{u}\sigma')$ $(x_{[S]} \notin vars(f(\bar{u})))$, by I.H. we can derive $f(\bar{u}) :_{\mathcal{E}} s, G'\sigma$, and $u_i\sigma{:}s_i \rightarrow_{R/\mathcal{E}} x'_{[s_i]}\sigma{:}s_i$, so also $f(u_1\sigma, \ldots, u_n\sigma) \rightarrow_{R/\mathcal{E}} f(u_1\sigma, \ldots, x'_{[s_i]}\sigma \ldots, u_n\sigma)$, or $f(\bar{u})\sigma{:}s \rightarrow_{R/\mathcal{E}} x'_{[s_i]}\sigma{:}s_i$, and σ is a solution for G in $\rightarrow_{R/\mathcal{E}}$.

2. Completeness: We prove that if ρ is an $R/\mathcal{E}$-normalized idempotent answer for a reachability problem G in $\rightarrow_{R/\mathcal{E}}$ $(=\!\!\rightarrow_{ER,B})$, then there is σ idempotent, with $\rho \ll_{\mathcal{E}} \sigma_{vars(G)}$, such that $G \rightsquigarrow^*_\sigma \square$. Inferred sorts are treated as in the proof of completeness of the calculus for unification. We don't show the inferred sorts here.

   The proof uses induction in the number of calculus rules applied from Figure 3.3:

   Base step: one rule. Then $G = u{:}s \Rightarrow v{:}t$, $u\rho =_{\mathcal{E}} v\rho$, $u\rho : s$, and $v\rho : t$, so $u{:}s \Rightarrow v{:}t \rightsquigarrow_{[X]} u{:}s = v{:}t \rightsquigarrow^*_\sigma \square$, with $\rho \ll_{\mathcal{E}} \sigma_{vars(G)}$ by correctness of the calculus for unification.

   Induction step: more than one rule. We prove a required result, followed by the proof of the induction step.

   (a) We prove that if $G = u{:}s \Rightarrow^1 x_{[s]}{:}[s], G'$, $u\rho : s$, and there exists a term v in $\mathcal{T}_\Sigma$ such that $u\rho \xrightarrow[c,p,\theta]{}{}^1_{R(E),B} v$, then there exists σ, with $\rho \cup \{x_{[s]}{:}[s] \mapsto v\} \ll_{\mathcal{E}} \sigma_{vars(G)}$, such that $G \rightsquigarrow^*_\sigma G'\sigma$.

   By definition, p in $Pos(u\rho)$ has the form $p_1.\cdots.p_n$, c has the form $c : l \rightarrow r \text{ if } C$, with fresh variables, $u\rho|_p =_B l\theta$, $C\theta$ holds under $\rightarrow_{ER,B}$, and $v = (u\rho)[r\theta]_p$. By the same reasoning that we used in the proof of the completeness of the calculus for unification, p must be a nonvariable position in u, otherwise ρ would not be $R/\mathcal{E}$-normalized, so $u|_p\rho = u\rho|_p =_B l\theta$.

   Let $\sigma_1 = \{x_{[s]} \mapsto u[x'_{[t]}]_p\}$, $t = ls(u|_p)$, $q_i = p_1.\cdots.p_i$, and $t_i = ls(u|_{q_i})$, for $1 \leq i < n$. As $u|_p\rho =_B l\theta$ and c has fresh variables, then there exists a substitution $\sigma_2 (= \rho' \cup \theta')$ in $CSU_B(u|_p{=}l)$, with $\rho \ll_{\mathcal{E}} \rho'$, $\theta \ll_{\mathcal{E}} \theta'$, so $G \rightsquigarrow^*_{[I],\sigma_1} u|_p{:}t \Rightarrow^1 x'_{[t]}{:}[t], u{:}s, u|_{q_1}{:}t_1, \cdots, u|_{q_{n-1}}{:}t_{n-1}, G'\sigma_1 = G_1$. We can discard the subgoals $u|_{q_1}{:}t_1, \cdots, u|_{q_{n-1}}{:}t_{n-1}$, that will hold trivially for any substitution, so $G_1 \rightsquigarrow_{[R],\sigma_2} C\sigma_2, u\sigma_2{:}s, u\sigma_2|_p{:}t, G'\sigma_1\sigma_2 = G_2$. Again, we can discard the subgoal $u\sigma_2|_p{:}t$ that will hold trivially for any substitution.

   As $\sigma_2 = \rho' \cup \theta'$, then $G_2 = C\theta', u\rho'{:}s, G'\sigma_1\sigma_2$ so, as $C\theta$ holds under $\rightarrow_{ER,B}$ and $\theta \ll_{\mathcal{E}} \theta'$, there exists σ_3 such that $\theta \ll_{\mathcal{E}} (\theta'\sigma_3)_{vars(G_2)}$, $\rho \ll_{\mathcal{E}} (\rho'\sigma_3)_{vars(G_2)}$, and $G_2 \rightsquigarrow^*_{\sigma_3} u\rho'\sigma_3{:}s, G'\sigma_1\sigma_2\sigma_3 = G_3$.

   As $u\rho : s$ and $\rho \ll_{\mathcal{E}} \rho'\sigma_3$ then, by completeness of the calculus for unification, there exists σ_4 such that $\rho \ll_{\mathcal{E}} (\rho'\sigma_3\sigma_4)_{vars(G_3)}$ and $G_3 \rightsquigarrow^*_{\sigma_4} G'\sigma_1\sigma_2\sigma_3\sigma_4$. Then, $\sigma = \sigma_1\sigma_2\sigma_3\sigma_4$ and $\rho \cup \{x_{[s]}{:}[s] \mapsto v\} \ll_{\mathcal{E}} \sigma$.

   (b) We prove that if $u\rho \rightarrow_{R/\mathcal{E}} v\rho$ (so $u\rho \rightarrow_{ER,B} v\rho$), $u\rho : s$, $v\rho : t$, and ρ is a solution of G', then $G = u{:}s \Rightarrow v{:}t, G' \rightsquigarrow^*_\sigma \square$, with $\rho \ll_{\mathcal{E}} \sigma_{vars(G)}$. We distinguish two cases:

   - $u\rho =_{\mathcal{E}} v\rho$. Then $u{:}s \Rightarrow v{:}t, G' \rightsquigarrow_{[X]} u{:}s = v{:}t \rightsquigarrow^*_{\sigma'} G'\sigma'$, with $\rho \ll_{\mathcal{E}} \sigma'_{vars(G)}$ by correctness of the calculus for unification, so there exists ρ' such that $\sigma' \cdot \rho' =_{\mathcal{E}} \rho$.

As ρ is a solution for G', then ρ' is a solution for $G'\sigma'$. By I.H., there exists σ'', with $\rho' \ll_{\mathcal{E}} \sigma''_{vars(G'\sigma')}$, such that $G'\sigma' \rightsquigarrow^*_{\sigma''} \square$, let $\sigma = \sigma'\sigma''$. Then, $G \rightsquigarrow^*_{\sigma} \square$ and $\rho \ll_{\mathcal{E}} \sigma_{vars(G)}$.

- $u\rho \neq_{\mathcal{E}} v\rho$. Then rule transitivity is the first one to be applied. Recall that $\rightarrow^1_{ER,B}$ is $\rightarrow^*_{E,B};\rightarrow^1_{R(E),B}$ and $\rightarrow_{ER,B}$ $(=\rightarrow_{R/\mathcal{E}})$ is $\rightarrow^*_{ER,B};=_E$, so there exist terms u' and v' in $\mathcal{T}_\Sigma$ such that $u\rho =_{\mathcal{E}} u' \rightarrow^1_{R(E),B} v' \rightarrow_{ER,B} v\rho$. As $u\rho : s$ and $u\rho =_{\mathcal{E}} u'$ then $u' : s$ and then, as $u' \rightarrow^1_{R(E),B} v'$, also $[s] = [ls(v')]$.
  As $u{:}s \Rightarrow v{:}t, G' \rightsquigarrow_{[t]} u{:}s \rightarrow x_{[s]}{:}s, x_{[s]}{:}s \Rightarrow^1 x'_{[s]}{:}[s], x'_{[s]}{:}[s] \Rightarrow v{:}t, G' = G_1$ and $\rho' = \rho \cup \{x_{[s]} \mapsto u'\}$ is a solution of $G'_1 = u{:}s \rightarrow x_{[s]}{:}s$, because $u\rho =_{\mathcal{E}} u'$, then, by completeness of the calculus for unification, there exists σ', with $\rho' \ll_B \sigma'_{vars(G'_1)}$, so there exists σ'_1 such that $\rho' =_B \sigma'_{vars(G'_1)} \cdot \sigma'_1$ and $u' =_B x_{[s]}\sigma'\sigma'_1$, where $G'_1 \rightsquigarrow^*_{\sigma'} \square$, and $G_1 \rightsquigarrow^*_{\sigma'} (x_{[s]}{:}s \Rightarrow^1 x'_{[s]}{:}[s], x'_{[s]}{:}[s] \Rightarrow v{:}t, G')\sigma' = G_2$.
  As $u' \rightarrow^1_{R(E),B} v'$ and $u' =_B x_{[s]}\sigma'\sigma'_1$ then, by $\mathcal{E}$-consistency of $\rightarrow^1_{R(E),B}$ with B, there exists a term v'' in $\mathcal{T}_\Sigma$ such that $v' =_{\mathcal{E}} v''$ and $x_{[s]}\sigma'\sigma'_1 \rightarrow^1_{R(E),B} v''$, so $\rho'' = \sigma'_1 \cup \{x'_{[s]} \mapsto v''\}$ is a solution of $G'_2 = x_{[s]}\sigma'{:}s \Rightarrow^1 x'_{[s]}{:}[s]$.
  By (a), there exists σ'', with $\rho'' \ll_{\mathcal{E}} \sigma''_{vars(G'_2)}$, so there exists σ''_1 such that $\rho'' =_{\mathcal{E}} \sigma''_{vars(G'_2)} \cdot \sigma''_1$ and $v'' =_{\mathcal{E}} x'_{[s]}\sigma''\sigma''_1$, where $G_2 \rightsquigarrow^*_{\sigma''} (x'_{[s]}{:}[s] \Rightarrow v{:}t, G')\sigma'\sigma'' = G_3$.
  As $v' \rightarrow_{ER,B} v\rho$, so $v' \rightarrow_{R/\mathcal{E}} v\rho$, $v' =_{\mathcal{E}} v'' =_{\mathcal{E}} x'_{[s]}\sigma''\sigma''_1 = x'_{[s]}\sigma'\sigma''\sigma''_1$, and $v\rho = v\rho' =_B v\sigma'\sigma'_1 = v\sigma'\rho'' =_{\mathcal{E}} v\sigma'\sigma''\sigma''_1$, then $x'_{[s]}\sigma'\sigma''\sigma''_1 \rightarrow_{R/\mathcal{E}} v\sigma'\sigma''\sigma''_1$, so σ''_1 is a solution of $G'_3 = (x'_{[s]}{:}[s] \Rightarrow v{:}t)\sigma'\sigma''$.
  By I.H., there exists σ''', with $\sigma''_1 \ll_{\mathcal{E}} \sigma'''_{vars(G_3)}$, so there exists σ'''_1 such that $\sigma''_1 =_{\mathcal{E}} \sigma'''_{vars(G_3)} \cdot \sigma'''_1$ and $G_3 \rightsquigarrow^*_{\sigma'''} G'\sigma'\sigma''\sigma''' = G_4$, let $\alpha = \sigma'\sigma''\sigma'''$.
  As $vars(G') \subseteq vars(G)$ and $(\alpha\sigma'''_1)_{vars(G)} = (\sigma'\sigma''\sigma'''\sigma'''_1)_{vars(G)} =_E$ $(\sigma'\sigma''\sigma''_1)_{vars(G)} =_E (\sigma'\rho'')_{vars(G)} =_E (\sigma'\sigma'_1)_{vars(G)} =_B \rho'_{vars(G)} = \rho_{vars(G)}$ then σ'''_1 is a solution of G_4.
  By I.H., there exists β, with $\sigma'''_1 \ll_{\mathcal{E}} \beta_{vars(G_4)}$, so there exists γ such that $\sigma'''_1 =_{\mathcal{E}} \beta_{vars(G_4)} \cdot \gamma$ and $G_4 \rightsquigarrow^*_{\beta} \square$.
  Let $\sigma = \alpha\beta$. As $V_{G'} \subseteq vars(G)$, $(\alpha\sigma'''_1)_{vars(G)} =_{\mathcal{E}} \rho_{vars(G)}$ and $\sigma'''_1 =_{\mathcal{E}} \beta_{vars(G_4)} \cdot \gamma$ then $\sigma_{vars(G)} \cdot \gamma = (\sigma\gamma)_{vars(G)} = (\alpha\beta\gamma)_{vars(G)} =_{\mathcal{E}} (\alpha\sigma'''_1)_{vars(G)} = (\sigma'\sigma''\sigma'''\sigma'''_1)_{vars(G)} =_{\mathcal{E}} \rho_{vars(G)}$, so $\rho \ll_{\mathcal{E}} \rho_{vars(G)} \ll_{\mathcal{E}} \sigma_{vars(G)}$, i.e., $\rho \ll_{\mathcal{E}} \sigma_{vars(G)}$.

$\square$

Chapter 4

Sentence-normalized conditional narrowing modulo

Once we have proved that conditional narrowing modulo is achievable, we address one of the sources of state explosion in the first calculus: the fact that unification and reachability steps can be interleaved in many ways in the narrowing paths generated by the calculus.

Two versions of the calculus in this chapter have been published [AMPP15, AMPP18]. We discuss in this chapter the second one, which is a major revision of the former.

Our main contributions in this chapter are the definition of two new concepts, *fresh pattern property* (FPP from now on) and *narrowable rewrite theory*, and the development of two new narrowing calculi for these definitions, with the following characteristics:

- a larger class of rewrite theories is accepted by the calculus in comparison with the first calculus, admitting extra variables anywhere in the rules;
- a larger class of reachability goals is admitted for solving, compared to the first calculus;
- all terms are normalized before each calculus step, reducing the state space, i.e., some unification steps take precedence over any other feasible narrowing step;
- only normalized instantiations are allowed for reachability terms and extra variables, also reducing the state space; and
- both calculi are sound and weakly complete, i.e., complete with respect to idempotent normalized answers.

4.1 Concurrency specification example

A concurrency specification will be used as running example to explain the definitions in a less abstract way. We review the needed terms. There are `Users` (abbreviated to `u`) `u1`, `u2`, `u3`, and `Tools` (`t`) `t1`, `t2`, `t3`. Several `Users`, separated by commas, are a `UserSet` (`us`) if all the `Users` are different. `emptyU` is the empty `UserSet`. Several `Tools`, separated by semicolons, are a `ToolBox` (`tb`). There will be two `ToolBoxes`, the second one can be seen as a workbench. `emptyT` is the empty `ToolBox`. Each `User` needs two different `Tools` to work which can only be grabbed from the workbench: `u1` needs `t2` and `t3`, `u2`

needs t1 and t3, u3 needs t1 and t2. We also have natural numbers, called Nat (n), with constant 0 and function successor s. We can count the number of elements in a ToolBox, obtaining a Nat, and compare two Nats with the function $<$ obtaining a Boolean (b) value of ok when the comparison holds. A State (st) is composed of two UserSets, and two ToolBoxes, separated by | symbols. The first UserSet holds the Users that are not working; the second UserSet holds the Users that are working. The first ToolBox is the main one, while the second ToolBox is the workbench. There are two conditions that a State must verify: first, the union of both UserSets must be a UserSet, i.e., each User appears only once in the State; second, due to the size of the ToolBox, the total number of Tools, including those being used, cannot exceed four. The initial State is called init.

The rules that a State must follow are:

1. In the initial State nobody is working, and there are no Tools in the workbench.
2. When the workbench is empty, any two Tools that may be in the first ToolBox can be put in the workbench.
3. When there are two Tools in the workbench and a User who is not working needs those tools, he can grab them and work.
4. When a User finishes working, he puts the two Tools that he was using back in the first ToolBox.

In Maude, the specification is:

```
mod CONCURRENCY is
sorts User UserSet Tool ToolBox Nat Boolean State .
subsorts User < UserSet .
subsorts Tool < ToolBox .

ops u1 u2 u3 : -> User .
op emptyU : -> UserSet .
ops t1 t2 t3 : -> Tool .
op emptyT : -> ToolBox .
op 0 : -> Nat .
op s : Nat -> Nat .
op ok : -> Boolean .
op init : -> State .
op _,_ : [UserSet] [UserSet] -> [UserSet] [comm assoc id: emptyU] .
op _;_ : ToolBox ToolBox -> ToolBox [comm assoc id: emptyT] .
op _|_|_|_ : UserSet UserSet ToolBox ToolBox -> [State] .
op count : ToolBox -> Nat .
op _<_ : Nat Nat -> Boolean .

vars M N : Nat .
vars U U' : User .
vars US US' : UserSet .
vars T T' : Tool .
vars TB TB' : ToolBox .
```

```
mb u1, u2 : UserSet .
mb u1, u3 : UserSet .
mb u2, u3 : UserSet .
mb u1, u2, u3 : UserSet .
cmb US | emptyU | TB | TB' : State
    if count(TB ; TB') < s(s(s(s(s(0))))) = ok [label M1] .
cmb US | U | TB | TB' : State
    if U, US : UserSet /\ count(TB ; TB') < s(s(s(0))) = ok .
cmb US | U, U' | emptyT | emptyT : State if U, U', US : UserSet .

eq count(emptyT) = 0 [label E1] .
eq count(T ; TB) = s(count(TB)) [label E2] .
eq 0 < s(N) = ok .
eq s(M) < s(N) = M < N .

crl init => US | emptyU | TB | emptyT
    if US | emptyU | TB | emptyT : State [label R1 nonexec] .
crl US | US' | T ; T' ; TB | emptyT => US | US' | TB | T ; T'
    if US | US' | TB | T ; T' : State [label R2] .
crl u1, US | US' | TB | t2 ; t3 => US | u1, US' | TB | emptyT
    if US | u1, US' | TB | emptyT : State [label R3] .
crl u2, US | US' | TB | t1 ; t3 => US | u2, US' | TB | emptyT
    if US | u2, US' | TB | emptyT : State [label R4] .
crl u3, US | US' | TB | t1 ; t2 => US | u3, US' | TB | emptyT
    if US | u3, US' | TB | emptyT : State [label R5] .
crl US | u1, US' | TB | TB' => u1, US | US' | t2 ; t3 ; TB | TB'
    if u1, US | US' | t2 ; t3 ; TB | TB' : State [label R6] .
crl US | u2, US' | TB | TB' => u2, US | US' | t1 ; t3 ; TB | TB'
    if u2, US | US' | t1 ; t3 ; TB | TB' : State [label R7] .
crl US | u3, US' | TB | TB' => u3, US | US' | t1 ; t2 ; TB | TB'
    if u3, US | US' | t1 ; t2 ; TB | TB' : State [label R8] .
endm
```

The `nonexec` attribute on rule `R1` instructs Maude to not care about the new variables that appear in the right side of the rule when checking its syntax. Maude will not use the rule for rewriting purposes. The reflective capabilities of Maude allow the use of this rule and all the others through the metalevel. The rule is used for narrowing purposes in the example in Section 4.6. The prototypes for the other calculi in this dissertation use the metalevel version of the given reachability problem and rewrite theory, including its `nonexec` rules, to look for solutions.

4.1.1 Signature

In the concurrency specification example, $\Sigma = (K, S, F)$ is:

- $K = \{[\mathtt{us}], [\mathtt{tb}], [\mathtt{n}], [\mathtt{b}], [\mathtt{st}]\}$,
- $S = \{S_{[\mathtt{us}]}, S_{[\mathtt{tb}]}, S_{[\mathtt{n}]}, S_{[\mathtt{b}]}, S_{[\mathtt{st}]}\}$, where

$S_{[\mathtt{us}]} = \{\mathtt{u}, \mathtt{us}\}, S_{[\mathtt{tb}]} = \{\mathtt{t}, \mathtt{tb}\}, S_{[\mathtt{n}]} = \{\mathtt{n}\}, S_{[\mathtt{b}]} = \{\mathtt{b}\}, S_{[\mathtt{st}]} = \{\mathtt{st}\}$.

- $F = \{\{\_ \mid \_ \mid \_ \mid \_\}_{\mathtt{us\ us\ tb\ tb},[\mathtt{st}]}, \{\_, \_\}_{[\mathtt{us}]\ [\mathtt{us}],[\mathtt{us}]}, \{\_;\_\}_{\mathtt{tb\ tb},\mathtt{tb}}, \{\mathtt{count}\}_{\mathtt{tb},\mathtt{n}}, \{\mathtt{s}\}_{\mathtt{n\ n},\mathtt{n}}, \{\_<\_\}_{\mathtt{n\ n},\mathtt{b}}, \{\mathtt{u1}, \mathtt{u2}, \mathtt{u3}\}_{\mathtt{u}}, \{\mathtt{emptyU}\}_{\mathtt{us}}, \{\mathtt{t1}, \mathtt{t2}, \mathtt{t3}\}_{\mathtt{t}}, \{\mathtt{emptyT}\}_{\mathtt{tb}}, \{\mathtt{0}\}_{\mathtt{n}}, \{\mathtt{ok}\}_{\mathtt{b}}, \{\mathtt{init}\}_{\mathtt{s}}\}$

4.1.2 MEL theory

The MEL theory for the concurrency specification example consists of $\Sigma = (K, S, F)$ and the set of MEL sentences $\mathcal{E}$ in Table 4.1, where the first row of MEL sentences represents the subsort ordering in S. We omit the implicit subsorts for each kind, and the implicit memberships for each constant, variable, and kinded function. By the executability requirements of the theory, the associativity, commutativity, and identity axioms are defined over kinds.

$x_{\mathtt{u}} : \mathtt{us}$	$x_{\mathtt{t}} : \mathtt{tb}$	(subsorts)
$(x_{[\mathtt{us}]}, y_{[\mathtt{us}]}), z_{[\mathtt{us}]} = x_{[\mathtt{us}]}, (y_{[\mathtt{us}]}, z_{[\mathtt{us}]})$	$(x_{[\mathtt{tb}]}; y_{[\mathtt{tb}]}); z_{[\mathtt{tb}]} = x_{[\mathtt{tb}]}; (y_{[\mathtt{tb}]}; z_{[\mathtt{tb}]})$	(associativity)
$x_{[\mathtt{us}]}, y_{[\mathtt{us}]} = y_{[\mathtt{us}]}, x_{[\mathtt{us}]}$	$x_{[\mathtt{tb}]}; y_{[\mathtt{tb}]} = y_{[\mathtt{tb}]}; x_{[\mathtt{tb}]}$	(commutativity)
$x_{[\mathtt{us}]}, \mathtt{emptyU} = x_{[\mathtt{us}]}$	$x_{[\mathtt{tb}]}; \mathtt{emptyT} = x_{[\mathtt{tb}]}$	(identity)

$x_{\mathtt{us}} \mid \mathtt{emptyU} \mid z_{\mathtt{tb}} \mid w_{\mathtt{tb}} : \mathtt{st}$ *if* $\mathtt{count}(z_{\mathtt{tb}}; w_{\mathtt{tb}}) < \mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{0}))))) = \mathtt{ok}$

$x_{\mathtt{us}} \mid y_{\mathtt{u}} \mid z_{\mathtt{tb}} \mid w_{\mathtt{tb}} : \mathtt{st}$ *if* $y_{\mathtt{u}}, x_{\mathtt{us}} : \mathtt{us} \wedge \mathtt{count}(z_{\mathtt{tb}}; w_{\mathtt{tb}}) < \mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{0}))) = \mathtt{ok}$

$x_{\mathtt{us}} \mid y_{\mathtt{u}}, y'_{\mathtt{u}} \mid \mathtt{emptyT} \mid \mathtt{emptyT} : \mathtt{st}$ *if* $y_{\mathtt{u}}, y'_{\mathtt{u}}, x_{\mathtt{us}} : \mathtt{us}$

$\mathtt{u1}, \mathtt{u2} : \mathtt{us} \quad \mathtt{u1}, \mathtt{u3} : \mathtt{us} \quad \mathtt{u2}, \mathtt{u3} : \mathtt{us} \quad \mathtt{u1}, \mathtt{u2}, \mathtt{u3} : \mathtt{us}$

$\mathtt{count}(\mathtt{emptyT}) = \mathtt{0} \quad \mathtt{count}(x_{\mathtt{t}}; y_{\mathtt{tb}}) = \mathtt{s}(\mathtt{count}(y_{\mathtt{tb}}))$

$\mathtt{0} < \mathtt{s}(x_{\mathtt{n}}) = \mathtt{ok} \quad \mathtt{s}(x_{\mathtt{n}}) < \mathtt{s}(y_{\mathtt{n}}) = x_{\mathtt{n}} < y_{\mathtt{n}}$

Table 4.1: MEL sentences for the concurrency specification example

The conditional membership sentences for `State` (`st`) take into account that, when checking the total number of `Tools`, any working `User` is holding two `Tools`. When two `Users` are working, both `ToolBoxes` must be empty. If necessary it is also checked that the union of the working `UserSet` and the non working `UserSet` is also a `UserSet`.

4.1.3 Rewrite theory

In the concurrency specification example, R has as elements the conditional rewrite rules in Table 4.2.

4.2 Narrowing and narrowable rewrite theories

In this chapter we will use a stricter relation $\rightarrow^1_{ER,B}$ compared to the one in Section 2.3.6. Instead of allowing any number of E, B rewrite steps before an $R(E), B$ rewrite step, now we force the generation of an E, B-normalized term before an $R(E), B$ rewrite step. Nonetheless, we will prove that this new and stricter relation is enough to mimic $\rightarrow^1_{R/\mathcal{E}}$.

Definition 3 (ER, B-rewriting)**.** *We define* $\rightarrow^1_{ER,B}$ *as* $(\rightarrow^!_{E,B}; \rightarrow^1_{R(E),B})$.

$$\texttt{init} \rightarrow x_{\texttt{us}} \mid \texttt{emptyU} \mid z_{\texttt{tb}} \mid \texttt{emptyT}\ \textit{if}\ x_{\texttt{us}} \mid \texttt{emptyU} \mid z_{\texttt{tb}} \mid \texttt{emptyT} : \texttt{s}$$

$$x_{\texttt{us}} \mid y_{\texttt{us}} \mid u_{\texttt{t}}; v_{\texttt{t}}; z_{\texttt{tb}} \mid \texttt{emptyT} \rightarrow x_{\texttt{us}} \mid y_{\texttt{us}} \mid z_{\texttt{tb}} \mid u_{\texttt{t}}; v_{\texttt{t}}\ \textit{if}\ x_{\texttt{us}} \mid y_{\texttt{us}} \mid z_{\texttt{tb}} \mid u_{\texttt{t}}; v_{\texttt{t}} : \texttt{s}$$

$$\texttt{u1}, x_{\texttt{us}} \mid y_{\texttt{us}} \mid z_{\texttt{tb}} \mid \texttt{t2}; \texttt{t3} \rightarrow x_{\texttt{us}} \mid \texttt{u1}, y_{\texttt{us}} \mid z_{\texttt{tb}} \mid \texttt{emptyT}\ \textit{if}\ x_{\texttt{us}} \mid \texttt{u1}, y_{\texttt{us}} \mid z_{\texttt{tb}} \mid \texttt{emptyT} : \texttt{s}$$

$$\texttt{u2}, x_{\texttt{us}} \mid y_{\texttt{us}} \mid z_{\texttt{tb}} \mid \texttt{t1}; \texttt{t3} \rightarrow x_{\texttt{us}} \mid \texttt{u2}, y_{\texttt{us}} \mid z_{\texttt{tb}} \mid \texttt{emptyT}\ \textit{if}\ x_{\texttt{us}} \mid \texttt{u2}, y_{\texttt{us}} \mid z_{\texttt{tb}} \mid \texttt{emptyT} : \texttt{s}$$

$$\texttt{u3}, x_{\texttt{us}} \mid y_{\texttt{us}} \mid z_{\texttt{tb}} \mid \texttt{t1}; \texttt{t2} \rightarrow x_{\texttt{us}} \mid \texttt{u3}, y_{\texttt{us}} \mid z_{\texttt{tb}} \mid \texttt{emptyT}\ \textit{if}\ x_{\texttt{us}} \mid \texttt{u3}, y_{\texttt{us}} \mid z_{\texttt{tb}} \mid \texttt{emptyT} : \texttt{s}$$

$$x_{\texttt{us}} \mid \texttt{u1}, y_{\texttt{us}} \mid z_{\texttt{tb}} \mid w_{\texttt{tb}} \rightarrow \texttt{u1}, x_{\texttt{us}} \mid y_{\texttt{us}} \mid \texttt{t2}; \texttt{t3}; z_{\texttt{tb}} \mid w_{\texttt{tb}}\ \textit{if}\ \texttt{u1}, x_{\texttt{us}} \mid y_{\texttt{us}} \mid \texttt{t2}; \texttt{t3}; z_{\texttt{tb}} \mid w_{\texttt{tb}} : \texttt{s}$$

$$x_{\texttt{us}} \mid \texttt{u2}, y_{\texttt{us}} \mid z_{\texttt{tb}} \mid w_{\texttt{tb}} \rightarrow \texttt{u2}, x_{\texttt{us}} \mid y_{\texttt{us}} \mid \texttt{t1}; \texttt{t3}; z_{\texttt{tb}} \mid w_{\texttt{tb}}\ \textit{if}\ \texttt{u2}, x_{\texttt{us}} \mid y_{\texttt{us}} \mid \texttt{t1}; \texttt{t3}; z_{\texttt{tb}} \mid w_{\texttt{tb}} : \texttt{s}$$

$$x_{\texttt{us}} \mid \texttt{u3}, y_{\texttt{us}} \mid z_{\texttt{tb}} \mid w_{\texttt{tb}} \rightarrow \texttt{u3}, x_{\texttt{us}} \mid y_{\texttt{us}} \mid \texttt{t1}; \texttt{t2}; z_{\texttt{tb}} \mid w_{\texttt{tb}}\ \textit{if}\ \texttt{u3}, x_{\texttt{us}} \mid y_{\texttt{us}} \mid \texttt{t1}; \texttt{t2}; z_{\texttt{tb}} \mid w_{\texttt{tb}} : \texttt{s}$$

Table 4.2: Rewrite rules for the concurrency example

The setting for narrowing that we present in this chapter relaxes the requirements for the new variables that may appear in the rules in R, so $\mathcal{R}$ will stop being executable and become *narrowable*, a new concept that is formalized later. With this definition, we can use the metalevel of Maude to solve narrowing problems, but Maude's rewrite engine cannot be used to verify the solutions obtained for the theories belonging to this wider class of narrowable rewrite theories, except when they are also executable.

The plan is to replace $=_{\mathcal{E}}$ and $:_{\mathcal{E}}$ with $\rightarrow_{E,B}$, and $\rightarrow_{R/\mathcal{E}}$ with $\rightarrow_{ER,B}$, but, as explained previously in Section 2.3.8, there is a problem that must be addressed to make these replacements feasible. Consider a rewrite theory $\mathcal{R}$ with only one sort s, and whose only rule is $f(a,b) \rightarrow c$, where f is associative and commutative ($E = \emptyset$). The term $f(f(a,a),b)$ is a normal form in $\rightarrow^1_{ER,B}$, but $f(f(a,a),b) \rightarrow^1_{R/\mathcal{E}} f(a,c)$, because $f(f(a,a),b) =_B f(a,f(a,b))$, so the relations are different. This problem would not happen if $\mathcal{R}$ had another rule $f(x_{[s]}, f(a,b)) \rightarrow f(x_{[s]}, c)$ that could be applied on top of the term $f(f(a,a),b)$ with matching $x_{[s]} \mapsto a$, modulo associativity and commutativity, leading to $f(f(a,a),b) \rightarrow^1_{ER,B} f(a,c)$. Rewrite theories, including those associated to a MEL theory, that have these rules, avoiding such problems, are called *closed under B-extensions* [Mes17].

4.2.1 Closure under B-extensions

Let $\mathcal{R} = (\Sigma, E \cup B, R)$ be a rewrite theory, and let $c : l \rightarrow r\ \textit{if}\ C$ be a rule in R. Without loss of generality we asume that $vars(B) \cap vars(c) = \emptyset$. If this is not the case, *only the variables of B* will be renamed; the variables of c *will never be* renamed. We then define the set of *B-extensions* of c as the set:

$$Ext_B(c) = \{u[l]_p \rightarrow u[r]_p\, \textit{if}\, C \mid u = v \in B \cup B^{-1} \wedge p \in Pos_\Sigma(u) - \{\epsilon\} \wedge CSU_B(l = u|_p) \neq \emptyset\}$$

where, by definition, $B^{-1} = \{v = u \mid u = v \in B\}$.

Given two rules $l \rightarrow r\ \textit{if}\ C$ and $l' \rightarrow r'\ \textit{if}\ C$ with the *same* condition C we say that $l \rightarrow r\ \textit{if}\ C$ *B-subsumes* $l' \rightarrow r'\ \textit{if}\ C$ iff there is a substitution σ such that: (i) $dom(\sigma) \cap vars(C) = \emptyset$, (ii) $l' =_B l\sigma$, and (iii) $r' =_B r\sigma$.

In the general case presented in [Mes17], computing closures as a fixed point of an algorithm may generate sets of rules that are infinite.

From now on, we will consider as valid those rewrite theories $\mathcal{R} = (\Sigma, E \cup B, R)$ whose axioms B are any combination of **associativity**, **commutativity**, and **identity** (ACU

Figure 4.1: Strict coherence of $\rightarrow^1_{E,B}$ and $\rightarrow^1_{R(E),B}$

rewrite theories).

Let $c : f(t_1, t_2) \rightarrow t_3 \in R$ .The only axiom that requires the addition of new rules in $Ext_B(c)$ is associativity.

- If f has the associative property, then $Ext_B(c)$ will have a rule $f(x_{[s]}, f(t_1, t_2)) \rightarrow f(x_{[s]}, t_3)$ (it could also be $f(f(t_1, t_2), x_{[s]}) \rightarrow f(t_3, x_{[s]})$).
- If f has the commutative property, $u = v \in B \cup B^{-1}$, with $u = f(x_{[s]}, y_{[s]})$ and $v = f(y_{[s]}, x_{[s]})$, then u has no non-variable subterms so, by the definition of $Ext_B(c)$, no rule has to be added.
- If f has the identity property, assuming 0 is the identity for f, $f(x_{[s]}, 0) = x_{[s]}$, the non-variable subterm 0 only matches rules of the form $0 \rightarrow t$ yielding a rule $f(x_{[s]}, 0) \rightarrow f(x_{[s]}, t)$, which is subsumed by the original rule $0 \rightarrow t$ with the substitution $\{x_{[s]} \mapsto 0\}$, since $f(0, 0) =_B 0$ and $f(0, t) =_B t$, so also no rule has to be added.

Definition 4 (Closed under B-extensions rewrite theory)**.** *We call* $\mathcal{R} = (\Sigma, E \cup B, R)$ closed under B-extensions *iff for any rule* c *in* R*, each rule in* $Ext_B(c)$ *is subsumed by some rule in* R*.*

Theorem 2 and Corollary 3 in [Mes17] can be applied in a straightfordward way to $\rightarrow^1_{E,B}$ and $\rightarrow^1_{R(E),B}$, and we get the following Lemmas.

Lemma 2 (Strict coherence of $\rightarrow^1_{E,B}$)**.** *Given a* MEL *theory* $(\Sigma, E \cup B)$ *and its associated rewrite theory* $\mathcal{R}_E$*, if* $\mathcal{R}_E$ *is closed under* B*-extensions, then* $\rightarrow^1_{E,B}$ *is strictly coherent, i.e., for all* t_1, t_2, t_3 *if* $t_1 \rightarrow^1_{E,B} t_2$ *and* $t_1 =_B t_3$*, then there exists* t_4 *such that* $t_3 \rightarrow^1_{E,B} t_4$ *and* $t_2 =_B t_4$*.*

A diagram of Lemma 2 is shown in the left side of Fig. 4.1, where filled lines are used for universal quantification and dotted lines are used for existential quantification.

Lemma 3 (Strict coherence of $\rightarrow^1_{R(E),B}$)**.** *Given a rewrite theory* $\mathcal{R} = (\Sigma, E \cup B, R)$*, if* R *is closed under* B*-extensions then* $\rightarrow^1_{R(E),B}$ *is strictly coherent, i.e., for all* t_1, t_2, t_3 *if* $t_1 \rightarrow^1_{R(E),B} t_2$ *and* $t_1 =_B t_3$*, then there exists* t_4 *such that* $t_3 \rightarrow^1_{R(E),B} t_4$ *and* $t_2 =_B t_4$ *(see Fig. 4.1, right side).*

Strict coherence of $\rightarrow^1_{E,B}$ and $\rightarrow^1_{R(E),B}$ will be used later in the chapter to prove the equivalence of $\rightarrow^1_{R/\mathcal{E}}$ and $\rightarrow^!_{E,B}; \rightarrow^1_{R(E),B}; =_{\mathcal{E}}$ for narrowable rewrite theories.

Definition 5 (Normal form)**.** *Given a* MEL *theory* $(\Sigma, E \cup B)$ *and a term* $t \in T_\Sigma(\mathcal{X})$*, if the associated rewrite theory* $\mathcal{R}_E = (\Sigma', B, R_E)$ *is closed under* B*-extensions and* $\rightarrow^1_{E,B}$ *is sort-decreasing, terminating, and confluent, then the* normal form *of* t *is the unique (up to* B*-equivalence and new variable renaming) canonical term* $t{\downarrow_{E,B}}$*.*

When the MEL theory is also admissible then $t{\downarrow_{E,B}}$ is unique up to B-equivalence.

4.2.2 FPP theories. Narrowable rewrite theory

In this section we will first define the new concepts of FPP MEL and rewrite theory. Then we show how any MEL or rewrite theory can be turned into an equivalent FPP one by changing the definition of some equations or rules. Finally, we will define the concept of narrowable rewrite theory, which has the following assumptions relaxed with respect to executable rewrite theories [CDE+07]:

- a narrowable rewrite theory doesn't need to be operationally terminating,
- it admits extra variables anywhere in the conditions, and
- it has no restrictions on equational, membership or rewrite conditions; only matching equations have restrictions.

We want to apply narrowing only to normal terms, reducing the state space of our narrowing problems. Matching with normal forms may not be safe in general. The use of FPP theories will ensure the completeness of this procedure [CEM14].

Definition 6 (FPP MEL theory)**.** *A* MEL *theory* $(\Sigma, E \cup B)$ *has the* Fresh Pattern Property (FPP) *if for each sentence* $t = t'$ *if* $\bigwedge_{i=1}^{n} A_i$ *or* $t : s$ *if* $\bigwedge_{i=1}^{n} A_i$ *in* E*, if* A_i *has the form* $u_i := v_i$*, then* $(vars(t) \cup vars(v_i) \cup \bigcup_{j=1}^{i-1} vars(A_j)) \cap vars(u_i) = \emptyset$*.*

A matching equation in an FPP MEL theory is similar to a "let" expression in functional programming, allowing us to define locally some value that is needed later in the condition, or in the right part of a conditional equation. For narrowing purposes, the restriction that we impose on matching equations will allow us to instantiate the extra variables in u_i (we call them *matching variables*) by B-unification of u_i with the normal form of some instance of v_i, instead of performing a needless unification by E-narrowing. The main difference with respect to "let" expressions is that this matching is done modulo the axioms B, so we gain expressiveness.

Example 7. *Let* $(\Sigma, E \cup B)$ *be a* MEL *theory, with sorts* $item(i)$*,* $multiset(m)$*, and* $state(s)$*; subsorts* $i \leq m$*; constants* $a :\to i$*,* $b :\to i$*, and* $empty :\to i$*; functions* $\_\,;\_ : m\ m \to m$ *(with associative, commutative, and identity axioms, i.e., those corresponding to multisets), and* $[\_] : m \to s$*.*

If $E = \{[x_m] = [y_m]$ *if* $a; y_m := x_m\}$ *then* $(\Sigma, E \cup B)$ *is FPP because the* Σ*-equation* $[x_m] = [y_m]$ *applies to states, not to multisets, so* $a; y_m$ *is a* Σ*-pattern, and* y_m *does not appear in the left part of the* Σ*-equation.*

What this equation does is to remove any occurrence of the constant a *in the multiset included within a state, just by matching the multiset with the* Σ*-pattern (modulo the axioms of the function* $\_\,;\_$*), leaving a state holding a multiset whose elements are the remaining* b*'s, or a multiset whose only element is empty, if there were none.*

Example 8. *Consider the* MEL *theory* $(\Sigma, E \cup B)$ *where:*

$$K = \{k\}, S = \{S_k\}, S_k = \{s\}, F = \{\{a, b, c, d\}_s, \{f, [\_\,, \_\,]\}_{ss,s}\},$$

with $B = \emptyset$ *and equations:*

$$E = \{a = b, \quad c = d, \quad f(x, y) = z \text{ if } [x, z] := [x, y]\}$$

The MEL *theory is not FPP because in the conditional equation the variable x appears both in the left side of the matching equation $[x,z] := [x,y]$ and in the left side of the Σ-equation $f(x,y) = z$. Among others, its associated rewrite theory, R_E, has rules:*

$$\{eq(w_k,w_k) \to \mathtt{tt},\ a \to b,\ c \to d,\ f(x,y) \to z \text{ if } [x,y] \to [x,z]\} \subseteq R_E$$

E is admissible; $\to^1_{E,B}$ is confluent, terminating, and sort-decreasing. We have omitted the sort in the variables when it is s. Rewriting the term $f(a,c)$ in $\to^1_{E,B}$ generates the condition $[a,c] \to [a,z]$. If we match $[a,c]$ with $[a,z]$ before rewriting $[a,c]$ in $\to^1_{E,B}$ we get the match $z \mapsto c$, so $f(a,c) \to c$. However, if we rewrite $[a,c]$ to its normal form $[b,d]$ we get the condition $[b,d] \to [a,z]$ that does not match, so $f(a,c)$ cannot be rewritten.

Using FPP theories we can rewrite any term to its normal form before matching. An easy transformation allows us to turn any rewrite or MEL theory into an FPP one

Example 8 (continued)**.** *The transformed FPP* MEL *theory $(\Sigma, E \cup B)$ has equations:*

$$E = \{a = b, \quad c = d, \quad f(x,y) = z \text{ if } [x',z] := [x,y] \wedge x = x'\}$$

where we have added a new variable x', and a new condition $x = x'$ that forces both variables, x and x', to be instantiated to E,B-equivalent terms. Now, in the associated rewrite theory R_E:

$$\{eq(w_k,w_k) \to \mathtt{tt},\ a \to b,\ c \to d,\ f(x,y) \to z \text{ if } [x,y] \to [x',z] \wedge eq(x,x') \to \mathtt{tt}\} \subseteq R_E$$

E is admissible; $\to^1_{E,B}$ is confluent, terminating, and sort-decreasing. Rewriting the term $f(a,c)$ generates the condition $[a,c] \to [x',z] \wedge eq(a,x') \to \mathtt{tt}$ now. Using rules $a \to b$ and $c \to d$ we get $[b,d] \to [x',z] \wedge eq(a,x') \to \mathtt{tt}$, where $[b,d]$ is a normal form. Substitution $\sigma = \{x' \mapsto b, z \mapsto d\}$ solves the first part of the condition, and the second part of the condition becomes $eq(a,b) \to \mathtt{tt}$ which, using the rule $a \to b$, rewrites to $eq(b,b) \to \mathtt{tt}$, true with rule $eq(w_k,w_k) \to \mathtt{tt}$ and substitution $\{w_k \mapsto b\}$. Then $f(a,c)$ rewrites to d, which is the normal form of c, so the rewritings in both examples are E,B-equivalent, i.e., equivalent modulo $\mathcal{E}$.

Definition 7 (FPP rewrite theory)**.** *A rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ is FPP if:*

1. *the* MEL *theory $(\Sigma, E \cup B)$ is FPP and*
2. *for each rule $l \to r$ if $\bigwedge_{i=1}^n A_i$ in R:*
   - *$vars(r) \subseteq vars(l) \cup \bigcup_{i=1}^n vars(A_i)$ and*
   - *if A_i has form $u_i := v_i$, then:*
     - *u_i is a Σ-pattern and*
     - *$(vars(l) \cup vars(v_i) \cup \bigcup_{j=1}^{i-1} vars(A_j)) \cap vars(u_i) = \emptyset$.*

This definition means that for each rule $l \to r$ *if* C in an FPP rewrite theory we admit extra variables anywhere in C, even on the right side of matching equations, but not in the right term r, and again we demand that for matching equations $u_i := v_i$ the variables in u_i haven't appeared before in the rule. We can relax these requirements because we only need the rewrite theories to be narrowable (see definition below), while we need the MEL theories to be executable, so we can get the normal form of any term by E,B-rewriting.

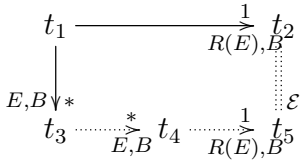

Figure 4.2: $\mathcal{E}$-coherence of $\to^1_{R(E),B}$ with $\to^1_{E,B}$

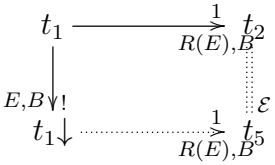

Figure 4.3: Canonical $\mathcal{E}$-coherence of $\to^1_{R(E),B}$ with $\to^1_{E,B}$

Definition 8 (Narrowable rewrite theory)**.** *A rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ is* narrowable *if Σ is preregular modulo B; E, B, and R are finite; no left term in E and R is a variable; the* MEL *theory $(\Sigma, E \cup B)$ is admissible; and $\mathcal{R}$ satisfies the following requirements:*

1. *$\mathcal{R}$ is FPP, and both $\mathcal{R}$ and the associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$ are closed under B-extensions.*

2. *The axioms in B are regular, linear, and sort-preserving. Any variable appearing in an axiom must be a kinded variable. Furthermore, equality modulo B must be decidable and there must exist a finitary matching algorithm modulo B producing a finite number of B-matching substitutions, $Match_B(t_1, t_2) = \{\sigma_i\}_{i=1}^n$ meaning that $t_1 =_B t_2\sigma_i$ for $i = 1, \ldots, n$, or failing otherwise.*

3. *The relation $\to^1_{E,B}$ is sort-decreasing, terminating, confluent, and operationally terminating.*

4. *$\to^1_{R(E),B}$ is* $\mathcal{E}$-coherent *with $\to^1_{E,B}$ (see Fig. 4.2), i.e., for all t_1, t_2, t_3 in $\mathcal{T}_\Sigma(\mathcal{X})$ we have that if $t_1 \to^1_{R(E),B} t_2$ and $t_1 \to^*_{E,B} t_3$ then there exist t_4, t_5 such that $t_3 \to^*_{E,B} t_4$, $t_4 \to^1_{R(E),B} t_5$, and $t_2 =_\mathcal{E} t_5$. Again, we use a diagram with filled lines for universal quantification and dotted lines for existential quantification*

Remark 5. *Figure 4.3 is a special case of Figure 4.2, where we rewrite t_1 to its canonical form $t_1{\downarrow}$, so $t_3 = t_4$. It shows that $\mathcal{E}$-coherence of $\to^1_{R(E),B}$ with $\to^1_{E,B}$ implies that t_1 has at least the same rewrite steps in $\to^1_{ER,B}$ as it has in $\to^1_{R(E),B}$, up to $\mathcal{E}$-equality.*

Remark 6. *For narrowable rewrite theories, where $(\Sigma, \mathcal{E})$ is executable, $=_\mathcal{E}, :=_\mathcal{E}$, and $:_\mathcal{E}$ can be replaced with $\to_{E,B}$. In this case, as both $\to^1_{R(E),B}$ and $\to_{ER,B}$ become restrictions of $\to^1_{R/\mathcal{E}}$ and $\to_{R/\mathcal{E}}$, respectively, then $\to^1_{R(E),B} \subseteq \to^1_{R/\mathcal{E}}$ and $\to_{ER,B} \subseteq \to_{R/\mathcal{E}}$.*

Example 9. *Consider a rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$, where $S = \{s\}$, $F = \{\{a, b, c\}_s, \{f\}_{s,s}\}$, with $B = \emptyset$, $E = \{a = b\}$ and $R = \{f(a) \to c\}$.*

In this theory $f(a) \to^1_{R(E),B} c$, but $f(a) \to^1_{E,B} f(b)$ and $f(b)$ cannot be further rewritten in $\to^1_{R(E),B}$, so the theory is not $\mathcal{E}$-coherent. If we add the rule $f(b) \to c$ to R then $f(a) \to^1_{E,B} f(b) \to^1_{R(E),B} c$, and we have an $\mathcal{E}$-coherent rewrite theory.

Example 10. *The rewrite theory for the concurrency specification example is narrowable if we decompose $\mathcal{E}$ in the following way: the set B contains the associative, commutative, and identity equations in $\mathcal{E}$; the set E contains the rest of equations and all memberships in $\mathcal{E}$.*

For narrowable rewrite theories we can implement $\rightarrow_{R/\mathcal{E}}$ using $\rightarrow_{ER,B}$. This theorem links them and also $\rightarrow^1_{R/\mathcal{E}}$ with $\rightarrow^1_{R(E),B}$.

Theorem 4 (Reduction of $\rightarrow^1_{R/\mathcal{E}}$ to $\rightarrow^1_{R(E),B}$ and $\rightarrow_{R/\mathcal{E}}$ to $\rightarrow_{ER,B}$)**.** *Let $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ be a narrowable rewrite theory. Then, for any t_1, t_2 in $\mathcal{T}_\Sigma(\mathcal{X})$: (i) $t_1 \rightarrow^1_{R/\mathcal{E}} t_2$ if and only if $t_1{\downarrow} \rightarrow^1_{R(E),B} t_3$ for some $t_3 =_\mathcal{E} t_2$ (where $t_3{\downarrow} =_B t_2{\downarrow}$ serves as check) and (ii) $t_1 \rightarrow_{R/\mathcal{E}} t_2$ if and only if $t_1 \rightarrow_{ER,B} t_2$.*

See proof on page 80.

4.2.3 Reachability goal. Solutions

Definition 9 (Reachability goal)**.** *Given a rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$, a* reachability goal *$G$ is a conjunction of the form $u_1 \Rightarrow v_1 \wedge \ldots \wedge u_n \Rightarrow v_n \wedge G'$ where for $1 \leq i \leq n$, $u_i, v_i \in T_\Sigma(\mathcal{X})_{\kappa_i}$ for appropriate κ_i, and G' is a unification goal in $\mathcal{R}_E$ associated to a system of sentences F in the* MEL *theory $(\Sigma, E \cup B)$. The subgoals $u_i \Rightarrow v_i$ can be interleaved with the subgoals in G'. We define $vars(G) = \bigcup_{i=1}^n (vars(u_i) \cup vars(v_i)) \cup vars(G')$.*

Definition 10 (Solution)**.** *A substitution σ is a solution of G if σ is an $\mathcal{E}$-solution for G', and $u_i\sigma \rightarrow_{R/\mathcal{E}} v_i\sigma$, for $1 \leq i \leq n$.*

If the substitution is idempotent we also say that the solution is *idempotent*. We define $\mathcal{E}(G)$ to be the system of sentences $u_1 = v_1 \wedge \ldots \wedge u_n = v_n \wedge F$. We say that σ is a *trivial solution* of G if it is an $\mathcal{E}$-solution for $\mathcal{E}(G)$. We say that G is trivial if the identity substitution *id* is a trivial solution of G.

Theorem 5 (Equivalence of $\mathcal{E}$-solutions for systems of sentences and unification goals)**.** *Given a* MEL *theory $(\Sigma, E \cup B)$ and its associated rewrite theory $\mathcal{R}_E$, if $\rightarrow^1_{E,B}$ is terminating, confluent, sort-decreasing, and closed under B-extensions, then for any system of sentences F, an idempotent normal substitution σ is an $\mathcal{E}$-solution for F iff σ is an $\mathcal{E}$-solution for its associated unification goal (G) in $\rightarrow_{E,B}$.*

See proof on page 82.

As a conclusion, we can verify that σ is an $\mathcal{E}$-solution for F by checking $G\sigma$ using the relation $\rightarrow_{E,B}$. Conversely, if we find an $\mathcal{E}$-solution σ for G, then σ is an $\mathcal{E}$-solution for F.

4.2.4 E, B-narrowing. $R(E), B$-narrowing. ER, B-narrowing.

Three narrowing relations are now presented for narrrowable rewrite theories $\mathcal{R} = (\Sigma, E \cup B, R)$:

- the first one is used, given a term t, to find its instances $t\sigma$ such that $t\sigma \rightarrow^1_{E,B} t'$, for proper t';
- the second one is used, given a term t, to find its instances $t\sigma$ such that $t\sigma \rightarrow^1_{R(E),B} t'$, for proper t';

- the third one, the union of the two previous narrowing relations, is the one that is implemented in the calculus for reachability in this chapter.

Definition 11 (E, B-narrowing). *Given an FPP executable* MEL *theory* $(\Sigma, E \cup B)$*, its associated rewrite theory* $\mathcal{R}_E$*, a term* t *in* $T_\Sigma(\mathcal{X})$*, and a rule* $c : l \to r$ *if* C *in* R_E*, properly renamed so* $vars(c) \cap vars(t) = \emptyset$*, if there exists a non-variable position* p *in* $Pos_\Sigma(t)$*, and a substitution* σ *such that* $t|_p\sigma =_B l\sigma$ *and* $C\sigma$ *holds in* $\to_{E,B}$*, then we write* $t \rightsquigarrow^1_{p,\sigma,E,B} t[r]_p\sigma$ *and say that there is an* E, B-narrowing step *from* t *to* $t[r]_p\sigma$*.*

Definition 12 ($R(E), B$-narrowing). *Given a narrowable rewrite theory* $\mathcal{R} = (\Sigma, E \cup B, R)$*, its associated rewrite theory* $\mathcal{R}_E$*, a term* t *in* $T_\Sigma(\mathcal{X})$*, and a rule* $c : l \to r$ *if* C *in* R*, properly renamed so* $vars(c) \cap vars(t) = \emptyset$*, if there exists a non-variable position* p *in* $Pos_\Sigma(t)$*, and a substitution* σ *such that* $t|_p\sigma =_B l\sigma$ *and* $C\sigma$ *holds in* $\to_{ER,B}$*, then we write* $t \rightsquigarrow^1_{p,\sigma,R(E),B} t[r]_p\sigma$ *and say that there is an* $R(E), B$-narrowing step *from* t *to* $t[r]_p\sigma$*.*

Definition 13 (ER, B-narrowing). *We define* $\rightsquigarrow^1_{ER,B}$ *as* $\rightsquigarrow^1_{R(E),B} \cup \rightsquigarrow^1_{E,B}$*.*

In conditional narrowing, given a term t in $T_\Sigma(\mathcal{X})$ and a rule $c : l \to r$ *if* C in R, we start with a unifier $\sigma' \in CSU_B(t|_p = l)$, for appropriate p, and recursively solve the new goal $C\sigma'$, using $\rightsquigarrow^1_{ER,B}$ for reachability conditions and $\rightsquigarrow^1_{E,B}$ for the rest, obtaining some σ'' as solution. Then $\sigma = \sigma'\sigma''$ is the desired substitution such that $t \rightsquigarrow^1_{p,\sigma,ER,B} t[r]_p\sigma$.

Example 11. *Consider* $\mathcal{R} = (\Sigma, E \cup B, R)$*, where* $S = \{s\}$*,* $F = \{\{a, b, c\}_s, \{f, g\}_{ss,s}\}$*,* $\mathcal{E} = \emptyset$*, and* $R = \{g(b, c) \to c, f(a, z_s) \to b \text{ if } g(b, z_s) \to c\}$*.*

Now, if we try to narrow the term $f(x_s, y_s)$ *with rule* $f(a, z_s) \to b$ *if* $g(b, z_s) \to c$ *and unifier* $\sigma' = \{x_s \mapsto a, y_s \mapsto w_s, z_s \mapsto w_s\}$ *we have to prove the condition* $g(b, w_s) \to c$*, which can be narrowed with rule* $g(b, c) \to c$ *and substitution* $\sigma'' = \{w_s \mapsto c\}$*, so* $g(b, z_s) \rightsquigarrow_{\sigma'',ER,B} c$*. Then, by composition of the substitutions* σ' *and* σ''*, we get* $\sigma = \{x_s \mapsto a, y_s \mapsto c, z_s \mapsto c\}$ *and we have* $f(x_s, y_s) \rightsquigarrow_{\sigma,ER,B} b$*. As a consequence, that will be later proved,* $f(x_s, y_s)\sigma \to_{ER,B} b$*, i.e.,* $f(a, c) \to_{ER,B} b$*.*

4.3 Sentence-normalized substitution and rewriting

We develop in this section the concepts of *sentence-normalized substitution* and *sentence-normalized rewriting*.

Let $(\Sigma, E \cup B)$ be an FPP executable MEL theory, and $\mathcal{R}_E = (\Sigma', B, R_E)$ its associated rewrite theory. Executability allows us to incrementally construct the substitutions used on $\to^1_{E,B}$ in such a way that we will only generate normal substitutions for matching variables.

Let $t \in T_\Sigma$ be a term, and $c' : l \to r$ *if* $\bigwedge_{i=1}^n A'_i$ a conditional rule in R_E (we don't have to prove anything for unconditional rules). If l matches t using σ_0 ($t =_B l\sigma_0$) then for all i, $1 \le i \le n$, if A'_i has no matching variables we define $\sigma_i = id$; else if $A'_i = t'_i \to t_i$ has matching variables (because the corresponding sentence c in E has the condition $A_i = t_i := t'_i$), then $(\Sigma, E \cup B)$ being FPP implies that each substitution σ_j, $1 \le j < i$ instantiates different variables, so $\bigcup_{j=0}^{i-1} \sigma_j$ is properly defined, and $dom(\bigcup_{j=0}^{i-1} \sigma_j) \cap vars(t_i) = \emptyset$. Then $(t'_i \bigcup_{j=0}^{i-1} \sigma_j \to t_i \bigcup_{j=0}^{i-1} \sigma_j) = (t'_i \bigcup_{j=0}^{i-1} \sigma_j \to t_i)$. We define σ_i to be a matching of t_i with $(t'_i \bigcup_{j=0}^{i-1} \sigma_j)\downarrow$, that is $t_i\sigma_i =_B (t'_i \bigcup_{j=0}^{i-1} \sigma_j)\downarrow$. As we are matching against an E, B-irreducible term, σ_i must be normal. The only exception is the first substitution σ_0, which may not be normal but doesn't instantiate matching variables.

The *extended* substitution σ that we need to apply the rule is $\sigma = \bigcup_{i=0}^{n} \sigma_i$, where the instantiation of all matching variables, $\bigcup_{i=1}^{n} \sigma_i$, uses normal terms.

Sentence-normalized substitution and sentence-normalized rewriting are based on this fact. We show their connection to E, B-rewriting and ER, B-rewriting for our unification and reachability goals. As we have already shown the link between $=_{\mathcal{E}}$, $:_{\mathcal{E}}$, and E, B-rewriting, and also the link between $R/\mathcal{E}$-rewriting and ER, B-rewriting, all the properties for the narrowing calculi to be presented in the next sections will only have to be related to sentence-normalized rewriting.

Definition 14 (Sentence-normalized substitution)**.** *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$, its FPP executable* Mel *theory $(\Sigma, E \cup B)$, and the associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, for any conditional rule, $c : l \to r$ if C in R_E or R and substitution σ, the* sentence-normalized substitution *σ_c is defined as $\sigma_c = \sigma|_{vars(l)} \cup \sigma{\downarrow}|_{Extra(c)}$, where $Extra(c) = vars(c) \setminus vars(l)$ is the set of new extra variables in c. In the case of rules in R_E, $Extra(c)$ will only contain matching variables.*

We are interested in computing normal solutions for unification and reachability goals using only sentence-normalized substitutions, hence reducing the state space. We define several new relations, where we introduce new notation for easier reading.

Definition 15 (Sentence-normalized rewriting)**.** *We will use the term* sentence-normalized rewriting (SNR from now on) *and write $t \to^1_N t'$ ($t \to_N t'$) instead of $t \to^1_{E,B} t'$ (resp., $t \to_{E,B} t'$), and also write $t \Rightarrow^1_N t'$ ($t \Rightarrow_N t'$) instead of $t \to^1_{R(E),B} t'$ (resp., $t \to_{ER,B} t'$), to imply that only sentence-normalized substitutions have been applied in all rewrite steps.*

Note that $\to^1_N \subseteq \to^1_{E,B}$, $\to_N \subseteq \to_{E,B}$, $\Rightarrow^1_N \subseteq \to^1_{R(E),B}$, and $\Rightarrow_N \subseteq \to_{ER,B}$, so each relation is sound with respect to its superset. Also, by the new definition of $\to^1_{R(E),B}$, if $t \Rightarrow^1_N t'$ then $t \to^!_N t{\downarrow} \Rightarrow^1_N t'$.

The main result is that $\to_N$ is complete with respect to $\to_{E,B}$ when rewriting to normal form.

Lemma 4 (Completeness of sentence-normalized rewriting to normal form)**.** *Given an FPP executable* Mel *theory $(\Sigma, E \cup B)$, its associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, and terms $t, t' \in T_{\Sigma'}(\mathcal{X})$, if $t \to_{E,B} t'$ and t' is E, B-irreducible then $t \to_N t'$.*

See proof on page 85.

As a consequence, it is always the case that $t \to^!_N t{\downarrow}$.

Proposition 1 (Rewriting in $\Rightarrow_N$ with normal forms)**.** *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$, its FPP executable* Mel *theory $(\Sigma, E \cup B)$, the associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, and two terms t, t' in $T_\Sigma(\mathcal{X})$, if $t \Rightarrow_N t'$ then $t{\downarrow} \Rightarrow_N t'$.*

See proof on page 85.

The main result for $\Rightarrow_N$ is that $\Rightarrow_N$ is complete with respect to $\to_{R/\mathcal{E}}$.

Lemma 5 (Completeness of Sentence-normalized Rewriting for $\to_{R/\mathcal{E}}$)**.** *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$, its FPP executable* Mel *theory $(\Sigma, E \cup B)$, the associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, and terms t, t' in $T_\Sigma(\mathcal{X})$, if $t \to_{R/\mathcal{E}} t'$ then $t \Rightarrow_N t'$.*

See proof on page 86.

4.3.1 Results for solutions to unification and reachability goals

As a direct consequence of Lemma 4 we get that with respect to associated unification goals the normal E, B-solutions are the same using $\rightarrow_{E,B}$, or $\rightarrow_N$ (which we call *N-solutions*). Recall that $\sigma_c = \sigma$ for any normal substitution σ and rule c in R_E.

Theorem 6 (Equivalence of SNR for Solutions of Associated Unification Goals)**.** *Given an FPP executable* MEL *theory $(\Sigma, E \cup B)$ and its associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, a normal substitution σ is an $\mathcal{E}$-solution of a system of sentences F and an E, B-solution of its associated unification goal $G = \bigwedge_{i=1}^{n}(t_i \rightarrow t'_i)$ (so $t_i\sigma \rightarrow_{E,B} t'_i\sigma$, for $1 \leq i \leq n$) iff $t_i\sigma \rightarrow_N t'_i\sigma$, $i = 1, \ldots, n$ (i.e. σ is an N-solution for G).*

See proof on page 86.

We also have that conditions and canonical conditions have the same E, B-normalized solutions. This result is important because it will allow us to reduce the state space in the narrowing problems by working only with canonical forms.

Proposition 2. *Given an FPP executable* MEL *theory $(\Sigma, \mathcal{E})$ and its associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, for any conditional* MEL *sentence c in E, and corresponding rule $c' \equiv s'$ if $\bigwedge_{i=1}^{n} u_i \rightarrow v_i$ in $\mathcal{R}_E$, if there is an E, B-normalized substitution σ such that $u_i\sigma \rightarrow_{E,B} v_i\sigma$, for $1 \leq i \leq n$, then $u_i{\downarrow}\sigma \rightarrow_N v_i\sigma$, for $1 \leq i \leq n$.*

See proof on page 86.

Another result is that the normal E, B-solutions of a unification goal are the same as the N-solutions for $G{\downarrow}$.

Lemma 6 (Equivalence of Solutions for Normal Unification Goals)**.** *Given an FPP executable* MEL *theory $(\Sigma, E \cup B)$ and its associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, a normal substitution σ is an E, B-solution of the unification goal $G = \bigwedge_{i=1}^{n}(t_i \rightarrow t'_i)$, associated to a system of sentences F, iff $t_i{\downarrow}\sigma \rightarrow_N t'_i\sigma$, for $1 \leq i \leq n$.*

See proof on page 87.

The last result in this section shows that for reachability goals the normal E, B-solutions are the same using $\rightarrow_{ER,B}$ or $\Rightarrow_N$ (which we again call *N-solutions*).

Lemma 7 (Equivalence of Solutions for Normalized Reachability Goals)**.** *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$, its FPP executable* MEL *theory $(\Sigma, E \cup B)$, the associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, and a reachability goal $G = u_1 \Rightarrow v_1 \wedge \ldots \wedge u_n \Rightarrow v_n \wedge G'$, where G' is a unification goal associated to a system of sentences F, and an idempotent normal substitution σ, these three assertions are equivalent:*

1. *σ is a solution for G,*
2. *σ is an N-solution for $G{\downarrow}$,*
3. *σ is a solution for $G{\downarrow}$.*

See proof on page 87.

4.4 Conditional Narrowing for $\mathcal{E}$-solutions

Narrowing allows us to compute solutions for reachability goals. We implement narrowing using a calculus with the following properties:

1. The calculus is *weakly complete*, i.e., for any idempotent ER, B-normalized solution of a reachability goal G, the calculus can compute a more general answer for G.

2. The calculus is *sound*, i.e., if the calculus computes an answer σ for a reachability goal G, then σ is a solution of G.

We are going to split the calculus into two parts: the one that solves unification goals and the one that solves reachability goals. We assume that we have a complete B-unification algorithm that for any equation $t = t'$ returns $CSU_B(t = t')$ away from all the variables in G, so all the unifiers are idempotent.

4.4.1 Transformation for unification with memberships

As the current existing unification algorithms in Maude are only valid for order-sorted theories, we are going to develop a transformation that allows us to apply these algorithms to our MEL theories at the kind level, and later takes into account membership information provided by the variables in the calculus. As this transformation can impose a lot of extra work to our calculus, because it doesn't make use of order-sorted information for computing B-unifiers, it would be desirable to identify which terms or subterms are not suitable for order-sorted unification and apply the transformation only on those terms or subterms. We show an algorithm that identifies the sorts that cannot be involved in an order-sorted unification. We will apply the transformation only to terms of those sorts.

Non order-sorted unifiable sorts

From S, the set of sorts in our rewrite theory, we define the subset $MB(S)$ of non order-sorted unifiable sorts, see [LM09], as the smallest subset of S such that

1. if $t : s(\ if\ c)$ in E is not a subsort declaration then $s \in MB(S)$.

2. if $s \in MB(S)$ and $s \leq s'$, with s, s' in S then $s' \in MB(S)$.

3. if $f : s_1 \cdots s_n \to s$ is an function declaration, with s in S and $s_i \in MB(S)$ for some i, $1 \leq i \leq n$, then $s \in MB(S)$.

Recall that for simplicity we only allow overloading of functions when their codomains belong to the same kind. We define $OS(S) = S - MB(S)$. $OS(S)$ is the set of sorts whose terms can be unified by order-sorted unification, no memberships can be involved directly or indirectly, via functions, when checking whether a term has any of these sorts or not.

Example 12. *In the concurrency specification example, as* $\mathtt{u1}, \mathtt{u2} : \mathtt{us}$ *is in* E*, then* $\mathtt{us} \in MB(S)$ *using case 1. Also, as* $x_{\mathtt{us}} \mid y_{\mathtt{u}}, y'_{\mathtt{u}} \mid \mathtt{emptyT} \mid \mathtt{emptyT} : s$ *if* $y_{\mathtt{u}}, y'_{\mathtt{u}}, x_{\mathtt{us}} : \mathtt{us}$ *is in* E*, then* $\mathtt{s} \in MB(S)$ *using case 1. Then* $MB(S) = \{\mathtt{us}, \mathtt{s}\}$ *and* $OS(S) = \{\mathtt{u}, \mathtt{t}, \mathtt{tb}, \mathtt{n}, \mathtt{b}\}$.

Kinded function

Given two terms t and t', if $ls(t) \in OS(S)$ and $ls(t') \in OS(S)$ then we unify them directly. Else, we compute the non well-formed substitution $\rho = \{x^i_{s_i} \mapsto y^i_{[s_i]} \mid x^i_{s_i} \in vars(t) \cup vars(t') \wedge s_i \in S\}$, with each $y^i_{[s_i]}$ a fresh unsorted variable, and unify $s\rho$ and $t\rho$, terms that only have unsorted variables. From $\rho = \{x^i_{s_i} \mapsto y^i_{[s_i]}\}^n_{i=1}$ we generate the system of sentences $C = \bigwedge_{i=1}^n y^i_{[s_i]} : s_i$.

C and ρ are computed by $kinded(t, t')$ defined below in a Maude-like algorithm. V is the auxiliary set of already processed variables, function k processes lists of terms, function $k1$ processes individual terms, and function $k0$ discards V and returns the pair (C, ρ):

$kinded(t, t') = (\emptyset, id)$ if $ls(t) \in OS(S)$ and $ls(t') \in OS(S)$
$kinded(t, t') = k0(k((t, t'), (\emptyset, id, \emptyset)))$ otherwise

$k((t_1, \ldots, t_n), (C, \rho, V)) = k((t_2, \ldots, t_n), k1(t_1, (C, \rho, V)))$
$k((t), (C, \rho, V)) = k1(t, (C, \rho, V))$

$k1(f(t_1, \ldots, t_n), (C, \rho, V)) = k((t_1, \ldots, t_n), (C, \rho, V))$
$k1(c, (C, \rho, V)) = (C, \rho, V)$ where c constant.
$k1(x^i_{\kappa_i}, (C, \rho, V)) = (C, \rho, V)$ if $x^i_{\kappa_i} \in V$ or κ is a kind.
$k1(x^i_{\kappa_i}, (C, \rho, V)) = (C \wedge y^i_{[s_i]} : s_i, \rho \cup \{x^i_{\kappa_i} \mapsto y^i_{[s_i]}\}, V \cup \{x^i_{\kappa_i}\})$ otherwise, with $y^i_{[s_i]}$ a fresh variable.

$k0((C, \rho, V)) = (C, \rho)$

The computed substitution ρ replaces the variables that belong to sorts that cannot be unified with order-sorted algorithms with variables of the corresponding kind. The computed condition C ensures that the new kinded variables are instantiated to terms with the same sort as the original variables.

Lemma 8. *Given an FPP executable* MEL *theory* $(\Sigma, E \cup B)$ *and its associated rewrite theory* $\mathcal{R}_E = (\Sigma', B, R_E)$*, a substitution* σ*, with* $dom(\sigma) = vars(t) \cup vars(t')$ *(i.e., all variables are at least renamed), is an idempotent* B*-unifier of two terms* t, t' *in* $T_\Sigma(\mathcal{X})$ *if and only if* $\sigma =_B \rho\gamma\gamma'$*, where* $kinded(t, t') = (C, \rho)$*,* γ *is an idempotent* B*-unifier of* $t\rho$ *and* $t'\rho$*, and* γ' *is an* $\mathcal{E}$*-solution for the system of sentences* $C\gamma$*, where all substitutions are always away from all the variables that have previously appeared.*

See proof on page 88.

Example 13. *In the concurrency specification example, let* $t = \mathtt{u2}, x_{\mathtt{u}}$ *and let* $t' = y_{\mathtt{us}}$*. Then* $\sigma = \{x_{\mathtt{u}} \mapsto \mathtt{u1}, y_{\mathtt{us}} \mapsto \mathtt{u1}, \mathtt{u2}\}$ *is a* B*-unifier of* t *and* t' *(because* $\mathtt{u1}, \mathtt{u2} =_B \mathtt{u2}, \mathtt{u1}$*). As* $ls(t) = [\mathtt{us}]$ *and* $ls(t') = \mathtt{us}$*, neither of them belonging to* $OS(S)$*, we don't have direct* B*-unification algorithms for* t *and* t'*, so we compute* $kinded(t, t')$*, with answer* $\rho = \{x_{\mathtt{u}} \mapsto z_{[\mathtt{us}]}, y_{\mathtt{us}} \mapsto v_{[\mathtt{us}]}\}$*, and* $C = z_{[\mathtt{us}]} : \mathtt{us} \wedge v_{[\mathtt{us}]} : \mathtt{us}$*. Now* $t\rho = \mathtt{u2}, z_{[\mathtt{us}]}$ *and* $t'\rho = v_{[\mathtt{us}]}$*. There is a complete many-sorted* B*-unification algorithm at the kind level for* $t\rho$ *and* $t'\rho$ *that returns the answer* $\gamma = \{z_{[\mathtt{us}]} \mapsto w_{[\mathtt{us}]}, v_{[\mathtt{us}]} \mapsto \mathtt{u2}, w_{[\mathtt{us}]}\}$*. As* $C\gamma = w_{[\mathtt{us}]} : \mathtt{us} \wedge \mathtt{u2}, w_{[\mathtt{us}]} : \mathtt{us}$*, then* $\gamma' = \{w_{[\mathtt{us}]} \mapsto \mathtt{u1}\}$ *is an* $\mathcal{E}$*-solution for the system of sentences* $C\gamma$*, and* $\{x_{\mathtt{u}} \mapsto \mathtt{u1}, y_{\mathtt{us}} \mapsto \mathtt{u1}, \mathtt{u2}\} = \sigma =_B \rho\gamma\gamma' = \{x_{\mathtt{u}} \mapsto \mathtt{u1}, y_{\mathtt{us}} \mapsto \mathtt{u2}, \mathtt{u1}\}$*, thanks to the commutativity axiom for the function* $\_\,,\_$*.*

4.4.2 Calculus for unification strategies and rules

The calculus for unification uses the following strategies:

- Inference rules are applied with leftmost strategy.
- As we are computing E, B-normalized solutions and we have already shown the equivalence of SNR-rewriting with respect to E, B-normalized solutions for unification goals, we have built-in the following strategy in our calculus: we only apply a calculus rule if the composition of all computed substitutions remains idempotent E, B-normalized with respect to all extra variables and all the variables in the initial unification problem. This means that we must keep track of all extra variables that have not been instantiated to ground terms in order to be able to discard any narrowing step that violates this principle.
- As we have also proved the equivalence of E, B-normalized solutions with respect to normalized unification goals, we follow a second strategy in our calculus consisting of normalizing the unification problem after each use of rules reduction or elimination.

It is later proved that we don't miss any answer with these reductions of the state space. Our calculus for $\mathcal{E}$-solutions is defined by the inference rules in Figure 4.4. These rules transform *unification problems* of the form $t \to t'$, or $t|_p \to^1 x_k, t[x_k]_p \to t'$ (x_k fresh variable, with $k = [ls(t|_p)]$), both having the same meaning: find a substitution σ such that $t\sigma \to_{E,B} t'\sigma$. Note that in the second type of subproblem, t can be easily reconstructed as $t = t[x_k]_p\rho$, with $\rho = \{x_k \mapsto t|_p\}$. The goal G' represents the rest of unification subproblems that have not been processed yet, if they exist. We show G' in an inference rule only when it can be affected by instantiation and further normalization.

Note that unification goals are a subset of unification problems. For any subproblem of the form $t \to t'$, if $t =_B t'$ then we always apply rule elimination with substitution id, which is more general than any other possible computed answer. After applying rule transitivity we get a unification problem $t|_p \to^1 x_k, t[x_k]_p \to t'$, where we perform an actual narrowing step in t using rule reduction, or we perform an actual narrowing step in some proper subterm of t, applying several times rule congruence to reach the desired subterm, followed by an application of rule reduction. Such narrowing steps have a kinded variable x_k as target, because although t may have some sort s when it is sufficiently instantiated with some substitution σ, t will have kind $k = [s]$ for partial instantiations, but it will usually have no sort. Rule membership is needed to lower the type of a variable so it has a desired sort. It is the most general way of instantiating the variable.

Our transformation of the rules in R_E and R generates additional membership subgoals. Many of them are trivial and don't need any further instantiation after, or become trivial after several calculus steps are applied to other subgoals. By normalization, these trivial membership subgoals $t{:}s \to \texttt{tt}$, with t ground usually, become $\texttt{tt} \to \texttt{tt}$ and they will be removed from the problem with the elimination rule and substitution id. If subgoals were not normalized, there would be a significant overhead in the calculus only in order to prove these trivial subgoals, not to mention all the overhead generated by applying narrowing to all the subgoals that may exist in a rewriting path between a subgoal g and its normalized version $g{\downarrow}$.

When we apply one of the calculus rules for unification to a unification problem G_i with some inference rule $[i]$, substitution σ_i, and (maybe) rule c from R_E, yielding another unification problem G_{i+1}, we display it as $G_i \rightsquigarrow_{[i](,c),\sigma_i} G_{i+1}$ and say that there exists a

- $[t]$ transitivity

$$\frac{t \to t' \ (\wedge G')}{t \to^1 x_k, x_k \to t' \ (\wedge G')}$$

where $k = [ls(t)]$

- $[c]$ congruence

$$\frac{t|_p \to^1 x_k, t[x_k]_p \to t' \ (\wedge G')}{t|_{p.i} \to^1 y_{k'}, t[y_{k'}]_{p.i} \to t' \ (\wedge G')}$$

where $t|_p = f(t_1, \ldots, t_n)$, $1 \leq i \leq n$, $k' = [ls(t|_{p.i})]$, and $y_{k'}$ is a fresh variable

- $[r]$ reduction

$$\frac{t|_p \to^1 x_k, t[x_k]_p \to t' \ (\wedge G')}{(((C) \wedge t[r]_p \to t'(\wedge G'))\theta)\downarrow}$$

where $t|_p \notin \mathcal{X}$, $l \to r \ (if \ C)$ is a fresh instance of a rule in R_E,
$k = [ls(t)]$, and $\theta \in CSU_B(t|_p = l)$

- $[e]$ elimination

$$\frac{t \to t' \ (\wedge G')}{(G'\theta)\downarrow}$$

where $\theta \in CSU_B(t = t')$

Figure 4.4: Inference rules for $\mathcal{E}$-solution by conditional narrowing.

narrowing step from G_i to G_{i+1} using the substitution σ_i, inference rule $[i]$, and (maybe) rule c. $[i]$, c, and σ_i may be omitted when their actual values are irrelevant or can be inferred.

Proposition 3. *Given an FPP executable* MEL *theory* $(\Sigma, E \cup B)$ *and its associated rewrite theory* $\mathcal{R}_E = (\Sigma', B, R_E)$*, after applying the* transitivity *rule followed by zero or more applications of the* congruence *rule to a unification problem of the form* $t \to t'$ *we get another unification problem of the form* $t|_p \to^1 x_k, t[x_k]_p \to t'$ *with* k *some kind in* Σ'*.*

See proof on page 88.

This proposition means that after applying the transitivity rule $[t]$ to a unification subgoal and before applying the reduction rule $[r]$, all generated unification subproblems that use the $\to^1$ symbol will have the same shape: $t|_p \to^1 x_k, t[x_k]_p \to t'$, for some $p \in Pos(t)$, with $k = [ls(t|_p)]$, and x_k fresh variable. As we always start our inferences from a unification goal, we can assume that a unification subproblem has this shape when there is a $\to^1$ symbol within the subproblem.

Again, from a unification goal G a derivation is made applying rules of the calculus. If the derivation ends in the empty goal, denoted by $\square$ and written $G \rightsquigarrow_{\sigma_1} G_1 \ldots \rightsquigarrow_{\sigma_n} \square$, or $G \rightsquigarrow^*_{\sigma} \square$, with $\sigma = \sigma_1 \ldots \sigma_n$, then $\sigma_{vars(G)}$ is a computed answer for G.

Definition 16 (normalized goal)**.** *Given a narrowable rewrite theory* $\mathcal{R} = (\Sigma, \mathcal{E}, R)$*, its associated rewrite theory* $\mathcal{R}_E = (\Sigma', B, R_E)$ *and a condition in* R_E *(or a unification goal)* $G = \bigwedge_{i=1}^n t_i \to t'_i$*, we define the* normalized condition (or goal) $G{\downarrow}$ *as* $\bigwedge_{i=1}^n t_i{\downarrow} \to t'_i{\downarrow}$*.*

Recall that for a unification goal associated to a system of sentences or a condition in R_E, $G{\downarrow} = \bigwedge_{i=1}^n t_i{\downarrow} \to t'_i$ because t'_i is always `tt` or a Σ-pattern.

The calculus for unification is sound and weakly complete, i.e., complete with respect to normalized goals and idempotent ER, B-normalized solutions. We prove the completeness of the calculus with respect to normalized goals (by Lemma 6) and E, B-normalized idempotent solutions (more general than ER, B-normalized solutions). In this way, we can independently apply this part of the calculus to any FPP executable MEL theory, even if it is the case that the MEL theory is not underlying some rewrite theory.

Theorem 7 (Soundness of the Calculus for $\mathcal{E}$-solutions)**.** *Given a narrowable rewrite theory* $\mathcal{R} = (\Sigma, \mathcal{E}, R)$*, its FPP executable* MEL *theory* $(\Sigma, E \cup B)$*, the associated rewrite theory* $\mathcal{R}_E = (\Sigma', B, R_E)$*, a system of sentences, and its associated unification goal* G*, if* σ *is a computed answer for* $G{\downarrow}$ *then* σ *is an idempotent* E, B*-normalized* $\mathcal{E}$*-solution for* G*.*

See proof on page 89.

Theorem 8 (Weak Completeness of the Calculus for $\mathcal{E}$-solutions)**.** *Given a narrowable rewrite theory* $\mathcal{R} = (\Sigma, \mathcal{E}, R)$*, its FPP executable* MEL *theory* $(\Sigma, E \cup B)$*, the associated rewrite theory* $\mathcal{R}_E = (\Sigma', B, R_E)$*, a system of sentences* F*, and its associated unification goal* G*, if* σ *is an idempotent* E, B*-normalized* $\mathcal{E}$*-solution for* G *then there is an idempotent* E, B*-normalized substitution* γ*, with* $\sigma \ll_B \gamma_{vars(G)}$*, such that* $G{\downarrow} \rightsquigarrow^*_{\gamma} \square$*.*

See proof on page 90.

Example 14. *In example 13 we had* $C\gamma = w_{[\mathtt{us}]} : \mathtt{us} \wedge \mathtt{u2}, w_{[\mathtt{us}]} : \mathtt{us}$*, and we used* $\gamma' = \{w_{[\mathtt{us}]} \mapsto \mathtt{u1}\}$ *as an* $\mathcal{E}$*-solution for the system of sentences* $C\gamma$*. The associated*

unification goal G *for* $C\gamma$ *is* $w_{[\mathtt{us}]} : \mathtt{us} \to \mathtt{tt} \wedge \mathtt{u2}, w_{[\mathtt{us}]} : \mathtt{us} \to \mathtt{tt}$, *and* $G{\downarrow} = G$. *The narrowing derivation that finds* γ', *where we write* T *as a shortcut for Truth, uses the rule* $E1 : u_1, u_2{:}\mathtt{us} \to \mathtt{tt}$ *in* R_E *that comes from the membership* $u_1, u_2{:}\mathtt{us}$ *in* $\mathcal{E}$*:*
$G{\downarrow} \rightsquigarrow_{[m],\{w_{[\mathtt{us}]} \mapsto y_{\mathtt{us}}\}} \mathtt{u2}, y_{\mathtt{us}} : \mathtt{us} \to \mathtt{tt} \rightsquigarrow_{[t]} \mathtt{u2}, y_{\mathtt{us}} : \mathtt{us} \to^1 x_{[T]}, x_{[T]} \to \mathtt{tt} \rightsquigarrow_{[r],E1,\{y_{\mathtt{us}} \mapsto u_1, x_{[T]} \mapsto \mathtt{tt}\}} \mathtt{tt} \to \mathtt{tt} \rightsquigarrow_{[e]} \square$.

The inference rule $[r]$ *can be applied thanks to the commutativity axiom for the function* $\_,\_$ *that yields* $u_1, u_2 : \mathtt{us} =_B u_2, u_1 : \mathtt{us}$. *As* $vars(G) = \{w_{[\mathtt{us}]}\}$, *and* $\sigma_1 = \{w_{[\mathtt{us}]} \mapsto y_{\mathtt{us}}\}$ *and* $\sigma_2 = \{y_{\mathtt{us}} \mapsto u_1, x_{[T]} \mapsto \mathtt{tt}\}$ *are the substitutions in the narrowing derivation, then* $(\sigma_1\sigma_2)_{vars(G)} = \gamma'$.

4.5 Reachability by conditional narrowing

In this part of the calculus, given an FPP narrowable rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ and a reachability goal G, we will solve the normalized reachability goal $G{\downarrow}$ and prove that it has the same E, B-normalized solutions as G. We will use a transformed set of rules $\tilde{R}$ where for each rule $l \to r$ (*if* $\bigwedge_{i=1}^{n} A_i$) in R, there is a rule $l \to r$ (*if* $\bigwedge_{i=1}^{n} A'_i$) in $\tilde{R}$ such that:

- if A_i has the form $t_i \to t'_i$ then A'_i is $t_i \Rightarrow t'_i$,
- if A_i has the form $t_i : s_i$ then A'_i is $t_i{:}s_i \to \mathtt{tt}$,
- if A_i has the form $t_i := t'_i$ then A'_i is $t'_i \to t_i$, and
- if A_i has the form $t_i = t'_i$ then A'_i is $eq(t_i, t'_i) \to \mathtt{tt}$.

That is, we apply the same transformation that we used in the rewrite theory associated to a Mel theory, and replace each $\to$ symbol in conditions with a new $\Rightarrow$ symbol, so we can distinguish reachability conditions from equational conditions.

4.5.1 Calculus for reachability strategies and rules

Reachability by conditional narrowing is achieved using the previous calculus rules in Figure 4.4, extended with the calculus rules in Figure 4.5. These new rules transform *reachability problems* that have the form $t \Rightarrow t'$, $t|_p \to^1 x_k, t[x_k]_p \Rightarrow t'$, or $t|_p \Rightarrow^1 x_k, t[x_k]_p \Rightarrow t'$ (x_k being a fresh variable, where $k = [ls(t|_p)]$), all of them having the same meaning: find a substitution σ such that $t\sigma \to_{ER,B} t'\sigma$. Like unification goals, reachability goals are a subset of reachability problems. We show the rest of the reachability goal, G', in an inference rule only when it can be affected by instantiation and further normalization.

The calculus for reachability uses the following strategies:

- Inference rules are applied with a leftmost strategy.
- As we are computing E, B-normalized solutions and SNR-rewriting is equivalent to normal rewriting with respect to E, B-normalized solutions for reachability goals, we only apply a calculus rule if the composition of all computed substitutions remains idempotent E, B-normalized with respect to all extra variables and all the variables in the initial reachability problem, that is, when we apply a rule $l \to r$ (*if* C) in $\tilde{R}$ the only variables that need not be instantiated with an idempotent E, B-normalized substitution are those in $vars(l)$.

- $[x]$ reflexivity

$$\frac{t \Rightarrow t' \ (\wedge G')}{eq(t,t')\downarrow \rightarrow \mathtt{tt} \ (\wedge G')}$$

- $[t]$ transitivity

$$\frac{t \Rightarrow t' \ (\wedge G')}{t \rightarrow^1 x_k, x_k \Rightarrow t' \ (\wedge G')} \qquad \frac{t \Rightarrow t' \ (\wedge G')}{t \Rightarrow^1 x_k, x_k \Rightarrow t' \ (\wedge G')}$$

where $t \notin \mathcal{X}$, and $k = [ls(t)]$

- $[c]$ congruence

$$\frac{t|_p \rightarrow^1 x_k, t[x_k]_p \Rightarrow t' \ (\wedge G')}{t|_{p.i} \rightarrow^1 y_{k'}, t[y_{k'}]_{p.i} \Rightarrow t' \ (\wedge G')} \qquad \frac{t|_p \Rightarrow^1 x_k, t[x_k]_p \Rightarrow t' \ (\wedge G')}{t|_{p.i} \Rightarrow^1 y_{k'}, t[y_{k'}]_{p.i} \Rightarrow t' \ (\wedge G')}$$

with $t|_p \equiv f(t_1, \ldots, t_n)$, $1 \leq i \leq n$, $t_i \notin \mathcal{X}$, $k' = [ls(t|_{p.i})]$, and $y_{k'}$ fresh variable

- $[r]$ reduction

$$\frac{t|_p \rightarrow^1 x_k, t[x_k]_p \Rightarrow t' \ (\wedge G')}{(((C) \wedge t[r]_p \Rightarrow t'(\wedge G'))\theta)\downarrow}$$

where $t|_p \notin \mathcal{X}$, $l \rightarrow r \ (\mathit{if}\ C)$ is a fresh rule in R_E and $\theta \in CSU_B(t|_p = l)$

- $[w]$ rewrite

$$\frac{t|_p \Rightarrow^1 x_k, t[x_k]_p \Rightarrow t' \ (\wedge G')}{(((C) \wedge t[r]_p \Rightarrow t' \ (\wedge G'))\theta)\downarrow}$$

where $t|_p \notin \mathcal{X}$, $l \rightarrow r \ (\mathit{if}\ C)$ is a fresh rule in $\tilde{R}$, $\theta \in CSU_B(t|_p = l)$, and $t\theta$ E,B-normalized

Figure 4.5: Inference rules for reachability by conditional narrowing.

- As there is also an equivalence of solutions and N-solutions with respect to normalized reachability goals, we follow a second strategy in our calculus consisting of normalizing the reachability problem after each use of a conditional rule in the calculus.

- Rule rewrite is applied on $t|_p$ with substitution θ only if **the whole term** $t\theta$ is E,B-normalized.

We explain the meaning of these rules, recall that $\rightarrow_{ER,B}$ is $\rightarrow^*_{ER,B}; =_{\mathcal{E}}$:

- The reflexivity rule applies the $=_{\mathcal{E}}$ part of the definition for $\rightarrow_{ER,B}$. It is the only rule having a $\Rightarrow$ symbol as an antecedent but not as a consequent, so it must be applied in every derivation from a subproblem of the form $t_i \Rightarrow t'_i$ to get rid of the $\Rightarrow$ symbol. If a solution σ generates a derivation with zero rewrite steps in $\rightarrow_{ER,B}$, this means that $t_i\sigma =_{\mathcal{E}} t'_i\sigma$, so we can find this substitution or a more general one by applying the reflexivity rule. The resulting subproblem $eq(t_i, t'_i)$ will be solved using the calculus rules for unification.

- The transitivity rule has been expanded. Now it can also apply the $\rightarrow^1_{R(E),B}$ part of the definition for $\rightarrow^1_{ER,B}$, invoking the use of the congruence and rewrite rules to generate one actual reachability step ($\Rightarrow^1$) for reachability subgoals $t \Rightarrow t'$.

- The congruence rule has been expanded to deal with $\rightarrow^1$ followed by $\Rightarrow$, and $\Rightarrow^1$ followed by $\Rightarrow$. It has the same meaning as in the calculus for unification.

- The reduction rule has been expanded to deal with $\rightarrow^1$ followed by $\Rightarrow$. It also has the same meaning as in the calculus for unification.

- The rewrite rule is the only one that may generate instantiations, by using some rule $\tilde{r}$ from $\tilde{R}$. It gets rid of the $\Rightarrow^1$ symbol generated by the transitivity rule, and propagated by the congruence rule, transforming equational and membership conditions in rule r from R into their equivalent unification conditions in $\mathcal{R}_E$. After the congruence rule has selected a subterm $t|_p$, we apply rewrite using rule $\tilde{r}$ with some B-unifier θ, *only if* the whole instantiated term $t\theta$ is E, B-normalized. This is an improvement over previous reachability calculi for narrowing that only required the instantiated subterm $t|_p\theta$ to be E, B-normalized.

The calculus for reachability is sound and weakly complete.

Theorem 9 (Soundness of the Calculus for Reachability)**.** *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$, its FPP executable* MEL *theory $(\Sigma, E \cup B)$, the associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, and a reachability goal G, if σ is a computed answer for $G{\downarrow}$, using the transformed set of rules $\tilde{R}$, then σ is a solution for G.*

See proof on page 92.

Theorem 10 (Weak Completeness of the Calculus for Reachability)**.** *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$, its FPP executable* MEL *theory $(\Sigma, E \cup B)$, the associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, and a reachability goal G, if σ is an idempotent ER, B-normalized solution for G then there is an idempotent E, B-normalized substitution γ, with $\sigma \ll_B \gamma_{vars(G)}$, such that $G{\downarrow} \rightsquigarrow^*_\gamma \Box$ using the transformed set of rules $\tilde{R}$.*

See proof on page 93.

4.6 Narrowing example: concurrency specification

This is an application of the calculus, using the concurrency specification example. Consider the reachability goal $G = \texttt{init} \Rightarrow x_{\texttt{u}} \mid y_{\texttt{us}} \mid z^1_{\texttt{t}}; z^2_{\texttt{t}} \mid z^1_{\texttt{t}}; z^2_{\texttt{t}}$, where from the initial `State init`, we want to reach a `[State]` with one waiting `User`, two `Tools` in the `ToolBox`, and the same two `Tools` in the workbench. The reachability goal is already normalized. An excerpt of the Maude module with the most relevant membership, equations, and rules follows:

```
mod CONCURRENCY is
...
cmb US | emptyU | TB | TB' : State
    if count(TB ; TB') < s(s(s(s(s(0))))) = ok [label M1] .
```

```
...
eq count(emptyT) = 0 [label E1] .
eq count(T ; TB) = s(count(TB)) [label E2] .
...
crl init => US | emptyU | TB | emptyT
    if US | emptyU | TB | emptyT : State [label R1 nonexec] .
crl US | US' | T ; T' ; TB | emptyT => US | US' | TB | T ; T'
    if US | US' | TB | T ; T' : State [label R2] .
...
endm
```

We abbreviate `emptyT` to ϵ_t, `emptyU` to ϵ_u, and `count` to `c`. We will also write F instead of $x_{\mathtt{u}} \mid y_{\mathtt{us}} \mid z^1_{\mathtt{t}}; z^2_{\mathtt{t}} \mid z^1_{\mathtt{t}}; z^2_{\mathtt{t}}$:

1. $\mathtt{init} \Rightarrow F \rightsquigarrow_{[t]}$

   Rule transitivity is always needed before an application of rule rewrite.

2. $\mathtt{init} \Rightarrow^1 x^1_{[\mathtt{s}]}, x^1_{[\mathtt{s}]} \Rightarrow F \rightsquigarrow_{[w],\mathtt{R1}}$

   Rule rewrite is applied with the transformed rule for `R1`, where the membership condition has now form $x^2_{\mathtt{us}} \mid \epsilon_u \mid x^3_{\mathtt{tb}} \mid \epsilon_t : s \to \mathtt{tt}$. We apply the transformation for unification with memberships and compute $\rho_0 = id$ and $C_0 = \emptyset$ because `init` has no variables.

3. $x^2_{\mathtt{us}} \mid \epsilon_u \mid x^3_{\mathtt{tb}} \mid \epsilon_t : s \to \mathtt{tt} \wedge x^2_{\mathtt{us}} \mid \epsilon_u \mid x^3_{\mathtt{tb}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[t]} \rightsquigarrow_{[r],\mathtt{M1},\sigma_1}$

   We apply rule transitivity again, followed by rule reduction with the fresh rule $x^4_{\mathtt{us}} \mid \epsilon_u \mid x^5_{\mathtt{tb}} \mid x^6_{\mathtt{tb}} : s \to \mathtt{tt}$ *if* $eq(\mathtt{c}(x^5_{\mathtt{tb}}; x^6_{\mathtt{tb}}) < \mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(0))))), \mathtt{ok}) \to \mathtt{tt}$ associated to the conditional membership `M1`. We omit the result of the transitivity step. We compute $\rho_1 = \{x^2_{\mathtt{us}} \mapsto x^2_{[\mathtt{us}]}, x^3_{\mathtt{tb}} \mapsto x^3_{[\mathtt{tb}]}, x^4_{\mathtt{us}} \mapsto x^4_{[\mathtt{us}]}, x^5_{\mathtt{tb}} \mapsto x^5_{[\mathtt{tb}]}, x^6_{\mathtt{tb}} \mapsto x^6_{[\mathtt{tb}]}\}$ and $C_1 = \{x^2_{[\mathtt{us}]} : \mathtt{us} \wedge x^3_{[\mathtt{tb}]} : \mathtt{tb} \wedge x^4_{[\mathtt{us}]} : \mathtt{us} \wedge x^5_{[\mathtt{tb}]} : \mathtt{tb} \wedge x^6_{[\mathtt{tb}]} : \mathtt{tb}\}$. Instead of solving the unification problem and use the obtained unifier, we apply ρ_1 to the whole reachability problem and add the condition associated to C_1 in $\mathcal{R}_E$ in front of the reachability problem, which is an equivalent approach for leftmost narrowing, because in this way we must solve the unification problem before we can continue with the reachability problem. The obtained unifier is $\sigma_1 = \{x^2_{[\mathtt{us}]} \mapsto x^7_{[\mathtt{us}]}, x^3_{[\mathtt{tb}]} \mapsto x^8_{[\mathtt{tb}]}, x^4_{[\mathtt{us}]} \mapsto x^7_{[\mathtt{us}]}, x^5_{[\mathtt{tb}]} \mapsto x^8_{[\mathtt{tb}]}, x^6_{[\mathtt{tb}]} \mapsto \epsilon_t\}$. The condition $x^6_{[\mathtt{tb}]} : \mathtt{tb} \to \mathtt{tt}$ becomes $\epsilon_t : \mathtt{tb} \to \mathtt{tt}$ which after normalization is $\mathtt{tt} \to \mathtt{tt}$. Also $\mathtt{c}(x^5_{\mathtt{tb}}; x^6_{\mathtt{tb}})$ becomes $\mathtt{c}(x^8_{[\mathtt{tb}]}; \epsilon_t)$, and then it becomes $\mathtt{c}(x^8_{[\mathtt{tb}]})$ after normalization.

4. $x^7_{[\mathtt{us}]} : \mathtt{us} \to \mathtt{tt} \wedge x^8_{[\mathtt{tb}]} : \mathtt{tb} \to \mathtt{tt} \wedge x^7_{[\mathtt{us}]} : \mathtt{us} \to \mathtt{tt} \wedge x^8_{[\mathtt{tb}]} : \mathtt{tb} \to \mathtt{tt} \wedge \mathtt{tt} \to \mathtt{tt} \wedge$
   $eq(\mathtt{c}(x^8_{[\mathtt{tb}]}) < \mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(0))))), \mathtt{ok}) \to \mathtt{tt} \wedge x^7_{[\mathtt{us}]} \mid \epsilon_u \mid x^8_{[\mathtt{tb}]} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[t]} \rightsquigarrow_{[r],\{x^7_{[\mathtt{us}]} \mapsto x^7_{\mathtt{us}}\}}$

5. $\mathtt{tt} \to \mathtt{tt} \wedge x^8_{[\mathtt{tb}]} : \mathtt{tb} \to \mathtt{tt} \wedge x^7_{[\mathtt{us}]} : \mathtt{us} \to \mathtt{tt} \wedge x^8_{[\mathtt{tb}]} : \mathtt{tb} \to \mathtt{tt} \wedge \mathtt{tt} \to \mathtt{tt} \wedge$
   $eq(\mathtt{c}(x^8_{[\mathtt{tb}]}) < \mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(0))))), \mathtt{ok}) \to \mathtt{tt} \wedge x^7_{[\mathtt{us}]} \mid \epsilon_u \mid x^8_{[\mathtt{tb}]} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[e]}$

6. $x^8_{[\mathtt{tb}]} : \mathtt{tb} \to \mathtt{tt} \wedge \mathtt{tt} \to \mathtt{tt} \wedge x^8_{[\mathtt{tb}]} : \mathtt{tb} \to \mathtt{tt} \wedge \mathtt{tt} \to \mathtt{tt} \wedge$
$eq(\mathtt{c}(x^8_{[\mathtt{tb}]}) < \mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{0}))))), \mathtt{ok}) \to \mathtt{tt} \wedge x^7_{\mathtt{us}} \mid \epsilon_u \mid x^8_{[\mathtt{tb}]} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[t]} \rightsquigarrow_{[r],\{x^8_{[\mathtt{tb}]} \mapsto x^8_{\mathtt{tb}}\}}$

7. $\mathtt{tt} \to \mathtt{tt} \wedge \mathtt{tt} \to \mathtt{tt} \wedge \mathtt{tt} \to \mathtt{tt} \wedge \mathtt{tt} \to \mathtt{tt} \wedge eq(\mathtt{c}(x^8_{\mathtt{tb}}) < \mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{0}))))), \mathtt{ok}) \to \mathtt{tt} \wedge$
$x^7_{\mathtt{us}} \mid \epsilon_u \mid x^8_{\mathtt{tb}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[e]} \rightsquigarrow_{[e]} \rightsquigarrow_{[e]} \rightsquigarrow_{[e]}$

   Rule elimination removes the trivial subgoals. We have finished solving the unification problem, with unifier $\{x^2_{\mathtt{us}} \mapsto x^7_{\mathtt{us}}, x^3_{\mathtt{tb}} \mapsto x^8_{\mathtt{tb}}, x^4_{\mathtt{us}} \mapsto x^7_{\mathtt{us}}, x^5_{\mathtt{tb}} \mapsto x^8_{\mathtt{tb}}, x^6_{\mathtt{tb}} \mapsto \epsilon_t\}$, so we continue with the reachability problem. We apply rules transitivity, congruence, and reduction several times using the rules associated to equations E1 and E2. As the involved sorts are in $OS(S)$ we can use the complete order-sorted unification algorithms without any transformation.

8. $eq(\mathtt{c}(x^8_{\mathtt{tb}}) < \mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{0}))))), \mathtt{ok}) \to \mathtt{tt} \wedge x^7_{\mathtt{us}} \mid \epsilon_u \mid x^8_{\mathtt{tb}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[t]}$

9. $eq(\mathtt{c}(x^8_{\mathtt{tb}}) < \mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{0}))))), \mathtt{ok}) \to^1 x^9_{[T]}, x^9_{[T]} \to \mathtt{tt} \wedge x^7_{\mathtt{us}} \mid \epsilon_u \mid x^8_{\mathtt{tb}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[c]}$

10. $\mathtt{c}(x^8_{\mathtt{tb}}) < \mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{0}))))) \to^1 x^{10}_{[b]}, eq(x^{10}_{[b]}, \mathtt{ok}) \to \mathtt{tt} \wedge x^7_{\mathtt{us}} \mid \epsilon_u \mid x^8_{\mathtt{tb}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[c]}$

11. $\mathtt{c}(x^8_{\mathtt{tb}}) \to^1 x^{11}_{[\mathtt{n}]}, eq(x^{11}_{[\mathtt{n}]} < \mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{0}))))), \mathtt{ok}) \to \mathtt{tt} \wedge x^7_{\mathtt{us}} \mid \epsilon_u \mid x^8_{\mathtt{tb}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[r],\mathtt{E2},\sigma_2}$

    Rule reduction is applied with equation E2 and unifier $\sigma_2 = \{x^8_{\mathtt{tb}} \mapsto x^{12}_{\mathtt{t}}; x^{13}_{\mathtt{tb}}\}$.

12. $eq(\mathtt{s}(\mathtt{c}(x^{13}_{\mathtt{tb}})) < \mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{0}))))), \mathtt{ok}) \to \mathtt{tt} \wedge x^7_{\mathtt{us}} \mid \epsilon_u \mid x^{12}_{\mathtt{t}}; x^{13}_{\mathtt{tb}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[t]}$

    We omit some steps here.

13. $eq(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}((\mathtt{c}(x^{17}_{\mathtt{tb}})))))) < \mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{0}))))), \mathtt{ok}) \to \mathtt{tt} \wedge$
$x^7_{\mathtt{us}} \mid \epsilon_u \mid x^{12}_{\mathtt{t}}; x^{14}_{\mathtt{t}}; x^{15}_{\mathtt{t}}; x^{16}_{\mathtt{t}}; x^{17}_{\mathtt{tb}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[t]}$

    We also omit some steps here.

14. $\mathtt{c}(x^{18}_{\mathtt{tb}}) \to^1 x^{19}_{[\mathtt{n}]}, eq_{[\mathtt{b}]}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}((x^{19}_{[\mathtt{n}]}))))) < \mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{0}))))), \mathtt{ok}) \to \mathtt{tt} \wedge$
$x^7_{\mathtt{us}} \mid \epsilon_u \mid x^{12}_{\mathtt{t}}; x^{14}_{\mathtt{t}}; x^{15}_{\mathtt{t}}; x^{16}_{\mathtt{t}}; x^{18}_{\mathtt{tb}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[r],\mathtt{E1},\sigma_3}$

    Now $\sigma_3 = \{x^{18}_{\mathtt{tb}} \mapsto \epsilon_t\}$, so $x^{19}_{[\mathtt{n}]} \mapsto 0$. Observe the great simplification of the reachability problem after normalization.

15. $\mathtt{tt} \to \mathtt{tt} \wedge x^7_{\mathtt{us}} \mid \epsilon_u \mid x^{12}_{\mathtt{t}}; x^{14}_{\mathtt{t}}; x^{15}_{\mathtt{t}}; x^{16}_{\mathtt{t}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[e]}$

16. $x^7_{\mathtt{us}} \mid \epsilon_u \mid x^{12}_{\mathtt{t}}; x^{14}_{\mathtt{t}}; x^{15}_{\mathtt{t}}; x^{16}_{\mathtt{t}} \mid \epsilon_t \Rightarrow F \rightsquigarrow_{[t]}$

17. $x^7_{\mathtt{us}} \mid \epsilon_u \mid x^{12}_{\mathtt{t}}; x^{14}_{\mathtt{t}}; x^{15}_{\mathtt{t}}; x^{16}_{\mathtt{t}} \mid \epsilon_t \Rightarrow^1 x^{20}_{[\mathtt{s}]}, x^{20}_{[\mathtt{s}]} \Rightarrow F \rightsquigarrow_{[w],\mathtt{R2}}$

    In order to apply rule rewrite with rule R2 we compute ρ_2 and C_2, as in step 3, and use them to obtain a B-unifier. We skip these steps, and show the resulting reachability problem, where $x^{12}_{\mathtt{t}}$ and $x^{14}_{\mathtt{t}}$ have been moved to the workbench. Observe that the condition in R2 is trivial after substitution and normalization.

18. $\mathtt{tt} \to \mathtt{tt} \wedge x^7_{\mathtt{us}} \mid \epsilon_u \mid x^{15}_{\mathtt{t}}; x^{16}_{\mathtt{t}} \mid x^{12}_{\mathtt{t}}; x^{14}_{\mathtt{t}} \Rightarrow F \rightsquigarrow_{[e]}$

19. $x^7_{\mathtt{us}} \mid \epsilon_u \mid x^{15}_{\mathtt{t}}; x^{16}_{\mathtt{t}} \mid x^{12}_{\mathtt{t}}; x^{14}_{\mathtt{t}} \Rightarrow x_{\mathtt{u}} \mid y_{\mathtt{us}} \mid z^1_{\mathtt{t}}; z^2_{\mathtt{t}} \mid z^1_{\mathtt{t}}; z^2_{\mathtt{t}} \rightsquigarrow_{[x]}$

    Now we remove the $\Rightarrow$ symbol, unifying both terms.

20. $eq(x^7_{\mathtt{us}} \mid \epsilon_u \mid x^{15}_{\mathtt{t}}; x^{16}_{\mathtt{t}} \mid x^{12}_{\mathtt{t}}; x^{14}_{\mathtt{t}}, x_{\mathtt{u}} \mid y_{\mathtt{us}} \mid z^1_{\mathtt{t}}; z^2_{\mathtt{t}} \mid z^1_{\mathtt{t}}; z^2_{\mathtt{t}}) \to \mathtt{tt} \rightsquigarrow_{[t]} \rightsquigarrow_{[r],\sigma_4}$

    Again, the B-unifier σ_4 is obtained by previously computing ρ_3, where all sorted variables are replaced with kinded variables and C_3, which forces each kinded variable to have the specific sort that it had before applying ρ_3. As a final result, we get the computed answer $\sigma = \sigma_4|_{vars(G)} = \{x_{\mathtt{u}} \mapsto x^{21}_{\mathtt{u}}, y_{\mathtt{us}} \mapsto \epsilon_u, z^1_{\mathtt{t}} \mapsto x^{22}_{\mathtt{t}}, z^2_{\mathtt{t}} \mapsto x^{23}_{\mathtt{t}}\}$

21. $\mathtt{tt} \to \mathtt{tt} \rightsquigarrow_{[e]} \square$

So we have found a very general computed answer for our reachability problem $G \equiv \mathtt{init} \Rightarrow x_{\mathtt{u}} \mid y_{\mathtt{us}} \mid z^1_{\mathtt{t}}; z^2_{\mathtt{t}} \mid z^1_{\mathtt{t}}; z^2_{\mathtt{t}}$, where $x_{\mathtt{u}}$ can be any `user`, $y_{\mathtt{us}}$ must be `emptyU`, and $z^1_{\mathtt{t}}$ and $z^2_{\mathtt{t}}$ can be any `tool`.

4.7 Results and proofs

Here we show the proofs for this chapter, together with some needed technical results.

Theorem 4 (Reduction of $\to^1_{R/\mathcal{E}}$ to $\to^1_{R(E),B}$ and $\to_{R/\mathcal{E}}$ to $\to_{ER,B}$)**.** *Let $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ be a narrowable rewrite theory. Then, for any t_1, t_2 in $\mathcal{T}_\Sigma(\mathcal{X})$: (i) $t_1 \to^1_{R/\mathcal{E}} t_2$ if and only if $t_1{\downarrow} \to^1_{R(E),B} t_3$ for some $t_3 =_\mathcal{E} t_2$ (where $t_3{\downarrow} =_B t_2{\downarrow}$ serves as check) and (ii) $t_1 \to_{R/\mathcal{E}} t_2$ if and only if $t_1 \to_{ER,B} t_2$.*

Proof. The if part of (i) is immediate just by noticing that as $\mathcal{R}$ is narrowable then $t_1 =_\mathcal{E} t_1{\downarrow}$. As $t_1{\downarrow} \to^1_{R(E),B} t_3$ then, by Remark 6, $t_1{\downarrow} \to^1_{R/\mathcal{E}} t_3$ and, as a consequence, for any t_2 such that $t_2 =_\mathcal{E} t_3$ we get $t_1 \to^1_{R/\mathcal{E}} t_2$.

The if part of (ii) follows trivially from Remark 6.

We prove the only if parts of (i) and (ii) by induction on the *depth* of the derivations, i.e., if we represent the reachability conditions in a derivation as a tree, where the root is either $t_1{\downarrow} \to^1_{R/\mathcal{E}} t_2$ or $t_1{\downarrow} \to_{R/\mathcal{E}} t_2$ (in this case there will be as many branches as $\to^1_{R/\mathcal{E}}$ rewrite steps from t_1 to t_2), its depth is the length of the longest branch. We also prove that each derivation with $\to_{ER,B}$ (also having one branch for each $\to^1_{ER,B}$ step) has, at most, the same depth as the corresponding derivation with $\to_{R/\mathcal{E}}$.

As $(\Sigma, E \cup B)$ is executable then, by Section 2.3.8 (Executable MEL theory), $\to_{E/B} = \to_{E,B}$ so each condition in a rule that is not a reachability one will hold in $\to_{E/B}$ if and only it holds in $\to_{E,B}$, so we do not have to represent them in these trees.

- Base case, depth 1.

    (i) We have a rule with no reachability conditions on it where if $t_1 \to^1_{R/\mathcal{E}} t_2$ then $t_1 =_\mathcal{E} t \to^1_{R(\mathcal{E})} t' =_\mathcal{E} t_2$, for appropriate t and t'. Then, by Remark 1-(ii), we have that $t \to^1_R t'$, so also $t \to^1_{R(E),B} t'$, and we can draw the diagrams

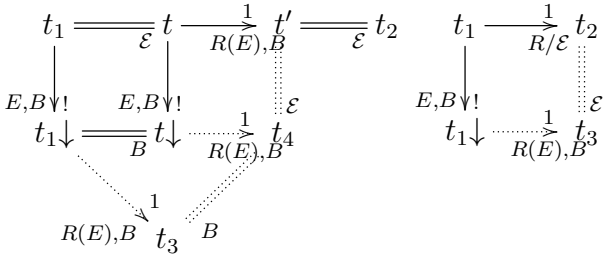

Figure 4.6: Reduction of $\to^1_{R/\mathcal{E}}$ to $\to^1_{R(E),B}$

on Figure 4.6. In the left diagram, the upper line represents our assumption on $t_1 \to^1_{R/\mathcal{E}} t_2$, replacing $t \to^1_{R(\mathcal{E})} t'$ with $t \to^1_{R(E),B} t'$; the left square follows from confluence and termination of E modulo B; the right square follows from $\mathcal{E}$-coherence of $\to^1_{R(E),B}$, since $t \to^1_{R(E),B} t'$; finally, the inverted triangle is a distorted version of the square that represents the strict coherence of $\to^1_{R(E),B}$. In the right diagram, which is a graphical representation of the statement (i), the upper line is the same, with an equivalent representation, the rest of the diagram is drawn from the left one just by taking in account the fact that $t_3 =_\mathcal{E} t_4 =_\mathcal{E} t' =_\mathcal{E} t_2$ implies $t_3 =_\mathcal{E} t_2$, where the derivation for $t_1{\downarrow} \to^1_{R(E),B} t_3$ has depth 1. This derivation can also be seen as $t_1 \to_{ER,B} t_2$.

(ii) As the depth is 1, the derivation for $t_1 \to_{R/\mathcal{E}} t'$ has no $\to^1_{R/\mathcal{E}}$ rewrite steps, which only happens when $t_1 =_E t'$. Then, by definition, $t_1 \to_{ER,B} t'$ with no $\to^1_{R(E),B}$ rewrite steps, hence also with depth 1.

- Induction case, depth $n > 1$.

  (i) $t_1 =_\mathcal{E} t \to^1_{R(\mathcal{E})} t' =_\mathcal{E} t_2$. As the rewrite step $t \to^1_{R(\mathcal{E})} t'$, using some instance of a rule c, has depth n, then each reachability condition derivation for $u_i \to_{R/\mathcal{E}} v_i$ from that instance of c has depth $n-1$ at most so, by induction hypothesis (I.H. from ow on), $u_i \to_{ER,B} v_i$ also having depth $n-1$ at most, hence $t \to^1_{R(E),B} t'$ with depth n at most, and Figure 4.6 again serves as proof for the theorem in this case.

  (ii) As $\to_{R/\mathcal{E}}$ is $\to^*_{R/\mathcal{E}} \cup =_\mathcal{E}$, recall that $\to_{ER,B}$ is $\to^*_{ER,B}; =_\mathcal{E}$ and $\to^1_{ER,B}$ is $(\to^!_{R,E}; \to^1_{R(E),B})$, there exists a rewrite path $t_1 \to^1_{R/\mathcal{E}} t_2 \ldots \to^1_{R/\mathcal{E}} t_{m-1} \to^1_{R/\mathcal{E}} t'$ where each one of the $\to^1_{R/\mathcal{E}}$ rewrite steps must have at most depth $n-1$. Then, as in the proof for the induction case (i), $t_i \to^1_{ER,B} t'_{i+1} =_\mathcal{E} t_{i+1}$ (†) having depth $n-1$ at most, for $1 \leq i < m$, where $t_m =_\mathcal{E} t'$. For $2 \leq i < m$, as $t'_i =_\mathcal{E} t_i$, then also $t'_i \to^1_{R/\mathcal{E}} t'_{i+1}$, and $t_1 \to^1_{R/\mathcal{E}} t'_2 \ldots \to^1_{R/\mathcal{E}} t'_{m-1} \to^1_{R/\mathcal{E}} t'_m =_\mathcal{E} t'$. Then, by (†), also $t_1 \to^1_{ER,B} t'_2 \ldots \to^1_{ER,B} t'_{m-1} \to^1_{ER,B} t'_m =_\mathcal{E} t'$, i.e., $t_1 \to_{ER,B} t'$, where each derivation has depth $n-1$ at most, so the derivation for $t_1 \to_{ER,B} t'$ has depth n at most.

□

Theorem 5 (Equivalence of $\mathcal{E}$-solutions for systems of sentences and unification goals)**.** *Given a* MEL *theory* $(\Sigma, E \cup B)$ *and its associated rewrite theory* $\mathcal{R}_E$*, if* $\to^1_{E,B}$ *is terminating, confluent, sort-decreasing, and closed under* B*-extensions, then for any system*

of sentences F, an idempotent normal substitution σ is an $\mathcal{E}$-solution for F iff σ is an $\mathcal{E}$-solution for its associated unification goal (G) in $\rightarrow_{E,B}$.

Proof. First we prove that if σ is an $\mathcal{E}$-solution of F then σ is an $\mathcal{E}$-solution for G. We prove it by induction on the number of deduction rules for MEL theories applied. We consider each possible type of sentence in F.

1. $t = t'$, and $t\sigma =_{\mathcal{E}} t'\sigma$, so $(t\sigma)\!\downarrow =_B (t'\sigma)\!\downarrow$. In G we have the subgoal $eq(t, t') \rightarrow \mathtt{tt}$, and $eq(t\sigma, t'\sigma) \rightarrow^*_{E,B} eq((t\sigma)\!\downarrow, (t'\sigma)\!\downarrow) \rightarrow^1_{E,B} \mathtt{tt} =_B \mathtt{tt}$, with $k = [ls((t\sigma)\!\downarrow)]$, rule $eq(x_k, x_k) \rightarrow \mathtt{tt}$, and substitution $\{x_k \mapsto (t\sigma)\!\downarrow\}$ so, by definition $eq(t\sigma, t'\sigma) \rightarrow_{E,B} \mathtt{tt}$. Then σ is an $\mathcal{E}$-solution for $eq(t, t') \rightarrow \mathtt{tt}$.

2. $t := t'$, and $t\sigma =_{\mathcal{E}} t'\sigma$, so $(t\sigma)\!\downarrow =_B (t'\sigma)\!\downarrow$. As t is a Σ-pattern and σ is E, B-normalized then $(t\sigma)\!\downarrow \equiv t\sigma$ so $t\sigma =_B (t'\sigma)\!\downarrow$. In G we have the subgoal $t' \rightarrow t$, and $t'\sigma \rightarrow^*_{E,B} (t'\sigma)\!\downarrow =_B t\sigma$. Then, by definition, $t'\sigma \rightarrow_{E,B} t\sigma$ so σ is an $\mathcal{E}$-solution for $t' \rightarrow t$ in $\rightarrow_{E,B}$.

3. $t : s$, and $t\sigma :_{\mathcal{E}} s$. We consider two subcases, depending on the deduction rule applied.

   - Rule replacement. Then we infer $t\sigma :_{\mathcal{E}} s$ because there is a sentence $l : s$ *if* $\bigwedge_{i=1}^n A_i$ in E, and a substitution ρ such that $t\sigma \equiv l\rho$, and ρ is an $\mathcal{E}$-solution for A_i, $1 \leq i \leq n$. In G we have the subgoal $t{:}s \rightarrow \mathtt{tt}$, and in R_E we have the rule $l{:}s \rightarrow \mathtt{tt}$ *if* $\bigwedge_{i=1}^n A'_i$ where, by I.H., ρ is a solution for A'_i, $1 \leq i \leq n$, in $\rightarrow_{E,B}$, so $l\rho{:}s \rightarrow^1_{E,B} \mathtt{tt}$, that is, $t\sigma{:}s \rightarrow^1_{E,B} \mathtt{tt} =_B \mathtt{tt}$ so, by definition, $t\sigma{:}s \rightarrow_{E,B} \mathtt{tt}$ so σ is an $\mathcal{E}$-solution for $t{:}s \rightarrow \mathtt{tt}$ in $\rightarrow_{E,B}$.
   - Rule membership. Then we infer $t\sigma :_{\mathcal{E}} s$ because $t\sigma =_{\mathcal{E}} t'$ and $t' :_{\mathcal{E}} s$ is deduced with rule replacement. The case where several rules membership are applied before applying rule replacement is easily reduced to this one using rule transitivity: if $t\sigma =_{\mathcal{E}} t_1 =_{\mathcal{E}} \ldots =_{\mathcal{E}} t'$, then $t\sigma =_{\mathcal{E}} t'$, so $(t\sigma)\!\downarrow =_B t'\!\downarrow$. There is a sentence $l : s$ *if* $\bigwedge_{i=1}^n A_i$ in E, and a substitution ρ such that $t' \equiv l\rho$, and ρ is an $\mathcal{E}$-solution for A_i, $1 \leq i \leq n$. In G we have the subgoal $t{:}s \rightarrow \mathtt{tt}$, and in R_E we have the rule $l{:}s \rightarrow \mathtt{tt}$ *if* $\bigwedge_{i=1}^n A'_i$ where, by I.H., ρ is a solution for A'_i, $1 \leq i \leq n$, in $\rightarrow_{E,B}$, so $l\rho{:}s \rightarrow^1_{E,B} \mathtt{tt}$, that is, $t'{:}s \rightarrow^1_{E,B} \mathtt{tt}$. As $t' \rightarrow^*_{E,B} t'\!\downarrow$, we also can apply the same rules to t' in the context $t'{:}s$, so $t'{:}s \rightarrow^*_{E,B} t'\!\downarrow{:}s$. As $\mathtt{tt}$ is a canonical form and $t'{:}s \rightarrow^1_{E,B} \mathtt{tt}$ then, by confluence, $t'\!\downarrow{:}s \rightarrow^*_{E,B} \mathtt{tt}$. But, as $(t\sigma)\!\downarrow{:}s =_B t'\!\downarrow{:}s$ (because $(t\sigma)\!\downarrow =_B t'\!\downarrow$) then, by strict coherence of $\rightarrow^1_{E,B}$, $t'\!\downarrow{:}s \rightarrow^*_{E,B} \mathtt{tt}$ implies $(t\sigma)\!\downarrow{:}s \rightarrow^*_{E,B} \mathtt{tt}$. As $t\sigma{:}s \rightarrow^*_{E,B} (t\sigma)\!\downarrow{:}s$, we conclude that $t\sigma{:}s \rightarrow^*_{E,B} \mathtt{tt} =_B \mathtt{tt}$ so, by definition, $t\sigma{:}s \rightarrow_{E,B} \mathtt{tt}$ so σ is an $\mathcal{E}$-solution for $t{:}s \rightarrow \mathtt{tt}$ in $\rightarrow_{E,B}$.

Now we prove that if σ is an $\mathcal{E}$-solution for G then σ is an $\mathcal{E}$-solution of F. We prove it by induction on the total number of rewrite steps applied. We consider each possible type of subgoal in G.

1. $eq(t, t') \rightarrow \mathtt{tt}$. Then $eq(t\sigma, t'\sigma) \rightarrow_{E,B} \mathtt{tt}$.

   - One rewrite step: then $eq(t\sigma, t'\sigma) \rightarrow_{E,B} \mathtt{tt}$ with rule $eq(x_k, x_k) \rightarrow \mathtt{tt}$ because $t\sigma =_B t'\sigma$, so $t\sigma =_{\mathcal{E}} t'\sigma$ because $A \subseteq \mathcal{E}$. The sentence in F is $t = t'$, and $t\sigma =_{\mathcal{E}} t'\sigma$, so σ is an $\mathcal{E}$-solution for $t = t'$.

- $n > 1$ rewrite steps: without loss of generality we assume that the rewritten term in the first rewrite step is t. Then $eq(t\sigma, t'\sigma) \rightarrow^1_{E,B} eq((t\sigma)[r\rho]_p, t'\sigma) \rightarrow_{E,B}$ $\mathtt{tt}$ with rule $l \rightarrow r$ *if* c' in R_E, because $(t\sigma)|_p =_B l\rho$ (so $(t\sigma)|_p =_{\mathcal{E}} l\rho$) and ρ is an $\mathcal{E}$-solution for all the conditions in c'. Then there must be a corresponding equation $l = r$ *if* c in E (the only rules that don't have a counterpart are those related to the new sort *Truth*, and no subterm of t and t' can have a sort *Truth* or kind [*Truth*]) where, by I.H., ρ is an $\mathcal{E}$-solution for all the conditions in c. By replacement rule, we can deduce $l\rho =_{\mathcal{E}} r\rho$. Then, by repeated application of the congruence rule, we can deduce $(t\sigma)[l\rho]_p =_{\mathcal{E}} (t\sigma)[r\rho]_p$. As $(t\sigma)|_p =_{\mathcal{E}} l\rho$ we can also deduce $t\sigma =_{\mathcal{E}} (t\sigma)[l\rho]_p$ by repeated application of the congruence rule. Then, using the transitivity rule, we can deduce $t\sigma =_{\mathcal{E}} (t\sigma)[r\rho]_p$. As σ is idempotent, r has fresh variables, and $Ran(\rho)$ is away from $vars((t\sigma)[r])$, then $(t\sigma)[r\rho]_p\sigma = (t\sigma)[r\rho]_p$, and $eq((t\sigma)[r\rho]_p\sigma, t'\sigma) \rightarrow_{E,B} \mathtt{tt}$ with less than n rewrite steps so, by I.H., σ is an $\mathcal{E}$-solution for the sentence $(t\sigma)[r\rho]_p = t'$, and then $t\sigma =_{\mathcal{E}} t'\sigma$.

2. $t{:}s \rightarrow \mathtt{tt}$. Then $t\sigma{:}s \rightarrow_{E,B} \mathtt{tt}$.

   - One rewrite step: then $t\sigma{:}s \rightarrow_{E,B} \mathtt{tt}$ with rule $l{:}s \rightarrow \mathtt{tt}$ in R_E because there is a substitution ρ such that $t\sigma =_B l\rho$. The sentence in F is $t : s$, and there is a sentence $l : s$ in E and $t\sigma =_B l\rho$, so $t\sigma :_{\mathcal{E}} s$ and σ is an $\mathcal{E}$-solution for $t : s$.
   - $n > 1$ rewrite steps: then $t\sigma{:}s \rightarrow^1_{E,B} (t\sigma)[r\rho]_p{:}s \rightarrow_{E,B} \mathtt{tt}$ with rule $l \rightarrow r$ *if* c' in R_E, because $(t\sigma)|_p =_B l\rho$ (so $(t\sigma)|_p =_{\mathcal{E}} l\rho$) and ρ is an $\mathcal{E}$-solution for all the conditions in c'. Then there must be a corresponding equation $l = r$ *if* c in E where, by I.H., ρ is an $\mathcal{E}$-solution for all the conditions in c. By replacement rule, we can deduce $l\rho =_{\mathcal{E}} r\rho$. Then, by repeated application of the congruence rule, we can deduce $(t\sigma)[l\rho]_p =_{\mathcal{E}} (t\sigma)[r\rho]_p$. As $(t\sigma)|_p =_{\mathcal{E}} l\rho$ we can also deduce $t\sigma =_{\mathcal{E}} (t\sigma)[l\rho]_p$ by repeated application of the congruence rule. Then, using the transitivity rule, we can deduce $t\sigma =_{\mathcal{E}} (t\sigma)[r\rho]_p$. As σ is idempotent, r has fresh variables, and $Ran(\rho)$ is away from $vars((t\sigma)[r])$, then $(t\sigma)[r\rho]_p\sigma = (t\sigma)[r\rho]_p$, and $(t\sigma)[r\rho]_p\sigma{:}s \rightarrow_{E,B} \mathtt{tt}$ with less than n rewrite steps so, by I.H., σ is an $\mathcal{E}$-solution for the sentence $(t\sigma)[r\rho]_p : s$, and then by rule membership $t\sigma :_{\mathcal{E}} s$.

3. $t' \rightarrow t$, with $t \neq \mathtt{tt}$. Then $t'\sigma \rightarrow_{E,B} t\sigma$.

   - Zero rewrite steps: then $t\sigma =_B t'\sigma$, so $t\sigma =_{\mathcal{E}} t'\sigma$ because $A \subseteq \mathcal{E}$. The sentence in F is $t := t'$, and $t\sigma =_{\mathcal{E}} t'\sigma$, so σ is an $\mathcal{E}$-solution for $t := t'$.
   - $n > 0$ rewrite steps: then $t\sigma \rightarrow^1_{E,B} (t\sigma)[r\rho]_p \rightarrow_{E,B} t'\sigma$ with rule $l \rightarrow r$ *if* c' in R_E, because $(t\sigma)|_p =_B l\rho$ (so $(t\sigma)|_p =_{\mathcal{E}} l\rho$) and ρ is an $\mathcal{E}$-solution for all the conditions in c'. Then there must be a corresponding equation $l = r$ *if* c in E where, by I.H., ρ is an $\mathcal{E}$-solution for all the conditions in c. By replacement rule, we can deduce $l\rho =_{\mathcal{E}} r\rho$. Then, by repeated application of the congruence rule, we can deduce $(t\sigma)[l\rho]_p =_{\mathcal{E}} (t\sigma)[r\rho]_p$. As $(t\sigma)|_p =_{\mathcal{E}} l\rho$ we can also deduce $t\sigma =_{\mathcal{E}} (t\sigma)[l\rho]_p$ by repeated application of the congruence rule. Then, using the transitivity rule, we can deduce $t\sigma =_{\mathcal{E}} (t\sigma)[r\rho]_p$. As σ is idempotent, r has fresh variables, and $Ran(\rho)$ is away from $vars((t\sigma)[r])$, then $(t\sigma)[r\rho]_p\sigma = (t\sigma)[r\rho]_p$, and $(t\sigma)[r\rho]_p\sigma \rightarrow_{E,B} t'\sigma$ with less than n rewrite steps so, by I.H., σ is an $\mathcal{E}$-solution for the sentence $(t\sigma)[r\rho]_p = t'$, and then $t\sigma =_{\mathcal{E}} t'\sigma$, i.e., σ is an $\mathcal{E}$-solution for $t := t'$.

□

Proposition 4. *Given an FPP executable* MEL *theory* $(\Sigma, \mathcal{E})$*, its associated rewrite theory* $\mathcal{R}_E = (\Sigma', B, R_E)$*, and a term* $t \in T_{\Sigma'}(\mathcal{X})$*, if* $t \rightarrow^1_{E,B} t[r\sigma]_p$ *using a rule* $c \equiv l \rightarrow r$ *if* $\bigwedge_{i=1}^n A'_i$ *in* R_E *and an idempotent substitution* σ*, then* $t \rightarrow^1_{E,B} t[r\sigma_c]_p$ *using the same rule* c *and substitution* σ_c*.*

Proof. For unconditional rules there is nothing to prove, because $\sigma_c \equiv \sigma$, so we focus on conditional rules. By definition of σ_c we have that as $t =_B l\sigma$ then $t =_B l\sigma_c$. Now we prove that for $i = 1, \ldots, n$ if the condition $A'_i\sigma$ is verified then the condition $A'_i\sigma_c$ is also verified. We prove it for the three types of conditions:

1. Case $eq(t_i\sigma, t'_i\sigma) \rightarrow_{E,B} \texttt{tt}$. Then as $eq(t_i\sigma, t'_i\sigma) \rightarrow^*_{E,B} eq(t_i\sigma_c, t'_i\sigma_c)$, by confluence of $\rightarrow^1_{E,B}$, as $\texttt{tt}$ is a unique normal form, we get $eq(t_i\sigma_c, t'_i\sigma_c) \rightarrow_{E,B} \texttt{tt}$.

2. Case $t_i\sigma{:}s \rightarrow_{E,B} \texttt{tt}$. We use the equivalence between $\rightarrow_{E,B}$ and $:_{\mathcal{E}}$. Then $t_i\sigma :_{\mathcal{E}} s$. As $t_i\sigma \rightarrow^*_{E,B} t_i\sigma_c$ and $\rightarrow^1_{E,B}$ is sort decreasing then $t_i\sigma_c$ has sort s, so $t_i\sigma_c :_{\mathcal{E}} s$ must be derivable. By the equivalence between $\rightarrow_{E,B}$ and $:_{\mathcal{E}}$, $t_i\sigma_c{:}s \rightarrow_{E,B} \texttt{tt}$

3. Case $t'_i\sigma \rightarrow_{E,B} t_i\sigma$ (from $t_i := t'_i$). Then $t'_i\sigma \rightarrow^*_{E,B} u =_B t_i\sigma$. As for all rules $c \equiv l \rightarrow r$ *if* C if $l\rho =_B u$ we also have that $l\rho =_B t_i\sigma$, then the rewrite steps in $\rightarrow^1_{E,B}$ are the same for u and $t_i\sigma$. As $t_i\sigma \rightarrow^*_{E,B} t_i\sigma_c$, we have $t'_i\sigma \rightarrow^*_{E,B} t_i\sigma_c$. We also have $t'_i\sigma \rightarrow^*_{E,B} t'_i\sigma_c$. But $t_i\sigma_c$ is a normal form because t_i is a Σ-pattern, all the variables in t_i are matching variables in c and σ_c is E, B-normalized with respect to all matching variables in c. Then, by confluence, it must be the case that $t'_i\sigma_c \rightarrow^*_{E,B} t' =_B t_i\sigma_c$, so $t'_i\sigma_c \rightarrow_{E,B} t_i\sigma_c$.

□

Proposition 5. *Given a narrowable rewrite theory* $\mathcal{R} = (\Sigma, \mathcal{E}, R)$*, its FPP executable* MEL *theory* $(\Sigma, E \cup B)$*, the associated rewrite theory* $\mathcal{R}_E = (\Sigma', B, R_E)$*, and two terms* t, t' *in* $T_{\Sigma'}(\mathcal{X})$*, if* $t \rightarrow_{ER,B} t'$ *then* $t{\downarrow} \rightarrow_{ER,B} t'$*.*

Proof. By induction on the sum of $\rightarrow^1_{E,B}$ and $\rightarrow^1_{R(E),B}$ steps. Remember that $=_B \subseteq =_{\mathcal{E}}$.

Base case. Zero steps. Then $t =_{\mathcal{E}} t'$, so $[t]_{\mathcal{E}} = [t']_{\mathcal{E}}$. As $[t]_{\mathcal{E}} = [t{\downarrow}]_{\mathcal{E}}$ then $[t{\downarrow}]_{\mathcal{E}} = [t']_{\mathcal{E}}$, so $t{\downarrow} =_{\mathcal{E}} w$.

Induction case. We consider two cases depending on the first rule used.

- $t \rightarrow^1_{E,B} u \rightarrow_{ER,B} t'$. By I.H. $u{\downarrow} \rightarrow^*_{ER,B} w =_{\mathcal{E}} t'$, but $u{\downarrow} =_B t{\downarrow}$ by confluence of $\rightarrow^1_{E,B}$. Then, by strict coherence of $\rightarrow^1_{E,B}$ and $\rightarrow^1_{R(E),B}$, $t{\downarrow} \rightarrow^*_{ER,B} w' =_B w$, so $t{\downarrow} \rightarrow_{ER,B} t'$.

- $t \rightarrow^1_{R(E),B} u \rightarrow_{ER,B} t'$. By I.H. $u{\downarrow} \rightarrow_{ER,B} t'$. As $t \rightarrow^!_{E,B} t{\downarrow}$, then by $\mathcal{E}$-consistency of $\rightarrow^1_{R(E),B}$ with $\rightarrow^1_{E,B}$, $t{\downarrow} \rightarrow^1_{R(E),B} u' =_{\mathcal{E}} u$. By confluence of $\rightarrow^1_{E,B}$, $u' \rightarrow^!_{E,B} u'{\downarrow} =_B u{\downarrow}$ so, by strict coherence of $\rightarrow^1_{E,B}$ and $\rightarrow^1_{R(E),B}$, $u'{\downarrow} \rightarrow_{ER,B} w =_B t'$. Putting all together, and by definition of $\rightarrow_{ER,B}$, $t{\downarrow} \rightarrow^1_{R(E),B} u' \rightarrow^!_{E,B} u'{\downarrow} \rightarrow^*_{ER,B} w' =_{\mathcal{E}} w =_B t'$, so $t{\downarrow} \rightarrow_{ER,B} t'$.

□

Proposition 6. *Given a narrowable rewrite theory* $\mathcal{R} = (\Sigma, \mathcal{E}, R)$*, its FPP executable* MEL *theory* $(\Sigma, E \cup B)$*, the associated rewrite theory* $\mathcal{R}_E = (\Sigma', B, R_E)$*, and a term* $t \in T_{\Sigma'}(\mathcal{X})$*, if* $t \rightarrow^1_{R(E),B} t[r\sigma]_p$ *using a rule* $c \equiv l \rightarrow r$ *if* $\bigwedge_{i=1}^n A_i$ *in* R *and an idempotent substitution* σ*, then* $t \rightarrow^1_{R(E),B} t[r\sigma_c]_p$ *using the same rule* c *and substitution* σ_c*.*

Proof. For unconditional rules there is nothing to prove, because $\sigma_c \equiv \sigma$, so we focus on conditional rules. By definition of σ_c we have that as $t =_B l\sigma$ then $t =_B l\sigma_c$. Now we prove that for $i = 1, \ldots, n$ if the condition $A_i\sigma$ is verified then the condition $A_i\sigma_c$ is also verified. We have already proved it for the three types of equational conditions in Proposition 4 using the associated condition A'_i, so the only case left to prove is the one where $A_i \equiv t_i \to t'_i$ and $t_i\sigma \to_{ER,B} t'_i\sigma$. We prove it by induction on the sum of $\to^1_{E,B}$ and $\to^1_{R(E),B}$ steps steps, including those due to all rewriting conditions in the rewriting path.

- Base case. Zero steps. Then $t_i\sigma =_{\mathcal{E}} t'_i\sigma$, and as $t_i\sigma \to_{E,B} t_i\sigma_c$, and $t'_i\sigma \to_{E,B} t'_i\sigma_c$ then $t_i\sigma_c =_{\mathcal{E}} t'_i\sigma_c$.
- Induction case. We consider two cases depending on the first rule used.
  - $t_i\sigma \to^1_{E,B} u \to_{ER,B} t'_i\sigma$. As σ is idempotent then $t_i\sigma \to^1_{E,B} u\sigma \to_{ER,B} t'_i\sigma$. As in case 3 of Proposition 4, $t_i\sigma_c \to_{E,B} t'\sigma_c$, and by I.H. $u\sigma_c \to_{ER,B} t'_i\sigma_c$, so $t_i\sigma_c \to_{ER,B} t'_i\sigma_c$.
  - $t_i\sigma \to^1_{R(E),B} u \to_{ER,B} t'_i\sigma$. As σ is idempotent then $t_i\sigma \to^1_{R(E),B} u\sigma \to_{ER,B} t'_i\sigma$. As $t_i\sigma \to^*_{E,B} t_i\sigma_c$ then, by $\mathcal{E}$-consistency of $\to^1_{R(E),B}$ with $\to^1_{E,B}$, $t_i\sigma_c \to^*_{E,B} \to^1_{R(E),B} w =_{\mathcal{E}} u\sigma$. $u\sigma \to^*_{E,B} u\sigma_c \to^!_{E,B} (u\sigma)\!\downarrow$ and, by I.H., $u\sigma_c \to_{ER,B} t'_i\sigma_c$. Then, by Proposition 5 $(u\sigma)\!\downarrow \to^*_{ER,B} w' =_{\mathcal{E}} w$. As $w =_{\mathcal{E}} u\sigma$ then $w \to^!_{E,B} w\!\downarrow =_B (u\sigma)\!\downarrow$. Then, by strict coherence of $\to^1_{E,B}$ and $\to^1_{R(E),B}$, $w\!\downarrow \to^*_{ER,B} w'' =_B w'$. Putting all together, $t_i\sigma_c \to^*_{E,B} \to^1_{R(E),B} w \to^!_{E,B} w\!\downarrow \to^*_{ER,B} w'' =_B w' =_{\mathcal{E}} t'_i\sigma_c$, so $t_i\sigma_c \to_{ER,B} t'_i\sigma_c$.

□

Lemma 4 (Completeness of sentence-normalized rewriting to normal form)**.** *Given an FPP executable* MEL *theory* $(\Sigma, E \cup B)$*, its associated rewrite theory* $\mathcal{R}_E = (\Sigma', B, R_E)$*, and terms* $t, t' \in T_{\Sigma'}(\mathcal{X})$*, if* $t \to_{E,B} t'$ *and* t' *is* E, B*-irreducible then* $t \to_N t'$*.*

Proof. By induction on the total number of $\to^1_{E,B}$ steps.

- Base case: $t \to_{E,B} t'$ with zero $\to^1_{E,B}$ steps. Then $t' =_B t$, so $t' \to_N t$.
- Induction case: $t \to^1_{E,B} t[r\sigma]_p \to_{E,B} t'$ with a rule $c \equiv l \to r$ *if* $\bigwedge_{i=1}^n u_i \to v_i$ and a substitution σ. By Proposition 4, $t \to^1_{E,B} t[r\sigma_c]_p$ with rule c and substitution σ_c, so $u_i\sigma_c \to_{E,B} v_i\sigma_c$ for $1 \le i \le n$. For $1 \le i \le n$ the term v_i in rule c must be a Σ-pattern (maybe with form `tt`), and σ_c is E, B-normalized with respect to $vars(v_i)$, so $v_i\sigma_c$ is E, B-irreducible. Then, by I.H., $u_i\sigma_c \to_N v_i\sigma_c$ for $1 \le i \le n$, so $t \to^1_N t[r\sigma_c]_p$ by definition. We choose the derivation $t \to^1_N t[r\sigma_c]_p \to_{E,B} t'$ which must exist by confluence of $\to^1_{E,B}$ because $t[r\sigma]_p \to^*_{E,B} t[r\sigma_c]_p$, t' is E, B-irreducible, and $\to^1_N \subseteq \to^1_{E,B}$. By I.H. $t[r\sigma_c]_p \to_N t'$, so $t \to_N t'$.

□

Proposition 1 (Rewriting in $\Rightarrow_N$ with normal forms)**.** *Given a narrowable rewrite theory* $\mathcal{R} = (\Sigma, E \cup B, R)$*, its FPP executable* MEL *theory* $(\Sigma, E \cup B)$*, the associated rewrite theory* $\mathcal{R}_E = (\Sigma', B, R_E)$*, and two terms* t, t' *in* $T_\Sigma(\mathcal{X})$*, if* $t \Rightarrow_N t'$ *then* $t\!\downarrow \Rightarrow_N t'$*.*

Proof. We distinguish two cases depending on the number of $\Rightarrow^1_N$ steps.

- Zero $\Rightarrow_N^1$ steps. Then $t =_{\mathcal{E}} t'$. As $t =_{\mathcal{E}} t{\downarrow}$ then $t{\downarrow} =_{\mathcal{E}} t'$, so $t{\downarrow} \Rightarrow_N t'$.
- At least one $\Rightarrow_N^1$ step. Then, by definition, $t \Rightarrow_N^1 u \Rightarrow_N t'$, i.e., $t \rightarrow_{E,B}^! t{\downarrow} \rightarrow_{R(E),B}^1 u \Rightarrow_N t'$ using SNR, hence $t{\downarrow} \rightarrow_{R(E),B}^1 u \Rightarrow_N t'$, which is $t{\downarrow} \Rightarrow_N^1 u \Rightarrow_N t'$ in this case, hence $t{\downarrow} \Rightarrow_N t'$.

□

Lemma 5 (Completeness of Sentence-normalized Rewriting for $\rightarrow_{R/\mathcal{E}}$)**.** *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$, its FPP executable* MEL *theory $(\Sigma, E \cup B)$, the associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, and terms t, t' in $T_\Sigma(\mathcal{X})$, if $t \rightarrow_{R/\mathcal{E}} t'$ then $t \Rightarrow_N t'$.*

Proof. By Theorem 4, as $t \rightarrow_{R/\mathcal{E}} t'$ then $t \rightarrow_{ER,B} t'$. We distinguish two cases depending on the number of $\Rightarrow_N^1$ steps.

- Zero $\rightarrow_{ER,B}^1$ steps. Trivial because then $t =_{\mathcal{E}} t'$.
- At least one $\rightarrow_{ER,B}^1$ step. Then $t \rightarrow_{E,B}^! t{\downarrow} \rightarrow_{R(E),B}^1 u \rightarrow_{ER,B} t'$. By Lemma 4, $t \rightarrow_N^! t{\downarrow}$, By Proposition 1, $t{\downarrow} \Rightarrow_N^1 u$, and, by I.H., $u \rightarrow_N t'$. Putting all together: $t \rightarrow_N^! t{\downarrow} \Rightarrow_N^1 u \Rightarrow_N t'$, i.e., $t \Rightarrow_N^1 u \Rightarrow_N t'$, hence $t \Rightarrow_N t'$.

□

Theorem 6 (Equivalence of SNR for Solutions of Associated Unification Goals)**.** *Given an FPP executable* MEL *theory $(\Sigma, E \cup B)$ and its associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, a normal substitution σ is an $\mathcal{E}$-solution of a system of sentences F and an E, B-solution of its associated unification goal $G = \bigwedge_{i=1}^n (t_i \rightarrow t_i')$ (so $t_i\sigma \rightarrow_{E,B} t_i'\sigma$, for $1 \leq i \leq n$) iff $t_i\sigma \rightarrow_N t_i'\sigma$, $i = 1, \ldots, n$ (i.e. σ is an N-solution for G).*

Proof. We prove each part of the double implication separately.

- $\Rightarrow$: for $1 \leq i \leq n$ the term t_i' is a Σ-pattern (maybe with form `tt`) by definition of system of sentences and associated unification goal, so $t_i'\sigma$ is E, B-irreducible. As $t_i\sigma \rightarrow_{E,B} t_i'\sigma$ then, by Lemma 4, $t_i\sigma \rightarrow_N t_i'\sigma$.
- $\Leftarrow$: Immediate since $\rightarrow_N^1 \subseteq \rightarrow_{E,B}^1$.

□

Proposition 2. *Given an FPP executable* MEL *theory $(\Sigma, \mathcal{E})$ and its associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, for any conditional* MEL *sentence c in E, and corresponding rule $c' \equiv s'$ if $\bigwedge_{i=1}^n u_i \rightarrow v_i$ in $\mathcal{R}_E$, if there is an E, B-normalized substitution σ such that $u_i\sigma \rightarrow_{E,B} v_i\sigma$, for $1 \leq i \leq n$, then $u_i{\downarrow}\sigma \rightarrow_N v_i\sigma$, for $1 \leq i \leq n$.*

Proof. Immediate since FPP and executability imply that, for $1 \leq i \leq n$, v_i is a Σ-pattern and σ is E, B-normalized, so $v_i\sigma$ is E, B-irreducible, and $u_i\sigma \rightarrow_{E,B} v_i\sigma$. $u_i \rightarrow_{E,B} u_i{\downarrow}$ implies $u_i\sigma \rightarrow_{E,B} u_i{\downarrow}\sigma$ so, by confluence of $\rightarrow_{E,B}^1$, $u_i{\downarrow}\sigma \rightarrow_{E,B} v_i\sigma$. Then by Lemma 4, as $v_i\sigma$ is E, B-irreducible, $u_i{\downarrow}\sigma \rightarrow_N v_i\sigma$. □

Lemma 6 (Equivalence of Solutions for Normal Unification Goals)**.** *Given an FPP executable* MEL *theory $(\Sigma, E \cup B)$ and its associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, a normal substitution σ is an E, B-solution of the unification goal $G = \bigwedge_{i=1}^n (t_i \rightarrow t_i')$, associated to a system of sentences F, iff $t_i{\downarrow}\sigma \rightarrow_N t_i'\sigma$, for $1 \leq i \leq n$.*

Proof. We have written $t'_i\sigma$ instead of $t'_i{\downarrow}\sigma$ because for unification goals associated to a system of sentences it is always the case that $t'_i{\downarrow} \equiv t'_i$ (t'_i must be $\mathtt{tt}$ or a Σ-pattern). We prove each part of the double implication separately.

- $\Rightarrow$: $t_i\sigma \rightarrow_{E,B} t'_i\sigma$. Since a unification goal associated to a system of sentences has the same form and restrictions as the conditions of the rules in R_E, then, by Proposition 2, $t_i{\downarrow}\sigma \rightarrow_N t'_i\sigma$.
- $\Leftarrow$: $t_i{\downarrow}\sigma \rightarrow_N t'_i\sigma$. As $\rightarrow^1_N \subseteq \rightarrow^1_{E,B}$ then $t_i{\downarrow}\sigma \rightarrow_{E,B} t'_i\sigma$. $t_i \rightarrow^*_{E,B} t_i{\downarrow}$ implies $t_i\sigma \rightarrow^*_{E,B} t_i{\downarrow}\sigma$. As a consequence of the last two deductions $t_i\sigma \rightarrow_{E,B} t'_i\sigma$.

□

Lemma 7 (Equivalence of Solutions for Normalized Reachability Goals)**.** *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$, its FPP executable* Mel *theory $(\Sigma, E \cup B)$, the associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, and a reachability goal $G = u_1 \Rightarrow v_1 \wedge \ldots \wedge u_n \Rightarrow v_n \wedge G'$, where G' is a unification goal associated to a system of sentences F, and an idempotent normal substitution σ, these three assertions are equivalent:*

1. *σ is a solution for G,*
2. *σ is an N-solution for $G{\downarrow}$,*
3. *σ is a solution for $G{\downarrow}$.*

Proof. We prove $1 \Rightarrow 2$, $2 \Rightarrow 3$, and $3 \Rightarrow 1$.

- $1 \Rightarrow 2$

  By induction on the number of $\Rightarrow^1_{R/\mathcal{E}}$ steps.

  Base case: zero $\Rightarrow^1_{R/\mathcal{E}}$ steps. Then, by Definition 4.2.3, σ is a trivial solution of G, so $u_i\sigma =_{\mathcal{E}} v_i\sigma$, for $1 \leq i \leq n$, and σ is an $\mathcal{E}$-solution for G'. Then, by Lemma 6, σ is also a solution for $u_i{\downarrow} =_{\mathcal{E}} v_i{\downarrow}$, for $1 \leq i \leq n$, and also for $G'{\downarrow}$, i.e., σ is a trivial solution for $G{\downarrow}$.

  Induction case: without losing generality, we assume that $u_1\sigma \Rightarrow^1_{R/\mathcal{E}} t \Rightarrow_{R/\mathcal{E}} v_1\sigma$. As σ is idempotent, then $u_1\sigma \Rightarrow^1_{R/\mathcal{E}} t\sigma \Rightarrow_{R/\mathcal{E}} v_1\sigma$. By definition of $=_{\mathcal{E}}$ we have that $u_1 =_{\mathcal{E}} u_1{\downarrow}$ and $t =_{\mathcal{E}} t{\downarrow}$, so $u_1\sigma =_{\mathcal{E}} u_1{\downarrow}\sigma$ and $t\sigma =_{\mathcal{E}} t{\downarrow}\sigma$. Then, by definition of $\Rightarrow^1_{R/\mathcal{E}}$, $u_1{\downarrow}\sigma \Rightarrow^1_{R/\mathcal{E}} t{\downarrow}\sigma$ and, by Lemma 5, $u_1{\downarrow}\sigma \Rightarrow^1_N t{\downarrow}\sigma$. By I.H., $t{\downarrow}\sigma \Rightarrow_N v_1{\downarrow}\sigma$, so $u_1{\downarrow}\sigma \Rightarrow_N v_1{\downarrow}\sigma$. Also by I.H. $u_i{\downarrow}\sigma \Rightarrow_N v_i{\downarrow}\sigma$, for $2 \leq i \leq n$, and σ is an N-solution for $G'{\downarrow}$, so σ is an N-solution for $G{\downarrow}$.

- $2 \Rightarrow 3$

  Trivial because $\Rightarrow^1_N \subseteq \Rightarrow^1_{ER,B} \subseteq \Rightarrow^1_{R/\mathcal{E}}$ and σ is an N-solution for $G{\downarrow}$.

- $3 \Rightarrow 1$

  Again by induction on the number of $\Rightarrow^1_{R/\mathcal{E}}$ steps.

  Base case: zero $\Rightarrow^1_{R/\mathcal{E}}$ steps. Then, by Definition 4.2.3, σ is a trivial solution of $G{\downarrow}$, so $u_i{\downarrow}\sigma =_{\mathcal{E}} v_i{\downarrow}\sigma$, for $1 \leq i \leq n$, and σ is an $\mathcal{E}$-solution for $G'{\downarrow}$. Then, by Lemma 6, σ is also a solution for $u_i =_{\mathcal{E}} v_i$, for $1 \leq i \leq n$, and also for G', i.e., σ is a trivial solution for G.

Induction case: without losing generality, we assume that $u_1{\downarrow}\sigma \Rightarrow^1_{R/\mathcal{E}} t \Rightarrow_{R/\mathcal{E}} v_1{\downarrow}\sigma$. As $t =_{\mathcal{E}} t{\downarrow}$ then $u_1{\downarrow}\sigma \Rightarrow^1_{R/\mathcal{E}} t{\downarrow} \Rightarrow_{R/\mathcal{E}} v_1{\downarrow}\sigma$ by definition of $\Rightarrow^1_{R/\mathcal{E}}$. As σ is idempotent, then $u_1{\downarrow}\sigma \Rightarrow^1_{R/\mathcal{E}} t{\downarrow}\sigma \Rightarrow_{R/\mathcal{E}} v_1{\downarrow}\sigma$. By definition of $=_{\mathcal{E}}$ we have that $u_1 =_{\mathcal{E}} u_1{\downarrow}$ and $t =_{\mathcal{E}} t{\downarrow}$, so $u_1\sigma =_{\mathcal{E}} u_1{\downarrow}\sigma$ and $t\sigma =_{\mathcal{E}} t{\downarrow}\sigma$. Then, by definition of $\Rightarrow^1_{R/\mathcal{E}}$, $u_1\sigma \Rightarrow^1_{R/\mathcal{E}} t\sigma$. By I.H. $t\sigma \Rightarrow_{R/\mathcal{E}} v_1\sigma$, so $u_1\sigma \Rightarrow_{R/\mathcal{E}} v_1\sigma$. Also by I.H. $u_i\sigma \Rightarrow_{R/\mathcal{E}} v_i\sigma$, for $2 \leq i \leq n$, and σ is a solution for G', so σ is a solution for G.

□

Lemma 8. *Given an FPP executable* MEL *theory* $(\Sigma, E \cup B)$ *and its associated rewrite theory* $\mathcal{R}_E = (\Sigma', B, R_E)$*, a substitution* σ*, with* $dom(\sigma) = vars(t) \cup vars(t')$ *(i.e., all variables are at least renamed), is an idempotent* B*-unifier of two terms* t, t' *in* $T_\Sigma(\mathcal{X})$ *if and only if* $\sigma =_B \rho\gamma\gamma'$*, where* $kinded(t, t') = (C, \rho)$*,* γ *is an idempotent* B*-unifier of* $t\rho$ *and* $t'\rho$*, and* γ' *is an* $\mathcal{E}$*-solution for the system of sentences* $C\gamma$*, where all substitutions are always away from all the variables that have previously appeared.*

Proof. We prove each implication separately.

⇒)

The proof for the case where $ls(t) \in OS(S)$ and $ls(t') \in OS(S)$ is trivial because no memberships are involved in the unification of t and t', C is empty, $\rho = id$, $\gamma = \sigma$ and $\gamma' = id$.

Otherwise, if σ, with $dom(\sigma) = \{x^1_{s_1}, \ldots, x^n_{s_n}, z^1_{k_1}, \ldots, z^m_{k_m}\}$ such that s_i in S $(1 \leq i \leq n)$ and k_j in K $(1 \leq j \leq m)$, is an idempotent B-unifier of t and t' then, by construction, for each $x^i_{s_i}$, with $1 \leq i \leq n$, there is a fresh variable $y^i_{[s_i]}$ such that $\rho = \{x^i_{s_i} \mapsto y^i_{[s_i]}\}^n_{i=1}$ and $C = \bigwedge^n_{i=1} y^i_{[s_i]} : s_i$.

We define $\sigma' = \{y^i_{[s_i]} \mapsto x^i_{s_i}\sigma\}^n_{i=1} \cup \{z^j_{k_j} \mapsto z^j_{k_j}\sigma\}^m_{j=1}$. By construction $\sigma = \rho\sigma'$. σ' is idempotent because each $y^i_{[s_i]}$, $1 \leq i \leq n$, is a fresh variable that doesn't appear anywhere else and σ is idempotent. As $t\sigma =_B t'\sigma$ then $(t\rho)\sigma' =_B (t'\rho)\sigma'$, so σ' is a B-unifier for $(t\rho)$ and $(t'\rho)$. Then, there is a substitution $\gamma \in CSU_B(t\rho, t'\rho)$, so $t\rho\gamma =_B t'\rho\gamma$, such that $\sigma' \ll_B \gamma$, and there exists a substitution γ' such that $\sigma' =_B \gamma\gamma'$, so $\sigma =_B \rho\gamma\gamma'$. For each condition $y^i_{[s_i]} : s_i$ in C, $1 \leq i \leq n$, we have that $x^i_{s_i}\rho = y^i_{[s_i]}$, and $x^i_{s_i}\rho\gamma\gamma' =_B x^i_{s_i}\sigma$, so $y^i_{[s_i]}\gamma\gamma' =_B x^i_{s_i}\sigma$. As $x^i_{s_i}\sigma$ has sort s_i because σ is well-formed, then γ' is a solution for $y^i_{[s_i]}\gamma : s_i$, so γ' is an $\mathcal{E}$-solution for the system of sentences $C\gamma$.

⇐)

If $\rho = \{x^i_{s_i} \mapsto y^i_{[s_i]}\}^n_{i=1}$, $C = \bigwedge^n_{i=1} y^i_{[s_i]} : s_i$, γ is an idempotent B-unifier of $t\rho$ and $t'\rho$, and γ' is an $\mathcal{E}$-solution for $C\gamma$, we call $\sigma' = \gamma\gamma'$ and $\sigma = \rho\sigma'$, so $t\sigma =_B t'\sigma$. Now we prove that σ is well-formed. The sorted variables in $vars(t) \cup vars(t')$ are $\{x^i_{s_i}\}^n_{i=1}$. We have that $x^i_{s_i}\sigma = y^i_{[s_i]}\sigma'$,$1 \leq i \leq n$, and γ' is an $\mathcal{E}$-solution for $y^i_{[s_i]}\gamma : s_i$, so $y^i_{[s_i]}\gamma\gamma' : s_i$, i.e, $x^i_{s_i}\sigma : s_i$. σ is idempotent because σ' is away from $vars(t) \cup vars(t') \cup vars(C)$. □

Proposition 3. *Given an FPP executable* MEL *theory* $(\Sigma, E \cup B)$ *and its associated rewrite theory* $\mathcal{R}_E = (\Sigma', B, R_E)$*, after applying the* transitivity *rule followed by zero or more applications of the* congruence *rule to a unification problem of the form* $t \to t'$ *we get another unification problem of the form* $t|_p \to^1 x_k, t[x_k]_p \to t'$ *with* k *some kind in* Σ'*.*

Proof. Immediate, by induction on the number of congruence rules applied.

- Base case: zero congruence rules. Then $t \to t' \rightsquigarrow_{[t]} t \to^1 x_k, x_k \to t'$, with $k = [ls(t)]$. In this case $p = \epsilon$ and $t[x_k]_\epsilon \equiv x_k$.
- Induction case. We assume that after applying the congruence rule zero or more times we have: $t \to t' \rightsquigarrow_{[t]} \rightsquigarrow^*_{[c]} (t|_p \to^1 x_k, t[x_k]_p \to t')$. Then, if we apply the congruence rule again, by definition of the rule we get the unification problem $t|_{p.i} \to^1 y_{k'}, t[y_{k'}]_{p.i} \to t'$.

□

Theorem 7 (Soundness of the Calculus for $\mathcal{E}$-solutions)**.** *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, its FPP executable* MEL *theory $(\Sigma, E \cup B)$, the associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, a system of sentences, and its associated unification goal G, if σ is a computed answer for $G{\downarrow}$ then σ is an idempotent E, B-normalized $\mathcal{E}$-solution for G.*

Proof. By Lemma 6, we only have to prove that σ is an $\mathcal{E}$-solution for $G{\downarrow}$. By construction of σ all computed answers are idempotent E, B-normalized. Now, we prove that σ is an $\mathcal{E}$-solution for each unification subproblem generated by narrowing from an initial unification subproblem $u{\downarrow} \to v$, by induction on the total number of narrowing steps. We prove that if σ is a computed answer for $t \to t'$, or $t|_p \to^1 x^p_{k_p}, t[x^p_{k_p}]_p \to t'$ (where t and all of its subterms are always canonical forms by definition of the calculus, so $t|_p{\downarrow} \equiv t|_p$), then $t\sigma \to_{E,B} t'\sigma$ so σ is an $\mathcal{E}$-solution for $t \to t'$.

Base case, one narrowing step:

- Elimination rule $[e]$. There are two subcases:
  - $\mathtt{tt} \to \mathtt{tt}$. Trivial with $\sigma = id$.
  - $t \to t'$ and $t\sigma =_B t'\sigma$, so $t \to t' \rightsquigarrow_{[e],\sigma} \square$. Then, by definition of $\to_{E,B}$, $t\sigma \to_{E,B} t'\sigma$.

Induction case:

- Transitivity rule $[t]$. $t \to t' \rightsquigarrow_{[t]} t \to^1 x_k, x_k \to t' \rightsquigarrow^*_\sigma \square$. By I.H. $t\sigma \to_{E,B} t'\sigma$.
- Reduction rule [r]. The unification subproblem has form $t|_p \to^1 x_k, t[x_k]_p \to t'$, with x_k fresh variable. We apply rule $[r]$ because there is a rule $c \equiv l \to r$ *if* $\bigwedge_{i=1}^n t_i \to t'_i$ in R_E and there is an idempotent substitution θ, with $dom(\theta) \subseteq vars(t|_p) \cup vars(l)$ and $\theta_{vars(t)}$ E, B-normalized, such that $t\theta =_B l\theta$, and also $dom(\theta) \cap vars(t'_i) = \emptyset$ because c has fresh variables and t'_i is $\mathtt{tt}$ or an FPP Σ-pattern.

  Then, the narrowing derivation is $t|_p \to^1 x_k, t[x_k]_p \to t' \rightsquigarrow_{[r],\rho,\theta} \bigwedge_{i=1}^n (t_i\theta){\downarrow} \to t'_i \wedge (t[r]_p\theta){\downarrow} \to t' \rightsquigarrow^*_{\sigma'} \square$, $(t'\theta){\downarrow} \equiv t'$ because $dom(\theta) \cap vars(t') = \emptyset$, and t' is $\mathtt{tt}$ or an FPP Σ-pattern, with σ' E, B-normalized with respect to all variables in $\bigwedge_{i=1}^n (t_i\theta){\downarrow} \to t'_i \wedge (t[r]_p\theta){\downarrow} \to t'$. Then $\sigma = \theta\sigma'$.

  For $1 \leq i \leq n$, $t_i\theta \to_{E,B} (t_i\theta){\downarrow}$, so $(t_i\theta)\sigma' \to_{E,B} (t_i\theta){\downarrow}\sigma'$. By I.H. $(t_i\theta){\downarrow}\sigma' \to_{E,B} t'_i\sigma'$, so $t_i\sigma \to_{E,B} t'_i\sigma'$, and also $t_i\sigma \to_{E,B} t'_i\sigma$ because $dom(\theta) \cap vars(t'_i) = \emptyset$, so $t'_i\theta\sigma' \equiv t'_i\sigma'$. As $t|_p\theta =_B l\theta$ then $t|_p\sigma =_B l\sigma$ so $t|_p\sigma \to^1_{E,B} r\sigma$. Then, by definition of $\to^1_{E,B}$, $t[t|_p\sigma]_p \to^1_{E,B} t[r\sigma]_p$, so $t[t|_p\sigma]_p\sigma \to^1_{E,B} t[r\sigma]_p\sigma$ which, as σ is idempotent and $t[t|_p]_p \equiv t$, is equivalent to $t\sigma \to^1_{E,B} t[r]_p\sigma$.

  $t[r]_p\theta \to_{E,B} (t[r]_p\theta){\downarrow}$, so $t[r]_p\theta\sigma' \to_{E,B} (t[r]_p\theta){\downarrow}\sigma'$. As by I.H. $(t[r]_p\theta){\downarrow}\sigma' \to_{E,B} t'\sigma'$, then $t[r]_p\theta\sigma' \to_{E,B} t'\sigma'$, i.e., $t[r]_p\sigma \to_{E,B} t'\sigma$. As $t\sigma \to^1_{E,B} t[r]_p\sigma$, then $t\sigma \to_{E,B} t'\sigma$.

- Congruence rule $[c]$. By I.H. $t\sigma \to_{E,B} t'\sigma$.

  As $t|_{p.i}\sigma \to^1_{E,B} y_{k'}\sigma$, then $t[t|_{p.i}\sigma]_{p.i} \to^1_{E,B} t[y_{k'}\sigma]_{p.i}$ and $t[t|_{p.i}\sigma]_{p.i}\sigma \to^1_{E,B} t[y_{k'}\sigma]_{p.i}\sigma$ which, as σ is idempotent and $t[t|_{p.i}]_{p.i} \equiv t$, is equivalent to $t\sigma \to^1_{E,B} t[y_{k'}]_{p.i}\sigma$. Again, as $t[y_{k'}]_{p.i}\sigma \to_{E,B} t'\sigma$, then $t\sigma \to_{E,B} t'\sigma$.

□

Theorem 8 (Weak Completeness of the Calculus for $\mathcal{E}$-solutions)**.** *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, its FPP executable* MEL *theory $(\Sigma, E \cup B)$, the associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, a system of sentences F, and its associated unification goal G, if σ is an idempotent E, B-normalized $\mathcal{E}$-solution for G then there is an idempotent E, B-normalized substitution γ, with $\sigma \ll_B \gamma_{vars(G)}$, such that $G{\downarrow} \rightsquigarrow^*_\gamma \square$.*

Proof. Every computed answer γ is idempotent E, B-normalized by definition of the calculus. If σ is an $\mathcal{E}$-solution for G then, by Lemma 6, for each unification subgoal $t \to t'$, $t{\downarrow}\sigma \to_N t'\sigma$. We prove the theorem using induction on the number of unification subgoals plus the number of $\to^1_N$ rewrite steps, including the subgoals and rewrite steps due to conditions.

Base case. One subgoal, zero rewrite steps. There are several cases:

- $F \equiv t = t'$, and $eq(t, t'){\downarrow}\sigma \to_N \mathtt{tt}$ because $t{\downarrow}\sigma =_B t'{\downarrow}\sigma$. There are two subcases:
  - $t{\downarrow} =_B t'{\downarrow}$. Then $G{\downarrow} \equiv \mathtt{tt} \to \mathtt{tt} \rightsquigarrow_{[e],id} \square$, and trivially $\sigma \ll_B id$.
  - $t{\downarrow} \neq_B t'{\downarrow}$. Then $G{\downarrow} \equiv eq(t{\downarrow}, t'{\downarrow}) \to \mathtt{tt}$, and there exists $\gamma \in CSU_B(t{\downarrow} = t'{\downarrow})$ such that $\sigma \ll_B \gamma$, so $t{\downarrow}\gamma =_B t'{\downarrow}\gamma$.
    Then $G{\downarrow} \equiv eq(t{\downarrow}, t'{\downarrow}) \to \mathtt{tt} \rightsquigarrow_{[t]} eq(t{\downarrow}, t'{\downarrow}) \to^1 x_k, x_k \to \mathtt{tt} \rightsquigarrow_{[r],eq(y_{[k]},y_{[k]})\to\mathtt{tt},\gamma'} \mathtt{tt} \to \mathtt{tt} \rightsquigarrow_{[e,id]} \square$, where $\gamma' = \gamma \cup \{y_{[k]} \mapsto t\gamma\}$.
- $F \equiv t := t'$, and $t'{\downarrow}\sigma \to_N t\sigma$ because $t{\downarrow}\sigma =_B t{\downarrow}\sigma$ ($t'{\downarrow} \equiv t$ because t is a Σ-pattern). Then there exists $\gamma \in CSU_B(t'{\downarrow} = t)$ such that $\sigma \ll_B \gamma$, so $G{\downarrow} \equiv t'{\downarrow} \to t \rightsquigarrow_{[e],\gamma} \square$.
- $F \equiv t : s$, and $t{:}s{\downarrow}\sigma \to_N \mathtt{tt}$ because $t{:}s{\downarrow} \equiv \mathtt{tt}$ (i.e., t and $t{\downarrow}$ have sort s). Then, again, $G{\downarrow} \equiv \mathtt{tt} \to \mathtt{tt} \rightsquigarrow_{[e],id} \square$, and trivially $\sigma \ll_B id$.

Induction case. We consider two subcases:

- Several subgoals in the initial problem (there may be zero $\to^1_N$ rewrite steps): $G \equiv t \to t' \wedge G'$. As σ is an $\mathcal{E}$-solution for $t \to t'$, which has at most the same number of rewrite steps and one less subgoal than G, so I.H. applies and there exists an idempotent E, B-normalized substitution γ such that $\sigma \ll_B \gamma_{vars(t\to t')}$, so $\sigma = \gamma_{vars(t\to t')}\rho$ for some idempotent E, B-normalized substitution ρ, such that $t \to t' \rightsquigarrow^*_\gamma \square$. Then $t \to t' \wedge G' \rightsquigarrow^*_\gamma (G'\gamma){\downarrow}$.

  As $\sigma = \gamma_{vars(t\to t')}\rho$ and $vars(G') \cap dom(\gamma) \subseteq vars(t \to t')$ then $G'\gamma \equiv G'\gamma_{vars(t\to t')}$, so ρ is an $\mathcal{E}$-solution for $G'\gamma$ because $\sigma = \gamma_{vars(t\to t')}\rho$. Then I.H applies to $(G'\gamma){\downarrow}$, which has at most the same number of rewrite steps and one less subgoal than G, and there exists an idempotent E, B-normalized substitution θ such that $\rho \ll_B \theta_{vars(G'\gamma))\downarrow}$ and $(G'\gamma){\downarrow} \rightsquigarrow^*_\theta \square$.

  $vars((G'\gamma){\downarrow}) \subseteq vars(G\gamma)$ $dom(\theta) \cap vars(G\gamma) \subseteq vars((G'\gamma){\downarrow})$, so $\theta_{vars((G'\gamma)\downarrow)} = \theta_{vars(G\gamma)}$, and $\rho \ll_B \theta_{vars(G\gamma)}$. Let $v = vars(G)$. As $v \cap dom(\gamma) \subseteq vars(t \to t')$ then $\gamma_{vars(t\to t')} = \gamma_v$, and $\sigma = \gamma_v\rho$. Recall that $dom(\theta_{vars(G\gamma)}) \subseteq Ran(\gamma_v) \cup v$. Then $\sigma = \gamma_v\rho \ll_B \gamma_v\theta_{vars(G\gamma)} = \gamma_v(\theta_{Ran(\gamma_v)} \cup \theta_v) = (\gamma\theta)_v$.

- One subgoal in the initial problem and at least one $\rightarrow_N^1$ rewrite step: $G \equiv t \rightarrow t'$, $t{\downarrow}\sigma \rightarrow_N^1 t'' \rightarrow_N t'\sigma$, σ is an N-solution for $G{\downarrow}$, and $t'\sigma$ is a canonical form.

  We check each type of rule that can have been applied in $t{\downarrow}\sigma \rightarrow_N^1 t''$:

  1. $c \equiv eq(x_k, x_k) \rightarrow \mathtt{tt}$, so $t \equiv eq(t_1, t_2)$, with $t_1{\downarrow} \neq_B t_2{\downarrow}$ (else $t{\downarrow} \equiv \mathtt{tt}$ and there would not be any $\rightarrow_N^1$ step because $\mathtt{tt}$ is a canonical form), $t_1{\downarrow}\sigma =_B t_2{\downarrow}\sigma$, $t' \equiv \mathtt{tt}$, and $eq(t_1{\downarrow}, t_2{\downarrow}) \rightarrow_N^1 \mathtt{tt} \rightarrow_N \mathtt{tt}$. Then there exists $\gamma \in CSU_B(t_1{\downarrow} = t_2{\downarrow})$ such that $\sigma \ll_B \gamma$, so $eq(t_1{\downarrow}, t_2{\downarrow}) \rightarrow \mathtt{tt} \rightsquigarrow_{[t]} eq(t_1{\downarrow}, t_2{\downarrow}) \rightarrow^1 y_k, y_k \rightarrow \mathtt{tt} \rightsquigarrow_{[r],c,\gamma'} \mathtt{tt} \rightarrow \mathtt{tt} \rightsquigarrow_{[e]} \Box$, where $\gamma' = \gamma \cup \{y_{[k]} \mapsto t\gamma\}$.

  2. $c \equiv l{:}s \rightarrow \mathtt{tt}$ *if* C, so $t \equiv t_1{:}s$, with $ls(t_1{\downarrow}) \not\leqslant s$ (else $t{\downarrow} \equiv \mathtt{tt}$), $t_1{\downarrow}\sigma =_B l\sigma'_c$, σ'_c is an idempotent $\mathcal{E}$-solution for C (E, B-normalized with respect to $Extra(C)$), $t' \equiv \mathtt{tt}$, $t'' \equiv \mathtt{tt}$, and $t_1{\downarrow}\sigma{:}s \rightarrow_N^1 \mathtt{tt} \rightarrow_N \mathtt{tt}$. $dom(\sigma) \cap dom(\sigma'_c) = \emptyset$, so $\sigma \cup \sigma'_c$ is a B-unifier for $t_1{\downarrow} = l$. Let $v = vars(t) = vars(G)$, and $w = vars(l)$. Then there exists $\gamma \equiv \gamma_v \cup \gamma_w \in CSU_B(t_1{\downarrow} = l)$ such that $t_1\gamma_v =_B l\gamma_w$, and $\sigma \cup \sigma'_c \ll_B \gamma$, so $\sigma \cup \sigma'_c =_B \gamma\rho$ for some idempotent substitution ρ. σ is E, B-normalized, σ'_c is E, B-normalized except maybe for some subset of $dom(\gamma)$, so ρ must be E, B-normalized. Then $t_1{:}s \rightarrow \mathtt{tt} \rightsquigarrow_{[t]} t_1{:}s \rightarrow^1 x_k, x_k \rightarrow \mathtt{tt} \rightsquigarrow_{[r],\gamma} (C\gamma){\downarrow}$. As C has fresh variables then $dom(\sigma) \cap vars(C) = \emptyset$, so $C\sigma'_c \equiv C(\sigma \cup \sigma'_c) \equiv C(\gamma\rho)$.

     σ'_c is an $\mathcal{E}$-solution for C with less than n rewrite steps, so ρ is an idempotent E, B-normalized $\mathcal{E}$-solution for $C\gamma$ with less than n rewrite steps, with $(C\gamma){\downarrow} \equiv (C\gamma_w){\downarrow}$ because $dom(\gamma_v) \cap vars(C) = \emptyset$. By I.H. there exists θ, with $\rho \ll_B \theta_{vars(C\gamma)}$ such that $(C\gamma){\downarrow} \rightsquigarrow_\theta^* \Box$. The composition of the substitutions in the narrowing derivation is $\gamma\theta$. We have to prove that $\sigma \ll_B (\gamma\theta)_v$. As $vars(C\gamma) \cap v = \emptyset$ then $dom(\theta) \cap v = \emptyset$. As $\sigma \cup \sigma'_c =_B \gamma\rho$, $dom(\sigma'_c) \cap v = \emptyset$, and $dom(\sigma) \subseteq v$, then $\sigma = (\gamma_v\rho_{Ran(\gamma_v)}) \cup \rho_v$.

     $(C\gamma){\downarrow} \rightsquigarrow_\theta^* \Box$ so $dom(\theta) \subseteq vars(C\gamma) \cup v'$, with v' a set of fresh variables generated by the narrowing calculus, and $Ran(\gamma) = Ran(\gamma_v) = Ran(\gamma_w)$ because $\gamma \in CSU_B(t_1{\downarrow} = l)$ and B is regular. Then, $dom(\theta) \cap Ran(\gamma) = dom(\theta) \cap Ran(\gamma_v) \subseteq vars(C\gamma)$, and $\theta_{vars(C\gamma)} \ll_B \theta_{Ran(\gamma)}$. As $\rho \ll_B \theta_{vars(C\gamma)}$, then $\rho \ll_B \theta_{Ran(\gamma_v)}$, and $\rho_{Ran(\gamma_v)} \ll_B \theta_{Ran(\gamma_v)}$.

     Now, $\gamma_v\rho_{Ran(\gamma_v)} \ll_B \gamma_v\theta_{Ran(\gamma_v)} = \gamma_v\theta_{Ran(\gamma_v)} \cup \theta_v$ because $dom(\theta) \cap v = \emptyset$, so $\theta_v = id$. But $\gamma_v\theta_{Ran(\gamma_v)} \cup \theta_v = (\gamma\theta)_v$, so $\gamma_v\rho_{Ran(\gamma_v)} \ll_B (\gamma\theta)_v$.

     In conclusion: $\sigma = \gamma_v\rho_{Ran(\gamma_v)} \cup \rho_v \ll_B (\gamma\theta)_v$.

  3. $c \equiv l \rightarrow r$ *if* C, not in cases 1 or 2. Then $t' \neq \mathtt{tt}$, $t'' \neq \mathtt{tt}$, and $t{\downarrow}\sigma \rightarrow_N^1 (t{\downarrow}\sigma)[r\sigma'_c]_p \rightarrow_N t'\sigma$ because $(t{\downarrow}\sigma)|_p =_B l\sigma'_c$ and σ'_c is an $\mathcal{E}$-solution for C (E, B-normalized with respect to $Extra(C)$). As σ is E, B normalized and $l \notin \mathcal{X}$, then we cannot rewrite inside a position instantiated by σ or a variable position, so p must be an already existing non-variable position in $t{\downarrow}$ (i.e., $p \in Pos_\Sigma(t{\downarrow})$). Also $(t{\downarrow}\sigma)[r\sigma'_c]_p \equiv t{\downarrow}[r\sigma'_c]_p\sigma$ because $dom(\sigma) \cap vars(r\sigma'_c) = \emptyset$, and $t{\downarrow}[r\sigma'_c]_p\sigma \equiv t{\downarrow}[r]_p(\sigma \cup \sigma'_c)$ because $dom(\sigma'_c) \cap vars(t{\downarrow}) = \emptyset$ and $dom(\sigma) \cap dom(\sigma'_c)) = \emptyset$.

     As in the previous subcase, $t{\downarrow}\sigma \rightarrow_N^1 t{\downarrow}[r]_p(\sigma \cup \sigma'_c) \rightarrow_N t'\sigma$, and there exists $\gamma = \gamma_v \cup \gamma_w \in CSU_B(t{\downarrow}|_p = l)$, with $v = vars(G)$, and $w = vars(c)$, such that $t{\downarrow}|_p\gamma_v =_B l\gamma_w$. Then there exists $\gamma \equiv \gamma_v \cup \gamma_w \in CSU_B(t{\downarrow}|_p{\downarrow} = l)$ such that $t{\downarrow}|_p\gamma_v =_B l\gamma_w$, and $\sigma \cup \sigma'_c \ll_B \gamma$, so $\sigma \cup \sigma'_c =_B \gamma\rho$ for some idempotent substitution ρ. σ is E, B-normalized, σ'_c is E, B-normalized except maybe for some subset of $dom(\gamma)$, so ρ must be E, B-normalized.

Although t' is an FPP Σ-pattern, we reason in this part of the proof as if t' could be any term, for compatibility with the equivalent proof for the calculus for reachability, obtaining then a more general result.

Then $t{\downarrow} \to t'{\downarrow} \rightsquigarrow_{[t]} t{\downarrow} \to^1 x_k, x_k \to t'{\downarrow} \rightsquigarrow^*_{[c]} t{\downarrow}|_p \to^1 y_{k'}, t{\downarrow}[y_{k'}]_p \to t'{\downarrow} \rightsquigarrow_{[r],\gamma} (C\gamma){\downarrow} \wedge (t{\downarrow}[r]_p\gamma){\downarrow} \to (t'{\downarrow}\gamma){\downarrow}$ (recall that $(t'{\downarrow}\gamma){\downarrow} \equiv t'$). Let $u = vars(C\gamma \wedge t{\downarrow}[r]_p\gamma)$. ρ is an $\mathcal{E}$-solution for $C\gamma \wedge t{\downarrow}[r]_p\gamma \to (t'{\downarrow}\gamma){\downarrow}$ with one less subgoal and one less rewriting step than the $\mathcal{E}$-solution σ for $t \to t'$, so I.H. applies and there exist an E, B-normalized substitution θ such that $\rho \ll_B \theta_u$ and $(C\gamma){\downarrow} \wedge (t{\downarrow}[r]_p\gamma){\downarrow} \to (t'{\downarrow}\gamma){\downarrow} \rightsquigarrow^*_\theta \Box$.

As $\rho \ll_B \theta_u$ then $\rho \ll_B \theta_{vars(C\gamma)}$, so $\rho_{vars(C\gamma)} \ll_B \theta_{vars(C\gamma)}$. $\sigma \cup \sigma'_c =_B \gamma\rho$, so $\sigma = (\sigma \cup \sigma'_c)_v = (\gamma\rho)_v$. Then we have $\sigma = \gamma_v\rho_{Ran(\gamma_v)} \cup \rho_v$. As $dom(\gamma_v) \cap dom(\rho_v) = \emptyset$ then $\gamma_v\rho_{Ran(\gamma_v)} \cup \rho_v = \gamma_v(\rho_{Ran(\gamma_v)} \cup \rho_v)$, and $\sigma = \gamma_v(\rho_{Ran(\gamma_v)} \cup \rho_v) = \gamma_v\rho_{vars(C\gamma)} \ll_B \gamma_v\theta_{vars(C\gamma)} = \gamma_v(\theta_{Ran(\gamma_v)}) \cup \theta_v)$. As $dom(\gamma_v) \cap dom(\theta_v) = \emptyset$, then $\gamma_v(\rho_{Ran(\gamma_v)} \cup \rho_v) = \gamma_v\rho_{Ran(\gamma_v)} \cup \rho_v$, so $\sigma \ll_B \gamma_v\rho_{Ran(\gamma_v)} \cup \rho_v = (\gamma\rho)_v$.

□

Theorem 9 (Soundness of the Calculus for Reachability)**.** *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$, its FPP executable* Mel *theory $(\Sigma, E \cup B)$, the associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, and a reachability goal G, if σ is a computed answer for $G{\downarrow}$, using the transformed set of rules $\tilde{R}$, then σ is a solution for G.*

Proof. We prove that given a reachability problem $G \equiv g(\wedge\, G')$, if $G{\downarrow} \rightsquigarrow^*_\sigma \Box$ then σ is a solution for G in $\to_{R/\mathcal{E}}$. In particular, we prove that if $g \equiv t \dashrightarrow^1 x_k, x_k \Rightarrow t' \rightsquigarrow^*_\sigma \Box$, where $\dashrightarrow$ can be either $\to$ or $\Rightarrow$, then σ is a solution for $t \Rightarrow t'$ in $\to_{R/\mathcal{E}}$. As $\mathcal{R}$ is narrowable, and by Lemma 7, it is enough to prove that σ is an N-solution for $G{\downarrow}$. Soundness of the calculus for reachability is proved by induction on the total number of narrowing steps for each unification subproblem generated by narrowing from $G{\downarrow}$. By our previous proof of soundness in Theorem 7, we know that if we compute a solution σ for $t \to t'$ then $t\sigma \to_N t'\sigma$.

Base case: one narrowing step. The calculus rules in Figure 4.5 cannot compute a solution in one narrowing step, so we are in one of the base cases already proved for Theorem 7, with some unification goal $G{\downarrow} \equiv t \to t'$, so σ is an $\mathcal{E}$-solution for $G{\downarrow}$, hence a solution for $G{\downarrow}$ in $\to_{ER,B}$.

Induction case: The cases where the first rule applied to $g{\downarrow}$ is shown in Figure 4.4 have already been proved for unification goals in Theorem 7. The same proof is valid for reachability goals *mutatis mutandis*, so we only check the cases where the first rule applied to $g{\downarrow}$ is one of the rules in Figure 4.5.

- Reflexivity rule: any computed answer σ is a solution for $t = t'$ and $G'{\downarrow}$. Then, as seen in the base case, σ is a solution for $g \equiv t \Rightarrow t'$ and, by I.H., σ is also a solution for $G'{\downarrow}$, so σ is a solution for $G{\downarrow}$ in $\to_{ER,B}$. We skip the part of the proof related to G' in the rest of cases, as it is always the same.

- Transitivity rule: by I.H. σ is a solution for $t \Rightarrow t'$ in $\to_{ER,B}$.

- Congruence rule: by I.H. σ is a solution for $t \Rightarrow t'$ in $\to_{ER,B}$.

- Reduction rule: the reachability subproblem has form $t|_p \to^1 x_k, t[x_k]_p \Rightarrow t'$, with x_k a fresh variable. We apply rule $[r]$ because there is a rule $c \equiv l \to r$ *if* $\bigwedge_{i=1}^n t_i \to t'_i$

in R_E and there is an idempotent substitution θ, with $dom(\theta) \subseteq vars(t|_p) \cup vars(l)$ and $\theta_{vars(t)}$ E,B-normalized, such that $t\theta =_B l\theta$, and also $dom(\theta) \cap vars(t'_i) = \emptyset$ because c has fresh variables and t'_i is $\mathtt{tt}$ or an FPP Σ-pattern.

Then, the narrowing derivation is $t|_p \rightarrow^1 x_k, t[x_k]_p \Rightarrow t' \rightsquigarrow_{[n],c,\theta} \bigwedge_{i=1}^n (t_i\theta)\downarrow \rightarrow t'_i \wedge (t[r]_p\theta)\downarrow \Rightarrow (t'\theta)\downarrow \rightsquigarrow^*_{\sigma'} \square$, with σ' E,B-normalized with respect to all variables in $\bigwedge_{i=1}^n (t_i\theta)\downarrow \rightarrow t'_i \wedge (t[r]_p\theta)\downarrow \Rightarrow (t'\theta)\downarrow$. Then $\sigma = \theta\sigma'$.

For $1 \leq i \leq n$, $t_i\theta \rightarrow_{E,B} (t_i\theta)\downarrow$, so $(t_i\theta)\sigma' \rightarrow_{E,B} (t_i\theta)\downarrow\sigma'$. By I.H. $(t_i\theta)\downarrow\sigma' \rightarrow_{E,B} t'_i\sigma'$, so $t_i\sigma \rightarrow_{E,B} t'_i\sigma'$, and also $t_i\sigma \rightarrow_{E,B} t'_i\sigma$ because $dom(\theta) \cap vars(t'_i) = \emptyset$, so $t'_i\theta\sigma' \equiv t'_i\sigma'$. As $t|_p\theta =_B l\theta$ then $t|_p\sigma =_B l\sigma$ so $t|_p\sigma \rightarrow^1_{E,B} r\sigma$. Then, by definition of $\rightarrow^1_{E,B}$, $t[t|_p\sigma]_p \rightarrow^1_{E,B} t[r\sigma]_p$, so $t[t|_p\sigma]_p\sigma \rightarrow^1_{E,B} t[r\sigma]_p\sigma$ which, as σ is idempotent and $t[t|_p]_p \equiv t$, is equivalent to $t\sigma \rightarrow^1_{E,B} t[r]_p\sigma$.

$t[r]_p\theta \rightarrow_{E,B} (t[r]_p\theta)\downarrow$. The $t[r]_p\theta\sigma' \rightarrow_{E,B} (t[r]_p\theta)\downarrow\sigma'$ and $t[r]_p\theta\sigma' \rightarrow^!_{E,B} ((t[r]_p\theta)\downarrow\sigma')\downarrow$. By I.H. $(t[r]_p\theta)\downarrow\sigma' \rightarrow_{ER,B} (t'\theta)\downarrow\sigma'$, so also $((t[r]_p\theta)\downarrow\sigma')\downarrow \rightarrow_{ER,B} (t'\theta)\downarrow\sigma'$, hence $t[r]_p\theta\sigma' \rightarrow_{ER,B} (t'\theta)\downarrow\sigma'$.

- Rewrite rule: the reachability subproblem has form $t|_p \Rightarrow^1 x_k, t[x_k]_p \Rightarrow t'$, with x_k a fresh variable. We apply rule $[r]$ because there is a rule $c \equiv l \Rightarrow r$ *if* $\bigwedge_{i=1}^n t_i \dashrightarrow t'_i$ in $\tilde{R}$ (where $\dashrightarrow$ can be either $\rightarrow$ or $\Rightarrow$) and there is an idempotent substitution θ, with $dom(\theta) \subseteq vars(t|_p) \cup vars(l)$ and $\theta_{vars(t)}$ E,B-normalized, such that $t\theta =_B l\theta$.

  Then, the narrowing derivation is $t|_p \Rightarrow^1 x_k, t[x_k]_p \Rightarrow t' \rightsquigarrow_{[w],c,\theta} \bigwedge_{i=1}^n (t_i\theta)\downarrow \dashrightarrow (t'_i\theta)\downarrow \wedge (t[r]_p\theta)\downarrow \Rightarrow (t'\theta)\downarrow \rightsquigarrow^*_{\sigma'} \square$, with σ' E,B-normalized with respect to all variables in $\bigwedge_{i=1}^n (t_i\theta)\downarrow \dashrightarrow (t'_i\theta)\downarrow \wedge (t[r]_p\theta)\downarrow \Rightarrow (t'\theta)\downarrow$. Then $\sigma = \theta\sigma'$.

  By I.H. σ' is an solution of $(t_i\theta)\downarrow \dashrightarrow (t'_i\theta)\downarrow$, for $1 \leq i \leq n$. Then, by Lemma 7, σ' is a solution for $t_i\theta \dashrightarrow t'_i\theta$, so σ is a solution for $t_i \dashrightarrow t'_i$, and $t|_p\sigma \rightarrow^1_{ER,B} r\sigma$.

  Then, by definition of $\rightarrow^1_{ER,B}$, $t[t|_p\sigma]_p \rightarrow^1_{ER,B} t[r\sigma]_p$, so $t[t|_p\sigma]_p\sigma \rightarrow^1_{ER,B} t[r\sigma]_p\sigma$ which, as σ is idempotent and $t[t|_p]_p \equiv t$, is equivalent to $t\sigma \rightarrow^1_{ER,B} t[r]_p\sigma$.

  By I.H. σ' is a solution for $(t[r]_p\theta)\downarrow \Rightarrow (t'\theta)\downarrow$. Then, by Lemma 7, σ' is a solution for $t[r]_p\theta \Rightarrow t'\theta$, so σ is a solution for $t[r]_p \Rightarrow t'$, and $t[r]_p\sigma \rightarrow_{ER,B} t'\sigma$. As $t\sigma \rightarrow^1_{ER,B} t[r]_p\sigma$, then $t\sigma \rightarrow_{ER,B} t'\sigma$.

□

Theorem 10 (Weak Completeness of the Calculus for Reachability)**.** *Given a narrowable rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$, its FPP executable* Mel *theory $(\Sigma, E \cup B)$, the associated rewrite theory $\mathcal{R}_E = (\Sigma', B, R_E)$, and a reachability goal G, if σ is an idempotent ER,B-normalized solution for G then there is an idempotent E,B-normalized substitution γ, with $\sigma \ll_B \gamma_{vars(G)}$, such that $G\downarrow \rightsquigarrow^*_\gamma \square$ using the transformed set of rules $\tilde{R}$.*

Proof. Every computed answer γ is idempotent E,B-normalized by definition of the calculus. We prove the theorem using induction on the number of reachability subgoals plus the number of $\Rightarrow^1_N$ rewrite steps, including the subgoals and rewrite steps due to conditions. The proof for equational subgoals is exactly the same already shown in Theorem 8, so there are only two cases left to prove.

- The first case is the base case with one subgoal and zero $\Rightarrow^1_N$ rewrite steps. Then $t\sigma =_\mathcal{E} t'\sigma$, so σ is a solution for the sentence $t = t'$ and also for the unification goal $eq(t,t') \rightarrow \mathtt{tt}$. There are two subcases.

- If $t{\downarrow} =_{\mathcal{E}} t'{\downarrow}$, then $t{\downarrow} \Rightarrow t'{\downarrow} \rightsquigarrow_{[x]} \mathtt{tt} \rightarrow \mathtt{tt} \rightsquigarrow_{[e]} \square$, and $\gamma = id$, so $\sigma \ll_B \gamma_{vars(G)}$.
- If $t{\downarrow} \neq_{\mathcal{E}} t'{\downarrow}$, then $t{\downarrow} \Rightarrow t'{\downarrow} \rightsquigarrow_{[x]} eq(t{\downarrow}, t'{\downarrow}) \rightarrow \mathtt{tt}$, but we have already proved in Theorem 8 that if σ is an $\mathcal{E}$-solution for this unification problem then there exists a substitution γ, with $\sigma \ll_B \gamma_{vars(G)}$, such that $eq(t{\downarrow}, t'{\downarrow}) \rightarrow \mathtt{tt} \rightsquigarrow^*_{\gamma} \square$ and $\sigma \ll_B \gamma_{vars(G)}$.

- The second case is the induction subcase for one subgoal and at least one $\Rightarrow^1_N$ rewrite step, where we apply a rule $c \equiv l \rightarrow r \; if \; C$ in R to $t\sigma$. As σ is a solution for G then, by Lemma 7, σ is an N-solution for $G{\downarrow}$. Then $G{\downarrow} \equiv t{\downarrow} \Rightarrow t'{\downarrow}$, and $t{\downarrow}\sigma \Rightarrow^1_N (t{\downarrow}\sigma)[r\sigma'_c]_p \Rightarrow_N t'{\downarrow}\sigma$ because $(t{\downarrow}\sigma)|_p =_B l\sigma'_c$ and σ'_c is a solution for C (E, B-normalized with respect to $Extra(c)$). As σ is ER, B normalized and $l \notin \mathcal{X}$, then we cannot rewrite inside a position instantiated by σ or a variable position, so p must be an already existing non-variable position in $t{\downarrow}$ (i.e., $p \in Pos_{\Sigma}(t{\downarrow})$). Also $(t{\downarrow}\sigma)[r\sigma'_c]_p \equiv t{\downarrow}[r\sigma'_c]_p\sigma$ because $dom(\sigma) \cap vars(r\sigma'_c) = \emptyset$, and $t{\downarrow}[r\sigma'_c]_p\sigma \equiv t{\downarrow}[r]_p(\sigma \cup \sigma'_c)$ because $dom(\sigma'_c) \cap vars(t{\downarrow}) = \emptyset$ and $dom(\sigma) \cap dom(\sigma'_c)) = \emptyset$.

  Then, $t{\downarrow}\sigma \Rightarrow^1_N t{\downarrow}[r]_p(\sigma \cup \sigma'_c) \Rightarrow_N t'{\downarrow}\sigma$, and there exists $\gamma = \gamma_v \cup \gamma_w \in CSU_B(t{\downarrow}|_p = l)$, with $v = vars(G)$, and $w = vars(c)$, such that $t{\downarrow}|_p\gamma_v =_B l\gamma_w$. Then there exists $\gamma \equiv \gamma_v \cup \gamma_w \in CSU_B(t{\downarrow}|_p{\downarrow} = l)$ such that $t{\downarrow}|_p\gamma_v =_B l\gamma_w$, and $\sigma \cup \sigma'_c \ll_B \gamma$, so $\sigma \cup \sigma'_c =_B \gamma\rho$ for some idempotent substitution ρ. σ is E, B-normalized, σ'_c is E, B-normalized except maybe for some subset of $dom(\gamma)$, so ρ must be E, B-normalized.

  Then $t{\downarrow} \Rightarrow t'{\downarrow} \rightsquigarrow_{[t]} t{\downarrow} \Rightarrow^1 x_k, x_k \Rightarrow t'{\downarrow} \rightsquigarrow^*_{[c]} t{\downarrow}|_p \Rightarrow^1 y_{k'}, t{\downarrow}[y_{k'}]_p \Rightarrow t'{\downarrow} \rightsquigarrow_{[r],\gamma} (C\gamma){\downarrow} \wedge (t{\downarrow}[r]_p\gamma){\downarrow} \Rightarrow t'{\downarrow}$. Let $u = vars(C\gamma \wedge t{\downarrow}[r]_p\gamma)$. ρ is a solution for $C\gamma \wedge t{\downarrow}[r]_p\gamma \rightarrow (t'{\downarrow}\gamma){\downarrow}$ (recall that $(t'{\downarrow}\gamma){\downarrow} \equiv (t'\gamma){\downarrow}$) with one less subgoal and one less rewriting step than the solution σ for $t \Rightarrow t'{\downarrow}$, so I.H. applies and there exist an E, B-normalized substitution θ such that $\rho \ll_B \theta_u$ and $(C\gamma){\downarrow} \wedge (t{\downarrow}[r]_p\gamma){\downarrow} \Rightarrow (t'\gamma){\downarrow} \rightsquigarrow^*_{\theta} \square$.

  As $\rho \ll_B \theta_u$ then $\rho \ll_B \theta_{vars(C\gamma)}$, so $\rho_{vars(C\gamma)} \ll_B \theta_{vars(C\gamma)}$. $\sigma \cup \sigma'_c =_B \gamma\rho$, so $\sigma = (\sigma \cup \sigma'_c)_v = (\gamma\rho)_v$. Then we have $\sigma = \gamma_v\rho_{Ran(\gamma_v)} \cup \rho_v$. As $dom(\gamma_v) \cap dom(\rho_v) = \emptyset$ then $\gamma_v\rho_{Ran(\gamma_v)} \cup \rho_v = \gamma_v(\rho_{Ran(\gamma_v)} \cup \rho_v)$, and $\sigma = \gamma_v(\rho_{Ran(\gamma_v)} \cup \rho_v) = \gamma_v\rho_{vars(C\gamma)} \ll_B \gamma_v\theta_{vars(C\gamma)} = \gamma_v(\theta_{Ran(\gamma_v)}) \cup \theta_v)$. As $dom(\gamma_v) \cap dom(\theta_v) = \emptyset$, then $\gamma_v(\rho_{Ran(\gamma_v)} \cup \rho_v) = \gamma_v\rho_{Ran(\gamma_v)} \cup \rho_v$, so $\sigma \ll_B \gamma_v\rho_{Ran(\gamma_v)} \cup \rho_v = (\gamma\rho)_v$.

□

Chapter 5

Conditional narrowing modulo SMT plus axioms

In this chapter we consider order-sorted rewrite theories where the conditions that appear in their rules are either rewrite conditions or quantifier-free SMT formulas, with no restriction regarding the variables that appear in these rules or in the reachability problems. As explained in Section 2.2.3, the underlying equational theories of these rewrite theories must be decomposable in $\mathcal{E} = E_0 \cup B$ where E_0 is a subset of the theories handled by SMT solvers, and B is a set of axioms for the algebraic data types not handled by the SMT solvers. Abstracting the SMT subterms of the left hand side of the rules and adding compensating equations in the conditions, as already done in [RMM17], play a significant role now.

The calculus presented here extends the use of SMT solvers in rewriting from [RMM17] to the narrowing environment by allowing: (i) rewrite conditions in the rules and (ii) non-SMT variables in the reachability problems.

The main contribution in this chapter is the development of a sound and weakly complete, i.e., complete with respect to idempotent $R/\mathcal{E}$-normalized answers, narrowing calculus for conditional narrowing modulo $E_0 \cup B$ for the considered rewrite theories. The soundness and weak completeness of the calculus is proved using the complete unification algorithm for B and the assumption that there exists an oracle for E_0, the SMT solver itself, instead of using the whole equational theory $\mathcal{E}$, which is usually required for this kind of proofs. All the results for this chapter, together with their corresponding proofs, can be found in Section 5.6.

Although the narrowing calculus is based on the theoretical existence of a SMT oracle, after finishing the work on it [AMPP17], an alpha version of Maude 3, with an initial support for SMT solvers, was released for testers, so we used it to develop two prototypes, that are discussed in Section 5.5. While in the theoretical part of this chapter we stick to [AMPP17], where the theoretical SMT solver is used as an oracle, in the examples and explanations that follow we refer both to [AMPP17] and to the developed prototypes.

5.1 Toast example

Toast cooking will be used as running example in this chapter. A toast is well-cooked if both sides of the toast have been cooked for exactly five seconds. No overcooking is allowed. Fresh toasts are taken from a toast bag, and they are cooked using a frying pan that can toast up to two toasts simultaneously, toasting one side of each toast. There is a

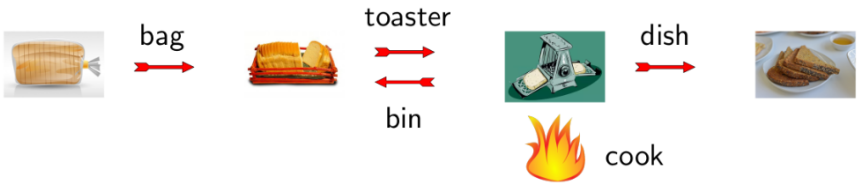

Figure 5.1: Toast cooking

bin, where fresh toasts are put when taken from the bag. A toast can be flipped directly over the pan, or returned to the bin. Finally, there is a dish where well-cooked toasts can be output. The frying pan is a metaphor of a computer resource that allows a maximum number of concurrent users.

A `Toast` (abbreviated to `t`) can be a `RealToast` (`rt`), represented as an ordered pair of natural numbers, each one with sort `Integer` (`i`), storing the seconds that each side has already been toasted, or an `EmptyToast` (`et`) which has a constant `zt`, representing the absence of `Toasts`; a `Pan` (`p`) is an unordered pair of `Toasts`; a `Kitchen` (`k`) has a timer, represented by a natural number, and a `Pan`; a `Bin` (`b`) is a multiset of `Toasts`; the bag and the dish are represented by natural numbers, the number of `RealToasts` in each one; the `System` (`s`) has a bag, a `Bin`, a `Kitchen`, and a dish. When a `RealToast` is in the pan, the side being toasted is represented by the first integer of the ordered pair. There is one auxiliary function, `cook`. The rules for `Toast` cooking are the following:

1. The function call `cook`(x, y) will return the `Kitchen` obtained from `Kitchen` x after y seconds, where y is a positive integer, only if no `RealToast` in the `Pan` gets overcooked.

2. A fresh `RealToast` can pass from a non-empty bag to the `Bin`.

3. A `RealToast` can pass from the `Bin` to the `Pan` if there is room in the `Pan`. The `RealToast` cannot be flipped during this action.

4. A `Kitchen` with at least one `RealToast` in the `Pan` can `cook` the `RealToasts` that are laying on the pan any given integer number of seconds.

5. A `RealToast` in the `Pan` can be flipped over the `Pan`.

6. A `RealToast` in the `Pan` can be returned to the `Bin`, without getting flipped.

7. A well-cooked `RealToast` can be taken out to the dish. This operation takes one second, so if there is another `RealToast` in the `Pan`, it will get `cook`ed for one second.

We show the Maude specification used in our narrowing prototypes. It cannot be used for rewriting due to the existence of non-executable rules, among other things:

```
load smt
mod TOASTS is
  protecting INTEGER .

  sorts RealToast EmptyToast Toast Pan Kitchen Bin System .
  subsort RealToast EmptyToast < Toast < Bin .
```

```
  vars A B C D N OK Y Z : Integer .
  var H : RealToast .
  var V : Toast .
  var T : Bin .
  var K : Kitchen .

  op zt : -> EmptyToast .
  op [_,_] : Integer Integer -> RealToast .
  op __ : Toast Toast -> Pan [comm] .
  op _;_ : Bin Bin -> Bin [comm assoc id: zt] .
  op _;_ : Integer Pan -> Kitchen .
  op cook : Kitchen Integer -> [Kitchen] .
  op _/_/_/_ : Integer Bin Kitchen Integer -> System .

  crl cook(Y ; zt zt, Z) => Y + Z ; zt zt
      if (Z > 0) = (true).Boolean [label r1a] .
  crl cook(Y ; [A, B] zt, Z) => Y + Z ; [A + Z, B] zt
      if ((A >= 0) and (Z > 0) and (Z + A <= 5)) = (true).Boolean
          [label r1b] .
  crl cook(Y ; [A, B] [C, D], Z) => Y + Z ; [A + Z, B] [C + Z, D]
      if ((A >= 0) and (C >= 0) and (Z > 0) and
         (A + Z <= 5) and (C + Z <= 5)) = (true).Boolean [label r1c] .
  crl N / T / K / OK => N - 1 / [0, 0] ; T / K / OK
      if (N > 0) = (true).Boolean [label r2] .
  rl N / H ; T / Y ; zt V / OK => N / T / Y ; H V / OK [label r3] .
  rl Y ; H V => cook(Y ; H V, Z) [label r4 nonexec] .
  rl Y ; [A, B] V => Y ; [B, A] V [label r5] .
  rl N / T / Y ; [A, B] V / OK => N / [A, B] ; T / Y ; zt V / OK
      [label r6] .
  crl N / T / Y ; [5, 5] V / OK => N / T / K / OK + 1
      if cook(Y ; zt V, 1) => K [label r7] .
endm
```

The first command loads the module `smt.maude` that, among other things, defines the sorts `Boolean` (abbreviated to `bool`) and `Integer` that are accepted by the SMT oracle. The command `protecting INTEGER` states that the functional module `INTEGER` from `smt.maude` is used "as is", without any modification, so the SMT oracle can be used to verify the satisfiability of any integer arithmetic formula from $QF(\mathcal{X}_0)$. At the same time, the functional module `INTEGER` contains the command `protecting BOOLEAN`, which is the functional module where the sort `Boolean` is defined.

One point to bear in mind is that the module `smt.maude` has sort and function definitions, but no equations, so any term within its sorts is canonical. This means, for instance, that the functional module `INTEGER` will understand the term `2 + 2` as having sort `Integer`, but it will not reduce it to `4`. What the SMT solver can tell us is that the `Boolean` formula `2 + 2 === 4` from $QF(\mathcal{X}_0)$ is satisfiable, hence valid since the formula is ground. The equality function `op _===_ : Integer Integer -> Boolean` in `INTEGER` has three equality symbols because the `_=_` and `_==_` functions, that usually represent the equality between terms, have a special meaning for the Maude parser.

The sort `System` represents all the elements of the example: bag (`Integer`), `Bin`, `Kitchen`, and dish (`Integer`), separated by slashes, where the `Kitchen` is the pair formed by the elapsed time (`Integer`) and the `Pan`, separated by a semicolon. For design reasons it is desirable to have these two items attached one to another.

The system module `TOASTS` is order-sorted since is has subsorts but no memberships. All of its conditional rules but one ($r7$) have SMT constraints, which we distinguish in Maude by checking them against the condition `true` of sort `Boolean` from `smt.maude`, so that the prototypes can handle them in an appropriate way. Rule $r7$ is the one that has a reachability condition: `cook(Y ; zt V, 1) => K`. When there are two `RealToast`s in the `Pan`, one of them is completely cooked, and we want to put it in the output dish, this condition prevents the `RealToast` that remains in the `Pan`, `V`, from getting overcooked, by checking that the result of cooking it for 1 second matches a variable K with sort `Kitchen` (a subsort of `[Kitchen]`, the codomain of `cook`), that forces the application of rule `r1b`, in this case. If there is only the well-cooked `RealToast` in the `Kitchen`, then rule `r1a` is applied to the condition, incrementing the timer one second, since its instantiated SMT constraint, `1 > 0`, is valid.

5.1.1 Signature

In the toast example, $\Sigma = (S, \leq, F)$ is, omitting the implied kind for each connected component of S:

- $S = \{\mathtt{bool}, \mathtt{i}, \mathtt{rt}, \mathtt{et}, \mathtt{t}, \mathtt{p}, \mathtt{k}, \mathtt{b}, \mathtt{s}\}$
- $\leq = \{(\mathtt{rt}, \mathtt{t}), (\mathtt{et}, \mathtt{t}), (\mathtt{t}, \mathtt{b})\}$
- $F = \{\{[\_, \_]\}_{\mathtt{i}\,\mathtt{i},\mathtt{rt}}, \{\_\_\}_{\mathtt{t}\,\mathtt{t},\mathtt{p}}, \{\_;\_\}_{\mathtt{b}\,\mathtt{b},\mathtt{b}}, \{\_;\_\}_{\mathtt{i}\,\mathtt{p},\mathtt{k}}, \{\mathtt{cook}\}_{\mathtt{k}\,\mathtt{i},[\mathtt{k}]},$
  $\{\_/\_/\_/\_\}_{\mathtt{i}\,\mathtt{b}\,\mathtt{k}\,\mathtt{i},\mathtt{s}}, \{\mathtt{zt}\}_{\mathtt{et}}\}$

5.1.2 Order-sorted theory

The order-sorted theory $(\Sigma, \mathcal{E})$ for the cooking example has $\Sigma = (S, \leq, F)$ and $\mathcal{E} = E_0 \cup B$, where E_0 is the set of equations for integer arithmetic and Boolean calculus (not displayed), and B consists of the four equations:

- $(x_{[\mathtt{b}]}; y_{[\mathtt{b}]}); z_{[\mathtt{b}]} = x_{[\mathtt{b}]}; (y_{[\mathtt{b}]}; z_{[\mathtt{b}]})$
- $x_{[\mathtt{b}]}; y_{[\mathtt{b}]} = y_{[\mathtt{b}]}; x_{[\mathtt{b}]}$
- $x_{[\mathtt{b}]}; \mathtt{zt} = x_{[\mathtt{b}]}$
- $x_{[\mathtt{t}]} y_{[\mathtt{t}]} = y_{[\mathtt{t}]} x_{[\mathtt{t}]}$

stating that the `Bin` is a multiset and that the position of the `Toast`s in the `Pan` is irrelevant.

5.1.3 Rewrite theory

In the toast example, R is the following translation of the rules shown in Section 5.1, where the abbreviations used for the subscripts, as established before, are `rt`−`RealToast`, `t`−`Toast`, `k`−`Kitchen`, `b`−`Bin`, and we assume sort `i`−`Integer` when no subscript is shown:

- $[r1a]: \texttt{cook}(y; \texttt{zt}\ \texttt{zt}, z) \rightarrow y+z; \texttt{zt}\ \texttt{zt}$ *if* $z > 0$
- $[r1b]: \texttt{cook}(y; [a,b]\ \texttt{zt}, z) \rightarrow y+z; [a+z, b]\ \texttt{zt}$ *if* $z > 0 \wedge a+z \leq 5$
- $[r1c]: \texttt{cook}(y; [a,b]\ [c,d], z) \rightarrow y+z; [a+z, b]\ [c+z, d]$ *if* $z > 0 \wedge a+z \leq 5 \wedge c+z \leq 5$
- $[r2]: n/x_{\texttt{b}}/g_{\texttt{k}}/ok \rightarrow (n-1)/[0,0]; x_{\texttt{b}}/g_{\texttt{k}}/ok$ *if* $n > 0$
- $[r3]: n/h_{\texttt{rt}}; x_{\texttt{b}}/y; \texttt{zt}\ v_{\texttt{t}}/ok \rightarrow n/x_{\texttt{b}}/y; h_{\texttt{rt}}\ v_{\texttt{t}}\ /ok$
- $[r4]: y; h_{\texttt{rt}}\ v_{\texttt{t}} \rightarrow \texttt{cook}(y; h_{\texttt{rt}}\ v_{\texttt{t}}, z)$
- $[r5]: y; [a,b]\ v_{\texttt{t}} \rightarrow y; [b,a]\ v_{\texttt{t}}$
- $[r6]: n/x_{\texttt{b}}/y; [a,b]\ v_{\texttt{t}}/ok \rightarrow n/[a,b]; x_{\texttt{b}}/y; \texttt{zt}\ v_{\texttt{t}}/ok$
- $[r7]: n/x_{\texttt{b}}/y; [5,5]\ v_{\texttt{t}}/ok \rightarrow n/x_{\texttt{b}}/g_{\texttt{k}}/ok+1$ *if* $\texttt{cook}(y; \texttt{zt}\ v_{\texttt{t}}, 1) \rightarrow g_{\texttt{k}}$

Normal form of the rules

The only rule in R whose normal form differs from the rule itself is $[r7]$, whose normal form is

$$[r7^{\circ}]: n/x_{\texttt{r}}/y; [a,b]\ v_{\texttt{t}}/ok \rightarrow n/x_{\texttt{r}}/g_{\texttt{k}}/ok+1 \text{ if } \texttt{cook}(y; \texttt{zt}\ v_{\texttt{t}}, 1) \rightarrow g_{\texttt{k}} \mid a = 5 \wedge b = 5$$

5.2 Expressiveness. Properties of top_{Σ_0}. Rewriting with B-extensions

5.2.1 Expressiveness of $\rightarrow^1_R$ and $\rightarrow^1_{R/\mathcal{E}}$

The use of normal rewrite rules will allow our narrowing calculus to split $\mathcal{E}$-unification into B-unification plus $\mathcal{E}$-satisfiability.
In conditional rewriting with built-ins, rewriting modulo is more expressive than rewriting ($\rightarrow^1_R \subsetneq \rightarrow^1_{R/\mathcal{E}}$): from their definitions in Sections 2.3.3 and 2.3.4, it is clear that $\rightarrow^1_R \subseteq \rightarrow^1_{R/\mathcal{E}}$; in the next example we prove that, in general, $\rightarrow^1_{R/\mathcal{E}} \not\subseteq \rightarrow^1_R$.

Example 15. *Let us assume a rewrite theory* $\mathcal{R} = (\Sigma, E_0 \cup B, R)$*, where* E_0 *is integer arithmetic,* f *and* g *are function symbols in* Σ_1*;* $B = \{f(x,y) = f(y,x)\}$*; and the only rule in* R *is* $c: f(2+x, 0) \rightarrow g(x)$*. Then* $f(0,3)$ *cannot be rewritten in* R *because* $f(0,3) \neq f(2+x, 0)\sigma$*, syntactically speaking, for any substitution* σ*, but* $f(0,3) \rightarrow^1_{R/\mathcal{E}} g(1)$ *with* $\sigma = \{x \mapsto 1\}$*, because* $3 =_{E_0} 2+1$*, so* $f(0,3) =_{E_0} f(0,2+1) =_B f(2+1,0) = f(2+x,0)\sigma$*.*

Rewriting modulo with built-ins can be imitated using a B-matching algorithm and an oracle for E_0 instead of an $\mathcal{E}$-matching algorithm.

Example 16. *In example 15 we can convert the* $\mathcal{E}$*-matching problem* $f(0,1+2) =_{\mathcal{E}} f(2+x,0)$ *into the* B*-matching problem* $f(0,1+2) =_B f(z,0)$*, with the condition* $z =_{E_0} 2+x$ *(this is a simplification of the real algorithm, to make the example more understandable). The* B*-matching algorithm returns the answer* $\{z \mapsto 1+2\}$*, that gives us the condition* $1+2 =_{E_0} 2+x$*, and the oracle for* E_0 *recognizes it as satisfiable. Then,* $f(0,1+2)$

rewrites modulo $\mathcal{E}$ *to* $g(x)$ *with the condition* $1 + 2 =_{E_0} 2 + x$*, that has as only solution* $\{x \mapsto 1\}$*. The pair* $g(x)|(1 + 2 = 2 + x)$*, is called a* constrained term*. The* E_0 *subscript in the equality sign is removed from the constrained terms since it is always implied by the rewrite theory involved.*

The narrowing calculus presented in this chapter is based on the use of constrained terms, a complete B-unification algorithm, and an oracle for E_0.

5.2.2 Properties of top_{Σ_0}

We show in this section the main properties related to the concept of set of topmost Σ_0-positions presented in Section 2.3.7.

Proposition 7 (Invariants of top_{Σ_0} under E_0-equality)**.** *Let* $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ *be a rewrite theory with built-in subtheory* (Σ_0, E_0)*. If* t *and* t' *are two terms in* $\mathcal{H}_\Sigma(\mathcal{X})$ *such that* $t =_{E_0} t'$ *then:*

1. $top_{\Sigma_0}(t) = top_{\Sigma_0}(t')$,
2. $ls(t|_q) = ls(t'|_q)$ *and* $t|_q =_{E_0} t'|_q$ *for all positions* q *in* $top_{\Sigma_0}(t)$,
3. $t|_{q'} =_{E_0} t'|_{q'}$ *for all positions* q' *such that* $t|_{q'} \in \mathcal{H}_\Sigma(\mathcal{X})$*, and*
4. *if* $top_{\Sigma_0}(t) = \{q_1, \ldots, q_n\}$ *then* $t' = t[t'|_{q_1}]_{q_1} \cdots [t'|_{q_n}]_{q_n}$.

See proof on page 125.

Proposition 8 (Relation between $abstract_{\Sigma_1}$ and top_{Σ_0})**.** *Let* $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ *be a rewrite theory with built-in subtheory* (Σ_0, E_0)*. If* t *is a term in* $\mathcal{H}_\Sigma(\mathcal{X})$*,* $abstract_{\Sigma_1}(t, \mathcal{Y}) = \langle \lambda \bar{x}.t^\circ; \theta^\circ; \phi^\circ \rangle$*, where* $\bar{x} = \{x_1, \ldots, x_n\}$ *and* $t^\circ = t[x_1]_{q_1} \cdots [x_n]_{q_n}$*, then*

(i) $top_{\Sigma_0}(t) = \{q_1, \ldots, q_n\}$*, and*

(ii) for every substitution $\sigma : \bar{x} \to \mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$ *it holds that* $top_{\Sigma_0}(t^\circ\sigma) = top_{\Sigma_0}(t)$.

See proof on page 125.

Proposition 9 (Bijection between top_{Σ_0} positions in B-equal terms)**.** *Given an OS equational theory* $\mathcal{E} = (\Sigma, E_0 \cup B)$ *and two terms* u *and* v *in* $\mathcal{H}_\Sigma(\mathcal{X})$ *such that* $u =_B v$*, where* $u = u_0 \overset{ax_1}{\longleftrightarrow}_B \cdots \overset{ax_n}{\longleftrightarrow}_B u_n = v$*,* $\overline{ax} = ax_1, \ldots, ax_n$*, with* $\hat{ax} \subset B \cup B^{-1}$*, if* $top_{\Sigma_0}(u) = \hat{p}$ *and* $top_{\Sigma_0}(v) = \hat{q}$ *then there exists a bijective function* $dest_{\overline{ax}} : \hat{p} \to \hat{q}$ *such that* $u|_{p_i} = v|_{dest_{\overline{ax}}(p_i)}$*, for each position* p_i *in* $\hat{p}$.

See proof on page 126.

Corollary 2 (Bijection between top_{Σ_0} positions in $\mathcal{E}$-equal terms)**.** *Given an OS equational theory* $\mathcal{E} = (\Sigma, E_0 \cup B)$ *and two terms* u *and* v *in* $\mathcal{H}_\Sigma(\mathcal{X})$ *such that* $u =_{\mathcal{E}} v$*, if* $top_{\Sigma_0}(u) = \hat{p}$ *and* $top_{\Sigma_0}(v) = \hat{q}$ *then there exists a bijective function* $dest : \hat{p} \to \hat{q}$*, hence* $\hat{q} = dest(\hat{p})$*, such that* $u|_{p_i} =_{E_0} v|_{dest(p_i)}$*, for each position* p_i *in* $\hat{p}$.

See proof on page 127.

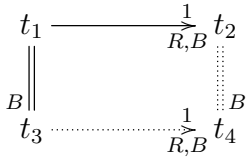

Figure 5.2: Strict coherence of $\to^1_{R,B}$

Lemma 9 (Relation between $\mathcal{E}$-unifiers and B-unifiers of abstractions)**.** *Given an OS equational theory $\mathcal{E} = (\Sigma, E_0 \cup B)$ and two terms u and v in $\mathcal{H}_\Sigma(\mathcal{X})$, if $abstract_{\Sigma_1}((u,v)) = \langle \lambda(\bar{x},\bar{y}).(u^\circ, v^\circ); (\theta^\circ_u, \theta^\circ_v); (\phi^\circ_u, \phi^\circ_v)\rangle$ and σ' is a ground substitution such that $V_{u,v} \subseteq dom(\sigma')$, $u\sigma' =_{\mathcal{E}} v\sigma'$, and $dom(\sigma') \cap (\hat{x} \cup \hat{y}) = \emptyset$ then there exists another ground substitution σ° such that $u^\circ\sigma^\circ =_B v^\circ\sigma^\circ$, $E_0 \vdash (\phi^\circ_u \wedge \phi^\circ_v)\sigma^\circ$, $dom(\sigma^\circ) = dom(\sigma') \cup \hat{x} \cup \hat{y}$, so $V_{(u^\circ, v^\circ, \phi^\circ_u, \phi^\circ_v)\sigma^\circ} = \emptyset$, and $\sigma' =_{E_0} \sigma^\circ_{dom(\sigma')}$.*

See proof on page 127.

5.2.3 One-step B-deduction and E_0-deduction

Let $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ be a rewrite theory with built-in subtheory (Σ_0, E_0). Let $B^{-1} = \{v = w \mid w = v \in B\}$, then we write $l \leftrightarrow_B r$ iff there exists $v = w$ in $B \cup B^{-1}$, a position p in l and a substitution σ such that $l|_p = v\sigma$ and $r = l[w\sigma]_p$. Let $E_0^{-1} = \{v = w \text{ if } C \mid w = v \text{ if } C \in E_0\}$, then we write $l \leftrightarrow_{E_0} r$ iff there exists $v = w$ if C in $E_0 \cup E_0^{-1}$, a position p in l and a substitution σ such that $l|_p = v\sigma$, $r = l[w\sigma]_p$, and $E_0 \vdash C\sigma$. We define $\leftrightarrow_{\mathcal{E}} = \leftrightarrow_B \cup \leftrightarrow_{E_0}$. This notion is needed for the proof of the following result.

Proposition 10 (Decomposition of $\mathcal{E}$-equality in B-equality plus E_0-equality)**.** *Let $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ be a rewrite theory with built-in subtheory (Σ_0, E_0). If t and t'' are terms in $\mathcal{H}_\Sigma(\mathcal{X})$ and $t =_{\mathcal{E}} t''$ then there exists a term t' in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $t =_B t' =_{E_0} t''$.*

See proof on page 129.

5.2.4 Rewriting with B-extensions

When working with a closed under B-extensions rewrite theory $\mathcal{R}$, as in Definition 4.2.1, the relation between rewriting modulo SMT plus axioms and rewriting with B-extensions, is slightly different to the one in Chapter 4, since it also involves the normal rewrite rules from $\mathcal{R}$.

Corollary 2 in [Mes17] can be applied in a straightforward way to $\to^1_{R,B}$, yielding the following lemma.

Lemma 10 (Independence of R, B-rewriting modulo B for rewrite theories closed under B-extensions)**.** *Let $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ be a rewrite theory with built-in subtheory (Σ_0, E_0). If R is closed under B-extensions then $\to^1_{R,B}$ is strictly coherent, i.e., for all t_1, t_2, t_3 if $t_1 \to^1_{R,B} t_2$ and $t_1 =_B t_3$ then there exists t_4 such that $t_3 \to^1_{R,B} t_4$ and $t_2 =_B t_4$ (see Fig. 5.2).*

Example 17. *In the cooking example, R is closed under B-extensions because the subterms of the equations in B have sorts* `Toast`*,* `Bin`*, or* `Pan`*, and no head of any rule in R has any of these sorts.*

Theorem 11 (Equivalence of $R/\mathcal{E}$ and R,B-rewriting for rewrite theories closed under B-extensions)**.** *Let $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ be a rewrite theory with built-in subtheory (Σ_0, E_0). If $\mathcal{R}$ is closed under B-extensions then $\rightarrow^1_{R,B} \subseteq \rightarrow^1_{R/\mathcal{E}}$, $\rightarrow_{R,B} = \rightarrow_{R/\mathcal{E}}$, and if t and w are terms in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $t \rightarrow^1_{R/\mathcal{E}} w$ then there exists t' in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $t \rightarrow^1_{R,B} t' =_E w$.*

See proof on page 130.

Corollary 3. *Let $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ be a rewrite theory with built-in subtheory (Σ_0, E_0). If $\mathcal{R}$ is closed under B-extensions then any substitution is $R/\mathcal{E}$-normalized iff it is R,B-normalized.*

5.3 Reachability by conditional narrowing modulo SMT plus axioms

In this section, the narrowing calculus for reachability is introduced, its soundness and weak completeness are stated, and completeness for *topmost* rewrite theories is also stated.

Narrowing is like R,B-rewriting, where unification is used instead of matching, allowing the inspection of a set of initial states, namely the ground instances of the given symbolic initial state, which can have variables both in its SMT and non-SMT subterms, in contrast with [RMM17] whose initial states can only have variables in its SMT subterms.

Consider a reachability problem $P = \bigwedge_{i=1}^n t_i \rightarrow v_i \mid \psi$. For simplicity of the explanation, let $n = 1$. A way to solve this problem using narrowing is to find a series of narrowing steps $t_1 \rightarrow v_1 \mid \psi \rightsquigarrow_{\sigma_1} \cdots \rightsquigarrow_{\sigma_{m-1}} t' \rightarrow v_1\sigma_1 \cdots \sigma_{m-1} \mid \phi$ and then find a substitution σ_m such that $t'\sigma_m =_E v_1\sigma_1 \cdots \sigma_m$ and $E_0 \vdash \phi\sigma_m$. It is immediate to show, using induction on the number of narrowing steps, that $\sigma = \sigma_1 \cdots \sigma_m$ is a solution for P.

From the definition of narrowing for rewrite theories with built-in in in Section 2.3.11:

"$t \rightsquigarrow^1 t'$ if there is a position $p \in Pos(t)$, a rule $c \in R$, with normal form $c^\circ : l^\circ \rightarrow r$ if $\bigwedge_{j=1}^m l_j \rightarrow r_j \mid \phi \wedge \phi'$, and a substitution σ such that $rep(t|_p)\sigma =_B l\sigma$, $t' = (t[r]_p)\sigma$, $l_j\sigma \rightarrow_{R,B} r_j\sigma$, for $1 \leq j \leq m$, and $E_0 \vdash (\phi \wedge \phi')\sigma$",

in order to perform a narrowing step from the term t_1 in P with rule $c \in R$ it is required that $rep(t_1|_p)\sigma =_B l^\circ\sigma$, $l_j\sigma \rightarrow_{R,B} r_j\sigma$, for $1 \leq j \leq m$, and $E_0 \vdash (\psi \wedge \phi \wedge \phi')\sigma$. This is not trivial. A method to achieve this task is to consider a position p in $Pos_\Sigma(t_1)$, and a B-unifier ρ_0 in $CSU_B(rep(t_1|_p) = l)$ such that the SMT condition $(\psi \wedge \phi \wedge \phi')\rho_0$ is satisfiable. Then $l_j\rho_0 \rightarrow r_j\rho_0$, for $1 \leq j \leq m$, is a set of reachability problems. Each problem $l_j\rho_0 \cdots \rho_{j-1} \rightarrow r_j\rho_0 \cdots \rho_{j-1}$, for $1 \leq j \leq m$, is solved recursively by narrowing, with the previous satisfiable SMT condition as new reachability formula, yielding the next substitution ρ_j and another satisfiable SMT condition as solution. If we take $\sigma = \rho_0 \cdots \rho_m$ and call Ψ the last satisfiable SMT condition obtained, then $P \rightsquigarrow^1_{p,c,\sigma} (t_1[r]_p \rightarrow v_1 \wedge \bigwedge_{i=2}^n t_i \rightarrow v_i)\sigma \mid \Psi$.

The method sketched in the previous paragraphs is the one used in the *calculus for reachability by conditional narrowing modulo SMT plus axioms*, whose calculus rules are shown in Figure 5.3. This calculus handles reachability goals, which in this context are an extension of reachability problems.

5.3.1 Reachability goal

Given a rewrite theory $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ with built-in subtheory (Σ_0, E_0), a reachability goal G is an expression with the form

1. $\bigwedge_{i=1}^{n} u_i \to v_i \mid \phi$, or
2. $u_1 \to^1 x_k, u_2[x_k]_p \to v_2 \wedge \bigwedge_{i=3}^{n} u_i \to v_i \mid \phi$,

where $n \geq 0$, $u_i, v_i \in \mathcal{H}_\Sigma(\mathcal{X})$, for $1 \leq i \leq n$, $\phi \in QF(\mathcal{X}_0)$, $p \in Pos(u_2)$, $k = [ls(u_1)]$, the kind of the least sort of u_1, and x_k appears exactly twice in G in case (2). We say that x_k is the *connecting variable* of the goal.

Reachability problems are reachability goals with the first form; reachability goals with the second form are generated by the calculus rules; this second form prevents the repeated application of rule transitivity in a derivation, forcing the calculus of a narrowing step in the first subgoal of the reachability problem.

The notations $P \rightsquigarrow_{[r]} P'$, $P \rightsquigarrow_{[r],\sigma} P'$, or $P \rightsquigarrow_{[r],c,\sigma} P'$, will be used in the calculus to indicate that rule $[r]$ of the calculus has been applied (with substitution σ, if needed, and using rule c from R in the case that $[r]$ is the rewrite rule) to P, yielding P'.

We extend the definition of solution of a reachability problem in $\to^1_{R/\mathcal{E}}$ and $\to^1_{R,B}$ to reachability goals with the second form.

Solution of a reachability goal in $\to^1_{R/\mathcal{E}}$

Given a reachability goal $G = u_1 \to^1 x_k, u_2[x_k]_p \to v_2 \wedge \bigwedge_{i=3}^{n} u_i \to v_i \mid \phi$, a substitution $\sigma : vars(G) \to \mathcal{T}_\Sigma$ is a *solution* of G in $\to^1_{R/\mathcal{E}}$ if $u_1\sigma \to^1_{R/\mathcal{E}} x_k\sigma$, $u_2[x_k]_p\sigma \to_{R/\mathcal{E}} v_2\sigma$, $u_i\sigma \to_{R/\mathcal{E}} v_i\sigma$, for $3 \leq i \leq n$, and $E_0 \vdash \phi\sigma$.

Solution of a reachability goal in $\to^1_{R,B}$

Given a reachability goal $G = u_1 \to^1 x_k, u_2[x_k]_p \to v_2 \wedge \bigwedge_{i=3}^{n} u_i \to v_i \mid \phi$, a substitution $\sigma : vars(G) \to \mathcal{T}_\Sigma$ is a *solution* of G in $\to^1_{R,B}$ if $u_1\sigma \to^1_{R,B} x_k\sigma$, $u_2[x_k]_p\sigma \to_{R,B} v_2\sigma$, $u_i\sigma \to_{R,B} v_i\sigma$, for $3 \leq i \leq n$, and $E_0 \vdash \phi\sigma$.

By Theorem 11, when $\mathcal{R}$ is closed under B-extensions, the solutions of any reachability goal under $\to^1_{R/\mathcal{E}}$ and $\to^1_{R,B}$ are the same.

5.3.2 Empty goal. Narrowing path. Computed answer

Empty goal and narrowing path

We call $nil \mid \phi$, where ϕ is satisfiable, an *empty goal*. Given a closed under B-extensions rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ with built-in subtheory (Σ_0, E_0), a reachability problem P in $\to^1_{R/\mathcal{E}}$ is solved by applying the calculus rules in Figure 5.3, starting with P and in a top-down manner, until an empty goal is obtained, generating a *narrowing path*.

Computed answer

Given a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ with built-in subtheory (Σ_0, E_0), and a reachability goal G, if there is a narrowing path $G \rightsquigarrow_{\sigma_1} G_1 \rightsquigarrow_{\sigma_2} \cdots \rightsquigarrow_{\sigma_{n-1}} G_{n-1} \rightsquigarrow_{\sigma_n} nil \mid \psi$, using the calculus rules in Figure 5.3, hence ψ is satisfiable, then we write $G \rightsquigarrow^*_{\sigma} nil \mid \psi$, with $\sigma = \sigma_1 \cdots \sigma_n$, and we call $\sigma_{vars(G)} \mid \psi$ a *computed answer* for G. As the unifiers σ_i,

- $[u]$ unification

$$\frac{u \to v \ \wedge \ \Delta \mid \phi}{\Delta\theta \mid \psi}$$

where $abstract_{\Sigma_1}((u, v)) = \langle\lambda(\bar{x}, \bar{y}).(u^\circ, v^\circ); (\theta_u^\circ, \theta_v^\circ); (\phi_u^\circ, \phi_v^\circ)\rangle$,
σ in $CSU_B(u^\circ = v^\circ)$, $vars(\psi) \subseteq vars((\phi \wedge \phi_u^\circ \wedge \phi_v^\circ)\sigma)$,
$E_0 \vdash \psi \Leftrightarrow (\phi \wedge \phi_u^\circ \wedge \phi_v^\circ)\sigma$, and ψ is satisfiable

- $[t]$ transitivity

$$\frac{u \to v \ (\wedge \Delta) \mid \phi}{u \to^1 x_k, x_k \to v \ (\wedge \Delta) \mid \phi}$$

where $u \notin \mathcal{X}$, $k = [ls(u)]$, and x_k fresh variable

- $[c]$ congruence

$$\frac{u|_p \to^1 x_k, u[x_k]_p \to v \ (\wedge \Delta) \mid \phi}{u_i \to^1 y_{k'}, u[y_{k'}]_{p.i} \to v \ (\wedge \Delta) \mid \phi}$$

where $u|_p = f(u_1, \ldots, u_n)$, $u_i \notin \mathcal{X} \cup \mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$,
$k' = [ls(u_i)]$, $1 \leq i \leq n$, and $y_{k'}$ fresh variable

- $[r]$ rewrite

$$\frac{u|_p \to^1 x_k, u[x_k]_p \to v \ (\wedge \Delta) \mid \phi}{(C \wedge u[r]_p \to v \ (\wedge \Delta))\theta \mid \psi}$$

where $u|_p \notin \mathcal{X}$, $l \to r$ *if* $C \mid \phi'$ fresh rule in R,
$abstract_{\Sigma_1}((u|_p, l)) = \langle\lambda(\bar{x}, \bar{y}).(u^\circ, l^\circ); (\theta_u^\circ, \theta_l^\circ); (\phi_u^\circ, \phi_l^\circ)\rangle$,
θ in $CSU_B(u^\circ = l^\circ)$, $vars(\psi) \subseteq vars((\phi \wedge \phi' \wedge \phi_l^\circ \wedge \phi_u^\circ)\theta)$,
$E_0 \vdash \psi \Leftrightarrow (\phi \wedge \phi' \wedge \phi_l^\circ \wedge \phi_u^\circ)\theta$, and ψ is satisfiable

Figure 5.3: Inference rules for reachability by conditional narrowing modulo SMT plus axioms.

$1 \leq i \leq n$, returned by CSU_B are idempotent and away from all the variables that have previously appeared in the computation, so $ran(\sigma_i) \cap \bigcup_{j=1}^{i-1} ran(\sigma_j) = \emptyset$, then σ is also idempotent.

Rules unification and rewrite allow for simplifications in the reachability formulas obtained, i.e., $(\phi \wedge \phi^{\circ})\theta$ can be replaced with another formula ψ under the assumptions stated in both rules. For instance $X - Y + Z > 0 \wedge X = Y$ can be replaced with $Z > 0$. It is always possible to obtain the same computed answer without using simplifications.

Proposition 11 (Existence of canonical paths). *Given a rewrite theory $\mathcal{R} = (\Sigma, E, R)$ with built-in subtheory (Σ_0, E_0), and a narrowing path from a reachability goal G, $G = \Delta_0 \mid \psi_0 \rightsquigarrow_{\sigma_1} \Delta_1 \mid \psi_1 \rightsquigarrow_{\sigma_2} \cdots \Delta_{m-1} \mid \psi_{m-1} \rightsquigarrow_{\sigma_m} nil \mid \psi_m$, there exists another narrowing path $G = \Delta_0 \mid \psi_0 \rightsquigarrow_{\sigma_1} \Delta_1 \mid \chi_1 \rightsquigarrow_{\sigma_2} \cdots \Delta_{m-1} \mid \chi_{m-1} \rightsquigarrow_{\sigma_m} nil \mid \chi_m$, where if one rule is applied at step i in one path then the same rule is applied at step i in the other path, for $1 \leq i \leq m$, there is no simplification of the reachability formula when rule unification or rewrite is applied, and $E_0 \vdash \psi_i \Leftrightarrow \chi_i$, for $1 \leq i \leq m$.*

See proof on page 132.

5.3.3 Soundness and weak completeness of the calculus

In this section it is stated that the calculus rules are a sound method for solving reachability goals in $\rightarrow^1_{R,B}$. A distinction is made depending on the form of the reachability goal. For goals of the second form it is necessary to be very careful with the connecting variable of the goal, since this variable does not appear in the original reachability problem.

Theorem 12 (Soundness in $\rightarrow^1_{R,B}$ of the Calculus for Reachability Goals). *Given a closed under B-extensions rewrite theory $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ with built-in subtheory (Σ_0, E_0), where $\mathcal{E} = E_0 \cup B$, and a narrowing path from a reachability goal G, $G = \Delta_1 \mid \psi_1 \rightsquigarrow_{\sigma_1} \Delta_2 \mid \psi_2 \rightsquigarrow_{\sigma_2} \cdots \Delta_m \mid \psi_m \rightsquigarrow_{\sigma_m} nil \mid \psi$, let $\sigma = \sigma_1 \cdots \sigma_m$, then:*

1. *if $\Delta_1 = \bigwedge_{i=1}^n u_i \rightarrow v_i$ and $\rho : \mathcal{X} \rightarrow \mathcal{T}_\Sigma$ is a substitution such that $dom(\rho) = vars(G\sigma)$ and $\psi\rho$ is satisfiable then $(\sigma\rho)_{vars(G)}$ is a solution for G in $\rightarrow^1_{R,B}$, and*
2. *if $\Delta_1 = u_1|_p \rightarrow^1 x, u_1[x]_p \rightarrow v_1 \wedge \bigwedge_{i=2}^n u_i \rightarrow v_i \mid \psi_1$ and $\rho : \mathcal{X} \setminus \{x\} \rightarrow \mathcal{T}_\Sigma$ is a substitution such that $dom(\rho) = vars(G\sigma) \setminus \{x\}$ and $\psi\rho$ is satisfiable then*
   (a) *$(\sigma\rho)_{vars(G)\setminus\{x\}}$ is a solution for $\bigwedge_{i=1}^n u_i \rightarrow v_i \mid \psi_1$ in $\rightarrow^1_{R,B}$ and*
   (b) *there exists a substitution $\rho_x : \{x\} \rightarrow \mathcal{T}_\Sigma$ such that $(\sigma(\rho \cup \rho_x))_{vars(G)}$ is a solution for G in $\rightarrow^1_{R,B}$.*

See proof on page 132.

The soundness of the calculus rules with respect to the solutions of reachability problems in $\rightarrow^1_{R/\mathcal{E}}$, for rewrite theories closed under B-extensions, is now a consequence of the soundness of the calculus rules in $\rightarrow^1_{R,B}$ and the fact that reachability problems are a special case of reachability goals.

Theorem 13 (Soundness in $\rightarrow^1_{R/\mathcal{E}}$ of the Calculus for Reachability Problems). *Given a closed under B-extensions rewrite theory $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ with built-in subtheory (Σ_0, E_0), and a narrowing path from a reachability problem G, $G = \bigwedge_{i=1}^n u_i \rightarrow v_i \mid \psi_1 \rightsquigarrow^*_\sigma nil \mid \psi$, if $\rho : \mathcal{X} \rightarrow \mathcal{T}_\Sigma$ is a substitution such that $dom(\rho) = vars(G\sigma)$ and $\psi\rho$ is satisfiable, then $(\sigma\rho)_{vars(G)}$ is a solution for G in $\rightarrow^1_{R/\mathcal{E}}$.*

Proof. By Theorem 12 (1), $(\sigma\rho)_{vars(G)}$ is a solution for G in $\rightarrow^1_{R,B}$. As $\mathcal{R}$ is closed under B-extensions, then, by Theorem 11, $(\sigma\rho)_{vars(G)}$ is also a solution for G in $\rightarrow^1_{R/\mathcal{E}}$. □

Theorem 14 (Weak Completeness in $\rightarrow^1_{R/\mathcal{E}}$ of the Calculus for Reachability Problems)**.** *Given a closed under B-extensions rewrite theory $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ with built-in subtheory (Σ_0, E_0), where $\mathcal{E} = E_0 \cup B$, and a reachability problem $P = \bigwedge_{i=1}^n u_i \rightarrow v_i \mid \phi$, if σ is an idempotent $R/\mathcal{E}$-normalized solution for P in $\rightarrow^1_{R/\mathcal{E}}$ then there exist a formula $\psi \in QF(\mathcal{X}_0)$, and substitutions γ and δ, such that $P \rightsquigarrow^*_\gamma nil \mid \psi$, $\sigma = (\gamma\delta)_{vars(P)}$, and $\psi\delta$ is satisfiable.*

See proof on page 135.

Completeness in $\rightarrow_{R/E}$ of the calculus, for topmost rewrite theories

In the proof of weak completeness of the calculus for reachability, the only place where the hypothesis of σ being R/E-normalized is used is in the induction case, (ii), where it limits the positions where rewriting can happen at some proper subterm of $u_1\sigma$, an instance of the first term in the reachability problem P (u_1). It is immediate then to prove the *completeness of the calculus for topmost rewrite theories*, those rewrite theories $\mathcal{R} = (\Sigma, E, R)$ such that for some top sort `state`, no operator in Σ has `state` as argument sort and each rule $l \rightarrow r$ *if* $\bigwedge_{i=1}^n l_i \rightarrow r_i \mid \phi$ in R satisfies $l, r \in \mathcal{T}_\Sigma(\mathcal{X})_{\texttt{state}}$ and $l_i, r_i \in \mathcal{T}_\Sigma(\mathcal{X})_{\texttt{state}}$, for $1 \leq i \leq n$, since rewriting always happens at position ϵ of $u_1\sigma$, so the hypothesis of σ being R/E-normalized is not needed for this type of rewrite theories in the proof of completeness.

5.4 Narrowing example: toast cooking

An application of the calculus using the running example is shown. All the subscripts for variables with sort `Integer` are omitted for readability. Recall the rest of subscripts in the example: $\texttt{p} - \texttt{Pan}, \texttt{rt} - \texttt{RealToast}, \texttt{t} - \texttt{Toast}, \texttt{k} - \texttt{Kitchen}, \texttt{b} - \texttt{Bin}, \texttt{s} - \texttt{System}$. Consider the reachability goal $G = n_1/\texttt{zt}/0; \texttt{zt}\ \texttt{zt}/0 \rightarrow m/x_\texttt{r}/t; y_\texttt{p}/1 \mid n_1 > 0 \wedge n_1 < 3 \wedge t < 12$, where from an initial `System` consisting of a bag containing one or two `RealToast`s, an empty `Bin`, an empty `Kitchen` (with zero seconds of elapsed time), and no well-cooked `RealToast`s, it is desired to reach a `System` with one well-cooked `RealToast` in less than twelve seconds. Let $F = m/x_\texttt{r}/t; y_\texttt{p}/1$, $\phi_1 = n_1 > 0 \wedge n_1 < 3 \wedge t < 12$, and $\phi_2 = n_2 > 0 \wedge n_2 < 3 \wedge t < 12$. Then $abstract_{\Sigma_1}(F) = \langle \lambda ok.F^\circ; \theta^\circ; \phi^\circ \rangle$, with $F^\circ = m/x_\texttt{r}/t; y_\texttt{p}/ok$ and $\phi^\circ = (ok = 1)$. Narrowing steps involving rules $[t]$ or $[c]$ have been joined in multiple narrowing steps, for instance $\rightsquigarrow^*_{[t],[c],[r]}$, after their first occurrences. The interaction between unification, SMT operations, and satisfiability is explained using the number of well-cooked `RealToast`s, $0 + 1$, in step 15:

1. $n_1/\texttt{zt}/0; \texttt{zt}\ \texttt{zt}/0 \rightarrow F \mid \phi_1 \rightsquigarrow_{[t]}$
2. $n_1/\texttt{zt}/0; \texttt{zt}\ \texttt{zt}/0 \rightarrow^1 x1_{[\texttt{s}]}, x1_{[\texttt{s}]} \rightarrow F \mid \phi_1 \rightsquigarrow_{[r]r2,\sigma_2=\{n_1 \mapsto n_2\}}$
3. $n_2 - 1/[0,0]/0; \texttt{zt}\ \texttt{zt}/0 \rightarrow F \mid \phi_2 \rightsquigarrow_{[t]}$
4. $n_2 - 1/[0,0]/0; \texttt{zt}\ \texttt{zt}/0 \rightarrow^1 x2_{[\texttt{s}]}, x2_{[\texttt{s}]} \rightarrow F \mid \phi_2 \rightsquigarrow_{[r]r3}$

5. $n_2 - 1/\mathtt{zt}/0; [0,0]\,\mathtt{zt}/0 \to F \mid \phi_2 \rightsquigarrow^*_{[t]}$

6. $n_2 - 1/\mathtt{zt}/0; [0,0]\,\mathtt{zt}/0 \to^1 x3_{[\mathtt{s}]}, x3_{[\mathtt{s}]} \to F \mid \phi_2 \rightsquigarrow^*_{[c]}$

7. $0; [0,0]\,\mathtt{zt} \to^1 y2_{[\mathtt{k}]}, n_2 - 1/\mathtt{zt}/y2_{[\mathtt{k}]}/0 \to F \mid \phi_2 \rightsquigarrow_{[r]r4}$

8. $n_2 - 1/\mathtt{zt}/\mathtt{cook}(0; [0,0]\,\mathtt{zt}, z_1)/0 \to F \mid \phi_2 \wedge z_1 > 0 \rightsquigarrow^*_{[t],[c],[r]r1b}$

9. $n_2 - 1/\mathtt{zt}/z_1; [z_1,0]\,\mathtt{zt}/0 \to F \mid \phi_2 \wedge z_1 > 0 \wedge z_1 \leq 5 \rightsquigarrow^*_{[t],[c],[r]r5}$

10. $n_2 - 1/\mathtt{zt}/z_1; [0,z_1]\,\mathtt{zt}/0 \to F \mid \phi_2 \wedge z_1 > 0 \wedge z_1 \leq 5 \rightsquigarrow^*_{[t],[c],[r]r4}$

11. $n_2 - 1/\mathtt{zt}/\mathtt{cook}(z_1; [0,z_1]\,\mathtt{zt}, z_2)/0 \to F \mid \phi_2 \ \wedge\ z_1 > 0 \wedge z_1 \leq 5 \rightsquigarrow^*_{[t],[c],[r]r1b}$

12. $n_2 - 1/\mathtt{zt}/z_1 + z_2; [z_2,z_1]\,\mathtt{zt}/0 \to F \mid \phi_2 \ \wedge\ z_1 > 0 \wedge z_1 \leq 5 \wedge z_2 > 0 \wedge z_2 \leq 5 \rightsquigarrow_{[t]}$

13. $n_2 - 1/\mathtt{zt}/z_1 + z_2; [z_2,z_1]\,\mathtt{zt}/0 \to^1 x4_{[\mathtt{s}]}, x4_{[\mathtt{s}]} \to F \mid \phi_2 \wedge$
$\wedge z_1 > 0 \wedge z_1 \leq 5 \wedge z_2 > 0 \wedge z_2 \leq 5 \rightsquigarrow_{[r]r7^\circ}$

14. $\mathtt{cook}(z_1 + z_2; \mathtt{zt}\,\mathtt{zt}, 1) \to y3_{[\mathtt{k}]} \wedge n_2 - 1/\mathtt{zt}/y3_{[\mathtt{k}]}/0 + 1 \to F \mid \phi_2 \wedge$
$\wedge z_1 = 5 \wedge z_2 = 5 \rightsquigarrow^*_{[t],[c],[r]r1c}$

15. $n_2 - 1/\mathtt{zt}/1 + z_1 + z_2; \mathtt{zt}\,\mathtt{zt}/0 + 1 \to F \mid \ \wedge$
$\wedge n_2 > 0 \wedge n_2 < 3 \wedge t < 12 \wedge z_1 = 5 \wedge z_2 = 5 \rightsquigarrow_{[u],F^\circ,\phi_0^\circ,\sigma_{15}}$

16. $nil \mid n_2 > 0 \wedge n_2 < 3 \wedge z_1 = 5 \wedge z_2 = 5$

The last narrowing step, the unification of $n_2 - 1/\mathtt{zt}/1 + z_1 + z_2; \mathtt{zt}\,\mathtt{zt}/0 + 1$ with F°, i.e., $m/x_\mathtt{r}/t; y_\mathtt{p}/ok$, is explained in detail. The unifier, indeed a matching, is $\sigma_{15} = \{m \mapsto n_2 - 1, x_\mathtt{r} \mapsto \mathtt{zt}, t \mapsto 1 + z_1 + z_2, y_\mathtt{p} \mapsto \mathtt{zt}\,\mathtt{zt}, ok \mapsto 0 + 1\}$. As there is a substitution $\sigma_2 = \{n_1 \mapsto n_2\}$ in step 2, then $\sigma_{vars(G)} = (\sigma_1 \cdots \sigma_{15})_{vars(G)} = \{n_1 \mapsto n_2, x_\mathtt{r} \mapsto \mathtt{zt}, t \mapsto 1 + z_1 + z_2, y_\mathtt{p} \mapsto \mathtt{zt}\,\mathtt{zt}, ok \mapsto 0 + 1\}$, where ok does not map to 1, but to $0 + 1$.

The new condition, including $\phi^\circ = (ok = 1)$, becomes $n_2 > 0 \wedge n_2 < 3 \wedge 1 + z_1 + z_2 < 12 \wedge z_1 = 5 \wedge z_2 = 5 \wedge 0 + 1 = 1$, which simplifies to $n_2 > 0 \wedge n_2 < 3 \wedge z_1 = 5 \wedge z_2 = 5$.

The computed answer for the reachability goal shows two different solutions, one with $n_2 = 1$ and another one with $n_2 = 2$. As $t = 1 + z_1 + z_2$, $z_1 = 5$, and $z_2 = 5$, then from a bag with one or two `RealToast`s, it is possible to reach a `System` with one well-cooked `RealToast` in $1 + 5 + 5$, i.e. 11, seconds, hence fulfilling all the requirements of the problem. The actions that lead to this answer correspond to one application of rule unification ($[u]$), that has already been explained, and with each application of rule rewrite ($[r]$):

- in step (2) a `RealToast` is taken from the bag and put in the `Bin`,
- in step (4) the `RealToast` passes from the `Bin` to the `Kitchen`,
- in steps (7) and (8) one side of the `RealToast` cooks for some time z_1 that is added to the timer,
- in step (9) the `RealToast` is flipped,
- in steps (10) and (11) the other side of the `RealToast` cooks for some time z_2 that is added to the timer,
- in steps (13) and (14) the `RealToast` becomes a well-cooked `RealToast`, forcing $z_1 = z_2 = 5$, and taken out to the dish; one second is added to the timer.

5.5 Prototypes

In this section we present two prototypes of the narrowing calculus shown in this chapter, that have been implemented using Maude. These prototypes not only implement the usual search for partial solutions via unification, but also the extraction of partial solutions for the SMT variables in the reachability problems via the inspection of the evolving SMT constraints of the computation. They can be found here, in the web page for this thesis https://maude.ucm.es/cnarrowing, together with the instructions to run them. The main difference between both prototypes is:

- the first prototype is a system module, where the search engine of Maude is used to find the solutions of the reachability problems;
- the functional prototype implements its own search engine in a functional module. In this prototype the SMT solver is used to discard already visited states by checking the SMT equivalence of the SMT constraints of the generated reachability problems instead of their syntactic equality.

At first only one prototype was developed, the system module, but during the tests we found states in the search space that were equivalent, but not syntactically equal, so the development of the functional prototype was decided.

When the work on the prototypes began after the calculus development was finished, an experimental feature, which is formalized in the next chapter, was added to them: the admission of variable SMT parameters in the specifications and problems. Reachability problems beyond those allowed by the narrowing calculus in this chapter, where only constant SMT parameters are allowed in the rules of the specifications, can be expressed using this new feature.

Several versions of each prototype have been developed to check whether certain modifications would be an improvement or not. These modifications are explained in detail later on.

5.5.1 Common functionality of the prototypes

In Maude, rewrite theories are defined using system modules and equational theories using functional modules. While the first prototype was developed as a system module, where the search engine in Maude does all the in-house work, keeping track of the search tree and discarding the duplicated states that it finds, the functional prototype was been developed as a functional module that included an implementation of its own SMT-based search engine.

We present here the common functionality of the prototypes, followed by their distinctive features. Finally, we describe several improvements that have been tested on them.

Variable SMT parameters

Constant SMT parameters are directly handled by the calculus but, in the calculus shown in this chapter, it is not possible to include variable SMT parameters in the rules to find feasible values for them given a reachability goal because each instance of a rule that is applied in the calculus gets new variable names, and the variable parameters must always have the same name. This problem has been solved in the prototypes by:

(i) turning the variable SMT parameters that appear in the rules of the specification into new SMT constants of the same name and sort;

(ii) each time that a fresh version of a rule is requested, all the new SMT constants that appear in the fresh rule are replaced with the original SMT variables. In this way, the variable parameters are never renamed.

In the next chapter a different, more flexible, approach has been taken:

(i) the parameters remain as variables,

(ii) a list of parameters must be always supplied in the reachability problem, and

(iii) it is also possible to provide with the reachability problem a substitution that instantiates partially or totally each parameter.

An extension of the module `META-LEVEL`

The prototypes were developed for an alpha version of Maude 2.7, which supports SMT satisfiability through a file called `SMT.maude` that includes four functional modules. These modules define new sorts `Boolean`, `Integer`, and `Real`, different from the sorts `Bool`, `Int`, and `Rat` that appear in all versions of Maude. The modules are inert, i.e., there are neither axioms nor equations in them, so every SMT term is a normal form. Each constant and operator in the modules just hold a hook to the C++ code that handles it. The operator `metaCheck`, found in the module `META-LEVEL` from `prelude.maude`, is used in the prototypes to check the satisfiability of any given SMT metaterm.

Maude recognizes two SMT terms as equal only if they are syntactically equal. For instance, the variable $X{:}Integer$ and the term $1.Integer{*}X{:}Integer$ are not recognized as equal by Maude because the latter SMT expression is not simplified by the `SMT` modules.

We add simplification capabilities to the SMT expressions in our module `SMTLOGIC`. In this way, the search engine can better prune the search tree, for instance by simplifying the previously mentioned term $1.Integer * X{:}Integer$ to $X{:}Integer$.

Our module `SMTLOGIC` defines SMT arithmetic metaterms and Boolean metaconditions. This extension of the module `META-LEVEL` allows the prototypes to:

- find partial solutions, consisting of ground assignments for the SMT variables in satisfiable SMT constraints,
- remove ground SMT expressions from satisfiable SMT constraints, and
- simplify SMT subterms and conditions, for instance, removing duplicated conditions.

The module `SMTLOGIC` adds simplification functionality to the prototypes. As the arithmetic operators $+$, $-$, and $*$ are overloaded, we defined new metaoperators, $+I$, $-I$, $*I$, $+R$, $-R$, and $*R$ in our module for the metaexpressions, also separating integer expressions from real expressions. We defined top sorts `SmtCondi`, for `Boolean` metaterms, and `SmtTerm`. The sort `SmtAterm` (for *SMT arithmetic term*) is reserved for ground arithmetic metaterms and also for arithmetic metaterms having neither `Boolean` metavariables nor metaoperators with sort `Real` in them. `Boolean` metavariables may appear in a non-`Boolean` metaterm in the conditional part of a ternary operator *if ... then ... else*, which deserves special treatment, hence the need for sort `SmtAterm`.

The chosen subsort ordering in SMTLOGIC reflects several pre-existing relations for the terms shared between META-LEVEL and SMTLOGIC, as well as the desired subsort ordering for the new sorts in SMTLOGIC:

```
subsort Term < SmtCondi .
subsort Term < SmtATerm < SmtTerm .
subsort GroundTerm < GroundSmtTerm < SmtATerm .
subsort GroundTerm < GroundSmtCondi < SmtCondi .
```

In this way:

- variables with sort Boolean, that become terms with sort Term in the META-LEVEL, have also sort SmtCondi in SMTLOGIC,
- non-Boolean SMT variables, that become terms with sort Term in the META-LEVEL, have also sort SmtATerm in SMTLOGIC,
- non-Boolean SMT constants, that become terms with sort GroundTerm in the META-LEVEL, have also sort GroundSmtTerm in SMTLOGIC, a sort that is needed to characterize the terms with sort GroundSmtCondi, and
- the metaconstants 'true.Boolean and 'false.Boolean have sort GroundTerm in the META-LEVEL, but they also have sort GroundSmtCondi in SMTLOGIC, a sort that helps in the task of removing all the redundant ground subexpressions that may appear in a satisfiable SMT constraint.

The module has equations that simplify tautological Boolean conditions, equations that simplify arithmetic operations, like those involving the identity constants for each operation, and, finally, equations that exchange the left and right terms in some expressions, aiming to:

- obtain a canonical representation of the expressions, helping the search engine of Maude to identify more terms as equal, and
- simplify the rules of the prototypes that are in charge of extracting partial solutions from satisfiable SMT constraints.

The function smtSimplify simplifies SMT constraints. Given an SMTLOGIC metacondition, the function downSmt, that modifies the arithmetic and Boolean operators and also modifies the sort of each constant and variable, is called on each arithmetic SMT subterm or ground Boolean SMT subterm, generating new metasubterms. When applying the META-LEVEL operator downTerm to each one of these new metasubterms, we obtain terms with sorts (Int, Rat, or Bool) that Maude simplifies automatically. Then we apply the META-LEVEL operator upTerm followed by a call to another SMTLOGIC operator MaudeSort2SmtSort that replaces the sort of each variable or constant with the corresponding one for SMT expressions. Finally, in the case of arithmetic expressions a call to the SMTLOGIC operators term2iExpr, term2rExpr, or term2cExpr restores the SMTLOGIC metaoperators in each metasubterm.

There exist also a pair of functions, constants2variables and variables2constants that are used in the extension of the prototypes to support variable parameters in the rules of the specifications, as explained in Section 5.5.4.

Normal form of rules and states in the system under test. Motivation for the functional prototype

Each prototype uses a normal form of the rules in the specification. This normal form is slightly different in each prototype, to obtain a better performance.

Given a rewrite theory $\mathcal{R} = (\Sigma, E_0 \cup B, R)$, both prototypes use a normal form of the rules in R, called R°. After computing a closure under B-extensions of R, R° is generated from this closure by abstracting all the SMT terms in the head of the rules, for the rule-based prototype, and all the SMT terms in the rules, for the functional one. The abstraction is achieved in each rule by:

(i) replacing each selected non-variable SMT term of the rule with a new SMT variable and

(ii) adding a new condition to the SMT constraint in the rule, stating the equality between the SMT term and the SMT variable.

Removal of already visited states is done by the built-in search engine of Maude in the rule-based prototype and using our own defined operator in the functional one, motivating the consideration of different normal forms in each prototype:

Consider one state, $s(\mathtt{SMT_1})|\mathtt{SMT_C}$, where $\mathtt{SMT_1}$ is a SMT subexpression and $\mathtt{SMT_C}$ is the condition associated to the state. Two different abstractions of this state, $s(X)|\mathtt{SMT_C}\ \&\ X = \mathtt{SMT_1}$ and $s(Y)|\mathtt{SMT_C}\ \&\ Y = \mathtt{SMT_1}$, are not identified as equal by the built-in rewrite engine of Maude in the rule-based prototype, because $X \neq Y$ and the engine searches for equality modulo axioms to remove states, so it is better not to abstract the expressions in the rule-based prototype.

The functional prototype identifies the states in the previous example as equal. Also, if $\mathtt{SMT_1}$ and $\mathtt{SMT_2}$ are two equivalent expressions that are syntactically different, so Maude cannot identify them as being equal, then our own defined operator will also recognize as being the same state two abstractions $s(X)|\mathtt{SMT_C}\ \&\ X = \mathtt{SMT_1}$ and $s(Y)|\mathtt{SMT_C}\ \&\ Y = \mathtt{SMT_2}$, but it will not recognize as being the same state the non-abstracted versions $s(\mathtt{SMT_1})|\mathtt{SMT_C}$ and $s(\mathtt{SMT_2})|\mathtt{SMT_C}$, because our operator uses the built-in unification method of Maude to unify the non-SMT part of the states, and this method can unify $s(X)$ and $s(Y)$, but it fails to unify $s(\mathtt{SMT_1})$ and $s(\mathtt{SMT_2})$. Of course, the rule-based prototype fails on both cases of this paragraph, which is the motivation for the development of the functional one.

Depth-oriented generation of unifiers

One design decision, that is shared by both prototypes, concerns the depth at which each unifier is tried in the search tree of the reachability problem. This search tree is generated using the rules of the specification, and controlled by the prototypes. We have designed the rules of the rule-based prototype and the equations of the functional one so that the set of unifiers for any given pair of terms is generated in a depth-oriented way instead of in a width-oriented way, i.e., if the first unifier is tried at level n of the search tree, then the second unifier is tried at level $n + 1$, and so on. Level n of the search tree can be also used to generate other narrowing paths. This design allows the search tree of the reachability problem to be more fair to all the rules that can be applied to any state in the search tree, and also to support potentially infinite unification of terms, like in the associative case, without losing completeness.

The normal form of the rules includes the kind of every rule and every rewrite condition in the rule. As the kind of each reachability subgoal in the original problem is also computed, we only have to select rules with the same kind as the current subgoal, using the matching capabilities of Maude, to try to unify the subgoal with the left side of the rule as a first step towards generating a narrowing step, discarding the rest of the rules. The normal form of the rules also includes a serial number that is used with some memoized operators. The set of normal forms of the rules together with their kind and serial number is memoized in the operator `normalRls`.

A version of the specification without the rules is also memoized. It is used for equational simplification of the metaterms generated by the prototypes. The other memoized set holds the SMT variable parameters that are inferred from the specification. It consists of all the constants with SMT sort.

Syntax of the reachability problems

The reachability problem that we want to solve, P, is represented in our extension of the `META-LEVEL` as a term with sort `OrigPrCond`. The signature for this sort is:

```
op nilOP : -> OrigPr [ctor] .
op _=>*_ : Term Term -> OrigPr [ctor] .
op _&_ : OrigPr OrigPr -> OrigPr [ctor assoc id: nilOP] .
op _&&_ : OrigPr Term -> OrigPrCond [ctor] .
```

For each reachability problem P in our examples, we have obtained each term with sort `Term` using the `upTerm` operator on the corresponding term of P.

Each subterm with the form `u =>* v` is the metarepresentation of a subgoal of P, the `&` symbol joins the different subgoals of P, and the term after the `&&` symbol is the metarepresentation of the SMT constraint of P.

This term with sort `OrigPrCond` is then processed by the operator `problem`:

```
op problem : OrigPrCond -> Problem .
```

producing a term with sort `Problem`, the key sort of the prototypes, with signature:

```
op _;_;_;_ : ReachGoal TermList Int Substitution -> Problem [ctor prec 79] .
```

The `TermList` holds the list of initial variables of the problem, `Int` has the number of the next fresh variable, and `Substitution` holds the computed substitution so far; `ReachGoal` is a supersort of `ReachProblem`. A reachability problem, with sort `ReachProblem`, holds the metarepresentation of all the subproblems and the SMT constraint of a given `Problem`. The subproblems within a `ReachProblem` are metarepresented using the sort `RuleCondi`, and the SMT constraint of the problem with a term with sort `SmtCondi`. Sort `ReachGoal` is used to represent the intermediate states in the generation of a narrowing step, after rules transitivity, congruence, or rewrite of the narrowing calculus are applied. When rule unification of the narrowing calculus is applied to a `ReachGoal`, finishing the generation of a narrowing step, the obtained term has again sort `ReachProblem`.

```
op nilR : -> RuleCondi [ctor] .
op errorR : -> RuleCondi [ctor] .
op _/_=>*_ : Kind Term Term -> RuleCondi [ctor] .
op _&_ : RuleCondi RuleCondi -> RuleCondi [ctor assoc id: nilR] .
op _||_ : RuleCondi SmtCondi -> ReachProblem [ctor] .
```

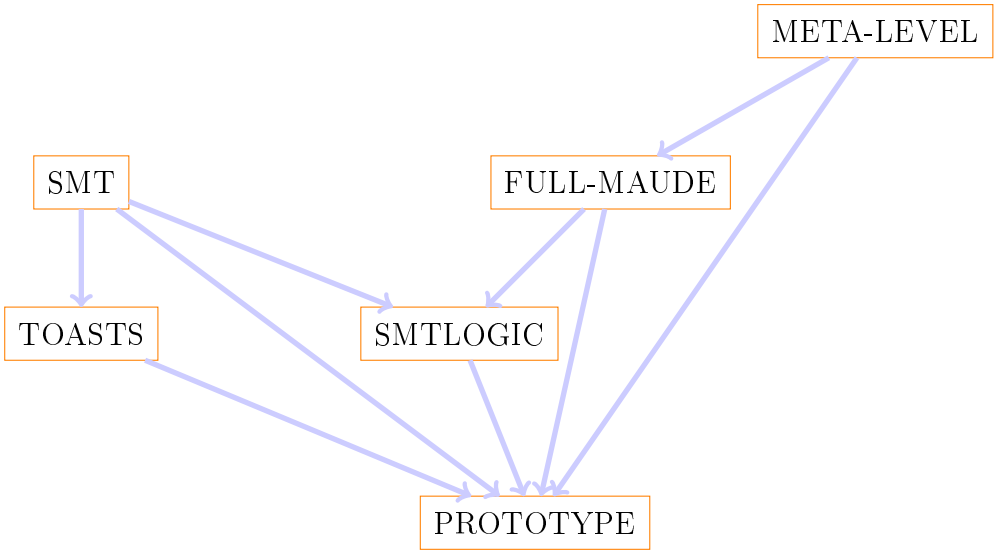

Figure 5.4: Dependencies between modules

Sort `RuleCondi`, together with the metarepresentation in `META-LEVEL` of the left and right side of the rule and the SMT constraint, is also used in the metarepresentation of the conditional rules in the normal form of R, R°:

```
op _&&_ : RuleCondi Term -> FullCondi [ctor] .
op _=>_if_ : Term Term FullCondi -> SmtRule [ctor] .
op kNone : -> KindIntSmtRuleSet [ctor] .
op _/_/_ : Kind Int SmtRule -> KindIntSmtRule [ctor] .
op __ : KindIntSmtRuleSet KindIntSmtRuleSet
        -> KindIntSmtRuleSet [ctor assoc comm id: kNone] .
```

The memoized set `normalRls`, with sort `KindIntSmtRuleSet`, includes the normal form of all the rules, together with their kind and serial number. The kind of each subproblem and rule is included in their signatures. In this way, the rules that do not have same kind as the subproblem that is being solved at a certain moment are not used by the search engines.

Prototype modular design

Figure 5.4 shows the dependency relation between the different modules used in the implementation.

Three constants must be given an appropriate value in the module `PROTOTYPE` before it is loaded. For the running example, we define:

```
eq target = 'TOASTS .
eq congruenceKinds1 = getKind(moduleNoRls, 'System) .
eq congruenceKinds2 = getKind(moduleNoRls, 'Kitchen) .
```

The values given to the constants mean that:

- the target module is `TOASTS`,
- rule `congruence` has to be applied only to terms whose sort is in the connected component of sort `System`, and

- when rule `congruence` is applied to a term, it will be applied only to the subterms whose sort is in the connected component of sort `Kitchen`.

The prototypes depend on the specification under test, in this case the module `TOASTS`, because the inclusion of the constant `target` allows us to memoize the normal form R° of the specification rules R and other constant parameters derived from it. In this way, the computations do not have to include the specification, or a transformed version of it, as part of the term that represents the current state in the rule-based prototype, or as part of the whole computation in the functional one, improving the performance of both prototypes.

The definition of the constants `congruenceKinds1` and `congruenceKinds2` is an important improvement that is explained in Section 5.5.4. Currently both constants are hardwired into the prototypes, being their computation from the specification one possible enhancement of the prototypes.

5.5.2 Rule-based prototype

The system module that implements the rule-based prototype is called `REW-NAR` (for 'narrowing using rewrite').

The operator `problem` of the prototype together with the operator `upTerm` of the module `META-LEVEL` of Maude are used to express in an easy way the reachability problems that we want to solve in our specification, and a module called `TEST` is used to load all the modules shown in Figure 5.4 in appropriate order.

Example 18. *The search command*

```
search [9, 90] in TEST :
problem(upTerm(1 / W / 0 ; zt zt / 0) =>* upTerm(N2 / zt / Y ; zt zt / 1)
        && upTerm(Y < 12)) =>* SOL .
```

asks for 9 solutions, with a depth limit of 90 steps in the search, to the reachability problem: is it possible from a `State` with just one `Toast` in the bag and another `Toast` (`W`) in the `Tray`, which may be an `EmptyToast`, `zt`, to reach a `State` such that there is one well-cooked `Toast`, the tray is empty, and the pan is empty, in less than 12 seconds?

The operator `upTerm` *returns the meta-representation of the initial and final states of the reachability problem, and also of the SMT constraint of the reachability problem.*

The variable `SOL` *has sort* `Solution`*, a subsort of sort* `Problem` *that the states of the computation in the rule-based prototype have when a solution to a reachability problem, with respect to the narrowing calculus, is found.*

The operator `problem` *returns as output a term P with sort* `Problem` *of the form* `K / L =>* R & RC || SC ; OVL ; NV ; AN`*, where* `OVL` *is the list of variables in P,* `NV` *is the next number of variable,* `AN` *is* `none`*, the computed answer so far, and* `K / L =>* R & RC || SC` *is the reachability problem to solve. Here,* `K / L =>* R` *represents the first subgoal of P, together with its kind* `K`*,* `RC` *holds the rest of subgoals of P and their respective kinds, and* `SC` *is the reachability formula of P.*

5.5.3 Functional prototype

The reduction engine of Maude, used with the equations, is more efficient than its rewrite engine, used with the rules. As the functional prototype does not use the search engine

of Maude, a functional module, called `RED-NAR` (for 'narrowing using reduce'), that has equations instead of rules, was developed. The functional prototype uses its own complete search engine, which in its initial version was SMT-based.

When we began testing possible improvements for the prototypes, some set-based versions of the functional prototype were developed, to check how much performance is lost by replacing the search engine of Maude with a functional version of it.

In the set-based versions of the functional prototype, a set of visited states is kept and each new generated state is compared against the elements of this set, using the set-matching capabilities of Maude, to verify whether it has already been visited or not.

Each set-based version of the functional prototype is a counterpart of some version of the rule-based prototype. The reason for the development of each version is explained later in this section.

As the rewrite engine of Maude is not used, then there is no need for a variable `SOL` to identify the end states of the reachability problems. On the other hand, in the functional prototype the number of answers and the depth of the search have to be included in the call to the operator `problem`, to control the end in the search for solutions using the reduction engine of Maude. The syntax of the operator `problem` becomes:

```
op problem : Nat Nat OrigPrCond -> State .
```

There are new sorts, aimed at implementing the search engine in the functional module:

```
op nilNP : -> NatProblem [ctor] .
op _|_ : Nat Problem -> NatProblem [ctor prec 81] .
op _||_ : NatPrList NatPrList -> NatPrList [ctor assoc id: nilNP prec 83] .
op __ : ProblemSet ProblemSet -> ProblemSet [ctor assoc comm id: nilP prec 81] .
op _/_/_/_/_ : Nat Nat NatPrList NatPrList ProblemSet -> State [ctor] .
```

Sort `NatProblem` associates to each problem the remaining depth of the search. If the remaining depth reaches zero then the problem is discarded. Sort `ProblemSet` keeps all visited problems. Sort `State` holds the full computation, which consists of the state number, the number of answers left to find, a list of current problems and a list of found solutions, both with sort `NatPrList`, and the set of already visited problems, with sort `ProblemSet`.

Then, reachability problems in the functional prototype have different syntax to those in the rule-based prototype. A similar problem to the one shown in Example 18 is:

Example 19. *The reduce command:*

```
red in TEST :
problem(3, 90, (upTerm(N / W / 0 ; zt zt / 0)
            =>* upTerm(0 / zt / Y ; zt zt / 2) && upTerm(Y < 12))) .
```

asks the prototype to find 3 solutions, using its own search engine and with a depth limit of 90 steps in the search, to the reachability problem: is it possible from a `State` with any number of `Toast`s in the bag and another `Toast` (`W`) in the `Tray`, which may be an `EmptyToast`, `zt`, to reach a `State` such that there are two well-cooked `Toast`s, the tray is empty, and the pan is empty, in less than 12 seconds?

Example 20. *An optimized version of the previous example could consist in the strengthening of its SMT constraint* `Y < 12`. *For instance, replacing it with the condition* `N < 3 and Y < 12` *tells the prototype that we do not care for solutions that involve a bag that has more than two* `Toast`s *in the initial state, since no feasible solution is possible in that case.*

Both versions are compared later in this section. The different performance of the same prototype with each of them shows the importance of having a deep understanding of both the specification and the problem under test to get the best results.

The operator `problemCore` generates the abstracted version of the problem; the constant `normalRls` holds in this prototype a normalized version of the rules where all the terms in the rules, instead of only the head, have been abstracted. This approach has also been tested in one version of the rule-based prototype.

The operator `processState` implements the search engine logic. It uses the operator `processNatPr` to generate the candidate children of a problem in the search tree, which are then pruned with operator `isNewProblem` that searches through the whole list of already visited problems using operator `sameProblem`:

```
ceq sameProblem((K / L' =>* R' & RC' || SC' ; OVL ; NV' ; AN'),
                (K / L =>* R & RC || SC ; OVL ; NV ; AN))
  = true if {SU, SU', NV''} := metaDisjointUnify(thisModule,
             upTerm(K / L =>* R & RC) =?
             upTerm(K / L' =>* R' & RC'), max(NV, NV'), 0) /\
     isRenaming(SU) /\ isRenaming(SU') /\
     metaCheck(smtModule, downSmt((SC <<* su2smtSu(SU, smtParams)) ===
               (SC' <<* su2smtSu(SU', smtParams)) , false)) .
eq sameProblem(PR, PR') = false [owise] .
```

Identical problems have been previously removed, using the multiset axioms of sort `ProblemSet`, with the equation:

```
eq processState(STATES, ANSWERS, DEPTH | PR || NP0, NPL, SOLS, PR PS)
  = processState(STATES, ANSWERS, NP0, NPL, SOLS, PR PS) .
```

5.5.4 Improvements in the prototypes

We present now several improvements, some of them already mentioned, that are included in the versions of the prototypes that we have developed. The idea of testing each of these improvements came to our mind during the different stages of the debugging of the prototypes, when the inspection of the generated states, including their SMT constraints, showed some flaw that prevented a better pruning of the state space, as a way to try to overcome that flaw.

The first improvement, the support for variable SMT parameters, aims at the enhancement of the kind of expressible problems within the prototypes and has already been explained. It is included in all versions of the prototypes. The rest of the improvements, explained below, aim at speed up the execution time. The relation between the versions of the prototypes and the improvements applied to each of them is shown in the next section in Tables 5.1 and 5.2.

1. Many times the congruence rule may be applied to a term of a given kind, trying to generate a narrowing step from a subterm or any of its proper subterms, each one having some sort, and there does not exist any rule for any of these sorts, so the narrowing step will not be generated. To prevent this issue, the prototypes may include as constants two sets of kinds:

   - one set holds the kinds of terms where the congruence rule can be applied,

- the other set holds the kinds of the subterms where the congruence rule can be applied.

These sets are different for every given specification. Currently they are hardwired into the prototypes, as previously explained. The constant `allKinds`, that holds all the kinds of the specification, can be used to give value to both constants, guaranteeing the completeness of the narrowing process. This speed up improvement is included in all the versions of the prototypes but the first one, that serves as reference.

2. Reachability problems that are semantically equal may not be identified as such by the search engine in Maude. The new variables added by the narrowing engines have the form $\#n$, with n increasing over time without limit. One of the approaches that can be used to deal with this issue involves two alternative strategies:

   (a) rename the new variables in the reachability problems that we obtain after each narrowing step, using the names $\#1, \#2, \ldots$, trying to generate a canonical version of the problems where the highest variable number in each new problem is limited by the number of new variables in it, or

   (b) use an SMT-based engine, that keeps a list of all generated reachability problems and tries to identify every new generated reachability problem as already known by variable renaming and checking for SMT equivalence of the SMT constraints.

3. Another approach that can be used, together with 2(a) to remove semantically equal states involves the simplification of the SMT subterms in the reachability problems after each application of the unification or rewrite rules. Each one of these SMT subterms is turned into a corresponding `Int`, `Rat`, or `Bool` term. We let Maude simplify the term, and then the obtained term is turned back into a term with SMT sorts.

4. The following simplifications, besides the above-mentioned, can be applied to the SMT constraint of a reachability problem after each application of the unification or rewrite rules:

   - since the SMT constraint is satisfiable and has the form $\bigwedge_i t_i$, by definition of both rules, if any of the terms t_i has sort `GroundSmtCondi` then it must be valid, and it can be safely removed without affecting the satisfiability of the SMT constraint,
   - remove the temporal variables, of the form $\#n$, from the SMT constraint if their value in the computed solution so far is another variable,
   - extract the SMT constraints of the form 'metavariable = ground arithmetic metaterm' as assignments in the computed solution, and
   - extract the SMT constraints of the form 'metavariable = ground SMT metacondition' as assignments of the form $v \mapsto \mathit{true}$ or $v \mapsto \mathit{false}$ in the computed solution. The ground SMT metacondition is converted into its corresponding ground SMT constraint, Maude reduces this condition either to *true* or to *false*, and this is the value assigned to the variable.

When a new assignment is generated, this new assignment is also applied to the rest of the problem, thus getting rid of all the instances of the metavariable being removed.

As a full simplification of the SMT constraints, which is better for pruning the search space, may involve a big number of rewrites, slowing down the generation of new states, two alternative strategies are used in the versions of the prototypes to check their performance:

(a) always apply full simplification after each calculus step, or

(b) partial simplification, removing some costly steps, for new reachability problems and full simplification for every found solution, since full simplification improves its readability and it will be only called for a few final states.

5. A call to the SMT solver is only needed when the SMT constraint has changed, which may not happen all the times. Some versions of the prototypes can control these changes and act consequently.

6. Full abstraction of the reachability problems and rules is required by the SMT-based search engine. It is also implemented in some versions of the prototypes that do not have this engine to check the overhead caused by this modification.

7. As calling the SMT solver is an expensive operation, we also developed lazy versions of both prototypes that called the SMT solver after each unification step, but not after any rewrite step, allowing faster state generation at the cost of generating unfeasible states that will be later discarded.

5.5.5 Testing the prototypes

The versions of the rule-based prototype have the form `REW-NAR-?` and the versions of the functional one have the form `RED-NAR-?`, where the ending character `?` is a letter. These names refer to Maude's reserved words `rewrite`, used in system modules, and `reduce`, used in functional modules.

Table 5.1 and Table 5.2 show the improvements implemented in each version of the rule-based and the functional prototypes, respectively. We use the same ending letter on both tables when the set of improvements selected for the corresponding versions of the prototypes is also the same.

We explain now the motivation for the improvements selected in each version of the rule-based prototype:

- `REW-NAR-A`: implements renaming of variables (2a), simplification of problems (3), and full simplification of SMT constraints (4a). It is the only version without the optimization in the use of the congruence rule (1), and it serves as a reference for the other versions of the rule-based prototype.

- `REW-NAR-B`: during the testing of `REW-NAR-A` we came up with the idea of the optimization in the use of the congruence rule (1), so we added it. It also seemed to us that the overhead caused by the simplification of the reachability problem (3) in each new state exceeded the benefits of the extra already visited states removed, so we excluded this simplification.

Table 5.1: REW-NAR versions

Improvements		**Version**					
Number	**Description**	**A**	**B**	**C**	**D**	**E**	**F**
1	congruenceKinds		**X**	**X**	**X**	**X**	**X**
2a	renaming of variables	**X**	**X**	**X**	**X**	**X**	**X**
3	simplification of problems	**X**					
4a	full simplification of SMT constraints	**X**	**X**				
4b	partial simplification of SMT constraints			**X**	**X**	**X**	**X**
5	SMT check only for modified conditions				**X**		
6	full abstraction of problems and rules						**X**
7	SMT check only for unification					**X**	

Table 5.2: RED-NAR versions

Improvements		**Version**				
Number	**Description**	**C**	**E**	**F**	**G**	**H**
1	congruenceKinds	**X**	**X**	**X**	**X**	**X**
2a	renaming of variables	**X**	**X**	**X**		
2b	SMT-based search engine				**X**	**X**
4b	partial simplification of SMT constraints	**X**	**X**	**X**	**X**	**X**
5	SMT check only for modified conditions					**X**
6	full abstraction of problems and rules			**X**	**X**	**X**
7	SMT check only for unification		**X**			

The comparison of their performance showed that REW-NAR-B was way better so we decided to keep trying to improve it and discard REW-NAR-A.

- REW-NAR-C: we also wanted to check whether full simplification of SMT constraints in REW-NAR-B was better than partial simplification or not, due to the overhead caused by full simplification, so we developed a version implementing partial simplification of SMT constraints (4b).

  The comparison of their performance showed that REW-NAR-C was better (about 10%) both in time and number of rewrites, while generating few additional states, so we decided to keep trying to improve it and discard REW-NAR-B.

  Taking as base REW-NAR-C, we developed the rest of the versions of the rule-based prototype, each of them implementing a different improvement.

- REW-NAR-D: when testing REW-NAR-C we came to see that many times we were checking the satisfiability of a SMT constraint that we had already checked, so we implemented a version that checks for SMT satisfiability only when the SMT constraint has changed (5).

- REW-NAR-E: we added the check for SMT satisfiability only when the rule unification is applied (7) to REW-NAR-C, because we wanted to test a lazy version of the calculus that generates a possible solution for each subproblem before calling the SMT solver.

- REW-NAR-F: as full abstraction of problems and rules (6) is used in the functional prototype, we added the same abstraction to REW-NAR-C, to check the overhead introduced by this approach.

The functional prototype only has two versions where the SMT-based rewrite engine is implemented, RED-NAR-G and RED-NAR-H. The other versions use a simpler set-based rewrite engine, as explained before. A pair of versions, one of each prototype, that has the same ending letter also has the same set of improvements implemented, allowing us to compare the performance of both prototypes under the same setting.

The versions of the functional prototype are:

- RED-NAR-C: reference for the rest of the versions of the functional prototype. It includes the optimization in the use of the congruence rule (1), the renaming of variables (2a), and the partial simplification of SMT constraints (4b). We chose REW-NAR-C as model, since it was one of the best versions of the rule-based prototypes, and also the base for several other versions.

- RED-NAR-E: as in the system modules, we added the check for SMT satisfiability only when the rule unification is applied (5) to RED-NAR-C.

- RED-NAR-F: full abstraction of problems and rules (6) is used in the rest of the versions, so the first step was to add this improvement to RED-NAR-C.

- RED-NAR-G: the SMT-based search engine (2b) is added to RED-NAR-F. Renaming of variables (2a) is removed, since the unification used in (2b) makes this renaming unnecessary.

- RED-NAR-H: as in the case of REW-NAR-D we added the check for SMT satisfiability only when the SMT constraint has changed (5) to RED-NAR-G.

The versions have been tested with the problems shown in Examples 18, 19, and 20, using two compiled versions of Maude alpha 115, one with CVC4 and the other with Yices as SMT solvers. The compiled version with Yices has been consistently faster in all the tests, so only the times for this version are shown.

Table 5.3: Performance for the problem of Example 18

	7 Answers			**9 Answers**		
Version	**States**	**Rewrites**	**Time(s)**	**States**	**Rewrites**	**Time(s)**
REW-NAR-A	11.440	4.552.431	5,471	20.745	9.148.848	10,355
REW-NAR-B	5.507	2.573.367	2,968	9.443	4.950.374	5,647
REW-NAR-C	5.516	2.332.723	2,770	9.460	4.393.577	5,163
REW-NAR-D	**5.516**	**2.330.945**	**2,779**	**9.460**	**4.391.021**	**5,227**
REW-NAR-E	36.834	14.345.972	13,015	73.108	29.882.513	27,598
REW-NAR-F	6.303	3.334.769	3,911	11.650	6.841.007	8,506
RED-NAR-C	**5.508**	**2.346.495**	**3,488**	**9.452**	**4.418.403**	**7,103**
RED-NAR-E	36.866	14.685.756	55,203	-	-	-
RED-NAR-F	6.344	3.366.954	4,606	11.723	6.871.891	10,715
RED-NAR-G	5.151	5.265.088	92,628	-	-	-
RED-NAR-H	5.151	5.264.014	92,520	-	-	-

Table 5.3 shows the performance of the different versions of the prototypes when they are required to find seven and nine answers for the reachability problem in Example 18. Sometimes the result of the search for nine answers is not shown, meaning that the running time was over 1.000 seconds and we stopped the search.

Table 5.4 shows the performance of a selection from the prototypes in Table 5.3 when they are required to find one or three answers for the original reachability problem in Example 19 and also for the optimized reachability problem in Example 20.

The versions of the rule-based prototype that have a counterpart version in the functional one are those ending with 'C', 'E', and 'F'. For any problem that we have checked with any of these versions, the number of rewrites and states generated is quite similar in both prototypes, as expected. The difference in their running time, always in favor of the prototype that uses the rewrite engine of Maude, is the one expected from the use of multisets in the meta-search engine, instead of the C++ optimized structures used by the search engine of Maude.

The versions of the functional prototype that use the SMT-based search engine, those ending with 'G', and 'H' are the ones that generated less states, the suspicion behind their development, at the cost of more rewrites and their, expected, very bad running time. We could get a better measure of the real value of this approach if a SMT-based search engine were added to Maude, but the results obtained with these versions of the prototype suggest that a deeper pruning of the state space could be accomplished by using this kind of engines.

Let's review the data in Table 5.3:

- The first two data rows, `REW-NAR-A` vs `REW-NAR-B`, show the effect of the optimization in the use of the congruence rule even when the simplification of problems is removed. When we ask for 9 answers we get over 50% reduction in the number of states generated (9.443 vs 20.745) and over 45% reduction in running time (5,647 vs 10,355 seconds).

- `REW-NAR-C`, with partial simplification of the SMT constraints, shows a slight increase in the number of states (9.460 vs 9.443) with respect to `REW-NAR-B`, that has full simplification of the SMT constraints, because less simplification implies worse pruning of the search space, but it also shows better running time performance (5,163 vs 5,647 seconds) due to around a 10% reduction in the number of rewrites (4.393.577 vs 4.950.374), all figures from the 9 answers query.

- `REW-NAR-D` is like `REW-NAR-C` with the addition of checking for SMT satisfiability only when the SMT constraint has changed. When we compare both, `REW-NAR-D` always generates the same number of states, as expected, but although it requires less rewrites (4.391.021 vs 4.393.577) it is slightly slower (5,227 vs 5.163 seconds), all figures from the 9 answers query.

  As we will see in Table 5.4, when the search space gets bigger `REW-NAR-D` is faster than `REW-NAR-C` due to the lower amount of calls to the SMT solver.

- The lazy approach in `REW-NAR-E` and `RED-NAR-E`, which are like their corresponding 'C' versions but where we only check for SMT satisfiability when the rule unification is applied, does not work well, because even for this simple specification it allows the generation of a big number of states, more than 14 million in the 7 answers query, that are pruned to 2,3 million with the aid of the SMT solver in the 'C' versions of the prototypes.

- `REW-NAR-F` and `RED-NAR-F`, which are like their corresponding 'C' versions but with full abstraction of problems and rules, show that the number of visited states increases, over 10% (from around 5.500 to over 6.300) for finding 7 answers and over 20% (from around 9.450 to over 11.600) for finding 9 answers, i.e., the increase is nonlinear.

- In `RED-NAR-G` the combined effect of full abstraction, as in `RED-NAR-F`, and the use of the SMT-based search engine generates the smallest number of states for finding 7 answers, a 6% reduction at least with respect to all the other versions (5.151 vs 5.508 in `RED-NAR-C`).

- The improvement added in `RED-NAR-H` to `RED-NAR-G`, SMT check after each unification or rewrite step only for modified conditions, shows a negligible reduction both in number of rewrites and running time.

We show the performance of `RED-NAR-C` also in Table 5.4 to compare the performance of both prototypes when similar versions of each one are used.

It can be seen in Table 5.4 that the running time of the functional prototype `RED-NAR-C` is not competitive against the rule-based prototype versions, as it happened in Example 18.

Also, the results of the comparison of the performance of the different versions of the rule-based prototype in each of the examples in Table 5.4 are similar to those in Table 5.3, so we do not need to add anything in this respect.

What is important about Table 5.4 is that it shows the importance of having a deep understanding of the problems that we want to solve: formulating in Example 20 the problem from Example 19 but with more precision has a big impact in the performance of all the versions of the prototypes.

Table 5.4: Performance for the problems of Examples 19 and 20

Example 19	**1 Answer**			**3 Answers**		
Version	**States**	**Rewrites**	**Time(s)**	**States**	**Rewrites**	**Time(s)**
REW-NAR-B	75.794	46.899.795	55,400	124.178	81.209.130	97,929
REW-NAR-C	75.977	42.744.050	52,723	124.444	73.345.384	92,898
REW-NAR-D	**75.977**	**42.788.069**	**52,175**	**124.444**	**73.422.931**	**92,177**
RED-NAR-C	75.834	42.594.805	334,544	124.257	73.103.677	939,022
Example 20	**1 Answer**			**3 Answers**		
REW-NAR-B	35.979	22.447.879	26,832	55.789	36.742.261	43,476
REW-NAR-C	36.074	20.057.910	24,951	55.918	32.471.854	40,670
REW-NAR-D	**36.074**	**20.084.576**	**24,200**	**55.918**	**32.515.258**	**39,876**
RED-NAR-C	36.018	20.063.240	69,392	55.855	32.494.540	158,288

Giving more information about the desired search space in Example 20 results in at least a 50% gain in performance for the versions of the rule-based prototype with respect to Example 19. In the case of faster version for this problem `REW-NAR-D`, for the 3 answers query there is an improvement of over 55% in the number of generated states (55.918 vs. 124.444), rewrites (32.5 million vs 73.4 million), and running time (39,876 vs 92,177 seconds).

For the functional prototype `RED-NAR-C`, the gain of performance in running time is around 80% (69,392 vs 334,544 seconds) for 1 answer and over 83% (158,288 vs 939,022 seconds) for 3 answers, showing the increasing overhead of using the meta-search engine as the set of generated states gets bigger.

As a conclusion, the version that has had better performance is `REW-NAR-D`, but an implementation in C++ of a SMT-based search engine like the one in `RED-NAR-H` would presumably be a heavy contender due to the space state reduction for `RED-NAR-H` shown in Table 5.3.

5.6 Results and proofs

This section holds some needed technical results for this chapter, together with the proofs for all the results.

Proposition 12 (Relation between Σ-terms and abstractions in rewrite theories)**.** *Let $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ be a rewrite theory with built-in subtheory (Σ_0, E_0), and t be a term in $\mathcal{H}_\Sigma(\mathcal{X})$, with $abstract_{\Sigma_1}(t) = \langle \lambda \bar{x}.t^\circ; \theta^\circ; \phi^\circ \rangle$. For any substitution σ such that $E_0 \vdash \phi^\circ\sigma$, it follows that $t^\circ\sigma =_{\mathcal{E}} t\sigma$.*

Proof. By definition $\theta^\circ = \{x^1_{\kappa_1} \mapsto t_1, \ldots, x^n_{\kappa_n} \mapsto t_n\}$, $\phi^\circ = \bigwedge_{i=1}^n x^i_{\kappa_i} = t_i$, $t = t[t_1]_{p_1} \cdots [t_n]_{p_n}$, and $t^\circ = t[x^1_{\kappa_1}]_{p_1} \cdots [x^n_{\kappa_n}]_{p_n}$. Also, as $E_0 \vdash \phi^\circ\sigma$ then $x^i_{\kappa_i}\sigma =_{E_0} t_i\sigma$, for $1 \leq i \leq n$, so $t^\circ\sigma = (t[x^1_{\kappa_1}]_{p_1} \cdots [x^n_{\kappa_n}]_{p_n})\sigma = t\sigma[x^1_{\kappa_1}\sigma]_{p_1} \cdots [x^n_{\kappa_n}\sigma]_{p_n} =_{E_0}$
$t\sigma[t_1\sigma]_{p_1} \cdots [t_n\sigma]_{p_n} = (t[t_1]_{p_1} \cdots [t_n]_{p_n})\sigma = t\sigma$.

As the theory inclusion is protecting then also $t^\circ\sigma =_{\mathcal{E}} t\sigma$. □

Proposition 13 (Invariants of top_{Σ_0} under E_0-deduction)**.** *Let $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ be a rewrite theory with built-in subtheory (Σ_0, E_0). If t and t' are two terms in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $t \leftrightarrow_{E_0} t'$ then:*

1. $top_{\Sigma_0}(t) = top_{\Sigma_0}(t')$,
2. $ls(t|_q) = ls(t'|_q)$ *and* $t|_q =_{E_0} t'|_q$ *for all positions* q *in* $top_{\Sigma_0}(t)$,
3. $t|_{q'} =_{E_0} t'|_{q'}$ *for all positions* q' *such that* $t|_{q'} \in \mathcal{H}_\Sigma(\mathcal{X})$, *and*
4. *if* $top_{\Sigma_0}(t) = \{q_1, \ldots, q_n\}$ *then* $t' = t[t'|_{q_1}]_{q_1} \cdots [t'|_{q_n}]_{q_n}$.

Proof. As $t \in \mathcal{H}_\Sigma(\mathcal{X})$ and $t|_p \in \mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$, there must exist a position p' in $top_{\Sigma_0}(t)$ such that $p' \leq p$, so $p = p'.r$ for suitable r. The first two points are proved simultaneously.

1. and (2) As p' in $top_{\Sigma_0}(t)$, then $p' = p''.i$ for suitable position p'' and natural number i, and $t|_{p''} \in \mathcal{H}_\Sigma(\mathcal{X})$, so $t' = t[w\sigma]_{p'.r} = t[w\sigma]_{p''.i.r}$. If q in $top_{\Sigma_0}(t)$ and $q \neq p'$ then neither $q \leq p'$ nor $p' \leq q$, so $t|_q$ is unaffected by the E_0-deduction step. Then $t|_q = t'|_q$, hence $ls(t|_q) = ls(t'|_q)$ and $t|_q =_{E_0} t'|_q$, so q in $top_{\Sigma_0}(t')$.

   For $top_{\Sigma_0}(t)$ and $top_{\Sigma_0}(t')$ to be equal it must be proved that $p' \in top_{\Sigma_0}(t')$. As (Σ_0, E_0) is many-sorted then $ls(v\sigma) = ls(w\sigma)$, so we are replacing in $t|_{p'}$ the subterm $t|_{p'.r}$, having the value $v\sigma$, with the subterm $w\sigma$ both of them with the same sort, so also $t|_{p'}$ and $t|_{p'}[w\sigma]_r$ have the same sort in S_0. As $t|_{p''} = (t[v\sigma]_{p''.i.r})|_{p''} \in \mathcal{H}_\Sigma(\mathcal{X})$, and $v\sigma$ and $w\sigma$ have the same sort then also $(t[w\sigma]_{p''.i.r})|_{p''} \in \mathcal{H}_\Sigma(\mathcal{X})$, so $p' \in top_{\Sigma_0}(t')$. As $t \leftrightarrow_{E_0} t[w\sigma]_p = t'$ and $p = p'.r$, then $t'|_{p'} = t|_{p'}[w\sigma]_r$ and $t|_{p'} \leftrightarrow_{E_0} t|_{p'}[w\sigma]_r = t'|_{p'}$, so $t|_{p'} =_{E_0} t'|_{p'}$.

3. If q' is a position such that $t|_{q'} \in \mathcal{H}_\Sigma(\mathcal{X})$ then:

   - $p' \not\leq q'$, because as p' in $top_{\Sigma_0}(t)$ if $p' \leq q'$ then $t|_{q'} \in \mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$, in contradiction with $t|_{q'} \in \mathcal{H}_\Sigma(\mathcal{X})$,
   - if $q' = p'$ then $t|_{q'} = t'|_{q'}$, so $t|_{q'} =_{E_0} t'|_{q'}$, and
   - if $q' < p'$ then $p' = q'.r'$, for suitable position r', so $p = p'.r = q'.r'.r$. As $t' = t[w\sigma]_p$ then $t'|_{q'} = t|_{q'}[w\sigma]_{r'.r}$, and as $t|_p = t|_{q'.r'.r} = v\sigma$, then $t|_{q'} = t|_{q'}[v\sigma]_{r'.r}$, so $t|_{q'} \leftrightarrow_{E_0} t'|_{q'}$ with the same equation c and substitution σ at position $r'.r$, and $t|_{q'} =_{E_0} t'|_{q'}$.

4. As p' in $top_{\Sigma_0}(t)$, without loss of generality take $p' = q_1$, so $t'|_{q_j} = t|_{q_j}$, for $2 \leq j \leq n$. $t' = t[w\sigma]_{p'.r} = t[w\sigma]_{q_1.r} = t[t|_{q_1}[w\sigma]_r]_{q_1}$. As $t'|_{q_1.r} = w\sigma$ then $t'|_{q_1} = t|_{q_1}[w\sigma]_r$, so $t' = t[t'|_{q_1}]_{q_1}$. As $t = t[t|_{q_1}]_{q_1}[t|_{q_2}]_{q_2} \cdots [t|_{q_n}]_{q_n} = t[t|_{q_1}]_{q_1}[t'|_{q_2}]_{q_2} \cdots [t'|_{q_n}]_{q_n}$ and $t' = t[t'|_{q_1}]_{q_1}$, then $t' = (t[t|_{q_1}]_{q_1}[t'|_{q_2}]_{q_2} \cdots [t'|_{q_n}]_{q_n})[t'|_{q_1}]_{q_1} = t[t'|_{q_1}]_{q_1}[t'|_{q_2}]_{q_2} \cdots [t'|_{q_n}]_{q_n}$.

□

Proposition 7 (Invariants of top_{Σ_0} under E_0-equality)**.** *Let* $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ *be a rewrite theory with built-in subtheory* (Σ_0, E_0). *If* t *and* t' *are two terms in* $\mathcal{H}_\Sigma(\mathcal{X})$ *such that* $t =_{E_0} t'$ *then:*

1. $top_{\Sigma_0}(t) = top_{\Sigma_0}(t')$,
2. $ls(t|_q) = ls(t'|_q)$ *and* $t|_q =_{E_0} t'|_q$ *for all positions* q *in* $top_{\Sigma_0}(t)$,
3. $t|_{q'} =_{E_0} t'|_{q'}$ *for all positions* q' *such that* $t|_{q'} \in \mathcal{H}_\Sigma(\mathcal{X})$, *and*
4. *if* $top_{\Sigma_0}(t) = \{q_1, \ldots, q_n\}$ *then* $t' = t[t'|_{q_1}]_{q_1} \cdots [t'|_{q_n}]_{q_n}$.

Proof. Immediate since either $t = t'$ and all invariants follow trivially, or induction can be used in combination with Proposition 13, since $t =_{E_0} t'$ can be seen as a series of one-step deductions $t \leftrightarrow_{E_0} t_1 \cdots \leftrightarrow_{E_0} t_n \cdots \leftrightarrow_{E_0} t'$ for suitable t_i, $1 \leq i \leq n$. □

Proposition 8 (Relation between $abstract_{\Sigma_1}$ and top_{Σ_0})**.** *Let $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ be a rewrite theory with built-in subtheory (Σ_0, E_0). If t is a term in $\mathcal{H}_\Sigma(\mathcal{X})$, $abstract_{\Sigma_1}(t, \mathcal{Y}) = \langle \lambda \bar{x}.t^\circ; \theta^\circ; \phi^\circ \rangle$, where $\bar{x} = \{x_1, \ldots, x_n\}$ and $t^\circ = t[x_1]_{q_1} \cdots [x_n]_{q_n}$, then*

(i) $top_{\Sigma_0}(t) = \{q_1, \ldots, q_n\}$, and

(ii) for every substitution $\sigma : \bar{x} \to \mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$ it holds that $top_{\Sigma_0}(t^\circ\sigma) = top_{\Sigma_0}(t)$.

Proof. By definition of $abstract_{\Sigma_1}$, $t^\circ\theta^\circ = t$, $vars(t^\circ) \cap \mathcal{X}_0 = \bar{x}$, t° in $\mathcal{H}_\Sigma(\mathcal{X})$, $t|_{q_i}$ in $\mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$, and $x_i\theta^\circ = t|_{q_i}$, for $1 \leq i \leq n$.

(i) $top_{\Sigma_0}(t) = \{q_1, \ldots, q_n\}$:

1.- $\{q_1, \ldots, q_n\} \subseteq top_{\Sigma_0}(t)$.
For $1 \leq i \leq n$, as t in $\mathcal{H}_\Sigma(\mathcal{X})$ and $t|_{q_i}$ in $\mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$, then $q_i \neq \epsilon$, so $q_i = q_i'.j_i$, for suitable position q_i' and integer j_i. As $t^\circ = t[x_1]_{q_1} \cdots [x_n]_{q_n} \in \mathcal{H}_\Sigma(\mathcal{X})$ and $q_i' < q_i$, then $t^\circ|_{q_i'}$ is a term in $\mathcal{H}_\Sigma(\mathcal{X})$. Now, as $\theta^\circ : \bar{x} \to \mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$ then $t^\circ|_{q_i'}\theta^\circ$ is a term in $\mathcal{H}_\Sigma(\mathcal{X})$. But $t^\circ|_{q_i'}\theta^\circ = t^\circ\theta^\circ|_{q_i'} = t|_{q_i'}$, so $t|_{q_i'}$ in $\mathcal{H}_\Sigma(\mathcal{X})$, and q_i in $top_{\Sigma_0}(t)$.

2.- $top_{\Sigma_0}(t) \subseteq \{q_1, \ldots, q_n\}$.
Let p be a position in $top_{\Sigma_0}(t)$. By definition $t|_p$ in $\mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$ and $t = t[t|_p]_p$.

(a) $t^\circ = t[x_1]_{q_1} \cdots [x_n]_{q_n} = t[t|_p]_p[x_1]_{q_1} \cdots [x_n]_{q_n}$ if $p \not\leq q_i$ and $q_i \not\leq p$, for $1 \leq i \leq n$. As $t|_p$ in $\mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$ then either t° is not a term in $\mathcal{H}_\Sigma(\mathcal{X})$ or $vars(t^\circ) \cap \mathcal{X}_0 \neq \bar{x}$, both in contradiction with t° being an abstraction.

(b) If $q_i < p$ for some $1 \leq i \leq n$, then $p = q_i.q_i'.j_i$ for suitable (possibly empty) position q_i' and integer j_i. As $x_i\theta^\circ = t|_{q_i} \in \mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$ then $t|_{q_i.q_i'} \in \mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$ so $p = q_i.q_i'.j_i$ is not a position in $top_{\Sigma_0}(t)$.

(c) If $Q = \{q_i \mid q_i \in \{q_1, \ldots, q_n\} \wedge p < q_i\}$ is non-empty, so $Q = \{q_{i_1}, \ldots, q_{i_m}\}$, then $q_{i_j} = p.q_{i_j}'$, for suitable $q_{i_j}' \neq \epsilon$, for $1 \leq j \leq m$. As $t|_p$ in $\mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$ then $t^\circ|_p = t|_p[x_{i_1}]_{q_{i_1}'} \cdots [x_{i_m}]_{q_{i_m}}$ is a term in $\mathcal{T}_{\Sigma_0}(\mathcal{X}_0) \setminus \mathcal{X}_0$, so t° is not a term in $\mathcal{H}_\Sigma(\mathcal{X})$.

The only possibility left is $p = q_i$ for some $1 \leq i \leq n$, so $p \in \{q_1, \ldots, q_n\}$.

(ii) If $\sigma : \bar{x} \to \mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$ then $top_{\Sigma_0}(t^\circ\sigma) = top_{\Sigma_0}(t)$.

1.- $top_{\Sigma_0}(t^\circ\sigma) \subseteq top_{\Sigma_0}(t)$
For $1 \leq i \leq n$, $x_i\sigma \in \mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$. As $vars(t^\circ) \cap \mathcal{X}_0 = \bar{x}$ and $t^\circ|_p \in \mathcal{H}_\Sigma(\mathcal{X}) \setminus \mathcal{X}_0$ if $p \in Pos(t^\circ) \setminus \{q_1, \ldots, q_n\}$, then if $q \in top_{\Sigma_0}(t^\circ\sigma)$ there must exist some i, with $1 \leq i \leq n$, such that $q_i \leq q$. But as $t^\circ\sigma|_{q_i} = x_i\sigma \in \mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$ then $q_i \not< q$, so $q_i = q$ and $q \in top_{\Sigma_0}(t)$.

2.- $top_{\Sigma_0}(t) \subseteq top_{\Sigma_0}(t^\circ\sigma)$
$top_{\Sigma_0}(t) = \{q_i\}_{i=1}^n$. For $1 \leq i \leq n$ there exists a position q_i' and an integer j_i such that $q_i = q_i'.j_i$; also as $x_i\sigma \in \mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$ then $(t^\circ\sigma)|_{q_i} = x_i\sigma \in \mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$. Let $\{q_{i_1}, \ldots, q_{i_m}\} = \{q_j \mid q_i' < q_j, 1 \leq j \leq n\}$. Then $q_{i_k} = q_i'.p_{i_k}$ for suitable $p_{i_k} \neq \epsilon$, for $1 \leq k \leq m$, so $t|_{q_i'} = t|_{q_i'}[t|_{q_{i_1}}]_{p_{i_1}} \cdots [t|_{q_{i_m}}]_{p_{i_m}}$ and, as

$t^\circ|_{q'_i} = t|_{q'_i}[x_{i_1}]_{p_{i_1}} \cdots [x_{i_m}]_{p_{i_m}}$, then $t^\circ\sigma|_{q'_i} = t|_{q'_i}[x_{i_1}\sigma]_{p_{i_1}} \cdots [x_{i_m}\sigma]_{p_{i_m}}$. Σ_0 is many sorted, σ is well-formed, and $ls(x_{i_k}) = ls(t|_{q_{i_k}})$ by definition of $abstract_{\Sigma_1}$, so $ls(x_{i_k}\sigma) = ls(x_{i_k}) = ls(t|_{q_{i_k}})$, for $1 \leq k \leq m$. Then $ls(t|_{q'_i}) = ls(t^\circ\sigma|_{q'_i})$, so $t^\circ\sigma|_{q'_i} \in \mathcal{H}_\Sigma(\mathcal{X})$ and $q_i \in top_{\Sigma_0}(t^\circ\sigma)$.

□

Proposition 9 (Bijection between top_{Σ_0} positions in B-equal terms)**.** *Given an OS equational theory $\mathcal{E} = (\Sigma, E_0 \cup B)$ and two terms u and v in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $u =_B v$, where $u = u_0 \overset{ax_1}{\longleftrightarrow}_B \cdots \overset{ax_n}{\longleftrightarrow}_B u_n = v$, $\overline{ax} = ax_1, \ldots, ax_n$, with $\hat{ax} \subset B \cup B^{-1}$, if $top_{\Sigma_0}(u) = \hat{p}$ and $top_{\Sigma_0}(v) = \hat{q}$ then there exists a bijective function $dest_{\overline{ax}} : \hat{p} \to \hat{q}$ such that $u|_{p_i} = v|_{dest_{\overline{ax}}(p_i)}$, for each position p_i in $\hat{p}$.*

Proof. We inductively define the function $dest_l$ that tracks the final position of a subterm for a list of axioms $l = a_1, \ldots, a_m$. Given a position p':

1. $dest_{nil}(p') = p'$,
2. for a_1 in $B \cup B^{-1}$ with the form $f[\bar{x}]_{\bar{q}} = f'[\bar{x}]_{\bar{r}}$, where $vars(f[\bar{x}]_{\bar{q}}) = vars(f'[\bar{x}]_{\bar{r}}) = \hat{x}$, if $p' = q_j.s_j$, with q_j in $\hat{q}$, then $dest_{a_1}(p') = r_j.s_j$, else $dest_{a_1}(p') = p'$, and
3. for $l = a_1, \ldots, a_m$, with $m > 1$, if $dest_{a_1}(p') = p''$ then $dest_l(p') = dest_{a_2,\ldots,a_m}(p'')$.

As, by definition, the axioms in B are regular, linear, and only have function symbols from F_1, then in each step $u_{i-1} \overset{ax_i}{\longleftrightarrow}_B u_i$, $1 \leq i \leq n$, if ax_i has the form $f[\bar{x}]_{\bar{q}} = f'[\bar{x}]_{\bar{r}}$, where $vars(f[\bar{x}]_{\bar{q}}) = vars(f'[\bar{x}]_{\bar{r}}) = \hat{x}$ and it is used in a subterm $u_{i-1}|_p$ then:

- if ax_i moves a subterm in a position $p.q_j$ from $top_{\Sigma_0}(u_{i-1})$, where q_j in $\hat{q}$, with parent in F_1 since ax_i has only symbols in F_1, then the subterm is moved to the position $p.r_j$, with parent also in F_1 for the same reason as before, hence it remains a top_{Σ_0} position,
- if ax_i moves a subterm t in a position $p.q_j.s_j.k_j$ from $top_{\Sigma_0}(u_{i-1})$, where q_j in $\hat{q}$, s_j may be ϵ, k_j is an integer, and the parent of t in position $p.q_j.s_j$ is a function symbol f'' from F_1, then t is moved to the position $p.r_j.s_j.k_j$, where its parent at position $p.r_j.s_j$ is the same function symbol f'' from F_1, since f'' is also moved by ax_i from $p.q_j.r_j$ to $p.q_j.s_j$, hence it remains a top_{Σ_0} position,
- the rest of positions in $top_{\Sigma_0}(u_{i-1})$ remain unchanged.

Then $dest_{\overline{ax}}$ is injective, by its definition, and it also has to be surjective, since any position in $\hat{q}$ not in the image of $dest_{\overline{ax}}$ could be always related to a single position in $\hat{p}$ just by using the list of axioms ${ax_n}^{-1}, \ldots, {ax_1}^{-1}$, all of them in $B \cup B^{-1}$, a contradiction with $dest$ being total and surjective. We will write $dest$ instead of $dest_{\overline{ax}}$ when $\overline{ax}$ is irrelevant, homomorphically extend the definition of $dest$ to lists and sets of positions, and define $orig = dest^{-1}$. □

Corollary 2 (Bijection between top_{Σ_0} positions in $\mathcal{E}$-equal terms)**.** *Given an OS equational theory $\mathcal{E} = (\Sigma, E_0 \cup B)$ and two terms u and v in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $u =_\mathcal{E} v$, if $top_{\Sigma_0}(u) = \hat{p}$ and $top_{\Sigma_0}(v) = \hat{q}$ then there exists a bijective function $dest : \hat{p} \to \hat{q}$, hence $\hat{q} = dest(\hat{p})$, such that $u|_{p_i} =_{E_0} v|_{dest(p_i)}$, for each position p_i in $\hat{p}$.*

Proof. As $u =_{\mathcal{E}} v$ then, by Proposition 10, there exists a term w in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $u =_{E_0} w =_B v$. As $u =_{E_0} w$ then, by Proposition 7, $top_{\Sigma_0}(u) = top_{\Sigma_0}(w) = \hat{p}$ and $u|_{p_i} =_{E_0} w|_{p_i}$, for each position p_i in $\hat{p}$. But, by Proposition 9, $w|_{p_i} = v|_{dest(p_i)}$, so $u|_{p_i} =_{E_0} w|_{p_i} = v|_{dest(p_i)}$, for each position p_i in $\hat{p}$. □

Lemma 9 (Relation between $\mathcal{E}$-unifiers and B-unifiers of abstractions)**.** *Given an OS equational theory $\mathcal{E} = (\Sigma, E_0 \cup B)$ and two terms u and v in $\mathcal{H}_\Sigma(\mathcal{X})$, if $abstract_{\Sigma_1}((u,v)) = \langle \lambda(\bar{x},\bar{y}).(u^\circ, v^\circ); (\theta_u^\circ, \theta_v^\circ); (\phi_u^\circ, \phi_v^\circ)\rangle$ and σ' is a ground substitution such that $V_{u,v} \subseteq dom(\sigma')$, $u\sigma' =_{\mathcal{E}} v\sigma'$, and $dom(\sigma') \cap (\hat{x} \cup \hat{y}) = \emptyset$ then there exists another ground substitution σ° such that $u^\circ\sigma^\circ =_B v^\circ\sigma^\circ$, $E_0 \vdash (\phi_u^\circ \wedge \phi_v^\circ)\sigma^\circ$, $dom(\sigma^\circ) = dom(\sigma') \cup \hat{x} \cup \hat{y}$, so $V_{(u^\circ, v^\circ, \phi_u^\circ, \phi_v^\circ)\sigma^\circ} = \emptyset$, and $\sigma' =_{E_0} \sigma^\circ_{dom(\sigma')}$.*

Proof. Let $\bar{x} = \{x_1, \ldots, x_{i_x}\}$ and $\bar{y} = \{y_1, \ldots, y_{i_y}\}$, so $u^\circ = u[\bar{x}]_{\bar{p}}$, $\phi_u^\circ = (\bigwedge_{i=1}^{i_x} x_i = u|_{p_i})$, $v^\circ = v[\bar{y}]_{\bar{q}}$, $\phi_v^\circ = (\bigwedge_{j=1}^{i_y} y_j = v|_{q_j})$, for appropriate $\bar{p}$ and $\bar{q}$ such that $\hat{p} = top_{\Sigma_0}(u)$ and $\hat{q} = top_{\Sigma_0}(v)$. Also, let $u = u[\bar{x}']_{\bar{p}'}$ and $v = v[\bar{y}']_{\bar{q}'}$, where $pos_{\mathcal{X}_1}(u) = \hat{p}'$, $V_u \cap \mathcal{X}_1 = \hat{x}'$, $pos_{\mathcal{X}_1}(v) = \hat{q}'$, and $V_v \cap \mathcal{X}_1 = \hat{y}'$, so $u^\circ = u[\bar{x}]_{\bar{p}}[\bar{x}']_{\bar{p}'}$ and $v^\circ = v[\bar{y}]_{\bar{q}}[\bar{y}']_{\bar{q}'}$. As $u\gamma'$ and $v\sigma'$ are ground terms then $\hat{x}' \cup \hat{y}' \subseteq dom(\sigma')$.

As $u^\circ = u[\bar{x}]_{\bar{p}}$ then $pos_{\mathcal{X}_0}(u^\circ) = \hat{p}$, hence $V_{u[]_{\bar{p}}} \cap \mathcal{X}_0 = \emptyset$, i.e., $V_{u[]_{\bar{p}}} = \hat{x}' \subset \mathcal{X}_1$, so $V_{u[]_{\bar{p}}[]_{\bar{p}'}} = \emptyset$, and $u[]_{\bar{p}}[]_{\bar{p}'} = u\sigma'[]_{\bar{p}}[]_{\bar{p}'}$. In the same way, $V_{v[]_{\bar{q}}} = \hat{y}' \subset \mathcal{X}_1$, $V_{v[]_{\bar{q}}[]_{\bar{q}'}} = \emptyset$, and $v[]_{\bar{q}}[]_{\bar{q}'} = v\sigma'[]_{\bar{q}}[]_{\bar{q}'}$.

Let $\hat{t}^{\sigma'}_{\Sigma_0} = \bigcup_{t \in dom(\sigma')\sigma'} t|_{top_{\Sigma_0}(t)}$, i.e., the set of all top_{Σ_0} terms that appear in $z\sigma'$, where z ranges over the variables in $dom(\sigma')$. Now, let $\hat{t} = u|_{\hat{p}}\sigma' \cup v|_{\hat{q}}\sigma' \cup \hat{t}^{\sigma'}_{\Sigma_0}$. As $\hat{x}' \cup \hat{y}' \subseteq dom(\sigma')$, then $\hat{t}$ includes all the top_{Σ_0}-terms that appear in $u\sigma'$ and $v\sigma'$, either from their top_{Σ_0} positions or as subterms of the instances of the variables in their $\mathcal{X}_1$ positions.

Define $\sigma^\circ = rep_{\hat{t}}(\sigma') \cup \{x_i \mapsto rep_{\hat{t}}(u|_{p_i}\sigma') \mid x_i \in \hat{x}\} \cup \{y_j \mapsto rep_{\hat{t}}(v|_{q_j}\sigma') \mid y_j \in \hat{y}\}$, so $rep_{\hat{t}}(\sigma) = \sigma^\circ_{dom(\sigma')}$, hence $dom(\sigma^\circ) = dom(\sigma') \cup \hat{x} \cup \hat{y}$ and $\sigma' =_{E_0} \sigma^\circ_{dom(\sigma')}$. Then:

- as $\hat{x}' \cup \bar{y}' \subseteq dom(\sigma') = dom(\sigma'_{rep})$ then $\hat{x} \cup \hat{y} \cup \hat{x}' \cup \hat{y}' \subseteq dom(\sigma^\circ)$,
- $u^\circ\sigma^\circ = u[\bar{x}]_{\bar{p}}[\bar{x}']_{\bar{p}'}\sigma^\circ = u[\bar{x}\sigma^\circ]_{\bar{p}}[\bar{x}'\sigma^\circ]_{\bar{p}'} = u[rep_{\hat{t}}(u|_{\bar{p}}\sigma')]_{\bar{p}}[\bar{x}'\sigma'_{rep}]_{\bar{p}'} =_{E_0} u[u|_{\bar{p}}\sigma']_{\bar{p}}[\bar{x}'\sigma']_{\bar{p}'} = u\sigma'[u|_{\bar{p}}\sigma']_{\bar{p}}[\bar{x}'\sigma']_{\bar{p}'} = u[u|_{\bar{p}}]_{\bar{p}}[\bar{x}]'_{\bar{p}'}\sigma' = u\sigma'$,
- $v^\circ\sigma^\circ = v[\bar{y}]_{\bar{q}}[\bar{y}']_{\bar{q}'}\sigma^\circ = v[\bar{y}\sigma^\circ]_{\bar{q}}[\bar{y}'\sigma^\circ]_{\bar{q}'} = v[rep_{\hat{t}}(v|_{\bar{q}}\sigma')]_{\bar{q}}[\bar{y}'\sigma'_{rep}]_{\bar{q}'} =_{E_0} v[v|_{\bar{q}}\sigma']_{\bar{q}}[\bar{y}'\sigma']_{\bar{q}'} = v\sigma'[v|_{\bar{q}}\sigma']_{\bar{q}}[\bar{y}'\sigma']_{\bar{q}'} = v[v|_{\bar{q}}]_{\bar{q}}[\bar{y}']_{\bar{q}'}\sigma' = v\sigma'$,
- as $u\sigma' =_{\mathcal{E}} v\sigma'$, then $u^\circ\sigma^\circ =_{E_0} u\sigma' =_{\mathcal{E}} v\sigma' =_{E_0} v^\circ\sigma^\circ$, i.e., $u^\circ\sigma^\circ =_{\mathcal{E}} v^\circ\sigma^\circ$.

By Proposition 10, there exists a term w such that $u^\circ\sigma^\circ =_B w =_{E_0} v^\circ\sigma^\circ$, let $\hat{r} = top_{\Sigma_0}(w)$. We prove $v^\circ\sigma^\circ = w$, so $u^\circ\sigma^\circ =_B v^\circ\sigma^\circ$:

- as $u^\circ\sigma^\circ = u[rep_{\hat{t}}(u|_{\bar{p}}\sigma')]_{\bar{p}}[\bar{x}'\sigma'_{rep}]_{\bar{p}'} =_B w$ then, by Proposition 9, there exists a bijection $dest_1$ such that $dest_1(top_{\Sigma_0}(u^\circ\sigma^\circ)) = \hat{r}$ and $w|_{r_i} = u^\circ\sigma^\circ|_{orig_1(r_i)}$, for each position r_i in $\hat{r}$. As $V_{u[]_{\bar{p}}[]_{\bar{p}'}} = \emptyset$ then either:
  - (i) $orig_1(r_i)$ is a position p_j in $\hat{p}$, so $w|_{r_i} = rep_{\hat{t}}(u|_{p_j}\sigma')$. As $u|_{p_j}\sigma'$ is an element of $\hat{t}$, then $w|_{r_i}$ is an element of $rep_{\hat{t}}(\hat{t})$; or
  - (ii) $orig_1(r_i)$ has the form $p'_j.s_k$, where p'_j is a position in $\hat{p}'$, so s_k is a top_{Σ_0}-position of $u^\circ\sigma^\circ|_{p'_j}$. Then the variable x'_j in $\hat{x}'$, let $\hat{s} = top_{\Sigma_0}(x'_j\sigma'_{rep})$ so $s_k \in \hat{s}$, verifies $x'_j\sigma'_{rep} = rep_{\hat{t}}(x'_j\sigma')$, so $s_k \in top_{\Sigma_0}(rep_{\hat{t}}(x'_j\sigma'))$, $rep_{\hat{t}}(x'_j\sigma') = x'_j\sigma'[rep^\circ_{\hat{t}}(x'_j\sigma'|_{\bar{s}})]_{\bar{s}}$, and $w|_{r_i} = (x'_j\sigma'_{rep})|_{s_k} = rep_{\hat{t}}(x'_j\sigma')|_{s_k} = rep^\circ_{\hat{t}}(x'_j\sigma'|_{s_k}) = rep_{\hat{t}}(x'_j\sigma'|_{s_k})$. Then, as $rep_{\hat{t}}(x'_j\sigma'|_{\hat{s}}) \subseteq rep_{\hat{t}}(\hat{t})$, $w|_{r_i}$ is an element of $rep_{\hat{t}}(\hat{t})$.

In conclusion, $w|_{\hat{r}} \subseteq rep_{\hat{t}}(\hat{t})$, hence $w = w[rep_{\hat{t}}(w|_{\bar{r}})]_{\bar{r}}$.

- $v^\circ\sigma^\circ = v[rep_{\hat{t}}(v|_{\bar{q}}\sigma')]_{\bar{q}}[\bar{y}'\sigma'_{rep}]_{\bar{q}'} =_{E_0} w = w[rep_{\hat{t}}(w|_{\bar{r}})]_{\bar{r}}$. By Proposition 7, $top_{\Sigma_0}(v^\circ\sigma^\circ) = top_{\Sigma_0}(w) = \hat{r}$. As $v = v[\bar{y}']_{\bar{q}'}$, $V_{v[]_{\bar{q}}} = \hat{y}'$, and $top_{\Sigma_0}(v) = \hat{q}$ then, for each position r_i in $\hat{r}$, either:

  (i) r_i is a position q_j in $\hat{q}$, so $rep_{\hat{t}}(w|_{r_i}) = w_{r_i} =_{E_0} v^\circ\sigma^\circ|_{q_j} = rep_{\hat{t}}(v|_{q_j}\sigma')$. As $rep_{\hat{t}}(w|_{r_i}) =_{E_0} rep_{\hat{t}}(v|_{q_j}\sigma')$ then, by Remark 4, $rep_{\hat{t}}(w|_{r_i}) = rep_{\hat{t}}(v|_{q_j}\sigma')$, i.e., $w_{r_i} = v^\circ\sigma^\circ|_{q_j} = v^\circ\sigma^\circ|_{r_i}$; or

  (ii) r_i has the form $q'_j.s_k$, where q'_j is a position in $\hat{q}'$. As $\bar{y}' \subseteq dom(\sigma'_{rep})$ then $v^\circ\sigma^\circ|_{q'_j} = y'_j\sigma'_{rep} = rep_{\hat{t}}(y'_j\sigma')$, let $\hat{s} = top_{\Sigma_0}(rep_{\hat{t}}(y'_j\sigma'))$, so $s_k \in \hat{s}$ and $rep_{\hat{t}}(w|_{r_i}) = w_{r_i} =_{E_0} v^\circ\sigma^\circ|_{q'_j.s_k} = rep_{\hat{t}}(y'_j\sigma')|_{s_k} = rep^\circ_{\hat{t}}(y'_j\sigma'|_{s_k}) = rep_{\hat{t}}(y'_j\sigma'|_{s_k})$. As $rep_{\hat{t}}(w|_{r_i}) =_{E_0} rep_{\hat{t}}(y'_j\sigma'|_{s_k})$ then, by Remark 4, $rep_{\hat{t}}(w|_{r_i}) = rep_{\hat{t}}(y'_j\sigma'|_{s_k})$, i.e., $w_{r_i} = v^\circ\sigma^\circ|_{q_j.s_k} = v^\circ\sigma^\circ|_{r_i}$.

  In conclusion, as $v^\circ\sigma^\circ[]_{\bar{r}} = w[]_{\bar{r}}$, $v^\circ\sigma^\circ = v^\circ\sigma^\circ[v^\circ\sigma^\circ|_{\bar{r}}]_{\bar{r}} = w[v^\circ\sigma^\circ|_{\bar{r}}]_{\bar{r}} = w[w|_{\bar{r}}]_{\bar{r}} = w$.

We have just proved $u^\circ\sigma^\circ =_B v^\circ\sigma^\circ$, but also:

- as $\phi^\circ_u = (\bigwedge_{i=1}^{i_x} x_i = u|_{p_i})$ and $\bar{x}\sigma^\circ = rep_{\hat{t}}(u|_{\bar{p}}\sigma') =_{E_0} u|_{\bar{p}}\sigma' = u|_{\bar{p}}\sigma^\circ$ then $E_0 \vdash \phi^\circ_u\sigma^\circ$, and
- as $\phi^\circ_v = (\bigwedge_{j=1}^{i_y} y_j = v|_{q_j})$ and $\bar{y}\sigma^\circ = rep_{\hat{t}}(v|_{\bar{q}}\sigma') =_{E_0} v|_{\bar{q}}\sigma' = v|_{\bar{q}}\sigma^\circ$ then $E_0 \vdash \phi^\circ_v\sigma^\circ$,

so $E_0 \vdash (\phi^\circ_u \wedge \phi^\circ_v)\sigma^\circ$. □

Proposition 14. *Let $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ be a rewrite theory with built-in subtheory (Σ_0, E_0). If t_0, t_1, and t_2 are terms in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $t_0 \leftrightarrow_{E_0} t_1 \leftrightarrow_B t_2$ then there exists a term t'_1 in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $t_0 \leftrightarrow_B t'_1 \leftrightarrow_{E_0} t_2$.*

Proof. By Proposition 13, $top_{\Sigma_0}(t_0) = top_{\Sigma_0}(t_1)$. The one-step E_0-deduction $t_0 \leftrightarrow_{E_0} t_1$ is performed at some position p of the term t_0 with a Σ_0-sentence $v_0 = w_0$ *if* C in $E_0 \cup E_0^{-1}$, and substitution σ_0, so $t_0|_p$ in $\mathcal{T}_{\Sigma_0}$. Let $q = top_{\Sigma_0}(t_0, p)$. Then $p = q.q'$, for suitable q', $t_0 = t_0[v_0\sigma_0]_{q.q'}$, and $t_1 = t_0[w_0\sigma_0]_{q.q'}$.

The one-step B-deduction $t_1 \leftrightarrow_B t_2$ uses a regular Σ_1-equation $v = w$ and a substitution σ at some position r of t_1, so $t_1|_r$ in $\mathcal{H}_\Sigma(\mathcal{X})$ and $t_1|_r = v\sigma$. Let $vars(v) \cap \mathcal{X}_0 = \{x_1, \ldots, x_m\}$. As B is regular then $vars(w) \cap \mathcal{X}_0 = vars(v) \cap \mathcal{X}_0$. For $0 \leq i \leq m$, as B is linear, let r_i be the position of the variable x_i in v and let s_i be the position of the variable x_i in w. As v in $\mathcal{H}_\Sigma(\mathcal{X})$, so there are no function symbols from Σ_0, then there exists a position r'_i and a natural number j_i such that $r_i = r'_i.j_i$, $v|_{r_i}$ in $\mathcal{X}_0 \subseteq \mathcal{T}_{\Sigma_0}(\mathcal{X}_0)$, and $v|_{r'_i}$ in $\mathcal{H}_\Sigma(\mathcal{X})$, so r_i in $top_{\Sigma_0}(v)$.

As all Σ_0-subterms of $t_1|_r$ must be matched with the term v through the variables $\{x_1, \ldots, x_m\}$, then $top_{\Sigma_0}(t_1|_r) = \{r_1, \ldots, r_m\}$, and $v\sigma|_{r_i} = t_1|_{r.r_i}$ is a topmost Σ_0-subterm of t_1, that is moved to position $r.s_i$ in t_2.

There are three cases to consider regarding the relative position of p and r:

1. $p \not\leq r$ because:

   (a) if $p = r$ then $t_1|_p$ is both a term in $\mathcal{T}_{\Sigma_0}$ and in $\mathcal{H}_\Sigma$, a contradiction, and

   (b) if $p < r$ then $r = p.p'$, for suitable $p' \neq \epsilon$, and $t_1|_p$ is a term in $\mathcal{T}_{\Sigma_0}$ that has a subterm in $\mathcal{H}_\Sigma$ at position p', in contradiction with (Σ_0, E_0) being a subsignature of $(\Sigma, \mathcal{E})$.

2. If $p \not\leq r$ and $r \not\leq p$ then both one-step deductions are independent and $t_2 = (t_0[w_0\sigma_0]_p)[w\sigma]_r = (t_0[w\sigma]_r)[w_0\sigma_0]_p$. Applying the one-step B-deduction before the one-step E_0-deduction yields $t_1' = t_0[w\sigma]_r$.

3. If $r < p$, as $p = q.q'$ and $top_{\Sigma_0}(t_0) = top_{\Sigma_0}(t_1)$, then $t_1|_q \in \mathcal{T}_{\Sigma_0}$, $t_0|_r \in \mathcal{H}_\Sigma$, and (Σ_0, E_0) is a subsignature of $(\Sigma, \mathcal{E})$, then $q < r$, so $q = r.r_l$, for some $r_l \in top_{\Sigma_0}(t_0|_r)$, because q is a topmost E_0-position of t_0 and t_1, and $t_0 = t_0[v_0\sigma_0]_{r.r_l.q'}$. As both one-step deductions take place below position r, let $t_0|_r = t'$ for simplicity of notation. Then $t' = t'[v_0\sigma_0]_{r_l.q'} \leftrightarrow_{E_0} t'[w_0\sigma_0]_{r_l.q'} = t'[t'|_{r_l}[w_0\sigma_0]_{q'}]_{r_l} = v\sigma \leftrightarrow_B w\sigma = w\sigma[t'|_{r_l}[w_0\sigma_0]_{q'}]_{s_l}$, where $v|_{r_l} = x_l$ and $x_l\sigma = t'|_{r_l}[w_0\sigma_0]_{q'}$, so $t_2 = t_0[w\sigma[t'|_{r_l}[w_0\sigma_0]_{q'}]_{s_l}]_r$.

   Let $\sigma' = \sigma_{dom(\sigma)\setminus x_l} \cup \{x_l \mapsto t'|_{r_l}\}$. As v is linear and $r_l \leq r_l.q'$ then $v\sigma' = (t'[w_0\sigma_0]_{r_l.q'})[t'|_{r_l}]_{r_l} = t'$. But $t' = t'[v_0\sigma_0]_{r_l.q'}$, so $t'[v_0\sigma_0]_{r_l.q'} = t' = v\sigma' \leftrightarrow_B w\sigma' = w\sigma[t'|_{r_l}]_{s_l} = w\sigma[t_0|_{r.r_l}]_{s_l}$. As $(w\sigma[t'|_{r_l}]_{s_l})|_{s_l.q'} = t'|_{r_l.q'}$ then $w\sigma[t'|_{r_l}]_{s_l} \leftrightarrow_{E_0} (w\sigma[t'|_{r_l}]_{s_l})[w_0\sigma_0]_{s_l.q'} = w\sigma[t'|_{r_l}[w_0\sigma_0]_{q'}]_{s_l}$. In conclusion, $t' \leftrightarrow_B w\sigma[t'|_{r_l}]_{s_l} \leftrightarrow_{E_0} w\sigma[t'|_{r_l}[w_0\sigma_0]_{q'}]_{s_l}$. Then, take $t_1' = t_0[w\sigma[t'|_{r_l}]_{s_l}]_r = t_0[w\sigma[t_0|_{r.r_l}]_{s_l}]_r$, so $t_0 = t_0[t']_r \leftrightarrow_B t_0[w\sigma[t'|_{r_l}]_{s_l}]_r = t_1' \leftrightarrow_{E_0} t_0[w\sigma[t'|_{r_l}[w_0\sigma_0]_{q'}]_{s_l}]_r = t_2$.

□

Proposition 15. *Let $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ be a rewrite theory with built-in subtheory (Σ_0, E_0). If $\{t_0, \ldots, t_{n+1}\}$ is a set of terms in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $t_0 \leftrightarrow_{E_0} t_1 \leftrightarrow_{E_0} \cdots \leftrightarrow_{E_0} t_{n-1} \leftrightarrow_B t_n$ then there exists a set of terms $\{t_1', \ldots, t_{n-1}'\}$ in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $t_0 \leftrightarrow_B t_1' \leftrightarrow_{E_0} \cdots \leftrightarrow_{E_0} t_{n-1}' \leftrightarrow_{E_0} t_n$.*

Proof. By induction on n.

- Base case, $n = 2$. $t_0 \leftrightarrow_{E_0} t_1 \leftrightarrow_B t_2$, so by Proposition 14 there exists a term t_1' in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $t_0 \leftrightarrow_B t_1' \leftrightarrow_{E_0} t_2$.

- Induction case. As $t_1 \leftrightarrow_{E_0} \cdots \leftrightarrow_{E_0} t_{n-1} \leftrightarrow_B t_n$, by I.H., there exists a set of terms $\{t_2', \ldots, t_{n-1}'\}$ in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $t_1 \leftrightarrow_B t_2' \leftrightarrow_{E_0} \cdots \leftrightarrow_{E_0} t_{n-1}' \leftrightarrow_{E_0} t_n$, so $t_0 \leftrightarrow_{E_0} t_1 \leftrightarrow_B t_2' \leftrightarrow_{E_0} \cdots \leftrightarrow_{E_0} t_{n-1}' \leftrightarrow_{E_0} t_n$. By Proposition 14, as $t_0 \leftrightarrow_{E_0} t_1 \leftrightarrow_B t_2'$, there exists t_1' in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $t_0 \leftrightarrow_B t_1' \leftrightarrow_{E_0} t_2'$, and $t_0 \leftrightarrow_B t_1' \leftrightarrow_{E_0} \cdots \leftrightarrow_{E_0} t_{n-1}' \leftrightarrow_{E_0} t_n$.

□

Proposition 10 (Decomposition of $\mathcal{E}$-equality in B-equality plus E_0-equality)**.** *Let $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ be a rewrite theory with built-in subtheory (Σ_0, E_0). If t and t'' are terms in $\mathcal{H}_\Sigma(\mathcal{X})$ and $t =_\mathcal{E} t''$ then there exists a term t' in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $t =_B t' =_{E_0} t''$.*

Proof. By induction on the number of applications of axioms in B.

- Base case: zero B-axioms. Then $t =_{E_0} t''$. Take $t' = t$, so $t =_B t' =_{E_0} t''$.

- Induction case. There are two cases to consider depending on the position of the first $\leftrightarrow_B$ step.

  - If $t \leftrightarrow_B t_1 \leftrightarrow_\mathcal{E} \cdots \leftrightarrow_\mathcal{E} t''$, by I.H., there exists a term t' in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $t_1 =_B t' =_{E_0} t''$, so $t \leftrightarrow_B t_1 =_B t' =_{E_0} t''$, i.e., $t =_B t' =_{E_0} t''$.

– $t \leftrightarrow_{E_0} \cdots \leftrightarrow_{E_0} t_{j-1} \leftrightarrow_B t_j \leftrightarrow_{\mathcal{E}} \cdots \leftrightarrow_{\mathcal{E}} t''$. By Proposition 15, as $t \leftrightarrow_{E_0} \cdots \leftrightarrow_{E_0} t_{j-1} \leftrightarrow_B t_j$, there exists a set $\{t'_1, \ldots, t'_{j-1}\}$ in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $t \leftrightarrow_B t'_1 \leftrightarrow_{E_0} \cdots \leftrightarrow_{E_0} t'_{j-1} \leftrightarrow_{E_0} t_j$. Then, by I.H., in $t'_1 \leftrightarrow_{E_0} \cdots \leftrightarrow_{E_0} t'_{j-1} \leftrightarrow_{E_0} t_j \cdots \leftrightarrow_{\mathcal{E}} t_n \cdots \leftrightarrow_{\mathcal{E}} t''$, there exists a term t' in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $t'_1 =_B t' =_{E_0} t''$, so $t \leftrightarrow_B t'_1 =_B t' =_{E_0} t''$, i.e., $t =_B t' =_{E_0} t''$.

□

Theorem 11 (Equivalence of $R/\mathcal{E}$ and R,B-rewriting for rewrite theories closed under B-extensions)**.** *Let $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ be a rewrite theory with built-in subtheory (Σ_0, E_0). If $\mathcal{R}$ is closed under B-extensions then $\rightarrow^1_{R,B} \subseteq \rightarrow^1_{R/\mathcal{E}}$, $\rightarrow_{R,B} = \rightarrow_{R/\mathcal{E}}$, and if t and w are terms in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $t \rightarrow^1_{R/\mathcal{E}} w$ then there exists t' in $\mathcal{H}_\Sigma(\mathcal{X})$ such that $t \rightarrow^1_{R,B} t' =_E w$.*

Proof. There is a special case to consider when there are no rewrite steps involved in the deductions.
(i) $\rightarrow^1_{R,B} \subseteq \rightarrow^1_{R/\mathcal{E}}$ and $\rightarrow_{R,B} \subseteq \rightarrow_{R/\mathcal{E}}$.

In the special case, $t \rightarrow_{R,B} v$ with no rewrite steps. As $\rightarrow_{R,B} = (\rightarrow^*_{R,B}; =_{\mathcal{E}})$ then $t =_{\mathcal{E}} v$, so $t \rightarrow_{R/\mathcal{E}} v$. The other cases are proved using induction on the total number of $\rightarrow^1_{R,B}$ rewrite steps in the proof tree.

- Base case:

  $t \rightarrow_{R,B} w$ has the form $t \rightarrow^1_{R,B} v =_{\mathcal{E}} w$ with only one $\rightarrow^1_{R,B}$ rewrite step in the proof tree, so there is a rule $c : l \rightarrow r \ \textit{if} \ \bigwedge_{i=1}^m l_i \rightarrow r_i \mid \phi$ in R, with normal form $c^\circ : l^\circ \rightarrow r \ \textit{if} \ \bigwedge_{i=1}^m l_i \rightarrow r_i \mid \phi \wedge \phi^\circ$, a position p in $pos_{\Sigma_1}(t)$ (let $t' = rep(t|_p) =_{E_0} t|_p$), and a substitution σ such that $t' =_B l^\circ\sigma$, $l_i\sigma =_{\mathcal{E}} r_i\sigma$, for $1 \leq i \leq m$, and $E_0 \vdash (\phi \wedge \phi^\circ)\sigma$, so $t[r\sigma]_p = v$.

  As $\phi^\circ = \bigwedge_{j=1}^n (x_j = l|_{q_j})$, then $l^\circ = l[x_1]_{q_1} \cdots [x_n]_{q_n}$ and $l^\circ\sigma = l\sigma[x_1\sigma]_{q_1} \cdots [x_n\sigma]_{q_n}$ so, as $E_0 \vdash \phi^\circ\sigma$, $x_j\sigma =_{E_0} l\sigma|_{q_j}$, for $1 \leq j \leq n$, and $l\sigma = l\sigma[l\sigma|_{q_1}]_{q_1} \cdots [\sigma_{q_n}]_{q_n} =_{E_0} l\sigma[x_1\sigma]_{q_1} \cdots [x_n\sigma]_{q_n} =_B t' =_{E_0} t|_p$, i.e., $l\sigma =_{\mathcal{E}} t|_p$.

  As $t|_p =_{\mathcal{E}} l\sigma$ and $l_i\sigma =_{\mathcal{E}} r_i\sigma$, for $1 \leq i \leq m$, then $t = t[t|_p]_p =_{\mathcal{E}} t[l\sigma]_p \rightarrow^1_{R(\mathcal{E})} t[r\sigma]_p = v =_{\mathcal{E}} w$ with rule c in R, that is, $t \rightarrow^1_{R/\mathcal{E}} v$ and also $t \rightarrow_{R/\mathcal{E}} w$.

- Induction case. There are two subcases to consider:

  1. $t \rightarrow_{R,B} w$ has the form $t \rightarrow^1_{R,B} v =_{\mathcal{E}} w$ with several $\rightarrow^1_{R,B}$ rewrite steps in the derivation. As in the base case, there is a rule $c : l \rightarrow r \ \textit{if} \ \bigwedge_{i=1}^m l_i \rightarrow r_i \mid \phi$ in R, with normal form $c^\circ : l^\circ \rightarrow r \ \textit{if} \ \bigwedge_{i=1}^m l_i \rightarrow r_i \mid \phi \wedge \phi^\circ$, a position p in $pos_{\Sigma_1}(t)$ (let $t' = rep(t|_p) =_{E_0} t|_p$), and a substitution σ such that $t' =_B l^\circ\sigma$, $l_i\sigma \rightarrow_{R,B} r_i\sigma$, for $1 \leq i \leq m$, and $E_0 \vdash (\phi \wedge \phi^\circ)\sigma$, so $t[r\sigma]_p = v$.
     By I.H. $l_i\sigma \rightarrow_{R/\mathcal{E}} r_i\sigma$, for $1 \leq i \leq m$. As in the base case, $E_0 \vdash (\phi \wedge \phi^\circ)\sigma$ implies $t|_p =_{\mathcal{E}} l\sigma$, so $t = t[t|_p]_p =_{\mathcal{E}} t[l\sigma]_p \rightarrow^1_{R(\mathcal{E})} t[r\sigma]_p =_{\mathcal{E}} v$, i.e., $t \rightarrow^1_{R/\mathcal{E}} v$ and also $t \rightarrow_{R/\mathcal{E}} w$.
  2. $t \rightarrow^1_{R,B} u \rightarrow^+_{R,B} v =_{\mathcal{E}} w$. By the previous subcase $t \rightarrow^1_{R/\mathcal{E}} u \rightarrow^+_{R,B} v =_{\mathcal{E}} w$, and, by I.H., $t \rightarrow^1_{R/\mathcal{E}} u \rightarrow^+_{R/\mathcal{E}} v =_{\mathcal{E}} w$, i.e., $t \rightarrow^*_{R/\mathcal{E}} v =_E w$, or $t \rightarrow_{R/\mathcal{E}} w$.

(ii) $\rightarrow_{R/\mathcal{E}} \subseteq \rightarrow_{R,B}$, $t \rightarrow^1_{R/\mathcal{E}} w \implies \exists t' \, s.t. \, t \rightarrow^1_{R,B} t' =_E w$.

In the special case, $t \rightarrow_{R/\mathcal{E}} v$ with no rewrite steps because $t =_{\mathcal{E}} v$, as $\rightarrow_{R,B} = (\rightarrow^*_{R,B}; =_E)$ then $t \rightarrow_{R,B} v$. The other cases are proved using induction in the total number of $\rightarrow^1_{R/\mathcal{E}}$ rewrite steps in the derivation.

- Base case: $t \to^1_{R/\mathcal{E}} v =_{\mathcal{E}} w$ with only one $\to^1_{R/\mathcal{E}}$ rewrite step in the derivation using a rule $c : l \to r \ \mathit{if}\ C$ in R, with $C = \bigwedge_{i=1}^m l_i \to r_i \mid \phi \wedge \phi^\circ$, and a substitution σ, so no rewriting steps are taken to solve $C\sigma$, because $l_i\sigma = \mathcal{E}r_i\sigma$, for $1 \leq i \leq m$.

  By Proposition 10 there exists a term t' in $\mathcal{H}_\Sigma$ such that $t =_B t' =_{E_0} t'' \to^1_{R(\mathcal{E})} u =_{\mathcal{E}} v$. We have $t \overset{ax_1}{\longleftrightarrow}_B \cdots \overset{ax_l}{\longleftrightarrow}_B t'$, where ax_i (linear and regular), for $1 \leq i \leq l$ has the form $w_i = w_i'$, let $\overline{ax_i}$ be $w_i' = w_i$, so each top_{Σ_0} subterm of t is moved by $ax_1 \cdots ax_l$ and becomes another top_{Σ_0} subterm of t'. Then, $\overline{ax_l} \cdots \overline{ax_1}$ moves the top_{Σ_0} subterms of t'' in the opposite way, so there exists a term t_0 in $\mathcal{T}_\Sigma$ such that $t'' \overset{\overline{ax_l}}{\longleftrightarrow}_B \cdots \overset{\overline{ax_1}}{\longleftrightarrow}_B t_0 =_{E_0} t$.

  We have $t =_{E_0} t_0 =_B t'' = t''[l\sigma]_p$, so $t''|_p = l\sigma$. The more general case, where $t_0 =_B t''|_p =_B l\sigma$ is studied in Theorem 2 and Corollary 2 in [Mes17], where it is proved that there is a position q in $pos(t_0)$, a rule $c_0 : l_0 \to r_0 \ \mathit{if}\ C$ in R, maybe the original c, and a substitution σ_0, such that $t_0|_q =_B l_0\sigma_0$, $t_0[r_0\sigma_0]_q =_B u$, and $C\sigma_0 = C\sigma$, which is also valid for our particular case where $t''|_p = l\sigma$. As, by definition of rule, $l_0 \in \mathcal{H}_\Sigma(\mathcal{X})$, then $q \in pos_{\Sigma_1}(t_0)$, so $t_0|_q =_{E_0} t|_q$. Let $top_{\Sigma_0}(t|_q) = \hat{z}$. Then $rep_{t|_{q.\hat{z}}}$ is the function that given a term in $\mathcal{T}_\Sigma$ returns the same term with each top_{Σ_0} term on it replaced with the representative for that top_{Σ_0} term in $rep(t|_q)$, if it exists, so $rep(t|_q) = rep_{t|_{q.\hat{z}}}(t_0|_q) =_B rep_{t|_{q.\hat{z}}}(l_0\sigma_0)$.

  Let $abstract_{\Sigma_1}(l_0) = \langle \lambda\bar{y}.l_0^\circ; \theta_0^\circ; \phi_0^\circ \rangle$, $\bar{y} = y_1, \ldots, y_k$, $l_0^\circ = l_0[\bar{y}]_{\bar{o}}$, $\phi^\circ = \bigwedge_{j=1}^k y_j = l_0|_{o_j}$. Define $\sigma' : dom(\sigma_0) \cup \hat{y} \to \mathcal{T}_\Sigma$ as: if $z = y_j \in \hat{y}$ then $z\sigma' = rep_{t|_{q.\hat{z}}}(l_0|_{o_j}\sigma_0)$ else $z\sigma' = rep_{t|_{q.\hat{z}}}(z\sigma_0)(=_{E_0} z\sigma_0)$. As, for $1 \leq j \leq k$, $y_j\sigma' = rep_{t|_{q.\hat{z}}}(l_0|_{o_j}\sigma_0) =_{E_0} l_0|_{o_j}\sigma_0 =_{E_0} l_0|_{o_j}\sigma'$, because $\hat{y} \cap V_{l_0|_{o_j}} = \emptyset$, then $E_0 \vdash \phi^\circ\sigma'$. Also, as $C\sigma_0 = C\sigma$ and if $z \in dom(\sigma_0)$ then $z\sigma' =_{E_0} z\sigma_0$ then $\bar{l}\sigma' =_{E_0} \bar{l}\sigma_0 =_E \bar{r}\sigma_0 =_{E_0} \bar{r}\sigma'$, i.e., $\bar{l}\sigma' =_{\mathcal{E}} \bar{r}\sigma'$, and $\phi\sigma' =_{E_0} \phi\sigma_0 = \phi\sigma$, so $E_0 \vdash \phi\sigma'$. As $\phi\sigma'$ and $\phi^\circ\sigma'$ are ground, because $rep_{t|_{q.\hat{z}}}$ is replacing each ground subterm with another ground subterm, then $E_0 \vdash (\phi \wedge \phi^\circ)\sigma'$.

  As

  - $l_0[]_{\bar{o}}\sigma' = l_0\sigma'[]_{\bar{o}} = rep_{t|_{q.\hat{z}}}(l_0\sigma_0[]_{\bar{o}})$, and
  - $y_j\sigma' = rep_{t|_{q.\hat{z}}}(l_0|_{o_j}\sigma_0)$, for $1 \leq j \leq k$,

  then $l_0^\circ\sigma' = l_0[\bar{y}]_{\bar{o}}\sigma' = rep_{t|_{q.\hat{z}}}(l_0\sigma_0[l_0|_{\bar{o}}\sigma_0]_{\bar{o}}) = rep_{t|_{q.\hat{z}}}(l_0[l_0|_{\bar{o}}]_{\bar{o}}\sigma_0) = rep_{t|_{q.\hat{z}}}(l_0\sigma_0) =_B rep(t|_q)$, i.e., $rep(t|_q) =_B l_0^\circ\sigma'$ so, as $t[r_0\sigma']_q =_{E_0} t[r_0\sigma_0]_q =_{E_0} t_0[r_0\sigma_0]_q =_B u =_{\mathcal{E}} v$, i.e., $t[r_0\sigma']_q =_{\mathcal{E}} v$, we have $t \to^1_{R,B} t[r_0\sigma']_q =_{\mathcal{E}} v =_{\mathcal{E}} w$, or $t \to_{R,B} w$.

- Induction case:

  again, there are two subcases to consider:

  1. $t \to^1_{R/\mathcal{E}} v =_{\mathcal{E}} w$ with several $\to^1_{R/\mathcal{E}}$ rewrite steps in the derivation. The proof is the same as the one in the base case, except that instead of having $\bar{l}\sigma' =_{\mathcal{E}} \bar{r}\sigma'$ now we have $l_i\sigma' \to_{R/\mathcal{E}} r_i\sigma'$, for $1 \leq i \leq m$, so by I.H., also $l_i\sigma' \to_{R,B} r_i\sigma'$ hence $t \to^1_{R,B} t[r_0\sigma']_q =_{\mathcal{E}} v =_{\mathcal{E}} w$, or $t \to_{R,B} w$.
  2. $t \to^1_{R/\mathcal{E}} u \to^+_{R/\mathcal{E}} v =_{\mathcal{E}} w$. By the previous subcase $t \to^1_{R,B} u' =_{\mathcal{E}} u \to^+_{R/\mathcal{E}} v =_{\mathcal{E}} w$, and, by I.H., as $u' \to^+_{R/\mathcal{E}} v$, $t \to^1_{R,B} u' \to^+_{R,B} = u'' =_{\mathcal{E}} v =_{\mathcal{E}} w$, i.e., $t \to^*_{R,B} u'' =_{\mathcal{E}} w$, or $t \to_{R,B} w$.

□

Proposition 11 (Existence of canonical paths). *Given a rewrite theory $\mathcal{R} = (\Sigma, E, R)$ with built-in subtheory (Σ_0, E_0), and a narrowing path from a reachability goal G, $G = \Delta_0 \mid \psi_0 \rightsquigarrow_{\sigma_1} \Delta_1 \mid \psi_1 \rightsquigarrow_{\sigma_2} \cdots \Delta_{m-1} \mid \psi_{m-1} \rightsquigarrow_{\sigma_m} nil \mid \psi_m$, there exists another narrowing path $G = \Delta_0 \mid \psi_0 \rightsquigarrow_{\sigma_1} \Delta_1 \mid \chi_1 \rightsquigarrow_{\sigma_2} \cdots \Delta_{m-1} \mid \chi_{m-1} \rightsquigarrow_{\sigma_m} nil \mid \chi_m$, where if one rule is applied at step i in one path then the same rule is applied at step i in the other path, for $1 \leq i \leq m$, there is no simplification of the reachability formula when rule* unification *or* rewrite *is applied, and $E_0 \vdash \psi_i \Leftrightarrow \chi_i$, for $1 \leq i \leq m$.*

Proof. As the applied rule at each step i only depends on Δ_{i-1} which is the same on both paths, as long as ψ_i and χ_i are satisfiable, all that it has to be proved is $E_0 \vdash \psi_i \Leftrightarrow \chi_i$. Then as ψ_i is satisfiable so is χ_i.

Let $\chi_0 = \psi_0$, so $E_0 \vdash \psi_0 \Leftrightarrow \chi_0$. Then $E_0 \vdash \psi_{i-1} \Leftrightarrow \chi_{i-1}$ implies $E_0 \vdash \psi_i \Leftrightarrow \chi_i$, for $1 \leq i \leq m$. The proof for rules transitivity and congruence is trivial, since $\psi_i = \psi_{i-1}$, $\chi_i = \chi_{i-1}$, and $E_0 \vdash \psi_{i-1} \Leftrightarrow \chi_{i-1}$, and it is also trivial for rules unification and rewrite since $\chi_i = (\chi_{i-1} \wedge \phi^\circ)\sigma_i$ and $E_0 \vdash \psi_i \Leftrightarrow (\chi_{i-1} \wedge \phi^\circ)\sigma_i$. As $E_0 \vdash \psi_0 \Leftrightarrow \chi_0$ then $E_0 \vdash \psi_i \Leftrightarrow \chi_i$, for $1 \leq i \leq m$. □

Theorem 12 (Soundness in $\rightarrow^1_{R,B}$ of the Calculus for Reachability Goals). *Given a closed under B-extensions rewrite theory $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ with built-in subtheory (Σ_0, E_0), where $\mathcal{E} = E_0 \cup B$, and a narrowing path from a reachability goal G, $G = \Delta_1 \mid \psi_1 \rightsquigarrow_{\sigma_1} \Delta_2 \mid \psi_2 \rightsquigarrow_{\sigma_2} \cdots \Delta_m \mid \psi_m \rightsquigarrow_{\sigma_m} nil \mid \psi$, let $\sigma = \sigma_1 \cdots \sigma_m$, then:*

1. *if $\Delta_1 = \bigwedge_{i=1}^n u_i \rightarrow v_i$ and $\rho : \mathcal{X} \rightarrow \mathcal{T}_\Sigma$ is a substitution such that $dom(\rho) = vars(G\sigma)$ and $\psi\rho$ is satisfiable then $(\sigma\rho)_{vars(G)}$ is a solution for G in $\rightarrow^1_{R,B}$, and*
2. *if $\Delta_1 = u_1|_p \rightarrow^1 x, u_1[x]_p \rightarrow v_1 \wedge \bigwedge_{i=2}^n u_i \rightarrow v_i \mid \psi_1$ and $\rho : \mathcal{X} \setminus \{x\} \rightarrow \mathcal{T}_\Sigma$ is a substitution such that $dom(\rho) = vars(G\sigma) \setminus \{x\}$ and $\psi\rho$ is satisfiable then*
   (a) *$(\sigma\rho)_{vars(G)\setminus\{x\}}$ is a solution for $\bigwedge_{i=1}^n u_i \rightarrow v_i \mid \psi_1$ in $\rightarrow^1_{R,B}$ and*
   (b) *there exists a substitution $\rho_x : \{x\} \rightarrow \mathcal{T}_\Sigma$ such that $(\sigma(\rho \cup \rho_x))_{vars(G)}$ is a solution for G in $\rightarrow^1_{R,B}$.*

Proof. By structural induction on the length of the narrowing path and the first inference rule applied.

(i) Base case

$G = u_1 \rightarrow v_1 \mid \psi_1 \rightsquigarrow_{[u],\sigma} nil \mid \psi$, where $abstract_{\Sigma_1}(v_1) = \langle \lambda\bar{x}.v_1^\circ; \theta^\circ; \phi^\circ \rangle$, $vars(\psi) \subseteq vars((\psi_1 \wedge \phi^\circ)\sigma)$, $E_0 \vdash \psi \Leftrightarrow (\psi_1 \wedge \phi^\circ)\sigma$, $\bar{x} = \{x_1, \ldots, x_l\}$, $v_1^\circ = v_1[x_1]_{q_1} \cdots [x_l]_{q_l}$, $\phi^\circ = \bigwedge_{i=1}^l x_i = v_1|_{q_i}$, σ in $CSU_B(u_1 = v_1^\circ)$, so $u_1\sigma =_B v_1^\circ\sigma$, and ψ is satisfiable. As ρ is a substitution such that $dom(\rho) = vars(G\sigma)$ and $\psi\rho$ is satisfiable, so $(\psi_1 \wedge \phi^\circ)\sigma\rho$ is also satisfiable, then $\psi_1\sigma\rho \in \mathcal{T}_{\Sigma_0}$, so $E_0 \vdash \psi_1\sigma\rho$, and $\phi^\circ\sigma\rho = \bigwedge_{i=1}^l x_i\sigma\rho = v_1|_{q_i}\sigma\rho$ is satisfiable, where $v_1|_{q_i}\sigma\rho \in \mathcal{T}_\Sigma$, for $1 \leq i \leq l$, so there exists a substitution $\rho' : \bigcup_{i=1}^l vars(x_i\sigma\rho) \rightarrow \mathcal{T}_\Sigma$ such that $x_i\sigma\rho\rho' =_{E_0} v_1|_{q_i}\sigma\rho\rho' = v_1|_{q_i}\sigma\rho$, for $1 \leq i \leq l$. Let $\gamma = \sigma\rho\rho'$.

As $u_1\sigma\rho$ and $v_1\sigma\rho$ in $\mathcal{T}_\Sigma$, the theory inclusion $(\Sigma_0, E_0) \subseteq (\Sigma, E)$ is protecting, and $u_1\sigma\rho =_B v_1^\circ\sigma\rho$, then $u_1\sigma\rho = u_1\gamma =_B v_1^\circ\gamma = v_1\gamma[x_1\gamma]_{q_1} \cdots [x_l\gamma]_{q_l} = v_1\sigma\rho[x_1\gamma]_{q_1} \cdots [x_l\gamma]_{q_l} =_{E_0} v_1\sigma\rho[v_1|_{q_1}\sigma\rho]_{q_1} \cdots [v_1|_{q_l}\sigma\rho]_{q_l} = v_1\sigma\rho$, so $u_1\sigma\rho =_E v_1\sigma\rho$, and $u_1\sigma\rho \rightarrow_{R,B} v_1\sigma\rho$. Also as $vars(\{u_1, v_1, \psi_1\})$ is a subset of $vars(G)$ then $u_1(\sigma\rho)_{vars(G)} \rightarrow_{R,B} v_1(\sigma\rho)_{vars(G)}$ and $E_0 \vdash \psi_1(\sigma\rho)_{vars(G)}$.

(ii) Induction case

$G = \bigwedge_{i=1}^{n} u_i \to v_i \mid \psi_1$ or $G = u_1|_p \to^1 x, u_1[x]_p \to v_1 \wedge \bigwedge_{i=2}^{n} u_i \to v_i \mid \psi_1$. Let $\Delta = \bigwedge_{i=2}^{n} u_i \to v_i$. There is one case for each inference rule.

1. Rule transitivity: $G = u_1 \to v_1 \wedge \Delta \mid \psi_1 \rightsquigarrow_{[t]} u_1 \to^1 x, x \to v_1 \wedge \Delta \mid \psi_1 \rightsquigarrow_{\sigma}^{*} nil \mid \psi$, so $\sigma_{vars(G)} \mid \psi$ is a computed answer for G. Let $G_1 = u_1 \to^1 x, x \to v_1 \wedge \Delta \mid \psi_1$. As $dom(\rho) = vars(G_1\sigma)$ and $vars(G_1) = vars(G) \cup \{x\}$ then $dom(\rho) = vars(G_1\sigma) \setminus \{x\}$ so, by I.H., $(\sigma\rho)_{vars(G)}$ is a solution for G in $\to_{R,B}$.

2. Rule congruence: $G = u_1|_p \to^1 x, u_1[x]_p \to v_1 \wedge \Delta \mid \psi_1 \rightsquigarrow_{[c]} u_1|_{p.i} \to^1 y, u_1[y]_{p.i} \to v_1 \wedge \Delta \mid \psi_1 \rightsquigarrow_{\sigma}^{*} nil \mid \psi$, where each one of x and y appears exactly twice in the narrowing path. Let $G' = u_1|_{p.i} \to^1 y, u_1[y]_{p.i} \to v_1 \wedge \Delta \mid \psi_1$. As $vars(u_1|_p, u_1[x]_p) = vars(u_1) \cup \{x\}$ and $vars(u_1|_{p.i}, u_1[y]_{p.i}) = vars(u_1) \cup \{y\}$ then $vars(G') = (vars(G) \cup \{y\}) \setminus \{x\}$. Then $dom(\rho) = vars(G\sigma) \setminus \{x\} = vars(G\sigma) \setminus \{y\}$ so, by I.H., $(\sigma\rho)_{vars(G)}$ is a solution for $\bigwedge_{i=1}^{n} u_i \to v_i \mid \psi_1$ in $\to_{R/E}$ and there exists a substitution $\rho_y : \{y\} \to \mathcal{T}_\Sigma$ such that $(\rho \cup \rho_y)_{vars(G')}$ is a solution for G' in $\to_{R,B}$ so, as $(\rho \cup \rho_y)_{vars(G')} = (\rho \cup \rho_y)$ when applied to any term in G', $u_1(\rho \cup \rho_y)|_{p.i} \to^1_{R,B} y(\rho \cup \rho_y)$, $u_1(\rho \cup \rho_y)[y(\rho \cup \rho_y)]_{p.i} \to_{R,B} v_1(\rho \cup \rho_y)$ and $u_i(\rho \cup \rho_y) \to v_i(\rho \cup \rho_y)$, for $i \leq 2 \leq n$. As y appears exactly twice in G', this is equivalent to: $u_1\rho|_{p.i} \to^1_{R,B} y\rho_y$, $u_1\rho[y\rho_y]_{p.i} \to_{R,B} v_1\rho$ and $u_i\rho \to v_i\rho$, for $i \leq 2 \leq n$. Let $\rho_x = \{x \mapsto (u_1\rho[y\rho_y]_{p.i})|_p\}$. Then:

   (a) as $u_1\rho|_{p.i} \to^1_{R,B} y\rho_y$ then $u_1\rho|_p \to^1_{R,B} (u_1\rho[y\rho_y]_{p.i})|_p$, i.e. $u_1\rho|_p \to^1_{R,B} x\rho_x$, and

   (b) as $u_1\rho[y\rho_y]_{p.i} \to_{R,B} v_1\rho$ then $x\rho_x \to_{R,B} v_1\rho$.

   As x appears exactly twice in G, this is equivalent to $u_1(\rho \cup \rho_x)|_p \to^1_{R,B} x(\rho \cup \rho_x)$, $x(\rho \cup \rho_x) \to_{R,B} v_1(\rho \cup \rho_x)$, and $u_i(\rho \cup \rho_x) \to v_i(\rho \cup \rho_x)$, for $i \leq 2 \leq n$, so $(\rho \cup \rho_x)_{vars(G)}$ is a solution for G in $\to_{R,B}$.

3. Rule unification: $G = u_1 \to v_1 \wedge \Delta \mid \psi_1 \rightsquigarrow_{[u],\sigma_1} \Delta\sigma_1 \mid \psi_2 \rightsquigarrow_{\beta}^{*} nil \mid \psi$, where $\beta = \sigma_2 \cdots \sigma_m$ and $\sigma = \sigma_1\beta$. Consider the canonical path $G = u_1 \to v_1 \wedge \Delta \mid \chi_1 \rightsquigarrow_{[u],\sigma_1} \Delta\sigma_1 \mid \chi_2 \rightsquigarrow_{\beta}^{*} nil \mid \chi$, where $\chi_1 = \psi_1$. By Proposition 11, $E_0 \vdash \psi \Leftrightarrow \chi$ so, as ρ is a substitution such that $\psi\rho$ is satisfiable then $\chi\rho$ is also satisfiable. As $dom(\rho) = vars(G\sigma)$ then the proof can be done over the canonical path. The first narrowing step is as in the base case, with $\chi_2 = (\chi_1 \wedge \phi^\circ)\sigma_1$. Let $G' = \Delta \mid \chi_1 \wedge \phi^\circ$. As in the base case $\phi^\circ = (\bigwedge_{i=1}^{l} x_i = v_1|_{q_i})$, and σ_1 in $CSU_B(u_1 = v_1^\circ)$, so $u_1\sigma_1 =_B v_1^\circ\sigma_1$.

   Let $\rho = \rho_1 \cup \rho_2$, with $\rho_1 = \rho_{vars(G\sigma) \cap vars(G'\sigma)}$ and $\rho_2 = \rho_{vars(G\sigma) \setminus vars(G'\sigma)}$. As ρ is a substitution such that $\chi\rho$ is satisfiable, then $\chi\rho_1$, a more general formula, is also satisfiable, so there exists a substitution $\rho' : (vars(\chi) \setminus vars(G\sigma)) \cap vars(G'\sigma) \to \mathcal{T}_{\Sigma_0}$ such that $\chi(\rho_1 \cup \rho')$ is satisfiable. Let δ be a substitution $\delta : vars(G'\sigma(\rho_1 \cup \rho')) \to \mathcal{T}_\Sigma$, which must exist because all signatures have non-empty sorts, and let $\gamma = \rho_1 \cup \rho' \cup \delta$. Then $dom(\gamma) = vars(G'\sigma)$, $ran(\gamma) = \emptyset$ and $\chi\gamma$, equal to $\chi(\rho_1 \cup \rho')$, is satisfiable.

   $G'\sigma_1 \rightsquigarrow_{\beta}^{*} nil \mid \chi$, $\beta = \sigma_2 \cdots \sigma_m$, $dom(\gamma) = vars(G'\sigma) = vars((G'\sigma_1)\beta)$, and $\chi\gamma$ is satisfiable so, by I.H., $(\beta\gamma)_{vars(G'\sigma_1)}$ is a solution for $G'\sigma_1$ in $\to_{R,B}$.

   As, trivially, $(\beta\gamma)_{vars(G'\sigma_1)} = \beta\gamma$ when applied to $G'\sigma_1$, then $u_i\sigma_1\beta\gamma \to_{R,B} v_i\sigma_1\beta\gamma$, for $2 \leq i \leq n$, and $E_0 \vdash (\chi_1 \wedge \phi^\circ)\sigma_1\beta\gamma$. Now, as $\sigma_1\beta = \sigma$ then $u_i\sigma\gamma \to_{R,B} v_i\sigma\gamma$, for $2 \leq i \leq n$, and $E_0 \vdash (\chi_1 \wedge \phi^\circ)\sigma\gamma$, so also $E_0 \vdash \chi_1\sigma\gamma$ and $E_0 \vdash \phi^\circ\sigma\gamma$, ground formulas.

   As $dom(\rho_2) = vars(G\sigma) \setminus vars(G'\sigma)$ and $dom(\gamma) = vars(G'\sigma)$, then $dom(\rho_2 \cup \gamma) = vars(G\sigma) \cup vars(G'\sigma)$. But $\rho_2 \cup \gamma = \rho_2 \cup \rho_1 \cup \rho' \cup \delta = \rho \cup \rho' \cup \delta$, where $dom(\rho \cup \rho' \cup \delta) =$

$vars(G\sigma) \cup vars(G'\sigma)$, so $u_i\sigma(\rho_2 \cup \gamma) \rightarrow_{R,B} v_i\sigma(\rho_2 \cup \gamma)$, for $2 \leq i \leq n$, $E_0 \vdash \chi_1\sigma(\rho_2 \cup \gamma)$, and $E_0 \vdash \phi^\circ\sigma(\rho_2 \cup \gamma)$, i.e., $u_i\sigma(\rho \cup \rho' \cup \delta) \rightarrow_{R,B} v_i\sigma(\rho \cup \rho' \cup \delta)$, for $2 \leq i \leq n$, $E_0 \vdash \chi_1\sigma(\rho \cup \rho' \cup \delta)$, and $E_0 \vdash \phi^\circ\sigma(\rho \cup \rho' \cup \delta)$. Then, as $vars(\chi_1\sigma) \cup \bigcup_{i=1}^n vars(u_i) \cup \bigcup_{i=1}^n vars(v_i) \subseteq vars(G\sigma) = dom(\rho)$, $u_i\sigma\rho \rightarrow_{R,B} v_i\sigma\rho$, for $2 \leq i \leq n$, and $E_0 \vdash \chi_1\sigma\rho$.

As $\phi^\circ = (\bigwedge_{i=1}^l x_i = v_1|_{q_i})$ and $E_0 \vdash \phi^\circ\sigma(\rho \cup \rho' \cup \delta)$ then $E_0 \vdash \bigwedge_{i=1}^l x_i\sigma(\rho \cup \rho' \cup \delta) = v_1|_{q_i}\sigma(\rho \cup \rho' \cup \delta)$, but $vars(v_1\sigma) \subseteq vars(G\sigma) = dom(\rho)$, so $E_0 \vdash \bigwedge_{i=1}^l x_i\sigma(\rho \cup \rho' \cup \delta) = v_1|_{q_i}\sigma\rho$. As $u_1\sigma_1 =_B v_1^\circ\sigma_1$ and $\sigma = \sigma_1\beta$ then also $u_1\sigma(\rho \cup \rho' \cup \delta) =_B v_1^\circ\sigma(\rho \cup \rho' \cup \delta)$. Now, as $v_1^\circ = v_1[x_1]_{q_1} \cdots [x_l]_{q_l}$ and $vars(u_1\sigma) \subseteq vars(G\sigma) = dom(\rho)$, then $u_1\sigma\rho = u_1\sigma(\rho \cup \rho' \cup \delta) =_B v_1^\circ\sigma(\rho \cup \rho' \cup \delta) = (v_1[x_1]_{q_1} \cdots [x_l]_{q_l})\sigma(\rho \cup \rho' \cup \delta) = v_1\sigma\rho[x_1\sigma(\rho \cup \rho' \cup \delta)]_{q_1} \cdots [x_l\sigma(\rho \cup \rho' \cup \delta)]_{q_l}) =_{E_0}$
$(v_1\sigma\rho[v_1|_{q_1}\sigma\rho]_{q_1} \cdots [v_1|_{q_l}\sigma\rho]_{q_l}) = v_1\sigma\rho$, i.e., $u_1\sigma\rho =_E v_1\sigma\rho$, so $u_1\sigma\rho \rightarrow_{R,B} v_1\sigma\rho$ and $(\sigma\rho)_{vars(G)}$ is a solution for G in $\rightarrow_{R,B}$.

4. Rule rewrite: $G = u_1|_p \rightarrow^1 x, u_1[x]_p \rightarrow v_1 \wedge \Delta \mid \psi_1 \rightsquigarrow_{[r],\sigma_1} \Delta_2 \mid \psi_2 \rightsquigarrow_\beta^* nil \mid \psi$, where $\beta = \sigma_2 \cdots \sigma_m$ and $\sigma = \sigma_1\beta$. Consider the canonical path $G = u_1|_p \rightarrow^1 x, u_1[x]_p \rightarrow v_1 \wedge \Delta \mid \chi_1 \rightsquigarrow_{[u],\sigma_1} \Delta_2 \mid \chi_2 \rightsquigarrow_\beta^* nil \mid \chi$, where $\chi_1 = \psi_1$, as in the previous case. The first narrowing step uses a rule $c : l \rightarrow r \; if \; \bigwedge_{i=1}^k l_i \rightarrow r_i \mid \phi$ in R, where $abstract_{\Sigma_1}((u_1|_p, l)) = \langle \lambda(\bar{x}, \bar{y}).(u^\circ, l^\circ); (\theta_u^\circ, \theta_l^\circ); (\phi_u^\circ, \phi_l^\circ) \rangle$, let $\phi_1 = \phi \wedge \phi_u^\circ \wedge \phi_l^\circ$. This narrowing step has the form: $u_1|_p \rightarrow^1 x, u_1[x]_p \rightarrow v_1 \wedge \Delta \mid \psi_1 \rightsquigarrow_{[r],\sigma_1} (\bigwedge_{i=1}^k l_i \rightarrow r_i \wedge u_1[r]_p \rightarrow v_1 \wedge \Delta)\sigma_1 \mid (\chi_1 \wedge \phi_1)\sigma_1$, where σ_1 in $CSU_B(u^\circ = l^\circ)$, so $u^\circ\sigma_1 =_B l^\circ\sigma_1$. As x appears exactly twice in G and $ran(\sigma)$ is away from all the variables that have appeared before in the narrowing path, then $dom(\sigma) \cap \{x\} = \emptyset$ and $ran(\sigma) \cap \{x\} = \emptyset$. Let $G' = \bigwedge_{i=1}^k l_i \rightarrow r_i \wedge u_1[r]_p \rightarrow v_1 \wedge \Delta \mid \chi_1 \wedge \phi_1$.

Let $\rho = \rho_1 \cup \rho_2$, with $\rho_1 = \rho_{(vars(G\sigma) \setminus \{x\}) \cap vars(G'\sigma)}$ and $\rho_2 = \rho_{(vars(G\sigma) \setminus \{x\}) \setminus vars(G'\sigma)}$. As ρ is a substitution such that $\chi\rho$ is satisfiable, then $\chi\rho_1$, a more general formula, is also satisfiable, so there exists a substitution $\rho' : (vars(\chi) \setminus (vars(G\sigma) \setminus \{x\})) \cap vars(G'\sigma) \rightarrow \mathcal{T}_{\Sigma_0}$ such that $\chi(\rho_1 \cup \rho')$ is satisfiable. Let δ be a substitution $\delta : vars(G'\sigma(\rho_1 \cup \rho')) \rightarrow \mathcal{T}_\Sigma$, which must exist because all signatures have non-empty sorts, and let $\gamma = \rho_1 \cup \rho' \cup \delta$. As ρ and ρ_1 are ground substitutions then $dom(\gamma) = vars(G'\sigma)$ and $ran(\gamma) = \emptyset$. Also, as $vars(\chi) \cap vars(G'\sigma) \subseteq dom(\rho_1 \cup \rho)$ then $\chi\gamma = \chi(\rho_1 \cup \rho')$, so $\chi\gamma$ is satisfiable.

$G'\sigma_1 \rightsquigarrow_\beta^* nil \mid \chi$, $\beta = \sigma_2 \cdots \sigma_m$, $dom(\gamma) = vars(G'\sigma) = vars((G'\sigma_1)\beta)$, and $\chi\gamma$ is satisfiable so, by I.H., $(\beta\gamma)_{vars(G'\sigma_1)}$ is a solution for $G'\sigma_1$ in $\rightarrow_{R,B}$. As $(\beta\gamma)_{vars(G'\sigma_1)} = \beta\gamma$ when applied to $G'\sigma_1$ then $l_i\sigma\gamma \rightarrow_{R,B} r_i\sigma\gamma$, for $1 \leq i \leq k$, $u_j\sigma\gamma \rightarrow_{R,B} v_j\sigma\gamma$, for $2 \leq j \leq n$, $u_1[r]_p\sigma\gamma \rightarrow_{R,B} v_1\sigma\gamma$, and $E_0 \vdash (\chi_1 \wedge \phi_1)\sigma\gamma$, so also $E_0 \vdash \chi_1\sigma\gamma$ and $E_0 \vdash \phi_1\sigma\gamma$, ground formulas.

Let $\alpha = \rho \cup \rho' \cup \delta$. In the same way that has been proven for rule unification, as $\rho_2 \cup \gamma = \alpha$, then $u_j\sigma\rho \rightarrow_{R,B} v_j\sigma\rho$, for $2 \leq j \leq n$, and $E_0 \vdash \chi_1\sigma\rho$.

As $u_1[r]_p\sigma\gamma \rightarrow_{R,B} v_1\sigma\gamma$, both ground terms, and $vars(v_1\sigma) \subseteq (vars(G\sigma) \setminus \{x\}) = dom(\rho)$ then $u_1[r]_p\sigma(\rho_2 \cup \gamma) = u_1[r]_p\sigma\gamma \rightarrow_{R,B} v_1\sigma\gamma = v_1\sigma(\rho_2 \cup \gamma)$, i.e., $u_1[r]_p\sigma\alpha \rightarrow_{R,B} v_1\sigma\alpha = v_1\sigma\rho$.

As $E_0 \vdash \phi_1\sigma_1\gamma$ then $E_0 \vdash (\phi \wedge \phi_u^\circ \wedge \phi_l^\circ)\sigma_1\alpha$, so $u^\circ\sigma\alpha =_{E_0} u_1|_p\sigma\alpha$ and $l^\circ\sigma\alpha =_{E_0} l\sigma\alpha$. Also as $l_i\sigma\gamma \rightarrow_{R,B} r_i\sigma\gamma$ then $l_i\sigma\alpha \rightarrow_{R,B} r_i\sigma\alpha$ and, by Theorem 11, $l_i\sigma\alpha \rightarrow_{R/\mathcal{E}} r_i\sigma\alpha$, for $1 \leq i \leq k$.

As $u^\circ\sigma_1 =_B l^\circ\sigma_1$ then $u_1|_p\sigma\alpha =_{E_0} u^\circ\sigma\alpha =_B l^\circ\sigma\alpha =_{E_0} l\sigma\alpha$, i.e., $u_1|_p\sigma\alpha =_{\mathcal{E}} l\sigma\alpha$, so $u_1|_p\sigma\alpha \to^1_{R/\mathcal{E}} r\sigma\alpha$, with rule c, hence there exist $t \in \mathcal{H}_\Sigma$ such that $u_1|_p\sigma\alpha \to^1_{R,B} t =_{\mathcal{E}} r\sigma\alpha$, again by Theorem 11. In consequence, since t is ground, $u_1\sigma\alpha \to^1_{R,B} u_1[t]_p\sigma\alpha =_{\mathcal{E}} u_1[r]_p\sigma\alpha$ so, as $vars(u_1\sigma) \subseteq (vars(G\sigma) \setminus \{x\}) = dom(\rho)$, $u_1\sigma\rho = u_1\sigma\alpha \to^1_{R,B} u_1[t]_p\sigma\alpha =_{\mathcal{E}} u_1[r]_p\sigma\alpha$.

Using several times Theorem 11, as $u_1[r]_p\sigma\alpha \to_{R,B} v_1\sigma\rho$ then $u_1[r]_p\sigma\alpha \to_{R/\mathcal{E}} v_1\sigma\rho$ so, as $u_1[t]_p\sigma\alpha =_{\mathcal{E}} u_1[r]_p\sigma\alpha$, also $u_1[t]_p\sigma\alpha \to_{R/\mathcal{E}} v_1\sigma\rho$, and then there exists $t' \in \mathcal{H}_\Sigma$ such that $u_1[t]_p\sigma\alpha \to_{R,B} t' =_E v_1\sigma\rho$.

As $u_1\sigma\rho \to^1_{R,B} u_1[t]_p\sigma\alpha$ and $u_1[t]_p\sigma\alpha \to_{R,B} t' =_E v_1\sigma\rho$ then $u_1\sigma\rho \to_{R,B} v_1\sigma\rho$, so $(\sigma\rho)_{vars(G)\setminus\{x\}}$ is a solution for $\bigwedge_{i=1}^n u_i \to v_i \mid \psi_1$ in $\to_{R,B}$.

Take $\rho_x = \{x \mapsto r\sigma\alpha\}$. As $dom(\sigma) \cap \{x\} = \emptyset$ and $ran(\sigma) \cap \{x\} = \emptyset$, then $u_1|_p\sigma(\rho \cup \rho_x) = u_1|_p\sigma\rho = u_1|_p\sigma\alpha \to^1_{R,B} r\sigma\alpha = x\rho_x = x\sigma\rho_x = x\sigma(\rho \cup \rho_x)$. Also, as $vars(u_1\sigma, v1\sigma) \subseteq (vars(G\sigma) \setminus \{x\}) = dom(\rho)$ and $ran(\rho) = \emptyset$, $u_1[x]_p\sigma(\rho \cup \rho_x) = u_1\sigma\rho[r\sigma\alpha]_p = u_1\sigma\alpha[r\sigma\alpha]_p = u_1[r]_p\sigma\alpha \to_{R,B} v_1\sigma\rho = v_1\sigma(\rho \cup \rho_x)$. Then, as $vars(\Delta) \cap dom(\rho_x) = \emptyset$, $(\sigma(\rho \cup \rho_x))_{vars(G)}$ is a solution for G in $\to_{R,B}$.

□

Theorem 14 (Weak Completeness in $\to^1_{R/\mathcal{E}}$ of the Calculus for Reachability Problems)**.** *Given a closed under B-extensions rewrite theory $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ with built-in subtheory (Σ_0, E_0), where $\mathcal{E} = E_0 \cup B$, and a reachability problem $P = \bigwedge_{i=1}^n u_i \to v_i \mid \phi$, if σ is an idempotent $R/\mathcal{E}$-normalized solution for P in $\to^1_{R/\mathcal{E}}$ then there exist a formula $\psi \in QF(\mathcal{X}_0)$, and substitutions γ and δ, such that $P \rightsquigarrow^*_\gamma nil \mid \psi$, $\sigma = (\gamma\delta)_{vars(P)}$, and $\psi\delta$ is satisfiable.*

Proof. By Theorem 11, $u_i\sigma \to_{R,B} v_i\sigma$, for $1 \le i \le n$. The proof is by induction on the number of R, B-rewrite steps. No simplification is applied to the reachability formulas that appear in the generated path.

(i) Base case: zero rewrite steps. The proof is by induction on n, the size of the conjunction.

- Base case, $n = 1$: $P = u_1 \to v_1 \mid \phi$, $u_1\sigma =_{\mathcal{E}} v_1\sigma$, and $E_0 \vdash \phi\sigma$. By Proposition 10, there exists a term w such that $u_1\sigma =_B w =_{E_0} v_1\sigma$. As σ is idempotent then also $u_1\sigma =_B w\sigma =_{E_0} v_1\sigma$. Let $abstract_{\Sigma_1}(v_1) = \langle \lambda\bar{x}.v_1^\circ; \theta^\circ; \phi^\circ\rangle$, with $\bar{x} = \{x_1, \ldots, x_n\}$, and let $top_{\Sigma_0}(v_1^\circ) = \{q_1, \ldots, q_n\}$. Let $\sigma^\circ = \sigma \cup \bigcup_{i=1}^n \{x_i \mapsto w|_{q_i}\sigma\}$. By Proposition 8, $top_{\Sigma_0}(v_1^\circ\sigma) = top_{\Sigma_0}(v_1)$. Then, by Proposition 7, $w\sigma = v_1\sigma[w|_{q_1}\sigma]_{q_1} \cdots [w|_{q_n}\sigma]_{q_n}$, where $w|_{q_i}\sigma =_{E_0} v_1|_{q_i}\sigma$, for $1 \le i \le n$. Let $\sigma^\circ = \sigma \cup \bigcup_{i=1}^n \{x_i \mapsto w|_{q_i}\sigma\}$. Then $v_1^\circ\sigma^\circ = (v_1[x_1]_{q_1} \cdots [x_n]_{q_n})\sigma^\circ = v_1\sigma[w|_{q_1}\sigma]_{q_1} \cdots [w|_{q_n}\sigma]_{q_n} = w\sigma =_B u_1\sigma = u_1\sigma^\circ$, because $vars(P) \cap \bar{x} = \emptyset$ implies $v_1\sigma^\circ = v_1\sigma$ and $u_1\sigma = u_1\sigma^\circ$. As $u_1\sigma^\circ =_B v_1^\circ\sigma^\circ$, there exist substitutions γ_1 and δ_1 such that $\gamma_1 \in CSU_B(u_1 = v_1^\circ)$ and $\sigma^\circ = \gamma_1\delta_1$, so $\sigma = (\gamma_1\delta_1)_{vars(P)}$.

  $vars(\phi) \cap \bar{x} = \emptyset$ implies $\phi\sigma^\circ = \phi\sigma$ so $E_0 \vdash \phi\sigma^\circ$, because $E_0 \vdash \phi\sigma$. Also, as $\phi^\circ\sigma^\circ = \bigwedge_{i=1}^n (w|_{q_i}\sigma = v_1|_{q_i}\sigma)$ because $vars(v) \cap \bar{x} = \emptyset$, $E_0 \vdash \phi^\circ\sigma^\circ$, so $E_0 \vdash (\phi \wedge \phi^\circ)\sigma^\circ$. Let $\psi = (\phi \wedge \phi^\circ)\gamma_1$. Then, as $\sigma^\circ = \gamma_1\delta_1$, ψ and $\psi\delta$ are satisfiable.

  As $\gamma_1 \in CSU_B(u = v^\circ)$ and ψ is satisfiable then $u_1 \to v_1 \mid \phi \rightsquigarrow_{[u],\gamma_1} nil \mid \psi$.

- Induction case, $n > 1$: $P = \bigwedge_{i=1}^{n} u_i \to v_i \mid \phi$, and $u_i\sigma =_{\mathcal{E}} v_i\sigma$, for $1 \leq i \leq n$. Using the same proof of the base case $u_1 \to v_1 \wedge \bigwedge_{i=2}^{n} u_i \to v_i \mid \phi \rightsquigarrow_{[u],\gamma_1} \bigwedge_{i=2}^{n} u_i\gamma_1 \to v_i\gamma_1 \mid (\phi \wedge \phi^\circ)\gamma_1$ and $\sigma^\circ = \gamma_1\delta_1$. Let $G' = \bigwedge_{i=2}^{n} u_i\gamma_1 \to v_i\gamma_1 \mid \phi\gamma_1 \wedge \phi^\circ\gamma_1$. Then δ_1 is a solution of G' because $\sigma^\circ = \gamma_1\delta_1$, $E_0 \vdash (\phi \wedge \phi^\circ)\sigma^\circ$, and $u_i\sigma^\circ = u_i\sigma =_E v_i\sigma = v_i\sigma^\circ$, for $2 \leq i \leq n$. By I.H., there exist a formula $\psi \in QF(\mathcal{X}_0)$, and substitutions γ_2 and δ such that $G' \rightsquigarrow^*_{\gamma_2} nil \mid \psi$, $\delta_1 = (\gamma_2\delta)_{vars(G')}$, and $\psi\delta$ is satisfiable. Let $\gamma = \gamma_1\gamma_2$. Then $P \rightsquigarrow_{[u],\gamma_1} G' \rightsquigarrow^*_{\gamma_2} nil \mid \psi$, i.e., $P \rightsquigarrow^*_{\gamma} nil \mid \psi$, and $(\gamma\delta)_{vars(P)} = (\gamma_1\gamma_2\delta)_{vars(P)} = (\gamma_1\delta_1)_{vars(P)} = (\sigma^\circ)_{vars(P)} = \sigma$.

(*ii*) Induction case: at least there is one rewrite step. If there are not rewrite steps in $u_1\sigma \to_{R,B} v_1\sigma$ then reorder the reachability problem in a way that there is one rewrite step in the first subproblem. Let $\Delta = \bigwedge_{i=2}^{n} u_i \to v_i$. Then $P = u_1 \to v_1 \wedge \Delta \mid \phi$, $u_1\sigma \to^1_{R,B} u' \to_{R,B} w =_{\mathcal{E}} v_1\sigma$, $E_0 \vdash \phi\sigma$, and $u_i\sigma \to_{R,B} v_i\sigma$, for $2 \leq i \leq n$. As σ is idempotent then also $u_1\sigma \to^1_{R,B} u'\sigma \to_{R,B} w\sigma =_{\mathcal{E}} v_1\sigma$.

$u_1\sigma \to^1_{R,B} u'\sigma$ using a rule $c : l \to r \; if \; \bigwedge_{j=1}^{m} l_j \to r_j \mid \phi'$ in R, let $abstract_{\Sigma_1}((u_1|_p, l)) = \langle \lambda(\bar{x}, \bar{y}).(u^\circ, l^\circ); (\theta_u^\circ, \theta_l^\circ); (\phi_u^\circ, \phi_l^\circ) \rangle$, with normal form $c^\circ : l^\circ \to r \; if \; \bigwedge_{j=1}^{m} l_j \to r_j \mid \phi' \wedge \phi_l^\circ$, and some substitution δ at a position p in $pos_{\Sigma_1}(u_1\sigma)$, such that $rep(u_1\sigma|_p) =_B l^\circ\delta$, $E_0 \vdash (\phi' \wedge \phi_l^\circ)\delta$, $u' = u_1\sigma[r\delta]_p$, and $l_j\delta \to_{R,B} r_j\delta$, for $1 \leq j \leq m$. It has to be the case that $p \in pos_{\Sigma_1}(u_1)$ because otherwise there exists some integer k, $1 \leq k \leq m$, such that either $u_1|_p = y_k$ or the rewriting takes place at some position in a subterm of a term of the form $y_k\sigma$. In both cases σ would be neither R, B-normalized, nor $R/\mathcal{E}$-normalized.

As $u_1|_p\sigma =_{\mathcal{E}} l^\circ\delta$ and $E_0 \vdash \phi_l^\circ\delta$, so $l^\circ\delta =_{E_0} l\delta$, ground terms, then $u_1|_p(\sigma \cup \delta) =_{\mathcal{E}} l(\sigma \cup \delta)$. By Lemma 9, there exists a substitution γ such that $u^\circ\gamma =_B l^\circ\gamma$, $E_0 \vdash (\phi_u^\circ \wedge \phi_l^\circ)\gamma$ and $\sigma \cup \delta =_{E_0} \gamma_{dom(\sigma\cup\delta)}$.

Then, there exist idempotent substitutions α and β such that $\alpha \in CSU_B(u^\circ = l^\circ)$, $E_0 \vdash (\phi_u^\circ \wedge \phi_l^\circ)\alpha$ and $\gamma = \alpha \cdot \beta$, so $\sigma \cup \delta =_{E_0} (\alpha \cdot \beta)_{dom(\sigma\cup\delta)}$. As $E_0 \vdash \phi\sigma$ and $E_0 \vdash \phi'\delta$ then $E_0 \vdash (\phi \wedge \phi' \wedge \phi_u^\circ \wedge \phi_l^\circ)\alpha \cdot \beta$.

Then $u_1 \to v_1 \wedge \Delta \mid \phi \rightsquigarrow_{[t]} u_1 \to^1 z_\kappa, z_\kappa \to v_1 \wedge \Delta \mid \phi \rightsquigarrow^*_{[c]} u_1|_p \to^1 z'_{\kappa'}, u_1[z'_{\kappa'}]_p \to v_1 \wedge \Delta \mid \phi \rightsquigarrow_{[w],c,\alpha} \bigwedge_{j=1}^{m} l_j\alpha \to r_j\alpha \wedge u_1[r]_p\alpha \to v_1\alpha \wedge \Delta\alpha \mid (\phi \wedge \phi' \wedge \phi_u^\circ \wedge \phi_l^\circ)\alpha$.

As $\sigma \cup \delta =_{E_0} (\alpha \cdot \beta)_{dom(\sigma\cup\delta)}$ and $\to_{R,B} = \to_{R/\mathcal{E}}$, by Theorem 11, then:

1. for $1 \leq j \leq m$:

   $l_j\delta \to_{R,B} r_j\delta$ implies $l_j\delta \to_{R/\mathcal{E}} r_j\delta$. As $l_j\alpha\beta =_{E_0} l_j\delta$ and $r_j\delta =_{E_0} r_j\alpha\beta$ then also $l_j\alpha\beta \to_{R/\mathcal{E}} r_j\alpha\beta$, so $l_j\alpha\beta \to_{R,B} r_j\alpha\beta$;

2. in the same way, as $u_i\alpha\beta =_{E_0} u_i\delta$, $u_i\delta \to_{R,B} v_i\delta$, and $v_i\delta =_{E_0} v_i\alpha\beta$, then $u_i\alpha\beta \to_{R,B} v_i\alpha\beta$, for $2 \leq i \leq n$; and

3. $u_1[r]_p\alpha\beta = u_1\alpha\beta[r\alpha\beta]_p =_{E_0} u_1\sigma[r\delta]_p$, $u_1\sigma[r\delta]_p \to_{R,B} v_1\sigma$, and $v_1\sigma =_{E_0} v_1\alpha\beta$ implies $u_1[r]_p\alpha\beta \to_{R,B} v_1\alpha\beta$.

Then $\beta_{vars(G')}$ is a solution of the reachability problem $G' = \bigwedge_{i=1}^{n} l_i\alpha \to r_i\alpha \wedge u_1[r]_p\alpha \to v_1\alpha \wedge \Delta\alpha \mid (\phi \wedge \phi' \wedge \phi_u^\circ \wedge \phi_l^\circ)\alpha$ that takes one less rewrite step than those taken to prove $P\sigma$ so, by I.H., there exist a formula $\psi \in QF(\mathcal{X}_0)$, and substitutions α' and δ' such that $G' \rightsquigarrow^*_{\alpha'} nil \mid \psi$, $\psi\delta'$ is satisfiable, and $\beta_{vars(G')} = (\alpha'\delta')_{vars(G')}$, so $P \rightsquigarrow_{\alpha} G' \rightsquigarrow^*_{\alpha'} nil \mid \psi$, i.e., $P \rightsquigarrow^*_{\alpha\alpha'} nil \mid \psi$. Let $\gamma = \alpha\alpha'$ and $\delta = \delta' \cup \{y \mapsto y\beta \mid y \in dom(\beta) \setminus vars(G')\}$.

For all variables x in $vars(P)$ and all variables y in $vars(x\alpha)$:

- if $y \in vars(G')$ then $y\alpha'\delta = y(\alpha'\delta')_{vars(G')} = y(\beta)_{vars(G')} = y\beta$, and

- if $y \notin vars(G')$ then $y\alpha' = y$. As $y \in vars(x\alpha)$ implies $y \in dom(\beta)$, then $y \in dom(\beta) \setminus vars(G')$, so $y\alpha'\delta = y\delta = y\beta$.

Then $x\gamma\delta = x\alpha\alpha'\delta = x\alpha\beta = x\sigma$, so $\sigma = (\gamma\delta)_{vars(P)}$. Also $vars(\psi) \cap (dom(\beta) \setminus vars(G')) = \emptyset$, because $vars(\psi)$ may only have variables in $vars(G')$ together with new fresh variables not in $vars(P)$, hence also not in $dom(\beta)$, so $\psi\delta = \psi\delta'$ and $\psi\delta$ is satisfiable. □

Chapter 6

Strategies in conditional narrowing modulo SMT plus axioms

The contents of Chapter 5 are extended with the addition of two elements, the first one (strategies) for explicit control of the rewrite steps that can be applied, and the second one (parameters) for an enhanced expressivity of the reachability problems. The addition of the strategies has forced the development of a completely new narrowing calculus [AMPP23]. Section 6.8 presents the prototype for narrowing with strategies developed to implements this calculus. We show in Section 6.7.3 a link between the use of strategies and the SNR-rewrite relation from Chapter 4. We briefly present the aforementioned two elements:

1. *Strategies.* In [AMPP17] we found several sources of state space explosion, namely:

   (a) the order of application of the rules,

   (b) the application of unneeded rules, and

   (c) that checking a SMT constraint that applied to any state was only possible for candidate final states,

   that even prevented the state space of some problems from being finite. These problems can be addressed with the use of strategies that, as a side effect, also allow for the specification of OS equational theories beyond $E_0 \cup B$ by giving precedence to a subset of the rules (see Section 6.7.3).

2. *Parameters.* We also found out in [AMPP17] that the scope of the calculus could be broadened if we included the support for parameters in the specifications, i.e., a subset of the variables in them, either SMT or not, to be considered as *common constants* that needed to be given a value in the reachability problem, either as a prerequisite or as part of its solution, allowing, for instance, the fine tuning of a proposed specification.

The strategy language for narrowing, presented in Section 2.4, together with the proof tree based interpretation of its semantics, can be used to drive the search of solutions to reachability problems or to specify either algorithms or OS theories. It is a subset of the Maude strategy language [MOMV04, EMOMV07, RMPV21].

The main difference between the Maude strategy language and our strategy language for narrowing is the scope of the if-then-else strategy, which in Maude has the form

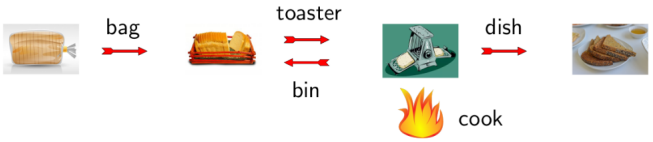

Figure 6.1: Toast cooking with strategies

$\mathtt{ST}\,?\,\mathtt{ST}_1\ :\ \mathtt{ST}_2$, where the condition is any strategy $\mathtt{ST}$. If narrowing was applied to a reachability problem $t \to v/\mathtt{ST}\,?\,\mathtt{ST}_1\ :\ \mathtt{ST}_2$, for any instance $t\delta$ of t such that $\mathtt{ST}@[t\delta]_{\mathcal{E}} = \emptyset$ we would get the new problem $t\delta \to v\delta/\mathtt{ST}_2\delta$. This means that we would have to generate all the instances of t to explore the state space.

That is why we have restricted the if-then-else strategy to open goals of the form $t \to v/\mathtt{match}\,u\,\mathtt{s.t.}\,\phi\,?\,\mathtt{ST}_1\ :\ \mathtt{ST}_2$, where we require that there exists a substitution δ such that $t =_{\mathcal{E}} u\delta$ to apply the strategy, and continuing with $t \to v/\mathtt{ST}_1\delta$, if $E_0 \vdash \phi\delta$, or $t \to v/\mathtt{ST}_2\delta$, otherwise, thus avoiding the need to generate all the instances of any symbolic term in the narrowing calculus.

The narrowing calculus developed in this chapter is completely new, and it includes the strategy language and the use of parameters, both in the rewrite theories and in the strategies. Under certain requirements, the calculus is sound and weakly complete.

6.1 Toast example

Toast cooking will be used again as a concurrency running example, but now in a slightly modified version that includes the use of parameters. A toast is well-cooked if both sides of the toast have been cooked for exactly `cookTime` (abbreviated to `ct`) seconds. No overcooking is allowed. Fresh toasts are taken from a toast bag, and they are cooked using a frying pan that can toast up to two toasts simultaneously, well cooking one side of each toast in the pan. There is a bin, where fresh toasts are put when taken from the bag. A toast in the pan can be returned to the bin, being flipped in this process. Finally, there is a dish where well-cooked toasts can be output. There is a limit of `failTime` (`ft`) seconds to reach the desired final state. In this example, `ct` and `ft` will be the parameters, i.e., they are the variables that represent the common constants of the specification that must be given a value either by the conditions of the problem or by its solution.

A `Toast` (abbreviated to `t`) can be either a `RealToast` (`rt`), represented as an ordered pair of natural numbers, each one with sort `Integer` (`i`), storing the seconds that each side has already been toasted, or an `EmptyToast` (`et`) which has a constant `zt`, representing the absence of `Toasts`; a `Pan` (`p`) is an unordered pair of `Toasts`; a `Kitchen` (`k`) has a timer, represented by a natural number, and a `Pan`; a `Bin` (`b`) is a multiset of `Toasts`; the bag and the dish are represented by natural numbers, the number of `RealToasts` in each one; the `System` (`s`) has a bag, a `Bin`, a `Kitchen`, and a dish. When a `RealToast` is in the pan, the side being toasted is represented by the first `Integer` of the ordered pair. We will use two auxiliary functions, `cook` and `toast` (in lowercase). The instructions for `Toast` cooking are the following:

1. The function call $\mathtt{cook}(x_{\mathtt{k}}, y_{\mathtt{i}})$ will return the `Kitchen` obtained from `Kitchen` $x_{\mathtt{k}}$ after $y_{\mathtt{i}}$ seconds, by calling the function $\mathtt{toast}(v_{\mathtt{t}}, y_{\mathtt{i}})$ for each `Toast` $v_{\mathtt{t}}$ in `Kitchen` $x_{\mathtt{k}}$.

2. The function call $\texttt{toast}(\texttt{zt}, y_{\texttt{i}})$ will return $\texttt{zt}$.

3. The function call $\texttt{toast}(r_{\texttt{rt}}, y_{\texttt{i}})$ will return the `RealToast` obtained from `RealToast` $r_{\texttt{rt}}$ after toasting it for $y_{\texttt{i}}$ seconds, where $y_{\texttt{i}} > 0$, only if the side of $r_{\texttt{rt}}$ that is in contact with the `Pan` gets well-cooked.

4. A fresh `RealToast` can pass from a non-empty bag, represented in the `System` with a positive `Integer`, to the `Bin`.

5. A `RealToast` can pass from the `Bin` to the `Pan` if there is room in the `Pan`.

6. A `Kitchen` with at least one `RealToast` in the `Pan` can `cook` the `RealToast`s that are laying on the pan any given `Integer` number of seconds.

7. A `RealToast` in the `Pan` can be returned to the `Bin`, where it is flipped. This is the only way that a toast gets flipped.

8. A well-cooked `RealToast` can be taken out to the dish.

6.1.1 Signature and order-sorted theory

Signature

In the toast cooking example, omitting the implied kind for each connected component of S, $\Sigma = (S, \leq, F)$ is very similar to the one in the previous chapter:

$S = \{\texttt{Boolean}, \texttt{Integer}, \texttt{RealToast}, \texttt{EmptyToast}, \texttt{Toast}, \texttt{Pan}, \texttt{Kitchen}, \texttt{Bin}, \texttt{System}\}$,
$\leq = \{(\texttt{RealToast}, \texttt{Toast}), (\texttt{EmptyToast}, \texttt{Toast}), (\texttt{Toast}, \texttt{Bin})\}$,
$F = \{\{[\_, \_]\}_{\texttt{i i,rt}}, \{\_\_\}_{\texttt{t t,p}}, \{\_;\_\}_{\texttt{b b,b}}, \{\_; \_\}_{\texttt{i p,k}}, \{\texttt{cook}\}_{\texttt{k i,[k]}}, \{\texttt{toast}\}_{\texttt{t i,[t]}},$
$\{\_/\_/\_/\_\}_{\texttt{i b k i,s}}, \{\texttt{zt}\}_{\texttt{et}}\}$.

Order-sorted theory

We write $\mathcal{E}$ as a shortcut for $E_0 \cup B$ throughout this chapter. For the cooking example, $\Sigma = (S, \leq, F)$ and $\mathcal{E}$ are the same ones from the previous chapter, i.e., E_0 is the set of equations for integer arithmetic and Boolean calculus (not displayed), and B consists of the equations:

- $(x_{[\texttt{b}]}; y_{[\texttt{b}]}); z_{[\texttt{b}]} = x_{[\texttt{b}]}; (y_{[\texttt{b}]}; z_{[\texttt{b}]})$
- $x_{[\texttt{b}]}; y_{[\texttt{b}]} = y_{[\texttt{b}]}; x_{[\texttt{b}]}$
- $x_{[\texttt{b}]}; \texttt{zt} = x_{[\texttt{b}]}$
- $x_{[\texttt{t}]} y_{[\texttt{t}]} = y_{[\texttt{t}]} x_{[\texttt{t}]}$

stating that the `bin` is a multiset and that the position of the `toasts` in the `pan` is irrelevant.

6.1.2 Rewrite theory

The uniqueness requirement for the labels of the rules in a rewrite theory will be relaxed, but not removed, later in this chapter. From now on we will write "rewrite theory" as a shortcut for "conditional rewrite theory with built-in subtheory plus axioms".

In our running example for this chapter, R is the following translation of the instructions for cooking, shown at the beginning of this section. The subscript `i` will be omitted from now on, for a better readability of the examples:

$[kitchen] : y; h_{\mathtt{rt}}\, v_{\mathtt{t}} \;\rightarrow\; \mathtt{cook}(y; h_{\mathtt{rt}}\, v_{\mathtt{t}}, z)$ *if* $z > 0$
$[cook] : \mathtt{cook}(y; h_{\mathtt{rt}}\, v_{\mathtt{t}}, z) \;\rightarrow\; y + z; h'_{\mathtt{rt}}\, v'_{\mathtt{t}}$ *if* $\mathtt{toast}(h_{\mathtt{rt}}, z) \;\rightarrow\; h'_{\mathtt{rt}} \wedge \mathtt{toast}(v_{\mathtt{t}}, z) \;\rightarrow\; v'_{\mathtt{t}}$
$[toast1] : \mathtt{toast}(\mathtt{zt}, z) \;\rightarrow\; \mathtt{zt}$
$[toast2] : \mathtt{toast}([a, b], z) \;\rightarrow\; [a + z, b]$ *if* $a \geq 0 \wedge a + z = \mathtt{ct}$
$[bag] : n/x_{\mathtt{b}}/g_{\mathtt{k}}/ok \;\rightarrow\; (n-1)/[0,0]; x_{\mathtt{b}}/g_{\mathtt{k}}/ok$ *if* $n > 0$
$[pan] : n/h_{\mathtt{rt}}; x_{\mathtt{b}}/y; \mathtt{zt}\, v_{\mathtt{t}}/ok \;\rightarrow\; n/x_{\mathtt{b}}/y; h_{\mathtt{rt}}\, v_{\mathtt{t}}\, /ok$
$[bin] : n/x_{\mathtt{b}}/y; [a, b]\, v_{\mathtt{t}}/ok \;\rightarrow\; n/[b, a]; x_{\mathtt{b}}/y; \mathtt{zt}\, v_{\mathtt{t}}\, /ok$
$[dish] : n/x_{\mathtt{b}}/y; [\mathtt{ct}, \mathtt{ct}]\, v_{\mathtt{t}}/ok \;\rightarrow\; n/x_{\mathtt{b}}/\mathtt{zt}\, v_{\mathtt{t}}/ok + 1$

The Maude specification for the running example is:

```
load smtlogic

mod TOASTS is
  protecting SMTLOGIC .

  sorts RealToast EmptyToast Toast Pan Kitchen Bin System .
  subsort RealToast EmptyToast < Toast < Bin .

  vars C D N OK Y Z cookTime : Integer .
  var R : RealToast .
  vars V W V1 W1 : Toast .
  var B : Bin .
  var K : Kitchen .
  var S : System .
  var I : Int .
  var CO : Constant .
  var TE : Term .

  op zt : -> EmptyToast .
  op [_,_] : Integer Integer -> RealToast .
  op __ : Toast Toast -> Pan [comm] .
  op _;_ : Bin Bin -> Bin [comm assoc id: zt] .
  op _;_ : Integer Pan -> Kitchen .
  op cook : Kitchen Integer -> Kitchen .
  op toast : Toast Integer -> Toast .
  op _/_/_/_ : Integer Bin Kitchen Integer -> System .

  crl [kitchen] : Y ; R V => cook(Y ; R V, Z)
    if (Z > 0) = (true).Boolean [nonexec] .
```

```
  crl [cook] : cook(Y ; V W, Z) => Y + Z ; V1 W1
    if toast(V, Z) => V1 /\ toast(W, Z) => W1 .
  rl [toast1] : toast(zt, Z) => zt .
  crl [toast2] : toast([C, D], Z) => [C + Z, D]
    if (C >= 0 and C + Z === cookTime) = (true).Boolean [nonexec] .
  crl [bag] : N / B / K / OK => N - 1 / [0, 0] ; B / K / OK
    if (N > 0) = (true).Boolean .
  rl [pan] : N / R ; B / Y ; zt V / OK
    => N / B / Y ; R V / OK .
  rl [bin] : N / B / Y ; [C, D] V / OK
    => N / [D, C] ; B / Y ; zt V / OK .
  rl [dish] : N / B / Y ; [cookTime, cookTime] V / OK
    => N / B / Y ; zt V / OK + 1 [nonexec] .
endm
```

In this case, the first command loads the module `smtlogic.maude` that in turn loads the module `smt.maude` that was loaded in the example of the previous chapter. A different way in the use of the syntax of Maude to label the rules is shown here: the label goes between brackets after the `rl` or `crl` reserved word and is followed by a colon.

The parameter `cooktime` is defined as a variable here, so that it can be included in the rules, which are marked as `nonexec`, since this parameter should be given a ground value before using any rule that includes it.

6.2 Closure under B-extensions revisited

We extend the definitions given in Section 4.2.1 to the case of rewrite theories with built-in.

Let $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ be a rewrite theory, where R may have repeated labels, and let $c : l \to r \; \mathit{if}\; C$ be a rule in R. It is assumed that $vars(B) \cap vars(c) = \emptyset$. If this is not the case, *only the variables of* B will be renamed; the variables of c *will never be* renamed. We define the set of B-*extensions* of c as the set:
$Ext_B(c) = \{c : u[l]_p \to u[r]_p \; \mathit{if}\; C \mid u = v \in B \cup B^{-1} \wedge p \in pos_\Sigma(u) \setminus \{\epsilon\} \wedge CSU_B(l, u|_p) \neq \emptyset\}$
where, by definition, $B^{-1} = \{v = u \mid u = v \in B\}$.

All the rules in $Ext_B(c)$ have label c. Given two rules $c : l \to r \; \mathit{if}\; C$ and $c_1 : l' \to r' \; \mathit{if}\; C$ with the *same* condition C, c *subsumes* c_1 iff there is a substitution δ such that: (i) $dom(\delta) \cap vars(C) = \emptyset$, (ii) $l' =_B l\delta$, and (iii) $r' =_B r\delta$.

We say that $\mathcal{R}$ is *closed under* B-*extensions* iff for any rule with label c in R, each rule in $Ext_B(c)$ is subsumed by one rule with label c in R.

6.2.1 Finite closure under B-extensions of a rule

Given an equational theory $(\Sigma, \mathcal{E})$, with built-in subtheory (Σ_0, E_0), and a rule with label c, we denote by c_B the set of rules in any finite closure under B-extensions of the rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, \{c\})$.

6.2.2 Associated rewrite theory closed under B-extensions

Given a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ with no repeated rule labels, any rewrite theory $\mathcal{R}_B = (\Sigma, \mathcal{E}, \bigcup_{c \in R} c_B)$ is called an *associated rewrite theory closed under B-extensions* of $\mathcal{R}$. If $\mathcal{R}$ is an associated rewrite theory closed under B-extensions of itself then we say that $\mathcal{R}$ is *closed under B-extensions.*

Example 21. *In the toast example, if we look at the set $\{u = v \in B \cup B^{-1}\}$, there are three equations in it such that $pos_\Sigma(u) \setminus \{\epsilon\}$ is not empty:*

1. $(x_{[\mathtt{b}]}; y_{[\mathtt{b}]}); z_{[\mathtt{b}]} = x_{[\mathtt{b}]}; (y_{[\mathtt{b}]}; z_{[\mathtt{b}]})$ *at position 1, where* $u|_1 = x_{[\mathtt{b}]}; y_{[\mathtt{b}]}$,
2. $x_{[\mathtt{b}]}; (y_{[\mathtt{b}]}; z_{[\mathtt{b}]}) = (x_{[\mathtt{b}]}; y_{[\mathtt{b}]}); z_{[\mathtt{b}]}$ *at position 2, where* $u|_2 = y_{[\mathtt{b}]}; z_{[\mathtt{b}]}$, *and*
3. $x_{[\mathtt{b}]}; \mathtt{zt} = x_{[\mathtt{b}]}$ *at position 2, where* $u|_2 = \mathtt{zt}$.

In the three cases, the subterms have sorts in the kind $[\mathtt{Bin}]$. *The only rules in R with head in that kind are* `toast1` *and* `toast2` *with head* $\mathtt{toast}(\mathtt{zt}, \mathtt{Z})$ *and* $\mathtt{toast}([\mathtt{C}, \mathtt{D}], \mathtt{Z})$ *respectively.*

As no unification modulo B is possible between any of these two heads with any of the three subterms in $\{u|_p \mid u = v \in B \cup B^{-1} \wedge p \in pos_\Sigma(u) \setminus \{\epsilon\}\}$, then $\mathcal{R}$ is closed under B-extensions.

Example 22. *Consider a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ with only one sort s, $R = \{l : f(a, b) \to c\}$, where f is associative and commutative ($E_0 = \emptyset$). Then, one possible instance of l_B is $l_B = R \cup \{l : f(x_s, f(a, b)) \to f(x_s, c)\}$, because the left side of the associative rule $f(x_s, f(y_s, z_s)) = f(f(x_s, y_s), z_s)$ has a subterm at position 2, $f(y_s, z_s)$, that matches with $f(a, b)$. If we take $R_B = l_B$ then $\mathcal{R}_B = (\Sigma, \mathcal{E}, R_B)$ is an associated rewrite theory closed under B-extensions of $\mathcal{R}$.*

By definition, associated rewrite theories closed under B-extensions are allowed to have several rules with the same label. The only condition is that all the rules sharing a label must conform a finite closure under B-extensions of one of these rules. Rewriting modulo does not change if we use a rewrite theory or any of its associated rewrite theories closed under B-extensions.

6.3 R_B, B-rewriting

In this chapter, we will use a modification of the definition of R, B-rewriting. The relation $\to_{R_B,B}$ is defined as $\to^+_{R_B,B} \cup =_\mathcal{E}$, where the relation $\to^1_{R_B,B}$ is inductively defined below.

Definition 17. *Given associated rewrite theory closed under B-extensions $\mathcal{R}_B = (\Sigma, \mathcal{E}, R_B)$, where $\mathcal{E} = E_0 \cup B$, terms t, t' in $\mathcal{H}_\Sigma$, and a rule $c : l \to r$ if $\bigwedge_{i=1}^n l_i \to r_i \mid \phi$ in R_B, if c° has the form $c^\circ : l^\circ \to r$ if $\bigwedge_{i=1}^n l_i \to r_i \mid \phi \wedge \phi^\circ$, and there exist a position p in $pos_{\Sigma_1}(t)$ and a substitution $\sigma : vars(c^\circ) \to \mathcal{T}_\Sigma$ such that $rep(t|_p) =_B l^\circ\sigma$, $t' =_\mathcal{E} t[r\sigma]_p$, $l_i\sigma \to_{R_B,B} r_i\sigma$, for $1 \le i \le n$, and $E_0 \vdash (\phi \wedge \phi^\circ)\sigma$, then there is a one-step transition $t \to^1_{R_B,B} t'$.*

We write $t \xrightarrow[c,p,\sigma]{}{}^1_{R_B,B} t'$, when we need to make explicit the rule, position, and substitution. Any of these items can be omitted when it is irrelevant.

The addition of the equality modulo $\mathcal{E}$ between t' and $t[r\sigma]_p$ will pose no problem in this chapter.

Example 23. *In example 22, as $E_0 = \emptyset$, no abstraction of terms has to be performed when rewriting with $\to^1_{R_B,B}$ ($abstract_{\Sigma_1}(l) = \langle \lambda.l; none; true\rangle$ for any left side l of a Σ-rule). Then, the term $f(f(a,a),b)$ is not a normal form in $\to^1_{R_B,B}$ because R_B has the rule $l : f(x_s, f(a,b)) \to f(x_s,c)$ that can be applied on top of the term $f(f(a,a),b)$ with matching $x_s \mapsto a$, modulo associativity and commutativity, leading to $f(f(a,a),b) \to^1_{R_B,B} f(a,c)$. Also $f(f(a,a),b) \to^1_{R/\mathcal{E}} f(a,c)$ and $f(f(a,a),b) \to^1_{R_B/\mathcal{E}} f(a,c)$, because $f(f(a,a),b) =_{\mathcal{E}} f(a,f(a,b))$. The added rule $l : f(x_s,f(a,b)) \to f(x_s,c)$ has allowed us to imitate $\to^1_{R/\mathcal{E}}$ $(=\to^1_{R_B/\mathcal{E}})$ with $\to^1_{R_B,B}$.*

Lemma 11 (Equivalence of $R/\mathcal{E}$-rewriting and $R_B/\mathcal{E}$-rewriting)**.** *If $\mathcal{R}_B = (\Sigma, \mathcal{E}, R_B)$ is an associated rewrite theory of $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ closed under B-extensions then $\to^1_{R/\mathcal{E}} = \to^1_{R_B/\mathcal{E}}$ and $\to_{R/\mathcal{E}} = \to_{R_B/\mathcal{E}}$.*

See proof on page 173.

Theorem 15 (Equivalence of $R_B/\mathcal{E}$ and R_B, B-rewriting)**.** *If $\mathcal{R}_B = (\Sigma, \mathcal{E}, R_B)$ is an associated rewrite theory closed under B-extensions then $\to^1_{R_B,B} = \to^1_{R_B/\mathcal{E}}$ and $\to_{R_B,B} = \to_{R_B/\mathcal{E}}$.*

See proof on page 174.

Corollary 4. *If $\mathcal{R}_B = (\Sigma, \mathcal{E}, R_B)$ is an associated rewrite theory of $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ closed under B-extensions then $\to^1_{R_B,B} = \to^1_{R/\mathcal{E}}$ and $\to_{R_B,B} = \to_{R/\mathcal{E}}$.*

Corollary 5. *If $\mathcal{R}_{\mathcal{B}} = (\Sigma, \mathcal{E}, R_B)$ is an associated rewrite theory closed under B-extensions, then any substitution is $R_B/\mathcal{E}$-normalized iff it is R_B, B-normalized.*

6.4 Strategies

We present in this section the full semantics for the strategies language presented in Section 2.4, together with some relevant results.

As previously stated, the semantics defines the result of the application of a strategy to the equivalence class of a term, which is based on the construction of closed proof trees. It is given by a function (in mix-fix notation)

$$\_^{@}\_ \;:\; Strat_{\mathcal{R},Call_{\mathcal{R}}} \times \mathcal{T}_{\Sigma_1/\mathcal{E}} \longrightarrow \mathcal{P}(\mathcal{T}_{\Sigma_1/\mathcal{E}}),$$

with $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ and $\mathcal{E} = E_0 \cup B$, where $[v]_{\mathcal{E}}$ is an element of $ST\,@\,[t]_{\mathcal{E}}$ if and only if a c.p.t. with head $t \to v/ST$ can be constructed using the derivation rules in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$, defined below.

If $[v]_{\mathcal{E}} \in ST\,@\,[t]_{\mathcal{E}}$, as any subtree of a c.p.t. for $t \to v/ST$, with head say $t' \to v'/ST'$, is closed then also $[v']_{\mathcal{E}} \in ST' @ [t']_{\mathcal{E}}$.

The set $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$ does not need to be computable. We will prove in this work that if a c.p.t. can be formed from an instance $G\sigma$ of a goal G (i.e., σ is a *solution* of G), then the narrowing calculus that we present can find a more general solution to the goal G, i.e., one that can be instantiated to σ.

In this work we also assume, without loss of generality, that $vars(B) \cap vars(ST) = \emptyset$ for any strategy ST in $Call_{\mathcal{R}}$, by renaming the variables in B. Now, we define $Call_{\mathcal{R}}$, $Strat_{\mathcal{R},Call_{\mathcal{R}}}$, and $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$.

We will use the following set of strategies for narrowing, which is a subset of the Maude strategy language for rewriting [MOMV04, EMOMV07, RMPV21]:

`Idle` **and** `fail`

These are constant strategies that always belong to $Strat_{\mathcal{R},Call_{\mathcal{R}}}$. While the first always succeeds, the second always fails. For each equivalence class $[t]_{\mathcal{E}} \in \mathcal{T}_{\Sigma_1/\mathcal{E}}$ there is a derivation rule $\frac{}{t \to t/\texttt{idle}}$ in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$. There are no derivation rules for `fail`. Then, $\texttt{idle} @ [t]_{\mathcal{E}} = \{[t]_{\mathcal{E}}\}$ and $\texttt{fail} @ [t]_{\mathcal{E}} = \emptyset$. We define $vars(\texttt{idle}) = vars(\texttt{fail}) = \emptyset$. For any substitution δ we define $\texttt{idle}\,\delta = \texttt{idle}$, and $\texttt{fail}\,\delta = \texttt{fail}$.

Rule application

A rule of R that has no rewrite conditions and a substitution form a *rule application*. The syntax of all the strategies in this chapter is shown using pseudo Backus–Naur form notation.

⟨AlphaNum⟩	::=	$A \mid \cdots \mid Z \mid a \mid \cdots \mid z \mid 0 \mid \cdots \mid 9$
⟨Label⟩	::=	⟨AlphaNum⟩
	|	⟨AlphaNum⟩⟨Label⟩
⟨Assignment⟩	::=	⟨Variable⟩ ↦ ⟨$\mathcal{T}_\Sigma(\mathcal{X})$-term⟩
⟨Assignment List⟩	::=	⟨Assignment⟩
	|	⟨Assignment⟩ ; ⟨Assignment List⟩
⟨Substitution⟩	::=	*none*
	|	⟨Assignment List⟩
⟨RuleApplic⟩	::=	⟨Label⟩ [⟨Substitution⟩]
⟨Strat⟩	::=	⟨RuleApplic⟩

If $c : l \to r \ \mathit{if}\ \psi$ is a rule in R, and $\gamma : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X} \setminus V_{\mathcal{R},Call_{\mathcal{R}}})$ is a substitution such that $dom(\gamma) \subseteq vars(c)$, then $c[\gamma]$ is a rule application in $Strat_{\mathcal{R},Call_{\mathcal{R}}}$. For each pair of terms t, v in $\mathcal{H}_\Sigma$, if $t \underset{c\gamma\ R}{\overset{1}{\longrightarrow}} v$ then there is a derivation rule

$$\frac{}{t \to v/c[\gamma]}$$

in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$.

We define $vars(c[\gamma]) = ran(\gamma)$. The application of $\delta : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X} \setminus V_{\mathcal{R},Call_{\mathcal{R}}})$ to $c[\gamma]$ is defined as $c[\gamma]\delta = c[(\gamma\delta)_{dom(\gamma)}]$.

Example 24. *The set $Call_{\mathcal{R}}$ for the running example contains the rule application* `kitchen`$[none]$.

For rules with rewrite conditions, a strategy must be supplied for each rewrite condition.

⟨StratList⟩	::=	⟨Strat⟩
	|	⟨Strat⟩ , ⟨StratList⟩
⟨RuleApplic⟩	::=	⟨Label⟩ [⟨Substitution⟩] { ⟨StratList⟩ }

If $c : l \to r \ \mathit{if}\ \bigwedge_{j=1}^{m} l_j \to r_j \mid \psi$ is a rule in R, $\gamma : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X} \setminus V_{\mathcal{R},Call_{\mathcal{R}}})$ is a substitution such that $dom(\gamma) \subseteq vars(c)$, and $\overline{ST} = ST_1, \ldots, ST_m$ is an ordered list of strategies such that $dom(\gamma) \cap vars(\overline{ST}) = \emptyset$, then $RA = c[\gamma]\{\overline{ST}\}$ is a rule application in $Strat_{\mathcal{R},Call_{\mathcal{R}}}$.

We define $vars(RA) = ran(\gamma) \cup vars(\overline{ST})$. The application of $\delta : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X} \setminus V_{\mathcal{R},Call_\mathcal{R}})$ to RA is defined as $RA\delta = c[(\gamma\delta)_{dom(\gamma)}]\{\overline{ST}\delta\}$. For each substitution $\delta : vars(c\gamma) \to \mathcal{T}_\Sigma$ such that $E_0 \vdash \psi\gamma\delta$, each term u in $\mathcal{H}_\Sigma$, and each position p in $pos(u)$ such that $u|_p = l\gamma\delta$ there is a derivation rule

$$\frac{l_1\gamma\delta \to r_1\gamma\delta/ST_1\delta \cdots l_m\gamma\delta \to r_m\gamma\delta/ST_m\delta}{u \to u[r\gamma\delta]_p/RA}$$

in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$.

$[t]_\mathcal{E} \in c[\gamma]@[u|_p]_\mathcal{E}$ implies $[u[t]_p]_\mathcal{E} \in c[\gamma]@[u]_\mathcal{E}$, and $[t]_\mathcal{E} \in c[\gamma]\{\overline{ST}\}@[u|_p]_\mathcal{E}$ implies $[u[t]_p]_\mathcal{E} \in c[\gamma]\{\overline{ST}\}@[u]_\mathcal{E}$ because no specific position is required for rewriting using a rule application.

Example 25. *The set $Call_\mathcal{R}$ for the running example contains an enhanced version of the rule application*

$$\texttt{cook}[none]\{(\texttt{toast1}[none] \mid \texttt{toast2}[none]), (\texttt{toast1}[none] \mid \texttt{toast2}[none])\})$$

where the symbol $\mid$ represents the `or` *strategy (defined below). Rule*

$$[cook] : \texttt{cook}(y; h_{\texttt{rt}}\ v_{\texttt{t}}, z) \to y + z; h'_{\texttt{rt}}\ v'_{\texttt{t}} \textit{ if } \texttt{toast}(h_{\texttt{rt}}, z) \to h'_{\texttt{rt}} \wedge \texttt{toast}(v_{\texttt{t}}, z) \to v'_{\texttt{t}}$$

will be applied only if we can apply either the rule application `toast1`$[none]$ *or the rule application* `toast2`$[none]$ *to each condition in the rule.*

Top

It is possible to restrict the application of a rule in R only to the top of the term. This is useful for structural rules, that are applied to the whole state, or for the strategies applied on the conditional part of a rule, as will be shown in our running example.

$\langle \text{Strat} \rangle$ ::= `top(` $\langle \text{RuleApplic} \rangle$ `)`

If $c : l \to r$ *if* ψ is a rule in R and $\gamma : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X} \setminus V_{\mathcal{R},Call_\mathcal{R}})$ is a substitution such that $dom(\gamma) \subseteq vars(c)$, then $\texttt{top}(c[\gamma])$ is a strategy in $Strat_{\mathcal{R},Call_\mathcal{R}}$. We define $vars(\texttt{top}(c[\gamma])) = vars(c[\gamma])$ and $\texttt{top}(c[\gamma])\delta = \texttt{top}(c[\gamma]\ \delta)$. For each substitution $\delta : vars(c\gamma) \to \mathcal{T}_\Sigma$ such that $E_0 \vdash \psi\gamma\delta$ there is a derivation rule

$$\frac{}{l\gamma\delta \to r\gamma\delta/\texttt{top}(c[\gamma])}$$

in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$.

If $c : l \to r$ *if* $\bigwedge_{j=1}^m l_j \to r_j \mid \psi$ is a rule in R, $\gamma : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X} \setminus V_{\mathcal{R},Call_\mathcal{R}})$ is a substitution such that $dom(\gamma) \subseteq vars(c)$, $\overline{ST} = ST_1, \ldots, ST_m$ is an ordered list of strategies such that $dom(\gamma) \cap vars(\overline{ST}) = \emptyset$ and we let $RA = c[\gamma]\{\overline{ST}\}$, then $\texttt{top}(RA)$ is a strategy in $Strat_{\mathcal{R},Call_\mathcal{R}}$. We define $vars(\texttt{top}(RA)) = vars(RA)$ and $\texttt{top}(RA)\delta = \texttt{top}(RA\ \delta)$, for $\delta : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X} \setminus V_{\mathcal{R},Call_\mathcal{R}})$. For each substitution $\delta : vars(c\gamma) \to \mathcal{T}_\Sigma$ such that $E_0 \vdash \psi\gamma\delta$, there is a derivation rule

$$\frac{l_1\gamma\delta \to r_1\gamma\delta/ST_1\delta \cdots l_m\gamma\delta \to r_m\gamma\delta/ST_m\delta}{l\gamma\delta \to r\gamma\delta/\texttt{top}(RA)}$$

in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$.

Example 26. *Whenever a rule application appears in the set $Call_\mathcal{R}$ for the running example, it is as part of a* `top` *strategy, e.g.,* $\texttt{top}(\texttt{kitchen}[none])$.

Call strategy

Call strategy definitions allow the use of parameters and the implementation of recursive strategies. A call strategy definition can be either unconditional or conditional.

⟨VarList⟩	::=	⟨Variable⟩
	|	⟨Variable⟩ , ⟨VarList⟩
⟨Equational Condition⟩	::=	⟨$\mathcal{H}_\Sigma(\mathcal{X})$-term⟩ = ⟨$\mathcal{H}_\Sigma(\mathcal{X})$-term⟩
	|	⟨Equational Condition⟩ ∧ ⟨Equational Condition⟩
⟨Strat Condition⟩	::=	⟨quantifier-free formula⟩
	|	⟨Equational Condition⟩ ∧ ⟨quantifier-free formula⟩
⟨Arguments⟩	::=	⟨$\mathcal{H}_\Sigma(\mathcal{X})$-term⟩
	|	⟨$\mathcal{H}_\Sigma(\mathcal{X})$-term⟩ , ⟨Arguments⟩
⟨Strat⟩	::=	⟨Label⟩
	|	⟨Label⟩ (⟨Arguments⟩)
⟨Call Strat⟩	::=	`sd` ⟨Label⟩ ::= ⟨Strat⟩
	|	`sd` ⟨Label⟩(⟨VarList⟩) := ⟨Strat⟩
	|	`csd` ⟨Label⟩(⟨VarList⟩) := ⟨Strat⟩ `if` ⟨Strat Condition⟩

The semantics for *call strategy invocations*, given a pair of terms t and v in $\mathcal{H}_\Sigma$ such that $ls(t) \equiv_{\leq} ls(v)$, is:

- If `sd` $CS := ST \in Call_{\mathcal{R}}$ then the call strategy invocation CS is a strategy in $Strat_{\mathcal{R},Call_{\mathcal{R}}}$. We define $vars(CS) = \emptyset$ and, for any substitution δ, $CS\delta = CS$. For every renaming γ such that $dom(\gamma) \subseteq vars(ST)$ and $ran(\gamma)$ is away from any known variable, there is a derivation rule

$$\frac{t \to v/ST\gamma}{t \to v/CS}$$

  in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$.

- If `sd` $CS(\bar{x}) := ST \in Call_{\mathcal{R}}$, where $\bar{x} = x^1_{s_1}, \ldots, x^n_{s_n}$ are the *parameters* of CS, $\hat{x} \subseteq vars(ST)$, $t_1, \ldots, t_n$ are terms in $\mathcal{T}_\Sigma(\mathcal{X} \setminus V_{\mathcal{R},Call_{\mathcal{R}}})$, with sorts $s_1, \ldots, s_n$ respectively, and we let $\bar{t} = t_1, \ldots, t_n$, then the call strategy invocation $CS(\bar{t})$ is a strategy in $Strat_{\mathcal{R},Call_{\mathcal{R}}}$. If $\rho = \{\bar{x} \mapsto \bar{t}\}$ then $vars(CS(\bar{t})) = ran(\rho)$. If $\delta : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X} \setminus \hat{x})$, then we define $CS(\bar{t})\delta = CS(\bar{t}\delta)$. For every renaming γ such that $dom(\gamma) \subseteq vars(ST) \setminus \hat{x}$ and $ran(\gamma)$ is away from any known variable, there is a derivation rule

$$\frac{t \to v/ST(\gamma \cup \rho)}{t \to v/CS(\bar{t})}$$

  in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$.

- If `csd` $CS(\bar{x}) := ST$ *if* $C \in Call_{\mathcal{R}}$, with $\bar{x} = x^1_{s_1}, \ldots, x^n_{s_n}$ and $C = \bigwedge_{j=1}^m (l_j = r_j) \wedge \phi$, $V_{CS} = vars(ST) \cup vars(C)$, $\hat{x} \subseteq V_{CS}$, $t_1, \ldots, t_n$ are terms in $\mathcal{T}_\Sigma(\mathcal{X} \setminus V_{\mathcal{R},Call_{\mathcal{R}}})$, with sorts $s_1, \ldots, s_n$ respectively, $\bar{t} = t_1, \ldots, t_n$, then the call strategy invocation $CS(\bar{t})$ is a strategy in $Strat_{\mathcal{R},Call_{\mathcal{R}}}$. If $\rho = \{\bar{x} \mapsto \bar{t}\}$ then $vars(CS(\bar{t})) = ran(\rho)$. If $\delta : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X} \setminus (ran(\rho) \cup V_{\mathcal{R},Call_{\mathcal{R}}}))$, then we define $CS(\bar{t})\delta = CS(\bar{t}\delta)$. For every renaming γ such that $dom(\gamma) \subseteq V_{CS} \setminus \hat{x}$ and $ran(\gamma)$ is away from any known variable,

and each substitution $\delta : vars(C(\gamma \cup \rho)) \to \mathcal{T}_\Sigma$ such that $l_j(\gamma \cup \rho)\delta =_\mathcal{E} r_j(\gamma \cup \rho)\delta$, for $1 \leq j \leq n$, and $E_0 \vdash \phi(\gamma \cup \rho)\delta$, there is a derivation rule

$$\frac{t \to v/ST(\gamma \cup \rho)\delta}{t \to v/CS(\bar{t})}$$

in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$.

The meaning of γ in all three cases is that the names of the variables in ST that we could call free, with respect to CS, have no relevance. By using renaming, different instances of a call strategy will get different variable names in the narrowing calculus that we have developed.

Example 27. *The call strategy definition*

$$\texttt{sd toasts} \;:=\; \texttt{top(toast1}[none]) \mid \texttt{top(toast2}[none])$$

allows us to rewrite the strategy in Example 25 as $\texttt{top(cook}[none]\{\texttt{toasts,toasts}\})$.

Tests

Tests are strategies that check a property on an equivalence class $[t]_\mathcal{E}$ in $\mathcal{T}_{\Sigma_1/\mathcal{E}}$. If the property holds then the test returns a set containing $[t]_\mathcal{E}$ as its only element. Otherwise, the test returns the empty set.

$$\begin{array}{lll} \langle \text{Test} \rangle & ::= & \texttt{match}\ \langle \mathcal{H}_\Sigma(\mathcal{X})\text{-term} \rangle\ \texttt{s.t.}\ \langle \text{Strat Condition} \rangle \\ \langle \text{Strat} \rangle & ::= & \langle \text{Test} \rangle \end{array}$$

For simplicity of notation, there will always be one quantifier-free formula $\phi \in QF(\mathcal{X}_0)$ as last element of the test condition, which will be the Boolean term *true* if there are no built-in conditions to check.

For each equivalence class $[t]_\mathcal{E}$ in $\mathcal{T}_{\Sigma_1/\mathcal{E}}$, and each strategy $TS = \texttt{match}\ u\ \texttt{s.t.}\ \bigwedge_{j=1}^m (l_j = r_j) \wedge \phi$, if there exists a substitution $\delta : vars(TS) \to \mathcal{T}_\Sigma$, where we define $vars(TS) = vars(u) \cup vars(\phi) \cup \bigcup_{j=1}^m vars((l_j, r_j))$, such that $t =_\mathcal{E} u\delta$, $l_j\delta =_\mathcal{E} r_j\delta$, for $1 \leq j \leq m$, and $E_0 \vdash \phi\delta$, then there is a rule

$$\frac{}{t \to t/\texttt{match}\ u\ \texttt{s.t.}\ \bigwedge_{j=1}^m (l_j = r_j) \wedge \phi}$$

in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$. If $\delta : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X} \backslash vars(TS))$ then $TS\delta = \texttt{match}\ u\delta\ \texttt{s.t.}\ \bigwedge_{j=1}^m (l_j\delta = r_j\delta) \wedge \phi\delta$.

Example 28. *The set $Call_\mathcal{R}$ for the running example contains the definition*
$\texttt{sd test} \;:=\; \texttt{match}\ N/B_\texttt{b}/Y; V_\texttt{t} W_\texttt{t}/OK\ \texttt{s.t.}\ Y < \texttt{ft}$.
This test will be used to verify that the system has not reached the fail time.

If-then-else

Strategies can be combined to be applied over execution paths in several ways. The first way is the if-then-else strategy where a subset of the test strategies, called *simple test*, is used. The term must match some pattern u. If the quantifier-free formula ϕ instantiated with the matching substitution holds, the strategy in the then clause is applied; if not, the strategy in the else clause is applied.

$$\langle \text{Simple Test} \rangle ::= \texttt{match}\ \langle \mathcal{H}_\Sigma(\mathcal{X})\text{-term} \rangle\ \texttt{s.t.}\ \langle \text{quantifier-free formula} \rangle$$
$$\langle \text{Strat} \rangle ::= \langle \text{Simple Test} \rangle\ ?\ \langle \text{Strat} \rangle\ :\ \langle \text{Strat} \rangle$$

For each pair of equivalence classes $[t]_\mathcal{E}$ and $[v]_\mathcal{E}$ in $\mathcal{T}_{\Sigma_1/\mathcal{E}}$, each if-then-else strategy $IS = \texttt{match}\ u\ \texttt{s.t.}\ \phi\ ?\ ST_1\ :\ ST_2$ and each substitution $\delta : vars(u) \cup vars(\phi) \to \mathcal{T}_\Sigma$ such that $t =_\mathcal{E} u\delta$, if $E_0 \vdash \phi\delta$, then there is a rule

$$\frac{t \to v/ST_1\delta}{t \to v/\texttt{match}\ u\ \texttt{s.t.}\ \phi\ ?\ ST_1 : ST_2}$$

in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$, and if $E_0 \vdash \neg\phi\delta$ then there is a rule

$$\frac{t \to v/ST_2\delta}{t \to v/\texttt{match}\ u\ \texttt{s.t.}\ \phi\ ?\ ST_1 : ST_2}$$

in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$. We define $vars(IS) = vars(u) \cup vars(\phi) \cup vars(ST_1) \cup vars(ST_2)$.

$IS\delta = \texttt{match}\ u\delta\ \texttt{s.t.}\ \phi\delta\ ?\ ST_1\delta\ :\ ST_2\delta$, for any substitution $\delta : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X} \backslash vars(IS))$.

The restriction to *SMT* constraints is needed to ensure the completeness of the narrowing calculus since, in general, a reachability condition cannot be proved false.

Example 29. *One alternative set $Call_\mathcal{R}$ for the running example contained the definition*
$\texttt{sdcheckExtract} := \texttt{match}\, N/B_\texttt{b}/Y; [\texttt{ct}, \texttt{ct}]V_\texttt{t}/OK\ \texttt{s.t.}\, true\, ?\, \texttt{top}(\texttt{dish}[none])\ :\ \texttt{idle}$
This if-then-else strategy was meant to force the extraction of a fully cooked toast to the dish, pruning the state space of the search for a solution.

Regular expressions

Another way of combining strategies is the use of regular expressions.

<Strat>	::=	<Strat> ; <Strat>	concatenation
<Strat>	::=	<Strat> | <Strat>	union
<Strat>	::=	<Strat> +	iteration (1 or more)
<Strat>	::=	<Strat> *	iteration (0 or more)

Of course, $ST*$ can be defined as $\texttt{idle} \mid ST+$. Let ST and ST' be strategies, and let t, v and u be terms in $\mathcal{H}_\Sigma$ such that $ls(t) \equiv_\leq ls(u) \equiv_\leq ls(v)$. Then, we have rules

$$\frac{t \to u/ST_1 \quad u \to v/ST_2}{t \to v/ST_1\ ;\ ST_2}$$
$$\frac{t \to v/ST_1}{t \to v/ST_1 \mid ST_2}$$
$$\frac{t \to v/ST_2}{t \to v/ST_1 \mid ST_2}$$
$$\frac{t \to v/ST}{t \to v/ST+}$$
$$\frac{t \to v/ST\ ;\ ST+}{t \to v/ST+}$$

in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$.

We define $vars(ST_1\,;ST_2) = vars(ST_1 \mid ST_2) = vars(ST_1) \cup vars(ST_2)$; we define $vars(ST+) = vars(ST)$. The concatenation and union combinators are defined to be right associative, e.g., $ST_1\,;ST_2\,;ST_3 = ST_1\,;(ST_2\,;ST_3)$. The scope of this work is restricted to concatenated and iterated strategies that have no variables in common apart form the parameters of the reachability problem being solved. Substitutions are applied to all the strategies in the regular expression.

Example 30. *The set $Call_{\mathcal{R}}$ for the running example contains the definition*
`sd kitchCook := top(kitchen[`*none*`]) ; top(cook[`*none*`]{toasts,toasts}).`
After applying the strategy `top(kitchen[`*none*`])` *to a term with sort* `Kitchen`*, the strategy* `top(cook[`*none*`]{toasts,toasts})` *will be applied to each term in the resulting set.*

Rewriting of subterms

The `matchrew` combinator allows the selection of a subterm to apply a rule and extends the scope of the substitution that validates a test strategy to subsequent steps of the execution path.

$$\begin{array}{lcl} \langle \text{VarStList} \rangle & ::= & \langle \text{Variable} \rangle \texttt{ using } \langle \text{Strat} \rangle \\ & \mid & \langle \text{VarStList} \rangle \texttt{ , } \langle \text{VarStList} \rangle \\ \langle \text{Strat} \rangle & ::= & \texttt{matchrew } \langle \mathcal{H}_\Sigma(\mathcal{X})\text{-term} \rangle \texttt{ s.t. } \langle \text{Strat Condition} \rangle \texttt{ by } \langle \text{VarStList} \rangle \end{array}$$

A `Matchrew` strategy MS has the form

$$\texttt{matchrew } u \texttt{ s.t. } \bigwedge_{j=1}^{m} (l_j = r_j) \wedge \phi \texttt{ by } x^1_{s_1} \texttt{ using } ST_1, \ldots, x^n_{s_n} \texttt{ using } ST_n$$

where $\bar{x} = x^1_{s_1}, \ldots, x^n_{s_n}$ are the *match parameters* of MS, $\hat{x}$ is a subset of the set of non-built-in variables $\mathcal{X}_1$, $|\hat{x}| = n$, $u = u[\bar{x}]_{\bar{p}}$, for appropriate $\bar{p}$, $\hat{l} \cup \hat{r} \subset \mathcal{H}_\Sigma(\mathcal{X})$, and, for $1 \leq i \leq n$, $x^i_{s_i}$ does not appear as a match parameter of another `matchrew` strategy in $\overline{ST}$ and for each $i \in \{1, \ldots, n\}$ such that $ST_i \neq \texttt{idle}$ there exists $j \in \{1, \ldots, m\}$ such that $l_j = x^i_{s_i}$ and $r_j \in \mathcal{H}_\Sigma(\mathcal{X}) \setminus \mathcal{X}$. We define $vars(MS) = V_{u,\phi,\bar{l},\bar{r},\overline{ST}}$.

We will also use the short-form $MS = \texttt{matchrew } u \texttt{ s.t. } \bar{l} = \bar{r} \wedge \phi \texttt{ by } \bar{x} \texttt{ using } \overline{ST}$. If $\delta : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X} \backslash vars(MS))$, let $\delta' = \delta_{\backslash \hat{x}}$, then $MS\delta = \texttt{matchrew } u\delta' \texttt{ s.t. } \bar{l}\delta' = \bar{r}\delta' \wedge \phi\delta' \texttt{ by } \bar{x} \texttt{ using } \overline{ST}\delta'$.

For each n-tuple $(t_1, \ldots, t_n)$ of terms in $\mathcal{T}^n_{\Sigma_1}$ such that $ls(\bar{t}) \leq \bar{s}$, and each substitution δ such that $\delta_{vars(MS)} : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X} \backslash vars(MS))$, so $\delta_{vars(MS)}$ is idempotent, $u\delta \in \mathcal{T}_\Sigma$, $\{l_j\delta, r_j\delta\}_{j=1}^m \subset \mathcal{T}_\Sigma$, $\bar{l}\delta =_{\mathcal{E}} \bar{r}\delta$, $\phi\delta \in \mathcal{T}_\Sigma$, and $E_0 \vdash \phi\delta$, so $ran(\delta_{vars(MS)}) \subseteq vars(\overline{ST}\delta)$, there is a derivation rule

$$\frac{x^1_{s_1}\delta \to t_1/ST_1\delta \cdots x^n_{s_n}\delta \to t_n/ST_n\delta}{u\delta \to u\delta[t_1, \ldots, t_n]_{p_1,\ldots,p_n}/MS}$$

in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$.

Definition 18. *For any structure Δ, we call $matchParam(\Delta)$ the set of all the match parameters that appear in Δ.*

In narrowing, rewrite rules have to be applied, using unification, to non-variable terms. Requiring each variable $x^i_{s_i}$ to match with a non-variable term of $\mathcal{H}_\Sigma(\mathcal{X})$ will allow the narrowing calculus to apply the rewrite rules to the non-variable instances of $x^i_{s_i}$.

Example 31. *The set $Call_{\mathcal{R}}$ for the running example contains the definition*

`sd cook1` := `matchrew` $N/B_{\mathtt{b}}/K_{\mathtt{k}}/OK$ `s.t.` $K_{\mathtt{k}} = Y; R_{\mathtt{rt}} V_{\mathtt{t}}$ `by` $K_{\mathtt{k}}$ `using kitchCook`.

The strategy `kitchCook` *will be applied to the* ***`Kitchen`*** $K_{\mathtt{k}}$ *of a* ***`State`****, whenever there is a* ***`RealToast`*** *($R_{\mathtt{rt}}$) in $K_{\mathtt{k}}$, and $K_{\mathtt{k}}$ will get instantiated to a non-variable term by the condition.*

Definition 19 (Subterms, holes, and replacement in a strategy)**.** *We extend the use of subterms and holes to strategies. If ST is a strategy, i is a positive integer, p is a position, and t is a term, then $ST|_{i.p}$ is the subterm that appears at position p in the term i of the tuple formed by all terms that appear in ST, taken from left to right, $ST[]_{i.p}$ consists in the replacement in $ST|_i$ of its subterm at position p with $[]$, and $ST[t]_{i.p}$ consists in the replacement in $ST|_i$ of its subterm at position p with t.*

Definition 20 (Equality modulo of strategies)**.** *Given two strategies ST and ST', we say that ST is equal modulo $\mathcal{E}$ to ST', and write $ST =_{\mathcal{E}} ST'$ iff $ST = ST'[\bar{t}]_{\bar{p}}$, for appropriate $\bar{t}$ and $\bar{p}$, and for each position p in $\bar{p}$ $ST|_p =_{\mathcal{E}} ST'|_p$ and $V_{ST|_p} = V_{ST'|_p}$.*

6.4.1 Interpretation of the semantics. Generalization of strategies

Lemma 12 (Interpretation of the semantics)**.** *Given a rewrite theory $\mathcal{R} = (\Sigma, E_0 \cup B, R)$, a set of call strategy definitions $Call_{\mathcal{R}}$, and terms $t, v \in \mathcal{H}_\Sigma$, for each c.p.t. T formed using the rules in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$ with head $t \to v/ST$, so $[v]_{\mathcal{E}} \in ST@[t]_{\mathcal{E}}$, each renaming α such that $ran(\alpha) \cap (V_T \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$, and each strategy $ST' =_{\mathcal{E}} ST$, it holds that:*

1. *Main property: $t \to_{R/\mathcal{E}} v$ and there exist closed proof trees for $[v]_{\mathcal{E}} \in ST\alpha@[t]_{\mathcal{E}}$ and $[v]_{\mathcal{E}} \in ST'@[t]_{\mathcal{E}}$ with the same depth and number of nodes as T.*
2. *If $ST =$ `idle` then $[t]_{\mathcal{E}} = [v]_{\mathcal{E}}$.*
3. *If $ST = c[\gamma]$ then $t \underset{c\gamma}{\longrightarrow}^1_{R/\mathcal{E}} v$.*
4. *If $ST =$ `top`$(c[\gamma])$, then $t \underset{c\gamma,\epsilon}{\longrightarrow}^1_{R/\mathcal{E}} v$ (i.e., the rewrite happens at the top position of t).*
5. *If $ST =$ `match` u `s.t.` $\bigwedge_{j=1}^m (l_j = r_j) \wedge \phi$ then $[t]_{\mathcal{E}} = [v]_{\mathcal{E}}$ and there exists a substitution σ such that $t =_{\mathcal{E}} u\sigma$, $l_j\sigma =_{\mathcal{E}} r_j\sigma$, for $1 \leq j \leq m$, and $E_0 \vdash \phi\sigma$.*
6. *If $ST = ST_1 \,;\, ST_2$ then there exists a term $u \in \mathcal{H}_\Sigma$ such that $[u]_{\mathcal{E}} \in ST_1@[t]_{\mathcal{E}}$ and $[v]_{\mathcal{E}} \in ST_2@[u]_{\mathcal{E}}$.*
7. *If $ST = ST_1+$ then there exist $i+1$ terms $u_0 = t, u_1, \ldots, u_{i-1}, u_i = v \in \mathcal{H}_\Sigma$, with $i > 0$, such that $[u_j]_{\mathcal{E}} \in ST_1@[u_{j-1}]_{\mathcal{E}}$, for $1 \leq j \leq i$, where i is equal to one plus the number of times that a rule with the form $\frac{w_1 \to w_2/ST_1 \,;\, ST_1+}{w_1 \to w_2/ST_1+}$, followed by the application of a rule with the form $\frac{\overline{w_1 \to w'/ST_1} \quad \overline{w' \to w_2/ST_1+}}{w_1 \to w_2/ST_1 \,;\, ST_1+}$, is applied in the rightmost branch of the subtree before applying a rule with the form $\frac{w_1 \to w_2/ST_1}{w_1 \to w_2/ST_1+}$.*
8. *If $ST = ST_1 \mid ST_2$ then $[v]_{\mathcal{E}} \in ST_1@[t]_{\mathcal{E}}$ or $[v]_{\mathcal{E}} \in ST_2@[t]_{\mathcal{E}}$.*

9. *If* $ST = \texttt{match}\ u\ \texttt{s.t.}\ \phi\,?\,ST_1 : ST_2$ *then there exists a substitution* δ *such that* $t =_{\mathcal{E}} u\delta$ *and either* $E_0 \vdash \phi\delta$ *and* $[v]_{\mathcal{E}} \in ST_1\delta@[t]_{\mathcal{E}}$ *or* $E_0 \vdash \neg\phi\delta$ *and* $[v]_{\mathcal{E}} \in ST_2\delta@[t]_{\mathcal{E}}$.

10. *If* $ST = CS$, *where* $\texttt{sd}\ CS := ST_1 \in Call_{\mathcal{R}}$, *then: (i)* $[v]_{\mathcal{E}} \in ST_1@[t]_{\mathcal{E}}$, *and (ii)* $[v]_{\mathcal{E}} \in ST_1\gamma@[t]_{\mathcal{E}}$, *for every renaming* γ *such that* $dom(\gamma) \subseteq vars(ST_1) \setminus V_{\mathcal{R}}$ *and* $ran(\gamma) \cap V_{\mathcal{R},Call_{\mathcal{R}}} = \emptyset$.

11. *If* $ST = CS(\bar{t})$, *where* $\texttt{sd}\ CS(\bar{x}) := ST_1 \in Call_{\mathcal{R}}$, $\bar{x} = x^1_{s_1}, \ldots, x^n_{s_n}$, $\bar{t} = t_1, \ldots, t_n$, *and* $\rho = \{\bar{x} \mapsto \bar{t}\}$, *then: (i)* $[v]_{\mathcal{E}} \in ST_1\rho@[t]_{\mathcal{E}}$ *and (ii) if* γ *is a renaming such that* $dom(\gamma) \subseteq vars(ST_1) \setminus \hat{x}$ *and* $ran(\gamma) \cap (ran(\rho) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$ *(so* $\frac{t \to v/ST_1(\gamma \cup \rho)}{t \to v/CS(\bar{t})} \in \mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$*), then* $[v]_{\mathcal{E}} \in ST_1(\gamma \cup \rho)@[t]_{\mathcal{E}}$.

12. *If* $ST = CS(\bar{t})$, *where* $\texttt{csd}\ CS(\bar{x}) := ST_1$ *if* $C \in Call_{\mathcal{R}}$, *with* $\bar{x} = x^1_{s_1}, \ldots, x^n_{s_n}$ *and* $C = \bigwedge_{j=1}^m (l_j = r_j) \wedge \phi$, $V_{CS} = vars(ST_1) \cup vars(C)$, $\hat{x} \subseteq V_{CS}$, $\bar{t} = t_1, \ldots, t_n$, *and* $\rho = \{\bar{x} \mapsto \bar{t}\}$, *then (i) there exists a substitution* $\delta_1 : vars(C\rho) \to \mathcal{T}_\Sigma$, *such that* $l_j\rho\delta_1 =_{\mathcal{E}} r_j\rho\delta_1$, *for* $1 \leq j \leq n$, $E_0 \vdash \phi\rho\delta_1$ *(so* $\frac{t \to v/ST_1\rho\delta_1}{t \to v/CS(\bar{t})} \in \mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$*), and* $[v]_{\mathcal{E}} \in ST_1\rho\delta_1@[t]_{\mathcal{E}}$, *and (ii) for every renaming* γ *such that* $dom(\gamma) \subseteq V_{CS} \setminus \hat{x}$ *and* $ran(\gamma) \cap (ran(\rho) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$, *there exists a substitution* $\delta_2 : vars(C(\gamma \cup \rho)) \to \mathcal{T}_\Sigma$, *such that* $l_j(\gamma \cup \rho)\delta_2 =_{\mathcal{E}} r_j(\gamma \cup \rho)\delta_2$, *for* $1 \leq j \leq n$, $E_0 \vdash \phi(\gamma \cup \rho)\delta_2$ *(so* $\frac{t \to v/ST_1(\gamma \cup \rho)\delta_2}{t \to v/CS(\bar{t})} \in \mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$*), and* $[v]_{\mathcal{E}} \in ST_1(\gamma \cup \rho)\delta_2@[t]_{\mathcal{E}}$.

13. *If* $ST = c[\gamma]\{ST_1, \ldots, ST_m\}$, *with* $c : l \to r$ *if* $\bigwedge_{j=1}^m l_j \to r_j \mid \psi$ *a rule in* R, *then there is a substitution* δ *such that* $[r_i\gamma\delta]_{\mathcal{E}} \in ST_i\delta\ @\ [l_i\gamma\delta]_{\mathcal{E}}$, *for* $1 \leq i \leq m$, *and* $t \xrightarrow[c,\gamma\delta]{1}_{R/\mathcal{E}} v$.

14. *If* $ST = \texttt{top}(c[\gamma]\{ST_1, \ldots, ST_m\})$, *with* $c : l \to r$ *if* $\bigwedge_{j=1}^m l_j \to r_j \mid \psi$ *a rule in* R *then there is a substitution* δ *such that* $[r_i\gamma\delta]_{\mathcal{E}} \in ST_i\delta\ @\ [l_i\gamma\delta]_{\mathcal{E}}$, *for* $1 \leq i \leq m$, *and* $t \xrightarrow[c,\epsilon,\gamma\delta]{1}_{R/\mathcal{E}} v$.

15. *If* $ST = \texttt{matchrew}\ u\ \texttt{s.t.}\ \bigwedge_{j=1}^m (l_j = r_j) \wedge \phi\ \texttt{by}\ x^1_{s_1}\ \texttt{using}\ ST_1, \ldots, x^n_{s_n}\ \texttt{using}\ ST_n$, *where* $u = u[x^1_{s_1}, \ldots, x^n_{s_n}]_{p_1 \ldots p_n}$ *then there exist a substitution* δ, *where* $\delta_{V_{u,\phi,\bar{l},\bar{r}}}$ *is ground, and terms* $t_1, \ldots, t_n \in \mathcal{H}_\Sigma$ *such that* $t =_{\mathcal{E}} u\delta$, $l_j\delta =_{\mathcal{E}} r_j\delta$, *for* $1 \leq j \leq m$, $E_0 \vdash \phi\delta$, $[t_i]_{\mathcal{E}} \in ST_i\delta\ @\ [x^i_{s_i}\delta]_{\mathcal{E}}$, *for* $1 \leq i \leq n$, *and* $v =_{\mathcal{E}} u\delta[t_1, \ldots, t_n]_{p_1 \ldots p_n}$.

See proof on page 178.

Lemma 13 (Generalization of strategies)**.** *Given a rewrite theory* $\mathcal{R} = (\Sigma, E_0 \cup B, R)$, *a set of call strategy definitions* $Call_{\mathcal{R}}$, *terms* $t, v \in \mathcal{H}_\Sigma$, *a strategy* $ST \in Strat_{\mathcal{R},Call_{\mathcal{R}}}$, *and a substitution* σ *such that* $dom(\sigma) \cap V_R = \emptyset$ *and* $ran(\sigma) \cap (V_R \cup V_{ST}) = \emptyset$, *if* $[v]_{\mathcal{E}} \in ST\sigma@[t]_{\mathcal{E}}$ *can be proved with a c.p.t.* T *then* $[v]_{\mathcal{E}} \in ST@[t]_{\mathcal{E}}$ *and a c.p.t.* T' *with head* $t \to v/ST$ *and the same depth as* T *can be constructed.*

See proof on page 185.

6.4.2 Call strategies in the toast cooking example

The Maude strategy module that we will use in the running example problems in Section 6.7, where these call strategies are explained, is:

```
smod TOASTS-STRAT is
  protecting TOASTS .

  vars NS OKS YS failTime : Integer .
  var RS : RealToast .
  vars VS WS : Toast .
  var BS : Bin .
  var KS : Kitchen .

  strat test : @ System .
  strat cook1 : @ System .
  strat kitchCook : @ Kitchen .
  strat toasts : @ Toast .
  strat noCook : @ System .
  strat loop : @ System .
  strat solve1 : @ System .
  strat solve2 : @ System .

  sd test := match NS / BS / YS ; VS WS / OKS s.t.
             (YS < failTime) = (true).Boolean [nonexec] .
  sd cook1 := matchrew NS / BS / KS / OKS s.t.
              YS ; RS VS := KS by KS using kitchCook .
  sd kitchCook := top(kitchen) ; top(cook{toasts, toasts})
  sd toasts := top(toast1) | top(toast2) .
  sd noCook := top(bin) | top(pan) | top(dish) .
  sd loop := (noCook | (cook1 ; test ; noCook)) + .
  sd solve1 := top(bag) ; top(bag) ; top(bag) ; loop .
  sd solve2 := top(bag) ; top(bag) ; (top(bag) | idle) ; loop .
endsm
```

All the variables are named with an 'S' (for strategy) at their end, except `failtime` which is a parameter, to avoid collisions with other variable names either in the rules or in the reachability problem. This strategy module in Maude corresponds to the following set of strategy definitions using our syntax:

- sd test := match $NS/BS_{\mathtt{b}}/YS$; $VS_{\mathtt{t}}\,WS_{\mathtt{t}}/OKS$ s.t. $YS <$ ft
- sd cook1 := matchrew $NS/BS_{\mathtt{b}}/KS_{\mathtt{k}}/OKS$ s.t. YS; $RS_{\mathtt{rt}}\,VS_{\mathtt{t}} := KS_{\mathtt{k}}$ by $KS_{\mathtt{k}}$ using kitchCook
- sd kitchCook := top(kitchen[*none*]) ; top(cook[*none*]{toasts, toasts})
- sd toasts := top(toast1[*none*]) | top(toast2[*none*])
- sd noCook := top(bin[*none*]) | top(pan[*none*]) | top(dish[*none*])
- sd loop := (noCook | (cook1 ; test ; noCook))+
- sd solve1 := top(bag[*none*]) ; top(bag[*none*]) ; top(bag[*none*]) ; loop
- sd solve2 := top(bag[*none*]) ; top(bag[*none*]) ; (top(bag[*none*]) | idle) ; loop

There are two main differences between both syntaxes:

1. Maude can decide whether an alphanumeric label corresponds to a call strategy invocation or a rule application and adds the *none* substitution when parsing the strategy module, if it is missing.

   In our syntax for strategies, we require that all rule applications are invoked with a substitution, for simplicity of the prototype.

2. The SMT constraint must always appear in the Maude strategy module, again for simplicity of the prototype. If there is no SMT constraint, then we use the SMT constant `(true).Boolean`.

   As the Maude syntax checker only recognizes conditions with sort `Bool` and any SMT constraint has sort `Boolean`, we check for equality against `(true).Boolean` to obtain a term with sort `Bool`.

   Internally, our narrowing prototype will only check the satisfiability of the SMT constraint, through the metalevel `metaCheck` function, ignoring this equality.

One possible improvement would be to redefine the call strategy `noCook`, using an if-then-else strategy, as:

- $\texttt{sd noCook} := \texttt{match}\ NS/BS_{\texttt{b}}/YS;\ [\texttt{ct},\texttt{ct}]\ VS_{\texttt{t}}/OKS\ ?$
  $\texttt{top}(\texttt{dish}[none])\ :\ \texttt{top}(\texttt{bin}[none])\ |\ \texttt{top}(\texttt{pan}[none])$

or, in Maude syntax:

```
sd noCook := match NS / BS / YS ; [cookTime, cookTime] VS / OKS
             s.t. (true).Boolean = (true).Boolean ?
             top(dish) : (top(bin) | top(pan)) .
```

This would force the extraction of any `RealToast` after it gets well-cooked, reducing the state space.

The performance of both definitions is compared in Section 6.7.2.

6.5 Strategies in reachability problems

In this section we present the concept of reachability problem for this chapter, which is completely based on the use of strategies, together with its solutions and the properties that a solution to one of these problems has.

6.5.1 Reachability problems

Definition 21 (Reachability problem)**.** *Given a rewrite theory* $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ *and a set of call strategy definitions* $Call_{\mathcal{R}}$*, a* reachability problem *is an expression* P *with the form* $\bigwedge_{i=1}^{n} u_i \rightarrow v_i/ST_i \mid \phi \mid V, \nu$*, where* u_i *and* v_i *are terms in* $\mathcal{H}_\Sigma(\mathcal{X})$*,* ST_i *is a strategy in* $Strat_{\mathcal{R},Call_{\mathcal{R}}}$*,* $\phi \in QF(\mathcal{X}_0)$*,* V *is the finite set of* parameters *of the problem, i.e., variables of* $\mathcal{X}$ *that have to be given a ground value, and* ν *is a substitution such that* $dom(\nu) \subseteq V$ *and* $ran(\nu)$ *consists only of new variables, not seen before, that may hold the initial values, either constants or patterns, of some of these parameters. The formula* ϕ *is the* reachability formula *of* P*. We define* $vars(P) = vars(\bar{u}, \bar{v}, \phi)$*. The set* V *allows the declaration of variables in* $V_{\mathcal{R},Call_{\mathcal{R}}}$ *or* $V_{\overline{ST}}$*, as parameters of the problem.* V *must always verify:*

1. *$vars(P) \subseteq V$, $vars(B) \cap V = \emptyset$, and $V_{\mathcal{R}} \cap V_{Call_{\mathcal{R}}} \subseteq V$, i.e, $V_{\mathcal{R}}$ and $V_{Call_{\mathcal{R}}}$ have no variables in common, with the exception of the parameters of the problem,*

2. *concatenated strategies may have in common only variables from V, since they will be given a ground value; this is also mandatory for strategies from different open goals; also, only variables from V may appear in iterated strategies and call strategy invocations, since they may become concatenated ones, and*

3. *V cannot contain:*

   - *any variable in $dom(\gamma)$ for any strategy $c[\gamma]$ that may appear in $Call_{\mathcal{R}}$ or ST_i, $1 \leq i \leq n$,*
   - *any variable in $matchParam(\overline{ST}) \cup matchParam(Call_{\mathcal{R}})$ (see Def. 18).*

6.5.2 Instances and solutions

Definition 22 (Instances). *Given a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, a set of call strategy declarations $Call_{\mathcal{R}}$, and a substitution σ such that $vars(B) \cap (dom(\sigma) \cup ran(\sigma)) = \emptyset$, the* instance $\mathcal{R}^\sigma$ *of $\mathcal{R}$ is the rewrite theory that results from the simultaneous replacement of every instance in R of any variable $x \in dom(\sigma)$ with $x\sigma$, $Call^\sigma_{\mathcal{R}}$ is the set of call strategy declarations that results from the simultaneous replacement of every instance in $Call_{\mathcal{R}}$ of any variable $x \in dom(\sigma)$ with $x\sigma$, and $Strat^\sigma_{\mathcal{R},Call_{\mathcal{R}}}$ is their set of associated strategies. For every strategy ST in $Strat_{\mathcal{R},Call_{\mathcal{R}}}$ we denote by ST^σ its corresponding strategy in $Strat^\sigma_{\mathcal{R},Call_{\mathcal{R}}}$. We denote by $\mathcal{D}^\sigma_{\mathcal{R},Call_{\mathcal{R}}}$ the associated set of derivation rules. If γ is a substitution, $dom(\gamma) \cap (dom(\sigma) \cup ran(\sigma)) = \emptyset$, and $ST = ST_1\gamma$ then $ST^\sigma = ST_1^\sigma(\gamma \cdot \sigma)$. If $t \in \mathcal{T}_\Sigma(\mathcal{X})$, then $t^\sigma = t\sigma$. If $\phi \in QF(\mathcal{X}_0)$, then $\phi^\sigma = \phi\sigma$. For any structure S formed with terms, formulas and strategies, the instance S^σ of S will consist of the instantiation with σ of each one of its elements.*

Although the label, say c, of an instantiated rule remains the same, we will use superscripts, say c^σ, when we need to distinguish which instance of the rule we are referring to.

Proposition 16 (Equality of $(R^\sigma)_B$ and $(R_B)^\sigma$). *For any rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ and any substitution σ such that $vars(B) \cap (dom(\sigma) \cup ran(\sigma)) = \emptyset$, it holds that $(R^\sigma)_B = (R_B)^\sigma$.*

See proof on page 188.

We will write R^σ_B to refer to either $(R^\sigma)_B$ or $(R_B)^\sigma$, indistinctly.

Definition 23 (Solution of a reachability problem). *Given a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, and a set of call strategy definitions $Call_{\mathcal{R}}$, a solution of the reachability problem $P = \bigwedge_{i=1}^n u_i \to v_i/ST_i \mid \phi \mid V, \nu$ is a substitution $\sigma : V \to \mathcal{T}_\Sigma$ such that $\sigma = \nu \cdot \sigma'$ for some substitution σ', $E_0 \vdash \phi\sigma$, and $[v_i\sigma]_{\mathcal{E}} \in ST_i^\sigma @ [u_i\sigma]_{\mathcal{E}}$ (hence $u_i\sigma \to_{R^\sigma/\mathcal{E}} v_i\sigma$), for $1 \leq i \leq n$.*

The reachability problem $P' = \bigwedge_{i=1}^n u_i \to v_i/ST_i$; `idle` $\mid \phi \mid V, \nu$, has the same solutions as P: for any solution σ of P, $E_0 \vdash \phi\sigma$ and $[v_i\sigma]_{\mathcal{E}}$ in $ST_i^\sigma @ [u_i\sigma]_{\mathcal{E}}$, for $1 \leq i \leq n$, so there are closed proof trees

$$\frac{F_i}{u_i\sigma \to v_i\sigma/ST_i^\sigma},$$

where $1 \leq i \leq n$, formed with the rules in $\mathcal{D}^{\sigma}_{\mathcal{R},Call_{\mathcal{R}}}$. Then, also

$$\frac{\dfrac{F_i}{u_i\sigma \to v_i\sigma/ST_i^{\sigma}} \quad \dfrac{}{v_i\sigma \to v_i\sigma/\mathtt{idle}}}{u_i\sigma \to v_i\sigma/ST_i^{\sigma}\,;\mathtt{idle}},$$

where $1 \leq i \leq n$, are closed proof trees, so σ is a solution of P', and vice versa.

Given a reachability problem $\bigwedge_{i=1}^{n} u_i \to v_i/ST_i \mid \phi \mid V, \nu$, we will solve the equivalent problem $\bigwedge_{i=1}^{n} u_i \to v_i/ST_i\,;\mathtt{idle} \mid \phi \mid V, \nu$, since it will allow us to use a smaller set of narrowing rules, by not having to distinguish between those strategies that are a concatenation of strategies, to process one strategy after the other, and those that are not, except for the `idle` strategy, that will require two rules.

6.6 Strategies in reachability by conditional narrowing modulo SMT plus axioms

In this section, the narrowing calculus for reachability with strategies is introduced and its soundness, weak completeness, and completeness for topmost rewrite theories are stated.

6.6.1 Reachability goals and calculus

Some definitions and the calculus for reachability with strategies by conditional narrowing modulo SMT plus axioms are presented now.

Instance of a set of variables

Given a set of variables V and a substitution ν, we define the *instance* V^{ν} of V as the set $V^{\nu} = (V \setminus dom(\nu)) \cup ran(\nu_V)$.

Reachability goal

Given a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ and a set of call strategy definitions $Call_{\mathcal{R}}$, a *reachability goal* G is an expression with the form

1. $(\bigwedge_{i=1}^{n} u_i' \to v_i'/ST_i \mid \phi')^{\nu}\varrho_{\nu} \mid V, \nu$, or
2. $(u_1'|_p \to^1 x_k, u_1'[x_k]_p \to v_1'/ST_1 \wedge \bigwedge_{i=2}^{n} u_i' \to v_i'/ST_i \mid \phi')^{\nu}\varrho_{\nu} \mid V, \nu$,

where ν and ϱ_{ν} are substitutions, $dom(\nu) \subseteq V$, $dom(\varrho_{\nu}) \cap (V \cup V^{\nu}) = \emptyset$, $V \subset \mathcal{X}$ is finite, $(\bar{u}, \bar{v}, \phi) = (\bar{u}', \bar{v}', \phi')^{\nu}\varrho_{\nu}$, $n \geq 1$, u_i' and v_i' are terms in $\mathcal{H}_{\Sigma}(\mathcal{X})$, $ST_i \in Strat_{\mathcal{R},Call_{\mathcal{R}}}$, for $1 \leq i \leq n$, and $\phi \in QF(\mathcal{X}_0)$; also, in the second case, $p \in pos(u_1)$, $k = [ls(u_1|_p)]$, the kind of the least sort of $u_1|_p$, $x_k \notin V_{\bar{u}',\bar{v}',\phi',\overline{ST}} \cup V \cup ran(\nu) \cup dom(\varrho_{\nu}) \cup ran(\varrho_{\nu})$, and ST_1 has the form $RA\,;ST$, with RA a rule application.

In the first case, each one of the elements in the conjunction is an open goal, for which we define $V_{u \to v/ST} = V_{u,v}$, and $V_G = V_{\bar{u},\bar{v},\phi} \cup V^{\nu}$; in the second case, we say that x_k is the *connecting variable* of the goal and we define $V_G = \{x_k\} \cup V_{\bar{u},\bar{v},\phi} \cup V^{\nu}$. We will write 'goal' as a synonym of reachability goal from now on.

Reachability goals with the second form, where we always can recover u_1 from $u_1|_p$ and $u_1[x_k]_p$ since $u_1 = u_1[u_1|_p]_p$, can be generated by the calculus rules in Figures 6.2, 6.3, and 6.4 from a reachability goal with the first form when the first open goal has the

form $u_1 \to v_1/RA; ST$, with RA a rule application strategy. This second form prevents the repeated application in a derivation of rule transitivity, that maintains the problem in the second form, forcing the application to the first open goal of the rule application rule, that reverts the problem to the first form.

The substitution ϱ_ν will be used in our calculus to hold instantiations or renamings, that will be generated by the calculus rules, of the variables not in V.

Definition 24 (Instance of a goal)**.** *If G is a goal of the form $(\bigwedge_{i=1}^n S_i \mid \phi)^\nu \varrho_\nu \mid V, \nu$ and σ is a substitution such that $dom(\sigma) \cap V^\nu \neq \emptyset$, then we define the* instance *$G\sigma$ of G as $G\sigma = (\bigwedge_{i=1}^n S_i \mid \phi)^\mu \varrho_\mu \mid V, \mu$, where $\mu = (\nu\sigma)_V$ and $\varrho_\mu = (\varrho_\nu\sigma)_{V_G \setminus V}$.*

Definition 25 (Instance of a conjunction of open goals)**.** *If G is a goal of the form $(\bigwedge_{i=1}^n S_i \mid \phi)^\nu \varrho_\nu \mid V, \nu$ and σ is a substitution such that $dom(\sigma) \cap V^\nu \neq \emptyset$, let $SG = (\bigwedge_{i=1}^n S_i)^\nu \varrho_\nu$ and define the* instance *$SG\sigma$ of SG as $SG\sigma = (\bigwedge_{i=1}^n S_i)^\mu \varrho_\mu$, where $\mu = (\nu\sigma)_V$ and $\varrho_\mu = (\varrho_\nu\sigma)_{V_{SG} \setminus V}$.*

When $dom(\sigma) \cap V^\nu = \emptyset$, σ is directly applied to every term and formula in G and SG, respectively, thus avoiding circularity in these definitions.

Admissible goals

From now on, we will only consider in this chapter two types of goals:

(a) those goals coming from a reachability problem $\bigwedge_{i=1}^n u_i \to v_i/ST_i \mid \phi \mid V, \nu$, which is transformed into the goal $\bigwedge_{i=1}^n u_i\nu \to v_i\nu/ST_i^\nu; \texttt{idle} \mid \phi\nu \mid V, \nu$, with $\varrho_\nu = none$, and

(b) those goals generated by repeatedly applying the calculus rules in Figures 6.2 to 6.4 to one goal of type (a).

The notation $G \rightsquigarrow_{[r],\sigma} G'$, will be used in the calculus to indicate that rule $[r]$ of the calculus has been applied with substitution σ to G, yielding G'. We call this application a *narrowing step*. If σ is the identity substitution it can be omitted. The rule $[r]$ can also be omitted in the expression. The superscripts $\rightsquigarrow^n$, with $n > 0$, $\rightsquigarrow^+$, and $\rightsquigarrow^*$ will be used with their standard meanings, maybe with no rule in the subscript ($\rightsquigarrow$ and $\rightsquigarrow^1$ are equivalent).

Definition 26 (Solution of a goal)**.** *Given a rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, a set of call strategy definitions $Call_\mathcal{R}$ for $\mathcal{R}$, and a goal G, a substitution $\sigma : vars(G) \to \mathcal{T}_\Sigma$, where $\nu' = (\nu\sigma)_V$ and $\varrho_{\nu'} = (\varrho_\nu\sigma)_{\setminus V}$, is a* solution of G *iff:*

1. *if $G = \bigwedge_{i=1}^n u_i \to v_i/ST_i^\nu \varrho_\nu \mid \phi \mid V, \nu$ then $E_0 \vdash \phi\sigma$ and $[v_i\sigma]_\mathcal{E} \in ST_i^{\nu'} \varrho_{\nu'}@[u_i\sigma]_\mathcal{E}$ (hence $u_i\sigma \to_{R^{\nu'}/\mathcal{E}} v_i\sigma$), for $1 \leq i \leq n$, and*

2. *if $G = u_1|_p \to^1 x_k, u_1[x_k]_p \to v_1/ST_1^\nu\varrho_\nu \wedge \bigwedge_{i=2}^n u_i \to v_i/ST_i^\nu\varrho_\nu \mid \phi \mid V, \nu$, where $ST_1 = RA; ST$, then $E_0 \vdash \phi\sigma$, $[x_k\sigma]_\mathcal{E} \in RA^{\nu'} \varrho_{\nu'}@[u_1\sigma|_p]_\mathcal{E}$, $[v_1\sigma]_\mathcal{E} \in ST^{\nu'} \varrho_{\nu'}@[u_1[x_k]_p\sigma]_\mathcal{E}$, and $[v_i\sigma]_\mathcal{E} \in ST_i^{\nu'} \varrho_{\nu'}@[u_i\sigma]_\mathcal{E}$, for $2 \leq i \leq n$.*

In the second case, as $[x_k\sigma]_\mathcal{E} \in RA^{\nu'} \varrho_{\nu'}@[u_1|_p\sigma]_\mathcal{E}$ implies $[u_1[x_k]_p\sigma]_\mathcal{E} \in RA^{\nu'} \varrho_{\nu'}@[u_1\sigma]_\mathcal{E}$, and $[v_1\sigma]_\mathcal{E} \in ST^{\nu'} \varrho_{\nu'}@[u_1[x_k]_p\sigma]_\mathcal{E}$ then $[v_1\sigma_{\setminus\{x_k\}}]_\mathcal{E} \in ST_1^{\nu'}(\varrho_{\nu'})_{\setminus\{x_k\}}@[u_1\sigma_{\setminus\{x_k\}}]_\mathcal{E}$, i.e., $\sigma_{\setminus\{x_k\}}$ is a solution of $\bigwedge_{i=1}^n u_i \to v_i/ST_i^\nu \varrho_\nu \mid \phi \mid V, \nu$.

We call $nil \mid \phi \mid V, \nu$, where ϕ is satisfiable and $\nu : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X})$ such that $dom(\nu) \subseteq V$, an *empty goal*.

Given $\mathcal{R}_B = (\Sigma, \mathcal{E}, R_B)$, a closed under B-extensions associated rewrite theory of $\mathcal{R} = (\Sigma, \mathcal{E}, R)$, both with built-in subtheory (Σ_0, E_0), a reachability problem $P = \bigwedge_{i=1}^n u_i \to v_i/ST_i \mid \phi \mid V, \nu$ is solved by applying the calculus rules in Figures 6.2 - 6.4, starting with $G = \bigwedge_{i=1}^n u_i\nu \to v_i\nu/(ST_i^\nu \,;\, \mathtt{idle}) \mid \phi\nu \mid V, \nu$ in a top-down manner, until an empty goal is obtained, where $(\bigwedge_{i=1}^n u_i \to v_i/ST_i^\nu \varrho_\nu)\sigma = \bigwedge_{i=1}^n u_i\sigma \to v_i\sigma/ST_i^{(\nu\sigma)_V}(\varrho_\nu\sigma)_{\setminus V}$.

We briefly explain rule $[w]$ (matchrew): we rename the matching parameters from $\bar{z}$ to the fresh variables $\bar{x}$ with γ. Once abstracted u and $t[\bar{x}]_{\bar{p}}$ to u° and t° and chosen a B-unifier σ of u° and t°, we abstract $\bar{l}\gamma\sigma$ and $\bar{r}\gamma\sigma$, and choose a B-unifier of these abstractions, say α, using the $[d1]$ (idle) strategy. Then, the open goals $(\bar{x}\sigma \to \bar{y}/\overline{ST}\gamma\sigma)\alpha$, where $\bar{y}$ is fresh, will try to find a substitution β that makes $[y_i\beta]_\mathcal{E}$ an element of $ST_i\gamma\sigma\alpha\beta@[x_i\sigma\alpha\beta]_\mathcal{E}$, for $1 \leq i \leq n$. If successful, we continue trying to find solutions for the open goal $(t[\bar{y}]_{\bar{p}} \to v/ST)\sigma\alpha\beta$.

Narrowing path and computed answer

Given $\mathcal{R}_B = (\Sigma, \mathcal{E}, R_B)$, an associated rewrite theory of $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ closed under B-extensions and a goal G with set of parameters V and substitution ν_0, if there is a *narrowing path* $G \rightsquigarrow_{\sigma_1} G_1 \rightsquigarrow_{\sigma_2} \cdots \rightsquigarrow_{\sigma_{n-1}} G_{n-1} \rightsquigarrow_{\sigma_n} nil \mid \psi \mid V, \nu$, using the calculus rules in Figures 6.2 and 6.3, hence ψ is satisfiable, then we write $G \rightsquigarrow_\sigma^n nil \mid \psi \mid V, \nu$, where $\sigma = \sigma_1 \cdots \sigma_n$, and we let $\nu \mid \psi$ a *computed answer* for G.

If $\nu_0 = none$ then ν is the restriction of σ to V by construction. In this case, as the unifiers σ_i, $1 \leq i \leq n$, returned by CSU_B are idempotent and away from all the variables that have previously appeared in the computation, so $ran(\sigma_i) \cap \bigcup_{j=1}^{i-1} ran(\sigma_j) = \emptyset$, then ν is also idempotent.

Although several rules allow for simplification in the reachability formula obtained, e.g., we can replace $X - Y + Z > 0 \wedge X = Y$ with $Z > 0$, it is always possible to obtain the same computed answer without using simplifications.

Proposition 17 (Canonical narrowing path). *Given $\mathcal{R}_B = (\Sigma, E_0 \cup B, R_B)$, an associated rewrite theory of $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ closed under B-extensions, and a narrowing path from a goal G (with set of parameters V), $G = \Delta_0 \mid \psi_0 \mid V, none \rightsquigarrow_{\sigma_1} \Delta_1 \mid \psi_1 \mid V, \nu_1 \rightsquigarrow_{\sigma_2} \cdots \Delta_{m-1} \mid \psi_{m-1} \mid V, \nu_{m-1} \rightsquigarrow_{\sigma_m} nil \mid \psi_m \mid V, \nu_m$, there exists another narrowing path $G = \Delta_0 \mid \psi_0 \mid V, none \rightsquigarrow_{\sigma_1} \Delta_1 \mid \chi_1 \mid V, \nu_1 \rightsquigarrow_{\sigma_2} \cdots \Delta_{m-1} \mid \chi_{m-1} \mid V, \nu_{m-1} \rightsquigarrow_{\sigma_m} nil \mid \chi_m \mid V, \nu_m$, where the same inference rule, with the same substitution, is applied at each step in both paths, there is no simplification of the reachability formula on the second path, and $E_0 \vdash \psi_i \Leftrightarrow \chi_i$, for $1 \leq i \leq m$.*

See proof on page 192.

6.6.2 Soundness and weak completeness of the calculus

The soundness and weak completeness, i.e., completeness with respect to $R/\mathcal{E}$-normalized solutions, of the calculus for reachability problems are now stated.

Theorem 16 (Soundness of the Calculus for Reachability Goals). *Given an associated rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ closed under B-extensions and a reachability goal G, if $\nu \mid \psi$ is a computed answer for G then for each substitution $\rho : V^\nu \to \mathcal{T}_\Sigma$ such that $\psi\rho$ is satisfiable, $\nu \cdot \rho$ is a solution for G.*

- $[d1]$ idle

$$\frac{u \to v/\texttt{idle}\ (\wedge\ \Delta) \mid \phi \mid V, \nu}{(\Delta\sigma) \mid \psi \mid V, (\nu\sigma)_V}$$

where $abstract_{\Sigma_1}((u,v)) = \langle\lambda(\bar{x},\bar{y}).(u^\circ, v^\circ); (\theta_u^\circ, \theta_v^\circ); (\phi_u^\circ, \phi_v^\circ)\rangle$, σ in $CSU_B(u^\circ = v^\circ)$, $vars(\psi) \subseteq vars((\phi \wedge \phi_u^\circ \wedge \phi_v^\circ)\sigma)$, $E_0 \vdash \psi \Leftrightarrow (\phi \wedge \phi_u^\circ \wedge \phi_v^\circ)\sigma$, and ψ is satisfiable

- $[d2]$ idle

$$\frac{u \to v/\texttt{idle}\ ;\ ST\ (\wedge\ \Delta) \mid \phi \mid V, \nu}{u \to v/ST\ (\wedge\ \Delta) \mid \phi \mid V, \nu}$$

- $[o1]$ or

$$\frac{u \to v/(ST_1 \mid ST_2)\ ;\ ST\ (\wedge\ \Delta) \mid \phi \mid V, \nu}{u \to v/ST_1\ ;\ ST\ (\wedge\ \Delta) \mid \phi \mid V, \nu}$$

- $[o2]$ or

$$\frac{u \to v/(ST_1 \mid ST_2)\ ;\ ST\ (\wedge\ \Delta) \mid \phi \mid V, \nu}{u \to v/ST_2\ ;\ ST\ (\wedge\ \Delta) \mid \phi \mid V, \nu}$$

- $[p1]$ plus

$$\frac{u \to v/ST_1{+}\ ;\ ST\ (\wedge\ \Delta) \mid \phi \mid V, \nu}{u \to v/ST_1\ ;\ ST\ (\wedge\ \Delta) \mid \phi \mid V, \nu}$$

- $[p2]$ plus

$$\frac{u \to v/ST_1{+}\ ;\ ST\ (\wedge\ \Delta) \mid \phi \mid V, \nu}{u \to v/ST_1\ ;\ ST_1{+}\ ;\ ST\ (\wedge\ \Delta) \mid \phi \mid V, \nu}$$

- $[s1]$ star

$$\frac{u \to v/ST_1{*}\ ;\ ST\ (\wedge\ \Delta) \mid \phi \mid V, \nu}{u \to v/ST\ (\wedge\ \Delta) \mid \phi \mid V, \nu}$$

- $[s2]$ star

$$\frac{u \to v/ST_1{*}\ ;\ ST\ (\wedge\ \Delta) \mid \phi \mid V, \nu}{u \to v/ST_1{+}\ ;\ ST\ (\wedge\ \Delta) \mid \phi \mid V, \nu}$$

- $[i1]$ if then else

$$\frac{u \to v/\texttt{match}\ t\ \texttt{s.t.}\ \phi'\ ?\ ST_1 : ST_2\ ;\ ST\ (\wedge\ \Delta) \mid \phi \mid V, \nu}{(u \to v\ /ST_1\ ;\ ST\ (\wedge\ \Delta))\sigma \mid \psi \mid V, (\nu\sigma)_V}$$

where $abstract_{\Sigma_1}((u,t)) = \langle\lambda(\bar{x},\bar{y}).(u^\circ, t^\circ); (\theta_u^\circ, \theta_t^\circ); (\phi_u^\circ, \phi_t^\circ)\rangle$, σ in $CSU_B(u^\circ = t^\circ)$, $vars(\psi) \subseteq vars((\phi \wedge \phi' \wedge \phi_u^\circ \wedge \phi_t^\circ)\sigma)$, $E_0 \vdash \psi \Leftrightarrow (\phi \wedge \phi' \wedge \phi_u^\circ \wedge \phi_t^\circ)\sigma$, and ψ is satisfiable

- $[i2]$ if then else

$$\frac{u \to v/\texttt{match}\ t\ \texttt{s.t.}\ \phi'\ ?\ ST_1 : ST_2\ ;\ ST\ (\wedge\ \Delta) \mid \phi \mid V, \nu}{(u \to v\ /ST_2\ ;\ ST\ (\wedge\ \Delta))\sigma \mid \psi \mid V, (\nu\sigma)_V}$$

where $abstract_{\Sigma_1}((u,t)) = \langle\lambda(\bar{x},\bar{y}).(u^\circ, t^\circ); (\theta_u^\circ, \theta_t^\circ); (\phi_u^\circ, \phi_t^\circ)\rangle$, σ in $CSU_B(u^\circ = t^\circ)$, $vars(\psi) \subseteq vars((\phi \wedge \neg\phi' \wedge \phi_u^\circ \wedge \phi_t^\circ)\sigma)$, $E_0 \vdash \psi \Leftrightarrow (\phi \wedge \neg\phi' \wedge \phi_u^\circ \wedge \phi_t^\circ)\sigma$, and ψ is satisfiable

Figure 6.2: Inference rules for reachability with strategies modulo SMT plus axioms I

- $[t]$ transitivity

$$\frac{u \to v/RA\,;ST\ (\wedge\,\Delta) \mid \phi \mid V, \nu}{u \to^1 x_k, x_k \to v/RA\,;ST\ (\wedge\,\Delta) \mid \phi \mid V, \nu}$$

where RA is a rule application, $u \in \mathcal{H}_\Sigma(\mathcal{X}) \setminus \mathcal{X}$, $k = [ls(u)]$, and x_k fresh variable

- $[c]$ congruence

$$\frac{u|_p \to^1 x_k, u[x_k]_p \to v/RA\,;ST\ (\wedge\,\Delta) \mid \phi \mid V, \nu}{u_i \to^1 y_{k'}, u[y_{k'}]_{p.i} \to v/RA\,;ST\ (\wedge\,\Delta) \mid \phi \mid V, \nu}$$

where RA is a rule application, $u|_p = f(u_1, \ldots, u_n)$, $u_i \in \mathcal{H}_\Sigma(\mathcal{X}) \setminus \mathcal{X}$,
$k' = [ls(u_i)]$, $y_{k'}$ fresh variable, and $\sigma_1 = \{x_k \mapsto u|_p[y_{k'}]_i\}$

- $[r]$ rule application

$$\frac{u|_p \to^1 x_k, u[x_k]_p \to v/c[\gamma]\{ST_1, \ldots, ST_n\}\,;ST\ (\wedge\,\Delta) \mid \phi \mid V, \nu}{(\bigwedge_{i=1}^n (l_i\gamma \to r_i\gamma/ST_i; \texttt{idle}) \wedge u[r\gamma]_p \to v\ /ST\ (\wedge\,\Delta))\sigma \mid \psi \mid V, (\nu\sigma)_V}$$

where $c : l \to r$ *if* $\bigwedge_{i=1}^n (l_i \to r_i) \mid \phi'$ fresh version, except for $dom(\gamma) \cup V^\nu$, of a rule c in R^ν,
$abstract_{\Sigma_1}((u|_p, l\gamma)) = \langle \lambda(\bar{u}, \bar{y}).(u^\circ, l^\circ); (\sigma_u^\circ, \sigma^\circ); (\phi_u^\circ, \phi_l^\circ)\rangle$, σ' in $CSU_B(u^\circ = l^\circ)$,
$\sigma = \sigma' \cup \{x_k \mapsto r\gamma\sigma'\}$, $vars(\psi) \subseteq vars((\phi \wedge \phi_u^\circ \wedge \phi_l^\circ \wedge (\phi'\gamma))\sigma)$,
$E_0 \vdash \psi \Leftrightarrow (\phi \wedge \phi_u^\circ \wedge \phi_l^\circ \wedge (\phi'\gamma))\sigma$, and ψ is satisfiable

- $[tp]$ top

$$\frac{u \to v/\texttt{top}(c[\gamma]\{ST_1, \ldots, ST_n\})\,;ST\ (\wedge\,\Delta) \mid \phi \mid V, \nu}{(\bigwedge_{i=1}^n (l_i\gamma \to r_i\gamma/ST_i; \texttt{idle}) \wedge r\gamma \to v\ /ST\ (\wedge\,\Delta))\sigma \mid \psi \mid V, (\nu\sigma)_V}$$

where $c : l \to r$ *if* $\bigwedge_{i=1}^n (l_i \to r_i) \mid \phi'$ fresh version, except for $dom(\gamma) \cup V^\nu$, of a rule c in R^ν,
$abstract_{\Sigma_1}((u, l\gamma)) = \langle \lambda(\bar{u}, \bar{y}).(u^\circ, l^\circ); (\sigma_u^\circ, \sigma^\circ); (\phi_u^\circ, \phi_l^\circ)\rangle$, σ in $CSU_B(u^\circ = l^\circ)$,
$vars(\psi) \subseteq vars((\phi \wedge \phi_u^\circ \wedge \phi_l^\circ \wedge (\phi'\gamma))\sigma)$, $E_0 \vdash \psi \Leftrightarrow (\phi \wedge \phi_u^\circ \wedge \phi_l^\circ \wedge (\phi'\gamma))\sigma$, and ψ is satisfiable

Figure 6.3: Inference rules for reachability with strategies modulo SMT plus axioms II

- $[m]$ match

$$\frac{u \to v/\texttt{match}\ t\ \texttt{s.t.}\ \bigwedge_{j=1}^{m}(l_j = r_j) \wedge \phi'\ ;\ ST\ (\wedge\ \Delta)\ \mid\ \phi\ \mid\ V, \nu}{(\bigwedge_{j=1}^{m}(l_j \to r_j/\texttt{idle}) \wedge u \to v\ /ST\ (\wedge\ \Delta))\sigma\ \mid\ \psi\ \mid\ V, (\nu\sigma)_V}$$

where $abstract_{\Sigma_1}((u,t)) = \langle \lambda(\bar{x},\bar{y}).(u^\circ, t^\circ); (\theta_u^\circ, \theta_t^\circ); (\phi_u^\circ, \phi_t^\circ)\rangle$, σ in $CSU_B(u^\circ = t^\circ)$,
$vars(\psi) \subseteq vars((\phi \wedge \phi' \wedge \phi_u^\circ \wedge \phi_t^\circ)\sigma)$, $E_0 \vdash \psi \Leftrightarrow (\phi \wedge \phi' \wedge \phi_u^\circ \wedge \phi_t^\circ)\sigma$, and ψ is satisfiable

- $[w]$ matchrew

$$\frac{u \to v/\texttt{matchrew}\ t[\bar{z}]_{\bar{p}}\ \texttt{s.t.}\ \bigwedge_{j=1}^{m}(l_j = r_j) \wedge \phi'\ \texttt{by}\ \bar{z}\ \texttt{using}\ \overline{ST}\ ;\ ST\ (\wedge\ \Delta)\ \mid\ \phi\ \mid\ V, \nu}{(\bigwedge_{j=1}^{m}(l_j\gamma \to r_j\gamma/\texttt{idle}) \wedge \bigwedge_{i=1}^{n}(x_i \to y_i/ST_i\gamma; \texttt{idle}) \wedge t[\bar{y}]_{\bar{p}} \to v/\ ST\ (\wedge\ \Delta))\sigma\ \mid\ \psi\ \mid\ V, (\nu\sigma)_V}$$

where $\bar{z} = z_1, \ldots, z_n$, $\overline{ST} = ST_1, \ldots, ST_n$, $\bar{x}$ and $\bar{y}$ fresh versions of $\bar{z}$, γ renaming from $\bar{z}$ to $\bar{x}$,
$abstract_{\Sigma_1}((u, t[\bar{x}]_{\bar{p}})) = \langle \lambda(\bar{w},\bar{w}').(u^\circ, t^\circ); (\theta_u^\circ, \theta_t^\circ); (\phi_u^\circ, \phi_t^\circ)\rangle$, σ in $CSU_B(u^\circ = t^\circ)$,
$vars(\psi) \subseteq vars((\phi \wedge \phi' \wedge \phi_u^\circ \wedge \phi_t^\circ)\sigma)$, $E_0 \vdash \psi \Leftrightarrow (\phi \wedge \phi' \wedge \phi_u^\circ \wedge \phi_t^\circ)\sigma$, and ψ is satisfiable

- $[c1]$ call strategy

$$\frac{u \to v/CS\ ;\ ST\ (\wedge\ \Delta)\ \mid\ \phi\ \mid\ V, \nu}{u \to v/ST_2\ ;\ ST\ (\wedge\ \Delta)\ \mid\ \phi\ \mid\ V, \nu} \qquad \frac{u \to v/CS(\bar{t})\ ;\ ST\ (\wedge\ \Delta)\ \mid\ \phi\ \mid\ V, \nu}{u \to v/ST_2\gamma\ ;\ ST\ (\wedge\ \Delta)\ \mid\ \phi\ \mid\ V, \nu}$$

where $\texttt{sd}\ CS := ST_1$, or $\texttt{sd}\ CS(\bar{x}) := ST_1$ in $Call_{\mathcal{R}}^{\nu}$, $\gamma = \{\bar{x} \mapsto \bar{t}\}$,
and ST_2 fresh version of ST_1, except for $dom(\gamma) \cup V^\nu$

- $[c2]$ call strategy

$$\frac{u \to v/CS(\bar{t})\ ;\ ST\ (\wedge\ \Delta)\ \mid\ \phi\ \mid\ V, \nu}{\bigwedge_{j=1}^{m}(l_j\gamma \to r_j\gamma/\texttt{idle}) \wedge u \to v/ST_2\gamma\ ;\ ST\ (\wedge\ \Delta)\ \mid\ \psi\ \mid\ V, \nu}$$

where $\texttt{csd}\ CS(\bar{x}) := ST_1\ \textit{if}\ C$ in $Call_{\mathcal{R}}^{\nu}$, $\gamma = \{\bar{x} \mapsto \bar{t}\}$,
$ST_2\ \textit{if}\ \bigwedge_{j=1}^{m}(l_j = r_j) \wedge \phi'$ fresh version of $ST_1\ \textit{if}\ C$, except for $dom(\gamma) \cup V^\nu$,
$vars(\psi) \subseteq vars(\phi'\gamma \wedge \phi)$, $E_0 \vdash \psi \Leftrightarrow \phi'\gamma \wedge \phi$, and ψ is satisfiable

Figure 6.4: Inference rules for reachability with strategies modulo SMT plus axioms III

See proof on page 193.

Theorem 17 (Weak Completeness of the Calculus for Reachability Goals)**.** *Given an associated rewrite theory* $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ *closed under* B*-extensions and a reachability problem* $P = \bigwedge_{i=1}^{n} u_i \to v_i / ST_i \mid \phi \mid V, \mu$*, where* μ *is* $R/\mathcal{E}$*-normalized, if* $\sigma : V \to \mathcal{T}_\Sigma$ *is a* $R/\mathcal{E}$*-normalized solution for* P *then there exist a formula* $\psi \in QF(\mathcal{X}_0)$ *and two substitutions, say* λ *and* ρ*, such that* $\bigwedge_{i=1}^{n} u_i\mu \to v_i\mu / ST_i^{\mu}; \texttt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow_{\lambda}^{+} nil \mid \psi \mid V, \nu$*,* $\sigma =_{\mathcal{E}} \nu \cdot \rho$*, and* $\psi\rho$ *is satisfiable, where* $\nu = (\mu\lambda)_V$*.*

See proof on page 223.

Remark 7. *In the previous theorem, by Definition 23 there exists a substitution* σ' *such that* $\sigma = \mu \cdot \sigma'$*. As* σ *is* $R/\mathcal{E}$*-normalized then, by Proposition 18,* μ *has to be* $R/\mathcal{E}$*-normalized too. Also, as* σ *is* $R/\mathcal{E}$*-normalized and the substitution* η *obtained after each narrowing step is always a generalization of* σ *then, by Proposition 19,* η *is* $R/\mathcal{E}$*-normalized too.*

6.6.3 Completeness of the calculus, for topmost rewrite theories

In the proof of weak completeness of the calculus for reachability, the only places where the hypothesis of σ being $R/\mathcal{E}$-normalized is used are in the initial substitution μ and in the induction case, (ii), where it limits the positions where rewriting can happen at some proper subterm of $u_1\sigma$, an instance of the first term in the reachability problem P (u_1). It is immediate then to prove the *completeness of the calculus for topmost rewrite theories*, since rewriting always happens at position ϵ of $u_1\sigma$, so the hypothesis of σ being $R/\mathcal{E}$-normalized is not needed for this type of rewrite theories in the proof of completeness, where no variable in V has sort `state`, so μ is $R/\mathcal{E}$-normalized.

6.7 Narrowing example: toasts with strategies

Three applications of the calculus using the running example are shown. Recall the abbreviations: `i` − `Integer`, `p` − `Pan`, `rt` − `RealToast`, `t` − `Toast`, `k` − `Kitchen`, `b` − `Bin`, `s` − `System`, $\texttt{ct}_\texttt{i}$ − `cookTime`, and $\texttt{ft}_\texttt{i}$ − `failTime`. We will omit the use of the subscript `i` in all variables for readability. In all cases we take `ct` = 20.

6.7.1 Applications

- In the first application, from an initial system with an empty `Pan`, an empty `Dish`, and at most one `Toast` in the `Bin`, we want to reach in no more than 60 seconds a final system where there are three `RealToast`s in the `Dish` and all the remaining elements are empty.

- In the second application, we want to know if there is any value for `ft` lower than 61 seconds that allows us to get from an initial system where there are three `RealToast`s in the bag and the remaining elements are empty to the same final system as in the previous case.

- The third application is like the second one but increasing the time limit from 61 to 62 seconds.

In Section 6.4.2 we saw the syntax in Maude of the following set $Call_{\mathcal{R}}$ of call strategy definitions, designed to solve the problems for this example:

- sd test := match $NS/BS_{\mathtt{b}}/YS$; $VS_{\mathtt{t}}\,WS_{\mathtt{t}}/OKS$ s.t. $YS < \mathtt{ft}$
- sd cook1 := matchrew $NS/BS_{\mathtt{b}}/KS_{\mathtt{k}}/OKS$ s.t. YS; $RS_{\mathtt{rt}}\,VS_{\mathtt{t}} := KS_{\mathtt{k}}$ by $KS_{\mathtt{k}}$ using kitchCook
- sd kitchCook := top(kitchen[*none*]) ; top(cook[*none*]{toasts, toasts})
- sd toasts := top(toast1[*none*]) | top(toast2[*none*])
- sd noCook := top(bin[*none*]) | top(pan[*none*]) | top(dish[*none*])
- sd loop := (noCook | (cook1 ; test ; noCook))+
- sd solve1 := top(bag[*none*]) ; top(bag[*none*]) ; top(bag[*none*]) ; loop
- sd solve2 := top(bag[*none*]) ; top(bag[*none*]) ; (top(bag[*none*]) | idle) ; loop

Our reachability problems are:

$P_1 = N \,/\, T_{\mathtt{t}} \,/\, 0 \,;\, \mathtt{zt}\,\mathtt{zt} \,/\, 0 \;\rightarrow\; 0 \,/\, \mathtt{zt} \,/\, Y \,;\, \mathtt{zt}\,\mathtt{zt} \,/\, 3 \,/\, \mathtt{solve2} \;\mid\; N > 0 \wedge N < 3 \;\mid\; \{\mathtt{ct}, \mathtt{ft}, N, T_{\mathtt{t}}, Y\}, \{\mathtt{ct} \mapsto 20, \mathtt{ft} \mapsto 61\}$

$P_2 = 3/\mathtt{zt}/0; \mathtt{ztzt}/0 \rightarrow 0/\mathtt{zt}/Y; \mathtt{ztzt}/3/\mathtt{solve1} \mid \mathtt{ft} < 61 \mid \{\mathtt{ct}, \mathtt{ft}, Y\}, \{\mathtt{ct} \mapsto 20\}$

$P_3 = 3/\mathtt{zt}/0; \mathtt{ztzt}/0 \rightarrow 0/\mathtt{zt}/Y; \mathtt{ztzt}/3/\mathtt{solve1} \mid \mathtt{ft} < 62 \mid \{\mathtt{ct}, \mathtt{ft}, Y\}, \{\mathtt{ct} \mapsto 20\}$

The most important feature of $Call_{\mathcal{R}}$ is the invocation of the call strategy test after each invocation of cook1 in the loop strategy definition:

- sd test := match $NS/BS_{\mathtt{b}}/YS$; $VS_{\mathtt{t}}\,WS_{\mathtt{t}}/OKS$ s.t. $YS < \mathtt{ft}$
- sd loop := (noCook | (cook1 ; test ; noCook))+

This renders the search state space of both problems finite, since the strategy nocook+ has a finite state space from any initial state and cook1 always increases the timer, so there is a limit in both problems in the value of ft, that gets checked against the timer, which initially has value 0, through the invocation of test.

Further pruning of the search tree is achieved through several facts: (i) all rule applications are used inside top strategies, preventing rule congruence of the narrowing calculus to be applied, (ii) in the call strategy definition cook1, where a rule must be applied in a subterm of the state, the matchrew strategy selects the precise subterm where to apply a top strategy in a more efficient way than the blind search of rule applications, and (iii) the use of the call strategy noCook after test prevents consecutive calls to cook1 since rule toast2 always well-toasts one side, so it cannot be invoked in the next strategy call.

In P_1, as we can infer from the problem that, initially, there must be either two or three toasts in the bag, we impose in solve2 the application of the rule bag twice, followed by the nondeterministic strategy top(bag[*none*]) | idle, before applying any other rule, also preventing its application later, pruning the search tree. In the initial state we use the

variable T_t to represent the bin. This use is valid because `Toast` is a subsort of `Bin`, and it also covers both initial cases: the one without `RealToast`s in the `Bin` and the one with one `RealToast` in the `Bin`, since both `EmptyToast` and `RealToast` are subsorts of `Toast`.

Among the answers returned by the prototype for narrowing with strategies (see Section 6.8) we have:

a - $\texttt{ct} \mapsto 20, \texttt{ft} \mapsto 61, N \mapsto 3,\ Y \mapsto 60,\ T_t \mapsto \texttt{zt}$,

b - $\texttt{ct} \mapsto 20, \texttt{ft} \mapsto 61, N \mapsto 2,\ Y \mapsto 60,\ T_t \mapsto [0, 0]$,

c - $\texttt{ct} \mapsto 20, \texttt{ft} \mapsto 61, N \mapsto 2,\ Y \mapsto 40,\ T_t \mapsto [20, 20]$, and

d - $\texttt{ct} \mapsto 20, \texttt{ft} \mapsto 61, N \mapsto 2,\ Y \mapsto 40 + U + V,\ T_t \mapsto [C, D]$ *such that*
$C + U = 20 \wedge D + V = 20 \wedge U + V \leq 20 \wedge U > 0 \wedge V > 0$,

stating that we need 60 seconds when (a) 3 `RealToast`s are in the bag or (b) 2 `RealToast`s are in the bag and one fresh `RealToast` is in the `Bin` ($T_t \mapsto [0, 0]$). The required amount of time can be smaller: (c) 40 seconds if the `RealToast` in the `Bin` is well-cooked ($T_t \mapsto [20, 20]$), or, if not well-cooked, (d) 40 seconds plus the remaining toasting time for the `RealToast` in the `Bin`, as long as this remaining time is not above 20 seconds ($U+V \leq 20$).

In P_2, as we know that there are three `RealToast`s in the bag, we impose in `solve1` the application of the rule `bag` three times before applying any other rule, also preventing its application later, pruning the search tree. This problem has only one initial state, but what we are trying to find is a value for the parameter `ft` that fits the constraints of the problem. The search for a solution ends, because our search state space is finite thanks to the call strategy `test` that prunes all the narrowing paths where $Y \geq 60$, without finding a solution, so `ft` cannot be given a value below 61.

For P_3, where we allow `ft` to be below 62 seconds instead of 61, the prototype returns the answer $Y \mapsto 60$ *such that* $\texttt{ft} < 62 \wedge \texttt{ft} > 60$, i.e., we can cook three toasts in 60 seconds when $\texttt{ft} = 61$, fulfilling all the constraints of the problem.

6.7.2 Optimization of the call strategy `noCook`

In Section 6.4.2 a second version of the call strategy definition `noCook` was suggested as an improvement to the toast cooking specification. In the one used for the applications in the previous section:

- `sd noCook` := `top(bin`[*none*]) | `top(pan`[*none*]) | `top(dish`[*none*])

a well-cooked `RealToast` can go back to the `Bin` instead of the `Dish` because the `top(bin)` strategy is available. With its improved definition

- `sd noCook` := `match` $NS/BS_b/YS$; [`ct`, `ct`] VS_t/OKS ?
`top(dish`[*none*] : `top(bin`[*none*]) | `top(pan`[*none*])

if there is a well-cooked `RealToast` in the `Pan` then it has to be moved to the `Dish` since the if-then-else strategy will select the `top(dish)` strategy as the only available option.

When compared against the first problem from the previous section
$P_1 = N$ / T_t / 0 ; `zt zt` / 0 → 0 / `zt` / Y ; `zt zt` / 3 / `solve2` | $N > 0 \wedge N < 3$ |
$\{\texttt{ct}, \texttt{ft}, N, T_t, Y\}, \{\texttt{ct} \mapsto 20, \texttt{ft} \mapsto 61\}$

where from an initial system with an empty `Pan`, an empty `Dish`, and at most one `Toast` in the `Bin`, we want to reach in no more than 60 seconds a final system where there are three `RealToast`s in the `Dish` and all the remaining elements are empty, we get the following results, all against the original definition (i) and in favor of the improved definition (ii):

- First answer:

  (i) `states: 338779 rewrites: 58524365 in 54312 ms`

  (ii) `states: 242985 rewrites: 40792938 in 34336 ms`

- Last answer:

  (i) `states: 4896122 rewrites: 1215048778 in 1101436 ms`

  (ii) `states: 1621706 rewrites: 321709975 in 267756 ms`

- End of search:

  (i) `states: 5389440 rewrites: 1387211660 in 1278016 ms`

  (ii) `states: 1731977 rewrites: 353839361 in 293280 ms`

This simple optimization in the definition of one call strategy has allowed for over a 67% prunning of the state space, over a 74% reduction in the number of rewrites and over a 77% reduction in running time, way beyond our own expectations: we could never have imagined that allowing to put a well-cooked `RealToast` back in the `Bin` would account for two thirds of the state space.

6.7.3 Relation with sentence-normalized rewriting

This section is an informal one, where we point out some details about the specification of our running example. No formal proof of our claims is given.

The calculus in this chapter only admits equational theories of the form $(\Sigma, E_0 \cup B)$, with no other equations. A more natural specification, shown below, of the running example requires the use of other equations apart from the admitted ones.

In our specification, the use of strategies allows us to overcome this limit and emulate sentence-normalized rewriting for OS rewrite theories, with the added support for strategies.

Although the system module `TOASTS` in the example has no equations, except for E_0 and the hidden SMT specification, the definition of R combined with the strategy module `TOASTS-STRAT`, designed to give priority to some rules over the rest, allows some of these rules to act as if they were oriented equations.

The function definitions and rules, in Maude syntax:

```
op cook : Kitchen Integer -> Kitchen .
op toast : Toast Integer -> Toast .

crl [kitchen] : Y ; R V => cook(Y ; R V, Z)
  if (Z > 0) = (true).Boolean [nonexec] .
crl [cook] : cook(Y ; V W, Z) => Y + Z ; V1 W1
  if toast(V, Z) => V1 /\ toast(W, Z) => W1 .
```

```
rl [toast1] : toast(zt, Z) => zt .
crl [toast2] : toast([C, D], Z) => [C + Z, D]
  if (C >= 0 and C + Z === cookTime) = (true).Boolean [nonexec] .
```

and the strategy definitions:

```
sd cook1 := matchrew NS / BS / KS / OKS s.t.
            YS ; RS VS := KS by KS using kitchCook .
sd kitchCook := top(kitchen) ; top(cook{toasts, toasts}) .
sd toasts := top(toast1) | top(toast2) .
```

when combined act as an OS-theory where reachability steps are applied only when no unification step is possible, but extended with the use of strategies. The aforementioned OS equational theory that is equivalent to the specification of our running example has, apart from E_0 and the axioms B, the equations

```
ceq [cook] : cook(Y ; V W, Z) = Y + Z ; V1 W1
  if V1:= toast(V, Z) /\ W1 := toast(W, Z) .
eq [toast1] : toast(zt, Z) = zt .
ceq [toast2] : toast([C, D], Z) = [C + Z, D]
  if (C >= 0 and C + Z === cookTime) = (true).Boolean [nonexec] .
```

replacing their corresponding rules, and the call strategy definition

```
sd cook1 := matchrew NS / BS / KS / OKS s.t.
            YS ; RS VS := KS by KS using top(kitchen) .
```

replacing the three call strategy definitions shown above.

If a narrowing calculus similar to the one for sentence-normalized conditional narrowing modulo developed in Chapter 4, extended with the use of strategies, were to be used, it would work in the following way, where we focus on the `System` evolution and omit the SMT constraints: after narrowing with rule `kitchen`, the current `System`, say $N/B/Y;R\ V/OK$, becomes a `System` $N/B/\texttt{cook}(Y;R\ V,Z)/OK$, to which only unification steps apply, using the added equations, until a normalized term, i.e., one where neither the function `cook` nor the function `toast` appear.

This is exactly what the strategy definitions `cook1`, `kitchCook`, and `toasts` also achieve in the running example: after applying a narrowing step with rule `kitchen`, we force the application of a narrowing step with rule `cook`, using for each condition the suitable rule, either `toast1` or `toast2`, that is applied to each one of the `Toasts` in the `Pan`.

6.8 Prototype for narrowing with strategies

In this section we present the prototype of the narrowing calculus with strategies and SMT solvers shown in this chapter, that has been implemented using version 3 of the Maude engine. It can be found in the web page for this thesis https://maude.ucm.es/cnarrowing.

This prototype inherits the management of the SMT constraints developed for the prototypes in the previous chapter, that has been explained in Section 5.5, so it implements not only the usual search for partial solutions via unification, but also the

extraction of partial solutions for the SMT variables in the reachability problems via the inspection of the evolving SMT constraints of the computation. It also inherits the capability to simplify SMT expressions through the functional module `SMTLOGIC`, presented in Section 5.5.1.

The prototype, developed as a system module, gives support to the two new features in the calculus: the use of a strategy language and the admission of variable SMT parameters both in the specifications and problems. A list of parameters together with a substitution that instantiates partially or totally these parameters are provided with each reachability problem. The list of parameters may be empty and the substitution may be the identity substitution.

As the prototype has to handle strategies with SMT constraints at the metalevel, an extension of the existing sorts in the metalevel of Maude 3 has been defined to give support to these strategies.

Meta SMT strategies syntax

From the functional module `META-STRATEGY` in the prelude of Maude, that supports strategies at the metalevel, we have developed an extension of it: the functional module `META-SMT-STRAT`, found in the file `meta-smt-strat.maude`, to deal at the metalevel with our strategies with mixed equational and SMT conditions. The constructor function for this type of conditions is:

```
op _&&_ : EqCondition SmtCondi -> SmtStratCondi [ctor prec 19] .
```

The module has neither equations nor rules, it just defines the syntax of the new sorts for our prototype, being the main sort `SmtStrategy`, an extension of the sort `Strategy` in `META-STRATEGY` that admits strategies with an `SmtStratCondi`. From this sort, the sort `SmtStratDef`, an extension of the sort `StratDef` in `META-STRATEGY` that admits `SmtStrategies`, is also defined in this module.

Meta SMT strategies parsing

We have used the fact that SMT constraints have sort `Boolean` to include them in rules and strategies as equational conditions, by comparing the equality of each SMT constraint with the constant `true` of sort `Boolean`, so that the parser in Maude 3 does not complain when parsing any rule or strategy.

Then, we have developed the functional module `PARSE-SMT-STRAT`, found in the file `smtStratParse.maude`, to take a term with sort `EqCondition` and generate a term with sort `SmtStratCondi`, using the function `ec2ssc`, that can be handled by the prototype.

The main function in this module, called `parseStrat`, is applied to every defined strategy to compute the constant `smtStratDefSet`, i.e., the set of defined call strategies with SMT constraints, that has sort `SmtStratDefSet`, an extension of the sort `StratDefSet` in the prelude of Maude. The function `parseStrat` is recursively invoked to parse some of the admitted strategies.

The module has another function, called `parseSmtStrat`, that given a list of `Qids` generates a `SmtStrategy`. A `Qid`, quoted identifier, for instance `'TOASTS`, is the base sort of the metalevel. As an example, the sorts `Constant` and `Variable` of the metalevel of Maude are both subsorts of `Qid`.

The function `parseSmtStrat` is used in our prototype to parse the strategy given with the reachability problem to solve, since it is expressed as a string of characters.

Syntax of the reachability problems

In the prototypes developed for the calculus of the previous chapter, the name of the system module under test had to be defined inside the prototype, as a constant named `target`, before posing any problem.

To avoid the need to modify the source code of the prototype for each new module to test, in the prototype developed for this calculus it is mandatory to define in another file the functional module `TARGET-AND-STRATEGIES`. This module has to include the constants `target` and `strat` with the meta-names of the system and the strategy modules that the problems will use, respectively. These constants will be used in the `META-LEVEL` by the system modules `PARSE-SMT-STRAT` and `NARROW-SMT-STRAT`, the main module of the prototype, that are loaded after loading the `target` and `strat` modules. At this moment any problem related to the current target and strategy modules can be posed. For instance, the code for the third application in Section 6.7.1 is:

```
fmod TARGET-AND-STRATEGIES is
protecting QID .

  op target : -> Qid .
  eq target = 'TOASTS .

  op strat : -> Qid .
  eq strat = 'TOASTS-STRAT .
endfm

load toasts
load toasts-strat
load smtStratNarrow

mod TEST is
protecting REAL-INTEGER .
protecting TOASTS .
protecting NARROW-SMT-STRAT .

  vars failTime cookTime : Integer  .
  var YP : Integer  .
  var SOL : Solution .
endm

search in TEST :
problem(upTerm(3 / zt / 0 ; zt zt / 0) ->* upTerm(0 / zt / YP ; zt zt / 3)
        && upTerm(failTime < 62), "solve1",
        (upTerm(cookTime) ; upTerm(failTime)),
        upTerm(cookTime) <- '20.Integer)
  =>* SOL .
```

We first define the meta-names of the target and the strategy modules as `'TOASTS` and `'TOASTS-STRAT`, respectively. The files that contain the system module `TOASTS` and the strategy module `TOASTS-STRAT` are then loaded, followed by the file `smtStratNarrow.maude`,

that contains the main module of the prototype. Finally we define a system module called `TEST`, holding the system under test, in this case `TOASTS`, the name of the prototype module, `NARROW-SMT-STRAT`, and the parameters and variables that are used in the problem.

The problem is stated inside the `TEST` module using the following terms, most of them converted to `META-LEVEL TERM`s by the function `upTerm`:

- the initial state: `3 / zt / 0 ; zt zt / 0 ;`
- the final state: `0 / zt / YP ; zt zt / 3;`
- the initial SMT constraint: `failTime < 62;`
- the strategy to use, in this case a call strategy inside the `TOASTS-STRAT` strategy module: `solve1`;
- the parameters of the specification: `cookTime` and `failTime`; and
- the initial solution that sets a constant value for one parameter: {`cookTime` $\mapsto$ 20}.

The function `problem` uses these items to generate a term with sort `SimpleProblem`, of which the sort `Solution` is a subsort, and the Maude engine uses the rules in the prototype to try to rewrite this term with sort `SimpleProblem` to a term with sort `Solution`. The only equation that defines this function is:

```
ceq problem(OP && TC, S, VS, SU)
 = sp((op2Rc(OP, nilRCS, getSmtStrat(parseSmtStrat(tokenize(S), 0, OVS)))
     <<< SU || smtSimplifyC(t2smtCondi(TC <<< SU)) - OVS ; 0 ; SU), nilU)
   if OVS := (VS ; getVarSetOp(OP) ; getVarSet(TC)) .
```

In this equation:

- `OP` is the original problem: the function `_->*_`, that has as parameters the metalevel versions of the initial and final states;
- `TC` is the metalevel version of the SMT constraint that the solutions have to verify;
- `S` is the string that defines the strategy that has to be applied to the initial state;
- `VS` is the set of non-evident parameters of the problem (see explanation below); and
- `SU` is the metalevel version of the initial solution. In our example, `upTerm(cookTime) <- '20.Integer` shows a mixed use of `upTerm` and a term with sort `Qid`.

`VS` may not be the real set of parameters of the problem since, by definition, all the variables in the problem are parameters of it, i.e., `OVS` is the set of parameters of the problem. `VS` allows us to include as a parameter any variable that does not appear in the problem.

The function `tokenize` turns the string `S` into a list of `Qid`s. From this list, the function `parseSmtStrat` generates the `SmtStrategy` that will be applied to the initial state.

The function `_<<<_` applies metalevel substitutions to the new sorts in the prototype.

The function `op2Rc` turns the instantiated initial state `L`, final state `R`, and SMT strategy `SST`, all in their metalevel forms, into the term `L =>* R / SST`, that has sort `ReachCondiStrat`.

The function `t2smtCondi` turns the instantiated term of the metalevel `TC <<< SU`, that has sort `Term`, into a term with sort `SmtCondi`, that can be handled by the module `SMTLOGIC` of the prototype, in this case by applying its `smtSimplifyC` function that simplifies terms with sort `SmtCondi`.

Finally, the function `sp` serves as a context that allows the application of the rules of the prototype only where it is desirable.

Normal form of the applied rules and states in the system under test

The prototype does not compute a normal form of the rules in the specification. Given a rewrite theory $\mathcal{R} = (\Sigma, E_0 \cup B, R)$, the prototype computes a closure under B-extensions of R, and generates a set of transformed rules. This is an excerpt of this generation:

```
op qSmtRls : -> QidSmtRuleSet [memo] .
eq qSmtRls = transformRls(getRls(axCohComplete(upModule(target, false)))) .
...
eq transformRl(crl L => R if CO /\ (TC = 'true.Boolean) [label(Q) Ats] .) =
   [ Q // L => R if rc2Src(CO) && t2smtCondi(TC) ] .
...
```

The constant `qSmtRls` holds the transformed version of each rule in the closure under B-extensions of the `target` module, that is generated by the `axCohComplete` function provided by the `AX-COHERENCE-COMPLETION` function module in the `full-maude31.maude` file included in Maude 3.

The function `transformRls` calls the function `transformRl` with each rule in this closure under B-extensions. In the equation that we have selected as an example, the label `Q` may not be unique, since it comes from a closure, and is used as a selector to filter out all the rules that may not be applied by a `rule application` or `top` strategy, `CO` is the conditional equation of the rule, and `TC` is the SMT constraint, distinguished by the use of the comparison against `'true.Boolean`. There is some internal reformatting for ease of use.

Each time that a `rule application` or `top` strategy is applied the term that is going to be unified with the head of a rule is abstracted, for instance:

```
crl [T] :
sp(L =>* R / top(Q[SU]{SSTL}) ;; SST & RCS || SC - OVS ; NV ; AN, UPL)
 => sp(nt(L0 =>1 V & V =>* R / SST & RCS || SC and SC',
       OVS ; NV' ; AN, Q, SU, SSTL), UPL)
    if V := newVarMemo(NV, getKindTermMemo(L)) /\
       (L0 / SC' ; NV') := abstract(L, s(NV), OVS) .
```

Then, the label of the rule in the strategy is used to select one of the rules in `qSmtRls` with the same label, a fresh version of the rule, except for the parameters of the rule, is generated, the substitution that appears in the strategy is applied to this fresh version, the head of this instance is abstracted, and the first unifier of both abstractions, with index `0`, is computed. For instance, if we want to unify a constant and the head of a rule:

```
crl [nt] :
nt(C =>1 V & V =>* R / SST & RCS || SC, OVS ; NV ; AN, Q, SU, SSTL)
```

```
  => n*(1 / NV'', L0, R', RC / reverseSub (SU', OVS) / C =>1 V & V =>* R /
     SST & RCS || SC and SC' and SC'', OVS ; NV''' ; AN, SSTL)
     if [ Q // SRL ] QSRLS := qSmtRls /\
     NV' ; L' => R' if RC && SC' :=
     (freshRule(SRL, NV, (OVS ; domain(SU))) <<< (AN ; SU)) /\
     (L0 / SC'' ; NV'') := abstract(L', NV', OVS) /\
     {SU', NV'''} := metaUnify(targetNoRls, C =? L0, NV'', 0) .
```

There are different rules to apply the generated unifier or to try to generate another unifier. Both rules will be used by the search engine in Maude, that will also discard any generated state if it has already been visited.

Depth-oriented generation of unifiers

Again, we have designed the rules of the prototype so that the set of unifiers for any given pair of terms is generated in a depth-oriented way, i.e., if the first unifier is tried at level n of the search tree, then the second unifier is tried at level $n + 1$, and so on, so that the search tree of the reachability problem is fairer to all the rules that can be applied to any state in the search tree, and also to support potentially infinite unification of terms without losing completeness.

Example 32. *Our search command of the third application in Section 6.7.1:*

```
search in TEST :
problem(upTerm(3 / zt / 0 ; zt zt / 0) ->* upTerm(0 / zt / YP ; zt zt / 3)
        && upTerm(failTime < (62).Integer), "solve1",
        upTerm(cookTime) ; upTerm(failTime),
        upTerm(cookTime) <- '20.Integer) =>* SOL .
```

asks for all the solutions to the reachability problem: is it possible from a `State` with just 3 `Toasts` in the bag to reach a `State` with just 3 well-cooked `Toasts` in the dish if `failTime` is lower than 62 seconds and it takes 20 seconds to cook each side of a `Toast`?

The constructor for the sort `Solution` *of the variable* `Sol` *is:*

```
op _where_ : SubstSol BoolSol -> Solution [ctor] .
```

Then, the search engine of Maude 3 returns:

```
Solution 1
SOL --> (cookTime <- (20).Integer ; YP <- (60).Integer)
where failTime < (62).Integer /\ failTime > (60).Integer

No more solutions.
```

This answer means that there is only one solution where a total cook time (`YP`*) of 60 seconds has been added to the initial substitution and the SMT constraint for this solution is* `failTime < (62).Integer / failTime > (60).Integer`*, i.e.,* `failTime` *has to be 61 seconds.*

6.9 Results and proofs

This section holds some needed technical results for this chapter, together with the proofs for all the results.

Lemma 11 (Equivalence of $R/\mathcal{E}$-rewriting and $R_B/\mathcal{E}$-rewriting)**.** *If $\mathcal{R}_B = (\Sigma, \mathcal{E}, R_B)$ is an associated rewrite theory of $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ closed under B-extensions then $\rightarrow^1_{R/\mathcal{E}} = \rightarrow^1_{R_B/\mathcal{E}}$ and $\rightarrow_{R/\mathcal{E}} = \rightarrow_{R_B/\mathcal{E}}$.*

Proof. Since $R \subseteq R_B$ then $\rightarrow^1_{R/\mathcal{E}} \subseteq \rightarrow^1_{R_B/\mathcal{E}}$ and $\rightarrow_{R/\mathcal{E}} \subseteq \rightarrow_{R_B/\mathcal{E}}$.

In order to prove $\rightarrow^1_{R_B/\mathcal{E}} \subseteq \rightarrow^1_{R/\mathcal{E}}$ and $\rightarrow_{R/\mathcal{E}} \subseteq \rightarrow_{R_B/\mathcal{E}}$, we will prove a stronger pair of assertions:

(i) if $t \underset{c,u}{\longrightarrow^1}_{R_B/\mathcal{E}} v$, where c in R_B, then $t \underset{c,u}{\longrightarrow^1}_{R/\mathcal{E}} v$ using the same number of rewrite steps, and

(ii) if $t \rightarrow_{R_B/\mathcal{E}} v$ then $t \rightarrow_{R/\mathcal{E}} v$ using the same number of rewrite steps.

We use induction on the number of $\rightarrow^1_{R_B/\mathcal{E}}$ rewrite steps of the derivations, including those in the condition of the rule.

Base cases:

(i) one rewrite step: $t \underset{c,u,p,\sigma}{\longrightarrow^1}_{R_B/\mathcal{E}} v$ with a rule $c : \tilde{l} \rightarrow \tilde{r}$ *if* $\bigwedge_{i=1}^n l_i \rightarrow r_i \mid \phi$ in R_B. As there is only one rewrite step in the derivation, it must be the case that $l_i\sigma \rightarrow_{R_B/\mathcal{E}} r_i\sigma$ in zero rewrite steps, $1 \leq i \leq n$. Then $l_i\sigma =_{\mathcal{E}} r_i\sigma$, so $l_i\sigma \rightarrow_{R/\mathcal{E}} r_i\sigma$ in zero rewrite steps, $1 \leq i \leq n$. Also, $t =_{\mathcal{E}} u = u[\tilde{l}\sigma]_p$, $u[\tilde{r}\sigma]_p =_{\mathcal{E}} v$, and $E_0 \vdash \phi\sigma$.

- If the rule c belongs to R then $t \underset{c,u,p,\sigma}{\longrightarrow^1}_{R/\mathcal{E}} v$ using the same derivation that has only one rewrite step,
- else c belongs to $c_B \backslash R$, so there is another rule $c : l \rightarrow r$ *if* $\bigwedge_{i=1}^n l_i \rightarrow r_i \mid \phi$ in R such that, by definition of c_B, $\tilde{l} = w[l]_{\tilde{p}}$ and $\tilde{r} = w[r]_{\tilde{p}}$, where $w = w' \in B \cup B^{-1}$ and $\tilde{p} \in pos_\Sigma(w) - \{\epsilon\}$.
  Now, $t =_{\mathcal{E}} u = u[\tilde{l}\sigma]_p = u[w[l]_{\tilde{p}}\sigma]_p = u[w\sigma[l\sigma]_{\tilde{p}}]_p$. Then $u_{p.\tilde{p}} = l\sigma$, so $u = u[l\sigma]_{p.\tilde{p}}$. As $u[r\sigma]_{p.\tilde{p}} = u[w\sigma[r\sigma]_{\tilde{p}}]_p = u[w[r]_{\tilde{p}}\sigma]_p = u[\tilde{r}\sigma]_p =_{\mathcal{E}} v$, $l_i\sigma \rightarrow_{R/\mathcal{E}} r_i\sigma$ in zero rewrite steps, $1 \leq i \leq n$, and $E_0 \vdash \phi\sigma$, then $t \underset{c,u,p.\tilde{p},\sigma}{\longrightarrow^1}_{R/\mathcal{E}} v$ in one rewrite step.

(ii) zero rewrite steps: $t \rightarrow_{R_B/\mathcal{E}} v$ because $t =_{\mathcal{E}} v$. Then, also $t \rightarrow_{R/\mathcal{E}} v$.

Inductive step:

(i) $t \underset{c,u,p,\sigma}{\longrightarrow^1}_{R_B/\mathcal{E}} v$ in $n > 1$ rewrite steps, with a rule $c : \tilde{l} \rightarrow \tilde{r}$ *if* $\bigwedge_{i=1}^n l_i \rightarrow r_i \mid \phi$ in R_B. Then, $l_i\sigma \rightarrow_{R_B/\mathcal{E}} r_i\sigma$ with less than n rewrite steps, $1 \leq i \leq n$ so, by I.H, $l_i\sigma \rightarrow_{R/\mathcal{E}} r_i\sigma$, $1 \leq i \leq n$, using the same number of rewrite steps in each derivation. Now, using the same proof shown in the base case, we get $t \underset{c,u,p,\sigma}{\longrightarrow^1}_{R/\mathcal{E}} v$ if c in R, or else $t \underset{c,u,p.\tilde{p},\sigma}{\longrightarrow^1}_{R/\mathcal{E}} v$ using the same number of rewrite steps.

(ii) $t \to_{R_B/\mathcal{E}} v$ in $n > 0$ rewrite steps. We distinguish two cases:

- $t \to^1_{R_B/\mathcal{E}} w \to_{R_B/\mathcal{E}} v$. If the derivation $w \to_{R_B/\mathcal{E}} v_1$ has no rewrite steps, then $w =_{\mathcal{E}} v$, so $t \to^1_{R_B/\mathcal{E}} v$ and the proof in subcase (i) holds. Else, the derivations of both $t \to^1_{R_B/\mathcal{E}} w$ and $w \to_{R_B/\mathcal{E}} v$ have less than n rewrite steps so, by I.H., $t \to^1_{R/\mathcal{E}} w$ and $w \to_{R/\mathcal{E}} v$ with derivations using the same number of rewrite steps as the original ones, and then $t \to_{R/\mathcal{E}} v$ with a derivation that uses n rewrite steps.
- $t \xrightarrow[c,u,p,\sigma]{}{}^1_{R_B/\mathcal{E}} v$ in $n > 0$ rewrite steps. This case is exactly the same as the one in the subcases (i) of the base case and the inductive step, so the same proofs hold.

□

Theorem 15 (Equivalence of $R_B/\mathcal{E}$ and R_B, B-rewriting)**.** *If $\mathcal{R}_B = (\Sigma, \mathcal{E}, R_B)$ is an associated rewrite theory closed under B-extensions then $\to^1_{R_B,B} = \to^1_{R_B/\mathcal{E}}$ and $\to_{R_B,B} = \to_{R_B/\mathcal{E}}$.*

Proof. There is a special case to consider when there are no rewrite steps involved in the deductions.
(*i*) $\to^1_{R_B,B} \subseteq \to^1_{R_B/\mathcal{E}}$ and $\to_{R_B,B} \subseteq \to_{R_B/\mathcal{E}}$.

In the special case, $t \to_{R_B,B} v$ with no rewrite steps. As $\to_{R_B,B} = (\to^+_{R_B,B} \cup =_{\mathcal{E}})$ then $t =_{\mathcal{E}} v$, so $t \to_{R_B/\mathcal{E}} v$. The other cases are proved using induction on the total number of $\to^1_{R_B,B}$ rewrite steps in the derivation.

- Base case

  $t \to^1_{R_B,B} t[r\sigma]_p =_{\mathcal{E}} v$ with only one $\to^1_{R_B,B}$ rewrite step in the derivation, where $c : l \to r$ *if* $\bigwedge_{i=1}^m l_i \to r_i \mid \phi$ in R_B, $abstract_{\Sigma_1}(l) = \langle \lambda \bar{x}.l^\circ; \theta^\circ; \phi^\circ \rangle$, $\bar{x} = x_1, \ldots, x_n$, $l^\circ = l[\bar{x}]_{\bar{q}}$, $\phi^\circ = \bigwedge_{j=1}^n (x_j = l|_{q_j})$), p in $pos_{\Sigma_1}(t)$, and $\sigma : \bar{x} \cup vars(c) \to \mathcal{T}_\Sigma$ such that $rep(t|_p) =_B l^\circ\sigma$, $v =_{\mathcal{E}} t[r\sigma]_p$, $\bar{l}\sigma =_{\mathcal{E}} \bar{r}\sigma$, and $E_0 \vdash (\phi \wedge \phi^\circ)\sigma$.

  As $E_0 \vdash \phi^\circ\sigma$ then $l\sigma = l\sigma[l\sigma|_{q_1}]_{q_1} \cdots [\sigma_{q_n}]_{q_n} =_E l\sigma[x_1\sigma]_{q_1} \cdots [x_n\sigma]_{q_n} = l^\circ\sigma =_B rep(t|_p) =_{E_0} t|_p$, so $l\sigma =_{\mathcal{E}} t|_p$.

  As $t|_p =_{\mathcal{E}} l\sigma$ and $\bar{l}\sigma =_{\mathcal{E}} \bar{r}\sigma$, then $t = t[t|_p]_p =_{\mathcal{E}} t[l\sigma]_p \to^1_{R_B} t[r\sigma]_p =_{\mathcal{E}} v$ with rule c in R_B, that is, $t \to^1_{R_B/\mathcal{E}} v$, so $t \to_{R_B/\mathcal{E}} v$.

- Induction case

  There are two subcases to consider:

  1. $t \to^1_{R_B,B} t[r\sigma]_p =_{\mathcal{E}} v$ with several $\to^1_{R_B,B}$ rewrite steps in the derivation. As in the base case, $c : l \to r$ *if* $\bigwedge_{i=1}^m l_i \to r_i \mid \phi$ in R_B, $abstract_{\Sigma_1}(l) = \langle \lambda \bar{x}.l^\circ; \theta^\circ; \phi^\circ \rangle$, $\bar{x} = x_1, \ldots, x_n$, $l^\circ = l[\bar{x}]_{\bar{q}}$, $\phi^\circ = \bigwedge_{j=1}^n (x_j = l|_{q_j})$), p in $pos_{\Sigma_1}(t)$, and $\sigma : \bar{x} \cup vars(c) \to \mathcal{T}_\Sigma$ such that $rep(t|_p) =_B l^\circ\sigma$, $v =_{\mathcal{E}} t[r\sigma]_p$, $\bar{l}\sigma =_{\mathcal{E}} \bar{r}\sigma$, and $E_0 \vdash (\phi \wedge \phi^\circ)\sigma$.

     By induction hypothesis $l_i\sigma \to_{R_B/\mathcal{E}} r_i\sigma$, for $1 \leq i \leq m$. As in the base case, $E_0 \vdash \phi\sigma$ and $t|_p =_{\mathcal{E}} l\sigma$, so $t = t[t|_p]_p =_{\mathcal{E}} t[l\sigma]_p \to^1_{R_B} t[r\sigma]_p =_{\mathcal{E}} v$, i.e., $t \to^1_{R_B/\mathcal{E}} v$, so $t \to_{R_B/\mathcal{E}} v$.
  2. $t \to^1_{R_B,B} u \to^+_{R_B,B} w =_{\mathcal{E}} v$. By the previous subcase $t \to^1_{R_B/\mathcal{E}} u \to^+_{R_B,B} w =_{\mathcal{E}} v$, and, by I.H., $t \to^1_{R_B/\mathcal{E}} u \to^+_{R_B/\mathcal{E}} w =_E v$, i.e., $t \to^*_{R_B/\mathcal{E}} w =_{\mathcal{E}} v$, or $t \to_{R_B/\mathcal{E}} v$.

(ii) $\to^1_{R_B/\mathcal{E}} \subseteq \to^1_{R_B,B}$ and $\to_{R_B/\mathcal{E}} \subseteq \to_{R_B,B}$.

In the special case, $t \to_{R_B/\mathcal{E}} v$ with no rewrite steps because $t =_\mathcal{E} v$, as $\to_{R_B,B} = (\to^+_{R_B,B} \cup =_E)$ then $t \to_{R_B,B} v$. The other cases are proved using induction in the total number of $\to^1_{R_B/\mathcal{E}}$ rewrite steps in the derivation.

- Base case: $t \to^1_{R_B/\mathcal{E}} v$ with only one $\to^1_{R_B/\mathcal{E}}$ rewrite step in the derivation using a rule $c : l \to r \; if \; C$ in R_B, where $C = \bigwedge_{i=1}^m l_i \to r_i \mid \phi$, and a substitution σ. We can assume that c is a rule in R_0 since any $\to^1_{R_B/\mathcal{E}}$ step given at position p of t'' using a rule $c_1 : w[l]_q \to w[r]_q \; if \; C$ in $R \setminus R_0$ can also be achieved using rule c at position $p.q$ of t'', so $t =_\mathcal{E} t'' \to^1_{R_B} u =_\mathcal{E} v$, $t'' = t''[l\sigma]_p$, $u = t''[r\sigma]_p$, $\bar{l}\sigma =_\mathcal{E} \bar{r}\sigma$, and $E_0 \vdash \phi\sigma$. By Proposition 10 there exists a term t' in $\mathcal{H}_\Sigma$ such that $t =_B t' =_{E_0} t'' \to^1_{R_B} u =_\mathcal{E} v$. We have $t \overset{ax_1}{\longleftrightarrow}_B \cdots \overset{ax_l}{\longleftrightarrow}_B t'$, where ax_i (linear and regular), for $1 \leq i \leq l$ has the form $w_i = w'_i$, let $\overline{ax_i}$ be $w'_i = w_i$, so each top_{Σ_0} subterm of t is moved by $ax_1 \cdots ax_l$ and becomes another top_{Σ_0} subterm of t'. Then, $\overline{ax_l} \cdots \overline{ax_1}$ moves the top_{Σ_0} subterms of t'' in the opposite way, so there exists a term t_0 in $\mathcal{T}_\Sigma$ such that $t'' \overset{\overline{ax_l}}{\longleftrightarrow}_B \cdots \overset{\overline{ax_1}}{\longleftrightarrow}_B t_0 =_{E_0} t$.

  We have $t =_{E_0} t_0 =_B t'' = t''[l\sigma]_p$, so $t''|_p = l\sigma$. The more general case, where $t_0 =_B t''|_p =_B l\sigma$ is studied in Theorem 2 and Corollary 2 in [Mes17], where it is proved that there is a position q in $pos(t_0)$, a rule $c_0 : l_0 \to r_0 \; if \; C$ in R_B, maybe the original c, and a substitution σ_0, such that $t_0|_q =_B l_0\sigma_0$, $t_0[r_0\sigma_0]_q =_B u$, and $C\sigma_0 = C\sigma$, which is also valid for our particular case where $t''|_p = l\sigma$. As, by definition of rule, $l_0 \in \mathcal{H}_\Sigma(\mathcal{X})$, then $q \in pos_{\Sigma_1}(t_0)$, so $t_0|_q =_{E_0} t|_q$. Let $top_{\Sigma_0}(t|_q) = \hat{z}$. Then $rep_{t|_{q.\hat{z}}}$ is the function that given a term in $\mathcal{T}_\Sigma$ returns the same term with each top_{Σ_0} term on it replaced with the representative for that top_{Σ_0} term in $rep(t|_q)$, if it exists, so $rep(t|_q) = rep_{t|_{q.\hat{z}}}(t_0|_q) =_B rep_{t|_{q.\hat{z}}}(l_0\sigma_0)$.

  Let $abstract_{\Sigma_1}(l_0) = \langle \lambda\bar{y}.l_0^\circ; \theta_0^\circ; \phi_0^\circ \rangle$, $\bar{y} = y_1, \ldots, y_k$, $l_0^\circ = l_0[\bar{y}]_{\bar{o}}$, $\phi^\circ = \bigwedge_{j=1}^k y_j = l_0|_{o_j}$. Define $\sigma' : dom(\sigma_0) \cup \hat{y} \to \mathcal{T}_\Sigma$ as: if $z = y_j \in \hat{y}$ then $z\sigma' = rep_{t|_{q.\hat{z}}}(l_0|_{o_j}\sigma_0)$ else $z\sigma' = rep_{t|_{q.\hat{z}}}(z\sigma_0)(=_{E_0} z\sigma_0)$. As, for $1 \leq j \leq k$, $y_j\sigma' = rep_{t|_{q.\hat{z}}}(l_0|_{o_j}\sigma_0) =_{E_0} l_0|_{o_j}\sigma_0 =_{E_0} l_0|_{o_j}\sigma'$, because $\hat{y} \cap V_{l_0|_{o_j}} = \emptyset$, then $E_0 \vdash \phi^\circ\sigma'$. Also, as $C\sigma_0 = C\sigma$ and if $z \in dom(\sigma_0)$ then $z\sigma' =_{E_0} z\sigma_0$ then $\bar{l}\sigma' =_{E_0} \bar{l}\sigma_0 =_E \bar{r}\sigma_0 =_{E_0} \bar{r}\sigma'$, i.e., $\bar{l}\sigma' =_\mathcal{E} \bar{r}\sigma'$, and $\phi\sigma' =_{E_0} \phi\sigma_0 = \phi\sigma$, so $E_0 \vdash \phi\sigma'$. As $\phi\sigma'$ and $\phi^\circ\sigma'$ are ground, because $rep_{t|_{q.\hat{z}}}$ is replacing each ground subterm with another ground subterm, then $E_0 \vdash (\phi \wedge \phi^\circ)\sigma'$.

  As

  - $l_0[]_{\bar{o}}\sigma' = l_0\sigma'[]_{\bar{o}} = rep_{t|_{q.\hat{z}}}(l_0\sigma_0[]_{\bar{o}})$, and
  - $y_j\sigma' = rep_{t|_{q.\hat{z}}}(l_0|_{o_j}\sigma_0)$, for $1 \leq j \leq k$,

  then $l_0^\circ\sigma' = l_0[\bar{y}]_{\bar{o}}\sigma' = rep_{t|_{q.\hat{z}}}(l_0\sigma_0[l_0|_{\bar{o}}\sigma_0]_{\bar{o}}) = rep_{t|_{q.\hat{z}}}(l_0[l_0|_{\bar{o}}]_{\bar{o}}\sigma_0) = rep_{t|_{q.\hat{z}}}(l_0\sigma_0) =_B rep(t|_q)$, i.e., $rep(t|_q) =_B l_0^\circ\sigma'$ so, as $t[r_0\sigma']_q =_{E_0} t[r_0\sigma_0]_q =_{E_0} t_0[r_0\sigma_0]_q =_B u =_\mathcal{E} v$, i.e., $t[r_0\sigma']_q =_\mathcal{E} v$, we have $t \to^1_{R_B,B} v$.

- Induction case:

  again, there are two subcases to consider:

  1. $t \to^1_{R_B/\mathcal{E}} t[r\sigma]_p =_\mathcal{E} v$ with several $\to^1_{R_B/\mathcal{E}}$ rewrite steps in the derivation. The proof is the same as the one in the base case, except that instead of having $\bar{l}\sigma' =_\mathcal{E} \bar{r}\sigma'$ now we have $l_i\sigma \to_{R_B/\mathcal{E}} r_i\sigma$, for $1 \leq i \leq m$, so by I.H., as $(l_i, r_i)\sigma \to_{R_B/\mathcal{E}} (l_i, r_i)\sigma'$, also $l_i\sigma' \to_{R_B,B} r_i\sigma'$ hence $t \to^1_{R_B,B} v$.

2. $t \to^1_{R_B/\mathcal{E}} u \to^+_{R_B/\mathcal{E}} w =_\mathcal{E} v$. By the previous subcase $t \to^1_{R_B,B} u \to^+_{R_B/\mathcal{E}} w =_\mathcal{E} v$, and, by I.H., $t \to^1_{R_B,B} u \to^+_{R_B,B} w =_\mathcal{E} v$, i.e., $t \to^*_{R_B,B} w =_\mathcal{E} v$, or $t \to_{R_B,B} v$.

□

Proposition 18 (Decomposition of a normalized substitution)**.** *Let $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ be a rewrite theory with built-in subtheory (Σ_0, E_0). If σ is an $R/\mathcal{E}$-normalized substitution and $\sigma = \sigma_1 \cdot \sigma_2$, with $dom(\sigma_1) \cap (ran(\sigma_1) \cup dom(\sigma_2)) = \emptyset$, then σ_1 and σ_2 are $R/\mathcal{E}$-normalized.*

Proof. We prove that each substitution is normalized by reductio ad absurdum:

- If σ_1 is not $R/\mathcal{E}$-normalized, then there exists a variable x in $dom(\sigma_1) \subseteq dom(\sigma)$ and a term t such that $x\sigma_1$ is in $\mathcal{H}_\Sigma$, so $x\sigma_1 = x\sigma_1\sigma_2 = x\sigma$, and $x\sigma_1 \to^1_{R/\mathcal{E}} t$. As $x\sigma_1 = x\sigma$, then also $x\sigma \to^1_{R/\mathcal{E}} t$ hence, as x is in $dom(\sigma)$, σ is not $R/\mathcal{E}$-normalized, a contradiction.
- If σ_2 is not $R/\mathcal{E}$-normalized, then there exists a variable x in $dom(\sigma_2)$ and a term t such that $x\sigma_2$ is in $\mathcal{H}_\Sigma$ and $x\sigma_2 \to^1_{R/\mathcal{E}} t$, where either x in $dom(\sigma)$ or not.
  - If x is in $dom(\sigma)$ then $x\sigma_2 = x\sigma$, so also $x\sigma \to^1_{R/\mathcal{E}} t$ hence, as x is in $dom(\sigma)$, σ is not $R/\mathcal{E}$-normalized, a contradiction.
  - If x is not in $dom(\sigma)$ then, as $\sigma = \sigma_1\sigma_2$, x is in $ran(\sigma_1)$, so there exists y in $dom(\sigma_1) \subseteq dom(\sigma)$ and a position p such that $y\sigma_1|_p = x$. Then $y\sigma|_p = y\sigma_1\sigma_2|_p = y\sigma_1|_p\sigma_2 = x\sigma_2$, so $y\sigma|_p \to^1_{R/\mathcal{E}} t$, hence also $y\sigma \to^1_{R/\mathcal{E}} t$. As y is in $dom(\sigma)$, then σ is not $R/\mathcal{E}$-normalized, a contradiction.

□

Proposition 19 (Preservation of the normalized property under generalization)**.** *Let $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ be a rewrite theory with built-in subtheory (Σ_0, E_0). If ρ is an $R/\mathcal{E}$-normalized substitution and σ is a more general substitution than ρ, then σ is $R/\mathcal{E}$-normalized.*

Proof. We proceed again by reductio ad absurdum. By definition of $\ll_\mathcal{E}$, there exisst a substitution η such that $\rho_\mathcal{V} =_\mathcal{E} (\sigma\eta)_\mathcal{V}$. If σ is not $R/\mathcal{E}$-normalized, then there exist a variable x in $dom(\sigma) \subseteq dom(\rho)$ and a term t such that $x\sigma$ is in $\mathcal{H}_\Sigma$, so $x\sigma = x\sigma\eta =_\mathcal{E} x\rho$, and $x\sigma \to^1_{R/\mathcal{E}} t$. But then, also $x\rho \to^1_{R/\mathcal{E}} t$ so, as x is in $dom(\rho)$, ρ is not $R/\mathcal{E}$-normalized, a contradiction. □

Lemma 12 (Interpretation of the semantics)**.** *Given a rewrite theory $\mathcal{R} = (\Sigma, E_0 \cup B, R)$, a set of call strategy definitions $Call_\mathcal{R}$, and terms $t, v \in \mathcal{H}_\Sigma$, for each c.p.t. T formed using the rules in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$ with head $t \to v/ST$, so $[v]_\mathcal{E} \in ST@[t]_\mathcal{E}$, each renaming α such that $ran(\alpha) \cap (V_T \cup V_{\mathcal{R},Call_\mathcal{R}}) = \emptyset$, and each strategy $ST' =_\mathcal{E} ST$, it holds that:*

1. *Main property: $t \to_{R/\mathcal{E}} v$ and there exist closed proof trees for $[v]_\mathcal{E} \in ST\alpha@[t]_\mathcal{E}$ and $[v]_\mathcal{E} \in ST'@[t]_\mathcal{E}$ with the same depth and number of nodes as T.*
2. *If $ST =$ `idle` then $[t]_\mathcal{E} = [v]_\mathcal{E}$.*
3. *If $ST = c[\gamma]$ then $t \underset{c\gamma}{\longrightarrow}^1_{R/\mathcal{E}} v$.*

4. *If* $ST = \texttt{top}(c[\gamma])$, *then* $t \underset{c\gamma,\epsilon\ R/\mathcal{E}}{\longrightarrow^1} v$ *(i.e., the rewrite happens at the top position of* t*).*

5. *If* $ST = \texttt{match}\ u\ \texttt{s.t.}\ \bigwedge_{j=1}^{m}(l_j = r_j) \wedge \phi$ *then* $[t]_{\mathcal{E}} = [v]_{\mathcal{E}}$ *and there exists a substitution* σ *such that* $t =_{\mathcal{E}} u\sigma$, $l_j\sigma =_{\mathcal{E}} r_j\sigma$, *for* $1 \leq j \leq m$, *and* $E_0 \vdash \phi\sigma$.

6. *If* $ST = ST_1\,;ST_2$ *then there exists a term* $u \in \mathcal{H}_\Sigma$ *such that* $[u]_{\mathcal{E}} \in ST_1@[t]_{\mathcal{E}}$ *and* $[v]_{\mathcal{E}} \in ST_2@[u]_{\mathcal{E}}$.

7. *If* $ST = ST_1+$ *then there exist* $i+1$ *terms* $u_0 = t, u_1, \ldots, u_{i-1}, u_i = v \in \mathcal{H}_\Sigma$, *with* $i > 0$, *such that* $[u_j]_{\mathcal{E}} \in ST_1@[u_{j-1}]_{\mathcal{E}}$, *for* $1 \leq j \leq i$, *where* i *is equal to one plus the number of times that a rule with the form* $\frac{w_1 \to w_2/ST_1\,;ST_1+}{w_1 \to w_2/ST_1+}$, *followed by the application of a rule with the form* $\frac{\overline{w_1 \to w'/ST_1}\quad \overline{w' \to w_2/ST_1+}}{w_1 \to w_2/ST_1\,;ST_1+}$, *is applied in the rightmost branch of the subtree before applying a rule with the form* $\frac{w_1 \to w_2/ST_1}{w_1 \to w_2/ST_1+}$.

8. *If* $ST = ST_1 \mid ST_2$ *then* $[v]_{\mathcal{E}} \in ST_1@[t]_{\mathcal{E}}$ *or* $[v]_{\mathcal{E}} \in ST_2@[t]_{\mathcal{E}}$.

9. *If* $ST = \texttt{match}\ u\ \texttt{s.t.}\ \phi\,?\,ST_1 : ST_2$ *then there exists a substitution* δ *such that* $t =_{\mathcal{E}} u\delta$ *and either* $E_0 \vdash \phi\delta$ *and* $[v]_{\mathcal{E}} \in ST_1\delta@[t]_{\mathcal{E}}$ *or* $E_0 \vdash \neg\phi\delta$ *and* $[v]_{\mathcal{E}} \in ST_2\delta@[t]_{\mathcal{E}}$.

10. *If* $ST = CS$, *where* $\texttt{sd}\ CS := ST_1 \in Call_{\mathcal{R}}$, *then: (i)* $[v]_{\mathcal{E}} \in ST_1@[t]_{\mathcal{E}}$, *and (ii)* $[v]_{\mathcal{E}} \in ST_1\gamma@[t]_{\mathcal{E}}$, *for every renaming* γ *such that* $dom(\gamma) \subseteq vars(ST_1) \setminus V_{\mathcal{R}}$ *and* $ran(\gamma) \cap V_{\mathcal{R},Call_{\mathcal{R}}} = \emptyset$.

11. *If* $ST = CS(\bar{t})$, *where* $\texttt{sd}\ CS(\bar{x}) := ST_1 \in Call_{\mathcal{R}}$, $\bar{x} = x^1_{s_1}, \ldots, x^n_{s_n}$, $\bar{t} = t_1, \ldots, t_n$, *and* $\rho = \{\bar{x} \mapsto \bar{t}\}$, *then: (i)* $[v]_{\mathcal{E}} \in ST_1\rho@[t]_{\mathcal{E}}$ *and (ii) if* γ *is a renaming such that* $dom(\gamma) \subseteq vars(ST_1) \setminus \hat{x}$ *and* $ran(\gamma) \cap (ran(\rho) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$ *(so* $\frac{t \to v/ST_1(\gamma \cup \rho)}{t \to v/CS(\bar{t})} \in \mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$*), then* $[v]_{\mathcal{E}} \in ST_1(\gamma \cup \rho)@[t]_{\mathcal{E}}$.

12. *If* $ST = CS(\bar{t})$, *where* $\texttt{csd}\ CS(\bar{x}) := ST_1$ *if* $C \in Call_{\mathcal{R}}$, *with* $\bar{x} = x^1_{s_1}, \ldots, x^n_{s_n}$ *and* $C = \bigwedge_{j=1}^{m}(l_j = r_j) \wedge \phi$, $V_{CS} = vars(ST_1) \cup vars(C)$, $\hat{x} \subseteq V_{CS}$, $\bar{t} = t_1, \ldots, t_n$, *and* $\rho = \{\bar{x} \mapsto \bar{t}\}$, *then (i) there exists a substitution* $\delta_1 : vars(C\rho) \to \mathcal{T}_\Sigma$, *such that* $l_j\rho\delta_1 =_{\mathcal{E}} r_j\rho\delta_1$, *for* $1 \leq j \leq n$, $E_0 \vdash \phi\rho\delta_1$ *(so* $\frac{t \to v/ST_1\rho\delta_1}{t \to v/CS(\bar{t})} \in \mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$*), and* $[v]_{\mathcal{E}} \in ST_1\rho\delta_1@[t]_{\mathcal{E}}$, *and (ii) for every renaming* γ *such that* $dom(\gamma) \subseteq V_{CS} \setminus \hat{x}$ *and* $ran(\gamma) \cap (ran(\rho) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$, *there exists a substitution* $\delta_2 : vars(C(\gamma \cup \rho)) \to \mathcal{T}_\Sigma$, *such that* $l_j(\gamma \cup \rho)\delta_2 =_{\mathcal{E}} r_j(\gamma \cup \rho)\delta_2$, *for* $1 \leq j \leq n$, $E_0 \vdash \phi(\gamma \cup \rho)\delta_2$ *(so* $\frac{t \to v/ST_1(\gamma \cup \rho)\delta_2}{t \to v/CS(\bar{t})} \in \mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$*), and* $[v]_{\mathcal{E}} \in ST_1(\gamma \cup \rho)\delta_2@[t]_{\mathcal{E}}$.

13. *If* $ST = c[\gamma]\{ST_1, \ldots, ST_m\}$, *with* $c : l \to r$ *if* $\bigwedge_{j=1}^{m} l_j \to r_j \mid \psi$ *a rule in* R, *then there is a substitution* δ *such that* $[r_i\gamma\delta]_{\mathcal{E}} \in ST_i\delta\ @\ [l_i\gamma\delta]_{\mathcal{E}}$, *for* $1 \leq i \leq m$, *and* $t \underset{c,\gamma\delta\ R/\mathcal{E}}{\longrightarrow^1} v$.

14. *If* $ST = \texttt{top}(c[\gamma]\{ST_1, \ldots, ST_m\})$, *with* $c : l \to r$ *if* $\bigwedge_{j=1}^{m} l_j \to r_j \mid \psi$ *a rule in* R *then there is a substitution* δ *such that* $[r_i\gamma\delta]_{\mathcal{E}} \in ST_i\delta\ @\ [l_i\gamma\delta]_{\mathcal{E}}$, *for* $1 \leq i \leq m$, *and* $t \underset{c,\epsilon,\gamma\delta R/\mathcal{E}}{\longrightarrow^1} v$.

15. *If* $ST = \texttt{matchrew}\ u\ \texttt{s.t.}\ \bigwedge_{j=1}^{m}(l_j = r_j) \wedge \phi\ \texttt{by}\ x^1_{s_1}\ \texttt{using}\ ST_1, \ldots,\ x^n_{s_n}\ \texttt{using}\ ST_n$, *where* $u = u[x^1_{s_1}, \ldots, x^n_{s_n}]_{p_1 \ldots p_n}$ *then there exist a substitution* δ, *where* $\delta_{V_{u,\phi,\bar{l},\bar{r}}}$ *is*

ground, and terms $t_1, \ldots, t_n \in \mathcal{H}_\Sigma$ *such that* $t =_\mathcal{E} u\delta$, $l_j\delta =_\mathcal{E} r_j\delta$, *for* $1 \leq j \leq m$, $E_0 \vdash \phi\delta$, $[t_i]_\mathcal{E} \in ST_i\delta \,@\, [x^i_{s_i}\delta]_\mathcal{E}$, *for* $1 \leq i \leq n$, *and* $v =_\mathcal{E} u\delta[t_1, \ldots, t_n]_{p_1 \ldots p_n}$.

Proof. The proof for the first property is done by induction on the depth of the c.p.t. T for $t \to v/ST$. The rest of the properties are proved when the related strategy is treated in the proof for the first property. As $ran(\alpha) \cap vars(ST) = \emptyset$ then $vars(ST) \cap dom(\alpha^{-1}) = \emptyset$, so $ST\alpha\alpha^{-1} = ST$.

- There are five strategies in the base case: `fail`, `idle`, $c[\gamma]$, `top`$(c[\gamma])$, and the `match` test. The depth and number of nodes of all the closed proof trees is one in this case.

  1. As there are no derivation rules for `fail`, there is nothing to prove in this case.
  2. If $[v]_\mathcal{E} \in$ `idle`$@[t]_\mathcal{E} = \{[t]_\mathcal{E}\}$ then, as shown in example 5, $[v]_\mathcal{E} = [t]_\mathcal{E}$ (**property 2**), so $v =_\mathcal{E} t$ and, by definition, $t \to_{R/\mathcal{E}} v$. As `idle` $\alpha =$ `idle` then also $[v]_\mathcal{E} \in$ `idle` $\alpha@[t]_\mathcal{E}$ using the original c.p.t. T. As only `idle` $=_\mathcal{E}$ `idle`, there is nothing to prove about the strategies that are equal modulo E to `idle`.
  3. If $[v]_\mathcal{E} \in c[\gamma]@[t]_\mathcal{E}$, with $c : l \to r \; if \; \phi$, then $\frac{}{t \to v/c[\gamma]}$ must come from a derivation rule $\frac{}{t' \to v'/c[\gamma]}$ in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$, where $t' \xrightarrow[c,p,\gamma\delta]{1}_R v'$ for appropriate p and δ such that $t =_\mathcal{E} t' = t'[l\gamma\delta]_p$, $v =_\mathcal{E} v' = t'[r\gamma\delta]_p$, and $E_0 \vdash \phi\gamma\delta$, so $t \xrightarrow[c,p,\gamma\delta]{1}_{R/\mathcal{E}} v$ (**property 3**).
     $c[\gamma]\alpha = c[(\gamma\alpha)_{dom(\gamma)}]$, let $\beta = (\gamma\alpha)_{dom(\gamma)}$ and let $\delta' = \alpha^{-1}\delta$. As $ran(\alpha) \cap (V_T \cup V_{\mathcal{R},Call_\mathcal{R}}) = \emptyset$ then $c\beta\delta' = c(\gamma\alpha)_{dom(\gamma)}\alpha^{-1}\delta = c\gamma\delta$, so also $t \xrightarrow[c,\beta\delta']{1}_{R/\mathcal{E}} v$, and there is a derivation rule $\frac{}{t' \to v'/c[\beta]} \in \mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$, so $\frac{}{t \to v/c[\gamma]\alpha}$ is a c.p.t. for $[v]_\mathcal{E} \in c[\gamma]\alpha@[t]_\mathcal{E}$ because $t =_\mathcal{E} t'$, $v =_\mathcal{E} v'$, and $c[\gamma]\alpha = c[\beta]$.
     As $ST = c[\gamma] =_\mathcal{E} ST'$, then $ST' = c[\gamma']$ where $\gamma =_\mathcal{E} \gamma'$, so $(l, r, \phi)\gamma =_\mathcal{E} (l, r, \phi)\gamma'$, with $V_{l\gamma} = V_{l\gamma'}$ and $V_{r\gamma} = V_{r\gamma'}$, hence $E_0 \vdash \phi\gamma'\delta$, $t =_\mathcal{E} t'[l\gamma\delta]_p =_\mathcal{E} t'[l\gamma'\delta]_p$ and $v =_\mathcal{E} t'[r\gamma\delta]_p =_\mathcal{E} t'[r\gamma'\delta]_p$, ground terms, and $t'[l\gamma'\delta]_p \xrightarrow[c,p,\gamma'\delta]{1}_R t'[r\gamma'\delta]_p$. Then, there is a derivation rule $\frac{}{t'[l\gamma'\delta]_p \to t'[r\gamma'\delta]_p/c[\gamma']}$ in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$, so $\frac{}{t \to v/c[\gamma']}$ is a c.p.t. for $[v]_\mathcal{E} \in c[\gamma']@[t]_\mathcal{E}$.
  4. If $[v]_\mathcal{E} \in$ `top`$(c[\gamma])@[t]_\mathcal{E}$, where $c : l \to r$ is a rule in R, then $T = \frac{}{t \to v/\texttt{top}(c[\gamma])}$ must come from a derivation rule $\frac{}{l\gamma\delta \to r\gamma\delta/\texttt{top}(c[\gamma])} \in \mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$, meaning that $l\gamma\delta \xrightarrow[c,\epsilon,\gamma\delta]{1}_R r\gamma\delta$, such that $l\gamma\delta =_\mathcal{E} t$ and $r\gamma\delta =_\mathcal{E} v$, so $t \xrightarrow[c,\epsilon,\gamma\delta]{1}_{R/\mathcal{E}} v$ (**property 4**). Let $\beta = (\gamma\alpha)_{dom(\gamma)}$. As in the previous case, `top`$(c[\gamma])\alpha =$ `top`$(c[\gamma]\alpha) =$ `top`$(c[\beta])$. If we take $\delta' = \alpha^{-1}\delta$, then $c\beta\delta' = c\gamma\delta$ so also $l\gamma\delta \xrightarrow[c,\epsilon,\beta\delta']{1}_R r\gamma\delta$ and $\frac{}{l\gamma\delta \to r\gamma\delta/\texttt{top}(c[\beta])} \in \mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$, so $\frac{}{t \to v/\texttt{top}(c[\gamma])\alpha}$ is a c.p.t. for $[v]_\mathcal{E} \in$ `top`$(c[\gamma])\alpha@[t]_\mathcal{E}$, because $l\gamma\delta =_\mathcal{E} t$, $r\gamma\delta =_\mathcal{E} v$, and `top`$(c[\gamma])\alpha =$ `top`$(c[\beta])$.
     As $ST =$ `top`$(c[\gamma]) =_\mathcal{E} ST'$ then $ST' =$ `top`$(c[\gamma'])$ where $\gamma =_\mathcal{E} \gamma'$, so $(l, r, \phi)\gamma =_\mathcal{E} (l, r, \phi)\gamma'$, with $V_{l\gamma} = V_{l\gamma'}$ and $V_{r\gamma} = V_{r\gamma'}$, hence $E_0 \vdash \phi\gamma'\delta$, $t =_\mathcal{E} l\gamma\delta =_\mathcal{E} l\gamma'\delta$ and $v =_\mathcal{E} r\gamma\delta =_\mathcal{E} r\gamma'\delta$, ground terms, and $l\gamma'\delta \xrightarrow[c,\epsilon,\gamma'\delta]{1}_R r\gamma'\delta$. Then, there is a derivation rule $\frac{}{t'[l\gamma'\delta]_p \to t'[r\gamma'\delta]_p/\texttt{top}(c[\gamma'])}$ in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$, so $\frac{}{t \to v/\texttt{top}(c[\gamma'])}$ is a c.p.t. for $[v]_\mathcal{E} \in$ `top`$(c[\gamma'])@[t]_\mathcal{E}$.
  5. If $ST =$ `match` u `s.t.` $\bigwedge_{j=1}^m (l_j = r_j) \wedge \phi$ and $[v]_\mathcal{E} \in ST@[t]_\mathcal{E}$, then $T = \frac{}{t \to v/ST}$ must come from a rule $\frac{}{w \to w/ST}$ in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$ such that $t =_\mathcal{E} w$ and $v =_\mathcal{E} w$, so

$t =_{\mathcal{E}} v$ (i.e. $[t]_{\mathcal{E}} = [v]_{\mathcal{E}}$), and there exists a substitution σ such that $w =_{\mathcal{E}} u\sigma$, so $t =_{\mathcal{E}} u\sigma$, $l_j\sigma =_{\mathcal{E}} r_j\sigma$, for $1 \leq j \leq m$, and $E_0 \vdash \phi\sigma$ (**property 5**). As $t =_{\mathcal{E}} v$ then, by definition, $t \to_{R/\mathcal{E}} v$.

As $ST\alpha = \texttt{match}\, u\alpha\, \texttt{s.t.}\, \bigwedge_{j=1}^{m}(l_j\alpha = r_j\alpha) \wedge \phi\alpha$, if we take $\sigma' = \alpha^{-1}\sigma$ then, trivially, $w =_{\mathcal{E}} u\alpha\sigma'$, so $t =_{\mathcal{E}} u\alpha\sigma'$, $l_j\alpha\sigma' =_{\mathcal{E}} r_j\alpha\sigma'$, for $1 \leq j \leq m$, and $E_0 \vdash \phi\alpha\sigma'$, so there is a rule $\frac{}{w \to w/ST\alpha} \in \mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$, hence $\frac{}{t \to v/ST\alpha}$ is a c.p.t. for $[v]_{\mathcal{E}} \in ST\alpha@[t]_{\mathcal{E}}$.

- Inductive step:

6. $ST = ST_1\,;ST_2$.

   If $[v]_{\mathcal{E}} \in ST@[t]_{\mathcal{E}}$ then $T = \frac{T_1 \quad T_2}{t \to v/ST_1\,;ST_2}$ comes from a rule $\frac{t \to u/ST_1 \quad u \to v/ST_2}{t \to v/ST_1\,;ST_2}$, where T_1 and T_2 are closed proof trees with head $t \to u/ST_1$ and $u \to v/ST_2$, respectively, so $[u]_{\mathcal{E}} \in ST_1@[t]_{\mathcal{E}}$ and $[v]_{\mathcal{E}} \in ST_2@[u]_{\mathcal{E}}$ (**property 6**). As these closed proof trees are of a smaller depth then, by I.H. and property 1, $t \to_{R/\mathcal{E}} u$ and $u \to_{R/\mathcal{E}} v$, so $t \to_{R/\mathcal{E}} v$.

   As $ST\alpha = ST_1\alpha;ST_2\alpha$, we can apply the I.H. to T_1 and T_2, so there are closed proof trees T_1' and T_2' with head $t \to u/ST_1\alpha$ and $u \to v/ST_2\alpha$, respectively. As there is a rule $\frac{t \to u/ST_1\alpha \quad u \to v/ST_2\alpha}{t \to v/ST\alpha} \in \mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$ then $\frac{T_1' \quad T_2'}{t \to v/ST\alpha}$ is a c.p.t. for $[v]_{\mathcal{E}} \in ST\alpha@[t]_{\mathcal{E}}$.

   As $ST =_{\mathcal{E}} ST'$, then $ST' = ST_1'\,;ST_2'$ where $ST_1 =_{\mathcal{E}} ST_1'$ and $ST_2 =_{\mathcal{E}} ST_2'$. As T_1 and T_2 are of a smaller depth than T then, by I.H., there are closed proof trees T_1' and T_2' for $[u]_{\mathcal{E}} \in ST_1'@[t]_{\mathcal{E}}$ and $[v]_{\mathcal{E}} \in ST_2'@[u]_{\mathcal{E}}$, with the same depth and number of nodes as T_1 and T_2, respectively, and $\frac{T_1' \quad T_2'}{t \to v/ST'}$ is a c.p.t. for $[v]_{\mathcal{E}} \in ST'@[t]_{\mathcal{E}}$ with the same depth and number of nodes as T.

7. $ST = ST_1+$.

   T must be either of the form $\frac{T_1}{t \to v/ST_1+}$ or $\frac{T_2}{t \to v/ST_1+}$, where T_1 has head $t \to v/ST_1$ or T_2 has head $t \to v/ST_1\,;ST_1+$.

   In the first case, $i = 1$ because no rule with the form $\frac{w_1 \to w_2/ST_1\,;ST_1+}{w_1 \to w_2/ST_1+}$ has been applied, and there are 2 terms, u_0 (we take t) and u_1 (we take v), in $\mathcal{H}_\Sigma$ such that $u_0 = t$, $u_1 = v$, and $[u_1]_{\mathcal{E}} \in ST_1@[u_0]_{\mathcal{E}}$, because we have a c.p.t. for $t \to v/ST_1$.

   In the second case, we can apply I.H. to the c.p.t. for $u_1 \to v/ST_1+$ so there are i terms $w_0 = u_1, \ldots, w_{i-2}, w_{i-1} = v$ such that $[w_j]_{\mathcal{E}} \in ST_1@[w_{j-1}]_{\mathcal{E}}$, for $1 \leq j \leq i-1$. As there is a c.p.t. for $t \to u_1/ST_1$ in the left branch, then also $[u_1]_{\mathcal{E}} \in ST_1@[t]_{\mathcal{E}}$. Taking $u_0 = t$ and $u_{j+1} = w_j$ for $1 \leq j \leq i-1$ we get $u_0 = t$, $u_i = w_{i-1} = v$, and $[u_j]_{\mathcal{E}} \in ST_1@[u_{j-1}]_{\mathcal{E}}$, for $1 \leq j \leq i$ (**property 7**).

   In either case we also have a c.p.t. of a smaller depth whose head has the form $t \to v/\ldots$ so, by I.H., $t \to_{R/\mathcal{E}} v$. Also by I.H., we have either a c.p.t. T_1' with head $= t \to v/ST_1\alpha$ or T_2' with head $t \to v/ST_1\alpha\,;ST_1\alpha+$ with depth equal, whichever the case, to $depth(T)-1$. As $ST_1+\alpha = ST_1\alpha+$ and there are rules $\frac{t \to v/ST_1\alpha}{t \to v/ST_1\alpha+}$ and $\frac{t \to v/ST_1\alpha\,;ST_1\alpha+}{t \to v/ST_1\alpha+}$ in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$ then either $\frac{T_1'}{t \to v/ST_1\alpha+}$ or $\frac{T_2'}{t \to v/ST_1\alpha+}$ is a c.p.t. for $[v]_{\mathcal{E}} \in ST\alpha@[t]_{\mathcal{E}}$ with the same depth and number of nodes as T.

   As $ST =_{\mathcal{E}} ST'$, then $ST' = ST_1'+$ where $ST_1 =_{\mathcal{E}} ST_1'$. As T_j, where j in $\{1,2\}$, has smaller depth than T then, by I.H., there is a c.p.t. T' for

$[v]_{\mathcal{E}} \in ST'_1@[t]_{\mathcal{E}}$ or $[v]_{\mathcal{E}} \in ST'_1 ; ST_1 + @[t]_{\mathcal{E}}$ with the same depth and number of nodes as T_j, and $\frac{T'}{t \to v/ST'}$ is a c.p.t. for $[v]_{\mathcal{E}} \in ST'@[t]_{\mathcal{E}}$ with the same depth and number of nodes as T.

8. $ST = ST_1 \mid ST_2$.

   T must be either of the form $\frac{T_1}{t \to v/ST_1 \mid ST_2}$ or $\frac{T_2}{t \to v/ST_1 \mid ST_2}$, where T_1 has head $t \to v/ST_1$ or T_2 has head $t \to v/ST_2$, so either $[v]_{\mathcal{E}} \in ST_1@[t]_{\mathcal{E}}$ or $[v]_{\mathcal{E}} \in ST_2@[t]_{\mathcal{E}}$ must hold (**property 8**) and, by I.H., $t \to_{R/\mathcal{E}} v$. Also by I.H. there is a c.p.t T'_1, with head $t \to v/ST_1\alpha$, or T'_2, with head $t \to v/ST_2\alpha$ with depth equal, whichever the case, to $depth(T) - 1$.

   As $ST\alpha = ST_1\alpha \mid ST_2\alpha$ and there are rules $\frac{t \to v/ST_1\alpha}{t \to v/ST_1\alpha \mid ST_2\alpha}$ and $\frac{t \to v/ST_2\alpha}{t \to v/ST_1\alpha \mid ST_2\alpha}$ in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$, then either $\frac{T'_1}{t \to v/ST_1\alpha \mid ST_2\alpha}$ or $\frac{T'_2}{t \to v/ST_1\alpha \mid ST_2\alpha}$ is a c.p.t. for $[v]_{\mathcal{E}} \in ST\alpha@[t]_{\mathcal{E}}$ with the same depth and number of nodes as T.

   As $ST =_{\mathcal{E}} ST'$, then $ST' = ST'_1 \mid ST'_2$ where $ST_1 =_{\mathcal{E}} ST'_1$ and $ST_2 =_{\mathcal{E}} ST'_2$. As T_j, where j in $\{1, 2\}$, has smaller depth than T then, by I.H., there is a c.p.t. T' for $[v]_{\mathcal{E}} \in ST'_1@[t]_{\mathcal{E}}$ or $[v]_{\mathcal{E}} \in ST'_2@[t]_{\mathcal{E}}$, with the same depth and number of nodes as T_j, and $\frac{T'}{t \to v/ST'}$ is a c.p.t. for $[v]_{\mathcal{E}} \in ST'@[t]_{\mathcal{E}}$ with the same depth and number of nodes as T.

9. $ST = \texttt{match}\ u\ \texttt{s.t.}\ \phi\ ?\ ST_1\ :\ ST_2$.

   By the definition of the derivation rules for the if-then-else strategy, T must be of the form $\frac{T_1}{t \to v/ST}$ or $\frac{T_2}{t \to v/ST}$, where T_1 has head $t \to v/ST_1\delta$ or T_2 has head $t \to v/ST_2\delta$, coming from the application of a rule with the form $\frac{t' \to v'/ST_1\delta}{t' \to v'/ST}$ or $\frac{t' \to v'/ST_2\delta}{t' \to v'/ST}$, with $t =_{\mathcal{E}} t' =_{\mathcal{E}} u\delta$ and $v =_{\mathcal{E}} v'$. In the first case, by definition of the rule, $E_0 \vdash \phi\delta$ and, as T_1 is a c.p.t. for $t \to v/ST_1\delta$, $[v]_{\mathcal{E}} \in ST_1\delta@[t]_{\mathcal{E}}$; in the second case, also by definition of the rule, $E_0 \vdash \neg\phi\delta$ and, as T_2 is a c.p.t. for $t \to v/ST_2\delta$, $[v]_{\mathcal{E}} \in ST_2\delta@[t]_{\mathcal{E}}$ (**property 9**). In either case, as T_1 and T_2 are closed proof trees of a smaller depth whose head has the form $t \to v/\ldots$ then, by I.H., $t \to_{R/\mathcal{E}} v$.

   $ST\alpha = \texttt{match}\ u\alpha\ \texttt{s.t.}\ \phi\alpha\ ?\ ST_1\alpha\ :\ ST_2\alpha$. If we take $\delta' = \alpha^{-1}\delta$ then $\alpha\delta' = \delta$, so $u\alpha\delta' = u\delta$, $\phi\alpha\delta' = \phi\delta$, $ST_1\alpha\delta' = ST_1\delta$, and $ST_2\alpha\delta' = ST_2\delta$.

   - If $E_0 \vdash \phi\alpha\delta'$ (so $E_0 \vdash \phi\delta$) then T_1 exists and there is a rule $\frac{t' \to v'/ST_1\alpha\delta'}{t' \to v'/ST\alpha}$ (i.e., $\frac{t' \to v'/ST_1\delta}{t' \to v'/ST\alpha}$) in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$, so $\frac{T_1}{t \to v/ST\alpha}$ is a c.p.t. for $[v]_{\mathcal{E}} \in ST\alpha@[t]_{\mathcal{E}}$ with the same depth and number of nodes as T.
   - Else, T_2 exists and there is a rule $\frac{t' \to v'/ST_2\alpha\delta'}{t' \to v'/ST\alpha}$ (i.e., $\frac{t' \to v'/ST_2\delta}{t' \to v'/ST\alpha}$ in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$), so $\frac{T_2}{t \to v/ST\alpha}$ is a c.p.t. for $[v]_{\mathcal{E}} \in ST\alpha@[t]_{\mathcal{E}}$ with the same depth and number of nodes as T.

   As $ST =_{\mathcal{E}} ST'$, then $ST' = \texttt{match}\ u'\ \texttt{s.t.}\ \phi'\ ?\ ST'_1\ :\ ST'_2$ where $u =_{\mathcal{E}} u'$, $\phi =_{\mathcal{E}} \phi'$, $ST_1 =_{\mathcal{E}} ST'_1$, $ST_2 =_{\mathcal{E}} ST'_2$, $V_u = V_{u'}$, $V_\phi = V_{\phi'}$, $V_{ST_1} = V_{ST'_1}$, and $V_{ST_2} = V_{ST'_2}$. We prove the case where $E_0 \vdash \phi\delta$, the case where $E_0 \vdash \neg\phi\delta$ is proved in exactly the same way. As $\phi =_{\mathcal{E}} \phi'$ and $V_\phi = V_{\phi'}$ then $E_0 \vdash \phi'\delta$, ground formula. Also, as $u =_{\mathcal{E}} u'$ and $V_u = V_{u'}$, then $t =_{\mathcal{E}} t' =_{\mathcal{E}} u\delta =_{\mathcal{E}} u'\delta$, so there is a derivation rule $\frac{t' \to v'/ST'_1\delta}{t' \to v'/ST'}$. As $ST_1 =_{\mathcal{E}} ST'_1$ then $ST_1\delta =_{\mathcal{E}} ST'_1\delta$ so, by I.H. since $t =_{\mathcal{E}} t'$, $v =_{\mathcal{E}} v'$, and T_1 has smaller depth than T, there is a c.p.t. $T'_1 = \frac{T'}{t \to v/ST'_1\delta}$ for $[v]_{\mathcal{E}} \in ST'_1\delta@[t]_{\mathcal{E}}$, with the same depth and number of

nodes as T_1, and $\frac{T_1'}{t \to v/ST'}$ is a c.p.t. for $[v]_{\mathcal{E}} \in ST'@[t]_{\mathcal{E}}$ with the same depth and number of nodes as T.

10. $ST = CS$, where $\mathtt{sd}\ CS := ST_1$, and γ renaming such that $dom(\gamma) \subseteq vars(ST_1) \setminus V_{\mathcal{R}}$ and $ran(\gamma) \cap V_{\mathcal{R},Call_{\mathcal{R}}} = \emptyset$.
T must be of the form $\frac{T_1}{t \to v/CS}$, where T_1 has head $t \to v/ST_1\beta$, so $t \to_{R/\mathcal{E}} v$, by I.H., for some renaming β such that $ran(\beta) \cap V_{\mathcal{R},Call_{\mathcal{R}}} = \emptyset$ (hence $dom(\beta^{-1}) \cap V_{\mathcal{R}} = \emptyset$). Also by I.H., if we take β^{-1}, as $dom(\beta^{-1}) \cap V_{\mathcal{R}} = \emptyset$ then there is a c.p.t. T_1' with head $t \to v/ST_1$ and the same depth and number of nodes as T_1, so $[v]_{\mathcal{E}} \in ST_1@[t]_{\mathcal{E}}$ (i), and if we take $\gamma' = \beta^{-1}\gamma$, as also $dom(\gamma) \cap V_{\mathcal{R}} = \emptyset$, there must be a c.p.t. with head $t \to v/ST_1\beta\gamma'$ (i.e., $t \to v/ST_1\gamma$), with the same depth and number of nodes as T_1, so $[v]_{\mathcal{E}} \in ST_1\gamma@[t]_{\mathcal{E}}$ (ii) (**property 10**).
As $dom(\alpha) \subseteq vars(CS) = \emptyset$ then $\alpha = \mathtt{none}$, so $ST\alpha = CS$ and T is also a c.p.t. for $[v]_{\mathcal{E}} \in ST\alpha@[t]_{\mathcal{E}}$.
As $ST' =_{\mathcal{E}} ST$, then $ST' = CS = ST$, and T is also a c.p.t. for $[v]_{\mathcal{E}} \in ST'@[t]_{\mathcal{E}}$.

11. $ST = CS(\bar{t})$, where $\bar{t} = t_1, \ldots, t_n$, $\mathtt{sd}\, CS(\bar{x}) := ST_1 \in Call_{\mathcal{R}}$, $\bar{x} = x^1_{s_1}, \ldots, x^n_{s_n}$, $\hat{x} \subseteq V_{CS}$, $\rho = \{x^1_{s_1} \mapsto t_1, \ldots, x^n_{s_n} \mapsto t_n\}$, with $ran(\rho) \subset \mathcal{X} \setminus V_{\mathcal{R},Call_{\mathcal{R}}}$ by the definition of call strategy, and γ is a renaming such that $dom(\gamma) \subseteq vars(ST_1) \setminus \hat{x}$ and $ran(\gamma) \cap (ran(\rho) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$.
T must be of the form $\frac{T_1}{t \to v/CS(\bar{t})}$, where T_1 has head $t \to v/ST_1(\beta \cup \rho)$ (so, by I.H., $t \to_{R/\mathcal{E}} v$) for some renaming β such that $dom(\beta) \subseteq vars(ST_1) \setminus (\hat{x} \cup V_{\mathcal{R}})$ and $ran(\beta) \cap (ran(\rho) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$, hence $(\beta \cup \rho)\beta^{-1} = \rho$. Then, by I.H., there must exist a c.p.t T_2 with head $t \to v/ST_1\rho$ and the same depth and number of nodes as T_1 so $[v]_{\mathcal{E}} \in ST_1\rho@[t]_{\mathcal{E}}$ (i).
As $dom(\gamma) \subseteq vars(ST_1) \setminus \hat{x} \subseteq V_{\mathcal{R},Call_{\mathcal{R}}}$, $dom(\rho) = \hat{x}$, and $ran(\rho) \subset \mathcal{X} \setminus V_{\mathcal{R},Call_{\mathcal{R}}}$, so $ran(\rho) \cap dom(\gamma) = \emptyset$, then $ST_1(\gamma \cup \rho) = ST_1\rho\gamma$, with $dom(\gamma) \subseteq vars(ST_1\rho)$. Then $dom(\gamma) \subseteq vars(ST_1\rho)$. As T_2 has head $t \to v/ST_1\rho$ and the same depth and number of nodes as T_1, $dom(\gamma) \subseteq vars(ST_1\rho \setminus V_{\mathcal{R}})$, and $ran(\gamma) \cap (vars(ST_1\rho) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$ then, by I.H., there must exist a c.p.t. T_3 with head $t \to v/ST_1\rho\gamma$ (i.e., $t \to v/ST_1(\gamma \cup \rho)$), so $[v]_{\mathcal{E}} \in ST_1(\gamma \cup \rho)@[t]_{\mathcal{E}}$ (ii) (**property 11**).
As $dom(\alpha) \subseteq vars(ST) \setminus \hat{x} = vars(CS(\bar{t})) \setminus \hat{x} = ran(\rho)$, because $\hat{x} \notin vars(CS(\bar{t}))$ and $ran(\rho) \cap V_{\mathcal{R},Call_{\mathcal{R}}} = \emptyset$, then $ST\alpha = CS(\bar{t}\alpha)$ and as $ran(\rho) \subset \mathcal{X} \setminus V_{\mathcal{R},Call_{\mathcal{R}}} \subset \mathcal{X} \setminus vars(ST_1)$, so $dom(\alpha) \cap vars(ST_1) = \emptyset$, then $ST_1(\rho\alpha) = (ST_1\rho)\alpha$ and there is a derivation rule $\frac{t \to v/(ST_1\rho)\alpha}{t \to v/CS(\bar{t}\alpha)}$ in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$. Now, as T_2 has head $t \to v/ST_1\rho$ and depth one less than the depth of T, $dom(\alpha) \subseteq ran(\rho) \subseteq vars(ST_1\rho)$ and $vars(ST_1) \subseteq V_{\mathcal{R},Call_{\mathcal{R}}}$, so $ran(\alpha) \cap (vars(ST_1\rho) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) \subseteq ran(\alpha) \cap (ran(\rho) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = ran(\alpha) \cap (vars(ST) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$ then, by I.H., there is a c.p.t. T_4 with head $t \to v/(ST_1\rho)\alpha$ and the same depth and number of nodes as T_1, so $\frac{T_4}{t \to v/ST\alpha}$ is a c.p.t. for $[v]_{\mathcal{E}} \in ST\alpha@[t]_{\mathcal{E}}$ with the same depth and number of nodes as T.
As $ST' =_{\mathcal{E}} ST$, then $ST' = CS(\bar{t}')$, where $\bar{t} =_{\mathcal{E}} \bar{t}'$. Let $\rho' = \bar{x} \mapsto \bar{t}'$, so $\rho' =_{\mathcal{E}} \rho$. As $T = \frac{T_1}{t \to v/CS(\bar{t})}$, where T_1 has head $t \to v/ST_1(\beta \cup \rho)$, then there is a derivation rule $\frac{t \to v/ST_1(\beta \cup \rho)}{t \to v/CS(\bar{t})}$, so there is also a derivation rule $\frac{t \to v/ST_1(\beta \cup \rho')}{t \to v/CS(\bar{t}')}$. As $ST_1(\beta \cup \rho) =_{\mathcal{E}} ST_1(\beta \cup \rho')$ then, by I.H., there is a c.p.t. T_1' for $[v]_{\mathcal{E}} \in$

$ST_1(\beta \cup \rho')@[t]_{\mathcal{E}}$ with the same depth and number of nodes as T_1, so $\frac{T_1'}{t \to v/CS(\bar{t}')}$ is a c.p.t. for $[v]_{\mathcal{E}} \in CS(\bar{t}')@[t]_{\mathcal{E}}$ with the same depth and number of nodes as T.

12. $ST = CS(\bar{t})$, where $\bar{t} = t_1, \ldots, t_n$, $\texttt{csd}\ CS(\bar{x}) := ST_1$ *if* $C \in Call_{\mathcal{R}}$, with $\bar{x} = x^1_{s_1}, \ldots, x^n_{s_n}$, $\hat{x} \subseteq V_{CS}$, and $C = \bigwedge_{j=1}^{m}(l_j = r_j) \wedge \phi$, let $V_C = vars(C)$, $V_{CS} = vars(ST_1) \cup V_C$, and $\rho = \{x^1_{s_1} \mapsto t_1, \ldots, x^n_{s_n} \mapsto t_n\}$, with $ran(\rho) \cap V_{\mathcal{R},Call_{\mathcal{R}}} = \emptyset$, and γ is a renaming such that $dom(\gamma) \subseteq V_{CS} \setminus \hat{x} = V_{CS} \setminus dom(\rho)$ and $ran(\gamma) \cap (ran(\rho) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$, so $C(\gamma \cup \rho)(\gamma_{vars(C)})^{-1} = C\rho$.

T must be of the form $\frac{T_1}{t \to v/CS(\bar{t})}$, where T_1 has head $t \to v/ST_1(\beta \cup \rho)\delta$ (so, by I.H., $t \to_{R/\mathcal{E}} v$) for some renaming β such that $dom(\beta) \subseteq V_{CS} \setminus \hat{x} = V_{CS} \setminus dom(\rho)$, so $dom(\beta) \cap dom(\rho) = \emptyset$ and $ran(\beta) \cap (ran(\rho) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$, so $ran(\beta) \cap (ran(\rho) \cup dom(\rho)) = \emptyset$ hence $\rho\beta = \beta \cup \rho$, and some substitution $\delta : vars(C(\beta \cup \rho)) \to \mathcal{T}_\Sigma$ such that $\bar{l}(\beta \cup \rho)\delta =_{\mathcal{E}} \bar{r}(\beta \cup \rho)\delta$ and $E_0 \vdash \phi(\beta \cup \rho)\delta$.

Let $\delta_1 = \beta\delta$. As $\rho\beta = \beta \cup \rho$ then $\delta_1 : vars(C\rho) \to \mathcal{T}_\Sigma$ is a substitution such that $l_j\rho\delta_1 =_{\mathcal{E}} r_j\rho\delta_1$, for $1 \leq j \leq n$, $E_0 \vdash \phi\rho\delta_1$. Also as $\rho\beta = \beta \cup \rho$, so $(\beta \cup \rho)\delta = \rho\beta\delta = \rho\delta_1$, T_1 is a c.p.t with head $t \to v/ST_1\rho\delta_1$ so, by definition, $[v]_{\mathcal{E}} \in ST_1\rho\delta_1@[t]_{\mathcal{E}}$ (i).

As $C(\gamma \cup \rho)(\gamma_{V_C})^{-1} = C\rho$ then $C(\gamma \cup \rho)(\gamma_{V_C})^{-1}\delta_1 = C\rho\delta_1$, let $\delta_2 = (\gamma_{V_C})^{-1}\delta_1$, hence $\delta_2 : vars(C(\gamma \cup \rho)) \to \mathcal{T}_\Sigma$ is a substitution such that $l_j(\gamma \cup \rho)\delta_2 =_{\mathcal{E}} r_j(\gamma \cup \rho)\delta_2$, for $1 \leq j \leq n$, and $E_0 \vdash \phi(\gamma \cup \rho)\delta_2$. As $dom(\delta_1) = vars(C\rho)$ then $ST_1(\gamma \cup \rho)\delta_2 = ST_1(\gamma \cup \rho)(\gamma_{V_C})^{-1}\delta_1 = ST_1(\gamma_{V_C} \cup \gamma_{\setminus V_C} \cup \rho)(\gamma_{V_C})^{-1}\delta_1 = ST_1(\gamma_{\setminus V_C} \cup \rho)\delta_1 = ST_1(\gamma_{\setminus V_C} \cup \rho\delta_1)$, because as $ran(\gamma) \cap (ran(\rho) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$ and $vars(ST_1) \subseteq V_{\mathcal{R},Call_{\mathcal{R}}}$ then after $\gamma_{\setminus V_C}$ instantiates ST_1 in $ST_1(\gamma_{\setminus V_C} \cup \rho)$, δ_1 does not instantiate any renamed variable in $ran(\gamma_{\setminus V_C})$. Now, as δ_1 ground implies $ran(\rho\delta_1) \subseteq ran(\rho)$, $ran(\rho) \cap V_{\mathcal{R},Call_{\mathcal{R}}} = \emptyset$, and $dom(\gamma_{\setminus V_C}) \subseteq vars(ST_1) \subseteq V_{\mathcal{R},Call_{\mathcal{R}}}$, then $ST_1(\gamma_{\setminus V_C} \cup \rho\delta_1) = ST_1\rho\delta_1\gamma_{\setminus V_C}$, i.e., $ST_1(\gamma \cup \rho)\delta_2 = ST_1\rho\delta_1\gamma_{\setminus V_C}$.

In order to use I.H. we need to prove $ran(\gamma_{\setminus V_C}) \cap (vars(ST_1\rho\delta_1) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$ and $dom(\gamma_{\setminus V_C}) \subseteq vars(ST_1\rho\delta_1)$.

- By definition, $ran(\gamma) \cap (ran(\rho) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$. As $ran(\rho\delta_1) \subseteq ran(\rho)$ then also $ran(\gamma) \cap (ran(\rho\delta_1) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$, so $ran(\gamma_{\setminus V_C}) \cap (vars(ST_1\rho\delta_1) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$ because $vars(ST_1) \subseteq V_{\mathcal{R},Call_{\mathcal{R}}}$.
- As $dom(\gamma) \subseteq V_{CS} \setminus dom(\rho)$ and $V_{CS} = vars(ST_1) \cup V_C$ then $dom(\gamma_{\setminus V_C}) \subseteq vars(ST_1) \setminus (dom(\rho) \cup V_C)$ so $dom(\gamma_{\setminus V_C}) \subseteq vars(ST_1\rho) \setminus V_C$. Now, as $ran(\rho) \cap V_{\mathcal{R},Call_{\mathcal{R}}} = \emptyset$, so $ran(\rho) \cap vars(ST_1) = \emptyset$, and $dom(\gamma_{\setminus V_C}) \subseteq vars(ST_1) \setminus V_C$, then $dom(\gamma_{\setminus V_C}) \subseteq vars(ST_1\rho) \setminus (V_C \cup ran(\rho))$ so, as $dom(\delta_1) = vars(C\rho) \subseteq V_C \cup ran(\rho)$, then $dom(\gamma_{\setminus V_C}) \subseteq vars(ST_1\rho\delta_1)$.

Then, by I.H., there is a c.p.t. for $[v]_{\mathcal{E}} \in ST_1\rho\delta_1\gamma_{\setminus V_C}@[t]_{\mathcal{E}}$ hence, as $ST_1(\gamma \cup \rho)\delta_2 = ST_1\rho\delta_1\gamma_{\setminus V_C}$, also $[v]_{\mathcal{E}} \in ST_1(\gamma \cup \rho)\delta_2@[t]_{\mathcal{E}}$ (ii) (**property 12**).

As $dom(\alpha) \subseteq vars(ST) \setminus \hat{x} = vars(CS(\bar{t})) \setminus \hat{x} = ran(\rho)$, because $\hat{x} \notin vars(CS(\bar{t}))$ and $ran(\rho) \cap V_{\mathcal{R},Call_{\mathcal{R}}} = \emptyset$, then $ST\alpha = CS(\bar{t}\alpha)$. Also, as $ran(\alpha) \cap (vars(ST) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$, then $ran(\alpha) \cap (ran(\rho) \cup dom(\rho)) = \emptyset$ and $dom(\alpha^{-1}) \cap (vars(ST) \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$ so, as $V_{CS} \subseteq V_{\mathcal{R},Call_{\mathcal{R}}}$, $C\rho\alpha\alpha^{-1} = C\rho$ and $ST_1\rho\alpha\alpha^{-1} = ST_1\rho$, hence $C\rho\alpha\alpha^{-1}\delta_1 = C\rho\delta_1$ and $ST_1\rho\alpha\alpha^{-1}\delta_1 = ST_1\rho\delta_1$, let $\delta_3 = \alpha^{-1}\delta_1$, so $\delta_3 : vars(C\rho\alpha) \to \mathcal{T}_\Sigma$ is a substitution such that $l_j\rho\alpha\delta_3 =_{\mathcal{E}} r_j\rho\alpha\delta_3$, for $1 \leq j \leq n$ and $E_0 \vdash \phi\rho\alpha\delta_3$ and there is a derivation rule $\frac{t \to v/ST_1\rho\alpha\delta_3}{t \to v/CS(\bar{t}\alpha)} \in \mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$.

Then, as $ST_1\rho\alpha\delta_3 = ST_1\rho\delta_1$ implies $t \to v/ST_1\rho\alpha\delta_3 = t \to v/ST_1\rho\delta_1$ and T_1 has head $t \to v/ST_1\rho\delta_1$, $\frac{T_1}{t\to v/CS(\bar{t}\alpha)}$ is a c.p.t. for $[v]_\mathcal{E} \in ST\alpha@[t]_\mathcal{E}$ with the same depth and number of nodes as T.

As $ST' =_\mathcal{E} ST$, then $ST' = CS(\bar{t}')$, where $\bar{t} =_\mathcal{E} \bar{t}'$. Let $\rho' = \bar{x} \mapsto \bar{t}'$, so $\rho' =_\mathcal{E} \rho$. As $T = \frac{T_1}{t\to v/CS(\bar{t})}$, where T_1 has head $t \to v/ST_1(\beta \cup \rho)$, then there is a derivation rule $\frac{t\to v/ST_1(\beta\cup\rho)}{t\to v/CS(\bar{t})}$. As $\rho =_\mathcal{E} \rho'$, then $\bar{l}(\beta \cup \rho')\delta =_\mathcal{E} \bar{l}(\beta \cup \rho)\delta =_\mathcal{E} \bar{r}(\beta \cup \rho)\delta =_\mathcal{E} \bar{r}(\beta \cup \rho')\delta$ and $E_0 \vdash \phi(\beta \cup \rho')\delta$, so there is also a derivation rule $\frac{t\to v/ST_1(\beta\cup\rho')}{t\to v/CS(\bar{t}')}$. As $ST_1(\beta \cup \rho) =_\mathcal{E} ST_1(\beta \cup \rho')$ then, by I.H., there is a c.p.t. T_1' for $[v]_\mathcal{E} \in ST_1(\beta \cup \rho')@[t]_\mathcal{E}$ with the same depth and number of nodes as T_1, so $\frac{T_1'}{t\to v/CS(\bar{t}')}$ is a c.p.t. for $[v]_\mathcal{E} \in CS(\bar{t}')@[t]_\mathcal{E}$ with the same depth and number of nodes as T.

13. $ST = c[\gamma]\{\overline{ST}\}$, with $c : l \to r$ *if* $\bigwedge_{j=1}^m l_j \to r_j \mid \psi$ a rule in R, $\overline{ST} = ST_1, \ldots, ST_m$, and $dom(\gamma) \cap vars(\overline{ST}) = \emptyset$.

T must be of the form $\frac{T_1\cdots T_m}{t\to v/c[\gamma]\{\overline{ST}\}}$, where T_i, $1 \le i \le m$, are closed proof trees with head $l_i\gamma\delta \to r_i\gamma\delta/ST_i\delta$ (so, by I.H., $l_i\gamma\delta \to_{R/\mathcal{E}} r_i\gamma\delta$ and $[r_j\gamma\delta]_\mathcal{E} \in ST_j\delta@[l_j\gamma\delta]_\mathcal{E}$), because there is a derivation rule $\frac{l_1\gamma\delta\to r_1\gamma\delta/ST_1\delta\cdots l_m\gamma\delta\to r_m\gamma\delta/ST_m\delta}{u\to u[r\gamma\delta]_p/c[\gamma]\{\overline{ST}\}} \in \mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$, where $u \in \mathcal{H}_\Sigma$, $p \in pos(u)$, $\delta : vars(c\gamma) \to \mathcal{T}_\Sigma$, $u = u[l\gamma\delta]_p =_\mathcal{E} t$, $u[r\gamma\delta]_p =_\mathcal{E} v$, and $E_0 \vdash \psi\gamma\delta$ so, by definition as also $l_i\gamma\delta \to_{R/\mathcal{E}} r_i\gamma\delta$, $1 \le i \le m$, $t \xrightarrow[c,u,p,\gamma\delta]{1}_{R/\mathcal{E}} v$ (**property 13**).

Let $\gamma' = (\gamma\alpha)_{dom(\gamma)}$ so $ST\alpha = c[\gamma']\{\overline{ST}\alpha\}$.

If we take $\delta' = \alpha^{-1}\delta$, as $dom(\alpha^{-1}) = ran(\alpha)$, $ran(\alpha) \cap (V_T \cup V_{\mathcal{R},Call_\mathcal{R}}) = \emptyset$, $\delta : vars(c\gamma) \to \mathcal{T}_\Sigma$, then $c\gamma'\delta' = c(\gamma\alpha)_{dom(\gamma)}\alpha^{-1}\delta = c\gamma\delta$, so $\delta' : vars(c\gamma') \to \mathcal{T}_\Sigma$ with $E_0 \vdash \psi\gamma'\delta'$, $u|_p = l\gamma'\delta'$, and $\overline{ST}\alpha\delta' = \overline{ST}\delta$.

Then, $\frac{l_1\gamma'\delta'\to r_1\gamma'\delta'/ST_1\alpha\delta'\cdots l_m\gamma'\delta'\to r_m\gamma'\delta'/ST_m\alpha\delta'}{u\to u[r\gamma'\delta']_p/ST\alpha}$, i.e., $\frac{l_1\gamma\delta\to r_1\gamma\delta/ST_1\delta\cdots l_m\gamma\delta\to r_m\gamma\delta/ST_m\delta}{u\to u[r\gamma\delta]_p/ST\alpha}$ is a derivation rule in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$, so $\frac{T_1\cdots T_m}{t\to v/ST\alpha}$ is a c.p.t. for $[v]_\mathcal{E} \in ST\alpha@[t]_\mathcal{E}$ with the same depth and number of nodes as T.

As $ST = c[\gamma]\{\overline{ST}\} =_\mathcal{E} ST'$, then $ST' = c[\gamma']\{\overline{ST}'\}$ where $\overline{ST} =_\mathcal{E} \overline{ST}'$ and $\gamma =_\mathcal{E} \gamma'$, so $(l, r, \psi, \bar{l}, \bar{r})\gamma =_\mathcal{E} (l, r, \psi, \bar{l}', \bar{r}')\gamma'$, with $V_{l\gamma} = V_{l\gamma'}$, $V_{r\gamma} = V_{r\gamma'}$, $V_{\bar{l}\gamma} = V_{\bar{l}\gamma'}$ and $V_{\bar{r}\gamma} = V_{\bar{r}\gamma'}$, hence $E_0 \vdash \psi\gamma'\delta$, $t =_\mathcal{E} t'[l\gamma\delta]_p =_\mathcal{E} t'[l\gamma'\delta]_p$ and $v =_\mathcal{E} t'[r\gamma\delta]_p =_\mathcal{E} t'[r\gamma'\delta]_p$, ground terms and formula. Then, $\frac{l_1\gamma'\delta\to r_1\gamma'\delta/ST_1'\delta\cdots l_m\gamma'\delta\to r_m\gamma'\delta/ST_m'\delta}{u\to u[r\gamma'\delta]_p/c[\gamma']\{\overline{ST}'\}}$ is a derivation rule in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$. Again, by I.H., since $\overline{ST}\delta =_\mathcal{E} \overline{ST}'\delta$ and $(\bar{l}, \bar{r})\gamma\delta =_\mathcal{E} (\bar{l}, \bar{r})\gamma'\delta$, there exist a c.p.t. T_j' with the same depth and number of nodes as T_j for $[r_j\gamma'\delta]_\mathcal{E} \in ST_j'\delta@[l_j\gamma'\delta]_\mathcal{E}$, for $1 \le j \le m$, so $\frac{T_1'\cdots T_m'}{t\to v/c[\gamma']\{\overline{ST}'\}}$ is a c.p.t. for $[v]_\mathcal{E} \in c[\gamma']\{\overline{ST}'\}@[t]_\mathcal{E}$.

14. $ST = \mathtt{top}(c[\gamma]\{\overline{ST}\})$, with $c : l \to r$ *if* $\bigwedge_{j=1}^m l_j \to r_j \mid \psi$ a rule in R, $\overline{ST} = ST_1, \ldots, ST_m$, and $dom(\gamma) \cap vars(\overline{ST}) = \emptyset$.

T must be of the form $\frac{T_1\cdots T_m}{t\to v/c[\gamma]\{\overline{ST}\}}$, where T_i, $1 \le i \le m$, are closed proof trees with head $l_i\gamma\delta \to r_i\gamma\delta/ST_i\delta$ (so, by I.H., $l_i\gamma\delta \to_{R/\mathcal{E}} r_i\gamma\delta$ and $[r_j\gamma\delta]_\mathcal{E} \in ST_j\delta@[l_j\gamma\delta]_\mathcal{E}$), because there is a derivation rule $\frac{l_1\gamma\delta\to r_1\gamma\delta/ST_1\delta\cdots l_m\gamma\delta\to r_m\gamma\delta/ST_m\delta}{l\gamma\delta\to r\gamma\delta/\mathtt{top}(c[\gamma]\{ST_1,\ldots,ST_m\})} \in \mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$, where $\delta : vars(c\gamma) \to \mathcal{T}_\Sigma$, $l\gamma\delta =_\mathcal{E} t$, $r\gamma\delta =_\mathcal{E} v$, and $E_0 \vdash \psi\gamma\delta$.

As $l_i\gamma\delta \to_{R/\mathcal{E}} r_i\gamma\delta$, $1 \le i \le m$, $t =_\mathcal{E} l\gamma\delta$, $v =_\mathcal{E} r\gamma\delta$, and $E_0 \vdash \psi\gamma\delta$ then, by definition, $t \xrightarrow[c,u,\epsilon,\gamma\delta]{1}_{R/\mathcal{E}} v$ (**property 14**).

The proofs for the existence of a c.p.t. for $[v]_{\mathcal{E}} \in ST\alpha@[t]_{\mathcal{E}}$ and $[v]_{\mathcal{E}} \in ST'@[t]_{\mathcal{E}}$ with the same depth and number of nodes as T are the same proofs shown in the previous subcase, particularized for the position $p = \epsilon$, so $u = l\gamma\delta$ and $u[r\gamma\delta]_p = r\gamma\delta$.

15. $ST = \texttt{matchrew}\ u\ \texttt{s.t.}\ C\ \texttt{by}\ x^1_{s_1}\ \texttt{using}\ ST_1, \ldots, x^n_{s_n}\ \texttt{using}\ ST_n$.
Let $\bar{x} = x^1_{s_1}, \ldots, x^n_{s_n}$, where $C = \bigwedge_{j=1}^m (l_j = r_j) \wedge \phi$, $u = u[x^1_{s_1}, \ldots, x^n_{s_n}]_{p_1 \ldots p_n}$, and $\hat{x} = \{\bar{x}\}$.
T must be of the form $\frac{T_1 \cdots T_n}{t \to v/ST}$, where each T_i is a c.p.t. with head $x^i_{s_i}\delta \to t_i/ST_i\delta$, $1 \leq i \leq n$, by application of a rule $\frac{x^1_{s_1}\delta \to t_1/ST_1\delta \cdots x^n_{s_n}\delta \to t_n/ST_n\delta}{u\delta \to u\delta[\bar{t}]_{\bar{p}}/ST} \in \mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$, so $V_{(u,\bar{l},\bar{r},\phi)\delta} = \emptyset$ and $V_{\overline{ST}\delta} \subseteq V_T$, where $\delta_{V_{ST}} : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X} \backslash V_{ST})$, $ran(\delta_{V_{ST}}) \subseteq V_{\overline{ST}\delta}$, $t =_{\mathcal{E}} u\delta$, $v =_{\mathcal{E}} u\delta[\bar{t}]_{\bar{p}}$, $\bar{l}\delta =_{\mathcal{E}} \bar{r}\delta$, and $E_0 \vdash \phi\delta$ so, by I.H., $[t_j]_{\mathcal{E}} \in ST_j\delta@[x^j_{s_j}\delta]_{\mathcal{E}}$, for $1 \leq j \leq n$ (**property 15**). Also by I.H., $x^j_{s_j}\delta \to_{R/\mathcal{E}} t_j$, for $1 \leq j \leq n$. Then, by congruence of rewriting, $t =_{\mathcal{E}} u\delta[x^1_{s_1}\delta, \ldots, x^n_{s_n}\delta]_{p_1 \ldots p_n} \to_{R/\mathcal{E}} u\delta[\bar{t}]_{\bar{p}} =_{\mathcal{E}} v$ (i.e., $t \to_{R/\mathcal{E}} v$).
Let $\alpha' = \alpha_{\backslash \bar{x}}$. Then $ST\alpha$ has the form

$$\texttt{matchrew}\ u\alpha'\ \texttt{s.t.}\ C\alpha'\ \texttt{by}\ x^1_{s_1}\ \texttt{using}\ ST_1\alpha', \ldots, x^n_{s_n}\ \texttt{using}\ ST_n\alpha'$$

i.e., $ST\alpha = ST\alpha'$, with $ran(\alpha) \cap (V_T \cup V_{\mathcal{R},Call_{\mathcal{R}}}) = \emptyset$. Let $\delta' = (\alpha')^{-1}\delta$. As $ran(\alpha) \cap V_T = \emptyset$, $ran(\delta_{V_{ST}}) \subseteq V_{\overline{ST}\delta} \subseteq V_T$, and $ran(\delta_{V_{ST}}) \cap V_{ST} = \emptyset$, then $ran(\alpha) \cap ran(\delta_{V_{ST}}) = \emptyset$, hence $ran(\alpha') \cap ran(\delta_{V_{ST}}) = \emptyset$. As also $V_{ST} \cap ran(\delta_{V_{ST}}) = \emptyset$ and $V_{ST\alpha'} \subseteq V_{ST} \cup ran(\alpha')$ then, for each $x \in V_{ST}$, $x\alpha'\delta' = x\delta$ and:

- if $x \in dom(\delta)$ then $V_{x\delta} \subseteq ran(\delta_{V_{ST}})$, so $V_{x\delta} \cap V_{ST\alpha'} = \emptyset$, i.e., $V_{x\alpha'\delta'} \cap V_{ST\alpha'} = \emptyset$, and
- if $x \notin dom(\delta)$ then $x\delta = x$ and:
  - if $x \in dom(\alpha')$ then, as $ran(\alpha) \cap V_T = \emptyset$, hence also $ran(\alpha') \cap V_T = \emptyset$, and $x \in V_{ST} \subseteq V_T$, then $x \notin V_{ST\alpha'}$, i.e., $\emptyset = V_{x\delta} \cap V_{ST\alpha'} = V_{x\alpha'\delta'} \cap V_{ST\alpha'}$;
  - if $x \notin dom(\alpha')$ then $x \in V_{ST\alpha'} \setminus ran(\alpha') = V_{ST\alpha'} \setminus dom((\alpha')^{-1})$, so $x\delta' = x(\alpha')^{-1}\delta = x\delta = x$, i.e., $x \notin dom(\delta'_{V_{ST\alpha'}})$.

Then $\delta'_{V_{ST\alpha'}} : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X} \backslash vars(ST\alpha'))$ and $ST\alpha\delta' = ST\alpha'\delta' = ST\delta$, hence $t =_{\mathcal{E}} u\alpha'\delta' = u\delta \in \mathcal{T}_\Sigma$, $\bar{l}\alpha'\delta' = \bar{l}\delta =_{\mathcal{E}} \bar{r}\delta = \bar{r}\alpha'\delta'$, so $\{l_j\alpha'\delta', r_j\alpha'\delta'\}_{j=1}^m \subset \mathcal{T}_\Sigma$, $\phi\alpha'\delta' = \phi\delta \in \mathcal{T}_\Sigma$, and $E_0 \vdash \phi\alpha'\delta'$, hence there is a derivation rule $\frac{x^1_{s_1}\alpha'\delta' \to t_1/ST_1\alpha'\delta' \cdots x^n_{s_n}\alpha'\delta' \to t_n/ST_n\alpha'\delta'}{u\alpha'\delta' \to u\alpha'\delta'[\bar{t}]_{\bar{p}}/ST\alpha'}$ in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$. As $u\alpha'\delta' = u\delta$, $ST\alpha = ST\alpha'$, $\overline{ST}\alpha'\delta' = \overline{ST}\delta$, and $\bar{x}\alpha'\delta' = \bar{x}\delta$, because $\bar{x} \subseteq mp(ST)$, this is the same as $\frac{x^1_{s_1}\delta \to t_1/ST_1\delta \cdots x^n_{s_n}\delta \to t_n/ST_n\delta}{u\delta \to u\delta[\bar{t}]_{\bar{p}}/ST\alpha} \in \mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$. Then, as $t =_{\mathcal{E}} u\delta$ and $v =_{\mathcal{E}} u\delta[\bar{t}]_{\bar{p}}$, $\frac{T_1 \cdots T_n}{t \to v/ST\alpha}$ is a c.p.t. for $[v]_{\mathcal{E}} \in ST\alpha@[t]_{\mathcal{E}}$.

As $ST =_{\mathcal{E}} ST'$, then $ST' = \texttt{matchrew}\ u'\ \texttt{s.t.}\ C'\ \texttt{by}\ \bar{x}\ \texttt{using}\ \overline{ST}'$ where $\overline{ST} =_{\mathcal{E}} \overline{ST}'$ $C =_{\mathcal{E}} C' = \bigwedge_{j=1}^m (l'_j = r'_j) \wedge \phi'$, so $(\phi, \bar{l}, \bar{r}) =_{\mathcal{E}} (\phi', \bar{l}', \bar{r}')$, with $V_u = V_{u'} = \hat{x}$, $V_\phi = V_{\phi'}$, $V_{\bar{l}\gamma} = V_{\bar{l}\gamma'}$ and $V_{\bar{r}\gamma} = V_{\bar{r}\gamma'}$, so $t =_{\mathcal{E}} u\delta =_{\mathcal{E}} u'\delta$, $v =_{\mathcal{E}} u\delta[\bar{t}]_{\bar{p}} =_{\mathcal{E}} u\delta'[\bar{t}]_{\bar{p}}$, $\bar{l}\delta =_{\mathcal{E}} \bar{r}\delta$, and $E_0 \vdash \phi'\delta$, ground terms and formula.

Then, there is a derivation rule $\frac{x^1_{s_1}\delta \to t_1/ST'_1\delta \cdots x^n_{s_n}\delta \to t_n/ST'_n\delta}{u'\delta \to u\delta[\bar{t}]_{\bar{p}}/ST'} \in \mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$. Again, by I.H., since $\overline{ST}\delta =_{\mathcal{E}} \overline{ST}'\delta$, there exist a c.p.t. T'_j with the same depth and

number of nodes as T_j, for $[t_j]_{\mathcal{E}} \in ST'_j\delta@[x^j_{s_j}\delta]_{\mathcal{E}}$, for $1 \leq j \leq n$, so $\frac{T'_1 \cdots T'_n}{t \to v/ST'}$ is a c.p.t. for $[v]_{\mathcal{E}} \in \overline{ST}'@[t]_{\mathcal{E}}$ with the same depth and number of nodes as T.

□

Lemma 13 (Generalization of strategies)**.** *Given a rewrite theory $\mathcal{R} = (\Sigma, E_0 \cup B, R)$, a set of call strategy definitions $Call_{\mathcal{R}}$, terms $t, v \in \mathcal{H}_\Sigma$, a strategy $ST \in Strat_{\mathcal{R},Call_{\mathcal{R}}}$, and a substitution σ such that $dom(\sigma) \cap V_R = \emptyset$ and $ran(\sigma) \cap (V_R \cup V_{ST}) = \emptyset$, if $[v]_{\mathcal{E}} \in ST\sigma@[t]_{\mathcal{E}}$ can be proved with a c.p.t. T then $[v]_{\mathcal{E}} \in ST@[t]_{\mathcal{E}}$ and a c.p.t. T' with head $t \to v/ST$ and the same depth as T can be constructed.*

Proof. The proof is done by induction on the depth of T.

- There are five strategies in the base case: `fail`, `idle`, $c[\gamma]$, `top`$(c[\gamma])$, and the `match` test. The depth of all the closed proof trees is one in this case.
  - As there are no derivation rules for `fail`, there is nothing to prove in this case.
  - If $ST =$ `idle` then $ST\sigma = ST$ and $T' = T$.
  - If $ST = c[\gamma]$ then $ST\sigma = c[(\gamma\sigma)_{dom(\gamma)}]$. As $dom(\sigma) \cap V_R = \emptyset$ then $c(\gamma\sigma)_{dom(\gamma)} = c\gamma\sigma_{ran(\gamma)}$. $T = \frac{}{t \to v/ST\sigma}$ because c has the form $c : l \to r \; if \; \phi$, and there exist $u \in \mathcal{H}_\Sigma$, $p \in pos(u)$, and $\delta : V_{c\gamma\sigma_{ran(\gamma)}} \to \mathcal{T}_\Sigma$ such that $u \xrightarrow[c\gamma\sigma_{ran(\gamma)},p,\delta]{1}_R w$, i.e., $u = u[l\gamma\sigma_{ran(\gamma)}\delta]_p$ and $E_0 \vdash \phi\gamma\sigma_{ran(\gamma)}\delta$, so there is a derivation rule $\frac{}{u \to w/ST\sigma}$ in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$, $t =_{\mathcal{E}} u$, and $w = u[r\gamma\sigma_{ran(\gamma)}\delta]_p =_E v$. Then, also $u \xrightarrow[c\gamma,p,\sigma_{ran(\gamma)}\delta]{1}_R w$, because as, by definition, $dom(\gamma) \subseteq vars(c)$ then $\sigma_{ran(\gamma)}\delta : V_{c\gamma} \to \mathcal{T}_\Sigma$, so there is a derivation rule $\frac{}{u \to w/ST}$ in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$ and $T' = \frac{}{t \to v/ST}$.
  - if $ST =$ `top`$(c[\gamma])$ then $ST\sigma =$ `top`$(c[\gamma\sigma_{ran(\gamma)}])$. As $dom(\sigma) \cap V_R = \emptyset$ then $c(\gamma\sigma)_{dom(\gamma)} = c\gamma\sigma_{ran(\gamma)}$. $T = \frac{}{t \to v/ST\sigma}$ because c has the form $c : l \to r if \phi$, there exists $\delta : V_{c\gamma\sigma_{ran(\gamma)}} \to \mathcal{T}_\Sigma$ such that $E_0 \vdash \phi\gamma\sigma_{ran(\gamma)}\delta$, so $\frac{}{l\gamma\sigma_{ran(\gamma)} \to r\gamma\sigma_{ran(\gamma)}/ST\sigma}$ is a derivation rule in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$, $t =_{\mathcal{E}} l\gamma\sigma_{ran(\gamma)}\delta$, and $r\gamma\sigma_{ran(\gamma)}\delta =_{\mathcal{E}} v$. Again, by definition, $dom(\gamma) \subseteq vars(c)$ so $\sigma_{ran(\gamma)}\delta : V_{c\gamma} \to \mathcal{T}_\Sigma$ and, as $E_0 \vdash \phi\gamma\sigma_{ran(\gamma)}\delta$, there is a derivation rule $\frac{}{l\gamma\sigma_{ran(\gamma)} \to r\gamma\sigma_{ran(\gamma)}/ST}$ in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$, so $T' = \frac{}{t \to v/ST}$.
  - if $ST =$ `match` u `s.t.` $\bigwedge_{j=1}^m (l_j = r_j) \wedge \phi$ then there exists a substitution δ such that $t =_{\mathcal{E}} u\sigma\delta$, $l_j\sigma\delta =_{\mathcal{E}} r_j\sigma\delta$, for $1 \leq j \leq m$, and $E_0 \vdash \phi\sigma\delta$, so there are derivation rules $\frac{}{w \to w/ST\sigma}$ and $\frac{}{w \to w/ST}$ in $\mathcal{D}_{\mathcal{R},Call_{\mathcal{R}}}$, where $w =_{\mathcal{E}} u\sigma\delta$, and $T = \frac{}{t \to v/ST\sigma}$ because $t =_{\mathcal{E}} w =_{\mathcal{E}} v$, so also $T' = \frac{}{t \to v/ST}$.
- Inductive step:
  - $ST = ST_1;ST_2$ and T has the form $\frac{\frac{T_1}{t \to w/ST_1\sigma} \quad \frac{T_2}{w \to v/ST_2\sigma}}{t \to v/ST\sigma}$. By I.H there are closed proof trees with the forms $\frac{T'_1}{t \to w/ST_1}$ and $\frac{T'_2}{w \to v/ST_2}$ where T'_1 and T'_2 have the same depth and number of nodes as T_1 and T_2, respectively, so $T' = \frac{\frac{T'_1}{t \to w/ST_1} \quad \frac{T'_2}{t \to w/ST_2}}{t \to v/ST}$ is a c.p.t. with the same depth and number of nodes as T.
  - $ST = ST_1+$ and T must be either of the form $\frac{\frac{T_1}{t \to v/ST_1\sigma}}{t \to v/ST\sigma}$ or $\frac{\frac{T_2}{t \to v/ST_1\sigma \; ; \; ST_1\sigma+}}{t \to v/ST\sigma}$. As $ST_1\sigma \; ; ST_1\sigma+ = (ST_1 \; ; ST_1+)\sigma$ then, by I.H., there is either a c.p.t. with the form $\frac{T'_1}{t \to v/ST_1}$ or $\frac{T'_2}{t \to v/ST_1 \; ; ST_1+}$, hence either $T' = \frac{\frac{T'_1}{t \to v/ST_1}}{t \to v/ST}$ or $T' = \frac{\frac{T'_2}{t \to v/ST_1 \; ; ST_1+}}{t \to v/ST}$.

- $ST = ST_1 \,|\, ST_2$ and T must be either of the form $\frac{\frac{T_1}{t\to v/ST_1\sigma}}{t\to v/ST\sigma}$ or $\frac{\frac{T_2}{t\to v/ST_2\sigma}}{t\to v/ST\sigma}$. Then, by I.H., there is either a c.p.t. with the form $\frac{T_1'}{t\to v/ST_1}$ or $\frac{T_2'}{t\to v/ST_2}$, hence either $T' = \frac{\frac{T_1'}{t\to v/ST_1}}{t\to v/ST}$ or $T' = \frac{\frac{T_2'}{t\to v/ST_2}}{t\to v/ST}$.
- $ST = \mathtt{match}\ u\ \mathtt{s.t.}\ \phi ?\, ST_1 : ST_2$ and T must be either of the form $\frac{\frac{T_1}{t\to v/ST_1\sigma\delta}}{t\to v/ST\sigma}$ or $\frac{\frac{T_2}{t\to v/ST_2\sigma\delta}}{t\to v/ST\sigma}$ where $\delta : V_{u\sigma,\phi\sigma} \to \mathcal{T}_\Sigma$, $t =_\mathcal{E} u\sigma\delta$, and either $E_0 \vdash \phi\sigma\delta$ or $E_0 \vdash \neg\phi\sigma\delta$, respectively.
  Let $\alpha = \sigma_{V_{u,\phi}}$, so $dom(\delta) = V_{u,\phi} \setminus dom(\alpha)$, and $\beta = \sigma_{\setminus V_{u,\phi}}$, so $dom(\delta) \cap dom(\beta) = \emptyset$. Then $\sigma = \alpha \uplus \beta$, $(u\sigma\delta, \phi\sigma\delta) = (u\alpha\delta, \phi\alpha\delta)$, so $E_0 \vdash \phi\sigma\delta$ iff $E_0 \vdash \phi\alpha\delta$, and $\alpha\delta : V_{u,\phi} \to \mathcal{T}_\Sigma$, so there is a derivation rule of the form $\frac{t\to v/ST_1\alpha\delta}{t\to v/ST}$ or $\frac{t\to v/ST_2\alpha\delta}{t\to v/ST}$ in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$. Consider the open goal $t \to v/(ST_i\alpha\delta)\beta$, where $i = 1$ if $E_0 \vdash \phi\alpha\delta$ and $i = 2$ if $E_0 \vdash \neg\phi\alpha\delta$. As δ is ground and $dom(\delta) \cap dom(\beta) = \emptyset$ then $\alpha\delta\beta = \alpha\beta\delta = \sigma\delta$ and $\frac{T_i}{t\to v/(ST_i\alpha\delta)\beta}$ is a c.p.t. so, by I.H., there is a c.p.t. with the form $\frac{T_i'}{t\to v/ST_i\alpha\delta}$, where T_i' has the same depth and number of nodes as T_i, and $T' = \frac{\frac{T_i'}{t\to v/ST_i\alpha\delta}}{t\to v/ST}$.
- $ST = CS$, where $\mathtt{sd}\ CS := ST_1 \in Call_\mathcal{R}$ and T has the form $\frac{\frac{T_1}{t\to v/ST_1\gamma}}{t\to v/ST\sigma}$, for some renaming γ, because $ST\sigma = CS\sigma = CS = ST$, so $T' = \frac{\frac{T_1}{t\to v/ST_1\gamma}}{t\to v/ST}$.
- $ST = CS(\bar{t})$, where $\mathtt{sd}\, CS(\bar{x}) := ST_1 \in Call_\mathcal{R}$, $\bar{x} = x^1_{s_1}, \ldots, x^n_{s_n}$, $\bar{t} = t_1, \ldots, t_n$, and $\rho = \{\bar{x} \mapsto \bar{t}\}$, let $\rho' = \{\bar{x} \mapsto \bar{t}\sigma\}$, and T has the form $\frac{\frac{T_1}{t\to v/ST_1(\gamma\cup\rho')}}{t\to v/ST\sigma}$, because $ST_1\sigma = CS(\bar{t})\sigma = CS(\bar{t}\sigma)$, and for some renaming γ such that $dom(\gamma) \subseteq V_{ST_1} \setminus \hat{x}$ and $ran(\gamma)$ is away from any known variable, so $V_{ST_1} = \bar{x} \cup ran(\gamma)$. As we also have $dom(\rho') = dom(\rho) = \hat{x}$, then $ST_1(\gamma\cup\rho') = ST_1\gamma\rho' = ST_1\gamma\rho\sigma$ and also $ST_1(\gamma\cup\rho) = ST_1\gamma\rho$. As $\frac{T_1}{t\to v/ST_1\gamma\rho\sigma}$ is a c.p.t. then, by I.H., there is a c.p.t. $\frac{T_1'}{t\to v/ST_1\gamma\rho}$, and $T' = \frac{\frac{T_1'}{t\to v/ST_1(\gamma\cup\rho)}}{t\to v/ST}$.
- $ST = CS(\bar{t})$, where $\mathtt{csd}\ CS(\bar{x}) := ST_1\ \mathit{if}\ C \in Call_\mathcal{R}$, with $\bar{x} = x^1_{s_1}, \ldots, x^n_{s_n}$ and $C = \bigwedge_{j=1}^m (l_j = r_j) \wedge \phi$, $\hat{x} \subseteq V_{CS}$, $\bar{t} = t_1, \ldots, t_n$, and $\rho = \{\bar{x} \mapsto \bar{t}\}$, let $\rho' = \{\bar{x} \mapsto \bar{t}\sigma\}$, and T has the form $\frac{\frac{T_1}{t\to v/ST_1(\gamma\cup\rho')\delta}}{t\to v/ST\sigma}$, because $ST_1\sigma = CS(\bar{t})\sigma = CS(\bar{t}\sigma)$, and for some renaming γ such that $dom(\gamma) \subseteq V_{ST_1} \setminus \hat{x}$ and $ran(\gamma)$ is away from any known variable, so $V_{ST_1} = \bar{x}\cup ran(\gamma)$, and there is a substitution $\delta : vars(CS(\gamma\cup\rho')) \to \mathcal{T}_\Sigma$ such that $l_j(\gamma\cup\rho')\delta =_\mathcal{E} r_j(\gamma\cup\rho')\delta$, for $1 \le j \le n$, and $E_0 \vdash \phi(\gamma\cup\rho')\delta$.
  Let $\delta' = \delta_{ran(\gamma)} \cup (\sigma\delta_{\setminus ran(\gamma)})$. As δ is ground and $ran(\gamma)$ is away from all known variables, then $(\gamma\cup\rho)\delta' = (\gamma\cup\rho)\delta_{ran(\gamma)} \cup (\sigma\delta_{\setminus ran(\gamma)}) = (\gamma\delta_{ran(\gamma)}) \cup (\rho\sigma\delta_{\setminus ran(\gamma)})) = (\gamma\delta_{ran(\gamma)}) \cup (\rho'\delta_{\setminus ran(\gamma)})) = (\gamma\cup\rho')\delta$, so $\delta' : vars(C(\gamma\cup\rho)) \to \mathcal{T}_\Sigma$ verifies $l_j(\gamma\cup\rho)\delta' =_\mathcal{E} r_j(\gamma\cup\rho)\delta'$, for $1 \le j \le n$, and $E_0 \vdash \phi(\gamma\cup\rho)\delta'$, and there is a derivation rule $\frac{t\to v/ST_1(\gamma\cup\rho)\delta'}{t\to v/ST}$ in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$. Since $(\gamma\cup\rho)\delta' = (\gamma\cup\rho')\delta$, then $T' = \frac{\frac{T_1}{t\to v/ST_1(\gamma\cup\rho)\delta'}}{t\to v/ST}$.
- $ST = c[\gamma]\{ST_1, \ldots, ST_m\}$. As $dom(\sigma) \cap V_R = \emptyset$ then $c(\gamma\sigma)_{dom(\gamma)} = c\gamma\sigma_{ran(\gamma)}$. $c : l \to r\ \mathit{if}\ \bigwedge_{j=1}^m l_j \to r_j \mid \psi$ is a rule in R and T has the form $\frac{T_1 \cdots T_m}{t\to v/ST\sigma}$,

where T_i has the form $\frac{T'_i}{l_i\gamma\delta'\to r_i\gamma\delta'/ST_i\sigma\delta}$, for $1 \le i \le m$, $\delta : vars(c\gamma\sigma_{ran(\gamma)}) \to \mathcal{T}_\Sigma$, $\delta' = \sigma_{ran(\gamma)}\delta$, $E_0 \vdash \psi\gamma\sigma_{ran(\gamma)}\delta$, and there are u in $\mathcal{H}_\Sigma$ and p in $pos(u)$ such that $t =_\mathcal{E} u$, $u|_p = l\gamma\sigma_{ran(\gamma)}\delta$, and $u[r\gamma\sigma_{ran(\gamma)}\delta]_p =_\mathcal{E} v$.

As $\delta' = \sigma_{ran(\gamma)}\delta$, then $\delta' : vars(c\gamma) \to \mathcal{T}_\Sigma$, and $E_0 \vdash \psi\gamma\delta'$, $u|_p = l\gamma\delta'$, so there is a derivation rule $\frac{l_1\gamma\delta'\to r_1\gamma\delta'/ST_1\delta'\cdots l_m\gamma\delta'\to r_m\gamma\delta'/ST_m\delta'}{u\to u[r\gamma\delta']_p/ST}$ in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$. Also $u[r\gamma\delta']_p = u[r\gamma\sigma_{ran(\gamma)}\delta]_p =_\mathcal{E} v$.

As $dom(\sigma) \cap V_R = \emptyset$ and $dom(\delta) \subseteq V_c \cup ran(\gamma) \subseteq V_R \cup V_{ST}$ then $dom(\delta) \cap dom(\sigma_{\setminus ran(\gamma)}) = \emptyset$ so, as $ran(\sigma) \cap (V_R \cup V_{ST}) = \emptyset$ and δ is ground, $\sigma_{\setminus ran(\gamma)}\delta = \delta\sigma_{\setminus ran(\gamma)}$ and $\sigma\delta = (\sigma_{ran(\gamma)} \uplus \sigma_{\setminus ran(\gamma)})\delta = \sigma_{ran(\gamma)}\sigma_{\setminus ran(\gamma)}\delta = \sigma_{ran(\gamma)}\delta\sigma_{\setminus ran(\gamma)} = \delta'\sigma_{\setminus ran(\gamma)}$, hence, for $1 \le i \le m$, $T_i = \frac{T'_i}{l_i\gamma\delta'\to r_i\gamma\delta'/ST_i\delta'\sigma_{\setminus ran(\gamma)}}$, and, by I.H., there is a c.p.t. T''_i with the form $\frac{T'''_i}{l_i\gamma\delta'\to r_i\gamma\delta'/ST_i\delta'}$ and the same depth and number of nodes as T_i. Then, as $t =_\mathcal{E} u$ and $u[r\gamma\delta']_p =_\mathcal{E} v$, $T' = \frac{T''_1\cdots T''_m}{t\to v/ST}$.

– $ST = \mathtt{top}(c[\gamma]\{ST_1, \ldots, ST_m\})$. As $dom(\sigma) \cap V_R = \emptyset$ then $c(\gamma\sigma)_{dom(\gamma)} = c\gamma\sigma_{ran(\gamma)}$. $c : l \to r$ *if* $\bigwedge_{j=1}^m l_j \to r_j \mid \psi$ is a rule in R and T has the form $\frac{T_1\cdots T_m}{t\to v/ST\sigma}$, where T_i has the form $\frac{T'_i}{l_i\gamma\delta'\to r_i\gamma\delta'/ST_i\sigma\delta}$, for $1 \le i \le m$, $\delta : vars(c\gamma\sigma_{ran(\gamma)}) \to \mathcal{T}_\Sigma$, $\delta' = \sigma_{ran(\gamma)}\delta$, $E_0 \vdash \psi\gamma\sigma_{ran(\gamma)}\delta$, $t =_\mathcal{E} l\gamma\sigma_{ran(\gamma)}\delta$, and $r\gamma\sigma_{ran(\gamma)}\delta =_\mathcal{E} v$.

As $\delta' = \sigma_{ran(\gamma)}\delta$, then $\delta' : vars(c\gamma) \to \mathcal{T}_\Sigma$ and $E_0 \vdash \psi\gamma\delta'$, then there is a derivation rule $\frac{l_1\gamma\delta'\to r_1\gamma\delta'/ST_1\delta'\cdots l_m\gamma\delta'\to r_m\gamma\delta'/ST_m\delta'}{l\gamma\delta'\to r\gamma\delta'/ST}$ in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$. Also $t =_\mathcal{E} l\gamma\sigma_{ran(\gamma)}\delta = l\gamma\delta'$ and $r\gamma\delta' = r\gamma\sigma_{ran(\gamma)}\delta =_\mathcal{E} v$.

As in the previous case, for $1 \le i \le m$ there is a c.p.t. T''_i with the form $\frac{T'''_i}{l_i\gamma\delta'\to r_i\gamma\delta'/ST_i\delta'}$ and the same depth and number of nodes as T_i. Then, as $t =_\mathcal{E} l\gamma\delta'$ and $r\gamma\delta' =_\mathcal{E} v$, $T' = \frac{T''_1\cdots T''_m}{t\to v/ST}$.

– $ST = \mathtt{matchrew}\ u\ \mathtt{s.t.}\ \bigwedge_{j=1}^m (l_j = r_j) \wedge \phi\ \mathtt{by}\ x^1_{s_1}\ \mathtt{using}\ ST_1, \ldots, x^n_{s_n}\ \mathtt{using}\ ST_n$, where $u = u[x^1_{s_1}, \ldots, x^n_{s_n}]_{p_1\ldots p_n}$ and T has the form $\frac{T_1\cdots T_m}{t\to v/ST\sigma}$, where T_i has head $x^i_{s_i}\delta \to t_i/ST_i\sigma\delta$, for $1 \le i \le n$, with $\hat{t} \subset \mathcal{T}_\Sigma$, $\delta_{V_{ST\sigma}} : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X}\setminus V_{ST\sigma})$ such that, $ran(\delta_{V_{ST\sigma}}) \subseteq V_{\overline{ST}\sigma\delta}$, $t =_\mathcal{E} u\sigma\delta \in \mathcal{T}_\Sigma$, $u\sigma\delta[\bar{t}]_{\bar{p}} =_\mathcal{E} v$, $\{l_j\sigma\delta, r_j\sigma\delta\}_{j=1}^m \subset \mathcal{T}_\Sigma$, $\bar{l}\sigma\delta =_\mathcal{E} \bar{r}\sigma\delta$, $\phi\sigma\delta \in \mathcal{T}_\Sigma$, and $E_0 \vdash \phi\sigma\delta$.

The fact that $ran(\delta_{V_{ST\sigma}}) \subseteq V_{\overline{ST}\sigma\delta}$ does not ensure that $ran(\delta_{V_{ST\sigma}}) \cap V_{ST} = \emptyset$. Let α be a renaming such that $dom(\alpha) = V_{\overline{ST}} \cap ran(\delta_{V_{ST\sigma}})$ and $ran(\alpha)$ is away from all known variables and let $\delta' = \sigma\delta\alpha$. Then $\delta'_{V_{ST}} : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X}\setminus V_{ST})$. By Lemma 12, as T_i has head $x^i_{s_i}\delta \to t_i/ST_i\sigma\delta$, there is also a c.p.t. with the form $\frac{T'_i}{x^i_{s_i}\delta\to t_i/ST_i\delta'}$, for $1 \le i \le n$.

As $\delta'_{V_{ST}} : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X}\setminus V_{ST})$, $t =_\mathcal{E} u\delta = u\delta' \in \mathcal{T}_\Sigma$, $u\delta'[\bar{t}]_{\bar{p}} =_\mathcal{E} v$, $\bar{l}\delta' = \bar{l}\delta =_\mathcal{E} \bar{r}\delta = \bar{r}\delta'$, so $\{l_j\delta', r_j\delta'\}_{j=1}^m \subset \mathcal{T}_\Sigma$, and $\phi\delta' = \phi\delta \in \mathcal{T}_\Sigma$, so $E_0 \vdash \phi\delta'$ then there is a derivation rule $\frac{x^1_{s_1}\delta'\to t_1/ST_1\delta'\cdots x^n_{s_n}\delta'\to t_n/ST_n\delta'}{u\delta'\to u\delta'[\bar{t}]_{\bar{p}}/ST}$ in $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}$. As $u\delta'[\bar{x}\delta']_{\bar{p}} = u\delta' = u\delta = u\delta[\bar{x}\delta]_{\bar{p}}$, so also $\bar{x}\delta' = \bar{x}\delta'$, the derivation rule can be written $\frac{x^1_{s_1}\delta\to t_1/ST_1\delta'\cdots x^n_{s_n}\delta\to t_n/ST_n\delta'}{u\delta\to u\delta[\bar{t}]_{\bar{p}}/ST}$, hence $T' = \frac{\frac{T'_1}{x^1_{s_1}\delta\to t_1/ST_1\delta'}\cdots\frac{T'_n}{x^n_{s_n}\delta\to t_n/ST_n\delta'}}{u\delta\to u\delta[\bar{t}]_{\bar{p}}/ST}$.

□

Proposition 16 (Equality of $(R^\sigma)_B$ and $(R_B)^\sigma$)**.** *For any rewrite theory* $\mathcal{R} = (\Sigma, \mathcal{E}, R)$

and any substitution σ *such that* $vars(B) \cap (dom(\sigma) \cup ran(\sigma)) = \emptyset$, *it holds that* $(R^\sigma)_B = (R_B)^\sigma$.

Proof. We prove $(c^\sigma)_B = (c_B)^\sigma$ for every rule $c \in R$. If $c : l \to r \ if \ C \in R$ then, by definition, $l \in \mathcal{H}_\Sigma(\mathcal{X}) \setminus \mathcal{X}$, so l has the form $f(\bar{l})$, for appropriate f and $\bar{l}$.

- If f is binary associative then c has the form $c : f(l_1, l_2) \to r \ if \ C \in R$, and $c : f(x_s, f(l_1, l_2)) \to r \ if \ C \in c_B$, so $c : f(x_s, f(l_1\sigma, l_2\sigma)) \to r\sigma \ if \ C\sigma \in (c_B)^\sigma$ since $x_s\sigma = x_s$. Then, $c : f(l_1\sigma, l_2\sigma) \to r\sigma \ if \ C\sigma \in R^\sigma$, so also $c : f(x_s, f(l_1\sigma, l_2\sigma)) \to r\sigma \ if \ C\sigma \in (c^\sigma)_B$, and $(c^\sigma)_B = (c_B)^\sigma$.
- Else, $c_B = \{c\}$, and $(c_B)^\sigma = \{c^\sigma\}$. Now, c^σ has the form $c : f(\bar{l}\sigma) \to r\sigma \ if \ C\sigma$ where f is not binary associative, so also $(c^\sigma)_B = \{c^\sigma\}$, hence $(c^\sigma)_B = (c_B)^\sigma$.

□

Proposition 20 (Invariants of the goals)**.** *Given a rewrite theory* $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ *and a set of call strategy definitions* $Call_\mathcal{R}$, *and an admissible goal* G *with the form*

- $\bigwedge_{i=1}^n u_i \to v_i / ST_i^\nu \varrho_\nu \mid \phi \mid V, \nu$, *or*
- $u_1|_p \to^1 x_k, u_1[x_k]_p \to v_1 / ST_1^\nu \varrho_\nu \wedge \bigwedge_{i=2}^n u_i \to v_i / ST_i^\nu \varrho_\nu \mid \phi \mid V, \nu$,

if G_0 *is a goal of type (a), with substitution* ν_0 ($\varrho_{\nu_0} = none$ *by definition), and* $G_0 \rightsquigarrow_\theta^* G$ *then the following invariants hold:*

1. $vars(B) \cap V = \emptyset$ *and* $V_\mathcal{R} \cap V_{Call_\mathcal{R}} \subseteq V$,
2. $V \cap ran(\nu) = \emptyset$ *and* $\nu = (\nu_0\theta)_V$, *hence* $dom(\nu) \subseteq V$, *so* $dom(\nu)$ *satisfies the restrictions given for* V *in Definition 21.2,*
3. $\varrho_\nu = \theta_{\setminus V}$, *hence* $dom(\varrho_\nu) \cap V = \emptyset$ *and* ϱ_ν *is idempotent,*
4. $ran(\theta) \cap (V \cup V_{\mathcal{R}, Call_\mathcal{R}} \cup vars(\overline{ST})) = \emptyset$ *and* $ran(\varrho_\nu) \cap V = \emptyset$,
5. $dom(\varrho_\nu) \cap ran(\nu) = \emptyset$,
6. $dom(\varrho_\nu) \cap V^\nu = \emptyset$,
7. $V_{\mathcal{R}^\nu} \cap V_{Call_{\mathcal{R}^\nu}} \subseteq V^\nu$,
8. *if* $t \in \mathcal{T}_\Sigma(\mathcal{X})$ *then* $t^\nu \varrho_\nu = t(\nu \uplus \varrho_\nu)$,
9. u_i, v_i, $1 \leq i \leq n$, *and each term in* $\hat{\phi}$ *have the form* $t^\nu \varrho_\nu$,
10. $vars(\bar{u}, \bar{v}, \phi) \cap dom(\nu) = \emptyset$, *and*
11. G *has also the form* $G_1^\nu \varrho'_\nu$, *where* $\varrho'_\nu = \theta_{V_{G_1} \setminus V}$, *so* $dom(\varrho'_\nu) \subseteq V_{G_1} \setminus V$.

Proof. By induction on the number of applied calculus rules from Figures 6.2, 6.3, and 6.4. We consider the two types of goals separately in this proof.

- If G is a goal of type (a) then we have that $G = G_0$, $\theta = none$, and $\varrho_\nu = \varrho_{\nu_0} = none$. The invariants $1 - 7$ and 11 are direct consequences of the definitions of reachability problem and goal of type (a), and the fact if $\theta = \sigma_1 \ldots \sigma_m$ then $ran(\sigma_i)$ is away from any known variable, for $1 \leq i \leq m$, by the definition of the calculus rules. We prove invariants $8 - 10$.

8. As $\varrho_\nu = none$, then $t^\nu \varrho_\nu = t^\nu = t\nu = t(\nu \uplus \varrho_\nu)$.

9. We have to prove $w \in \hat{u} \cup \hat{v} \cup \hat{\phi} \implies \exists t, , w = t^\nu \varrho_\nu$. As, by the previous point, $t^\nu \varrho_\nu = t\nu$, then we prove $w \in \hat{u} \cup \hat{v} \cup \hat{\phi} \implies \exists t, , w = t\nu$. Now, as G is a goal of type (a), G has the form $\bigwedge_{i=1}^{n} u_i^0 \nu \to v_i^0 \nu / ST_i^\nu \mid \phi^0 \nu \mid V, \nu$, so $\bar{u} = \bar{u}^0 \nu$, $\bar{v} = \bar{v}^0 \nu$, $\bar{\phi} = \bar{\phi}^0 \nu$, hence $w \in \hat{u} \cup \hat{v} \cup \hat{\phi} \implies \exists t, , t \in \hat{u}^0 \cup \hat{v}^0 \cup \hat{\phi}^0 \wedge w = t\nu$.

10. As G is a goal of type (a) then $dom(\nu) \cap ran(\nu) = \emptyset$. By the previous point, there exists $\hat{u}^0 \cup \hat{v}^0 \cup \hat{\phi}^0$ such that $\hat{u} \cup \hat{v} \cup \hat{\phi} = \hat{u}^0 \nu \cup \hat{v}^0 \nu \cup \hat{\phi}^0 \nu$. As $dom(\nu) \cap ran(\nu) = \emptyset$ then $vars(\bar{u}^0 \nu, \bar{v}^0 \nu, \phi^0 \nu) \cap dom(\nu) = \emptyset$, i.e., $vars(\bar{u}, \bar{v}, \phi) \cap dom(\nu) = \emptyset$.

- We prove the invariants for goals of type (b) by induction on the number of applied calculus rules from Figures 6.2, 6.3, and 6.4 in $G_0 \rightsquigarrow_{\sigma'}^* G' \rightsquigarrow_{[r],\sigma} G$, so $\theta = \sigma'\sigma$, using the fact that the properties hold in G'. We call $\bar{u}'$, $\bar{v}'$, ϕ', ν', and $\overline{ST'}$ the structures in G' in place of $\bar{u}$, $\bar{v}$, ϕ, ν, and $\overline{ST}$, so either $\nu = \nu'$ or there is a substitution σ such that $\nu = (\nu'\sigma)_V$ where, for appropriate t_1 and t_2, $\sigma \in CSU_B(t_1, t_2)$ so $V \cap ran(\sigma) = \emptyset$ by definition of CSU_B. Also, as $dom(\nu_0) \cap ran(\nu_0) = \emptyset$, $\nu = (\nu_0\theta)_V$ and θ is a composition of several CSUs, so $ran(\theta)$ is away from all known variables, then $dom(\nu) \cap ran(\nu) = \emptyset$ and, as $V^\nu = (V \setminus dom(\nu)) \cup ran(\nu)$, also $dom(\nu) \cap V^\nu = \emptyset$.

1. Immediate, since the invariant holds in G', by I.H, and no rule modifies V.

2. As either $\nu = \nu'$ or $\nu = (\nu'\sigma)_V$, $V \cap ran(\sigma) = \emptyset$, and $V \cap ran(\nu') = \emptyset$, by I.H., then $V \cap ran(\nu) = \emptyset$ in either case. Also, by I.H., $\nu' = (\nu_0\sigma')_V$, so $\nu = (\nu'\sigma)_V = ((\nu_0\sigma')_V\sigma)_V = (\nu_0\sigma'\sigma)_V = (\nu_0\theta)_V$.

3. By I.H., $\varrho_{\nu'} = \sigma'_{\setminus V}$, with $dom(\varrho_{\nu'}) \cap (V \cup ran(\nu')) = \emptyset$ and $ran(\varrho_{\nu'}) \cap V = \emptyset$, i.e., $ran(\sigma'_{\setminus V}) \cap V = \emptyset$. Then:

   - If $[r]$ computes a CSU_B of two terms, say σ, then we can find in G (depending on the actual calculus $[r]$ applied):
     - open goals that are an instance with σ of one open goal in G' with the form $u' \to v'/ST^{\nu'}\varrho_{\nu'}$. The strategy of one open goal in G will be an instance with σ of part of $ST^{\nu'}\varrho_{\nu'}$ in the case of rules if then else and match,
     - new open goals with the form $(u \to v/\texttt{idle})\sigma$ which are equal to $(u \to v/\texttt{idle}\ \varrho_{\nu'})\sigma$, or
     - new open goals with the form $(u \to v/ST\varrho_{\nu'}; \texttt{idle})\sigma$, where $ST\varrho_{\nu'}$ is an already existing strategy in G', which are equal to $(u \to v/(ST; \texttt{idle})\varrho_{\nu'})\sigma$.

     In any of these cases, by Def. 24, $\varrho_\nu = (\varrho_{\nu'}\sigma)_{\setminus V}$, hence $\varrho_\nu = (\varrho_{\nu'}\sigma)_{\setminus V} = (\sigma'_{\setminus V}\sigma)_{\setminus V} = (\sigma'\sigma)_{\setminus V} = \theta_{\setminus V}$.
   - If $[r]$ is a `call strategy` rule, applied to a open goal with the form $u' \to v'/CS\ ;\ ST^{\nu'}\varrho_{\nu'}$ or $u' \to v'/CS(\bar{t}^{\nu'}\varrho_{\nu'})\ ;\ ST^{\nu'}\varrho_{\nu'}$, where CS has parameters $\bar{x}$, then $\sigma = none$, $\nu = \nu'$, $\varrho_\nu = \varrho_{\nu'} = \sigma'_{\setminus V} = (\sigma'\sigma)_{\setminus V} = \theta_{\setminus V}$, and $dom(\varrho_\nu) \cap (V \cup ran(\nu)) = \emptyset$. Apart from the rest of existing open goals, that remain unchanged, we can find in G:
     - for conditional call strategies, new open goals with the form $u \to v/\texttt{idle}$ which are equal to $u \to v/\texttt{idle}\ \varrho_\nu$, and

- a new open goal $u \to v/ST_2^\nu\gamma\,;\, ST^\nu\varrho_\nu$, where if the call strategy has no parameters then: (i) $\gamma = none$, let $\gamma_0 = none$, or else (ii) $\gamma = \{\bar{x} \mapsto \bar{t}^\nu\varrho_\nu\}$, let $\gamma_0 = \{\bar{x} \mapsto \bar{t}\}$, and ST_2^ν is a fresh version of the strategy ST_1^ν in the call strategy definition for CS in $Call_{\mathcal{R}}^\nu$, except for $dom(\gamma) \cup V^\nu$. As $dom(\varrho_\nu) \cap (V \cup ran(\nu)) = \emptyset$ and $vars(ST_2^\nu) \cap dom(\nu) = \emptyset$ then $vars(ST_2^\nu) \cap dom(\varrho_\nu) = \emptyset$ so, if either (i) or (ii) holds, $ST_2^\nu\gamma = (ST_2\gamma_0)^\nu\varrho_\nu$.

- $\sigma = none$ for the rest of the rules, so $\nu = \nu'$ and $\varrho_\nu = \varrho_{\nu'} = \sigma'_{\setminus V} = (\sigma'\sigma)_{\setminus V} = \theta_{\setminus V}$, and no new strategies are added in G, let $tokens(ST+) = tokens(ST)$, $tokens(ST_1\, op\, ST_2) = tokens(ST_1) \cup tokens(ST_2)$ if op is a binary combinator, and $tokens(ST) = ST$ otherwise. In these rules, for any open goal $u' \to v'/ST_G \in G$ there is one open goal $u' \to v'/ST^{\nu'}\varrho_{\nu'} \in G'$ such that if $ST_1 \in tokens(ST_G)$ then $ST_1 \in tokens(ST^{\nu'}\varrho_{\nu'})$, so ST_1 has the form $ST_2^{\nu'}\varrho_{\nu'}$, i.e., $ST_2^\nu\varrho_\nu$.

4. Immediate, since θ is a composition of several $CSUs$, where the range of each CSU is away from all known variables (so $dom(\theta) \cap ran(\theta) = \emptyset$), including V, and, by the previous point, $\varrho_\nu = \theta_{\setminus V}$.

5. If $\sigma = none$ there is nothing to prove. Else, as $\varrho_\nu = (\varrho_{\nu'}\sigma)_{\setminus V}$ and $\nu = (\nu'\sigma)_V$ then $dom(\varrho_\nu) = dom(\varrho_{\nu'}) \cup (dom(\sigma) \setminus (V \cup ran(\varrho_{\nu'}))$ and $ran(\nu) = ran(\sigma_V) \cup (ran(\nu') \setminus dom(\sigma))$.

   As $ran(\sigma)$ is away from all known variables and, by I.H., $dom(\varrho_{\nu'}) \cap ran(\nu') = \emptyset$ then

   $dom(\varrho_\nu) \cap ran(\nu) =$

   $(dom(\varrho_{\nu'}) \cup (dom(\sigma) \setminus (V \cup ran(\varrho_{\nu'}))) \;\cap\; (ran(\sigma_V) \cup (ran(\nu') \setminus dom(\sigma))) =$

   $(dom(\varrho_{\nu'}) \cup (dom(\sigma) \setminus (V \cup ran(\varrho_{\nu'}))) \;\cap\; (ran(\nu') \setminus dom(\sigma)) =$

   $(dom(\sigma) \setminus (V \cup ran(\varrho_{\nu'})) \cap (ran(\nu') \setminus dom(\sigma)) \subseteq dom(\sigma) \cap (ran(\nu') \setminus dom(\sigma)) = \emptyset.$

6. As $dom(\varrho_\nu) \cap V = \emptyset$, $dom(\varrho_\nu) \cap ran(\nu) = \emptyset$, and $V^\mu \subseteq V \cup ran(\nu)$, then $dom(\varrho_\nu) \cap V^\nu = \emptyset$.

7. Immediate, since $V_{\mathcal{R}} \cap V_{Call_{\mathcal{R}}} \subseteq V$, in $\mathcal{R}$ and $Call_{\mathcal{R}}$ we are replacing each variable $v \in dom(\nu)$ with $v\nu$, and $V^\nu = ran(\nu) \cup (V \setminus dom(\nu))$.

8. Immediate, since $dom(\nu) \subseteq V$ and $dom(\varrho_\nu) \cap (V \cup ran(\nu)) = \emptyset$, invariant 5, imply $t^\nu\varrho_\nu = t(\nu \uplus \varrho_\nu)$

9. Let $w \in \bar{u}' \cup \bar{v}' \cup \phi'$ such that $w\sigma \in \bar{u} \cup \bar{v} \cup \phi$. By I.H., $w = t^{\nu'}\varrho_{\nu'}$, for appropriate t. By I.H. and the previous point, $w = t(\nu' \uplus \varrho_{\nu'})$. As, by I.H., $dom(\nu') \subseteq V$ and $dom(\varrho_{\nu'}) \cap V = \emptyset$, then $w\sigma = t(\nu' \uplus \varrho_{\nu'})\sigma = t(\nu'_V \uplus (\varrho_{\nu'})_{\setminus V})\sigma = t((\nu'\sigma)_V \uplus (\varrho_{\nu'}\sigma)_{\setminus V}) = t(\nu \uplus \varrho_\nu)$ so, by the previous point, $w\sigma = t^\nu\varrho_\nu$.

10. By I.H., $vars(\bar{u}', \bar{v}', \phi') \cap dom(\nu') = \emptyset$, with $dom(\nu') \subseteq V$. As $\nu = (\nu'\sigma)_V$, then $dom(\nu) = dom(\nu') \cup dom(\sigma_V)$, so $vars(\bar{u}'\sigma, \bar{v}'\sigma, \phi'\sigma) \cap dom(\nu) = \emptyset$. Then we only have to check $vars(\bar{u}, \bar{v}, \phi) \setminus vars(\bar{u}'\sigma, \bar{v}'\sigma, \phi')\sigma$, i.e., those variables introduced by the rule that do not belong to the instantiation of $vars(\bar{u}', \bar{v}', \phi')$ with σ.

    - Each one of the variables, say x, introduced by $abstract_{\Sigma_1}$ is new so, as $\nu = (\nu'\sigma)_V$:

  - if $x \in dom(\sigma)$ then $vars(x\sigma) \cap dom(\nu) \subseteq ran(\sigma) \cap dom(\nu) \subseteq ran(\sigma) \cap V = \emptyset$, and
  - if $x \notin dom(\sigma)$ then, as x is new (so $x \notin V$), $vars(x\sigma) \cap dom(\nu) = \{x\} \cap dom(\nu) \subseteq \{x\} \cap V = \emptyset$ (†).

  This covers all the rules in Figure 6.2. It also covers rule match and it partially covers the rest of rules in Figures 6.3 and 6.4.

- Both rules transitivity and congruence introduce one new variable not in $dom(\nu)$, so (†) applies ($\sigma = none$).
- Rule matchrew introduces one vector of new variables ($\bar{y}$) not in $dom(\nu)$, so (†) applies.
- The next case is rule rule application, with strategy $c[\gamma]\{\overline{ST}\}$ and substitution σ. By I.H. $c[\gamma]\{\overline{ST}\}$ has the form $(c[\delta]\{\overline{ST'}\})^{\nu'}\varrho_{\nu'}$, for appropriate δ, so $c[\gamma] = c^{\nu'}[\delta(\nu' \uplus \varrho_{\nu'})_{ran(\delta)}]$, where $dom(\delta) = dom(\gamma)$. The calculus rule uses a version, say $c_1^{\nu'}$, of $c^{\nu'}$ where all the variables are new except for $dom(\gamma) \cup V^{\nu'}$. The new variables of $vars(c_1^{\nu'})$ are not in $dom(\nu)$, so (†) applies. We check the rest of the variables in $vars(c_1^{\nu'})$. For each $x \in vars(c_1^{\nu'}) \cap (dom(\gamma) \cup V^{\nu'})$:
  - if $x \in V^{\nu'}$ then:
    - if $x \in dom(\sigma)$ then $vars(x\sigma) \subseteq ran(\nu)$ so, as $dom(\nu) \cap ran(\nu) = \emptyset$ then $vars(x\sigma) \cap dom(\nu) = \emptyset$;
    - else $x\sigma = x$, so $x \in V^{\nu}$, and:
      - if $x \in V$ then $x \notin dom(\nu)$ so, as $x\sigma = x$, $vars(x\sigma) \cap dom(\nu) = \emptyset$;
      - else $x \in ran(\nu')$ so, as $x\sigma = x$, $x \in ran(\nu)$. Then, as $x\sigma = x$ and $dom(\nu) \cap ran(\nu) = \emptyset$, $vars(x\sigma) \cap dom(\nu) = \emptyset$;
  - else $x \in dom(\gamma)$ $(= dom(\delta))$, and $x\gamma\sigma = x\delta(\nu' \uplus \varrho_{\nu'})_{ran(\delta)}\sigma = x\delta(\nu \uplus \varrho_{\nu})_{ran(\delta)}$, let $\alpha = (\nu \uplus \varrho_{\nu})_{ran(\delta)}$. By definition of the rule application strategy, $ran(\delta) \subseteq \mathcal{T}_{\Sigma}(\mathcal{X} \setminus V_{\mathcal{R},Call_{\mathcal{R}}})$ so, as $dom(\delta) \subseteq V_{\mathcal{R},Call_{\mathcal{R}}}$, $ran(\delta) \cap dom(\delta) = \emptyset$. Then for each $y \in vars(x\delta)$, $y \in ran(\delta)$, $x \neq y$, and:
    - if $y \notin dom(\alpha)$ then $y\alpha = y$ and $y \notin dom(\nu)_{ran(\delta)}$. In particular, as $y \in ran(\delta)$, $y \notin dom(\nu)$, so $vars(y\alpha) \cap dom(\nu) = \emptyset$;
    - else $y \in dom(\alpha)$ $(= dom((\nu \uplus \varrho_{\nu})_{ran(\delta)}))$. Then:
      - if $y \in dom(\nu_{ran(\delta)})$ then $vars(y\alpha) \subseteq ran(\nu)$ so, as $dom(\nu) \cap ran(\nu) = \emptyset$, $vars(y\alpha) \cap dom(\nu) = \emptyset$, and
      - if $y \in dom((\varrho_{\nu})_{ran(\delta)})$ then, as we have already proved $\varrho_{\nu} = \theta_{\setminus V}$ and θ is a composition of several CSUs, so $ran(\theta)$ is away from all known variables, $vars(y\alpha) \cap dom(\nu) = \emptyset$.

  In conclusion, $vars(x\gamma\sigma) \cap dom(\nu) = \emptyset$.
- The proof for rule top, with strategy $\mathsf{top}(c[\gamma]\{\overline{ST}\})$ and substitution σ, is exactly the same as the previous one.
- In rule [$c1$] call strategy, $(\bar{u}, \bar{v}, \phi) = (\bar{u}'\sigma, \bar{v}'\sigma, \phi'\sigma)$, where $\sigma = none$, so there is nothing to prove.
- Now, we check rule [$c2$] call strategy with strategy invocation $CS(\bar{t})$ and substitution $\gamma = \{\bar{x} \mapsto \bar{t}\}$. By I.H. $CS(\bar{t})$ has the form $(CS(\bar{w}))^{\nu'}\varrho_{\nu'}$, for appropriate $\bar{w}$, so $\bar{t} = \bar{w}(\nu' \uplus \varrho_{\nu'}) = \bar{w}(\nu \uplus \varrho_{\nu}) = \bar{w}(\nu \uplus \varrho_{\nu})$ $(\sigma = none)$, hence $\gamma = \{\bar{x} \mapsto \bar{w}(\nu \uplus \varrho_{\nu})\}$, let $\alpha = \nu \uplus \varrho_{\nu}$. The calculus rule uses a version of the

condition C in the right-side of the call strategy definition, call it C', where all the variables are new except for $dom(\gamma) \cup V^{\nu'}$. The new variables in C' are not in $dom(\nu)$, so (†) applies. We check the rest of the variables in C'. For each $x \in vars(C') \cap (dom(\gamma) \cup V^{\nu'})$:

- if $x \in V^{\nu'}$ then $x \in V^{\nu}$, because $\sigma = none$, and:
  - if $x \in V$ then $x \notin dom(\nu)$ so, as $x\sigma = x$, $vars(x\sigma) \cap dom(\nu) = \emptyset$;
  - else $x \in ran(\nu)$. Then, as $x\sigma = x$ and $dom(\nu) \cap ran(\nu) = \emptyset$, $vars(x\sigma) \cap dom(\nu) = \emptyset$;
- else $x \in dom(\gamma)$ $(= \bar{x})$, say $x = x_i$, so $x\gamma = w_i\alpha$ $(\alpha = \nu \uplus \varrho_\nu)$. For every $y \in vars(w_i)$:
  - if $y \in dom(\nu)$ then $vars(y\alpha) \subseteq ran(\nu)$ so, as $dom(\nu) \cap ran(\nu) = \emptyset$, $vars(y\alpha) \cap dom(\nu) = \emptyset$,
  - if $y \in dom(\varrho_\nu)$ then, as we have already proved $\varrho_\nu = \theta_{\setminus V}$ and θ is a composition of several CSUs, so $ran(\theta)$ is away from all known variables, $vars(y\alpha) \cap dom(\nu) = \emptyset$,
  - else $y \notin (dom(\nu) \cup dom(\varrho_\nu))$, so $y\alpha = y$. Then, as $y \notin dom(\nu)$, $vars(y\alpha) \cap dom(\nu) = \emptyset$.

  In conclusion, $vars(x\gamma) \cap dom(\nu) = \emptyset$.

11. The last calculus rule applied to get G from a goal of the form $G_1^\nu \varrho_\nu$, where $G_0 \rightsquigarrow^*_{\theta'} G_1^\nu \varrho_\nu$ and, by I.H. and invariant 3, $\varrho_\nu = \theta'_{\setminus V}$:

    - may have generated G as an instance of $G_1^\nu \varrho_\nu$ with a substitution σ, so $\theta = \theta'\sigma$. Then Definition 24 ensures that $\varrho_\mu = (\varrho_\nu\sigma)_{V_G \setminus V} = (\theta'_{\setminus V}\sigma)_{V_G \setminus V} = (\theta'\sigma)_{V_G \setminus V} = \theta_{V_G \setminus V}$, and we take $\varrho'_\mu = \varrho_\mu$, or
    - it may have not generated an instance, so $\theta = \theta'$, and we take $\varrho'_\mu = (\varrho_\nu)_{V_G} = (\theta'_{\setminus V})_{V_G} = \theta'_{V_G \setminus V} = \theta_{V_G \setminus V}$.

□

Proposition 17 (Canonical narrowing path)**.** *Given $\mathcal{R}_B = (\Sigma, E_0 \cup B, R_B)$, an associated rewrite theory of $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ closed under B-extensions, and a narrowing path from a goal G (with set of parameters V), $G = \Delta_0 \mid \psi_0 \mid V, none \rightsquigarrow_{\sigma_1} \Delta_1 \mid \psi_1 \mid V, \nu_1 \rightsquigarrow_{\sigma_2} \cdots \Delta_{m-1} \mid \psi_{m-1} \mid V, \nu_{m-1} \rightsquigarrow_{\sigma_m} nil \mid \psi_m \mid V, \nu_m$, there exists another narrowing path $G = \Delta_0 \mid \psi_0 \mid V, none \rightsquigarrow_{\sigma_1} \Delta_1 \mid \chi_1 \mid V, \nu_1 \rightsquigarrow_{\sigma_2} \cdots \Delta_{m-1} \mid \chi_{m-1} \mid V, \nu_{m-1} \rightsquigarrow_{\sigma_m} nil \mid \chi_m \mid V, \nu_m$, where the same inference rule, with the same substitution, is applied at each step in both paths, there is no simplification of the reachability formula on the second path, and $E_0 \vdash \psi_i \Leftrightarrow \chi_i$, for $1 \leq i \leq m$.*

Proof. As the applied rule at each step i only depends on Δ_{i-1} which is the same on both paths, as long as ψ_i and χ_i are satisfiable, all that it has to be proved is $E_0 \vdash \psi_i \Leftrightarrow \chi_i$. Then as ψ_i is satisfiable so is χ_i.

By the definition of the proposition, $\chi_0 = \psi_0$, so $E_0 \vdash \psi_0 \Leftrightarrow \chi_0$. The check for $E_0 \vdash \psi_{i-1} \Leftrightarrow \chi_{i-1}$ implies $E_0 \vdash \psi_i \Leftrightarrow \chi_i$, for $1 \leq i \leq m$, is trivial since there are only two type of inference rules in the calculus:

- those rules that do not modify the formula, so $\psi_i = \psi_{i-1}$, $\chi_i = \chi_{i-1}$, and $E_0 \vdash \psi_{i-1} \Leftrightarrow \chi_{i-1}$ implies $E_0 \vdash \psi_i \Leftrightarrow \chi_i$, and

- those rules where $\chi_i = (\chi_{i-1} \wedge \chi'_{i-1})\theta$, for suitable χ'_{i-1} and θ, and $E_0 \vdash \psi_i \Leftrightarrow (\chi_{i-1} \wedge \chi'_{i-1})\theta$, i.e., $E_0 \vdash \psi_i \Leftrightarrow \chi_i$.

□

Theorem 16 (Soundness of the Calculus for Reachability Goals). *Given an associated rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ closed under B-extensions and a reachability goal G, if $\nu \mid \psi$ is a computed answer for G then for each substitution $\rho : V^\nu \to \mathcal{T}_\Sigma$ such that $\psi\rho$ is satisfiable, $\nu \cdot \rho$ is a solution for G.*

Proof. By induction on the depth of the corresponding canonical narrowing path and the first inference rule applied.

Remember that $V^\mu = (V \setminus dom(\mu)) \cup ran(\mu)$, $(\bigwedge_{i=1}^n u_i \to v_i / ST_i^\mu(\varrho_\mu)_i)\sigma = \bigwedge_{i=1}^n u_i\sigma \to v_i\sigma / ST_i^{(\mu\sigma)_V}((\varrho_\mu)_i\sigma)_{\setminus V}$, and $vars(G) = vars(\phi) \cup \bigcup_{i=1}^n vars(\{u_i, v_i\}) \cup V^\mu$ or $vars(G) = \{x_k\} \cup vars(\phi) \cup \bigcup_{i=1}^n vars(\{u_i, v_i\}) \cup V^\mu$ (for rules $[c]$ and $[r]$).

- Base case

  Rule $[d1]$ (idle):

  $G = u_1 \to v_1/\mathtt{idle} \mid \psi_1 \mid V, \mu \rightsquigarrow_{[d1],\sigma} nil \mid \psi \mid V, (\mu\sigma)_V$, where $abstract_{\Sigma_1}((u, v)) = \langle \lambda(\bar{x}, \bar{y}).(u_1^\circ, v_1^\circ); (\theta_u^\circ, \theta_v^\circ); (\phi_u^\circ, \phi_v^\circ) \rangle$, $\psi = (\psi_1 \wedge \phi_u^\circ \wedge \phi_v^\circ)\sigma$, $\bar{x} = \{x_1, \ldots, x_{i_x}\}$, $u_1^\circ = u_1[\bar{x}]_{\bar{p}}$, $\phi_u^\circ = (\bigwedge_{i=1}^{i_x} x_i = u_1|_{p_i})$, $\bar{y} = \{y_1, \ldots, y_{i_y}\}$, $v_1^\circ = v_1[\bar{y}]_{\bar{q}}$, $\phi_v^\circ = (\bigwedge_{j=1}^{i_y} y_j = v_1|_{q_j})$, $\sigma \in CSU_B(u_1^\circ = v_1^\circ)$, so $u_1^\circ\sigma =_B v_1^\circ\sigma$, and ψ is satisfiable, for appropriate $\bar{p}$ and $\bar{q}$.

  As ρ is a ground substitution such that $dom(\rho) = vars(G\sigma)$ and $\psi\rho$ is satisfiable, i.e., $(\psi_1 \wedge \phi_u^\circ \wedge \phi_v^\circ)\sigma\rho$ is satisfiable, then $\psi_1\sigma\rho$ is ground, so $E_0 \vdash \psi_1\sigma\rho$, and $(\phi_u^\circ \wedge \phi_v^\circ)\sigma\rho$ is satisfiable, where $u_1\sigma\rho$ and $v_1\sigma\rho$ are ground terms, so there exists a substitution $\rho' : V_{\bar{x}\sigma\rho, \bar{y}\sigma\rho} \to \mathcal{T}_\Sigma$ such that $\bar{x}\sigma\rho\rho' =_{E_0} u_1|_{\bar{p}}\sigma\rho\rho' = u_1|_{\bar{p}}\sigma\rho$ and $\bar{y}\sigma\rho\rho' =_{E_0} v_1|_{\bar{q}}\sigma\rho\rho' = v_1|_{\bar{q}}\sigma\rho$.

  Let $\gamma = \sigma\rho\rho'$. As $u_1\sigma\rho$ and $v_1\sigma\rho$ are terms in $\mathcal{T}_\Sigma$, the theory inclusion $(\Sigma_0, E_0) \subseteq (\Sigma, E)$ is protecting, and $u_1^\circ\sigma\rho =_B v_1^\circ\sigma\rho$, then $u_1\sigma\rho = u_1\sigma\rho[u_1|_{\bar{p}}\sigma\rho]_{\bar{p}} =_{E_0} u_1\sigma\rho[\bar{x}\gamma]_{\bar{p}} = u_1\gamma[\bar{x}\gamma]_{\bar{p}} = u_1^\circ\gamma =_B v_1^\circ\gamma = v_1\gamma[\bar{y}\gamma]_{\bar{q}} = v_1\sigma\rho[\bar{y}\gamma]_{\bar{q}} =_{E_0} v_1\sigma\rho[v_1|_{\bar{q}}\sigma\rho]_{\bar{q}} = v_1\sigma\rho$, so $u_1\sigma\rho =_E v_1\sigma\rho$. As $vars(\{u_1, v_1, \psi_1\}) \subseteq vars(G)$ then $u_1\sigma_{vars(G)}\rho = u_1\sigma\rho =_E v_1\sigma\rho = v_1\sigma_{vars(G)}\rho$ and $E_0 \vdash \psi_1\sigma\rho$ implies $E_0 \vdash \psi_1\sigma_{vars(G)}\rho$ so, as in Example 5, $[v_1\sigma_{vars(G)}\rho]_E \in \mathtt{idle}@[u_1\sigma_{vars(G)}\rho]_E$, and $\sigma_{vars(G)}\rho$ is a solution of G.

- Inductive step

  $G = \bigwedge_{i=1}^n u_i \to v_i/ST_i^\mu(\varrho_\mu)_i \mid \psi_1 \mid V, \mu$ or $G = u_1|_p \to^1 x, u_1[x]_p \to v_1/ST_1^\mu(\varrho_\mu)_1 \wedge \bigwedge_{i=2}^n u_i \to v_i/ST_i^\mu(\varrho_\mu)_i \mid \psi_1 \mid V, \mu$. We let $\Delta = \bigwedge_{i=2}^n u_i \to v_i/ST_i^\mu(\varrho_\mu)_i$. When the substitution applied in the first narrowing step is *none*, Δ, ψ_1, and μ remain unchanged, so I.H. ensures that Δ and ψ_1 comply with the thesis of the theorem, as it is shown in the proof for the second subcase. We will omit this proof in the rest of related subcases, as the proof is always the same.

  1. Rule $[d1]$ (idle):
     $G = u_1 \to v_1/\mathtt{idle} \wedge \Delta \mid \psi_1 \mid V, \mu \rightsquigarrow_{[d1],\sigma_1} \Delta \circ \sigma_1 \mid \psi_1\sigma_1 \wedge \phi^\circ\sigma_1 \mid V, (\mu\sigma_1)_V = G'\sigma_1$, with $G' = \Delta \mid \psi_1 \wedge \phi^\circ \mid V, \mu$, where $abstract_{\Sigma_1}(v_1) = \langle \lambda\bar{x}.v_1^\circ; \theta^\circ; \phi^\circ \rangle$, $\bar{x} = \{x_1, \ldots, x_l\}$, $v_1^\circ = v_1[x_1, \ldots, x_l]_{q_1 \ldots q_l}$, $\phi^\circ = (\bigwedge_{i=1}^l x_i = v_1|_{q_i})$, $\sigma_1 \in CSU_B(u_1 = v_1^\circ)$, $\psi_1\sigma_1 \wedge \phi^\circ\sigma_1$ is satisfiable, and $G'\sigma_1 \rightsquigarrow_{\sigma'}^+ nil \mid \psi \mid V, \nu$, let $\sigma = \sigma_1\sigma'$, so $\sigma_{vars(G)} \mid \psi$ is a computed answer for G, and $\sigma'_{vars(G'\sigma_1)} \mid \psi$ is a computed answer for $G'\sigma_1$.

If $\rho : vars(G\sigma) \to \mathcal{T}_\Sigma$ is a substitution such that $\psi\rho$ is satisfiable, then let $\rho_1 = \rho_{vars(G'\sigma)}$, so also $\psi\rho_1$ is satisfiable. As $dom(\rho) = vars(G\sigma)$ then $dom(\rho_1) = vars(G\sigma) \cap vars(G'\sigma)$. Let $\rho_2 = \rho_{vars(G\sigma)\setminus vars(G'\sigma)}$, so $\rho = \rho_1 \uplus \rho_2$, and let $\rho'_1 : vars(G'\sigma) \setminus vars(G\sigma) \to \mathcal{T}_\Sigma$, so $dom(\rho_1) \cap dom(\rho'_1) = \emptyset$ and $dom(\rho_1) \cup dom(\rho'_1) = vars(G'\sigma)$, such that $\psi(\rho_1 \uplus \rho'_1)$ is satisfiable, and let $\rho' = \rho_1 \uplus \rho'_1$, so $\rho' : vars(G'\sigma) \to \mathcal{T}_\Sigma$.

As $dom(\rho'_1) = vars(G'\sigma) \setminus vars(G\sigma)$ and $dom(\rho_1) = vars(G\sigma) \cap vars(G'\sigma) \subseteq vars(G\sigma)$, then $\rho'_{vars(G\sigma)} = (\rho_1 \uplus \rho'_1)_{vars(G\sigma)} = (\rho_1)_{vars(G\sigma)} = \rho_1$, so by I.H., as $\rho' : vars(G'(\sigma_1\sigma')) \to \mathcal{T}_\Sigma$ and $\psi\rho'$ is satisfiable, $\sigma'_{vars(G'\sigma_1)}\rho'$ is a solution for $G'\sigma_1$, meaning that $E_0 \vdash (\psi_1 \wedge \phi^\circ)\sigma_1\sigma'_{vars(G'\sigma_1)}\rho'$ and there are closed proof trees for each open goal in $\Delta\sigma_1\sigma'_{vars(G'\sigma_1)}\rho'$ with respect to the instantiation $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}^{(\mu\sigma_1\sigma'_{vars(G'\sigma_1)}\rho')_V}$. We prove (a) $\Delta\sigma_1\sigma'_{vars(G'\sigma_1)}\rho' = \Delta\sigma_{vars(G)}\rho$ and (b) $(\mu\sigma_1\sigma'_{vars(G'\sigma_1)}\rho')_V = (\mu\sigma_{vars(G)}\rho)_V$:

(a) As Δ appears both in G and G' then $vars(\Delta\sigma_1) \subseteq vars(G\sigma_1) \cap vars(G'\sigma_1) \subseteq vars(G'\sigma_1)$, so $\Delta\sigma_1\sigma'_{vars(G'\sigma_1)} = \Delta\sigma_1\sigma'$ and $V_{\Delta\sigma_1\sigma'_{vars(G'\sigma_1)}} = V_{\Delta\sigma_1\sigma'} \subseteq V_{G\sigma_1\sigma'} \cap V_{G'\sigma_1\sigma'} = V_{G\sigma} \cap V_{G'\sigma}$, hence $\Delta\sigma_1\sigma'_{vars(G'\sigma_1)}\rho' = \Delta\sigma_1\sigma'\rho' = \Delta\sigma_1\sigma'\rho_1 = \Delta\sigma_1\sigma'\rho = \Delta\sigma\rho = \Delta\sigma_{vars(G)}\rho$.

(b) If $v \in V$ then either
- $v \notin dom(\mu)$ and $v\mu = v$, so $vars(v\mu) \subseteq V \setminus dom(\mu) \subseteq vars(G)$, or
- $v \in dom(\mu)$, so $vars(v\mu) \subseteq ran(\mu) \setminus dom(\mu) \subseteq vars(G)$.

Also, either
- $v \notin dom(\mu\sigma_1)$ and $v\mu\sigma_1 = v$, so $vars(v\mu\sigma_1) \subseteq V \setminus dom(\mu\sigma_1) \subseteq vars(G\sigma_1) \cap vars(G'\sigma_1)$, or
- $v \in dom(\mu\sigma_1)$, so $vars(v\mu\sigma_1) \subseteq ran(\mu\sigma_1)\setminus dom(\mu\sigma_1)$, and $vars(v\mu\sigma_1) \subseteq vars(G\sigma_1) \cap vars(G'\sigma_1)$.

As in case (a), $v\mu\sigma_1\sigma'_{vars(G'\sigma_1)} = v\mu\sigma_1\sigma'$, $vars(v\mu\sigma) = vars(v\mu\sigma_1\sigma') = vars(v\mu\sigma_1\sigma'_{vars(G'\sigma_1)}) \subseteq vars(G\sigma_1\sigma'_{vars(G'\sigma_1)}) \cap vars(G'\sigma_1\sigma'_{vars(G'\sigma_1)}) = vars(G\sigma_1\sigma') \cap vars(G'\sigma_1\sigma') = vars(G\sigma) \cap vars(G'\sigma)$.
Then $v\mu\sigma_1\sigma'_{vars(G'\sigma_1)}\rho' = v\mu\sigma\rho' = v\mu\sigma\rho_1 = v\mu\sigma\rho = v\mu\sigma_{vars(G)}\rho$ hence $(\mu\sigma_1\sigma'_{vars(G'\sigma_1)}\rho')_V = (\mu\sigma_{vars(G)}\rho)_V$.

Then, from (a) and (b), the same closed proof trees are also valid for each open goal in $\Delta\sigma_{vars(G)}\rho$ with respect to the instantiation $\mathcal{D}_{\mathcal{R},Call_\mathcal{R}}^{(\mu\sigma_{vars(G)}\rho)_V}$.
As $vars(\psi_1 \wedge \phi^\circ) \subseteq vars(G')$ then $(\psi_1 \wedge \phi^\circ)\sigma_1\sigma'_{vars(G'\sigma_1)} = (\psi_1 \wedge \phi^\circ)\sigma_1\sigma' = (\psi_1 \wedge \phi^\circ)\sigma$, so $E_0 \vdash (\psi_1 \wedge \phi^\circ)\sigma\rho'$, hence $E_0 \vdash \psi_1\sigma\rho'$ and $E_0 \vdash \phi^\circ\sigma\rho'$, where $(\psi_1 \wedge \phi^\circ)\sigma\rho'$ is ground, because $vars((\psi_1 \wedge \phi^\circ)\sigma) \subseteq vars(G'\sigma)$ and $\rho' : vars(G'\sigma) \to \mathcal{T}_\Sigma$. Now, $dom(\rho_1) = vars(G\sigma) \cap vars(G'\sigma)$, and $vars(v_1|_{q_i}) \subseteq vars(G\sigma) \cap vars(G'\sigma)$ implies $v_1|_{q_i}\sigma\rho_1 \in \mathcal{T}_\Sigma$ so, as $\rho' = \rho_1 \uplus \rho'_1$, $v_1|_{q_i}\sigma\rho' = v_1|_{q_i}\sigma(\rho_1 \uplus \rho'_1) = v_1|_{q_i}\sigma\rho_1 = v_1|_{q_i}\sigma(\rho_1 \uplus \rho_2) = v_1|_{q_i}\sigma\rho$, for $1 \le i \le l$, hence $\phi^\circ\sigma\rho' = (\bigwedge_{i=1}^{l} x_i\sigma\rho' = v_1|_{q_i}\sigma\rho)$. As also $vars(\psi_1\sigma) \subseteq vars(G\sigma) \cap vars(G'\sigma)$ then, reasoning exactly in the same way, $\psi_1\sigma\rho' = \psi_1\sigma\rho$, so $E_0 \vdash \psi_1\sigma\rho$.

Let $\gamma = \sigma(\rho_1 \uplus \rho_2 \uplus \rho'_1)$, where $\rho_1 \uplus \rho_2 \uplus \rho'_1 = \rho \uplus \rho'_1 = \rho' \uplus \rho_2$. As $u_1\sigma_1 =_B v_1^\circ\sigma_1$ then $u_1\sigma =_B v_1^\circ\sigma$, so $u_1\gamma =_B v_1^\circ\gamma$. Also, $u_1\sigma\rho$ and $v_1\sigma\rho \in \mathcal{T}_\Sigma$, because $vars(\{u_1\sigma, v_1\sigma\}) \subseteq dom(\rho) = vars(G\sigma)$, so $u_1\gamma = u_1\sigma\rho$ and $v_1\gamma = v_1\sigma\rho$. Finally, $\phi^\circ\sigma\rho'$ ground implies $x_i\sigma\rho'$ ground, so $x_i\sigma\rho' = x_i\gamma$, for $1 \le i \le l$. Then,

$u_1\sigma\rho = u_1\gamma =_B v_1^\circ\gamma = v_1\gamma[x_1\gamma,\ldots,x_l\gamma]_{q_1\ldots q_l} = v_1\sigma\rho[x_1\sigma\rho',\ldots,x_l\sigma\rho']_{q_1\ldots q_l} =_{E_0} v_1\sigma\rho[v_1|_{q_1}\sigma\rho,\ldots,v_1|_{q_l\ldots q_l} = v_1\sigma\rho$, so, as $E = B \cup E_0$, $u_1\sigma\rho =_E v_1\sigma\rho$, and, as $vars(\{u_1, v_1\}) \subseteq vars(G)$, $u_1\sigma_{vars(G)}\rho =_E v_1\sigma_{vars(G)}\rho$. Then, as in Example 5, $[v_1\sigma_{vars(G)}\rho]_E \in \mathtt{idle}@[u_1\sigma_{vars(G)}\rho]_E$. As also $E_0 \vdash \psi_1\sigma_{vars(G)}\rho$, and there are closed proof trees for each open goal in $\Delta\sigma_{vars(G)}\rho$ with respect to $\mathcal{D}^{(\mu\sigma_{vars(G)}\rho)_V}_{\mathcal{R},Call_\mathcal{R}}$, then $\sigma_{vars(G)}\rho$ is a solution of G.

2. Rule [$d2$] (idle):

   $G = u_1 \to v_1/\mathtt{idle}; ST^\mu\varrho_\mu \wedge \Delta \mid \psi_1 \mid V, \mu \rightsquigarrow_{[d2],none} u_1 \to v_1/ST^\mu\varrho_\mu \wedge \Delta \mid \psi_1 \mid V, \mu = G'$ and $G' \rightsquigarrow^+_\sigma nil \mid \psi \mid V, \nu$, where $\nu = (\mu\sigma)_V$, so $\sigma_{vars(G)} \mid \psi$ is a computed answer for both G and G', since $vars(G) = vars(G')$. For any substitution ρ that satisfies the premises of the theorem, by I.H., $\sigma_{vars(G)}\rho$ is a solution for G', let $\delta = \sigma_{vars(G)}\rho$, $\nu' = (\mu\delta)_V$, and $\varrho_{\nu'} = (\varrho_\mu\delta)_{\setminus V}$, so $E_0 \vdash \psi_1\delta$, there are closed proof trees for each open goal in $\Delta\delta$, and also a c.p.t. $\frac{F}{u_1\delta\to v_1\delta/ST^{\nu'}\varrho_{\nu'}}$, all of them with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$. As there is a rule $\frac{u_1\delta\to u_1\delta/\mathtt{idle} \quad u_1\delta\to v_1\delta/ST^{\nu'}\varrho_{\nu'}}{u_1\delta\to v_1\delta/\mathtt{idle}\,;\,ST^{\nu'}\varrho_{\nu'}} \in \mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$, then $\frac{\frac{}{u_1\delta\to u_1\delta/\mathtt{idle}} \quad \frac{F}{u_1\delta\to v_1\delta/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta\to v_1\delta/\mathtt{idle}\,;\,ST^{\nu'}\varrho_{\nu'}}$ is also a c.p.t., with respect to $\mathcal{D}^{\nu'}_{\mathcal{R}}$, so $\sigma_{vars(G)}\rho$, is also a solution of G.

3. Rules [$o1$] and [$o2$] (or):

   we prove [$o1$]; the proof for [$o2$] is exactly the same, with ST_2 instead of ST_1. $G = u_1 \to v_1/((ST_1^\mu \mid ST_2^\mu)\,;\,ST^\mu)\varrho_\mu \wedge \Delta \mid \psi_1 \mid V, \mu \rightsquigarrow_{[o1],none} u_1 \to v_1/(ST_1^\mu\,;\,ST^\mu)\varrho_\mu \wedge \Delta \mid \psi_1 \mid V, \mu = G'$, so $vars(G) = vars(G')$, and $G' \rightsquigarrow^+_\sigma nil \mid \psi \mid V, \nu$, where $\nu = (\mu\sigma)_V$, so $\sigma_{vars(G)}|\psi$ is a computed answer for G and $\sigma_{vars(G')}|\psi$ is a computed answer for G'. Let $\Delta_1 = u_1 \to v_1/(ST_1^\mu\,;\,ST^\mu)\varrho_\mu$. By I.H., for any substitution $\rho : vars(G'\sigma) \to \mathcal{T}_\Sigma$ such that $\psi\rho$ is satisfiable, $\sigma_{vars(G')}\rho$ is a solution for G', let $\delta = \sigma_{vars(G')}\rho$ $(= \sigma_{vars(G)}\rho)$, $\nu' = (\mu\delta)_V$, and $\varrho_{\nu'} = (\varrho_\mu\delta)_{\setminus V}$, so there is a c.p.t. for $\Delta_1\delta$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$. The c.p.t. has the form $\frac{\frac{F_1}{u_1\delta\to t/ST_1^{\nu'}\varrho_{\nu'}} \quad \frac{F_2}{t\to v_1\delta/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta\to v_1\delta/(ST_1^{\nu'}\,;\,ST^{\nu'})\varrho_{\nu'}}$ for some term $t \in \mathcal{H}_\Sigma$. As there are rules $\frac{u_1\delta\to t/(ST_1^{\nu'}|ST_2^{\nu'})\varrho_{\nu'} \quad t\to v_1\delta/ST^{\nu'}\varrho_{\nu'}}{u_1\delta\to v_1\delta/((ST_1^{\nu'}|ST_2^{\nu'})\,;\,ST^{\nu'})\varrho_{\nu'}}$ and $\frac{u_1\delta\to t/ST_1^{\nu'}\varrho_{\nu'}}{u_1\delta\to t/(ST_1^{\nu'}|ST_2^{\nu'})\varrho_{\nu'}}$ in $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$, then the proof tree

$$\frac{\frac{\frac{F_1}{u_1\delta\to t/ST_1^{\nu'}\varrho_{\nu'}}}{u_1\delta\to t/(ST_1^{\nu'}|ST_2^{\nu'})\varrho_{\nu'}} \quad \frac{F_2}{t\to v_1\delta/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta \to v_1\delta/((ST_1^{\nu'} \mid ST_2^{\nu'})\,;\,ST^{\nu'})\varrho_{\nu'}}$$

   is closed, so, as $vars(G) = vars(G')$, $\rho : vars(G\sigma) \to \mathcal{T}_\Sigma$, $\psi\rho$ is satisfiable, and $\sigma_{vars(G)}\rho$ is a solution of G.

4. Rule [$p1$] (plus):

   $G = u_1 \to v_1/(ST_1^\mu{+};ST^\mu)\varrho_\mu{\wedge}\Delta \mid \psi_1 \mid V, \mu \rightsquigarrow_{[p1],none} u_1 \to v_1/(ST_1^\mu; ST^\mu)\varrho_\mu{\wedge}\Delta \mid \psi_1 \mid V, \mu = G'$, so $vars(G) = vars(G')$, and $G' \rightsquigarrow^+_\sigma nil \mid \psi \mid V, \nu$, where $\nu = (\mu\sigma)_V$, hence $\sigma_{vars(G)}|\psi$ is a computed answer for both G and G'. Let $\Delta_1 = u_1 \to v_1/(ST_1^\mu; ST^\mu)\varrho_\mu$. By I.H., for any substitution $\rho : vars(G\sigma) \to \mathcal{T}_\Sigma$ such that $\psi\rho$ is satisfiable, $\sigma_{vars(G)}\rho$ is a solution for G', let $\delta = \sigma_{vars(G')}\rho$ (=

$\sigma_{vars(G)}\rho)$, $\nu' = (\mu\delta)_V$, and $\varrho_{\nu'} = (\varrho_\mu\delta)_{\setminus V}$, so there is a c.p.t. for $\Delta_1\delta$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$.

The c.p.t. has the form $\dfrac{\frac{F_1}{u_1\delta\to t/ST_1^{\nu'}\varrho_{\nu'}} \quad \frac{F_2}{t\to v_1\delta/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta\to v_1\delta/(ST_1^{\nu'}\,;ST^{\nu'})\varrho_{\nu'}}$ for some term $t \in \mathcal{H}_\Sigma$. As there are rules $\dfrac{u_1\delta\to t/(ST_1^{\nu'}\varrho_{\nu'})+ \quad t\to v_1\delta/ST^{\nu'}\varrho_{\nu'}}{u_1\delta\to v_1\delta/(ST_1^{\nu'}+\,;ST^{\nu'})\varrho_{\nu'}}$ and $\dfrac{u_1\delta\to t/ST_1^{\nu'}\varrho_{\nu'}}{u_1\delta\to t/(ST_1^{\nu'}\varrho_{\nu'})+}$ in $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$, then

$$\dfrac{\dfrac{\frac{F_1}{u_1\delta\to t/ST_1^{\nu'}\varrho_{\nu'}}}{u_1\delta\to t/(ST_1^{\nu'}\varrho_{\nu'})+} \quad \dfrac{F_2}{t\to v_1\delta/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta\to v_1\delta/(ST_1^{\nu'}+\,;\,ST^{\nu'})\varrho_{\nu'}}$$

is a c.p.t., so $\rho : vars(G\sigma) \to \mathcal{T}_\Sigma$, $\psi\rho$ is satisfiable, and $\sigma_{vars(G)}\rho$ is a solution of G.

5. Rule [$p2$] (plus):

$G = u_1 \to v_1/(ST_1^\mu+\,;\,ST^\mu)\varrho_\mu \wedge \Delta \mid \psi_1 \mid V, \mu$.

$G \rightsquigarrow_{[p2],none} u_1 \to v_1/(ST_1^\mu\,;\,ST_1^\mu+;\,ST^\mu)\varrho_\mu \wedge \Delta \mid \psi_1 \mid V, \mu = G'$, so $vars(G) = vars(G')$, and $G' \rightsquigarrow_\sigma^+ nil \mid \psi \mid V, \nu$, where $\nu = (\mu\sigma)_V$, hence $\sigma_{vars(G)}|\psi$ is a computed answer for both G and G'. Let $\Delta_1 = u_1 \to v_1/(ST_1^\mu;ST_1^\mu+)\varrho_\mu;ST^\mu$. By I.H., for any substitution $\rho : vars(G\sigma) \to \mathcal{T}_\Sigma$ such that $\psi\rho$ is satisfiable, $\sigma_{vars(G)}\rho$ is a solution for G', let $\delta = \sigma_{vars(G')}\rho$ $(= \sigma_{vars(G)}\rho)$, $\nu' = (\mu\delta)_V$, and $\varrho_{\nu'} = (\varrho_\mu\delta)_{\setminus V}$, so there is a c.p.t. for $\Delta_1\delta$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$. The c.p.t. has the form $\dfrac{\frac{F_1}{u_1\delta\to t_1/ST_1^{\nu'}\varrho_{\nu'}} \quad \dfrac{\frac{F_2}{t_1\to t_2/(ST_1^{\nu'}\varrho_{\nu'})+} \quad \frac{F_3}{t_2\to v_1\delta/ST^{\nu'}\varrho_{\nu'}}}{t_1\to v_1\delta/(ST_1^{\nu'}+\,;ST^{\nu'})\varrho_{\nu'}}}{u_1\delta\to v_1\delta/(ST_1^{\nu'}\,;ST_1^{\nu'}+\,;ST^{\nu'})\varrho_{\nu'}}$, for terms t_1 and $t_2 \in \mathcal{H}_\Sigma$. As there are rules $\dfrac{u_1\delta\to t_2/(ST_1^{\nu'}\varrho_{\nu'})+ \quad t_2\to v_1\delta/ST^{\nu'}\varrho_{\nu'}}{u_1\delta\to v_1\delta/(ST_1^{\nu'}+\,;ST^{\nu'})\varrho_{\nu'}}$, $\dfrac{u_1\delta\to t_2/(ST_1^{\nu'}\,;ST_1^{\nu'}+)\varrho_{\nu'}}{u_1\delta\to t_2/(ST_1^{\nu'}\varrho_{\nu'})+}$, and $\dfrac{u_1\delta\to t_1/ST_1^{\nu'}\varrho_{\nu'} \quad t_1\to t_2/(ST_1^{\nu'}\varrho_{\nu'})+}{u_1\delta\to t_2/ST_1^{\nu'}\varrho_{\nu'}\,;ST_1^{\nu'}+}$ in $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$, then

$$\dfrac{\dfrac{\dfrac{\frac{F_1}{u_1\delta\to t_1/ST_1^{\nu'}\varrho_{\nu'}} \quad \dfrac{F_2}{t_1\to t_2/(ST_1^{\nu'}\varrho_{\nu'})+}}{u_1\delta\to t_2/(ST_1^{\nu'}\,;ST_1^{\nu'}+)\varrho_{\nu'}}}{u_1\delta\to t_2/(ST_1^{\nu'}\varrho_{\nu'})+} \quad \dfrac{F_3}{t_2\to v_1\delta/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta\to v_1\delta/(ST_1^{\nu'}+\,;\,ST^{\nu'})\varrho_{\nu'}}$$

is a c.p.t., so $\rho : vars(G\sigma) \to \mathcal{T}_\Sigma$, $\psi\rho$ is satisfiable, and $\sigma_{vars(G)}\rho$ is a solution of G.

6. Rule [$s1$] (star):

$G = u_1 \to v_1/(ST_1^\mu*\,;\,ST^\mu)\varrho_\mu \wedge \Delta \mid \psi_1 \mid V, \mu \rightsquigarrow_{[s1],none} u_1 \to v_1/ST^\mu\varrho_\mu \wedge \Delta \mid \psi_1 \mid V, \mu = G'$, so $vars(G) = vars(G')$, and $G' \rightsquigarrow_\sigma^+ nil \mid \psi \mid V, \nu$, where $\nu = (\mu\sigma)_V$, so $\sigma_{vars(G)}|\psi$ is a computed answer for G and $\sigma_{vars(G')}|\psi$ is a computed answer for G'. Let $\Delta_1 = u_1 \to v_1/ST^\mu\varrho_\mu$. By I.H., for any substitution $\rho : vars(G'\sigma) \to \mathcal{T}_\Sigma$ such that $\psi\rho$ is satisfiable, $\sigma_{vars(G')}\rho$ is a solution for G', let $\delta = \sigma_{vars(G')}\rho$ $(= \sigma_{vars(G)}\rho)$, $\nu' = (\mu\delta)_V$, and $\varrho_{\nu'} = (\varrho_\mu\delta)_{\setminus V}$, so there is a c.p.t. for $\Delta_1\delta$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$. The c.p.t. has the form $\frac{F_1}{u_1\delta\to v_1\delta/ST^{\nu'}\varrho_{\nu'}}$.

As, by definition, $(ST_1^{\nu'}\varrho_{\nu'})* = \texttt{idle} \mid (ST_1^{\nu'}\varrho_{\nu'})+$ and there are rules $\frac{}{u_1\delta \to u_1\delta/\texttt{idle}}$, $\frac{u_1\delta \to u_1\delta/\texttt{idle}}{u_1\delta \to u_1\delta/\texttt{idle}|(ST_1^{\nu'}\varrho_{\nu'})+}$, and $\frac{u_1\delta \to u_1\delta/\texttt{idle}|(ST_1^{\nu'}\varrho_{\nu'})+ \quad u_1\delta \to v_1\delta/ST^{\nu'}\varrho_{\nu'}}{u_1\delta \to v_1\delta/(\texttt{idle}|ST_1^{\nu'}+\,;\,ST^{\nu'})\varrho_{\nu'}}$ in $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$, then the proof tree

$$\frac{\dfrac{\overline{u_1\delta \to u_1\delta/\texttt{idle}}}{u_1\delta \to u_1\delta/\texttt{idle}|(ST_1^{\nu'}\varrho_{\nu'})+} \quad \dfrac{F_1}{u_1\delta \to v_1\delta/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta \to v_1\delta/((\texttt{idle} \mid ST_1^{\nu'}+)\,;\,ST^{\nu'})\varrho_{\nu'}}$$

is closed, so $\rho : vars(G\sigma) \to \mathcal{T}_\Sigma$, $\psi\rho$ is satisfiable, and $\sigma_{vars(G)}\rho$ is a solution of G.

7. Rule [$s2$] (star):
$G = u_1 \to v_1/(ST_1^{\mu}*; ST^{\mu})\varrho_\mu \wedge \Delta \mid \psi_1 \mid V, \mu \rightsquigarrow_{[s2],none} u_1 \to v_1/(ST_1^{\mu}+; ST^{\mu})\varrho_\mu \wedge \Delta \mid \psi_1 \mid V, \mu = G'$, so $vars(G) = vars(G')$, and $G' \rightsquigarrow_\sigma^+ nil \mid \psi \mid V, \nu$, where $\nu = (\mu\sigma)_V$, hence $\sigma_{vars(G)}|\psi$ is a computed answer for both G and G'. Let $\Delta_1 = (ST_1^{\mu}+\,;\,ST^{\mu})\varrho_\mu$. By I.H., for any substitution $\rho : vars(G\sigma) \to \mathcal{T}_\Sigma$ such that $\psi\rho$ is satisfiable, $\sigma_{vars(G)}\rho$ is a solution for G', let $\delta = \sigma_{vars(G')}\rho$ $(= \sigma_{vars(G)}\rho)$, $\nu' = (\mu\delta)_V$, and $\varrho_{\nu'} = (\varrho_\mu\delta)_{\setminus V}$, so there is a c.p.t. for the goal $\Delta_1\delta$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$.
The c.p.t. has the form $\frac{\frac{F_1}{u_1\delta \to t/(ST_1^{\nu'}\varrho_{\nu'})+} \quad \frac{F_2}{t \to v_1\delta/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta \to v_1\delta/(ST_1^{\nu'}\varrho_{\nu'})+\,;\,ST^{\nu'}\varrho_{\nu'}}$ for some term $t \in \mathcal{H}_\Sigma$.
As, by definition, $(ST_1^{\nu'}\varrho_{\nu'})* = \texttt{idle} \mid (ST_1^{\nu'}\varrho_{\nu'})+$ and $\frac{u_1\delta \to t/(ST_1^{\nu'}\varrho_{\nu'})+}{u_1\delta \to t/\texttt{idle}|(ST_1^{\nu'}\varrho_{\nu'})+}$ and $\frac{u_1\delta \to t/\texttt{idle}|(ST_1^{\nu'}\varrho_{\nu'})+ \quad t \to v_1\delta/ST_1\delta}{u_1\delta \to v_1\delta/((\texttt{idle}|ST_1^{\nu'}+)\,;\,ST^{\nu'})\varrho_{\nu'}}$ are rules in $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$, then

$$\frac{\dfrac{\dfrac{F_1}{u_1\delta \to t/(ST_1^{\nu'}\varrho_{\nu'})+}}{u_1\delta \to t/\texttt{idle}|(ST_1^{\nu'}\varrho_{\nu'})+} \quad \dfrac{F_2}{t \to v_1\delta/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta \to v_1\delta/((\texttt{idle} \mid ST_1^{\nu'}+)\,;\,ST^{\nu'})\varrho_{\nu'}}$$

is a c.p.t., so $\rho : vars(G\sigma) \to \mathcal{T}_\Sigma$, $\psi\rho$ is satisfiable, and $\sigma_{vars(G)}\rho$ is a solution of G.

8. Rule [$i1$] (if then else):
$G = u_1 \to v_1 \,/\, (\texttt{match}\ t_1\ \texttt{s.t.}\ \phi_1\ ?\ ST_1 : ST_2\,;ST)^\mu\varrho_\mu\,(\wedge\,\Delta) \mid \psi_1 \mid V, \mu \rightsquigarrow_{[i1],\sigma_1} (u_1 \to v_1/(ST_1\,;ST)^\mu\varrho_\mu(\wedge\Delta) \mid \psi_2 \mid V, \mu)\sigma_1 = G'\sigma_1$, let $t = t_1^\mu\varrho_\mu$ and $\phi = \phi_1^\mu\varrho_\mu$, where $abstract_{\Sigma_1}(t) = \langle \lambda\bar{x}.t^\circ; \sigma^\circ; \phi^\circ\rangle$, $t^\circ = t[\bar{x}]_{\bar{q}}$, with $\bar{x} = x_1, \ldots, x_l$ and $\bar{q} = q_1, \ldots, q_l$, $\phi^\circ = (\bigwedge_{i=1}^{l} x_i = t|_{q_i})$, hence $V_{t^\circ} \cup V_{\phi^\circ} = V_t \cup \hat{x}$, $\sigma_1 \in CSU_B(u_1 = t^\circ)$, $\psi_2 = \psi_1 \wedge \phi \wedge \phi^\circ$, so $V_G \subseteq V_{G'}$, $\psi_2\sigma_1$ is satisfiable, and $G'\sigma_1 \rightsquigarrow_{\sigma'}^+ nil \mid \psi \mid V, \nu$, let $\sigma = \sigma_1\sigma'$, where $\nu = (\mu\sigma)_V = (\mu\sigma_1\sigma')_V$, so $\sigma_{V_G} \mid \psi$ is a computed answer for G and $\sigma'_{V_{G'\sigma_1}} \mid \psi$ is a computed answer for $G'\sigma_1$.
Let $\rho : V_{G\sigma} \to \mathcal{T}_\Sigma$ be a substitution such that $\psi\rho$ is satisfiable, $\delta = \sigma_{V_G}\rho$, $\nu' = (\mu\delta)_V$, so $dom(\nu') = V$ and $ran(\nu') = \emptyset$, and $\varrho_{\nu'} = (\varrho_\mu\delta)_{\setminus V}$, so $\delta : V_G \to \mathcal{T}_\Sigma$. As $dom(\rho) = V_{G\sigma}$ and $V_G \subseteq V_{G'}$, so $V_{G\sigma} \subseteq V_{G'\sigma}$, then $dom(\rho) \subseteq V_{G'\sigma}$. Let $\rho'_1 : V_{G'\sigma} \setminus V_{G\sigma} \to \mathcal{T}_\Sigma$, so $dom(\rho) \cup dom(\rho'_1) = V_{G'\sigma}$, such that $\psi(\rho \uplus \rho'_1)$ is satisfiable, and $\rho' = \rho \uplus \rho'_1$, so $\rho' : V_{G'\sigma} \to \mathcal{T}_\Sigma$ and $\rho'_{V_{G\sigma}} = \rho$.

By I.H., as $\rho' : V_{G'\sigma_1\sigma'} \to \mathcal{T}_\Sigma$ and $\psi\rho'$ is satisfiable, $\sigma'_{V_{G'\sigma_1}}\rho'$ is a solution for $G'\sigma_1$, let $\delta' = \sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$, $\varrho' = (\varrho_\mu\delta')_{\setminus V}$, and $\rho'' = \delta'_{V_{t,\phi}\setminus V_G}$.

We prove several intermediate results:

- $(\mu\delta)_V = (\mu\delta')_V$.
  We prove the equivalent fact, $x \in vars(V\mu) \implies x\delta = x\delta'$: as $V^\mu = (V \setminus dom(\mu)) \cup ran(\mu)$ then $V_{V\mu} = V^\mu$ so if $x \in V_{V\mu} = V^\mu \subseteq V_G$ then $x \in V_G$, $x\sigma_1 \in V_{G\sigma_1} \subseteq V_{G'\sigma_1}$, and $x(\sigma_1\sigma')_{V_G} = x\sigma_1\sigma'_{V_{G'\sigma_1}}$. Now, as $x\delta$ $(= x\sigma_{V_G}\rho)$ is ground, $x\delta = x\sigma_{V_G}\rho = x\sigma_{V_G}(\rho \uplus \rho'_1) = x\sigma_{V_G}\rho' = x(\sigma_1\sigma')_{V_G}\rho' = x\sigma_1\sigma'_{V_{G'\sigma_1}}\rho' = x\delta'$.
- $V_{(t\sigma,\phi\sigma)} \subseteq V_{G'\sigma}$.
  As $V_{t^\circ} \cup V_{\phi^\circ} = V_t \cup \hat{x}$, $\psi_2 = \psi_1 \wedge \phi \wedge \phi^\circ$, and $\sigma_1 \in CSU_B(u_1 = t^\circ)$, so $V_{t^\circ\sigma_1} = V_{u_1\sigma_1} \subseteq V_{G\sigma_1}$, because B is regular, hence $V_{G\sigma_1} \cup V_{t^\circ\sigma_1} = V_{G\sigma_1}$, then $V_{G'\sigma_1} = V_{G\sigma_1} \cup V_{\phi^\circ\sigma_1} \cup V_{\phi\sigma_1} = V_{G\sigma_1} \cup V_{t^\circ\sigma_1} \cup V_{\phi^\circ\sigma_1} \cup V_{\phi\sigma_1} = V_{G\sigma_1} \cup V_{t\sigma_1} \cup V_{\bar{x}\sigma_1} \cup V_{\phi\sigma_1} = V_{G\sigma_1} \cup V_{(t\sigma_1,\phi\sigma_1)} \cup V_{\bar{x}\sigma_1}$, so $V_{(t\sigma_1,\phi\sigma_1)} \subseteq V_{G'\sigma_1}$, hence $V_{(t\sigma,\phi\sigma)} \subseteq V_{G'\sigma}$.
- $V_{(t_1^{\nu'},\phi_1^{\nu'})} \subseteq V_{(t_1^\mu,\phi_1^\mu)}$.
  This is immediate since $dom(\mu) \subseteq V$, $\nu' = (\mu\delta)_V$, so $dom(\mu) \subseteq dom(\nu')$, and $\nu' : V \to \mathcal{T}_\Sigma$.
- $V_{(t_1^\mu,\phi_1^\mu)} \setminus V_{(t_1^{\nu'},\phi_1^{\nu'})} \subseteq V^\mu$.
  As $dom(\mu) \subseteq V$ and $\nu' = (\mu\delta)_V$ then the variables in $V_{(t_1^\mu,\phi_1^\mu)}$ instantiated in $V_{(t_1^{\nu'},\phi_1^{\nu'})}$ must belong either to $V \setminus dom(\mu)$ or to $ran(\mu)$, i.e., to V^μ. Since $\nu' : V \to \mathcal{T}_\Sigma$ then $V_{(t_1^{\nu'},\phi_1^{\nu'})} \setminus V_{(t_1^\mu,\phi_1^\mu)} = \emptyset$ and the result follows.
- $\phi\sigma\rho' = \phi_1^{\nu'}\varrho_{\nu'}\rho''$.
  As $(\mu\delta)_V = (\mu\delta')_V$ then $\phi\sigma\rho' = \phi\delta' = (\phi_1^\mu\varrho_\mu)\delta' = \phi_1^{(\mu\delta')_V}(\varrho_\mu\delta')_{\setminus V} = \phi_1^{(\mu\delta)_V}\varrho' = \phi_1^{\nu'}\varrho'$, so we prove the equivalent $\phi_1^{\nu'}\varrho' = \phi_1^{\nu'}\varrho_{\nu'}\rho''$ by proving $x \in V_{\phi_1} \implies x^{\nu'}\varrho' = x^{\nu'}\varrho_{\nu'}\rho''$. We consider two cases:
  - if $x \in V$ then $x^{\nu'}$ is ground, so $x^{\nu'}\varrho' = x^{\nu'}\varrho_{\nu'}\rho''$.
  - if $x \notin V$ then $x^{\nu'} = x$, so $x^{\nu'} \notin V$. Also, as $x \notin V$, $x^\mu = x$ so, as $x \in V_{\phi_1}$, $x \in V_{\phi_1^\mu}$. As $x \notin V$ and $x^{\nu'} = x$ then $x^{\nu'}\varrho' = x\varrho' = x(\varrho_\mu\delta')_{\setminus V} = x\varrho_\mu\delta'$ and $x^{\nu'}\varrho_{\nu'}\rho'' = x\varrho_{\nu'}\rho'' = x(\varrho_\mu\delta)_{\setminus V}\rho'' = x\varrho_\mu\delta\rho''$, so we check $x\varrho_\mu\delta' = x\varrho_\mu\delta\rho''$ by checking $y \in V_{x\varrho_\mu} \implies y\delta' = y\delta\rho''$:
    - as $x \in V_{\phi_1^\mu}$ and $y \in V_{x\varrho_\mu}$ then $y \in V_{\phi_1^\mu\varrho_\mu}$, i.e., $y \in V_\phi$;
    - again, we consider two cases:
      (a) if $y \in V_G$ then $y\delta$ is ground, so $y\delta\rho'' = y\delta = y\sigma_{V_G}\rho = y\sigma\rho = y\sigma_1\sigma'\rho$. Also, as $V_G \subseteq V_{G'}$, $y \in V_{G'}$ and $V_{y\sigma_1} \subseteq V_{G'\sigma_1}$, so $y\delta' = y\sigma_1\sigma'_{V_{G'\sigma_1}}\rho = y\sigma_1\sigma'\rho = y\delta\rho''$;
      (b) if $y \notin V_G$ then, as $y \in V_\phi$, $y \in V_{\phi\setminus G} \subseteq V_{(\phi,t)\setminus G}$ so, as also $y \notin V_G$ and $\delta : V_G \to \mathcal{T}_\Sigma$, $y\delta\rho'' = y\rho'' = y\delta'_{V_{(\phi_1,t)\setminus G}} = y\delta'$.
- $t\sigma\rho' = t_1^{\nu'}\varrho_{\nu'}\rho''$.
  The proof is the same as the previous one, just exchanging ϕ and t everywhere, even when they appear with subscripts and/or superscripts.

As $\sigma'_{V_{G'\sigma_1}}\rho'$ is a solution for $G'\sigma_1$ then, by I.H.:

(a) $E_0 \vdash \psi_2\delta'$, i.e., $E_0 \vdash (\psi_1 \wedge \phi \wedge \phi^\circ)\delta'$,

(b) there are closed proof trees for each open goal in $\Delta\delta'$, with respect to $\mathcal{D}^{(\mu\delta')_V}_{\mathcal{R},Call_\mathcal{R}}$ ($=\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$, we use ν' instead of $(\mu\delta')_V$ in (c)), and

(c) $[v_1\delta']_E \in (ST_1\,;ST)^{\nu'}\varrho'@[u_1\delta']_E$,

so:

(a) i. $V_{\psi_2} \subseteq V_{G'}$ implies $\psi_2\sigma_1\sigma'_{V_{G'\sigma_1}} = \psi_2\sigma_1\sigma' = \psi_2\sigma$, so $E_0 \vdash \psi_2\sigma\rho'$, where $\psi_2\sigma\rho'$ is ground, because $V_{\psi_2\sigma} \subseteq V_{G'\sigma}$ and $\rho' : V_{G'\sigma} \to \mathcal{T}_\Sigma$, hence $E_0 \vdash \psi_1\sigma\rho'$, $E_0 \vdash \phi^\circ\sigma\rho'$, and $E_0 \vdash \phi\sigma\rho'$, all ground expressions.

ii. $V_{\psi_1\sigma} \subseteq V_{G\sigma}$ and $dom(\rho) = V_{G\sigma}$ implies $\psi_1\sigma\rho \in \mathcal{T}_\Sigma$ so, as $\rho' = \rho \uplus \rho'_1$, $\psi_1\sigma\rho' = \psi_1\sigma(\rho \uplus \rho'_1) = \psi_1\sigma\rho = \psi_1\delta$, hence $E_0 \vdash \psi_1\delta$ (†).

(b) As in subcase (a)-ii, $V_\Delta \subseteq V_G$ implies $\Delta\delta' = \Delta\delta$, and the same closed proof trees are valid for each open goal in $\Delta\delta$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$ (††).

(c) Again, $V_{v_1,u_1} \subseteq V_G$ implies that $v_1\delta' = v_1\delta$ and $u_1\delta' = u_1\delta$. Then there is a c.p.t. of the form $\dfrac{\dfrac{F_1}{u_1\delta \to w/ST_1^{\nu'}\varrho'} \quad \dfrac{F_2}{w \to v_1\delta/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta \to v_1\delta/(ST_1;ST)^{\nu'}\varrho'}$, for some term $w \in \mathcal{H}_\Sigma$, with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$.

We prove (a) $ST^{\nu'}\varrho_{\nu'}\rho'' = ST^{\nu'}\varrho'$ and (b) $dom(\rho'') = V_{t,\phi} \setminus V_G$:

(a) As $\varrho_{\nu'} = (\varrho_\mu\delta)_{\setminus V}$, $\delta = \sigma_{V_G}\rho$, $\sigma = \sigma_1\sigma'$, $\delta' = \sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$, $\varrho' = (\varrho_\mu\delta')_{\setminus V}$, and $V_{ST^{\nu'}} \cap V = \emptyset$ this is the same as $ST^{\nu'}\varrho_\mu(\sigma_1\sigma')_{V_G}\rho\rho'' = ST^{\nu'}\varrho_\mu\sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$. Let $y \in V_{ST^{\nu'}\varrho_\mu}$, so $y \notin V$. There are two options:

i. $y \in V_G$. Then $V_{y\sigma_1} \subseteq V_{G\sigma_1} \subseteq V_{G'\sigma_1}$, so $y(\sigma_1\sigma')_{V_G} = y\sigma_1\sigma' = y\sigma_1\sigma'_{V_{G'\sigma_1}}$. Also $y(\sigma_1\sigma')_{V_G} = y\sigma$, hence $V_{y(\sigma_1\sigma')_{V_G}} \subseteq V_{G\sigma}$. Then, as $\rho : V_{G\sigma} \to \mathcal{T}_\Sigma$, $y(\sigma_1\sigma')_{V_G}\rho$ is ground, so $y(\sigma_1\sigma')_{V_G}\rho\rho'' = y(\sigma_1\sigma')_{V_G}\rho = y\sigma_1\sigma'_{V_{G'\sigma_1}}\rho = y\sigma_1\sigma'_{V_{G'\sigma_1}}(\rho \cup \rho'_1) = y\sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$;

ii. $y \notin V_G$, so $y(\sigma_1\sigma')_{V_G} = y$. As $ran(\sigma) \cap V_{ST^\mu\varrho_\mu} = \emptyset$ and $V_{ST^{\nu'}\varrho_\mu} \subseteq V_{ST^\mu\varrho_\mu}$ then $ran(\sigma) \cap V_{ST^{\nu'}\varrho_\mu} = \emptyset$ so $y \notin V_{G\sigma}$ and, as $dom(\rho) = V_{G\sigma}$, $y(\sigma_1\sigma')_{V_G}\rho = y$. Then:

A. if $y \in V_{t,\phi}$ then $y(\sigma_1\sigma')_{V_G}\rho\rho'' = y\rho'' = y\delta'_{V_{t,\phi}\setminus V_G} = y\delta' = y\sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$, ground term because $V_{y\sigma_1} \subseteq V_{t\sigma_1,\phi\sigma_1} \subseteq V_{G'\sigma_1}$ and $\rho' : V_{G'\sigma_1\sigma'} \to \mathcal{T}_\Sigma$;

B. if $y \notin V_{t,\phi}$ then $y(\sigma_1\sigma')_{V_G}\rho\rho'' = y\rho'' = y\delta'_{V_{t,\phi}\setminus V_G} = y$. As $dom(\sigma_1) \subseteq (V_{u_1} \cup V_{t^\circ}) \subseteq (V_G \cup V_{t,\phi} \cup \bar{x})$ and $y \notin (V_G \cup V_{t,\phi})$ then $y\sigma_1 = y$ so, as $ran(\sigma_1) \cap V_{ST^{\nu'}\varrho_\mu} = \emptyset$, $y\sigma_1 \notin V_{G\sigma_1}$, and $y\sigma_1\sigma'_{V_{G'\sigma_1}} = y \notin V_{G\sigma_1\sigma'_{V_{G'\sigma_1}}}$ so, as $\rho' : V_{G'\sigma_1\sigma'} \to \mathcal{T}_\Sigma$, $y\sigma_1\sigma'_{V_{G'\sigma_1}}\rho' = y = y(\sigma_1\sigma')_{V_G}\rho\rho''$.

(b) As $dom(\rho'') \subseteq (V_{t,\phi} \setminus V_G)$ and, from (a.ii.A), $y \in (V_{t,\phi} \setminus V_G) \implies V_{y\rho''} = \emptyset$ then $dom(\rho'') = V_{t,\phi} \setminus V_G$, hence $\rho'' : V_{t,\phi} \setminus V_G \to \mathcal{T}_\Sigma$.

In exactly the same way as the proof for (a), we have that $ST_1^{\nu'}\varrho_{\nu'}\rho'' = ST_1^{\nu'}\varrho'$ and $ST_2^{\nu'}\varrho_{\nu'}\rho'' = ST_2^{\nu'}\varrho'$.

Now, we prove (a) $dom(\rho'') = V_{(t_1^{\nu'}\varrho_{\nu'},\phi_1^{\nu'}\varrho_{\nu'})}$, (b) $E_0 \vdash \phi_1^{\nu'}\varrho_{\nu'}\rho''$, and (c) $u_1\delta =_E t_1^{\nu'}\varrho_{\nu'}\rho''$:

(a) As $dom(\rho'') = V_{t,\phi} \setminus V_G$, $V_{(t_1^{\nu'},\phi_1^{\nu'})} \subseteq V_{(t_1^\mu,\phi_1^\mu)}$, $V_{(t_1^\mu,\phi_1^\mu)} \setminus V_{(t_1^{\nu'},\phi_1^{\nu'})} \subseteq V^\mu \subseteq V_G$, and $dom(\varrho_\mu) \cap V^\mu = \emptyset$, then $dom(\rho'') = V_{(t,\phi)} \setminus V_G = V_{(t_1^\mu\varrho_\mu,\phi_1^\mu\varrho_\mu)} \setminus V_G = V_{(t_1^{\nu'}\varrho_\mu,\phi_1^{\nu'}\varrho_\mu)} \setminus V_G$. As $\varrho_{\nu'} = (\varrho_\mu\delta)_{\setminus V}$ and $V_{(t_1^{\nu'},\phi_1^{\nu'})} \cap V = \emptyset$, then $V_{(t_1^{\nu'}\varrho_{\nu'},\phi_1^{\nu'}\varrho_{\nu'})} = V_{(t_1^{\nu'}(\varrho_\mu\delta)_{\setminus V},\phi_1^{\nu'}(\varrho_\mu\delta)_{\setminus V})} = V_{(t_1^{\nu'}\varrho_\mu\delta,\phi_1^{\nu'}\varrho_\mu\delta)}$.

Then we prove $V_{(t_1^{\nu'}\varrho_\mu\delta,\phi_1^{\nu'}\varrho_\mu\delta)} = V_{(t_1^{\nu'}\varrho_\mu,\phi_1^{\nu'}\varrho_\mu)} \setminus V_G$, which is trivial, since $\delta : V_G \to \mathcal{T}_\Sigma$.

(b) Immediate, since $E_0 \vdash \phi\sigma\rho'$ and $\phi\sigma\rho' = \phi_1^{\nu'}\varrho_{\nu'}\rho''$.

(c) $u_1\sigma_1 =_B t^\circ\sigma_1$ and $\sigma = \sigma_1\sigma'$ imply $u_1\sigma =_B t^\circ\sigma$ so, as $V_{u_1} \subseteq V_G$, $u_1\sigma_{vars(G)} = u_1\sigma =_B t^\circ\sigma$. As $\rho : V_{G\sigma} \to \mathcal{T}_\Sigma$, so $u_1\sigma_{V_G}\rho$ is a ground term, and $\rho' = \rho \uplus \rho'_1$ then $u_1\delta = u_1\sigma_{V_G}\rho = u_1\sigma_{V_G}\rho' =_B t^\circ\sigma\rho' = t[\bar{x}]_{\bar{q}}\sigma\rho' = t\sigma\rho'[\bar{x}\sigma\rho']_{\bar{q}}$. As $E_0 \vdash \phi^\circ\sigma\rho'$ then $t\sigma\rho'[\bar{x}\sigma\rho']_{\bar{q}} =_{E_0} t\sigma\rho'[t|_{q_1}\sigma\rho', \ldots, t|_{q_l}\sigma\rho']_{\bar{q}} = t\sigma\rho'[t\sigma\rho'|_{\bar{q}}]_{\bar{q}} = t\sigma\rho' = t_1^{\nu'}\varrho_{\nu'}\rho''$, because $t\sigma\rho' = t_1^{\nu'}\varrho_{\nu'}\rho''$, so $u_1\delta =_B t^\circ\sigma\rho' =_{E_0} t_1^{\nu'}\varrho_{\nu'}\rho''$, i.e., $u_1\delta =_E t_1^{\nu'}\varrho_{\nu'}\rho''$.

As $\rho'' : V_{(t_1^{\nu'}\varrho_{\nu'},\phi_1^{\nu'}\varrho_{\nu'})} \to \mathcal{T}_\Sigma$, $E_0 \vdash \phi_1^{\nu'}\varrho_{\nu'}\rho''$, $u_1\delta =_E t_1^{\nu'}\varrho_{\nu'}\rho''$, and $ST_1^{\nu'}\varrho_{\nu'}\rho'' = ST_1^{\nu'}\varrho'$, then there is a derivation rule $\frac{u_1\delta \to w/ST_1^{\nu'}\varrho'}{u_1\delta \to w/\mathtt{match}\, t_1^{\nu'}\varrho_{\nu'}\ \mathtt{s.t.}\ \phi_1^{\nu'}\varrho_{\nu'}\ ?\ ST_1^{\nu'}\varrho_{\nu'} : ST_2^{\nu'}\varrho_{\nu'}}$ in $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$. Now,

$$\frac{\dfrac{\dfrac{F_1}{u_1\delta \to w/ST_1^{\nu'}\varrho'}}{u_1\delta \to w/\mathtt{match}\, t_1^{\nu'}\varrho_{\nu'}\ \mathtt{s.t.}\ \phi_1^{\nu'}\varrho_{\nu'}\ ?\ ST_1^{\nu'}\varrho_{\nu'} : ST_2^{\nu'}\varrho_{\nu'}} \quad \dfrac{F_2}{w \to v_1\delta/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta \to v_1\delta/(\mathtt{match}\, t_1^{\nu'}\varrho_{\nu'}\ \mathtt{s.t.}\ \phi_1^{\nu'}\varrho_{\nu'}\ ?\ ST_1^{\nu'}\varrho_{\nu'} : ST_2^{\nu'}\varrho_{\nu'}); ST^{\nu'}\varrho_{\nu'}}$$

is a c.p.t., $\rho : vars(G\sigma) \to \mathcal{T}_\Sigma$, $\psi\rho$ is satisfiable, $E_0 \vdash \psi_1\delta$ (†), and there are closed proof trees for each open goal in $\Delta\delta$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$ (††), hence $\sigma_{vars(G)}\rho$ is a solution of G.

9. Rule $[i2]$ (if then else):

$G = u_1 \to v_1 \,/\, (\mathtt{match}\ t_1\ \mathtt{s.t.}\ \phi_1\ ?\ ST_1 : ST_2\,; ST)^\mu\varrho_\mu\,(\wedge\,\Delta) \mid \psi_1 \mid V, \mu \rightsquigarrow_{[i1],\sigma_1} (u_1 \to v_1/(ST_2; ST)^\mu\varrho_\mu(\wedge\Delta) \mid \psi_2 \mid V, \mu)\sigma_1 = G'\sigma_1$, let $t = t_1^\mu\varrho_\mu$ and $\phi = \phi_1^\mu\varrho_\mu$, where $abstract_{\Sigma_1}(t) = \langle\lambda\bar{x}.t^\circ; \sigma^\circ; \phi^\circ\rangle$, $t^\circ = t[\bar{x}]_{\bar{q}}$, with $\bar{x} = x_1, \ldots, x_l$ and $\bar{q} = q_1, \ldots, q_l$, $\phi^\circ = (\bigwedge_{i=1}^{l} x_i = t|_{q_i})$, hence $V_{t^\circ} \cup V_{\phi^\circ} = V_t \cup \hat{x}$, $\sigma_1 \in CSU_B(u_1 = t^\circ)$, $\psi_2 = \psi_1 \wedge \neg\phi \wedge \phi^\circ$, so $V_G \subseteq V_{G'}$, $\psi_2\sigma_1$ is satisfiable, and $G'\sigma_1 \rightsquigarrow^+_{\sigma'} nil \mid \psi \mid V, \nu$.

The proof is the same as the one for rule $[i1]$, just replacing ϕ with $\neg\phi$, and exchanging ST_1 and ST_2 everywhere except in the `match` strategy at the beginning "`match` t_1 `s.t.` ϕ_1 ? $ST_1 : ST_2\,; ST$".

10. Rule $[t]$ (transitivity):

$G = u_1 \to v_1/(RA\,;\ ST)^\mu\varrho_\mu\,(\wedge\,\Delta) \mid \psi_1 \mid V, \mu$

$G \rightsquigarrow_{[t]} u_1 \to^1 x_k, x_k \to v_1/(RA\,;\ ST)^\mu\varrho_\mu\,(\wedge\,\Delta) \mid \psi_1 \mid V, \mu = G'$, so $V_G \subseteq V_G \cup \{x_k\} = V_{G'}$, and $G' \rightsquigarrow^+_\sigma nil \mid \psi \mid V, \nu$, where $\nu = (\mu\sigma)_V$, hence $\sigma_{V_G}|\psi$ is a computed answer for G and $\sigma_{V_{G'}}|\psi$ is a computed answer for G'. Let $\rho : V_{G\sigma} \to \mathcal{T}_\Sigma$ such that $\psi\rho$ is satisfiable, let $\delta = \sigma_{V_G}\rho$, $\nu' = (\mu\delta)_V$, and $\varrho_{\nu'} = (\varrho_\mu\delta)_{\setminus V}$, where $dom(\nu') = V$ and $ran(\nu') = \emptyset$, let $\varrho : V_{G'\sigma} \setminus V_{G\sigma} \to \mathcal{T}_\Sigma$, such that $\psi(\rho \uplus \varrho)$ is satisfiable, let $\rho' = \rho \uplus \varrho$, and let $\delta' = \sigma_{V_{G'}}\rho'$. As $V_G \subseteq V_{G'}$ then $\rho' : V_{G'\sigma} \to \mathcal{T}_\Sigma$ and $G\delta' = G\delta$.

By I.H., as $\rho' : V_{G'\sigma} \to \mathcal{T}_\Sigma$ and $\psi\rho'$ is satisfiable, δ' is a solution for G', so $[x_k\delta']_E \in RA^\mu\varrho_\mu\delta' \,@\, [u_1\delta']_E$ and $[v_1\delta']_E \in ST^\mu\varrho_\mu\delta'@[x_k\delta']_E$. This is equivalent, since $G\delta' = G\delta$, to $[x_k\delta']_E \in RA^\mu\varrho_\mu\delta \,@\, [u_1\delta]_E$ and $[v_1\delta]_E \in ST^\mu\varrho_\mu\delta@[x_k\delta']_E$,

i.e., $[x_k\delta']_E \in RA^{\nu'}\varrho_{\nu'}$ @ $[u_1\delta]_E$ and $[v_1\delta]_E \in ST^{\nu'}\varrho_{\nu'}@[x_k\delta']_E$, so there are closed proof trees of the forms $\frac{F_1}{u_1\delta\to x_k\delta'/RA^{\nu'}\varrho_{\nu'}}$ and $\frac{F_2}{x_k\delta'\to v_1\delta/ST^{\nu'}\varrho_{\nu'}}$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$. As there is a rule $\frac{u_1\delta\to x_k\delta'/RA^{\nu'}\varrho_{\nu'}\quad x_k\delta'\to v_1\delta/ST^{\nu'}\varrho_{\nu'}}{u_1\delta\to v_1\delta/(RA\,;\,ST)^{\nu'}\varrho_{\nu'}} \in \mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$, then

$$\frac{\frac{F_1}{u_1\delta\to x_k\delta'/RA^{\nu'}\varrho_{\nu'}}\quad\frac{F_2}{x_k\delta'\to v_1\delta/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta\to v_1\delta/(RA\,;\,ST)^{\nu'}\varrho_{\nu'}}$$

is a c.p.t. with $\rho : vars(G\sigma)\to\mathcal{T}_\Sigma$ and $\psi\rho$ satisfiable, so $\sigma_{vars(G)}\rho$ is a solution of G.

11. Rule $[c]$ (congruence):
$G = u_1|_p \to^1 x_k, u_1[x_k]_p \to v_1/(RA\,;\,ST)^\mu\varrho_\mu\ (\wedge\,\Delta) \mid \psi_1 \mid V, \mu \rightsquigarrow_{[t],\sigma_1} u_i' \to^1 y_{k'}, u_1[y_{k'}]_{p.i} \to v_1/(RA\,;\,ST)^\mu\varrho_\mu\ (\wedge\,\Delta) \mid \psi_1 \mid V, \mu = G'$, where $u_1|_p = f(u_1',\ldots,u_m')$, $u_i'\in\mathcal{H}_\Sigma(\mathcal{X})\setminus\mathcal{X}$, $y_{k'}$ fresh variable, and $\sigma_1 = \{x_k\mapsto u_1|_p[y_{k'}]_i\}$, so $(\mu\sigma_1)_V = \mu$ and $V_{G\sigma_1} = V_{G'}$, and $G' \rightsquigarrow^+_{\sigma'} nil \mid \psi \mid V, \nu$, let $\sigma = \sigma_1\sigma'$, where $\nu = (\mu\sigma')_V = (\mu\sigma)_V$, hence $\sigma_{V_G}|\psi$ is a computed answer for G and $\sigma'_{V_{G'}}|\psi$ is a computed answer for G'.
Let $\rho : V_{G\sigma}\to\mathcal{T}_\Sigma$ such that $\psi\rho$ is satisfiable, let $\delta = \sigma_{V_G}\rho$, $\nu' = (\mu\delta)_V$, and $\varrho_{\nu'} = (\varrho_\mu\delta)_{\setminus V}$, where $dom(\nu') = V$ and $ran(\nu') = \emptyset$. As $V_{G\sigma_1} = V_{G'}$ then $V_{G\sigma} = V_{G\sigma_1\sigma'} = V_{G'\sigma'}$, so also $\rho : V_{G'\sigma'}\to\mathcal{T}_\Sigma$, let $\delta' = \sigma'_{V_{G'}}\rho$.
For every variable $z\in V_G\cap V_{G'}$, as $dom(\sigma_1) = \{x_k\}$ and $x_k\notin V_{G'}$, $z\delta = z\sigma_{V_G}\rho = z\sigma\rho = z\sigma_1\sigma'\rho = z\sigma'\rho = z\sigma'_{V_{G'}}\rho = z\delta'$. As $vars(u_1|_p)\subseteq V_G$ and $vars(u_1[y_{k'}]_{p.i})\subseteq V_{G'}$ then $vars(u_1|_p[]_i)\subseteq V_G\cap V_{G'}$ so $u_1|_p\delta[]_i = u_1|_p\delta'[]_i$.
By I.H., as $\rho : V_{G'\sigma'}\to\mathcal{T}_\Sigma$ and $\psi\rho$ is satisfiable, $\sigma'_{V_{G'}}\rho$ is a solution for G', so $[y_k'\delta']_E \in RA^\mu\varrho_\mu\delta'$ @ $[u_i'\delta']_E$ and $[v_1\delta']_E \in ST^\mu\varrho_\mu\delta'@[u_1[y_k']_{p.i}\delta']_E$. As $V_{G'} = \{y_{k'}\}\cup V_G\setminus\{x_k\}$ and $V_{RA^\mu\varrho_\mu}\cap\{x_k,y_{k'}\} = \emptyset$, so $RA^\mu\varrho_\mu(\sigma_1\sigma')_{V_G} = RA^\mu\varrho_\mu\sigma'_{V_G}$, then $RA^\mu\varrho_\mu\delta' = RA^\mu\varrho_\mu\sigma'_{V_{G'}}\rho = RA^\mu\varrho_\mu\sigma'_{V_G}\rho = RA^\mu\varrho_\mu(\sigma_1\sigma')_{V_G}\rho = RA^\mu\varrho_\mu\sigma_{V_G}\rho = RA^\mu\varrho_\mu\delta = RA^{\nu'}\varrho_{\nu'}$. In the same way, $ST^\mu\varrho_\mu\delta' = ST^{\nu'}\varrho_{\nu'}$. Then, $[y_k'\delta']_E \in RA^{\nu'}\varrho_{\nu'}$ @ $[u_i'\delta']_E$, $[v_1\delta']_E \in ST^{\nu'}\varrho_{\nu'}@[u_1[y_k']_{p.i}\delta']_E$, and there are closed proof trees of the forms (1) $\frac{}{u_i'\delta'\to y_k'\delta'/RA^{\nu'}\varrho_{\nu'}}$ or (2) $\frac{F_1}{u_i'\delta'\to y_k'\delta'/RA^{\nu'}\varrho_{\nu'}}$, and (3) $\frac{F_2}{u_1[y_k']_{p.i}\delta'\to v_1\delta'/ST^{\nu'}\varrho_{\nu'}}$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$.
   - Case (1): $RA^{\nu'} = c^{\nu'}[\gamma]$, so $RA^{\nu'}\varrho_{\nu'} = c^{\nu'}[\gamma(\varrho_{\nu'})_{ran(\gamma)}]$, $c^{\nu'} : l\to r$ *if* ϕ and there exist a substitution η, a position q, and terms $t,t'\in\mathcal{H}_\Sigma$ such that $E_0\vdash\phi\gamma(\varrho_{\nu'})_{ran(\gamma)}\eta$, $t \xrightarrow[c^{\nu'}\gamma(\varrho_{\nu'})_{ran(\gamma)},q,\eta_R]{}{}^1 t'$, so $\frac{}{t\to t'/RA^{\nu'}\varrho_{\nu'}}$ is a derivation rule in $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$, $u_i'\delta' =_E t$, and $t' =_E y_k'\delta'$. By definition of $\to^1_R$, also $u_1\delta|_p[t]_i \xrightarrow[c^{\nu'}\gamma(\varrho_{\nu'})_{ran(\gamma)},i.q,\eta_R]{}{}^1 u_1\delta|_p[t']_i$ so there is a derivation rule $\frac{}{u_1\delta|_p[t]_i\to u_1\delta|_p[t']_i/RA^{\nu'}\varrho_{\nu'}}\in\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$.
     Now, $u_i'\delta = u_i'\delta' =_E t$, so $u_1\delta|_p[t]_i =_E u_1\delta|_p[u_i'\delta]_i = u_1|_p[u_i']_i\delta = u_1|_p\delta$, and $x_k\in V_G$, so $x_k\delta = x_k\sigma_{vars(G)}\rho = x_k\sigma\rho = x_k\sigma_1\sigma'\rho = u_1|_p[y_{k'}]_i\sigma'\rho = u_1\sigma'|_p[y_{k'}\sigma']_i\rho = u_1\sigma|_p[y_{k'}\sigma']_i\rho = u_1\sigma\rho|_p[y_{k'}\sigma'\rho]_i = u_1\delta|_p[y_{k'}\delta']_i =_E u_1\delta|_p[t']_i$.
     If we apply the previous derivation rule, with $u_1\delta|_p[t]_i =_E u_1|_p\delta$ and $x_k\delta =_E u_1\delta|_p[t']_i$, then we get the c.p.t. $\frac{}{u_1|_p\delta\to x_k\delta/RA^{\nu'}\varrho_{\nu'}}$, so $[x_k\delta]_E\in RA^{\nu'}\varrho_{\nu'}@[u_1|_p\delta]_E$.

– Case (2):
$RA^{\nu'} = c^{\nu'}[\gamma]\{ST_1^{\nu'}, \ldots, ST_m^{\nu'}\}$, $RA^{\nu'}\varrho_{\nu'} = c^{\nu'}[\gamma(\varrho_{\nu'})_{ran(\gamma)}]\{\overline{ST}^{\nu'}\varrho_{\nu'}\}$,
$c^{\nu'}\gamma(\varrho_{\nu'})_{ran(\gamma)}$ has the form $l \to r$ *if* $\bigwedge_{j=1}^{m} l_j \to r_j \mid \phi$ and there exist a substitution η, a term $t \in \mathcal{H}_\Sigma$, and a position $q \in pos(t)$ such that $t|_q = l\eta$ and $E_0 \vdash \phi\eta$, so there is a derivation rule $\frac{l_1\eta \to r_1\eta/ST_1^{\nu'}\varrho_{\nu'}\eta \cdots l_m\eta \to r_m\eta/ST_m^{\nu'}\varrho_{\nu'}\eta}{t \to t[r\eta]_q/RA^{\nu'}\varrho_{\nu'}} \in \mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$, T_j is a c.p.t. with root $l_j\eta \to r_j\eta/ST_j^{\nu'}\varrho_{\nu'}\eta$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$, for $1 \le j \le m$, $u_i'\delta' =_E t$, and $t[r\eta]_q =_E y_k'\delta'$.
We take $w = u_1|_p\delta[t]_i$ and the position $i.q$. Then, as $E_0 \vdash \phi\eta$ and $w|_{i.q} = t|_q = l\eta$, there is also a derivation rule $\frac{l_1\eta \to r_1\eta/ST_1^{\nu'}\varrho_{\nu'}\eta \cdots l_m\eta \to r_m\eta/ST_m^{\nu'}\varrho_{\nu'}\eta}{w \to w[r\eta]_{i.q}/RA^{\nu'}\varrho_{\nu'}} \in \mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$.
We have $u_i'\delta' =_E t$ and $t[r\eta]_q =_E y_k'\delta'$. From the previous subcase we also know that $u_i'\delta = u_i'\delta'$ and $x_k\delta = u_1|_p\delta[y_{k'}\delta']_i$.
Then $w = u_1|_p\delta[t]_i =_E u_1|_p\delta[u_i'\delta']_i = u_1|_p\delta[u_i'\delta]_i = u_1|_p\delta$ and $w[r\eta]_{i.q} = (u_1|_p\delta[t]_i)[r\eta]_{i.q} = u_1|_p\delta[t[r\eta]_q]_i =_E u_1|_p\delta[y_k'\delta']_i$.
As $\sigma_1 = \{x_k \mapsto u_1|_p[y_{k'}]_i\}$, $x_k \in V_G$, $V_{u_1[y_{k'}]_{p.i}} \subseteq V_{G'}$, so $V_{u_1|_p[y_{k'}]_i} \subseteq V_{G'}$, and $u_1|_p\delta[]_i = u_1|_p\delta'[]_i$, then $x_k\delta = x_k\sigma_{V_G}\rho = x_k(\sigma_1\sigma')_{V_G}\rho = u_1|_p[y_{k'}]_i\sigma'\rho = u_1|_p[y_{k'}]_i\sigma'_{V_{G'}}\rho = u_1|_p[y_{k'}]_i\delta' = u_1|_p\delta'[y_{k'}\delta']_i = u_1|_p\delta[y_{k'}\delta']_i$ then $w[r\eta]_{i.q} =_E x_k\delta$ so, as $w =_E u_1|_p\delta$, we can apply the derivation rule with $u_1|_p\delta$ and $x_k\delta$ and complete a c.p.t. with $T_1, \ldots, T_m$, yielding $\frac{F_1}{u_1|_p\delta \to x_k\delta/RA^{\nu'}\varrho_{\nu'}}$, hence $[x_k\delta]_E \in RA^{\nu'}\varrho_{\nu'}@[u_1|_p\delta]_E$.

As $V_{(v_1,u_1)} \subseteq V_G \cap V_{G'}$ then $v_1\delta' = v_1\delta$ and $v_1\delta' = v_1\delta$, so $u_1[x_k]_p\delta = u_1\delta[x_k\delta]_p = u_1\delta[u_1|_p\delta[y_{k'}\delta']_i]_p = u_1\delta[y_{k'}\delta']_{p.i} = u_1\delta'[y_{k'}\delta']_{p.i}$, and the c.p.t. (3) can also be written as $\frac{F_2}{u_1[x_k]_p\delta \to v_1\delta/ST^{\nu'}\varrho_{\nu'}}$, hence $[v_1\delta]_E \in ST^{\nu'}\varrho_{\nu'}@[u_1[x_k]_p\delta]_E$. As also $[x_k\delta]_E \in RA^{\nu'}\varrho_{\nu'}@[u_1|_p\delta]_E$, either for case (1) or (2), and $\psi\rho$ is satisfiable then $\sigma_{vars(G)}\rho$ is a solution of G.

12. Rule $[r]$ (rule application):

We prove this case for conditional rules. For rules without rewrite conditions, the proof is the same just with the part dealing with the conditions removed from it.

$G = u|_p \to^1 x_k, u[x_k]_p \to v/(c[\gamma_r]\{ST_1, \ldots, ST_m\}\ ;\ ST)^\mu\varrho_\mu\ (\wedge\ \Delta) \mid \psi_1 \mid V, \mu \rightsquigarrow_{[r],\sigma_1} (\bigwedge_{i=1}^{n}(l_i\gamma \to r_i\gamma/ST_i^\mu\varrho_\mu; \mathtt{idle}) \wedge u[r\gamma]_p \to v\ /ST^\mu\varrho_\mu\ (\wedge\ \Delta) \mid \psi_2)\sigma_1 \mid V, (\mu\sigma_1)_V = G'\sigma_1$, where:

- $\gamma = (\gamma_r^\mu\varrho_\mu)_{dom(\gamma_r^\mu)}$ (so $ran(\gamma) \subseteq V_G$), $c \in R$, $c_0 \in c_B \subseteq R_B$ has the form $c : l^c \to r^c$ *if* $\bigwedge_{i=1}^{n}(l_i^c \to r_i^c) \mid \phi^c$, $c_{\gamma'} : l \to r$ *if* $\bigwedge_{i=1}^{n}(l_i \to r_i) \mid \phi$ is a fresh version with some renaming γ' of $c_0^\mu \in R_B^\mu$, with $dom(\gamma') = vars(c_0^\mu) \setminus (dom(\gamma_r) \uplus V^\mu)$, so $c_{\gamma'} = c_0^\mu\gamma'$, let $l' = l\gamma$;
- $abstract_{\Sigma_1}(u|_p) = \langle\lambda\bar{u}.u^\circ; \sigma_u^\circ; \phi_u^\circ\rangle$, $u^\circ = u|_p[\bar{x}]_{\bar{p}}$, with $\bar{x} = x_1, \ldots, x_u$ and $\bar{p} = p_1, \ldots, p_u$, $\phi_u^\circ = (\bigwedge_{j=1}^{u} x_j = u|_{p.p_j})$;
- $abstract_{\Sigma_1}(l') = \langle\lambda\bar{y}.l^\circ; \sigma^\circ; \phi^\circ\rangle$, $l^\circ = l'[\bar{y}]_{\bar{q}}$, with $\bar{y} = y_1, \ldots, y_l$ and $\bar{q} = q_1, \ldots, q_l$, $\phi^\circ = (\bigwedge_{i=1}^{l} y_i = l'|_{q_i})$;
- $\sigma_1' \in CSU_B(u^\circ = l^\circ)$, $\sigma_1 = \sigma_1' \cup \{x_k \mapsto r\gamma\sigma_1\sigma_1'\}$, $\psi_2 = \psi_1 \wedge \phi^\circ \wedge \phi_u^\circ \wedge \phi\gamma$, $\psi_2\sigma_1$ is satisfiable;

Then $G'\sigma_1 \rightsquigarrow^+_{\sigma'} nil \mid \psi \mid V, \nu$, let $\sigma = \sigma_1\sigma'$, where $\nu = (\mu\sigma)_V = (\mu\sigma_1\sigma')_V = (\mu\sigma'_1\sigma')_V$, so $\sigma_{V_G} \mid \psi$ is a computed answer for G and $\sigma'_{V_{G'\sigma_1}} \mid \psi$ is a computed answer for $G'\sigma_1$.

As $\gamma = (\gamma^\mu_r \varrho_\mu)_{dom(\gamma^\mu_r)}$ then $dom(\gamma_r) = dom(\gamma^\mu_r) = dom(\gamma)$.
Let $\rho : V_{G\sigma} \to \mathcal{T}_\Sigma$ be a substitution such that $\psi\rho$ is satisfiable, let $\delta = \sigma_{V_G}\rho$ and $\varrho_{\nu'} = (\varrho_\mu\delta)_{\setminus V}$, so $\delta : V_G \to \mathcal{T}_\Sigma$, $\rho_1 = \rho_{V_{G'\sigma}}$, so also $\psi\rho_1$ is satisfiable, and let $\nu' = (\nu\rho)_V$, where $dom(\nu') = V$ and $ran(\nu') = \emptyset$. As $dom(\rho) = V_{G\sigma}$ then $dom(\rho_1) = V_{G\sigma} \cap V_{G'\sigma}$. Let $\rho_2 = \rho_{V_{G\sigma} \setminus V_{G'\sigma}}$, so $\rho = \rho_1 \uplus \rho_2$, and let $\rho'_1 : V_{G'\sigma} \setminus V_{G\sigma} \to \mathcal{T}_\Sigma$, so $dom(\rho_1) \cap dom(\rho'_1) = \emptyset$ and $dom(\rho_1) \cup dom(\rho'_1) = V_{G'\sigma}$, such that $\psi(\rho_1 \uplus \rho'_1)$ is satisfiable, and let $\rho' = \rho_1 \uplus \rho'_1$, so $\rho' : V_{G'\sigma} \to \mathcal{T}_\Sigma$.

We prove several intermediate results:

- As $\nu = (\mu\sigma)_V$, $V^\mu \subseteq V_G \cap V_{G'}$, and $dom(\rho_1) = V_{G\sigma} \cap V_{G'\sigma}$ then $V^\nu \subseteq dom(\rho_1)$ so, as $dom(\nu') = V$ and $ran(\nu') = \emptyset$, $\nu' = (\nu\rho)_V = (\nu\rho_1)_V = (\nu\rho')_V$. Also, as $dom(\mu) \subseteq V$, $\nu' = (\nu\rho)_V = ((\mu\sigma)_V\rho)_V = (\mu\sigma\rho)_V = \mu(\sigma\rho)_{V^\mu}$.
- As $dom(\rho'_1) = V_{G'\sigma} \setminus V_{G\sigma}$ and $dom(\rho_1) = V_{G\sigma} \cap V_{G'\sigma} \subseteq V_{G\sigma}$, then $\rho'_{V_{G\sigma}} = (\rho_1 \uplus \rho'_1)_{V_{G\sigma}} = (\rho_1)_{V_{G\sigma}} = \rho_1$.
- As $\delta_{V^\mu} = (\sigma_{V_G}\rho)_{V^\mu}$, $\rho : V_{G\sigma} \to \mathcal{T}_\Sigma$, and $V^\mu \subseteq V_G$, then $\delta_{V^\mu} = (\sigma\rho)_{V^\mu}$, $ran(\delta_{V^\mu}) = \emptyset$, and $dom(\delta_{V^\mu}) = V^\mu$ $(= (V \setminus dom(\mu)) \cup ran(\mu)$, so $ran(\mu) \subseteq dom(\delta_{V^\mu}))$. Then $\nu' = (\mu\sigma_{V_G}\rho)_V = \mu\delta_{V^\mu}$ and $c_0^{\nu'} = c_0\nu' = c_0\mu\delta_{V^\mu}$.

As $\sigma'_{V_{G'\sigma_1}} \mid \psi$ is a computed answer for $G'\sigma_1$, $\rho' : V_{G'\sigma_1\sigma'} \to \mathcal{T}_\Sigma$, and $\psi\rho'$ is satisfiable then, by I.H., $\sigma'_{V_{G'\sigma_1}}\rho'$ is a solution for $G'\sigma_1$, let $\delta' = \sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$ and $\varrho' = (\varrho_\mu\delta')_{\setminus V}$, meaning that:

(a) $E_0 \vdash \psi_2\delta'$,

(b) there are closed proof trees for each open goal in $\Delta\delta'$, with respect to $\mathcal{D}^{(\nu\rho')_V}_{\mathcal{R},Call_\mathcal{R}}$ $(= \mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$, we use ν' instead of $(\nu\rho')_V$ in (c) and (d)),

(c) $[v\delta']_E \in ST^{\nu'}\varrho'@[u[r\gamma]_p\delta']_E$, i.e., $[v\delta']_E \in ST^{\nu'}\varrho'@[u\delta'[r\gamma\delta']_p]_E$, and

(d) $[r_i\gamma\delta']_E \in ST^{\nu'}_i\varrho'@[l_i\gamma\delta']_E$, for $1 \leq i \leq n$.

Then:

(a) i. $V_{\psi_2} \subseteq V_{G'}$ implies $V_{\psi_2\sigma_1} \subseteq V_{G'\sigma_1}$ and $V_{\psi_2\sigma} \subseteq V_{G'\sigma}$, so $\psi_2\delta' = \psi_2\sigma_1\sigma'_{V_{G'\sigma_1}}\rho' = \psi_2\sigma_1\sigma'\rho' = \psi_2\sigma\rho'$, hence $E_0 \vdash \psi_2\sigma\rho'$, where $\psi_2\sigma\rho'$ is ground, because $V_{\psi_2\sigma} \subseteq V_{G'\sigma}$ and $\rho' : V_{G'\sigma} \to \mathcal{T}_\Sigma$. As $\psi_2 = \psi_1 \wedge \phi^\circ \wedge \phi^\circ_u \wedge \phi\gamma$, then $\psi_1\delta' = \psi_1\sigma\rho'$, $E_0 \vdash \psi_1\sigma\rho'$, $E_0 \vdash \phi^\circ\sigma\rho'$, $E_0 \vdash \phi^\circ_u\sigma\rho'$, and $E_0 \vdash \phi\gamma\sigma\rho'$, all ground formulas.

ii. Also as $\psi_2 = \psi_1 \wedge \phi^\circ \wedge \phi\gamma$, so $V_{\psi_1} \subseteq V_G \cap V_{G'}$ hence $V_{\psi_1\sigma} \subseteq V_{G\sigma} \cap V_{G'\sigma}$, and $dom(\rho_1) = V_{G\sigma} \cap V_{G'\sigma}$ imply $\psi_1\sigma\rho_1 \in \mathcal{T}_\Sigma$. Then, as $\rho' = \rho_1 \uplus \rho'_1$, we have $\psi_1\sigma\rho' = \psi_1\sigma(\rho_1 \uplus \rho'_1) = \psi_1\sigma\rho_1 = \psi_1\sigma(\rho_1 \uplus \rho_2) = \psi_1\sigma\rho = \psi_1\delta$, so $E_0 \vdash \psi_1\delta$ (1).

iii. As $\psi_1\delta' = \psi_1\sigma\rho'$ and $\psi_1\sigma\rho' = \psi_1\delta$ then $\psi_1\delta' = \psi_1\delta$.

(b) As in subcases (a)-ii and (a)-iii, $V_\Delta \subseteq V_G \cap V_{G'}$ implies $\Delta\delta' = \Delta\delta$, and the same closed proof trees are valid for each open goal in $\Delta\delta$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$ (2).

(c) i. Again, $V_{v,u[\,]_p} \subseteq V_G \cap V_{G'}$ implies that $v\delta' = v\delta$ and $u\delta'[\,]_p = u\delta[\,]_p$.

ii. We prove that $ST^{\nu'}\varrho' = ST^{\nu'}\varrho_{\nu'}$.
As $\varrho_{\nu'} = (\varrho_\mu\delta)_{\setminus V}$, $\delta = \sigma_{V_G}\rho$, $\sigma = \sigma_1\sigma'$, $\delta' = \sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$, $\varrho' = (\varrho_\mu\delta')_{\setminus V}$, and $V_{ST^{\nu'}} \cap V = \emptyset$, this is the same as $ST^{\nu'}\varrho_\mu(\sigma_1\sigma')_{V_G}\rho = ST^{\nu'}\varrho_\mu\sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$.
Let $x \in V_{ST^{\nu'}\varrho_\mu}$. As $V_{ST^{\nu'}\varrho_\mu} \subseteq V_{ST^\mu\varrho_\mu} \subseteq V_G \cap V_{G'}$, then $x \in V_G \cap V_{G'}$ and $V_{x\sigma_1} \subseteq V_{G\sigma_1} \cap V_{G'\sigma_1} \subseteq V_{G'\sigma_1}$, so $x(\sigma_1\sigma')_{V_G} = x\sigma_1\sigma' = x\sigma_1\sigma'_{V_{G'\sigma_1}}$.
Also $x(\sigma_1\sigma')_{V_G} = x\sigma$, hence $V_{x(\sigma_1\sigma')_{V_G}} \subseteq V_{G\sigma}$. Then, as $\rho : V_{G\sigma} \to \mathcal{T}_\Sigma$, $x(\sigma_1\sigma')_{V_G}\rho$ is ground, so $x(\sigma_1\sigma')_{V_G}\rho = x\sigma_1\sigma'_{V_{G'\sigma_1}}\rho = x\sigma_1\sigma'_{V_{G'\sigma_1}}(\rho \cup \rho'_1) = x\sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$.

iii. As in subcase (a)-i, $V_{r\gamma} \subseteq V_{G'}$ implies $r\gamma\delta' = r\gamma\sigma\rho'$.
As $x_k\sigma_1 = r\gamma\sigma_1$ and $\sigma = \sigma_1\sigma'$ then $x_k\sigma = r\gamma\sigma$ so, as $x_k \in V_G$, $r\gamma\sigma = x_k\sigma_{V_G}$ and $r\gamma\sigma\rho' = x_k\sigma_{V_G}\rho'$, ground terms. But, as $V_{x_k\sigma_{V_G}} \subseteq V_{G\sigma}$ then $x_k\sigma_{V_G}\rho' = x_k\sigma_{V_G}(\rho_1 \uplus \rho'_1) = x_k\sigma_{V_G}\rho_1 = x_k\sigma_{V_G}(\rho_1 \uplus \rho_2) = x_k\sigma_{V_G}\rho = x_k\delta$, so $r\gamma\sigma\rho' = x_k\delta$ (3).

From (i)-(iii), $[v\delta]_E \in ST^{\nu'}\varrho_{\nu'}@[u\delta[x_k\delta]_p]_E$, i.e, $[v\delta]_E \in ST^{\nu'}\varrho_{\nu'}@[u[x_k]_p\delta]_E$ (4), holds.

(d) Using the same proof as in the previous case, $[r_i\gamma\delta']_E \in ST_i^{\nu'}\varrho'@[l_i\gamma\delta']_E$, $V_{l_i\gamma, r_i\gamma} \subseteq V_{G'}$, and $V_{ST_i^{\nu'}\varrho'} \subseteq V_G \cap V_{G'}$ imply $[r_i\gamma\sigma\rho']_E \in ST_i^{\nu'}\varrho_{\nu'}@[l_i\gamma\sigma\rho']_E$, for $1 \leq i \leq n$, where each term and strategy are ground (5).

Now:

(a) $V_{u|_p} \subseteq V_G$ imply $u|_p\sigma_{V_G} = u|_p\sigma$, hence $u|_p\sigma_{V_G}\theta = u|_p\sigma\theta$, and $u^\circ\sigma'_1 =_B l^\circ\sigma'_1$ imply $u^\circ\sigma\theta =_B l^\circ\sigma\theta$.

(b) As $E_0 \vdash \phi^\circ_u\sigma\theta$, ground formula, then $u^\circ\sigma\theta = u|_p[\bar{x}]_{\bar{p}}\sigma\theta = u|_p\sigma\theta[\bar{x}\sigma\theta]_{\bar{p}} =_{E_0} u|_p\sigma\theta[u|_{p.\bar{p}}\sigma\theta]_{\bar{p}} = u|_p\sigma\theta$, all ground terms.

(c) As $E_0 \vdash \phi^\circ\sigma\theta$, ground formula, then $l^\circ\sigma\theta = l'[\bar{y}]_{\bar{q}}\sigma\theta = l'\sigma\theta[\bar{y}\sigma\theta]_{\bar{q}} =_{E_0} l'\sigma\theta[l'|_{\bar{q}}\sigma\theta]_{\bar{q}} = l'\sigma\theta$, all ground terms (6).

(d) As $\rho : V_{G\sigma} \to \mathcal{T}_\Sigma$, so $u|_p\sigma_{V_G}\rho$ $(= u|_p\delta)$ is a ground term, then $u|_p\delta = u|_p\sigma_{V_G}\rho = u|_p\sigma_{V_G}\theta = u|_p\sigma\theta =_{E_0} u^\circ\sigma\theta =_B l^\circ\sigma\theta =_{E_0} l'\sigma\theta = l\gamma\sigma\theta$ (7).

Recall that $c[\gamma_r]\{\overline{ST}\}^{\nu'}\varrho_{\nu'} = c\nu'[(\gamma_r^{\nu'}\varrho_{\nu'})_{dom(\gamma_r^{\nu'})}]\{\overline{ST}^{\nu'}\varrho_{\nu'}\}$. Now, we prove $[x_k\delta]_E \in c[\gamma_r]\{\overline{ST}\}^{\nu'}\varrho_{\nu'}$ @ $[u|_p\delta]_E$.
As $dom(\gamma') = vars(c_0^\mu) \setminus (dom(\gamma) \uplus V^\mu)$, $c_0^\mu = c_0\mu$, and $dom(\delta_{V^\mu}) = V^\mu$, then $V_{c_0\mu} \subseteq dom(\delta_{V^\mu}) \uplus dom(\gamma) \uplus dom(\gamma')$. Then, as $c_0\nu' = c_0\mu\delta_{V^\mu}$ and δ_{V^μ} is a ground substitution, it follows that $V_{c_0\nu'} = dom(\gamma) \uplus dom(\gamma')$, hence $V_{c_0\nu'(\gamma\delta)_{dom(\gamma)}} = V_{ran(\gamma)\delta_{ran(\gamma)}} \cup V_{dom(\gamma')(\gamma\delta)_{dom(\gamma)}}$.
Then:

- As $(\gamma\delta)_{dom(\gamma)}$ is a ground substitution, if z is a variable in $ran(\gamma)$ then $z\delta_{ran(\gamma)}$ is a ground term, so $V_{ran(\gamma)\delta_{ran(\gamma)}} = \emptyset$.
- As $dom(\gamma) \cap dom(\gamma') = \emptyset$, if z is a variable in $dom(\gamma')$ then $z(\gamma\delta)_{dom(\gamma)} = z$, so $V_{dom(\gamma')(\gamma\delta)_{dom(\gamma)}} = dom(\gamma')$.

In conclusion, $V_{c_0\nu'\gamma\delta_{ran(\gamma)}} = dom(\gamma')$.
Let $\nu'' = \nu'(\gamma\delta)_{dom(\gamma)}$ $(= \nu' \uplus (\gamma\delta)_{dom(\gamma)}$ because $dom(\nu') \cap dom(\gamma) = V \cap dom(\gamma) = \emptyset)$. We must find a substitution $\tau : V_{c_0\nu''} \to \mathcal{T}_\Sigma$ such that $E_0 \vdash \phi^c\nu''\tau$. Let $\theta = \rho_2 \uplus \rho_1 \uplus \rho'_1 (= \rho_2 \uplus \rho')$, so $dom(\theta) = V_{G\sigma} \cup V_{G'\sigma}$. We choose

$\tau = (\gamma'\sigma\theta)_{dom(\gamma')} = \gamma'(\sigma\theta)_{ran(\gamma')}$, so $dom(\tau) = dom(\gamma') = V_{c_0\nu''}$ and $(c_0\nu'')\tau = (c_0\nu'')\gamma'\sigma\theta$.

We prove that τ is a ground substitution by proving that $(c_0\nu'')\gamma'\sigma\theta$ is ground. Let $\delta'' = \delta_{V^\mu}\gamma\delta_{ran(\gamma)}$. As δ_{V^μ} and $\gamma\delta_{ran(\gamma)}$ are ground substitutions, $dom(\delta_{V^\mu}) \cap (dom(\gamma') \cup ran(\gamma')) = \emptyset$, and $V_{c_0\nu'} = dom(\gamma) \uplus dom(\gamma')$, then $(c_0\nu'')\gamma' = c_0\nu'(\gamma\delta_{ran(\gamma)} \uplus \gamma') = c_0^\mu\delta_{V^\mu}(\gamma\delta_{ran(\gamma)} \uplus \gamma') = c_0^\mu\delta_{V^\mu}\gamma'\gamma\delta_{ran(\gamma)} = c_0^\mu\gamma'\delta_{V^\mu}\gamma\delta_{ran(\gamma)} = c_0^\mu\gamma'\delta'' = c_{\gamma'}\delta''$. If $z \in V_{c_{\gamma'}\delta''}$ then, as δ_{V^μ} is ground, either $z \in V_{G'}$ or $z \in V_{l'} \setminus V_{G'}$, because l' is the only term of $c_{\gamma'}\gamma$ that does not appear in G'. then:

- If $z \in V_{G'}$ then $V_{z\sigma} \subseteq V_{G'\sigma}$, so $z\sigma\theta$ is a ground term because $dom(\theta) = V_{G\sigma} \cup V_{G'\sigma}$.
- If $z \in V_{l'} \setminus V_{G'}$, as $z \in V_{l'}$ and, by (6), $l'\sigma\theta$ is ground, then $z\sigma\theta$ is a ground term.

Now, we prove $E_0 \vdash \phi^c\nu''\tau$.

- As $ran(\gamma) \subseteq V_G$ and δ is a ground substitution, then $\gamma\delta_{ran(\gamma)}$ is a ground substitution so, as $c_0^{\nu'} = c_0\mu\delta_{V^\mu}$ and $\nu'' = \nu'\gamma\delta_{ran(\gamma)}$, $\phi^c\nu''\tau = \phi^c\mu\delta_{V^\mu}\gamma\delta_{ran(\gamma)}\tau = \phi^c\mu\delta_{V^\mu}(\gamma\delta_{ran(\gamma)} \uplus \tau)$.
- As δ_{V^μ} is a ground substitution, $V_{\phi^c\mu\delta_{V^\mu}} \subseteq V_{c_0^\mu\delta_{V^\mu}} = dom(\gamma) \uplus dom(\gamma')$, $dom(\tau) = dom(\gamma')$, and $dom(\gamma\delta_{ran(\gamma)}) = dom(\gamma)$ then $\phi^c\mu\delta_{V^\mu}(\gamma\delta_{ran(\gamma)} \uplus \tau) = \phi^c\mu(\delta_{V^\mu} \uplus \gamma\delta_{ran(\gamma)} \uplus \tau) = \phi^c\mu((\sigma\rho)_{V^\mu} \uplus \gamma(\sigma\rho)_{ran(\gamma)} \uplus \tau) = \phi^c\mu((\sigma\theta)_{V^\mu} \uplus \gamma(\sigma\theta)_{ran(\gamma)} \uplus \gamma'(\sigma\theta)_{ran(\gamma')})$, because as $\phi^c\mu\delta''\tau$ is ground, it remains the same if we substitute the occurrences of ρ, ground substitution, with $\theta = \rho \uplus \rho'$.
- As $(\sigma\theta)_{V^\mu}$ is ground then $\phi^c\mu((\sigma\theta)_{V^\mu} \uplus \gamma(\sigma\theta)_{ran(\gamma)} \uplus \gamma'(\sigma\theta)_{ran(\gamma')}) = \phi^c\mu(\gamma' \uplus \gamma)\sigma\theta$, the last equality because as the formula is ground, no new instantiation will come from an unrestricted substitution.
- As $dom(\gamma) \cap dom(\gamma') = \emptyset$ and $dom(\gamma) \cap ran(\gamma') = \emptyset$, we can apply the substitutions one after the other, so $\phi^c\mu(\gamma' \uplus \gamma)\sigma\theta = \phi^c\mu\gamma'\gamma\sigma\theta = \phi\gamma\sigma\theta$.
- As $V_{\phi\gamma\sigma} \subseteq V_{G'\sigma}$, $\rho' : V_{G'\sigma} \to \mathcal{T}_\Sigma$, and $\theta = \rho_2 \uplus \rho'$ then $\phi\gamma\sigma\theta = \phi\gamma\sigma\rho'$.

Joining all the equalities, we get $\phi^c\nu''\tau = \phi\gamma\sigma\rho'$. Then, as $E_0 \vdash \phi\gamma\sigma\rho'$, also $E_0 \vdash \phi^c\nu''\tau$.

Now, we prove the existence of a needed derivation rule in $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$. As $\varrho_{\nu'} = (\varrho_\mu\delta)_{\setminus V}$ and $\nu' = (\mu\delta)_V$, both ground, $\bigcup_{i=1}^m V^\mu_{ST_i}\varrho_\mu \subseteq V_G$, and $\delta : V_G \to \mathcal{T}_\Sigma$, then $ST_i^\mu\varrho_\mu\delta = ST_i^{\nu'}\varrho_{\nu'}$ and $V_{ST_i^{\nu'}\varrho_{\nu'}} = \emptyset$, for $1 \le i \le m$, and $(c[\gamma_r])^\mu\varrho_\mu\delta = c^\mu[(\gamma_r^\mu\varrho_\mu)_{dom(\gamma_r^\mu)}]\delta = c^\mu[\gamma]\delta = c^{\nu'}[(\gamma\delta)_{dom(\gamma)}]$.

Recall that $c_0 \in R_B$ has the form $c : l^c \to r^c$ *if* $\bigwedge_{i=1}^n(l_i^c \to r_i^c) \mid \phi^c$ and $\nu'' = \nu'(\gamma\delta)_{dom(\gamma)}$. There are two cases to consider now:

(a) $c_0 \in R$:
as $\tau : V_{c_0\nu''} \to \mathcal{T}_\Sigma$, $E_0 \vdash \phi^c\nu''\tau$, $l^c\nu''\tau$ and $r^c\nu''\tau$ are terms in $\mathcal{H}_\Sigma$, ϵ is a position in $pos(l^c\nu''\tau)$ such that $(l^c\nu''\tau)|_\epsilon = l^c\nu''\tau$, and $\overline{ST}^{\nu'}\varrho_{\nu'}$ are ground strategies, then there is a derivation rule

$$\frac{l_1^c\nu''\tau \to r_1^c\nu''\tau/ST_1^{\nu'}\varrho_{\nu'} \cdots l_m^c\nu''\tau \to r_m^c\nu''\tau/ST_m^{\nu'}\varrho_{\nu'}}{l^c\nu''\tau \to r^c\nu''\tau/c[(\gamma\delta)_{dom(\gamma)}]\{\overline{ST}^{\nu'}\varrho_{\nu'}\}}$$

in $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$.

(b) $c_0 \notin R$:
then there is a rule $c_1 : f(t,t') \to t''$ *if* $C \in R$ such that c_0 has the form $c : f(x_s, f(t,t')) \to f(x_s,t'')$ *if* C, where $dom(\gamma_r) \subseteq V_{c_1}$ and $C = \bigwedge_{i=1}^n (l_i^c \to r_i^c) \mid \phi^c$. Let $\tau' = \tau_{V_{c_1\nu''}}$. As $V_{c_1} \subset V_{c_0}$ and $\tau : V_{c_0\nu''} \to \mathcal{T}_\Sigma$ then $\tau' : V_{c_1\nu''} \to \mathcal{T}_\Sigma$. Also, as $V_{c_1} \subset V_{c_0}$ and $E_0 \vdash \phi^c\nu''\tau$ then $E_0 \vdash \phi^c\nu''\tau'$.
As $l^c\nu''\tau$ is a term in $\mathcal{H}_\Sigma$, 2 is a position in $pos(l^c\nu''\tau)$ such that $(l^c\nu''\tau)|_2 = f(t,t')\nu''\tau$, $E_0 \vdash \phi^c\nu''\tau'$, $t''\nu''\tau' = t''\nu''\tau$, and $\overline{ST}^{\nu'}\varrho_{\nu'}$ are ground strategies, then there is a derivation rule

$$\frac{l_1^c\nu''\tau \to r_1^c\nu''\tau/ST_1^{\nu'}\varrho_{\nu'} \cdots l_m^c\nu''\tau \to r_m^c\nu''\tau/ST_m^{\nu'}\varrho_{\nu'}}{l^c\nu''\tau \to l^c\nu''\tau[t''\nu''\tau]_2/c[(\gamma\delta)_{dom(\gamma)}]\{\overline{ST}^{\nu'}\varrho_{\nu'}\}}$$

in $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$. As $l^c[]_2 = r^c[]_2 = f(x_s,[])$, and $r^c\nu''\tau[t''\nu''\tau]_2 = r^c[t'']_2\nu''\tau = r^c\nu''\tau$, this is the same as

$$\frac{l_1^c\nu''\tau \to r_1^c\nu''\tau/ST_1^{\nu'}\varrho_{\nu'} \cdots l_m^c\nu''\tau \to r_m^c\nu''\tau/ST_m^{\nu'}\varrho_{\nu'}}{l^c\nu''\tau \to r^c\nu''\tau/c[(\gamma\delta)_{dom(\gamma)}]\{\overline{ST}^{\nu'}\varrho_{\nu'}\}},$$

so in both cases we have the same derivation rule. Now, as:

- $\nu'' = \nu' \uplus (\gamma\delta)_{dom(\gamma)}$ is ground, $\nu' = \mu\delta_{V^\mu}$, $\delta = \sigma_{V_G}\rho$, $\theta = \rho \uplus \rho_1'$, and $dom(\delta_{V^\mu}) = V^\mu$,
- $\tau = \gamma'(\sigma\theta)_{ran(\gamma')}$ and $\delta_{V^\mu} \uplus (\gamma\delta)dom(\gamma)$ are ground substitutions,
- $c_0 : l^c \to r^c$ *if* $\bigwedge_{i=1}^n (l_i^c \to r_i^c) \mid \phi^c$ and $c_0\nu''\tau$ is ground,
- $c_{\gamma'} : l \to r$ *if* $\bigwedge_{i=1}^n (l_i \to r_i) \mid \phi$, and
- $c_{\gamma'}$ is a fresh version of c_0^μ except for $dom(\gamma) \uplus dom(\delta_{V^\mu})$, with renaming $\gamma' : vars(c_0^\mu) \setminus (dom(\gamma) \uplus dom(\delta_{V^\mu})) \to vars(c_{\gamma'}) \setminus (dom(\gamma) \uplus dom(\delta_{V^\mu}))$,

then, $c_0\nu''\gamma' = c_0(\nu'' \uplus \gamma') = c_0(\nu' \uplus (\gamma\delta)_{dom(\gamma)} \uplus \gamma') = c_0((\mu\delta_{V^\mu}) \uplus (\gamma\delta)_{dom(\gamma)} \uplus \gamma') = c_0^\mu(\delta_{V^\mu} \uplus (\gamma\delta)_{dom(\gamma)} \uplus \gamma') = c_{\gamma'}(\delta_{V^\mu} \uplus (\gamma\delta)_{dom(\gamma)})$, so $c_0\nu''\tau = c_0\nu''\gamma'(\sigma\theta)_{ran(\gamma')} = c_{\gamma'}(\delta_{V^\mu} \uplus (\gamma\delta)_{dom(\gamma)})(\sigma\theta)_{ran(\gamma')} = c_{\gamma'}(\delta_{V^\mu} \uplus (\gamma\delta)_{dom(\gamma)} \uplus \sigma\theta) = c_{\gamma'}(\delta \uplus \gamma\delta \uplus \sigma\theta) = c_{\gamma'}((\sigma_{V_G}\rho) \uplus (\gamma\sigma_{V_G}\rho) \uplus \sigma\theta) = c_{\gamma'}((\sigma\rho) \uplus (\gamma\sigma\rho) \uplus \sigma\theta) = c_{\gamma'}((\sigma\theta) \uplus (\gamma\sigma\theta) \uplus \sigma\theta) = c_{\gamma'}\gamma\sigma\theta$, all because $c_0\nu''\tau$ is ground, and we can write the derivation rule as

$$\frac{l_1\gamma\sigma\theta \to r_1\gamma\sigma\theta/ST_1^{\nu'}\varrho_{\nu'} \cdots l_m\gamma\sigma\theta \to r_m\gamma\sigma\theta/ST_m^{\nu'}\varrho_{\nu'}}{l\gamma\sigma\theta \to r\gamma\sigma\theta/c[(\gamma\delta)_{dom(\gamma)}]\{\overline{ST}^{\nu'}\varrho_{\nu'}\}} \quad (8)$$

Also, as $\delta' = \sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$, $V_{r_i\gamma\sigma_1, l_i\gamma\sigma_1} \subseteq V_{G'\sigma_1}$, $[r_i\gamma\delta']_E \in ST_i^{\nu'}\varrho_{\nu'}@[l_i\gamma\delta']_E$, for $1 \le i \le n$, where each term is ground, $\sigma = \sigma_1\sigma'$, and $\theta = \rho' \uplus \rho_2$, then $r_i\gamma\delta' = r_i\gamma\sigma_1\sigma'_{V_{G'\sigma_1}}\rho' = r_i\gamma\sigma_1\sigma'\rho' = r_i\gamma\sigma\rho' = r_i\gamma\sigma\theta$ (and $l_i\gamma\delta' = l_i\gamma\sigma\theta$), so $[r_i\gamma\sigma\theta]_E \in ST_i^{\nu'}\varrho_{\nu'}@[l_i\gamma\sigma\theta]_E$, and there are closed proof trees of the form $\frac{F_i}{l_i\gamma\sigma\theta \to r_i\gamma\sigma\theta/ST_i^{\nu'}\varrho_{\nu'}}$, with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$.
As $dom(\gamma') = vars(c_0^\mu) \setminus (dom(\gamma) \uplus V^\mu)$ then $dom(\gamma) \cap dom(\gamma') = \emptyset$, so $\gamma'\gamma = \gamma' \uplus \gamma = \gamma\gamma'$. We already know that $r\gamma\sigma\rho' = x_k\delta$ (3) and $u|_p\delta =_E l\gamma\sigma\theta$ (7) so, as $r\gamma\sigma\rho'$ is ground and $\theta = \rho' \uplus \rho_2$, then also $r\gamma\sigma\theta = r\gamma\sigma\rho' = x_k\delta$, and we can apply the derivation rule (8) to $u|_p\delta$ and $x_k\delta$ and construct the c.p.t. for $[x_k\delta]_E \in (c[\gamma_r]\{\overline{ST}\})^\mu\varrho_\mu\delta\,@\,[u|_p\delta]_E$, i.e., $[x_k\delta]_E \in c[\gamma\delta_{ran(\gamma)}]\{\overline{ST}^{\nu'}\varrho_{\nu'}\}@[u|_p\delta]_E$,

with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$:

$$\frac{\frac{F_1}{l_1\gamma\sigma\theta\to r_1\gamma\sigma\theta/ST_1\delta}\cdots\frac{F_m}{l_m\gamma\sigma\theta\to r_m\gamma\sigma\theta/ST_m\delta}}{u|_p\delta\to x_k\delta/c[(\gamma\delta)_{dom(\gamma)}]\{\overline{ST}^{\nu'}\varrho_{\nu'}\}}.$$

As we have shown before that $E_0 \vdash \psi_1\delta$ (1), that there are closed proof trees for each open goal in $\Delta\delta$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$ (2), and that $[v\delta]_E \in ST^{\nu'}\varrho_{\nu'}@[u[x_k]_p\delta]_E$ (4), then $\delta = \sigma_{vars(G)}\rho$ is a solution of G.

13. Rule $[tp]$ (top):

Again, we prove this case for conditional rules. For unconditional rules the proof is the same, just with the part dealing with the conditions removed from it.

$G = u \to v/(\mathtt{top}(c[\gamma_r]\{ST_1,\ldots,ST_m\})\ ;\ ST)^\mu\varrho_\mu\ (\wedge\ \Delta)\ |\ \psi_1\ |\ V,\mu \rightsquigarrow_{[tp],\sigma_1} (\bigwedge_{i=1}^n(l_i\gamma\to r_i\gamma/ST_i^\mu\varrho_\mu;\mathtt{idle})\wedge r\gamma\to v\ /ST^\mu\varrho_\mu\ (\wedge\ \Delta)\ |\ \psi_2)\sigma_1\ |\ V,(\mu\sigma_1)_V = G'\sigma_1$, where:

- $\gamma = (\gamma_r^\mu\varrho_\mu)_{dom(\gamma_r^\mu)}$ (so $ran(\gamma)\subseteq V_G$), $c\in R$, $c_0\in c_B\subseteq R_B$ has the form $c : l^c\to r^c$ *if* $\bigwedge_{i=1}^n(l_i^c\to r_i^c)\ |\ \phi^c$, $c_{\gamma'} : l\to r$ *if* $\bigwedge_{i=1}^n(l_i\to r_i)\ |\ \phi$ is a fresh version with some renaming γ' of $c_0^\mu\in R_B^\mu$, with $dom(\gamma') = vars(c_0^\mu)\setminus(dom(\gamma_r)\uplus V^\mu)$, so $c_{\gamma'} = c_0^\mu\gamma'$, let $l' = l\gamma$;
- $abstract_{\Sigma_1}(u|_p) = \langle\lambda\bar u.u^\circ;\sigma_u^\circ;\phi_u^\circ\rangle$, $u^\circ = u|_p[\bar x]_{\bar p}$, with $\bar x = x_1,\ldots,x_u$ and $\bar p = p_1,\ldots,p_u$, $\phi_u^\circ = (\bigwedge_{j=1}^u x_j = u|_{p.p_j})$;
- $abstract_{\Sigma_1}(l') = \langle\lambda\bar y.l^\circ;\sigma_1^\circ;\phi^\circ\rangle$, $l^\circ = l'[\bar y]_{\bar q}$, with $\bar y = y_1,\ldots,y_l$ and $\bar q = q_1,\ldots,q_l$, $\phi^\circ = (\bigwedge_{i=1}^l y_i = l'|_{q_i})$;
- $\sigma_1\in CSU_B(u^\circ = l^\circ)$, $\psi_2 = \psi_1\wedge\phi^\circ\wedge\phi_u^\circ\wedge\phi\gamma$, $\psi_2\sigma_1$ is satisfiable;

Then $G'\sigma_1 \rightsquigarrow^+_{\sigma'} nil\ |\ \psi\ |\ V,\nu$, let $\sigma = \sigma_1\sigma'$, where $\nu = (\mu\sigma)_V = (\mu\sigma_1\sigma')_V = (\mu\sigma_1'\sigma')_V$, so $\sigma_{V_G}\ |\ \psi$ is a computed answer for G and $\sigma'_{V_{G'\sigma_1}}\ |\ \psi$ is a computed answer for $G'\sigma_1$.

As $\gamma = \gamma_r^\mu(\varrho_\mu)_{ran(\gamma_r^\mu)}$ then $dom(\gamma_r) = dom(\gamma_r^\mu) = dom(\gamma)$.

Let $\rho : V_{G\sigma}\to\mathcal{T}_\Sigma$ be a substitution such that $\psi\rho$ is satisfiable, let $\delta = \sigma_{V_G}\rho$ and $\varrho_{\nu'} = (\varrho_\mu\delta)_{\setminus V}$, so $\delta : V_G\to\mathcal{T}_\Sigma$, $\rho_1 = \rho_{V_{G'\sigma}}$, so also $\psi\rho_1$ is satisfiable, and let $\nu' = (\nu\rho)_V$, where $dom(\nu') = V$ and $ran(\nu') = \emptyset$. As $dom(\rho) = V_{G\sigma}$ then $dom(\rho_1) = V_{G\sigma}\cap V_{G'\sigma}$. Let $\rho_2 = \rho_{V_{G\sigma}\setminus V_{G'\sigma}}$, so $\rho = \rho_1\uplus\rho_2$, and let $\rho_1' : V_{G'\sigma}\setminus V_{G\sigma}\to\mathcal{T}_\Sigma$, so $dom(\rho_1)\cap dom(\rho_1') = \emptyset$ and $dom(\rho_1)\cup dom(\rho_1') = V_{G'\sigma}$, such that $\psi(\rho_1\uplus\rho_1')$ is satisfiable, and let $\rho' = \rho_1\uplus\rho_1'$, so $\rho' : V_{G'\sigma}\to\mathcal{T}_\Sigma$.

We prove several intermediate results:

- As $\nu = (\mu\sigma)_V$, $V^\mu\subseteq V_G\cap V_{G'}$, and $dom(\rho_1) = V_{G\sigma}\cap V_{G'\sigma}$ then $V^\nu\subseteq dom(\rho_1)$ so, as $dom(\nu') = V$ and $ran(\nu') = \emptyset$, $\nu' = (\nu\rho)_V = (\nu\rho_1)_V = (\nu\rho')_V$. Also, as $dom(\mu)\subseteq V$, $\nu' = (\nu\rho)_V = ((\mu\sigma)_V\rho)_V = (\mu\sigma\rho)_V = \mu(\sigma\rho)_{V^\mu}$.
- As $dom(\rho_1') = V_{G'\sigma}\setminus V_{G\sigma}$ and $dom(\rho_1) = V_{G\sigma}\cap V_{G'\sigma}\subseteq V_{G\sigma}$, then $\rho'_{V_{G\sigma}} = (\rho_1\uplus\rho_1')_{V_{G\sigma}} = (\rho_1)_{V_{G\sigma}} = \rho_1$.
- As $\delta_{V^\mu} = (\sigma_{V_G}\rho)_{V^\mu}$, $\rho : V_{G\sigma}\to\mathcal{T}_\Sigma$, and $V^\mu\subseteq V_G$, then $\delta_{V^\mu} = (\sigma\rho)_{V^\mu}$, $ran(\delta_{V^\mu}) = \emptyset$, and $dom(\delta_{V^\mu}) = V^\mu$ $(= (V\setminus dom(\mu))\cup ran(\mu)$, so $ran(\mu)\subseteq dom(\delta_{V^\mu}))$. Then $\nu' = (\mu\sigma_{V_G}\rho)_V = \mu\delta_{V^\mu}$ and $c_0^{\nu'} = c_0\nu' = c_0\mu\delta_{V^\mu}$.

As $\sigma'_{V_{G'\sigma_1}} \mid \psi$ is a computed answer for $G'\sigma_1$, $\rho' : V_{G'\sigma_1\sigma'} \to \mathcal{T}_\Sigma$, and $\psi\rho'$ is satisfiable then, by I.H., $\sigma'_{V_{G'\sigma_1}}\rho'$ is a solution for $G'\sigma_1$, let $\delta' = \sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$ and $\varrho' = (\varrho_\mu\delta')_{\setminus V}$, meaning that:

(a) $E_0 \vdash \psi_2\delta'$,

(b) there are closed proof trees for each open goal in $\Delta\delta'$, with respect to $\mathcal{D}^{(\nu\rho')_V}_{\mathcal{R},Call_\mathcal{R}}$ $(=\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$, we use ν' instead of $(\nu\rho')_V$ in (c) and (d)),

(c) $[v\delta']_E \in ST^{\nu'}\varrho'@[r\gamma\delta']_E$, and

(d) $[r_i\gamma\delta']_E \in ST^{\nu'}_i\varrho'@[l_i\gamma\delta']_E$, for $1 \leq i \leq n$.

Then:

(a) i. $V_{\psi_2} \subseteq V_{G'}$ implies $V_{\psi_2\sigma_1} \subseteq V_{G'\sigma_1}$ and $V_{\psi_2\sigma} \subseteq V_{G'\sigma}$, so $\psi_2\delta' = \psi_2\sigma_1\sigma'_{V_{G'\sigma_1}}\rho' = \psi_2\sigma_1\sigma'\rho' = \psi_2\sigma\rho'$, hence $E_0 \vdash \psi_2\sigma\rho'$, where $\psi_2\sigma\rho'$ is ground, because $V_{\psi_2\sigma} \subseteq V_{G'\sigma}$ and $\rho' : V_{G'\sigma} \to \mathcal{T}_\Sigma$. As $\psi_2 = \psi_1 \wedge \phi^\circ \wedge \phi\gamma$, then $\psi_1\delta' = \psi_1\sigma\rho'$, $E_0 \vdash \psi_1\sigma\rho'$, $E_0 \vdash \phi^\circ\sigma\rho'$, $E_0 \vdash \phi^\circ_u\sigma\rho'$, and $E_0 \vdash \phi\gamma\sigma\rho'$, all ground formulas.

ii. Also as $\psi_2 = \psi_1 \wedge \phi^\circ \wedge \phi\gamma$, so $V_{\psi_1} \subseteq V_G \cap V_{G'}$ hence $V_{\psi_1\sigma} \subseteq V_{G\sigma} \cap V_{G'\sigma}$, and $dom(\rho_1) = V_{G\sigma} \cap V_{G'\sigma}$ imply $\psi_1\sigma\rho_1 \in \mathcal{T}_\Sigma$. Then, as $\rho' = \rho_1 \uplus \rho'_1$, we have $\psi_1\sigma\rho' = \psi_1\sigma(\rho_1 \uplus \rho'_1) = \psi_1\sigma\rho_1 = \psi_1\sigma(\rho_1 \uplus \rho_2) = \psi_1\sigma\rho = \psi_1\delta$, so $E_0 \vdash \psi_1\delta$ (1).

iii. As $\psi_1\delta' = \psi_1\sigma\rho'$ and $\psi_1\sigma\rho' = \psi_1\delta$ then $\psi_1\delta' = \psi_1\delta$.

(b) As in subcases (a)-ii and (a)-iii, $V_\Delta \subseteq V_G \cap V_{G'}$ implies $\Delta\delta' = \Delta\delta$, and the same closed proof trees are valid for each open goal in $\Delta\delta$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$ (2).

(c) i. Again, $V_v \subseteq V_G \cap V_{G'}$ implies that $v\delta' = v\delta$.

ii. We prove that $ST^{\nu'}\varrho' = ST^{\nu'}\varrho_{\nu'}$.
As $\varrho_{\nu'} = (\varrho_\mu\delta)_{\setminus V}$, $\delta = \sigma_{V_G}\rho$, $\sigma = \sigma_1\sigma'$, $\delta' = \sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$, $\varrho' = (\varrho_\mu\delta')_{\setminus V}$, and $V_{ST^{\nu'}} \cap V = \emptyset$, we have that $ST^{\nu'}\varrho_\mu(\sigma_1\sigma')_{V_G}\rho = ST^{\nu'}\varrho_\mu\sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$. Let $x \in V_{ST^{\nu'}\varrho_\mu}$. As $V_{ST^{\nu'}\varrho_\mu} \subseteq V_{ST^{\mu}\varrho_\mu} \subseteq V_G \cap V_{G'}$, then $x \in V_G \cap V_{G'}$ and $V_{x\sigma_1} \subseteq V_{G\sigma_1} \cap V_{G'\sigma_1} \subseteq V_{G'\sigma_1}$, so $x(\sigma_1\sigma')_{V_G} = x\sigma_1\sigma' = x\sigma_1\sigma'_{V_{G'\sigma_1}}$. Also $x(\sigma_1\sigma')_{V_G} = x\sigma$, hence $V_{x(\sigma_1\sigma')_{V_G}} \subseteq V_{G\sigma}$. Then, as $\rho : V_{G\sigma} \to \mathcal{T}_\Sigma$, $x(\sigma_1\sigma')_{V_G}\rho$ is ground, so $x(\sigma_1\sigma')_{V_G}\rho = x\sigma_1\sigma'_{V_{G'\sigma_1}}\rho = x\sigma_1\sigma'_{V_{G'\sigma_1}}(\rho \cup \rho'_1) = x\sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$.

iii. As in subcase (a)-i, $V_{r\gamma} \subseteq V_{G'}$ implies $r\gamma\delta' = r\gamma\sigma\rho'$.

Joining all the results, we get $[v\delta]_E \in ST^{\nu'}\varrho_{\nu'}@[r\gamma\sigma\rho']_E$, so there is a c.p.t. of the form $\frac{F}{r\gamma\sigma\rho' \to v\delta/ST^{\nu'}\varrho_{\nu'}}$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$ (3).

(d) Using the same proof as in the previous case, $[r_i\gamma\delta']_E \in ST^{\nu'}_i\varrho'@[l_i\gamma\delta']_E$, $V_{l_i\gamma,r_i\gamma} \subseteq V_{G'}$, and $V_{ST^{\nu'}_i\varrho'} \subseteq V_G \cap V_{G'}$ imply $[r_i\gamma\sigma\rho']_E \in ST^{\nu'}_i\varrho_{\nu'}@[l_i\gamma\sigma\rho']_E$, for $1 \leq i \leq n$, where each term and strategy are ground (4).

Now:

(a) $V_u \subseteq V_G$ imply $u\sigma_{V_G} = u\sigma$, hence $u\sigma_{V_G}\theta = u\sigma\theta$, and $u^\circ\sigma'_1 =_B l^\circ\sigma'_1$ imply $u^\circ\sigma\theta =_B l^\circ\sigma\theta$.

(b) As $E_0 \vdash \phi^\circ_u\sigma\theta$, ground formula, then $u^\circ\sigma\theta = u[\bar{x}]_{\bar{p}}\sigma\theta = u\sigma\theta[\bar{x}\sigma\theta]_{\bar{p}} =_{E_0} u\sigma\theta[u|_{\bar{p}}\sigma\theta]_{\bar{p}} = u\sigma\theta$, all ground terms.

(c) As $E_0 \vdash \phi^\circ\sigma\theta$, ground formula, then $l^\circ\sigma\theta = l'[\bar{y}]_{\bar{q}}\sigma\theta = l'\sigma\theta[\bar{y}\sigma\theta]_{\bar{q}} =_{E_0} l'\sigma\theta[l'|_{\bar{q}}\sigma\theta]_{\bar{q}} = l'\sigma\theta$, all ground terms (5).

(d) As $\rho : V_{G\sigma} \to \mathcal{T}_\Sigma$, so $u\sigma_{V_G}\rho$ $(= u\delta)$ is a ground term, then $u\delta = u\sigma_{V_G}\rho = u\sigma_{V_G}\theta = u\sigma\theta =_{E_0} u^\circ\sigma\theta =_B l^\circ\sigma\theta =_{E_0} l'\sigma\theta = l\gamma\sigma\theta$ (6).

We need to prove $[r\gamma\sigma\rho']_E \in c[\gamma_r]\{\overline{ST}\}^{\nu'}\varrho_{\nu'}$ @ $[u\delta]_E$, where $c[\gamma_r]\{\overline{ST}\}^{\nu'}\varrho_{\nu'} = c\nu'[(\gamma_r^{\nu'}\varrho_{\nu'})_{dom(\gamma_r^{\nu'})}]\{\overline{ST}^{\nu'}\varrho_{\nu'}\}$.
As $dom(\gamma') = vars(c_0^\mu) \setminus (dom(\gamma) \uplus V^\mu)$, $c_0^\mu = c_0\mu$, and $dom(\delta_{V^\mu}) = V^\mu$, then $V_{c_0\mu} \subseteq dom(\delta_{V^\mu}) \uplus dom(\gamma) \uplus dom(\gamma')$. Then, as $c_0\nu' = c_0\mu\delta_{V^\mu}$ and δ_{V^μ} is a ground substitution, it follows that $V_{c_0\nu'} = dom(\gamma) \uplus dom(\gamma')$, hence $V_{c_0\nu'\gamma} = ran(\gamma) \cup dom(\gamma')$ and $V_{c_0\nu'(\gamma\delta)_{dom(\gamma)}} = V_{ran(\gamma)\delta_{ran(\gamma)}} \cup V_{dom(\gamma')(\gamma\delta)_{dom(\gamma)}}$.
Then:

- As $(\gamma\delta)_{dom(\gamma)}$ is a ground substitution, if z is a variable in $ran(\gamma)$ then $z\delta_{ran(\gamma)}$ is a ground term, so $V_{ran(\gamma)\delta_{ran(\gamma)}} = \emptyset$.
- As $dom(\gamma) \cap dom(\gamma') = \emptyset$, if z is a variable in $dom(\gamma')$ then $z(\gamma\delta)_{dom(\gamma)} = z$, so $V_{dom(\gamma')(\gamma\delta)_{dom(\gamma)}} = dom(\gamma')$.

In conclusion, $V_{c_0\nu'(\gamma\delta)_{dom(\gamma)}} = dom(\gamma')$.
Let $\nu'' = \nu'(\gamma\delta)_{dom(\gamma)}$ $(= \nu' \uplus (\gamma\delta)_{dom(\gamma)}$ because $dom(\nu') \cap dom(\gamma) = V \cap dom(\gamma) = \emptyset)$. We must find a substitution $\tau : V_{c_0\nu''} \to \mathcal{T}_\Sigma$ such that $E_0 \vdash \phi^c\nu''\tau$. Let $\theta = \rho_2 \uplus \rho_1 \uplus \rho_1'(= \rho_2 \uplus \rho')$, so $dom(\theta) = V_{G\sigma} \cup V_{G'\sigma}$. We choose $\tau = (\gamma'\sigma\theta)_{dom(\gamma')} = \gamma'(\sigma\theta)_{ran(\gamma')}$, so $dom(\tau) = dom(\gamma') = V_{c_0\nu''}$ and $(c_0\nu'')\tau = (c_0\nu'')\gamma'\sigma\theta$.

We prove that τ is a ground substitution by proving that $(c_0\nu'')\gamma'\sigma\theta$ is ground. Let $\delta'' = \delta_{V^\mu}\gamma\delta_{ran(\gamma)}$. As δ_{V^μ} and $\gamma\delta_{ran(\gamma)}$ are ground substitutions, $dom(\delta_{V^\mu}) \cap (dom(\gamma') \cup ran(\gamma')) = \emptyset$, and $V_{c_0\nu'} = dom(\gamma) \uplus dom(\gamma')$, then $(c_0\nu'')\gamma' = c_0\nu'(\gamma\delta_{ran(\gamma)} \uplus \gamma') = c_0^\mu\delta_{V^\mu}(\gamma\delta_{ran(\gamma)} \uplus \gamma') = c_0^\mu\delta_{V^\mu}\gamma'\gamma\delta_{ran(\gamma)} = c_0^\mu\gamma'\delta_{V^\mu}\gamma\delta_{ran(\gamma)} = c_0^\mu\gamma'\delta'' = c_{\gamma'}\delta''$. If $z \in V_{c_{\gamma'}\delta''}$ then, as δ_{V^μ} is ground, either $z \in V_{G'}$ or $z \in V_{l'} \setminus V_{G'}$, because l' is the only term of $c_{\gamma'}\gamma$ that does not appear in G'. We check each case:

- If $z \in V_{G'}$ then $V_{z\sigma} \subseteq V_{G'\sigma}$, so $z\sigma\theta$ is a ground term because $dom(\theta) = V_{G\sigma} \cup V_{G'\sigma}$.
- If $z \in V_{l'} \setminus V_{G'}$, as $z \in V_{l'}$ and, by (5), $l'\sigma\theta$ is ground, then $z\sigma\theta$ is a ground term.

We prove $E_0 \vdash \phi^c\nu''\tau$.

- As $ran(\gamma) \subseteq V_G$ and δ is a ground substitution, then $\gamma\delta_{ran(\gamma)}$ is ground so, as $c_0^{\nu'} = c_0\mu\delta_{V^\mu}$ and $\nu'' = \nu'\gamma\delta_{ran(\gamma)}$, $\phi^c\nu''\tau = \phi^c\mu\delta_{V^\mu}\gamma\delta_{ran(\gamma)}\tau = \phi^c\mu\delta_{V^\mu}(\gamma\delta_{ran(\gamma)} \uplus \tau)$.
- As δ_{V^μ} is a ground substitution, $V_{\phi^c\mu\delta_{V^\mu}} \subseteq V_{c_0^\mu\delta_{V^\mu}} = dom(\gamma) \uplus dom(\gamma')$, $dom(\tau) = dom(\gamma')$, and $dom(\gamma\delta_{ran(\gamma)}) = dom(\gamma)$ then $\phi^c\mu\delta_{V^\mu}(\gamma\delta_{ran(\gamma)} \uplus \tau) = \phi^c\mu(\delta_{V^\mu} \uplus \gamma\delta_{ran(\gamma)} \uplus \tau) = \phi^c\mu((\sigma\rho)_{V^\mu} \uplus \gamma(\sigma\rho)_{ran(\gamma)} \uplus \tau) = \phi^c\mu((\sigma\theta)_{V^\mu} \uplus \gamma(\sigma\theta)_{ran(\gamma)} \uplus \gamma'(\sigma\theta)_{ran(\gamma')})$,
  because as $\phi^c\mu\delta''\tau$ is ground, it remains the same if we substitute the occurrences of ρ, ground substitution, with $\theta = \rho \uplus \rho'$.

- As $(\sigma\theta)_{V^\mu}$ is ground then $\phi^c\mu((\sigma\theta)_{V^\mu} \uplus \gamma(\sigma\theta)_{ran(\gamma)} \uplus \gamma'(\sigma\theta)_{ran(\gamma')}) = \phi^c\mu(\gamma' \uplus \gamma)\sigma\theta$, the last equality because as the formula is ground, no new instantiation will come from an unrestricted substitution.
- As $dom(\gamma) \cap dom(\gamma') = \emptyset$ and $dom(\gamma) \cap ran(\gamma') = \emptyset$, we can apply the substitutions one after the other, so $\phi^c\mu(\gamma' \uplus \gamma)\sigma\theta = \phi^c\mu\gamma'\gamma\sigma\theta = \phi\gamma\sigma\theta$.
- As $V_{\phi\gamma\sigma} \subseteq V_{G'\sigma}$, $\rho' : V_{G'\sigma} \to \mathcal{T}_\Sigma$, and $\theta = \rho_2 \uplus \rho'$ then $\phi\gamma\sigma\theta = \phi\gamma\sigma\rho'$.

Joining all the equalities, we get $\phi^c\nu''\tau = \phi\gamma\sigma\rho'$. Then, as $E_0 \vdash \phi\gamma\sigma\rho'$, also $E_0 \vdash \phi^c\nu''\tau$.

Now, we prove the existence of a needed derivation rule in $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$. As $\varrho_{\nu'} = (\varrho_\mu\delta)_{\setminus V}$ and $\nu' = (\mu\delta)_V$, both ground, $\bigcup_{i=1}^m V^\mu_{ST_i}\varrho_\mu \subseteq V_G$, and $\delta : V_G \to \mathcal{T}_\Sigma$, then $ST^\mu_i\varrho_\mu\delta = ST^{\nu'}_i\varrho_{\nu'}$ and $V_{ST^{\nu'}_i\varrho_{\nu'}} = \emptyset$, for $1 \leq i \leq m$, and $(c[\gamma_r])^\mu\varrho_\mu\delta = c^\mu[(\gamma^\mu_r\varrho_\mu)_{dom(\gamma^\mu_r)}]\delta = c^\mu[\gamma]\delta = c^{\nu'}[(\gamma\delta)_{dom(\gamma)}]$.

Recall that $c_0 \in R$ has the form $c : l^c \to r^c$ *if* $\bigwedge_{i=1}^n(l^c_i \to r^c_i) \mid \phi^c$ and $\nu'' = \nu'\gamma\delta_{ran(\gamma)}$. As $\tau : V_{c_0\nu''} \to \mathcal{T}_\Sigma$, $E_0 \vdash \phi^c\nu''\tau$, $l^c\nu''\tau$ and $r^c\nu''\tau$ are terms in $\mathcal{H}_\Sigma$, ϵ is a position in $pos(l^c\nu''\tau)$ such that $(l^c\nu''\tau)|_\epsilon = l^c\nu''\tau$, and $\overline{ST}^{\nu'}\varrho_{\nu'}$ are ground strategies, then there is a derivation rule

$$\frac{l^c_1\nu''\tau \to r^c_1\nu''\tau/ST^{\nu'}_1\varrho_{\nu'} \cdots l^c_m\nu''\tau \to r^c_m\nu''\tau/ST^{\nu'}_m\varrho_{\nu'}}{l^c\nu''\tau \to r^c\nu''\tau/c[(\gamma\delta)_{dom(\gamma)}]\{\overline{ST}^{\nu'}\varrho_{\nu'}\}}$$

in $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$.

Now, as:

- $\nu'' = \nu' \uplus (\gamma\delta)_{dom(\gamma)}$ is ground, $\nu' = \mu\delta_{V^\mu}$, $\delta = \sigma_{V_G}\rho$, $\theta = \rho \uplus \rho'_1$, and $dom(\delta_{V^\mu}) = V^\mu$,
- $\tau = \gamma'(\sigma\theta)_{ran(\gamma')}$ and $\delta_{V^\mu} \uplus (\gamma\delta)_{dom(\gamma)}$ are ground substitutions,
- $c_0 : l^c \to r^c$ *if* $\bigwedge_{i=1}^n(l^c_i \to r^c_i) \mid \phi^c$ and $c_0\nu''\tau$ is ground,
- $c_{\gamma'} : l \to r$ *if* $\bigwedge_{i=1}^n(l_i \to r_i) \mid \phi$, and
- $c_{\gamma'}$ is a fresh version of c^μ_0 except for $dom(\gamma) \uplus dom(\delta_{V^\mu})$, with renaming $\gamma' : vars(c^\mu_0) \setminus (dom(\gamma) \uplus dom(\delta_{V^\mu})) \to vars(c_{\gamma'}) \setminus (dom(\gamma) \uplus dom(\delta_{V^\mu}))$,

then, $c_0\nu''\gamma' = c_0(\nu'' \uplus \gamma') = c_0(\nu' \uplus (\gamma\delta)_{dom(\gamma)} \uplus \gamma') = c_0((\mu\delta_{V^\mu}) \uplus (\gamma\delta)_{dom(\gamma)} \uplus \gamma') = c^\mu_0(\delta_{V^\mu} \uplus (\gamma\delta)_{dom(\gamma)} \uplus \gamma') = c_{\gamma'}(\delta_{V^\mu} \uplus (\gamma\delta)_{dom(\gamma)})$, so $c_0\nu''\tau = c_0\nu''\gamma'(\sigma\theta)_{ran(\gamma')} = c_{\gamma'}(\delta_{V^\mu} \uplus (\gamma\delta)_{dom(\gamma)})(\sigma\theta)_{ran(\gamma')} = c_{\gamma'}(\delta_{V^\mu} \uplus (\gamma\delta)_{dom(\gamma)} \uplus \sigma\theta) = c_{\gamma'}(\delta \uplus (\gamma\delta) \uplus \sigma\theta) = c_{\gamma'}((\sigma_{V_G}\rho) \uplus (\gamma\sigma_{V_G}\rho) \uplus \sigma\theta) = c_{\gamma'}((\sigma\rho) \uplus (\gamma\sigma\rho) \uplus \sigma\theta) = c_{\gamma'}((\sigma\theta) \uplus (\gamma\sigma\theta) \uplus \sigma\theta) = c_{\gamma'}\gamma\sigma\theta$, all because $c_0\nu''\tau$ is ground, and we can write the derivation rule as

$$\frac{l_1\gamma\sigma\theta \to r_1\gamma\sigma\theta/ST^{\nu'}_1\varrho_{\nu'} \cdots l_m\gamma\sigma\theta \to r_m\gamma\sigma\theta/ST^{\nu'}_m\varrho_{\nu'}}{l\gamma\sigma\theta \to r\gamma\sigma\theta/c[(\gamma\delta)_{dom(\gamma)}]\{\overline{ST}^{\nu'}\varrho_{\nu'}\}} \quad (7)$$

Also, as $\delta' = \sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$, $V_{r_i\gamma\sigma_1, l_i\gamma\sigma_1} \subseteq V_{G'\sigma_1}$, $[r_i\gamma\delta']_E \in ST^{\nu'}_i\varrho_{\nu'}@[l_i\gamma\delta']_E$, for $1 \leq i \leq n$, where each term is ground (4), $\sigma = \sigma_1\sigma'$, and $\theta = \rho' \uplus \rho_2$, then $r_i\gamma\delta' = r_i\gamma\sigma_1\sigma'_{V_{G'\sigma_1}}\rho' = r_i\gamma\sigma_1\sigma'\rho' = r_i\gamma\sigma\rho' = r_i\gamma\sigma\theta$ (and $l_i\gamma\delta' = l_i\gamma\sigma\theta$), so $[r_i\gamma\sigma\theta]_E \in ST^{\nu'}_i\varrho_{\nu'}@[l_i\gamma\sigma\theta]_E$, and there are closed proof trees of the form $\frac{F_i}{l_i\gamma\sigma\theta \to r_i\gamma\sigma\theta/ST^{\nu'}_i\varrho_{\nu'}}$, with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$.

There is also a derivation rule $\frac{u\delta \to r\gamma\sigma\theta/c[(\gamma\delta)_{dom(\gamma)}]\{\overline{ST}^{\nu'}\varrho_{\nu'}\} \quad r\gamma\sigma\rho' \to v\delta/ST^{\nu'}\varrho_{\nu'}}{u\delta \to v\delta/(c[(\gamma\delta)_{dom(\gamma)}]\{\overline{ST}^{\nu'}\varrho_{\nu'}\};ST^{\nu'}\varrho_{\nu'}}$ in $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$, as seen in Section 6.4.

We already know that there is a c.p.t. of the form $\frac{F}{r\gamma\sigma\rho' \to v\delta/ST^{\nu'}\varrho_{\nu'}}$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$ (3). As $r\gamma\sigma\rho'$ is ground and $\theta = \rho' \uplus \rho_2$ then $r\gamma\sigma\rho' = r\gamma\sigma\theta$, hence $\frac{F}{r\gamma\sigma\theta \to v\delta/ST^{\nu'}\varrho_{\nu'}}$ is a c.p.t. with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$.
We also know that $u\delta =_E l\gamma\sigma\theta$ (6), so we can apply the derivation rule (7) to $u\delta$ and $r\gamma\sigma\theta$, and construct the c.p.t. for $[v\delta]_E \in (c[\gamma_r]\{\overline{ST}\})^{\mu}\varrho_{\mu}\delta \mathbin{@} [u\delta]_E$, i.e., $[v\delta]_E \in c[\gamma\delta_{ran(\gamma)}]\{\overline{ST}^{\nu'}\varrho_{\nu'}\} \mathbin{@} [u\delta]_E$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$:

$$\frac{\dfrac{\dfrac{F_1}{l_1\gamma\sigma\theta \to r_1\gamma\sigma\theta/ST_1^{\nu'}\varrho_{\nu'}} \cdots \dfrac{F_m}{l_m\gamma\sigma\theta \to r_m\gamma\sigma\theta/ST_m^{\nu'}\varrho_{\nu'}}}{u\delta \to r\gamma\sigma\theta/c[(\gamma\delta)_{dom(\gamma)}]\{\overline{ST}^{\nu'}\varrho_{\nu'}\}} \quad \dfrac{F}{r\gamma\sigma\theta \to v\delta/ST^{\nu'}\varrho_{\nu'}}}{u\delta \to v\delta/c[(\gamma\delta)_{dom(\gamma)}]\{\overline{ST}^{\nu'}\varrho_{\nu'}\}\,;ST^{\nu'}\varrho_{\nu'}}.$$

As we have shown before that $E_0 \vdash \psi_1\delta$ (1) and that there are closed proof trees for each open goal in $\Delta\delta$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$ (2), then $\delta = \sigma_{vars(G)}\rho$ is a solution of G.

14. Rule [$c1$] (call strategy):
There are two versions of the rule where in $Call_{\mathcal{R}}$ we have either (a) $\mathtt{sd}\ CS := ST_1$ or (b) $\mathtt{sd}\ CS(\bar{x}) := ST_1$.

(a) $G = u_1 \to v_1/CS^{\mu}\varrho_{\mu}\,;ST^{\mu}\varrho_{\mu}\ (\wedge\ \Delta) \mid \psi_1 \mid V, \mu$, where $CS^{\mu}\varrho_{\mu} = CS^{\mu}$, $G \rightsquigarrow_{[c1]} u_1 \to v_1/ST_2\,;ST^{\mu}\varrho_{\mu}\ (\wedge\ \Delta) \mid \psi_1 \mid V, \mu = G'$ and $G' \rightsquigarrow^{+}_{\sigma} nil \mid \psi \mid V, \nu$, where $\mathtt{sd}\ CS := ST_1 \in Call^{\mu}_{\mathcal{R}}$, $\nu = (\mu\sigma)_V$, and ST_2 is a fresh version of ST_1, with some renaming γ', where $dom(\gamma') = V_{ST_1} \setminus V^{\mu}$, so $ST_2 = ST_1\gamma'$, hence $\sigma_{V_G}|\psi$ is a computed answer for G and $\sigma_{V_{G'}}|\psi$ is a computed answer for G'. As $V_{CS} = \emptyset$ then $V_G \subseteq V_{G'}$, so $ran(\sigma_{V_G}) \subseteq ran(\sigma_{V_{G'}})$. Then:
   i. as $\mathtt{sd}\ CS := ST_1 \in Call^{\mu}_{\mathcal{R}}$ then there is $\mathtt{sd}\ CS := ST_0 \in Call_{\mathcal{R}}$ such that $ST_0^{\mu} = ST_1$, hence $ST_2 = ST_1\gamma' = ST_0^{\mu}\gamma' = (ST_0\gamma')^{\mu}$, since $dom(\gamma') \cap V^{\mu} = \emptyset$ and $ran(\gamma') \cap dom(\mu) = \emptyset$, and
   ii. as $dom(\varrho_{\mu}) \cap V^{\mu} = \emptyset$, invariant for admissible goals, and ST_2 has only new variables except for V^{μ}, then $ST_2 = ST_2\varrho_{\mu} = (ST_0\gamma')^{\mu}\varrho_{\mu}$.

Let $\rho : V_{G\sigma} \to \mathcal{T}_{\Sigma}$ such that $\psi\rho$ is satisfiable, let $\delta = \sigma_{V_G}\rho$, so $\delta : V_G \to \mathcal{T}_{\Sigma}$, and let $\nu' = (\nu\rho)_V$, where $dom(\nu') = V$ and $ran(\nu') = \emptyset$. Let $\rho_1 : V_{G'\sigma} \setminus V_{G\sigma} \to \mathcal{T}_{\Sigma}$, so $dom(\rho) \cap dom(\rho_1) = \emptyset$ and $dom(\rho) \cup dom(\rho_1) = V_{G'\sigma}$, such that $\psi(\rho \uplus \rho_1)$ is satisfiable. Let $\rho' = \rho \uplus \rho_1$, so $\rho' : V_{G'\sigma} \to \mathcal{T}_{\Sigma}$, and let $\delta' = \sigma_{V_{G'}}\rho'$, so $\delta' : V_{G'} \to \mathcal{T}_{\Sigma}$. As $dom(\nu') = V$ and $ran(\nu') = \emptyset$ then $(\nu\rho')_V = (\nu\rho)_V = \nu'$. Then $G\delta' = G\sigma_{V_{G'}}\rho' = G\sigma_{V_{G'}}(\rho \uplus \rho_1) = G(\sigma_{V_G} \uplus \sigma_{V_{G'}\setminus V_G})(\rho \uplus \rho_1) = G(\sigma_{V_G}\rho \uplus \sigma_{V_{G'}\setminus V_G}\rho_1) = G\sigma_{V_G}\rho = G\delta$.
By I.H, $E_0 \vdash \psi_1\delta'$ and there is a c.p.t. $\frac{\frac{F_1}{u_1\delta' \to t/(ST_0\gamma')^{\nu'}\varrho_{\nu'}} \quad \frac{F_2}{t \to v_1\delta'/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta' \to v_1\delta'/(ST_0\gamma'\,;ST))^{\nu'}\varrho_{\nu'}}$, for some term $t \in \mathcal{H}_{\Sigma}$ and appropriate F_1 and F_2, with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$. By Lemma 13, there is also a c.p.t. of the form $\frac{F_3}{u_1\delta' \to t/(ST_0\gamma')^{\nu'}}$.
As $CS^{\nu'}\varrho_{\nu'} = CS$, $(ST_0\gamma')^{\nu'} = ST_0^{\nu'}\gamma'$, since $(dom(\gamma') \cup ran(\gamma')) \cap V = \emptyset$, and there are derivation rules $\frac{u_1\delta' \to t/CS \quad t \to v_1\delta'/ST^{\nu'}\varrho_{\nu'}}{u_1\delta' \to v_1\delta'/(CS\,;ST)^{\nu'}\varrho_{\nu'}}$ and $\frac{u_1\delta' \to t/ST_0^{\nu'}\gamma'}{u_1\delta' \to t/CS}$, i.e., $\frac{u_1\delta' \to t/(ST_0\gamma')^{\nu'}}{u_1\delta' \to t/CS}$ in $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$, then $\frac{\frac{\frac{F_3}{u_1\delta' \to t/(ST_0\gamma')^{\nu'}}}{u_1\delta' \to t/CS} \quad \frac{F_2}{t \to v_1\delta'/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta' \to v_1\delta'/(CS\,;ST)^{\nu'}\varrho_{\nu'}}$ is a c.p.t., so $v_1\delta' \in (CS\,;ST)^{\nu'}\varrho_{\nu'} @ u_1\delta'$.

As $G\delta' = G\delta$, this is the same as $v_1\delta \in (CS\,;ST)^{\nu'}\varrho_{\nu'}@u_1\delta$ and $E_0 \vdash \psi_1\delta$, so $\sigma_{vars(G)}\rho$ is a solution of G.

(b) $G = u_1 \to v_1/CS(\bar{t})^{\mu}\varrho_{\mu}\,;ST^{\mu}\varrho_{\mu}\ (\wedge\,\Delta) \mid \psi_1 \mid V, \mu$, where $CS(\bar{t})^{\mu}\varrho_{\mu} = CS^{\mu}(\bar{t}\mu\varrho_{\mu})$, $G \rightsquigarrow_{[cI]} u_1 \to v_1/ST_2\gamma\,;ST : (\wedge\,\Delta) \mid \psi_1 \mid V, \mu = G'$, and $G' \rightsquigarrow_{\sigma}^{+} nil \mid \psi \mid V, \nu$, where $\nu = (\mu\sigma)_V$, we let $\varrho_{\nu} = (\varrho_{\mu}\sigma)_{\setminus V}$, $\mathtt{sd}\ CS(\bar{x}) := ST_1 \in Call^{\mu}_{\mathcal{R}}$, $\gamma = \{\bar{x} \mapsto \bar{t}\mu\varrho_{\mu}\}$, ST_2 is a fresh version of ST_1, with some renaming γ', where $dom(\gamma') = V_{ST_1} \setminus (\hat{x} \cup V^{\mu})$, so $ST_2 = ST_1\gamma'$, hence $\sigma_{V_G}|\psi$ is a computed answer for G and $\sigma_{V_{G'}}|\psi$ is a computed answer for G'. As $V_{CS^{\mu}(\bar{t}\mu\varrho_{\mu})} = ran(\gamma)$ and $\hat{x} \subseteq V_{ST_2}$ then $V_G \subseteq V_{G'}$, so $ran(\sigma_{V_G}) \subseteq ran(\sigma_{V_{G'}})$. $ST_2 = ST_2[\bar{x}']_{\bar{p}}$, for appropriate $\bar{x}'$ and $\bar{p}$, where $\hat{x}' = \hat{x}$ and $V_{ST_2[]_{\bar{p}}} \cap \bar{x} = \emptyset$, so $ST_2\gamma = ST_2[\bar{x}']_{\bar{p}}\gamma = ST_2[\bar{x}'\gamma]_{\bar{p}}$.
Let $\gamma_0 = \{\bar{x} \mapsto \bar{t}\}$, so $\bar{x}'\gamma = \bar{x}'\gamma_0\mu\varrho_{\mu}$. Then:

i. as $dom(\varrho_{\mu}) \cap V^{\mu} = \emptyset$, invariant for admissible goals, and $ST_2[]_{\bar{p}}$ has only new variables except for V^{μ}, then $ST_2[]_{\bar{p}} = ST_2[]_{\bar{p}}\varrho_{\mu} = ST_2\varrho_{\mu}[]_{\bar{p}} = ST_1\gamma'\varrho_{\mu}[]_{\bar{p}}$, and

ii. as $\mathtt{sd}\ CS(\bar{x}) := ST_1 \in Call^{\mu}_{\mathcal{R}}$ then there is a definition $\mathtt{sd}\ CS(\bar{x}) := ST_0$ in $Call_{\mathcal{R}}$ such that $ST_0^{\mu} = ST_1$. Then, we get $ST_2\gamma = ST_2[\bar{x}'\gamma]_{\bar{p}} = ST_1\gamma'\varrho_{\mu}[\bar{x}'\gamma]_{\bar{p}} = ST_1\gamma'\varrho_{\mu}[\bar{x}'\gamma_0\mu\varrho_{\mu}]_{\bar{p}} = (ST_0^{\mu}\gamma'[\bar{x}'\gamma_0\mu]_{\bar{p}})\varrho_{\mu} = ((ST_0\gamma')^{\mu}[\bar{x}'\gamma_0\mu]_{\bar{p}})\varrho_{\mu} = (ST_0\gamma'[\bar{x}'\gamma_0]_{\bar{p}})^{\mu}\varrho_{\mu}$, since $dom(\gamma') \cap V^{\mu} = \emptyset$ and $ran(\gamma') \cap dom(\mu) = \emptyset$.

Let $\rho : V_{G\sigma} \to \mathcal{T}_{\Sigma}$ such that $\psi\rho$ is satisfiable, let $\delta = \sigma_{V_G}\rho$, so $\delta : V_G \to \mathcal{T}_{\Sigma}$, let $\nu' = (\nu\rho)_V$, where $dom(\nu') = V$ and $ran(\nu') = \emptyset$, and let $\varrho_{\nu'} = (\varrho_{\nu}\rho)_{\setminus V}$. Let $\rho_1 : V_{G'\sigma}\setminus V_{G\sigma} \to \mathcal{T}_{\Sigma}$, so $dom(\rho) \cap dom(\rho_1) = \emptyset$ and $dom(\rho) \cup dom(\rho_1) = V_{G'\sigma}$, such that $\psi(\rho \uplus \rho_1)$ is satisfiable. Let $\rho' = \rho \uplus \rho_1$, so $\rho' : V_{G'\sigma} \to \mathcal{T}_{\Sigma}$, and let $\delta' = \sigma_{V_{G'}}\rho'$, so $\delta' : V_{G'} \to \mathcal{T}_{\Sigma}$. As $dom(\nu') = V$ and $ran(\nu') = \emptyset$ then $(\nu\rho')_V = (\nu\rho)_V = \nu'$. Also, as $V_G \subseteq V_{G'}$ and $dom(\rho_1) = V_{G'\sigma} \setminus V_{G\sigma}$, then $G\delta' = G\sigma_{V_{G'}}\rho' = G\sigma_{V_G}\rho' = G\sigma\rho' = G\sigma(\rho \uplus \rho_1) = G\sigma\rho = G\sigma_{V_G}\rho = G\delta$.
By I.H, $E_0 \vdash \psi_1\delta'$, so $\dfrac{\dfrac{F_1}{u_1\delta' \to w/(ST_0\gamma'[\bar{x}'\gamma_0]_{\bar{p}})^{\nu'}\varrho_{\nu'}} \quad \dfrac{F_2}{w \to v_1\delta'/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta' \to v_1\delta'/(ST_0\gamma'[\bar{x}'\gamma_0]_{\bar{p}}\,;ST))^{\nu'}\varrho_{\nu'}}$ is a c.p.t., for some term $w \in \mathcal{H}_{\Sigma}$, with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$. As ϱ_{ν} is idempotent and ρ is ground then $\varrho_{\nu'}$ is also idempotent. Then, as ν' is ground, $dom(\nu') = V$, and $dom(\varrho_{\nu'}) \cap V = \emptyset$, we can write $\dfrac{F_1}{u_1\delta' \to w/(ST_0\gamma'[\bar{x}'\gamma_0]_{\bar{p}})^{\nu'}\varrho_{\nu'}}$ as $\dfrac{F_1}{u_1\delta' \to w/(ST_0\gamma'[\bar{x}'\gamma_0\varrho_{\nu'}]_{\bar{p}})^{\nu'}\varrho_{\nu'}}$. Let α be a renaming such that $dom(\alpha) = V_{\varrho_{\nu'}}$ and $ran(\alpha)$ is away from all known variables. By Lemma 12 there is a c.p.t. of the form $\dfrac{F_3}{u_1\delta' \to w/(ST_0\gamma'[\bar{x}'\gamma_0\varrho_{\nu'}]_{\bar{p}})^{\nu'}(\varrho_{\nu'}\alpha)}$. Now, we can apply Lemma 13, so there is also a closed proof tree of the form $\dfrac{F_4}{u_1\delta' \to w/(ST_0\gamma'[\bar{x}'\gamma_0\varrho_{\nu'}]_{\bar{p}})^{\nu'}}$. This c.p.t. shows that partial generalization of $dom(\varrho_{\nu'})$ is also valid.
As $(ST_0\gamma'[\bar{x}'\gamma_0\varrho_{\nu'}]_{\bar{p}})^{\nu'} = ST_0^{\nu'}\gamma'[\bar{x}'\gamma_0\varrho_{\nu'}\nu']_{\bar{p}} = ST_0^{\nu'}\gamma'[\bar{x}'\gamma_0\nu'\varrho_{\nu'}]_{\bar{p}} = ST_0^{\nu'}(\gamma' \cup \gamma'')$, where $\gamma'' = \{\bar{x} \mapsto \bar{t}\nu'\varrho_{\nu'}\}$, since $(dom(\gamma') \cup ran(\gamma')) \cap V = \emptyset$, ν' is ground, $dom(\nu') = V$, and $dom(\varrho_{\nu'}) \cap V = \emptyset$, and also $CS(\bar{t})^{\nu'}\varrho_{\nu'} = CS(\bar{t}\nu'\varrho_{\nu'})$, then $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$ has derivation rules $\dfrac{u_1\delta' \to w/ST_0^{\nu'}(\gamma' \cup \gamma'')}{u_1\delta' \to w/CS(\bar{t}\nu'\varrho_{\nu'})}$ or, equiv-

alently, $\frac{u_1\delta' \to w/(ST_0\gamma'[\bar{x}'\gamma_0\varrho_{\nu'}]_{\bar{p}})^{\nu'}}{u_1\delta' \to w/CS(\bar{t}\nu'\varrho_{\nu'})}$, and $\frac{u_1\delta' \to w/CS(\bar{t}\nu'\varrho_{\nu'}) \quad w \to v_1\delta'/ST^{\nu'}\varrho_{\nu'}}{u_1\delta' \to v_1\delta'/(CS(\bar{t})\,;ST)^{\nu'}\varrho_{\nu'}}$. Then

$$\frac{\dfrac{\dfrac{F_4}{u_1\delta' \to w/(ST_0\gamma'[\bar{x}'\gamma_0\varrho_{\nu'}]_{\bar{p}})^{\nu'}}}{u_1\delta' \to w/CS(\bar{t}\nu'\varrho_{\nu'})} \quad \dfrac{F_2}{w \to v_1\delta'/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta' \to v_1\delta'/(CS(\bar{t})\,;ST)^{\nu'}\varrho_{\nu'}}$$

is a c.p.t., so $[v_1\delta']_E \in (CS(\bar{t})\,;ST)^{\nu'}\varrho_{\nu'}@[u_1\delta']_E$. As $G\delta' = G\delta$, this is the same as $[v_1\delta]_E \in (CS(\bar{t})\,;ST)^{\nu'}\varrho_{\nu'}@[u_1\delta]_E$ and $E_0 \vdash \psi_1\delta'$ is the same as $E_0 \vdash \psi_1\delta$, so $\sigma_{vars(G)}\rho$ is a solution of G.

15. Rule [$c2$] (call strategy):
$G = u_1 \to v_1/CS(\bar{t})^{\mu}\varrho_{\mu}; ST^{\mu}\varrho_{\mu}(\wedge\Delta) \mid \psi_1 \mid V, \mu$, where $CS(\bar{t})^{\mu}\varrho_{\mu} = CS^{\mu}(\bar{t}\mu\varrho_{\mu})$, $G \rightsquigarrow_{[c2]} \bigwedge_{j=1}^{m}(l_j\gamma'\gamma \to r_j\gamma'\gamma/\texttt{idle}) \wedge u_1 \to v_1/ST_2\gamma\,; ST^{\mu}\varrho_{\mu}(\wedge\Delta) \mid \psi_2 \mid V, \mu = G'$, $G' \rightsquigarrow^{*}_{\sigma'} u_1\sigma' \to v_1\sigma'/(ST_2\gamma\,; ST^{\mu}\varrho_{\mu})\sigma'\ (\wedge\,\Delta\sigma') \mid \psi_3 \mid V, (\mu\sigma')_V = G''$, and $G'' \rightsquigarrow^{+}_{\sigma''} nil \mid \psi \mid V, \nu$, let $\sigma = \sigma'\sigma''$, where $\nu = (\mu\sigma)_V$, $\texttt{csd}\ CS(\bar{x}) := ST_1$ *if* $C \in Call^{\mu}_{\mathcal{R}}$, $C = \bigwedge_{j=1}^{m} l_j \to r_j \wedge \phi$, $\gamma = \{\bar{x} \mapsto \bar{t}\mu\varrho_{\mu}\}$, let $\overline{C} = \bar{l}, \bar{r}, \phi$, ST_2 *if* $C\gamma'$ is a fresh version of ST_1 *if* C, with some renaming γ', $dom(\gamma') = V_{ST_1,C} \setminus (\hat{x} \cup V^{\mu})$, so $ST_2 = ST_1\gamma'$, $\psi_2 = \psi_1 \wedge \phi\gamma'\gamma$, and $\psi_3 = \psi_2\sigma' \wedge \psi_4 = \psi_1\sigma' \wedge \phi\gamma'\gamma\sigma' \wedge \psi_4$, for appropriate ψ_4, hence $V_{C\gamma'\gamma\sigma'} \subseteq V_{G'\sigma'} \subseteq V_{G''}$, let $\psi_5 = \phi\gamma'\gamma\sigma' \wedge \psi_4$, $\sigma_{V_G}|\psi$ is a computed answer for G, and $\sigma_{V_{G'}}|\psi$ is a computed answer for G', where $\psi = \psi_3\sigma'' \wedge \psi_6$, for appropriate ψ_6. We let $\varrho_{\nu} = (\varrho_{\mu}\sigma)_{V_{G'}\setminus V}$.
By invariant 11, G has the form $G_0^{\mu}\varrho_{\mu}$, so $(u_1, v_1, \psi_1) = (u_0, v_0, \psi_0)\mu\varrho_{\mu}$, for appropriate u_0, v_0, and ψ_0, and there exists Δ_0 such that $\Delta = \Delta_0^{\mu}\varrho_{\mu}$. As $V_{CS^{\mu}(\bar{t}\mu\varrho_{\mu})} = ran(\gamma)$ and $\hat{x} \subseteq V_{\overline{C}\gamma',ST_2}$ then $V_G \subseteq V_{G'}$, so $\sigma_{V_G} = (\sigma_{V_{G'}})_{V_G}$ and G' has the form $G_1^{\mu}\varrho_{\mu}$, where $\varrho_{\mu} = (\varrho_{\mu})_{V_{G_1}\setminus V}$, by invariant 11. Also by invariant 11, G'' has the form $G_2^{\mu'}\varrho_{\mu'}$, where $\mu' = (\mu\sigma')_V$ and $\varrho_{\mu'} = (\varrho_{\mu}\sigma')_{V_{G_2}\setminus V}$. Then, $\psi_3 = \psi_1\sigma' \wedge \psi_5 = (\psi_0\mu\varrho_{\mu})\sigma' \wedge \psi_5 = \psi_0\mu'\varrho_{\mu'} \wedge \psi_5$ has the form $(\psi_0 \wedge \phi_0)\mu'\varrho_{\mu'}$, for appropriate ϕ_0, and $(u_1, v_1)\sigma' = (u_0, v_0)\mu\varrho_{\mu}\sigma' = (u_0, v_0)\mu'\varrho_{\mu'}$, so $V_{G_0} \subseteq V_{G_2}$, hence $V_G = V_{G_0^{\mu}\varrho_{\mu}} \subseteq V_{G_2^{\mu}\varrho_{\mu}}$, $V_{G\sigma'} \subseteq V_{G_2^{\mu}\varrho_{\mu}\sigma'} = V_{G_2^{\mu'}\varrho_{\mu'}} = V_{G''}$, and $V_{G\sigma} \subseteq V_{G_2^{\mu}\varrho_{\mu}\sigma} = V_{G_2^{\mu}\varrho_{\mu}\sigma'\sigma''} = V_{G_2^{\mu'}\varrho_{\mu'}\sigma''} = V_{G''\sigma''}$.
$(ST_2, \overline{C}) = (ST_2[\bar{x}']_{\bar{p}}, \overline{C}[\bar{x}'']_{\bar{q}})$, for appropriate $(\bar{x}'', \bar{x}', \bar{q}, \bar{p})$, where $V_{(ST_2[]_{\bar{p}}, \overline{C}[]_{\bar{q}}} \cap \hat{x} = \emptyset$ and $\hat{x}' \cup \hat{x}'' = \hat{x}$, and $\overline{C}\gamma' = \overline{C}[\bar{x}'']_{\bar{q}}\gamma' = \overline{C}\gamma'[\bar{x}''\gamma']_{\bar{q}} = \overline{C}\gamma'[\bar{x}'']_{\bar{q}}$, since $dom(\gamma') \cap \hat{x} = \emptyset$, so $(ST_2, \overline{C}\gamma')\gamma = (ST_2[\bar{x}']_{\bar{p}}, \overline{C}\gamma'[\bar{x}'']_{\bar{q}})\gamma = (ST_2[\bar{x}'\gamma]_{\bar{p}}, \overline{C}\gamma'[\bar{x}''\gamma]_{\bar{q}})$.
Let $\gamma_0 = \{\bar{x} \mapsto \bar{t}\}$, so $\bar{x}'\gamma = \bar{x}'\gamma_0\mu\varrho_{\mu}$ and $\bar{x}''\gamma = \bar{x}''\gamma_0\mu\varrho_{\mu}$. Then:

(a) as $dom(\varrho_{\mu}) \cap V^{\mu} = \emptyset$, by Proposition 20.6, and $(ST_2[]_{\bar{p}}, \overline{C}\gamma'[]_{\bar{q}})$ has only new variables except for V^{μ}, then $(ST_2[]_{\bar{p}}, \overline{C}\gamma'[]_{\bar{q}}) = (ST_2[]_{\bar{p}}, \overline{C}\gamma'[]_{\bar{q}})\varrho_{\mu} = (ST_2\varrho_{\mu}[]_{\bar{p}}, \overline{C}\gamma'\varrho_{\mu}[]_{\bar{q}}) = (ST_1\gamma'\varrho_{\mu}[]_{\bar{p}}, \overline{C}\gamma'\varrho_{\mu}[]_{\bar{q}})$, and

(b) as $\texttt{sd}\ CS(\bar{x}) := ST_1$ *if* $C \in Call^{\mu}_{\mathcal{R}}$ then there is a call strategy definition $\texttt{sd}\ CS(\bar{x}) := ST_0$ *if* $C' \in Call_{\mathcal{R}}$, $C' = \bigwedge_{j=1}^{m} l'_j \to r'_j \wedge \phi'$, let $\overline{C}' = \bar{l}', \bar{r}', \phi'$, such that $(ST_0, \overline{C}')^{\mu} = (ST_1, \overline{C})$, so $\overline{C}'\mu = \overline{C} = \overline{C}[\bar{x}'']_{\bar{q}}$, hence $\overline{C}'\mu = \overline{C}'\mu[\bar{x}'']_{\bar{q}}$ and $\overline{C}' = \overline{C}'[\bar{x}'']_{\bar{q}}$, since $dom(\mu) \cap \hat{x} = \emptyset$. Then, since $dom(\gamma') \cap V^{\mu} = \emptyset$ and $ran(\gamma') \cap dom(\mu) = \emptyset$:
   – $ST_2\gamma = ST_2[\bar{x}'\gamma]_{\bar{p}} = ST_1\gamma'\varrho_{\mu}[\bar{x}'\gamma]_{\bar{p}} = ST_0^{\mu}\gamma'[\bar{x}'\gamma_0\mu]_{\bar{p}}\varrho_{\mu} = ST_0^{\mu}\gamma'[\bar{x}'\gamma_0\mu]_{\bar{p}}\varrho_{\mu} = (ST_0\gamma'[\bar{x}'\gamma_0]_{\bar{p}})^{\mu}\varrho_{\mu}$, let $ST'_0 = ST_0\gamma'[\bar{x}'\gamma_0]_{\bar{p}}$, and

– $\overline{C}\gamma' = \overline{C}\gamma'[\bar{x}''\gamma]_{\bar{q}} = \overline{C}\gamma'\varrho_\mu[\bar{x}''\gamma]_{\bar{q}} = \overline{C}'^{\mu}\gamma'\varrho_\mu[\bar{x}''\gamma]_{\bar{q}} = \overline{C}'^{\mu}\gamma'\varrho_\mu[\bar{x}''\gamma_0\mu\varrho_\mu]_{\bar{q}} = \overline{C}'^{\mu}\gamma'[\bar{x}''\gamma_0\mu]_{\bar{q}}\varrho_\mu = (\overline{C}'\gamma'[\bar{x}''\gamma_0]_{\bar{q}})^{\mu}\varrho_\mu = (\overline{C}'\gamma'[\bar{x}'']_{\bar{q}}\gamma_0)^{\mu}\varrho_\mu = (\overline{C}'[\bar{x}'']_{\bar{q}}\gamma'\gamma_0)^{\mu}\varrho_\mu = (\overline{C}'\gamma'\gamma_0)^{\mu}\varrho_\mu$.

As $G'' = G_2^{\mu'}\varrho_{\mu'}$, then $G_2 = u_0 \to v_0/ST'_0\,;ST\,(\wedge\,\Delta_0) \mid \psi_0 \wedge \phi_0 \mid V, none$, hence $(ST'_0\,;ST)^{\mu'}\varrho_{\mu'}$ is a strategy in G''.

Let $\rho : V_{G\sigma} \to \mathcal{T}_\Sigma$ such that $\psi\rho$ is satisfiable, let $\delta = \sigma_{V_G}\rho$, so $\delta : V_G \to \mathcal{T}_\Sigma$, let $\nu' = (\nu\rho)_V = (\mu\sigma\rho)_V$, where $dom(\nu') = V$ and $ran(\nu') = \emptyset$, and let $\varrho_{\nu'} = (\varrho_\nu\rho)_{V_{G_1}\setminus V} = (\varrho_\mu\sigma\rho)_{V_{G_1}\setminus V}$. As $dom(\rho) = V_{G\sigma}$ and $V_G \subseteq V_{G_2^\mu\varrho_\mu}$, so $V_{G\sigma} \subseteq V_{G_2^\mu\varrho_\mu\sigma} = V_{G_2^\mu\varrho_\mu\sigma'\sigma''} = V_{G''\sigma''}$, then $dom(\rho) \subseteq V_{G''}\sigma''$. Let $\rho_1 : V_{G''\sigma''} \setminus V_{G\sigma} \to \mathcal{T}_\Sigma$, so $dom(\rho) \cap dom(\rho_1) = \emptyset$ and $dom(\rho) \cup dom(\rho_1) = V_{G''\sigma''}$, such that $\psi(\rho \uplus \rho_1)$ is satisfiable. Let $\rho' = \rho \uplus \rho_1$, so $\rho' : V_{G''\sigma''} \to \mathcal{T}_\Sigma$ and $\rho'_{V_{G\sigma}} = \rho$.

By I.H., as $\rho' : V_{G''\sigma''} \to \mathcal{T}_\Sigma$ and $\psi\rho'$ is satisfiable, $\sigma''_{V_{G''}}\rho'$ is a solution for G'', let $\delta' = \sigma''_{V_{G''}}\rho'$, so $\delta' : V_{G''} \to \mathcal{T}_\Sigma$ and $\psi_1\sigma'\delta'$ is ground. As $dom(\nu') = V$ and $ran(\nu') = \emptyset$ then $(\nu\rho')_V = (\nu\rho)_V = \nu'$. Also, as $V_{G\sigma'} \subseteq V_{G''}$ and $dom(\rho_1) = V_{G''\sigma''} \setminus V_{G\sigma}$, then $G\sigma'\delta' = G\sigma'\sigma''_{V_{G''}}\rho' = G\sigma'\sigma''_{V_{G\sigma'}}\rho' = G\sigma'\sigma''\rho' = G\sigma\rho' = G\sigma(\rho\uplus\rho_1) = G\sigma\rho = G\sigma_{V_G}\rho = G\delta$. Also, as $V_{\psi_1} \subseteq V_G$, so $\psi_1\sigma'\delta' = \psi_1\delta$, and $\psi_1\sigma'\delta'$ is a subformula of $\psi\rho'$, so $\psi_1\sigma'\delta'$ is ground and satisfiable, then $E_0 \vdash \psi_1\delta$.

As δ' is a solution for $G'' = G_2^{\mu'}\varrho_{\mu'}$ and $G_2 = u_0 \to v_0/ST'_0\,;ST\,(\wedge\,\Delta_0) \mid \psi_0 \wedge \phi_0 \mid V, none$, then $[v_0\mu'\varrho_{\mu'}\delta']_E \in (ST'_0\,;ST)^{\mu'}\varrho_{\mu'}\delta'@[u_0\mu'\varrho_{\mu'}\delta']_E$ (†). Now, as $(u_0, v_0)\mu\varrho_\mu = (u_1, v_1)$ and $\delta' = \sigma''_{V_{G''}}\rho'$, then we can write (†) as $[v_1\delta']_E \in (ST'_0\,;ST)^{\mu'}\varrho_{\mu'}\sigma''_{V_{G''}}\rho'@[u_1\delta']_E$ (††).

As $(ST'_0\,;ST)^{\mu'}\varrho_{\mu'} \in G''$, then $(ST'_0\,;ST)^{\mu'}\varrho_{\mu'}\sigma''_{V_{G''}}\rho' = (ST'_0\,;ST)^{\mu'}\varrho_{\mu'}\sigma''\rho' = (ST'_0;ST)^{\mu}\varrho_\mu\sigma'\sigma''\rho' = (ST'_0;ST)^{\mu}\varrho_\mu\sigma\rho' = (ST'_0;ST)^{\nu}\varrho_\nu\rho' = (ST'_0;ST)^{\nu}\varrho_\nu\rho = (ST'_0\,;ST)^{\nu'}\varrho_{\nu'}$, because $G\sigma\rho' = G\sigma\rho$ and $(ST'_0\,;ST)^{\nu}\varrho_\nu$ is a strategy in $G\sigma$, so we can write (††) as $[v_1\delta']_E \in (ST'_0;ST)^{\nu'}\varrho_{\nu'}@[u_1\delta']_E$, hence there is a c.p.t. $\dfrac{\dfrac{F_1}{u_1\delta'\to w/(ST'_0)^{\nu'}\varrho_{\nu'}} \quad \dfrac{F_2}{w\to v_1\delta'/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta'\to v_1\delta'/(ST'_0\,;ST)^{\nu'}\varrho_{\nu'}}$, for some term $w \in \mathcal{H}_\Sigma$, with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$.

As ϱ_ν is idempotent and ρ is ground then $\varrho_{\nu'}$ is also idempotent. Then, as ν' is ground, $dom(\nu') = V$, and $dom(\varrho_{\nu'}) \cap V = \emptyset$, we can write $\dfrac{F_1}{u_1\delta'\to w/(ST'_0)^{\nu'}\varrho_{\nu'}}$ as $\dfrac{F_1}{u_1\delta'\to w/(ST_0\gamma'[\bar{x}'\gamma_0\varrho_{\nu'}]_{\bar{p}})^{\nu'}\varrho_{\nu'}}$. Let α be a renaming such that $dom(\alpha) = V_{\varrho_{\nu'}}$ and $ran(\alpha)$ is away from all known variables. By Lemma 12 there is a c.p.t. of the form $\dfrac{F_3}{u_1\delta'\to w/(ST_0\gamma'[\bar{x}'\gamma_0\varrho_{\nu'}]_{\bar{p}})^{\nu'}(\varrho_{\nu'}\alpha)}$. Now, we can apply Lemma 13, so there is also a c.p.t. of the form $\dfrac{F_4}{u_1\delta'\to w/(ST_0\gamma'[\bar{x}'\gamma_0\varrho_{\nu'}]_{\bar{p}})^{\nu'}}$.

As $G' \rightsquigarrow^*_{\sigma'} G''$ and all the calculus rules apply always to the leftmost open goal of any goal, then also $G''' = \bigwedge_{j=1}^m(l_j\gamma'\gamma \to r_j\gamma'\gamma/\texttt{idle}) \mid \psi_2 \mid V, \mu \rightsquigarrow^*_{\sigma'} nil \mid \psi_2\sigma' \wedge \psi_4 \mid V, (\mu\sigma')_V$. Then, by I.H., for every substitution $\theta : V_{G'''\sigma'} \to \mathcal{T}_\Sigma$ such that $(\psi_2\sigma' \wedge \psi_4)\theta$ is satisfiable, $\sigma'_{V_{G'''}}\theta$ is a solution of G''', so $\bar{l}\gamma'\gamma\sigma'\theta =_E \bar{r}\gamma'\gamma\sigma'\theta$ (†).

Let $\gamma'' = \{\bar{x} \mapsto \bar{t}(\nu' \uplus \varrho_{\nu'})\}$, so $CS^{\nu'}(\bar{t}\nu'\varrho_{\nu'}) = ST_0^{\nu'}\gamma''$ *if* $(C')^{\nu'}\gamma''$, let $C'' = (C')^{\nu'}(\gamma' \cup \gamma'') = C'\nu'(\gamma' \cup \gamma'')$, and let $\delta'' = \sigma'\delta'$. Then:

– $C''\delta''_{V_{C''}} = C''\delta'' = C'\nu'(\gamma'\cup\gamma'')\delta'' = C'\nu'(\gamma'\cup\gamma'')\delta'' = C'\nu'[\bar{x}'']_{\bar{q}}(\gamma'\cup\gamma'')\delta'' = C'\gamma'\nu'[\bar{x}'']_{\bar{q}}\gamma''\delta'' = C'\gamma'\nu'[\bar{t}(\nu'\uplus\varrho_{\nu'})]_{\bar{q}}\delta'' = C'\gamma'[\bar{t}\varrho_{\nu'}]_{\bar{q}}\nu'\delta'' = C'\gamma'[\bar{t}\varrho_{\nu'}]_{\bar{q}}\nu'\sigma'\delta' = C'\gamma'[\bar{t}\varrho_{\nu'}]_{\bar{q}}\nu'\sigma'\sigma''_{V_{G''}}\rho'$ since $(dom(\gamma') \cup ran(\gamma')) \cap V = \emptyset$, ν' is ground, $dom(\nu') = V$, and $dom(\varrho_{\nu'}) \cap V = \emptyset$,

– $C\gamma'\gamma = C'\mu\gamma'\gamma = C'\mu[\bar{x}'']_{\bar{q}}\gamma'\gamma = C'\gamma'\mu[\bar{x}'']_{\bar{q}}\gamma = C'\gamma'\mu[\bar{t}\mu\varrho_\mu]_{\bar{q}} = C'\gamma'[\bar{t}\varrho_\mu]_{\bar{q}}\mu =$

$C'\gamma'[\bar{t}(\varrho_\mu)_{V_{G_1}\setminus V}]_{\bar{q}}\mu$, because γ' is a renaming such that $(dom(\gamma')\cup ran(\gamma'))\cap (dom(\mu)\cup ran(\mu)\cup\hat{x})=\emptyset$ and $V_{\bar{t}}\subseteq V_{G_1}\setminus V$,

- as $C\gamma'\gamma = C'\gamma'[\bar{t}(\varrho_\mu)_{V_{G_1}\setminus V}]_{\bar{q}}\mu$, $V_{C\gamma'\gamma\sigma'}\subseteq V_{G''}$, $\sigma=\sigma'\sigma''$ is idempotent, and $\sigma'\sigma''_{V_{G''}}$ is a restriction of σ, hence also idempotent, then $C\gamma'\gamma\sigma'\delta' = C\gamma'\gamma\sigma'\sigma''_{V_{G''}}\rho' = C'\gamma'[\bar{t}(\varrho_\mu)_{V_{G_1}\setminus V}]_{\bar{q}}\mu\sigma'\sigma''_{V_{G''}}\rho' = C'\gamma'[\bar{t}(\varrho_\mu\sigma'\sigma''_{V_{G''}})_{V_{G_1}\setminus V}]_{\bar{q}}(\mu\sigma'\sigma''_{V_{G''}})_V\sigma'\sigma''_{V_{G''}}\rho' = C'\gamma'[\bar{t}(\varrho_\mu\sigma'\sigma'')_{V_{G_1}\setminus V}]_{\bar{q}}(\mu\sigma'\sigma'')_V\sigma'\sigma''_{V_{G''}}\rho' = C'\gamma'[\bar{t}(\varrho_\mu\sigma)_{V_{G_1}\setminus V}]_{\bar{q}}(\mu\sigma)_V\sigma'\sigma''_{V_{G''}}\rho'$, and
- as $\rho'=\rho\uplus\rho_1$, and ρ is ground, then $C'\gamma'[\bar{t}(\varrho_\mu\sigma)_{V_{G_1}\setminus V}]_{\bar{q}}(\mu\sigma)_V\sigma'\sigma''_{V_{G''}}\rho' = C'\gamma'[\bar{t}(\varrho_\mu\sigma\rho)_{V_{G_1}\setminus V}]_{\bar{q}}(\mu\sigma\rho)_V\sigma'\sigma''_{V_{G''}}\rho' = C'\gamma'[\bar{t}\varrho_{\nu'}]_{\bar{q}}\nu'\sigma'\sigma''_{V_{G''}}\rho'$,

so $C''\delta''_{V_{C''}} = C\gamma'\gamma\sigma'\delta'$. As $C\gamma'\gamma\sigma'\delta'$ is ground, then $\delta''_{V_{C''}}$ is ground, i.e., $\delta''_{V_{C''}}: V_{C''}\to\mathcal{T}_\Sigma$, and $\phi\gamma'\gamma\sigma'\delta'$ is ground.
As $\psi\rho'$ is satisfiable and $\psi\rho' = \psi_3\sigma''\rho'\wedge\psi_6\rho'$, then also $\psi_3\sigma''\rho' = (\psi_2\sigma'\wedge\psi_4)\sigma''\rho' = (\psi_2\sigma'\wedge\psi_4)\sigma''_{V_{G''}}\rho' = (\psi_2\sigma'\wedge\psi_4)\delta' = ((\psi_1\wedge\phi\gamma'\gamma)\sigma'\wedge\psi_4)\delta'$ is satisfiable, so $\phi\gamma'\gamma\sigma'\delta'$, i.e., $\phi\gamma'\gamma\delta''$, is satisfiable. As $\phi\gamma'\gamma\sigma'\delta'$ is also ground, then $E_0\vdash\phi\gamma'\gamma\delta''$.

By (†), as $(\psi_2\sigma'\wedge\psi_4)\delta'$ is satisfiable, $\bar{l}\gamma'\gamma\sigma'\delta' =_E \bar{r}\gamma'\gamma\sigma'\delta'$, i.e., $\bar{l}\gamma'\gamma\delta'' =_E \bar{r}\gamma'\gamma\delta''$. As also $CS(\bar{t})^{\nu'}\varrho_{\nu'} = CS(\bar{t}\nu'\varrho_{\nu'})$, then there are derivation rules $\frac{u_1\delta'\to w/ST_0^{\nu'}(\gamma'\cup\gamma'')\delta''}{u_1\delta'\to w/CS(\bar{t}\nu'\varrho_{\nu'})}$, i.e., $\frac{u_1\delta'\to w/(ST_0\gamma'[\bar{x}'\gamma_0\varrho_{\nu'}]_{\bar{p}})^{\nu'}}{u_1\delta'\to w/CS(\bar{t}\nu'\varrho_{\nu'})}$, and $\frac{u_1\delta'\to w/CS(\bar{t}\nu'\varrho_{\nu'})\quad w\to v_1\delta'/ST^{\nu'}\varrho_{\nu'}}{u_1\delta'\to v_1\delta'/(CS(\bar{t})\,;ST)^{\nu'}\varrho_{\nu'}}$ in $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$, so there is a c.p.t. $\dfrac{\dfrac{}{u_1\delta'\to w/(ST_0\gamma'[\bar{x}'\gamma_0\varrho_{\nu'}]_{\bar{p}})^{\nu'}}F_4}{u_1\delta'\to w/CS(\bar{t}\nu'\varrho_{\nu'})}\quad\dfrac{}{w\to v_1\delta'/ST^{\nu'}\varrho_{\nu'}}F_2}{u_1\delta'\to v_1\delta'/(CS(\bar{t})\,;ST)^{\nu'}\varrho_{\nu'}}$, and $v_1\delta'\in(CS(\bar{t})\,;ST)^{\nu'}\varrho_{\nu'}@u_1\delta'$.

As $G\delta'=G\delta$, this is the same as $v_1\delta\in(CS(\bar{t})\,;ST)^{\nu'}\varrho_{\nu'}@u_1\delta$ so, as $E_0\vdash\psi_1\delta$, $\sigma_{vars(G)}\rho$ is a solution of G.

16. Rule $[m]$ (match):

$G = u_1\to v_1/(\texttt{match}\,t_1\,\texttt{s.t.}\,\bigwedge_{j=1}^m(l'_j=r'_j)\wedge\phi_1;ST)^\mu\varrho_\mu(\wedge\Delta)\mid\psi_1\mid V,\mu\rightsquigarrow_{[m],\sigma_1}(\bigwedge_{j=1}^m(l'_j\to r'_j/\texttt{idle})^\mu\varrho_\mu\wedge u_1\to v_1/ST^\mu\varrho_\mu\,(\wedge\,\Delta)\mid\psi_2\mid V,\mu)\sigma_1 = G'\sigma_1$, let $t=t_1^\mu\varrho_\mu$, $\phi=\phi_1^\mu\varrho_\mu$, $\bar{l}=(\bar{l}')^\mu\varrho_\mu$, and $\bar{r}=(\bar{r}')^\mu\varrho_\mu$, where $abstract_{\Sigma_1}(t)=\langle\lambda\bar{x}.t^\circ;\sigma^\circ;\phi^\circ\rangle$, $t^\circ=t[\bar{x}]_{\bar{q}}$, with $\bar{x}=x_1,\ldots,x_l$ and $\bar{q}=q_1,\ldots,q_l$, $\phi^\circ=(\bigwedge_{i=1}^l x_i=t|_{q_i})$, hence $V_{t^\circ}\cup V_{\phi^\circ}=V_t\cup\hat{x}$, $\sigma_1\in CSU_B(u_1=t^\circ)$, $\psi_2=\psi_1\wedge\phi\wedge\phi^\circ$, so $V_G\subseteq V_{G'}$, $\psi_2\sigma_1$ is satisfiable, and $G'\sigma_1\rightsquigarrow^+_{\sigma'} nil\mid\psi\mid V,\nu$, let $\sigma=\sigma_1\sigma'$, where $\nu=(\mu\sigma)_V=(\mu\sigma_1\sigma')_V$, so $\sigma_{V_G}\mid\psi$ is a computed answer for G and $\sigma'_{V_{G'\sigma_1}}\mid\psi$ is a computed answer for $G'\sigma_1$.

Let $\rho: V_{G\sigma}\to\mathcal{T}_\Sigma$ be a substitution such that $\psi\rho$ is satisfiable, $\delta=\sigma_{V_G}\rho$, so $\delta: V_G\to\mathcal{T}_\Sigma$, $\rho_1=\rho_{V_{G'\sigma}}$, so also $\psi\rho_1$ is satisfiable, $\nu'=(\nu\rho)_V$, where $dom(\nu')=V$ and $ran(\nu')=\emptyset$, and $\varrho_{\nu'}=(\varrho_\mu\delta)_{\setminus V}$. As $dom(\rho)=V_{G\sigma}$ then $dom(\rho_1)=V_{G\sigma}\cap V_{G'\sigma}$. Let $\rho_2=\rho_{V_{G\sigma}\setminus V_{G'\sigma}}$, so $\rho=\rho_1\uplus\rho_2$, and $\rho'_1: V_{G'\sigma}\setminus V_{G\sigma}\to\mathcal{T}_\Sigma$, so $dom(\rho_1)\cap dom(\rho'_1)=\emptyset$ and $dom(\rho_1)\cup dom(\rho'_1)=V_{G'\sigma}$, such that $\psi(\rho_1\uplus\rho'_1)$ is satisfiable, and let $\rho'=\rho_1\uplus\rho'_1$, so $\rho': V_{G'\sigma}\to\mathcal{T}_\Sigma$. By definition of ν and ρ_1, $ran(\nu)\cup(V\setminus dom(\nu))\subseteq dom(\rho_1)$ so, as $dom(\nu')=V$ and $ran(\nu')=\emptyset$, $\nu'=(\nu\rho)_V=(\nu\rho_1)_V=(\nu\rho')_V$.

By I.H., as $\rho': V_{G'\sigma_1\sigma'}\to\mathcal{T}_\Sigma$ and $\psi\rho'$ is satisfiable, $\sigma'_{V_{G'\sigma_1}}\rho'$ is a solution for $G'\sigma_1$, let $\delta'=\sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$, $\varrho'=(\varrho_\mu\delta')_{\setminus V}$, and $\rho''=\delta'_{V_{t,\phi,\bar{l},\bar{r}}\setminus V_G}$.

As in rule $[i1]$, if then else, and using the fact that $V_{\bar{l},\bar{r}} \subseteq V_{G'}$, we have the following intermediate results:

- $(\mu\delta)_V = (\mu\delta')_V$,
- $V_{(t,\phi,\bar{l},\bar{r})\sigma} \subseteq V_{G'\sigma}$,
- $V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\nu'}} \subseteq V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\mu}}$,
- $V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\mu}} \setminus V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\nu'}} \subseteq V^{\mu}$, and
- $(t,\phi)\sigma\rho' = (t_1,\phi_1)^{\nu'}\varrho_{\nu'}\rho''$.

Using the proof for the last result we also get $(\bar{l},\bar{r})\sigma\rho' = (\bar{l}',\bar{r}')^{\nu'}\varrho_{\nu'}\rho''$.

As $\sigma'_{V_{G'\sigma_1}}\rho'$ is a solution for $G'\sigma_1$ then, by I.H.:

(a) $E_0 \vdash \psi_2\delta'$, i.e., $E_0 \vdash (\psi_1 \wedge \phi \wedge \phi^{\circ})\delta'$,

(b) there are closed proof trees for each open goal in $\Delta\delta'$, with respect to $\mathcal{D}^{(\mu\delta')_V}_{\mathcal{R},Call_{\mathcal{R}}}$ $(=\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$, we use ν' instead of $(\mu\delta')_V$ in (c) and (d)),

(c) $[v_1\delta']_E \in ST^{\nu'}\varrho'@[u_1\delta']_E$, and

(d) $[r_j\delta']_E \in \mathtt{idle}@[l_j\delta']_E$, for $1 \leq j \leq m$, i.e., $\bar{l}\delta' =_E \bar{r}\delta'$,

so:

(a) i. $V_{\psi_2} \subseteq V_{G'}$ implies $\psi_2\sigma_1\sigma'_{V_{G'\sigma_1}} = \psi_2\sigma_1\sigma' = \psi_2\sigma$, so $E_0 \vdash \psi_2\sigma\rho'$, where $\psi_2\sigma\rho'$ is ground, because $V_{\psi_2\sigma} \subseteq V_{G'\sigma}$ and $\rho' : V_{G'\sigma} \to \mathcal{T}_{\Sigma}$, hence $E_0 \vdash \psi_1\sigma\rho'$, $E_0 \vdash \phi^{\circ}\sigma\rho'$, and $E_0 \vdash \phi\sigma\rho'$, all ground expressions.

ii. $V_{\psi_1\sigma} \subseteq V_{G\sigma}$ and $dom(\rho) = V_{G\sigma}$ implies $\psi_1\sigma\rho \in \mathcal{T}_{\Sigma}$ so, as $\rho' = \rho \uplus \rho'_1$, $\psi_1\sigma\rho' = \psi_1\sigma(\rho \uplus \rho'_1) = \psi_1\sigma\rho = \psi_1\delta$, hence $E_0 \vdash \psi_1\delta$ (†).

(b) As in subcase (a)-ii, $V_{\Delta} \subseteq V_G$ implies $\Delta\delta' = \Delta\delta$, and the same closed proof trees are valid for each open goal in $\Delta\delta$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$ (††).

(c) Again, $V_{v_1,u_1} \subseteq V_G$ implies that $v_1\delta' = v_1\delta$ and $u_1\delta' = u_1\delta$. Then there is a c.p.t. of the form $\frac{F}{u_1\delta \to v_1\delta / ST^{\nu'}\varrho'}$, with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$.

(d) As $(\bar{l},\bar{r})\delta' = (\bar{l},\bar{r})\sigma_1\sigma'_{V_{G'\sigma_1}}\rho' = (\bar{l},\bar{r})\sigma_1\sigma'\rho' = (\bar{l},\bar{r})\sigma\rho' = (\bar{l}',\bar{r}')^{\nu'}\varrho_{\nu'}\rho''$, then $(\bar{l}')^{\nu'}\varrho_{\nu'}\rho'' =_E (\bar{r}')^{\nu'}\varrho_{\nu'}\rho''$.

We prove (a) $ST^{\nu'}\varrho_{\nu'}\rho'' = ST^{\nu'}\varrho'$ and (b) $\rho'' : V_{t,\phi,\bar{l},\bar{r}} \setminus V_G \to \mathcal{T}_{\Sigma}$:

(a) As $\varrho_{\nu'} = (\varrho_{\mu}\delta)_{\setminus V}$, $\delta = \sigma_{V_G}\rho$, $\sigma = \sigma_1\sigma'$, $\delta' = \sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$, $\varrho' = (\varrho_{\mu}\delta')_{\setminus V}$, and $V_{ST^{\nu'}} \cap V = \emptyset$ this is the same as $ST^{\nu'}\varrho_{\mu}(\sigma_1\sigma')_{V_G}\rho\rho'' = ST^{\nu'}\varrho_{\mu}\sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$. Let $y \in V_{ST^{\nu'}\varrho_{\mu}}$, so $y \notin V$. There are two options:

i. $y \in V_G$. Then $V_{y\sigma_1} \subseteq V_{G\sigma_1} \subseteq V_{G'\sigma_1}$, so $y(\sigma_1\sigma')_{V_G} = y\sigma_1\sigma' = y\sigma_1\sigma'_{V_{G'\sigma_1}}$. Also $y(\sigma_1\sigma')_{V_G} = y\sigma$, hence $V_{y(\sigma_1\sigma')_{V_G}} \subseteq V_{G\sigma}$. Then, as $\rho : V_{G\sigma} \to \mathcal{T}_{\Sigma}$, $y(\sigma_1\sigma')_{V_G}\rho$ is ground, so $y(\sigma_1\sigma')_{V_G}\rho\rho'' = y(\sigma_1\sigma')_{V_G}\rho = y\sigma_1\sigma'_{V_{G'\sigma_1}}\rho = y\sigma_1\sigma'_{V_{G'\sigma_1}}(\rho \cup \rho'_1) = y\sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$;

ii. $y \notin V_G$, so $y(\sigma_1\sigma')_{V_G} = y$. As $ran(\sigma) \cap V_{ST^{\mu}\varrho_{\mu}} = \emptyset$ and $V_{ST^{\nu'}\varrho_{\mu}} \subseteq V_{ST^{\mu}\varrho_{\mu}}$ then $ran(\sigma) \cap V_{ST^{\nu'}\varrho_{\mu}} = \emptyset$ so $y \notin V_{G\sigma}$ and, as $dom(\rho) = V_{G\sigma}$, $y(\sigma_1\sigma')_{V_G}\rho = y$. Then:

A. if $y \in V_{t,\phi,\bar{l},\bar{r}}$ then $y(\sigma_1\sigma')_{V_G}\rho\rho'' = y\rho'' = y\delta'_{V_{t,\phi,\bar{l},\bar{r}}\setminus V_G} = y\delta' = y\sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$, ground term because $V_{y\sigma_1} \subseteq V_{(t,\phi,\bar{l},\bar{r})\sigma_1} \subseteq V_{G'\sigma_1}$ and $\rho' : V_{G'\sigma_1\sigma'} \to \mathcal{T}_{\Sigma}$;

B. if $y \notin V_{t,\phi,\bar{l},\bar{r}}$ then $y(\sigma_1\sigma')_{V_G}\rho\rho'' = y\rho'' = y\delta'_{V_{t,\phi,\bar{l},\bar{r}}\setminus V_G} = y$. As $dom(\sigma_1) \subseteq (V_{u_1} \cup V_{t^\circ}) \subseteq (V_G \cup V_{t,\phi} \cup \bar{x}) \subseteq (V_G \cup V_{t,\phi,\bar{l},\bar{r}} \cup \bar{x})$ and $y \notin (V_G \cup V_{t,\phi,\bar{l},\bar{r}})$ then $y\sigma_1 = y$ so, as $ran(\sigma_1) \cap V_{ST^{\nu'}\varrho_\mu} = \emptyset$, $y\sigma_1 \notin V_{G\sigma_1}$, and $y\sigma_1\sigma'_{V_{G'\sigma_1}} = y \notin V_{G\sigma_1\sigma'_{V_{G'\sigma_1}}}$ so, as $\rho' : V_{G'\sigma_1\sigma'} \to \mathcal{T}_\Sigma$, $y\sigma_1\sigma'_{V_{G'\sigma_1}}\rho' = y = y(\sigma_1\sigma')_{V_G}\rho\rho''$.

(b) As $dom(\rho'') \subseteq (V_{t,\phi,\bar{l},\bar{r}} \setminus V_G)$ and, from (a.ii.A), $y \in (V_{t,\phi,\bar{l},\bar{r}} \setminus V_G) \implies V_{y\rho''} = \emptyset$ then $dom(\rho'') = V_{t,\phi,\bar{l},\bar{r}} \setminus V_G$, hence $\rho'' : V_{t,\phi,\bar{l},\bar{r}} \setminus V_G \to \mathcal{T}_\Sigma$.

Now, we prove (a) $dom(\rho'') = V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\nu'}\varrho_{\nu'}}$, (b) $E_0 \vdash \phi_1^{\nu'}\varrho_{\nu'}\rho''$, and (c) $u_1\delta =_E t_1^{\nu'}\varrho_{\nu'}\rho''$:

(a) As $dom(\rho'') = V_{t,\phi,\bar{l},\bar{r}}\setminus V_G$, $V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\nu'}} \subseteq V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\mu}}$, $V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\mu}}\setminus V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\nu'}} \subseteq V^\mu \subseteq V_G$, and $dom(\varrho_\mu) \cap V^\mu = \emptyset$, then $dom(\rho'') = V_{(t,\phi,\bar{l},\bar{r})} \setminus V_G = V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\mu}\varrho_\mu} \setminus V_G = V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\nu'}\varrho_\mu} \setminus V_G$. Also, as $\varrho_{\nu'} = (\varrho_\mu\delta)_{\setminus V}$ and $V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\nu'}} \cap V = \emptyset$, then $V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\nu'}\varrho_{\nu'}} = V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\nu'}(\varrho_\mu\delta)_{\setminus V}} = V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\nu'}\varrho_\mu\delta}$, so we prove $V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\nu'}\varrho_\mu\delta} = V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\nu'}\varrho_\mu} \setminus V_G$, which is trivial, since $\delta : V_G \to \mathcal{T}_\Sigma$.

(b) Immediate, since $E_0 \vdash \phi\sigma\rho'$ and $\phi\sigma\rho' = \phi_1^{\nu'}\varrho_{\nu'}\rho''$.

(c) $u_1\sigma_1 =_B t^\circ\sigma_1$ and $\sigma = \sigma_1\sigma'$ imply $u_1\sigma =_B t^\circ\sigma$ so, as $V_{u_1} \subseteq V_G$, $u_1\sigma_{vars(G)} = u_1\sigma =_B t^\circ\sigma$. As $\rho : V_{G\sigma} \to \mathcal{T}_\Sigma$, so $u_1\sigma_{V_G}\rho$ is a ground term, and $\rho' = \rho \uplus \rho'_1$ then $u_1\delta = u_1\sigma_{V_G}\rho = u_1\sigma_{V_G}\rho' =_B t^\circ\sigma\rho' = t[\bar{x}]_{\bar{q}}\sigma\rho' = t\sigma\rho'[\bar{x}\sigma\rho']_{\bar{q}}$.
As $E_0 \vdash \phi^\circ\sigma\rho'$ then $t\sigma\rho'[\bar{x}\sigma\rho']_{\bar{q}} =_{E_0} t\sigma\rho'[t|_{q_1}\sigma\rho', \ldots, t|_{q_l}\sigma\rho']_{\bar{q}} = t\sigma\rho'[t\sigma\rho'|_{\bar{q}}]_{\bar{q}} = t\sigma\rho' = t_1^{\nu'}\varrho_{\nu'}\rho''$, because $t\sigma\rho' = t_1^{\nu'}\varrho_{\nu'}\rho''$, so $u_1\delta =_B t^\circ\sigma\rho' =_{E_0} t_1^{\nu'}\varrho_{\nu'}\rho''$, i.e., $u_1\delta =_E t_1^{\nu'}\varrho_{\nu'}\rho''$.

Then, as $\rho'' : V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\nu'}\varrho_{\nu'}} \to \mathcal{T}_\Sigma$, $E_0 \vdash \phi_1^{\nu'}\varrho_{\nu'}\rho''$, and $(\bar{l}')^{\nu'}\varrho_{\nu'}\rho'' =_E (\bar{r}')^{\nu'}\varrho_{\nu'}\rho''$, there is a derivation rule $\frac{}{w \to w/\mathtt{match}\, t_1^{\nu'}\varrho_{\nu'}\ \mathtt{s.t.}\ \phi_1^{\nu'}\varrho_{\nu'}} \in \mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$, for some term w such that $t_1^{\nu'}\varrho_{\nu'}\rho'' =_E w$. As $u_1\delta =_E t_1^{\nu'}\varrho_{\nu'}\rho''$, then

$$\frac{\dfrac{}{u_1\delta \to u_1\delta/\mathtt{match}\, t_1^{\nu'}\varrho_{\nu'}\ \mathtt{s.t.}\ \phi_1^{\nu'}\varrho_{\nu'}} \quad \dfrac{F}{u_1\delta \to v_1\delta/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta \to v_1\delta/\mathtt{match}\, t_1^{\nu'}\varrho_{\nu'}\ \mathtt{s.t.}\ \phi_1^{\nu'}\varrho_{\nu'}; ST^{\nu'}\varrho_{\nu'}}$$

is a c.p.t., $\rho : vars(G\sigma) \to \mathcal{T}_\Sigma$, $\psi\rho$ is satisfiable, $E_0 \vdash \psi_1\delta$ (†), and there are closed proof trees for each open goal in $\Delta\delta$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$ (††), hence $\sigma_{vars(G)}\rho$ is a solution of G.

17. Rule $[w]$ (matchrew):

$MS = \mathtt{matchrew}\ t_1\ \mathtt{s.t.}\ C_1\ \mathtt{by}\ z_1\ \mathtt{using}\ ST_1, \ldots, z_n\ \mathtt{using}\ ST_n$, let $\bar{z} = \{z_1, \ldots, z_n\}$, where $t_1 = t_1[\bar{z}]_{\bar{p}}$, for appropriate $\bar{p} = \{p_1, \ldots, p_n\}$. $G = u_1 \to v_1/(MS; ST)^\mu\varrho_\mu(\wedge\Delta) \mid \psi_1 \mid V, \mu$, where $C_1 = \bigwedge_{j=1}^m (l'_j = r'_j) \wedge \phi_1$, let $t = t_1^\mu\varrho_\mu$, $\phi = \phi_1^\mu\varrho_\mu$, $\bar{l} = (\bar{l}')^\mu\varrho_\mu$, and $\bar{r} = (\bar{r}')^\mu\varrho_\mu$.

Now, $G \rightsquigarrow_{[w],\sigma_1} (\bigwedge_{j=1}^m (l_j\gamma \to r_j\gamma/\mathtt{idle}) \wedge \bigwedge_{i=1}^n (x_i \to y_i/ST_i^\mu\varrho_\mu\gamma; \mathtt{idle}) \wedge t[\bar{y}]_{\bar{p}} \to v_1/\, ST^\mu\varrho_\mu(\wedge\Delta) \mid \psi_2 \mid V, \mu)\sigma_1 = G'\sigma_1$, where $\bar{x}$ and $\bar{y}$ are fresh versions of $\bar{z}$, γ is a renaming from $\bar{z}$ to $\bar{x}$, $abstract_{\Sigma_1}(t[\bar{x}]_{\bar{p}}) = \langle \lambda\bar{z}.t^\circ; \sigma^\circ; \phi^\circ\rangle$, $t^\circ = t[z_1, \ldots, z_l]_{q_1\ldots q_l}$, $\phi^\circ = (\bigwedge_{i=1}^l z_i = t|_{q_i})$, $\sigma_1 \in CSU_B(u_1 = t^\circ)$, $\psi_2 = \psi_1 \wedge \phi \wedge \phi^\circ$, so $V_G \subseteq V_{G'}$, $\psi_2\sigma_1$ is satisfiable, $G'\sigma_1 \rightsquigarrow^*_{\sigma_2} \bigwedge_{i=1}^n (x_i \to y_i/ST_i^\mu\varrho_\mu\gamma; \mathtt{idle})\sigma_1\sigma_2 \wedge (t[\bar{y}]_{\bar{p}} \to v_1/\, ST^\mu\varrho_\mu)\sigma_1\sigma_2\ (\wedge\, \Delta\sigma_1\sigma_2) \mid \psi_3 \mid V, (\mu\sigma_1\sigma_2)_V = G''$, and $G'' \rightsquigarrow^+_{\sigma''}$

$nil \mid \psi \mid V, \nu$, let $\sigma' = \sigma_2\sigma''$ and $\sigma = \sigma_1\sigma'$, where $\nu = (\mu\sigma)_V$, so $\sigma_{V_G} \mid \psi$ is a computed answer for G and $\sigma'_{V_{G'\sigma_1}} \mid \psi$ is a computed answer for $G'\sigma_1$.

Let $\rho : V_{G\sigma} \to \mathcal{T}_\Sigma$ be a substitution such that $\psi\rho$ is satisfiable, $\delta = \sigma_{V_G}\rho$, so $\delta : V_G \to \mathcal{T}_\Sigma$, $\rho_1 = \rho_{V_{G'\sigma}}$, so also $\psi\rho_1$ is satisfiable, and $\nu' = (\nu\rho)_V$, where $dom(\nu') = V$ and $ran(\nu') = \emptyset$. As $dom(\rho) = V_{G\sigma}$ then $dom(\rho_1) = V_{G\sigma} \cap V_{G'\sigma}$. Let $\rho_2 = \rho_{V_{G\sigma} \setminus V_{G'\sigma}}$, so $\rho = \rho_1 \uplus \rho_2$, and $\rho'_1 : V_{G'\sigma} \setminus V_{G\sigma} \to \mathcal{T}_\Sigma$, so $dom(\rho_1) \cap dom(\rho'_1) = \emptyset$ and $dom(\rho_1) \cup dom(\rho'_1) = V_{G'\sigma}$, such that $\psi(\rho_1 \uplus \rho'_1)$ is satisfiable, and let $\rho' = \rho_1 \uplus \rho'_1$, so $\rho' : V_{G'\sigma} \to \mathcal{T}_\Sigma$, $\delta' = \sigma_{V_{G'}}\rho'$, $\delta'_{\bar{x}} = \delta'_{V_{\bar{x}}}$, and $\delta'_{\bar{y}} = \delta'_{V_{\bar{y}}}$. By definition of ν and ρ_1, $ran(\nu) \cup (V \setminus dom(\nu)) \subseteq dom(\rho_1)$ so, as $dom(\nu') = V$ and $ran(\nu') = \emptyset$, $\nu' = (\nu\rho)_V = (\nu\rho_1)_V = (\nu\rho')_V$.

By I.H., as $\rho' : V_{G'\sigma_1\sigma'} \to \mathcal{T}_\Sigma$ and $\psi\rho'$ is satisfiable, $\sigma'_{V_{G'\sigma_1}}\rho'$ is a solution for $G'\sigma_1$, let $\delta' = \sigma_1\sigma'_{V_{G'\sigma_1}}\rho'$, $\varrho' = (\varrho_\mu\delta')_{\setminus V}$, and $\rho'' = \delta'_{V_{t,\phi,\bar{l},\bar{r}} \setminus V_G}$.

As in rule $[m]$, match, we have the following intermediate results:

- $(\mu\delta)_V = (\mu\delta')_V$,
- $V_{(t,\phi,\bar{l},\bar{r})\sigma} \subseteq V_{G'\sigma}$,
- $V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\nu'}} \subseteq V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\mu}}$,
- $V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\mu}} \setminus V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\nu'}} \subseteq V^\mu$, and
- $(t,\phi,\bar{l},\bar{r})\sigma\rho' = (t_1,\phi_1,\bar{l}',\bar{r}')^{\nu'}\varrho_{\nu'}\rho''$.

As $\sigma'_{V_{G'\sigma_1}}\rho'$ is a solution for $G'\sigma_1$ then, by I.H.:

(a) $E_0 \vdash \psi_2\delta'$, i.e., $E_0 \vdash (\psi_1 \wedge \phi \wedge \phi^\circ)\delta'$,

(b) there are closed proof trees for each open goal in $\Delta\delta'$, with respect to $\mathcal{D}^{(\mu\delta')_V}_{\mathcal{R},Call_\mathcal{R}}$ $(=\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$, we use ν' instead of $(\mu\delta')_V$ in (c)-(e)),

(c) $[v_1\delta']_E \in ST^{\nu'}\varrho'@[t[\bar{y}]_{\bar{p}}\delta']_E$,

(d) $[r_j\delta']_E \in \texttt{idle}@[l_j\delta']_E$, for $1 \le j \le m$, i.e., $\bar{l}\delta' =_E \bar{r}\delta'$, and

(e) $[y_i\delta']_E \in ST^{\nu'}_i\varrho'@[x_i\delta']_E$, for $1 \le i \le n$,

so:

(a) i. $V_{\psi_2} \subseteq V_{G'}$ implies $\psi_2\sigma_1\sigma'_{V_{G'\sigma_1}} = \psi_2\sigma_1\sigma' = \psi_2\sigma$, so $E_0 \vdash \psi_2\sigma\rho'$, where $\psi_2\sigma\rho'$ is ground, because $V_{\psi_2\sigma} \subseteq V_{G'\sigma}$ and $\rho' : V_{G'\sigma} \to \mathcal{T}_\Sigma$, hence $E_0 \vdash \psi_1\sigma\rho'$, $E_0 \vdash \phi^\circ\sigma\rho'$, and $E_0 \vdash \phi\sigma\rho'$, so also $E_0 \vdash \phi_1^{\nu'}\varrho_{\nu'}\rho''$ (†), all ground expressions.

ii. $V_{\psi_1\sigma} \subseteq V_{G\sigma}$ and $dom(\rho) = V_{G\sigma}$ implies $\psi_1\sigma\rho \in \mathcal{T}_\Sigma$ so, as $\rho' = \rho \uplus \rho'_1$, $\psi_1\sigma\rho' = \psi_1\sigma(\rho \uplus \rho'_1) = \psi_1\sigma\rho = \psi_1\delta$, hence $E_0 \vdash \psi_1\delta$ (††).

(b) As in subcase (a)-ii, $V_\Delta \subseteq V_G$ implies $\Delta\delta' = \Delta\delta$, and the same closed proof trees are valid for each open goal in $\Delta\delta$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$ (†††).

(c) Again, $V_{v_1} \subseteq V_G$ implies that $v_1\delta' = v_1\delta$. Then there is a c.p.t. of the form $\frac{F}{t[\bar{y}]_{\bar{p}}\delta' \to v_1\delta / ST^{\nu'}\varrho'}$, with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$.

(d) As $(\bar{l},\bar{r})\delta' = (\bar{l},\bar{r})\sigma_1\sigma'_{V_{G'\sigma_1}}\rho' = (\bar{l},\bar{r})\sigma_1\sigma'\rho' = (\bar{l},\bar{r})\sigma\rho' = (\bar{l}',\bar{r}')^{\nu'}\varrho_{\nu'}\rho''$, then $(\bar{l}')^{\nu'}\varrho_{\nu'}\rho'' =_E (\bar{r}')^{\nu'}\varrho_{\nu'}\rho''$.

(e) As in the previous subcase, $(\bar{x},\bar{y})\delta' = (\bar{x}',\bar{y}')^{\nu'}\varrho_{\nu'}\rho''$, so there are closed proof trees of the form $\frac{F_i}{x_i^{\nu'}\varrho_{\nu'}\rho'' \to y_i^{\nu'}\varrho_{\nu'}\rho'' / ST_i^{\nu'}\varrho'}$, for $1 \le i \le n$, with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_\mathcal{R}}$.

Using the same proofs shown in rule $[m]$, match, we get $ST^{\nu'}\varrho_{\nu'}\rho'' = ST^{\nu'}\varrho'$ and $\rho'' : V_{t,\phi,\bar{l},\bar{r}} \setminus V_G \to \mathcal{T}_\Sigma$.
Also using these proofs, we get: (a) $dom(\rho'') = V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\nu'}\varrho_{\nu'}}$, (b) $E_0 \vdash \phi_1^{\nu'}\varrho_{\nu'}\rho''$, and (c) $u_1\delta =_E t_1^{\nu'}\varrho_{\nu'}\rho''$.
As $V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\nu'}\varrho_{\nu'}} \subseteq V_{(t_1,\phi_1,\bar{l}',\bar{r}')^{\mu}\varrho_{\mu}} \subseteq V_{MS^{\mu}\varrho_{\mu}}$, then $\rho''_{V_{MS^{\mu}\varrho_{\mu}}} = \rho''$, so $ran(\rho''_{V_{MS^{\mu}\varrho_{\mu}}}) \subseteq \mathcal{T}_\Sigma \subseteq \mathcal{T}_\Sigma(\mathcal{X})$ and, as $t_1 = t_1[\bar{x}]_{\bar{p}}$, $(\bar{l}')^{\nu'}\varrho_{\nu'}\rho'' =_E (\bar{r}')^{\nu'}\varrho_{\nu'}\rho''$ and $E_0 \vdash \phi_1^{\nu'}\varrho_{\nu'}\rho''$ (†), there is a derivation rule $\frac{x_1^{\nu'}\varrho_{\nu'}\rho''\to y_1^{\nu'}\varrho_{\nu'}\rho''/ST_1^{\nu'}\varrho'\cdots x_n^{\nu'}\varrho_{\nu'}\rho''\to y_n^{\nu'}\varrho_{\nu'}\rho''/ST_n^{\nu'}\varrho'}{t_1^{\nu'}\varrho_{\nu'}\rho''\to t_1[\bar{y}]_{\bar{p}}^{\nu'}\varrho_{\nu'}\rho''/MS^{\nu'}\varrho'}$ in $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$.
Also $\frac{u_1\delta\to t_1[\bar{y}]_{\bar{p}}^{\nu'}\varrho_{\nu'}\rho''/MS^{\nu'}\varrho_{\nu'} \quad t_1[\bar{y}]_{\bar{p}}^{\nu'}\varrho_{\nu'}\rho''\to v_1\delta/ST^{\nu'}\varrho_{\nu'}}{u_1\delta\to v_1\delta/(MS\,;ST)^{\nu'}\varrho_{\nu'}}$ is a derivation rule in $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$.
As $u_1\delta =_E t_1^{\nu'}\varrho_{\nu'}\rho''$, then

$$\frac{\dfrac{\dfrac{F_1}{x_1^{\nu'}\varrho_{\nu'}\rho''\to y_1^{\nu'}\varrho_{\nu'}\rho''/ST_1^{\nu'}\varrho'}\cdots\dfrac{F_n}{x_n^{\nu'}\varrho_{\nu'}\rho''\to y_n^{\nu'}\varrho_{\nu'}\rho''/ST_n^{\nu'}\varrho'}}{u_1\delta\to t_1[\bar{y}]_{\bar{p}}^{\nu'}\varrho_{\nu'}\rho''/MS^{\nu'}\varrho_{\nu'}} \quad \dfrac{F}{t_1[\bar{y}]_{\bar{p}}^{\nu'}\varrho_{\nu'}\rho''\to v_1\delta/ST^{\nu'}\varrho_{\nu'}}}{u_1\delta \to v_1\delta/(MS\,;ST)^{\nu'}\varrho_{\nu'}}$$

is a c.p.t. with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$. As $\rho : vars(G\sigma) \to \mathcal{T}_\Sigma$, $\psi\rho$ is satisfiable, $E_0 \vdash \psi_1\delta$ (††), and there are closed proof trees for each open goal in $\Delta\delta$ with respect to $\mathcal{D}^{\nu'}_{\mathcal{R},Call_{\mathcal{R}}}$ (†††), then $\sigma_{vars(G)}\rho$ is a solution of G.

□

The following lemma will be used in the proof of the weak completeness of the calculus.

Lemma 14 (Narrowing of equational conditions)**.** *Given an associated rewrite theory $\mathcal{R} = (\Sigma, E_0 \cup B, R)$ closed under B-extensions, and a goal $G = \bigwedge_{j=1}^m (l_j \to r_j/\texttt{idle}) \wedge \Delta^\mu\varrho_\mu \mid \psi \mid V, \mu$, if α is a ground substitution such that $V_G \subseteq dom(\alpha)$, $E_0 \vdash \psi\alpha$, and $\bar{l}\alpha =_E \bar{r}\alpha$, then there exist*

- *a ground substitution α°,*
- *substitutions $\beta_1, \ldots, \beta_m$ from CSUs, let $\beta_i^k = \beta_i\beta_{i+1}\cdots\beta_k$, and*
- *abstractions $abstract_{\Sigma_1}((l_j\beta_1^{j-1}, r_j\beta_1^{j-1})) = \langle\lambda(\bar{x}_j, \bar{y}_j).(l_j^\circ, r_j^\circ); (\theta_{l_j}^\circ, \theta_{r_j}^\circ); (\phi_{l_j}^\circ, \phi_{r_j}^\circ)\rangle$, for $1 \le j \le m$, where $\beta_1^0 = none$, let $\beta = \beta_1^m$,*

such that $dom(\alpha^\circ) = dom(\alpha) \cup V_{\hat{x},\hat{y}}$, $\alpha =_{E_0} \alpha^\circ_{dom(\alpha)}$, $\bar{l}^\circ\alpha^\circ =_E \bar{r}^\circ\alpha^\circ$, $\alpha^\circ \ll_E \beta_{dom(\alpha^\circ)}$, $G \rightsquigarrow^m_{[d1]} \Delta^\nu\varrho_\nu \mid \psi\beta \wedge \bigwedge_{j=1}^m (\phi_{l_j}^\circ \wedge \phi_{r_j}^\circ)\beta_j^m \mid V, \nu$, and for every pair of substitutions ρ and γ such that $ran(\rho)$ is away from all known variables, $\alpha^\circ \ll_E (\beta\rho)_{dom(\alpha^\circ)}$, and $\alpha^\circ =_E (\beta\rho)_{dom(\alpha^\circ)} \cdot \gamma$, it holds that $E_0 \vdash (\psi\beta \wedge \bigwedge_{j=1}^m (\phi_{l_j}^\circ \wedge \phi_{r_j}^\circ)\beta_j^m)\rho\gamma$ and $\Delta^\mu\varrho_\mu\alpha =_E \Delta^\mu\varrho_\mu\beta\rho\gamma$.

Proof. The proof is by induction on m, the number of equational conditions. We also prove that $dom(\beta) \subseteq dom(\alpha^\circ) \cup \bigcup_{j=1}^{m-1} ran(\beta_j)$ (∗).

1. Base case, $m = 1$:

   $\mathbf{G} = l \to r/\texttt{idle} \wedge \Delta^\mu\varrho_\mu \mid \psi \mid V, \mu$, α is a ground substitution, $V_G \subseteq dom(\alpha)$, $E_0 \vdash \psi\alpha$, $l\alpha =_E r\alpha$, and $abstract_{\Sigma_1}((l\beta_1^0, r\beta_1^0)) = abstract_{\Sigma_1}((l, r)) = \langle\lambda(\bar{x}, \bar{y}).(l^\circ, r^\circ); (\theta_l^\circ, \theta_r^\circ); (\phi_l^\circ, \phi_r^\circ)\rangle$, where $l^\circ = l[\bar{x}]_{\bar{p}}$, $r^\circ = r[\bar{y}]_{\bar{q}}$, $\phi_l^\circ = \bigwedge_{i=1}^{i_x} x_i = l|_{p_i}$, and $\phi_r^\circ = \bigwedge_{i=1}^{i_y} y_i = r|_{q_i}$ for appropriate $\bar{p}$, $\bar{q}$, i_x, and i_y, so $V_{l^\circ,r^\circ,\phi_l^\circ,\phi_r^\circ} = V_{l,r} \cup \hat{x} \cup \hat{y} \subseteq V_G \cup \hat{x} \cup \hat{y} \subseteq dom(\alpha) \cup \hat{x} \cup \hat{y} = dom(\alpha^\circ)$, hence $V_{\phi_l^\circ,\phi_r^\circ} \subseteq dom(\alpha^\circ)$. As $V_{\phi_l^\circ,\phi_r^\circ} \subset \mathcal{X}_0$ then also $V_{\phi_l^\circ,\phi_r^\circ} \subseteq dom(\alpha^\circ) \cap \mathcal{X}_0$. Then:

- by Lemma 9, there exists a ground substitution α° such that $l^\circ\alpha^\circ =_B r^\circ\alpha^\circ$, $E_0 \vdash (\phi_l^\circ \wedge \phi_r^\circ)\alpha^\circ$, $dom(\alpha^\circ) = dom(\alpha) \cup \hat{x} \cup \hat{y}$, so $V_{(l^\circ,r^\circ,\phi_l^\circ,\phi_r^\circ)\sigma^\circ} = \emptyset$, and $\alpha =_{E_0} \alpha^\circ_{dom(\alpha)}$, hence there also exists a substitution $\beta_1 \in CSU_B(l^\circ = r^\circ)$, where in this base case $\beta = \beta_1^1 = \beta_1$, such that $dom(\beta) \subseteq dom(\alpha^\circ) = dom(\alpha) \cup \hat{x} \cup \hat{y}$ $(*)$ and $\alpha^\circ \ll_B \beta$. As $\beta \ll \beta_{dom(\alpha^\circ)}$ then $\alpha^\circ \ll_B \beta_{dom(\alpha^\circ)}$, hence $\alpha^\circ \ll_E \beta_{dom(\alpha^\circ)}$;
- as $E_0 \vdash (\phi_l^\circ \wedge \phi_r^\circ)\alpha^\circ$, $\psi\alpha$ is satisfiable, $dom(\alpha^\circ) = dom(\alpha) \cup \hat{x} \cup \hat{y}$, $V_\psi \cap (\hat{x} \cup \hat{y}) = \emptyset$, so $\psi\alpha =_{E_0} \psi\alpha^\circ$ hence $\psi\alpha^\circ$ is satisfiable, and $\alpha^\circ \ll_B \beta$, so $\alpha^\circ_{\mathcal{X}_0} \ll \beta_{\mathcal{X}_0}$, then $(\psi \wedge \phi_l^\circ \wedge \phi_r^\circ)\beta$ is satisfiable, and $\mathbf{G} \rightsquigarrow^1_{[d1]} \Delta^\mu \varrho_\mu \beta \mid (\psi \wedge \phi_l^\circ \wedge \phi_r^\circ))\beta \mid V, (\mu\beta)_V$;
- let ρ such that $\alpha^\circ \ll_E (\beta\rho)_{dom(\alpha^\circ)}$ and let γ such that $\alpha^\circ =_E (\beta\rho)_{dom(\alpha^\circ)} \cdot \gamma$. Then:
  (a) as $V_G \subseteq dom(\alpha)$, $V_\psi \subset \mathcal{X}_0$, and $dom(\alpha^\circ) = dom(\alpha) \cup \hat{x} \cup \hat{y}$ then $\psi\alpha = \psi\alpha^\circ =_{E_0} \psi(\beta\rho)_{dom(\alpha^\circ)}\gamma = \psi\beta\rho\gamma$ so, as $E_0 \vdash \psi\alpha$, also $E_0 \vdash \psi\beta\rho\gamma$;
  (b) as $V_{\phi_l^\circ,\phi_r^\circ} \subseteq dom(\alpha^\circ) \cap \mathcal{X}_0$, then $(\phi_l^\circ \wedge \phi_r^\circ)\alpha^\circ =_{E_0} (\phi_l^\circ \wedge \phi_r^\circ)(\beta\rho)_{dom(\alpha^\circ)}\gamma = (\phi_l^\circ \wedge \phi_r^\circ)\beta\rho\gamma$ so, as $E_0 \vdash (\phi_l^\circ \wedge \phi_r^\circ)\alpha^\circ$, also $E_0 \vdash (\phi_l^\circ \wedge \phi_r^\circ)\beta\rho\gamma$.
  From (a) and (b) we get $E_0 \vdash (\psi \wedge \phi_l^\circ \wedge \phi_r^\circ)\beta\rho\gamma$.
- As $V_G \subseteq dom(\alpha) \subseteq dom(\alpha^\circ)$, $dom(\beta) \subseteq dom(\alpha^\circ)$, $\alpha^\circ =_E (\beta\rho)_{dom(\alpha^\circ)} \cdot \gamma$, and $ran(\rho)$ is away from all known variables, then $\Delta^\mu \varrho_\mu \alpha =_{E_0} \Delta^\mu \varrho_\mu \alpha^\circ =_E \Delta^\mu \varrho_\mu (\beta\rho)_{dom(\alpha^\circ)} \cdot \gamma = \Delta^\mu \varrho_\mu \beta\rho\gamma$.

2. Induction step, $m > 1$:

   $\mathbf{G} = l_1 \to r_1/\texttt{idle} \wedge \bigwedge_{j=2}^m (l_j \to r_j/\texttt{idle}) \wedge \Delta^\mu \varrho_\mu \mid \psi \mid V, \mu$, let $\Delta_2^m = \bigwedge_{j=2}^m (l_j \to r_j/\texttt{idle})$.

   As in the base case, there exist a ground substitution δ° and a substitution $\beta_1 \in CSU_B(l_1^\circ = r_1^\circ)$, so $ran(\beta_1) \cap (V_G \cup V_{\hat{x},\hat{y}} \cup V_{l_1^\circ,r_1^\circ}) = \emptyset$, such that $\alpha =_{E_0} \delta^\circ_{dom(\alpha)}$, $dom(\beta_1) \subseteq dom(\delta^\circ) = dom(\alpha) \cup \hat{x}_1 \cup \hat{y}_1$, $\delta^\circ \ll_B \beta_1 \ll (\beta_1)_{dom(\delta^\circ)}$, $(\psi \wedge \phi_l^\circ \wedge \phi_r^\circ)\beta_1$ is satisfiable, so $\mathbf{G} \rightsquigarrow^1_{[d1]} (\Delta_2^m \wedge \Delta^\mu \varrho_\mu)\beta_1 \mid (\psi \wedge \phi_{l_1}^\circ \wedge \phi_{r_1}^\circ))\beta_1 \mid V, (\mu\beta_1)_V = \mathbf{G_1}$, and for every pair of substitutions ρ and γ such that $\delta^\circ \ll_E (\beta_1\rho)_{dom(\delta^\circ)}$ and $\delta^\circ =_E (\beta_1\rho)_{dom(\delta^\circ)} \cdot \gamma$ it holds that $E_0 \vdash (\psi \wedge \phi_l^\circ \wedge \phi_r^\circ)\beta_1\rho\gamma$ and $(\Delta_2^m \wedge \Delta^\mu \varrho_\mu)\alpha =_E (\Delta_2^m \wedge \Delta^\mu \varrho_\mu)\beta_1\rho\gamma$.

   As $\delta^\circ \ll_B \beta_1$ and δ° is ground, then there exists a ground substitution δ_1 such that $dom(\delta_1) = ran(\beta_1) \cup (dom(\delta^\circ) \setminus dom(\beta_1))$, where $ran(\beta_1) \cap V_G = \emptyset$, and $\delta^\circ =_B \beta_1 \cdot \delta_1$, so $dom(\beta_1\delta_1) = ran(\beta_1) \cup dom(\delta^\circ) = ran(\beta_1) \cup dom(\alpha) \cup \hat{x}_1 \cup \hat{y}_1$. Then:

   - as $\delta^\circ =_B (\beta_1\delta_1)_{\setminus ran(\beta_1)}$, so $\delta^\circ_{\mathcal{X}_0} = (\beta_1\delta_1)_{\mathcal{X}_0 \setminus ran(\beta_1)}$, $dom(\delta^\circ) = dom(\alpha) \cup \hat{x}_1 \cup \hat{y}_1$, and $\alpha =_{E_0} \delta^\circ_{dom(\alpha)} = \delta^\circ_{\setminus(\hat{x}_1 \cup \hat{y}_1)}$, then $\alpha =_{E_0} \delta^\circ_{\setminus(\hat{x}_1 \cup \hat{y}_1)} =_B (\beta_1\delta_1)_{\setminus(ran(\beta_1) \cup \hat{x}_1 \cup \hat{y}_1)} = (\beta_1\delta_1)_{dom(\alpha)}$, i.e., $\alpha =_E (\beta_1\delta_1)_{dom(\alpha)}$;
   - $V_{\Delta_2^m} \cap (\hat{x}_1 \cup \hat{y}_1) = \emptyset$ implies $\Delta_2^m\beta_1\delta_1 = \Delta_2^m(\beta_1\delta_1)_{\setminus(\hat{x}_1 \cup \hat{y}_1)} =_E \Delta_2^m\alpha$. Then, since $\bigwedge_{j=2}^m (l_j\alpha =_E r_j\alpha)$, $\bigwedge_{j=2}^m (l_j\beta_1\delta_1 =_E r_j\beta_1\delta_1)$ $(\dagger)$;
   - as $E_0 \vdash \psi\alpha$, $V_\psi \subset \mathcal{X}_0$, $\delta^\circ_{\mathcal{X}_0} = (\beta_1\delta_1)_{\mathcal{X}_0 \setminus ran(\beta_1)}$, and $V_\psi \cap (ran(\beta_1) \cup \hat{x}_1 \cup \hat{y}_1) = \emptyset$, then $\psi\beta_1\delta_1 = \psi(\beta_1\delta_1)_{\setminus(\hat{x}_1 \cup \hat{y}_1)} =_{E_0} \psi\alpha$, so $E_0 \vdash \psi\beta_1\delta_1$;
   - as $\delta^\circ_{\mathcal{X}_0} = (\beta_1\delta_1)_{\mathcal{X}_0 \setminus ran(\beta_1)}$ and $V_{\phi_{l_1}^\circ,\phi_{r_1}^\circ} \cap ran(\beta_1) = \emptyset$, then $(\phi_{l_1}^\circ \wedge \phi_{r_1}^\circ)\beta_1\delta_1 = (\phi_{l_1}^\circ \wedge \phi_{r_1}^\circ)\delta^\circ$ so, as $E_0 \vdash (\phi_{l_1}^\circ \wedge \phi_{r_1}^\circ)\delta^\circ$, also $E_0 \vdash (\phi_{l_1}^\circ \wedge \phi_{r_1}^\circ)\beta_1\delta_1$; and
   - as $E_0 \vdash (\phi_{l_1}^\circ \wedge \phi_{r_1}^\circ)\beta_1\delta_1$ and $E_0 \vdash \psi\beta_1\delta_1$, then $E_0 \vdash (\psi \wedge \phi_{l_1}^\circ \wedge \phi_{r_1}^\circ)\beta_1\delta_1(\dagger\dagger)$.

Then, by (†) and (††), we can apply the I.H. and there exist a ground substitution δ_1°, substitutions $\beta_2, \ldots, \beta_m$ from CSUs, and, for $2 \leq j \leq m$, abstractions $abstract_{\Sigma_1}((l_j\beta_1\beta_2^{j-1}, r_j\beta_1\beta_2^{j-1})) = \langle\lambda(\bar{x}_j, \bar{y}_j).(l_j^\circ, r_j^\circ); (\theta_{l_j}^\circ, \theta_{r_j}^\circ); (\phi_{l_j}^\circ, \phi_{r_j}^\circ)\rangle$, where $\beta_2^1 = none$, such that $dom(\beta_2^m) \subseteq dom(\delta_1^\circ) \cup \bigcup_{j=2}^{m-1} ran(\beta_j)$, $dom(\delta_1^\circ) = dom(\beta_1\delta_1) \cup (V_{\hat{x},\hat{y}} \setminus (\hat{x}_1 \cup \hat{y}_1))$, $\beta_1\delta_1 =_{E_0} (\delta_1^\circ)_{dom(\beta_1\delta_1)}$, $l_j^\circ\delta_1^\circ =_E r_j^\circ\delta_1^\circ$, for $2 \leq j \leq m$, $\delta_1^\circ \ll_E (\beta_2^m)_{dom(\delta_1^\circ)}$, $\mathbf{G_1} \rightsquigarrow_{[d1]}^{m-1} \Delta^\mu\varrho_\mu\beta_1\beta_2^m \mid (\psi \wedge \phi_{l_1}^\circ \wedge \phi_{r_1}^\circ)\beta_1\beta_2^m \wedge \bigwedge_{j=2}^m(\phi_{l_j}^\circ \wedge \phi_{r_j}^\circ)\beta_j^m \mid V, (\mu\beta_1\beta_2^m)_V$, and for every pair of substitutions ρ and γ such that $\delta_1^\circ \ll_E (\beta_2^m\rho)_{dom(\delta_1^\circ)}$ and $\delta_1^\circ =_E (\beta_2^m\rho)_{dom(\delta_1^\circ)} \cdot \gamma$ it holds that $E_0 \vdash ((\psi \wedge \phi_{l_1}^\circ \wedge \phi_{r_1}^\circ)\beta_1\beta_2^m \wedge \bigwedge_{j=2}^m(\phi_{l_j}^\circ \wedge \phi_{r_j}^\circ)\beta_j^m)\rho\gamma$ and $(\Delta_2^m \wedge \Delta^\mu\varrho_\mu)\beta_1\delta_1 =_E (\Delta_2^m \wedge \Delta^\mu\varrho_\mu)\beta_1\beta_2^m\rho\gamma$.

As $\beta_1\beta_2^m = \beta_1^m = \beta$, this is the same as $\mathbf{G_1} \rightsquigarrow_{[d1]}^{m-1} \Delta^\mu\varrho_\mu\beta \mid \psi\beta \wedge \bigwedge_{j=1}^m(\phi_{l_j}^\circ \wedge \phi_{r_j}^\circ)\beta_j^m \mid V, (\mu\beta)_V$, $E_0 \vdash (\psi\beta \wedge \bigwedge_{j=1}^m(\phi_{l_j}^\circ \wedge \phi_{r_j}^\circ)\beta_j^m)\rho\gamma$, and $(\Delta_2^m \wedge \Delta^\mu\varrho_\mu)\beta_1\delta_1 =_E (\Delta_2^m \wedge \Delta^\mu\varrho_\mu)\beta\rho\gamma$ (†††).

As $dom(\beta_1) \subseteq dom(\delta^\circ) = dom(\alpha) \cup \hat{x}_1 \cup \hat{y}_1$, $dom(\alpha^\circ) = dom(\alpha) \cup V_{\hat{x},\hat{y}}$, so $dom(\beta_1) \cup dom(\delta^\circ) \subseteq dom(\alpha^\circ)$, $dom(\delta_1^\circ) = dom(\beta_1\delta_1) \cup (V_{\hat{x},\hat{y}} \setminus (\hat{x}_1 \cup \hat{y}_1))$, $dom(\delta_1) = ran(\beta_1) \cup (dom(\delta^\circ) \setminus dom(\beta_1))$, and $dom(\beta_2^m) \subseteq dom(\delta_1^\circ) \cup \bigcup_{j=2}^{m-1} ran(\beta_j)$, then:
$dom(\beta) = dom(\beta_1\beta_2^m) = dom(\beta_1) \cup dom(\beta_2^m) \subseteq dom(\alpha) \cup \hat{x}_1 \cup \hat{y}_1 \cup dom(\delta_1^\circ) \cup \bigcup_{j=2}^{m-1} ran(\beta_j) = dom(\alpha) \cup \hat{x}_1 \cup \hat{y}_1 \cup dom(\beta_1\delta_1) \cup (V_{\hat{x},\hat{y}} \setminus (\hat{x}_1 \cup \hat{y}_1)) \cup \bigcup_{j=2}^{m-1} ran(\beta_j) = dom(\alpha) \cup V_{\hat{x},\hat{y}} \cup dom(\beta_1\delta_1) \cup \bigcup_{j=2}^{m-1} ran(\beta_j) = dom(\alpha^\circ) \cup dom(\beta_1\delta_1) \cup \bigcup_{j=2}^{m-1} ran(\beta_j) \subseteq (dom(\alpha^\circ) \cup dom(\beta_1) \cup dom(\delta^\circ)) \cup (ran(\beta_1) \cup \bigcup_{j=2}^{m-1} ran(\beta_j)) = dom(\alpha^\circ) \cup \bigcup_{j=1}^{m-1} ran(\beta_j)$ $(*)$.

Let $\alpha^\circ = \delta^\circ(\delta_1^\circ)_{V_{\hat{x},\hat{y}} \setminus (\hat{x}_1 \cup \hat{y}_1)} = \delta^\circ \cup (\delta_1^\circ)_{V_{\hat{x},\hat{y}} \setminus (\hat{x}_1 \cup \hat{y}_1)}$, since δ° is ground and $dom(\delta^\circ) = dom(\alpha) \cup \hat{x}_1 \cup \hat{y}_1$. Then:

(a) as $\mathbf{G} \rightsquigarrow_{[d1]}^1 \mathbf{G_1}$, then:
$\mathbf{G} \rightsquigarrow_{[d1]}^m \Delta^\mu\varrho_\mu\beta \mid \psi\beta \wedge \bigwedge_{j=1}^m(\phi_{l_j}^\circ \wedge \phi_{r_j}^\circ)\beta_j^m \mid V, (\mu\beta)_V$.

(b) as $dom(\delta^\circ) = dom(\alpha) \cup \hat{x}_1 \cup \hat{y}_1$, then:
$dom(\alpha^\circ) = dom(\delta^\circ \cup (\delta_1^\circ)_{V_{\hat{x},\hat{y}} \setminus (\hat{x}_1 \cup \hat{y}_1)}) = dom(\delta^\circ) \cup (V_{\hat{x},\hat{y}} \setminus (\hat{x}_1 \cup \hat{y}_1)) = dom(\alpha) \cup \hat{x}_1 \cup \hat{y}_1 \cup (V_{\hat{x},\hat{y}} \setminus (\hat{x}_1 \cup \hat{y}_1)) = dom(\alpha) \cup V_{\hat{x},\hat{y}}$, i.e., $dom(\alpha^\circ) = dom(\alpha) \cup V_{\hat{x},\hat{y}}$;

(c) as $\alpha =_{E_0} \delta_{dom(\alpha)}^\circ$, $dom(\alpha) \cap V_{\hat{x},\hat{y}} = \emptyset$ and δ° is ground, then:
$\alpha_{dom(\alpha)}^\circ = (\delta^\circ \cup (\delta_1^\circ)_{V_{\hat{x},\hat{y}} \setminus (\hat{x}_1 \cup \hat{y}_1)})_{dom(\alpha)} = \delta_{dom(\alpha)}^\circ =_{E_0} \alpha$, i.e., $\alpha =_{E_0} \alpha_{dom(\alpha)}^\circ$;

(d) as $\delta^\circ =_B (\beta_1\delta_1)_{\setminus ran(\beta_1)}$, $\beta_1\delta_1 =_{E_0} (\delta_1^\circ)_{dom(\beta_1\delta_1)}$, and $dom(\delta_1^\circ) = dom(\beta_1\delta_1) \cup (V_{\hat{x},\hat{y}} \setminus (\hat{x}_1 \cup \hat{y}_1))$, then:
$\alpha^\circ = \delta^\circ \cup (\delta_1^\circ)_{V_{\hat{x},\hat{y}} \setminus (\hat{x}_1 \cup \hat{y}_1)} =_B (\beta_1\delta_1)_{\setminus ran(\beta_1)} \cup (\delta_1^\circ)_{V_{\hat{x},\hat{y}} \setminus (\hat{x}_1 \cup \hat{y}_1)} =_{E_0} (\delta_1^\circ)_{dom(\beta_1\delta_1) \setminus ran(\beta_1)} \cup (\delta_1^\circ)_{V_{\hat{x},\hat{y}} \setminus (\hat{x}_1 \cup \hat{y}_1)} = (\delta_1^\circ)_{\setminus ran(\beta_1)}$, i.e., $\alpha^\circ =_E (\delta_1^\circ)_{\setminus ran(\beta_1)}$, so:

- as $ran(\beta_1) \cap V_G = \emptyset$ and $l_j^\circ\delta_1^\circ =_E r_j^\circ\delta_1^\circ$, for $2 \leq j \leq m$ then $l_j^\circ\alpha^\circ =_E r_j^\circ\alpha^\circ$, for $2 \leq j \leq m$
- as $ran(\beta_1) \cap V_{l_1^\circ, r_1^\circ} = \emptyset$ and $l_1^\circ\beta_1 =_B r_1^\circ\beta_1$, then:
  - $l_1^\circ(\beta_1)_{\setminus ran(\beta_1)} = l_1^\circ\beta_1 =_B r_1^\circ\beta_1 = r_1^\circ(\beta_1)_{\setminus ran(\beta_1)}$,
  - $l_1^\circ(\beta_1\delta_1)_{\setminus ran(\beta_1)} =_B r_1^\circ(\beta_1\delta_1)_{\setminus ran(\beta_1)}$,
  - $l_1^\circ(\delta_1^\circ)_{dom(\beta_1\delta_1) \setminus ran(\beta_1)} =_E r_1^\circ(\delta_1^\circ)_{dom(\beta_1\delta_1) \setminus ran(\beta_1)}$, and
  - $l_1^\circ(\delta_1^\circ)_{\setminus ran(\beta_1)} =_E r_1^\circ(\delta_1^\circ)_{\setminus ran(\beta_1)}$, i.e., $l_1^\circ\alpha^\circ =_E r_1^\circ\alpha^\circ$.

In conclusion, $\bar{l}^\circ\alpha^\circ =_E \bar{r}^\circ\alpha^\circ$;

(e)
- as $dom(\beta_1\delta_1) = ran(\beta_1) \cup dom(\alpha) \cup \hat{x}_1 \cup \hat{y}_1$, then $(\beta_1\delta_1)_{\setminus ran(\beta_1)} = (\beta_1\delta_1)_{dom(\alpha)\cup\hat{x}_1\cup\hat{y}_1}$;
- as $\beta_1\delta_1 =_{E_0} (\delta_1^\circ)_{dom(\beta_1\delta_1)} = (\delta_1^\circ)_{dom(\beta_1)} \cup (\delta_1^\circ)_{dom(\delta_1)}$ then $\delta_1 =_{E_0} (\delta_1^\circ)_{dom(\delta_1)}$;
- as $dom(\beta_1\delta_1) = ran(\beta_1) \cup dom(\alpha) \cup \hat{x}_1 \cup \hat{y}_1$ then $dom(\delta_1) \subseteq ran(\beta_1) \cup dom(\alpha) \cup \hat{x}_1 \cup \hat{y}_1$;
- then, as $\delta^\circ =_B (\beta_1\delta_1)_{\setminus ran(\beta_1)}$ and $dom(\delta_1^\circ) = dom(\alpha^\circ) \cup ran(\beta_1)$:
  $\alpha^\circ = \delta^\circ(\delta_1^\circ)_{V_{\hat{x},\hat{y}}\setminus(\hat{x}_1\cup\hat{y}_1)} = (\delta^\circ(\delta_1^\circ)_{V_{\hat{x},\hat{y}}\setminus(\hat{x}_1\cup\hat{y}_1)})_{dom(\alpha^\circ)} =_B$
  $((\beta_1\delta_1)_{\setminus ran(\beta_1)}(\delta_1^\circ)_{V_{\hat{x},\hat{y}}\setminus(\hat{x}_1\cup\hat{y}_1)})_{dom(\alpha^\circ)} =$
  $((\beta_1\delta_1)_{dom(\alpha)\cup\hat{x}_1\cup\hat{y}_1}(\delta_1^\circ)_{V_{\hat{x},\hat{y}}\setminus(\hat{x}_1\cup\hat{y}_1)})_{dom(\alpha^\circ)} =_{E_0}$
  $((\beta_1(\delta_1^\circ)_{dom(\delta_1)})_{dom(\alpha)\cup\hat{x}_1\cup\hat{y}_1}(\delta_1^\circ)_{V_{\hat{x},\hat{y}}\setminus(\hat{x}_1\cup\hat{y}_1)})_{dom(\alpha^\circ)} =$
  $((\beta_1\delta_1^\circ)_{dom(\alpha)\cup\hat{x}_1\cup\hat{y}_1}(\delta_1^\circ)_{V_{\hat{x},\hat{y}}\setminus(\hat{x}_1\cup\hat{y}_1)})_{dom(\alpha^\circ)} = ((\beta_1\delta_1^\circ)_{dom(\alpha)\cup V_{\hat{x},\hat{y}}})_{dom(\alpha^\circ)} =$
  $(\beta_1\delta_1^\circ)_{dom(\alpha^\circ)} \ll_E (\beta_1(\beta_2^m)_{dom(\delta_1^\circ)})_{dom(\alpha^\circ)} = (\beta_1\beta_2^m)_{dom(\alpha^\circ)} = (\beta_1^m)_{dom(\alpha^\circ)}$.

In conclusion, $\alpha^\circ \ll_E (\beta_1^m)_{dom(\alpha^\circ)}$;

(f) let ρ and γ such that $\alpha^\circ \ll_E (\beta\rho)_{dom(\alpha^\circ)}$ and $\alpha^\circ =_E (\beta\rho)_{dom(\alpha^\circ)} \cdot \gamma$. Then:
- as $\delta_1 =_{E_0} (\delta_1^\circ)_{dom(\delta_1)}$ then $(\beta_1\delta_1)_{dom(\beta_1\delta_1)} =_{E_0} (\beta_1\delta_1^\circ)_{dom(\beta_1\delta_1)}$, hence $(\beta_1\delta_1)_{dom(\alpha)\cup\hat{x}_1\cup\hat{y}_1} =_{E_0} (\beta_1\delta_1^\circ)_{dom(\alpha)\cup\hat{x}_1\cup\hat{y}_1}$;
- then, as $\delta^\circ =_B (\beta_1\delta_1)_{\setminus ran(\beta_1)}$ and $dom(\beta_1\delta_1) = ran(\beta_1) \cup dom(\alpha) \cup \hat{x}_1 \cup \hat{y}_1$:
  $\alpha^\circ = \delta^\circ(\delta_1^\circ)_{V_{\hat{x},\hat{y}}\setminus(\hat{x}_1\cup\hat{y}_1)} =_B (\beta_1\delta_1)_{\setminus ran(\beta_1)}(\delta_1^\circ)_{V_{\hat{x},\hat{y}}\setminus(\hat{x}_1\cup\hat{y}_1)} =$
  $(\beta_1\delta_1)_{dom(\alpha)\cup\hat{x}_1\cup\hat{y}_1}(\delta_1^\circ)_{V_{\hat{x},\hat{y}}\setminus(\hat{x}_1\cup\hat{y}_1)} =_{E_0} (\beta_1\delta_1^\circ)_{dom(\alpha)\cup\hat{x}_1\cup\hat{y}_1}(\delta_1^\circ)_{V_{\hat{x},\hat{y}}\setminus(\hat{x}_1\cup\hat{y}_1)} =$
  $(\beta_1\delta_1^\circ)_{dom(\alpha^\circ)}$, i.e., $(\beta_1\delta_1^\circ)_{dom(\alpha^\circ)} =_E \alpha^\circ$;
- as $dom(\delta_1^\circ) = dom(\alpha^\circ) \cup ran(\beta_1)$ and $(\beta_1\delta_1^\circ)_{dom(\alpha^\circ)} =_E \alpha^\circ \ll_E (\beta\rho)_{dom(\alpha^\circ)} = (\beta_1\beta_2^m\rho)_{dom(\alpha^\circ)}$, then $(\delta_1^\circ)_{dom(\alpha^\circ)\cup ran(\beta_1)} \ll_E (\beta_2^m\rho)_{dom(\alpha^\circ)\cup ran(\beta_1)}$, i.e., $\delta_1^\circ \ll_E (\beta_2^m\rho)_{dom(\delta_1^\circ)}$;
- as $dom(\delta_1^\circ) = dom(\alpha^\circ) \cup ran(\beta_1)$ and $(\beta_1\beta_2^m\rho)_{dom(\alpha^\circ)}\gamma = (\beta\rho)_{dom(\alpha^\circ)}\gamma =_E \alpha^\circ =_E (\beta_1\delta_1^\circ)_{dom(\alpha^\circ)}$, then $(\beta_2^m\rho)_{dom(\alpha^\circ)\cup ran(\beta_1)}\gamma =_E (\delta_1^\circ)_{dom(\alpha^\circ)\cup ran(\beta_1)}$, i.e., $\delta_1^\circ =_E (\beta_2^m\rho)_{dom(\delta_1^\circ)}\gamma$.

In conclusion, as $\delta_1^\circ \ll_E (\beta_2^m\rho)_{dom(\delta_1^\circ)}$ and $\delta_1^\circ =_E (\beta_2^m\rho)_{dom(\delta_1^\circ)}\gamma$ then, by (†††), $E_0 \vdash (\psi\beta \wedge \bigwedge_{j=1}^m (\phi_{l_j}^\circ \wedge \phi_{r_j}^\circ)\beta_j^m)\rho\gamma$;

- Also by (†††), $(\Delta_2^m \wedge \Delta^\mu\varrho_\mu)\beta_1\delta_1 =_E (\Delta_2^m \wedge \Delta^\mu\varrho_\mu)\beta\rho\gamma$, so $\Delta^\mu\varrho_\mu\beta_1\delta_1 =_E \Delta^\mu\varrho_\mu\beta\rho\gamma$;
- As $\alpha =_{E_0} \delta_{dom(\alpha)}^\circ$, $\delta^\circ =_B \beta_1 \cdot \delta_1$, β_1 is a CSU, so $V_\Delta^\mu\varrho_\mu \cap ran(\beta_1) = \emptyset$, and $V_\Delta^\mu\varrho_\mu \subseteq V_G \subseteq dom(\alpha)$, then:
  $\Delta^\mu\varrho_\mu\beta\rho\gamma =_E \Delta^\mu\varrho_\mu\beta_1\delta_1 =_B \Delta^\mu\varrho_\mu\delta^\circ = \Delta^\mu\varrho_\mu\delta_{dom(\alpha)}^\circ =_{E_0} \Delta^\mu\varrho_\mu\alpha$.

So also $\Delta^\mu\varrho_\mu\alpha =_E \Delta^\mu\varrho_\mu\beta\rho\gamma$.

□

Theorem 17 (Weak Completeness of the Calculus for Reachability Goals)**.** *Given an associated rewrite theory $\mathcal{R} = (\Sigma, \mathcal{E}, R)$ closed under B-extensions and a reachability problem $P = \bigwedge_{i=1}^n u_i \to v_i/ST_i \mid \phi \mid V, \mu$, where μ is $R/\mathcal{E}$-normalized, if $\sigma : V \to \mathcal{T}_\Sigma$ is a $R/\mathcal{E}$-normalized solution for P then there exist a formula $\psi \in QF(\mathcal{X}_0)$ and two substitutions, say λ and ρ, such that $\bigwedge_{i=1}^n u_i\mu \to v_i\mu/ST_i^\mu; \texttt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow_\lambda^+ nil \mid \psi \mid V, \nu$, $\sigma =_\mathcal{E} \nu\cdot\rho$, and $\psi\rho$ is satisfiable, where $\nu = (\mu\lambda)_V$.*

Proof. The proof is by induction on the sum $\mathbf{h}$ of the number of nodes in each c.p.t. for the solution σ. No simplification is applied to the reachability formulas that appear in the generated path.

In the following we will make use of the following two facts. For any term t and substitution α it holds that:

1. $pos_\Sigma(t) \subseteq pos_\Sigma(t\alpha)$ because, by definition, the variables of t that α instantiates are located at positions in $pos_\mathcal{X}(t)$, and

2. $top_{\Sigma_0}(t) \subseteq top_{\Sigma_0}(t\alpha)$, because α only may add new top_{Σ_0} positions for non-Σ_0 variables in its domain, but cannot remove any existing position in $top_{\Sigma_0}(t)$.

Let $u = u_1\mu$ and $v = v_1\mu$. In all cases $\sigma = \mu \cdot \sigma'$, for appropriate σ' such that $dom(\sigma') = V^\mu$, $[v_1\sigma]_E \in ST_1^\sigma@[u_1\sigma]_E$, and $E_0 \vdash \phi\sigma$. As σ is ground and R/E-normalized, then σ' has to be also ground and, by Proposition 10, R/E-normalized.

(i) Base step: $\mathbf{h} = 1$.

Then P has the form $u_1 \to v_1/ST_1 \mid \phi \mid V, \mu$, with $V_P = V_{u_1,v_1,\phi} \subseteq V$ and the c.p.t. T for P_0 and σ has the form $\dfrac{\overline{u_1\sigma \to v_1\sigma/ST_1^\sigma} \quad \overline{v_1\sigma \to v_1\sigma/\mathtt{idle}}}{u_1\sigma \to v_1\sigma/ST_1^\sigma;\mathtt{idle}}$.

There are four strategies in the base case: `idle`, $c[\gamma]$, `top`$(c[\gamma])$, and the `match` test.

1. $ST_1 = \mathtt{idle}$.

   $P = u_1 \to v_1/\mathtt{idle} \mid \phi \mid V, \mu$. As, by definition 21, $V_{u_1,v_1,\phi} \subseteq V$ then $V_{u,v,\phi\mu} \subseteq V^\mu = dom(\sigma')$, and as $[v_1\sigma]_E \in \mathtt{idle}@[u_1\sigma]_E$ then, as shown in example 5, $u_1\sigma =_E v_1\sigma$, i.e., $u\sigma' =_E v\sigma'$, all ground terms.

   Let $abstract_{\Sigma_1}((u, v)) = \langle\lambda(\bar{x}, \bar{y}).(u^\circ, v^\circ); (\theta_u^\circ, \theta_v^\circ); (\phi_u^\circ, \phi_v^\circ)\rangle$. As $dom(\sigma') = V^\mu$ then, by Lemma 9, there exists a ground substitution σ° such that $u^\circ\sigma^\circ =_B v^\circ\sigma^\circ$, $E_0 \vdash (\phi_u^\circ \wedge \phi_v^\circ)\sigma^\circ$, $dom(\sigma^\circ) = V^\mu \cup \hat{x} \cup \hat{y}$, and $\sigma' =_{E_0} \sigma^\circ_{V^\mu}$.

   As $u^\circ\sigma^\circ =_B v^\circ\sigma^\circ$, then there exist substitutions ν' and ρ' such that $\nu' \in CSU_B(u^\circ = v^\circ)$ and $\sigma^\circ =_B \nu'\cdot\rho'$, let $\nu = (\mu\nu')_V$ and $\rho = \rho'_{ran(\nu)\cup(V\setminus dom(\nu))}$. As $dom(\mu) \subseteq V$ and $\sigma' =_{E_0} \sigma^\circ_{V^\mu}$ then:

   $\sigma = \mu\sigma' =_{E_0} \mu\sigma^\circ_{V^\mu} =_B \mu(\nu'\rho')_{V^\mu} = (\mu\nu'\rho')_V = (\mu\nu')_V\cdot\rho'_{ran((\mu\nu')_V)\cup(V\setminus dom((\mu\nu')_V))} = \nu\cdot\rho'_{ran(\nu)\cup(V\setminus dom(\nu))} = \nu\cdot\rho$, i.e., $\sigma =_E \nu\cdot\rho$.

   As $E_0 \vdash \phi\sigma$, $V_{\phi\mu} \subseteq V^\mu$, and $\sigma' =_{E_0} \sigma^\circ_{V^\mu}$ then $\phi\mu\sigma^\circ =_{E_0} \phi\mu\sigma' = \phi\sigma$, so $E_0 \vdash \phi\mu\sigma^\circ$. Now, as $E_0 \vdash (\phi_u^\circ \wedge \phi_v^\circ)\sigma^\circ$, then $E_0 \vdash (\phi\mu \wedge \phi_u^\circ \wedge \phi_v^\circ)\sigma^\circ$, let $\psi^\circ = \phi\mu \wedge \phi_u^\circ \wedge \phi_v^\circ$ and $\psi = \psi^\circ\nu'$. As $E_0 \vdash \psi^\circ\sigma^\circ$, $\sigma^\circ =_B \nu'\cdot\rho'$, and $V_{\psi^\circ} \cap ran(\nu') = \emptyset$, so $\psi^\circ\sigma^\circ = \psi^\circ\nu'\rho'$, and ρ is more general than ρ', then $\psi^\circ\nu'\rho$, i.e., $\psi\rho$, is satisfiable, hence ψ is also satisfiable.

   As $u = u_1\mu$, $v = v_1\mu$, and $\nu' \in CSU_B(u^\circ = v^\circ)$, then $u \to v/\mathtt{idle};\mathtt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow_{[d2]} u \to v/\mathtt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow_{[d1],\nu'} nil \mid \psi \mid V, \nu$, where ψ is satisfiable and $\sigma =_E \nu\rho$.

2. $ST_1 = c[\gamma]$.

   $P = u_1 \to v_1/c[\gamma] \mid \phi \mid V, \mu$, with $c : l \to rif\chi \in R$, and $[v_1\sigma]_E \in c^\sigma[\gamma\sigma_{ran(\gamma)}]@[u_1\sigma]_E$. Then, by Lemma 12 point 3, $u_1\sigma \xrightarrow[c^\sigma\gamma\sigma_{ran(\gamma)}]{}{}^1{}_{R^\sigma/E} v_1\sigma$, so $E_0 \vdash \chi\sigma\gamma\sigma_{ran(\gamma)}$. Let $c' = c^\sigma\gamma\sigma_{ran(\gamma)}$ ($= c^{\gamma\sigma}$ because σ is ground and, by definition, $dom(\gamma) \cap dom(\sigma) = \emptyset$,

hence $E_0 \vdash \chi\gamma\sigma$), $\mathcal{R}(c') = (\Sigma, E_0 \cup B, \{c'\})$, and $\mathcal{R}_B(c') = (\Sigma, E_0 \cup B, c'_B)$. Then also $u_1\sigma \underset{c'\ \{c'\}/E}{\rightarrow^1} v_1\sigma$ so, by Theorem 15, $u_1\sigma \rightarrow^1_{\{c'\},B} v_1\sigma$.

As $u_1\sigma \rightarrow^1_{\{c'\},B} v_1\sigma$ and $vars(B) \cap vars(c\gamma) = \emptyset$, then this rewrite step uses a rule $c'_1 \in c'_B$ where:

- if $c'_1 = c'$ then c'_1 has the form $c'_1 : l\gamma\sigma \rightarrow r\gamma\sigma$ *if* $\chi\gamma\sigma$, let $l_0 = l$ and $r_0 = r$, and
- if $c'_1 \neq c'$ then c'_1 has the form $c'_1 : w[l\gamma\sigma]_{p'} \rightarrow w[r\gamma\sigma]_{p'}$ *if* $\chi\gamma\sigma$, by Definition 6.2, for appropriate w and p'. As by Definition 6.2, $V_w \cap V_{c'} = \emptyset$, by Definition 21, $V_w \cap V = \emptyset$, and also $dom(\gamma) \subseteq V_{c'}$ and $dom(\sigma) \subseteq V$, this is the same as $c'_1 : w[l]_{p'}\gamma\sigma \rightarrow w[r]_{p'}\gamma\sigma$ *if* $\chi\gamma\sigma$, let $l_0 = w[l]_{p'}$ and $r_0 = w[r]_{p'}$.

In either case, c'_1 has the form $c'_1 : l_0\gamma\sigma \rightarrow r_0\gamma\sigma$ *if* $\chi\gamma\sigma$. Let $c_0 : l_0 \rightarrow r_0$ *if* χ. As $c'_1 \in c'^{\gamma\sigma}_B$ and $c'_1 = c_0^{\gamma\sigma}$ then, by Proposition 6.5.2, $c_0 \in c_B$. Since $\sigma = \mu\sigma'$, if we let $l_1 = l_0\gamma\mu$ and $r_1 = r_0\gamma\mu$ then c'_1 has also the form $c'_1 : l_1\sigma' \rightarrow r_1\sigma'$ *if* $\chi\gamma\sigma$.
Let $c_2 : l_2 \rightarrow r_2$ *if* χ_2 be a fresh version of c_0^μ except for $dom(\gamma) \cup V^\mu$ $(= dom(\gamma) \cup dom(\sigma'))$, and let τ be the renaming that verifies $c_2 = c_0^\mu\tau$, so $(l_2, r_2, \chi_2) = (l_0, r_0, \chi)(\mu \uplus \tau)$, where $(dom(\tau) \cup ran(\tau)) \cap (dom(\gamma) \cup V^\mu) = \emptyset$. Then $l_2(\gamma\mu)_{dom(\gamma)} = l_0(\mu \uplus \tau)(\gamma\mu)_{dom(\gamma)} = l_0((\gamma\mu)_{dom(\gamma)} \uplus \mu \uplus \tau) = l_0((\gamma\mu)_{dom(\gamma)} \uplus \mu)\tau = l_0\gamma\mu\tau = l_1\tau$, so also $r_2(\gamma\mu)_{dom(\gamma)} = r_1\tau$ and $\chi_2(\gamma\mu)_{dom(\gamma)} = \chi\gamma\mu\tau$. Let $l_c = l_2(\gamma\mu)_{dom(\gamma)}$ and $\sigma'' = \tau^{-1}\sigma'$. Then $l_c\sigma'' = l_1\tau\tau^{-1}\sigma' = l_1\sigma'$. Now:

(a) $abstract_{\Sigma_1}(l_c) = \langle \lambda\bar{y}.l^\circ; \theta^\circ_l; \phi^\circ_l \rangle$, where $\bar{y} = y_1, \ldots, y_{i_y}$, $l^\circ = l_c[\bar{y}]_{\bar{p}}$, $\bar{p} = p_1, \ldots, p_{i_y}$, $\hat{p} = top_{\Sigma_0}(l_c)$, $\theta^\circ_l = \bigcup_{i=1}^{i_y}\{y_i \mapsto l_c|_{p_i}\}$, and $\phi^\circ_l = \bigwedge_{i=1}^{i_y} y_i = l_c|_{p_i}$;

(b) since $l_1\sigma' = l_c\sigma''$ and $top_{\Sigma_0}(l_c) \subseteq top_{\Sigma_0}(l_c\sigma'')$ then $abstract_{\Sigma_1}(l_1\sigma') = abstract_{\Sigma_1}(l_c\sigma'') = \langle \lambda\bar{y}\bar{z}.l^\circ_{c\sigma''}; \theta^\circ_{c\sigma''}; \phi^\circ_{c\sigma''} \rangle$, where $\bar{z} = z_1, \ldots, z_{i_z}$, $l^\circ_{c\sigma''} = l_c\sigma''[\bar{y}]_{\bar{p}}[\bar{z}]_{\bar{q}}$, $\hat{q} = top_{\Sigma_0}(l_c\sigma'') \setminus top_{\Sigma_0}(l_c)$, $\theta^\circ_{c\sigma''} = \bigcup_{i=1}^{i_y}\{y_i \mapsto l_c|_{p_i}\sigma''\} \cup \bigcup_{j=1}^{i_z}\{z_j \mapsto l_c\sigma''|_{q_j}\}$, and $\phi^\circ_{c\sigma''} = (\bigwedge_{i=1}^{i_y} y_i = l_c|_{p_i}\sigma'' \wedge \bigwedge_{j=1}^{i_z} z_j = l_c\sigma''|_{q_j})$;

(c) as $u_1\sigma \rightarrow^1_{\{c'\},B} v_1\sigma$ with c'_1, then there are a position p in $pos_{\Sigma_1}(u_1\sigma)$ and a substitution $\delta : \hat{y} \cup \hat{z} \cup V_{c'_1} \rightarrow \mathcal{T}_\Sigma$ such that $rep(u_1\sigma|_p) =_B l^\circ_{c\sigma''}\delta$, $v_1\sigma =_E u_1\sigma[r_0\gamma\sigma\delta]_p = u_1\sigma[r_0\gamma\mu\sigma'\delta]_p = u_1\sigma[r_1\sigma'\delta]_p$, and $E_0 \vdash (\chi\gamma\sigma \wedge \phi^\circ_{c\sigma''})\delta$, so $E_0 \vdash \chi\gamma\sigma\delta$, i.e., $E_0 \vdash \chi\gamma\mu\sigma'\delta$, $\bar{y}\delta =_{E_0} l_c|_{\bar{p}}\sigma''\delta$ and $\bar{z}\delta =_{E_0} l_c\sigma''|_{\bar{q}}\delta$;

(d) as $p \in pos_{\Sigma_1}(u_1\sigma)$ and σ is R/E-normalized, hence R, E-normalized by Theorem 15, then $p \in pos_{\Sigma_1}(u_1)$, so $u_1\sigma|_p = u_1|_p\sigma = u_1|_p\mu\sigma' = u_1\mu|_p\sigma' = u|_p\sigma'$; and

(e) as τ is a fresh renaming then $\emptyset = V_u \cap ran(\tau) = V_u \cap dom(\tau^{-1})$, so $u|_p\tau^{-1}\sigma' = u|_p\sigma' = u_1\sigma|_p =_{E_0} rep(u_1\sigma|_p) =_B l^\circ_{c\sigma''}\delta = l_c\sigma''[\bar{y}]_{\bar{p}}[\bar{z}]_{\bar{q}}\delta =_{E_0} l_c\sigma'' = l_c\tau^{-1}\sigma'$, i.e., $u|_p\tau^{-1}\sigma' =_E l_c\tau^{-1}\sigma'$.

Let $abstract_{\Sigma_1}(u|_p) = \langle \lambda\bar{x}.u^\circ; \theta^\circ_u; \phi^\circ_u \rangle$. As $dom(\tau^{-1}\sigma') = ran(\tau) \cup V^\mu$ then, by Lemma 9, there exists a ground substitution σ° such that $u^\circ\sigma^\circ =_B l^\circ\sigma^\circ$, $E_0 \vdash (\phi^\circ_u \wedge \phi^\circ_l)\sigma^\circ$, $dom(\sigma^\circ) = dom(\tau^{-1}\sigma') \cup \hat{x} \cup \hat{y} = ran(\tau) \cup V^\mu \cup \hat{x} \cup \hat{y}$, and $\tau^{-1}\sigma' =_{E_0} \sigma^\circ_{dom(\tau^{-1}\sigma')} = \sigma^\circ_{ran(\tau)\cup V^\mu}$, so $(\tau^{-1}\sigma')_{V^\mu} =_{E_0} \sigma^\circ_{V^\mu}$. As $(dom(\tau) \cup ran(\tau)) \cap V^\mu = \emptyset$ and $dom(\sigma') = V^\mu$ then $\sigma' = \sigma'_{V^\mu} = (\tau^{-1}\sigma')_{V^\mu} =_{E_0} \sigma^\circ_{V^\mu}$.

As $u^\circ\sigma^\circ =_B l^\circ\sigma^\circ$, then there exist substitutions ν' and ρ' such that $\nu' \in CSU_B(u^\circ = l^\circ)$ and $\sigma^\circ =_B \nu'\rho'$, let $\nu = (\mu\nu')_V$ and $\rho = \rho'_{ran(\nu)\cup(V \setminus dom(\nu))}$. As $dom(\mu) \subseteq V$ and $\sigma' =_{E_0} \sigma^\circ_{V^\mu}$ then:

$\sigma = \mu\sigma' =_{E_0} \mu\sigma^\circ_{V^\mu} =_B \mu(\nu'\rho')_{V^\mu} = (\mu\nu'\rho')_V = (\mu\nu')_V\rho'_{ran((\mu\nu')_V)\cup(V\setminus dom((\mu\nu')_V))} = \nu\rho'_{ran(\nu)\cup(V\setminus dom(\nu))} = \nu\rho$, i.e., $\sigma =_E \nu\rho$.

As $\chi_2(\gamma\mu)_{dom(\gamma)} = \chi\gamma\mu\tau$, $dom(\sigma^\circ) = ran(\tau)\cup V^\mu\cup\hat{x}\cup\hat{y}$, and $\tau^{-1}\sigma' =_{E_0} \sigma^\circ_{ran(\tau)\cup V^\mu}$, then $\chi_2(\gamma\mu)_{dom(\gamma)}\sigma^\circ\delta = \chi\gamma\mu\tau\sigma^\circ_{ran(\tau)\cup V^\mu}\delta =_{E_0} \chi\gamma\mu\tau\tau^{-1}\sigma'\delta = \chi\gamma\mu\sigma'\delta$ so, as $E_0 \vdash \chi\gamma\mu\sigma'\delta$, $E_0 \vdash \chi_2(\gamma\mu)_{dom(\gamma)}\sigma^\circ\delta$.

As $E_0 \vdash \phi\sigma$, $V_{\phi\mu} \subseteq V^\mu$, and $\sigma' =_{E_0} \sigma^\circ_{V^\mu}$ then $\phi\mu\sigma^\circ =_{E_0} \phi\mu\sigma' = \phi\sigma$, so $E_0 \vdash \phi\mu\sigma^\circ$. Now, as $E_0 \vdash (\phi^\circ_u \wedge \phi^\circ_l)\sigma^\circ$, then $E_0 \vdash (\phi\mu \wedge \phi^\circ_u \wedge \phi^\circ_l)\sigma^\circ$ ground formula, so $E_0 \vdash (\phi\mu \wedge \phi^\circ_u \wedge \phi^\circ_l)\sigma^\circ\delta$ and $E_0 \vdash (\phi\mu \wedge \phi^\circ_u \wedge \phi^\circ_l \wedge \chi_2(\gamma\mu)_{dom(\gamma)})\sigma^\circ\delta$. Let $\varphi^\circ = \phi\mu \wedge \phi^\circ_u \wedge \phi^\circ_l \wedge \chi_2(\gamma\mu)_{dom(\gamma)}$, and let $\varphi = \varphi^\circ\nu'$. As $\sigma^\circ =_B \nu'\rho'$, so $\varphi^\circ\sigma^\circ = \varphi^\circ\nu'\rho' = \varphi\rho'$, then $E_0 \vdash \varphi\rho'\delta$, let $\delta' = \rho'\delta$, hence φ is also satisfiable.

$\mathbf{G_0} = u \to v/c^\mu[(\gamma\mu)_{dom(\gamma)}]; \texttt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow_{[t]} u \to^1 x_0, x_0 \to v/c^\mu[(\gamma\mu)_{dom(\gamma)}]; \texttt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow^*_{[c]} u|_p \to^1 x, u[x]_p \to v/c^\mu[(\gamma\mu)_{dom(\gamma)}]; \texttt{idle} \mid \phi\mu \mid V, \mu = \mathbf{G_1}$, where $u|_p$ cannot be a variable, say x_u, because as $p \in pos_\Sigma(u_1)$ then, by (c), also $x_u\sigma' \to^1_{R,B} r_0\gamma\sigma\delta$, so σ would not be R/E-normalized. As $c_2 : l_2 \to r_2$ *if* χ_2, where $r_2(\gamma\mu)_{dom(\gamma)} = r_1\tau$, and $\nu' \in CSU_B(u^\circ = l^\circ)$ then $\mathbf{G_1} \rightsquigarrow_{[r],\nu'\cup\{x\mapsto r_1\tau\nu'\}} (u[r_1\tau]_p \to v\,/\texttt{idle})\nu' \mid \varphi \mid V, \nu = \mathbf{G_2}$.

We already know that $E_0 \vdash \varphi\delta'$. We prove that $u[r_1\tau]_p\nu'\delta' =_E v\nu'\delta'$:

- as $\tau^{-1}\sigma' =_{E_0} \sigma^\circ_{dom(\tau^{-1}\sigma')}$ and $dom(\sigma^\circ) = dom(\tau^{-1}\sigma')\cup\hat{x}\cup\hat{y}$, then $\tau^{-1}\sigma' \uplus \sigma^\circ_{\hat{x}\cup\hat{y}} =_{E_0} \sigma^\circ_{dom(\tau^{-1}\sigma')} \uplus \sigma^\circ_{\hat{x}\cup\hat{y}} = \sigma^\circ$, where $V_{G_2} \cap (\hat{x}\cup\hat{y}) = \emptyset$, $u = u\tau^{-1}$, and $v = v\tau^{-1}$;
- $u[r_1\tau]_p\nu'\delta' =_B u[r_1\tau]_p\sigma^\circ\delta =_{E_0} u[r_1\tau]_p(\tau^{-1}\sigma'\uplus\sigma^\circ_{\hat{x}\cup\hat{y}})\delta = u[r_1\tau]_p\tau^{-1}\sigma'\delta = u[r_1]_p\sigma'\delta$;
- $v\nu'\delta' =_B v\sigma^\circ\delta =_{E_0} v(\tau^{-1}\sigma' \uplus \sigma^\circ_{\hat{x}\cup\hat{y}})\delta = v\tau^{-1}\sigma'\delta = v\sigma'\delta$;
- by (c), $v_1\sigma =_E u_1\sigma[r_1\sigma'\delta]_p$, i.e., $v\sigma' =_E u\sigma'[r_1\sigma'\delta]_p$, ground expression so, as δ is ground, $v\sigma'\delta =_E u\sigma'\delta[r_1\sigma'\delta]_p = u[r_1]_p\sigma'\delta$, hence $u[r_1\tau]_p\nu'\delta' =_E v\nu'\delta'$.

Let $abstract_{\Sigma_1}((u[r_1\tau]_p, v\nu')) = \langle\lambda(\bar{x}', \bar{y}').(r^\circ, v^\circ); (\theta^\circ_r, \theta^\circ_v); (\phi^\circ_r, \phi^\circ_v)\rangle$. By Lemma 9, there exists a ground substitution δ° such that $r^\circ\delta^\circ =_B v^\circ\delta^\circ$, $E_0 \vdash (\phi^\circ_r \wedge \phi^\circ_v)\delta^\circ$, $dom(\delta^\circ) = dom(\delta')\cup\hat{x}'\cup\hat{y}'$, and $\delta' =_{E_0} \delta^\circ_{dom(\delta')}$, so there exist substitutions ν'' and ρ'' such that $\nu'' \in CSU_B(r^\circ = v^\circ)$ and $\delta^\circ =_B \nu''\rho''$, let $\nu_1 = (\nu'\nu'')_V$ and $\rho_1 = \rho''_{ran(\nu_1)\cup(V\setminus dom(\nu_1))}$.

As $E_0 \vdash \varphi\delta'$, ground formula, and $\delta' =_{E_0} \delta^\circ_{dom(\delta')}$ then $E_0 \vdash \varphi\delta^\circ$ so $E_0 \vdash (\varphi \wedge \phi^\circ_r \wedge \phi^\circ_v)\delta^\circ$, let $\psi' = \varphi \wedge \phi^\circ_r \wedge \phi^\circ_v$. Now, as $\delta^\circ =_B \nu''\rho''$ implies $\psi'\delta^\circ = \psi'\nu''\rho''$, then also $E_0 \vdash \psi'\nu''\rho''$, let $\psi = \psi'\nu''$, so ψ and $\psi\rho_1$ are satisfiable.

As $\nu'' \in CSU_B(r^\circ = v^\circ)$ and ψ is satisfiable, then $\mathbf{G_2} \rightsquigarrow_{[d1],\nu''} nil \mid \psi \mid V, \nu_1$, where $\nu_1 = (\nu\nu'')_V$. Then, as $\psi\rho_1$ is satisfiable, all that is left to prove is $\sigma =_E \nu_1\rho_1$.

As $dom(\delta^\circ) = dom(\delta')\cup\hat{x}'\cup\hat{y}'$ and $\delta' =_{E_0} \delta^\circ_{dom(\delta')}$ then $dom(\delta^\circ)\cap V = dom(\delta')$ and $\delta^\circ_V =_{E_0} \delta'_V$, so, as $dom(\sigma) = V$ and σ $(=_B \mu(\nu'\rho')_{V^\mu})$ is ground, let $\sigma^B = \mu(\nu'\rho')_{V^\mu}$, then $\nu_1\rho_1 = (\nu\nu'')_V\rho''_{ran(\nu_1)\cup(V\setminus dom(\nu_1))} = (\nu\nu''\rho'')_V =_B (\nu\delta^\circ)_V =_{E_0} (\nu\delta')_V = (\mu\nu'\delta')_V = (\mu\nu'\rho'\delta)_V = \mu(\nu'\rho'\delta)_{V^\mu} = \sigma^B\delta_{ran(\sigma^B)\cup(V\setminus dom(\sigma^B))} =_B \sigma\delta_{ran(\sigma)\cup(V\setminus dom(\sigma))} = \sigma\delta_\emptyset = \sigma$, i.e., $\sigma =_E \nu_1\rho_1$.

3. $ST_1 = \mathtt{top}(c[\gamma])$.

   The proof is almost exactly the same as the previous one, particularized for the case $p = \epsilon$. The only difference is found in the initial narrowing steps, where instead of:

   - $\mathbf{G_0} = u \to v/c^{\mu}[(\gamma\mu)_{dom(\gamma)}]; \mathtt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow_{[t]} u \to^1 x_0, x_0 \to v/c^{\mu}[(\gamma\mu)_{dom(\gamma)}]; \mathtt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow^*_{[c]} u|_p \to^1 x, u[x]_p \to v/c^{\mu}[(\gamma\mu)_{dom(\gamma)}]; \mathtt{idle} \mid \phi\mu \mid V, \mu = \mathbf{G_1}$ and
   - $\mathbf{G_1} \rightsquigarrow_{[r],\nu' \cup \{x_0 \mapsto r_1\tau\nu'\}} (u[r_1\tau]_p \to v \,/\mathtt{idle})\nu' \mid \varphi \mid V, \nu = \mathbf{G_2}$,

   now we have:

   $\mathbf{G_0} = u \to v/\mathtt{top}(c^{\mu}[(\gamma\mu)_{dom(\gamma)}]); \mathtt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow_{[tp],\nu'} (r_1\tau \to v \,/\mathtt{idle})\nu' \mid \varphi \mid V, \nu = \mathbf{G_2}$.

4. $ST_1 = \mathtt{match}\, t\, \mathtt{s.t.}\, \bigwedge_{j=1}^{m}(l_j = r_j) \wedge \chi$.

   $P = u_1 \to v_1/ST_1 \mid \phi \mid V, \mu$, $V_P = V_{\bar{u}_1,\bar{v}_1,\phi} \subseteq V$, $ST_1^{\sigma} = \mathtt{match}\, t\sigma\, \mathtt{s.t.}\, \bigwedge_{j=1}^{m}(l_j\sigma = r_j\sigma) \wedge \chi\sigma$, and there exists a substitution $\delta : V_{ST_1^{\sigma}} \to \mathcal{T}_{\Sigma}$, such that $v_1\sigma =_E u_1\sigma =_E t\sigma\delta$, $l_j\sigma\delta =_E r_j\sigma\delta$, for $1 \leq j \leq m$, and $E_0 \vdash (\phi \wedge \chi)\mu\sigma'\delta$.

   Let $abstract_{\Sigma_1}((u, t\mu)) = \langle \lambda(\bar{x}, \bar{y}).(u^{\circ}, t^{\circ}); (\theta_u^{\circ}, \theta_t^{\circ}); (\phi_u^{\circ}, \phi_t^{\circ})\rangle$. As $u_1\sigma$ is ground then $u\sigma'\delta = u_1\mu\sigma'\delta = u_1\sigma\delta = u_1\sigma =_E t\sigma\delta = t\mu\sigma'\delta$ so, by Lemma 9, there exists a ground substitution σ° such that $u^{\circ}\sigma^{\circ} =_B t^{\circ}\sigma^{\circ}$, $E_0 \vdash (\phi_u^{\circ} \wedge \phi_t^{\circ})\sigma^{\circ}$, $dom(\sigma^{\circ}) = dom(\sigma'\delta) \cup \hat{x} \cup \hat{y}$, and $\sigma'\delta =_{E_0} \sigma^{\circ}_{dom(\sigma'\delta)}$.

   Let $\psi_1 = (\phi \wedge \chi)\mu \wedge \phi_u^{\circ} \wedge \phi_t^{\circ}$. As $E_0 \vdash (\phi \wedge \chi)\mu\sigma'\delta$, $V_{(\phi\wedge\chi)\mu\sigma'\delta} \cap (\hat{x} \cup \hat{y}) = \emptyset$, and $\sigma'\delta =_{E_0} \sigma^{\circ}_{dom(\sigma'\delta)} = \sigma^{\circ}_{\setminus(\hat{x}\cup\hat{y})}$, then $E_0 \vdash (\phi \wedge \chi)\mu\sigma^{\circ}$, so $E_0 \vdash \psi_1\sigma^{\circ}$.

   As $u^{\circ}\sigma^{\circ} =_B t^{\circ}\sigma^{\circ}$, then there exist substitutions ν and τ such that $\eta \in CSU_B(u^{\circ} = t^{\circ})$ and $\sigma^{\circ} =_B \eta{\cdot}\tau$, so $\psi_1\sigma^{\circ} = \psi_1\eta\tau$, hence $E_0 \vdash \psi_1\eta\tau$ and $\psi_1\eta$ is satisfiable.

   Now, $\mathbf{G_0} = u \to v/ST_1^{\mu}; \mathtt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow_{[m],\eta} (\bigwedge_{j=1}^{m}(l_j \to r_j/\mathtt{idle}) \wedge u_1 \to v_1 \,/\mathtt{idle})\mu\eta \mid \psi_1\eta \mid V, (\mu\eta)_V = \mathbf{G_1}$.

   As $\bar{l}\sigma\delta =_E \bar{r}\sigma\delta$, $\sigma'\delta =_{E_0} \sigma^{\circ}_{dom(\sigma'\delta)}$, $\sigma'\delta =_{E_0} \sigma^{\circ}_{\setminus(\hat{x}\cup\hat{y})}$, $\sigma^{\circ} =_B \eta{\cdot}\tau$, and $V_{\hat{l}\mu,\hat{r}\mu} \cap (\hat{x} \cup \hat{y} \cup ran(\eta)) = \emptyset$, then $(\bar{l}, \bar{r})\mu\eta\tau = (\bar{l}, \bar{r})\mu(\eta \cdot \tau) =_B (\bar{l}, \bar{r})\mu\sigma^{\circ} = (\bar{l}, \bar{r})\mu\sigma^{\circ}_{\setminus(\hat{x}\cup\hat{y})} =_{E_0} (\bar{l}, \bar{r})\mu\sigma'\delta$, i.e., $(\bar{l}, \bar{r})\mu\eta\tau =_E (\bar{l}, \bar{r})\sigma\delta$, so $\bar{l}\mu\eta\tau =_E \bar{r}\mu\eta\tau$.

   By Lemma 14, as τ is a substitution such that $E_0 \vdash \psi_1\eta\tau$ and $\bar{l}\mu\eta\tau =_E \bar{r}\mu\eta\tau$, then there exist a ground substitution τ°, substitutions $\beta_1, \ldots, \beta_m$, let $\beta = \beta_1^m$, and abstractions $abstract_{\Sigma_1}((l_j\beta_1^{j-1}, r_j\beta_1^{j-1})) = \langle \lambda(\bar{x}_j, \bar{y}_j).(l_j^{\circ}, r_j^{\circ}); (\theta_{l_j}^{\circ}, \theta_{r_j}^{\circ}); (\phi_{l_j}^{\circ}, \phi_{r_j}^{\circ})\rangle$, for $1 \leq j \leq m$, such that $dom(\tau^{\circ}) = dom(\tau) \cup V_{\hat{x},\hat{y}}$, $\tau =_{E_0} \tau^{\circ}_{dom(\tau)}$, $\bar{l}^{\circ}\tau^{\circ} =_E \bar{r}^{\circ}\tau^{\circ}$, $\tau^{\circ} \ll_E \beta_{dom(\tau^{\circ})}$, let $\psi_2 = \psi_1\eta\beta \wedge \bigwedge_{j=1}^{m}(\phi_{l_j}^{\circ} \wedge \phi_{r_j}^{\circ})\beta_j^m$, $\mathbf{G_1} \rightsquigarrow^m_{[d1]} (u_1 \to v_1 \,/\mathtt{idle})\mu\eta\beta \mid \psi_2 \mid V, (\mu\eta\beta)_V = \mathbf{G_2}$, and for every pair of substitutions ρ and γ such that $\tau^{\circ} \ll_E (\beta\rho)_{dom(\tau^{\circ})}$ and $\tau^{\circ} =_E (\beta\rho)_{dom(\tau^{\circ})} \cdot \gamma$ it holds that $E_0 \vdash \psi_2\rho\gamma$ and $(u_1 \to v_1 \,/\mathtt{idle})\mu\eta\tau =_E (u_1 \to v_1 \,/\mathtt{idle})\mu\eta\beta\rho\gamma$ (†).

   Take $\rho = none$. As $\tau^{\circ} \ll_E \beta_{dom(\tau^{\circ})}$, then there exists γ such that $\tau^{\circ} =_E \beta_{dom(\tau^{\circ})} \cdot \gamma$ and $ran(\tau^{\circ}) = ran(\beta_{dom(\tau^{\circ})} \cdot \gamma)$, so as τ° is ground then γ is ground. By (†), $E_0 \vdash \psi_2\gamma$ and $(u_1 \to v_1 \,/\mathtt{idle})\mu\eta\tau =_E (u_1 \to v_1 \,/\mathtt{idle})\mu\eta\beta\gamma$. Now, as $V_{u_1,v_1} \subseteq V = dom(\sigma)$ and σ is ground, then $V\sigma = V\sigma\delta = V\mu\sigma'\delta =_{E_0} V\mu\sigma^{\circ} =_B V\mu\eta\tau =_E V\mu\eta\beta\gamma$ so, as $u_1\sigma =_E v_1\sigma$, also $u_1\mu\eta\beta\gamma =_E v_1\mu\eta\beta\gamma$, ground Σ-equation, hence $V_{(u_1,v_1)\mu\eta\beta} \subseteq dom(\gamma)$.

   Let $abstract_{\Sigma_1}((u_1\mu\eta\beta, v_1\mu\eta\beta)) = \langle \lambda(\bar{x}', \bar{y}').(\mathfrak{u}^{\circ}, \mathfrak{v}^{\circ}); (\theta_{\mathfrak{u}}^{\circ}, \theta_{\mathfrak{v}}^{\circ}); (\phi_{\mathfrak{u}}^{\circ}, \phi_{\mathfrak{v}}^{\circ})\rangle$. As $u_1\mu\eta\beta\gamma =_E v_1\mu\eta\beta\gamma$, $V_{(u_1,v_1)\mu\eta\beta} \subseteq dom(\gamma)$, and γ is ground then, by Lemma 9, there exists a

ground substitution γ° such that $\mathfrak{u}^\circ\gamma^\circ =_B \mathfrak{v}^\circ\gamma^\circ$, $E_0 \vdash (\phi^\circ_{\mathfrak{u}} \wedge \phi^\circ_{\mathfrak{v}})\gamma^\circ$, $dom(\gamma^\circ) = dom(\gamma) \cup \hat{x}' \cup \hat{y}'$, and $\gamma =_{E_0} \gamma^\circ_{dom(\gamma)}$.

As $\mathfrak{u}^\circ\gamma^\circ =_B \mathfrak{v}^\circ\gamma^\circ$, then there exist substitutions α and ε such that $\alpha \in CSU_B(u^\circ = v^\circ)$ and $\gamma^\circ =_B \alpha \cdot \varepsilon$. Now, as $\tau^\circ =_E \beta_{dom(\tau^\circ)} \cdot \gamma =_{E_0} \beta_{dom(\tau^\circ)} \cdot \gamma^\circ_{dom(\gamma)} =_B \beta_{dom(\tau^\circ)} \cdot (\alpha \cdot \varepsilon)_{dom(\gamma)}$ and τ° is ground, then $\tau^\circ =_E (\beta\alpha\varepsilon)_{dom(\tau^\circ)}$, so $\tau^\circ =_E (\beta\alpha)_{dom(\tau^\circ)} \cdot \varepsilon$, hence $\tau^\circ \ll_E (\beta\alpha)_{dom(\tau^\circ)}$ and, by (†), $E_0 \vdash \psi_2\alpha\varepsilon$.

Let $\psi = (\psi_2 \wedge \phi^\circ_{\mathfrak{u}} \wedge \phi^\circ_{\mathfrak{v}})\alpha$, ground formula, and $\nu = (\mu\eta\beta\alpha)_V$. As $E_0 \vdash (\phi^\circ_{\mathfrak{u}} \wedge \phi^\circ_{\mathfrak{v}})\gamma^\circ$ and $\gamma^\circ =_B \alpha \cdot \varepsilon$ then $E_0 \vdash (\phi^\circ_{\mathfrak{u}} \wedge \phi^\circ_{\mathfrak{v}})\alpha \cdot \varepsilon$, so also $E_0 \vdash (\phi^\circ_{\mathfrak{u}} \wedge \phi^\circ_{\mathfrak{v}})\alpha\varepsilon$, hence, as $E_0 \vdash \psi_2\alpha\varepsilon$, $E_0 \vdash \psi\varepsilon$. Finally:

- as $E_0 \vdash \psi\varepsilon$ then ψ is satisfiable, so $\mathbf{G_2} \rightsquigarrow_{[d1],\alpha} nil \mid \psi \mid V, \nu$, i.e., $\mathbf{G_0} \rightsquigarrow^+ nil \mid \psi \mid V, \nu$,
- as $E_0 \vdash \psi\varepsilon$ then $\psi\varepsilon$ is satisfiable, and
- as $V\sigma =_E V\mu\eta\beta\gamma =_{E_0} V\mu\eta\beta\gamma^\circ_{dom(\gamma)} =_B V\mu\eta\beta(\alpha \cdot \varepsilon)_{dom(\gamma)}$ then: $\sigma = \sigma_V =_E (\mu\eta\beta(\alpha\cdot\varepsilon)_{dom(\gamma)})_V = (\mu\eta\beta\alpha\varepsilon)_V = (\mu\eta\beta\alpha)_V \cdot \varepsilon = \nu\cdot\varepsilon$, i.e., $\sigma =_E \nu\cdot\varepsilon$.

(i) Induction step: $\mathbf{h} > 1$.

- First, we prove the induction step when P has several open goals and the first open goal is one the base cases: $P = u_1 \to v_1/ST_1 \wedge \Omega \mid \phi \mid V, \mu$, $\Omega = \bigwedge_{i=2}^n u_i \to v_i/ST_i$ and $n > 1$, let $\Delta = \bigwedge_{i=2}^n u_i \to v_i/ST_i$; `idle`.

  We have proved for all of these cases that there exist a formula ψ_1 and substitutions λ', ν', and ρ' such that $\mathbf{G} = u_1\mu \to v_1\mu/ST_1^\mu$; `idle` $\mid \phi\mu \mid V, \mu \rightsquigarrow^+_{\lambda'} nil \mid \psi_1 \mid V, \nu'$, $\sigma =_E \nu'\cdot\rho'$, and $\psi_1\rho'$ is satisfiable. Then, also $\mathbf{G_0} = u_1\mu \to v_1\mu/ST_1^\mu$; `idle` $\wedge \Delta\mu \mid \phi\mu \mid V, \mu \rightsquigarrow^+_{\lambda'} \Delta(\mu\lambda') \mid \psi_1 \mid V, \nu' = \mathbf{G_1}$, where $\sigma =_E \nu'\cdot\rho'$ and $\psi_1\rho'$ is satisfiable.

  Now, we prove that $\mathbf{G_1} \rightsquigarrow^+_{\lambda''} nil \mid \psi \mid V, \nu$, for appropriate ν, ψ, and λ'', and that there exist a substitution ρ such that $\sigma =_E \nu\cdot\rho$ and $\psi\rho$ is satisfiable, so the theorem holds. This generic proof is valid for many of the other cases of the induction step, so we prove it only once. We provide a specific proof for each case where this proof does not apply.

  All the variables in $dom(\lambda')$ are either variables in V^μ or fresh variables generated by the calculus rules, so $V_{\Delta\mu} \cap dom(\lambda') \subseteq V^\mu$, hence $\Delta(\mu\lambda') = \Delta(\mu\lambda')_V = \Delta\nu'$ and $\mathbf{G_1} = \Delta\nu' \mid \psi_1 \mid V, \nu'$. As any narrowing step will preserve ϕ, instantiated with the substitution used in that step, as part of a conjunction of formulas, and $V_\phi \subseteq V$ then $\psi_1 = \phi\nu' \wedge \psi_2$, for appropriate ψ_2.

  As $\psi_1\rho'$ is satisfiable and $\phi\sigma$ is ground, so $\phi\nu'\rho'$ is ground, then there exists a ground substitution α such that $dom(\alpha) = V_{\psi_2\rho'}$, where all the variables are either fresh or belong to $ran(\nu')$, so $dom(\alpha) \cap ran(\nu') = \emptyset$, and $E_0 \vdash (\phi\nu' \wedge \psi_2)\rho'\alpha$, where $\phi\nu'\rho'\alpha = \phi\nu'\rho'$. As $\nu' \cdot \rho'$ is ground, so ρ' is also ground, and $dom(\alpha) \cap ran(\nu') = \emptyset$, then: (i) $\nu' \cdot (\rho' \cdot \alpha) = (\nu' \cdot \rho') \cdot \alpha = (\nu' \cdot \rho')\alpha$ and (ii) $\rho' \cdot \alpha = \rho'\alpha$, so $E_0 \vdash \psi_1(\rho' \cdot \alpha)$. Let $V' = V^{\nu'} \cup V_{\psi_2}$.

  Consider the problem $P' = \Omega\nu' \mid \psi_1 \mid V', none$ in $\mathcal{R}^{\nu'}$ and $Call^{\nu'}_{\mathcal{R}}$, whose corresponding goal is $\mathbf{G'_1} = \Delta\nu' \mid \psi_1 \mid V', none$. As $\sigma =_E \nu' \cdot \rho'$, both ground substitutions, then $V_{\Omega\sigma} = V_{\Omega(\nu'\cdot\rho')} \subseteq V_\Omega$, so $V_{\Omega(\nu'\cdot\rho')} \cap dom(\alpha) = \emptyset$ and $\Omega(\nu'\cdot\rho')\alpha = \Omega(\nu'\cdot\rho') =_E \Omega\sigma$.

As there is a c.p.t. for $[v_i\sigma]_E \in ST_i^\sigma[u_i\sigma]_E$, for $2 \leq i \leq n$, then, by Lemma 12, there are closed proof trees for all the open goals in $\Omega(\nu' \cdot \rho')$, i.e., $\Omega(\nu' \cdot \rho') \cdot \alpha$, each c.p.t. having the same depth and number of nodes as its corresponding c.p.t. for $\Omega\sigma$. As $E_0 \vdash \psi_1(\rho' \cdot \alpha)$ then $\rho' \cdot \alpha$ is a solution of P' with less nodes that those in the solution σ for P_0, since we have excluded the nodes in the c.p.t. for the first open goal, so we can apply the I.H. to P', and there exist a formula ψ and substitutions λ'' and ρ'', let $\lambda = \lambda'\lambda''$ and $\nu = (\mu\lambda)_V$, such that $\mathbf{G_1'} = \Delta\nu' \mid \psi_1 \mid V', none \rightsquigarrow^+_{\lambda''} nil \mid \psi \mid V', \lambda''_{V'}$, $\rho' \cdot \alpha =_E \lambda'' \cdot \rho''$, and $\psi\rho''$ is satisfiable, let $\rho = \rho''_{V \cup ran(\nu)}$. Then, also $\mathbf{G_1} = \Delta\nu' \mid \psi_1 \mid V, \nu' \rightsquigarrow^+_{\lambda''} nil \mid \psi \mid V, \nu$, so $\mathbf{G_0} \rightsquigarrow^+_{\lambda} nil \mid \psi \mid V, \nu$, and $\psi\rho$ is satisfiable. Finally, $\nu \cdot \rho = (\nu \cdot \rho'')_V = (\mu\lambda'\lambda''\rho'')_V =_E (\mu\lambda'\rho'\alpha)_V = (\nu'\rho'\alpha)_V =_E (\sigma\alpha)_V = \sigma$.

- In the rest of cases, the strategy in the first open goal of P may be a concatenation or not, so P has the form $u_1 \to v_1/ST_1(\,;ST) \wedge \Omega \mid \phi \mid V, \mu$, let $ST_0 = ST_1(\,;ST)$, with $\Omega = \bigwedge_{i=2}^n u_i \to v_i/ST_i$ and $n \geq 1$, let $\Delta = \bigwedge_{i=2}^n u_i \to v_i/ST_i; \texttt{idle}$, where ST is allowed to be a concatenation of strategies but ST_1 is not (in case of several concatenations), σ is a solution of the reachability problem $P' = \Omega \mid \phi \mid V, \mu$, so $[v_1\sigma]_E \in ST_0^\sigma @ [u_1\sigma]_E$ and, for $2 \leq i \leq n$, $[v_i\sigma]_E \in ST_i^\sigma @ [u_i\sigma]_E$, hence there is a c.p.t. for $[v_i\sigma]_E \in ST_i^\sigma[u_i\sigma]_E$ where the sum of the number of nodes in each c.p.t. for P' is lower than $\mathbf{h}$.

  1. $ST_1 = S_1 \mid S_2$.

     Then, one c.p.t., T, for P and σ has the form $\dfrac{\dfrac{F_1}{u_1\sigma \to w/S_i^\sigma} \quad (\dfrac{F_2}{w \to v_1\sigma/ST^\sigma})}{u_1\sigma \to v_1\sigma/ST_0^\sigma}$, with respect to $\mathcal{D}^\sigma_{\mathcal{R},Call_\mathcal{R}}$, where w ($= v_1\sigma$ if $ST_0 = ST_1$) is a term in $\mathcal{T}_\Sigma$ and i in $\{1,2\}$, let $S = S_i(;ST)$. Consider the problem $P' = u_1 \to v_1/S \mid \phi \mid V, \mu$ which for the same solution σ has a c.p.t. $T' = \dfrac{\dfrac{F_1}{u_1\sigma \to w/S_i^\sigma} \quad (\dfrac{F_2}{w \to v_1\sigma/ST^\sigma})}{u_1\sigma \to v_1\sigma/S^\sigma}$ with one less node than T.

     Then, by I.H., there exist a formula ψ_1 and two substitutions, λ' and ρ', let $\nu' = (\mu\lambda')_V$, such that $u \to v/S^\mu; \texttt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow^+_{\lambda'} nil \mid \psi_1 \mid V, \nu'$, $\sigma =_E \nu' \cdot \rho'$, and $\psi_1\rho'$ is satisfiable.

     But then, also:

     - if $n = 1$ then $\mathbf{G_0} = u \to v/ST_0^\mu; \texttt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow_{[o1\ or\ o2]} u \to v/S^\mu; \texttt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow^+_{\lambda'} nil \mid \psi_1 \mid V, \nu'$, $\sigma =_E \nu' \cdot \rho'$, and $\psi_1\rho'$ is satisfiable, so $\psi = \psi_1$, $\lambda = \lambda'$, $\nu = \nu'$, and $\rho = \rho'$;
     - else $\mathbf{G_0} = u \to v/ST_0^\mu; \texttt{idle} \wedge \Delta\mu \mid \phi\mu \mid V, \mu \rightsquigarrow_{[o1\ or\ o2]} u \to v/S^\mu; \texttt{idle} \wedge \Delta\mu \mid \phi\mu \mid V, \mu \rightsquigarrow^+_{\lambda'} \Delta(\mu\lambda') \mid \psi_1 \mid V, \nu' = \mathbf{G_1}$, $\sigma =_E \nu' \cdot \rho'$, and $\psi_1\rho'$ is satisfiable. The rest of the proof is the one given at the end of the induction step for the base cases.

  2. $ST_1 = S_1+$.

     Then there is a c.p.t. T of the form $\dfrac{\dfrac{T_1}{u_1\sigma \to w/ST_1^\sigma} \quad (\dfrac{F}{w \to v_1\sigma/ST^\sigma})}{u_1\sigma \to v_1\sigma/ST_0^\sigma}$, with respect to $\mathcal{D}^\sigma_{\mathcal{R},Call_\mathcal{R}}$, where w ($= v_1\sigma$ if $ST_0 = ST_1$) is a term in $\mathcal{T}_\Sigma$ and either $head(T_1) = u_1\sigma \to w/S_1^\sigma$ or $head(T_1) = u_1\sigma \to w/S_1^\sigma\,;S_1^\sigma+$, let $S = S_1$ or $S = S_1\,;S_1+$, depending on the case, and $S_0 = S(;ST)$.

Consider the problem $P' = u_1 \to v_1/S_0 \mid \phi \mid V, \mu$ which for the same solution σ has a c.p.t. $T' = \frac{T_1 \quad (\frac{F}{w \to v_1\sigma/ST^\sigma})}{u_1\sigma \to v_1\sigma/S_0^\sigma}$ with one less node than T.

Then, by I.H., there exist a formula ψ_1 and two substitutions, λ' and ρ', let $\nu' = (\mu\lambda')_V$, such that $u \to v/S_0^\mu; \texttt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow^+_{\lambda'} nil \mid \psi_1 \mid V, \nu'$, $\sigma =_E \nu' \cdot \rho'$, and $\psi_1\rho'$ is satisfiable.

But then, also:

- if $n = 1$ then $\mathbf{G_0} = u \to v/ST_0^\mu; \texttt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow_{[p1\ or\ p2]}$ $u \to v/S_0^\mu; \texttt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow^+_{\lambda'} nil \mid \psi_1 \mid V, \nu'$, $\sigma =_E \nu' \cdot \rho'$, and $\psi_1\rho'$ is satisfiable, so $\psi = \psi_1$, $\lambda = \lambda'$, $\nu = \nu'$, and $\rho = \rho'$;
- else $\mathbf{G_0} = u \to v/ST_0^\mu; \texttt{idle} \wedge \Delta\mu \mid \phi\mu \mid V, \mu \rightsquigarrow_{[p1\ or\ p2]} u \to v/S_0^\mu; \texttt{idle} \wedge \Delta\mu \mid \phi\mu \mid V, \mu \rightsquigarrow^+_{\lambda'} \Delta(\mu\lambda') \mid \psi_1 \mid V, \nu' = \mathbf{G_1}$, $\sigma =_E \nu' \cdot \rho'$, and $\psi_1\rho'$ is satisfiable. The rest of the proof is the one given at the end of the induction step for the base cases.

3. $ST_1 = CS$, where $\texttt{sd}\ CS := S$, let $S_0 = S(; ST)$.

Then there is a c.p.t. T of the form $\frac{\frac{T_1}{u_1\sigma \to w/ST_1^\sigma} \quad (\frac{F}{w \to v_1\sigma/ST^\sigma})}{u_1\sigma \to v_1\sigma/ST_0^\sigma}$, with respect to $\mathcal{D}^\sigma_{\mathcal{R}, Call_\mathcal{R}}$, where w ($= v_1\sigma$ if $ST_0 = ST_1$) is a term in $\mathcal{T}_\Sigma$ and $head(T_1) = u_1\sigma \to w/S^\sigma$.

Consider the problem $P' = u_1 \to v_1/S_0 \mid \phi \mid V, \mu$ which for the same solution σ has a c.p.t. $T' = \frac{T_1 \quad (\frac{F}{w \to v_1\sigma/ST^\sigma})}{u_1\sigma \to v_1\sigma/S_0^\sigma}$ with one less node than T.

Then, by I.H., there exist a formula ψ_1 and two substitutions, λ' and ρ', let $\nu' = (\mu\lambda')_V$, such that $u \to v/S_0^\mu; \texttt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow^+_{\lambda'} nil \mid \psi_1 \mid V, \nu'$, $\sigma =_E \nu' \cdot \rho'$, and $\psi_1\rho'$ is satisfiable.

But then, also:

- if $n = 1$ then $\mathbf{G_0} = u \to v/ST_0^\mu; \texttt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow_{[c1]}$ $u \to v/S_0^\mu; \texttt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow^+_{\lambda'} nil \mid \psi_1 \mid V, \nu'$, $\sigma =_E \nu' \cdot \rho'$, and $\psi_1\rho'$ is satisfiable, so $\psi = \psi_1$, $\lambda = \lambda'$, $\nu = \nu'$, and $\rho = \rho'$;
- else $\mathbf{G_0} = u \to v/ST_0^\mu; \texttt{idle} \wedge \Delta\mu \mid \phi\mu \mid V, \mu \rightsquigarrow_{[c1]} u \to v/S_0^\mu; \texttt{idle} \wedge \Delta\mu \mid \phi\mu \mid V, \mu \rightsquigarrow^+_{\lambda'} \Delta(\mu\lambda') \mid \psi_1 \mid V, \nu' = \mathbf{G_1}$, $\sigma =_E \nu' \cdot \rho'$, and $\psi_1\rho'$ is satisfiable. The rest of the proof is the one given at the end of the induction step for the base cases.

4. $ST_1 = CS(\bar{t})$, where $\texttt{sd}\ CS(\bar{x}) := S \in Call_\mathcal{R}$, let $\gamma = \{\bar{x} \mapsto \bar{t}\}$ and $S_0 = S\gamma(; ST)$.

Then there is a c.p.t. T of the form $\frac{\frac{T_1}{u_1\sigma \to w/ST_1^\sigma} \quad (\frac{F}{w \to v_1\sigma/ST^\sigma})}{u_1\sigma \to v_1\sigma/ST_0^\sigma}$, with respect to $\mathcal{D}^\sigma_{\mathcal{R}, Call_\mathcal{R}}$, where w ($= v_1\sigma$ if $ST_0 = ST_1$) is a term in $\mathcal{T}_\Sigma$ and $head(T_1) = u_1\sigma \to w/(S\gamma)^\sigma$.

Consider the problem $P' = u_1 \to v_1/S_0 \mid \phi \mid V, \mu$ which for the same solution σ has a c.p.t. $T' = \frac{T_1 \quad (\frac{F}{w \to v_1\sigma/ST^\sigma})}{u_1\sigma \to v_1\sigma/S_0^\sigma}$ with one less node than T.

Then, by I.H., there exist a formula ψ_1 and two substitutions, λ' and ρ', let $\nu' = (\mu\lambda')_V$, such that $u \to v/S_0^\mu; \texttt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow^+_{\lambda'} nil \mid \psi_1 \mid V, \nu'$, $\sigma =_E \nu' \cdot \rho'$, and $\psi_1\rho'$ is satisfiable.

But then, also:

- if $n = 1$ then $\mathbf{G_0} = u \to v/ST_0^\mu; \texttt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow_{[c1]}$

$u \to v/S_0^\mu; \texttt{idle} \mid \phi\mu \mid V, \mu \rightsquigarrow^+_{\lambda'} nil \mid \psi_1 \mid V, \nu', \sigma =_E \nu' \cdot \rho'$, and $\psi_1\rho'$ is satisfiable, so $\psi = \psi_1$, $\lambda = \lambda'$, $\nu = \nu'$, and $\rho = \rho'$;

- else $\mathbf{G_0} = u \to v/ST_0^\mu; \texttt{idle} \wedge \Delta\mu \mid \phi\mu \mid V, \mu \rightsquigarrow_{[c1]} u \to v/S_0^\mu; \texttt{idle} \wedge \Delta\mu \mid \phi\mu \mid V, \mu \rightsquigarrow^+_{\lambda'} \Delta(\mu\lambda') \mid \psi_1 \mid V, \nu' = \mathbf{G_1}$, $\sigma =_E \nu' \cdot \rho'$, and $\psi_1\rho'$ is satisfiable. The rest of the proof is the one given at the end of the induction step for the base cases.

5. $ST_1 = CS(\bar{t})$, where $\texttt{csd}\, CS(\bar{x}) := S\, if\, C \in Call_\mathcal{R}$, with C of the form $\bar{l} = \bar{r} \wedge \chi$, with $\mid \bar{l} \mid = \mid \bar{r} \mid = m$, let $\theta = \{\bar{x} \mapsto \bar{t}\}$ and let ϵ, with $dom(\epsilon) = V_{CS} \setminus (V \cup \hat{x})$, be a fresh renaming.

  Then there is a c.p.t. T of the form $\dfrac{\dfrac{F_1}{u_1\sigma \to w/(S\epsilon\theta\delta)^\sigma} \quad (\dfrac{F_2}{w \to v_1\sigma/ST^\sigma})}{u_1\sigma \to w/ST_1^\sigma}}{u_1\sigma \to v_1\sigma/ST_0^\sigma}$, with respect to $\mathcal{D}^\sigma_{\mathcal{R}, Call_\mathcal{R}}$, where w ($= v_1\sigma$ if $ST_0 = ST_1$) is a term in $\mathcal{T}_\Sigma$, $\delta : vars(C\epsilon\theta\sigma) \to \mathcal{T}_\Sigma$ is a substitution such that $\bar{l}\epsilon\theta\sigma\delta =_E \bar{r}\epsilon\theta\sigma\delta$, $E_0 \vdash \chi\epsilon\theta\sigma\delta$.

  As σ and δ are ground and $dom(\sigma) \cap dom(\delta) = \emptyset$, then $(S\epsilon\theta)^\sigma\delta = (S\epsilon\theta\delta)^\sigma$. Let $S_0 = S\epsilon\theta(; ST)$, $\tau = \sigma_1\delta$, $\mathbf{\Theta} = u_1 \to v_1/S_0; \texttt{idle}(\wedge\Delta)$, $\mathbf{\Theta'} = u_1 \to v_1/S_0(\wedge\Omega)$, and $\psi_1 = (\phi \wedge \chi\epsilon\theta)\mu$.

  Then $\mathbf{G_0} = u \to v/ST_0^\mu; \texttt{idle}(\wedge\Delta\mu) \mid \phi\mu \mid V, \mu \rightsquigarrow_{[c2]} \bigwedge_{j=1}^m (l_j\eta \to r_j\eta/\texttt{idle}) \wedge \mathbf{\Theta}\mu \mid \psi_1 \mid V, \mu = \mathbf{G_1}$. As $E_0 \vdash \phi\mu\sigma_1$, ground formula, because $V_\phi \subseteq V$, then $\phi\mu\sigma_1 = \phi\mu\tau$ and $E_0 \vdash \psi_1\tau$.

  By Lemma 14, as τ is a substitution such that $E_0 \vdash \psi_1\tau$ and $\bar{l}\eta\tau =_E \bar{r}\eta\tau$, then there exist a ground substitution τ°, substitutions $\beta_1, \ldots, \beta_m$, let $\beta = \beta_1^m$, and $abstract_{\Sigma_1}((l_j\eta\beta_1^{j-1}, r_j\eta\beta_1^{j-1})) = \langle\lambda(\bar{w}_j, \bar{w}'_j).(l_j^\circ, r_j^\circ); (\theta^\circ_{l_j}, \theta^\circ_{r_j}); (\phi^\circ_{l_j}, \phi^\circ_{r_j})\rangle$, for $1 \le j \le m$, such that $dom(\tau^\circ) = dom(\tau) \cup V_{\hat{w},\hat{w}'}$, $\tau =_{E_0} \tau^\circ_{dom(\tau)}$, $\bar{l}^\circ\tau^\circ =_E \bar{r}^\circ\tau^\circ$, $\tau^\circ \ll_E \beta_{dom(\tau^\circ)}$, let $\psi_2 = \psi_1\beta \wedge \bigwedge_{j=1}^m (\phi^\circ_{l_j} \wedge \phi^\circ_{r_j})\beta_j^m$, $\mathbf{G_1} \rightsquigarrow^m_{[d1],\beta} \mathbf{\Theta}\mu\beta \mid \psi_2 \mid V, (\mu\beta)_V = \mathbf{G_2}$, let $\xi = \mu\beta$, and for every pair of substitutions ρ and γ such that $\tau^\circ \ll_E (\beta\rho)_{dom(\tau^\circ)}$ and $\tau^\circ =_E (\beta\rho)_{dom(\tau^\circ)} \cdot \gamma$ it holds that $E_0 \vdash \psi_2\rho\gamma$ and $\mathbf{\Theta}\mu\tau =_E \mathbf{\Theta}\xi\rho\gamma$ (†).

  Consider the problem $P' = \mathbf{\Theta'}\xi \mid \psi_2 \mid V^\xi, none$ in $\mathcal{R}^{\xi_V}$ and $Call^{\xi_V}_\mathcal{R}$, whose corresponding goal is $\mathbf{G'} = \mathbf{\Theta}\xi \mid \psi_2 \mid V^\xi, none$, and take $\rho = none$. As $\tau^\circ \ll_E \beta_{dom(\tau^\circ)}$, then there exists γ' such that $\tau^\circ =_E \beta_{dom(\tau^\circ)} \cdot \gamma'$ and $ran(\tau^\circ) = ran(\beta_{dom(\tau^\circ)} \cdot \gamma')$, so as τ° is ground then γ' is ground. By (†), $E_0 \vdash \psi_2\gamma'$ and $\mathbf{\Theta}\mu\tau =_E \mathbf{\Theta}\xi\gamma'$, so also $\mathbf{\Theta'}\mu\tau =_E \mathbf{\Theta'}\xi\gamma'$, where all the terms and formulas are ground.

  Now, $\mathbf{\Theta'}\xi\gamma' =_E \mathbf{\Theta'}\mu\tau = (u_1 \to v_1/S_0(\wedge\Omega))\mu\tau = (u_1 \to v_1/S_0(\wedge\Omega))\mu\sigma_1\delta = (u_1 \to v_1/S_0(\wedge\Omega))\sigma\delta = (u_1\sigma \to v_1\sigma/S_0^\sigma\delta(\wedge\Omega^\sigma)) = \mathbf{\Theta''}$. For the first open goal of $\mathbf{\Theta''}$ there is a c.p.t. $T' = \dfrac{\dfrac{F_1}{u_1\sigma \to w/(S\epsilon\theta\delta)^\sigma} \quad (\dfrac{F_2}{w \to v_1\sigma/ST^\sigma})}{u_1\sigma \to v_1\sigma/S_0^\sigma\delta}$ with one less node than T, since $S_0^\sigma\delta = (S\epsilon\theta\delta)^\sigma(; ST^\sigma)$. As we have closed proof trees for all the other open goals in $\mathbf{\Theta''}$ then, by Lemma 12, there are closed proof trees for all the open goals in $\mathbf{\Theta'}\xi\gamma'$, each c.p.t. having the same depth and number of nodes as its correspondent c.p.t. in $\mathbf{\Theta''}$. As $E_0 \vdash \psi_2\gamma'$, then γ' is a solution for P', so we can apply the I.H. to $\mathbf{\Theta''}$, and there exist a formula ψ and substitutions ν' and ρ', such that $\mathbf{\Theta}\xi \mid \psi_2 \mid V^\xi, none \rightsquigarrow^+_{\nu'} nil \mid \psi \mid V^\xi, \nu', \gamma' =_E \nu' \cdot \rho'$,

and $\psi\rho'$ is satisfiable, where $dom(\nu') \subseteq V^\xi \subseteq ran(\xi)$. But then, let $\lambda = \beta\nu'$, $\nu = (\xi\nu')_V$, and $\rho = \rho'_{V \cup ran(\nu)}$, also $\mathbf{G_0} \rightsquigarrow^+_\beta \mathbf{\Theta}\xi \mid \psi_2 \mid V, \xi_V \rightsquigarrow^+_{\nu'} nil \mid \psi \mid V, \nu$, i.e., $\rho = \rho'_{V \cup ran(\nu)}$, so $\mathbf{G_0} \rightsquigarrow^+_\lambda nil \mid \psi \mid V, \nu$, and $\psi\rho$ is satisfiable.

As $dom(\nu) \subseteq V$, then $\nu \cdot \rho = (\nu\rho)_{\setminus ran(\nu)} = (\nu\rho'_{V \cup ran(\nu)})_{\setminus ran(\nu)} = (\rho'_V \cup \nu\rho'_{ran(\nu)})_{\setminus ran(\nu)} = \rho'_{V \setminus ran(\nu)} \cup (\nu\rho'_{ran(\nu)})_{\setminus ran(\nu)} = \rho'_V \cup (\nu\rho')_{V \setminus dom(\rho')} = (\nu\rho')_V = (\xi\nu'\rho')_V = (\mu\beta\nu'\rho')_V =_E (\mu\beta\gamma')_V =_E (\mu\tau^\circ)_V =_{E_0} (\mu\tau')_V =_B (\mu\sigma^\circ)_V =_{E_0} (\mu\sigma'\delta)_V = (\sigma\delta)_V = \sigma$, i.e., $\sigma =_E \nu \cdot \rho$.

Finally, as $\psi\rho'$ is satisfiable and ρ is more general than ρ', then $\psi\rho$ is also satisfiable.

6. $ST_1 = \mathtt{match}\ t\ \mathtt{s.t.}\ \chi\ ?\ S_1\ :\ S_2$ and there exists a substitution $\delta : V_{ST_1^\sigma} \to \mathcal{T}_\Sigma$ such that $u_1\sigma =_E t\sigma\delta$ and $E_0 \vdash (\phi \wedge \chi)\sigma\delta$ (the proof with S_2 instead of S_1, when $E_0 \vdash (\phi \wedge \neg\chi)\sigma\delta$, is exactly the same).

Then there is a c.p.t. T of the form $\dfrac{\dfrac{F_1}{\dfrac{u_1\sigma \to w/(S_1\delta)^\sigma}{u_1\sigma \to w/ST_1^\sigma}} \quad (\dfrac{F_2}{w \to v_1\sigma/ST^\sigma})}{u_1\sigma \to v_1\sigma/ST_0^\sigma}$, with respect to $\mathcal{D}^\sigma_{\mathcal{R}, Call_\mathcal{R}}$, where w ($= v_1\sigma$ if $ST_0 = ST_1$) is a term in $\mathcal{T}_\Sigma$.

As σ and δ are ground and $dom(\sigma) \cap dom(\delta) = \emptyset$, then $(S_1)^\sigma\delta = (S_1\delta)^\sigma$. Let $S_0 = S_1(; ST)$, $\tau = \sigma_1\delta$, $\mathbf{\Theta} = u_1 \to v_1/S_0; \mathtt{idle}(\wedge\Delta)$, and $\mathbf{\Theta'} = u_1 \to v_1/S_0(\wedge\Omega)$.

Let $abstract_{\Sigma_1}((u, t\mu)) = \langle\lambda(\bar{x}, \bar{y}).(u^\circ, t^\circ); (\theta^\circ_u, \theta^\circ_t); (\phi^\circ_u, \phi^\circ_t)\rangle$. As $u_1\sigma$ is ground then $u\sigma'\delta = u_1\mu\sigma'\delta = u_1\sigma\delta = u_1\sigma =_E t\sigma\delta = t\mu\sigma'\delta$ so, by Lemma 9, there exists a ground substitution σ° such that $u^\circ\sigma^\circ =_B t^\circ\sigma^\circ$, $E_0 \vdash (\phi^\circ_u \wedge \phi^\circ_t)\sigma^\circ$, $dom(\sigma^\circ) = dom(\sigma'\delta) \cup \hat{x} \cup \hat{y}$, and $\sigma'\delta =_{E_0} \sigma^\circ_{dom(\sigma'\delta)}$.

Let $\psi_1 = (\phi \wedge \chi)\mu \wedge \phi^\circ_u \wedge \phi^\circ_t$. As $E_0 \vdash (\phi \wedge \chi)\mu\sigma'\delta$, $V_{(\phi\wedge\chi)\mu\sigma'\delta} \cap (\hat{x} \cup \hat{y}) = \emptyset$, and $\sigma'\delta =_{E_0} \sigma^\circ_{dom(\sigma'\delta)} = \sigma^\circ_{\setminus(\hat{x}\cup\hat{y})}$, then $E_0 \vdash (\phi \wedge \chi)\mu\sigma^\circ$, so $E_0 \vdash \psi_1\sigma^\circ$.

As $u^\circ\sigma^\circ =_B t^\circ\sigma^\circ$, then there exist substitutions ν and τ such that $\eta \in CSU_B(u^\circ = t^\circ)$ and $\sigma^\circ =_B \eta\cdot\tau$, so $\psi_1\sigma^\circ = \psi_1\eta\tau$, hence $E_0 \vdash \psi_1\eta\tau$ and $\psi_1\eta$ is satisfiable. Let $\xi = \mu\eta$.

Now, $\mathbf{G_0} = (u_1 \to v_1/ST_1; \mathtt{idle}(\wedge\Delta))\mu \mid \phi\mu \mid V, \mu \rightsquigarrow_{[i1],\eta} (u_1 \to v_1\ /S_0; \mathtt{idle}(\wedge\Delta))\xi \mid \psi_1\eta \mid V, \xi_V = \mathbf{G_1}$.

Consider the problem $P' = \mathbf{\Theta'}\xi \mid \psi_1\eta \mid V^\xi, none$ in $\mathcal{R}^{\xi_V}$ and $Call^{\xi_V}_\mathcal{R}$, whose corresponding goal is $\mathbf{G'} = \mathbf{\Theta}\xi \mid \psi_1\eta \mid V^\xi, none$, and take $\rho = none$. Now, $\mathbf{\Theta'}\xi\tau = \mathbf{\Theta'}\mu\eta\tau =_B \mathbf{\Theta'}\mu\sigma^\circ = (u_1 \to v_1/S_0(\wedge\Omega))\mu\sigma^\circ = (u_1 \to v_1/S_0(\wedge\Omega))\mu\sigma^\circ =_{E_0} (u_1 \to v_1/S_0(\wedge\Omega))\mu\sigma_1\delta = (u_1 \to v_1/S_0(\wedge\Omega))\sigma\delta = (u_1\sigma \to v_1\sigma/S_0^\sigma\delta(\wedge\Omega^\sigma)) = \mathbf{\Theta''}$. For the first open goal of $\mathbf{\Theta''}$ there is a c.p.t. $T' = \dfrac{\dfrac{F_1}{u_1\sigma \to w/(S_1\delta)^\sigma} \quad (\dfrac{F_2}{w \to v_1\sigma/ST^\sigma})}{u_1\sigma \to v_1\sigma/S_0^\sigma\delta}$ with one less node than T, since $S_0^\sigma\delta = (S_1\delta)^\sigma(; ST^\sigma)$. As we have closed proof trees for all the other open goals in $\mathbf{\Theta''}$ then, by Lemma 12, there are closed proof trees for all the open goals in $\mathbf{\Theta'}\xi\tau$, each c.p.t. having the same depth and number of nodes as its correspondent c.p.t. in $\mathbf{\Theta''}$. As $E_0 \vdash \psi_1\eta\tau$, then τ is a solution for P', so we can apply the I.H. to $\mathbf{\Theta''}$, and there exist a formula ψ_2 and substitutions ν'' and ρ'', such that $\mathbf{\Theta}\xi \mid \psi_1\eta \mid V^\xi, none \rightsquigarrow^+_{\nu''} nil \mid \psi_2 \mid V^\xi, \nu''$, $\tau =_E \nu''\cdot\rho''$, and $\psi_2\rho''$ is satisfiable, where $dom(\nu'') \subseteq V^\xi \subseteq ran(\xi)$. Let $\lambda' = \eta\nu''$, $\nu' = (\xi\nu'')_V$, and $\rho' = \rho''_{V \cup ran(\nu')}$. As ρ' is more general than ρ'' and $\psi_2\rho''$ is satisfiable, then $\psi_2\rho'$ is satisfiable. Also, $\nu' \cdot \rho' = (\xi\nu'')_V \cdot \rho''_{V \cup ran(\nu')} = (\xi\nu''\rho'')_V = (\mu\eta\nu''\rho'')_V =_E (\mu\eta\tau)_V =_B (\mu\sigma^\circ)_V =_{E_0} (\mu\sigma'\delta)_V = (\sigma\delta)_V = \sigma_V = \sigma$, i.e., $\sigma =_E \nu' \cdot \rho'$. Now:

- if $n = 1$ then $\mathbf{G_0} \rightsquigarrow_{[i1],\eta} (u_1 \to v_1 \,/ S_0; \mathtt{idle})\xi \mid \psi_1\eta \mid V, \xi_V \rightsquigarrow^{+}_{\nu''} nil \mid \psi_2 \mid V, \nu'$, i.e., $\mathbf{G_0} \rightsquigarrow^{+}_{\lambda'} nil \mid \psi_2 \mid V, \nu'$, $\sigma =_E \nu' \cdot \rho'$, and $\psi_2\rho'$ is satisfiable, so $\psi = \psi_2$, $\lambda = \lambda'$, $\nu = \nu'$, and $\rho = \rho'$;
- else $\mathbf{G_0} \rightsquigarrow_{[i1],\eta} (u_1 \to v_1 \,/ S_0; \mathtt{idle} \wedge \Delta)\mu\eta \mid \psi_1\eta \mid V, \xi_V \rightsquigarrow^{+}_{\nu''} \Delta(\mu\lambda') \mid \psi_2 \mid V, \nu'$, i.e., $\mathbf{G_0} \rightsquigarrow^{+}_{\lambda'} \Delta(\mu\lambda') \mid \psi_2 \mid V, \nu'$, $\sigma =_E \nu' \cdot \rho'$, and $\psi_2\rho'$ is satisfiable. The rest of the proof is the one given at the end of the induction step for the base cases.

7. $ST_1 = \mathtt{matchrew}\ t\ \mathtt{s.t.}\ \bar{l} = \bar{r} \wedge \chi\ \mathtt{by}\ \bar{z}\ \mathtt{using}\ \bar{S}$, where $|\bar{z}| = k$, $|\bar{l}| = |\bar{r}| = m$, and $t = t[\bar{z}]_{\bar{p}}$.

   By definition, $V \cap \hat{z} = \emptyset$ and $\hat{z} \subset \mathcal{X}_1$. As $[v_1\sigma]_E \in ST_0^\sigma@[u_1\sigma]_E$ then there is a c.p.t., with respect to $\mathcal{D}^\sigma_{\mathcal{R},Call_\mathcal{R}}$, of the form $\dfrac{\frac{F_1}{z_1\delta \to t_1/S_1^\sigma\delta} \cdots \frac{F_k}{z_k\delta \to t_k/S_k^\sigma\delta}}{u_1\sigma \to t\sigma\delta[\bar{t}]_{\bar{p}}/ST_1^\sigma} \quad (\frac{F}{t\sigma\delta[\bar{t}]_{\bar{p}} \to v_1\sigma/ST^\sigma})}{u_1\sigma \to v_1\sigma/ST_0^\sigma}$, where $\hat{z} \subseteq dom(\delta)$, ground substitution, $u_1\sigma =_E t\sigma\delta$, $\bar{l}\sigma\delta =_E \bar{r}\sigma\delta$, and $E_0 \vdash \chi\sigma\delta$, with all these terms and the formula ground. Also, if $ST_0 = ST_1$ then $t\sigma\delta[\bar{t}]_{\bar{p}} =_E v_1\sigma$.

   Let $abstract_{\Sigma_1}((u, t\mu)) = \langle\lambda(\bar{w}, \bar{w}').(u^\circ, t^\circ); (\theta_u^\circ, \theta_t^\circ); (\phi_u^\circ, \phi_t^\circ)\rangle$. As $u_1\sigma$ is ground, then $u\sigma'\delta = u_1\mu\sigma'\delta = u_1\sigma\delta = u_1\sigma =_E t\sigma\delta = t\mu\sigma'\delta$ so, by Lemma 9, there exists a ground substitution σ° such that $u^\circ\sigma^\circ =_B t^\circ\sigma^\circ$, $E_0 \vdash (\phi_u^\circ \wedge \phi_t^\circ)\sigma^\circ$, $dom(\sigma^\circ) = dom(\sigma'\delta) \cup \hat{w} \cup \hat{w}'$, and $\sigma'\delta =_{E_0} \sigma^\circ_{dom(\sigma'\delta)}$.

   Let $\psi_1 = (\phi \wedge \chi)\mu \wedge \phi_u^\circ \wedge \phi_t^\circ$. As $E_0 \vdash (\phi \wedge \chi)\mu\sigma'\delta$, $V_{(\phi\wedge\chi)\mu} \cap (\hat{w} \cup \hat{w}') = \emptyset$, and $\sigma'\delta =_{E_0} \sigma^\circ_{dom(\sigma'\delta)} = \sigma^\circ_{\setminus(\hat{w}\cup\hat{w}')}$, then $E_0 \vdash (\phi \wedge \chi)\mu\sigma^\circ$, so $E_0 \vdash \psi_1\sigma^\circ$.

   As $u^\circ\sigma^\circ =_B t^\circ\sigma^\circ$, then there exist substitutions ν and τ such that $\eta \in CSU_B(u^\circ = t^\circ)$ and $\sigma^\circ =_B \eta\cdot\tau$, so $\psi_1\sigma^\circ = \psi_1\eta\tau$, hence $E_0 \vdash \psi_1\eta\tau$, ground formula, and $\psi_1\eta$ is satisfiable. Let $\mathbf{\Theta} = \bigwedge_{j=1}^k (x_j \to y_j/S_j; \mathtt{idle}) \wedge t[\bar{y}]_{\bar{p}} \to v_1/ST; \mathtt{idle} \wedge \Delta$, where $\bar{x}$ and $\bar{y}$ are fresh versions of $\bar{z}$, and let λ be the renaming from $\bar{x}$ to $\bar{z}$, i.e., $\bar{x}\lambda = \bar{z}$, and let $\mathbf{\Theta}' = \bigwedge_{j=1}^k (x_j \to y_j/S_j) \wedge t[\bar{y}]_{\bar{p}} \to v_1/ST \wedge \Omega$.

   Now, $\mathbf{G_0} = (u_1 \to v_1/ST_1; ST; \mathtt{idle} \wedge \Delta)\mu \mid \phi\mu \mid V, \mu \rightsquigarrow_{[m],\eta} (\bigwedge_{j=1}^m (l_j \to r_j/\mathtt{idle}) \wedge \mathbf{\Theta})\mu\eta \mid \psi_1\eta \mid V, (\mu\eta)_V = \mathbf{G_1}$, all ground terms.

   As $\bar{l}\sigma\delta =_E \bar{r}\sigma\delta$, $\sigma'\delta =_{E_0} \sigma^\circ_{dom(\sigma'\delta)}$, $\sigma'\delta =_{E_0} \sigma^\circ_{\setminus(\hat{w}\cup\hat{w}')}$, $\sigma^\circ =_B \eta\cdot\tau$, and $V_{\hat{l}\mu,\hat{r}\mu} \cap (\hat{w} \cup \hat{w}' \cup ran(\eta)) = \emptyset$, then $(\bar{l}, \bar{r})\mu\eta\tau = (\bar{l}, \bar{r})\mu(\eta\cdot\tau) =_B (\bar{l}, \bar{r})\mu\sigma^\circ = (\bar{l}, \bar{r})\mu\sigma^\circ_{\setminus(\hat{w}\cup\hat{w}')} =_{E_0} (\bar{l}, \bar{r})\mu\sigma'\delta$, i.e., $(\bar{l}, \bar{r})\mu\eta\tau =_E (\bar{l}, \bar{r})\sigma\delta$, so $\bar{l}\mu\eta\tau =_E \bar{r}\mu\eta\tau$, since $\bar{l}\sigma\delta =_E \bar{r}\sigma\delta$.

   In the same way, as $u_1\sigma =_E t\sigma\delta$, ground terms, and $v_1\sigma$ is also ground, then $V_{\theta\mu\eta\tau} = \hat{x} \cup \hat{y}$. Let $\tau' = \tau \cup \lambda \cdot \delta_{\hat{z}} \cup \{\bar{y} \mapsto \bar{t}\}$, so $V_{G_1\tau'} = \emptyset$. As $dom(\mu) \subseteq V$ then $\mathbf{\Theta}\mu = \mathbf{\Theta}^\mu$ so, by Lemma 14, as τ' is a ground substitution such that $V_{G_1} \subseteq dom(\tau')$, $E_0 \vdash \psi_1\eta\tau'$, and $\bar{l}\mu\eta\tau' =_E \bar{r}\mu\eta\tau'$, there exist a ground substitution τ°, substitutions $\beta_1, \ldots, \beta_m$, let $\beta = \beta_1^m$, and abstractions $abstract_{\Sigma_1}((l_j\beta_1^{j-1}, r_j\beta_1^{j-1})) = \langle\lambda(\bar{w}_j, \bar{w}_j').(l_j^\circ, r_j^\circ); (\theta_{l_j}^\circ, \theta_{r_j}^\circ); (\phi_{l_j}^\circ, \phi_{r_j}^\circ)\rangle$, for $1 \le j \le m$, such that $dom(\tau^\circ) = dom(\tau') \cup V_{\hat{w},\hat{w}'}$, $\tau' =_{E_0} \tau^\circ_{dom(\tau')}$, $\bar{l}^\circ\tau^\circ =_E \bar{r}^\circ\tau^\circ$, $\tau^\circ \ll_E \beta_{dom(\tau^\circ)}$, let $\xi = \mu\eta\beta$ and $\psi_2 = \psi_1\eta\beta \wedge \bigwedge_{j=1}^m (\phi_{l_j}^\circ \wedge \phi_{r_j}^\circ)\beta_j^m$, also $\mathbf{G_1} \rightsquigarrow^m_{[d1]} \mathbf{\Theta}\xi \mid \psi_2 \mid V, \xi_V = \mathbf{G_2}$, and for every pair of substitutions ρ and γ such that $\tau^\circ \ll_E (\beta\rho)_{dom(\tau^\circ)}$ and $\tau^\circ =_E (\beta\rho)_{dom(\tau^\circ)} \cdot \gamma$ it holds that $E_0 \vdash \psi_2\rho\gamma$ and $\mathbf{\Theta}\mu\eta\tau' =_E \mathbf{\Theta}\xi\rho\gamma$ (†).

   Consider the problem $P' = \mathbf{\Theta}'\xi \mid \psi_2 \mid (\hat{y} \cup \hat{x} \cup V)^\xi, none$ in $\mathcal{R}^{\xi_V}$ and $Call_\mathcal{R}^{\xi_V}$, whose corresponding goal is $\mathbf{G}' = \mathbf{\Theta}\xi \mid \psi_2 \mid (\hat{y} \cup \hat{x} \cup V)^\xi, none$, and take $\rho = none$. As $\tau^\circ \ll_E \beta_{dom(\tau^\circ)}$, then there exists γ' such that $\tau^\circ =_E \beta_{dom(\tau^\circ)} \cdot \gamma'$ and

$ran(\tau^\circ) = ran(\beta_{dom(\tau^\circ)} \cdot \gamma')$, so as τ° is ground then γ' is ground. By (†), $E_0 \vdash \psi_2\gamma'$ and $\mathbf{\Theta}\mu\eta\tau' =_E \mathbf{\Theta}\xi\gamma'$, so also $\mathbf{\Theta}'\mu\eta\tau' =_E \mathbf{\Theta}'\xi\gamma'$, where all the terms and formulas are ground. Now, $\mathbf{\Theta}'\xi\gamma' =_E \mathbf{\Theta}'\mu\eta\tau' = (\bigwedge_{j=1}^k (x_j \to y_j/S_j) \wedge t[\bar{y}]_{\bar{p}} \to v_1/ST; \mathtt{idle} \wedge \Delta)\mu\eta\tau' = (\bigwedge_{j=1}^k (z_j\delta \to t_j/S_j) \wedge t[\bar{t}]_{\bar{p}} \to v_1/ST; \mathtt{idle} \wedge \Delta)\mu\eta\tau =_B (\bigwedge_{j=1}^k (z_j\delta \to t_j/S_j) \wedge t[\bar{t}]_{\bar{p}} \to v_1/ST; \mathtt{idle} \wedge \Delta)\mu\sigma^\circ =_{E_0} (\bigwedge_{j=1}^k (z_j\delta \to t_j/S_j) \wedge t[\bar{t}]_{\bar{p}} \to v_1/ST; \mathtt{idle} \wedge \Delta)\mu\sigma'\delta = (\bigwedge_{j=1}^k (z_j\delta \to t_j/S_j) \wedge t[\bar{t}]_{\bar{p}} \to v_1/ST; \mathtt{idle} \wedge \Delta)\sigma\delta = \bigwedge_{j=1}^k (z_j\delta \to t_j/S_j^\sigma\delta) \wedge t[\bar{t}]_{\bar{p}} \to v_1/ST^\sigma; \mathtt{idle} \wedge \Delta^\sigma = \mathbf{\Theta}''$. As we have closed proof trees for all the open goals in $\mathbf{\Theta}''$ then, by Lemma 12, there are closed proof trees for all the open goals in $\mathbf{\Theta}'\xi\gamma'$, each c.p.t. having the same depth and number of nodes as its corresponding c.p.t. in $\mathbf{\Theta}''$. As $E_0 \vdash \psi_2\gamma'$, then γ' is a solution for P'. The difference with respect to the closed proof trees in the answer σ for the reachability problem P, is that we have two less nodes, $t\sigma\delta \to t\sigma\delta[\bar{t}]_{\bar{p}}/ST_1^\sigma$ and $t\sigma\delta \to v_1\sigma/ST_1^\sigma; ST^\sigma$, so we can apply the I.H. to $\mathbf{\Theta}'\xi\gamma'$, and there exist a formula ψ and substitutions ν' and ρ', such that $\mathbf{G}' = \mathbf{\Theta}\xi \mid \psi_2 \mid (\hat{y} \cup \hat{x} \cup V)^\xi, none \rightsquigarrow^+ nil \mid \psi \mid (\hat{y} \cup \hat{x} \cup V)^\xi, \nu', \gamma' =_E \nu' \cdot \rho'$, and $\psi\rho'$ is satisfiable, where $dom(\nu') \subseteq (\hat{y} \cup \hat{x} \cup V)^\xi \subseteq ran(\xi)$. But then, let $\nu = (\xi\nu')_V$ and $\rho = \rho'_{V \cup ran(\nu)}$, also $\mathbf{G_2} = \mathbf{\Theta}\xi \mid \psi_2 \mid V, \xi_V \rightsquigarrow^+ nil \mid \psi \mid V, \nu$.

As $dom(\nu) \subseteq V$, then $\nu \cdot \rho = (\nu\rho)_{\setminus ran(\nu)} = (\nu\rho'_{V \cup ran(\nu)})_{\setminus ran(\nu)} = (\rho'_V \cup \nu\rho'_{ran(\nu)})_{\setminus ran(\nu)} = \rho'_{V \setminus ran(\nu)} \cup (\nu\rho'_{ran(\nu)})_{\setminus ran(\nu)} = \rho'_V \cup (\nu\rho')_{V \setminus dom(\rho')} = (\nu\rho')_V = (\xi\nu'\rho')_V = (\mu\eta\beta\nu'\rho')_V =_E (\mu\eta\beta\gamma')_V =_E (\mu\eta\tau^\circ)_V =_{E_0} (\mu\eta\tau')_V =_B (\mu\sigma^\circ)_V =_{E_0} (\mu\sigma'\delta)_V = (\sigma\delta)_V = \sigma$, i.e., $\sigma =_E \nu \cdot \rho$.

Finally, as $\psi\rho'$ is satisfiable and ρ is more general than ρ', then $\psi\rho$ is also satisfiable.

8. $ST_1 = c[\gamma]\{\bar{S}\}$, with $c : l \to r$ *if* C a rule in R, $C = \bar{l} \to \bar{r} \mid \chi$, $\bar{S} = S_1, \ldots, S_m$, and $dom(\gamma) \cap vars(\bar{S}) = \emptyset$.

As $[v_1\sigma]_E \in ST_0^\sigma@[u_1\sigma]_E$ then there is a c.p.t. T, with respect to $\mathcal{D}^\sigma_{\mathcal{R}, Call_\mathcal{R}}$, of the form $\dfrac{\frac{T_1 \cdots T_m}{u_1\sigma \to w/ST_1^\sigma} \ (T_0)}{u_1\sigma \to v_1\sigma/ST_0^\sigma}$, where $T_i = \dfrac{F_i}{l_i\gamma\sigma\delta \to r_i\gamma\sigma\delta/S_i^\sigma\delta}$, for $1 \le i \le m$, $T_0 = \dfrac{F}{w \to v_1\sigma/ST^\sigma}$, $[w]_E \in c^\sigma[(\gamma\sigma)_{dom(\gamma)}]@[u_1\sigma]_E$, where $\delta : vars(c\gamma\sigma) \to \mathcal{T}_\Sigma$, with $E_0 \vdash \chi\gamma\sigma\delta$, there is $p \in pos(u_1\sigma)$ s.t. $u_1\sigma =_E u_1\sigma[l\gamma\sigma\delta]_p$, and $w = u_1\sigma[r\gamma\sigma\delta]_p$ if T_0 exists or $w = v_1\sigma$, otherwise. By Lemma 12.13, $\bar{l}\gamma\sigma\delta \to_{R^\sigma/E} \bar{r}\gamma\sigma\delta$, so $u_1\sigma \xrightarrow[c^\sigma, p, (\gamma\sigma)_{dom(\gamma)}\delta]{}{}^1_{R^\sigma/E} w$. Let $\alpha = \gamma\sigma\delta$ ($= \gamma\delta\sigma$ since $dom(\delta) \cap dom(\sigma) = \emptyset$ and both substitutions are ground), $\alpha' = \gamma\sigma$, and $c' = c^\sigma(\gamma\sigma)_{dom(\gamma)}$ ($= c\alpha'$ because σ is ground and, by definition, $dom(\gamma) \cap dom(\sigma) = \emptyset$). As $u_1\sigma \xrightarrow[c'\delta, p]{}{}^1_{R^\sigma/E} w$ then, by Theorem 15, $u_1\sigma \xrightarrow[c'_1\delta, p']{}{}^1_{R^\sigma, B} w$, since $\mathcal{R}$ is closed under B-extensions, with $c'_1 \in c'_B$ and appropriate p', as seen in the proof of Lemma 11, so also $u_1\sigma \xrightarrow[c'_1\delta, p']{}{}^1_{R^\sigma/E} w$, hence we can assume that $c' = c'_1$, $p = p'$, and T is the c.p.t. for $[w]_E \in c'_1@[u_1\sigma]_E$ using $u_1\sigma \xrightarrow[c'_1, p', \delta]{}{}^1_{R^\sigma/E} w$. Since $\sigma = \mu\sigma'$, if we let $l_1 = l\gamma\mu$ and $r_1 = r\gamma\mu$ then $c'(= c\alpha')$ has also the form $c\alpha' : l_1\sigma' \to r_1\sigma'$ *if* $C\alpha'$.

Let $c_2 : l_2 \to r_2$ *if* C_2 be a fresh version of c^μ except for $dom(\gamma) \cup V^\mu$ ($= dom(\gamma) \cup dom(\sigma')$), and let τ be the renaming that verifies $c_2 = c^\mu\tau$, so $(l_2, r_2, C_2) = (l, r, C)(\mu \uplus \tau)$, where $(dom(\tau) \cup ran(\tau)) \cap (dom(\gamma) \cup V^\mu) = \emptyset$. Then $l_2(\gamma\mu)_{dom(\gamma)} = l(\mu \uplus \tau)(\gamma\mu)_{dom(\gamma)} = l((\gamma\mu)_{dom(\gamma)} \uplus \mu \uplus \tau) = l((\gamma\mu)_{dom(\gamma)} \uplus$

$\mu)\tau = l\gamma\mu\tau = l_1\tau$, so also $r_2(\gamma\mu)_{dom(\gamma)} = r_1\tau$ and $C_2(\gamma\mu)_{dom(\gamma)} = C\gamma\mu\tau$. Let $l_c = l_2(\gamma\mu)_{dom(\gamma)}$ and $\sigma'' = \tau^{-1}\sigma'$; then $l_c\sigma'' = l_1\tau\tau^{-1}\sigma' = l_1\sigma'$. Now:

(a) $abstract_{\Sigma_1}(l_c) = \langle \lambda\bar{y}.l^\circ; \theta_l^\circ; \phi_l^\circ\rangle$, where $\bar{y} = y_1, \ldots, y_{i_y}$, $l^\circ = l_c[\bar{y}]_{\bar{p}}$, $\bar{p} = p_1, \ldots, p_{i_y}$, $\hat{p} = top_{\Sigma_0}(l_c)$, $\theta_l^\circ = \bigcup_{i=1}^{i_y}\{y_i \mapsto l_c|_{p_i}\}$, and $\phi_l^\circ = \bigwedge_{i=1}^{i_y} y_i = l_c|_{p_i}$;

(b) since $l_1\sigma' = l_c\sigma''$ and $top_{\Sigma_0}(l_c) \subseteq top_{\Sigma_0}(l_c\sigma'')$, then $abstract_{\Sigma_1}(l_1\sigma') = abstract_{\Sigma_1}(l_c\sigma'') = \langle \lambda\bar{y}\bar{z}.l^\circ_{c\sigma''}; \theta^\circ_{c\sigma''}; \phi^\circ_{c\sigma''}\rangle$, where $\bar{z} = z_1, \ldots, z_{i_z}$, $l^\circ_{c\sigma''} = l_c\sigma''[\bar{y}]_{\bar{p}}[\bar{z}]_{\bar{q}}$, $\hat{q} = top_{\Sigma_0}(l_c\sigma'') \setminus top_{\Sigma_0}(l_c)$, $\theta^\circ_{c\sigma''} = \bigcup_{i=1}^{i_y}\{y_i \mapsto l_c|_{p_i}\sigma''\} \cup \bigcup_{j=1}^{i_z}\{z_j \mapsto l_c\sigma''|_{q_j}\}$, and $\phi^\circ_{c\sigma''} = (\bigwedge_{i=1}^{i_y} y_i = l_c|_{p_i}\sigma'' \wedge \bigwedge_{j=1}^{i_z} z_j = l_c\sigma''|_{q_j})$;

(c) as $u_1\sigma \xrightarrow[c',p,\delta]{}{}^{1}_{R^\sigma,B} w$, then there is a substitution $\delta' : \hat{y} \cup \hat{z} \cup V_{c'} \to \mathcal{T}_\Sigma$, such that $\delta'_{V_{c'}} = \delta$, $rep(u_1\sigma|_p) =_B l^\circ_{c\sigma''}\delta'$, $w =_E u_1\sigma[r_1\sigma'\delta']_p = u_1\sigma[r_1\sigma'\delta]_p = u_1\sigma[r\gamma\mu\sigma'\delta]_p = u_1\sigma[r\gamma\sigma\delta]_p = u_1\sigma[r\alpha]_p$, and $E_0 \vdash (\chi\alpha' \wedge \phi^\circ_{c\sigma''})\delta'$, so $E_0 \vdash \chi\alpha$ (since $\chi\alpha'\delta' = \chi\alpha'\delta = \chi\alpha$), i.e., $E_0 \vdash \chi\gamma\delta\mu\sigma'$, $\bar{y}\delta' =_{E_0} l_c|_{\bar{p}}\sigma''\delta'$ and $\bar{z}\delta' =_{E_0} l_c\sigma''|_{\bar{q}}\delta'$;

(d) as $p \in pos_{\Sigma_1}(u_1\sigma)$ and σ is R/E-normalized, hence R, E-normalized by Theorem 15, then $p \in pos_{\Sigma_1}(u_1)$, so $u_1\sigma|_p = u_1|_p\sigma = u_1|_p\mu\sigma' = u_1\mu|_p\sigma' = u|_p\sigma'$; and

(e) as τ is a fresh renaming then $\emptyset = V_u \cap ran(\tau) = V_u \cap dom(\tau^{-1})$, so $u|_p\tau^{-1}\sigma' = u|_p\sigma' = u_1\sigma|_p =_{E_0} rep(u_1\sigma|_p) =_B l^\circ_{c\sigma''}\delta' = l_c\sigma''[\bar{y}]_{\bar{p}}[\bar{z}]_{\bar{q}}\delta' =_{E_0} l_c\sigma'' = l_c\tau^{-1}\sigma'$, i.e., $u|_p\tau^{-1}\sigma' =_E l_c\tau^{-1}\sigma'$;

Let $abstract_{\Sigma_1}(u|_p) = \langle \lambda\bar{x}.u^\circ; \theta_u^\circ; \phi_u^\circ\rangle$. As $dom(\tau^{-1}\sigma') = ran(\tau) \cup V^\mu$ then, by Lemma 9, there exists a ground substitution σ° such that $u^\circ\sigma^\circ =_B l^\circ\sigma^\circ$, $E_0 \vdash (\phi_u^\circ \wedge \phi_l^\circ)\sigma^\circ$, $dom(\sigma^\circ) = dom(\tau^{-1}\sigma') \cup \hat{x} \cup \hat{y} = ran(\tau) \cup V^\mu \cup \hat{x} \cup \hat{y}$, and $\tau^{-1}\sigma' =_{E_0} \sigma^\circ_{dom(\tau^{-1}\sigma')} = \sigma^\circ_{ran(\tau)\cup V^\mu}$, so $(\tau^{-1}\sigma')_{V^\mu} =_{E_0} \sigma^\circ_{V^\mu}$ and $\tau^{-1} = (\tau^{-1}\sigma')_{\setminus V^\mu} =_{E_0} \sigma^\circ_{\setminus V^\mu}$. As $(dom(\tau) \cup ran(\tau)) \cap V^\mu = \emptyset$ and $dom(\sigma') = V^\mu$ then $\sigma' = \sigma'_{V^\mu} = (\tau^{-1}\sigma')_{V^\mu} =_{E_0} \sigma^\circ_{V^\mu}$.

As $u^\circ\sigma^\circ =_B l^\circ\sigma^\circ$, then there exist substitutions ϑ and ζ' such that $\vartheta \in CSU_B(u^\circ = l^\circ)$ and $\sigma^\circ =_B \vartheta \cdot \zeta'$, let $\xi = \mu \cdot \vartheta$ and $\zeta = \zeta'_{ran(\xi_V)\cup(V\setminus dom(\xi_V))}$. As $dom(\mu) \subseteq V$ and $\sigma' =_{E_0} \sigma^\circ_{V^\mu}$ then $\sigma = \mu \cdot \sigma' =_{E_0} \mu \cdot \sigma^\circ_{V^\mu} =_B \mu \cdot (\vartheta \cdot \zeta')_{V^\mu} = \mu \cdot (\vartheta\zeta')_{V^\mu} = (\mu\vartheta\zeta')_V = \xi_V \cdot \zeta'_{ran(\xi_V)\cup(V\setminus dom((\xi_V))} = \xi_V \cdot \zeta'_{ran(\xi_V)\cup(V\setminus dom(\xi_V))} = \xi_V \cdot \zeta$, i.e., $\sigma =_E \xi_V \cdot \zeta$, so also $\sigma =_E (\mu\vartheta\zeta')_V$.

As $\chi_2(\gamma\mu)_{dom(\gamma)} = \chi\gamma\mu\tau$, $dom(\sigma^\circ) = ran(\tau) \cup V^\mu \cup \hat{x} \cup \hat{y}$, and $\tau^{-1}\sigma' =_{E_0} \sigma^\circ_{ran(\tau)\cup V^\mu}$, then $\chi_2(\gamma\mu)_{dom(\gamma)}\sigma^\circ\delta' = \chi\gamma\mu\tau\sigma^\circ_{ran(\tau)\cup V^\mu}\delta' =_{E_0} \chi\gamma\mu\tau\tau^{-1}\sigma'\delta' = \chi\gamma\mu\sigma'\delta' = \chi\gamma\mu\sigma'\delta = \chi\gamma\delta\mu\sigma'$ so, as $E_0 \vdash \chi\gamma\delta\mu\sigma'$, also $E_0 \vdash \chi_2(\gamma\mu)_{dom(\gamma)}\sigma^\circ\delta'$.

As $E_0 \vdash \phi\sigma$, $V_{\phi\mu} \subseteq V^\mu$, and $\sigma' =_{E_0} \sigma^\circ_{V^\mu}$ then $\phi\mu\sigma^\circ =_{E_0} \phi\mu\sigma' = \phi\sigma$, so $E_0 \vdash \phi\mu\sigma^\circ$. Now, as $E_0 \vdash (\phi_u^\circ \wedge \phi_l^\circ)\sigma^\circ$, then $E_0 \vdash (\phi\mu \wedge \phi_u^\circ \wedge \phi_l^\circ)\sigma^\circ$ ground formula, so $E_0 \vdash (\phi\mu \wedge \phi_u^\circ \wedge \phi_l^\circ)\sigma^\circ\delta'$ and $E_0 \vdash (\phi\mu \wedge \phi_u^\circ \wedge \phi_l^\circ \wedge \chi_2(\gamma\mu)_{dom(\gamma)})\sigma^\circ\delta'$. Let $\varphi^\circ = \phi\mu \wedge \phi_u^\circ \wedge \phi_l^\circ \wedge \chi_2(\gamma\mu)_{dom(\gamma)}$ and $\varphi = \varphi^\circ\vartheta$. As $\sigma^\circ =_B \vartheta \cdot \zeta'$, so $\varphi^\circ\sigma^\circ = \varphi^\circ\vartheta\zeta' = \varphi\zeta'$, then $E_0 \vdash \varphi\zeta'\delta'$, let $\delta'' = \zeta'\delta'$, hence φ is also satisfiable. Let $\mathbf{\Theta} = \bar{l}\gamma\tau \to \bar{r}\gamma\tau/\bar{S}\tau; \mathtt{idle}(\wedge u_1[r\gamma\tau]_p \to v_1/ST; \mathtt{idle}) \wedge \Delta$ and $\mathbf{\Theta}' = \bar{l}\gamma\tau \to \bar{r}\gamma\tau/\bar{S}\tau(\wedge u_1[r\gamma\tau]_p \to v_1/ST) \wedge \Omega$.

Now, $\mathbf{G_0} = (u_1 \to v_1/ST_1; ST; \mathtt{idle} \wedge \Delta)^\mu \mid \phi\mu \mid V, \mu \rightsquigarrow_{[t]}$

$u \to^1 x_0, x_0 \to v/ST_1^\mu; ST^\mu; \mathtt{idle} \wedge \Delta^\mu \mid \phi\mu \mid V, \mu \rightsquigarrow^*_{[c]}$
$u|_p \to^1 x, u[x]_p \to v/c^\mu[(\gamma\mu)_{dom(\gamma)}]\{\bar{S}^\mu\}; ST^\mu; \mathtt{idle} \wedge \Delta^\mu \mid \phi\mu \mid V, \mu = \mathbf{G_1}$,
where $u|_p$ cannot be a variable, say x_u, because as $p \in pos_\Sigma(u_1)$ then, by (c), also $x_u\sigma' \to^1_{R,B} r\alpha$, so σ would not be R/E-normalized.
As $\vartheta \in CSU_B(u^\circ = l^\circ)$ and $c_2 : l_2 \to r_2$ *if* C_2, where $r_2(\gamma\mu)_{dom(\gamma)} = r_1\tau = r\gamma\mu\tau$ and $C_2(\gamma\mu)_{dom(\gamma)} = C\gamma\mu\tau$, let $\vartheta' = \vartheta \cup \{x \mapsto r_1\tau\vartheta\}$, then $\mathbf{G_1} \rightsquigarrow_{[r],\vartheta'} \mathbf{\Theta}\xi \mid \varphi \mid V, \xi_V = \mathbf{G_2}$, let $V_0 = (\hat{y} \cup \hat{z} \cup V \cup V_{c\gamma\tau})^\xi$.

Consider the problem $P' = \mathbf{\Theta}'\xi \mid \varphi \mid V_0, none$ in $\mathcal{R}^{\xi_V}$ and $Call_{\mathcal{R}}^{\xi_V}$, whose corresponding goal is $\mathbf{G}' = \mathbf{\Theta}\xi \mid \varphi \mid V_0, none$. Now, $\mathbf{\Theta}'\xi\delta'' = \mathbf{\Theta}'\mu\vartheta\zeta'\delta' =_E \mathbf{\Theta}'(\sigma \cup (\vartheta\zeta'\delta')_{\setminus V^\mu}) =_B \mathbf{\Theta}'(\sigma \cup (\sigma^\circ\delta')_{\setminus V^\mu}) =_{E_0} \mathbf{\Theta}'(\sigma \cup \tau^{-1}\delta') = \mathbf{\Theta}'(\sigma \cup (\tau^{-1}\delta)_{V_{c'}} \cup \delta'_{\hat{y}\cup\hat{z}}) = \mathbf{\Theta}'(\sigma \cup (\tau^{-1}\delta)_{V_{c'}}) = \mathbf{\Theta}'\tau^{-1}\sigma\delta = \bar{l}\gamma\sigma\delta \to \bar{r}\gamma\sigma\delta/\bar{S}^\sigma\delta(\wedge u_1\sigma[r\gamma\sigma\delta]_p \to v_1\sigma/ST^\sigma) \wedge \Omega^\sigma = \mathbf{\Theta}''$.
We have closed proof trees $T_1, \ldots, T_m$ (and T_0 if ST_0 is a concatenation) for the open goals before Ω^σ, whose sum of nodes is two less than the number of nodes in T. As we have closed proof trees for all the other open goals in $\mathbf{\Theta}''$ then, by Lemma 12, there are closed proof trees for all the open goals in $\mathbf{\Theta}'\xi\delta''$, each c.p.t. having the same depth and number of nodes as its correspondent c.p.t. in $\mathbf{\Theta}''$. As $E_0 \vdash \varphi\delta''$, then δ'' is a solution for P', so we can apply the I.H. to $\mathbf{\Theta}'\xi\delta''$, and there exist a formula φ_2 and substitutions ν'' and ρ'', such that $\mathbf{\Theta}\xi \mid \varphi \mid V_0, none \rightsquigarrow^+_{\nu''} nil \mid \varphi_2 \mid V_0, \nu''_{V_0}$, $\delta'' =_E \nu''_{V_0}\cdot\rho''$, and $\varphi_2\rho''$ is satisfiable, where $dom(\nu''_{V_0}) \subseteq V_0 \subseteq ran(\xi)$. Let $\lambda' = \vartheta'\nu''$, $\nu' = (\xi\nu'')_V$, and $\rho' = \rho''_{V\cup ran(\nu')}$. As ρ' is more general than ρ'' and $\varphi_2\rho''$ is satisfiable then $\varphi_2\rho'$ is satisfiable. Also, as $V \subseteq V_0$ and $\sigma =_E (\mu\vartheta\zeta')_V$, $\nu'\cdot\rho' = (\xi\nu'')_V \cdot \rho''_{V\cup ran(\nu')} = (\xi\nu''\rho'')_V = (\mu\vartheta\nu''\rho'')_V =_E (\mu\vartheta\delta'')_V = (\mu\vartheta\zeta'\delta')_V =_E (\sigma\delta')_V = \sigma$, i.e., $\sigma =_E \nu'\cdot\rho'$. Now:

- if $n = 1$ then $\mathbf{G_0} \rightsquigarrow^+_{\vartheta'} \mathbf{G_2} \rightsquigarrow^+_{\nu''} nil \mid \varphi_2 \mid V, \nu'$, i.e., $\mathbf{G_0} \rightsquigarrow^+_{\lambda'} nil \mid \varphi'_2 \mid V, \nu'$, $\sigma =_E \nu'\cdot\rho'$, and $\varphi'_2\rho'$ is satisfiable, so $\psi = \varphi'_2$, $\lambda = \lambda'$, $\nu = \nu'$, and $\rho = \rho'$;
- else $\mathbf{G_0} \rightsquigarrow^+_{\vartheta'} \mathbf{G_2} \rightsquigarrow^+_{\nu''} \Delta(\mu\lambda') \mid \varphi'_2 \mid V, \nu'$, i.e., $\mathbf{G_0} \rightsquigarrow^+_{\lambda'} \Delta(\mu\lambda') \mid \varphi'_2 \mid V, \nu'$, $\sigma =_E \nu'\cdot\rho'$, and $\varphi'_2\rho'$ is satisfiable. The rest of the proof is the one given at the end of the induction step for the base cases.

9. $ST_1 = \mathtt{top}(c[\gamma]\{\bar{S}\})$.
The proof is almost exactly the same as the previous one, particularized for the case $p = \epsilon$, so $u|_p = u$, $u_1|_p = u_1$, $u_1[r\gamma\tau]_p = r\gamma\tau$, et cetera. The only difference is found in the initial narrowing steps, where instead of $\mathbf{G_0} \rightsquigarrow_{[t]} \rightsquigarrow^*_{[c]} \mathbf{G_1} \rightsquigarrow_{[r],\vartheta'} \mathbf{G_2}$ now we have $\mathbf{G_0} \rightsquigarrow_{[tp],\vartheta'} \mathbf{G_2}$.

□

Chapter 7

Conclusions

"The journey, not the destination matters."
—T.S. Eliot

7.1 First, some personal thoughts

So, the time to come to an end is upon us. It never seems to be the right time to finish one's work but, alas, this is our time to look behind and see what we have accomplished.

The journey that is about to end here has taken a long time to finish and is one for which I longed for years, something that would come now and then to my mind, but I never found the right moment to address. Finally, after being on Earth for more than half a century, some undesired professional events had the unexpected side effect of giving me the chance to accomplish my desire.

Not only the work on this Ph.D., but also that for my M.S., have been very rewarding. They have brought me back in some way to my early days in university, where everyday you would learn so many new, complex things. The wish to always absorb new bits of knowledge has been a faithful companion during all my life, even when my daily tasks would be completely away from any intellectual activity.

This journey is quite different from the one that a student in the beginning of their research career may take, first choosing an area of interest for this initial step, and seeking to get enough insight into it that may help them decide whether to follow that path for a long time or jump to another one as soon as their work there is finished. In this case, due to the natural evolution of us as human beings, probably there will be no future destination harbors, at least in such unexplored stormy theoretical oceans. Some Summer cruise near a safe bay may be nice, though, to keep that spark of interest in scientific matters alive. Anyway, I have no reason for complaining; I have enjoyed the experience and I am grateful for having had the opportunity to restart my academic studies almost 30 years later.

7.2 About my personal growth as a researcher

The one thing that has surprised me the most, and I was not aware of it, is the way that science is done when working on a Ph.D. When you are a graduate student you learn

things that have been already known for some time. The path is usually clear and you just have to follow it. Suddenly, after a few months reviewing the previous work in the field, the whole scenario becomes unknown territory. You still have some beacons over the horizon to serve as reference points, but mostly it is you alone in a back and forth game. For each paper that you begin working on, you plan your path, based on your prior knowledge, start following it, and when you hit a wall, which happens most of the time, you try to understand why and, with that added knowledge in your backpack, go back and plan a new path. Sometimes you will only need a small detour to avoid the wall, but other times you will need a different approach to overcome the difficulty. Finally, when the work is finished, everything is in place, it seems that you always had in mind the new definitions and intermediate results to present, because you already knew what you needed, and just had to write them down.

On the contrary, many definitions include restrictions that had to be added to make things work. Other definitions are concepts created to deal with some wall you hit, because it will be easier to prove some results... results that you found out that you needed when you hit the wall. This is how the different texts that I have worked on for this thesis have been written. From the more general setup that you think you can manage in your area of interest, you try to include the smallest set of restrictions that you can think of each time that you hit a wall. Many times I have wondered if it would be worth publishing some work with an inventory of the things that you tried to do, could not accomplish, and why they could not be done. If written in a most general way, perhaps then someone could be spared from hitting the same already known wall.

Apart from the contributions in my area of research, this resilience that I have been talking about, and the capacity to learn from failure, analyzing and taking steps to overcome each one of the failures are my biggest takeaways from all these years.

7.3 About the research plan

The research plan deserves some words. How can you anticipate what's happening in the years to come when you have not even begun working on your Ph.D. yet? That is what you are asked to do when writing your research proposal. Although you are given the opportunity to modify your goal each year, trying to fill in the form for the research plan for the first time makes you aware of the little that you know about your thesis subject.

When you look back and compare the first and last editions of the research plan, you realize how its surrounding environment has reshaped it, in a concrete application of Evolution Theory, where new features appear due to unexpected changes and other features are removed just because time is finite or because they no longer make sense.

In this case, as already explained in Chapter 1, the aim was adjusted twice due to new features added to the Maude language: the first time with the inclusion of SMT solvers; later on, with the inclusion of the strategy language, together with the use of parameters in the rewrite theories and reachability problems, an insight into the thesis subject provided by the research itself. In a way, research plans also unravel by themselves. As a side effect, the study of conditional narrowing for rewrite theories modulo membership equational logic with SMT solvers was removed from the research plan.

Another point worth thinking about is the subject of the investigation itself. Narrowing, and in particular conditional narrowing, have always had to deal with the state explosion problem, so they are deemed as almost unpractical in the real world, although there

has been success in some areas of interest, like cryptographic protocol analysis [MT07]. Why focus on this subject then? We chose to investigate conditional narrowing modulo membership equational theories because very little was known about it. There was almost no literature, there were even no precise definitions except for some particular cases, so it seemed that it would be worth looking into the matter with an ample focus to extend our theoretical knowledge. Nonetheless, this theoretical approach may be of use in practice, after all. The different techniques to deal with conditional narrowing that we present in each chapter of this dissertation are, up to some point, orthogonal between them, and could be combined with other known narrowing techniques, since the requirements and limitations of each of these techniques have been established.

7.4 Technical conclusions

In this dissertation we have studied conditional narrowing modulo in rewriting logic. Different approaches have been considered and for each of these approaches a narrowing calculus has been presented, and a proof of its soundness and weak completeness has also been given. The Maude engine has served as a base for the prototypes developed for some of the narrowing calculi.

We showed in Chapter 3 that conditional narrowing modulo was feasible.

Then, in Chapter 4 we showed a method to rein in the state explosion problem inherent to narrowing: first we defined the new concepts of fresh pattern property for MEL and rewrite theories; then, the concept of narrowable rewrite theory, extending the specifications and reachability problems supported for narrowing; finally, a transformation for dealing with unification for MEL theories. The main results are the calculi for unification and reachability, both calculi applying a leftmost strategy as a first limitation to control the state explosion problem, and imposing several restrictions to their corresponding valid narrowing steps, most of them based on the requirement of only generating E, B-normalized terms or substitutions when trying to apply any narrowing rule either for unification or for reachability.

In Chapter 5 we showed our first narrowing calculus with SMT solvers, that adds inductive proving to the theories supported by the SMT solver. New concepts of set of topmost Σ_0-positions and representative of a term have been presented to help define the R, B-rewrite relation with SMT solvers. After presenting the corresponding narrowing calculus and proving its correctness properties, we discussed two different approaches taken to develop prototypes of the calculus and showed a comparison of their performance when different improvements were added to them.

Finally, in Chapter 6 we extended the support for SMT solvers from the previous chapter with the addition of two elements: *strategies* for explicit control of the rewrite steps that can be applied, and *parameters* for enhanced expressivity of the reachability problems. The semantics of the supported strategy language was then presented and its main properties proved, followed by a narrowing calculus that supports both strategies and parameters and whose correctness properties were stated and proved. A prototype for this calculus, with some examples, was also developed.

The development of the prototypes in Chapters 5 and 6 was thought as an interesting complement for the new features being presented, and as a means to see the calculi at work with some toy-level examples. In the end, the prototypes proved to be very useful, since they were a source for novel ideas and also contributed to point out some flaws in

the calculi.

As an example of novel idea, while testing the first prototype from Chapter 5 we found that, even with the simplification provided by the module SMTLOGIC, the prototype could find pairs of solutions that only differ in equivalent SMT constraints. This led to the idea of developing the second prototype with its own SMT-based search engine.

Another novel idea, the admission of variable SMT parameters in the specifications and problems, emerged while trying to formulate more interesting reachability problems than those that we had already tried. The problems that we were thinking of required this new feature, so we experimentally introduced it in the prototypes for Chapter 5, giving theoretical support to it in Chapter 6.

Also a flaw in the calculus was found after the prototype for Chapter 6 was developed and before the soundness and completeness of the calculus were fully proved. While debugging a test problem that was failing, we found out that we have to use representatives of the Σ_0-terms in the definition of R,B-rewriting for rewrite theories with built-in in Section 2.3.7, so we rewrote the theory, to add all the required new definitions and results, and accordingly modified the prototype. Chapter 5 now includes a corrected version of the calculus presented in [AMPP17], that had the same error, the difference being that in this dissertation rules unification and rewrite in Figure 5.3 use abstracted terms to compute CSU_B.

7.5 Future work

Just a guideline for somebody who would take and carry the torch: future work should focus on broadening the applicability of the calculus with SMT solvers and strategies.

- One line of work could be the extension of the narrowing calculus to support OS equational theories of the form $(\Sigma, E_0 \cup E_1 \cup B)$, i.e., to admit other equations apart from E_0 and the axioms B.
- Another line of work would involve broadening the strategies and reachability problems supported by the calculus.

The first line of work may allow for a straightforward approach by designing a narrowing calculus for unification with SMT solvers similar to the one in Section 4.4 achieving the support for SMT solvers as in the rules in Figure 5.3. Of course, you never know, these kinds of extensions always seem easy before one gets down into business.

A different approach for this line of work would be the design of a narrowing calculus that uses both rules and strategies, instead of only rules, to control the generation of the states of the search tree. It would be nice to check the performance of this approach compared to the current one.

The second line of work deserves some motivation. The work on the semantics of the admitted strategies for this narrowing calculus grew out of the work in [MOMV04, EMOMV07] where a set-theoretic semantics for rewriting with strategies was presented. The semantics was adapted in this dissertation so that it could support narrowing instead of rewriting, and all the required technical results were proved.

When the work on proving the correctness of the calculus, using these technical results, was almost finished it became apparent a mismatch between the calculus and the strategies semantics for narrowing: while the calculus updates every instantiation of a variable everywhere in the reachability problem, the proof tree that is generated for the

correctness proofs can only update correctly the instances of the parameters, but not the instances of the rest of the variables if they are in different branches.

Due to this fact, the correctness of the narrowing calculus using the existing semantics could only be proved for reachability problems where the iterated and concatenated strategies have no variables in common apart from the parameters, a restriction included in Section 6.4.

The solution that I propose to overcome this obstacle is the second line of work: to develop a new environment-based semantics for the strategies, in a similar way to the small-step operational semantics proposed by Rubén Rubio in his Ph.D. dissertation [RC22], that propagates every instantiation of a variable in one part of a reachability problem to the rest of it.

As the existing prototype mimics the narrowing calculus, which does not need any change, the instantiations of any variable are propagated in the correct way, so it could be used to solve unrestricted problems once the correctness of the calculus with respect to the new semantics is proved.

Bibliography

[AEH94] Sergio Antoy, Rachid Echahed, and Michael Hanus. A needed narrowing strategy. In *Proceedings of the 21st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '94, page 268–279, New York, NY, USA, 1994. Association for Computing Machinery.

[AEH00] Sergio Antoy, Rachid Echahed, and Michael Hanus. A needed narrowing strategy. *J. ACM*, 47(4):776–822, jul 2000.

[AILS07] Luca Aceto, Anna Ingólfsdóttir, Kim G. Larsen, and Jiri Srba. *Reactive Systems: Modelling, Specification and Verification*. Cambridge University Press, August 2007.

[AMPP14] Luis Aguirre, Narciso Martí-Oliet, Miguel Palomino, and Isabel Pita. Conditional narrowing modulo in rewriting logic and maude. In Escobar [Esc14], pages 80–96.

[AMPP15] Luis Aguirre, Narciso Martí-Oliet, Miguel Palomino, and Isabel Pita. Sentence-normalized conditional narrowing modulo in rewriting logic and maude. In Narciso Martí-Oliet, Peter Csaba Ölveczky, and Carolyn L. Talcott, editors, *Logic, Rewriting, and Concurrency - Essays dedicated to José Meseguer on the Occasion of His 65th Birthday*, volume 9200 of *Lecture Notes in Computer Science*, pages 48–71. Springer, 2015.

[AMPP17] Luis Aguirre, Narciso Martí-Oliet, Miguel Palomino, and Isabel Pita. Conditional narrowing modulo SMT and axioms. In Wim Vanhoof and Brigitte Pientka, editors, *Proceedings of the 19th International Symposium on Principles and Practice of Declarative Programming, Namur, Belgium, October 09 - 11, 2017*, pages 17–28. ACM, 2017.

[AMPP18] Luis Aguirre, Narciso Martí-Oliet, Miguel Palomino, and Isabel Pita. Sentence-normalized conditional narrowing modulo in rewriting logic and maude. *J. Autom. Reason.*, 60(4):421–463, 2018.

[AMPP23] Luis Aguirre, Narciso Martí-Oliet, Miguel Palomino, and Isabel Pita. Strategies in conditional narrowing modulo SMT plus axioms. In Pedro López-García, John P. Gallagher, and Roberto Giacobazzi, editors, *Analysis, Verification and Transformation for Declarative Programming and Intelligent Systems - Essays Dedicated to Manuel Hermenegildo on the Occasion of His 60th Birthday*, volume 13160 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2023.

[BGM87] Pier Giorgio Bosco, Elio Giovannetti, and Corrado Moiso. Refined strategies for semantic unification. In Hartmut Ehrig, Robert Kowalski, Giorgio Levi, and Ugo Montanari, editors, *TAPSOFT '87*, pages 276–290, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg.

[BM06] Roberto Bruni and José Meseguer. Semantic foundations for generalized rewrite theories. *Theoretical Computer Science*, 360(1-3):386–414, 2006.

[BM12] Kyungmin Bae and José Meseguer. Model checking LTLR formulas under localized fairness. In Francisco Durán, editor, *Rewriting Logic and Its Applications - 9th International Workshop, WRLA 2012, Held as a Satellite Event of ETAPS, Tallinn, Estonia, March 24-25, 2012, Revised Selected Papers*, volume 7571 of *Lecture Notes in Computer Science*, pages 99–117. Springer, 2012.

[BM14] Kyungmin Bae and José Meseguer. Infinite-state model checking of LTLR formulas using narrowing. In Escobar [Esc14], pages 113–129.

[Boc93] Alexander Bockmayr. Conditional narrowing modulo a set of equations. *Applicable Algebra in Engineering, Communication and Computing*, 4:147–168, 1993.

[CDE+02] Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, and José F. Quesada. Maude: Specification and programming in rewriting logic. *Theoretical Computer Science*, 285(2):187–243, 2002.

[CDE+07] Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, and Carolyn Talcott. *All About Maude - A High-Performance Logical Framework: How to Specify, Program, and Verify Systems in Rewriting Logic*, volume 4350 of *Lecture Notes in Computer Science*. Springer, 2007.

[CDE+23] Manuel Clavel, Francisco Durán, Steven Eker, Santiago Escobar, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, Rubén Rubio, and Carolyn Talcott. Maude manual (version 3.3.1), 2023.

[CEM14] Andrew Cholewa, Santiago Escobar, and José Meseguer. Constrained Narrowing for Conditional Equational Theories Modulo Axioms. Technical report, http://hdl.handle.net/2142/50289, C.S. Department, University of Illinois at Urbana-Champaign, August 2014.

[CEM15] Andrew Cholewa, Santiago Escobar, and José Meseguer. Constrained narrowing for conditional equational theories modulo axioms. *Sci. Comput. Program.*, 112:24–57, 2015.

[DEE+20] Francisco Durán, Steven Eker, Santiago Escobar, Narciso Martí-Oliet, José Meseguer, Rubén Rubio, and Carolyn L. Talcott. Programming and symbolic computation in Maude. *J. Log. Algebr. Meth. Program.*, 110, 2020.

[DLM+08] Francisco Durán, Salvador Lucas, Claude Marché, José Meseguer, and Xavier Urbain. Proving operational termination of membership equational programs. *Higher-Order and Symbolic Computation*, 21(1-2):59–88, 2008.

[DM10] Francisco Durán and José Meseguer. A Church-Rosser checker tool for conditional order-sorted equational Maude specifications. In Peter Csaba Ölveczky, editor, *Rewriting Logic and Its Applications - 8th International Workshop, WRLA 2010, Held as a Satellite Event of ETAPS 2010, Paphos, Cyprus, March 20-21, 2010, Revised Selected Papers*, volume 6381 of *Lecture Notes in Computer Science*, pages 69–85. Springer, 2010.

[DM12] Francisco Durán and José Meseguer. On the Church-Rosser and coherence properties of conditional order-sorted rewrite theories. *Journal of Logic and Algebraic Programming*, 81(7-8):816–850, 2012.

[dMB08] Leonardo de Moura and Nikolaj Bjørner. Z3: an efficient SMT solver. In C. R. Ramakrishnan and Jakob Rehof, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340. Springer, 2008.

[DMT98] G. Denker, J. Meseguer, and C.L. Talcott. Protocol specification and analysis in maude. In N. Heintze, N. Heintze, and J. Wing, editors, *Proceedings of Workshop on Formal Methods and Security Protocols, June 25 1998, Indianapolis, Indiana*, 1998.

[DOS88] Nachum Dershowitz, Mitsuhiro Okada, and G. Sivakumar. Canonical conditional rewrite systems. In Ewing Lusk and Ross Overbeek, editors, *9th International Conference on Automated Deduction*, pages 538–549, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg.

[Ech90] Rachid Echahed. On completeness of narrowing strategies. *Theoretical Computer Science*, 72(2):133–146, 1990.

[EM19] Santiago Escobar and José Meseguer. Canonical narrowing with irreducibility constraints as a symbolic protocol analysis method. In Joshua D. Guttman, Carl E. Landwehr, José Meseguer, and Dusko Pavlovic, editors, *Foundations of Security, Protocols, and Equational Reasoning - Essays Dedicated to Catherine A. Meadows*, volume 11565 of *Lecture Notes in Computer Science*, pages 15–38. Springer, 2019.

[EMM09] Santiago Escobar, Catherine Meadows, and José Meseguer. Maude-NPA: Cryptographic protocol analysis modulo equational properties. In Alessandro Aldini, Gilles Barthe, and Roberto Gorrieri, editors, *Foundations of Security Analysis and Design V*, volume 5705 of *Lecture Notes in Computer Science*, pages 1–50. Springer, 2009.

[EMOMV07] Steven Eker, Narciso Martí-Oliet, José Meseguer, and Alberto Verdejo. Deduction, strategies, and rewriting. In Myla Archer, Thierry Boy de la Tour, and César Muñoz, editors, *Proceedings of the 6th International Workshop on Strategies in Automated Deduction, STRATEGIES 2006, Seattle, WA, USA, August 16, 2006*, volume 174(11) of *Electronic Notes in Theoretical Computer Science*, pages 3–25. Elsevier, 2007.

[EMS08] Santiago Escobar, José Meseguer, and Ralf Sasse. Variant narrowing and equational unification. In Grigore Rosu, editor, *Proceedings of the Seventh International Workshop on Rewriting Logic and its Applications, WRLA 2008, Budapest, Hungary, March 29-30, 2008*, volume 238 of *Electronic Notes in Theoretical Computer Science*, pages 103–119. Elsevier, 2008.

[Esc04] Santiago Escobar. Thesis: Strategies and analysis techniques in functional program optimization. *AI Commun.*, 17(1):35–37, 2004.

[Esc14] Santiago Escobar, editor. *Rewriting Logic and Its Applications - 10th International Workshop, WRLA 2014, Held as a Satellite Event of ETAPS, Grenoble, France, April 5-6, 2014, Revised Selected Papers*, volume 8663 of *Lecture Notes in Computer Science*. Springer, 2014.

[ESM12] Santiago Escobar, Ralf Sasse, and José Meseguer. Folding variant narrowing and optimal variant termination. *Journal of Logic and Algebraic Programming*, 81(7-8):898–928, 2012.

[Fay79] M. Fay. First-order unification in an equational theory. In *Proc. 4th Workshop on Automated Deduction*, pages 161–167, Austin, TX, USA, 1979. Academic Press.

[GM86a] Elio Giovannetti and Corrado Moiso. A completeness result for E-unification algorithms based on conditional narrowing. In Mauro Boscarol, Luigia Carlucci Aiello, and Giorgio Levi, editors, *Foundations of Logic and Functional Programming, Workshop, Trento, Italy, December 15-19, 1986, Proceedings*, volume 306 of *Lecture Notes in Computer Science*, pages 157–167. Springer, 1986.

[GM86b] Joseph A. Goguen and José Meseguer. EQLOG: equality, types, and generic modules for logic programming. In Doug DeGroot and Gary Lindstrom, editors, *Logic Programming: Functions, Relations, and Equations*, pages 295–363. Prentice-Hall, 1986.

[GS87] Jean H. Gallier and Wayne Snyder. A general complete e-unification procedure. In Pierre Lescanne, editor, *Rewriting Techniques and Applications*, pages 216–227, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg.

[Ham00] Mohamed Hamada. Strong completeness of a narrowing calculus for conditional rewrite systems with extra variables. *Electronic Notes in Theoretical Computer Science*, 31:89–103, 2000. CATS 2000 Computing: the Australasian Theory Symposium.

[Han97] Michael Hanus. Curry: A multi-paradigm declarative language (system description). In François Bry, Burkhard Freitag, and Dietmar Seipel, editors, *Twelfth Workshop Logic Programming, WLP 1997, 17-19 September 1997, München, Germany, Technical Report PMS-FB-1997-10*. Ludwig Maximilians Universität München, 1997.

[HKMN95] M. Hanus, H. Kuchen, and J.J. Moreno-Navarro. Curry: A truly functional logic language. In *Proc. ILPS'95 Workshop on Visions for the Future of Logic Programming*, pages 95–107, 1995.

[HL91] Gérard P. Huet and Jean-Jacques Lévy. Computations in orthogonal rewriting systems, I. In Jean-Louis Lassez and Gordon D. Plotkin, editors, *Computational Logic - Essays in Honor of Alan Robinson*, pages 395–414. The MIT Press, 1991.

[Höl89] Steffen Hölldobler. *Foundations of Equational Logic Programming*, volume 353 of *Lecture Notes in Computer Science*. Springer, 1989.

[Hul80] Jean-Marie Hullot. Canonical forms and unification. In Wolfgang Bibel and Robert A. Kowalski, editors, *5th Conference on Automated Deduction, Les Arcs, France, July 8-11, 1980, Proceedings*, volume 87 of *Lecture Notes in Computer Science*, pages 318–334. Springer, 1980.

[JKK83] Jean Pierre Jouannaud, Claude Kirchner, and Helene Kirchner. Incremental construction of unification algorithms in equational theories. In Josep Diaz, editor, *Automata, Languages and Programming, 10th Colloquium, Barcelona, Spain, July 18-22, 1983, Proceedings*, pages 361–373, Berlin, Heidelberg, 1983. Springer Berlin Heidelberg.

[Jou83] Jean-Pierre Jouannaud. Confluent and coherent equational term rewriting systems application to proofs in abstract data types. In Giorgio Ausiello and Marco Protasi, editors, *Trees in Algebra and Programming, CAAP'83, 8th Colloquium L'Aquila, March 9-11, 1983, Proceedings*, pages 269–283, Berlin, Heidelberg, 1983. Springer Berlin Heidelberg.

[Kap85] Stéphane Kaplan. Fair conditional term rewriting systems: Unification, termination and confluence. In Hans-Jörg Kreowski, editor, *Recent Trends in Data Type Specification*, pages 136–155, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg.

[KB70] Donald E. Knuth and Peter B. Bendix. Simple word problems in universal algebras. In JOHN LEECH, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon, 1970.

[KR94] Hélène Kirchner and Christophe Ringeissen. Constraint solving by narrowing in combined algebraic domains. In Pascal Van Hentenryck, editor, *Logic Programming, Proceedings of the Eleventh International Conference on Logic Programming, Santa Marherita Ligure, Italy, June 13-18, 1994*, pages 617–631. MIT Press, 1994.

[LE22] Raúl López-Rueda and Santiago Escobar. Canonical narrowing with irreducibility and SMT constraints as a generic symbolic protocol analysis method. In Kyungmin Bae, editor, *Rewriting Logic and Its Applications - 14th International Workshop, WRLA@ETAPS 2022, Munich, Germany, April 2-3, 2022, Revised Selected Papers*, volume 13252 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2022.

[LM09] Salvador Lucas and José Meseguer. Operational termination of membership equational programs: the order-sorted way. *Electronic Notes in Theoretical Computer Science*, 238(3):207–225, 2009.

[LMM05] Salvador Lucas, Claude Marché, and José Meseguer. Operational termination of conditional term rewriting systems. *Information Processing Letters*, 95(4):446–453, 2005.

[Mes90] José Meseguer. Rewriting as a unified model of concurrency. In J.C.M. Baeten and J.W. Klop, editors, *CONCUR '90 Theories of Concurrency: Unification and Extension*, volume 458 of *Lecture Notes in Computer Science*, pages 384–400. Springer, 1990.

[Mes92] José Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73–155, April 1992.

[Mes12] José Meseguer. Twenty years of rewriting logic. *Journal of Logic and Algebraic Programming*, 81(7-8):721–781, 2012.

[Mes17] José Meseguer. Strict coherence of conditional rewriting modulo axioms. *Theor. Comput. Sci.*, 672(C):1–35, April 2017.

[Mes20] José Meseguer. Generalized rewrite theories, coherence completion, and symbolic methods. *J. Log. Algebraic Methods Program.*, 110, 2020.

[Mes23] José Meseguer. Variants and satisfiability in the infinitary unification wonderland. *J. Log. Algebraic Methods Program.*, 134:100877, 2023.

[MH94] Aart Middeldorp and Erik Hamoen. Completeness results for basic narrowing. *Applicable Algebra in Engineering, Communication and Computing*, 5:213–253, 1994.

[MOI96] Aart Middeldorp, Satoshi Okui, and Tetsuo Ida. Lazy narrowing: Strong completeness and eager variable elimination. *Theoretical Computer Science*, 167(1):95–130, 1996.

[MOMV04] Narciso Martí-Oliet, José Meseguer, and Alberto Verdejo. Towards a strategy language for Maude. In Narciso Martí-Oliet, editor, *Proceedings of the Fifth International Workshop on Rewriting Logic and its Applications, WRLA 2004, Barcelona, Spain, March 27-April 4, 2004*, volume 117 of *Electronic Notes in Theoretical Computer Science*, pages 417–441. Elsevier, 2004.

[MT04] José Meseguer and Prasanna Thati. Symbolic reachability analysis using narrowing and its application to verification of cryptographic protocols. In Narciso Martí-Oliet, editor, *Proceedings of the Fifth International Workshop on Rewriting Logic and Its Applications, WRLA 2004, Barcelona, Spain, March 27-28, 2004*, volume 117 of *Electronic Notes in Theoretical Computer Science*, pages 153–182. Elsevier, 2004.

[MT07] José Meseguer and Prasanna Thati. Symbolic reachability analysis using narrowing and its application to verification of cryptographic protocols. *Higher-Order and Symbolic Computation*, 20(1-2):123–160, 2007.

[MW91] José Meseguer and Timothy C. Winkler. Parallel programmming in maude. In Jean-Pierre Banâtre and Daniel Le Métayer, editors, *Research Directions in High-Level Parallel Programming Languages, Mont Saint-Michel, France, June 17-19, 1991, Procedings*, volume 574 of *Lecture Notes in Computer Science*, pages 253–293. Springer, 1991.

[Ohl02] Enno Ohlebusch. *Advanced topics in term rewriting.* Springer, 2002.

[Plo72] Gordon Plotkin. Building in equational theories. *Machine Intelligence 7*, pages 73–90, 1972.

[RC22] Rubén Rafael Rubio Cuéllar. Model checking of strategy-controlled systems in rewriting logic. PhD Thesis, Universidad Complutense de Madrid, Facultad de Informática, January 2022.

[RMM17] Camilo Rocha, José Meseguer, and César A. Muñoz. Rewriting modulo SMT and open system analysis. *J. Log. Algebr. Meth. Program.*, 86(1):269–297, 2017.

[RMPV18] Rubén Rubio, Narciso Martí-Oliet, Isabel Pita, and Alberto Verdejo. Parameterized strategies specification in Maude. In José Luiz Fiadeiro and Ionut Tutu, editors, *Recent Trends in Algebraic Development Techniques - 24th IFIP WG 1.3 International Workshop, WADT 2018, Egham, UK, July 2-5, 2018, Revised Selected Papers*, volume 11563 of *Lecture Notes in Computer Science*, pages 27–44. Springer, 2018.

[RMPV21] Rubén Rubio, Narciso Martí-Oliet, Isabel Pita, and Alberto Verdejo. The semantics of the Maude strategy language. Technical report, https://docta.ucm.es/entities/publication/56811493-1c03-439b-979f-c962aa71cfe2, Facultad de Informática, Universidad Complutense de Madrid, August 2021.

[RW69] G. Robinson and L. Wos. Paramodulation and Theorem Proving in First-Order Theories with Equality. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*. Edinburgh University Press, 1969.

[Sla74] James R. Slagle. Automated theorem-proving for theories with simplifiers commutativity, and associativity. *J. ACM*, 21(4):622–642, 1974.

[Vir02] Patrick Viry. Equational rules for rewriting logic. *Theor. Comput. Sci.*, 285(2):487–517, 2002.