

UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE FILOLOGÍA



TESIS DOCTORAL

**SSG : simplified Spanish grammar : una gramática del español de tipo
HPSG de coste computacional reducido**

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Benjamín Ramírez González

Director

Fernando Sánchez León

Madrid, 2014

SSG: Simplified Spanish Grammar

Una gramática del español de tipo **HPSG** de coste
computacional reducido

Tesis doctoral



Universidad Complutense de Madrid

Instituto Universitario Ortega y Gasset

Doctorando: Benjamín Ramírez González

Director: Fernando Sánchez León

Agradecimientos

Este trabajo no hubiese visto nunca la luz sin la ayuda inestimable de varios profesores, compañeros, familiares y amigos. Estoy especialmente agradecido a mi tutor, Fernando Sánchez León, por lo mucho que he aprendido de él. Él me introdujo en el campo apasionante de la Lingüística Computacional, me enseñó los rudimentos de la programación y ha seguido con dedicación el desarrollo de mi investigación, corrigiendo, sugiriendo y abriendo horizontes. También agradezco su ayuda, su amistad y paciente dedicación al resto de compañeros del Departamento de Lingüística Computacional de la Real Academia Española (hoy Departamento de Tecnología), del cual formé parte entre los años 2007 y 2008. Aprendí de ellos multitud de cuestiones que han sido muy útiles para la elaboración de este trabajo. Agradezco las enseñanzas y la colaboración de los profesores y compañeros junto a las que seguí el programa de doctorado “Lingüística Teórica y sus aplicaciones”, entre los años 2005 y 2007, en Madrid, en la entonces llamada Fundación Ortega y Gasset. Recuerdo aquí a Olga Fernández Soriano, directora del programa, en nombre de todos ellos y en el suyo propio. Con estas personas he compartido ideas, ilusión, trabajo en equipo, buenos ratos y amistad.

Pero, aun habiendo estado rodeado de tan buenos profesores y compañeros, nunca hubiese terminado esta tesis sin el apoyo incondicional de mi familia y amigos. Les agradezco su paciencia y su comprensión. Han sabido entender mis ausencias y mi cansancio. Esta tesis debe, además, mucho a mis padres, que siempre han confiado en mí, me han animado a superarme y a perseguir mis sueños. Por último, quiero hacer una mención especial a

mis hijos (Juan y Teresa) y a mi mujer, Ana. Ella ha sido el testigo más cercano de mi trabajo. Me ha apoyado en todo momento, me ha ayudado a encontrar (a costa de su esfuerzo) el tiempo necesario y me ha dado el ánimo y la ilusión que he necesitado.

Summary

This thesis paper will present *SSG* (Simplified Spanish Grammar), an HPSG Spanish Grammar. *SSG* is a computational grammar useful for Natural Language Processing tasks. In its current development state, *SSG* is not the largest Spanish computational grammar in literature. Therefore, why is *SSG* a valuable scientific work? The reason lays in the fact that *SSG* features innovative analysis that helps to solve certain central problems of traditional computational grammars.

It is a fact that Computational Linguistics has not yet achieved a good emulation of human linguistic skills. From the middle of the twentieth century and despite discrepancies among different branches, linguistics has developed a common background of knowledge regarding the structure of human language. Nowadays, linguists know that every grammar of a natural language must face several well-known challenging phenomena: phrase structure, agreement, movement, binding, and so on. So what is the feature of the human language that makes it so complex? Why is it so difficult to handle natural language with an artificial processor?

Every phenomenon handled by a science must be formalized and codified by means of a formal language. In a specific computational sense, certain formal languages are more complex than others. This means that in order to form these languages, the grammar has to be more complex. The simpler the grammar, the easier it is to handle it with a computational device. Therefore, it is assumed that the human language structure has to be formalized

by means of a formal language which is too complex to be handled by an artificial device (a computer).

Noam Chomsky, in his first works, established a hierarchy of four possible kinds of formal languages, according to their computational complexity. These types of languages are sorted out according to their increasing order of complexity. Every possible formal language has to fall into one of these four categories. This hierarchy has been accepted as a right description, both in Linguistics and Computational Science. However, a very important question arises from it: What should be the formal language used to formalize natural language? In Linguistics, unlike other sciences, this happens to be a relevant question not only from a methodological point of view, but also from an empirical point of view. Natural language, unlike other phenomena, must be used in a specific way by human brains. Therefore, it matters in which specific way natural language is codified and used by human brains. In Computational Linguistics this question is crucial as well: the more complex the language, the greater the computational cost of its corresponding grammar. This means that an artificial device cannot deal with a great computational cost in a reasonable time.

Chomsky's hierarchy distinguishes among four kinds of languages. The first kind is made of the so-called regular languages. The sentences of these languages are one-dimensional strings: a linear sequence of items. It is plain that the sentences of human language are not one-dimensional strings, because, besides their linear nature, they are arranged according to phrasal relations. Hence, human language should not be formalized as a regular language.

The second kind of languages is the so-called context-free languages. The sentences of a context-free language are two-dimensional objects. As a matter of fact, the sentences of human languages are usually understood as two-dimensional objects: their items are arranged according to both linear and phrasal relations. In this case, human language can be formalized within a

context-free language, unless their sentences feature more complex phenomena.

The third set of languages in the hierarchy is context-sensitive languages. The sentences of these languages are multi-dimensional. In a sense, human language sentences can be conceived as multi-dimensional objects. Notice that agreement, binding and movement relations are non-linear relations established between items in different phrases.

There are different opinions in literature about this topic. For example, what kind of formal language should be used by linguists? Transformational Grammar considers sentences as complex objects with nested phrases, where relations can be established directly between different items in different phrases. Other generative models, such as Unification Grammars, and, specifically, HPSG (the model of SSG) are intended to be more plausible linguistic models (at least, from a computational point of view). Unification Grammars conceive sentences (any phrase, actually) as atomic objects that lack their constituents. According to this explanation, Transformational Grammar is a context-sensitive grammar, and HPSG a context-free grammar. According to this explanation, linguists would rather use HPSG than Transformational Grammar, because of its simpler computational nature. But is HPSG able to handle challenging phenomena such as agreement, movement or binding? In HPSG, any phrase or word is conceived as a feature structure. These feature structures are highly formalized descriptions of the nature of a word or phrase. Hence, in HPSG, a phrase lacks its constituents, but certain relevant information of these constituents can be saved in its feature structure by means of the feature-sharing. In a feature structure two or more features are allowed to share the value, so no matter what their value is, it has to be exactly the same. In HPSG, when a phrase is formed, relevant feature sharing relations are established between the resultant phrase and their constituents. By means of this device, it is possible to establish in HPSG relations like agreement, binding or movement.

It seems that every attempt to formalize human language has to face a great complexity, and, until now, Computational Linguistics has failed to provide a general solution to the problem. However, HPSG, which is the most popular grammatical model in Computational Linguistics, is supposed to be a plausible and efficient solution to the problem. What is exactly the problem? A Transformational Grammar conceives sentences as *trees of constituents* where non-local relations can be directly established. In these *trees* it is possible to identify a nested pronoun as the target of a binding or agreement relation, because the actual pronoun is in the structure, in the *tree*. It is also possible to find a nested NP or DP to move it to the left periphery of the sentence. These processes are very complex (context-sensitive processes). Nevertheless, this kind of process does not trigger ambiguity. Transformational Grammar does not need to blindly generate all possible behavioral patterns of words in the first stages of analysis. That is due to the fact that they are allowed to always remember the constituent structure and to change it in certain ways. In fact, they can always build the canonical version of words and wait for evidence to change this canonical structure only when needed.

On the other hand, in HPSG all actual processes are local. Even non-local relations are handled by means of a chain of local processes. These local processes are definitely less complex than the process of Transformational Grammar as they are in fact context-free processes. Nonetheless, they too trigger ambiguity in non-trivial situations. As long as grammar is not allowed to remember the constituents' structures, it cannot change these structures when needed. The only alternative is to generate all possible behavioral patterns of every word in a sentence in the first stages and then try all possible combinations. In fact, the result in non-trivial cases is a combinational explosion of hypothetical behavioral patterns.

For example, certain Spanish verbs such as *subir* have a transitive version, an intransitive version, a passive version and a passive version with *se*. All these patterns are compatible with a directional complement controlled

by the inner argument of the verb. Subsequently, all these resultant patterns are compatible with a dative complement. At the same time, these different complements may be involved in a cliticization process and different word order phenomena (scrambling, wh-movement, and so on). If a grammar deals with all these resultant patterns by means of different lexical rules, it will generate a huge amount of lexical patterns involved in the analysis of every sentence. These lexical patterns are the input to the merging of constituents. Therefore, a grammar with that system of lexical rules will lead to expensive and slow analysis processes.

This thesis aims to develop the core of an HPSG grammar of Spanish with a really small amount of lexical rules, which has been named Simplified Spanish Grammar (SSG). It is claimed that SSG analysis are elegant and theoretically motivated, and such analysis significantly reduces the computational cost of grammar. Three main groups of central phenomena in Spanish have been implemented in SSG.

The first phenomenon is diathesis alternations. From a computational point of view, this is one of the most challenging phenomena in natural languages as verbs can usually behave in different ways. They may have both active and passive versions and they may accept certain optional complements among other things. HPSG lexical rules are meant to deal with these alternations. In computational grammars, however, the use of lexical rules should be improved in specific ways.

Many verbs accept different kinds of phrases in a specific argumental position. For example, the inner argument of certain verbs may be either a nominal phrase or a verbal phrase. Computational grammars usually handle the same alternations tying different lexical units to the same verb. This implementation, however, fails to capture an important generalization. Moreover, many verbs have both active and passive versions or can feature other constructions, such as passive with *se* or pronominal construction. These verbs may also be used with optional complements, such as controlled structures,

datives, obliq complements and so on. If these criteria were to be crossed, they would result in a bunch of many possible patterns that a verb is allowed to use. Traditional computational grammars usually deal with this diversity by means of different lexical rules. They implement as many specific lexical rules and they are regarded as crossed patterns. According to this approach, there are specialized lexical rules or lexical units to: transitive verbs with nominal object, transitive verbs with nominal object and dative, transitive verbs with clausal object, transitive verbs with clausal object and dative, and so on. This traditional approach fails to capture due generalizations. Every grammatical reality (transitivity, passive, and a certain kind of dative complement) should be implemented just once. A verb happens to be transitive if it is an instance of the unique transitive pattern of the grammar; a verb has a certain dative complement only if it is an instance of the unique pattern that adds such a complement, and so on. Moreover, argumental positions can be filled with different types of phrases, which mean that both clausal and nominal objects should be considered different fillers available to the same argumental position in the same pattern.

This thesis paper develops a system in which every intuitive verbal pattern is implemented with a unique lexical rule. This system is claimed to be very simple, linguistically motivated, able to capture interesting generalizations and computationally more efficient than previous solutions.

The second central grammatical phenomenon implemented in SSG is the Spanish clitics system. The nature of clitics has been broadly discussed in literature. Regardless of whether they are considered a morphological or a syntactic phenomenon, clitization in HPSG has always been formalized by means of lexical rules. By following this approach, many lexical rules and clitization patterns can be added to grammar, which can become a great source of complexity in computational grammars. In Spanish, both accusative and dative arguments can suffer clitization. Moreover, depending on the context, a clitic can appear instead of its canonical object or beside it. Hence, Spanish grammars need a minimum of four lexical rules to handle clitization. Current

syntactic approaches use at least an extra specific rule to merge clitics and verb. Therefore, this thesis develops an analysis of clitics that avoids using any rule or lexical unit intended to deal with clitics. As previously stated this analysis is simple, linguistically motivated and reduces the combination-al explosion and the computational cost of the grammar.

The last grammatical phenomenon implemented in an innovative way in SSG can be related to word order. The possibilities of word order are a great source of complexity in every Spanish computational grammar. First of all, canonical preverbal subjects can be inverted in several contexts. That inversion has been implemented in traditional HPSG grammars by means of a lexical rule, which leads to a bigger combinational explosion of patterns. At the same time, post-verbal complements can switch their canonical positions, maybe only in a specific context, with certain intonation patterns and with different informational purposes. Finally, Spanish grammar must face well-known movement phenomena such as wh-movement, relative clauses or topicalization. These phenomena, according to their non-local nature, are a great source of complexity. Spanish data suggest that topicalization and the rest of phenomena are due to different grammatical processes. Therefore, both types of processes should be faced by means of different rules or lexical units. SSG is not a final solution to these tricky problems. However, this thesis is an interesting contribution because it proposes an innovative analysis of subject linearization in regards to word order. This proposal is plausible in a theoretical way and contributes to reduce the combinational explosion of grammar. At the same time, in SSG, post-verbal linearization of complements is implemented according to the classical Linearization Theory in HPSG. This application of the theoretical approach to a computational solution in Spanish is new and it is claimed to reduce also the grammar complexity and its computational cost. Finally, this thesis cannot propose a solution to classical non-bounded movements. These data have been exposed here because they support the analysis of clitics in SSG.

In the next paragraphs it will be explained how SSG can reach these goals.

Firstly, how does SSG shape diathesis alternation in Spanish? In this thesis it has been coined the term non-destructive rule. Usually, in HPSG, lexical rules are destructive and all verbs are supposed to have a canonical characterization. At the same time, lexical rules are intended to change that canonical pattern into another. These rules destruct a feature structure and create another one. Crucially, input and output are not supposed to be necessarily compatible. The result is that an HPSG rule is able to change its input in almost every way: it can add or remove an argument, change its category, its case, its position and so on. The previous paragraphs have shown that using lexical rules to achieve almost every type of change in verbal behavior is a common practice in HPSG. Unlike previous grammars, SSG avoids using this rules as lexical rules used by SSG are non-destructive rules. Non-destructive rules never change their input structure, they only specify them. In a non-destructive rule, input and output must share their feature structure and both structures must be identical. Those rules take an underspecified verb and specify it by adding information compatible with their original characterization. The non-destructive rule system is easier to implement and maintain than a traditional system. This thesis claims that this approach has theoretical significance. Every science aims to explain as much data as possible with a theoretical system in the simplest way possible. For example, early stages of Transformational Grammar achieved wide explanations of linguistic data by means of varied, language-specific and construction-specific transformational rules. This powerful system of rules lacked explanatory capacity, because it could operate almost every conceivable change in input. Hence, in the next stages of Transformational Grammar, universal principles were established and they drastically restricted the nature of conceivable changes in constituents' structures. The old language and specific construction rules were substituted by very few and simple universal principles, which turned out to be a huge methodological improvement. In a computational sense, early transformations and HPSG lexical rules are absolutely different. However, just like transformations, lexical rules can operate almost every conceivable change in input and this power reduces HPSG's explanatory capacity. This the-

sis claims that a non-destructive lexical rules system can entirely solve this problem. All non-destructive rules can be reduced, in fact, to a single universal operation: specification, application of an independently-legitimated behavioral pattern.

Can this restricted system face the great complexity of diathesis alternation in Spanish? This system must carefully balance the original characterization of lexical units. However, the main problem is that characterization will never be changed by lexical rules, but it will only be specified. The lexical unit must be under-specified, because it has to be compatible with very different behavioral patterns. At the same time, it has to be specific enough to avoid over-generation. Finding this perfect balance is the real challenge.

The solution in SSG is to deduce syntactic behavior of verbs from their semantic characterization. Verbs are really under-specified in a syntactic sense, but they feature a rich semantic characterization. It has been assumed that syntactic alternatives share a common semantic background. A classic semantic characterization has been used: verbs can be accomplishments, achievements, activities or states. According to this main classification, the semantic feature structure of verbs informs about the possible presence of an external argument, an inner argument, and the ability of the verb to receive a certain kind of dative complements or certain controlled predicates. Verbs are also crucially characterized by relevant syntactic features: their ability to assign accusative case or government idiosyncrasies. All these features are well-known verbal characterization criteria, so it is safe to say that they are natural and linguistically motivated. The interesting point is that, just by means of a system of several simple, classic notions, it is possible to develop a general grammar of diathesis alternations of Spanish verbs in a non-destructive fashion. SSG has only nine lexical rules that depict the main verbal patterns in Spanish. There is a lexical rule in charge of all transitive uses, another rule is in charge of passive uses, and another one deals with passives with *se*. Moreover, there are three rules that handle intransitive uses: unergative, unaccusative and pronominal uses. According to their

underlying semantic characterization, verbs are able to be featured by some of those rules. Furthermore, there are three rules that add extra arguments to the original characterization: a controlled predicate, an obliq argument or a dative argument. SSG features an innovative system that harnesses those rules to be applied to different inputs, regardless of their original length. All these rules restring the nature of their arguments in an interesting way. SSG has a general description of the general notion of argument and it also has a description of case: nominative, accusative, dative and obliq cases. The confluence of all these notions, as well as several semantic idiosyncrasies of certain verbs, successfully regulates the nature of the fillers of every argumental position.

Secondly, how does SSG shape clitization in Spanish? In traditional Spanish grammars previous complexity has to be combined with a bunch of lexical rules specifically devised to handle clitization of accusative and dative arguments. SSG avoids using all these rules. Moreover, in SSG clitics are verbal affixes. Thanks to this morphological approach, SSG avoids using a grammatical rule to merge clitics and verb. In SSG, clitics information is added to the verb by means of an inflectional rule. Note that inflectional rules do not trigger combinational explosion, because they are applied separately and only if pre-syntactic analysis (tokenization) has found actual clitics in the verb.

In SSG, clitics are not considered fillers available to an argumental position. Rather, they are only the morphological mark that certain words have left in the verb when they have filled their accusative or dative position. These words are personal pronouns, elliptic pronouns and traces left in topicalization processes. This thesis claims that these words exist in grammar independently of clitics. The outcome is a system of clitics that does not add complexity to the grammar. This analysis has been formalized by means of techniques broadly used in HPSG to handle non-local processes. In fact, clitics can suffer non-local processes, called clitic climbing. SSG analysis of clitics handles clitic climbing using non-local techniques in HPSG.

This analysis of clitics does not add computational complexity to the grammar. It is simple and natural and has been formalized by means of pre-existent devices.

Finally, SSG features innovative analysis of Spanish word order.

In Spanish, subjects are typically pre-verbal arguments. However, there are contexts in which they have to (or can) appear after the verb: the inversion process. According to typical HPSG formalization, this is a source of complexity and ambiguity. Inversion is usually handled by means of a lexical rule. The output of this rule combines with all other verbal patterns, and it results an increase of combinational explosion and complexity. Grammar with canonical preverbal subjects features a systematic ambiguity between local and topicalized subjects. In order to reach a simplified and computationally efficient analysis of subject linearization, SSG regards subjects as originally post-verbal arguments where pre-verbal subjects are the result of a topicalization. It is claimed that this approach is plausible in theoretical terms, it solves ambiguity (all preverbal subjects are topics) and reduces the computational cost of grammar.

Post-verbal complements in Spanish can be sorted in many ways. Accusative and dative complements can exchange their positions without any semantic or informational implication. This phenomenon can be understood as a scrambling process and it has been handled by traditional computational grammars in different ways. All of them are really expensive in a computational sense: different orders are handled by means of specific lexical or grammatical rules. However, HPSG has developed a whole Linearization Theory whose explanation of scrambling leads to a great simplification of grammar. The key idea is to use discontinuous constituents and all arguments are always listed in the same order in the verb. However, imagine that a parser is able to merge two constituents no matter if they are adjacent. In that case, all these arguments, which are always listed in the same order, can be found in different relative positions. This approach has not been applied

to traditional computational grammars because traditional parsers cannot deal with this kind of discontinuous constituents. In this thesis paper, it has been implemented a parser able to do that. For this reason, *SSG* does not need any rule to deal with scrambling as all complements are always listed in the verb according to a unique increasing order of obliquity.

After the verb, obliq complements can also appear as controlled predicates and subjects. All these post-verbal arguments can be sorted in different ways, but those possibilities involve specific intonational patterns and informational meanings. Hence, unlike accusative and dative alternations, these word order possibilities are not meant as scrambling. However, according to Linearization Theory, all these implications can be handled as extra conditions to the creation of certain discontinuous constituents. As a consequence, these word orders are explained by means of the same concept of discontinuous constituent, and only some orders are tied to extra conditions. In this thesis, Spanish post-verbal word order possibilities have been explained according to the standard Linearization Theory in HPSG. Therefore, *SSG* does not feature any kind of extra complexity to handle all these possibilities, which are handled by a parser able to build both continuous and discontinuous constituents. In principle, all complements after the verb can be sorted any possible way. This is definitely true to accusative and dative complements. In order to correctly shape other alternations, some restrictions should be added to order possibilities in feature research. Hence, thanks to this approach, grammar becomes simpler and easier to maintain. However, can it be said that this approach reduces combinational explosion? It is not clear. In purely grammatical terms, this approach does not add rules to grammar, but the corresponding parser builds discontinuous constituents. This question will be answered at the end of the present summary.

The last word order phenomena regarded in this thesis are classical unbounded dependencies, such as *wh*-movement and topicalization. Both *wh*-movement and topicalization have been implemented in the current version of *SSG*. Their analysis is not innovative, but they have been implemented and

documented because they are related to the clitics system. Notice that these movements left traces in their base position. In fact, these traces fill argumental positions. According to the analysis of clitics in SSG, and like all words, a trace, in an accusative or dative position may trigger the appearance of a clitic. According to Spanish data, topicalization traces must trigger the appearance of a clitic in those positions, and wh-movement traces must not (at least in accusative positions). Therefore, SSG considers two different traces: the first one can trigger cliticization, the second one cannot. This distinction is crucial to SSG analysis of clitics, but it is claimed that it is needed independently of clitics. Topicalization and wh-movement are different processes. For example, a sentence admits multiple topics, but just one wh-particle movement; in the left periphery of a sentence, topics must precede the wh-particle; topics and wh-particles can be moved from different places. In HPSG, it is not possible to formalize these movements with the same rules and lists because every movement is triggered by a different rule, so every rule can use a different trace.

Some SSG analyses have already appeared in theoretical versions of HPSG. However, they have never been applied to computational grammars of Spanish. The reason is that these analyses need technical support that current parsers hardly ever provide. For example, morphological analyses of clitics are popular in HPSG. However, in Spanish, clitics (proclitics, to be precise) and verbs are written as different words. As a result, a grammar whose clitics are affix needs a device able to join clitics and verb before the syntactic analysis. Hence, in order to handle scrambling and similar phenomena by means of the notion of discontinuous constituents, a parser has to be able to merge two constituents regardless of whether they are adjacent. In principle, such a parser can merge two constituents just when they do not overlap. In fact, during the development of SSG, SGP (Simplified Grammars Parser) has been developed. SGP is a bunch of libraries written in Perl with access to a databases server MySQL. SGP provides all the needed tools to analyze written text with HPSG grammars. Moreover, it provides all the needed tools to analyze with an original grammar as SSG and a library that joins clitics and verb, as

well as a parser compatible with discontinuous constituents.

To sum up, this summary aimed at explaining SSG, specifically three groups of phenomena in particular: diathesis alternation, clitics and word order. SSG provides innovative analysis for all these phenomena. It is claimed that these analysis are simple and linguistically motivated. Thanks to them, SSG is simpler than previous grammars, more scalable and easier to maintain. This analysis significantly reduces the computational cost of grammars. This last claim has been proved true by means of a compared analysis of the same test suit both with SSG and NSSG (Non Simplified Spanish Grammar). NSSG is a traditional grammar whose analysis of diathesis alternation, clitics and word order use the traditional lexical rules. Specifically, NSSG is SSG with eleven extra rules. NSSG uses a version of the same parser that does not allow discontinuous constituents. The result of the compared analysis is that SSG is absolutely faster than NSSG. This is a really predictable outcome, because SSG has achieved analysis able to significantly reduce the number of lexical rules in the grammar. At the same time, in regards to discontinuous constituents, this is not a trivial conclusion because discontinuous constituents are a source of computational complexity. This compared analysis proves that discontinuous constituents should be preferred to the combinational explosion of lexical rules.

Introducción

Este trabajo presenta *SSG* —*Simplified Spanish Grammar*—, una gramática del español creada dentro del marco teórico *HPSG* —*Head-Driven Phrase Structure Grammar*—. *SSG* es una gramática computacional, pensada para su uso en labores de procesamiento automático del lenguaje natural. Se ha implementado usando el lenguaje *TDL* —*Type Description Language*, definido en Copestake (2002)—.

Existe ya en la actualidad una gramática de este tipo que cubre un gran número de fenómenos del español: *SRG* —*Spanish Resource Grammar*¹—. *SSG* no pretende competir en cuanto a cobertura con *SRG*. En cambio, pretende solucionar ciertos problemas existentes en las gramáticas computacionales actuales; en concreto, ciertos problemas relevantes para el tratamiento del español. Las gramáticas computacionales actuales, según ganan cobertura, ganan también complejidad. Su uso real en labores de análisis profundo suele acarrear tiempos de procesamiento inasumibles. Un analizador automático, con una gramática de este tipo, debe calcular y recordar muchas hipótesis alternativas de análisis parcial. Una unidad atómica de la secuencia de entrada admite muchos patrones de realización. A partir de esta ambigüedad inicial, el proceso de combinación sintáctica genera también una gran ambigüedad estructural.

Todo ello, en usos no triviales de las gramáticas, genera una gran explosión combinatoria, con un coste de memoria y de tiempo de procesamiento

¹Marimon (2013)

inasumible. La solución a este problema es la gran encrucijada en la que se encuentra en la actualidad la Lingüística Computacional. Seguramente, dicha solución no la proporcionará un análisis propiamente lingüístico. La investigación sobre el lenguaje natural proporciona datos que suscitan reflexiones en campos como la Biología o la Inteligencia Artificial: ¿Cómo funciona el cerebro humano para dar lugar al procesamiento lingüístico en tiempo real? ¿Cómo es posible emular ese funcionamiento, de forma eficiente, mediante un procesador artificial? El análisis propiamente lingüístico ayuda a entender la complejidad estructural del lenguaje humano y la complejidad computacional que requiere su procesamiento. Pero estos datos suscitan dilemas de corte computacional cuya solución, seguramente, quede fuera de las posibilidades de la Lingüística misma.

No obstante, sí es necesaria una revisión crítica de los análisis lingüísticos habituales en las gramáticas computacionales. Esta revisión debe valorar el coste computacional de estos análisis; y debe buscar alternativas, lingüísticamente motivadas, que reduzcan la explosión combinatoria, y con ella el coste de procesamiento. SSG es el fruto de esta revisión para las gramáticas de tipo HPSG; en concreto, para las del español. En definitiva, SSG quiere ser una gramática del español más sencilla que las anteriores, que permita un coste computacional reducido; de ahí su nombre: *Simplified Spanish Grammar*.

Este trabajo documenta tres grandes aspectos de la gramática del español, a los que SSG da un tratamiento novedoso. En primer lugar, da cuenta de las alternancias de diátesis de los verbos de esta lengua. Cada verbo, normalmente, puede participar de patrones de comportamiento diferentes: versión transitiva, media, pasiva, pasiva refleja; con añadido o no de un dativo; etc. Además, es posible que la misma posición argumental de un verbo sea ocupada por elementos de distinta categoría gramatical: nombres, complementantes, verbos; marcados o no con preposición. SSG, para regular estas alternancias, utiliza solo un pequeño sistema de reglas léxicas. Esta sencillez simplifica la implementación, y reduce el coste computacional que asumían las gramáticas previas. Además, es un sistema conceptualmente novedoso

que restringe el concepto de regla léxica de HPSG, y aporta soluciones técnicas útiles.

En segundo lugar, SSG cuenta con un análisis novedoso de los clíticos del español. En HPSG, hasta el momento, se han utilizado reglas léxicas para modelar los procesos de clitización. SSG da cuenta de los mismos datos sin necesidad de uso de regla alguna. En cambio, en esta gramática, son ciertas unidades léxicas (pronombres y huellas) las que, cuando se unen al verbo como argumentos, desencadenan la aparición del clítico en este. Como se verá, tampoco se utilizan para ello unidades léxicas cuya existencia no esté justificada de forma independiente. Este análisis, igual que ocurriera con el de las alternancias de diátesis, al prescindir del uso de reglas, supone también una reducción del coste computacional de la gramática.

Por último, el español es una lengua con orden de palabras relativamente libre. Las gramáticas computacionales de tipo HPSG no manejan de forma satisfactoria esta libertad, y dan soluciones para ella de alto coste computacional. En este estudio se hace una reflexión al respecto, y se proponen análisis con reducido número de reglas; análisis, por tanto, que reducen la explosión combinatoria de las propuestas previas. En concreto, SSG cuenta con análisis novedosos para la inversión de sujeto, para el *scrambling* de complementos verbales en posición posnuclear, y para los casos más importantes de dependencias no acotadas.

En resumen, SSG es una gramática novedosa del español. Gracias a su motivación lingüística, es una gramática sencilla, con una implementación clara y coherente, fácil de ser ampliada con nuevos fenómenos. Además, reduce, con respecto a las propuestas previas, el número de reglas necesario para dar cuenta de las alternancias de diátesis, los fenómenos de clitización, y los órdenes de palabras. Con ello, SSG reduce la explosión combinatoria y el coste computacional.

Es importante definir cuál es el ámbito científico-técnico al que pertenece

este trabajo. Este no es, propiamente, un estudio lingüístico teórico. Sin embargo, sí se ha querido dotar a SSG de una cierta fundamentación teórica que le aporta coherencia interna, sencillez y elegancia conceptual. Además, su implementación ha ido acompañada de una reflexión sobre la complejidad computacional de los fenómenos tratados; sobre la complejidad computacional, en definitiva, de las lenguas naturales. Esta reflexión solo tiene sentido si los fenómenos en cuestión se han tratado, de un modo u otro, de acuerdo con las características irrenunciables del lenguaje humano. Pero, a pesar de ello, SSG es, fundamentalmente, una herramienta que ha de ser útil para su aplicación práctica. Por ello, la elección de uno u otro análisis ha estado condicionada por criterios de sencillez, rentabilidad del gasto computacional y ahorro de coste de procesamiento. Estos criterios no son siempre fáciles de conciliar con la fundamentación teórica antes mencionada. En aras de esta vocación aplicada, los análisis de SSG, en ocasiones, no se preocupan de cubrir datos problemáticos marginales. También en ocasiones se opta por análisis, siempre plausibles desde un punto de vista teórico, pero pendientes de demostración, o controvertidos entre los especialistas.

Queda fuera de las posibilidades y del interés de este trabajo hacer una defensa teórica sólida de cada uno de estos análisis. SSG pretende ser el germen de una solución computacional completa de la gramática del español: es fruto del engarce de un análisis diferente para cada uno de los muy diferentes fenómenos tratados. Este enfoque amplio no puede cubrir una demostración teórica de cada uno de estos fenómenos. Su fin es, en cambio, demostrar que es posible crear una gramática general del español, fundamentada en análisis teóricos plausibles, que reduzca sensiblemente la explosión combinatoria de las gramáticas anteriores.

Además, SSG, en su actual estado de desarrollo, es un trabajo de macroestructura. Pretende ser una solución adecuada para los grandes problemas que presenta la gramática del español. Pero queda por implementar y documentar la microestructura de la gramática. Por ejemplo, SSG cuenta, como se ha dicho, con un análisis de las alternancias de diátesis de los verbos del español:

un sistema que define las posibilidades de realización de los distintos tipos de verbos de esta lengua. Pero falta definir una ontología exhaustiva de clases verbales, con la debida adscripción de cada verbo a su clase. Esta limitación tiene relación con el valor de SSG como investigación novedosa y útil. Como ya se ha dicho, existe en la actualidad una gramática del español —SRG— que cubre un muy amplio abanico de fenómenos. SSG no sería, entonces, interesante por el hecho de volver a modelar estos fenómenos. El valor de SSG es, en cambio, su planteamiento general, la macroestructura. Esta sí es novedosa y útil dentro del campo de la implementación de gramáticas computacionales.

Por último, en varias ocasiones, las soluciones que da SSG a ciertos fenómenos son clásicas desde un punto de vista teórico, pero novedosas desde un punto de vista aplicado. Se observa, en las gramáticas aplicadas, un cierto alejamiento de algunas opciones teóricas preferidas. Es posible que este alejamiento se deba a falta de recursos técnicos. Los *parsers* utilizados habitualmente —el del entorno de desarrollo de gramáticas LKB: *Linguistic Knowledge Building*²— no están equipados para la utilización de dichas soluciones clásicas. Por ejemplo, es frecuente que los estudios de HPSG entiendan a los clíticos como afijos verbales, como realidades morfológicas. En cambio, las gramáticas aplicadas actuales los suelen entender como pronombres, como entes sintácticos autónomos. Téngase en cuenta que, en la escritura en español, el verbo y sus clíticos antepuestos —proclíticos— se escriben separados. Los *parsers* de texto escrito habituales entienden como unidad sintáctica atómica, en principio, todo aquello que esté separado por espacio en blanco. Por tanto, a falta de otros recursos técnicos pertinentes, estos *parsers* solo son compatibles con la interpretación sintactista del fenómeno de los clíticos. Del mismo modo, la Teoría de la Linealización clásica de HPSG contempla la existencia de constituyentes discontinuos en ciertas lenguas. Esta noción es la clave de la explicación en este modelo del fenómeno de *scrambling* de lenguas como el alemán o el español. Pero los *parsers* de uso habitual no contemplan la existencia de este tipo de constituyentes.

²Copestake (2002).

Precisamente, en SSG, los clíticos se conciben como afijos verbales; y los casos de *scrambling* se formalizan haciendo uso del concepto de constituyente discontinuo. Por tanto, SSG es una gramática computacional novedosa, en tanto que utiliza los análisis teóricos clásicos de HPSG. Para ello, en este trabajo se ha desarrollado, junto a SSG, SGP: *Simplified Grammars Parser*. SGP es un *parser* implementado en Perl que admite la existencia de constituyentes discontinuos. Además, cuenta con un módulo de preprocesamiento que une en una sola entidad al verbo y sus clíticos.

El trabajo que sigue se divide en cinco partes. En el capítulo 1 se explica cuál es el marco teórico en el que se ha desarrollado SSG. Se explicará qué es una gramática generativa, qué tipos de gramáticas existen y cuál es su complejidad computacional. Se explicará también qué es una gramática de unificación; y, en concreto, se explicará HPSG, modelo en el que se ha desarrollado SSG. En el capítulo 2, se explican ciertos problemas de corte computacional y teórico de HPSG, para los cuales SSG es una solución. En los tres capítulos siguientes se explica SSG propiamente dicha. En el capítulo 3, se explica el sistema de reglas léxicas mediante el cual SSG modela las alternancias de diátesis de los verbos del español. En el capítulo 4 se explica la solución que aporta esta gramática al fenómeno de los clíticos del español. Por último, en el capítulo 5, se explica el análisis que hace SSG del orden de palabras del español.

Además, el trabajo incluye dos apéndices. Por un lado, el apéndice A documenta el *parser* SGP. Por otro, el apéndice B presenta una *test suite* que recoge los fenómenos del español tratados en los capítulos 3, 4 y 5. Se muestran en él los resultados del análisis de esta *test suite* para el *parser* SGP, de acuerdo con la gramática SSG y de acuerdo con una gramática más tradicional. Se verá cómo con SSG los tiempos de análisis son sensiblemente menores.

Capítulo 1

Marco teórico

1.1. La Lingüística Generativa

En Chomsky (1957), una lengua natural cualquiera (el español, por ejemplo) se entiende como un conjunto infinito de oraciones de longitud finita, construidas todas ellas a partir de un conjunto finito de elementos. La tarea fundamental del lingüista que estudia una lengua natural consiste en crear una gramática de dicha lengua: un ingenio que determine qué secuencias de elementos forman parte de tal lengua y qué secuencias no. Cabe preguntarse, entonces, ¿qué es una gramática? ¿qué tipo de ingenio entiende Chomsky que permitiría discriminar qué es y qué no es una oración de una lengua natural dada?

Con los primeros trabajos de Chomsky, comenzó a plantearse la necesidad de representar al lenguaje natural, en tanto que objeto de estudio científico, mediante un lenguaje formal¹: un conjunto de teoremas derivables a partir de unos axiomas mediante un conjunto finito de reglas. Pues bien, una gramática es este conjunto de axiomas y reglas. De acuerdo con Chomsky

¹Posteriormente, la escuela chomskyana ha relajado la necesidad de formalización en el estudio del lenguaje. Pero otras escuelas, también generativas, hijas del trabajo original de Chomsky (Gramáticas de Unificación, Gramática Categorial...), siguen una formalización rigurosa. Véanse Shieber (1986) o Moreno Sandoval (2001) para las Gramáticas de Unificación, y Carpenter (1995) para la Gramática Categorial.

(2003)², en un lenguaje formal al uso (la aritmética, por ejemplo) el sistema de axiomas y reglas son meramente un modo de describir el lenguaje. El lenguaje mismo son los teoremas. Son estos los únicos que representan una parcela de la realidad. Los teoremas pueden representarse por medio de muy distintos sistemas de axiomas y reglas. Y, en principio, uno de estos sistemas no es preferible a otro. No cabe preguntarse, entonces, qué sistema de axiomas y reglas para derivar los teoremas de la aritmética son los verdaderos. Todos son sistemas válidos de representación de tales teoremas. En cambio, el lenguaje natural, entendido también como sistema formal, es diferente. El cerebro humano debe tener un sistema de cálculo que permita discriminar qué es una oración de una lengua natural y qué no. Y ese sistema de cálculo ha de ser de un modo concreto. Por tanto, en Lingüística sí es pertinente preguntarse qué sistema de axiomas y reglas derivativas es el que representa ese sistema de cálculo cerebral. Es pertinente, pues el cerebro ha de usar un sistema y no otro.

De cualquier modo, el problema central de la Lingüística Generativa, en cuanto a una lengua natural concreta, consiste en establecer la gramática que permite generar las oraciones de esta lengua. En términos formales, consiste en establecer el conjunto de reglas que, a partir de unos axiomas, generan todos los (y solo los) teoremas de la lengua en cuestión.

1.1.1. Lenguajes formales y Teoría de la Computación

Hasta la aparición de Chomsky (1957), el estudio del lenguaje hacía taxonomía de palabras, construcciones y fenómenos gramaticales. Pero en ningún momento se planteó la posibilidad de entender, como hace la Lingüística Generativa, los mecanismos que permiten generar las oraciones de una lengua. Fue necesaria para la aparición de este nuevo enfoque el desarrollo previo de ciertas ramas de la Matemática.

Estas fueron la Teoría de Lenguajes Formales y la Teoría de la Com-

²Chomsky (2003), p. 106.

putación. Los lenguajes formales son artificios creados para representar el objeto de estudio de una ciencia. Todo hecho que pueda contemplar una ciencia debe poder representarse en el lenguaje formal en el que esta se exprese. Si este hecho se concibe como verdadero, debe ser un teorema del lenguaje formal, es decir, debe ser demostrable dentro de tal lenguaje. En cambio, si se concibe como falso, debe no ser un teorema: no debe ser demostrable dentro del lenguaje. Pero ¿qué quiere decir que un enunciado sea o no demostrable dentro de un lenguaje formal? Un enunciado queda demostrado dentro de un lenguaje formal cuando es posible derivarlo a partir de los axiomas de dicho lenguaje, mediante la aplicación sucesiva de las reglas de inferencia de este. Un axioma es un enunciado del lenguaje que se entiende como teorema sin necesidad de demostración: se asume su veracidad. Por su parte, las reglas de inferencia son teoremas que permiten deducir otros teoremas a partir de ciertas premisas (teoremas previamente demostrados o axiomas). Entonces, demostrar un enunciado, demostrar que es un teorema, miembro del lenguaje formal, consiste en crear tal enunciado mediante la aplicación de las reglas del lenguaje a partir de sus axiomas. Dicho de otro modo, el lenguaje formal (el conjunto de sus teoremas) se describe, no de modo extensional (mediante el listado uno a uno de dichos teoremas), sino mediante el establecimiento de unos axiomas y reglas a partir de los cuales se pueden generar todos los (y solo los) teoremas.

Durante la primera mitad del siglo XX se dio un fuerte desarrollo de la Teoría de Lenguajes Formales. David Hilbert, matemático alemán, planteó en 1900 la posibilidad de demostrar todos los teoremas de la aritmética a partir de un mismo conjunto de axiomas y reglas de inferencia, de forma que tal sistema fuese totalmente coherente: no diese lugar a contradicciones³. En respuesta a ello, Bertrand Russell publicó en 1910, 1912 y 1913 sus *Prin-*

³Lo hizo en el Congreso Internacional de Matemáticos de París de 1900. Planteó como un reto para la comunidad científica veintitrés cuestiones de interés en aquel momento, la última de las cuales es la que se trata aquí. Véanse Hopcroft y otros (2001) p. 339 o Iranzo (2004) p. 72.

*incipia Mathematica*⁴, obra que definía un sistema de axiomas y reglas que debía poder demostrar todos los teoremas de la aritmética. En realidad, nadie sabía entonces si, realmente, la obra de Russell englobaba a toda la aritmética, ni si carecía de contradicciones. En 1931, Kurt Gödel, lógico y matemático austriaco, demostró que todo intento de axiomatización de la aritmética es incompleto⁵. En aquel momento, los matemáticos acostumbraban a buscar la demostración de enunciados aparentemente sencillos. Tenían la idea de que, por muy difícil que fuese dar con tales demostraciones, estas existían. Era cuestión de tiempo que alguien diese con ellas. Sin embargo, el descubrimiento de Gödel abrió la puerta a la posibilidad de que tales enunciados, u otros, fuesen intrínsecamente indemostrables. Si todo sistema formal que intente axiomatizar toda la aritmética es incompleto, un determinado enunciado puede caer en el grupo de los indemostrables. Dicho esto en otros términos, el descubrimiento de Gödel demostró que no existe un conjunto de axiomas y reglas de inferencia a partir de los cuales pueda ser demostrado como teorema o no teorema cualquier enunciado.

En esta situación, se planteaba en las matemáticas la pregunta crucial de qué se puede demostrar y qué no, qué enunciados o tipos de enunciados sí son demostrables y cuáles no, qué lenguajes formales se pueden describir mediante un sistema axiomático y cuáles no. Recuérdese que los enunciados de un lenguaje formal son la representación formalizada de problemas, hechos, estudiados por una ciencia que se expresa mediante este lenguaje formal. Por tanto, estas preguntas son equivalentes a preguntarse para qué tipo de problemas existe una solución. Como respuesta a estas preguntas surgió la Teoría de la Computación. Esta disciplina estudia qué lenguajes formales pueden describirse por medio de dispositivos computacionales. ¿Qué es un dispositivo computacional? Cualquier dispositivo concebible capaz de solucionar

⁴Véase Russell y Whitehead (1992).

⁵Gödel demostró que un sistema formal lo suficientemente complejo como para representarse a sí mismo no es completo mientras sea coherente. A esto se le conoce como Teorema de la Incompletitud de Gödel. El artículo original en alemán es Gödel (1931). La traducción en inglés es Gödel (1962).

problemas de modo algorítmico, mediante un algoritmo. Un algoritmo es un conjunto de reglas que dice paso a paso cómo proceder para conseguir un fin concreto. Por tanto, la Teoría de la Computación estudia para qué tipo de problemas existe un algoritmo que pueda solucionarlo. Para solucionar un problema mediante un algoritmo es necesario expresarlo de modo totalmente formalizado: mediante un lenguaje formal. Habrá tipos de problemas para cuya solución exista un algoritmo, otros para los cuales no; o, dicho en términos de lenguajes formales, habrá lenguajes que podrán describirse mediante un conjunto de axiomas y reglas, y otros que no.

Además, los lenguajes formales que sí cuentan con una posible descripción mediante un algoritmo pueden clasificarse en función de su complejidad computacional; es decir, en función de lo complejos que sean los algoritmos necesarios para la descripción. Chomsky⁶ estableció una jerarquía de posibles lenguajes en función de dicha complejidad computacional. Esta jerarquía se expone en el siguiente apartado, 1.1.2.

Recuérdese que se decía al comienzo de este apartado 1.1.1 que para el desarrollo de la Lingüística Generativa fue necesario un desarrollo previo de las teorías de Lenguajes Formales y de la Computación. Efectivamente, el lingüista generativo no describe una lengua haciendo una taxonomía de los tipos de oraciones de esta, sino creando una gramática: un conjunto de axiomas y reglas a partir de los cuales se derivan todas las (y solo las) oraciones de la lengua. Las teorías de Lenguajes Formales y de la Computación proporcionaron, como se ha visto, un fundamento matemático adecuado para ello. Como ya se ha dicho, en Lingüística se da la situación, extraña en otras ciencias, de que el conjunto de axiomas y reglas mediante el cual se describe un lenguaje formal no es una mera estrategia descriptiva de tal lenguaje, sino que este conjunto es objeto de estudio en sí mismo. El cerebro humano se concibe como ese dispositivo computacional que, mediante un algoritmo, genera las posibles oraciones de una lengua natural, o entiende una secuencia

⁶Véase, por ejemplo, Chomsky (1956).

de elementos como oración de esa lengua. Por tanto, ¿cuál es ese algoritmo?, ¿qué conjunto de axiomas y reglas son las pertinentes para describir una lengua natural en los términos en los que lo hace el cerebro humano? son preguntas pertinentes para la Lingüística⁷.

1.1.2. La jerarquía de lenguajes de Chomsky

De acuerdo con lo visto hasta ahora, una lengua natural, en tanto que objeto de estudio científico, debe expresarse mediante un lenguaje formal. El conjunto infinito de oraciones de esta será el conjunto, también infinito, de teoremas de ese lenguaje. La gramática que describa la lengua será el conjunto de axiomas y reglas de inferencia del lenguaje.

Como se dijo en 1.1.1, cabe distinguir varios tipos de lenguaje formal en función de la complejidad computacional necesaria para su descripción. Chomsky⁸ estableció una jerarquía de posibles lenguajes en orden creciente de complejidad. Se preguntó, además, cuál de todos esos tipos de lenguaje era el adecuado para describir una lengua natural. Existen diversos criterios para determinar la pertinencia de un lenguaje formal, de una gramática, para la descripción de una lengua natural. En primer lugar, es necesario lo que suele llamarse una adecuación observacional. Es necesario que el lenguaje en cuestión, la gramática, sea capaz de derivar como teoremas todas las oraciones pertenecientes a la lengua, y que no derive como teoremas aquellos que no sean oraciones de la lengua. Pero es un hecho que esta tarea la pueden cumplir bien para una lengua natural cualquiera muy distintos tipos

⁷En realidad, la Lingüística no aspira a describir una lengua en los términos exactos en que pueda contemplarla el cerebro humano, entendido este como dispositivo computacional. El funcionamiento del cerebro, el modo en que este represente su conocimiento es una cuestión biológica. La Lingüística trabaja en términos puramente lingüísticos, no biológicos. Pero como objetivo a largo plazo, el lingüista aspira a que esos términos tengan una correspondencia directa con los términos en los que la Biología exprese los procesos cerebrales, de igual modo que los procesos químicos se expresan en términos químicos que, en última instancia, se corresponden con procesos físicos fundamentales.

⁸Chomsky (1956). Véanse también Sánchez León (2006) y Serrano (1975).

de gramática. Por tanto, se pide a una gramática, además, que tenga lo que se ha dado en llamar adecuación descriptiva: debe describir adecuadamente la lengua. En general, se considera que una gramática describe una lengua mejor que otra si resulta más sencilla y si capta más generalizaciones. Estas son consideraciones comunes a cualquier ciencia. Es preferible una teoría científica que otra si, aun describiendo la misma parcela de realidad, una lo hace con menor aparato teórico, explicando diferentes hechos como resultado de los mismos principios generales.

Pero, en Lingüística, parece pertinente contemplar aún otros criterios de adecuación de una gramática como descripción de una lengua. La Lingüística es una ciencia empírica, que pretende estudiar una parcela del universo: el lenguaje humano. Este lenguaje es fruto de una actividad, o una situación cerebral concreta. La gramática, como ya se ha dicho, debe ser una teoría plausible sobre esa actividad cerebral concreta. Por tanto, el lingüista está constreñido, a la hora de crear una gramática, por la necesidad de buscar una cierta adecuación explicativa. Es decir, su gramática debe ser compatible con una cierta teoría general del lenguaje natural, concebido este como facultad humana universal. Su gramática debe ser un modelo plausible como descripción de la gramática que, se entiende, utiliza el cerebro humano de un modo concreto y observable.

En definitiva, Chomsky distinguió varios tipos posibles de lenguaje formal en función de su complejidad. Y es necesario elegir, para la descripción de las lenguas naturales, uno de estos tipos: el más adecuado desde un punto de vista observacional, descriptivo y explicativo. Como se verá, cualquiera de los tipos de gramática posibles es adecuado desde un punto de vista observacional para la descripción de una lengua. En cuanto a la adecuación descriptiva, se verá que este criterio hace ver que ciertos lenguajes son demasiado sencillos para describir las lenguas naturales de un modo adecuado, pues dan lugar a gramáticas muy complejas y redundantes. Por último, hay ciertos criterios relacionados con la adecuación explicativa que hacen preferir un tipo de gramática a otro. Por ejemplo, se suele entender que, en principio,

cuanto más sencillo sea un lenguaje formal, más plausible es como gramática de una lengua natural. La idea es que el cerebro humano debe enfrentarse a tareas de generación y procesamiento de oraciones, y estas tareas se antojan de un coste computacional enorme. Cuanto mayor sea la complejidad computacional de una gramática, más costoso será el desarrollo real de esas tareas para el cerebro humano. Por tanto, en principio, suele considerarse más adecuada una gramática cuanto menor sea su complejidad computacional. En general, los criterios de adecuación de tipo descriptivo y explicativo suelen entrar en conflicto. Una gramática suele captar más generalizaciones, suele ser más sencilla desde un punto de vista teórico, cuanto mayor sea su complejidad computacional; la cual, sin embargo, es un problema desde el punto de vista de la adecuación explicativa.

No sería justo obviar aquí lo que Chomsky entendió como criterio fundamental de adecuación explicativa. Para Chomsky el cerebro humano está equipado de forma innata con una Gramática Universal:⁹ un muy rico conocimiento gramatical que evolucionará de formas distintas en los distintos individuos al contacto con las lenguas con las que estos tengan experiencia. La idea es que el conocimiento gramatical necesario para el manejo de una lengua tiene una grandísima complejidad. Llama la atención el hecho de que un niño sea capaz de aprender esa enorme complejidad haciendo uso solo de una muy pobre experiencia, recibida de los usos lingüísticos que lo rodean en sus primeros años de vida. Para Chomsky, esto solo se explica si se supone que el grueso de ese conocimiento lingüístico lo tiene el niño de forma innata. A ese conocimiento innato, y por tanto universal, común a todo ser humano, es a lo que se llama Gramática Universal. Las distintas lenguas naturales son, entonces, esencialmente iguales entre sí. Si esto es cierto, se espera que la Lingüística sea capaz de describir esas distintas lenguas naturales mediante gramáticas esencialmente idénticas. Esta es una restricción muy grande a lo que puede ser una gramática de una lengua natural: debe ser compatible con una teoría general del lenguaje humano, debe utilizar mecanismos es-

⁹Véase, por ejemplo, Chomsky (1986b), cap. 1.

encialmente comunes a los que usan lenguas aparentemente muy diferentes. Esta es, claro, una restricción relacionada con la adecuación explicativa, que debe contemplar la Lingüística si se toma como cierta la tesis universalista de Chomsky.

Es en el conflicto entre la adecuación descriptiva y la explicativa (se entiende esta última de uno u otro modo) donde reside el valor empírico de la Lingüística Generativa. Esta no es un mero artificio matemático. Es una disciplina empírica constreñida por los datos. Debe lograr una teoría generalizadora y sencilla del lenguaje humano, compatible, al tiempo, con los datos de las distintas lenguas y con determinadas hipótesis de índole general.

En 1.1.2.1, 1.1.2.2, 1.1.2.3 y 1.1.2.4, se muestran los cuatro tipos de lenguajes que contempló Chomsky en su jerarquía. Se han colocado en orden creciente de complejidad. Hay que tener en cuenta que cada lenguaje engloba a todos los lenguajes de complejidad menor, y las gramáticas que definen cada lenguaje engloban también a todas las gramáticas de complejidad menor. Es decir, un tipo de gramática G es capaz de definir lenguajes L de hasta una determinada complejidad. Las gramáticas y lenguajes menos complejos que los de tipo G y L son casos concretos de gramáticas y lenguajes de tipo G y L que no alcanzan la complejidad máxima que pueden expresar estos.

1.1.2.1. Lenguajes regulares

El primer tipo de lenguaje que contempla Chomsky en su jerarquía son los lenguajes regulares. Son secuencias de elementos que obedecen a la estructura a^n . Es decir, son secuencias que constan de un número n de elementos de tipo a . Son meras secuencias lineales en las que sus constituyentes no mantienen otra relación entre sí que la de sus posiciones lineales relativas. Por ejemplo, para definir el lenguaje $a^n b^m$ —una secuencia lineal de n elementos de tipo a seguidos de m elementos de tipo b , donde n y m no son necesariamente iguales, pero son mayores que 0— es necesaria una gramática como la que se representa en la figura 1.1.

<i>Axioma</i>	A
<i>Regla1</i>	$A \rightarrow aA$
<i>Regla2</i>	$A \rightarrow aB$
<i>Regla3</i>	$B \rightarrow bB$
<i>Regla4</i>	$B \rightarrow b$

Figura 1.1: Lenguaje $a^n b^m$

En la figura 1.2 se muestra la derivación de $aaabb$, uno de los teoremas del lenguaje $a^n b^m$, de acuerdo con la gramática de la figura 1.1.

<i>Axioma</i>	A
<i>Regla1</i>	aA
<i>Regla1</i>	aaA
<i>Regla2</i>	$aaaB$
<i>Regla3</i>	$aaabB$
<i>Regla4</i>	$aaabb$

Figura 1.2: Derivación en $a^n b^m$

A las gramáticas que representan a los lenguajes regulares se les llama gramáticas regulares. Sus reglas toman como *input* una secuencia y la amplían necesariamente siempre por una de sus márgenes, siempre por la misma. La sucesiva aplicación de reglas a una secuencia hace crecer a esta siempre linealmente, siempre por un extremo, por el mismo. Obsérvese esto en la gramática de la figura 1.1. Los símbolos susceptibles de ser sustituidos — A y B — siempre se colocan en la margen derecha del *output* de las reglas. Concretamente, las reglas de las gramáticas regulares solo tienen la forma $A \rightarrow tB$ y $A \rightarrow t$ o $A \rightarrow Bt$ y $A \rightarrow t$, donde B es un signo rescribible y t no.

1.1.2.2. Lenguajes independientes del contexto

En el apartado 1.1.2.1 se han descrito los lenguajes regulares. Los teoremas de estos lenguajes eran meras secuencias lineales de elementos: secuencias

del tipo a^n , con n elementos de tipo a . Son el tipo más sencillo de lenguaje. Hay lenguajes más complejos. Es el caso, por ejemplo, de un lenguaje de tipo $a^n b^n$; es decir, un lenguaje formado por secuencias consistentes en un número n de elementos de tipo a seguidos de un mismo número n de elementos de tipo b . Este lenguaje no es regular, no puede ser descrito mediante una gramática regular¹⁰. Recuérdese que las gramáticas regulares solo permiten describir secuencias lineales, sin ningún tipo de estructura interna. Las secuencias se van creando con ellas mediante la sucesiva ampliación lineal de la misma margen de las secuencias anteriores. Por ello, para crear una secuencia $a^n b^n$ con una gramática regular habría que añadir a la secuencia primero a n veces, y después b , también n veces. Pero esto es imposible, pues no se puede saber cuántas veces se ha iterado a en la derivación en el momento en que se tiene que iterar b .

$$\begin{array}{ll} \text{Axioma} & A \\ \text{Regla1} & A \rightarrow aAb \\ \text{Regla2} & A \rightarrow ab \end{array}$$

Figura 1.3: Lenguaje $a^n b^n$

Por la misma razón, tampoco son regulares lenguajes como $a^n b^n c^n$, $a^n b^n c^n d^n$, o, en definitiva, cualquier lenguaje del tipo $(a^n)^m$, donde m sea mayor que 1. Los teoremas de estos lenguajes no son meras secuencias lineales de elementos, sino objetos multidimensionales. Para definirlos son necesarias gramáticas cuyas reglas definan un crecimiento parejo de las distintas dimensiones que conforman las secuencias. Por ejemplo, para definir el lenguaje $a^n b^n$ es necesaria una regla que añada a la secuencia una b por cada a añadida. Esta gramática se representa en la figura 1.3. En la figura 1.4 se muestra la

¹⁰Sí son lenguajes regulares $a^k b^k$, donde k es un número menor o igual a una cota determinada. Por ejemplo, sí se puede crear una gramática regular que defina secuencias formadas por k iteraciones de a seguidas de k iteraciones de b , donde k es, por ejemplo, un número menor o igual a diez. Esta sería una gramática trivial que enumeraría uno a uno cada uno de sus teoremas. También es regular un lenguaje como $a^n b^m$, donde n y m pueden tener o no el mismo valor.

derivación de uno de los teoremas del lenguaje $a^n b^n$: $aaaaabbbbb$.

<i>Axioma</i>	A
<i>Regla1</i>	aAb
<i>Regla1</i>	$aaAbb$
<i>Regla1</i>	$aaaAbbb$
<i>Regla1</i>	$aaaaAbbbb$
<i>Regla2</i>	$aaaaabbbbb$

Figura 1.4: Derivación en $a^n b^n$

Los lenguajes de tipo $a^n b^n$ se conocen como lenguajes independientes del contexto. Y las gramáticas que los generan —la de la figura 1.3, por ejemplo— son gramáticas independientes del contexto. Reciben este nombre por el hecho de que las reglas de estas gramáticas tienen siempre la estructura $A \rightarrow x$, donde x es una secuencia de símbolos. Es decir, son reglas que rescriben un símbolo de la entrada, A , sea cual sea el contexto en el que este aparece. De ahí el nombre: independientes del contexto. En el siguiente apartado 1.1.2.3 se verán reglas que permiten la rescritura de A , pero solo en determinados contextos —por ejemplo, en $xAy \rightarrow xay$, se definiría la posible rescritura de A como a , solo para aquellos casos en los que A aparezca precedido de una secuencia x y seguido de una secuencia y —. A las gramáticas con este tipo de reglas, y a los lenguajes que definen, se les llamará dependientes del contexto.

1.1.2.3. Lenguajes dependientes del contexto

En el apartado 1.1.2.1 se ha descrito el tipo de lenguaje más sencillo de la jerarquía de Chomsky: los lenguajes regulares. Las secuencias de estos lenguajes son secuencias lineales de símbolos. Podiera decirse que son secuencias unidimensionales, cuyos símbolos constituyentes se relacionan entre sí en una única dimensión: la única relación que se establece entre ellos es la de ordenación lineal. Son lenguajes de tipo a^n . En cambio, en el apartado 1.1.2.2 se vio un tipo más complejo de lenguaje, el segundo tipo en orden

creciente de complejidad en la jerarquía de Chomsky: los lenguajes independientes del contexto. Las secuencias de estos lenguajes son bidimensionales. Por ello pueden ser de tipo $a^n b^n$, pues las reglas que las definen pueden establecer un crecimiento parejo de la secuencia en sus dos dimensiones.

En este apartado se verá un tipo de lenguaje, también contemplado en la jerarquía de Chomsky, aún más complejo. En este grupo se encuadran los lenguajes de tipo $a^n b^n c^n$, $a^n b^n c^n d^n$, $a^n b^n c^n d^n e^n$, y, en general, cualquier lenguaje de tipo $(a^n)^m$ para el cual m sea mayor que 2. Si las secuencias de los lenguajes regulares eran unidimensionales y las de los lenguajes independientes del contexto bidimensionales, este nuevo tipo de lenguajes tendrá secuencias multidimensionales. Igual que las reglas de los lenguajes de tipo $a^n b^n$ podían definir un crecimiento uniforme de la secuencia en cada una de sus dos dimensiones, las reglas de un lenguaje como $a^n b^n c^n$ deben poder contemplar la secuencia como un objeto tridimensional y, así, definir un crecimiento uniforme de esta en sus tres dimensiones. Por tanto, podría pensarse que la gramática necesaria para definir un lenguaje de tipo $a^n b^n c^n$ es esencialmente idéntica a la necesaria para definir un lenguaje $a^n b^n$. Ambas deberían constar de reglas de tipo $A \rightarrow x$, donde un símbolo A se rescribe como la secuencia x . La única diferencia estriba en que, en un caso x es un objeto bidimensional y en el otro es tridimensional.

Pero esto no es cierto: un lenguaje de tipo $a^n b^n c^n$ no puede definirse mediante una gramática independiente del contexto. La razón de ello es que los símbolos constituyentes de toda secuencia de un lenguaje —ya se conciba esta como objeto unidimensional, bidimensional o multidimensional— guardan entre sí una relación de ordenación lineal. La hipotética regla $A \rightarrow x$ en la que x es un objeto tridimensional solo sería posible si los elementos de cada dimensión no estuvieran ordenados linealmente con respecto a los del resto de dimensiones. Pero en una secuencia del lenguaje $a^n b^n c^n$, los elementos de tipo a han de preceder linealmente a los de tipo b , y estos a los de tipo c . Como se vio en 1.1.2.2, esta ordenación lineal de los elementos de las distintas dimensiones de una secuencia puede establecerse mediante reglas del tipo de

$A \rightarrow x$ cuando las dimensiones son dos. Pero es imposible hacer otro tanto con tres o más dimensiones. Por tanto, una gramática para el lenguaje $a^n b^n c^n$ necesita reglas más complejas que las reglas independientes del contexto del lenguaje $a^n b^n$.

En la figura 1.5 se muestra una gramática que describe el lenguaje $a^n b^n c^n$. Sus dos primeras reglas (*Regla1* y *Regla2*) permiten generar secuencias con un mismo número n de elementos de tipo a , b y c . Estas reglas son independientes del contexto. Pero, además de estas reglas, son necesarias otras (*Regla3*, *Regla4* y *Regla5*) que reordenen los elementos de modo que todos los de tipo b precedan a los de tipo c . Obsérvese que estas reglas no son independientes del contexto (de tipo $A \rightarrow x$). En su parte izquierda no aparece solo un símbolo rescribible, sino varios símbolos, rescribibles o no. Concretamente, las reglas de esta gramática son del tipo $x \rightarrow y$, donde x e y son secuencias de símbolos rescribibles o no, para las cuales solo existe la restricción de que el número de símbolos de y debe ser mayor o igual al de x . A las gramáticas con este tipo de reglas se les llama dependientes del contexto; y a los lenguajes que describen, lenguajes dependientes del contexto. Esto se debe a que son reglas que permiten definir el contexto en el cual un símbolo puede rescribirse —como ocurre con *Regla4*, donde se define la rescritura de C como c , solo cuando C va precedida de c —. Pero, en realidad, también permiten definir reordenaciones de elementos, como la que se establece en *Regla3*.

<i>Axioma</i>	A
<i>Regla1</i>	$A \rightarrow aABC$
<i>Regla2</i>	$A \rightarrow abc$
<i>Regla3</i>	$cB \rightarrow Bc$
<i>Regla4</i>	$cC \rightarrow cc$
<i>Regla5</i>	$bB \rightarrow bb$

Figura 1.5: Lenguaje $a^n b^n c^n$

En la figura 1.6 se muestra la derivación de la secuencia $aaabbbccc$, conforme a la gramática de la figura 1.5. Los tres primeros pasos de la derivación son los que establecen que la secuencia tendrá tres elementos en cada una de sus dimensiones. Pero estos elementos no están adecuadamente ordenados linealmente: $aaabcBCBC$. El resto de pasos son necesarios para la reordenación de los elementos.

<i>Axioma</i>	A
<i>Regla1</i>	$aABC$
<i>Regla1</i>	$aaABCBC$
<i>Regla2</i>	$aaabcBCBC$
<i>Regla3</i>	$aaabBcCBC$
<i>Regla4</i>	$aaabBccBC$
<i>Regla3</i>	$aaabBcBcC$
<i>Regla4</i>	$aaabBcBcc$
<i>Regla3</i>	$aaabBBccc$
<i>Regla5</i>	$aaabbBccc$
<i>Regla5</i>	$aaabbbccc$

Figura 1.6: Derivación en $a^n b^n c^n$

En resumen, los lenguajes de tipo $a^n b^n c^n$, $a^n b^n c^n d^n$, $a^n b^n c^n d^n e^n$, etc. son lenguajes dependientes del contexto, para cuya descripción son necesarias gramáticas más complejas que las vistas en los apartados anteriores: gramáticas dependientes del contexto. Estas gramáticas permiten describir cualquier lenguaje recursivo: véase el apartado 1.1.2.4.

1.1.2.4. Lenguajes no recursivos

Todos los lenguajes contemplados por Chomsky en su jerarquía —los ya vistos en 1.1.2.1, 1.1.2.2 y 1.1.2.3 y los que se verán en este último apartado 1.1.2.4— son lo que se ha dado en llamar lenguajes recursivamente enu-

merables. Es decir, son lenguajes susceptibles de ser descritos mediante un sistema de axiomas y reglas de inferencia: estos generan como teoremas todas sus secuencias, y no generan secuencias ajenas al lenguaje. Fuera de este grupo quedan aquellos lenguajes (no enumerables) para los cuales no existe tal modo de descripción. Además, los lenguajes expuestos hasta ahora —los regulares, los independientes del contexto y los dependientes del contexto— son lenguajes recursivos. Un lenguaje es recursivo cuando es posible crear para él un procedimiento formal capaz de decidir, para toda secuencia dada, si esta forma parte o no del lenguaje en cuestión. El último tipo de lenguaje de la jerarquía de Chomsky es el de los no recursivos, es decir, lenguajes recursivamente enumerables, para los cuales no existe un modo formal de decidir, para cualquier secuencia, si esta es o no un teorema del lenguaje.

Imagínese la gramática G del lenguaje L , donde L es un lenguaje infinito. Si L es recursivamente enumerable, debe poder idearse una gramática G que genere todas sus secuencias, y solo esas. La demostración de que cualquier teorema t sí forma parte de L consiste en que G generará t en un tiempo finito, mediante un número finito de pasos. Pero imagínese que se quiere comprobar si una secuencia cualquiera s es o no un teorema de L . Si s es un teorema, G lo generará en un tiempo finito, con lo cual queda demostrado que sí pertenece a L . Pero, si no es un teorema, G nunca lo generará; aunque, como L es infinito, un observador podría esperar infinitamente por si s fuese el siguiente teorema generado por G .

Obsérvese que demostrar cuáles son los no teoremas de L es tanto como definir al lenguaje complementario de L , llámese \bar{L} . Por tanto, la demostración de que cualquier secuencia \bar{t} no es un teorema de L consiste, entonces, en demostrar que sí es un teorema de \bar{L} . La secuencia \bar{t} no forma parte de L si la gramática \bar{G} que define a \bar{L} genera \bar{t} en un tiempo finito. Por tanto, un lenguaje recursivamente enumerable L es, además, recursivo si es posible definir mediante una gramática \bar{G} a su lenguaje complementario \bar{L} ; es decir, si \bar{L} también es recursivamente enumerable. Para estos lenguajes, los recursivos, es posible decidir en un tiempo finito si una secuencia cualquiera

s forma o no parte del lenguaje.

Pero existen lenguajes recursivamente enumerables — L — cuyos complementarios — \bar{L} — no son recursivamente enumerables. Es decir las secuencias de L forman un conjunto coherente, definido mediante alguna gramática G ; pero las secuencias que no forman parte de L no forman un conjunto coherente definible mediante una gramática: no puede definirse un hipotético \bar{L} a través de \bar{G} .

Los lenguajes regulares, los independientes del contexto y los dependientes del contexto eran lenguajes recursivos. Los lenguajes que se añaden a la jerarquía en este apartado se diferencian de aquellos en que no son recursivos. Las gramáticas de los lenguajes recursivos son reglas de tipo $x \rightarrow y$ donde x e y son secuencias de símbolos sin restricción alguna. Recuérdese que en las reglas dependientes del contexto (las más libres de las vistas hasta ahora) el número de símbolos de y debe ser mayor o igual a los de x ; es decir, no se permitían reglas que redujesen el tamaño del *input*. Las gramáticas que generan lenguajes no recursivos se caracterizan, en cambio, por admitir reglas que definen tanto el crecimiento como la reducción de las secuencias.

En el apartado 1.1.3.4 se verán ciertos ejemplos de lenguajes no recursivos. Para la correcta comprensión de la naturaleza de estos lenguajes es útil cierta formalización que aparecerá en el siguiente apartado. De momento, entiéndase que es posible representar cada una de las gramáticas vistas en este apartado como una secuencia formada a partir de un mismo alfabeto. Si este fuera el caso, se podría definir el conjunto de las secuencias que representan todas las gramáticas regulares, todas las gramáticas independientes del contexto, todas las gramáticas dependientes del contexto, todas las gramáticas que definen un lenguaje de al menos una secuencia, etc. Pues bien, como se dirá en 1.1.3.4, estos lenguajes definatorios del conjunto de gramáticas que obedecen a una propiedad no trivial son necesariamente no recursivos (algunos de ellos tampoco son recursivamente enumerables). Valgan estos, entonces, como ejemplos de lenguajes no recursivos. Sería larga y

alejada de los intereses de este trabajo la exposición de las correspondientes demostraciones —véase, no obstante, 1.1.3.4—.

1.1.3. Máquinas abstractas

En el apartado 1.1.2 se mostraron los cuatro tipos de lenguaje formal que contempló Chomsky en su jerarquía. En este nuevo apartado se describirán esos mismos lenguajes desde una perspectiva nueva. De forma paralela al trabajo de Chomsky, en Teoría de la Computación se definieron una serie de máquinas abstractas capaces de definir distintos tipos de lenguaje¹¹. Se ha demostrado que estos tipos de lenguaje son, precisamente, los que propuso Chomsky en su jerarquía. Es decir, que estas máquinas abstractas y las gramáticas de la jerarquía de Chomsky son descripciones equivalentes de los mismos tipos de lenguaje. Los lenguajes regulares se describieron en términos de autómatas finitos —como se verá en el apartado 1.1.3.1—. Por su parte, los lenguajes independientes del contexto se describieron en términos de autómatas a pila —apartado 1.1.3.2—, y los dependientes del contexto como autómatas acotados linealmente —apartado 1.1.3.3—. Además, los lenguajes no recursivos se pueden describir mediante máquinas de Turing —apartado 1.1.3.4—.

La finalidad de este apartado es caracterizar los lenguajes de 1.1.2 desde un punto de vista no meramente descriptivo, sino computacional, algorítmico. Una máquina abstracta puede entenderse como un algoritmo capaz de decidir en un número finito de pasos, y con el uso de una determinada cantidad de memoria, si una secuencia forma o no parte de un lenguaje. Será posible, por tanto, evaluar si, para llevar a cabo ese cálculo, un determinado tipo de máquina necesita un número mayor o menor de pasos, o una cantidad mayor o menor de memoria. En este apartado se pretende, entonces, valorar el coste computacional que tiene la descripción de cada uno de los lenguajes de la jerarquía de Chomsky; coste valorado en términos de tiempo (número de pasos) y espacio (cantidad de memoria). Se verá que, cuanto más complejo

¹¹Se sigue aquí a Hopcroft y otros (2001) y Aranda y otros (2006).

es un lenguaje (en 1.1.2 se presentaron en orden creciente de complejidad) mayor será el coste computacional necesario para definirlo.

1.1.3.1. Autómatas finitos

Los autómatas finitos son máquinas capaces de definir solo lenguajes regulares. Constan de un conjunto Q de estados (entre los cuales debe haber un estado de inicio i y un subconjunto Φ de estados finales o de aceptación), un conjunto Σ de símbolos de entrada y una función de transición δ . Esta función toma como argumentos un estado de Q y un símbolo de Σ y devuelve un estado de Q .

Un autómata opera del siguiente modo. Su objetivo es leer una cinta de entrada —una secuencia de signos de Σ — y decidir si dicha secuencia es o no un teorema del lenguaje que define. Para ello, se encuentra en todo momento en uno de los estados de Q —llámese q — y en una de las posiciones de la cinta de entrada, leyendo el signo de Σ que se encuentra en tal posición —llámese s —. Si la función de transición δ define una transición del estado q al estado k , dada la lectura del signo s , entonces, el autómata pasa a estar en estado k , y pasa a leer el signo que aparezca en la siguiente posición de la cinta de entrada. Esta operación se repite hasta que terminan los signos de la cinta de entrada. La actividad del autómata comienza siempre en el estado de inicio, leyendo la cinta de entrada en uno de sus extremos. En cada transición el autómata pasará a un nuevo estado y pasará a leer el siguiente signo de la cinta de entrada. Se considera que la secuencia de dicha cinta es un teorema del lenguaje que se define si, tras leerse el último signo, el autómata se encuentra en un estado final o de aceptación. En cambio, una secuencia no es un teorema si, al terminar la lectura de la cinta, el autómata no se encuentra en un estado final; o no se ha podido terminar la lectura de la cinta porque, en algún momento, el autómata se encontraba en un estado q , leyendo un signo s , para los cuales δ no define una transición posible.

Es importante precisar que un autómata finito (a diferencia de otras

máquinas más complejas que se verán en los apartados siguientes) lee la cinta de entrada siempre en el mismo sentido. Comienza leyendo la cinta en uno de sus extremos, por convención el izquierdo, y, con cada transición, pasa a leer el signo de la siguiente posición a la derecha.

Recuérdese que en el apartado 1.1.2.1 se definió el lenguaje $a^n b^m$, formado por todas las posibles secuencias de n elementos de tipo a seguidos de m elementos de tipo b . Este lenguaje se formaliza ahora mediante un autómata finito, en el cual $Q = \{q^0, q^1, q^2\}$; $i = q^0$; $\Phi = \{q^2\}$; $\Sigma = \{a, b\}$; y δ se muestra en la figura 1.7, donde el resultado de la función para cada par estado y signo es el estado que se encuentra en la convergencia entre ambos. Este autómata es, al igual que la gramática vista en 1.1.2.1, una definición intensional del lenguaje $a^n b^m$.

	a	b
q^0	q^1	
q^1	q^1	q^2
q^2		q^2

Figura 1.7: Lenguaje $a^n b^m$: δ

q^0	a	q^1
q^1	aa	q^1
q^1	aaa	q^1
q^1	$aaab$	q^2
q^2	$aaabb$	q^2

Figura 1.8: Derivación en $a^n b^m$

De acuerdo con él, un teorema de $a^n b^m$ es una secuencia de signos de Σ para los cuales δ define una transición desde el estado de inicio (q^0) hasta un estado final (q^2). Por ejemplo, la secuencia $aaabb$ es un teorema de $a^n b^m$. Se muestra la derivación que lo prueba en 1.8, donde cada fila muestra una

transición. En la primera columna se muestra el estado en el que se encuentra un autómata antes de realizar la transición. En la segunda columna aparece la secuencia derivada hasta el momento. Y en la tercera columna aparece el estado al cual se mueve el autómata.

En definitiva, un autómata de estados finitos es una máquina abstracta que realiza movimientos desde un estado a otro, motivados por la lectura secuencial de los signos de una cinta de entrada. Un autómata finito define, como teorema del lenguaje que representa, a toda secuencia que motive un movimiento desde su estado de inicio hasta un estado final. Mediante autómatas finitos solo pueden representarse lenguajes regulares, del tipo a^n . Téngase en cuenta que son máquinas que solo realizan una lectura lineal de una secuencia; y, en un momento dado, no tienen memoria alguna de las transiciones que han efectuado hasta entonces. Por ello, no pueden definirse con autómatas finitos lenguajes independientes del contexto —del tipo $a^n b^n$ —, pues se necesitaría recordar, en el momento de determinar el número de elementos de tipo b de la secuencia, cuántos elementos de tipo a se establecieron. Los autómatas finitos carecen de esa memoria. Tampoco es posible definir con estas máquinas lenguajes dependientes del contexto —del tipo $a^n b^n c^n$ —. En 1.1.2.3 se vio cómo, para definir estos lenguajes, era necesario generar un número n de elementos de tipo a , b y c ; y, después, reordenarlos de forma que todos los de tipo a precediesen a los de tipo b , y estos a los de tipo c . Para realizar esta reordenación es necesario identificar un contexto en la cinta de entrada (identificar, por ejemplo, una c y a continuación una b), y volver hacia atrás en esta para efectuar el cambio de orden (en la primera posición de la cinta borrar la c para escribir b , y en la segunda cambiar b por c). Los autómatas finitos solo leen la cinta de entrada de modo secuencial, por lo que para ellos no son posibles estas operaciones, y no pueden, por tanto, definir lenguajes dependientes del contexto.

1.1.3.2. Autómatas a pila

En el apartado 1.1.3.1 se ha visto un tipo de máquina, el autómata finito, capaz de definir solo lenguajes regulares. Estas máquinas establecen un movimiento de estado a estado conforme se hace una lectura lineal de la secuencia de la cinta de entrada. En cada momento del proceso no son capaces de recordar qué se ha leído anteriormente en la cinta, ni por qué estados se ha transitado. Una máquina de estas características es incapaz de definir lenguajes independientes del contexto, de tipo $a^n b^n$, pues carecen de la memoria necesaria para recordar, a la hora de establecer el número de elementos de tipo b de la secuencia, con cuántos elementos de tipo a contaba esta.

Por tanto, para definir lenguajes independientes del contexto, es necesario dotar al autómata con una memoria: una pila. Una pila es una estructura de almacenamiento de información en la cual el último elemento introducido es el primero en salir. Estas máquinas aumentadas con una pila reciben el nombre de autómata a pila. Constan, como los autómatas finitos, de un conjunto de estados Q , con un estado inicial i y un subconjunto de estados finales Φ ; un conjunto Σ de símbolos de entrada; una función de transición δ ; y una pila, con un alfabeto de pila Γ . La función de transición δ , en un autómata a pila, toma tres y no dos argumentos: un estado de Q , un símbolo de Σ y un símbolo de pila Γ . El resultado de la función es un par formado por un estado de Q y un nuevo símbolo de pila de Γ . En definitiva, si δ , en un autómata finito, definía transiciones de un estado a otro, condicionadas por el símbolo leído en la cinta de entrada, en un autómata a pila define esas mismas transiciones, pero condicionadas por el símbolo leído en la cinta de entrada, y el leído en la pila. Con cada transición, el autómata pasa a leer el siguiente símbolo de la cadena de entrada y opera de un modo u otro con la pila: se elimina el elemento leído en esta, se añade un nuevo elemento, o se mantiene la pila inalterada. Un autómata a pila reconoce como teorema del lenguaje que define a aquella secuencia que motive una transición desde el estado de inicio a un estado final y deje la pila vacía.

Por ejemplo, el lenguaje $a^n b^n$ se puede definir mediante un autómata a pila en el cual $Q = q^0, q^1$; $i = q^0$; $\Phi = q^1$; $\Sigma = a, b$; $\Gamma = A$; y δ . Se muestra este autómata en la figura 1.9. En el eje horizontal se han colocado los pares que motivan la transición de un estado a otro: el primer elemento de cada par es un símbolo de Σ ; el segundo, un símbolo de Γ . La secuencia AA indica que se añade a la pila un símbolo A a otro símbolo A preexistente. Por último, ϵ denota un elemento nulo: indica que en la pila no se encuentra ningún símbolo (la pila está vacía), o que el símbolo leído en la pila se ha eliminado.

	a, ϵ	a, A	b, A
q^0	q^0, A	q^0, AA	q^1, ϵ
q^1			q^1, ϵ

Figura 1.9: Lenguaje $a^n b^n$: δ

Dicho de forma intuitiva, este autómata admite secuencias, primero con n elementos de tipo a , y a continuación con m elementos de tipo b . Para controlar que n y m sean iguales, se utiliza la pila: por cada elemento de tipo a de la cinta de entrada se añade a la pila un elemento A . Es decir, la pila es la memoria que establece el número de elementos de tipo a vistos en la secuencia. A continuación, el autómata sólo admite un elemento de tipo b en la cinta de entrada por cada elemento de tipo A guardado en la pila. En cada transición lee un elemento de tipo b en la cinta y elimina una A en la pila. Si la secuencia tiene más elementos de tipo a que de tipo b , al terminar el autómata la cinta de entrada, la pila no estará vacía; y, si tiene más de tipo b que de tipo a , la función de transición no permitirá terminar de leer la cinta de entrada (pues esta solo permite leer una b si en la pila hay una A). En cualquier caso, la secuencia en cuestión no se considerará un teorema de $a^n b^n$. Pero si la secuencia cuenta con tantos elementos de tipo a como de tipo b , el autómata, al leer n elementos de tipo a , añadirá A a la pila n veces; y, a continuación, al leer n elementos de tipo b , eliminará A de la pila n veces. De este modo, al terminar de leer la cinta de entrada, la pila estará vacía y la secuencia en cuestión sí se considerará un teorema de $a^n b^n$.

Este proceso puede observarse en la siguiente derivación de 1.10, que legitima a $aaabbb$ como teorema de $a^n b^n$. Cada fila representa una transición. En la primera columna aparece el estado de origen de dicha transición, y en la cuarta el estado de destino. En la segunda columna aparecen la secuencia que el autómata ha reconocido hasta el momento, donde el último símbolo es el que la transición en cuestión ha leído en la cinta de entrada. En la tercera, por último, aparece el estado en el que la transición deja la pila.

q^0	a	A	q^0
q^0	aa	AA	q^0
q^0	aaa	AAA	q^0
q^0	aaab	AA	q^1
q^1	aaabb	A	q^1
q^1	aaabbb	ϵ	q^1

Figura 1.10: Derivación en $a^n b^n$

Los autómatas a pila pueden, entonces, definir lenguajes independientes del contexto. Pero no pueden definir lenguajes dependientes del contexto — $a^n b^n c^n$ —. Una memoria de tipo pila puede recordar el número elementos de tipo a de una secuencia. Pero esta información se va perdiendo conforme se vacía la pila al reconocerse los elementos de tipo b . No habría modo, con este tipo de memoria, de recordar a continuación con cuántos elementos de tipo c ha de contar la secuencia. Parece, por tanto, que para definir un lenguaje dependiente del contexto es necesario un tipo de máquina diferente, más compleja.

1.1.3.3. Autómatas acotados linealmente

Los autómatas acotados linealmente son máquinas abstractas capaces de definir lenguajes dependientes del contexto; capaces, por tanto, conforme a lo visto en 1.1.2.4, de definir cualquier lenguaje recursivo. Son autómatas, como los vistos hasta el momento; pero capaces, a diferencia de estos, de

leer la cinta de entrada no solo de forma secuencial, de izquierda a derecha, sino en ambos sentidos. Además pueden rescribir los símbolos de la cinta de entrada. Cuentan con un conjunto de estados Q , con un estado inicial i y un subconjunto de estados finales Φ ; un conjunto de símbolos Σ ; y una función de transición δ . Esta función toma dos argumentos, un estado de Q y un símbolo de Σ ; y da como resultado un triplete formado por un estado de Q , un símbolo de Σ y una instrucción D o I . Es decir, las transiciones definidas mediante δ representan movimientos de un estado a otro, motivados por la lectura de un símbolo de la cinta de entrada. Pero, además, estas transiciones suponen la rescritura de ese símbolo de la cinta (quizá por el mismo símbolo, o por otro), y establecen si la siguiente posición que se leerá en la cinta de entrada será la ubicada inmediatamente a la derecha del último símbolo leído (instrucción D), o a su izquierda (instrucción I)¹². Se dice de estos autómatas que están acotados linealmente porque la cinta de entrada es necesariamente finita, acotada entre un símbolo de inicio $\#$ y un símbolo final $\$$, ambos miembros de Σ . El autómata no podrá leer la cinta de entrada a la izquierda de $\#$ ni a la derecha de $\$$.

Se puede definir $a^n b^n c^n$ mediante un autómata acotado linealmente. Para ello, $Q = q^0, q^1, \dots, q^{11}$, donde $i = q^0$ y $\Phi = q^{11}$; $\Sigma = a, b, c, A, B, C, \#, \$$; y δ se define como aparece en la figura 1.11. El resultado de δ se ha representado como un estado de destino con un subíndice con dos elementos: el símbolo por el que se rescribe la posición de la cinta de entrada, y la instrucción que indica si el autómata pasa a leer la cinta hacia la derecha o la izquierda. Este autómata lee una secuencia, en primer lugar, de izquierda a derecha. Cuando encuentra la primera a , la cambia por A ; cuando encuentra la primera b , la cambia por B ; y, cuando encuentra la primera c , la cambia por C . Una vez llega al final de la secuencia, lee la cinta en sentido inverso para volver al

¹²Como se ve, los autómatas acotados linealmente no cuentan con una pila. Esta no es necesaria, pues la posibilidad de rescribir los símbolos de la cinta de entrada de estos autómatas, así como el poder leer dicha cinta en ambos sentidos, de forma no necesariamente secuencial, cubre las posibilidades expresivas que aporta a un autómata el uso de una pila.

comienzo. Este proceso, si la secuencia en cuestión cuenta con un número n de elementos de tipo c , se repite n veces. Después, el autómata lee el símbolo de final de secuencia $\#$ y pasa al estado q^9 . Si el número de elementos de tipo a y b de la secuencia es también n , no deberían quedar en este momento en la cinta elementos de tipo a , b ni c ; lo cual comprueba el autómata recorriéndola de derecha a izquierda. Si, efectivamente, este es el caso, se llega al estado de aceptación q^{11} . Esto nunca ocurriría si el número de elementos de tipo a o de tipo b es menor que n —pues nunca se llegaría a q^9 —, ni si es mayor —pues algún símbolo a o b impediría pasar de q^9 a q^{11} —.

	a	b	c	A	B	C	$\#$	$\$$
q^0							$q_{\#,D}^1$	
q^1	$q_{A,D}^2$			$q_{A,D}^1$				
q^2	$q_{a,D}^2$	$q_{B,D}^3$			$q_{B,D}^5$			
q^3		$q_{b,D}^3$	$q_{C,D}^4$			$q_{C,D}^6$		
q^4			$q_{c,I}^7$					$q_{\$,I}^9$
q^5		$q_{B,D}^3$			$q_{B,D}^5$			
q^6			$q_{C,D}^4$			$q_{C,D}^6$		
q^7						$q_{C,I}^8$		
q^8	$q_{a,I}^8$	$q_{b,I}^8$	$q_{c,I}^8$	$q_{A,I}^8$	$q_{B,I}^8$	$q_{C,I}^8$	$q_{\#,D}^0$	
q^9	$q_{c,I}^{10}$							
q^{10}				$q_{a,I}^{10}$	$q_{b,I}^{10}$	$q_{c,I}^{10}$	$q_{\#,D}^{11}$	
q^{11}								

Figura 1.11: Lenguaje $a^n b^n c^n$: δ

La derivación de la secuencia $aabbcc$, conforme a este autómata, se representa en la figura 1.12. En ella, cada fila denota una transición. En la primera columna aparecen los estados de inicio, y en la tercera los de destino. En la columna central se muestra la secuencia derivada hasta el momento. Obsérvese que la posición de la cinta que lee cada transición aparece bajo una flecha que indica cuál va a ser la posición de cinta leída por la siguiente transición. En conclusión, para definir lenguajes dependientes del contexto

q^0	$\overrightarrow{\#}$	q^1
q^1	$\# \overrightarrow{A}$	q^2
q^2	$\# A \overrightarrow{a}$	q^2
q^2	$\# A a \overrightarrow{B}$	q^3
q^3	$\# A a B \overrightarrow{b}$	q^3
q^3	$\# A a B b \overrightarrow{C}$	q^4
q^4	$\# A a B b C \overleftarrow{c}$	q^7
q^7	$\# A a B b \overleftarrow{C} c$	q^8
q^8	$\# A a B \overleftarrow{b} C c$	q^8
q^8	$\# A a \overleftarrow{B} b C c$	q^8
q^8	$\# A \overleftarrow{a} B b C c$	q^8
q^8	$\# \overrightarrow{A} a B b C c$	q^1
q^1	$\# A \overrightarrow{A} B b C c$	q^1
q^2	$\# A A \overrightarrow{B} b C c$	q^5
q^5	$\# A A B \overrightarrow{B} C c$	q^3
q^3	$\# A A B B \overrightarrow{C} c$	q^6
q^6	$\# A A B B C \overrightarrow{C}$	q^4
q^4	$\# A A B B C C \overleftarrow{\$}$	q^9
q^9	$\# A A B B C \overleftarrow{c} \$$	q^{10}
q^{10}	$\# A A B B \overleftarrow{c} c \$$	q^{10}
q^{10}	$\# A A B \overleftarrow{b} c c \$$	q^{10}
q^{10}	$\# A A \overleftarrow{b} b c c \$$	q^{10}
q^{10}	$\# A \overleftarrow{a} b b c c \$$	q^{10}
q^{10}	$\# \overleftarrow{a} a b b c c \$$	q^{10}
q^{10}	$\overrightarrow{\#} a a b b c c \$$	q^{11}

Figura 1.12: Derivación en $a^n b^n c^n$

—como $a^n b^n c^n$, $a^n b^n c^n d^n$, $a^n b^n c^n d^n e^n \dots$ — son necesarias máquinas bastante más complejas que las vistas en los apartados anteriores. Son los autómatas acotados linealmente. Recuérdese que el conjunto de lenguajes dependientes del contexto engloba todos los lenguajes recursivos. Cualquier lenguaje recursivo puede, entonces, definirse mediante una máquina de este tipo. Véase el ejemplo relativo al lenguaje $a^n b^n c^n$ en la figura 1.12.

Como se ha visto, para definir estos lenguajes no es posible llevar a cabo un mero reconocimiento lineal de la secuencia, como ocurre con los lenguajes regulares o los independientes del contexto. La definición de lenguajes como $a^n b^n c^n$ requiere formas de reconocimiento más complejas. Por ejemplo, el autómata descrito en este apartado necesita, para reconocer una secuencia como miembro de $a^n b^n c^n$, recorrer la cinta de entrada $2n$ veces: n veces en un sentido y n veces en otro. Además, para definir tales lenguajes, es necesaria una memoria más poderosa que la pila utilizada en 1.1.3.2. Para los autómatas acotados linealmente, el hecho de poder retroceder en la lectura de la cinta de entrada supone poder recordar, en todo momento, la naturaleza de la secuencia derivada hasta entonces.

Sin embargo, los autómatas acotados linealmente no son capaces de reconocer lenguajes no recursivos. Todavía es necesario un tipo de máquina de mayor capacidad expresiva: la máquina de Turing.

1.1.3.4. Máquinas de Turing

Se ha dicho que Gödel probó que no toda proposición matemática es demostrable. Con ello, los matemáticos comenzaron a preguntarse qué proposiciones eran demostrables y cuáles no. Dicho en términos de la Teoría de la Computación, cobró importancia la pregunta de para qué problemas existía un método efectivo, un algoritmo de resolución. En definitiva, los matemáticos se estaban preguntando qué es computable y qué no. En los apartados anteriores se ha visto que cualquier problema formalizable como lenguaje recursivo es computable, pues existen máquinas abstractas capaces de definir

cualquiera de estos lenguajes. Pero ¿existe una máquina capaz de definir cualquier lenguaje recursivamente enumerable? Una máquina tal sería capaz de solucionar cualquier problema computable. Podría decirse que un problema que no cuente con solución mediante tal máquina no es computable. Esta máquina sería, entonces, la definición de la computabilidad.

Varios matemáticos utilizaron diversos lenguajes formales para lograr esta definición de lo computable. Alan Turing ideó lo que se ha dado en llamar máquina de Turing¹³, Alonzo Church creó el cálculo λ ¹⁴, Stephen Kleene definió las proposiciones recursivas generales¹⁵, y también Emil Post dio con un sistema: la máquina de Post¹⁶. Las cuatro aproximaciones son equivalentes¹⁷. A continuación se explica la más utilizada en la bibliografía: la máquina de Turing.

En realidad, la máquina de Turing es prácticamente igual a los autómatas acotados linealmente vistos en 1.1.3.3. La única diferencia estriba en que, en una máquina de Turing, la cinta de entrada es potencialmente infinita. Recuérdese que la cinta de entrada de los autómatas acotados linealmente tenía una longitud finita: la de la secuencia que se pasa a la máquina como entrada, más un símbolo de inicio de secuencia —#— y otro de final —\$—. La máquina no podía trabajar fuera de la cinta comprendida entre # y \$. En cambio, una máquina de Turing no tiene símbolos de inicio y final de secuencia, pues su cinta es infinita. Recuérdese que la cinta de los autómatas acotados linealmente actuaba como memoria: estos podían retroceder en la lectura de la cinta para comprobar así la secuencia derivada hasta el momento. Esto dotaba a tales autómatas de una capacidad expresiva mayor que la de los autómatas que solo hacían una lectura secuencial de la cinta. Por su parte, la máquina de Turing tiene una capacidad expresiva mayor

¹³Turing (1936).

¹⁴Church (1941).

¹⁵Kleene (1936).

¹⁶Post (1936) y Post (1946).

¹⁷Feynman (2003) pp. 65-66.

que la de los autómatas acotados linealmente gracias a que cuentan con una memoria más poderosa: una cinta infinita. Esta permite a la máquina de Turing guardar resultados intermedios. Obsérvese que un autómata acotado linealmente recuerda solo el último estado de la derivación. En cambio, una máquina de Turing puede recordar incluso la derivación completa. En definitiva, una máquina de Turing es un autómata que lee una cinta de entrada infinita, de forma no necesariamente secuencial, en ambos sentidos; y que es capaz de rescribir los símbolos de tal cinta.

Si los autómatas acotados linealmente eran capaces de definir cualquier lenguaje recursivo, las máquinas de Turing son capaces de definir cualquier lenguaje recursivamente enumerable, incluidos los no recursivos. Puede decirse, entonces, que un lenguaje L es computable si y solo si existe una máquina de Turing MT que lo defina: $MT(L)$. Recuérdese que los lenguajes no recursivos son aquellos que son computables, pero su lenguaje complementario no lo es. Es decir, es posible definir las secuencias de tal lenguaje como teoremas en un número finito de pasos, pero no es posible definir los no teoremas como tales en un número finito de pasos. Por tanto, una máquina de Turing que defina un lenguaje no recursivo L , y que esté comprobando si una secuencia σ forma parte de L , comprobará siempre que σ es un teorema, si este es el caso, en un número finito de pasos: parará en un estado de aceptación. Pero, si σ no es un teorema de L , quizá el proceso de comprobación no acabe nunca: la máquina de Turing nunca parará. Por tanto, puede definirse un lenguaje recursivo L como aquel para el que existe alguna máquina de Turing MT que lo define — $MT(L)$ — de tal modo que, para cualquier secuencia σ , MT se detiene, sea σ teorema de L o no. En cambio, un lenguaje L no es recursivo si la máquina de Turing MT que lo define, para alguna secuencia σ , no para nunca al intentar decidir si σ es un teorema de L .

Siguiendo a Hopcroft y otros (2001) (y en relación a lo apuntado en 1.1.2.4), todas las máquinas de Turing imaginables podrían formalizarse como distintas secuencias formadas a partir de un mismo alfabeto —por ejemplo, un alfabeto binario—. Todas estas secuencias formarían un único lengua-

je: el del conjunto de las máquinas de Turing posibles. A partir de él, podrían definirse lenguajes consistentes en subconjuntos de máquinas de Turing. Estos subconjuntos de máquinas de Turing podrían definir propiedades no triviales de los lenguajes: los lenguajes independientes del contexto, los lenguajes no vacíos (con al menos una secuencia), etc. Puede demostrarse que ninguno de estos lenguajes consistentes en subconjuntos de máquinas de Turing es recursivo. Sí pueden ser, en cambio, recursivamente enumerables. Valgan estos, entonces, como ejemplo de lenguajes recursivamente enumerables no recursivos. Quedan fuera del interés de este trabajo las correspondientes demostraciones¹⁸.

1.1.4. Un formalismo adecuado para el lenguaje natural

En 1.1.2 se vieron distintos tipos de gramáticas que definían distintos tipos de lenguajes, de mayor o menor complejidad: lenguajes regulares, independientes del contexto, dependientes del contexto o no recursivos. En 1.1.3 se mostró una definición alternativa de estos lenguajes, en términos de máquinas abstractas. Ambos modos de definición son equivalentes. En este apartado se reflexiona sobre si las lenguas naturales han de formalizarse como lenguajes regulares, independientes del contexto, dependientes del contexto o no recursivos. Parece razonable elegir, para la definición de las lenguas naturales, el formalismo más sencillo posible. Por ejemplo, si es posible definir de modo adecuado las lenguas naturales mediante lenguajes regulares, no hay razón para utilizar en tal tarea un formalismo más complejo. Pero si los lenguajes regulares no tienen suficiente capacidad expresiva para ello, sí estará justificado el uso de un formalismo más complejo: lenguajes independientes del contexto, etc.

Los lenguajes de la jerarquía de Chomsky se han presentado en orden creciente de complejidad. La complejidad computacional se mide en términos

¹⁸Se remite para ello a Hopcroft y otros (2001), cap. 9.

de tiempo y espacio¹⁹. El tiempo se entiende como el número de pasos que utiliza un algoritmo, una máquina de Turing — MT — para la resolución de un problema, y el espacio como la cantidad de memoria que emplea MT en tal tarea. Dada una entrada de longitud n para MT , puede decirse que $T(n)$ es el tiempo de resolución de MT para dicha entrada. Si T es una función polinómica²⁰, puede considerarse que MT podrá resolver el problema en un tiempo razonable para cualquier valor de n . En cambio, si T es una función exponencial²¹, el tiempo de ejecución de MT será enorme, excepto para valores muy bajos de n . Se entiende que un problema es tratable si existe una máquina de Turing capaz de resolverlo en tiempo polinómico, y es intratable si las máquinas de Turing capaces de resolverlo lo hacen en tiempo exponencial. Por su parte, el espacio E que necesita una máquina de Turing MT , dada una entrada de longitud n , es, como máximo, $T(n) + 1$. Por tanto, $E(n) = T(n) + 1$. Esto se debe a que MT podrá utilizar, en $T(n)$ pasos, como máximo, $T(n)$ celdas de la cinta de entrada, más aquella en la que se encuentra al inicio del proceso.

Dentro de los problemas tratables para alguna máquina de Turing, hay que distinguir varios tipos. Para ello, es necesario introducir el concepto de determinismo. Cualquiera de las máquinas abstractas vistas en 1.1.3, cualquier máquina de Turing en definitiva, puede ser determinista o no determinista. Una máquina de Turing es determinista cuando su función de transición δ —aquella que define las transiciones posibles entre sus estados—, es propiamente una función: cuando, para cualquier entrada, δ establece una sola salida posible. En cambio, una máquina de Turing es no determinista cuando, para alguna entrada, δ establece más de una salida. Es decir, una máquina de Turing es determinista si, dado $\delta(x, \sigma) = y$, donde x e y son, respectiva-

¹⁹La siguiente explicación de la complejidad computacional se ha tomado de Hopcroft y otros (2001) cap. 10.

²⁰Una función es polinómica cuando no presenta variables en sus exponentes. Por ejemplo, $T(n)$ sería polinómica si $T(n) = n^2$, pero no sería polinómica si $T(n) = 2^n$.

²¹Una función es exponencial si presenta variables en sus exponentes. Por ejemplo, $T(n)$ es exponencial si $T(n) = 2^n$.

mente, los estados de origen y destino que establece δ para una cadena de entrada σ , para cada valor de x solo existe un posible valor de y . En cambio, una máquina de Turing es no determinista si, dado $\delta(x, \sigma) = y$, existe algún valor de x para el cual hay varios valores de y .

Hay problemas que pueden ser resueltos por máquinas de Turing deterministas en tiempo polinómico. Y hay problemas, normalmente de tipo factorial o combinatorio, que pueden ser resueltos por máquinas de Turing no deterministas en tiempo polinómico; pero, si se tratan con máquinas de Turing no deterministas, necesitan un tiempo exponencial. Al primer conjunto de problemas se le suele llamar P , y al segundo NP . La distinción entre P y NP es relevante, pues los procesadores de los ordenadores actuales son necesariamente deterministas. Por tanto, para estos, solo son tratables los problemas de P . Un procesador de este tipo, para emular el comportamiento de una máquina de Turing no determinista, debe contemplar en cada momento solo una de las posibles transiciones que δ establece para una determinada entrada. El resto de posibilidades han de ser guardadas en memoria, y ha de existir un algoritmo que establezca cómo se van contemplando, una a una, las distintas posibilidades. Un hipotético procesador no determinista, en cambio, podría contemplar, al tiempo, como un único paso de procesamiento, las distintas opciones.

La Teoría de la Computación tiene abiertas en la actualidad muchas preguntas relativas a la complejidad computacional. Cabe preguntarse si, en realidad, cualquier problema de NP es potencialmente reducible a un problema de P . Se sabe que cualquier problema formalizable mediante un lenguaje regular pertenece a P . Además, es posible que los problemas formalizables mediante lenguajes independientes del contexto también pertenezcan siempre a P . En cambio, parece que, entre los problemas formalizables mediante lenguajes más complejos —dependientes del contexto y no recursivos— hay algunos cuya pertenencia a P es muy difícil.

Por tanto, ¿qué tipo de lenguaje conviene para la formalización de las

lenguas naturales? En cualquier caso, no sería deseable una formalización que necesitase un tiempo de procesamiento exponencial. Cabría preguntarse si el cerebro humano es un procesador determinista o no determinista. Si este fuese capaz de procesar de forma no determinista, podría tratar problemas de NP en tiempo polinómico. En tal caso, una formalización no determinista de las lenguas naturales perteneciente a NP , pero no a P , sería válida. Pero, si el objetivo de una formalización es la emulación del procesamiento lingüístico en un ordenador actual, determinista, esta formalización daría lugar a tiempos de procesamiento enormes. Para tales fines sería necesaria una formalización perteneciente a P .

En realidad, hasta el momento, la Lingüística se ha guiado por la idea intuitiva de que las gramáticas de la jerarquía de Chomsky presentan una complejidad computacional creciente; y es deseable, como teoría de las lenguas naturales, la gramática más sencilla de entre las de capacidad descriptiva suficiente. Lo mostrado en el apartado 1.1.3 puede ayudar a ilustrar esta idea intuitiva. El procesamiento de la cadena de entrada de las gramáticas más sencillas —los autómatas finitos, capaces de definir lenguajes regulares; y los autómatas a pila, capaces de definir lenguajes independientes del contexto— consiste en una mera lectura lineal: si la cadena de entrada tiene una longitud n , el número de pasos necesarios para efectuar dicha lectura es n . Se entiende que los autómatas a pila acarrear mayor coste computacional que los finitos por el uso de la pila: esta supone un gasto de memoria. De cualquier modo, ambos tipos de gramática tienen bajo coste computacional en comparación con los de los autómatas acotados linealmente, capaces de definir lenguajes dependientes del contexto. Estos autómatas, al permitir la lectura de la cinta de entrada en ambos sentidos, pueden requerir costes de procesamiento no lineales sobre la longitud de la cinta de entrada. Tómese como ejemplo el autómata que en 1.1.3.3 definía el lenguaje $a^n b^n c^n$. Este autómata lee una secuencia $a^n b^n c^n$, aproximadamente, $2n$ veces. Si la secuencia en cuestión es de longitud m , los pasos necesarios para su lectura son m^{2n} . Por último, las máquinas de Turing irrestrictas, capaces de definir incluso lenguajes no recursivos, son idénticas a los autómatas acotados linealmente, por lo que pueden

requerir costes computacionales como los de estos. Pero, además, admiten una cinta de entrada infinita, donde se pueden guardar resultados intermedios de la derivación. Por ello, permiten un gasto de memoria mucho mayor. En conclusión, la descripción de los distintos tipos de lenguajes formales en términos de máquinas abstractas da una idea intuitiva clara del coste computacional que acarrea el uso de cada uno de estos lenguajes.

En los siguientes subapartados se utilizan los conocidos criterios de adecuación observacional, descriptiva y explicativa para valorar si los lenguajes de la jerarquía de Chomsky son o no apropiados para la definición de las lenguas naturales. Dado lo visto en este apartado, se escogerá como formalismo el más sencillo de entre los que resulten adecuados.

1.1.4.1. Adecuación observacional

Se ha llamado capacidad generativa débil²² de una gramática al conjunto de secuencias que dicha gramática puede generar, con independencia del análisis que pueda asignarles. La capacidad generativa débil de una gramática, es, en definitiva, el lenguaje que define esta. Para que una gramática sea adecuada para la descripción de las lenguas naturales, es indispensable que tenga una capacidad generativa débil suficiente para tal fin: que pueda discriminar adecuadamente qué secuencias forman parte de una lengua y cuáles no. Una gramática es entonces adecuada para la descripción de las lenguas, desde un punto de vista observacional, si tiene una capacidad generativa débil suficiente para ello.

Chomsky (1957) demostró que las gramáticas regulares no tienen tal capacidad, por lo que no son observacionalmente adecuadas, y no interesan como teoría de las lenguas naturales. Recuérdese que estas gramáticas solo pueden definir lenguajes del tipo a^n , es decir, lenguajes formados por secuencias concebidas como mera concatenación de n elementos de tipo a . Quedan fuera de la capacidad generativa débil de estas gramáticas secuencias del tipo

²²Chomsky (1965), cap. 1.9.

$a^n b^n$, donde la aparición de n elementos de tipo a condiciona el que aparezca a continuación en la secuencia un mismo número n de elementos de tipo b . Chomsky observa que las lenguas naturales sí presentan secuencias con este patrón $a^n b^n$. Es un hecho que, en estas, es posible anidar una estructura de tipo α dentro de una misma estructura de tipo α , de forma recursiva, de modo que son potencialmente posibles infinitas anidaciones. Este tipo de situación acarrea, efectivamente, la aparición de patrones de tipo $a^n b^n$. Por ejemplo, la anidación recursiva de una estructura condicional dentro de otra —(1)— hace necesaria la aparición de n conjunciones *si*, y n oraciones que expresen la consecuencia del cumplimiento de una condición²³.

- (1) a. Si ganan el partido pasan a la final
 b. Si si ganan el partido pasan a la final, pasan a la final

Del mismo modo, es posible anidar una oración de relativo en el sujeto de otra oración. La repetición recursiva de esta operación da lugar a una secuencia de n sujetos seguida de sus correspondientes n predicados, como se puede observar en las oraciones de (2).

- (2) a. El concurso ha sido suspendido
 b. El concurso que el departamento había convocado ha sido suspendido
 c. El concurso que el departamento que tu hermana dirige había convocado ha sido suspendido

Valgan estos datos para demostrar que en las lenguas naturales se dan patrones de tipo $a^n b^n$. Como se vio en 1.1.2, estos no pueden ser descritos mediante gramáticas regulares. Estas son las gramáticas más sencillas de la jerarquía de Chomsky, las que acarrearán menor coste computacional, por tanto. Por ello, en principio, sería deseable poder usarlas para la definición de las

²³Para un ser humano resultarían incomprensibles, o muy difíciles de interpretar en tiempo real, oraciones con tres o más anidaciones de este tipo. Es comúnmente aceptado, no obstante, que esto se debe a limitaciones de memoria, y no al hecho de que tales oraciones fuesen agramaticales.

lenguas naturales, pero, como ha quedado demostrado, no tienen la suficiente capacidad generativa débil para ello.

De acuerdo con Chomsky (1957), las lenguas naturales no muestran patrones de complejidad mayor a $a^n b^n$. Si esto es cierto, bastaría con una gramática independiente del contexto —capaz de definir patrones $a^n b^n$ — para formalizar las lenguas naturales adecuadamente, desde el punto de vista observacional.

1.1.4.2. Adecuación descriptiva

En el apartado anterior, se vio cómo las gramáticas regulares no tienen la suficiente capacidad generativa débil para describir adecuadamente las lenguas naturales. En cambio, las gramáticas independientes del contexto sí cuentan con una capacidad generativa débil suficiente. Es decir, en principio, es posible, para una lengua natural dada, definir una gramática independiente del contexto que decida de forma adecuada si una secuencia es o no una oración de tal lengua. Partiendo de este supuesto, en este apartado se discutirá si estas gramáticas son además capaces de describir las lenguas naturales captando las debidas generalizaciones, evitando redundancias y dando una descripción adecuada de las secuencias de la lengua. Se discutirá, en definitiva, si estas gramáticas son herramientas descriptivamente adecuadas para la formalización de las lenguas naturales.

Se entiende por capacidad generativa fuerte²⁴ de una gramática la capacidad que tiene esta para describir de un modo u otro las secuencias de la lengua. Una gramática puede tener la suficiente capacidad generativa débil como para discriminar qué secuencias forman o no parte de una lengua. Pero quizá el análisis que hace de las secuencias en cuestión no es suficientemente rico como para describirlas adecuadamente. Se decía en el apartado anterior que las gramáticas regulares no eran capaces de definir secuencias de tipo $a^n b^n$, presentes en las lenguas naturales. En definitiva, el problema que

²⁴Chomsky (1965), cap. 1.9.

tienen estas gramáticas para la correcta descripción de las lenguas naturales es que las oraciones de las lenguas no son una mera concatenación lineal de palabras. Las palabras, en cambio, dentro de una oración, obedecen a una cierta ordenación interna: se agrupan en sintagmas de forma recursiva hasta formar la oración. Dada esta estructura sintagmática de las oraciones, son posibles en las lenguas naturales patrones de tipo $a^n b^n$. Las gramáticas regulares no son capaces de describir la estructura sintagmática de las oraciones, pues definen las secuencias como una mera concatenación lineal de elementos. Por tanto, las gramáticas regulares no tienen tampoco suficiente capacidad generativa fuerte para la correcta descripción de las lenguas naturales. En cambio, las gramáticas independientes del contexto sí permiten representar la estructura sintagmática de las oraciones, y por ello, en cuanto a este criterio, sí tienen suficiente capacidad generativa fuerte, sí son descriptivamente adecuadas para formalizar las lenguas naturales.

Obsérvese la siguiente gramática independiente del contexto de 1.13. Obsérvese que todas las secuencias que genere esta gramática han de constar de un SN , formado por un D y un N , seguido de un SV , formado a su vez por un V , etc. En una gramática de estas características, puede considerarse que las reglas definen relaciones sintagmáticas: la margen izquierda de la regla denota un sintagma cuyos constituyentes inmediatos son los que muestra la margen derecha. En este sentido, estas gramáticas asignan a las secuencias que generan un análisis en términos de estructura sintagmática.

<i>Axioma</i>	O
<i>Regla1</i>	$O \rightarrow SN SV$
<i>Regla2</i>	$SN \rightarrow D N$
<i>Regla3</i>	$SV \rightarrow V SN$
<i>Regla4</i>	$SN \rightarrow SN SP$
<i>Regla5</i>	$SV \rightarrow SV SP$
<i>Regla6</i>	$SP \rightarrow P SN$

Figura 1.13: Lengua natural I

Asúmase que *vio* es instancia de *V*; *un* y *una* son instancias de *D*; *hombre*, *tortuga* y *telescopio* son instancias de *N* y *con* es instancia de *P*. En tal caso, esta gramática generaría la secuencia *Un hombre vio una tortuga con un telescopio*. Esta oración es ambigua: puede significar que ‘un hombre usó un telescopio para ver una tortuga’, o que ‘un hombre vio una tortuga que tenía un telescopio’. El que la oración reciba una u otra interpretación depende de que se le suponga una u otra estructura sintagmática: si *con un telescopio* complementa al *SV vio una tortuga*, la interpretación será la primera; pero, si complementa al *SN una tortuga*, la interpretación es la segunda. La gramática de 1.13 admite ambas derivaciones, como se observa en la figura 1.14.

<i>Axioma</i>	<i>O</i>	<i>Axioma</i>	<i>O</i>
<i>Regla1</i>	<i>SN SV</i>	<i>Regla1</i>	<i>SN SV</i>
<i>Regla2</i>	[<i>D N</i>] <i>SV</i>	<i>Regla2</i>	[<i>D N</i>] <i>SV</i>
<i>Regla3</i>	[<i>D N</i>] [<i>V SN</i>]	<i>Regla5</i>	[<i>D N</i>] [<i>SV SP</i>]
<i>Regla4</i>	[<i>D N</i>] [<i>V [SN SP]</i>]	<i>Regla3</i>	[<i>D N</i>] [<i>V SN SP</i>]
<i>Regla2</i>	[<i>D N</i>] [<i>V [D N SP]</i>]	<i>Regla2</i>	[<i>D N</i>] [<i>V [D N] SP</i>]
<i>Regla6</i>	[<i>D N</i>] [<i>V [D N [P SN]]</i>]	<i>Regla6</i>	[<i>D N</i>] [<i>V [D N] [P SN]</i>]
<i>Regla2</i>	[<i>D N</i>] [<i>V [D N [P [D N]]]</i>]	<i>Regla2</i>	[<i>D N</i>] [<i>V [D N] [P [D N]]</i>]

Figura 1.14: Derivación en lengua natural I

Obsérvese que se han añadido corchetes que indican la estructura sintagmática que la gramática va asignando a la secuencia a lo largo de la derivación. Es importante el hecho de que estas ambigüedades interpretativas, comunes en las oraciones de las lenguas naturales, se capturan de forma satisfactoria con una gramática independiente del contexto como la de la figura 1.13, capaz de asignar a las secuencias análisis en términos de estructura sintagmática. Para una gramática regular esta ambigüedad es imposible de expresar, por el mero hecho de que no permite tal tipo de análisis.

Dado lo visto hasta el momento, parece que las gramáticas independientes

del contexto, además de ser capaces de generar todas las posibles secuencias de una lengua natural, les asignan a estas análisis adecuados en términos de estructura sintagmática. ¿Hay, no obstante, realidades de las lenguas naturales que este tipo de gramáticas no sean capaces de modelar de forma satisfactoria? A continuación se discutirán dos realidades que pudieran encontrarse en esta situación: la concordancia y el movimiento.

En las lenguas naturales es común el hecho de que diversas partes de una secuencia hayan de coincidir en cuanto al valor de ciertos rasgos. Por ejemplo, en español, en un sintagma nominal, es necesario que los determinantes, el nombre y los adjetivos presenten los mismos valores para sus rasgos de género y número. Del mismo modo, el sujeto y el verbo de una oración deben coincidir en el valor de sus rasgos de persona y número. Las oraciones de (3-b) y (3-c) son agramaticales por no cumplir este requerimiento.

- (3) a. Esos olivos milenarios han visto pasar muchas generaciones
b. *Esas olivos milenarios han visto pasar muchas generaciones
c. *Esos olivos milenarios ha visto pasar muchas generaciones

Las gramáticas independientes del contexto sí son capaces de capturar estos requerimientos de concordancia, como se muestra en la gramática de la figura 1.15. Para ello, es necesario multiplicar las reglas de la gramática de la figura 1.13 de forma que solo sea posible crear un *SN* o un *SV* cuando sus constituyentes tengan los mismos valores para los rasgos pertinentes. Esta solución es claramente indeseable, pues supone una enorme complicación de la gramática —compárense las figuras 1.13 y 1.15—. Además de complicada, la gramática de la figura 1.15 es muy redundante y no capta las debidas generalizaciones: lo que normalmente se entiende como una única relación de pertenencia a un sintagma se desglosa hasta en seis reglas independientes. En conclusión, estos fenómenos de concordancia los puede contemplar una gramática independiente del contexto, pero de un modo que atenta contra los requerimientos más básicos de adecuación descriptiva.

<i>Axioma</i>	O
<i>Regla1</i>	$O \rightarrow SN_{1sg} SV_{1sg}$
<i>Regla2</i>	$O \rightarrow SN_{1pl} SV_{1pl}$
<i>Regla3</i>	$O \rightarrow SN_{2sg} SV_{2sg}$
<i>Regla4</i>	$O \rightarrow SN_{2pl} SV_{2pl}$
<i>Regla5</i>	$O \rightarrow SN_{3sg} SV_{3sg}$
<i>Regla6</i>	$O \rightarrow SN_{3pl} SV_{3pl}$
<i>Regla7</i>	$SN_{3sg} \rightarrow D_{3msg} N_{3msg}$
<i>Regla8</i>	$SN_{3pl} \rightarrow D_{3mpl} N_{3mpl}$
<i>Regla9</i>	$SN_{3sg} \rightarrow D_{3fsg} N_{3fsg}$
<i>Regla10</i>	$SN_{3pl} \rightarrow D_{3fpl} N_{3fpl}$
<i>Regla11</i>	$SN_{1sg} \rightarrow SN_{1sg} SP$
<i>Regla12</i>	$SN_{1pl} \rightarrow SN_{1pl} SP$
<i>Regla12</i>	$SN_{2sg} \rightarrow SN_{2sg} SP$
<i>Regla13</i>	$SN_{2pl} \rightarrow SN_{2pl} SP$
<i>Regla14</i>	$SN_{3sg} \rightarrow SN_{3sg} SP$
<i>Regla15</i>	$SN_{3pl} \rightarrow SN_{3pl} SP$
<i>Regla16</i>	$SV_{1sg} \rightarrow SV_{1sg} SP$
<i>Regla17</i>	$SV_{1pl} \rightarrow SV_{1pl} SP$
<i>Regla18</i>	$SV_{2sg} \rightarrow SV_{2sg} SP$
<i>Regla19</i>	$SV_{2pl} \rightarrow SV_{2pl} SP$
<i>Regla20</i>	$SV_{3sg} \rightarrow SV_{3sg} SP$
<i>Regla21</i>	$SV_{3pl} \rightarrow SV_{3pl} SP$
<i>Regla22</i>	$SP \rightarrow P SN_{1sg}$
<i>Regla23</i>	$SP \rightarrow P SN_{1pl}$
<i>Regla24</i>	$SP \rightarrow P SN_{2sg}$
<i>Regla25</i>	$SP \rightarrow P SN_{2sg}$
<i>Regla26</i>	$SP \rightarrow P SN_{3sg}$
<i>Regla27</i>	$SP \rightarrow P SN_{3pl}$

Figura 1.15: Lengua natural II

De acuerdo con ello, parece necesario el uso de un tipo de gramática más complejo: una gramática dependiente del contexto.

Como se ha visto, estas gramáticas permiten entender las secuencias como objetos multidimensionales —véase 1.1.2.3—, donde estas dimensiones pueden entenderse como los distintos elementos concordantes. No hacen una sola lectura secuencial de las secuencias, sino que pueden retroceder sobre sus pasos en la lectura —véase 1.1.3.3—. De este modo, pueden establecer mecanismos de control de la concordancia. Sin embargo, como también se ha visto, las gramáticas dependientes del contexto tienen un coste computacional sensiblemente mayor al de las independientes del contexto.

La Lingüística moderna maneja un concepto que no se ha introducido aún en la discusión de este trabajo: el concepto de rasgo. Las distintas escuelas (Gramática Transformacional, Gramáticas de Unificación, Gramática Categorial) asumen de un modo u otro que las unidades gramaticales no son objetos atómicos, como se han venido presentado hasta el momento, sino que cuentan con rasgos; puede decirse, incluso, que son haces de rasgos, conjuntos de rasgos. En el ámbito de las Gramáticas de Unificación, se usa esta idea de forma generalizada, y se ha desarrollado toda una teoría totalmente formalizada al respecto. Otras escuelas, como la transformacional, también hacen un uso crucial del concepto de rasgo, si bien no le han dotado del debido aparato teórico. En cualquier caso, el concebir las unidades gramaticales como conjuntos de rasgos puede considerarse actualmente como un logro general de la Lingüística.

Este uso del concepto de rasgo permite tratar fenómenos como la concordancia mediante gramáticas independientes del contexto de forma descriptivamente adecuada, sin caer en problemas de complejidad, redundancia o falta de generalidad, como los vistos anteriormente. Si los objetos gramaticales presentes en una regla son conjuntos de rasgos, es posible establecer en tal regla que ciertos rasgos de dichos objetos gramaticales han de compartir el valor: sea su valor uno u otro, este ha de ser el mismo en todos ellos. De

acuerdo con ello, la *Regla1* de la figura 1.13 se reformularía como se muestra en la figura 1.16, donde se establece que el *SN* y el *SV* que forman *O* han de tener rasgos de persona y número —PER y NUM, respectivamente—. El valor de estos rasgos no se especifica en la regla, pero sí se especifica que, sea cual sea dicho valor, ha de coincidir en *SN* y *SV*.

$$\begin{bmatrix} O \\ \text{PER } \boxed{0} \\ \text{NUM } \boxed{1} \end{bmatrix} \rightarrow \begin{bmatrix} SN \\ \text{PER } \boxed{0} \\ \text{NUM } \boxed{1} \end{bmatrix} \begin{bmatrix} SV \\ \text{PER } \boxed{0} \\ \text{NUM } \boxed{1} \end{bmatrix}$$

Figura 1.16: *Regla1* con rasgos

Se considera que una gramática independiente del contexto que conciba sus unidades gramaticales como conjuntos de rasgos tiene la misma capacidad expresiva que una gramática dependiente del contexto²⁵. Por ello, dado que las gramáticas independientes del contexto acarrearán menor coste computacional, puede entenderse que, de momento, siguen siendo, con el añadido del uso de rasgos, el sistema de formalización de las lenguas naturales más adecuado.

1.1.4.3. Adecuación explicativa

Entre los elementos constituyentes de una secuencia se establecen relaciones de dependencia: los complementos de un determinado núcleo mantienen una relación de dependencia con este. Estas relaciones son locales si los complementos en cuestión se colocan en la secuencia de forma adyacente a su núcleo. Esta es la situación que se percibe como canónica. Pero es común que entre un núcleo y su complemento se dé una relación de dependencia no local: el complemento se explicita en la secuencia alejado de su núcleo. A este

²⁵Véase Joshi (1989). En realidad este autor entiende que el uso de rasgos y unificación permitiría modelar, incluso, máquinas de Turing irrestrictas. Pero los modelos gramaticales, de hecho, añaden restricciones que reducen esta capacidad generativa: de este modo son solo capaces de generar lenguajes dependientes del contexto —véase el apartado 1.3.3.2 de este trabajo—.

hecho se le suele dar el nombre metafórico de movimiento: movimiento desde la posición original adyacente al núcleo hasta la posición alejada de este en la que de hecho se explicita.

- (4) a. Quiere que le podéis muchas ramas al olivo de la entrada
 b. *¿Cuántas ramas_i* quiere que le podéis *h_i* al olivo de la entrada?

Por ejemplo, en (4-a), *muchas ramas* se explicita de forma adyacente a su núcleo: el verbo *podar*. Pero, en (4-b), ese mismo argumento del verbo *podar* —*Cuántas ramas*— se ha movido desde su posición canónica —marcada con *h*— hasta la margen izquierda de la oración. Se trata de un movimiento —propio del español y otras muchas lenguas— de sintagmas interrogativos a la margen izquierda de la oración.

Además, es común que la operación que aleja un complemento de su núcleo pueda aplicarse de forma recursiva, de modo que núcleo y complemento podrían alejarse, en principio, infinitamente. Se dice entonces de esta que es una relación de dependencia no acotada: una relación no local no acotada dentro de un determinado dominio. En (5) se muestra cómo el movimiento del sintagma nominal interrogativo de (4-b) puede efectuarse tantas veces como anidaciones de una oración dentro de otra se den, hasta que el constituyente movido alcanza la margen izquierda de la oración.

- (5) a. *Cuántas ramas_i* le podáis *h_i* al olivo
 b. [*Cuántas ramas_i* quiere [*h_i* que le podéis *h_i* al olivo]]
 c. [*Cuántas ramas_i* dice [*h_i* que quiere [*h_i* que le podéis *h_i* al olivo]]]

Existen otros muchos casos de movimiento en español —topicalización, focalización, salto de clíticos, ubicación de pronombres relativos, estructuras de ascenso, etc.—, y en las lenguas naturales en general. Este es, por tanto, un fenómeno usado recurrentemente en las lenguas.

Téngase en cuenta que las secuencias de las lenguas naturales deben hacer converger dos realidades muy distintas, dos componentes de la actividad cerebral: una secuencia tiene, por un lado, un significado, y por otro una realización fonética. Por tanto, una gramática adecuada desde un punto de vista explicativo debe ser capaz de explicar cómo se concilian en una secuencia estas dos dimensiones. Cabe imaginar que esta pregunta no tendrá una respuesta trivial. Podría definirse el movimiento como la situación en la cual un constituyente se realiza fonéticamente en un lugar distinto a aquel en el que se interpreta —alejado del núcleo del cual es argumento—. Por tanto, el movimiento es un fenómeno en el cual la dimensión fonética de la secuencia y la dimensión de significado no parecen estar bien conciliadas. Cabe preguntarse por qué se da el movimiento, por qué la realización fonética de las secuencias no es un calco perfecto de sus relaciones de significado. La respuesta a estas preguntas queda totalmente fuera de las posibilidades de este trabajo, y, de hecho, la Lingüística no tiene hoy en día una respuesta clara para ellas. Sí se ha aventurado²⁶ que quizá el movimiento sea el modo mediante el cual la gramática concilia dos tipos de significado presentes en las secuencias. Por un lado, en una secuencia se establecen relaciones semánticas argumentales: un núcleo tiene una serie de argumentos. Pero, por otro lado, los constituyentes de las secuencias tienen asociado un significado informativo —tema o rema, información ya conocida o información novedosa²⁷— y mantienen entre sí ciertas relaciones de ámbito. Todo ello tiene que reflejarse en una secuencia que, desde el punto de vista fonético es unidimensional: una mera secuencia lineal de sonidos. Quizá el movimiento sea la solución a este problema: las relaciones argumentales se establecen en las posiciones canónicas de los constituyentes; pero estos se mueven a otras posiciones asociadas a ciertos valores informativos, o desde las cuales pueden establecer determinadas relaciones de ámbito.

La pregunta relevante en este punto es si las gramáticas independientes del contexto aumentadas con rasgos son capaces de modelar el fenómeno del

²⁶Véase Chomsky (2003), pp. 96-97.

²⁷Véase Zubizarreta (1999) o Contreras (1978).

movimiento adecuadamente. Afróntese el problema desde esta perspectiva: el constituyente desplazado en un movimiento debe aparecer en la posición de destino del movimiento y no debe aparecer en la posición original. Es, por tanto, necesario que, en el momento de añadir o no a la secuencia el elemento desplazado en su posición de destino, se pueda recordar si este se añadió previamente en su posición original. Imagínese que la gramática ha generado una oración O a partir de un sintagma verbal transitivo SV . En principio, dicho SV debiera haberse generado mediante la unión de un verbo V y un complemento nominal SN . Si este es el caso, a O no se le podría añadir tal SN a su izquierda, pues ya se ha añadido en su posición canónica de complemento de V . Por el contrario, imagínese que la gramática permite generar SV a partir de un verbo transitivo V , sin unir V a su complemento SN , con la esperanza de que SN se añada una vez generada O , a la izquierda de esta. Tal proceso solo sería legítimo si, de hecho, se añade dicho SN a O . Si no fuera el caso, la secuencia sería ilegítima pues le faltaría el complemento del verbo transitivo V . En resumen, una gramática independiente del contexto necesita recordar, en el momento de la derivación en que se ha generado O , si a V ya se le añadió, o no, su complemento SN . Esto es posible gracias, de nuevo, al uso de rasgos. Es posible darle al SV resultante de la unión canónica de V y SN una caracterización distinta del formado exclusivamente a partir de V . Esta caracterización deben compartirla todas las proyecciones de V : SV y O , en este ejemplo. De este modo, es posible recordar, una vez generada O , si el complemento SN de V apareció ya en su posición canónica o no²⁸.

La alternativa a esta solución es la Gramática Transformacional de Chomsky. Chomsky (1957) defiende que para un modelado adecuado de las lenguas es necesaria una gramática independiente del contexto aumentada con reglas transformacionales. Las reglas de todas las gramáticas mostradas hasta ahora toman como *input* la secuencia resultante de la derivación efectuada hasta ese momento, y generan una nueva secuencia. En cambio, una regla transfor-

²⁸La primera formulación de este tipo de análisis la hizo G. Gazdar para el modelo gramatical GPSG: Gazdar y otros (1985). Véase también Sells (1985) cap. 3, y Moreno Moreno Sandoval (1998), pp. 109-110.

macional toma como *input* la derivación misma. Es decir, estas reglas tienen acceso a la información de la historia derivativa de la secuencia sobre la que operan. Para una regla de este tipo, el movimiento es una realidad muy fácil de modelar. Imagínese que el *input* de una regla transformacional T es una derivación que ha dado lugar a una oración en la que todos los constituyentes aparecen en posición canónica: O . T , como tiene acceso a la historia derivativa de O , conoce que O se ha formado a partir de un SV formado por un V al que se ha añadido como complemento un sintagma nominal SN que es interrogativo. T puede ser una regla tal que, ante un *input* de estas características, elimina SN de su posición canónica y lo añade a la izquierda de O . Es decir, lo que las reglas independientes del contexto hacen en varios pasos —se construye SV sin su complemento, se recuerda este hecho mediante el uso de rasgos, y se añade el SN desplazado a O — una regla transformacional lo lleva a cabo en un solo paso.

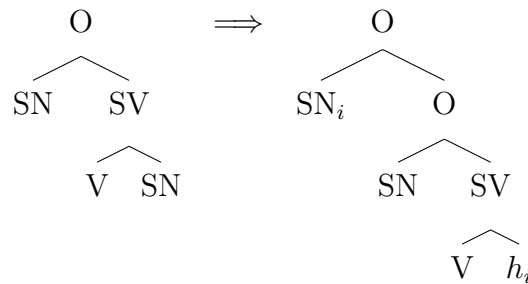


Figura 1.17: Regla transformacional

Dicho de un modo visual, una gramática transformacional entiende como objeto gramatical el árbol resultante de la derivación que ha efectuado, donde las ramas que unen los nodos definen relaciones entre estos. En cambio, para una gramática independiente del contexto, este árbol no existe: solo es una representación gráfica de la historia derivativa. Los objetos gramaticales existentes en una gramática independiente del contexto son los nodos del árbol. Por tanto, cuando una gramática independiente del contexto ha generado una oración O , la única información a la que tiene acceso es la del objeto que ha creado: O . En cambio, cuando una gramática transformacional genera O ,

en realidad ha generado el árbol completo que representa la historia derivativa que ha dado lugar a O . Por ello, dicho árbol, con toda su información, puede ser *input* de una regla transformacional, como se observa en la figura 1.17.

Para una gramática transformacional, por tanto, es muy fácil definir el contexto de aplicación de una regla que modele un movimiento, pues el objeto sobre el que opera —el árbol de O , en este caso— cuenta, por definición, con la información relevante para ello. En cambio, el objeto resultante de una derivación con una gramática independiente del contexto — O , en este caso— no cuenta con dicha información. Por ello es necesario suplir esta carencia con el uso de rasgos.

El problema de estas reglas es que son muy costosas desde un punto de vista computacional. Peters y Ritchie (1973) aportaron una demostración formal de que las gramáticas transformacionales son equivalentes a las máquinas de Turing irrestrictas²⁹. Es decir, son el tipo de gramática más costoso de la jerarquía de Chomsky. Recuérdese que la máquina de Turing es la única máquina abstracta que cuenta con una cinta infinita. En una gramática transformacional, es necesaria una cinta infinita de este estilo para recordar en cada momento del análisis la historia derivativa previa. La cinta del resto de máquinas solo cuenta con tantas celdas como sean necesarias para codificar la cadena de entrada. En ese espacio limitado no podrían memorizarse los pasos intermedios de la derivación. En cualquier caso, parece obvio que el coste de memoria necesario para recordar en todo momento la historia derivativa completa de una secuencia es mucho mayor del necesario para recordar solo su resultado. Y aplicar operaciones sobre un *input* de tal complejidad necesita también tiempos de procesamiento más altos —piénsese, por ejemplo, en la necesidad de decidir si una derivación entera es o no el *input* adecuado para la aplicación de una transformación—.

²⁹Como se verá en el apartado 1.2, esta afirmación habrá de ser matizada.

En resumen, en las lenguas naturales se da de modo generalizado un fenómeno al que se ha llamado movimiento: ciertos constituyentes se explicitan en las secuencias alejados del lugar en el que se interpretan, alejados del núcleo del que dependen. Este fenómeno puede modelarse con éxito mediante gramáticas transformacionales, aquellas que toman como *input* no la secuencia resultante de la derivación previa sino dicha derivación al completo. No obstante, se ha demostrado que estas gramáticas acarrearán un gran coste computacional. Se ha demostrado, por otro lado, que las gramáticas independientes del contexto, aumentadas con rasgos, también son capaces de dar cuenta del movimiento, sin el coste computacional inherente al modelo transformacional. ¿Cuál de los dos modelos es entonces más adecuado para la formalización de las lenguas naturales? Este es un problema de adecuación explicativa. Ambos modelos permiten tratar el movimiento de forma descriptivamente adecuada. Lo que cabe preguntarse es si existen razones para pensar que uno es más fácil de conciliar que el otro con lo que se sabe de la naturaleza del lenguaje, o del funcionamiento del cerebro. Concretamente, en lo sucesivo, se utilizarán como criterios de adecuación explicativa consideraciones relativas al coste computacional de los modelos. Ya se ha visto que a las gramáticas transformacionales se les supone, en tanto que miembros del grupo de gramáticas más complejo de la jerarquía de Chomsky, un gran coste computacional. Las gramáticas independientes de contexto aumentadas con rasgos son, en este sentido, menos costosas. Pero, en la práctica, también tienen problemas de este tipo. En los siguientes apartados se dará una explicación más detenida de ambos modelos, y se intentará dilucidar qué problemas de coste computacional presentan.

1.2. La Gramática Transformacional

En el apartado 1.1.4.3 se presentó la Gramática Transformacional que propuso Chomsky (1957). Se explicó cómo este modelo gramatical utiliza reglas transformacionales, lo cual hace de él un sistema costoso desde el punto de vista computacional. Peters y Ritchie (1973) demostraron que una

gramática transformacional equivale a una máquina de Turing irrestricta. Demostraron que es capaz de generar lenguajes no recursivos. Dado que este tipo de lenguaje solo puede ser generado por una máquina de Turing irrestricta, solo cabe concluir que, efectivamente, las gramáticas transformacionales son equivalentes a este tipo de máquinas de Turing. El hecho de que el modelo gramatical utilizado como teoría del lenguaje natural permita definir lenguajes no recursivos es contrario a las intuiciones que suelen manejar los lingüistas. Se entiende, normalmente, que un hablante es capaz de discriminar, dada una secuencia, si esta forma o no parte de la lengua. Pero si la lengua en cuestión puede ser un lenguaje no recursivo, esto, por definición, no ocurrirá para todas las secuencias posibles.

La demostración de Peters y Ritchie (1973) era válida para la Gramática Transformacional tal como se definía en las primeras formulaciones de esta —ellos remiten a Chomsky (1965)—. Cabe pensar que ciertas restricciones extra pudieran limitar el poder generativo de estas gramáticas, y asegurar así que solo generan lenguajes recursivos. Estos mismos autores aventuran en este artículo cuáles debieran ser estas limitaciones. Además, la Gramática Transformacional ha evolucionado desde 1973. La demostración de estos autores estaba fundamentada en el hecho de que una transformación es capaz de eliminar elementos de una secuencia³⁰. Las formulaciones posteriores de lo que es una regla transformacional —Chomsky (1986b)— eliminan esta posibilidad, lo cual puede hacer pensar que la Gramática Transformacional, con esta nueva formulación, pudiera estar libre de la indeseable generación de lenguajes no recursivos.

De hecho, en Chomsky (1965) se asume que las gramáticas transformacionales y las dependientes del contexto tienen una misma capacidad gene-

³⁰Recuérdese que las reglas de las gramáticas dependientes del contexto, capaces de generar cualquier lenguaje recursivo, solo tienen la restricción de que no pueden reducir la secuencia sobre la que operan: el número de símbolos del *output* ha de ser igual o mayor que el del *input*. Es cuando una gramática permite procesos de reducción de la secuencia cuando puede definir lenguajes no recursivos.

rativa débil³¹. La diferencia entre ambas residiría, entonces, en la capacidad generativa fuerte: en el tipo de análisis que estas son capaces de asignar a las secuencias. Pero esta diferencia en cuanto a capacidad generativa fuerte no es trivial. El coste de memoria y tiempo de procesamiento necesario para trabajar con unidades gramaticales tan complejas como las derivaciones mismas es muy alto.

1.2.1. La Teoría Estándar

En las primeras formulaciones de la Gramática Transformacional (las de Chomsky (1957) y Chomsky (1965)) esta contaba con un juego de reglas independientes del contexto que generaban las estructuras básicas de las secuencias. Sobre estas estructuras se aplicaban las reglas transformacionales para dar lugar a otras estructuras derivadas. Las secuencias de la lengua eran el resultado de la sucesiva aplicación de estos dos grupos de reglas. En 1.1.4.3 se vio cómo el uso de transformaciones soluciona de modo sencillo el problema del movimiento. Pero las transformaciones no se incluyeron en el modelo como una solución *ad hoc* para el movimiento. Son más bien lo que Chomsky entendió como solución óptima a la búsqueda de adecuación descriptiva. Es un hecho el que existen ciertas relaciones sistemáticas entre diversas secuencias de la lengua. Por ejemplo, en español, las oraciones de (6) tienen la misma estructura argumental. La diferencia entre ellas estriba en que una es una oración activa y otra es pasiva, pero su significado es idéntico.

- (6) a. El ingeniero ha encontrado la solución
b. La solución ha sido encontrada por el ingeniero

³¹Chomsky (1965), cap. 2.3.3. y 2.3.4. Recuérdese que las gramáticas independientes del contexto aumentadas con rasgos tienen también la misma capacidad generativa débil que las dependientes del contexto. De acuerdo con ello, el uso de rasgos igualaría la capacidad generativa débil de los tres tipos de gramáticas. No ocurre otro tanto con la capacidad generativa fuerte.

Una gramática que genere estas dos oraciones de modo totalmente independiente —por ejemplo, a partir de unidades léxicas independientes *encontrar* y *haber sido encontrado*— no da cuenta del hecho de que ambas comparten una realidad común. Esta gramática estaría perdiendo generalidad, atentaría contra la adecuación descriptiva. En cambio, una gramática transformacional recoge el hecho de que, desde el punto de vista de significado, (6-a) y (6-b) son idénticas, generando para ambas una misma estructura subyacente mediante reglas independientes del contexto, en la que se representara la realidad de significado. Esta estructura sería básicamente la esperable para la oración de (6-a). En cambio, (6-b) sería el resultado de la aplicación, sobre dicha estructura, de una transformación que convierta una oración activa en pasiva³². Podría decirse que el sistema transformacional es el vínculo que une en las secuencias del lenguaje natural la dimensión de significado y la realización fonética. La realidad de significado de una secuencia quedaría representada mediante la estructura subyacente creada con reglas independientes del contexto. Pero el desfase que existe entre esta realidad y la realización fonética de la secuencia lo regula el componente transformacional.

Durante las primeras etapas de la Gramática Transformacional —lo que se conoce como Teoría Estándar— se elaboraron gramáticas descriptivamente adecuadas de las lenguas. Para alcanzar la debida cobertura, estas gramáticas contaban con muchas reglas, diferentes entre lenguas. Además, estos sistemas de reglas modelaban directamente las construcciones que había contemplado hasta entonces la tradición gramatical: había reglas *ad hoc* para modelar, por ejemplo, las oraciones pasivas, las construcciones de relativo, las interrogativas, etc. Cada lengua tenía entonces sus construcciones propias modeladas directamente con muy distintas reglas específicas para esa lengua. Aunque se alcanzó un gran nivel de adecuación descriptiva, esta situación atentaba contra la adecuación explicativa, tal como la entendía Chomsky. La lingüística chomskyana siempre ha asumido como criterio crucial de adecuación explicativa la conocida hipótesis del innatismo: habida cuenta de que

³²O quizá una sucesión de transformaciones.

el ser humano desarrolla el lenguaje en un corto espacio de tiempo, y con unos datos extremadamente pobres, parece necesario suponer que el grueso de dicho conocimiento es innato. De acuerdo con esto, las distintas lenguas naturales no son sino instancias de una misma Gramática Universal. Las diferencias entre estas han de ser, en realidad, relativamente pequeñas. No resulta entonces adecuada una teoría gramatical que estipula un conjunto enorme de reglas muy distintas entre lenguas. Será necesaria una teoría en la cual las muy diversas construcciones propias de cada lengua se expliquen como epifenómenos deducibles de un aparato teórico común y sencillo.

1.2.2. El modelo de Principios y Parámetros

Tras la estela de las ideas vistas en 1.2.1, la Gramática Transformacional evolucionó desde finales de los años sesenta, hasta dar lugar a la llamada Teoría de Principios y Parámetros —tómense como referencia Chomsky (1986a) y Chomsky (1986b)—. En ella se abandona la idea de construcción —construcción pasiva, construcción de relativo, interrogativa, etc.— como objeto gramatical, como unidad de análisis. Estas pasan a considerarse epifenómenos resultantes de la convergencia de principios gramaticales mucho más abstractos, generales, comunes a las distintas lenguas. Se prescinde también del complejo conjunto de reglas de la Teoría Estándar. Las posibilidades de combinación de unidades gramaticales están solo restringidas por principios universales. Por ejemplo, en la Teoría Estándar la construcción de sintagmas estaba regulada por reglas independientes del contexto que establecían los constituyentes inmediatos de cada tipo de sintagma. Para distintos tipos de sintagma podían establecerse distintos tipos de constituyentes. En el modelo de Principios y Parámetros se establece que cualquier estructura sintagmática de cualquier lengua humana ha de respetar sencillamente lo establecido en la llamada Teoría de la X con Barra: todo sintagma es el resultado de dos niveles de proyección de su núcleo, uno logrado mediante la unión de este con su complemento y otro mediante la unión con su especificador. Del mismo modo, el conjunto de reglas transformacionales de la

Teoría Estándar —capaces de eliminar un elemento de una estructura sintagmática, cambiarlo de lugar o permutar su posición con otro elemento de la estructura— se sustituyeron por la única operación transformacional posible, consistente en mover un constituyente dentro de la estructura sintagmática. La posibilidad de aplicación de esta operación está constreñida por diversos principios universales.

El modelo se llamó de Principios y Parámetros porque concibe la gramática como un conjunto universal de principios universales que regulan los procesos generativos, y un conjunto de parámetros que explican las diferencias entre las lenguas. Las lenguas del mundo difieren entre sí en tanto que toman valores distintos para ciertos parámetros de la Gramática Universal. Además, se ha propuesto que las diferencias paramétricas residen exclusivamente en las unidades léxicas³³. Los principios gramaticales que regulan la combinatoria de unidades léxicas y los procesos de movimiento son comunes a toda lengua. Estos principios universales dan resultados distintos debido a que se aplican sobre unidades léxicas diferentes.

Cabe hacer ciertas consideraciones relativas al coste computacional de este nuevo modelo gramatical. En primer lugar, el hecho de que las transformaciones se reduzcan al movimiento de constituyentes dentro de la estructura sintagmática es un hecho relevante. En el modelo anterior existía también la posibilidad de que una regla transformacional eliminase un elemento. Como ya se ha dicho, esta situación hacía de las gramáticas transformacionales sistemas generativos capaces de definir lenguajes no recursivos. El hecho de que en el modelo de Principios y Parámetros no existan transformaciones de este tipo elimina dicha posibilidad, lo cual es deseable.

Sin embargo, el hecho de que el modelo gramatical haya pasado de estar fundamentado en reglas a estar fundamentado en principios de corrección

³³En las unidades léxicas tradicionales y en las categorías funcionales. Véase Eguren y Fernández Soriano (2004), pp. 205-206.

puede acarrear problemas de índole computacional. En este modelo, las operaciones gramaticales —combinación o movimiento de constituyentes— se pueden aplicar, en principio, libremente. Darán lugar a resultados legítimos o no en función de que se atengan a determinados principios generales de corrección. El problema consistiría en que el filtro que establece la legitimidad de las operaciones se aplicara *a posteriori*. La situación ideal, desde un punto de vista de coste computacional, es aquella en que la legitimidad de la aplicación de una operación se puede valorar en el momento mismo de su aplicación. Si este es el caso, si se establece que la operación en cuestión es ilegítima, sencillamente no se lleva a cabo, con lo cual el proceso ahorra un coste de tiempo y memoria inútil. Pero, si no es posible valorar la legitimidad de las operaciones en el momento mismo de llevarlas a cabo, el sistema se verá obligado a aplicarlas de forma ciega, dando lugar a resultados que *a posteriori* se revelarán inútiles, con el consecuente coste de tiempo y memoria.

Precisamente, en el modelo de Principios y Parámetros, se establecen hasta cuatro niveles de representación que actúan como filtro al resultado de las operaciones gramaticales. El proceso de generación de una secuencia es, en este modelo, como sigue. Las unidades léxicas se combinan de acuerdo con los principios de la Teoría de la X con Barra. Este proceso da lugar a la llamada Estructura Profunda. A continuación, se aplica a esta Estructura Profunda la operación de movimiento, de lo cual resulta la Estructura Superficial. Estructura Profunda y Estructura Superficial se conciben como niveles de representación sujetos a condiciones de buena formación. Una vez creada la Estructura Superficial, entran en juego dos nuevos niveles de representación: la Forma Lógica y la Forma Fonética. Son los niveles en los que se interpreta semántica y fonéticamente el resultado de la computación lingüística. Actúan como interficie entre el componente computacional propiamente lingüístico y los módulos de la mente que interactúan con este.

El proceso, por tanto, ha perdido su carácter derivacional y toma un carácter representacional. Las operaciones que generan una Estructura Profunda o una Estructura Superficial son muy sencillas: un proceso combina-

torio de unidades del léxico y la posibilidad de mover un elemento dentro de la estructura. En principio, desde un punto de vista derivativo, es posible aplicar estas operaciones de cualquier modo. Ahora, el resultado de estas operaciones será o no legítimo según se atenga o no a las condiciones de buena formación de los niveles de representación (Estructura Profunda, Estructura Superficial, Forma Lógica y Forma Fonética). Desde un punto de vista de coste computacional, sería fatal que la gramática pudiese aplicar libremente las operaciones, y solo después valorar si el resultado es legítimo conforme a lo establecido en los niveles de representación. Esto permitiría un número ingente de generaciones que a la postre se revelarían inútiles. Sería necesario, en cambio, que las condiciones de buena formación establecidas por los niveles de representación guiasen la aplicación de las operaciones. Por supuesto, no está claro cómo puede conseguirse este efecto. La Teoría de Principios y Parámetros carece de la formalización matemática necesaria para plantear estas cuestiones. No obstante, el valor que tiene este modelo es el de haber logrado explicar el lenguaje humano en términos lo suficientemente abstractos como para lograr una generalidad y sencillez compatibles con las tesis innatistas.

1.2.3. El Programa Minimista

La Teoría de Principios y Parámetros supuso un gran avance en la investigación lingüística. Desde un mero punto de vista metodológico, se logró una teoría sencilla y muy generalizadora de las lenguas naturales. Desde el punto de vista de la búsqueda de adecuación explicativa, se logró una teoría compatible con la hipótesis universalista del innatismo. Sin embargo, los principios de este modelo gramatical tienen un fuerte carácter estipulativo. Aun con un alto nivel de abstracción, y con ello de generalidad y capacidad predictiva interlingüística, no son sino meras descripciones de los hechos observados en las lenguas: se observa que en las lenguas del mundo solo es posible el movimiento en tales situaciones estructurales, se observa que las anáforas tienen que estar ligadas por un antecedente dentro de una determinada configuración

estructural, se observa que los sintagmas nominales con realización fonética deben haber recibido una marca de caso, etc.

Dada esta situación, el Programa Minimista —Chomsky (1995)— intenta reformular la gramática de tal forma que sus principios se deduzcan directamente de aquello que se sabe, o de las hipótesis que se establecen, sobre el funcionamiento del lenguaje en el cerebro. Hasta este momento, se pretendía elaborar una gramática compatible con las hipótesis existentes sobre la naturaleza del lenguaje: innatismo, universalismo... Pero los principios gramaticales mismos no tenían una motivación directa en tales hipótesis. En el Programa Minimista se intenta no solo lograr una gramática compatible con las hipótesis establecidas sobre el lenguaje humano. Se intenta también que los principios mismos de la gramática tengan una motivación independiente en dichas hipótesis.

Entonces, ¿qué se sabe del funcionamiento del lenguaje humano en el cerebro, es decir, qué criterios de adecuación explicativa se podrían usar? Se conjetura que el lenguaje debe ser una facultad de la mente innata y por tanto universal. La Teoría de Principios y Parámetros ya desarrolla una gramática compatible con estas hipótesis. Además, se sabe que el lenguaje humano consiste, en esencia, en un proceso recursivo de combinación de símbolos. Esta característica se encuentra en el corazón de su naturaleza. Por tanto, parece legítimo afirmar que la gramática debe contar con una operación combinatoria de este tipo: es una necesidad conceptual. ¿Qué más criterios de adecuación explicativa cabe utilizar? Se sabe que las secuencias resultantes de la operación combinatoria son susceptibles de interpretación en términos de significado y en términos de sonidos. Por tanto, puede decirse que es necesario que el resultado de tal operación combinatoria sea interpretable por los módulos de la mente encargados de gestionar el significado y el sonido. Por último, los cómputos lingüísticos, que se antojan complejos, han de poder efectuarse en tiempo real. Por tanto, también se sabe que el lenguaje ha de tener una complejidad computacional limitada.

Posiblemente no se sepa mucho más sobre el funcionamiento del lenguaje en el cerebro. El Programa Minimista parte, entonces, de estos supuestos, y solo se permite establecer principios gramaticales que se puedan entender como motivados por ellos. Desde este planteamiento, el lenguaje humano se concibe como una realidad muy sencilla, pues no pueden estipularse le características caprichosas de ningún tipo. En principio, esta hipótesis parece incompatible con la evidencia empírica: las lenguas humanas presentan fenómenos complejos como es el movimiento, continuas redundancias como la concordancia, etc. La hipótesis de Chomsky (1995) es que quizá las características aparentemente caprichosas del lenguaje no sean sino el modo más sencillo de lograr que el resultado de la computación lingüística dé lugar a un *output* interpretable por los dos módulos del cerebro que tienen que interactuar con ella. Asumida esta hipótesis, en la gramática del Programa Minimista solo se podrán definir operaciones que sean conceptualmente necesarias (el lenguaje es en esencia un proceso recursivo de combinación de símbolos) o que se presenten como necesarias para lograr un *output* de la computación interpretable por los módulos externos de la mente. Movimiento y concordancia, por ejemplo, los entiende Chomsky (1995) como el mecanismo óptimo para eliminar ciertos rasgos de los símbolos lingüísticos que no son interpretables por estos módulos externos: rasgos funcionales, como las marcas de caso o los rasgos de concordancia redundantes. Concretamente, en el Programa Minimista es clave el llamado Principio de Interpretación Plena, que establece que el resultado de la computación lingüística no puede incluir rasgos no interpretables por los módulos de la mente con los que esta interactúa: el módulo del significado solo reconoce estructuras de tipo predicado-argumento, u operador-variable; y el módulo del sonido reconoce estructuras fonéticas. Los diversos principios gramaticales que se establezcan han de entenderse, entonces, como estrategias para alcanzar el Principio de la Interpretación Plena.

La idea de que el lenguaje es un sistema sencillo, que no tiene peculiaridades idiosincrásicas, es una hipótesis necesitada de confirmación. En caso de ser correcta, el lenguaje sería un sistema extraño entre los sistemas biológicos,

caracterizados por la complejidad y la redundancia. Pero, aun en caso de no ser cierta la hipótesis, como método de trabajo es interesante. En principio, sería posible definir distintas gramáticas con la adecuación explicativa que manejaba la Teoría de Principios y Parámetros. Pero la Lingüística es una ciencia empírica, a la que le interesa la configuración real del lenguaje humano en el cerebro. Por tanto, es interesante un programa de investigación que se pregunte si la cobertura empírica de los principios desarrollados en la Teoría de Principios y Parámetros puede alcanzarse con un aparato teórico mínimo, no estipulativo, con motivación independiente en hipótesis generales sobre la naturaleza y funcionamiento del lenguaje en el cerebro. Quizá se alcance la conclusión de que no se puede explicar el lenguaje humano sin entender que este cuenta con características idiosincrásicas caprichosas, redundancias, etc. Pero para alcanzar tal conclusión es necesario evitar antes cualquier estipulación carente de motivación independiente en certezas o hipótesis falsables sobre el funcionamiento del lenguaje.

Además del Principio de Interpretación Plena, el Programa Minimista cuenta con ciertas máximas de economía. Se observa que ciertas derivaciones, en principio legítimas —convergentes con el Principio de Interpretación Plena—, quedaban bloqueadas por otras derivaciones parecidas, pero que, en cierto sentido bien definido, resultaban más económicas, de un coste computacional menor. La intuición que hay detrás de esta afirmación es que, a partir de un conjunto cerrado de unidades léxicas, el sistema de computación lingüística tiende a buscar la solución más económica para construir una secuencia determinada; y solo la solución óptima, en este sentido bien definido, es gramatical. Estas máximas de economía juegan el mismo papel que el Principio de Interpretación Plena, pues son principios defnitorios de la gramática fundamentados en hipótesis independientes sobre el funcionamiento del lenguaje: la necesidad de que la producción lingüística tenga un coste computacional limitado. No obstante, la noción de economía utilizada por la teoría no puede ser una valoración general sobre el coste computacional real de las derivaciones. Tiene que consistir, en cambio, en máximas muy precisas que, en un momento dado de la derivación, hagan imposible la aplicación de la operación

α por la existencia de una operación posible β más económica en un sentido preciso. Por ejemplo, parece que, si un constituyente Σ se mueve en la estructura desde una posición original ω , no podrá alcanzar una posición de destino δ_1 si entre ω y δ_1 hay una posición de destino δ_2 que es un destino posible para Σ . En resumen, parece que la gramática prima el movimiento más corto posible.

Por último, en su afán por evitar toda afirmación estipulativa, la teoría se reformula restringiendo de forma drástica el aparato teórico del modelo de Principios y Parámetros. Todos los principios de este, los niveles de representación, las subteorías, etc. se ponen en cuestión en tanto que primitivos gramaticales, se eliminan y se intenta deducir su contenido empírico como consecuencia necesaria de las operaciones gramaticales fundamentales, o como requerimientos del Principio de Interpretación Plena o de las máximas de economía. Fruto de este proceso es la eliminación de los niveles de representación de Estructura Profunda y Estructura Superficial, pues se observa que las condiciones de buena formación que se definían mediante estos niveles se pueden deducir de forma independiente. Sí se mantienen, en cambio, la Forma Lógica y la Forma Fonética, pues se entiende que son necesarios en tanto que interficie con los módulos externos de la mente. Son el lugar natural donde se aplica el Principio de Interpretación Plena. De hecho, la gramática, con el Programa Minimista, recupera su carácter eminentemente derivacional original. La operación recursiva de combinación de símbolos se aplica sucesivamente, paso a paso. Se minimiza el uso del nivel de representación como filtro que valora la legitimidad de una derivación. Se comienza a observar una preocupación³⁴ por la necesidad de que la valoración de la legitimidad de una operación se haga de forma local, en el momento mismo de su aplicación, y no *a posteriori*, en un nivel de representación. El modelo de fases —Chomsky (2008)— contribuye a la reducción del problema. Dentro de una fase, los objetos gramaticales son derivaciones, pero una vez cerrada esta, la computación no tendrá acceso a su historia derivativa. Sin embargo, siguen

³⁴Chomsky (1995), cap. 1.7.

entendiéndose la Forma Lógica y la Forma Fonética como niveles de representación que filtran las derivaciones que atentan contra el Principio de Interpretación Plena. También se han formulado máximas de economía que bloquean derivaciones completas (no operaciones concretas) por la existencia de otras derivaciones que, en conjunto, resultan más económicas. Mientras sea posible crear derivaciones de forma ciega confiando en que al final del proceso existe un método de evaluación de los resultados (un nivel de representación o una comparación de varias derivaciones en términos de economía) el sistema tendrá un coste computacional inasumible.

1.2.4. Conclusión

La Gramática Transformacional, desde su origen hasta la actualidad, ha entendido que el objeto sobre el que actúan las operaciones gramaticales son, no los sintagmas resultantes de las derivaciones, sino las derivaciones mismas. Es decir, los árboles completos donde cada sintagma cuenta con la información de sus constituyentes inmediatos. Efectivamente, existen fenómenos muy importantes de las lenguas naturales que hacen necesario tomar como objeto gramatical la derivación completa y no su resultado. Por ejemplo, el fenómeno del movimiento relaciona dos pasos diferentes de la derivación: los de las posiciones de origen y destino del elemento desplazado. El problema es que una gramática de este estilo tiene un coste computacional muy elevado. Aun asumiendo que —a partir de la Teoría de Principios y Parámetros— estas gramáticas no generan lenguajes no recursivos, es evidente que hacer que sea la derivación completa el objeto gramatical que se toma como *input* de las reglas tiene un alto coste computacional. En primer lugar, supone un alto coste de memoria. En segundo lugar, efectuar cualquier operación sobre un *input* de tal complejidad es muy costoso también en términos de tiempo.

1.3. Las Gramáticas de Unificación. HPSG

En el apartado 1.2 se ha presentado la Gramática Transformacional como una teoría del lenguaje natural adecuada desde el punto de vista descriptivo. Se ha discutido si también era adecuada desde el punto de vista explicativo, utilizando la complejidad computacional como criterio. Se ha visto que las reglas transformacionales son muy costosas en términos computacionales. ¿Habrá, por tanto, que optar por una gramática de estructura sintagmática independiente del contexto?

En este apartado se describe un modelo gramatical con reglas de este tipo: las Gramáticas de Unificación. Estas nacieron en los años 80, en buena medida, con el fin de dar respuesta a los problemas de procesamiento en tiempo real. Son, entonces, teorías del lenguaje humano que, frente al modelo chomskyano, buscan la moderación del coste computacional, así como la adecuación a ciertas características observables del procesamiento lingüístico.

Las Gramáticas de Unificación más importantes son GPSG, LFG y HPSG. GPSG (*Generalized Phrase Structure Grammar*) fue desarrollado por Gazdar, Pullum y Sag —Gazdar y otros (1985)—. Hoy se encuentra en desuso. En cambio, LFG (*Lexical Functional Grammar*) se encuentra en pleno desarrollo —Dalrymple y otros (1995) y Bresnan (2001)—. HPSG (*Head-Driven Phrase Structure Grammar*) es la evolución de GPSG, y será ampliamente estudiada en el siguiente apartado.

1.3.1. Características generales

Las Gramáticas de Unificación —véanse Shieber (1986) y Pollard y Sag (1994)³⁵— cuentan con las siguientes características.

- Están basadas en la superficie. Caracterizan directamente el orden lineal de constituyentes. Solo utilizan, por tanto, reglas de estructura

³⁵Pollard y Sag (1994), pp. 6-14.

sintagmática independientes del contexto. Como ya se ha visto, estas son reglas menos costosas que las transformaciones y, por ello, en principio, más adecuadas que estas en términos de adecuación explicativa.

- Son informacionales. Contemplan el lenguaje humano como un sistema de codificación y transmisión de información. Las secuencias de las lenguas se entienden, entonces, como elementos informacionales. También cada subsecuencia de una secuencia (en última instancia cada símbolo atómico) se contempla como un elemento informacional. Como se verá con detalle en 1.3.2, estos elementos informacionales tienen una definición precisa y formalizada: son estructuras de rasgos.
- El proceso de generación de secuencias es una combinación recursiva de elementos informacionales que da lugar a nuevos elementos, informacionalmente más específicos. Si los elementos informacionales se han formalizado como estructuras de rasgos, las posibilidades de combinación de estos están reguladas por la operación matemática de unificación de estructuras de rasgos. Por tanto, en las Gramáticas de Unificación, la descripción de los elementos gramaticales y la de sus posibilidades de combinación están totalmente formalizadas. Esta formalización hace de ellas teorías útiles para el trabajo en Lingüística Computacional.
- Son declarativas. Se define la combinación de subsecuencias en términos de qué combinaciones son permitidas, qué estructuras de rasgos unifican, y no de cómo se calculan. Se desvincula así la definición de la gramática del algoritmo de derivación de secuencias. En el modelo chomskyano estas dos realidades están unidas: la gramática explica cómo se va generando la secuencia a lo largo de la derivación. Una gramática de unificación, en cambio, solo es una declaración de qué elementos son compatibles. Esta declaración puede ser utilizada después por un algoritmo u otro (un *parser* u otro) para calcular en tiempo real las secuencias del lenguaje que define. Resulta natural, así, que una misma gramática pueda ser utilizada tanto en labores de procesamiento como de generación.

- Son monoestráticas. Cada secuencia y cada subsecuencia (cada elemento informacional) presenta en paralelo la información relativa a cada nivel lingüístico (fonológico, morfológico, sintáctico, semántico y contextual). No hay, por tanto, como en el modelo chomskyano, un orden en el cual se va solucionando cada uno de estos niveles, sino que todos se solucionan en paralelo. Todos ellos se formalizan como partes de la misma estructura de rasgos que es un elemento informacional.

Pollard y Sag (1994) aseguran que, frente a la gramática chomskyana, las de unificación, dadas estas características, logran tiempos de análisis razonables; y, además, explican de forma natural las siguientes propiedades del procesamiento lingüístico cerebral.

- El procesamiento lingüístico es incremental: los hablantes realizan interpretaciones parciales a medida que procesan la oración. Esto lo capta una gramática de unificación, pues sus símbolos son elementos informacionales parciales. Por tanto, una secuencia incompleta es, aun de forma parcial, informativa. En la gramática chomskyana, en cambio, es necesario haber procesado toda la oración para lograr una interpretación.
- El procesamiento lingüístico es también integrador: integra información lingüística, conocimiento del mundo y del contexto comunicativo. Cada uno de estos tipos de conocimiento establece una serie de restricciones, que conducen a elegir la interpretación apropiada entre varias posibles. Las teorías gramaticales actuales no combinan satisfactoriamente conocimiento lingüístico y extralingüístico. Pero, como formalismo, las estructuras de rasgos y su combinación mediante la unificación permitirían tal integración.
- Por último, según Pollard y Sag (1994), ha sido probado por experimentos psicolingüísticos que el orden en el cual se accede a la información está estrechamente ligado al orden de aparición de palabras. No se soluciona primero la fonología, después la morfología, la sintaxis y por

último la semántica, sino que a este conocimiento se accede en paralelo. Esta realidad la captan también las Gramáticas de Unificación, pues, en ellas, los distintos niveles lingüísticos forman parte de una misma estructura de rasgos.

Por tanto, si se utiliza como criterio de adecuación explicativa la necesidad de sencillez computacional, o el hecho de que el procesamiento lingüístico humano sea incremental, integrador o monoestrático, las Gramáticas de Unificación parecen más adecuadas que el modelo chomskyano. En los siguientes subapartados se explican con detalle ciertas cuestiones importantes que ya se han enunciado aquí. En 1.3.2, se explicarán los conceptos de rasgo, de estructura de rasgos y de unificación. En 1.3.3, se explicará la que es la gramática de unificación más utilizada en Lingüística Computacional, la utilizada en el resto de este trabajo: HPSG.

1.3.2. Estructuras de rasgos y unificación

Las Gramáticas de Unificación utilizan una definición formal muy precisa de qué es un elemento informacional, y de cuáles son las condiciones de combinación de dos elementos informacionales. Para ello se utilizan los conceptos de rasgo, estructura de rasgos y unificación³⁶.

El concepto de rasgo, de un modo u otro, es ampliamente utilizado por las distintas escuelas lingüísticas, pero las Gramáticas de Unificación hacen un uso generalizado y totalmente formalizado de él. Todo elemento informacional (presente en una secuencia o contemplado por una Gramática de Unificación) se concibe no como un ente atómico, sino como una estructura de rasgos. Una estructura de rasgos es un conjunto de rasgos, con un cierto valor informativo. Cada rasgo de ese conjunto es un par atributo valor: $F v$, en la figura 1.18³⁷.

³⁶Kasper y Rounds (1986) y Carpenter (1992).

³⁷Se seguirá a partir de ahora la convención habitual consistente en escribir los atributos en mayúscula y los valores, si fueran atómicos, en minúscula.

$$[F \ v]$$

Figura 1.18: Rasgo F

Un atributo F es un símbolo (un átomo, sin estructura interna) que da nombre al rasgo. Su valor v puede ser o un átomo o una estructura de rasgos. En la figura 1.19 se observa el rasgo F_1 , cuyo valor v_1 es una estructura de rasgos formada por el rasgo F_2 , cuyo valor es, a su vez, un elemento atómico: v_2 .

$$\left[F_1 \left[\begin{array}{l} v_1 \\ F_2 \ v_2 \end{array} \right] \right]$$

Figura 1.19: Rasgo F con valor complejo

Kay (1979) fue el primero en establecer una definición e implementación precisas de la noción de rasgo. Para entonces era muy común encontrar rasgos en las descripciones lingüísticas, pero solo con valores atómicos. Dicho trabajo es el que introduce la posibilidad de que, junto a los valores atómicos, se utilicen valores complejos (estructuras de rasgos). Esto permite contemplar un único tipo de unidades —los rasgos— válido para todos los niveles lingüísticos; de modo que toda categoría, toda realidad lingüística (las categorías clásicas, la noción de oración, la de concordancia, la de subcategorización, etc.) puede describirse en términos de estructuras de rasgos.

Puede decirse que los rasgos son funciones, de tal modo que solo es posible asignar un valor en una situación dada a un atributo. Por ejemplo, pudiera definirse para objetos lingüísticos un rasgo NUM (número) cuyo valor pudiese ser *sg* (singular) o *pl* (plural). En una determinada estructura de rasgos dicho valor pudiera no estar especificado, o pudiera ser *sg* o *pl*, pero nunca ambos al tiempo.

Por último, dentro de una estructura de rasgos, es posible estipular que dos rasgos —A y B— compartan valor. Es decir, se puede establecer para

dos rasgos A y B, que el valor de estos, sea uno u otro, ha de ser siempre el mismo³⁸. A partir de ahora se llamará a esta situación estructura compartida —Shieber (1986), *feature sharing*—. La estructura compartida tiene una aplicación obvia para modelar la concordancia en el lenguaje natural, pero en las Gramáticas de Unificación se hará un uso mucho más rico de ella³⁹.

$$\begin{bmatrix} A & \boxed{1} \\ B & \boxed{1} \end{bmatrix}$$

Figura 1.20: Rasgo F con estructura compartida

Las estructuras de rasgos cumplen una función informacional. Por ello, es posible hacer descripciones de objetos (objetos lingüísticos, por ejemplo) mediante estructuras de rasgos. Estas son descripciones parciales: la información total asociada a una secuencia es fruto de la combinación de la información parcial de cada una de las subsecuencias que la constituyen. La unificación es la operación matemática que regula esta combinación. Para explicar el concepto de unificación, es necesario explicar antes la relación de subsunción entre dos estructuras de rasgos.

$$\begin{bmatrix} D \\ A & a \end{bmatrix} \sqsubseteq \begin{bmatrix} D' \\ A & a \\ B & \begin{bmatrix} C & c \end{bmatrix} \end{bmatrix}$$

Figura 1.21: D subsume a D'

Si a una estructura de rasgos se le añade información (se le suman rasgos, se le especifica el valor de los que ya tiene, o se le introduce una situación de estructura compartida), se obtiene una estructura más informativa, más

³⁸No se confunda esta situación con lo que Shieber (1986) llama rasgos semejantes. Dos rasgos son semejantes, en una situación dada, si están evaluados del mismo modo. Pero esta coincidencia en los valores es casual, no motivada por estipulación alguna.

³⁹Se seguirá desde ahora la convención de representar las situaciones de estructura compartida entre dos rasgos añadiendo al valor del rasgo el mismo índice numérico recuadrado.

específica que la original. Se dice que una estructura de rasgos D subsume a otra D' ($D \sqsubseteq D'$) si la información de D es un subconjunto de la información contenida en D' : figura 1.21.

Imagínense ahora dos estructuras de rasgos A y B tales que no se encuentran en relación de subsunción: es decir, A y B tienen información diferente, compatible o no. Se dice que estas dos estructuras de rasgos A y B unifican ($A \sqcup B$) si existe una tercera estructura de rasgos C a la que subsumen A y B ($A \sqsubseteq C$ y $B \sqsubseteq C$). Dicho de un modo intuitivo, la unificación es la operación de combinación de información de dos o más estructuras de rasgos que da lugar a una nueva estructura de rasgos que contiene la información de todas ellas.

$$\begin{bmatrix} A \\ F_1 \ v_1 \\ F_2 \ [F_3 \ v_2] \end{bmatrix} \not\sqcup \begin{bmatrix} B \\ F_2 \ [F_3 \ v_3] \\ F_4 \ v_4 \end{bmatrix}$$

Figura 1.22: A y B no unifican

Esto es solo posible entre estructuras que contienen información compatible. Por ejemplo, en la figura 1.22, A y B no unifican, pues estas estructuras tienen valores incompatibles para el rasgo F_3 . Pero A y B sí unifican en la figura 1.23, pues, en este caso, no aparecen valores incompatibles.

$$\begin{bmatrix} A \\ F_1 \ v_1 \\ F_2 \ [F_3 \ v_3] \end{bmatrix} \sqcup \begin{bmatrix} B \\ F_2 \ [F_3 \ v_3] \\ F_4 \ v_4 \end{bmatrix}$$

Figura 1.23: A y B unifican

Hay que añadir que el resultado de la unificación no es cualquier estructura más informativa que las estructuras que se pretende unificar. Es, en cambio, aquella estructura menos informativa posible que es subsumida por todas las estructuras unificadas.

Las propiedades de la unificación son las siguientes.

- Idempotencia $D \sqcup D = D$.
- Conmutativa $D \sqcup B = B \sqcup D$.
- Asociativa $(D \sqcup B) \sqcup C = D \sqcup (B \sqcup C)$.

Estas propiedades son importantes, porque implican que la información asociada con la estructura unificada no depende del orden en que se obtiene: una gramática de unificación es una mera declaración de condiciones de compatibilidad de subsecuencias, donde no se especifica el orden de unificación. Será un algoritmo, externo a la gramática misma, el que opere dichos cálculos en un orden determinado.

1.3.3. HPSG

HPSG (*Head-Driven Phrase Structure Grammar*, o gramática sintagmática nuclear) surge a mediados de los 80. Es una teoría ecléctica, que recoge influencias de otras teorías lingüísticas y disciplinas relacionadas (filosofía del lenguaje, lógica y computación)⁴⁰. Puede decirse que hoy en día es el formalismo de unificación de aplicación más extendida en Lingüística Computacional.

Las obras de referencia son Pollard y Sag (1987), Pollard y Sag (1994) así como Sag y otros (2002). Como puede imaginarse, la formalización de la gramática mediante estructuras de rasgos varía inevitablemente de una obra a otra. En los subapartados que siguen, se irá creando progresivamente una gramática de tipo HPSG. La formalización de esta gramática es claramente deudora de estos trabajos clásicos, y también sigue ciertas convenciones propias de la gramática computacional **Matrix** —Bender y otros

⁴⁰En el campo semántico, la influencia viene dada por la semántica de situaciones —Barwise y Perry (1983), Devlin (1991)—, y de la teoría de representación del discurso o DRT —Kamp y Reyle (1993)—. En cuanto a las teorías gramaticales influyentes, hay que mentar a la Gramática Categorial, LFG, *Government and Binding*, *Arc Pair Grammar* —Johnson y Postal (1980)— y, de manera especial, GPSG, teoría de la cual es evolución directa.

(2002)—. Allí donde estas referencias discrepan, se ha optado por la solución más ajustada a los intereses de este trabajo. En definitiva, la formalización concreta que sigue, no es, en rigor, la explicación de ninguna de las versiones clásicas de HPSG, si bien las sigue muy de cerca. Puede considerarse que es ya la explicación del corazón de la gramática SSG que es objeto fundamental de este trabajo.

1.3.3.1. El signo lingüístico

Los elementos informacionales mentados en 1.3.1 se entienden en HPSG como signos lingüísticos. Un signo lingüístico es una asociación de significante y significado, sea una palabra, un sintagma, una oración o un discurso⁴¹.

Esta asociación se formaliza mediante una estructura de rasgos, en la que aparecen rasgos que caracterizan la naturaleza fonética y fonológica del signo; y otros que caracterizan su morfología, sintaxis y semántica. La estructura de rasgos de la figura 1.24 formaliza el signo lingüístico que se utilizará durante el resto de este trabajo. El significante del signo se codifica como valor del rasgo *STEM*, y su significado como valor del rasgo *SYNSEM*. En esta gramática, la caracterización del significante es muy sencilla: es la lista (valor *list*) de la representación escrita de las palabras que constituyen el signo. La caracterización del significado es más rica. Consta de la información categorial (rasgo *HEAD*), la información relativa a los constituyentes que el signo subcategoriza (*VAL*, y dentro de este sus rasgos anidados *SUBJ* y *COMPS*⁴²), los rasgos de concordancia (rasgo *AGR*) y la información semántica (*SEM*).

⁴¹El axioma de una gramática de tipo HPSG puede ser, en principio, cualquier signo lingüístico —en el modelo chomskyano, el axioma es la oración—. Esto es especialmente interesante para gramáticas que quieran dar cuenta de la actuación, y quieran analizar, por tanto, oraciones truncadas y tópicos, palabras o morfemas aislados, etc.

⁴²Pollard y Sag (1994) usan una sola lista *SUBCAT* que aquí, en la línea de Sag y otros (2002), se ha desglosado en *SUBJ* y *COMPS*. Se entiende que *SUBJ* cuenta con el argumento menos oblicuo de *SUBCAT*, y *COMPS* con el resto. Los elementos de *COMPS* se listan en orden creciente de oblicuidad.

Obsérvese que esta información relativa al significante (HEAD, VAL, AGR y SEM) se presenta en la figura 1.24 como valor del rasgo LOCAL. Existe, además, en el signo, el rasgo NON-LOCAL. La información de LOCAL es la información entendida como local al signo (su categoría, significado, complementos adyacentes a él, etc.). La información codificada en NON-LOCAL será la información no local: la de aquellos constituyentes subcategorizados por el signo, pero realizados en la secuencia alejados de él (tópicos, pronombres interrogativos, relativos, etc.); o la de otros rasgos propios del signo materializados lejos de él (clíticos en el salto de clíticos, si se entienden estos como afijos).

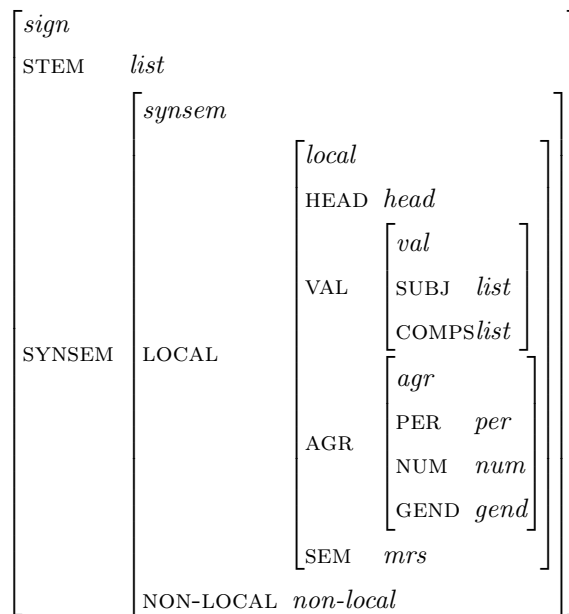


Figura 1.24: Signo lingüístico

En realidad la caracterización del signo utilizada en este trabajo es más compleja. En los capítulos sucesivos habrá ocasión de reformular la versión de la figura 1.24 para dar cuenta de esa complejidad. De momento, esta explicación es suficiente para entender cómo se conciben en HPSG los elementos informacionales que forman, y son, las secuencias de una lengua natural.

De acuerdo con la versión del signo de la figura 1.24, se pueden caracterizar palabras, sintagmas, oraciones, etc. Por ejemplo, el verbo del español

visitar, en tanto que unidad léxica, pudiera caracterizarse tal como se muestra en la figura 1.25.

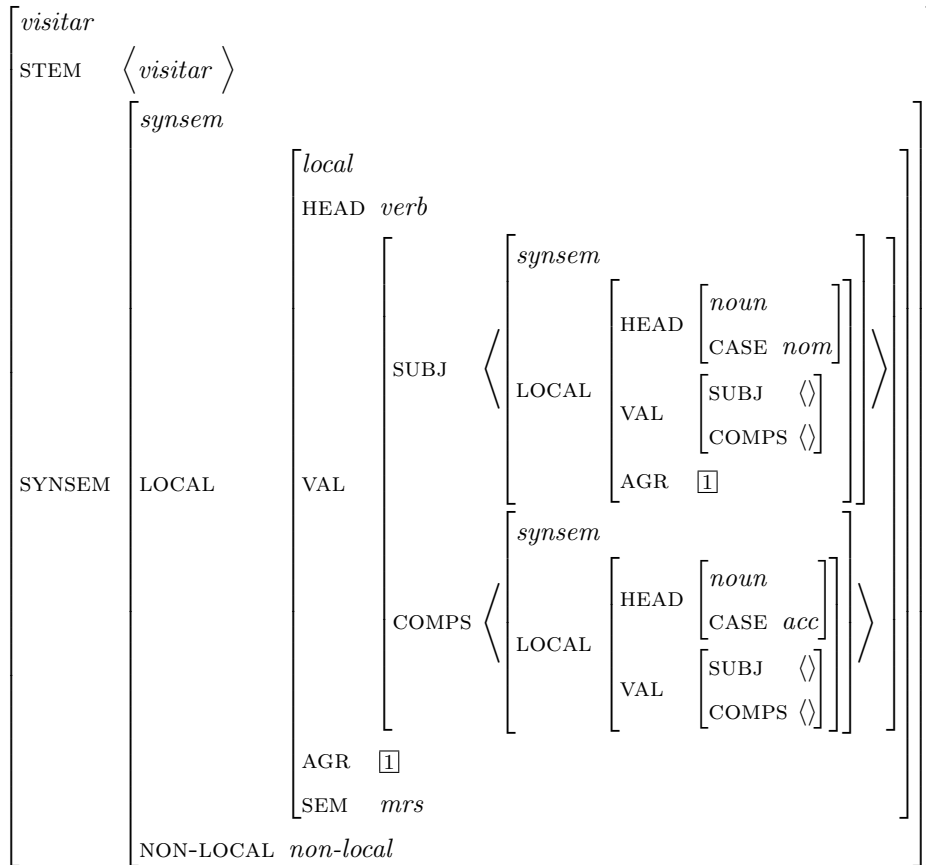


Figura 1.25: Signo lingüístico: *visitar*

Se ha caracterizado *visitar* como un verbo (HEAD **verb**) que subcategoriza un sujeto (SUBJ) y un complemento (COMPS). Estos deben ser un sintagma de categoría nominal (HEAD **noun**), con sus listas de subcategorización saturadas (SUBJ y COMPS <>). El caso ha de ser nominativo y acusativo (CASE **nom** y **acc**). Además, se establece concordancia de persona, género y número entre el sujeto y el verbo (estructura compartida de sus rasgos AGR)⁴³.

⁴³Pollard y Sag (1994), p. 36: “*subject-verb agreement arises directly from lexical specifications of the finite verb in conjunction with the structure sharing imposed by the Subcategorization Principle*”. Este *Subcategorization Principle* se explicará en breve. Es cierto que

Obsérvese que el valor de SUBJ y COMPS es una lista de estructuras de rasgos. La notación utilizada para representar las listas (los símbolos \langle y \rangle) es una simplificación. En realidad, una lista no es sino una estructura de rasgos más. La lista vacía $\langle \rangle$ es un elemento atómico *null*. Y la lista no vacía es un elemento no atómico con dos rasgos: al primero se le suele llamar FIRST y al segundo REST. El valor de FIRST será del tipo de los elementos listados. Si, como ocurre en SUBJ y COMPS, se están listando elementos de tipo *synsem*, FIRST tendrá por valor un elemento *synsem*. Por su parte, REST tiene por valor otra lista, que de nuevo tendrá su FIRST y su REST. Para denotar que la lista ha llegado a su fin, se establece como valor de REST una lista vacía *null*. Por tanto, las listas de cero, uno o dos elementos $\langle \rangle$, $\langle [] \rangle$ y $\langle [], [] \rangle$ son equivalentes a las siguientes estructuras de rasgos de la figura 1.26.

$$\left[\begin{array}{l} \textit{null} \\ \textit{FIRST} \textit{ synsem} \\ \textit{REST} \textit{ null} \end{array} \right] \left[\begin{array}{l} \textit{list} \\ \textit{FIRST} \textit{ synsem} \\ \textit{REST} \left[\begin{array}{l} \textit{list} \\ \textit{FIRST} \textit{ synsem} \\ \textit{REST} \textit{ null} \end{array} \right] \end{array} \right]$$

Figura 1.26: Estructuras de rasgos del tipo *list*

Cabe decir, por último, que los elementos subcategorizados por un signo lingüístico no son sino estructuras de rasgos que denotan subpartes de otros signos. Los elementos subcategorizados como sujeto o complementos son de tipo *synsem* —véase la figura 1.25—; es decir, son caracterizaciones de cómo ha de ser la naturaleza sintáctica y semántica de los signos que, en una secuencia, sean el sujeto o los complementos del núcleo. Los valores del rasgo *SYNSEM* de los signos lingüísticos que denoten a esos elementos habrán de

la concordancia entre sujeto y verbo debe establecerse solo para los verbos en forma finita. Se ha optado por incluirla aquí por resultar reveladora del modo en que muchos principios gramaticales se resuelven con naturalidad en HPSG mediante la caracterización léxica en forma de estructura de rasgos. Entiéndase, entonces, que la estructura aquí mostrada es caracterizadora de este tipo de signo.

unificar con estas caracterizaciones de su núcleo.

1.3.3.2. Reglas gramaticales

HPSG concibe entonces las secuencias de una lengua como signos lingüísticos. Estos se formalizan mediante estructuras de rasgos que aglutinan la información fonética, sintáctica, semántica, etc. de la secuencia. Una secuencia está formada por distintas subsecuencias, que son también signos lingüísticos, caracterizados, básicamente, mediante el mismo tipo de estructura de rasgos que caracteriza a la secuencia completa. La información que se le supone a la estructura de rasgos de una secuencia es el resultado de la unificación recursiva de la información de las estructuras de rasgos de sus subsecuencias. Por tanto, cada subsecuencia es portadora de una información parcial que, adecuadamente unificada con la información del resto de subsecuencias, da lugar a la información más específica de la secuencia completa.

Dada la lista de las palabras que forman una secuencia; es necesario que la gramática defina cuáles son las condiciones que permiten combinarlas para dar lugar a subsecuencias mayores, y, en última instancia, a la secuencia completa. Estas condiciones de compatibilidad se definen, fundamentalmente, en términos de subcategorización: una unidad léxica es compatible con un sintagma al que subcategoriza⁴⁴. Como se vio en el apartado 1.3.3.1, la estructura de rasgos que caracterizaba al verbo *visitar* establecía que este había de contar en la secuencia con un sujeto, y establecía ciertas propiedades (categoría, caso) que había de satisfacer este. Pollard y Sag (1994) formalizaron el modo en el que la gramática legitima la combinación de signos en virtud de la saturación argumental, mediante lo que llamaron *Subcategorization Principle*.

Este principio se pudiera concretar, entre otras posibilidades, en las siguientes reglas *comps-rule* y *subj-rule* —figuras 1.27 y 1.28—.

⁴⁴No obstante, una gramática de tipo HPSG también contempla otros esquemas de combinación de signos: la modificación, por ejemplo.

$$\left[\begin{array}{l} P \\ \text{SYNSEM|VAL|COMPS } \boxed{2} \end{array} \right] \longrightarrow \left[\begin{array}{l} p \\ \text{SYNSEM|VAL|COMPS } \langle \boxed{1} . \boxed{2} \rangle \end{array} \right], \left[\begin{array}{l} C \\ \text{SYNSEM } \boxed{1} \end{array} \right]$$

Figura 1.27: Regla de saturación de la lista COMPS: *comps-rule*

$$\left[\begin{array}{l} P \\ \text{SYNSEM|VAL|SUBJ } \boxed{2} \end{array} \right] \longrightarrow, \left[\begin{array}{l} C \\ \text{SYNSEM } \boxed{1} \end{array} \right], \left[\begin{array}{l} p \\ \text{SYNSEM|VAL|SUBJ } \langle \boxed{1} . \boxed{2} \rangle \end{array} \right]$$

Figura 1.28: Regla de saturación de la lista SUBJ: *subj-rule*

En *comps-rule* y *subj-rule* se establece que un sintagma P se forma mediante la unión de un núcleo p y un sintagma C . Para que esta unión sea posible es necesario que el valor de SYNSEM de C y el primer elemento de COMPS o SUBJ de p , unifiquen $\langle \boxed{1} \rangle$. Dicho de otro modo, *comps-rule* y *subj-rule* definen la proyección P de un núcleo p , dada la realización del argumento C de p . Para que C pueda satisfacer esa necesidad argumental de p , debe tener una estructura de rasgos unificable con lo que se ha establecido en p que debe ser el argumento en cuestión. Como se ve en las figuras 1.27 y 1.28, una vez C satisface la necesidad argumental de p , el argumento C se elimina en P $\langle \boxed{2} \rangle$. Por tanto, la aplicación de *comps-rule* o *subj-rule* sobre un núcleo va saturando sus argumentos de COMPS o SUBJ . Para que este proceso de saturación tenga lugar, es necesario que en la secuencia en la que aparece p aparezcan también los correspondientes elementos C , cuyas estructuras de rasgos han de unificar con las caracterizaciones que para ellas establece el núcleo p .

Obsérvese que la combinación de estas reglas modela lo que en el marco chomskyano se captaba mediante la Teoría de la X con Barra. A estas reglas se les da, entonces, en las obras clásicas de HPSG, una cierta vocación universalista —Pollard y Sag (1994) y Sag y otros (2002)—. Además, este sistema de proyección capta ya algo que después se asumiría en el Programa Minimista: los niveles de proyección no se restringen estipulativamente a dos. Una proyección será máxima o no según las reglas hayan saturado todas

sus necesidades o no. El nivel de proyección de una unidad es una cuestión gradual, entonces, relativa al número de elementos ya saturados, y a los pendientes de saturar.

No se ha dicho nada hasta ahora del orden lineal de constituyentes que establecen *comps-rule* y *subj-rule*. Este puede variar de unas lenguas a otras. En las figuras 1.27 y 1.28 se establece que, para *comps-rule*, *p* debe preceder inmediatamente en la secuencia a *C*; y, para *subj-rule*, *C* debe preceder inmediatamente a *p*. Con ello se está modelando el orden de palabras SVO (sujeto, verbo, objeto) propio de lenguas como el inglés o el español; el orden, en principio, interesante para este trabajo. Es frecuente, además, en las lenguas con este orden de palabras, que la aplicación de *comps-rule* deba preceder a la de *subj-rule*, de forma que núcleo y complementos crean un constituyente al que después se une el sujeto. Para lograr este efecto, es necesario estipular que el núcleo de *subj-rule* debe tener una lista *COMPS* vacía. Y deben proyectarse siempre, en *comps-rule*, el valor de *SUBJ* del núcleo; y en *subj-rule*, el valor de *COMPS* del núcleo. Por motivos de claridad, se obviará esta caracterización en la representación de las estructuras siguientes.

Si bien *comps-rule* y *subj-rule* recogen el *Subcategorization Principle* antes mentado, hay que añadirle cierta caracterización que formalice la idea intuitiva de que *P* es una estructura nucleada donde *p* es el núcleo. Pollard y Sag (1994) formalizaron esta realidad mediante lo que llamaron *Head Feature Principle*. Este principio establece que el rasgo *HEAD* de la proyección —*P*— comparte estructura con el rasgo *HEAD* del núcleo —*p*—. *HEAD*, como se ha visto, codifica la naturaleza categorial del signo. Mediante este principio, entonces, Pollard y Sag (1994) dan cuenta de la observación tradicional de que la categoría del núcleo la heredan todas sus proyecciones. Habrá que reformular, entonces, la regla de las figuras 1.27 y 1.28 para hacerla partícipe del *Head Feature Principle*. Esta reformulación se muestra en las figuras 1.29 y 1.30.

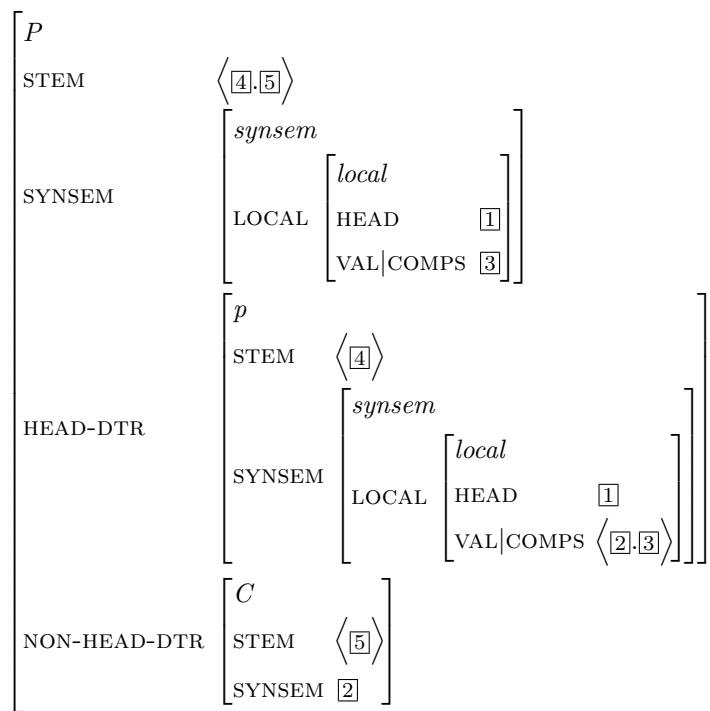


Figura 1.29: Regla de saturación de COMPS

En realidad, una regla no deja de ser un signo lingüístico más, caracterizable mediante la estructura de rasgos de la figura 1.24. Pudiera entenderse que los signos pueden ser de dos tipos fundamentales: léxicos o sintagmáticos. Una unidad léxica se formaliza mediante un signo léxico; y un sintagma (normalmente resultado de la unión de dos signos constituyentes), mediante un signo sintagmático, una regla. En las figuras 1.29 y 1.30, se vuelve a mostrar *comps-rule* y *subj-rule* en forma de estructura de rasgos, y en ellas se recoge ya el *Head Feature Principle*. Los rasgos STEM y SYNSEM caracterizan al signo lingüístico que es el resultado de la aplicación de la regla: la proyección P. Además, en HEAD-DTR se caracteriza al núcleo de la regla (p), y en NON-HEAD-DTR al complemento (C). El hecho de que esta regla participa del *Head Feature Principle* se capta haciendo que el rasgo HEAD de la proyección P y el del núcleo p compartan valor —1—. Las identidades de STEM reflejan el orden lineal requerido en cada caso —4 y 5—.

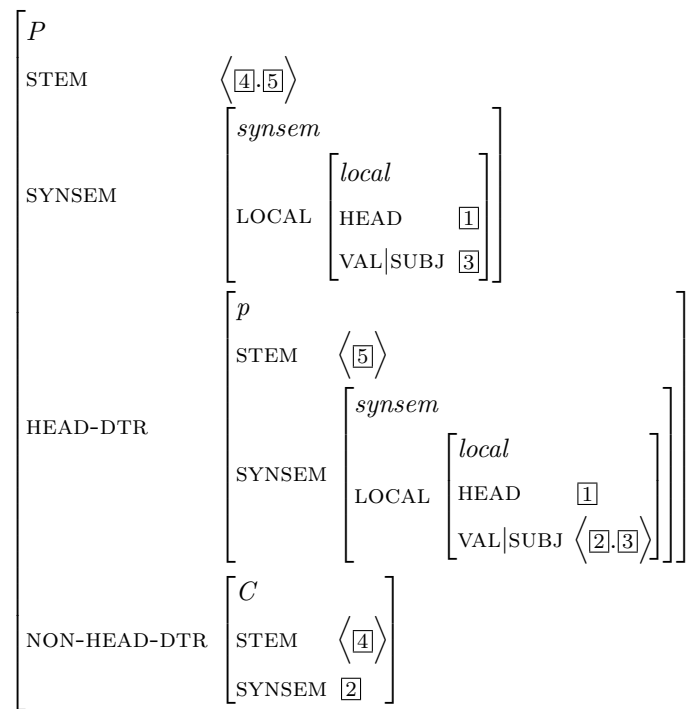


Figura 1.30: Regla de saturación de SUBJ

En resumen, una regla es un signo lingüístico más que define al sintagma resultante de la unión de un núcleo y un complemento —en su STEM y SYNSEM—. Núcleo y complemento se recogen en la estructura de rasgos como valores de HEAD-DTR y NON-HEAD-DTR —rasgos solo propios de los signos que son reglas—. La regla declara la posibilidad de proyectar un núcleo, dada la satisfacción de una de sus necesidades argumentales. Por tanto, si en la secuencia aparecen, en un determinado orden lineal, una unidad léxica que unifica con el valor de HEAD-DTR, otra que unifica con el de NON-HEAD-DTR, y la segunda unifica con los requerimientos que, para ella, establece la primera, se puede aplicar la regla en cuestión, y se crea así la proyección resultante. HPSG es un modelo gramatical radicalmente lexicista pues es en la caracterización léxica del núcleo donde se establece cómo debe ser el complemento. Esto permite que las reglas de la gramática sean pocas, y sean patrones tan genéricos como *comps-rule* y *subj-rule*. En ellos se unen dos elementos cualesquiera, confiando en que en la declaración léxica de cada núcleo ha sido codificada la caracterización de lo que deben ser sus complementos.

Llegada la discusión a este punto, se puede entender por qué no es trivial el que los elementos de SUBJ y COMPS sean *synsem*, y no signos completos *sign*. Gracias a esta decisión, se excluye la posibilidad de que un signo use, en la caracterización de sus argumentos, la naturaleza de los constituyentes de este. Nada impediría, si los elementos de SUBJ y COMPS fueran *sign*, que estos fueran caracterizados en cuanto a sus rasgos HEAD-DTR o NON-HEAD-DTR. Esta situación se parecería enormemente a una regla transformacional. Recuérdese que una regla transformacional toma como *input* una derivación, un árbol sintáctico. Una declaración de subcategorización que caracterice a los constituyentes del elemento subcategorizado hace otro tanto. Esto atenta contra un principio nuclear de las Gramáticas de Unificación, principio que hace de ellas teorías de menor coste computacional que la Gramática Transformacional: sus reglas deben ser de estructura sintagmática independientes del contexto. En versiones previas a Pollard y Sag (1994) —Pollard y Sag (1987)— los elementos de SUBCAT (lo que aquí es SUBJ y COMPS) eran de tipo *sign*. Esto obligaba a los autores a incluir un principio de localidad en

la gramática —*Locality Principle*— que excluía *ad hoc* esa posibilidad.

Además, en HPSG se contempla la existencia de otras reglas gramaticales: reglas que unen un adjunto al núcleo, reglas que unen un elemento desplazado (*filler*) a una estructura, etc. Sobre el primer tipo de regla no se volverá a hacer mención en el trabajo; pero las reglas que añaden un *filler* a la estructura sí se tratarán con detalle: apartado 5.3.

1.3.3.3. Semántica

Una vez visto el proceso de saturación de los argumentos de un núcleo en HPSG, se explica aquí el correlato semántico de ese proceso⁴⁵. Un signo lingüístico, una secuencia, es un par que une un significante y un significado. El proceso de análisis de una secuencia de una lengua natural debe consistir, entonces, en definitiva, en la traducción del significante en su significado, o viceversa. En este apartado se explicará ese proceso de traducción.

La información semántica de un signo se codifica en la estructura de rasgos como valor del rasgo SEM. Este rasgo codifica, en concreto, cuáles son las relaciones argumentales, y las relaciones de ámbito, que se establecen entre los constituyentes del signo. En este apartado se explicarán las relaciones argumentales. Estas se modelan, en una estructura de rasgos, mediante identidades entre los índices referenciales que identifican a cada individuo del discurso y los roles temáticos de los predicados. Cada signo lingüístico cuenta en SEM con un rasgo INDEX, cuyo valor puede ser *index* o *event*. El valor *index* se asigna al rasgo INDEX de los sintagmas que identifican individuos⁴⁶, mientras que los predicados tendrán por valor de INDEX *event*. Los predicados cuentan, además, con un rasgo por cada una de sus posiciones argumentales.

⁴⁵Será una explicación sencilla, ceñida a las necesidades propias de este trabajo. Para más información, véase Copestake y otros (1998).

⁴⁶Este tipo *index* se divide en Pollard y Sag (1994) en varios subtipos que distinguen los elementos nominales referenciales de los expletivos; los pronombres, las anáforas, los reflexivos y los recíprocos. Todo ello queda fuera del ámbito de esta explicación.

Se sigue en este trabajo la convención⁴⁷ consistente en utilizar atributos neutros para estos rasgos, atributos del tipo de ARG1, ARG2, ARG3, etc. Dado que no existe una teoría bien desarrollada sobre la nomenclatura de los posibles papeles temáticos, es preferible utilizar este tipo de atributos.

En cada unidad léxica predicativa debe establecerse cuál es la relación existente entre los argumentos subcategorizados por ella y sus roles temáticos. Si un verbo, *visitar* por ejemplo, subcategoriza en SUBJ un sujeto y en COMPS un complemento en acusativo, los índices de cada uno de ellos (INDEX index) deberán estar coindizados con los roles temáticos del predicado del verbo: ARG1 (adscrito para *visitar* a quien visita) y ARG2 (adscrito para *visitar* a lo que se visita). Cuando el proceso de saturación argumental identifique al signo que es el predicado, y al signo que es su complemento, el índice del complemento, con toda su información semántica asociada, se identificará con el rol temático correspondiente del predicado. Se establecerá así, para el verbo *visitar*, que ese individuo en cuestión es ‘quien visita’ o ‘lo que se visita’.

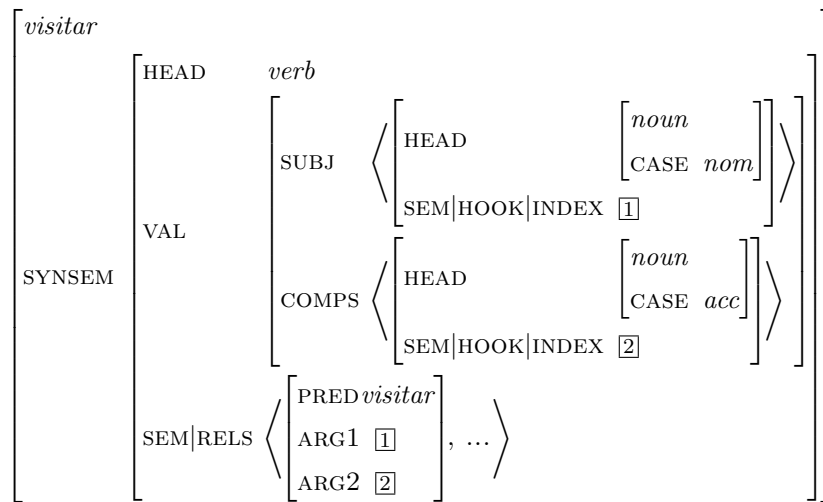


Figura 1.31: Correlato entre argumentos sintácticos y roles temáticos

Véase en la figura 1.31 una primera aproximación a la estructura de rasgos

⁴⁷Convención usada en la gramática Matrix. Los trabajos clásicos de HPSG utilizan para cada predicado atributos propios que caracterizan *ad hoc* sus papeles temáticos.

que establece estas relaciones. La siguiente estructura de rasgos de la figura 1.32 muestra de forma sencilla el modelado que hace HPSG de la naturaleza semántica de un signo de carácter predicativo.

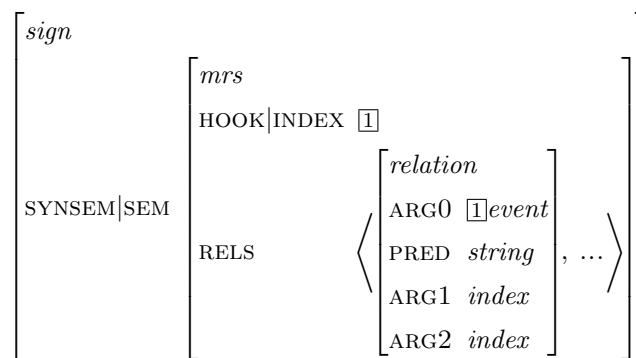


Figura 1.32: Signo lingüístico predicativo: rasgo SEM

En SEM se encuentran, fundamentalmente, dos rasgos: RELS y HOOK. HOOK es el lugar en el que aparece el índice semántico del signo⁴⁸ (INDEX *event*). RELS es una lista donde se van sumando los predicados en los que directa, o indirectamente, participa el índice de HOOK⁴⁹. Cada unidad léxica predicativa cuenta, normalmente, con un único elemento en la lista RELS: *relation*), con un rasgo PRED (identificador del predicado), un rasgo ARG0 que denota al índice de ese predicado (coindizado, como se puede imaginar, con el INDEX de HOOK) y un rasgo ARG1, ARG2, etc. por cada uno de los roles temáticos del predicado.

La caracterización semántica de un signo identificador de un individuo es similar. El índice del signo se coloca también en el INDEX de HOOK, pero es de tipo *index*. Su RELS incluye un elemento *relation* que representa al índice en cuestión en la lista. Además puede incluir otros elementos: predicados que predicán sobre dicho índice. Esto ocurre en cualquier sintagma nominal que incluya un nombre o un adjetivo. Se observa la estructura propia de este tipo de unidad léxica en la figura 1.33.

⁴⁸Junto con otras cuestiones, relativas al argumento externo del predicado, etc.

⁴⁹Junto con otras cuestiones relativas a las relaciones de ámbito, que exceden a las posibilidades de este trabajo.

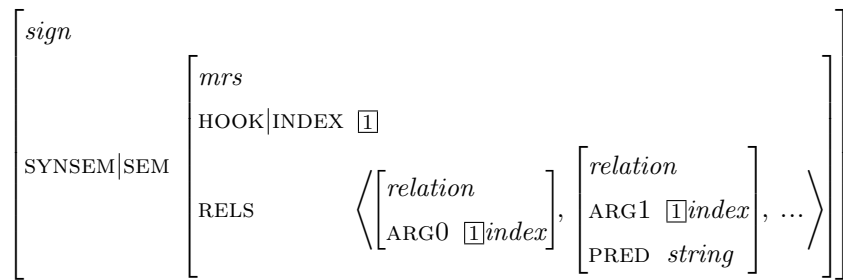


Figura 1.33: Signo lingüístico no predicativo: rasgo SEM

A las reglas sintácticas —apréciese de nuevo el lexicismo radical de HPSG— solo les queda sumar, en la proyección que crean, las listas RELS de los constituyentes que unen. Además, al menos en el tipo de reglas presentado aquí, identifican el índice semántico del núcleo con el de la proyección, mediante identidad de HOOK. Por tanto, a la caracterización de las reglas **comps-rule** y **subj-rule** hay que añadirle la caracterización de la figura 1.34.

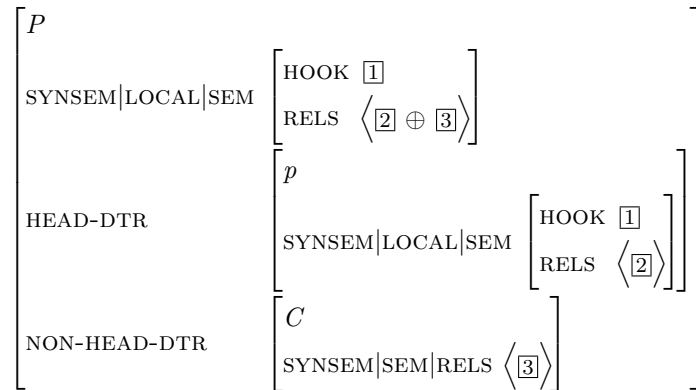


Figura 1.34: Reglas **comps-rule** y **subj-rule**. Información semántica

Así pues, con esta nueva versión de **comps-rule** y **subj-rule**, y dado que en las unidades léxicas se ha establecido la debida relación de índices entre los argumentos sintácticos y los roles temáticos —figura 1.31—, la gramática queda equipada para traducir significante a significado (entendido este, al menos, como juego de relaciones argumentales). Concretamente, al crear un sintagma, **comps-rule** y **subj-rule** suman en este las listas RELS de sus constituyentes. Se presupone, además, que, en la declaración léxica del núcleo,

se han establecido las debidas relaciones entre los índices de los argumentos sintácticos, y los roles temáticos —figura 1.31—. Con todo ello, la lista RELS del sintagma creado será una caracterización adecuada de las relaciones argumentales establecidas entre núcleo y complemento: constará, al menos, de un elemento *relation* que represente al predicado del núcleo, y otro que represente al individuo que es su complemento; y el índice del individuo y la posición argumental del predicado compartirán valor.

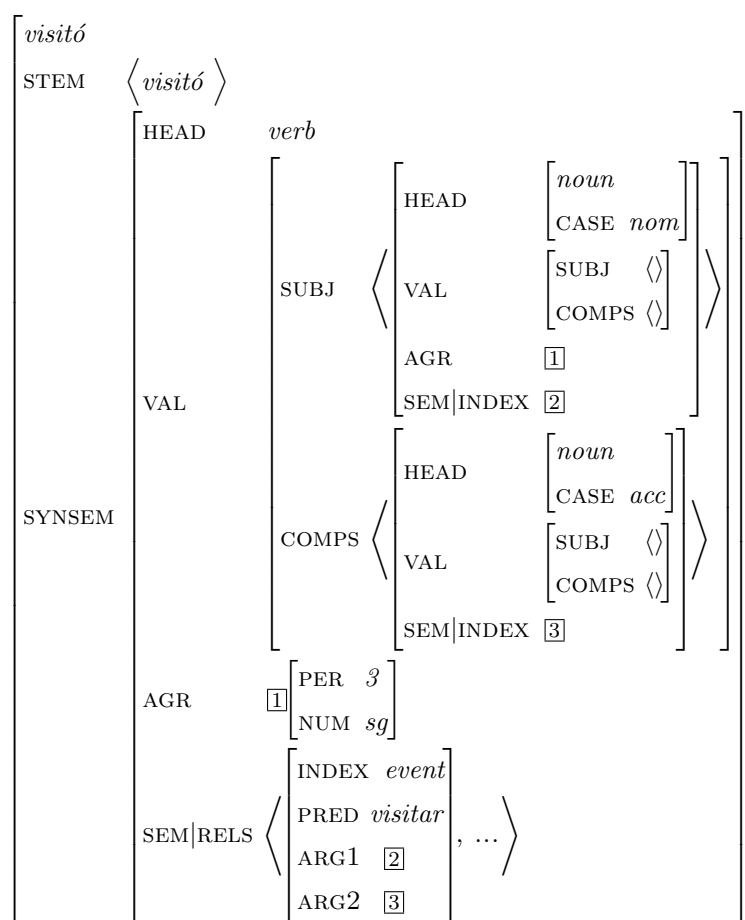


Figura 1.35: Signo lingüístico: *visitó*

Para ilustrar este proceso, se explica a continuación cómo una gramática de tipo HPSG opera para analizar una secuencia como *María visitó Madrid*. Obsérvese cómo el proceso recursivo de saturación sintáctica —saturación de SUBJ y COMPS— elabora, recursivamente también, las debidas relaciones

argumentales semánticas —codificadas en la lista RELS—. La estructura de *visitó* tiene la misma caracterización que se mostró en la figura 1.25 para la unidad léxica *visitar*, salvo por el añadido los rasgos de persona y número. Véase en la figura 1.35. Además, obsérvese que cada uno de los argumentos de SUBJ y COMPS tiene relacionado su índice con un rol temático del verbo (ARG1 y ARG2). Con esta relación, se formaliza la idea de que el individuo denotado por el sujeto de *visitar* será ‘quien visite’; mientras que, el individuo denotado por el complemento será ‘lo que se visite’.

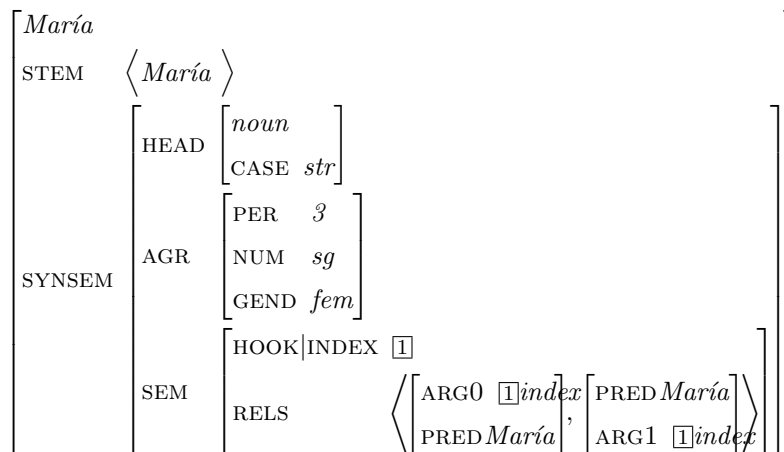


Figura 1.36: Signo lingüístico: *María*

Las estructuras de rasgos correspondientes a *María* y *Madrid* se muestran, respectivamente, en las figuras 1.36 y 1.37. Téngase en cuenta que el valor de CASE de ambos es **str**, es decir, estructural. Los sintagmas no tienen especificado si van a estar marcados por caso nominativo o acusativo, pero sí restringen sus posibilidades de marca a estos dos casos. Al unificar con la posición de sujeto o de complemento de *visitar*, el caso **str** pasará a ser **nom** o **acc**; casos compatibles con **str**, pero más específicos. Obsérvese la identidad de índices establecida, en ambos casos, entre el rasgo ARG0 de RELS y el rasgo INDEX de HOOK. Recuérdese que HOOK es el lugar donde se publica la información de RELS del signo que puede ser utilizada por otro signo. Precisamente, en la caracterización del verbo *visitar*, la coindización entre los

argumentos sintácticos y los roles temáticos se hace a través de INDEX, no de ARG0, de acceso más difícil. Por tanto, esta identidad es importante.

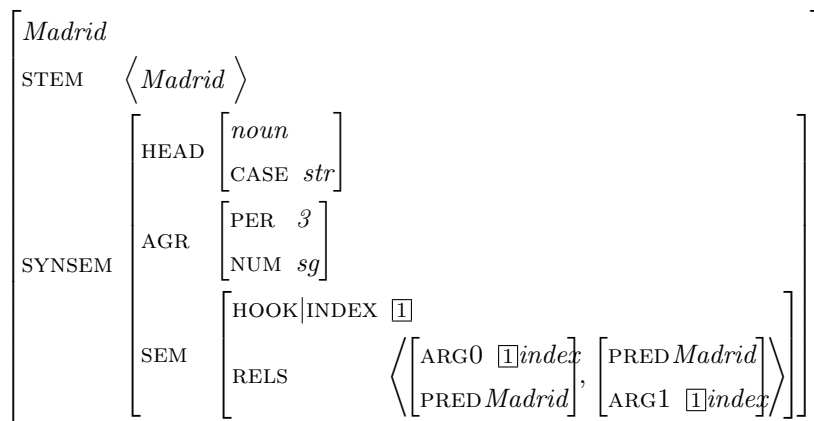


Figura 1.37: Signo lingüístico: *Madrid*

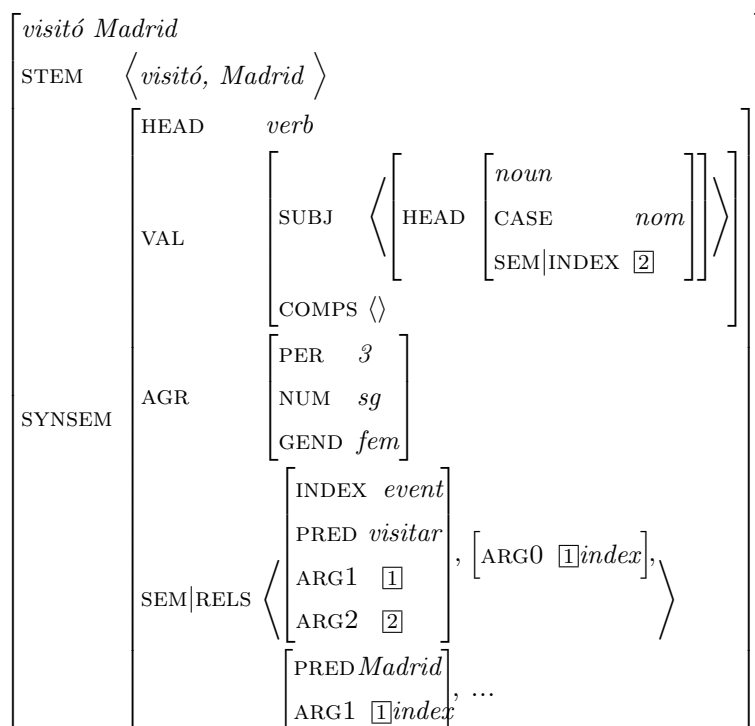


Figura 1.38: Signo lingüístico: *visitó Madrid*

Para crear una primera proyección del verbo, *visitó* y *Madrid* podrían unificar, respectivamente, con las posiciones HEAD-DTR y NON-HEAD-DTR de *comps-rule*. El sintagma *Madrid* unifica con los requerimientos que establece *visitó* a su primer argumento, con lo cual se puede crear una primera proyección de *visitó* en la que tal argumento queda saturado. Véase en la figura 1.38. La eliminación del complemento en COMPS supone, en cambio, el añadido a RELS del elemento *relation* de *Madrid*, con la debida relación de su índice con el ARG1 del predicado *visitar*.

Esta proyección puede volver a unificar con el valor del rasgo HEAD-DTR de *subj-rule*. El valor de NON-HEAD-DTR podría unificar también con la estructura correspondiente a *María*. Como esta segunda estructura unifica con los requerimientos que el verbo establece para su argumento, es posible crear una proyección máxima de este, correspondiente a la oración *María visitó Madrid*. Se muestra esta en la figura 1.39.

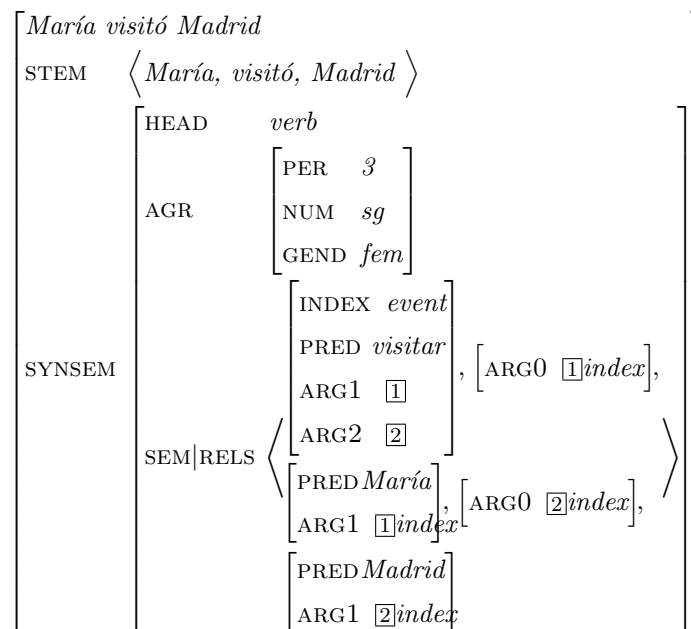


Figura 1.39: Signo lingüístico: *María visitó Madrid*

De nuevo, la eliminación del sujeto en SUBJ supone el añadido a RELS del elemento *relation* de *María*, con la adecuada relación de su índice con el ARG1 del predicado *visitar*.

Asúmase que el axioma de la gramática, como es tradicional, es un signo de categoría verbal con sus necesidades argumentales resueltas. En tal caso, la derivación que se acaba de mostrar legitimaría a *María visitó Madrid* como secuencia propia del lenguaje que la gramática define. Además, el proceso de análisis ha creado una proyección, la mostrada en la figura 1.39, cuya caracterización semántica refleja las relaciones argumentales que denota la secuencia en cuestión. Es decir, el proceso sintáctico de análisis de la secuencia ha construido el significado de esta.

En el apartado 1.3.3.2, solo se presentó uno de los tipos de reglas sintácticas que contempla HPSG: el que legitima la creación de un sintagma dada la satisfacción de una necesidad argumental de un núcleo. En este apartado solo se ha contemplado la relación semántica correspondiente a esa relación gramatical de núcleo y complemento. Quedan sin explicar otros patrones de relación semántica, igual que quedaron por explicar otros patrones de relación sintáctica. La relación existente entre un núcleo y un adjunto de tipo intersecutivo (sin ámbito sobre su núcleo) pedirá en una gramática un tipo diferente de regla y un patrón semántico propio. Otro tanto ocurre para la relación existente entre un núcleo y un modificador con ámbito sobre él, la que se establece entre un determinante y un sintagma nominal, o entre un *filler* y una oración. La modificación y el ligado de variables por parte de los determinantes quedan fuera de los intereses de este trabajo. Por tanto, las reglas y patrones semánticos correspondientes a estas cuestiones no se tratarán en él. Como ya se dijo en 1.3.3.2, sí se trabajará en el apartado 5.3 la relación sintáctica, y semántica también, entre una oración y un *filler*.

1.3.3.4. Lexicismo y reglas léxicas

HPSG es un modelo gramatical radicalmente lexicista. El lexicismo es pro-

pio también de las últimas versiones de la gramática chomskyana; es, quizá, una tendencia más o menos general de la Lingüística actual⁵⁰. Pero HPSG la asume de un modo radical. En este modelo, son las unidades léxicas las que proporcionan el grueso de la información sobre los elementos que subcategorizan. Son, entonces, las que establecen sus propias condiciones de compatibilidad. Asumido que las unidades léxicas caracterizan minuciosamente cómo ha de ser la estructura de rasgos de sus elementos subcategorizados, la gramática solo necesita unas pocas reglas muy generales.

En los apartados previos 1.3.3.2 y 1.3.3.3 ha quedado claro cómo HPSG vuelca el peso de la declaración de las condiciones de compatibilidad de las subsecuencias en el léxico. Las reglas **comps-rule** y **subj-rule** establecen de forma genérica que el complemento debe ser un sintagma ubicado en la secuencia de un determinado modo con respecto al núcleo, y que ha de satisfacer la caracterización que para él establece este. Pero esa caracterización se ha codificado en la declaración léxica del núcleo, no en las reglas. En lo que a la semántica se refiere, la declaración léxica del núcleo establece las debidas relaciones argumentales. A las reglas **comps-rule** y **subj-rule** solo les queda proyectar el índice del núcleo y sumar los elementos **relation** de los constituyentes.

Por tanto, una gramática de tipo HPSG es una colección de unidades léxicas construidas en forma de estructuras de rasgos muy ricamente especificadas. A partir de tales unidades, un sistema de reglas sintácticas sencillísimo (idealmente, una regla de especificador **subj-rule** y otra de complemento **comps-rule**) generan las secuencias posibles de la lengua. Dada esta situación, en HPSG ha sido necesaria la elaboración de una rica teoría sobre la elaboración y organización del léxico. El léxico se convierte en un repositorio enorme de información. Habrá que velar por que este sea lo más sencillo

⁵⁰Pollard y Sag (1994), p. 37: “*In the past decade, there has been a continuation of the general trend in syntactic theory toward the ‘lexicalization of grammar’, and concomitantly a tendency to collapse or eliminate construction-specific rules in favour of highly schematic immediate dominance templates*”.

posible, evite redundancias, y recoja las debidas generalizaciones sobre el funcionamiento de cada palabra. El siguiente apartado 1.3.3.5 ayudará a entender cómo se elabora un léxico de estas características. Se verá que no se codifica cada unidad léxica de forma independiente al resto. En cambio, se intentan implementar de forma independiente generalizaciones de las que participen distintas unidades léxicas. Estas unidades léxicas se codificarán después como instancias de esas generalizaciones. De este modo, se evita la redundancia. Por ejemplo, no es necesario ya implementar, repetidamente, cada verbo transitivo como tal, de forma independiente al resto, sino que se creará un signo lingüístico transitivo del que se hará partícipe a continuación a cada unidad léxica que obedezca a ese patrón.

La gramática chomskyana da cuenta del hecho de que, para una misma unidad léxica, existen distintas posibilidades de realización. El ejemplo clásico es la posibilidad de que un verbo transitivo —en español o en inglés, entre otras lenguas— pueda aparecer en voz activa o pasiva. Como se explicó en 1.2, la gramática chomskyana utiliza las transformaciones para dar a este fenómeno una explicación adecuada, al menos, desde el punto de vista descriptivo. No es descriptivamente adecuada una gramática que implemente dos entradas léxicas distintas para el mismo verbo transitivo: una activa y otra pasiva. Se perdería así la generalización consistente en que se conciben activa y pasiva como dos patrones de realización de la misma palabra. De hecho, ambos patrones dan lugar a una misma interpretación semántica, al menos a un mismo esquema de relaciones argumentales. Una gramática que no recoja estas relaciones entre las distintas secuencias de las lenguas que representa no es, entonces, descriptivamente adecuada. Por tanto, ¿cómo da HPSG cuenta de estas generalizaciones?

Utiliza para ello un nuevo tipo de signo lingüístico: las reglas léxicas. En principio, la codificación propiamente dicha de una unidad léxica recoge, exclusivamente, su realización canónica. El resto de posibilidades de realización se logran mediante reglas que modifican apropiadamente la unidad léxica original. Así pues, un verbo transitivo se modela, en principio, en su

versión activa. La versión pasiva será el resultado de la aplicación sobre esa unidad léxica de una regla que aplique al verbo morfología pasiva, elimine su argumento externo, y asigne caso nominativo al argumento interno. Si bien desde el punto de vista de coste computacional las reglas léxicas de HPSG y las transformaciones no son en absoluto equiparables —una regla léxica no opera sobre una derivación, sino sobre el resultado de esta—, desde un punto de vista funcional, quizá sí lo sean. Ambas consiguen relacionar adecuadamente las secuencias de las lenguas que deben estar relacionadas: dos patrones distintos de realización de la misma palabra están relacionados en tanto que corresponden a derivaciones distintas a partir de la misma unidad léxica. Dentro de este marco, las reglas léxicas no son sino un sistema generativo que crea nuevas unidades léxicas (nuevos posibles *inputs* para el sistema sintáctico) a partir de unidades léxicas preexistentes⁵¹.

Solo queda mostrar cómo se formaliza una regla léxica mediante una estructura de rasgos. Como cabe imaginar, una regla léxica se concibe, de nuevo, como el signo lingüístico caracterizador de su *output*. Por tanto, no será sino un tipo de signo, con su rasgo STEM y su rasgo SYNSEM. A este signo se le añade un rasgo DTR cuyo valor es el signo que representa al *input* de la regla. En la figura 1.40 se muestra la estructura de rasgos que formaliza estas reglas.

$$\left[\begin{array}{ll} \textit{sign} & \\ \text{STEM} & \textit{list} \\ \text{SYNSEM} & \textit{synsem} \\ \text{DTR} & \textit{sign} \end{array} \right]$$

Figura 1.40: Regla léxica

Mediante las adecuadas estructuras compartidas, se puede lograr que el signo que es el *output* comparta, hasta cierto punto, la caracterización del

⁵¹Se habla de unidades léxicas diferentes en tanto que son *inputs* léxicos distintos para el sistema sintáctico. La relación evidente entre realizaciones distintas de una misma unidad la preserva la mera derivación.

signo que es el *input*. Ahora, en tanto en cuanto *input* y *output* no compartan estructura, será posible modificar la caracterización del primero.

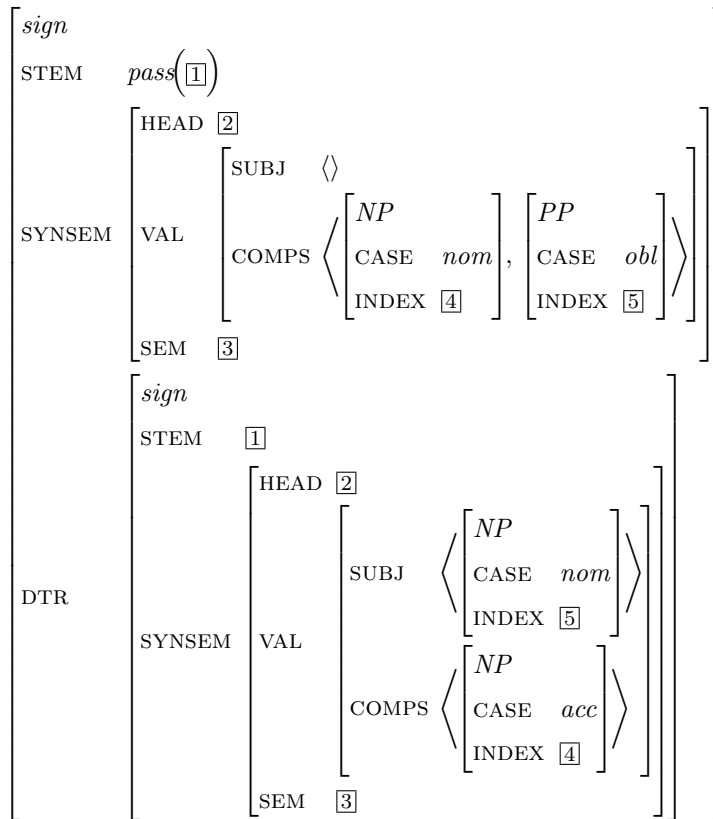


Figura 1.41: Regla léxica de pasivización

Por ejemplo, una hipotética regla de pasivización de un verbo transitivo activo debería identificar, al menos, la categoría verbal y la naturaleza semántica del *input* con las del *output*; pero no debería identificar sus condiciones de subcategorización, pues estas cambian de activa a pasiva. Valga la representación tentativa de esta regla de pasivización de la figura 1.41, donde *pass* es una función que devuelve la versión pasiva del significante de un verbo, y donde los argumentos se representan de forma simplificada.

Obsérvese cómo las relaciones argumentales del *input* se preservan en el *output*. Por un lado, el valor de SEM del primero se identifica con el del

segundo, con lo que se preserva la naturaleza predicativa de la unidad léxica: el predicado que denote, con sus roles temáticos. Además, los índices de los argumentos sintácticos —que el verbo identificará con los roles temáticos pertinentes— se identifican entre *input* y *output*. De este modo, el rol temático del sujeto activo se identifica con el rol temático del complemento agente pasivo; y el rol del objeto activo se identifica con el del sujeto pasivo. Esta regla, asegura, entonces, que las realizaciones activa y pasiva de un verbo darán lugar a un mismo juego de relaciones argumentales.

Es frecuente, por último, que una unidad léxica admita distintos patrones de realización, excluyentes entre sí, pero entre los cuales no es posible identificar uno canónico y otros derivados. Para modelar este tipo de fenómenos se utilizan reglas léxicas en las que todo el *input* comparte estructura con el *output*: ambos se identifican completamente. Se confía entonces en que ambos sean absolutamente compatibles; si bien se entiende que en el *output* se especificará en algún sentido la estructura del *input*.

$$\left[\begin{array}{l} \textit{sign} \\ \text{STEM} \quad \textit{infl}(\boxed{1}) \\ \text{SYNSEM} \quad \boxed{2} \\ \text{DTR} \quad \left[\begin{array}{l} \textit{sign} \\ \text{STEM} \quad \boxed{1} \\ \text{SYNSEM} \quad \boxed{2} \end{array} \right] \end{array} \right]$$

Figura 1.42: Regla léxica flexiva

Las gramáticas incluyen una subclase de regla léxica de este tipo, por ejemplo, para modelar el fenómeno de la flexión de ciertas unidades: *infl-rule*. Un verbo, por ejemplo, pudiera aparecer en tercera persona del singular, o en tercera persona del plural. Parece conveniente que la unidad léxica de tal verbo no tenga especificación alguna de persona y número; y que sean distintas reglas de tipo *infl-rule* las que le asignen una u otra caracterización. En la figura 1.42 se entiende que $\textit{infl}(\boxed{1})$ es una especificación de $\boxed{1}$, consistente en la concreción del significante de la palabra, de acuerdo con el valor de los

rasgos flexivos que asigne la regla.

Por ejemplo, el verbo *visitar*, como *input* de una hipotética regla *3-sg-infl-rule*, daría lugar a la palabra flexionada *visitó* —obviando la naturaleza tempoaspectual del verbo, que merece una formalización similar—. En cambio, como *input* de *3-pl-infl-rule*, daría lugar a la palabra flexionada *visitaron*. Véanse estas reglas en la figura 1.43, donde *3sg(visitar)* equivale a *visitó*, y donde *3pl(visitar)* equivale a *visitaron*.

$$\left[\begin{array}{l} \textit{3-sg-infl-rule} \\ \text{STEM} \quad \langle \textit{3SG}(\underline{1}) \rangle \\ \text{SYNSEM} \quad \underline{2} \left[\text{SYNSEM|AGR} \begin{array}{l} \text{PER} \quad \textit{3} \\ \text{NUM} \quad \textit{sg} \end{array} \right] \\ \text{DTR} \quad \begin{array}{l} \textit{sign} \\ \text{STEM} \quad \langle \underline{1} \textit{visitar} \rangle \\ \text{SYNSEM} \quad \underline{2} \end{array} \end{array} \right] \quad \left[\begin{array}{l} \textit{3-pl-infl-rule} \\ \text{STEM} \quad \langle \textit{3PL}(\underline{1}) \rangle \\ \text{SYNSEM} \quad \underline{2} \left[\text{SYNSEM|AGR} \begin{array}{l} \text{PER} \quad \textit{3} \\ \text{NUM} \quad \textit{pl} \end{array} \right] \\ \text{DTR} \quad \begin{array}{l} \textit{sign} \\ \text{STEM} \quad \langle \underline{1} \textit{visitar} \rangle \\ \text{SYNSEM} \quad \underline{2} \end{array} \end{array} \right]$$

Figura 1.43: Flexión del verbo *visitar*

1.3.3.5. Noción de tipo y noción de jerarquía

En las secciones previas se han mostrado distintas estructuras de rasgos, caracterizadoras de distintos tipos de signo lingüístico: unidades y reglas léxicas, y reglas gramaticales. De un modo no explícito, y no formal, se ha jugado con la idea intuitiva de que cada una de esas estructuras de rasgos modelaban un objeto gramatical de un tipo concreto, distinto de otros objetos de otro tipo: una unidad léxica, por ejemplo, es un objeto distinto de una regla gramatical. No es difícil imaginar que, dentro de una gramática, existirán distintos tipos de unidad léxica (verbos, nombres, adjetivos, etc.) y distintos tipos de reglas gramaticales o léxicas. Cada uno de esos tipos se formalizará mediante una estructura de rasgos característica.

HPSG hace un uso general y formalizado de la noción de tipo. Esta formalización es deudora de los trabajos de Kasper y Rounds (1986) y Carpenter (1992). En HPSG toda gramática G consiste en la siguiente cuádrupla: $\langle T, F, I, *top* \rangle$.

- T es un conjunto de tipos t . Es decir, $T = t_1, \dots, t_n$. Cada t_i es una estructura de rasgos con 0 o más rasgos de F declarados para sí.
- F es un conjunto de rasgos f . Esto es, $F = f_1, \dots, f_m$. Cada f_i es un par atributo valor $\langle f_i t_j \rangle$, tal que $t_j \in T$. Además, cada rasgo ha debido ser declarado como rasgo de un tipo de T , y solo de uno.
- I es una función que establece, para todo par de tipos $t_i \in T$ y $t_j \in T$, un valor 0 o 1 . Dicho de forma intuitiva, I denota una relación de herencia entre t_i y t_j . De este modo, si $I(t_i, t_j) = 1$, entonces t_i es subtipo de t_j —o lo que es lo mismo, t_j es supertipo de t_i —. Esta relación es relevante, porque se entiende que, si $I(t_i, t_j) = 1$, entonces $t_j \sqsubseteq t_i$. Es decir, cuando un tipo (t_i) se declara como subtipo de otro (t_j), entonces, el subtipo (t_i) hereda la estructura de rasgos de su supertipo (t_j): si para t_j se declaró el rasgo f_k , entonces, a la estructura de su subtipo t_i se le supone también el rasgo f_k .
- $*top*$ es el único miembro de T que no se ha declarado en G como subtipo de ningún otro tipo (para cualquier $t_j \in T$, $I(*top*, t_j) = 0$). Además, todo tipo $t_i \in T$ distinto de $*top*$ ha sido declarado en G , directa o indirectamente, como subtipo de $*top*$ —o bien $I(t_i, *top*) = 1$, o bien existe un subconjunto $R \subset T$, donde $R = r_1, \dots, r_n$; en tal caso, $r_1 = *top*$, además $r_n = t_i$ y, para todo r_i , $I(r_{i+1}, r_i) = 1$ —. En definitiva, de este modo, se establece que la gramática G es una única jerarquía de tipos, donde cada tipo es subtipo, al menos, de otro tipo. Esta jerarquía tiene una estructura arbórea: a partir del axioma $*top*$, el resto de tipos se declaran como subtipos de este, o como subtipos de otros tipos que, en última instancia, sí son subtipos de $*top*$.

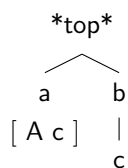


Figura 1.44: Gramática G

Es decir, una gramática de tipo HPSG es un conjunto de tipos que modelan distintos objetos gramaticales. Cada tipo es una estructura de rasgos. Cada rasgo se declara para un solo tipo; y su valor es, necesariamente, uno de los tipos de la gramática. Véase, por ejemplo, la gramática G en la figura 1.44. G consta de cuatro tipos: ***top***, **a**, **b** y **c**. Los tipos ***top***, **b** y **c** son atómicos (con estructuras de rasgos vacías []); mientras que **a** es una estructura de rasgos con el rasgo A. El valor de A es un tipo de la gramática: **c**. No hubiese sido posible de otro modo: todo rasgo debe tener por valor un tipo declarado como tal. De igual modo, no se hubiese podido declarar a A como rasgo de **a** y de **b**: cada rasgo se declara en un solo tipo de la gramática.

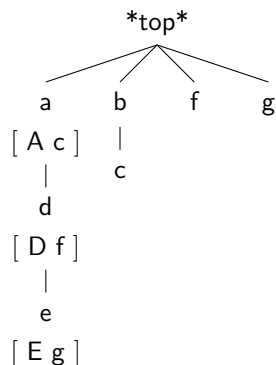


Figura 1.45: Gramática G'

Por lo demás, todos los tipos de G forman una jerarquía, están relacionados por herencia, formando una única estructura arbórea. El árbol de la figura 1.44 da a entender que **a** y **b** son subtipos de ***top***, mientras que **c** es subtipo de **b**. No hubiese sido posible que cualquier otro tipo distinto del axioma ***top*** no hubiese sido declarado como subtipo de algún tipo. En tanto que axioma, ***top***, por contra, no hubiese podido ser declarado como subtipo de otro tipo. Toda gramática debe obedecer, en definitiva, a un esquema arbóreo como el

de la figura 1.44. Si un tipo d es subtipo de otro tipo a , d hereda la estructura de rasgos de a . Además, la herencia es una relación transitiva: si un tipo e es subtipo de d , y d a su vez es subtipo de a ; entonces e heredará la estructura de d , e , indirectamente, también la de a . Valga como ejemplo la gramática G' , versión ampliada de la gramática anterior G , y representada en la figura 1.45.

En ella aparece solo la estructura de rasgos directamente declarada en cada tipo: A se ha declarado como rasgo de a , D se ha declarado como rasgo de d , y E se ha declarado como rasgo de e . Dada la relación de herencia, la estructura de rasgos de d debe incluir, no solo la caracterización propiamente declarada en d —el rasgo D —, sino también toda la caracterización heredada de sus supertipos. Por tanto, de acuerdo con G' , su estructura de rasgos, expandida, es la que se muestra en la figura 1.46.

$$\begin{bmatrix} d \\ A \ c \\ D \ f \end{bmatrix}$$

Figura 1.46: Estructura de rasgos expandida de d

Además, la relación de herencia, como se ha dicho, es transitiva. Por tanto, la estructura expandida del tipo e debe incluir la caracterización declarada en sí — E —, la declarada en sus supertipos inmediatos — D — y la declarada en los supertipos de sus supertipos hasta el axioma ***top*** — A —. En definitiva, en G' , la estructura de rasgos expandida de e es la que se observa en la figura 1.47.

$$\begin{bmatrix} d \\ A \ c \\ D \ f \\ E \ g \end{bmatrix}$$

Figura 1.47: Estructura de rasgos expandida de e

Suele entenderse, por último, que existen tres tipos de herencia: herencia simple, herencia múltiple y herencia por defecto. La herencia simple es la mostrada hasta el momento: es la situación en la cual un tipo *a* es subtipo de un único supertipo *b*. Pero es posible que un mismo tipo *a* herede de más de un supertipo (*b*, *c*, etc.). Tal situación recibe el nombre de herencia múltiple. En este caso, la estructura de rasgos del subtipo en cuestión debe incluir la caracterización de todos sus supertipos. Por último, se contempla la existencia de la herencia por defecto. Esta es la forma habitual de representar las excepciones. Se puede establecer en un tipo *a* que el valor para su rasgo *X* es, por defecto, *x* —se representa *X* /*x*—. En tal caso, todo subtipo de *a*, por defecto, tendrá por valor de *X*, *x*. Pero es posible declarar un subtipo *b* de *a* que sobrescriba ese valor por defecto; de tal forma que, para *b*, el valor de *X* sea *y*. En tal caso, *b* es una excepción de la generalización que denota *a*.

Se ha explicado, entonces, cómo una gramática de tipo HPSG es una jerarquía de tipos relacionados mediante herencia. Llegada la explicación a este punto, habrá que ordenar todas las realidades gramaticales vistas hasta el momento en tal jerarquía. Recuértese que se ha visto una caracterización concreta del signo lingüístico. Por tanto, el signo como tal será un tipo de esa jerarquía: *sign*. También se han vistos distintas clases de signo lingüístico: unidades léxicas, y reglas léxicas y gramaticales. Estas realidades deben formalizarse, por tanto, como tipos de esa misma jerarquía, concretamente, como subtipos de *sign*. En la siguiente figura 1.48 se muestra una primera aproximación de lo que debe ser la jerarquía en cuestión. Se sigue la práctica habitual consistente en distinguir dos grandes clases de tipos: los tipos atómicos —subtipos de *atom*— y los complejos —subtipos de *avm*⁵²—. Como se ve en la figura 1.48, el signo se clasifica como tipo complejo.

⁵²Léase *attribute value matrix*, o matriz de pares atributo valor.

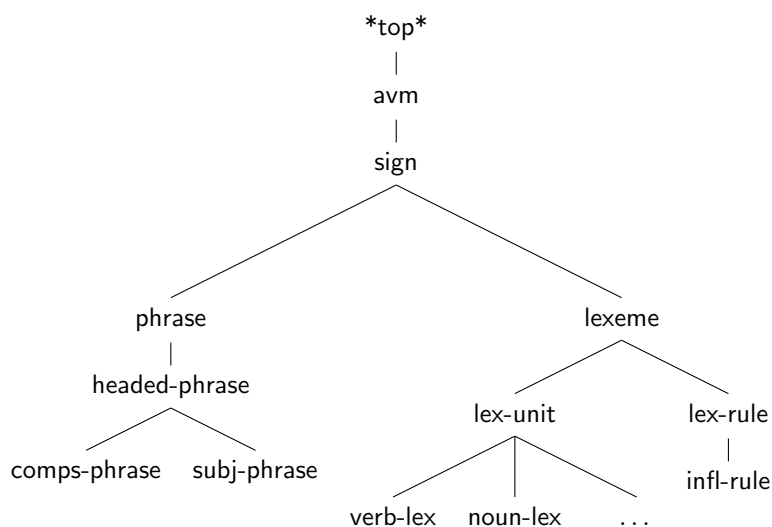


Figura 1.48: Gramática SSG. Jerarquía de tipos (I)

En la figura 1.48 se observa que un signo lingüístico puede ser o un sintagma (*phrase*) o un lexema (*lexeme*). Esta jerarquía, de momento, solo contempla la existencia de sintagmas nucleados (*headed-phrase*); concretamente, el resultante de la saturación de un complemento en el núcleo (*comps-phrase*), y el resultante de la saturación de un sujeto (*subj-phrase*). Los lexemas pueden ser unidades léxicas propiamente dichas (*lex-unit*): verbos (*verb-lex*), nombres (*noun-lex*), etc. Y también pueden ser reglas léxicas (*lex-rule*), útiles, entre otras cuestiones, para flexionar unidades léxicas (*infl-rule*). Por razones de claridad, no se incluyen en esta figura las estructuras de rasgos declaradas en cada tipo. Recuérdese que a *sign* se le suponen los rasgos *STEM* y *SYNSEM*, que formalizan, respectivamente, el significante y el significado del signo. En *headed-phrase* se añaden los rasgos *HEAD-DTR* y *NON-HEAD-DTR*, que representan a los constituyentes del sintagma denotado. A *lex-rule* se le supone un rasgo *DTR lexeme*, para modelar el *input* de la regla. Se han incluido, además, otras caracterizaciones en las estructuras de estos tipos, mediante estructuras compartidas (el *Headed Feature Principle* en *headed-phrase*, el *Subcategorization Principle* en *comps-phrase* y *subj-phrase*, etc.), o mediante especificación del valor de sus rasgos (la categoría de las unidades léxicas, sus condiciones de subcategorización, etc.). Todo ello ya ha sido explicado en los apartados

previos; solo queda observar cómo todo confluye en la jerarquía de la figura 1.48.

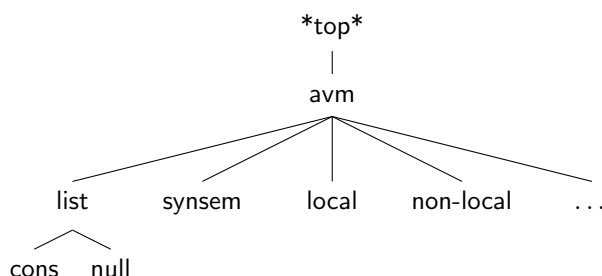


Figura 1.49: Gramática SSG. Jerarquía de tipos *avm* (II)

Pero la versión de la jerarquía de dicha figura es insuficiente. Recuérdese que se ha establecido que todo valor de un rasgo de un tipo de la jerarquía debe haber sido declarado, también, como tipo en la jerarquía. Por tanto, a la jerarquía de la figura 1.48, habrá que añadirle todos los tipos que, en cualquier nivel de anidación, caracterizan al signo lingüístico: su valor categorial (*head*), sus condiciones de subcategorización (*val*), su semántica (*mrs*), etc. En concreto, a la jerarquía de la figura 1.48 habrá que añadirle los tipos de las figuras 1.49, 1.50 y 1.51. En la figura 1.49, se recogen los tipos que son valores de los rasgos que caracterizan al signo en primera instancia. El signo tiene un rasgo *STEM* cuyo valor es la lista de los significantes de todos sus constituyentes. Por tanto la gramática ha de contar con un tipo *list* que formalice el concepto de lista. El tipo *list* tiene un subtipo *null*, que denota una lista vacía —véase el apartado 1.3.3.1—, y un subtipo *cons*, que denota una lista no vacía. Por su parte, la caracterización sintáctica y semántica del signo se modela como valor del rasgo *SYNSEM*. Ese valor es el tipo *synsem*, que ha de recogerse, entonces, en la jerarquía. Una vez incluido *synsem* en la gramática, es a su vez necesario incluir los tipos que son valores de sus rasgos —*LOCAL* y *NON-LOCAL*, donde se formaliza, respectivamente, la información local y no local del signo—. Estos tipos son *local* y *non-local*. En la siguiente figura 1.50, se recogen los últimos tipos complejos contemplados en la gramática hasta el momento, los rasgos del tipo *local*: recuérdese que *head* determina la categoría del signo, *val* sus condiciones de subcategorización

(con las listas SUBJ y COMPS), *agr* agrupa los rasgos de concordancia (PER, NUM y GEND) y *mrs* recoge la información semántica (con la lista de predicados *relation* RELS y el índice del signo INDEX, de valor *index*). Todos los tipos pertinentes para tal caracterización se recogen en la figura 1.50.

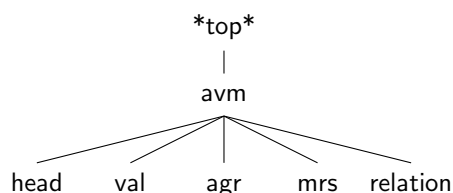


Figura 1.50: Gramática SSG. Jerarquía de tipos *avm* (III)

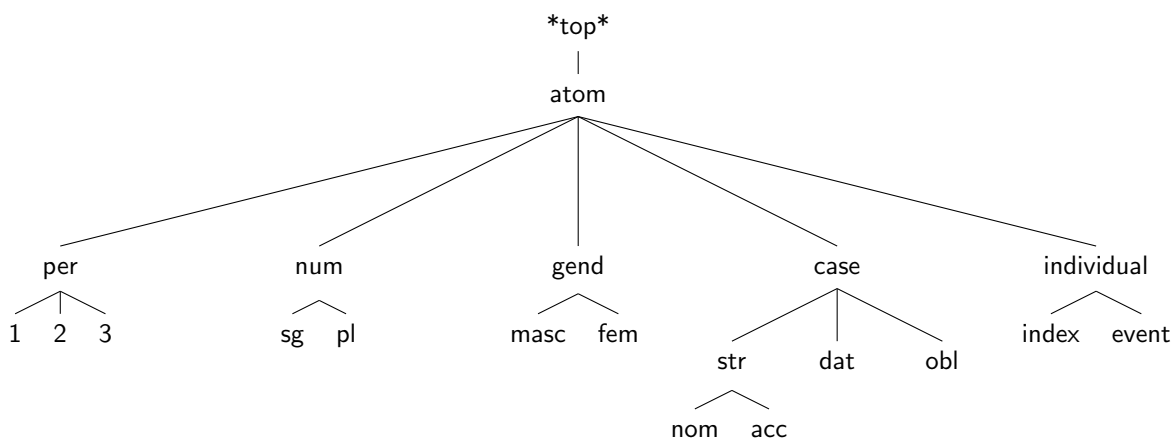


Figura 1.51: Gramática SSG. Jerarquía de tipos *atom*

Para acabar, se incluyen en la jerarquía todos los tipos atómicos vistos hasta el momento: valores de los rasgos PER, NUM, GEND, CASE e INDEX. Véase la figura 1.51. En versiones posteriores de la gramática, los tipos *case* e *individual* pasarán a contemplarse como estructuras complejas, pues se les añadirá cierta caracterización interna. Pero de momento, se clasifican del modo que intuitivamente resulta más adecuado: como tipos atómicos.

Véase cómo toda la caracterización de la gramática SSG que se ha ido desgranando a lo largo de los apartados 1.3.3.1, 1.3.3.2, 1.3.3.3 y 1.3.3.4 confluye en este último apartado en una única jerarquía de tipos. Toda gramática de

tipo HPSG se formaliza únicamente mediante una jerarquía de tipos. Toda realidad gramatical es un tipo de la jerarquía.

1.4. El modelo gramatical de SSG

Se ha visto en el apartado 1.3 qué es una Gramática de Unificación, y se ha usado como ejemplo HPSG. En el apartado anterior —1.2— se explicó, por su parte, la Gramática Transformacional chomskyana. Ambos son modelos gramaticales que han resultado descriptivamente adecuados. Pero sigue abierta la pregunta sobre cuál de los dos modelos es más adecuado desde un punto de vista explicativo. Como ya se ha dicho, cada uno de los modelos se ha desarrollado intentando ser solución a este problema de adecuación explicativa. Pero, en tal intento, han entendido cuestiones distintas por adecuación explicativa. El modelo chomskyano intenta ser compatible con sus hipótesis innatistas y universalistas, y las Gramáticas de Unificación intentan ser modelos plausibles de explicación del procesamiento del lenguaje en tiempo real. Estos no son criterios excluyentes, sino complementarios: forman un único cuerpo de datos empíricos a los cuales debe dar explicación cualquier teoría científica del lenguaje humano. Sin embargo, para este trabajo, dada su vocación aplicada, resulta relevante el criterio de procesamiento en tiempo real más que el de innatismo y universalismo.

De acuerdo con ese criterio, parece que los modelos de unificación son más adecuados: sus reglas son más sencillas, lo cual hace esperar tiempos de procesamiento asumibles; y respetan ciertas características observables en el procesamiento cerebral del lenguaje —véase el apartado 1.3.1—. Por tanto, queda excluido el modelo transformacional como herramienta útil para las labores de Lingüística Computacional que se abordarán en los siguientes capítulos. Se utilizará, en cambio, HPSG. No obstante, el siguiente capítulo 2 está destinado a explicar las debilidades que se observan en las versiones clásicas de HPSG y en sus aplicaciones para el logro de buenos tiempos de procesamiento. Estas debilidades consisten, en concreto, en que el uso masi-

vo de reglas léxicas da lugar, para casos no triviales, a una explosión combinatoria que hace lentas las herramientas de procesamiento basadas en estas gramáticas. SSG quiere ser una gramática del español que, aun siendo fiel al lexicismo de HPSG, minimice la proliferación de reglas léxicas, y con ella, la explosión combinatoria. Es cierto que los análisis propuestos necesitan, en ocasiones, de determinadas herramientas de preprocesamiento, o de algoritmos de *parsing* concretos. Pero, no son estos elementos externos a la gramática los que propician la mejora de los tiempos de procesamiento, sino el limitado uso de reglas que hace la gramática misma.

Capítulo 2

HPSG: problemas

El objetivo fundamental de este trabajo es presentar la gramática computacional SSG, y demostrar cómo, en esta, se da una solución elegante, y computacionalmente adecuada, a ciertos fenómenos problemáticos para las gramáticas previas: *Matrix* —Bender y otros (2002)— y *SRG* —Marimon (2013)—. *Matrix* es un gran logro. Pretende ser un corazón gramatical a partir del cual se pueda implementar una gramática computacional de cualquier lengua natural. Hoy en día, prácticamente cualquier gramática computacional de tipo HPSG está basada, efectivamente, en *Matrix*. Y *SRG*, gramática del español de gran cobertura, el trabajo de referencia en la actualidad, es un ejemplo. SSG no pretende competir con estas gramáticas, ni en cobertura ni en universalidad. Pretende, en cambio, demostrar que sería deseable asumir ciertas estrategias de implementación que favorecen la elegancia y el ahorro computacional. Fundamentalmente, SSG se comparará a partir de ahora con *Matrix*; pues se entiende que esta, en tanto que estándar comúnmente aceptado, recoge las estrategias usadas en la actualidad.

En los siguientes apartados —2.1, 2.2 y 2.3— se explicarán ciertos problemas a los que tiene que hacer frente un implementador de gramáticas computacionales basadas en HPSG. En el resto de capítulos se mostrará SSG y se defenderá que esta gramática soluciona esos problemas.

2.1. Explosión combinatoria

En el capítulo 1, se llegó a la conclusión de que HPSG es un modelo gramatical más adecuado que la Gramática Transformacional para el desarrollo de gramáticas computacionales: sus reglas no transformacionales son más sencillas y menos costosas. Sin embargo, es un hecho que, aun haciendo uso de formalismos gramaticales como HPSG, no se han logrado aún herramientas de procesamiento eficientes para dominios de aplicación amplios. El problema es el aumento inasumible de tiempo de procesamiento según aumenta la cobertura de la gramática. Este trabajo no es, ni con mucho, una solución al problema así planteado; pero sí quiere ser un pequeño avance en ese camino.

En realidad, tanto a HPSG como a la Gramática Transformacional les cuesta tratar eficientemente el hecho de que una misma unidad léxica pueda realizarse de muy distintos modos: piénsese en la alternancia entre activa o pasiva o en la posibilidad de mover o no un sintagma nominal, por ejemplo. Con ambos modelos, se genera una explosión combinatoria inasumible en las labores de procesamiento, al intentar gestionar esta multiplicidad de posibilidades.

Una transformación es muy costosa en sí misma. Pero este modelo puede, en principio, retrasar la decisión sobre si una unidad se está comportando de un modo u otro hasta el momento en el que la secuencia de entrada aporte evidencia sobre la cuestión¹.

En cambio, en HPSG es necesario un sistema generativo léxico —las reglas léxicas— que genere, a ciegas, las posibilidades hipotéticas de realización de cada palabra, antes de iniciarse el proceso sintáctico de saturación. Por ejem-

¹Por ejemplo, imagínese una gramática G utilizada en labores de análisis de una secuencia σ que contiene un verbo transitivo V , que pudiera aparecer en voz activa (V_{act}) o pasiva (V_{pass}). Si G es transformacional, podría generar una misma estructura transitiva —la correspondiente a V_{act} —. A partir de ese momento, solo si se descubren en σ indicios de que V funciona como verbo pasivo, se aplicará una transformación de pasivización sobre la estructura original.

plo, si una gramática de tipo HPSG está analizando una secuencia σ con un verbo transitivo V , tendrá que contemplar, al menos, la hipótesis (V_{act}) de que V funcione en voz activa en σ ; y la hipótesis (V_{pass}) de que V funcione en voz pasiva en σ . V_{act} y V_{pass} son, a efectos del análisis, unidades léxicas diferentes que se lanzan al proceso de saturación sintáctico. Si V funciona como verbo activo en σ , solo V_{act} dará lugar a una derivación exitosa; y si funciona como verbo pasivo, será V_{pass} la unidad léxica que dé lugar a la derivación exitosa. La unidad léxica desechada, no obstante, en tanto en cuanto σ se lo permita, dará lugar a una derivación parcial, luego implicará tiempo de procesamiento inútil.

Este no es un problema baladí. Las posibilidades de realización de cada palabra son muchas, si se cruzan los distintos criterios pertinentes. Cada una de ellas deberá ser creada mediante reglas léxicas, memorizada, y utilizada como desencadenante de derivaciones parciales. En adelante, se llamará explosión combinatoria a esta situación: el cruce de distintos criterios da lugar a un número muy elevado de comportamientos posibles para una palabra. Esta es una realidad inherente a una gramática que solo cuente con reglas de estructura sintagmática, como HPSG. Si se asume, como es el caso, que HPSG es útil como explicación del lenguaje natural; habrá que demostrar que esta característica del modelo, bien gestionada, no supone un coste de procesamiento inasumible. Este trabajo quiere ser un avance hacia el logro de esta buena gestión.

Durante el resto de este apartado —visto que la explosión combinatoria supone un serio coste computacional—, se intentará ilustrar hasta qué punto una palabra puede funcionar de muy distintos modos. Tómese como ejemplo el verbo *subir*. Un verbo como *subir* puede funcionar como transitivo en voz activa —ejemplo (1-a)— o en voz pasiva —(1-b)—. También puede aparecer en construcción de pasiva refleja —(1-c)— o puede funcionar como intransitivo (1-d)—.

- (1) a. El botones ha subido el equipaje

- b. Ha sido subido el equipaje
- c. Se ha subido el equipaje
- d. Ha subido el precio de la habitación

Todos estos patrones se cruzan con la posibilidad de que al verbo se le añada o no un argumento dativo: ejemplos de (2).

- (2) a. El botones nos ha subido el equipaje
- b. Nos ha sido subido el equipaje
- c. Se nos ha subido el equipaje
- d. Nos ha subido el precio de la habitación

A su vez, los patrones de realización de *subir* (hasta el momento ocho), se cruzan con la posibilidad de añadido al verbo de una construcción de control de objeto.

- (3) a. El botones ha subido el equipaje a la habitación
- b. Ha sido subido el equipaje a la habitación
- c. Se ha subido el equipaje a la habitación
- d. Ha subido el precio de las habitaciones a 100 euros

Con ello, *subir* podría obedecer a ocho patrones de realización más (dieciséis en total), los cuales se muestran en los ejemplos de (3) y de (4).

- (4) a. El botones nos ha subido el equipaje a la habitación
- b. Nos ha sido subido el equipaje a la habitación
- c. Se nos ha subido el equipaje a la habitación
- d. Nos ha subido el precio de las habitaciones a 100 euros

Además, ténganse en cuenta las diferentes posibilidades de ordenación lineal de los complementos del verbo *subir*. Obsérvese en los siguientes ejemplos que los complementos del verbo *subir* admiten, en posición posverbal, cualquier orden lineal relativo. Hasta ahora se han contemplado cuatro patrones con un complemento posverbal, ocho con dos complementos posverbales, y cuatro con tres. Por tanto, el añadido del criterio de orden lineal supone un total de

treinta y dos patrones de realización para el verbo *subir*. Se muestran en (5) las posibilidades de ordenación lineal de dos complementos del verbo *subir*.

- (5) a. El botones le ha subido el equipaje al guía
- b. El botones le ha subido al guía el equipaje
- c. El botones ha subido el equipaje a la habitación
- d. El botones ha subido a la habitación el equipaje

Por su parte, las siguientes, con su debido patrón entonativo, son algunas de las posibilidades de ordenación lineal de tres complementos posverbiales de este verbo: (6) y (7).

- (6) a. El botones le ha subido el equipaje al guía a la habitación
- b. El botones le ha subido al guía el equipaje a la habitación
- c. El botones le ha subido a la habitación el equipaje al guía
- (7) a. El botones le ha subido a la habitación al guía el equipaje
- b. El botones le ha subido el equipaje a la habitación al guía
- c. El botones le ha subido al guía a la habitación el equipaje

Téngase en cuenta que, a su vez, el sujeto de *subir* pudiera omitirse; y los complementos acusativo y dativo pudieran pronominalizarse como clíticos. Además, los tres tipos de argumento pueden desplazarse a la izquierda de la oración (como tópicos, como focos o como partículas interrogativas). Es fácil calcular que los treintaidós patrones de realización vistos hasta ahora, combinados con estos nuevos criterios, darán lugar a un número realmente elevado de posibilidades de realización del verbo *subir*.

Se puede discutir si estas cantidades son manejables, o no, para un procesador. Quizá dependa del tipo de procesador. Se puede discutir, por ejemplo, si son manejables para el cerebro humano, para un procesador determinista, para un procesador no determinista, etc. Este trabajo no puede aportar nada a esta discusión. Sí se pretende demostrar, en cambio, que las estrategias de implementación de SSG suponen una reducción significativa de estas cantidades, con respecto, por ejemplo, a las estrategias seguidas en **Matrix**.

Esta reducción, como es lógico, para los procesadores deterministas actuales, acarrea un ahorro en tiempo de procesamiento.

2.2. Adecuación descriptiva

En el apartado previo se explicó cómo las gramáticas computacionales basadas en HPSG sufren problemas de explosión combinatoria que hacen difícil el logro de buenos tiempos de procesamiento. SSG quiere ser una gramática que, gracias a sus estrategias de implementación lingüísticamente motivadas, reduzca esta explosión combinatoria. Estas estrategias tendrán mucha relación con el uso de las reglas léxicas.

Además, en este apartado se sugiere la necesidad de mejora del concepto de regla léxica en HPSG, por razones de elegancia teórica. A lo largo del trabajo se defenderá que SSG utiliza un sistema de reglas léxicas sencillo y elegante, desde un punto de vista teórico. Las reglas léxicas de HPSG, tal como se utilizan habitualmente, se parecen en cierto sentido a las transformaciones de las primeras versiones de la gramática chomskyana. Estas eran reglas muy poderosas. Podían añadir elementos a una derivación, modificarlos o eliminarlos. Podían también permutar las posiciones de varios elementos. Es cierto que desde un punto de vista de coste computacional las reglas léxicas son radicalmente distintas a estas transformaciones². Sin embargo, como ellas, las reglas léxicas permiten modificar un elemento prácticamente de cualquier modo: quitan o ponen argumentos, permutan sus posiciones, etc.

La gramática chomskyana ha evolucionado, desde sus primeras versiones, en busca de una adecuación explicativa, para la cual, ese tipo de reglas eran un problema. Se entendía que una gramática era adecuada desde un punto de vista explicativo si era compatible con las tesis universalistas. Un modelo con

²Las reglas léxicas operan sobre estados resultantes de las derivaciones, mientras que las transformaciones operan sobre derivaciones, lo cual las hace mucho más costosas.

muchas reglas, muy diferentes unas de otras, no era, entonces, adecuado: era difícil explicar que a esa compleja diversidad subyacía una Gramática Universal. Por ello, en el modelo de Principios y Parámetros, las diversas reglas de estructura sintagmática anteriores se redujeron a una única forma establecida por la Teoría de la X con Barra; y las diversas transformaciones se redujeron a una única operación de movimiento, sujeta a ciertas constricciones. Este modelo es más compatible que el anterior con las tesis universalistas. Es más adecuado, por tanto, desde un punto de vista explicativo.

Pero lo que interesa aquí es que es un modelo más sencillo, más elegante: en cualquier disciplina científica es preferible, entre dos modelos que expliquen los mismos datos, aquel que cuente con menor aparato teórico. Por tanto, independientemente de que el modelo de Principios y Parámetros alcanzase una determinada adecuación explicativa, desde luego, supuso una mejora en cuanto a adecuación descriptiva: lo que era un sistema de muy distintos tipos de reglas se redujo a unos pocos patrones.

Dada la similitud existente entre las primeras transformaciones y las reglas léxicas —ambas operan todo tipo de cambios sobre el *input*—, sería interesante para HPSG una revisión similar a la de la gramática chomskyana, en busca, quizá no de adecuación explicativa³, pero sí descriptiva. Sería interesante alcanzar un sistema de reglas léxicas que restringiese la forma y función de estas, lo cual daría lugar a un modelo conceptualmente más sencillo. Además, esta sencillez simplificaría las labores de implementación y mantenimiento de las gramáticas computacionales.

En el capítulo 3 se explicará el sistema de reglas léxicas de SSG. Este sistema pretende dar solución al problema planteado en este apartado. Se explicará en qué sentido concreto se restringe lo que puede hacer una regla léxica de tipo HPSG. Esta restricción dará lugar a un sistema en el que, efectivamente, todas las reglas obedecen a un mismo patrón de comportamiento

³HPSG no está tan vinculado con las tesis universalistas como el modelo chomskyano.

sencillo. Con ello se logra, por un lado, una teoría sencilla y elegante; y por otro, una práctica de implementación que facilita el desarrollo y mantenimiento de la gramática.

2.3. Motivación lingüística

Hay ciertos fenómenos que las gramáticas computacionales de tipo HPSG vienen tratando de forma, desde un punto de vista lingüístico, antinatural. Quizá haya casos en que estas prácticas sean inevitables, pero la tendencia debe ser crear gramáticas naturales desde un punto de vista teórico: con motivación lingüística. Esto, además del interés teórico que pueda suponer, favorece el que la gramática sea fácil de mantener y de ampliar; y, como se verá, suele ayudar al ahorro computacional.

Un ejemplo de esta situación es el siguiente. La gramática *Matrix* proporciona al implementador unos tipos de lexema básicos. Por ejemplo, *transitive-lex-item* modela los lexemas transitivos cuyos argumentos son de naturaleza nominal; mientras que *clausal-second-arg-trans-lex-item* modela los lexemas transitivos cuyo último argumento es oracional. De acuerdo con ello, la alternancia que se muestra en (8) se recogería en *Matrix* aplicando sobre *saber* el patrón *transitive-lex-item* en un caso, y *clausal-second-arg-trans-lex-item* en otro.

- (8) a. ¿Sabes el precio de la habitación?
b. ¿Sabes si el hotel tiene sala de reuniones?

Con este análisis se pierde la generalización consistente en que, para *saber*, la alternancia de (8) obedece a un único patrón de comportamiento transitivo. El motivo de la alternancia es, sencillamente, que tanto un sintagma nominal como un complementante unifican con la caracterización del argumento interno de ese único patrón. La utilización en *Matrix* de dos patrones distintos (*transitive-lex-item* y *clausal-second-arg-trans-lex-item*) no capta esa intuición. Además, favorece la explosión combinatoria. Si estos dos patrones

se pudiesen neutralizar en uno, se ganaría en generalidad y sencillez; y se ahorraría en tiempo de procesamiento.

Además, en *Matrix*, junto a *transitive-lex-item*, se contempla el tipo *ditransitive-lex-item*, que modela a las unidades léxicas ditransitivas. Con estos tipos se puede dar cuenta de los usos del verbo *subir* que aparecen en (9-a) y (9-b).

- (9)
- a. El botones ha subido el equipaje
 - b. El botones le ha subido el equipaje al guía
 - c. Ha subido el precio de las habitaciones
 - d. Les ha subido el precio de las habitaciones

De acuerdo con *Matrix*, (9-a) es resultado de la aplicación del patrón *transitive-lex-item* sobre el verbo *subir*; mientras que (9-b), con añadido de un dativo beneficiario, es resultado de la aplicación de *ditransitive-lex-item*. De nuevo, la existencia de estos dos patrones distintos no capta las debidas generalizaciones. Parece apropiado entender que el argumento acusativo de (9-a) y (9-b) aparece en ambas oraciones en virtud de un mismo proceso gramatical; al que después se puede añadir, o no, el argumento dativo. Pero, dados *transitive-lex-item* y *ditransitive-lex-item*, la aparición del acusativo viene legitimada por patrones diferentes en cada oración.

Del mismo modo, parece adecuado entender que el añadido de un dativo beneficiario es siempre un mismo patrón gramatical, independientemente de que se aplique sobre un verbo transitivo —(9-b)— o intransitivo —(9-d)—. De nuevo, *Matrix* no capta esta generalización, pues entiende a (9-a) y (9-d) como partícipes de dos patrones diferentes *ditransitive-lex-item* y *transitive-lex-item*, respectivamente.

Otro campo en el que se observan soluciones no motivadas lingüísticamente es el del tratamiento de los diferentes órdenes relativos entre complementos. Las gramáticas computacionales de tipo HPSG tienen ciertos problemas para tratar lenguas, como el español, en que los complementos de

un verbo admiten sin problemas diferentes ordenaciones lineales. Ya se vio este fenómeno para los complementos del verbo *subir*, en el apartado 2.1. Se repite la alternancia de orden entre acusativo y dativo en (10).

- (10) a. El botones le ha subido el equipaje al guía
b. El botones le ha subido al guía el equipaje

La solución que se da a estas situaciones en gramáticas al uso —véase el apartado 5.2.1— consiste en multiplicar las reglas que saturan los argumentos. Además de la regla común que satura el primer elemento de una lista de complementos, se añade otra que satura el segundo, otra que satura el tercero, etc. Otro modo obvio de dar cuenta de los órdenes de (10) es crear reglas léxicas que generen las posibilidades de ordenación pertinentes. Cualquiera de estas dos estrategias acarrea ciertos problemas técnicos, pero sobre todo son soluciones *ad hoc*, carentes de motivación lingüística.

En el capítulo 3 se defenderá que SSG modela las alternancias léxicas como las vistas en los ejemplos de (8) y (9) de forma lingüísticamente motivada, preservando las debidas generalizaciones. Incluso se defenderá que la solución utilizada reduce significativamente la explosión combinatoria. Además, se dará una solución, también motivada lingüísticamente, para el problema de orden lineal de los ejemplos de (10): capítulo 5.2. En este caso, la solución es la aplicación al español del análisis comúnmente aceptado en los estudios teóricos de HPSG relativos al *scrambling*.

2.4. Estado de la cuestión

El problema de la explosión combinatoria visto en 2.1 no ha pasado inadvertido en la bibliografía. En este apartado se mostrarán las distintas soluciones planteadas hasta el momento para tal problema. En primer lugar, la complejidad computacional resultante de la explosión combinatoria se ha intentado reducir desde planteamientos estadísticos. Los planteamientos es-

tadísticos⁴ utilizan información extraída de un corpus previamente anotado morfosintácticamente. Dado un nodo estructural, se observa qué reglas gramaticales permiten crear tal nodo en el corpus y qué probabilidad existe de que la regla utilizada sea una u otra. La probabilidad de que en un análisis se use una secuencia concreta de reglas es el producto de la probabilidad de cada una de las reglas implicadas. Aplicando este conocimiento al problema de la combinación de reglas léxicas, puede deducirse qué secuencias de reglas léxicas son más probables como análisis de una entrada léxica. Gracias a esta información, es posible implementar algoritmos de análisis que, en principio, solo contemplen los análisis más probables para una entrada léxica. Solo en caso de que tales análisis fallen se daría paso a los análisis menos probables. Gracias a ello, es posible obviar en multitud de situaciones buena parte de los patrones de comportamiento posibles para las unidades léxicas. Esta reducción de posibilidades reduce la explosión combinatoria y los tiempos de análisis. En definitiva, las aproximaciones estadísticas intentan mejorar la eficiencia del algoritmo de análisis y no la de la codificación misma de la gramática⁵.

Otro algoritmo interesante para el tratamiento eficiente de los sistemas de reglas léxicas en las gramáticas de tipo HPSG es el propuesto en Meurers y Minner (1997). En este estudio se documenta la implementación de un compilador de sistemas de reglas léxicas que reduce significativamente la explosión combinatoria. Para ello, el compilador toma el sistema de reglas léxicas de una gramática y, a partir de él, asigna a cada unidad léxica de esta una caracterización eficiente de sus posibilidades de realización. Una vez el compilador ha hecho su trabajo, la complejidad del análisis en tiempo de ejecución se ve seriamente reducida. En concreto, en Meurers y Minner (1997), durante el análisis, a una unidad léxica se le asignan solo las secuencias de reglas léxicas que sí conducen a un análisis posible para ella. Se evita así que el algoritmo pruebe reglas, o secuencias de reglas, que al final serán análisis

⁴Véase Moreno Sandoval (1998) pp. 198-203.

⁵Véase Tsuruoka y otros (2004) como ejemplo de uso de algoritmos con información probabilística con gramáticas de tipo HPSG.

imposibles para la unidad léxica en cuestión.

El compilador de Meurers y Minner (1997) distingue, además, para cada regla léxica, cuál es la información del *input* que la regla cambia en el *output* y cuál es la información que permanece inalterada. A la información inalterada se le llama *frame*. Dicho de forma intuitiva, el *frame* de una regla léxica es el escenario sobre el cual la regla opera. Dada la naturaleza de las estructuras de rasgos de HPSG, es frecuente que para una misma regla léxica deban distinguirse varios *frames*. Sin embargo, el compilador es capaz de seleccionar, para cada unidad léxica concreta, los *frames* de sus reglas léxicas que no unifican con ella. Gracias a ello, estos *frames* podrán no tenerse en cuenta en tiempo de ejecución para esta palabra en concreto.

Por último, el compilador de Meurers y Minner (1997) permite determinar, para cada unidad léxica, cuál es el *frame* aplicable a cualquiera de sus patrones de realización: el común a todas sus reglas léxicas. En un primer momento del análisis, basta con especificar cada palabra con la estructura de este *frame*. La aplicación de las secuencias de reglas léxicas puede posponerse, entonces, a estadios avanzados del análisis. Esta posposición siempre es deseable para la reducción de la explosión combinatoria, pues el proceso sintáctico especifica la estructura de la unidad léxica y reduce así el número de reglas compatibles con ella. En resumen, la propuesta de Meurers y Minner (1997) consiste en reducir en tiempo de compilación las posibilidades reales de aplicación de reglas léxicas a cada unidad del lexicón. De este modo, el proceso de análisis en tiempo de ejecución resulta mucho más sencillo.

Actualmente, este tipo de estrategias consistentes en la mejora del algoritmo de análisis son las más habituales en las aplicaciones computacionales. Pero, siguiendo a Flickinger (2000), es también necesario determinar qué tipo de análisis propiamente lingüísticos permiten reducir significativamente la complejidad computacional de las gramáticas. En Flickinger (2000) se docu-

mentan tres experimentos llevados a cabo en *English Resource Grammar*⁶ —ERG—, una gramática de tipo HPSG del inglés. En primer lugar, este autor eliminó los operadores disyuntivos que hasta entonces existían en las estructuras de rasgos de ERG. Las alternativas que permitía codificar este operador pasaron a modelarse como distintos subtipos de un supertipo general. Este cambio de estrategia acarrió una reducción muy significativa de los tiempos de análisis de ERG.

El segundo experimento que documenta Flickinger (2000) hace referencia a la codificación de complementos opcionales en los núcleos léxicos de ERG. La versión original de ERG contaba con reglas gramaticales específicas para eliminar tales complementos de las listas de subcategorización cuando estos no aparecían en la secuencia de entrada. Pero ERG ganó en eficiencia gracias a la eliminación de estas reglas. Los complementos opcionales de los núcleos pasaron a codificarse en un nuevo tipo de lista: *opt-nelist*. Esto permitía considerar a un núcleo como saturado tanto si sus rasgos de subcategorización tenían por valor una lista vacía —*null*— como si tenían una lista *opt-nelist*. De nuevo, el mero uso de una estrategia de codificación gramatical ayudó a la reducción significativa de los tiempos de análisis.

Por último, Flickinger (2000) documenta ciertos cambios en la codificación de ERG que condujeron a una reducción significativa de las estructuras de rasgos implicadas en los procesos de unificación. En ERG —y en cualquier gramática de tipo HPSG— hay algunos rasgos que aparecen con enorme frecuencia en las estructuras. Se trata de rasgos propios de cualquier signo lingüístico, como *CAT* que, en ERG, recoge la información categorial del signo. Estos rasgos tienen habitualmente complejas estructuras de rasgos anidadas. Sin embargo, en multitud de ocasiones, estas estructuras no presentan ninguna especificación añadida a las propias del tipo genérico para el que se declararon. Así y todo, los algoritmos de unificación las recorren también, pues no tienen modo de saber si en algún lugar de estas habrá información

⁶Copestake y Flickinger (2000).

relevante. Para evitar esta situación, en ERG se cambió el valor habitual de estos rasgos: si hasta entonces el valor del rasgo F había sido el tipo t (con su compleja estructura de rasgos anidada), el valor de F paso a ser un nuevo tipo t_{\min} sin estructura de rasgos anidada, donde t era subtipo inmediato de t_{\min} . De este modo, si un signo lingüístico cuenta con alguna caracterización específica de F , su valor será t . Pero si su caracterización de F es la declarada en la gramática para cualquier rasgo F , su valor será t_{\min} . Lo interesante es que solo en el primer caso el algoritmo de unificación deberá recorrer la estructura anidada en F . Flickinger (2000) documenta cómo esta estrategia consiguió también una reducción significativa de los tiempos de análisis de ERG.

En definitiva, Flickinger (2000) utiliza estos tres experimentos para ilustrar cómo la elección de una u otra estrategia de codificación tiene un impacto muy significativo en la eficiencia de la gramática. No obstante, tal como él explica, no abundan en la bibliografía investigaciones que demuestren qué prácticas de codificación son preferibles desde un punto de vista de coste computacional. El presente trabajo es, precisamente, una investigación de tal tipo: en él se documenta cómo ciertos análisis concretos del español conducen a la reducción del coste computacional de las gramáticas. Este tipo de análisis es compatible con el uso de algoritmos de análisis con información probabilística o con el algoritmo de Meurers y Minner (1997). Para el desarrollo de aplicaciones computacionales será necesario, por un lado, el uso de algoritmos de análisis eficientes y, por otro, el uso de una codificación gramatical eficiente como la propuesta en esta tesis.

2.5. Conclusión

En el presente capítulo 2 se han mostrado ciertos problemas que existen en la actualidad para la implementación de gramáticas computacionales de tipo HPSG.

Se ha discutido un problema de índole práctica al que se enfrentan las gramáticas de cobertura amplia: la explosión combinatoria, con su consecuente incremento de tiempos de procesamiento. En la práctica, una palabra puede realizarse de muy distintos modos, si se atiende a los muchos criterios existentes de caracterización de su comportamiento: número de argumentos, naturaleza y orden de estos, etc. Un *parser* debe contemplar todas las posibilidades de realización de cada palabra, y guardarlas en memoria. A continuación debe, para cada una de estas posibilidades, comprobar si la gramática contempla un análisis compatible con la cadena de entrada. Cuando los patrones de realización de las palabras son muchos, los pasos que debe seguir el *parser* son muchos, y la información que debe guardar en memoria también; y todo ello aumenta los tiempos de análisis. Es una preocupación del implementador el que la complejidad del proceso no alcance un nivel que pida tiempos de procesamiento excesivos.

También se han presentado problemas de tipo teórico. Las gramáticas computacionales —si bien no son estudios propiamente teóricos, ligados a teorías concretas— aspiran a tener una cierta fundamentación teórica, coherencia interna y elegancia en sus resoluciones. Se prefieren las soluciones generales al modelado *ad hoc* de cada fenómeno. Estas prácticas favorecen el que las gramáticas sean fáciles de mantener y aumentar. En este capítulo se ha defendido que las reglas léxicas de HPSG permiten cualquier tipo de alteración del *input*, y que sería deseable restringir esa posibilidad, de forma que todas estas reglas sigan un mismo patrón de comportamiento. Con ello, el modelo ganaría en sencillez y elegancia. También se han mostrado fenómenos concretos para los cuales las gramáticas computacionales, habitualmente, no dan soluciones que obedezcan a la intuición lingüística del implementador (ciertas alternancias de diátesis y distintos órdenes relativos entre complementos).

En los próximos capítulos se explica la gramática que es objeto fundamental de este estudio: SSG. Se defiende que esta gramática logra un modelado elegante y eficiente de buena parte de los fenómenos centrales del español.

Es un modelado que quiere ser solución a los problemas mostrados en este capítulo 2.

Capítulo 3

SSG: alternancias de diátesis

En el capítulo 1 se escogió HPSG como modelo teórico para la implementación de SSG. A continuación, en el capítulo 2, se mostraron ciertos problemas propios de las gramáticas computacionales basadas en este modelo —fundamentalmente, de *Matrix*—. Estos problemas eran relativos a los tiempos de procesamiento (la explosión combinatoria), o a la sencillez y elegancia teórica (excesivo potencial de las reglas léxicas, y estrategias de implementación que no reflejan las debidas generalizaciones). En 2 se decía que SSG es un avance en la solución de esos problemas.

De un modo u otro, todas las cuestiones expuestas en 2 tenían relación con el sistema de reglas léxicas. Este apartado explica el funcionamiento del grueso de las reglas léxicas de SSG. Se describe cómo la gramática da cuenta de las distintas construcciones de las que puede participar una palabra; en adelante, alternancias de diátesis. En realidad, el estudio se ciñe a las alternancias de diátesis propias de los verbos del español. En los capítulos siguientes se completará esta discusión con otro tipo de alternancias relativas a cliticización, omisión y desplazamiento de los argumentos verbales.

El capítulo se organiza del siguiente modo. En el primer apartado 3.1 se revisa la formalización del signo lingüístico hecha en 1.3.3.1. Se añadirán a este signo rasgos necesarios para formalizar las propuestas de este capítu-

lo. En 3.2 se explicará la clasificación semántica de tipos verbales de Levin y Rappaport (1995) y Levin y Rappaport (1998). Los tipos verbales propuestos por estas autoras serán los que se contemplen en SSG. En gran medida, la pertenencia de un verbo a una de estas clases semánticas condicionará su comportamiento sintáctico, en cuanto a número y tipo de argumentos, participación de ciertas construcciones, etc. En el apartado 3.3 se explicarán, entonces, las reglas léxicas mediante las cuales SSG modela todas aquellas construcciones verbales relacionadas con la presencia de argumentos de caso nominativo y acusativo (transitiva, pasiva, etc.). En el apartado 3.4 se presenta el caso especial de los verbos pronominales. A continuación, en 3.5 se explicará cómo se regula en SSG, mediante el uso de reglas léxicas, la posibilidad de añadido de un argumento de caso oblicuo o dativo, o de un complemento predicativo, a un verbo. Por último, en los apartados 3.6 y 3.7, se valora el interés teórico del sistema de reglas léxicas de SSG.

3.1. Revisión del signo lingüístico (I)

En este primer apartado se amplía la versión de SSG vista hasta el momento. El objetivo es dotar a SSG de las herramientas pertinentes para el logro de un correcto modelado de los fenómenos del español que se mostrarán en el resto del capítulo.

3.1.1. El rasgo `HOOK`

Recuérdese que la caracterización semántica del signo se formalizó en 1.3.3.3 mediante el tipo `mrs`, valor del rasgo local `SEM`. En aquella primera aproximación solo se le supusieron a `mrs` dos rasgos: `INDEX` y `RELS`. Ahora se asumirá —siguiendo a *Matrix*— la siguiente estructura de la figura 3.1. En la versión previa, `INDEX` era el rasgo en el que se publicaba el índice semántico del signo. Ese índice ya aparecía dentro de la lista `RELS` como valor de algún rasgo `ARG0`. Pero, como es complejo acceder al interior de esta lista, resultaba útil rescatar el índice en cuestión en `INDEX`. Una vez rescatado, puede ser utilizado desde fuera del signo mismo: quizá, por un núcleo que lo subcategorice, o por

un modificador que predique sobre él. Pero, como se verá, es común que en RELS haya más índices que merezcan ser rescatados. Es el caso, para los predicados, del índice del argumento sobre el que predicán. Este índice se rescatará de RELS en el rasgo XARG. Por ello, se asume la existencia de un rasgo HOOK en cuyo valor *hook* se recogen los distintos índices interesantes de un signo lingüístico (INDEX, el índice del signo mismo; y XARG, el índice sobre el que predica el signo si es un predicado).

$$\left[\begin{array}{l} mrs \\ \\ HOOK \left[\begin{array}{l} hook \\ XARG \textit{ index} \\ INDEX \textit{ individual} \end{array} \right] \\ RELS \textit{ list} \end{array} \right]$$

Figura 3.1: Tipo *mrs*: revisión (I)

La de la figura 3.1 es la versión del tipo *mrs* de la gramática **Matrix**. En SSG se ha asumido esa estructura, pero se ha enriquecido añadiendo a HOOK otros índices, que pueden ser interesantes para una correcta caracterización semántica del signo.

$$\left[\begin{array}{l} mrs \\ \\ HOOK \left[\begin{array}{l} hook \\ XARG \textit{ index} \\ IARG \textit{ semarg} \\ EARG \textit{ semarg} \\ CARG \textit{ semarg} \\ PARG \textit{ semarg} \\ INDEX \textit{ individual} \end{array} \right] \\ RELS \textit{ list} \end{array} \right]$$

Figura 3.2: Tipo *mrs*: revisión (II)

El valor de EARG será el índice del argumento externo del signo (si este lo tuviera), mientras que IARG es el índice correspondiente al argumento interno. EARG e IARG no son, necesariamente, índices diferentes al de XARG. La

idea es que, normalmente, un predicado con argumento externo (rescatado en EARG) predicará sobre este, con lo cual, se identificarán en él EARG y XARG. Como segunda opción, si el predicado no muestra argumento externo, XARG se identificará con el argumento interno, con IARG. En CARG se publica el índice del argumento sobre el que deberá predicar un hipotético complemento predicativo. Por último, PARG publica el índice del sujeto de los verbos pronominales. Véase la que será ya versión definitiva de *mrs* en la figura 3.2.

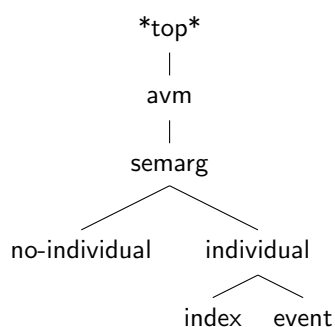


Figura 3.3: Índices semánticos: jerarquía de tipos

Además, en SSG se utiliza la siguiente jerarquía de tipos para formalizar la noción de índice semántico. La cuestión novedosa es la inclusión de un tipo *no-individual*, que denota la no existencia de un índice. Gracias a este tipo, es posible formalizar la idea de que un signo no tiene algún tipo de índice: un predicado no tiene argumento externo (EARG *no-individual*) o no tiene argumento interno (IARG *no-individual*).

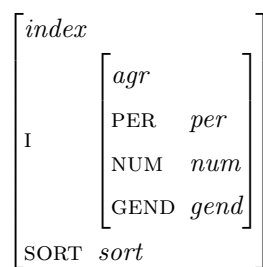


Figura 3.4: Tipo *index*

También se asume —como en Pollard y Sag (1994) y en *Matrix*— que los índices semánticos no son tipos atómicos. Estos tienen, en cambio, rasgos de

concordancia (l agr).

El tipo *sort*, o sus subtipos, no es necesariamente atómico. Como se verá en el próximo apartado 3.2, al tipo *sort* se le suponen, al menos, dos rasgos: ACT y BECOME, de valor booleano (*bool*). Es decir, independientemente de la clase semántica a la que pertenezca un signo, este va a estar caracterizado sobre si denota, o no, el desarrollo de una actividad (ACT *yes* o ACT *no*); y si denota, o no, un cambio de estado (BECOME *yes* o BECOME *no*).

Por último, se explicará una convención asumida en SSG, relativa a los papeles temáticos que definen los predicados verbales. Se ha visto cómo cada predicado identifica los índices de sus argumentos con el valor de los rasgos ARG1, ARG2, ARG3, ARG4, etc. Estos atributos se han tomado de *Matrix*. Son atributos que no dicen nada sobre el tipo de rol temático que denotan. Sin embargo, en SSG sí se les va a adscribir cierta identidad con roles temáticos concretos. Con ello se va a conseguir que la caracterización semántica del predicado diga mucho de su posible comportamiento sintáctico, sin necesidad de hacer una formalización concreta de este último. La creación de EARG e IARG tiene el mismo fin. Como se verá en adelante, esto será crucial para la creación del sistema de reglas léxicas de SSG. Así pues, ARG1 se identificará con los argumentos externos: normalmente el agente (*María* en *María corrió hasta el final*), pero también una causa (*La tormenta* en *La tormenta ha roto la ventana*), y los experimentantes (*María* en *María vio lo sucedido*). ARG2 y ARG3 se identificarán con distintos tipos de argumento interno: ARG2 lo hará con pacientes afectados por un cambio de estado (*la ventana*, en *María ha roto la ventana*), mientras que ARG3 lo hará con temas que no denotan un paciente afectado (*una cosa* en *María ha pensado una cosa*). Por último, el rasgo ARG4 se identificará con los argumentos que denotan al beneficiario del proceso (*a los vecinos* en *María les ha roto la ventana a los vecinos*), poseedores en dativo, etc.

Gracias a la combinación de ARG1, ARG2, ARG3, ARG4, EARG, IARG y XARG, CARG y PARG, SSG cuenta con una caracterización rica y accesible de la na-

turalidad semántica de un verbo. Esto le permitirá restringir adecuadamente qué verbos pueden participar de qué patrones de comportamiento, sin —esto es crucial— partir de una caracterización sintáctica rica; pues, normalmente, los patrones de comportamiento de un verbo piden caracterizaciones sintácticas incompatibles entre sí.

3.1.2. Teoría del ligamiento: ARG-ST

Pollard y Sag (1994) asumen la existencia de la siguiente jerarquía de tipos semánticos nominales, subtipos de *mrs*: figura 3.5.

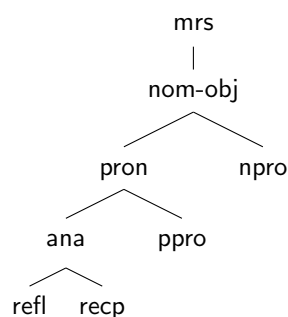


Figura 3.5: Tipos semánticos nominales

El tipo *pron* define la semántica propia de los pronombres, que pueden ser pronombres personales (*ppro*) o anáforas (*ana*). Las anáforas, a su vez, pueden ser reflexivas (*refl*) o recíprocas (*recp*). El tipo *npro* representa, por su parte, la semántica propia de los nombres propios y de los sintagmas nominales cuantificados. Para el español, el tipo *ppro* es el propio de los pronombres personales (*yo, tú...*); *refl* corresponde a la anáfora *a sí mismo*; y *recp* a la anáfora recíproca *entre sí*. Por último, *npro* define la semántica propia de cualquier nombre propio o sintagma nominal cuantificado. También las unidades léxicas sin realización fonética se pueden caracterizar conforme a esta clasificación semántica¹.

¹En el modelo de Principios y Parámetros, Chomsky entiende que el movimiento a posiciones argumentales deja como huella una anáfora; mientras que el que tiene como destino una posición no argumental deja como huella una variable —equivalente, en cuanto

El que una unidad léxica nominal cuente con uno u otro de estos patrones semánticos condiciona su comportamiento, en lo que a las condiciones de ligamiento se refiere. De este modo, los pronombres personales (**ppro**) no pueden ser correferentes con un elemento nominal que los preceda, dentro de determinada configuración sintáctica O —(1-b)—. Pero sí pueden ser correferentes con un elemento precedente, si este se encuentra fuera de dicha configuración O —(1-d)—. Los nombres propios y los sintagmas nominales cuantificados (**npro**), por su parte, no pueden ser correferentes, nunca, con un elemento nominal precedente —(1-a) y (1-c)—. En cambio, las anáforas (**ana**) sí deben ser correferentes con otro elemento nominal que las preceda, dentro de esa misma configuración sintáctica O —véase (1-e)—.

- (1) a. [$\acute{E}l_i$ vio a $Juan_j$] $_O$
 b. [$Juan_i$ lo vio a $\acute{e}l_j$] $_O$
 c. [$\acute{E}l_i$ dice que [$Teresa$ vio a $Juan_j$] $_O$] $_O'$
 d. [$Juan_i$ dice que [$Teresa$ lo vio a $\acute{e}l_i$] $_O$] $_O'$
 e. [$Juan_i$ se vio a sí mismo $_i$] $_O$

Estos contrastes se explicaron en la Gramática Transformacional mediante la llamada Teoría del Ligamiento de Chomsky (1986b). Pollard y Sag (1994) revisaron la propuesta de Chomsky para adaptarla a HPSG. Y Sag y otros (2002) hicieron una nueva formulación que mejoraba la anterior².

a condiciones de ligamiento, a un sintagma nominal cuantificado—. Además, las lenguas que, como el español, permiten la elisión del sujeto, tienen pronombres personales sin realización fonética (**pro**). Véase Chomsky (1986b).

²En buena medida, los esfuerzos de estas obras de HPSG están orientados a formular la propuesta de Chomsky de un modo no configuracional. Nótese que la anterior explicación de los contrastes de (1) ha hecho uso —como Chomsky (1986b)— de la noción de configuración sintáctica. Una configuración sintáctica es un objeto gramatical en el que se explicitan las relaciones de dependencia entre un sintagma y sus constituyentes; es decir, es un árbol sintáctico. HPSG no contempla, por tanto, la configuración sintáctica como objeto gramatical. Así pues, es un reto formular la Teoría del Ligamiento de Chomsky sin hacer uso de este concepto. De hecho, la coindización de (1-d) se establece entre dos elementos (*Juan* y *él*), tales que el primero se une a una proyección de la que es constituyente el

Para explicar los contrastes de las oraciones de (1), Sag y otros (2002) añaden a la estructura de rasgos del signo lingüístico el rasgo ARG-ST. En ARG-ST se listan todos los argumentos subcategorizados por el signo, que después se desglosarán, mediante estructura compartida, entre SUBJ y COMPS. En SSG, ARG-ST se ha definido para el tipo *lexeme*: figura 3.6.

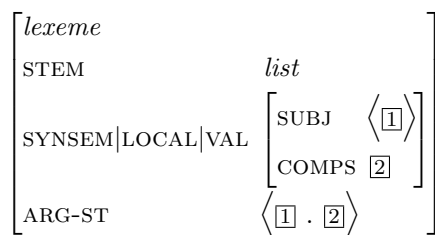


Figura 3.6: Relación entre ARG-ST y VAL

La saturación sintáctica se sigue haciendo en las listas de VAL, pero las relaciones de ligamiento se establecen en ARG-ST. ARG-ST es el único dominio dentro del cual se puede afirmar que dos elementos deben o no estar coindizados. Además, los argumentos se listan en ARG-ST en orden creciente de oblicuidad. De acuerdo con estas dos premisas, las condiciones de ligamiento consistirán, en cualquier caso, en que un elemento nominal debe, o no debe, estar coindizado con otro que lo preceda en ARG-ST. O, lo que es lo mismo, un elemento nominal debe, o no debe, estar coindizado con otro menos oblicuo que él, dentro del dominio de su lista ARG-ST. Concretamente, de acuerdo con todo ello, los contrastes de los ejemplos de (1) obedecen a los dos principios siguientes.

- Principio A: Toda anáfora (*ana*) debe estar coindizada con un elemento nominal que lo preceda en ARG-ST.
- Principio B: Todo pronombre (*ppro*) puede estar coindizado con un elemento nominal que no lo preceda en ARG-ST.

segundo. Parece, como el movimiento, la típica situación que solo gestiona adecuadamente una Gramática Transformacional, que contempla el árbol sintáctico como objeto.

Obsérvese que estos principios no dicen nada de las posibilidades de ligamiento de los sintagmas nominales cuantificados y nombres propios (*npro*). Se asume que los índices no se identifican, por defecto, a menos que los principios del ligamiento lo permitan explícitamente. De este modo, se excluye la posibilidad de que estos elementos nominales sean correferentes con elementos que los precedan en una configuración sintáctica, sin usar propiamente nociones configuracionales. Esta solución se toma de Sag y otros (2002)³.

No obstante, estos principios de ligamiento no están lo suficientemente formalizados como para aplicarse directamente en una gramática computacional. El Principio B tiene una formulación en negativo imposible en el formalismo de rasgos y unificación presentado hasta el momento en este trabajo. El concepto de precedencia tampoco se puede captar directamente en este formalismo. En definitiva, el problema consiste en que las relaciones de ligamiento, como el movimiento, se pueden establecer en dominios no locales no acotados. Por ello, los principios de ligamiento de Pollard y Sag (1994) y Sag y otros (2002), que quieren evitar dicho carácter no local, no resisten la formalización rigurosa necesaria para ser aplicados a gramáticas computacionales. En cualquier caso, ni SSG ni Matrix cuentan con una solución para este problema.

No obstante, durante el resto del trabajo, la lista ARG-ST —como se ve, en un principio vinculada al fenómeno del ligamiento— resultará una herramienta muy útil para el logro de un modelado elegante del sistema de reglas léxicas de SSG.

3.1.3. La noción de argumento: el tipo *argument*

SSG cuenta con un tipo que formaliza la noción de argumento sintáctico. Es el tipo *argument*, subtipo de *local*. Véase su estructura en la figura

³En Pollard y Sag (1994) sí se incluía un Principio C que regulaba explícitamente el comportamiento de los elementos nominales de tipo *npro*. Pero aquella formalización no resultaba adecuada, pues, necesariamente, hacía alusión a conceptos configuracionales.

3.7. De acuerdo con esta estructura, para que un signo pueda concebirse como argumento de otro, debe ser una proyección máxima⁴ (tener sus listas de subcategorización vacías) y no debe ser (o funcionar como) un elemento predicativo (PRD no). Por otro lado, se supone que todo argumento, en tanto que tal, ha de ser capaz de recibir caso, luego ha de llevar rasgo CASE.

$$\left[\begin{array}{l} \textit{argument} \\ \text{HEAD} \left[\begin{array}{l} \text{PRD} \quad \textit{no} \\ \text{CASE} \quad \textit{case} \end{array} \right] \\ \text{VAL} \left[\begin{array}{l} \text{SUBJ} \quad \langle \rangle \\ \text{COMPS} \langle \rangle \end{array} \right] \end{array} \right]$$
Figura 3.7: Tipo *argument*

Además, *argument* tiene dos subtipos: *inner* y *external*. Estos son la formalización, respectivamente, de las nociones de argumento interno y externo. La estructura de rasgos de *inner* no añade información a la que hereda de sus supertipos. En cambio, el tipo *external* cuenta con la caracterización extra que se observa en la figura 3.8.

$$\left[\begin{array}{l} \textit{external} \\ \text{HEAD|CASE} \quad \textit{nom} \\ \text{SEM|HOOK|INDEX} \quad \textit{index} \end{array} \right]$$
Figura 3.8: Tipo *external*

De acuerdo con la figura 3.8, en SSG, todo argumento externo recibirá caso nominativo (CASE *nom*) y tendrá naturaleza nominal: INDEX *index*⁵.

⁴En HPSG se contemplan relaciones argumentales donde el argumento no es propiamente una proyección máxima —Pollard y Sag (1994)—. Se trata de estructuras de ascenso y control en las cuales un verbo asume el sujeto, no saturado, de otro. No sería difícil modificar SSG para contemplar esta posibilidad.

⁵Dada esta caracterización del argumento externo, quizá haya datos del español que quedan fuera de SSG. Puede ser el caso de los verbos copulativos, o de identidad: *Que vengas es fantástico*, *Que vengas supondrá que nos lo pasaremos genial*. Pero la generalización

Por último, como ya se ha visto con *subir* en 2.1, hay verbos con complementos no argumentales: complementos predicativos. Estos complementos no denotan un individuo al que se asigna un rol temático. Son, en cambio, predicados que predicán sobre otro complemento del núcleo; este sí argumento, y menos oblicuo. Es el caso de *cantar en francés* en (2-a), *corriendo* en (2-b), *a la habitación* en (2-c), *dormido* en (2-d), etc.

- (2) a. Oigo a Teresa cantar en francés
 b. Juan viene corriendo
 c. Subió las cosas a la habitación
 d. Encontramos a Juan dormido

SSG cuenta con un tipo **predicative**. Su estructura se muestra en la figura 3.9. En ella se ve que un complemento predicativo debe ser, igual que los argumentos, una proyección máxima. Pero, a diferencia de aquellos, estos son predicativos: PRD yes.

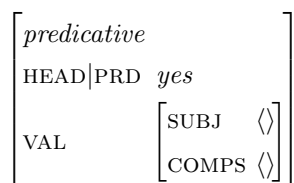


Figura 3.9: Tipo **predicative**

SSG hará un uso crucial del tipo **argument**, y en la medida en la que sea necesario, también de **predicative**. Téngase en cuenta que un argumento pudiera ser un signo de muy diversa naturaleza: un sintagma nominal, o un complementante, o una oración en infinitivo, o una oración interrogativa; o cualquiera de esas posibilidades marcada con una preposición *a* de acusativo, o con una preposición *en, con, de...* de caso oblicuo, etc. Es frecuente que, en las gramáticas basadas en **Matrix**, cada una de estas posibilidades de realización del argumento de un verbo se formalice, explícitamente, en un tipo es útil para la inmensa mayoría de los casos. Parece, entonces, necesario, complicar la jerarquía con un tipo intermedio external-min para el argumento externo, que no especifique su naturaleza semántica, pero sí su caso. El tipo external sería subtipo de aquel.

verbal distinto⁶.

En SSG, por elegancia y a fin de reducir la explosión combinatoria, se pretende evitar una formalización explícita de ese estilo. En cambio, solo el tipo **argument** y sus subtipos establecen, de forma general, cuál puede ser la naturaleza categorial de cualquier signo que se entienda como argumento. En principio, cualquier signo que unifique con ellos será válido.

Concretamente, la caracterización del tipo **argument** de la figura 3.7 unifica en SSG con los determinantes, nombres, preposiciones de marca de caso, complementantes y partículas interrogativas capaces de encabezar una oración. Por tanto, a falta de otra restricción, cualquiera de estas categorías puede colocarse en una posición argumental. Así pues, en SSG, para dar cuenta de los diferentes usos del verbo *saber* que se muestran en (3), basta con caracterizar al argumento interno de este verbo como **argument** (como **inner**, en realidad).

- (3)
- a. Sé precios
 - b. Sé los precios de la habitación
 - c. Sé que han subido los precios de la habitación
 - d. Sé cuánto han subido los precios de la habitación
 - e. Sé subir los precios de la habitación

En ocasiones (por ejemplo, en **external**) se pedirá a un argumento que tenga naturaleza semántica nominal, mediante la especificación de **INDEX** como **index**. Esto impedirá que los complementantes ocupen la posición argumental en cuestión. La especificación del caso también establecerá, de forma indirecta, restricciones categoriales a lo que puede ser un argumento. La jerarquía de subtipos de **case** en SSG se representa en la figura 3.10. Los casos **nom** y **acc** se han declarado como subtipos de **str** (estructural).

⁶Recuérdese **transitive-lex-item** para verbos con complementos nominales, frente a **clausal-second-arg-trans-lex-item** para verbos con complementos verbales.

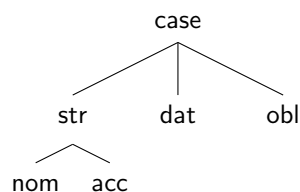


Figura 3.10: Jerarquía de casos (case)

Se asume, entonces, la división tradicional en la Gramática Transformacional entre casos estructurales (asignados por la ubicación del argumento en una posición de la estructura sintáctica) e inherentes (asignados directamente por una preposición encargada de marcar caso). No se asume aquí que sea una posición estructural concreta (complemento del verbo, o especificador de una categoría funcional determinada) la que asigna caso estructural. Pero sí se entiende que los casos nominativo y acusativo tienen una cierta naturaleza común: se asignan de forma regular a los argumentos menos oblicuos de los verbos. Y, como se verá en adelante, **str** resultará un tipo natural para especificar el caso, necesariamente estructural pero variable, de ciertos argumentos internos: acusativos en voz activa, pero nominativos en pasiva. La cuestión es que los determinantes, nombres y complementantes tendrán en SSG en el léxico caso **str**. Y las preposiciones de marca de caso tendrán el caso **acc**, **dat** u **obl** correspondiente. Por tanto, en SSG se entiende que un determinante, nombre o complementante tiene por defecto caso **str**. Para que se le asigne caso inherente deberá ser tomado como complemento de una preposición que sí tenga tal caso⁷.

⁷No se ha conseguido en este trabajo restringir adecuadamente las posibilidades de realización de los argumentos con caso acusativo. Un argumento de acusativo va marcado con preposición *a* cuando es definido y animado. Y no lleva marca preposicional en cualquier otro caso. Véase (I).

- (I)
- a. Veo *(a) Juan
 - b. Veo (*a) los barcos
 - c. Veo (*a) niños
 - d. Veo (*a) barcos

Parece difícil dar cuenta de estos datos sin crear reglas léxicas *ad hoc*. La solución real

Como cabe imaginar, los complementos de las preposiciones de marca de caso se han caracterizado también como *argument*. De acuerdo con este sistema, en SSG, al argumento interno de un verbo como *saber* se le asigna caso estructural: *CASE str*. Gracias a ello, este argumento puede adoptar las formas de (3), pero no la de preposición de caso oblicuo, por ejemplo. En cambio, ese mismo argumento interno, en el verbo *acordarse*, recibe caso oblicuo (*obl*). Esta especificación obliga a que el argumento interno de este verbo se satisfaga con una preposición de marca de oblicuo. Ahora bien, el complemento de esta preposición sí puede tener la naturaleza categorial de un nombre, un sintagma nominal, etc. Todo ello da cuenta de los datos de (4).

- (4)
- a. No me acuerdo de precios
 - b. No me acuerdo del precio de la habitación
 - c. No me acuerdo de que haya subido el precio de la habitación
 - d. No me acuerdo de cuánto ha subido el precio de la habitación
 - e. No me acuerdo de subir los precios de la habitación

Lo interesante de este sistema es que permite restringir la naturaleza categorial de un determinado argumento directamente en la caracterización léxica de su núcleo. Para dar cuenta de las distintas alternativas de (3) y (4), no es necesario el uso de reglas léxicas diferentes. En cambio, *Matrix* sugiere el uso de al menos dos patrones distintos para dar cuenta de estas oraciones: uno para los argumentos nominales y otro para los verbales.

Es comúnmente aceptado que solo los elementos nominales reciben marca de caso⁸. En cambio, como se ha visto, SSG concibe también a los complementantes como categorías susceptibles de recibir tal marca. No es posible

utilizada en este trabajo es externa a la gramática: se utiliza para ello uno de los módulos que procesa la secuencia de entrada antes de que se le pase al analizador sintáctico.

⁸En el modelo de Principios y Parámetros se asume que el caso se asigna a elementos nominales: Eguren y Fernández Soriano (2004) pp. 131-132. En HPSG, generalmente, se asume esta idea.

hacer aquí una defensa teórica de esta propuesta. Se ha asumido porque es necesaria para lograr un sistema de asignación de caso coherente y sencillo. Gracias a ello, SSG puede explicar el contraste entre (3-c) y (4-c) del mismo modo que se explica el de (3-b) y (4-b): como un proceso de asignación de caso diferente.

Además, es necesario que la gramática contemple como posibles argumentos oraciones con partícula interrogativa —(3-d) y (4-d)—, y predicados verbales en infinitivo —(3-e) y (4-e)—. No se han de contemplar como argumento, en cambio, las oraciones finitas sin partícula interrogativa. En un modelo gramatical con categorías funcionales, como el chomskyano, es fácil modelar esta asimetría. Las partículas interrogativas, en el modelo chomskyano, se suelen colocar en el especificador de una categoría funcional C (complementante). Por tanto, la presencia de estas implica que la construcción en cuestión, categorialmente, es un complementante C, con núcleo silente. Por su parte, los complementantes propiamente dichos (*que* o *si*, en español) son núcleos explícitos de esa categoría C. La oración finita suele entenderse como una categoría funcional portadora de la flexión F. Los predicados verbales no finitos, en infinitivo, se pudieran concebir, en principio, como núcleos verbales léxicos —V—, no como categorías funcionales. Estructuralmente, C es superior a F, y esta es superior a V⁹. Dada esta formalización, es posible caracterizar a C y a V como compatibles con el tipo *argument*, y a F como incompatible con él.

Pero, en HPSG el establecimiento de esta asimetría no es tan sencillo, pues este modelo concibe las tres construcciones como elementos categorialmente idénticos. En HPSG no existen categorías funcionales. En ella, los únicos núcleos existentes son los léxicos (en este caso, los verbos). Por ello, tanto las oraciones finitas, como las oraciones con partícula interrogativa y los predicados verbales en infinitivo son, categorialmente, verbos. Pueden idearse mecanismos *ad hoc* para restringir la aparición de estas construc-

⁹Véase Eguren y Fernández Soriano (2004) pp. 152-161.

ciones en la posición de objeto¹⁰. Pero la solución adecuada debe engarzarse en una teoría amplia que recoja en HPSG lo que otros modelos logran, de forma satisfactoria, haciendo uso crucial de las categorías funcionales.

3.1.4. Relación entre ARG-ST y VAL

Perlmutter (1978) propuso por primera vez que los verbos intransitivos (aquellos que no asignan caso acusativo) se pueden clasificar en dos grupos: pueden ser verbos inergativos o inacusativos. La diferencia entre ellos consiste en que el sujeto de un verbo inergativo es un argumento externo, mientras que el de un verbo inacusativo es un argumento interno. Esto es relevante, pues, si un sujeto es argumento externo, será un sujeto genuino; mientras que, si es interno, tendrá, aun con caso nominativo, la naturaleza de objeto, y un comportamiento propio de objeto. Por ejemplo, es posible —véase (5)— que el sujeto de un verbo inacusativo sea un nombre posverbal sin determinante; igual que pudiera haberlo sido el objeto de un verbo transitivo¹¹.

- (5) a. Siempre vienen mujeres

¹⁰SSG, en su actual estado de desarrollo, no contempla las partículas interrogativas como complementos posibles. Además, el problema de los infinitivos no se plantea en la versión actual de SSG. Esto se debe a que SSG recibe al infinitivo y al verbo del que es argumento como un único elemento atómico para el procesamiento sintáctico. Un módulo de preprocesamiento se encarga de presentarlos de tal modo al *parser* sintáctico

¹¹Se ha seguido a Mendikoetxea (1999), también en los ejemplos. Esta autora recoge además los siguientes contrastes entre verbos inergativos e inacusativos. Por un lado, los participios de los verbos inacusativos pueden aparecer en cláusulas de participio absoluto (no así los de los verbos inergativos).

- (I) a. Agotado el ozono en la atmósfera, el fin de la vida en la tierra es inminente
b. *Trabajado Juan, se dirigió a su casa

Por otro lado, los participios de verbos inacusativos pueden modificar a un sintagma nominal que sería su sujeto sintáctico. No ocurre lo mismo con los de verbos inergativos.

- (II) a. Un tesoro recientemente aparecido
b. *Un perro muy ladrado

- b. Todos los años llegan cigüeñas
- c. Existen problemas

En cambio, —(6)— no ocurre otro tanto con los inergativos¹².

- (6) a. *Duermen mujeres
- b. *Juegan niños
- c. *Trabajan jóvenes

Además, siguiendo a Contreras (1978), también en el ejemplo, para ciertos verbos con sujeto argumento interno, la posición canónica de este es siempre posverbal: véase el sujeto del verbo *estallar* en (7).

- (7) Ha estallado la revolución

Apoyada en estos datos, SSG entiende que, en español, la realización canónica de un argumento interno (sea este nominativo o acusativo) es posverbal. Y solo el argumento externo se realizará de forma canónica como preverbal. Por tanto, la anteposición al verbo de un argumento interno es un orden de palabras derivado, una relación de dependencia no acotada, también si ese argumento es nominativo. Para formalizar esta generalización, se han creado en SSG dos tipos: *unaccusative* y *unergative*. Estos tipos establecen distintos patrones de relación entre las listas ARG-ST, SUBJ y COMPS.

$$\left[\begin{array}{l} \textit{unergative} \\ \text{SYNSEM|LOCAL|VAL} \left[\begin{array}{l} \text{SUBJ} \langle \text{1} \rangle \\ \text{COMPS} \text{ 2} \end{array} \right] \\ \text{ARG-ST} \langle \text{1} \left[\text{LOCAL} \textit{external} \right]. \text{2} \rangle \end{array} \right]$$

Figura 3.11: Relación entre ARG-ST y VAL con argumento externo

¹²Sin embargo, existen ciertos contextos sintácticos que habilitan a tales verbos para participar de esta construcción —Torrego (1989)—: la compañía de un localizador (**Corren niños* vs. *En este parque corren niños*), la pregunta, la exclamación... (*¿Corren niños?*, *¡Mira, corren niños!*).

El tipo *unergative* representa a los núcleos que tienen argumento externo: el caso prototípico serán los verbos inergativos, pero también participarán de este patrón los verbos transitivos —figura 3.11—. Estas unidades léxicas colocarán su sujeto en *SUBJ* y el resto de complementos en *COMPS*. En cambio, para los núcleos que carezcan de argumento externo —los verbos inacusativos—, el patrón utilizado será el que se ha formalizado mediante el tipo *unaccusative*: colocan todos sus argumentos, incluido el sujeto, en *COMPS*, y dejan *SUBJ* vacío¹³ —véase 3.12—.

$$\left[\begin{array}{l} \textit{unaccusative} \\ \text{SYNSEM|LOCAL|VAL} \left[\begin{array}{l} \text{SUBJ} \langle \rangle \\ \text{COMPS} \boxed{1} \end{array} \right] \\ \text{ARG-ST} \quad \boxed{1} \langle [\text{LOCAL } \textit{inner}], \dots \rangle \end{array} \right]$$

Figura 3.12: Relación entre ARG-ST y VAL sin argumento externo

Según se ha definido SSG, el elemento colocado en *SUBJ* se realizará a la izquierda del núcleo, y los colocados en *COMPS*, a su derecha. Por tanto, los tipos *unergative* y *unaccusative* de SSG establecen que la realización canónica de un argumento externo (el sujeto de un verbo inergativo o transitivo) es preverbal, mientras que la realización canónica de un argumento interno (el sujeto de un verbo inacusativo, o el objeto de uno transitivo) es posverbal.

¹³Pudiera contemplarse la posibilidad de colocar un expletivo en las listas *SUBJ* aparentemente vacías. En español, este expletivo no tendría realización fonética. De este modo, la lista *SUBJ* siempre contaría con un elemento, lo cual es interesante. Por ejemplo, el análisis habitual en HPSG de las estructuras de ascenso o control confía en la presencia de un elemento en *SUBJ*. Un verbo auxiliar, por ejemplo, suele analizarse como una estructura de ascenso: toma como complemento un sintagma verbal cuyo elemento en *SUBJ* no ha sido saturado, y asume que el índice de su sujeto es el índice del sujeto de su complemento. Si un auxiliar *A* tomara como complemento un verbo inacusativo *V* con una lista *SUBJ* vacía, no encontraría sujeto. En cambio, si se hubiese colocado un expletivo en la lista *SUBJ* de *V*, *A* tomaría como sujeto en *SUBJ* al expletivo, igual que pudiera haber tomado un índice referencial si *V* hubiese sido inergativo. Este análisis también permitiría —como se propone en *Matrix*— colocar el cierre semántico oracional en la regla de saturación de *SUBJ*. En cambio, sin uso del expletivo, las oraciones con verbos inergativos no necesitan tal regla, y no recibirían cierre semántico.

De este modo se da una representación natural a los datos de (5), (6) y (7).

Esta propuesta es una revisión de lo visto en el apartado 3.1.2, donde solo se contemplaba un patrón de relación entre las listas en cuestión: el equivalente a *unergative*. A lo largo del trabajo —en el apartado 5.1—, se revisará de nuevo esta propuesta, a la luz de nuevos datos. En general, se remite al capítulo 5 para una justificación y explicación por extenso de todas las cuestiones relativas al orden de palabras.

3.2. Tipos verbales

Levin y Rappaport (1995) y Levin y Rappaport (1998) establecen que un verbo puede denotar una actividad, un logro, una realización o un estado. Esta es una clasificación semántica resultante del cruce de dos criterios: los verbos pueden denotar, o no, un acción que se lleva a cabo; y pueden denotar, o no, un cambio de estado que se sufre. Estos dos criterios se han formalizado en SSG mediante los rasgos semánticos (declarados para el tipo *sort*) ACT y BECOME, de valor booleano (*yes* o *no*). Véase la figura 3.13.

$$\begin{bmatrix} \textit{sort} & \\ \text{ACT} & \textit{bool} \\ \text{BECOME} & \textit{bool} \end{bmatrix}$$

Figura 3.13: ACT y BECOME

Una actividad es un evento en el cual un participante x está desarrollando una acción, está haciendo algo. Es el caso de *hablar*, *estudiar*, *nadar*, *pasear*, *caminar*, *bailar*, *correr* o *esquiar*, por ejemplo. Estos verbos cuentan, al menos, con un argumento: aquel que lleva a cabo la acción. Se trata de un argumento externo. Es el caso de *los turistas* en (8-a), o el pronombre omitido de primera persona del singular en (8-b).

- (8) a. ¿Pueden esquiar los turistas en primavera?

- b. ¿Puedo hablar en inglés con el personal del hotel?

En SSG se recoge esta clase de verbos en el tipo **activity**, que se muestra en la figura 3.14.

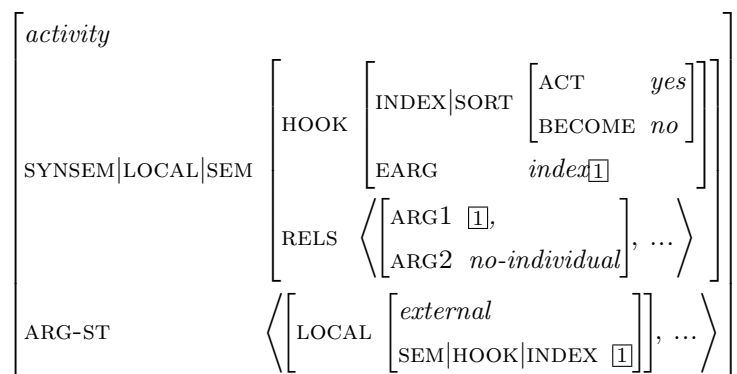


Figura 3.14: Actividades

Los verbos de clase **activity**, como denotan el desarrollo de una actividad, tienen un valor positivo para **ACT**; pero, como no denotan cambio de estado alguno, tienen valor negativo para **BECOME**. Además, dado que estos verbos cuentan con un argumento externo, se establece en su lista **ARG-ST** un primer elemento de tipo **external**, cuyo índice semántico se publica en el rasgo **EARG**. Además, este índice también se identifica con **ARG1**, con lo cual se da a entender que denota un agente¹⁴. Por último, aunque estos verbos pueden contar también con un argumento interno, este no será nunca un paciente afectado: por ello, se especifica **ARG2 no-individual**.

Por su parte, un logro no denota una acción que lleve a cabo nadie, sino un cambio de estado que sufre un participante x . Ejemplos de estos verbos son *floreecer*, *crecer*, *estallar*, *palidecer*, *arder* o *enrojecer*. También obedecen a este patrón los usos intransitivos de *empezar* o *subir*. Son lo que se ha llamado verbos de cambio de estado de causa interna. Es decir, la causa del cambio de estado que denotan no está en un agente externo que lo propicie, sino en la misma naturaleza del participante afectado por el cambio. Por ejemplo, para

¹⁴Recuérdese la explicación de 3.1.1.

empezar, la razón de que *x* empiece no está en que se contemple la existencia de un agente *y* que haga empezar a *x*. La causa está en la misma naturaleza de *x*. Estos verbos cuentan también, en principio, con un único argumento: el que denota aquello que sufre el cambio de estado. En (9-a) es *el espectáculo* el argumento en cuestión, y en (9-b), *el precio de las habitaciones*.

- (9) a. Ha empezado el espectáculo
- b. Ha subido el precio de las habitaciones

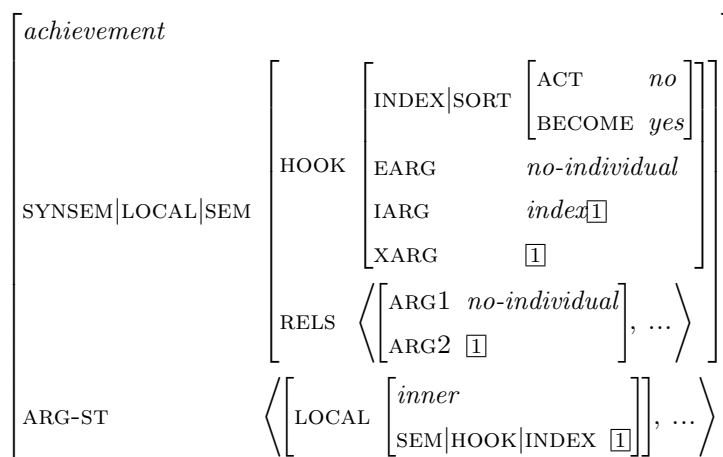


Figura 3.15: Logros

El tipo de la gramática que recoge esta clase verbal es *achievement*, que se muestra en la figura 3.15. En tanto que cambio de estado, su valor de BECOME es *yes*; y, en tanto que no denota una acción, su valor de ACT es *no*. Su argumento es interno, y no tiene argumento externo¹⁵. Por tanto,

¹⁵En cuanto a esta identidad rigurosa entre paciente afectado y argumento interno véase la *Nueva gramática de la lengua española*, II, p. 3056: “Se ha debatido largamente la cuestión de si la distinción entre verbos inacusativos e inergativos tiene correspondencia directa con las clases semánticas de verbos. Esta es una cuestión importante que no ha recibido todavía una respuesta satisfactoria. [...] Un buen número de verbos inacusativos denotan presencia, existencia, aparición, surgimiento o manifestación de algo [...]. Otro grupo numeroso de verbos inacusativos expresan cambios de estado”. En SSG no se entra a esta compleja discusión y se asume que estas son clases semánticas correspondientes al patrón inacusativo.

achievement cuenta con un elemento inner en ARG-ST, cuyo índice se publica en IARG. Como el argumento en cuestión es un paciente afectado, se identifica con la posición argumental ARG2. La falta de argumento externo se codifica evaluando EARG como *no-individual*. También se puede establecer la falta de un agente evaluando ARG1 como *no-individual*.

Por último, el argumento interno también se puede identificar con el rasgo XARG. Este rasgo denota el argumento que, en un uso concreto del verbo, es el más prominente, el menos oblicuo de entre los que tienen realización sintáctica. Una actividad podría tener un uso activo y otro pasivo. En el primer caso, XARG sería el argumento externo; y, en el segundo, el interno. Pero para un verbo intransitivo, como es todo logro, esa alternancia no es posible. Por ello, en cualquier caso, en *achievement*, XARG ha de estar identificado con su argumento interno.

Las realizaciones son verbos que también denotan el cambio de estado de un participante *x*. Pero, en este caso, la causa del cambio es externa a *x*. Es un agente *y* el que, desarrollando una actividad, propicia el cambio en *x*. Estos verbos, por tanto, denotan dos subeventos: un cambio de estado y una acción que propicia ese cambio. Verbos de este tipo son *romper*, *borrar*, *descolgar*, *estropear*, *fastidiar*, *hundir*, *dividir*, *emocionar*, *alegrar*, *contratar* o los usos transitivos de *subir*. Son verbos que presentan, al menos, dos argumentos: aquel que denota al agente causante del cambio, y el que denota al participante que sufre el cambio. El primero será un argumento externo y el segundo un argumento interno. En (10-a), *El botones* es el argumento que, llevando a cabo una acción, propicia que *el equipaje* (el segundo argumento) cambie de estado: quede en la habitación. Otro tanto ocurre con *El hotel* y *el guía* en (10-b).

- (10) a. El botones ha subido el equipaje a la habitación
 b. El hotel ha contratado al guía para la excursión

Las realizaciones quedan reflejadas en la gramática mediante el tipo accom-

plishment, mostrado en la figura 3.16.

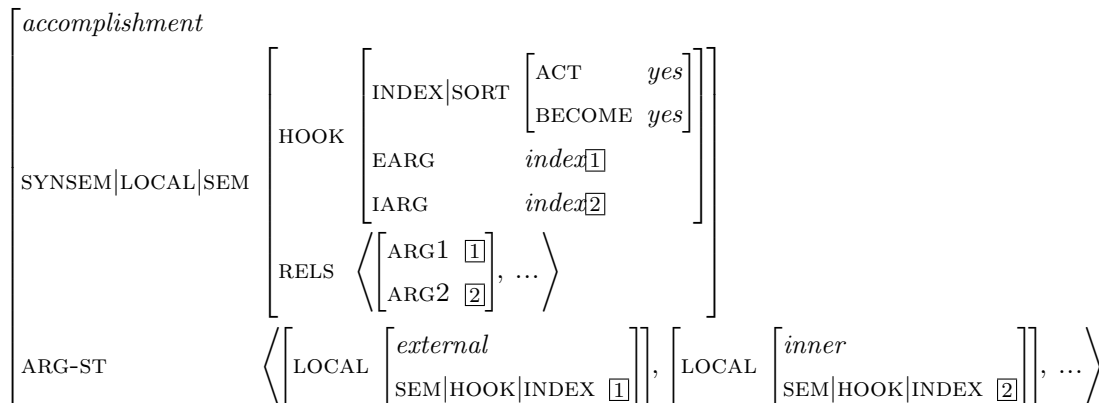


Figura 3.16: Realizaciones

Estos verbos, al denotar tanto acción como cambio de estado, presentan valor positivo para ACT y BECOME. El argumento que aporta el primer subevento (el agente del cambio) es externo. Por ello, se establece que el verbo cuenta con un argumento externo (EARG *index*), y se identifica a este como agente (estructura compartida con ARG1). En cuanto al segundo argumento (el paciente), es interno y paciente. Por ello, se establece IARG *index*, y se identifica ese índice con ARG2. Ambos argumentos semánticos pueden indentificarse, a su vez, con los índices de los correspondientes argumentos sintácticos (adecuadamente caracterizados como *external* e *inner*) que ocupan las dos primeras posiciones de ARG-ST. También en este caso, la posibilidad de que el verbo participe de un patrón activo o pasivo hace imposible especificar aquí el valor de XARG.

Por último, los estados son verbos que no denotan ni acciones ni cambios de estado. Son, sencillamente, verbos que denotan que un participante x se encuentra en un determinado estado. Dicho participante no lleva a cabo ninguna acción, ni cambia en ningún sentido. Ejemplos de esta clase de verbos son *existir*, *ocurrir*, *ver*, *tener*, *necesitar* o *costar*. En (11-a), se da a entender que el participante *el hotel* se encuentra en un estado de posesión de una sala de reuniones. En (11-b), *la entrada* se encuentra adscrita a un precio

concreto. Esta clase de verbos piden, en principio, un argumento: aquel que se encuentra en el estado en cuestión. Sería el caso de *el hotel* en (11-a) y *La entrada* en (11-b).

- (11) a. ¿Tiene el hotel sala de reuniones?
 b. La entrada cuesta 13 euros

El tipo que recoge a esta clase en la gramática es **state** —figura 3.17—.

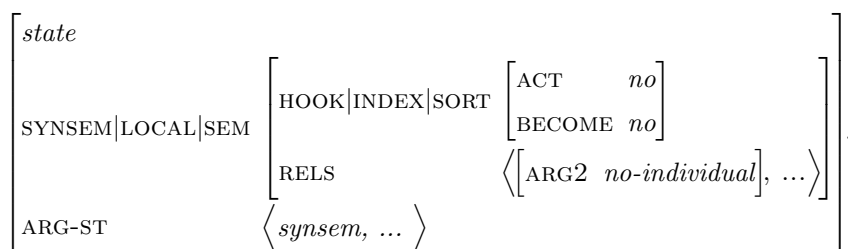


Figura 3.17: Estados

Al no denotar estos verbos ni acción ni cambio de estado tienen valor negativo para ACT y BECOME. Su único argumento parece en unos casos interno y en otros externo, con lo cual no se especifica su naturaleza en este tipo. La única caracterización propia de todos sus usos es que no denotará un paciente afectado (en tanto que BECOME no).

En resumen, en la gramática se contemplan cuatro grandes clases de tipos verbales, en función de su naturaleza semántica: actividades, logros, realizaciones y estados. Estas clases se han implementado, respectivamente, mediante los tipos **activity**, **achievement**, **accomplishment** y **state**. La naturaleza semántica de cada uno de estos tipos se caracteriza con los rasgos ACT y BECOME, de valores booleanos. Los verbos que denotan una acción (actividades y realizaciones) son ACT **yes**. Los verbos que denotan un cambio de estado (logros y realizaciones) son BECOME **yes**. Estos rasgos semánticos de los verbos condicionan su naturaleza sintáctica. Un verbo que denota una acción cuenta con un argumento externo en ARG-ST: el agente de esa acción. Por su parte, un verbo que denota un estado (o un cambio de estado) cuenta también con

un argumento en ARG-ST: aquel que se encuentra en el estado en cuestión, o aquel que deviene a tal estado. Por tanto, las actividades cuentan, en principio, con un único argumento: el agente de la acción. Los logros y los estados cuentan, también en principio, con un solo argumento: el participante que se encuentra en el estado en cuestión o que deviene a tal estado. Y las realizaciones cuentan, al menos, con dos argumentos: el agente que propicia el cambio de estado, y el paciente que deviene a tal estado. Levin y Rappaport (1998) llaman a estos argumentos legitimados por este significado regular del verbo argumentos estructurales. Obsérvese que todos los argumentos estructurales se han caracterizado con naturaleza nominal (INDEX index).

Además de estos argumentos estructurales, un verbo puede contar con otros argumentos legitimados por su significado idiosincrásico. Dos verbos pueden contar con un mismo esquema de significado básico (el de actividad, logro, realización o estado). Pero además, cada verbo tiene un cierto significado idiosincrásico que lo diferencia del resto de verbos de su clase. Este significado puede pedir ciertos argumentos extra. Por ejemplo, es común que las actividades y los estados pidan un tema que complete su significado. Es lo que ocurre con el estado *costar* en (11-b): cuenta con un argumento (*13 euros*) que especifica cuánto cuesta. Otro tanto ocurre con *tener* en (11-a): presenta un argumento (*sala de reuniones*) que establece lo que se tiene.

Por último, entiéndase que todos estos tipos están ordenados en SSG conforme a la siguiente jerarquía de la figura 3.18. Se entiende que *v-semantic-c* aglutina todos los tipos verbales caracterizados en cuanto a patrón semántico.

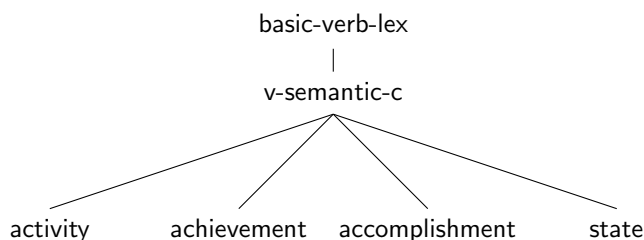


Figura 3.18: Tipos verbales según Levin y Rappaport (1998)

3.3. Argumentos estructurales

De acuerdo con lo visto en 3.2, los verbos, dada su naturaleza semántica, contarán con una serie determinada de argumentos estructurales en ARG-ST. Por ejemplo, un verbo que desde el punto de vista de su significado sea una realización, como *romper* o *alegrar*, se implementará en la gramática como instancia del tipo *accomplishment*, lo cual le aportará dos argumentos en ARG-ST, adscritos a ciertos valores temáticos: el primero será un agente y el segundo un paciente.

Una vez un verbo tiene una lista de argumentos, es necesario marcar a cada uno de ellos con un caso. El marcado de caso, como es comúnmente aceptado en HPSG¹⁶, es una caracterización léxica: el verbo establece en el léxico el caso de cada argumento. En SSG, el marcado de caso corre a cargo de ciertas reglas léxicas independientes de los tipos que aportan argumentos a los verbos. Las unidades léxicas serán instancias de tipos como *activity*, *accomplishment*, *achievement* o *state*, y no especificarán el caso de sus argumentos. El caso de estos se especificará posteriormente mediante la aplicación de reglas léxicas como *v-transitive-c* o *v-passive-c*. De este modo, SSG recoge la intuición de que a un mismo argumento se le pueden asignar distintos casos; y de que un mismo caso puede asignarse a distintos tipos de argumento: la naturaleza de un argumento (su rol temático, su carácter interno o externo, etc.) es independiente de la marca de caso. Concretamente, en *v-transitive-c* se implementa en la gramática el modelo de asignación de caso propio de los verbos transitivos en voz activa: el primer argumento en ARG-ST recibe caso nominativo y el segundo acusativo. En cambio, en *v-passive-c* se recoge el modelo de asignación de caso propio de los verbos transitivos en voz pasiva: el primer argumento en ARG-ST no se realiza y al segundo se le asigna nominativo. En definitiva, la gramática se ha implementado de tal forma que activa, pasiva y otras construcciones se conciben como patrones de asignación de caso alternativos aplicables a unidades léxicas subespecificadas en cuanto a la asignación de caso a sus argumentos.

¹⁶Sag y otros (2002).

Como se ve, este planteamiento se aleja de la idea tradicional, presente en las versiones clásicas de HPSG, de que la activa es la construcción canónica de un verbo transitivo, y la pasiva se construye a partir de la primera. En SSG, ambas son patrones de comportamiento en pie de igualdad, aplicables a unidades léxicas subespecificadas compatibles con ambos. Este planteamiento permitirá una implementación mucho más sencilla y elegante de lo que resulta habitual en las gramáticas computacionales de tipo HPSG.

Todos los tipos de la gramática encargados de modelar los patrones de asignación de caso —*v-transitive-c*, *v-passive-c*, etc.— heredan del tipo de la figura 3.19.

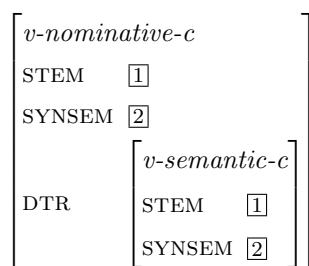


Figura 3.19: Patrones de asignación de caso (I)

El tipo *v-nominative-c* es una regla léxica que toma como *input* un tipo *v-semantic-c*, el supertipo general del cual heredan *activity*, *accomplishment*, *achievement* y *state*. Además, la estructura sintáctica y semántica del *input* y el *output* de la regla se identifican, mediante identidad del valor de sus rasgos SYNSEM. Por tanto, *v-nominative-c* es una regla que toma una unidad léxica —instancia de *v-semantic-c*—, asume su estructura, y le añade alguna caracterización compatible con esta: el caso de sus argumentos.

Los diferentes subtipos de *v-nominative-c* asumen, entonces, la estructura de los verbos sobre los que operan y les añaden el caso. Estos subtipos son, concretamente, *v-transitive-c*, *v-passive-c*, *v-rpass-c*, *v-unergative-c*, *v-unaccusative-c* y *v-pronominal-c*. La aplicación de una u otra de estas reglas

sobre un verbo como *contratar* (instancia léxica de *accomplishment*) da lugar a los patrones de comportamiento de (12).

- (12) a. El hotel ha contratado al guía para la excursión
 b. Ha sido contratado un guía para la excursión
 c. Se han contratado guías para la excursión

La versión transitiva activa de *contratar* de (12-a) está legitimada por la aplicación sobre el verbo de la regla *v-transitive-c*. La versión pasiva —(12-b)—, en cambio, obedece a la aplicación de *v-passive-c*; mientras que la construcción de pasiva refleja de (12-c) es el resultado de aplicar *v-rpass-c*. Las construcciones intransitivas (*v-unergative-c*, *v-unaccusative-c* y *v-pronominal-c*) no se pueden aplicar sobre *contratar*, pues este, inherentemente transitivo en tanto que instancia de *accomplishment*, es un *input* inaceptable para ellas.

$$\left[\begin{array}{l} \textit{unergative} \\ \text{SYNSEM|LOCAL|VAL} \left[\begin{array}{l} \text{SUBJ} \langle \underline{1} \rangle \\ \text{COMPS} \underline{2} \end{array} \right] \\ \text{ARG-ST} \langle \underline{1} \left[\text{LOCAL } \textit{external} \right]. \underline{2} \rangle \end{array} \right]$$

$$\left[\begin{array}{l} \textit{unaccusative} \\ \text{SYNSEM|LOCAL|VAL} \left[\begin{array}{l} \text{SUBJ} \langle \rangle \\ \text{COMPS} \underline{1} \end{array} \right] \\ \text{ARG-ST} \underline{1} \langle \left[\text{LOCAL } \textit{inner} \right], \dots \rangle \end{array} \right]$$

Figura 3.20: Patrones *unergative* y *unaccusative*

En SSG se ha entendido que estos patrones de asignación de caso se cruzan con los patrones de relación entre ARG-ST y las listas de VAL: SUBJ y COMPS. Como se verá, ambas son cuestiones relacionadas. En 3.1.4, se mostraron los patrones de relación entre listas *unergative* y *unaccusative*, cuyas estructuras de rasgos se repiten en la figura 3.20. Aquí se propone añadir a estos, en principio, un tercer patrón: *passive* —véase en la figura 3.21—. Este patrón es propio de verbos transitivos: su ARG-ST cuenta con un primer argumento externo seguido de un argumento interno. Sobre esta lista ARG-ST, *passive*

obvia el primer argumento, y opera con el resto de argumentos como si de un verbo inacusativo se tratase: deja SUBJ vacío e identifica COMPS con la lista completa, excluido su primer elemento.

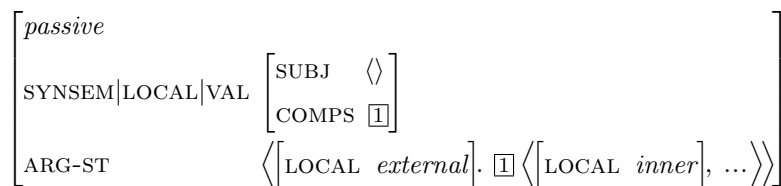


Figura 3.21: Relación pasiva entre ARG-ST y VAL

Los tipos *unergative*, *unaccusative* y *passive* son subtipos de *v-nominative-c* y, por tanto, reglas léxicas. Además, los patrones de asignación de caso (*v-transitive-c*, *v-passive-c*...) son subtipos de estos patrones de relación entre las listas. Todos ellos se organizan conforme a la siguiente jerarquía de tipos de la figura 3.22.

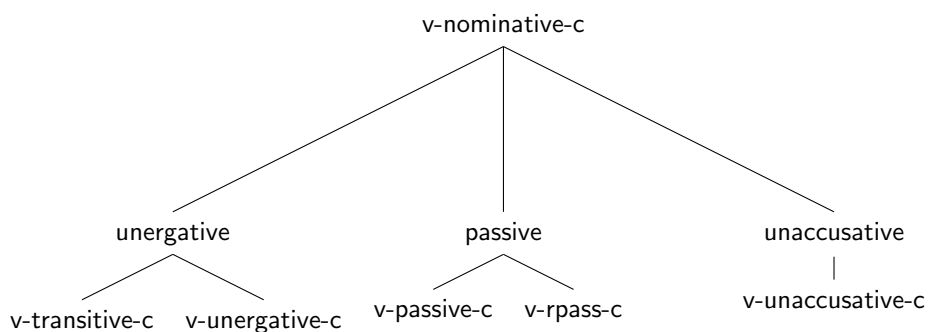


Figura 3.22: Jerarquía de *v-nominative-c*

Es una versión provisional, a la que habrá que añadir el tipo propio de los verbos pronominales *v-pronominal-c*.

Solo queda mostrar cómo los patrones de asignación de caso (*v-transitive-c*, *v-passive-c*...) especifican el valor del rasgo CASE de sus argumentos, en ARG-ST. Dada la jerarquía de la figura 3.22, esos argumentos se distribuirán convenientemente en las listas de VAL; pero, en adelante, se obviará esa cuestión,

ya explicada.

Perlmutter (1978) ya observó cómo la existencia de verbos inacusativos, y pasivos, hace pensar que la asignación de caso nominativo es prioritaria sobre la del caso acusativo. En principio, los verbos asignan caso acusativo a su argumento interno. Es lo que ocurre con los verbos transitivos. Los verbos inergativos no asignan acusativo porque no tienen argumento interno. Cabría pensar que los verbos inacusativos, al tener argumento interno, le asignarían a este caso acusativo. Sin embargo, este argumento recibe caso nominativo. Del mismo modo, las construcciones pasivas eliminan —de acuerdo con *passive*— su argumento externo, el que en condiciones normales recibe nominativo. Y a su argumento interno —que, en principio, debiera recibir acusativo— se le asigna nominativo.

Parece, entonces, que en las lenguas del mundo hay una fuerte tendencia a asignar caso nominativo al argumento más externo de un verbo. Cuando un verbo cuenta con argumento externo propiamente dicho, será este el que reciba caso nominativo, y el verbo podrá asignar acusativo al interno, si lo hubiera. Pero si un verbo no tiene argumento externo, o este se ha eliminado, su argumento más externo es el interno. En tal caso, a este se asigna caso nominativo, con lo cual no se le puede ya asignar acusativo. En SSG se capta esta generalización asignando caso nominativo al primer argumento de todo verbo en ARG-ST.

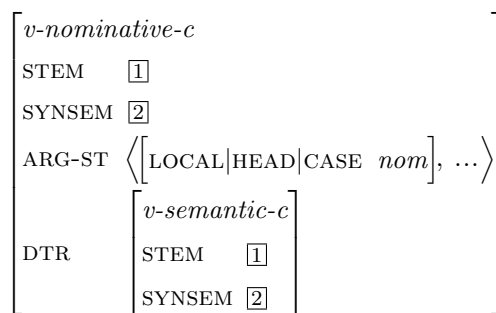


Figura 3.23: Patrones de asignación de caso (II)

El supertipo general de todos los patrones de asignación de caso —*v-nominative-c*— debe especificar, por tanto, que el primer argumento de su lista ARG-ST tiene caso nominativo. Este argumento tendrá naturaleza distinta en los diferentes subtipos de *v-nominative-c*: será argumento externo en *v-transitive-c* y *v-unergative-c*, pero interno en *v-unaccusative-c*. Para captar esta generalización, la estructura definitiva de *v-nominative-c* es la que aparece en la figura 3.23.

En las siguientes figuras aparecen los tipos caracterizadores de los distintos patrones concretos de asignación de caso. En ellas solo se muestra la información declarada propiamente en dichos tipos. Habrá que suponerles a estos tipos, además, la información de sus supertipos: de acuerdo con *v-nominative-c*, el primer argumento en ARG-ST siempre tendrá caso nominativo; y los argumentos de esta lista estarán repartidos en las listas de VAL conforme al supertipo pertinente *unergative*, *unaccusative* o *passive*.

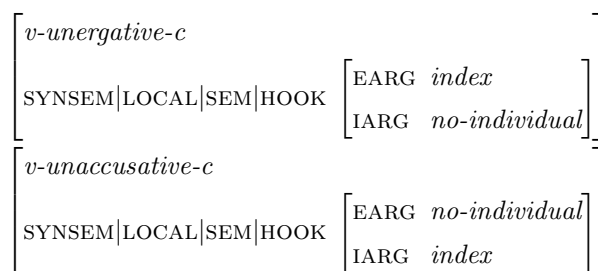


Figura 3.24: Patrones de asignación de caso intransitivos

En la figura 3.24 se observan las estructuras propias de los verbos intransitivos. Estos tipos restringen su *input* mediante los rasgos semánticos EARG e IARG, que dan información sobre si el verbo en cuestión tiene o no argumento interno o externo. Como se ve en 3.24, solo pueden participar de estos patrones los verbos intransitivos: aquellos para los que no se hayan explicitado tanto el argumento interno como el externo.

En la figura 3.25 se muestra el tipo *v-transitive-c*. Este es el único patrón de la gramática que asigna caso acusativo. Opera sobre verbos con un argumento

externo y otro interno; que, además, deben tener morfología activa: VOICE active. La asignación de acusativo tiene lugar sobre el segundo argumento de ARG-ST, el interno. El rasgo INTR regula que no participen de este patrón verbos que no asignen acusativo a su objeto. Cobrará sentido cuando se expliquen los objetos de caso oblicuo en 3.5.2.

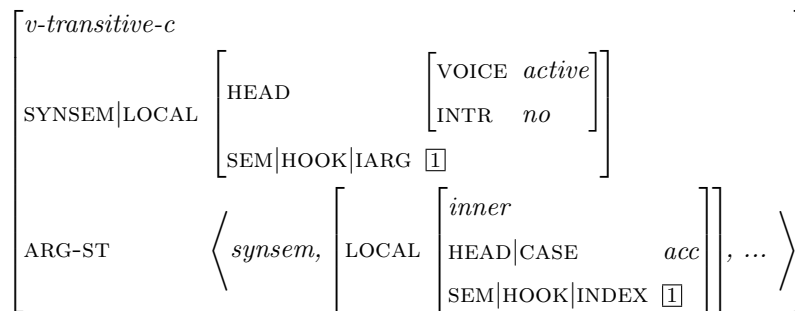


Figura 3.25: Patrón de asignación de caso transitivo

Por su parte, *v-passive-c* —figura 3.26— recoge un patrón de asignación de caso también propio de verbos con un argumento externo y otro interno: la construcción pasiva. Esta regla pide un verbo en voz pasiva (VOICE pass), obvia el argumento externo de este y asigna caso nominativo al interno.

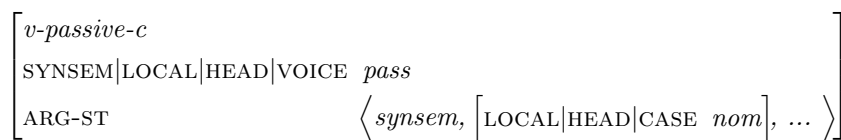


Figura 3.26: Patrón de asignación de caso pasivo

El tipo *v-rpass-c*, cuya estructura se observa en la figura 3.27, recoge un tercer patrón de asignación de caso: la construcción de pasiva refleja. Es una construcción propia también de verbos transitivos. Consiste, de nuevo, en obviar el argumento externo del verbo y asignar caso nominativo al interno. Se aplica a verbos en voz activa (VOICE active), a los cuales se añade un clítico de tercera persona *se*. En la figura 3.27 se observa cómo el clítico en cuestión se recoge en la lista CLITICS del argumento externo. Es un clítico

de tercera persona: *se*. Se ha establecido concordancia entre este y el sujeto paciente. Este análisis sugiere que es el argumento no realizado el que aporta propiamente el clítico; sugiere que hay una cierta relación entre la omisión de este y la aparición del clítico. Como se verá por extenso en el capítulo 4, en SSG, los clíticos de un verbo son la suma de los clíticos que aportan a este sus argumentos en ARG-ST.

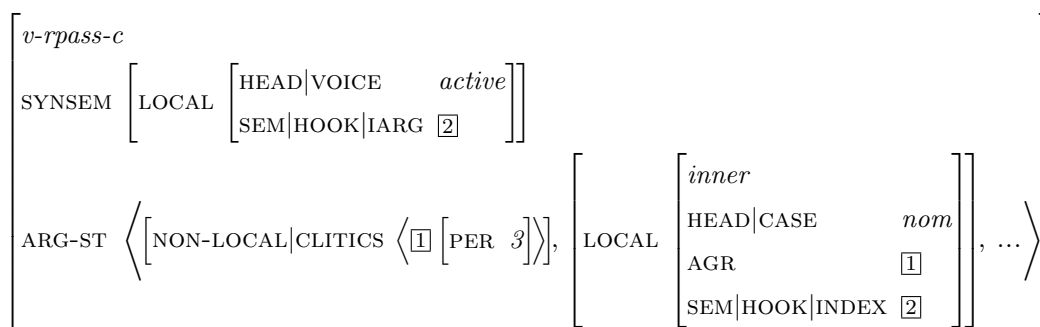


Figura 3.27: Patrón de asignación de caso pasiva refleja

En construcciones como la pasiva refleja, hay clíticos cuya adscripción a un argumento no es evidente. Aquí se preserva esa relación. Se asume la hipótesis de que el clítico de pasiva refleja lo aporta el argumento externo que, aun presente en ARG-ST, no se realiza explícitamente por no pasar a las listas de VAL¹⁷. Como se verá en adelante, no es esta una situación única en SSG: los verbos pronominales recibirán un análisis similar.

3.4. Verbos pronominales

En este apartado se completa el apartado previo —3.3— con un último patrón de asignación de caso para los argumentos estructurales: el propio de los verbos pronominales. Se sigue para ello la descripción que de este tipo de verbos hace la *Nueva gramática de la lengua española*¹⁸. Los verbos pronomi-

¹⁷Véase el capítulo 4.

¹⁸*Nueva gramática de la lengua española*, Real Academia Española, Espasa (2009), II, pp. 3099-3111.

nales son verbos, como *arrepentirse*, *desgañitarse*, *dignarse*, etc. que siempre se construyen con un clítico sin valor argumental. Este clítico concuerda con los rasgos de persona y número del verbo y el sujeto, y en tercera persona es *se*. Estos verbos tienen un comportamiento similar, entonces, al de los verbos reflexivos (*peinarse*, *lavarse*, etc.), pero, a diferencia de aquellos, para estos no es posible explicitar un sintagma correspondiente al clítico, pues este no tiene valor argumental. Véanse los ejemplos de (13), donde *arrepentirse* es pronominal y *peinarse* reflexivo.

- (13) a. ¿Se ha arrepentido de no ir a la playa?
 b. ¿Cómo puedo peinarme?
 c. *¿Se_i ha arrepentido a sí mismo_i de no ir a la playa?
 d. ¿Cómo puedo peinarme_i a mí mismo_i?

De acuerdo con la *Nueva gramática de la lengua española*, la mayoría de los verbos pronominales denotan cambios de estado o experiencias, en los que el paciente o el experimentante no ejerce un control sobre el proceso. A este esquema semántico, adscribible a un verbo, se le ha llamado tradicionalmente voz media. En latín, a ciertos verbos transitivos se les podía aplicar la voz media mediante el uso de morfología pasiva. En cambio, el añadido a estos verbos de un pronombre reflexivo les aportaba valor reflexivo. En su evolución, el latín perdió la morfología pasiva, lo cual propició la neutralización de las dos estrategias: la voz media tendió a expresarse también mediante el añadido al verbo de un pronombre reflexivo¹⁹. En algún momento, por tanto, el pronombre reflexivo debió resultar ambiguo: podía indicar tanto un sentido reflexivo como un sentido medio.

¹⁹De acuerdo con la *Nueva gramática de la lengua española*, véase cómo la forma activa latina *curāre*, con pronombre *se*, toma sentido reflexivo (*curāre se suamque aetātem*, ‘cuidarse, mantenerse en forma’), mientras que la forma pasiva *curāri* tiene sentido medio (*Corpōra curāri possunt*, ‘los cuerpos pueden curarse, sanar’). Con la desaparición de la morfología pasiva, ambos sentidos fueron recogidos en la forma activa con pronombre reflexivo. Por tanto, *Ille se curat* resultó ambiguo: ‘se cura a sí mismo’ o ‘sana’.

Este parece ser el origen de los actuales verbos pronominales: provienen de una construcción sintáctica reflexiva a la que se dotó de sentido medio. Esta construcción cuenta con un clítico (correspondiente a un argumento interno original) y un sujeto (el propio de la construcción reflexiva original). El único rol temático del patrón semántico medio se identificó con el sujeto sintáctico; con lo cual, el clítico correspondiente al argumento interno original perdió su valor argumental.

Por ello, en SSG, el patrón pronominal recibe una formalización similar a la de los verbos reflexivos. En SSG, la lista ARG-ST de un verbo reflexivo cuenta con un sujeto y un objeto, donde el objeto es una anáfora correferencial con el sujeto; y esta anáfora es la que aporta el clítico al verbo. Un verbo pronominal contará con una lista ARG-ST similar, con un sujeto seguido de un segundo elemento que aporta el clítico. No puede decirse propiamente que este segundo elemento sea un objeto, un argumento, etc. Intuitivamente, se concibe como el vestigio de lo que fue el objeto del patrón reflexivo. Formalmente, es un elemento *synsem* sin valor referencial (sin rol temático, por tanto), con presencia en ARG-ST (donde aporta el clítico), pero ausente en COMPS (y sin realización sintáctica, por tanto). No se especifica la naturaleza interna o externa de este sujeto, para hacer a este patrón compatible con verbos inergativos e inacusativos²⁰.

$$\left[\begin{array}{l} \textit{pronominal} \\ \text{SYNSEM|LOCAL} \left[\begin{array}{l} \text{VAL} \left[\begin{array}{l} \text{SUBJ} \langle \text{1} \rangle \\ \text{COMPS} \text{ 2} \end{array} \right] \\ \text{SEM|HOOK|PARG} \text{ 3 } \textit{index} \end{array} \right] \\ \text{ARG-ST} \langle \text{1} \left[\text{LOCAL|SEM|HOOK|INDEX} \text{ 3} \right], \textit{synsem} . \text{ 2} \rangle \end{array} \right]$$

Figura 3.28: Relación entre ARG-ST y VAL para verbos pronominales

²⁰Según la *Nueva gramática de la lengua española*, solo algunos verbos pronominales son inacusativos.

Para dar cuenta de todo ello se han creado dos tipos: *pronominal* y *v-pronominal-c*. El primero se concibe como un patrón de relación entre ARG-ST y VAL, alternativo a los ya vistos (*unergative*, *unaccusative* y *passive*). Es un patrón similar al inergativo, pero obvia el argumento interno en el paso de ARG-ST a COMPS —véase en la figura 3.28—.

Por último, es necesario introducir en *hook* un nuevo rasgo PARG que regule la adscripción de rol temático al sujeto del patrón pronominal. Este sujeto podría identificarse con una posición de argumento interno o externo (EARG o IARG), y con uno u otro rol temático (ARG1 o ARG2). Para permitir que cada verbo adscriba el índice de este sujeto a los rasgos adecuados, *pronominal* publica este índice en PARG. Será cada verbo el que identifique después el rasgo PARG con los rasgos IARG, EARG, ARG1 o ARG2 pertinentes. PARG permite, además, regular qué verbos admiten o no patrón pronominal: un verbo pronominal debe codificarse como PARG *index*, mientras que un verbo no pronominal debe ser PARG *no-individual*.

Como subtipo de *pronominal*, se ha creado *v-pronominal-c*, tipo que añade a aquel ciertas peculiaridades propias de los verbos pronominales: figura 3.29. Obsérvese en dicha figura que el argumento interno no tiene valor referencial (INDEX *no-individual*). En cambio, es el responsable de la aparición del clítico en el verbo: CLITICS. Además, se establece que se trata de un elemento en origen estructural mediante la caracterización CASE *str*.

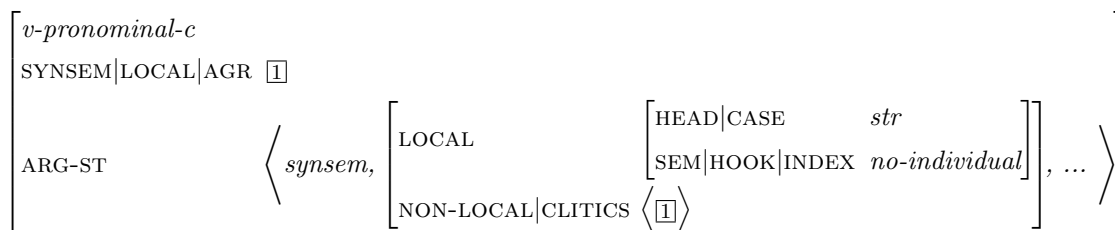


Figura 3.29: Patrón de asignación de caso para verbos pronominales

Esta formalización recoge la intuición consistente en que el patrón pronominal ha sido creado a partir de un patrón reflexivo original, como sugieren

los estudios diacrónicos. Como se verá en el capítulo 4, los clíticos se conciben en esta gramática como marcas morfológicas aportadas al verbo desde sus posiciones argumentales. Por ejemplo, la anáfora de un verbo reflexivo es la que le aporta a este su clítico. Por tanto, es natural que sea una posición de ARG-ST la responsable de la aparición del clítico de un verbo pronominal. Por otro lado, es natural en SSG que una determinada construcción obvie la existencia de un argumento presente en la lista ARG-ST del verbo. Así se ha formalizado la eliminación del argumento externo que suponen las construcciones pasivas. Del mismo modo, el patrón pronominal supone la eliminación de lo que fue un argumento interno. Por último, el clítico de un verbo pronominal lo aporta una posición de ARG-ST sin posterior realización sintáctica. Esta situación ya se vio en 3.3 en la formalización de la pasiva refleja: en ambos casos, el clítico característico de la construcción se entiende como el resto morfológico de la posición argumental eliminada.

La mayoría de los verbos pronominales admiten tanto el patrón pronominal como el transitivo. Es el caso de los verbos *secar* y *admirar*, en (14).

- (14)
- a. Me he secado el pelo
 - b. ¿Se ha secado la toalla?
 - c. Admiraba la belleza del paisaje
 - d. Se admiraba de la belleza del paisaje

El uso de uno u otro patrón puede aportar al verbo, o no, significados diferentes, con diferencias más o menos sutiles. En ocasiones, es una cuestión compleja el decidir si, dadas estas posibles diferencias, el uso pronominal y el transitivo deben codificarse como entradas léxicas diferentes, o como alternancias de una misma entrada. Es cierto que SSG pretende evitar la duplicación de entradas en el lexicon. Pero esta estrategia solo es adecuada cuando los distintos usos de una palabra no varían su significado, o cuando lo especifican de un modo regular.

Por ello, en SSG se ha optado por entender como alternancias solo los usos transitivos y pronominales que no impliquen grandes diferencias idiosincrásicas de significado; y obedezcan a un mismo esquema semántico básico. Por ejemplo, se entienden como diferentes versiones de una misma unidad léxica los usos transitivo y pronominal de los verbos que en ambos casos se interpretan como el mismo estado de experiencia: *admirar* en (14-c) y (14-d). También se entenderán como versiones distintas de una misma entrada léxica los usos transitivo y pronominal cuando se correspondan, respectivamente, con la interpretación causativa y no causativa de un mismo cambio de estado: *secar* en (14-a) y (14-b).

La alternancia de (14-a) y (14-b) refuerza, desde un punto de vista meramente sincrónico, la naturalidad del análisis propuesto en SSG para los verbos pronominales. Un verbo que —como *secar*— participe de esta alternancia debe contar en su codificación léxica con los dos argumentos estructurales propios de una realización —3.2—. A partir de esa codificación original, es posible la aplicación del patrón transitivo y el pronominal, que acarrearán las interpretaciones causativa y media, respectivamente. La eliminación del argumento interno, en este caso, es una alternativa a su realización. La situación es similar a la que se establece entre la versión causativa de este verbo y una posible versión pasiva —salvando las diferencias semánticas existentes—. Y, por tanto, SSG formaliza estas construcciones de un modo similar, coherente²¹.

3.5. Argumentos no estructurales

Hasta el momento —apartado 3.5— se ha visto que un verbo cuenta en el léxico con una serie de argumentos estructurales, legitimados por su perte-

²¹De acuerdo con lo expuesto, SSG da cuenta del que, según la *Nueva gramática de la lengua española*, es el prototipo de verbo pronominal del español: un verbo intransitivo con sentido medio. No obstante, el patrón v-pronominal-c también es compatible con otros verbos pronominales cuyo sujeto es un agente (*confesarse*, *contenerse*, *decidirse* o *justificarse*). Han quedado fuera de la gramática, en cambio, unos pocos verbos transitivos con carácter pronominal: *encontrarse* —*encontrárselo*—, *quedarse* —*quedárselo*—.

nencia a una clase verbal. Las actividades cuentan con un solo argumento: el agente de la acción que denotan. Los logros cuentan también con un argumento: el paciente del cambio de estado que denotan. Las realizaciones tienen, por su parte, dos argumentos: el agente o causa del cambio de estado que denotan, y el paciente de este. Por último, los estados tienen al menos un argumento: quien se encuentra en el estado denotado por el verbo. Estos son los argumentos básicos de los verbos, comunes a todos ellos, e ineludibles, de acuerdo con Levin y Rappaport (1998). En este apartado se explica cómo a estos argumentos se les puede añadir otros, legitimados por el significado idiosincrásico del verbo: los argumentos no estructurales.

3.5.1. Argumentos con caso nominativo o acusativo

Como se dijo en 3.2, los verbos pueden añadir a sus argumentos estructurales otros legitimados por su significado idiosincrásico. Por ejemplo, el verbo *tener*, en principio, en tanto que estado, solo denota un participante: aquel que posee —*el hotel* en (15-a)—. Pero el significado idiosincrásico de *tener* pide que se especifique el objeto de la posesión. Esto legitima la aparición de un segundo participante: lo que se tiene —*sala de reuniones* en (15-a)—. Este segundo argumento de *tener* está legitimado, entonces, no porque este verbo sea un estado; sino porque *tener* denota, de forma idiosincrásica, que se tiene algo.

- (15) a. ¿Tiene el hotel sala de reuniones?
 b. El hotel ha contratado al guía para la excursión

En cambio, en (15-b), *contratar* es una realización. Como se vio en 3.2, los objetos de las realizaciones sí son argumentos estructurales, que denotan algo que es común a todas las realizaciones. Todas las realizaciones denotan cambios de estado, y el objeto de estas denota siempre al paciente afectado por ese cambio. En concreto, los objetos que son argumentos estructurales tienen papel temático de paciente afectado, mientras que los que no son estructurales son temas no necesariamente afectados.

En cualquier caso, es interesante que dos verbos con naturaleza semántica diferente, como se ve en (15), pueden participar de un mismo patrón de comportamiento transitivo. En SSG esta realidad se captó estableciendo un único tipo caracterizador del verbo transitivo activo: *v-transitive-c*, cuya estructura se repite en la figura 3.30. Esta regla puede tomar como *input* tanto una instancia léxica de *accomplishment*, como de *state* o *activity*. Gracias a ello, en SSG, la asignación de caso para los argumentos no estructurales no es una cuestión idiosincrásica carente de generalidad. El objeto acusativo de una realización, como *contratar*, y el de un estado, como *tener*, tienen naturaleza semántica diferente. Sin embargo, reciben caso del mismo modo, mediante la aplicación del mismo patrón *v-accusative-c*.

$$\left[\begin{array}{l} v\text{-transitive-c} \\ \text{SYNSEM|LOCAL} \left[\begin{array}{l} \text{HEAD|VOICE} \quad \textit{active} \\ \text{SEM|HOOK|IARG} \quad \boxed{1} \end{array} \right] \\ \text{ARG-ST} \quad \left\langle \textit{synsem}, \left[\begin{array}{l} \text{LOCAL} \left[\begin{array}{l} \textit{inner} \\ \text{HEAD|CASE} \quad \textit{acc} \\ \text{SEM|HOOK|INDEX} \quad \boxed{1} \end{array} \right], \dots \right\rangle \end{array} \right.$$

Figura 3.30: Patrón de asignación de caso transitivo

Téngase en cuenta, no obstante, que el *input* de *v-accusative-c* es distinto si es una instancia léxica de *accomplishment*, frente a si lo es de *state* o *activity*. Si el verbo es *accomplishment*, el *input* ya caracteriza explícitamente la naturaleza del argumento interno, y *v-accusative-c* solo especifica el caso de este. Pero si el verbo es *state* o *activity*, la especificación que del argumento interno haga *v-accusative-c* será su primera caracterización explícita. Esto obliga a explicitar en *v-accusative-c* cierta estructura relativa a este argumento interno. Esta caracterización será redundante con respecto a la caracterización de *accomplishment*, pero nueva, y necesaria, con respecto a la de *state* y *activity*.

Por ello —figura 3.30— en *v-accusative-c* se ha caracterizado explícitamente el argumento interno como tal (*inner*), y se ha identificado su índice con el rasgo IARG del verbo. Téngase en cuenta que no es posible identificar

este argumento interno con un rol temático concreto, pues, como se ha visto, este no es siempre el mismo: para una realización, el argumento interno es paciente afectado (ARG2); mientras que para un estado o actividad es un tema no necesariamente afectado (ARG3). La estrategia seguida por SSG es identificar el índice con una posición neutra (IARG) y esperar que sea cada verbo el que identifique a su vez IARG con el rol temático oportuno (ARG2 o ARG3). Téngase en cuenta que un estado o actividad puede establecer esta identidad semántica sin explicitar el argumento sintáctico correspondiente en ARG-ST. Sigue pudiendo, por tanto, participar de patrones intransitivos.

Sin embargo, el origen diferente de los objetos estructurales y no estructurales supone diferencias de comportamiento sintáctico. Siguiendo a Levin y Rappaport (1998), un argumento estructural es ineludible, mientras que un argumento idiosincrásico puede no realizarse en una oración. Este contraste se muestra en las oraciones de (16).

- (16) a. Ese hotel tiene sala de reuniones. ¿Y nuestro hotel tiene?
 b. *Ese hotel ha contratado un guía. ¿Y nuestro hotel ha contratado?

También el hecho de que el papel temático de esos objetos sea distinto supone diferencias sintácticas. La construcción pasiva pide como sujeto un argumento que denote un paciente afectado. Los objetos de las realizaciones son pacientes afectados. Por tanto, estas admiten la construcción pasiva con gran naturalidad. En cambio, los objetos de otros verbos transitivos (estados o actividades) no son pacientes necesariamente afectados. La construcción pasiva no resulta natural, por ello, con ese tipo de verbos. Esto explica las diferencias de comportamiento de (17), entre los verbos *contratar* y *tener*.

- (17) a. *La sala de reuniones es tenida por el hotel
 b. El guía ha sido contratado por el hotel para la excursión

Dado lo expuesto hasta ahora, es fácil dar cuenta de estos contrastes en SSG. Las realizaciones (*accomplishment*), como se vio en 3.2, explicitan dos argu-

mentos sintácticos en ARG-ST. El segundo es el argumento interno en cuestión. Por ello, cualquier instancia léxica de *accomplishment* deberá explicitar en la oración, al menos, estos dos argumentos. Para ellas, la omisión del objeto de (15) es, por tanto, imposible. Pero los estados como *tener* (*state*), o las actividades (*activity*), solo explicitan un argumento en ARG-ST. Esta caracterización las hace compatibles con los patrones de comportamiento transitivos: a ese primer argumento en ARG-ST se le puede añadir después otro argumento interno. Pero también permite usos intransitivos como el que se observa en (16-a).

Por su parte, los contrastes relativos a la construcción pasiva —(17)— se pueden modelar haciendo uso del tipo *no-individual* y los rasgos ARG2 y ARG3. Un verbo que denota cambio de estado, como una realización, asigna siempre a su objeto el rol temático de paciente. Esto se formaliza identificando el índice (*index*) del argumento en cuestión con el rasgo ARG2 del verbo. En cambio, un verbo que no denota cambio de estado no tendrá nunca paciente afectado, lo cual se representa evaluando el rasgo ARG2 del verbo como *no-individual*. Este es el caso de los estados y las actividades. Con todo ello, para que la construcción de pasiva solo opere sobre verbos con paciente afectado, basta pedir al *input* de esta construcción valor *index* para su rasgo ARG2. Téngase en cuenta, por último, que el índice del objeto de los estados y actividades se identifica con el rasgo ARG3 del verbo, rasgo que denota el papel temático de tema no afectado.

De lo aquí expuesto se concluye que existen argumentos idiosincrásicos que, sin embargo, siguen los patrones generales de asignación de caso propios de los argumentos estructurales. Son objetos que, en activa, reciben caso acusativo. Habría que entender que estos mismos objetos, en pasiva, debieran recibir caso nominativo. No obstante, como se ha visto, el patrón *v-passive-c*, dado que necesita un *input* transitivo con paciente afectado, solo es aplicable sobre realizaciones, cuyo objeto es estructural. Pero sí es aplicable para los verbos con objeto idiosincrásico la construcción de pasiva refleja: *v-rpass-c*. Este patrón, efectivamente, contempla la existencia de un argumento interno, que es estructural en las realizaciones, pero no estructural para estados y

actividades transitivos. Como se observa en los ejemplos de (18), tanto el verbo *tener* como *contratar* admiten esta construcción.

- (18) a. Se tienen salas de reuniones
 b. Se han contratado guías para la excursión

Por ello, la estructura de rasgos de **v-rpass-c** —figura 3.27— explicita, como ocurre para **v-transitive-c**, la naturaleza interna del argumento en cuestión, e identifica el índice de este con el IARG del verbo.

3.5.2. Argumentos con caso oblicuo

Se observa que hay verbos que subcategorizan un objeto idiosincrásico —como *tener* en 3.5.1—, pero que no hacen uso del patrón general de asignación de caso **v-transitive-c**. En cambio, asignan caso oblicuo al objeto en cuestión²². Para ejemplificar tal situación valga una actividad: el verbo *jugar*. Las oraciones de (19-a) y (19-b) demuestran que *los bolos*, el objeto de *jugar*, recibe caso oblicuo mediante la preposición *a*, y no acusativo —los juicios de gramaticalidad son correspondientes al español de España—.

- (19) a. ¿Puedo jugar a los bolos cerca del hotel?
 b. *¿Puedo jugar los bolos cerca del hotel?

²²De nuevo se sigue aquí, para la descripción de los fenómenos, la *Nueva gramática de la lengua española* II, pp. 2715-2726. Esta obra recoge bajo el mismo término de complemento de régimen preposicional los distintos complementos preposicionales pedidos por la semántica de un predicado. No distingue entre los complementos cuya preposición es una mera marca de caso, y aquellos cuya preposición es un predicado. Los primeros son participantes del predicado que denota el verbo, marcados con la preposición; los segundos, predicados que predicán sobre otro participante de ese verbo. En esta gramática, se encuentran alusiones que invitan a contemplar estas diferencias, pero no se hace una distinción formal. En cualquier caso, SSG contemplará a estos dos tipos de complementos como realidades diferentes. Los primeros se presentan en este apartado, y los segundos en 3.5.3. Por lo demás, se obvia aquí gran parte de la casuística que muestra esta gramática. Existen verbos que piden de forma rigurosa una única preposición —*disentir de*—, pero otros permiten mayor variación: *hablar de, sobre, acerca de*. También existen verbos que subcategorizan dos argumentos oblicuos —*comprometerse a con*—.

c. Ellos han adecuado el hotel a la normativa

Esta solución es frecuente entre estados (*disfrutar con, olvidarse de, enterarse de...*) y actividades (*jugar a, comunicarse con, desentenderse de, hablar de...*). Estos verbos añaden un solo objeto idiosincrásico a su sujeto estructural, con caso oblicuo. Pero también es posible que una realización añada un segundo objeto en oblicuo a sus dos argumentos estructurales. Es lo que ocurre con *adecuar* en (19-c). Para este verbo existe un agente (*Ellos*) que hace que un paciente (*el hotel*) quede adecuado conforme a un tema (*la normativa*). Por tanto, en (19-c), al significado propio de toda realización, *adecuar* le añade un tercer argumento idiosincrásico (aquello conforme a lo que se adecuaba); igual que *jugar* o *tener* añaden un primer objeto a su sujeto estructural.

SSG no contempla un patrón de asignación de caso alternativo a **v-accusative-c** para cubrir estas situaciones. En cambio, se entiende que este tipo de oraciones son usos intransitivos —(19-a)— o transitivos —(19-c)— que deben ser modelados por los patrones **v-unergative-c**, **v-unaccusative-c** o **v-transitive-c**. En este caso, *jugar*, como verbo inergativo que es, asignará nominativo a su único argumento estructural mediante el patrón **v-unergative-c**. Y *adecuar* asignará nominativo y acusativo a sus dos primeros argumentos mediante **v-transitive-c**. Asumido esto, SSG sí contempla un patrón que añade un objeto en oblicuo a la lista ARG-ST de cualquier verbo que lo permita. El rol temático de este objeto será de tema no afectado (ARG3).

En principio, cualquier gramática computacional basada en HPSG necesitaría dos reglas distintas para añadir un argumento, por un lado a *jugar*, y por otro a *adecuar*. La primera pediría al *input* un único argumento en ARG-ST y añadiría a este el argumento oblicuo; y la segunda pediría dos argumentos en ARG-ST y añadiría el oblicuo como tercero. Otro tanto ocurriría para el añadido de dativos, o de predicativos. Con ello se pierde la generalidad deseada: la intuición dice que el añadido de un oblicuo, un dativo o un predicativo es un único proceso gramatical, aplicable a listas de argumentos

de longitud variable. Además, el contemplar muchas reglas léxicas supone un incremento del tiempo de procesamiento.

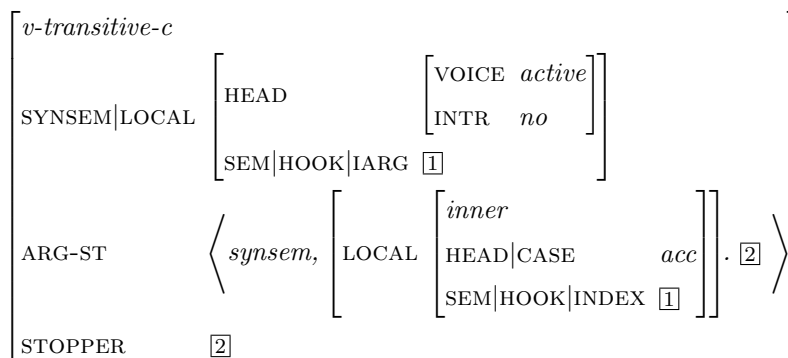


Figura 3.31: Patrón de asignación de caso transitivo

Por tanto, SSG cuenta con un sistema que añade un argumento a un tipo con independencia de la longitud de la lista ARG-ST de este. Para ello se ha utilizado una lista llamada STOPPER, adscrita al tipo *sign*. Los patrones vistos hasta el momento (*v-transitive-c*, *v-unergative-c*, etc.) deben identificar el resto de su lista ARG-ST con su lista STOPPER, como se muestra en la figura 3.31 para *v-transitive-c*. Dada esta caracterización, es posible añadirles a estos patrones un argumento, colocándolo en su lista STOPPER. De ello se encargan reglas como *v-obliq-c*: colocan el argumento nuevo en la lista STOPPER del *input*; confiando en que, en tal *input*, STOPPER y el resto de ARG-ST hayan sido identificados. En concreto, *v-obliq-c* es una regla que utiliza este sistema para añadir al *input* un argumento de caso oblicuo —figura 3.32—. Lo interesante es que la misma regla *v-obliq-c* puede operar tanto sobre un *input* transitivo como intransitivo. En ambos casos, la misma regla añade un argumento oblicuo, pero el que este sea segundo o tercer argumento en ARG-ST depende de la naturaleza del *input*.

En la figura 3.33 se representa la jerarquía de tipos dentro de la cual se entiende a *v-obliq-c*. De acuerdo con ella, *v-nominative-c* deja de ser el supertipo absoluto de todos los patrones de asignación de caso, y pasa a serlo solo de aquellos que asignan caso nominativo: *v-transitive-c*, *v-passive-c*, etc. Se crea un supertipo general *v-case-c* del que heredan, como hermanos,

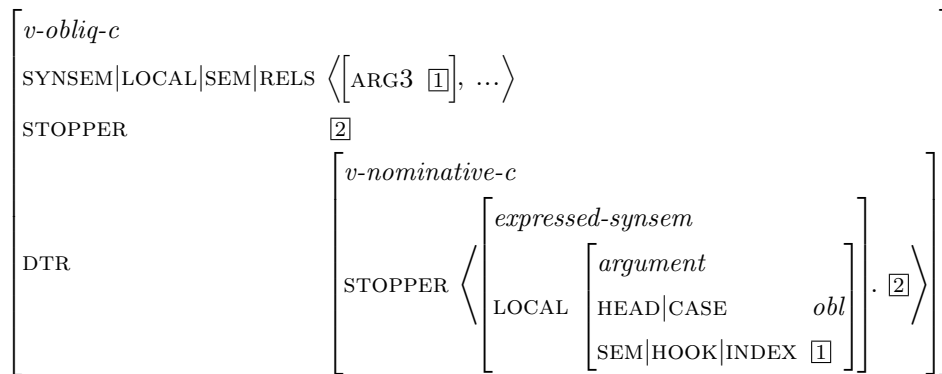


Figura 3.32: Patrón de añadido de argumento oblicuo

v-nominative-c y v-obliq-c. Además, se ha establecido que v-obliq-c toma como *input* cualquier subtipo de v-nominative-c.

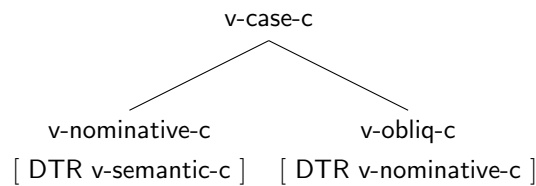


Figura 3.33: v-case-c: jerarquía de tipos

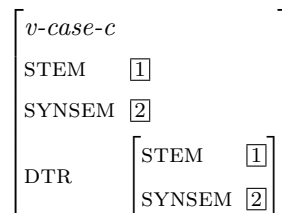


Figura 3.34: v-case-c

Téngase en cuenta que cierta caracterización de v-nominative-c debe pasarse al nuevo supertipo v-case-c, para que sea heredada también por v-obliq-c. Se trata de la identidad que se asumió en 3.3 para v-nominative-c entre los rasgos STEM y SYNSEM del *input* y el *output*. En todas las reglas léxicas en SSG se asume que la estructura de rasgos del *input* y el *output* es directamente unificable, con lo cual la regla nunca cambia la estructura del *input*: solo la

específica. Por tanto *v-case-c* cuenta con la estructura de rasgos de 3.34.

Si, además, *v-obliq-c* identifica el resto de la lista *STOPPER* del *input* con la lista *STOPPER* del *output*, *v-obliq-c* puede ser, a su vez, *input* de otra regla de añadido argumental (quizá de un dativo). SSG debe asegurar que este proceso se itera del modo deseado. En cualquier caso, es imprescindible que, tras la última aplicación de una regla léxica sobre una palabra, se cierre su lista *STOPPER*. Si quedase abierta, las reglas gramaticales podrían saturar infinitamente una lista potencialmente infinita.

Por tanto, cuando un signo léxico entra como *input* de cualquier regla de proyección gramatical, su lista *STOPPER* se evalúa como *null*, como lista vacía. Con ello se asegura que la lista *ARG-ST* de la unidad léxica —e, indirectamente, también *COMPS*— queda cerrada. Concretamente, el tipo *phrase* de SSG, supertipo general de toda regla gramatical, formalización del concepto de sintagma, recibe la siguiente caracterización de la figura 3.35.

$$\left[\begin{array}{l} \textit{phrase} \\ \text{HEAD-DTR|STOPPER} \quad \textit{null} \\ \text{NON-HEAD-DTR|STOPPER} \quad \textit{null} \end{array} \right]$$

Figura 3.35: Cierre de *STOPPER*

Además, en la figura 3.32 se ha codificado al argumento oblicuo en cuestión como *expressed-synsem*. Este, en SSG es el tipo de *synsem* que aparece en los signos lingüísticos con realización fonética explícita. Con esta especificación se evita la posibilidad indeseable de que un argumento se añada al verbo para que después sea elidido.

En definitiva, en SSG, *jugar* y *adecuar* asignan caso a sus argumentos estructurales mediante la aplicación de los patrones *v-unergative-c* y *v-transitive-c*, respectivamente. Esto preserva la generalidad en la gramática: todos los verbos con patrón de comportamiento inergativo obedecen a la regla *v-unergative-c*, y todos los transitivos a la regla *v-transitive-c*. La cuestión

tratada en este apartado es que algunos de estos verbos pueden añadir, a esos patrones generales, un argumento con caso oblicuo: *jugar* y *adecuar*. Este argumento lo añade SSG mediante la aplicación posterior de una sola regla: *v-obliq-c*. De nuevo se preserva la generalidad en la gramática, pues es una misma regla la que legitima la aparición de ese argumento, sea para un verbo transitivo o intransitivo.

Es necesario restringir qué verbos admiten un objeto en acusativo y qué verbos admiten un objeto de caso oblicuo. La intuición dice que un verbo, normalmente, asigna acusativo a su complemento más directo. Y los casos de objeto oblicuo son situaciones excepcionales donde un verbo, de forma idiosincrásica, no marca directamente a su complemento. Debe hacerlo, entonces, una preposición. Esta es una característica idiosincrásica de la sintaxis del verbo. Y como tal se caracteriza en SSG. Para ello, se utiliza el rasgo INTR, adscrito a *verb*. Un verbo caracterizado como INTR *yes* es aquel incapaz de asignar caso acusativo a su objeto. Estos verbos tendrán vedado su paso por los patrones transitivos y pasivos (*v-transitive-c*, *v-passive-c* y *v-rpass-c*).

Además, téngase en cuenta que un verbo que marca con caso oblicuo a su objeto lo hace siempre con la misma preposición: se puede jugar *a* los bolos, pero no se puede jugar *de* los bolos o *en* los bolos. Para establecer esta restricción, se ha utilizado un sistema inspirado en *Matrix*: el rasgo local KEYS, de valor *keys*. Se le suponen a *keys* dos rasgos: KEY y OKEY. En KEY se coloca el nombre del predicado principal del signo: el identificador del predicado establecido como valor de PRED —véase 1.3.3.3—. Y OKEY sirve para establecer restricciones idiosincrásicas a la subcategorización del objeto. Su valor será una lista con, en principio, un identificador: el del objeto subcategorizado. OKEY es un rasgo pensado para modelar colocaciones, frases hechas, etc. En SSG se utiliza para establecer otro caso de subcategorización idiosincrásica: la rección caprichosa, por parte de un verbo, de una determinada preposición. Por tanto, a la estructura de *v-obliq-c* vista anteriormente hay que añadirle la caracterización de la figura 3.36. En ella, se muestra cómo el valor de KEY del argumento ha de coincidir con el valor que pide el verbo a

dicho argumento: OKEY.

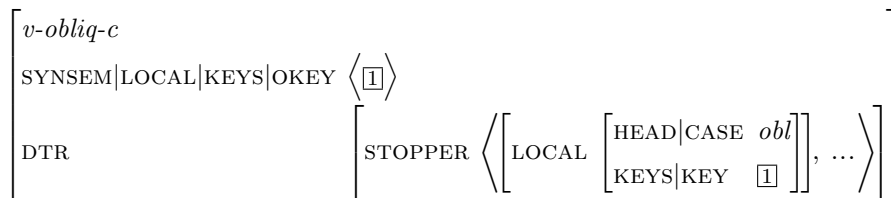


Figura 3.36: Rección de preposición en *v-obliq-c*

Como se ve, se ha supuesto que el valor de OKEY es una lista, dentro de la cual se encuentra el identificador en cuestión. Esto permitiría, si fuera necesario, establecer distintas posibilidades de rección. Pero sobre todo permite evaluar OKEY como lista vacía (null). Con ello, se daría a entender que un verbo no rige preposición. Es un modo natural de impedir el paso por *v-obliq-c* a verbos como *saber* o *tener*, cuyo objeto no estructural lleva siempre caso acusativo. Esta clase de verbos debe estar, por tanto, caracterizada en el léxico como OKEY null.

3.5.3. Complementos predicativos

En 3.5.2 se ha visto cómo ciertos verbos (*jugar a*, *adecuar a*) subcategorizan un sintagma preposicional como complemento. Se ha dicho que este sintagma es un argumento con caso oblicuo. Se trataba, por tanto, de un complemento preposicional cuya preposición era una mera marca de caso, carente de información semántica: no era un predicado. Hay verbos que subcategorizan, como aquellos, un complemento idiosincrásico preposicional, pero cuya preposición es un predicado²³. El añadido de estos complementos al

²³Esta clase de complementos ha sido tradicionalmente adscrita al grupo de lo que la *Nueva gramática de la lengua española* llama complementos de régimen —3.5.2—, o al grupo de los adjuntos; en función de que se hiciese primar, o no, su carácter de elemento pedido por el significado del núcleo. En SSG, en tanto que elementos pedidos por el significado del verbo, no se tratan como adjuntos. Y, en tanto que semánticamente diferentes de los argumentos de 3.5.2, no se tratarán como argumentos oblicuos. Son, en cambio, complementos pedidos por el verbo, pero con carácter predicativo: predicán sobre otro argumento verbal.

verbo crea una estructura de control, en la cual la preposición predica sobre el argumento del verbo inmediatamente anterior a él en ARG-ST²⁴. Son frecuentes estos usos entre estados, actividades, logros y realizaciones que denotan situación espacial o movimiento: *subir*, *bajar*, *correr*, *entrar*, *salir*, etc.

- (20) a. Ha subido el precio de la habitación a cien euros
 b. El botones ha subido el equipaje a la habitación

Como se puede observar en (20), este tipo de complementos se puede añadir tanto a un verbo transitivo como intransitivo. En (20-a), *subir* obedece a un patrón de comportamiento intransitivo: es un logro (*achievement*) con un único argumento estructural, interno, que recibe caso nominativo gracias al patrón de asignación *v-unaccusative-c*: *el precio de la habitación*. A la lista ARG-ST correspondiente se ha añadido un sintagma preposicional con la preposición predicativa *a*, que toma *el precio de la habitación* como argumento externo, y dice de él que ha ascendido hasta los *cien euros*. La misma relación se establece entre *el equipaje* y *la habitación* en (20-b). Pero en este caso *subir* funciona como verbo transitivo, de modo que *el equipaje*, el argumento interno, recibe caso acusativo. Este segundo uso de *subir* es resultado, en SSG, de la aplicación de *v-transitive-c* sobre el tipo caracterizador de las realizaciones: *accomplishment*.

Por tanto, SSG cuenta con una regla —*v-predicative-c*— que permite el añadido de un complemento predicativo preposicional a los verbos. Esta regla añade el predicativo del mismo modo que *v-obliq-c* añadía el argumento oblicuo. Asume como *input* cualquiera de los patrones básicos de asignación de caso: *v-transitive-c*, *v-passive-c*, etc. Y utiliza la lista STOPPER para añadir el complemento en cuestión.

²⁴La preposición de marca de caso oblicuo la suele establecer el verbo para su complemento, de forma que esta suele ser, para cada verbo, una en concreto. En cambio, para estos complementos predicativos la preposición varía, siempre dentro de un orden: la preposición debe denotar un movimiento o estado espacial compatible con el expresado por el verbo.

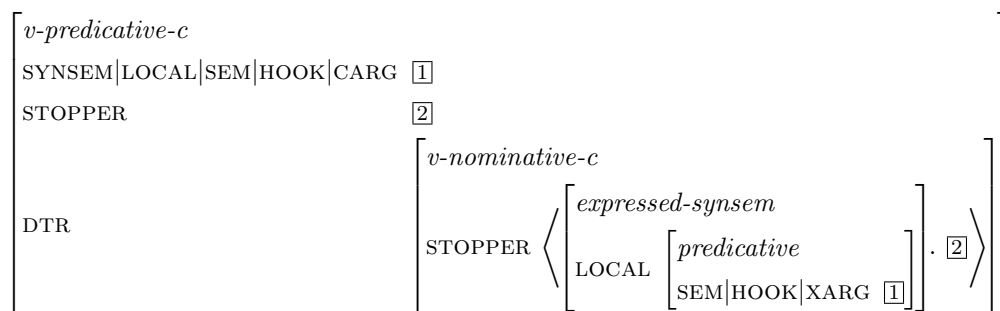


Figura 3.37: Patrón de añadido de complemento predicativo

Para restringir adecuadamente el *input* de esta regla se utiliza el rasgo CARG. El que un verbo admita o no un complemento de este tipo se debe especificar en su declaración léxica original. Si no lo permite, su valor de CARG debe ser *no-individual*. Pero la existencia de CARG está justificada, además, por ser el lugar donde se publica el índice del argumento sobre el que debe predicar el complemento que introduce *v-predicative-c*. Un verbo que admita predicativo debe identificar CARG con su argumento interno o externo (IARG o EARG), según deba ser uno u otro el objetivo de la predicación.

Obsérvese cómo, en la figura 3.37, el complemento predicativo añadido predica sobre el argumento cuyo índice se haya colocado en CARG: identidad entre el XARG del complemento y el CARG del verbo. Con ello —asumiendo que en *subir* se identificaron CARG e IARG— se logran las relaciones predicativas propias de todas las oraciones de (20), sin necesidad de explicitar en esta regla la posición en ARG-ST del objetivo de la predicación. Si *v-predicative-c* se hubiese aplicado sobre una actividad, en la que CARG se hubiese identificado con el índice del argumento externo (EARG), este mismo predicativo predicaría sobre tal argumento externo. Obsérvese, por último, que, en este caso, el elemento añadido es de tipo *predicative*, y no *argument*; lo cual restringe adecuadamente la naturaleza categorial del signo que ocupe esa posición.

Se han contemplado solo predicativos preposicionales. En realidad, en español, hay muchos otros casos de complementos predicativos que establecen estructuras de control. Por ejemplo, los verbos de percepción suelen contar con un verbo en infinitivo como predicativo: *Ya oigo a Teresa hablar de ti*. También son frecuentes los adjetivos y participios —*Tengo a los niños sucísimos* y *Tengo a los niños cansados de calor*— o los gerundios —*Llegaron cantando canciones*—. No se pueden restringir, entonces, estas estructuras a las preposiciones, ni a los verbos de situación espacial o movimiento. En principio, SSG está equipada para dar cuenta de todos estos datos. El tipo *predicative* unifica tanto con preposiciones como con verbos en forma no personal o adjetivos. Y el rasgo *CARG* permite establecer para cada verbo si admite o no este tipo de construcción. Pero falta una clasificación completa de los verbos que admiten predicativos. De momento, en la práctica, en SSG solo se han utilizado predicativos preposicionales para verbos de situación espacial y movimiento. Se deja a un trabajo futuro la tarea que falta al respecto.

Por último, la regla *v-predicative-c* se entiende en SSG dentro de la jerarquía de tipos de la figura 3.38. Es, como *v-obliq-c*, subtipo de *v-case-c*, y hermano de *v-nominative-c*, tipo al que toma como *input*.

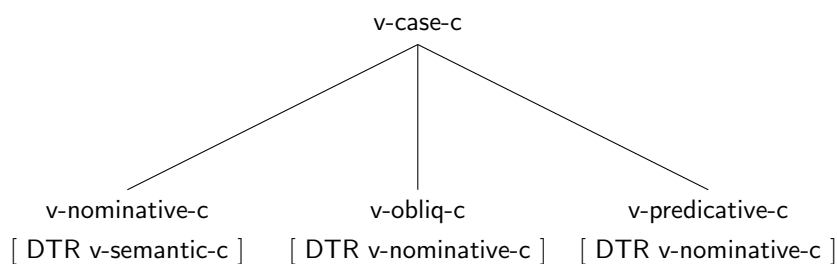


Figura 3.38: *v-case-c*: jerarquía de tipos

3.5.4. Argumentos con caso dativo

En este apartado se sigue la *Nueva gramática de la lengua española*²⁵. En español, ciertos verbos cuentan con complementos de caso dativo que for-

²⁵II, pp. 2655-2660.

man parte de su estructura argumental básica. Es el caso, entre otros, de los destinatarios de los verbos de transferencia (*dar*, *devolver...*), comunicación (*avisar*, *contestar*), etc. Y es el caso también de los experimentantes de los verbos psicológicos (*asustar*, *alegrar*), los de acaecimiento (*suced*, *ocurrir*), los de daño o provecho (*beneficiar*, *dañar*, *perjudicar*), etc.²⁶. También es posible encontrar otros muchos verbos (*abrir*, *subir*, *romper*, *contratar* etc.) para los que el complemento dativo, si bien es posible, no forma parte de su estructura argumental básica. Estos complementos denotan, normalmente, un beneficiario de la acción —*Me buscaron un albergue*—; un poseedor —*A la casa se le mojó el tejado*—; o un elemento alejado de la acción verbal, pero afectado en algún sentido por esta: un dativo ético —*No se me ponga nerviosa*—²⁷.

El dativo de *subir* no forma parte de su estructura argumental básica. En cambio, sí forma parte de esta estructura argumental básica el dativo de *dar*.

- (21) a. El botones le ha subido el equipaje al guía
 b. El guía le ha dado propina al botones
 c. ?El botones ha subido el equipaje al guía
 d. El guía ha dado propina al botones

Según Gutiérrez Ordóñez (1999), el que un dativo forme o no parte de la estructura argumental básica de un verbo tiene cierta relevancia sintáctica. Si el dativo sí forma parte de dicha estructura, es posible hacerlo aparecer en la oración sin explicitar su clítico correspondiente: (21-b) y (21-d). Pero es imposible omitir el clítico correspondiente a un dativo no argumental: (21-a) pero (21-c). Todo ello hace pensar que estos dos tipos de dativo han de obedecer en SSG a procesos gramaticales diferentes.

- (22) a. El botones ha subido el equipaje

²⁶No es esta una nómina exhaustiva.

²⁷Los ejemplos son de la *Nueva gramática de la lengua española*.

- b. El guía ha dado propina

No obstante, obsérvese que los dos tipos de dativo contemplados pueden omitirse: (22). Dada esta posibilidad, en SSG, en la caracterización léxica de los verbos, no se explicita el dativo en ARG-ST. En cualquier caso, será una regla léxica la que añada el dativo a tal lista: *v-dative-c*. Además, en favor de la simplicidad computacional, se ha optado por utilizar la misma regla para el añadido de los dativos argumentales y no argumentales²⁸.

Además, no todos los verbos admiten con el mismo grado de facilidad los dativos no argumentales. De acuerdo con la *Nueva gramática de la lengua española*, el nivel de aceptación depende de condiciones semánticas, aspectuales o discursivas. Un verbo puede admitir mejor o peor la existencia de un participante directa o indirectamente afectado. Estas distinciones son sutiles: cuando el verbo admite con facilidad el dativo, este se interpreta normalmente como beneficiario o poseedor. Cuando no lo admite con facilidad su uso resulta marcado, pero suele ser posible como dativo ético. En SSG se ha optado por permitir la aplicación de *v-dative-c* a todo verbo. Se entiende como una idiosincrasia léxica el que tal dativo se interprete como de uno u otro tipo; y que se admita con mayor o menor facilidad.

Como puede verse en la figura 3.39, la regla *v-dative-c* es similar a las reglas de añadido de complemento vistas hasta el momento (*v-obliq-c* y *v-predicative-c*): toma un patrón de asignación de caso de los contemplados hasta ahora y le añade el dativo mediante la lista STOPPER. El papel temático de este argumento será el que cada verbo suponga al rasgo ARG4²⁹, con el cual se

²⁸No se da cuenta por tanto en SSG de los contrastes de (22). Para ello habría que utilizar dos reglas diferentes en función de que el dativo añadido sea de uno u otro tipo. Pero, en este punto, SSG ha primado la sencillez en la implementación y el ahorro computacional. Y se ha optado por contemplar una sola regla *v-dative-c*, que añade de igual modo a todo verbo un complemento dativo.

²⁹Los dativos propios de ciertos verbos de experiencia como *gustar*, *encantar*, *fastidiar*, etc. son experimentantes. Han de ser identificados, por tanto, con el rasgo ARG1 del verbo. Estos verbos son equivalentes, desde el punto de vista de estructura semántica, a los estados

identifica el índice del argumento.

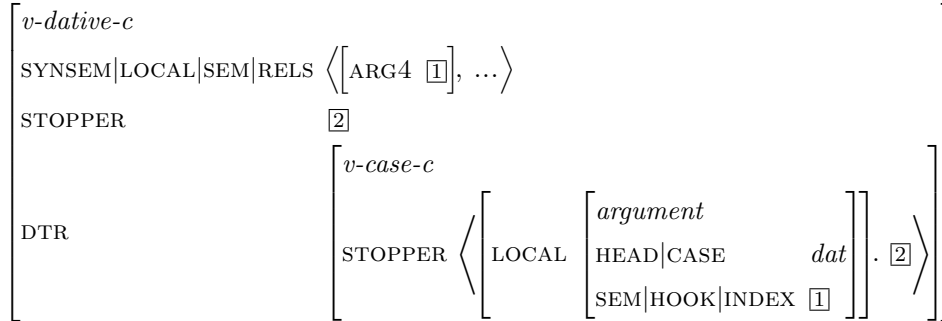


Figura 3.39: Patrón de añadido de argumento dativo

Se observa que este argumento en caso dativo se puede añadir a un verbo al que ya se haya añadido un complemento predicativo o argumento en caso oblicuo: (23-b) y (23-c).

- (23) a. El botones le ha subido el equipaje al guía
 b. El botones le ha subido el equipaje a la habitación al guía
 c. Ellos os han adecuado el hotel a la normativa.

Por tanto, *v-dative-c* debe poder aplicarse al resultado de una derivación que haya aplicado ya sobre el verbo, no solo un patrón básico de asignación de caso —*v-transitive-c* en (23-b) y (23-c)—, sino también *v-predicative-c* o *v-obliq-c*. Obviamente, también debe ser aplicable sobre el resultado de una derivación que solo haya aplicado el patrón básico de asignación de caso: *v-transitive-c* en (21-a).

transitivos cuyo sujeto es un experimentante, identificado con *ARG1* (*querer*, *preferir*, etc). Sería interesante que la gramática contemplase estos verbos como dos tipos de semántica idéntica, pero adscribibles a patrones de asignación de caso distintos. Para ello, sería necesario que el dativo experimentante de verbos como *gustar* fuese añadido al verbo mediante un patrón distinto de *v-dative-c*, que añada dativos beneficiarios (identificados con *ARG4*). Pero, en SSG, en favor de la simplicidad, se recogen esos dativos también mediante el patrón general *v-dative-c*.

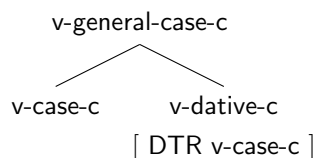


Figura 3.40: v-general-case-c: jerarquía de tipos

Para dar cuenta de esta situación, *v-dative-c* se ha declarado como subtipo de un supertipo general *v-general-case-c*, del cual hereda también *v-case-c*. Además, se establece en *v-dative-c* que el *input* debe ser de tipo *v-case-c*: supertipo que engloba tanto a *v-nominative-c* como a *v-pronominal-c* y *v-obliq-c*. En definitiva, la jerarquía de SSG debe modificarse conforme a lo que se muestra en la figura 3.40.

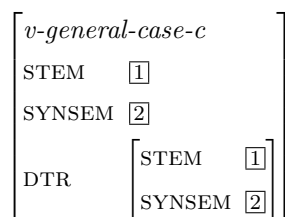


Figura 3.41: v-general-case

Como ya ocurrió en 3.5.2, una vez creado un nuevo supertipo de todas las reglas léxicas, es necesario colocar en él cierta caracterización que se había asignado anteriormente a uno de sus subtipos, en este caso, a *v-case-c*. La estructura de rasgos de *v-general-case-c* es, entonces, la que se muestra en la figura 3.41. Gracias a ello, también *v-dative-c* se concibe como una regla que asume la estructura de su *input* y la específica, pero nunca efectúa propiamente un cambio sobre ella.

3.6. Reglas léxicas no destructivas

Como se ha visto hasta ahora, un verbo, en la práctica, puede comportarse de muy diferentes formas. En HPSG se suele dar cuenta de este hecho suponiendo que el verbo en cuestión presenta, en el léxico, una caracteri-

zación que refleja su comportamiento canónico. El resto de posibilidades de comportamiento se modelan mediante reglas léxicas que modifican esa caracterización original. Un caso paradigmático es el tratamiento tradicional que se hace de los usos activo y pasivo de los verbos transitivos. Es comúnmente aceptado que el uso canónico de un verbo es el activo, y el pasivo es un uso derivado. Por tanto, un verbo transitivo se codifica normalmente en el léxico con su argumento externo como sujeto, en nominativo; y su argumento interno como objeto, en acusativo. Además, la gramática cuenta con una regla de pasivización que elimina el argumento externo y cambia el caso al interno. Llámese a este tipo de regla, regla destructiva. Son reglas que más allá de especificar la estructura sintáctica y semántica de su *input*, la modifican. Eliminan parte de su codificación explícita y la cambian por una alternativa.

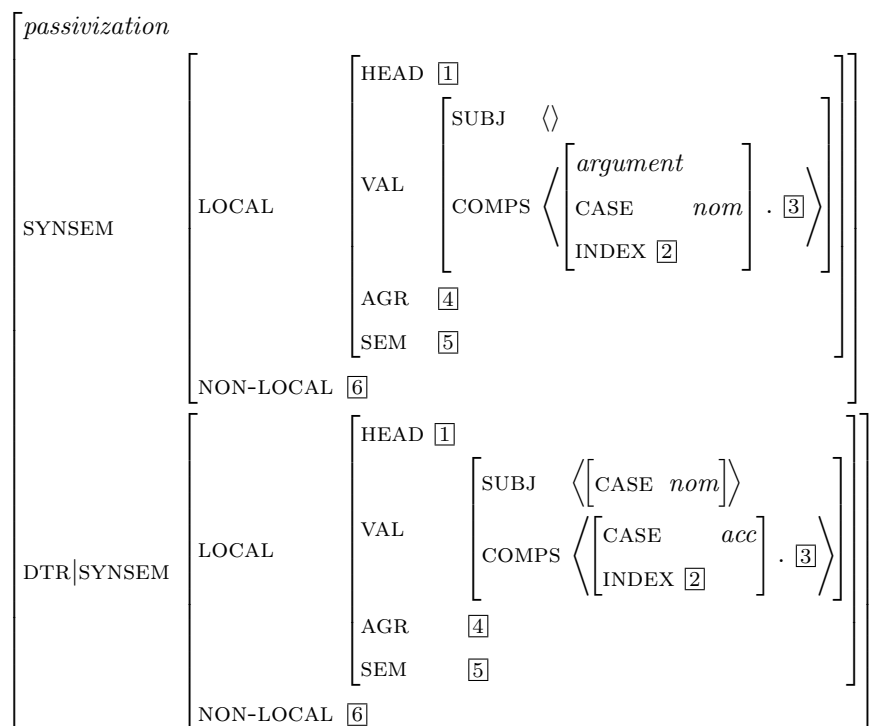


Figura 3.42: Regla destructiva de pasivización

En la práctica estas reglas son complicadas de implementar. Requieren una selección muy compleja de la información del *input* que pasa al *output*. Si

los rasgos del *input* que cambia la regla están muy anidados en su estructura, habrá que explicitar que el resto de rasgos hasta ese nivel pasan sin modificación al *output*. Tómese como ejemplo la regla hipotética *passivization* que se muestra en la figura 3.42. Se trata de una regla de pasivización que usa el método destructivo que se acaba de explicar. Esta regla debe vaciar en el *input* la lista SUBJ y debe cambiar el valor del rasgo CASE de su objeto de *acc* (acusativo) a *nom* (nominativo). Esos dos pequeños cambios requieren, como se observa, un gran número de identidades entre los rasgos del *input* y el *output*. Si no se explicitan estas identidades, todas las especificaciones presentes en los rasgos afectados del *input* se perderían en su paso al *output*.

Además, el hecho de que las reglas léxicas puedan cambiar su *input* de forma absolutamente libre no resulta elegante. Las reglas léxicas en HPSG no tienen la complejidad computacional de las reglas transformacionales: desde el punto de vista propiamente computacional, no son sino reglas de estructura sintagmática independientes del contexto. Sin embargo, esta libertad ilimitada para modificar el *input* sobre el que operan recuerda al estilo de las transformaciones de los comienzos de la gramática chomskyana. Una regla en aquella gramática podía directamente añadir argumentos, eliminarlos, permutar sus posiciones, etc. Mediante estas reglas se consiguió modelar muchos fenómenos de distintas lenguas. Pero para ello se multiplicaron los tipos de reglas: cada lengua, e incluso cada fenómeno en una misma lengua, contaba con sus reglas propias. Esto atentaba contra las aspiraciones de universalidad que tenía la gramática chomskyana. Y atentaba también contra las aspiraciones de generalidad de cualquier teoría científica. Por ello, aquella gramática evolucionó hacia lo que se llamó el Programa de Principios y Parámetros. En lo que a las transformaciones se refiere, todas ellas se redujeron a una única operación.

Es cierto que las reglas léxicas no acarrearán la complejidad computacional de las transformaciones. Y es cierto también que HPSG no tiene las mismas aspiraciones universalistas que la gramática chomskyana. Sin embargo, la total libertad con que modifican al *input* las reglas léxicas atenta, como lo hacían

las primeras transformaciones, contra las aspiraciones de generalidad de la teoría. Sería interesante restringir las posibilidades de estas reglas, reducirlas todas a un mismo esquema de funcionamiento.

HPSG es una teoría radicalmente lexicista. La sintaxis no consta, en esta teoría, sino de unas reglas muy simples, muy generales. Es en el léxico donde se recoge toda la complejidad de la gramática. Este se concibe como un sistema generativo (formado por las reglas léxicas) de las distintas unidades atómicas con las que opera la sintaxis. Lo que tradicionalmente se entiende como unidad léxica es un axioma de este sistema generativo, que da lugar a todas las posibilidades de realización de tal unidad léxica. En definitiva, las reglas léxicas forman un sistema que, a partir de las unidades léxicas, crean todas sus posibilidades de realización. Por ejemplo, a partir del verbo transitivo *contratar*, las reglas léxicas generan su versión en activa, en pasiva, con o sin argumento dativo, etc. Como se ha dicho, tradicionalmente, las gramáticas de tipo HPSG utilizan reglas destructivas: en la unidad léxica se codifica la realización canónica de la palabra, y las reglas modifican esa caracterización para dar cuenta de las alternativas no canónicas.

Imagínese, sin embargo, que se prescinde de la idea de realización canónica. Imagínese que, para un verbo transitivo, activo y pasivo son patrones de realización alternativos en pie de igualdad. En tal caso, no habría razón para codificar en la unidad léxica uno de estos patrones, y modificarlo con una regla para obtener el otro. En cambio, a la unidad léxica se le asignaría una caracterización compatible con ambos patrones, pendiente de ser especificada por una de las reglas léxicas que modela a estos. Esta es la idea directriz del sistema de reglas léxicas de SSG. En esta se usan exclusivamente reglas no destructivas. Estas reglas asumen la estructura del *input* y le añaden una caracterización compatible, unificable, con aquella.

Esto, por un lado, simplifica mucho la implementación de las reglas. El flujo de información del *input* al *output* de todas las reglas aplicables a una palabra sigue una lógica muy simple: toda la información del *input* aparece

en el *output*: no se cambia nada. Esto hace de la gramática un sistema sencillo, fácil de mantener y ampliar. Y, desde un punto de vista teórico, las reglas léxicas no se asemejan ya a las citadas transformaciones. Son, en cambio, meros tipos que modelan patrones de realización concretos. Son tipos que modelan realidades de la gramática. Las unidades léxicas necesitan una caracterización en cuanto a ciertos criterios: necesitan asignar caso a sus argumentos estructurales, concretar su lista de argumentos, concretar la categoría gramatical de estos, etc. Las reglas léxicas les aportan esta caracterización. Estas dejan de operar cambios sobre las palabras, para modelar ciertas realidades gramaticales de las que pueden participar tales palabras.

El contemplar las reglas léxicas como patrones no destructivos neutraliza las infinitas posibilidades de comportamiento de las reglas destructivas en uno solo: el añadido de especificidad; o, dicho de otro modo, la adscripción de la unidad léxica a un patrón regular de comportamiento, con el que es compatible. La existencia en la gramática de un patrón de comportamiento debe estar justificada con independencia de que se conciba o no como regla léxica. Asumido esto, efectivamente, la regla léxica no destructiva, en tanto que tal, siempre tiene la misma función: adscribir la unidad léxica a uno u otro patrón.

3.7. Alternancias de diátesis con reglas no destructivas

Se ha visto hasta el momento que los argumentos de un verbo pueden contar con un amplio abanico de posibilidades de realización: un mismo verbo puede contar con más o menos argumentos, o con argumentos de uno u otro tipo; en voz activa, pasiva, etc. Por ejemplo, un verbo como *jugar* admite, al menos, solo un sujeto —(24-a)— o un sujeto y un objeto —(24-b)—.

- (24) a. ¿Puedo jugar cerca del hotel?
b. ¿Puedo jugar a los bolos cerca del hotel?

La gramática ha de ser capaz de dar cuenta de los dos usos y adscribirlos a la misma unidad léxica. Se ha de evitar, en favor de la generalidad y la sencillez del lexicón, duplicar las entradas léxicas para dar cuenta de sus distintas posibilidades de realización. Por tanto, las distintas posibilidades de realización del verbo *jugar* en (24) han de generarse en la gramática mediante reglas léxicas aplicadas a una misma entrada. Si se usa, para ello, un sistema de reglas no destructivo, hay que hacer frente a ciertos problemas.

De acuerdo con lo explicado en 3.6, con un sistema de reglas no destructivo, una unidad léxica ha de estar lo suficientemente subespecificada como para ser directamente unificable con todos los patrones de comportamiento que le son propios. Esto se debe a que las reglas no destructivas asumen la caracterización del *input*, y solo le añaden especificidad; nunca efectúan sobre él cambios: no eliminan argumentos, ni los cambian de orden, etc. Por tanto, es necesario que, en el lexicón, *jugar* no explicita su objeto. Si lo explicitara para dar cuenta de (24-b), ninguna regla podría después eliminarlo para dar cuenta de (24-a)³⁰.

Pero, al tiempo, toda unidad léxica ha de estar lo suficientemente especificada como para evitar que se le apliquen reglas que modelen patrones que no le son propios. Estos dos requerimientos suelen ser difíciles de compaginar. Por ejemplo, se ha dicho que no es posible explicitar el objeto de (24-b) en la caracterización léxica del verbo *jugar*, para que este sea compatible con el uso de (24-a). Pero si no se explicita dicho objeto, ¿cómo se especifica cuál ha de ser su naturaleza, para aquellos casos en que el verbo sí cuente con él? ¿Cómo se logra, por ejemplo, que, en caso de aparecer el objeto de *jugar*, este reciba caso oblicuo y no acusativo, como parece necesario?

³⁰En 3.3 se mostraron reglas (pasivas) que eliminaban argumentos de ARG-ST. Es un concepto de eliminación distinto. Esas reglas eliminan (dentro del mismo *input*) un elemento en su paso de ARG-ST a SUBJ o COMPS; no en su paso del *input* al *output*. Lo que no puede hacer una regla no destructiva es tomar un elemento, por ejemplo de la lista COMPS del *input*, y eliminarlo en la lista COMPS del *output*.

En los siguientes apartados se muestra cómo el sistema de reglas no destructivas de SSG ha conseguido compaginar satisfactoriamente estos dos requerimientos. Con las nueve reglas léxicas mostradas hasta el momento, se da cuenta del grueso de las alternancias de diátesis de los verbos del español. Para ello, de nuevo, se utiliza como guía la clasificación clásica de Levin y Rappaport (1998).

3.7.1. Realizaciones y logros

Las realizaciones se han recogido en SSG en el tipo *accomplishment*. Como se vio en 3.2, cuentan con dos argumentos estructurales en ARG-ST: uno externo y otro interno. Ambos tienen naturaleza nominal. La presencia de estos argumentos en el verbo se recoge evaluando tanto EARG como IARG como *index*. Esta caracterización hace a cualquier instancia léxica de *accomplishment* incompatible con los patrones de asignación de caso intransitivos: *v-unergative-c*, *v-unaccusative-c* y *v-pronominal-c*. Véase (25-d). En cambio, no hay nada que haga incompatible a cualquier verbo de este tipo con *v-transitive-c* —(25-a)—, *v-passive-c* —(25-b)— o *v-rpass-c* —(25-c)—.

- (25) a. El hotel ha contratado al guía para la excursión
 b. Ha sido contratado un guía para la excursión
 c. Se han contratado guías para la excursión
 d. *El hotel ha contratado

Además, las realizaciones que hayan sido caracterizadas como *CARG index* son compatibles con la regla de añadido de un complemento predicativo: *v-predicative-c*. Es el caso de *subir* en (26-a). La estructura de rasgos de *subir* debe incluir también identidad entre CARG e IARG, de modo que el predicativo en cuestión predique sobre el argumento interno del verbo.

- (26) a. El botones ha subido el equipaje a la habitación
 b. Ellos han adecuado el hotel a la normativa

Del mismo modo, las realizaciones que contemplen la existencia de un segundo objeto con rol temático de tema no afectado (además del objeto paciente estructural) deben ser caracterizadas como ARG3 *index*. Estos verbos podrán participar de *v-obliq-c* y contar, así, con un tercer argumento de caso oblicuo —(26-b)—.

Por último, para cualquier realización, sea cual sea su patrón de comportamiento, es posible el añadido de un dativo —(27) y (28)—, mediante la aplicación de la regla *v-dative-c*.

- (27) a. El hotel nos ha contratado al guía para la excursión
 b. Nos ha sido contratado un guía para la excursión
 c. Se nos han contratado guías para la excursión
- (28) a. El botones os ha subido el equipaje a la habitación
 b. Ellos os han adecuado el hotel a la normativa

Los logros, por su parte, se han recogido en SSG en el tipo *achievement*. Cuentan con un único argumento estructural, interno, y necesariamente nominal: (29-a). Dada la caracterización consecuyente (EARG *no-individual* e IARG *index*), son compatibles únicamente con el patrón de asignación de caso *v-unaccusative-c*, que supone un único argumento estructural, interno. Son frecuentes los logros con significado de movimiento. Estos verbos suelen contar, además, con un complemento predicativo que predica sobre el argumento interno, e indica la dirección, el origen, etc. del movimiento que sufre este: (29-b). Estos verbos deben estar caracterizados como CARG *index*, y deben identificar CARG con IARG. Además, también para todo logro se supone la posibilidad de añadido de un argumento en dativo, ya sea junto al predicativo o no: (29-c).

- (29) a. Ha subido el precio de las habitaciones
 b. Ha subido el precio de las habitaciones a cien euros
 c. Nos ha subido el precio de las habitaciones a cien euros

En resumen, basta la caracterización semántica recogida, fundamentalmente, en el rasgo HOOK de estos verbos, para regular su comportamiento sintáctico: EARG e IARG determinan si el verbo tiene o no argumento externo e interno. Los patrones de asignación de caso transitivo e intransitivo piden, precisamente, verbos con una determinada caracterización en cuanto a la presencia de esos argumentos en HOOK. La posibilidad de añadido de argumento oblicuo se regula también mediante un rasgo semántico: ARG3, que determina si al verbo se le puede añadir un objeto con rol temático de tema no afectado. Y la presencia de predicativo la regula la existencia o no de un índice en CARG (el lugar en el que el verbo publica el objetivo de una posible predicación). Por su parte, el dativo tiene una aplicación universal.

Son todos rasgos que caracterizan al signo de forma lingüísticamente motivada. Es una caracterización semántica, cuya existencia está, entonces, justificada, con independencia del uso sintáctico que se pueda hacer de ella. Pero, de hecho, los procesos sintácticos pueden regularse satisfactoriamente haciendo uso de esa caracterización. La estrategia que usa SSG consiste, en definitiva, en regular la aplicación de procesos sintácticos usando como referencia la caracterización semántica del signo. Esto permite mantener subespecificada su dimensión sintáctica. Gracias a ello, es posible el uso de un sistema de reglas léxicas no destructivas: las unidades léxicas son, al tiempo, compatibles con todos los patrones sintácticos de los que participan; pero suficientemente específicas como para evitar patrones que no les son propios.

3.7.2. Actividades y estados

Las actividades y los estados cuentan con un único argumento estructural. Las actividades se han recogido en el tipo *activity*, y los estados en el tipo *state*. Ese único argumento estructural, en las actividades, es externo (EARG index, pero IARG no-individual). Esto las hace compatibles con *v-unergative-c*, pero no con *v-unaccusative-c*. En cambio, el tipo *state* no especifica si su único argumento estructural es interno o externo, de modo que habrá estados que

participen de *v-unergative-c* (*ver, saber, tener, etc.*) y otros que lo hagan de *v-unaccusative-c* (*sucedec, ocurrir, etc.*). En cualquier caso, estados y actividades —a diferencia de las realizaciones: (30-e)— admiten usos intransitivos: (30).

- (30) a. Yo confío en eso. ¿Tú confías?
 b. Yo no hablo de eso. ¿Tú hablas?
 c. Yo no sé de eso. ¿Tú sabes?
 d. Yo no tengo eso. ¿Tú tienes?
 e. Yo no he roto eso. *¿Tú has roto?

Además, las actividades y los estados con argumento externo, como ya se ha visto en 3.5, admiten usos muy parecidos a los de las realizaciones: pueden añadir al sujeto estructural un objeto. Pero este objeto está legitimado por el significado idiosincrásico del verbo, y recibe un papel temático de tema no afectado. Además, la naturaleza de este objeto es mucho más variable de lo que pudiera ser la del objeto de una realización: puede recibir caso oblicuo o acusativo, y puede tener naturaleza nominal o verbal. Véanse los ejemplos de (31).

- (31) a. ¿Confías en que baje el precio?
 b. ¿Hablas en inglés?
 c. ¿Sabes que han subido el precio de las habitaciones?
 d. ¿Tienes las entradas?

La caracterización del objeto de una realización como elemento necesariamente nominal es sencilla de establecer, pues este argumento sí se explicita en *accomplishment*. En cambio, los objetos no estructurales de (31) no se explicitan en las entradas léxicas. Los objetos oblicuos de (31-a) y (31-b) solo han sido explicitados en la regla *v-obliq-c* que los añade. Y los acusativos de (31-c) y (31-d) se explicitan por primera vez en la regla *v-transitive-c*. La estructura de rasgos que modela a estos objetos en tales reglas no especifica su naturaleza nominal o verbal. Además, dado que las actividades y los estados no explicitan en su caracterización el objeto, no es posible establecer direc-

tamente si este ha de ser acusativo —aportado por *v-transitive-c*— u oblicuo —aportado por *v-obliq-c*—.

Sin embargo, es obvio que estos verbos, en caso de presentar objeto, establecen ciertas restricciones sobre su naturaleza categorial y su caso: el verbo *confiar* no admite que su objeto tenga caso acusativo, ni *saber* o *tener* admiten un objeto con caso oblicuo; de igual modo, no parece natural que el objeto de verbos como *tener* tenga naturaleza verbal. Véanse estas posibilidades en (32).

- (32) a. ¿Tienes las entradas?
 b. *¿Tienes de las entradas?
 c. *Tengo que venga
 d. ¿Puedo confiar en ti?
 e. *¿Puedo confiar a ti?
 f. Confío en que venga

¿Tiene SSG capacidad expresiva suficiente para establecer adecuadamente estas restricciones? Por un lado, puede establecerse para cada verbo si el índice semántico del argumento en cuestión es necesariamente nominal (*index*) o no (*individual*). Esta especificación puede colocarse en el valor del rasgo ARG3 del verbo, con el cual se identifica el índice semántico de los objetos no estructurales (en tanto que temas no afectados). Por su parte, el que estos objetos puedan recibir uno u otro caso se regula mediante los rasgos INTR y OKEY —véase 3.5.2—. Los verbos que, como *confiar*, no admiten objetos en acusativo deben ser caracterizados como INTR *yes* (como incapaces de asignar caso acusativo). Esto los hace incompatibles con el patrón *v-transitive-c*. Además, los verbos que, como *tener* o *saber*, no admiten objeto en caso oblicuo se caracterizan como OKEY *null* (como carentes de preposición de rección para el objeto). Esta caracterización los hace incompatibles con el patrón *v-obliq-c*.

Como se ha dicho en 3.7.1, en SSG se ha regulado el comportamiento sintáctico de los verbos haciendo uso de caracterizaciones semánticas in-

dependientes. En cambio, INTR y OKEY son rasgos de tipo sintáctico. Esta asimetría está justificada, pues la asignación de uno u otro caso al objeto es una realidad rigurosamente sintáctica. No hubiese sido adecuado restringir también este aspecto desde la semántica. Cualquier rasgo que se hubiese propuesto en SEM para dar cuenta de ello hubiese sido una estipulación injustificada. En cambio, INTR y OKEY, rasgos sintácticos, sí tienen razón de ser. El que un verbo esté o no capacitado para asignar acusativo es una distinción fundamental de la sintaxis de estos, que aquí se recoge mediante INTR. Y el que un verbo con objeto oblicuo rija una u otra preposición es una realidad idiosincrásica del verbo que hay que codificar. La falta de esa preposición es el modo natural de establecer que el patrón de asignación de oblicuo no es propio del verbo en cuestión.

Además, en principio, los verbos que participan del patrón *v-transitive-c* admiten también el paso por las construcciones pasivas: *v-passive-c* y *v-rpass-c*. En cambio, a los verbos ajenos al patrón transitivo, no les son propias las construcciones pasivas. Por tanto, *v-passive-c* y *v-rpass-c*, al igual que *v-transitive-c*, han sido caracterizadas como INTR no; de modo que son incompatibles con los verbos que no asignan acusativo (INTR yes). Con tal caracterización, SSG da cuenta adecuadamente de los contrastes de (33).

- (33) a. ¿Se habla inglés?
 b. ¿Se sabe el precio de las habitaciones?
 c. *¿Se confía él?

Pero téngase en cuenta que, como se vio en 3.3, *v-passive-c* solo admite verbos con paciente afectado. Por ello, *v-passive-c* se ha caracterizado como ARG2 index. Las actividades y los estados no tienen paciente afectado —son ARG2 no-individual—. Por tanto, estos verbos no pueden participar de *v-passive-c*.

Por último, la regla *v-dative-c* es aplicable —como lo fuera para las realizaciones y los logros— también a toda actividad o estado. Véanse los ejemplos de (34) y (35).

- (34) a. ¿El personal del hotel os habla inglés?
 b. ¿Os puedo hablar en inglés?
 c. ¿Nos tienes las entradas?
- (35) a. ¿Puedes confiarme en él?
 b. El hotel se me desentiende de la situación
 c. Los turistas se os pueden personar en la oficina de turismo
 d. ¿Cómo se me puede comunicar con el personal del hotel?

Obviamente, no todos los verbos admiten el dativo del mismo modo. Ya se explicó esta situación en 3.5.4. Los usos de (34) resultan mucho más naturales que los de (35). No obstante, los dativos de (35) sí son posibles como dativos éticos, si denotan un elemento, alejado de la acción verbal, pero afectado en algún sentido por ella. SSG no trata de forma distinta estos dos tipos de argumento. La gramática, sencillamente, permite el añadido del argumento dativo. Se confía en que un estudio léxico posterior pudiera discriminar, para cada verbo, qué naturaleza semántica tiene tal elemento. En cualquier caso, sería posible regular desde el léxico qué verbos admiten o no un argumento dativo: basta establecer adecuadamente el valor de ARG4, rasgo identificador del índice adscrito al rol temático propio de estos argumentos (beneficiario, poseedor, destinatario, etc.).

3.7.3. Alternancia acusativo oblicuo

Se ha visto en 3.7.2 cómo existen estados y actividades con argumento externo e interno. Ese argumento interno, no estructural, puede recibir caso acusativo u oblicuo. Por ejemplo, *tener* le asigna caso acusativo a su objeto, mientras que el objeto de *confiar* recibe caso oblicuo. Pero son muy comunes, además, los verbos cuyo objeto puede recibir tanto caso acusativo como oblicuo. Este es el caso del verbo *hablar*, como se observa en los ejemplos de (36).

- (36) a. ¿El personal del hotel habla inglés?
 b. ¿Puedo hablar en inglés con el personal del hotel?

De esta misma alternancia participan también el verbo *saber* ('saber algo' frente a 'saber de algo'), *disfrutar* ('disfrutar algo' frente a 'disfrutar de algo'), *acertar* ('acertar algo' frente a 'acertar con algo'), *consultar* ('consultar a alguien' frente a 'consultar con alguien') y *cuidar* ('cuidar a alguien' frente a 'cuidar de alguien'), entre otros³¹.

¿Cómo dar cuenta de estos datos con los recursos expresivos de SSG? Haciendo uso de los rasgos INTR y OKEY. Un verbo que admite objeto en oblicuo debe especificar la preposición que marca el caso de tal objeto en la lista OKEY. Esta mera especificación lo habilita para participar del patrón v-obliq-c. Si además está caracterizado como capaz de asignar caso acusativo (INTR no), es al tiempo compatible con el patrón transitivo v-transitive-c. De acuerdo con ello, el verbo *hablar*, por ejemplo, para dar cuenta de los usos de (36), debe caracterizarse como OKEY < en > y como INTR no. En cambio, un verbo solo capaz de asignar acusativo a su objeto (*tener*) debe ser OKEY null e INTR no. Un verbo como *confiar*, por último, para ser solo compatible con el patrón de asignación de caso oblicuo, debe caracterizarse como OKEY < en > e INTR yes.

3.7.4. Alternancia causativa

Hasta el momento, se ha entendido que todo verbo debe codificarse como instancia de uno de los cuatro grandes grupos verbales de 3.2: realizaciones, logros, actividades o estados.

Pero hay datos del español que sugieren que estas grandes clases pudieran entenderse también como distintos patrones de realización aplicables a un mismo verbo. Por ejemplo, es frecuente que un cambio de estado admita

³¹*Nueva gramática de la lengua española*, Real Academia Española, Espasa (2009), II, pp. 2724-2726. Esta gramática contempla otros dos tipos de alternancia parecidos al aquí mostrado. Una es la propia de los verbos cuya versión intransitiva tiene carácter pronominal: se verá en 3.8. La otra es aquella de la que participan los verbos transitivos pronominales. Estos verbos, como se ha dicho anteriormente, se obvian en SSG.

tanto un uso causativo como un uso no causativo: funcione, en definitiva, como realización o como logro. Es lo que ocurre con el verbo *subir*, como ya se vio en 2.1. Este verbo denota en cualquier caso un cambio de estado: un paciente sufre un cambio de posición. Es posible que no exista una causa externa de tal cambio, como ocurre en (37-a). Pero también es posible que sí exista tal causa externa: (37-b). Es la misma alternancia que se da entre unidades léxicas diferentes como *morir* y *matar*.

- (37) a. Ha subido el precio de las habitaciones
 b. El botones ha subido el equipaje

Otro tanto ocurre con verbos como *bajar*, *abrir*, *cerrar*, *empezar*, etc. Una solución adecuada para todos ellos sería entender **accomplishment** y **achievement** (y también **state** y **activity**³²) como reglas: patrones alternativos aplicables a una misma entrada léxica. Si se caracterizara a estos verbos como **BECOME yes**, participarían de **accomplishment** y de **achievement**, pero no de **state** o **activity**.

SSG es perfectamente compatible con esta solución. Pero no se ha utilizado, pues complicaría el sistema de reglas léxicas, y aumentaría el tiempo de procesamiento. En cambio, estos verbos se han caracterizado como instancias del tipo **become-verb-lex**, representado en la figura 3.43. Este tipo solo especifica al verbo como cambio de estado (**BECOME yes**), lo cual implica la presencia de un argumento interno (IARG) cuyo rol temático es de paciente afectado (ARG2). Esto lo habilita para la participación de los patrones transitivo, inacusativo y pasivo. Pero impide su aparición en los patrones intransitivos inergativo o pronominal.

³²Por ejemplo, hay ciertos verbos de movimiento que pueden funcionar como agentivos o no, de acuerdo con la naturaleza semántica de su sujeto —véase (I)—. El verbo *girar*, en (I-a), parece funcionar como una actividad, pero en (I-b) parece un logro.

- (I) a. Juan, conscientemente, giró a la derecha
 b. La barca, a la deriva, giró a la derecha

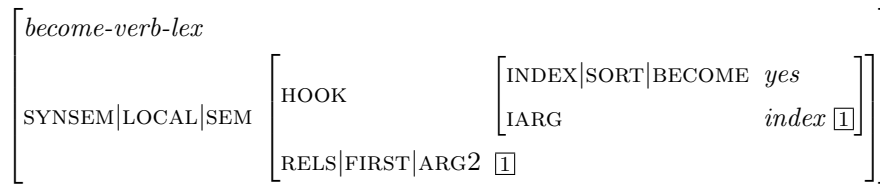


Figura 3.43: become-verb-lex

En el siguiente apartado 3.8 se completa esta discusión mostrando una versión de la alternancia causativa en la que el uso intransitivo es pronominal.

3.8. Alternancias con verbos pronominales

El que un verbo tenga o no carácter pronominal es una idiosincrasia léxica. Por tanto, se debe marcar, de forma idiosincrásica, mediante el rasgo PARG: un verbo que no admita uso pronominal debe ser PARG *no-individual*, y un verbo que sí lo admita debe ser PARG *index*. Esta caracterización es suficiente para evitar que los verbos no pronominales participen del patrón intransitivo *v-pronominal-c*, pues este pide verbos PARG *index*. Por su parte, los patrones intransitivos no pronominales deben pedir verbos PARG *no-individual*, para evitar que un verbo pronominal, en su versión intransitiva, pase por ellos.

Hay algunos verbos pronominales que solo admiten usos intransitivos. Estos pueden contar con un objeto, que será oblicuo —oraciones de (38)— o predicativo locativo —oraciones de (39)—, pero nunca acusativo.

- (38) a. El hotel se desentiende de la situación
 b. *El hotel desentiende la situación
 c. *El hotel se desentiende la situación
 d. *El hotel desentiende de la situación
- (39) a. Los turistas pueden personarse en la oficina de turismo a las 12:00
 b. *Los turistas pueden personar en la oficina de turismo a las 12:00

Estos verbos deben caracterizarse como INTR *yes* (lo cual evita que participen de los patrones transitivo y pasivos) y como PARG *index* (lo cual obliga a que el patrón intransitivo del que participen sea el pronominal).

Pero también existen verbos pronominales que admiten tanto uso intransitivo con objeto en oblicuo como uso transitivo. Estos verbos solo son pronominales en su versión intransitiva. Ya se vio en 3.7.3, para verbos no pronominales como *jugar*, una alternancia similar. Son verbos como *lamentar* (‘lamentar algo’ y ‘lamentarse de algo’), *olvidar* (‘olvidar algo’ y ‘olvidarse de algo’), *admirar* (‘admirar algo’ y ‘admirarse de algo’), etc. Véanse los ejemplos de (40) —tomados del apartado 3.4—.

- (40) a. Admiraba la belleza del paisaje
 b. Se admiraba de la belleza del paisaje

Estos verbos del tipo de *admirar* deben codificarse como INTR *no* y PARG *index*; lo cual obliga a que sus usos intransitivos sean pronominales, pero admite al tiempo usos transitivos (obligatoriamente no pronominales; pues, en SSG, la transitividad y el carácter pronominal son excluyentes). Por contra, los verbos no pronominales que alternan entre objeto acusativo y oblicuo (*hablar* en 3.7.3) deben ser INTR *no* (como se vio entonces), pero también PARG *no-individual*; para evitar que su uso intransitivo sea pronominal.

En el apartado 3.7.4 se mostró cómo ciertos verbos de cambio de estado (como *subir*) participaban de la llamada alternancia causativa. En este apartado se mostrará la misma alternancia para verbos que, en su versión intransitiva, tienen carácter pronominal: *secar* y *secarse*, *acabar* y *acabarse*, *acostar* y *acostarse*, por ejemplo³³. Como ya se vio en 3.4, *secar* y *secarse* tienen un mismo significado idiosincrásico; pero su versión transitiva es causativa (‘alguien seca algo’), mientras que la pronominal no lo es (‘algo se seca’). La alternancia causativa se estableció en 3.7.4 para verbos de cambio

³³*Nueva gramática de la lengua española*, Real Academia Española, Espasa (2009), II, pp. 3107-3108.

de estado no especificados en cuanto a su carácter transitivo o intransitivo: *become-verb-lex*. Visto que hay verbos partícipes de esta alternancia para los cuales la versión intransitiva es pronominal (*secarse*), y otros para los cuales no lo es (*subir*), las instancias de *become-verb-lex* deben ser especificadas en cuanto a este particular (PARG index o no-individual, respectivamente). Esta especificación independiente a la naturaleza semántica del verbo regulará el que su versión intransitiva sea una instancia del patrón no pronominal *v-unaccusative-c* o del pronominal *v-pronominal-c*.

3.9. Conclusión

En este apartado se ha mostrado cómo SSG cuenta con un sencillo sistema de reglas léxicas que da cuenta de las alternancias de diátesis de los verbos del español. La cobertura de este sistema no es exhaustiva, pero sí cubre el grueso de los datos, con lo cual se ha demostrado que es un sistema viable. Este sistema de reglas es novedoso e interesante por varios motivos.

En primer lugar, sus reglas son no destructivas: nunca alteran el *input* que reciben, solo lo especifican. Con ello se consigue una implementación clara y sencilla, fácil de mantener y ampliar. Además, conceptualmente, las reglas léxicas no destructivas son una realidad muy sencilla: son meras descripciones de patrones directamente adscribibles a las unidades léxicas; no operaciones que las alteran, sino patrones que las especifican. Un sistema de reglas léxicas tradicional es mucho más complejo desde un punto de vista práctico y conceptual: cada regla puede modificar el *input* prácticamente de cualquier modo.

En segundo lugar, SSG cuenta con un tipo de regla que añade un complemento al *input*, sea cual sea el número de complementos original de tal *input*. Este sistema reduce sensiblemente el número de reglas de la gramática. Normalmente, una gramática usa distintas reglas para añadir el mismo tipo de complemento: una lo añade como segundo complemento, otra lo añade como

tercero, etc. SSG neutraliza esas reglas en una sola, con lo cual la gramática gana en sencillez y elegancia conceptual.

Por último, SSG hace un uso crucial del concepto de argumento. Se ha definido mediante un único tipo qué puede ser un argumento (también se han definido los conceptos de argumento interno, externo y complemento predicativo). Gracias a ello, los patrones de comportamiento verbal contemplados en SSG no especifican la naturaleza categorial de sus argumentos, solo su caso y quizá su semántica. Se entiende que, en las posiciones argumentales, solo podrán aparecer elementos compatibles con la caracterización general de lo que es un argumento, y con las restricciones que imponga el caso a dicha posición.

Los patrones de comportamiento intransitivo, transitivo, pasivo, de pasiva refleja y pronominal se han entendido en SSG como el cruce de dos criterios: cada patrón es, por un lado, un modo de asignación de caso a los argumentos estructurales del verbo; y, además, un modo de proyección de esos argumentos desde la lista ARG-ST a las listas SUBJ y COMPS. Los patrones inergativo e inacusativo se han formalizado como distintos modos de proyección de los argumentos de ARG-ST a SUBJ y COMPS. Dada esta formalización, ha resultado natural entender que los usos pasivos obedecen a un tercer patrón de relación entre listas: este omite el argumento externo de ciertos verbos. El patrón pronominal, dada su procedencia histórica, se ha podido entender de modo similar. Nótese la elegancia conceptual de entender estos patrones como diferentes posibilidades de una misma realidad. A estos patrones hay que sumar aquellos que añaden a los argumentos estructurales un argumento no estructural (un objeto oblicuo, un complemento predicativo o un dativo).

El reto fundamental al que se enfrenta un sistema de reglas léxicas no destructivas, como el de SSG, es regular a qué unidades léxicas se puede aplicar cada patrón. Para lograrlo, SSG ha dotado al verbo de una rica caracterización semántica, consistente en la publicación de los índices de sus argumentos en el rasgo HOOK. Esta caracterización permite discriminar qué tipo

de argumentos le son propios a cada verbo. Además, se han creado dos rasgos sintácticos que regulan el caso que puede recibir el objeto de cada verbo. Toda esta caracterización es independiente de la aparición explícita de los argumentos implicados, por lo cual, es compatible con patrones sintácticos quizá incompatibles entre sí. Además, se ha defendido que es una caracterización lingüísticamente motivada.

Cabe terminar apuntando que esta buena gestión de las alternancias de diátesis del español es útil desde un punto de vista computacional. Con ella se ha conseguido un sistema de solo nueve reglas léxicas que da cuenta del grueso de los datos pertinentes. Esta sencillez reduce sensiblemente la explosión combinatoria, y con ella, el coste computacional.

Capítulo 4

SSG: clíticos

En el capítulo 3 se explicó cómo SSG da cuenta, con un sistema sencillo de nueve reglas léxicas no destructivas, del grueso de las posibilidades de realización de los verbos del español. Las gramáticas previas necesitaban muchas más reglas para dar cuenta de los mismos datos. La proliferación de reglas léxicas es un serio problema para la viabilidad computacional de una gramática. Por tanto, SSG es una buena solución desde el punto de vista del coste computacional. En este capítulo se completa esa discusión introduciendo un fenómeno central en las lenguas románicas: la cliticización de los argumentos. Este fenómeno, en definitiva, es un factor más de variación para el comportamiento de un verbo. Por tanto, en principio, al ser modelado en una gramática computacional, acarreará al análisis un coste añadido. En este capítulo se muestra cómo el análisis que de los clíticos del español hace SSG supera a los análisis previos al evitar absolutamente el uso de reglas. La eliminación de las reglas que, en otras gramáticas, regulaban este fenómeno supondrá también un ahorro computacional.

En el apartado 4.1 se explican ciertos aspectos técnicos sobre la formalización del signo lingüístico en SSG. Estos son interesantes para la comprensión del resto de apartados. En el apartado 4.2 se muestran los datos básicos a los que debe hacer frente SSG. En 4.3 se explican las soluciones habituales en los trabajos teóricos y gramáticas computacionales previas. En 4.4 se expone el

análisis que hace SSG del sistema de clíticos del español; y en 4.5 se muestran ciertos datos problemáticos para este análisis.

4.1. Revisión del signo lingüístico (II)

El fenómeno de los clíticos, como se verá, es, en cierto modo, un fenómeno no local. Este es el primer fenómeno no local que se formaliza en SSG. Por tanto, será este el momento adecuado para caracterizar al rasgo NON-LOCAL del signo lingüístico. De ello se ocupa el apartado 4.1.1. Además, la aparición de clíticos está muy ligada a la elisión de argumentos, o a su desplazamiento a la margen izquierda de la oración. Por ello, en 4.1.2 se explica cómo se saturan en SSG los elementos de COMPS que se realizan de este modo.

4.1.1. El rasgo NON-LOCAL

En el apartado 1.1.4.3 se explicó cómo cualquier gramática de una lengua natural debe enfrentarse al modelado de relaciones no locales. Efectivamente, dos constituyentes de una secuencia pueden estar relacionados, aun sin ser adyacentes: pueden mantener relaciones de dependencia estructural; o, sencillamente, la naturaleza de uno puede condicionar la del otro.

Atiéndase a los siguientes ejemplos de relaciones no locales. En todas las oraciones de (1), el objeto de *visitemos* se interpreta en la proyección 1, pero se explicita lejos de esta, en la proyección 4. En (1-a) y en (1-c) el movimiento está legitimado por la naturaleza interrogativa del objeto; y en (1-b) y en (1-d) por ser este un pronombre relativo. Pero, tanto en (1-c) como en (1-d), las palabras que confieren a los objetos esa naturaleza interrogativa o relativa se encuentran anidadas en proyecciones internas al objeto. Por tanto, para que, en estos casos, la proyección 2 sea un objeto desplazado legítimo; es necesario que la naturaleza interrogativa o relativa de *qué* y *que* trascienda su dominio local y se manifieste en la proyección 2. Son estas, entonces, manifestaciones no locales de la naturaleza de estas palabras.

- (1) a. ¿ [4 Qué_i [3 quieres [2 que [1 visitemos \mathcal{O}_i]]]] ?
 b. [La ciudad_i [4 que_i [3 quieres [2 que [1 visitemos \mathcal{O}_i]]]]]
 c. ¿ [[2 A [1 qué_i mujer]] [quieres [que [visitemos \mathcal{O}_i]]]] ?
 d. [La mujer_i [[2 a [1 la que_i]] [quieres [que [visitemos \mathcal{O}_i]]]]]

Ya se avanzó en 1.1.4.3 cómo afronta una gramática de unificación el tratamiento de estos fenómenos. Recuérdese que estas gramáticas, a diferencia de las transformacionales, solo entienden como objeto gramatical a cada proyección en sí, no a la estructura compleja que incluye a la proyección y a sus constituyentes. Por tanto, para una gramática de unificación, las estructuras de corchetes de (1) solo son representaciones del proceso derivativo, nunca entes gramaticales sobre los que se pueda operar. Estas gramáticas, entonces, no pueden regular directamente las relaciones no locales de (1), pues no contemplan la existencia de un objeto gramatical que incluya a los dos elementos afectados. Por ejemplo, en (1-a) no pueden saturar directamente el objeto de *visitemos* en 1 mediante la unión de *Qué* a 4. Una gramática transformacional, en cambio, al contemplar toda esa estructura encorchetada como un ente gramatical, sí puede.

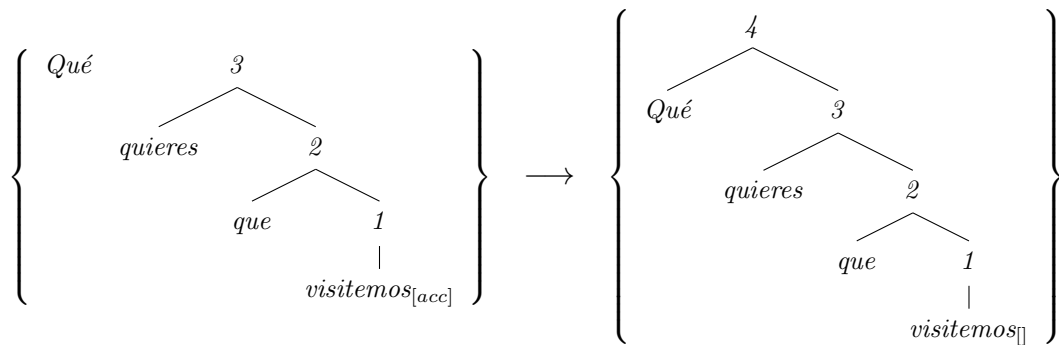


Figura 4.1: Regla transformacional

Véase en la figura 4.1 cómo una regla transformacional puede saturar la necesidad argumental de *visitemos_[acc]* en 1, mediante la unión de *Qué* a 3: *visitemos_[]*. La clave está en que esta regla toma como *input* la estructura de constituyentes completa, y puede operar sobre ella. La regla equivalente

en una gramática no transformacional, en cambio, solo operaría sobre la proyección \mathcal{P} misma. Véase en la figura 4.2. Esta regla no tiene información, en principio, sobre las necesidades argumentales presentes en una proyección sobre la que no opera: 1 .

$$\{ \textit{Qué } \mathcal{P} \} \longrightarrow \{ 4 \}$$

Figura 4.2: Regla no transformacional

Para solventar esta limitación, en una gramática de unificación, cada proyección creada deberá guardar en sí la información no local de sus constituyentes inmediatos. Este proceso se puede iterar tantas veces como sea necesario, de forma que cada proyección cuente con información relativa a lo que ocurrió en momentos previos de la derivación. Se confía en que, en algún momento de la derivación, una proyección pueda utilizar esa información heredada. Por ejemplo —como se muestra en la figura 4.3— para dar cuenta de (1-a), la proyección 1 se crea saturando gratuitamente el objeto de *visitemos*. Pero se guarda en 1 una señal *[acc]* que indica que ese objeto debe aparecer en algún momento posterior de la derivación. Después, cuando la gramática crea 2 , esa información de 1 se pasa también a 2 . Otro tanto ocurre en el paso de 2 a 3 . Por fin, esa información no local de 3 se puede cotejar con *Qué* al crearse la proyección 4 , con lo cual se legitima la saturación gratuita anterior.

$$\begin{array}{l} \{ \textit{Qué quieres que visitemos}_{[acc]} \} \longrightarrow \{ \textit{Qué quieres que } 1_{[acc]} \} \\ \{ \textit{Qué quieres que } 1_{[acc]} \} \longrightarrow \{ \textit{Qué quieres } 2_{[acc]} \} \\ \{ \textit{Qué quieres } 2_{[acc]} \} \longrightarrow \{ \textit{Qué } 3_{[acc]} \} \\ \{ \textit{Qué } 3_{[acc]} \} \longrightarrow \{ 4_{[]} \} \end{array}$$

Figura 4.3: Derivación no transformacional

En definitiva, una gramática de unificación, al contemplar las proyecciones

como entes atómicos¹, debe guardar en cada proyección la información no local de sus constituyentes, con la esperanza ciega de que alguna proyección posterior pueda hacer el debido uso de ella. Una gramática transformacional, en cambio, al contemplar las estructuras encorchetadas de (1) como entes gramaticales, puede establecer directamente relaciones entre elementos de distintas proyecciones.

Como se puede imaginar, la formalización de este proceso de paso de información de una proyección a otra descansa, para una gramática de unificación, en el uso de rasgos con estructura compartida. En HPSG, la información no local de un signo se codifica en su rasgo NON-LOCAL. Cada regla capaz de crear una proyección debe heredar, como valor de su rasgo NON-LOCAL, la información de NON-LOCAL de sus constituyentes. Pollard y Sag (1994) regularon el paso de información no local de proyección en proyección mediante lo que llamaron *Nonlocal Feature Principle*. Este principio, tal como se enuncia en esa obra, establece que el valor de NON-LOCAL en una proyección se define como la suma del valor de NON-LOCAL de sus constituyentes inmediatos.

Pero ¿cómo se formaliza la suma de los rasgos NON-LOCAL de los constituyentes en la proyección? ¿Qué tipo de información no local cabe codificar en NON-LOCAL, y cómo debe ser formalizada? Es comúnmente aceptado, tanto en los trabajos teóricos —Pollard y Sag (1994) y Sag y otros (2002)— como en las gramáticas computacionales —Matrix o SSG— que NON-LOCAL esté formado por una serie de listas². Cada una de ellas recoge elementos de aplicación no local de uno u otro tipo. Por tanto, para sumar la informa-

¹Para una gramática de unificación, las proyecciones son entes atómicos en el sentido de que no incluyen la explicitación de sus constituyentes. No lo son en tanto que consisten en estructuras de rasgos.

²En realidad, en Pollard y Sag (1994), estos rasgos tienen por valor un conjunto, no una lista. De este modo se da cuenta del hecho de que, cuando hay varios elementos desplazados, la saturación de estos no sigue un orden establecido —véase (1)—.

- (1) a. La ciudad_{*i*} a los turistas_{*j*} no se la hemos enseñado $t_i t_j$
 b. A los turistas_{*j*} la ciudad_{*i*}, no se la hemos enseñado $t_i t_j$

ción de NON-LOCAL de los constituyentes en la proyección, se debe definir cada lista de esta como la suma de las correspondientes listas de sus constituyentes.

Pollard y Sag (1994) y *Matrix* asignan al valor de NON-LOCAL tres rasgos: SLASH, QUE y REL. En SLASH se recogen los argumentos de un signo que no se han realizado en su dominio local y están pendientes de saturación en un dominio no local: como todos los objetos de *visitemos* en (1). En realidad, se recoge la información local de esos argumentos: elementos de tipo *local*. En QUE, en cambio, se recogen marcas que denotan la presencia de un elemento interrogativo en algún punto de la derivación: marcas útiles, por ejemplo, para denotar la naturaleza interrogativa de una proyección como *¿* en (1-c). En concreto, Pollard y Sag (1994) colocan en QUE elementos semánticos de tipo *npro* —subtipos de *mrs* propios de signos nominales no pronominales—. Por último, los elementos de REL son índices semánticos, útiles para establecer la identidad entre los pronombres relativos y las posiciones argumentales a las que obedezcan. Gracias a REL es posible establecer el carácter relativo de la proyección *¿* en (1-d). Así, entonces, que el valor de NON-LOCAL es un tipo *non-local*, que obedece a la representación de la figura 4.4.

$$\left[\begin{array}{l} \textit{non-local} \\ \text{SLASH} \langle \textit{local}, \dots \rangle \\ \text{QUE} \langle \textit{npro}, \dots \rangle \\ \text{REL} \langle \textit{index}, \dots \rangle \end{array} \right]$$

Figura 4.4: Valor de NON-LOCAL

Por tanto, la información no local de un signo se formaliza a través de

Pero en las gramáticas computacionales no existe una sintaxis definitoria de la noción de conjunto. Por tanto, en *Matrix* estos rasgos son meras listas.

Además, en esta obra, NON-LOCAL cuenta con una estructura más compleja que la que se presentará aquí, pensada para dar cuenta de los niveles de saturación de los argumentos desplazados. Todo ello son cuestiones técnicas que no aportan nada a esta discusión. Se obvian aquí, por tanto.

elementos, del tipo pertinente, listados en SLASH, QUE y REL. El *Nonlocal Feature Principle* de Pollard y Sag (1994) establece, entonces, que las listas SLASH, QUE y REL de una proyección se definen como la suma de las listas correspondientes de sus constituyentes. Con ello, se logra recordar a lo largo de la derivación la información no local pertinente. Como puede imaginarse, el *Nonlocal Feature Principle*, en una gramática computacional, se formalizará en las reglas gramaticales, tal como se representa en la figura 4.5.

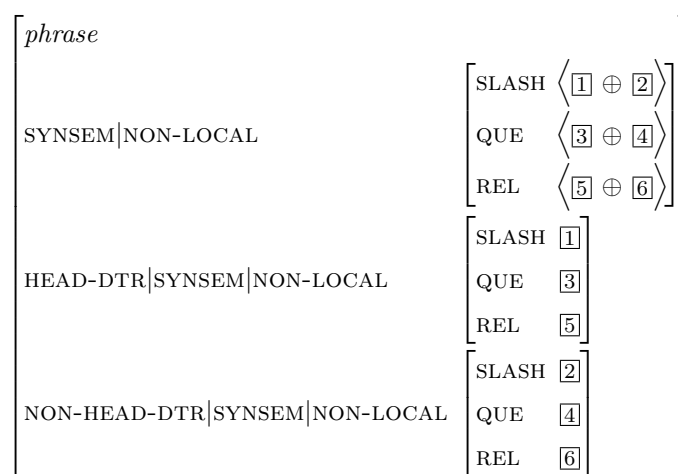


Figura 4.5: *Nonlocal Feature Principle* en Pollard y Sag (1994)

Ahora bien, en *Matrix*, se logra el mismo efecto del *Nonlocal Feature Principle* siguiendo una estrategia ligeramente distinta. En esta gramática, la suma de SLASH, QUE y REL no se establece directamente entre los constituyentes y las proyecciones (en las reglas); sino que se establece en el léxico. Las unidades léxicas que desencadenen procesos no locales colocarán los elementos pertinentes en sus listas SLASH, QUE y REL. Para el resto, SLASH, QUE y REL se definirán como la suma de las listas correspondientes de sus argumentos, listados en ARG-ST. Con ello, se consigue que cada núcleo de una proyección tenga, ya desde el léxico, su propia información no local, o la de cualquier complemento que se una a él como argumento. Una vez sumada en el núcleo la información no local de sus argumentos, esa información se propaga del núcleo a sus proyecciones gracias a una especificación gramati-

cal: las reglas de proyección deben identificar el valor de NON-LOCAL de la proyección con el valor de NON-LOCAL del núcleo. La combinación de estas dos caracterizaciones es el equivalente en Matrix al *Nonlocal Feature Principle* de Pollard y Sag (1994).

SSG sigue a Matrix en este punto. En esta gramática las reglas capaces de definir proyecciones son *comps-rule* y *subj-rule*. Así pues, el equivalente al *Nonlocal Feature Principle* en SSG, para una unidad léxica de dos argumentos (*transitive-lex-item*), consiste en la combinación de los tipos siguientes de las figuras 4.6³ y 4.7.

³En realidad, esta representación de la suma de elementos, si bien será utilizada en adelante en la discusión, es una simplificación. El operador \oplus no existe como primitivo en el lenguaje de implementación de gramáticas utilizado para el desarrollo de Matrix y SSG —TDL—.

En realidad, las listas definidas como suma de otras listas (SLASH y QUE) se modelan en Matrix y SSG con el tipo especial *diff-list*. Este tipo de listas cuentan con dos rasgos: LIST y LAST. LIST es una lista normal list, y LAST se concibe como la última lista anidada en LIST. De este modo, si se quiere sumar una lista λ a la lista LIST, basta con colocar λ como valor del LAST correspondiente a LIST.

Para que este proceso de suma de λ sea efectivo, cada lista *diff-list* debe establecer si tiene listados en sí elementos propios. Una lista *diff-list* que cuente con elementos propios debe codificar esos elementos en LIST, e identificar el resto de esa lista con LAST. Con ello se formaliza el que la lista consiste en sus elementos propios más los que se le aporten a través de LAST (λ). Si, en cambio, la lista no cuenta con elementos propios, debe identificar directamente LIST y LAST. Con ello se da a entender que esa lista consistirá solo en aquello que se le aporte a través de LAST (λ). Llámese al primer patrón *append-list* y al segundo *non-append-list*.

Por último, para sumar las listas *diff-list* de los argumentos de ARG-ST de un tipo a su núcleo se deben establecer en tal tipo las siguientes relaciones. Cada argumento suma su lista LIST a la lista LIST del argumento anterior, a través de LAST. Por tanto, todas esas listas quedan sumadas en la lista LIST del primer argumento de ARG-ST. Esa suma es la que se pasa a la lista LIST del núcleo. Además, el rasgo LAST del núcleo se identifica con el rasgo LAST del último elemento de ARG-ST. Con ello se asegura que, en el núcleo, LAST represente la lista más anidada de su correspondiente LIST. Esto es crucial si se quiere

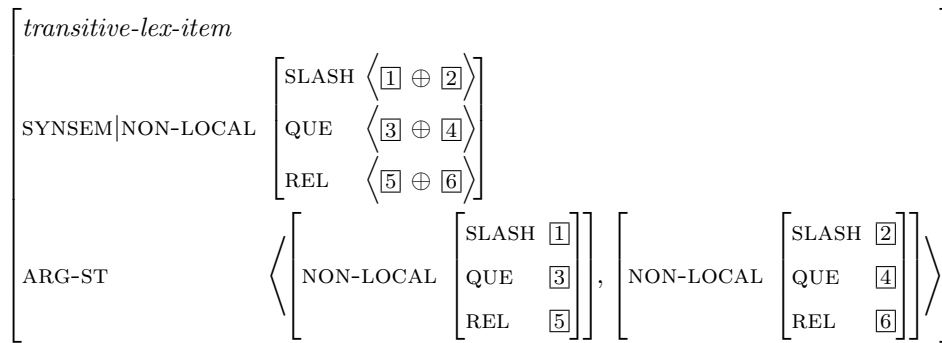


Figura 4.6: *Nonlocal Feature Principle* en SSG (I)

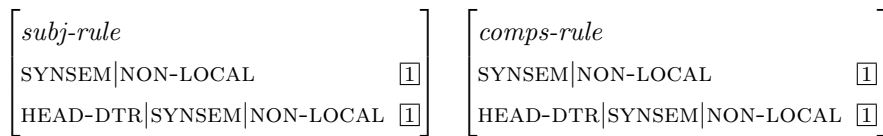


Figura 4.7: *Nonlocal Feature Principle* en SSG (II)

En resumen, dada la naturaleza no transformacional de las gramáticas de unificación, cuando estas crean una proyección, deben codificar en ella cierta información, entendida como no local, presente en sus constituyentes inmediatos. Esta información es reveladora de lo sucedido en un determinado estadio de la derivación. Este proceso se itera n veces, en espera de que, en otro estadio derivativo, tal información pueda ser utilizada por alguna proyección n . Este es el mecanismo que utilizan las gramáticas de unificación para dar cuenta de los fenómenos no locales.

4.1.2. La regla **comps-rule** con complementos elididos

Según lo expuesto en 4.1.1, en *Matrix* y en *SSG*, el desencadenante de todo proceso no local es el hecho de que una unidad léxica cuente con un elemento en una lista no local. En este apartado se reflexiona sobre la naturaleza de algunas de las unidades léxicas que desencadenan un proceso no local. Se iterar este proceso, haciendo que, en algún momento de la derivación, este núcleo funcione como argumento de otro.

repite, para ello, los ejemplos de (1) en (2).

- (2) a. ¿ [4 Qué_i [3 quieres [2 que [1 visitemos \emptyset_i]]]] ?
 b. [La ciudad_i [4 que_i [3 quieres [2 que [1 visitemos \emptyset_i]]]]]
 c. ¿ [[2 A [1 qué_i mujer]] [quieres [que [visitemos \emptyset_i]]]] ?
 d. [La mujer_i [2 a [1 la que_i]] [quieres [que [visitemos \emptyset_i]]]]]

En (2-c), la proyección 2 tiene carácter interrogativo. Y es este carácter interrogativo el que la habilita como objeto desplazado. La cuestión es que esa naturaleza se la debe 2 a la presencia de la partícula *qué* en la proyección 1. Esta, por tanto, trasciende su dominio local y propaga su carácter interrogativo hasta la proyección 2. Este hecho se formaliza, en **Matrix** y en **SSG**, caracterizando a *qué* en el léxico tal como se muestra en la figura 4.8. Como se observa, un determinante interrogativo como *qué*, instancia de **interrogative-specifier**, coloca su información semántica como elemento de su lista no local **QUE**.

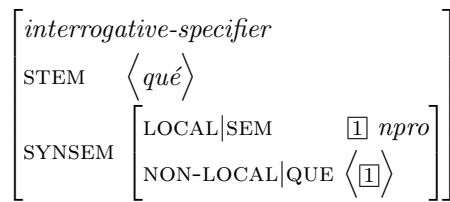


Figura 4.8: *interrogative-specifier*

Entiéndase que, en (2-c), *qué* se une al núcleo *mujer* como especificador de este. Los especificadores de un núcleo, en **Matrix**, también se colocan en su lista ARG-ST. En tal caso, conforme a lo visto en 4.1.1, el elemento **npro** de la lista **QUE** de *qué* pasa a la lista **QUE** de *mujer*. Desde ahí, como se sabe, dada la caracterización de **comps-rule** y **subj-rule**, se propaga a la proyecciones 1 y 2.

Del mismo modo, en (2-d), la proyección 2 tiene carácter relativo gracias a la presencia en esta del relativo *que*. Pero, de nuevo, *que* está anidado en la proyección 1. Por tanto, *que* debe ser caracterizado en el léxico de

modo tal que desencadene la propagación no local del carácter relativo hasta la proyección 2. Para ello, *que* se codifica en el léxico como instancia del siguiente tipo de la figura 4.9.

$$\left[\begin{array}{l} \textit{relative-pronoun} \\ \text{STEM} \quad \langle \textit{que} \rangle \\ \text{SYNSEM} \quad \left[\begin{array}{l} \text{LOCAL|SEM|HOOK|INDEX} \quad \boxed{1} \textit{ index} \\ \text{NON-LOCAL|REL} \quad \langle \boxed{1} \rangle \end{array} \right] \end{array} \right]$$

Figura 4.9: *relative-pronoun*

Por último, cabe preguntarse qué unidad léxica desencadena el desplazamiento del objeto de *visitemos*, en todas las oraciones de (2). En cualquier versión clásica de HPSG, como en el modelo chomskyano, suele entenderse, de un modo u otro, que, en estos casos, *visitemos* se une a una huella carente de realización fonética: \emptyset . Por tanto la proyección 1 de (2-a) y (2-b), y las correspondientes en (2-c) y (2-d), son fruto de la unión de *visitemos* y una huella. La caracterización léxica de esta huella debe incluir en su lista SLASH el elemento desencadenante del proceso no local; del mismo modo que *qué* incluye un elemento **npro** en su lista QUE, y *que* incluye un elemento **index** en su lista REL. En **Matrix** estas huellas se modelan como signos lingüísticos cuyo valor para el rasgo SYNSEM es de tipo **gap**. Por ejemplo, contémplese la existencia de la siguiente unidad léxica **trace** —figura 4.10—.

$$\left[\begin{array}{l} \textit{trace} \\ \text{STEM} \quad \langle \rangle \\ \text{SYNSEM} \quad \left[\begin{array}{l} \textit{gap} \\ \text{LOCAL} \quad \boxed{1} \textit{ local} \\ \text{NON-LOCAL|SLASH} \quad \langle \boxed{1} \rangle \end{array} \right] \end{array} \right]$$

Figura 4.10: *trace*

El tipo **trace** no tiene realización fonética. Además, su estructura gramatical se caracteriza como de tipo **gap**: la información local se guarda como

elemento de la lista SLASH. Por tanto, imagínese que un núcleo, como *visitemos* en (2), toma como argumento una unidad léxica de tipo **trace**. En tal caso se puede formar una proyección 1, donde el objeto no tendrá manifestación fonética. Pero ese objeto (en tanto que instancia de **trace**) aportará un elemento a la lista SLASH de *visitemos*. Esa información se propagará proyección tras proyección, hasta que un sintagma adecuado (*filler*) se una a la proyección 3, saturando su lista SLASH. En ese proceso de saturación, el *filler* identifica su información local con la codificada en SLASH, con la de la huella. Con ello, la información del *filler* se identifica con la del argumento de *visitemos*.

Por tanto, la huella en cuestión es capaz de ejercer de complemento de un núcleo en un contexto local. En SSG, en estos casos, la huella ocupa la posición NON-HEAD-DTR de la regla **comps-rule**, igual que pudiera hacerlo un signo con realización fonética. Para ello, SSG recoge las unidades léxicas sin realización fonética (huellas, pronombres omitidos, etc.) en un lexicón propio. Después, el sistema de *parsing* SGP contempla la posibilidad de crear proyecciones a partir del esquema **comps-rule**, dado un núcleo explícito en la secuencia de entrada (HEAD-DTR) y un complemento sin realización fonética tomado de tal lexicón (NON-HEAD-DTR).

$$\left[\begin{array}{l} \text{comps-rule} \\ \text{SYNSEM|LOCAL|VAL|COMPS} \quad \boxed{2} \\ \text{HEAD-DTR|SYNSEM|LOCAL|COMPS} \quad \langle \text{phr-synsem} \boxed{1} . \boxed{2} \rangle \\ \text{NON-HEAD-DTR} \quad \boxed{1} \end{array} \right]$$

Figura 4.11: **comps-rule** en **Matrix**

Esta es una solución novedosa entre las gramáticas computacionales. **Matrix** no codifica los signos sin realización fonética como entradas del lexicón. Por ello, debe crear una regla de proyección por cada signo de este tipo que contemple. En **Matrix**, **comps-rule** solo crea proyecciones a partir de dos signos identificados en la secuencia de entrada: el núcleo y el complemento. Los complementos explícitos se caracterizan en esta gramática como de tipo sintagmático: SYNSEM *phr-synsem*. Véase en la figura 4.11.

En cambio, para contemplar la posibilidad de que se cree una proyección mediante la unión de un núcleo y una huella, **Matrix** usa una regla similar: **gap-comps-rule**. Esta regla identifica un único signo en la cadena de entrada: el núcleo. A ese núcleo se le satura gratuitamente una necesidad argumental en COMPS; y esa necesidad argumental se caracteriza como huella: SYNSEM *gap*. Véase en la figura 4.12.

$$\left[\begin{array}{l} \textit{gap-comps-rule} \\ \text{SYNSEM|LOCAL|VAL|COMPS} \quad \boxed{\text{I}} \\ \text{HEAD-DTR|SYNSEM|LOCAL|COMPS} \quad \langle \textit{gap} . \boxed{\text{I}} \rangle \end{array} \right]$$

Figura 4.12: **gap-comps-rule** en **Matrix**

Otro tanto ocurre en **Matrix** para definir la unión de un núcleo con un pronombre sin realización fonética. En este caso se utiliza una regla llamada **opt-comps-rule**. Esta, como **gap-comps-rule**, identifica un único constituyente en la cadena de entrada: el núcleo, de nuevo. En ese núcleo también se satura gratuitamente una necesidad argumental de COMPS; pero, en este caso, se caracteriza a dicho argumento, no como huella (esta regla no desencadena proceso no local), sino como elemento elidido: SYNSEM *unexpressed*. Véase en la figura 4.13.

$$\left[\begin{array}{l} \textit{opt-comps-rule} \\ \text{SYNSEM|LOCAL|VAL|COMPS} \quad \boxed{\text{I}} \\ \text{HEAD-DTR|SYNSEM|LOCAL|COMPS} \quad \langle \textit{unexpressed} . \boxed{\text{I}} \rangle \end{array} \right]$$

Figura 4.13: **opt-comps-rule** en **Matrix**

SSG evita esta multiplicación de reglas, al entender que una huella o un pronombre omitido son unidades léxicas capaces de cubrir la posición de complemento de **comps-rule**. Con ello, la gramática gana en sencillez. No obstante, en principio, esta estrategia no supone ahorro computacional. Con SSG, para proyectar un núcleo con un complemento, el sistema de *parsing* debe

aplicar *comps-rule* tantas veces como unidades léxicas sin realización fonética se contemplen; del mismo modo que, con *Matrix*, debe aplicar tantas reglas diferentes como signos sin realización fonética se contemplen. Por tanto, en principio, ambos sistemas son equivalentes en cuanto a coste computacional.

4.2. Casuística

En español, los argumentos de dativo y acusativo de un verbo pueden manifestarse en este en forma de clíticos. El clítico aparece necesariamente si el argumento en cuestión está topicalizado —(3-a) y (3-b)—, cuando el argumento está omitido —(3-c) y (3-d)—, o cuando es un pronombre personal —(3-e) y (3-f)—.

- (3)
- a. La entrada, ¿puedo comprarla en la oficina de turismo?
 - b. Al encargado ¿le has dado la entrada?
 - c. ¿Puedo comprarla en la oficina de turismo?
 - d. Le ha dado la entrada
 - e. ¿La has contratado a ella?
 - f. Le ha dado la entrada a él

Cuando el argumento acusativo no se encuentra en ninguna de estas situaciones, no puede llevar clítico, como se ve en (4-a). En cambio, el argumento dativo, si no se encuentra en dichas situaciones, puede ir acompañado de clítico —(4-b)— o no⁴ —(4-c)—.

- (4)
- a. *¿Lo has contratado al guía?
 - b. ¿Le has dado la entrada al encargado?
 - c. ¿Has dado la entrada al encargado?

Obsérvese que, de acuerdo con lo anteriormente expuesto, cuando el argumento se antepone al verbo, funciona de distinto modo según sea o no tópico. Un argumento de dativo puede prescindir de clítico, o no, si es foco —(5-a)—,

⁴Conforme a lo que se mostró en 3.5.4, los verbos para los cuales el dativo no forma parte de su red argumental básica piden en cualquier caso la aparición del clítico.

partícula interrogativa —(5-b)— o relativo —(5-c)—; pero lo necesita si es tópico —(5-d)—.

- (5) a. ¡A LOS NIÑOS, (les) cuesta la entrada 13 euros, no a ti!
 b. ¿A quiénes (les) cuesta la entrada 13 euros?
 c. Los niños a quienes (les) cuesta la entrada 13 euros
 d. A los niños *(les) cuesta la entrada 13 euros

El acusativo, en cambio, no admite clítico si es foco —(6-a)—, partícula interrogativa —(6-b)— o relativo —(6-c)—; pero lo necesita si es tópico —(6-d)—.

- (6) a. ¡AL GUÍA (*lo) he contratado, no al botones!
 b. ¿A quién (*lo) has contratado?
 c. El guía al que (*lo) contrataste
 d. Al guía, *(lo) he contratado

Es importante, para la discusión que sigue, mostrar que estos mismos procesos de topicalización, elisión o pronominalización no suponen cliticización alguna para los argumentos de caso nominativo u oblicuo. Véase en las oraciones de (7).

- (7) a. El guía dicen que viene con nosotros
 b. Viene con nosotros
 c. ¿Viene él con nosotros?
 d. A los bolos juego cerca del hotel
 e. Juego cerca del hotel
 f. ¿Tú confías en él?

Además de estos clíticos relativos a los argumentos de acusativo y dativo, el español cuenta con los propios de las construcciones pronominal, pasiva refleja e impersonal. Los verbos pronominales, ya vistos en 3.4, son aquellos que solo pueden aparecer acompañados de un clítico no referencial, como *desentenderse* en (8-a). La pasiva refleja —véase el apartado 3.3— es una construcción que elimina el argumento externo de un verbo, asigna a su ar-

gumento interno caso nominativo y le añade un clítico *se*. El uso de *contratar* en (8-c) es un ejemplo de pasiva refleja, donde se ha eliminado el argumento externo presente en las versiones transitivas de este verbo —(8-b)—. Por último, la construcción impersonal elimina también el argumento externo del verbo y le añade el clítico *se*; pero, a diferencia de la pasiva refleja, mantiene el caso acusativo del argumento interno. Lo que interesa en este apartado es que, como se ve en (8), estas tres construcciones implican la aparición de un clítico sin valor referencial.

- (8)
- a. El hotel *(se) desentiende de la situación
 - b. Ellos han contratado guías para la excursión
 - c. Se han contratado guías para la excursión
 - d. Ellos han visto al tiburón
 - e. Se ha visto al tiburón

Para acabar este apartado 4.2, se hablará del llamado salto de clíticos. Un verbo puede aparecer en una oración como lo que se ha llamado verbo principal de otros verbos auxiliares, modales o aspectuales. Estos verbos ubican el evento denotado por el verbo principal conforme a una serie de coordenadas temporales, aspectuales y modales. Pues bien, los clíticos propios del verbo principal pueden colocarse en la oración junto a este, pero también junto a uno de sus verbos auxiliares (*haber*), modales (*poder*, *deber*, *querer*, etc.) o aspectuales (*empezar*, *terminar*, *continuar*, etc.). El clítico, por tanto, se interpreta como argumento del verbo principal, pero se puede realizar más o menos alejado de él. Los ejemplos de (9) muestran todo ello: los argumentos del verbo *pagar* pueden colocarse, como clíticos, en este, o en el verbo que lo subcategoriza (*querer*).

- (9)
- a. El hotel, lo queremos pagar al contado
 - b. El hotel, queremos pagarlo al contado
 - c. *El hotel, lo queremos pagarlo al contado

Este salto de clíticos está sujeto a ciertas reglas. Como se observa en (9), los verbos en forma personal llevan los clíticos antepuestos (proclíticos),

mientras que los infinitivos, gerundios e imperativos los llevan pospuestos (enclíticos). Además, todos los clíticos argumentos de un verbo deben realizar-se en el mismo lugar. Por ello, la oración de (10), donde el clítico de dativo de *pagar* se realiza en *querer* y el de acusativo en el mismo *pagar*, es agramatical.

- (10) a. El hotel, ¿se lo podemos pagar al contado?
 b. *El hotel, ¿le podemos pagarlo al contado?

Por último, no se establece salto de clíticos entre dos verbos entre los que media un complementante. Véase (11).

- (11) a. El hotel, ¿sabes si lo podemos pagar al contado?
 b. El hotel, ¿lo sabes si podemos pagar al contado?

4.3. Soluciones previas

El fenómeno de los clíticos, en los trabajos teóricos de HPSG, se ha solucionado siempre haciendo uso de reglas léxicas. Ya Pollard y Sag (1994) asumen la existencia de reglas de cliticización para el galés⁵. Monachesi (1998) propone también reglas de este estilo para el italiano. En ambos casos, se conciben los clíticos como afijos verbales. Por su parte, SRG, la gramática computacional de tipo HPSG de referencia en la actualidad, también utiliza reglas léxicas para el modelado de los clíticos del español. En SRG —en esto se aleja de los trabajos teóricos antes mencionados—, los clíticos son pronombres autónomos, no afijos.

En cualquier caso, todas estas propuestas siguen una estrategia similar. Dado un núcleo que subcategoriza un argumento, una regla léxica opera de algún modo sobre el argumento, y le añade el clítico correspondiente al núcleo. En el caso típico, la regla elimina el argumento de un verbo, y en su lugar le añade a este el clítico correspondiente. Si ese clítico añadido se

⁵Pollard y Sag (1994), pp. 351-358.

concibe como un afijo verbal, bastará con que el significante del verbo refleje la presencia del clítico. Pero si se concibe como un pronombre autónomo, ese clítico será un nuevo elemento subcategorizado, que deberá ser saturado de algún modo —hay que pensar que con una regla de proyección específica—.

A continuación, se explica con más detalle el sistema de clíticos de SRG. En concreto, se presenta la solución de Pineda y Meza (2005) para los clíticos del español, solución en el cual se basa la implementación de aquella gramática. Pineda y Meza (2005) entienden los clíticos del español como pronombres átonos subcategorizados, no en COMPS, como el resto de complementos, sino en una lista propia de clíticos: CL-LIST. Por tanto, en VAL, añaden CL-LIST a las listas de subcategorización SUBJ y COMPS. Desde esta lista, una nueva regla gramatical, *head-proclitic-rule*, satura los clíticos delante del verbo. En la figura 4.14 se muestra esta regla —simplificada y adaptada a la versión del signo utilizada hasta el momento—.

$$\left[\begin{array}{l} \textit{head-proclitic-rule} \\ \text{SYNSEM|LOCAL|VAL|CL-LIST} \quad \langle \rangle \\ \text{HEAD-DTR|SYNSEM|LOCAL|VAL|CL-LIST} \quad \boxed{1} \\ \text{NON-HEAD-DTR|SYNSEM|LOCAL} \quad \left[\begin{array}{l} \text{HEAD} \quad \textit{clitic} \\ \text{VAL|CL-LIST} \quad \boxed{1} \end{array} \right] \end{array} \right]$$

Figura 4.14: Regla de saturación de clíticos de Pineda y Meza (2005)

Estos argumentos realizados como clíticos se pasan directamente de ARG-ST a CL-LIST. Este proceso se regula con reglas léxicas. Por ejemplo, estos autores contemplan una regla a la que llaman *simple-clitization-rule*. Es una regla que toma un verbo ditransitivo, elimina de ARG-ST sus dos complementos, y los añade a la lista de clíticos CL-LIST —véase en la figura 4.15—. Dada la aplicación de *simple-clitization-rule*, los complementos canónicos del verbo quedan saturados, y serán sus clíticos correspondientes los que habrán de saturarse mediante *head-proclitic-rule*. El resultado de esta regla es, pues, una de las posibilidades de realización de un verbo ditransitivo en español: aquella en la que sus dos complementos —acusativo y dativo— se realizan

como clíticos.

$$\left[\begin{array}{l} \textit{simple-clitization-rule} \\ \text{SYNSEM|LOCAL} \left[\text{VAL|CL-LIST} \langle \boxed{3}, \boxed{2} \rangle \right] \\ \text{ARG-ST} \quad \langle \boxed{1} \rangle \\ \text{DTR} \quad \left[\begin{array}{l} \text{SYNSEM|LOCAL|VAL|CL-LIST} \langle \rangle \\ \text{ARG-ST} \quad \langle \boxed{1}, \boxed{2}, \boxed{3} \rangle \end{array} \right] \end{array} \right]$$

Figura 4.15: Clitización simple

Como puede imaginarse, serán necesarias reglas léxicas del tipo de **simple-clitization-rule** que pasen a CL-LIST solo el primer complemento de un verbo ditransitivo, o solo el segundo. También habrá que contemplar la existencia de una regla que pase a CL-LIST el único complemento de un verbo transitivo. Además, obsérvese que **simple-clitization-rule** solo da cuenta de las apariciones de clíticos motivadas por elisión del complemento. Como se ha visto en 4.2, un clítico aparece también en español cuando su argumento correspondiente es un pronombre personal o está topicalizado. Para cubrir estos datos, Pineda y Meza (2005) proponen **pleonastic-clitization-rule**, una regla que identifica los índices de los complementos de ARG-ST con sus clíticos correspondientes en CL-LIST, pero no elimina los argumentos de aquella lista —figura 4.16—.

$$\left[\begin{array}{l} \textit{pleonastic-clitization-rule} \\ \text{SYNSEM|LOCAL} \left[\text{VAL|CL-LIST} \langle \boxed{3}_i, \boxed{2} \rangle \right] \\ \text{ARG-ST} \quad \langle \boxed{1}, \boxed{3} \rangle \\ \text{DTR} \quad \left[\begin{array}{l} \text{SYNSEM|LOCAL|VAL|CL-LIST} \langle \rangle \\ \text{ARG-ST} \quad \langle \boxed{1}, \boxed{2}, \boxed{3}_i \rangle \end{array} \right] \end{array} \right]$$

Figura 4.16: Clitización pleonástica

Como puede observarse en la figura 4.16, **pleonastic-clitization-rule** opera sobre verbos ditransitivos. Concretamente, representa, para estos, el patrón de realización consistente en la clitización y elisión del acusativo, y la clitización sin elisión del dativo. Por ejemplo, para el verbo *dar*, **pleonastic-**

clitization-rule representa el patrón de realización propio de la oración de (12).

(12) Al guía se lo han subido a la habitación

De nuevo, será necesaria la inclusión en la gramática de reglas similares a *pleonastic-clitization-rule* que den cuenta de todas las posibilidades de cliticación (con elisión o sin ella) de los argumentos de los verbos ditransitivos y transitivos del español. Por tanto, una gramática que —como SRG— asuma este sistema, u otro similar, añadirá un buen número de reglas léxicas —*simple-clitization-rule*, *pleonastic-clitization-rule* y sus variantes— a las que utilice para otros menesteres. En concreto, si SSG utilizara este sistema, para calcular los posibles patrones de comportamiento de un verbo del español, debería cruzar estas reglas con las reglas vistas en el capítulo 3. Además, a las reglas de proyección *comps-rule* y *subj-rule* se les debería añadir *head-proclitic-rule*.

4.4. El sistema de clíticos de SSG

En este apartado, se explica cómo da cuenta SSG de los datos que aparecen en el apartado 4.2. Para ello, esta gramática se aleja de las soluciones previas vistas en 4.3 y gana en sencillez, elegancia y ahorro computacional.

4.4.1. Los clíticos como problema morfológico

¿Es el de los clíticos un fenómeno morfológico o sintáctico? Es decir, ¿deben concebirse estos como afijos verbales, o como pronombres autónomos que saturan una necesidad argumental de su verbo? Desde un punto de vista teórico, esta es una cuestión controvertida, sin solución cerrada. Este trabajo no puede aportar nada al respecto. Como se ha dicho en 4.3, los trabajos teóricos en HPSG —Pollard y Sag (1994) y Monachesi (1998)— tienden a entender el de los clíticos como un problema morfológico. Para una defensa teórica de este planteamiento morfológico se remite a Monachesi (1998).

SSG se suma a estos trabajos y entiende también los clíticos como afijos verbales. Se hace por razones de coste computacional. Como se vio en 4.3, la solución sintactista implica el añadido a la gramática de una regla de proyección específica. SSG pretende reducir la explosión combinatoria mediante la reducción del número de reglas. Por tanto, se asume el enfoque morfológico para evitar el uso de tal regla.

Sin embargo, el modelado de los clíticos como problema morfológico se enfrenta a problemas que no existen para la solución sintactista. Los trabajos teóricos de HPSG pueden obviar estos problemas, pero si se quiere crear una gramática computacional real, es necesario afrontarlos. Toda gramática computacional de análisis sintáctico necesita un procesamiento previo de la secuencia de entrada. En él se deben identificar las unidades atómicas del análisis sintáctico: *tokens*. Los *tokens* de una secuencia, para casos no triviales, no son siempre las palabras. Será necesario unir varias palabras en un mismo *token* (nombres propios complejos, locuciones, etc.), separar una misma palabra en varios *tokens* (amalgamas), obviar signos de puntuación, etc. Así pues, un sistema que conciba los clíticos como afijos verbales debe unir, en un mismo *token*, al verbo propiamente dicho con sus clíticos.

En (13) se muestra una propuesta de división en *tokens* de la secuencia *El hotel, ¿se lo podemos pagar al contado?* Cada subsecuencia encorchetada se corresponde con un *token*, y será, entonces, una unidad atómica para el análisis sintáctico. Como se ve, se han obviado todos los signos de puntuación. Además, *al* y *contado* se han unido en un mismo *token*, pues forman una locución adverbial que se contempla como una sola entrada léxica.

(13) [El] [hotel] , ¿ [se lo podemos] [pagar] [al contado]?

Del mismo modo, *se*, *lo* y *podemos* forman un único *token*. En concreto, para analizar (13), SSG, al concebir los clíticos como afijos verbales, necesita recibir como único *token* la subsecuencia *se lo podemos*. Sería incapaz de analizar (13) si recibe *se*, *lo* y *podemos* como *tokens* independientes.

SGP, el sistema de *parsing* desarrollado en este trabajo, sí cuenta con un módulo de preprocesamiento que une los clíticos al verbo en un mismo *token*; y asigna a tal *token* una regla flexiva correspondiente. Por ejemplo, SGP, al procesar la secuencia de (13), identifica como *token* la subsecuencia *se lo podemos*, y le asigna el lema *poder*. Además, le adscribe dos reglas flexivas: **verb-1-pl-present-indicative-irule** y **verb-se-lo-irule**. La primera especifica los rasgos de persona, número, tiempo y modo del verbo. La segunda, además, especifica la existencia en este de los clíticos *se* y *lo*. Esta última especificación restringirá las posibilidades de realización del verbo y sus argumentos.

Esta regla **verb-se-lo-irule** cuenta en SSG con la siguiente estructura de rasgos de la figura 4.17. En ella se ha añadido el rasgo CLITICS a la estructura del signo lingüístico. Su valor es una lista en la que se colocan los clíticos con los que cuenta un verbo, entendidos estos como afijos verbales.

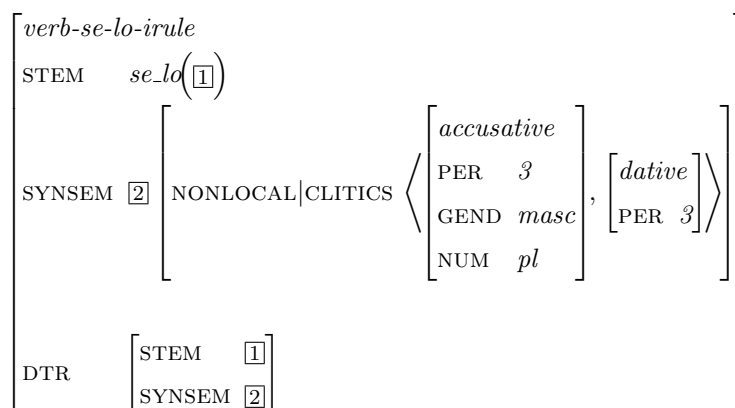


Figura 4.17: Regla flexiva de añadido de clíticos: **verb-se-lo-irule**

También puede observarse en la figura 4.17 que los clíticos se han caracterizado como conglomerados de rasgos de persona, género y número. En concreto, un clítico de acusativo se ha recogido en SSG mediante el tipo **accusative**, subtipo de **agr**. El tipo **dative**, también subtipo de **agr**, recoge el concepto de clítico de dativo. También existe en SSG un tercer subtipo de **agr** llamado **nominative**, que modela la concordancia de nominativo. Se conciben,

entonces, los clíticos como conglomerados de rasgos de concordancia adscritos a un caso determinado, distintas líneas de concordancia hábiles para un verbo⁶.

El tratamiento de los clíticos como cuestión morfológica evita ciertos problemas que sí deben abordar las propuestas sintactistas. Siguiendo a Fernández Soriano (1999), cuando un verbo cuenta con más de un clítico —tres como máximo—, estos deben aparecer en un orden fijo, y existen ciertas combinaciones imposibles. Por ejemplo, el clítico *se* marcador de pasiva refleja, impersonalidad, etc. solo puede aparecer en primera posición de la lista de clíticos —véase el contraste de (14)—.

- (14) a. Se le han perdido los billetes
 b. *Le se han perdido los billetes

Además, no pueden aparecer en un mismo verbo dos clíticos de primera persona, o dos clíticos de segunda persona: (15-a)⁷. En cambio, sí es posible

⁶Estos tipos *nominative*, *accusative* y *dative* no deben aparecer como valor del rasgo *l* de los índices semánticos. El valor de tal rasgo debe ser siempre *agr*, y no uno de sus subtipos. Esto es importante, pues hay situaciones en las que se coindizan argumentos de distinto caso, ya sean de un mismo predicado (una anáfora y su antecedente), o de predicados distintos (un pronombre relativo y su antecedente, un sujeto controlado y su antecedente, etc.).

⁷Ciertos casos particulares de esta restricción se cruzan con una restricción más general de la gramática: no puede haber solapamiento referencial entre los argumentos de un verbo. Es decir, como se observa en (1), no es posible la coexistencia en un verbo de dos argumentos (sujeto, objeto directo o indirecto) tales que uno sea de primera o segunda persona del singular y el otro de esa misma persona en plural. Desde este punto de vista, todas las oraciones de (1) son agramaticales en virtud del mismo principio.

- (1) a. *Me nos han contratado
 b. *Nos he contratado

Esta restricción, en la que se ven implicados el sistema de clíticos y la flexión de nominativo, es un problema para SSG. Si la flexión relativa a uno y otro fenómeno se modela con reglas flexivas distintas, como parece deseable, hay que idear un modo de que la interacción entre ellas contemple las restricciones de solapamiento referencial.

la aparición de dos clíticos de tercera persona, siempre y cuando el primero de ellos tome la forma *se*: véase (15-b) frente a (15-c).

- (15) a. {*Me me / *Te te} han contratado
 b. Se lo han contratado
 c. *Le lo han contratado

En caso de que aparezcan un clítico de segunda persona y un clítico de primera, el de segunda debe preceder al de primera: (16-a) frente a (16-b)⁸.

- (16) a. Te me han contratado
 b. *Me te han contratado

En caso de coaparición de un clítico de primera o segunda persona con un clítico de tercera, este último debe posponerse a aquel —(16-a) frente a (16-b)—. Además, en tales casos, como ocurre cuando conviven dos clíticos de tercera persona, el clítico de dativo precede siempre al de acusativo.

- (17) a. Te lo han contratado
 b. *Le te han contratado

Valgan estos datos para dar cuenta del hecho de que la coaparición de clíticos está sujeta a restricciones caprichosas difíciles de gestionar como fenómenos sintácticos. SSG no necesita hacer frente a tales restricciones. Dado su tratamiento morfológico del fenómeno de los clíticos, basta con establecer en la gramática solo reglas flexivas que aporten al verbo listas de clíticos legítimas. Por ejemplo, SSG contará con una regla flexiva que aporte a un verbo una lista de clíticos legítima *te lo*, pero no contará con la regla corres-

⁸Nótese que la oración de (16-a) es ambigua: si *Te* se concibe como clítico de acusativo, la oración denota que ‘tú has sido contratado para mí’, pero si se concibe como clítico de dativo, la oración denota que ‘yo he sido contratado para ti’. Parece que este requerimiento de orden en función de la persona gramatical es prioritario sobre el requerimiento general de que el clítico de dativo preceda al de acusativo. Esta ambigüedad será un problema para SSG, pues esta gramática presupone orden de clíticos fijo en función del caso: el dativo precede siempre al acusativo. Desde un punto de vista práctico no es un gran problema: la secuencia *te me* tiene un uso real marginal.

pondiente a la lista ilegítima *le te*.

4.4.2. Los clíticos motivados por la naturaleza del argumento

En cualquiera de los trabajos presentados hasta ahora —Pollard y Sag (1994), Monachesi (1998) o SRG— se hace un uso crucial de las reglas léxicas para regular la clitización de los argumentos verbales. SSG es, en cuanto a este particular, una gramática novedosa, pues no utiliza regla alguna para dar cuenta del fenómeno de los clíticos del español. Como se sugirió en 4.3, esto supondrá un serio ahorro de coste computacional.

En SSG se entiende que todo complemento de un verbo se saturará necesariamente en la lista COMPS. Se entiende, además, que, si un verbo cuenta con un clítico, es porque uno de sus argumentos ha sido saturado (desde COMPS) por un signo lingüístico que se lo ha aportado. Este signo lingüístico no es el clítico. El clítico es un mero afijo verbal. Este signo lingüístico es una unidad léxica (con realización fonética o sin ella) que desencadena en el verbo la aparición del clítico.

Entonces, ¿qué unidades léxicas aportan un clítico al verbo al que se unen como argumentos? Recuérdese que los procesos que desencadenan la aparición de clítico son la realización del argumento como pronombre personal, la omisión del argumento, y su topicalización. Por tanto, las unidades léxicas que aportan clítico al verbo son: los pronombres personales explícitos; sus correspondientes pronombres omitidos; y las huellas que queden en la posición de base de una topicalización. Si todos ellos se conciben como unidades léxicas capaces de saturar un elemento de la lista COMPS, pueden caracterizarse de forma que activen la aparición de un clítico en el verbo. El resto de unidades léxicas capaces de ser argumentos de un verbo (determinantes, nombres, complementantes, otro tipo de huellas...) deben caracterizarse de forma que no aporten clítico.

Por tanto, en SSG se entiende que, en (18-a), el clítico *La* es un afijo verbal cuya aparición ha sido desencadenada por el pronombre personal (*ella*) que ha ocupado la posición de argumento acusativo de *contratar*. En cambio, un sintagma nominal cuantificado, como *un guía* en (18-b), en la misma posición de argumento acusativo, no desencadena la aparición del clítico. Un argumento, bajo este análisis, nunca se realiza como clítico —el clítico es un mero afijo verbal—. Se realiza, por ejemplo, como pronombre personal o como sintagma nominal cuantificado. El primero tiene la propiedad de desencadenar la aparición del clítico, pero el segundo no.

- (18) a. ¿La has contratado a ella?
 b. ¿Has contratado un guía?
 c. ¿Puedo comprarla en la oficina de turismo?
 d. La entrada, ¿puedo comprarla en la oficina de turismo?

Además, de acuerdo con el análisis de SSG, en (18-c), el argumento acusativo ha sido saturado por un pronombre sin realización fonética. Como se ve, este tipo de pronombres también desencadena la aparición del clítico en el verbo. La idea es que este pronombre no es propiamente el clítico *la*. El pronombre no tiene realización fonética; pero, al igual que su versión explícita de (18-a), sí desencadena la aparición del clítico *la*.

Otro tanto ocurre con la huella que satura, en (18-d), al argumento acusativo del verbo *comprar*. La topicalización de un argumento —*La entrada* en (18-d)— es un caso de dependencia no acotada, de movimiento. El análisis habitual para estas construcciones en HPSG pasa por entender que, en la posición de base del movimiento (la lista COMPS de *comprar*), el argumento queda saturado por una huella. Una huella es un signo lingüístico sin realización fonética⁹. Pues bien, SSG caracteriza a las huellas de los elementos topicalizados de tal forma que también desencadenan la aparición del clítico en el verbo.

⁹Todo ello se explicará con detalle en el capítulo 5.

A continuación se presenta una primera formalización de este análisis. A la estructura del signo se le ha añadido una lista de clíticos: CLITICS. Además, recuérdese que los argumentos de un verbo se listan en ARG-ST. Pues bien, el valor de la lista CLITICS de un verbo será la suma del valor de la lista CLITICS de sus argumentos de ARG-ST. Por ejemplo, un verbo con dos argumentos en ARG-ST obedece a la siguiente caracterización de la figura 4.18. Como se ve, se ha entendido CLITICS como un rasgo no local. Esta decisión cobrará sentido cuando se explique la realización no local de los clíticos, el salto de clíticos —4.4.3—.

$$\left[\begin{array}{l} \text{SYNSEM|NON-LOCAL|CLITICS} \langle \boxed{1} \oplus \boxed{2} \rangle \\ \text{ARG-ST} \quad \left\langle \left[\text{NON-LOCAL|CLITICS} \boxed{1} \right], \left[\text{NON-LOCAL|CLITICS} \boxed{2} \right] \right\rangle \end{array} \right]$$

Figura 4.18: Valor de CLITICS para un verbo con dos argumentos

Por otro lado, véase cómo este análisis de los clíticos está inspirado en el análisis habitual en HPSG para cualquier fenómeno no local —este se explicó en 4.1.1—. No es una solución técnica *ad hoc*, sino la aplicación de una solución de uso general a un fenómeno nuevo. Imagínese que las listas CLITICS de los pronombres personales (omitidos o no), y las de las huellas de topicalización, cuentan con un clítico. Imagínese también que, en cambio, para el resto de unidades léxicas, la lista CLITICS está vacía. En tal caso, dada la caracterización verbal de la figura 4.18, la lista CLITICS de un verbo contaría con un clítico cada vez que uno de sus argumentos se realizase omitido (como pronombre sin realización fonética), topicalizado (como huella de topicalización) o en forma de pronombre personal. En cualquier otro caso, el argumento no aportaría clítico a la lista CLITICS del verbo.

Además, recuérdese —apartado 4.4.1— que los verbos entran al análisis sintáctico cargados con la información de los clíticos que aparecen en su significante. A cada verbo se le aplica una regla flexiva que especifica esa información en CLITICS. Obviamente, los clíticos que los argumentos aportan a un verbo y los clíticos que aparecen en su significante deben ser coherentes

—ambas dimensiones se codifican en CLITICS—.

Dicho de otro modo, un verbo cuyo significante aporte a CLITICS unos clíticos determinados solo permitirá derivaciones en las cuales sus complementos aporten a esa lista esos mismos clíticos. Un verbo con CLITICS vacío no admitirá, por ejemplo, un pronombre personal de acusativo como complemento. Y un verbo con un clítico de acusativo en CLITICS no admitirá como complemento acusativo un sintagma nominal cuantificado. En ambos casos, esos complementos configurarían la lista CLITICS del verbo de forma incompatible con la caracterización que le aportaron las reglas flexivas, de forma incoherente con lo que dice su significante.

Hasta el momento, se ha obviado en la discusión una cuestión de gran importancia. Se ha dicho que los pronombres personales, omitidos y explícitos, y las huellas de topicalización, aportan un clítico al verbo del que son argumentos. Se ha formalizado esta realidad codificando un clítico en su lista CLITICS. Pero, en realidad, estos pronombres y huellas solo aportan clítico cuando se les asigna un determinado caso: acusativo o dativo. Por tanto, para que la lista CLITICS de una unidad léxica esté cargada con un clítico, y lo aporte al verbo del que es argumento, es necesaria la convergencia de dos requisitos: que la unidad léxica sea una de las antes citadas, y que el caso que reciba como argumento verbal sea de acusativo o de dativo.

En SSG, las palabras no tienen especificado el caso en el lexicón. La caracterización del caso está codificada en la posición argumental del núcleo que las tomará como complemento. Como se vio en el capítulo 3, los verbos, con la aplicación de ciertas reglas léxicas, asignan caso a sus elementos listados en ARG-ST. Un argumento de caso nominativo estará caracterizado como CASE nom, uno acusativo será CASE acc, uno dativo será CASE dat, etc. Por tanto, para que un pronombre o una huella de los antes mencionados aporte clítico al verbo, es necesario que haya sido especificado como CASE acc o CASE dat. Pero esta especificación no llega hasta que ocupa la posición argumental. Por tanto, hasta ese momento, su lista CLITICS no debe ser especificada.

La intuición que se intenta formalizar es que, en español, un caso puede manifestarse mediante la presencia de determinadas marcas formales: un clítico en el verbo, flexión verbal tradicional o presencia de preposición. Así pues, la gramática del español establece que los casos nominativo y oblicuo no usan los clíticos como estrategia de manifestación formal. Sí los usan, en cambio, los casos acusativo y dativo. De acuerdo con ello, un argumento de un determinado caso (y realizado de un modo adecuado para la manifestación del clítico) aportará al verbo los clíticos que la gramática contemple como marca propia de tal caso. Para el español, si este caso es nominativo u oblicuo se aportarán al verbo cero clíticos; si es acusativo o dativo, un clítico.

Esta adscripción del uso del clítico a la naturaleza de cada caso se ha formalizado en SSG del siguiente modo. Se ha entendido que los distintos valores del rasgo CASE (*nom*, *acc*, *dat* y *obl*, subtipos de *case*) no son tipos atómicos. Cuentan, en cambio, con información relativa a las marcas formales que la gramática establece para cada uno de ellos. Por tanto, cuentan con un rasgo CLITIC¹⁰, cuyo valor es una lista de cero o un elementos. Los tipos *acc* y *dat* tendrán un clítico en su lista CLITIC, mientras que *nom* y *obl* tienen listas vacías (*null*). Véase la nueva caracterización de estos tipos en la figura 4.19.

$$\left[\begin{array}{c} \textit{nom} \\ \text{CLITIC } \langle \rangle \end{array} \right] \left[\begin{array}{c} \textit{acc} \\ \text{CLITIC } \langle \textit{accusative} \rangle \end{array} \right] \left[\begin{array}{c} \textit{dat} \\ \text{CLITIC } \langle \textit{dative} \rangle \end{array} \right] \left[\begin{array}{c} \textit{obl} \\ \text{CLITIC } \langle \rangle \end{array} \right]$$

Figura 4.19: Subtipos de *case*

A partir de esta caracterización, los pronombres personales, explícitos u omitidos, y las huellas de topicalización identifican su lista CLITICS con la lista CLITIC de su caso. De este modo, estas unidades léxicas aportan al verbo — en CLITICS— uno o cero clíticos: aquellos que su caso establezca como marca formal para sí —en CLITIC—. En SSG, los pronombres personales (explícitos u omitidos) y las huellas de topicalización se han recogido como instancias léxi-

¹⁰CLITIC es distinto del rasgo no local CLITICS mostrado anteriormente.

cas, respectivamente, de los tipos **personal-pronoun** y **topic-trace**¹¹. Véanse en la figura 4.20. En cambio, cualquier otra unidad léxica (determinantes, nombres, complementantes, otras huellas...) no establece esta identidad. Estas (independientemente del valor de CLITIC), al no ser contextos oportunos para la manifestación del clítico, se caracterizarán como CLITICS <>. Cualquiera de ellas, de un modo u otro, será instancia del tipo **non-clitics-lex-item**, representado en la figura 4.21.

$$\left[\begin{array}{l} \textit{personal-pronoun} \\ \text{SYNSEM} \left[\begin{array}{l} \text{LOCAL|HEAD|CASE|CLITIC} \quad \boxed{\mathbb{1}} \\ \text{NON-LOCAL|CLITICS} \quad \boxed{\mathbb{1}} \end{array} \right] \end{array} \right] \left[\begin{array}{l} \textit{topic-trace} \\ \text{SYNSEM} \left[\begin{array}{l} \text{LOCAL|HEAD|CASE|CLITIC} \quad \boxed{\mathbb{1}} \\ \text{NON-LOCAL|CLITICS} \quad \boxed{\mathbb{1}} \end{array} \right] \end{array} \right]$$

Figura 4.20: **personal-pronoun** y **topic-trace**

$$\left[\begin{array}{l} \textit{non-clitics-lex-item} \\ \text{SYNSEM} \left[\begin{array}{l} \text{LOCAL|HEAD|CASE} \quad \textit{case} \\ \text{NON-LOCAL|CLITICS} \quad \langle \rangle \end{array} \right] \end{array} \right]$$

Figura 4.21: **non-clitics-lex-item**

Según lo expuesto, en SSG se puede crear una proyección, mediante el esquema **comps-rule**, a partir de la existencia de un núcleo verbal (**HEAD-DTR**) y un complemento (**NON-HEAD-DTR**) identificado con una posición argumental del verbo. Si el elemento que se coloca en **NON-HEAD-DTR** es una instancia léxica de **personal-pronoun** o **topic-trace**, y la posición argumental es de acusativo o dativo, se activa la aparición de clítico.

Como ya se dijo en 4.1.2, SSG contempla la posibilidad, compatible con las versiones clásicas de HPSG, de que las unidades léxicas sin realización fonética (instancias de **personal-pronoun** y **topic-trace**, por ejemplo) ocupen la posición de complemento del esquema **comps-rule**. En aquel apartado se presentó el tipo **trace**, que caracterizaba a cualquier huella de la gramática.

¹¹Para una justificación y revisión de esta decisión, se remite al capítulo 5.

Hay que entender que el tipo *topic-trace*, visto en este apartado, es un subtipo de aquel. Por tanto, la estructura expandida de *topic-trace* es la que se muestra en la figura 4.22: lo propio de una huella es que desencadene un movimiento mediante el añadido de su información local a su lista SLASH. La existencia de un tipo *trace* y un subtipo de este —*topic-trace*— sugiere que, en SSG, se contempla también la existencia de otras huellas. Efectivamente, estas son necesarias para dar cuenta de otros movimientos que, como se vio en 4.2, no desencadenan la aparición de clítico: focalización, movimiento de partículas interrogativas y de relativos. Se sugiere, entonces, que en la posición de base de estos últimos movimientos aparece una huella *focus-trace*, cuya caracterización es la siguiente: figura 4.23.

$$\left[\begin{array}{l} \textit{topic-trace} \\ \text{SYNSEM} \left[\begin{array}{l} \text{LOCAL} \quad \boxed{1} \left[\text{HEAD|CASE|CLITIC} \quad \boxed{2} \right] \\ \text{NON-LOCAL} \left[\begin{array}{l} \text{CLITICS} \quad \boxed{2} \\ \text{SLASH} \quad \boxed{1} \end{array} \right] \end{array} \right] \end{array} \right]$$

Figura 4.22: *topic-trace*

$$\left[\begin{array}{l} \textit{focus-trace} \\ \text{SYNSEM} \left[\begin{array}{l} \text{LOCAL} \quad \boxed{1} \left[\text{HEAD|CASE} \quad \textit{case} \right] \\ \text{NON-LOCAL} \left[\begin{array}{l} \text{CLITICS} \quad \langle \rangle \\ \text{SLASH} \quad \boxed{1} \end{array} \right] \end{array} \right] \end{array} \right]$$

Figura 4.23: *focus-trace*

Matrix, en principio, solo cuenta con una huella (caracterizada como *gap*). Si en SSG se usan dos (*focus-trace* y *topic-trace*), dado que ninguna tiene realización fonética, el *parsing* tendrá que contemplar, a ciegas, dos proyecciones de cada verbo donde *Matrix* solo contemplaba una. Pero ¿es esta multiplicación de posibilidades fruto, propiamente, del modelado de los clíticos? Como se verá con detalle en el apartado 5.3, en español, si se quiere dar cuenta de forma apropiada de los distintos tipos de movimiento, es necesario contemplar dos tipos de huella distintos. Esta necesidad es independiente de la

aparición o no de clítico. Por tanto, las unidades léxicas que usa SSG para dar cuenta de los clíticos están justificadas de forma independiente. En cualquier caso, sería necesario, para dar cuenta de los datos del español, la existencia de *personal-pronoun*, *topic-trace* y *focus-trace*. Este análisis, por tanto, no hace uso ni de reglas léxicas, ni de unidades léxicas creadas *ad hoc* para añadir clíticos al verbo. Esta realidad será demostrada en 5.3.

Por último, como se dijo en 4.2, la construcción de pasiva refleja, la impersonal con *se* y los verbos pronominales piden la aparición de un clítico. El análisis que se hizo en 3.3 de estas construcciones¹² da solución inmediata al problema. Las reglas *v-rpass-c* y *v-pronominal-c* incluían en ARG-ST un argumento que no pasaba después a COMPS. Este argumento incluía un clítico en su lista CLITICS. Por tanto, dado el análisis presentado aquí, este argumento, aun no realizado sintácticamente por no pasar a COMPS, sí añade el clítico en cuestión al verbo. Estos argumentos tenían en 4.2 su razón de ser, con independencia de la aparición del clítico: el de *v-rpass-c* era el argumento externo que toda pasiva elimina —también presente en *v-passive-c*—. Y en *v-pronominal-c* se entendía que el clítico era el vestigio de un argumento interno, obviado en esta construcción, pero presente en los verbos reflexivos, similares a los pronominales.

En resumen, el fenómeno de los clíticos se resuelve en SSG mediante el uso de la lista CLITICS. Por un lado, en CLITICS se establecen los clíticos presentes en el significante del verbo. Estos se identifican en el preprocesamiento de la secuencia, y se codifican en CLITICS mediante la aplicación de reglas flexivas. Por otro lado, en CLITICS se suman los clíticos que los argumentos aportan al verbo. Obviamente, ambas caracterizaciones de la misma lista han de ser coherentes. Además, una unidad léxica aporta clítico a la lista CLITICS de un verbo, del que es argumento, si tiene codificado tal clítico en su propia lista CLITICS. Hay unidades léxicas que sí tienen codificado tal clítico y otras que

¹²La construcción impersonal, por ser marginal, no se ha tratado; pero podría modelarse siguiendo el mismo principio usado para la de pasiva refleja y la de verbos pronominales.

no. Para aquellas unidades que sí pueden aportar clítico, se ha visto cómo su caracterización es relativa al caso que recibe esta en su posición argumental.

Es labor de la codificación léxica de cada lengua establecer esta caracterización para sus unidades. Por tanto, en principio, este sistema es útil para cualquier lengua que deba modelar el fenómeno de los clíticos. Como es propio de HPSG, las diferencias de comportamiento dentro de la misma lengua, y entre lenguas diferentes, dependen de la caracterización de sus unidades léxicas.

4.4.3. El salto de clíticos

En 4.4.2 se ha dicho que es necesario que los clíticos observados en el significante de un verbo deben ser coherentes con los clíticos que a este aportan sus argumentos. Pero la relación entre estas dos realidades no es una cuestión trivial.

Como se vio en el apartado 4.2, en español se da lo que se ha llamado salto de clíticos: el clítico correspondiente al argumento de un verbo se explicita como afijo de otro verbo. Por ejemplo, en (19-b), el clítico *lo* es afijo de *querer*, pero se interpreta como el argumento de acusativo de *pagar*. Por tanto, *lo*, en tanto que clítico aportado por un argumento a un verbo, se codificará en la lista CLITICS de *pagar*; pero, en tanto que afijo, se codificará en la lista CLITICS de *querer*. Otro tanto ocurre en (19-c), donde el clítico *lo* salta desde *pagar*, no una, sino dos posiciones, hasta el auxiliar *haber*.

- (19) a. El hotel, queremos pagarlo al contado
 b. El hotel, lo queremos pagar al contado
 c. El hotel, lo hemos querido pagar al contado

Estos datos justifican la concepción de CLITICS como rasgo no local; pues, en las oraciones con salto de clíticos, el valor de CLITICS del verbo principal deberá trascender su dominio local, hasta encontrar en la lista CLITICS de un auxiliar o modal los afijos correspondientes. El salto de clíticos se establece,

entonces, en un dominio no local. Efectivamente, como se representa en la figura 4.24 para la oración de (19-c), este dominio está formado por un verbo finito que subcategoriza n verbos no finitos anidados.

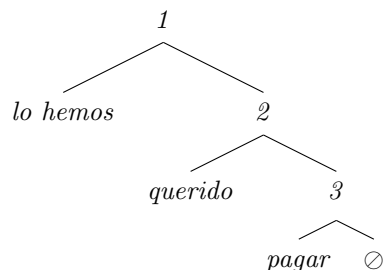


Figura 4.24: Dominio para el salto de clíticos

El ascenso de los clíticos dentro del dominio se capta automáticamente al ser CLITICS un rasgo no local. Como todo rasgo no local, CLITICS se define como la suma de las listas correspondientes de sus argumentos. En concreto, para los verbos modales o auxiliares, esa suma consiste en la identificación de la lista CLITICS del verbo con la lista correspondiente de su único complemento: un sintagma verbal en forma no personal. Gracias a ello, dentro de un dominio de salto de clíticos, los clíticos del verbo principal pueden ascender, en principio, hasta la lista CLITICS del límite del dominio: la del verbo finito. De este modo, es posible que cualquiera de los verbos del dominio explicita los afijos correspondientes a los clíticos del verbo principal, pues en cualquiera de ellos está disponible el valor de la lista CLITICS de dicho verbo.

Pero este sistema ha de complicarse si se quiere dar cuenta del hecho de que el salto de clíticos es un proceso de saturación. Si los argumentos del verbo principal aportan clíticos, estos deben explicitarse en algún punto del dominio. Si estos clíticos no se explicitan nunca, la oración es agramatical: (20-a). Además, todos los clíticos del dominio deben explicitarse en el mismo verbo: de ahí la gramaticalidad de (20-c) y (20-d), frente a la agramaticalidad de (20-e). Por último, una vez explicitado, no es posible una nueva aparición de un clítico en un verbo superior: (20-b).

- (20) a. *El hotel, queremos pagar al contado

- b. *El hotel, lo queremos pagarlo al contado
- c. El hotel, te lo queremos pagar al contado
- d. El hotel, queremos pagártelo al contado
- e. *El hotel, te queremos pagarlo al contado

En realidad, SSG confía en la existencia de un módulo de preprocesamiento —presente en SGP— que une en un único *token* todo lo que aquí se ha definido como posible dominio para el salto de clíticos: el verbo principal y sus n auxiliares o modales. Con ello, SSG mejora sensiblemente sus tiempos de procesamiento. Pero, además, esta solución elimina la necesidad de dar un tratamiento sintáctico al problema del salto de clíticos: estos han de encontrarse en cualquier punto del *token* formado por el verbo principal junto a sus auxiliares y modales; y cualquier restricción a su aparición se ha de tratar como un cuestión morfológica previa al procesamiento sintáctico. Esta no es una solución teórica adecuada, sino una simplificación útil para la elaboración de una herramienta de procesamiento eficaz. No obstante, SSG, como se muestra a continuación, es compatible con un modelado del salto de clíticos adecuado desde un punto de vista teórico.

Para dar cuenta adecuadamente del salto de clíticos, basta con asumir la formalización tradicional de la saturación de listas no locales en HPSG. En concreto, en Sag y otros (2002), las listas no locales se desglosan en dos listas: una representa el *input* y otra el *output* del proceso de saturación. El valor de las listas que denotan el *input* en un signo se definen como la suma de las listas que denotan el *output* correspondiente de sus argumentos. La saturación de la lista en un signo se representa estableciendo valores diferentes para el *input* y el *output*: el *input* hereda elementos de sus argumentos, pero estos no pasan al *output*; el *output* se define como lista vacía.

Este es el sistema de saturación de listas no locales clásico en HPSG¹³, perfectamente aplicable para modelar el salto de clíticos en SSG. Por ello,

¹³Las formalizaciones varían de una obra a otra —compárense Pollard y Sag (1994) y Sag y otros (2002)—. Pero la intuición formalizada es la misma.

la lista CLITICS de esta gramática debería desglosarse en dos: I-CLITICS y O-CLITICS, para la representación, respectivamente, del *input* y el *output* del proceso. De acuerdo con ello, los argumentos aportan a su verbo los clíticos que tienen pendientes de saturar: los codificados en O-CLITICS. El verbo, por su parte, cuando recibe esos clíticos, los suma como *input* en su lista I-CLITICS. Este proceso se observa en la figura 4.25 para los verbos con dos argumentos.

$$\left[\begin{array}{l} \textit{transitive-lex-item} \\ \text{SYNSEM|NON-LOCAL|I-CLITICS } \langle \boxed{1} \oplus \boxed{2} \rangle \\ \text{ARG-ST} \qquad \qquad \langle [\text{NON-LOCAL|O-CLITICS } \boxed{1}], [\text{NON-LOCAL|O-CLITICS } \boxed{2}] \rangle \end{array} \right]$$

Figura 4.25: Suma de CLITICS de los argumentos en el núcleo

Los verbos que no explicitan clíticos en su significante son aquellos que han sido flexionados con la regla flexiva **non-clitic-verb-irule**, que se representa en la figura 4.26.

$$\left[\begin{array}{l} \textit{non-clitic-verb-irule} \\ \text{SYNSEM|NON-LOCAL} \left[\begin{array}{l} \text{O-CLITICS } \boxed{1} \\ \text{I-CLITICS } \boxed{1} \end{array} \right] \end{array} \right]$$

Figura 4.26: non-clitic-verb-irule

Como se ve, en estos verbos se identifican las listas I-CLITICS y O-CLITICS. Es decir, en ellos, los clíticos del *input* pasan al *output*. Con ello se formaliza el que los clíticos aportados al verbo por los argumentos no se han saturado: quedan pendientes de saturar en un verbo superior del dominio. Efectivamente, cuando un segundo verbo toma al anterior como complemento, los clíticos de la lista O-CLITICS del complemento se suman a la lista I-CLITICS del verbo que lo subcategoriza. Con ello se formaliza el salto de los clíticos de un verbo hasta el modal o auxiliar que lo subcategoriza.

Mediante la iteración de este proceso de ascenso, los clíticos llegan a la lista I-CLITICS del verbo en el que se explicitan. Este verbo ha sido flexionado

con una regla —como *verb-se-lo-irule*, vista en el apartado 4.4.1— subtipo de *clitic-verb-irule*: véase este tipo en la figura 4.27. Estas reglas especifican el valor de la lista I-CLITICS del verbo. Por ejemplo, la regla *verb-se-lo-irule* caracteriza a I-CLITICS como una lista de dos clíticos de determinado tipo. Con ello, se establece el cotejo entre los clíticos aportados al verbo principal por sus argumentos y los afijos correspondientes. Una vez explicitados los clíticos, estos se consideran saturados dentro de su dominio, de modo que no pueden volver a aparecer en un verbo superior —(20-b)—. Para establecer esta restricción, *clitic-verb-irule* no pasa los clíticos de I-CLITICS a O-CLITICS, sino que evalúa esta última como lista vacía.

$$\left[\begin{array}{l} \textit{clitic-verb-irule} \\ \text{SYNSEM|NON-LOCAL|O-CLITICS } \langle \rangle \end{array} \right]$$

Figura 4.27: *clitic-verb-irule*

Por último, los verbos finitos son aquellos sobre los que se aplica una regla flexiva subtipo de *fin-verb-irule* —véase en la figura 4.28—. Al ser estos verbos límite del dominio, es necesario que su lista O-CLITICS sea una lista vacía, pues, alcanzado el límite del dominio, todo clítico ha debido ser saturado. Esta restricción se establece, efectivamente, en *fin-verb-irule*. Gracias a ella, se evita que en un dominio no lleguen nunca a saturarse los clíticos establecidos en la lista I-CLITICS del verbo principal.

$$\left[\begin{array}{l} \textit{fin-verb-irule} \\ \text{SYNSEM|NON-LOCAL|O-CLITICS } \langle \rangle \end{array} \right]$$

Figura 4.28: *fin-verb-irule*

En resumen, los clíticos que aportan los argumentos a un verbo principal no tienen por qué explicitarse como afijos de este. Es posible que se expliciten como afijos de alguno de los modales o auxiliares que forman lo que se ha llamado dominio de salto de clíticos. Esta posibilidad invita a entender la lista de codificación de clíticos como un rasgo no local. Gracias a ello, es posible que los clíticos asciendan de verbo en verbo hasta alcanzar aquel en el que

se explicitan. Este proceso de salto de clíticos es un proceso de saturación. Para ello, en cada verbo se debe establecer una lista I-CLITICS en la que el verbo recibe clíticos de sus argumentos, y otra O-CLITICS en la que el mismo verbo presenta los clíticos heredados de sus argumentos que no se explicitan en él. Si el verbo no cuenta con los afijos correspondientes, pasa los clíticos heredados en I-CLITICS a O-CLITICS. Si sí cuenta con tales afijos, los coteja con los clíticos heredados en I-CLITICS y los da por saturados dejando O-CLITICS vacía. Además, los verbos finitos (límite superior de todo dominio de salto de clíticos) tienen también la lista O-CLITICS vacía, con lo cual se evita que en un dominio no se expliciten los clíticos en ningún punto.

4.5. Problemas

El modelado de los clíticos que se ha presentado hasta el momento no cubre ciertos datos del español. En este apartado se explican estos fenómenos y se presenta la solución (provisional) que les da SSG. El análisis de 4.4.2 está fundamentado en el hecho de que un argumento aparece cuando se realiza en un determinado contexto. Estos contextos son la omisión del argumento, su topicalización o su realización como pronombre. Sin embargo, como se vio en 4.2, y se observa en las oraciones de (21), existen otros contextos en los que puede aparecer, concretamente, el clítico de dativo.

- (21)
- a. (Le) han dado el equipaje al botones
 - b. ¿A quién (le) han dado el equipaje?
 - c. El botones a quien (le) han dado el equipaje
 - d. ¡AL BOTONES (le) han dado el equipaje, no al guía!

Un argumento de dativo realizado en su posición canónica —como en (21-a)— puede llevar clítico. Esta posibilidad no está contemplada en el análisis de 4.4.2. Del mismo modo, los datos de (21) demuestran que las huellas que dejan en la posición de base ciertos movimientos de un argumento de dativo, distintos a la topicalización, también permiten la aparición del clítico: es el caso del movimiento de una partícula interrogativa —(21-b)—, un pronombre

relativo —(21-c)— o un foco —(21-d)—. Ninguno de estos clíticos está permitido tampoco por el análisis de 4.4.2.

De acuerdo con este análisis, en (21-a), las unidades léxicas que forman *al botones* (una preposición, un determinante y un nombre) no aportan clítico alguno al verbo. Por tanto, este análisis cubre la ausencia, posible, del clítico, pero no su presencia. Además, según se vio en 4.4.2, la huella de base de los movimientos de (21), en SSG, es *focus-trace*. Esta huella (a diferencia de *topic-trace*, la propia de las topicalizaciones) no añade clítico al verbo al que se une. Esto da cuenta de los casos en los que los movimientos de (21) no aportan clítico, pero no cubren la posibilidad de que ese clítico sí aparezca.

El único modo de dar cuenta de estas alternancias es crear unidades léxicas alternativas a las existentes que sí aporten clítico. Por ejemplo, frente a *focus-trace*, SSG debería contar con *focus-trace'*: una huella idéntica a aquella, útil para dar cuenta del mismo tipo de movimientos, pero que aporte al verbo un clítico de dativo. Esta huella solo unificaría, como parece deseable, con posiciones argumentales de dativo. El problema de este análisis es que la multiplicación de unidades léxicas favorece la explosión combinatoria. Y *focus-trace'* no está justificada más que por la aparición caprichosa del clítico de dativo en contextos distintos de los contemplados hasta ahora.

Por último, la oración de (21-a) pone a prueba la hipótesis defendida en este trabajo de que los clíticos son aportados al verbo por ciertas unidades léxicas, capacitadas para ello, que ocupan su posición argumental. En principio, no resulta fácil justificar que el clítico de (21-a) lo aporta el sintagma *al botones*. Desde un punto de vista técnico, no hay problema en establecer que la preposición de marca de caso dativo *a* puede aportar clítico al verbo. Bastaría, para ello, con contemplar la existencia de dos entradas léxicas para tal preposición (*a* y *a'*), cada una caracterizada de un modo en cuanto a este particular. Pero esta solución no es deseable. En primer lugar, esta multiplicación de entradas léxicas tiene un coste computacional. Y en segundo lugar, las unidades léxicas que hasta ahora aportaban clíticos eran de tipo nomi-

nal: pronombres y huellas. Es una solución *ad hoc* el añadido a ese grupo de la preposición *a*. Quizá pudiera estudiarse en un futuro la posibilidad de que el argumento de dativo de (21-a) haya sido saturado por una huella o pronombre; y el sintagma *al botones* esté legitimado en virtud de un esquema distinto a *comps-rule*.

No obstante, téngase en cuenta que las soluciones alternativas a la de SSG —las de 4.3— también tienen que multiplicar sus reglas léxicas si quieren hacer frente a estos datos. SSG, para dar cuenta de la mayor parte de los clíticos del español, no necesita crear ninguna regla, ni ninguna unidad léxica cuya existencia no esté justificada de forma independiente. Para dar cuenta de los clíticos de dativo de este apartado, sí crea *ad hoc* las unidades léxicas *focus-trace'* y *a'*. Pero todas estas unidades léxicas (o sus reglas correspondientes) deben aparecer también, de un modo u otro, en los análisis alternativos. A esa complejidad, estos análisis añaden reglas léxicas —y gramaticales— que se evitan en SSG.

4.6. Conclusión

En este capítulo se ha mostrado un análisis de los clíticos del español novedoso y económico, desde el punto de vista del coste computacional. Los análisis previos en HPSG de los fenómenos de cliticización descansaban en el uso de reglas léxicas. Esta estrategia suponía un aumento de la explosión combinatoria y del gasto computacional de la gramática. El análisis de SSG evita el uso de regla léxica alguna.

Para ello, se entiende que los clíticos que lleva un verbo deben ser especificados en este, antes del proceso sintáctico, como parte de su flexión morfológica. La presencia o ausencia de unos determinados clíticos en un verbo se entiende, entonces, como una especificación morfológica que recibe este mediante reglas flexivas, y que restringirá las posibilidades de realización sintáctica de sus argumentos. Efectivamente, después, en el proceso de satu-

ración, un verbo solo admitirá como argumentos a aquellos signos que resulten compatibles con dicha caracterización morfológica previa. En concreto, esta caracterización se ha formalizado como una lista de clíticos: CLITICS. La mayor parte de las unidades léxicas de la gramática no aportan ningún clítico a la lista CLITICS del verbo al que se unen como argumentos. En cambio, los pronombres personales (omitidos o no) y las huellas de topicalización, si tienen caso acusativo o dativo, sí aportan un clítico correspondiente a la lista CLITICS del verbo al que se unen. Estos clíticos aportados por los argumentos del verbo deben ser compatibles con la caracterización que de CLITICS hizo la regla flexiva que se aplicó a tal verbo.

En definitiva, SSG contempla el de los clíticos como un fenómeno morfológico. En concreto, se entienden los clíticos como afijos cuya aparición en el verbo viene desencadenada por la unión a este de un determinado tipo de unidad léxica: pronombre personal, omitido o no, o huella de topicalización. Así se da cuenta del hecho de que la clitización de un argumento tiene lugar cuando este se omite, cuando se pronominaliza o cuando se topicaliza.

Este tratamiento morfológico hace necesario el uso de un módulo de preprocesamiento que une en un solo *token* al verbo y sus clíticos, y establece qué regla flexiva debe codificar la presencia de tales clíticos en el verbo. SGP cuenta con tal módulo. Aparte del derivado del uso de este módulo de preprocesamiento, el tratamiento que de los clíticos hace SSG no acarrea coste computacional significativo a la gramática. Obsérvese cómo para dar cuenta de él no se ha establecido regla léxica alguna, ni reglas de proyección. Sí se ha contado con la existencia de unidades léxicas concretas que aportan clíticos. Pero estas las debe contemplar, de un modo u otro, cualquier gramática del español, con independencia del análisis de los clíticos que haga.

Capítulo 5

SSG: orden de palabras

El español es una lengua de orden de palabras relativamente libre. Igual que ocurría con las posibilidades de clitización vistas en 4, la libertad de ordenación lineal agrava el problema de la explosión combinatoria. Este capítulo pretende explicar cómo el análisis que hace SSG de dicha libertad reduce el coste computacional que asumen las soluciones previas al problema. Se tratarán a continuación tres cuestiones centrales relativas al orden de palabras del español: la ubicación del sujeto en relación al verbo (apartado 5.1), la libertad de ordenación relativa entre los complementos (apartado 5.2) y los distintos tipos de dependencias no acotadas (apartado 5.3).

El criterio de orden lineal de constituyentes ha sido muy utilizado en la creación de tipologías lingüísticas. ¿Qué tipo de lengua es el español, en lo que a orden de palabras se refiere? Se sigue, para dar respuesta a esta pregunta, a Fernández Soriano (1993). El español es una lengua configuracional de orden básico SVO (sujeto, verbo, objeto), con una libertad de ordenación lineal de constituyentes relativamente grande (en comparación con lenguas de su entorno, como el francés o el inglés).

Greenberg (1963) estableció una tipología de lenguas en función de la ordenación básica de sus constituyentes oracionales: sujeto, objeto y verbo. En esta tipología se hace una primera distinción entre lenguas configuracionales

y lenguas no configuracionales. En las lenguas configuracionales, los constituyentes de la oración se ordenan conforme a una cierta estructura sintáctica. Como se observa en la figura 5.1, en estas lenguas, los distintos argumentos de un núcleo se relacionan con este de un modo distinto desde el punto de vista estructural. El complemento se une en primera instancia al núcleo desnudo. Y el especificador se une, después, al complejo formado por el núcleo desnudo y su complemento.

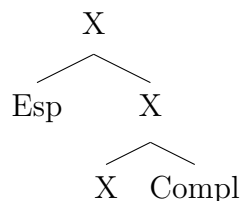


Figura 5.1: Lenguas configuracionales

Las lenguas configuracionales pueden tener más o menos libertad de ordenación lineal, pero siempre están constreñidas por la estructura: los constituyentes de un nodo estructural no pueden linealizarse de forma discontinua. Por ejemplo, dada la estructura de la figura 5.1, el especificador puede aparecer a derecha o izquierda del constructo formado por núcleo y complemento; pero nunca podrá aparecer entre los constituyentes de este. En cambio, las lenguas no configuracionales no están sujetas a una estructura como la de la figura 5.1, y su ordenación lineal no estará, por tanto, constreñida por ella. Estas lenguas permiten gran libertad de ordenación de elementos. Por ejemplo, estas lenguas permitirían, en principio, la aparición de lo que en la figura 5.1 se ha llamado especificador, entre el núcleo y el complemento.

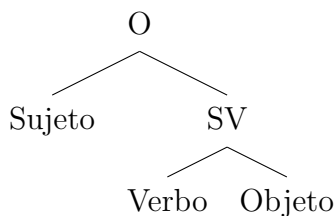


Figura 5.2: Sintagma verbal

Concretamente, las lenguas configuracionales cuentan con un constructo

al que pudiera llamarse sintagma verbal. Es el resultado de la unión del verbo con su complemento, el objeto. El sujeto se une, no al verbo propiamente dicho, sino al sintagma verbal completo. Así pues, la figura 5.2 muestra un ejemplo del esquema general de 5.1. En una lengua no configuracional, no existe un constructo de este tipo.

Según Fernández Soriano (1993), el español pertenece al grupo de las lenguas configuracionales. Presenta cierta libertad de ordenación de constituyentes, pero siempre a partir de un orden básico, canónico, establecido por la estructura sintáctica. Es decir, en principio, en español, sea cual sea el orden de sujeto, verbo y objeto, estos no pueden ordenarse de forma tal que los constituyentes del sintagma verbal (verbo y objeto) sean separados por un elemento externo a él (el sujeto).

Dentro de las lenguas configuracionales, se pueden establecer varios tipos en función de dos criterios. El primero es la posición en que estas lenguas linealizan el complemento con respecto al núcleo. Hay lenguas configuracionales que colocan el complemento delante de su núcleo y otras que lo colocan detrás. Parece claro que el español coloca al complemento detrás de su núcleo. Además, la posición canónica de aparición del sujeto en español (para verbos con sujeto y objeto) es la antepuesta al verbo. Por tanto, el español es una lengua configuracional que obedece a un orden básico SVO (sujeto, verbo, objeto).

Pero, además, dentro de las lenguas configuracionales, están aquellas que siguen su orden básico de constituyentes de forma estricta. Y otras que permiten, a partir de ese orden básico de constituyentes, cierta dislocación. Se ha llamado a las primeras lenguas de orden fijo, y a las segundas lenguas de orden libre. Se observa que las lenguas de orden libre cuentan con una morfología y un sistema de preposiciones mucho más rico que las lenguas de orden fijo. La cuestión es que, de uno u otro modo, es necesario que sea reconocible el papel temático que el núcleo asigna a sus argumentos en la oración. En una lengua de orden fijo, ese papel temático es reconocible porque el argumento

se ubica de forma estrictamente adyacente al núcleo que se lo asigna. En una lengua de orden libre, el papel temático que asigna el núcleo al argumento se reconoce porque este último está marcado con una preposición, una marca morfológica o una relación de concordancia que lo identifique. Esto permite que el argumento en cuestión se linealice alejado de su núcleo. El latín era una lengua de morfología muy rica. La identificación del papel temático de los sintagmas nominales se aseguraba para esta lengua mediante el uso de marcas morfológicas. Esto le permitía una gran libertad de ordenación de sus constituyentes. Las lenguas romances han perdido, en mayor o menor medida, estas marcas morfológicas, y, en consecuencia, la libertad de ordenación lineal. No obstante, el español, si se compara con el francés, conserva cierta libertad de ordenación lineal, gracias a su morfología rica y su sistema de preposiciones.

En resumen, el español es una lengua configuracional de orden básico SVO, más fijo que el latín, pero bastante menos que el francés o el inglés.

5.1. El sujeto

En el apartado 3.3, se mostró cómo los sujetos en español pueden ser tanto argumentos internos (en los verbos inacusativos o pasivos) o externos (en los verbos inergativos o transitivos). Se entendió entonces que la posición canónica de los sujetos consistentes en argumentos internos era la posverbal, lo cual se formalizó colocando tales sujetos en la lista COMPS. En cambio, la posición canónica de los sujetos consistentes en argumentos externos es la preverbal, de modo que estos fueron colocados en la lista SUBJ.

No obstante, ciertos contextos piden, o facilitan, la inversión de estos argumentos externos sujeto. Las oraciones interrogativas —(1-a)—, la anidación de la oración —(1-b)—, o la presencia de un tópico —(1-c) y (1-d)— facilitan la inversión.

- (1) a. ¿Han visitado ellos este museo?

- b. Creo que ya han visitado ellos el museo
- c. El museo ya lo han visitado ellos
- d. A las 12:00, ya habían visitado ellos el museo

Además, en el español de España, la aparición de un foco, o de una partícula interrogativa, obliga a la inversión del sujeto (o a su topicalización): (2).

- (2) a. *!ESE MUSEO ellos han visitado!
- b. !ESE MUSEO han visitado ellos!
- c. *¿Qué museo ellos han visitado?
- d. ¿Qué museo han visitado ellos?

En HPSG, se han formalizado estos casos de inversión de sujeto haciendo uso de una regla léxica —véase Sag y otros (2002)—. Esta regla, llámese *inversion*, se ha entendido siempre como una regla destructiva: altera el *input*, pasando el sujeto de su posición canónica en SUBJ, a COMPS. Se entiende que, desde COMPS, se realiza pospuesto al verbo. Ya se explicó en 3.6 que SSG no utiliza reglas destructivas, por razones teóricas y prácticas. SSG podría utilizar, no obstante, una versión no destructiva de *inversion*. Efectivamente, *inversion* podría entenderse como un patrón de comportamiento verbal, solo para verbos con argumento externo —alternativo, entonces, a *v-transitive-c* o *v-unergative-c*—. Este patrón definiría la ubicación del sujeto en COMPS, tal como se muestra en la figura 5.3.

$$\left[\begin{array}{l} \textit{inversion} \\ \text{SYNSEM|LOCAL|VAL} \left[\begin{array}{l} \text{SUBJ} \langle \rangle \\ \text{COMPS} \boxed{1} \end{array} \right] \\ \text{ARG-ST} \quad \boxed{1} \langle \textit{external}, \dots \rangle \end{array} \right]$$

Figura 5.3: Hipotética regla *inversion*

Obsérvese que *inversion*, tal como se muestra en la figura 5.3, y tal como se entiende habitualmente en HPSG¹, con reglas gramaticales como *comps-rule*,

¹Sag y otros (2002).

da lugar a una derivación como la de la figura 5.4. En ella, el sujeto es el primer elemento en unirse al verbo, y el objeto se une después al constructo formado por verbo y sujeto. Esta derivación no es compatible con la asunción de que, en las lenguas configuracionales, existe un sintagma verbal, formado por la unión inmediata de verbo y objeto.

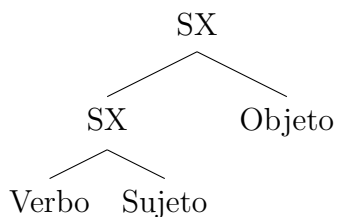


Figura 5.4: Derivación con inversión de sujeto en HPSG

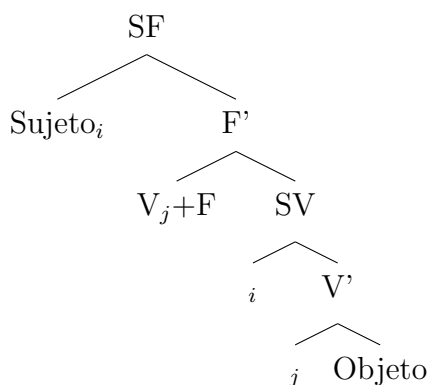


Figura 5.5: Derivación en el modelo chomskyano (I)

En el modelo gramatical chomskyano, es fácil preservar, en principio, la existencia del sintagma verbal en los casos de inversión. Este modelo contempla la existencia de categorías funcionales que toman a la proyección máxima del núcleo léxico como complemento. Por ejemplo, en el modelo de Principios y Parámetros², la oración se entiende como la proyección de una categoría *F* (flexión), cuyo complemento es la proyección máxima de un verbo *V*. El sujeto del verbo se genera en la posición de especificador de *V*, donde recibe papel temático de este. Después se mueve a la posición de especificador de

²El análisis siguiente está inspirado en Eguren y Fernández Soriano (2004), pp. 152-161.

F en busca de caso nominativo (el caso nominativo se marca en español mediante concordancia del argumento con la flexión verbal). Además, el núcleo V se mueve a la posición de núcleo de F (y de ahí la realización sincrética del verbo con la flexión). Véase todo ello en la figura 5.5. Obsérvese cómo este análisis permite dar cuenta del orden de constituyentes canónico SVO del español, de forma consistente con las constricciones estructurales propias de una lengua configuracional.

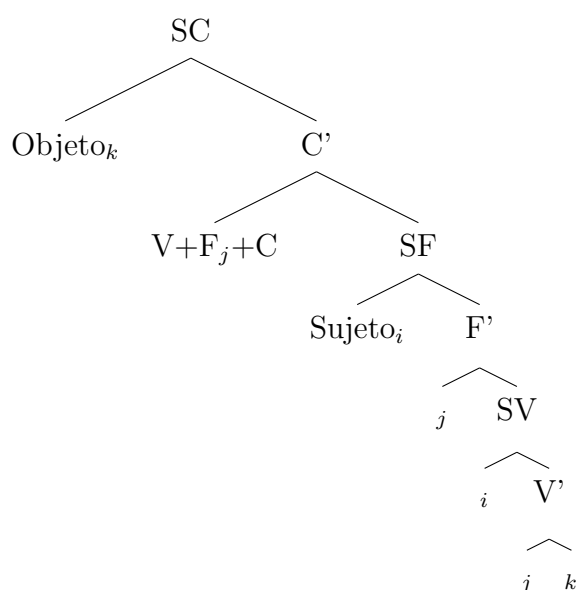


Figura 5.6: Derivación en el modelo chomskyano (II)

En el modelo de Principios y Parámetros se contempla además la existencia de otras categorías funcionales. La categoría C (complementante, definitoria de la modalidad oracional) toma como complemento la proyección máxima de F. Y su posición de especificador es el lugar de destino de ciertos movimientos, por ejemplo, los de partículas interrogativas. Así, el objeto de una oración se coloca, en forma de partícula interrogativa, efectivamente, como especificador de C. Así, en las oraciones interrogativas, el núcleo verbal asciende de F a C³. Véanse estos movimientos

³Eguren y Fernández Soriano (2004), p. 185 n. 95. Este análisis se ha propuesto, por ejemplo, para dar cuenta del orden inverso entre verbo y sujeto de las interrogativas totales del inglés. No se afirma aquí que sea válido para el español. Solo se conjetura con

en la figura 5.6.

La estructura de la figura 5.6 sugiere un orden de palabras OVS para este tipo de oraciones —como (2-d)—. Con ello, se da cuenta del hecho de que, en español, el sujeto de las oraciones interrogativas se puede posponer al verbo. Y se hace respetando las constricciones estructurales de una lengua configuracional: se crea un sintagma verbal, y son ciertos movimientos posteriores los que legitiman la inversión del sujeto.

No se pretende aquí hacer una explicación rigurosa o exhaustiva de los datos de (1) de acuerdo con la gramática chomskyana. Solo se pretende mostrar cómo su naturaleza transformacional y la existencia de categorías funcionales permiten explicar los casos de inversión; preservando la premisa de que, en una lengua configuracional, el objeto forma junto al verbo una primera proyección, a la que el sujeto se une después.

HPSG no contempla la existencia de categorías funcionales. En HPSG, el núcleo léxico dirige siempre la derivación. Esta es una asunción fundamental de la teoría —de ahí su nombre: *Head-Driven Phrase Structure Grammar*—. La información que en la gramática chomskyana se formaliza mediante categorías funcionales se recoge en HPSG como especificaciones de la estructura de rasgos del núcleo léxico. Por tanto, los núcleos léxicos son el eje de la derivación, y sus argumentos se pueden linealizar solo a derecha o izquierda de estos: no hay lugar para movimiento de núcleo a través de unas categorías funcionales que no existen.

Por tanto, se asume en este trabajo que esta limitación es inherente al modelo utilizado; y no se le intenta, por ello, dar solución. Sí concierne, en cambio, a este trabajo, reflexionar sobre el coste computacional del añadido a SSG de una regla como *inversion*.

tal posibilidad.

Al crear una gramática computacional del español, dados los datos de (1), es realmente complicado discriminar en qué contextos se puede o no se puede aplicar la inversión. En esta lengua, como demuestran (1-b), (1-c) y (1-d), la inversión no es un fenómeno ceñido a las oraciones interrogativas. En la gramática transformacional es comúnmente aceptado que la posición estructural original de los sujetos es interna al sintagma verbal⁴. Desde esa posición original, el sujeto asciende y se coloca como especificador de la categoría funcional T, lo cual explica su posición preverbal canónica. Se entiende que este ascenso está motivado por una necesidad intrínseca de T: la de contar con un especificador⁵. Cabe imaginar que, si esta necesidad fuese cubierta por un elemento distinto al sujeto, el ascenso de este sería innecesario. En tales casos, el sujeto se linealizaría en posición posverbal. Efectivamente, Zubizarreta (1998) o Gutiérrez-Bravo (2008) asumen que la posición de especificador de T puede estar ocupada también por tópicos, focos o partículas interrogativas. Según estas propuestas, cuando la necesidad de especificador de T está cubierta por estos elementos, el sujeto no asciende, y se linealiza pospuesto al verbo⁶.

Esta formalización no es aplicable a HPSG. Pero sí es útil la intuición consistente en que la posición original del sujeto es interna al sintagma verbal, y desde allí, posverbal; y la idea de que es la satisfacción de un principio ajeno a la mera proyección del núcleo verbal lo que explica la anteposición del sujeto. No parece estar justificado en SSG que este principio sea una necesidad oracional abstracta, como la del modelo transformacional. ¿Podría ser, en cambio, una necesidad informativa? ¿Quizá la necesidad de que, dado cierto

⁴Se ha llamado a esta asunción Hipótesis del sujeto interno al sintagma verbal: Eguren y Fernández Soriano (2004), pp. 160-169.

⁵Esta necesidad se ha formalizado en el modelo de Principios y Parámetros con el llamado PPA —Principio de Proyección Ampliado—. Véase Eguren y Fernández Soriano (2004), p. 129.

⁶Estas ideas no están reñidas con el hecho de que T asigne caso nominativo al sujeto. La asignación de caso y el movimiento al especificador de T son procesos independientes. Cuando el sujeto permanece en el interior del sintagma verbal, la asignación de caso a este sigue haciéndola T a distancia (Eguren y Fernández Soriano (2004), p. 294 n.).

patrón enunciativo, es necesaria la presencia a la izquierda del predicado de un tema, un elemento conocido sobre el que predicar? Cuando ese patrón enunciativo no existe —(1-a), (1-b) y (3)—, o cuando la función de tema la cubre otro elemento —(1-c) y (1-d)—, la anteposición del sujeto no sería necesaria y no tiene lugar.

(3) Ha empezado la visita

Todo ello, sin pasar de ser una hipótesis pendiente de demostrar, hace plausible el análisis de los sujetos que hace SSG. Efectivamente, en SSG se asumirá que todos los sujetos se colocan originariamente en COMPS, desde donde pueden realizarse en posición posverbal. Esto da cuenta de los datos de (1). Cualquier anteposición del sujeto se concibe, entonces, como fruto de una topicalización. Gracias a este análisis, SSG prescinde de la regla *inversion*, y se hace inútil la existencia de *subj-rule*. De este modo, todos los elementos linealizados a la derecha de un verbo se saturarán mediante *comps-rule*, y todos los linealizados a su izquierda se saturarán mediante la regla de realización de tópicos: *topic-rule*, como se verá en 5.3. Por tanto, el tipo *unergative*, del que dependen los patrones *v-transitive-c* y *v-unergative-c* se reformula conforme a lo que muestra la figura 5.7.

$$\left[\begin{array}{l} \textit{unergative} \\ \text{SYNSEM|LOCAL|VAL} \left[\begin{array}{l} \text{SUBJ} \langle \rangle \\ \text{COMPS} \underline{\mathbb{1}} \end{array} \right] \\ \text{ARG-ST} \quad \underline{\mathbb{1}} \langle \text{LOCAL } \textit{external}, \dots \rangle \end{array} \right]$$

Figura 5.7: Relación entre ARG-ST y VAL con argumento externo

La colocación del sujeto en COMPS soluciona ciertos problemas a los que se enfrenta cualquier gramática del español. Un sujeto, igual que un objeto, puede topicalizarse —(4-a)—. Para una gramática del español que sature sus sujetos desde SUBJ, a la izquierda del verbo, esto supone una ambigüedad

sistemática⁷. Cuando un sujeto está inmediatamente antepuesto a su verbo, puede haberse realizado desde SUBJ, o a través de una topicalización, como se representa en (4-b). El análisis de SSG hace desaparecer este problema.

- (4) a. [Ellos_i [creo que t_i ya han visitado el museo]]
 b. [Ellos_i [t_i ya han visitado el museo]]

Además, este análisis de SSG soluciona ciertas complicaciones relativas al modelado de los auxiliares. Los auxiliares en HPSG se han entendido habitualmente como estructuras de ascenso: tomaban como complemento un sintagma verbal con el sujeto pendiente de saturar, e identificaban ese sujeto con el suyo propio. Este proceso de ascenso puede iterarse de auxiliar en auxiliar, hasta que el sujeto del verbo principal llega a la lista SUBJ de un verbo finito, con rasgos de concordancia. Allí se establece la concordancia entre sujeto y flexión oracional; y desde allí el sujeto se satura, a la izquierda de todos los auxiliares. Pero, como se ha visto, un verbo puede sufrir, o no, inversión. Si esta se entiende como el paso del sujeto desde SUBJ hasta COMPS, un auxiliar podría tomar como complemento tanto un verbo con la lista SUBJ vacía como un verbo con la lista llena. Esta dualidad, de un modo u otro, pide contemplar en la gramática dos tipos de auxiliares: uno debe desencadenar ascenso del sujeto, y otro no.

SSG neutraliza automáticamente estos dos tipos de auxiliares, pues la lista SUBJ de sus verbos siempre está vacía. En SSG, la anteposición del sujeto al verbo y a sus auxiliares es un caso de topicalización; y su posposición es un caso de realización canónica desde la lista COMPS del verbo principal. En ambos casos, la concordancia existente entre el sujeto y la flexión del auxiliar finito se establece mediante ascenso exclusivo de los rasgos pertinentes: XARG. En concreto, en SSG un verbo auxiliar —como se muestra en la figura 5.8— toma como complemento a la proyección máxima de un verbo no finito (FORM

⁷Solo se evita la ambigüedad entre la realización canónica del sujeto desde SUBJ y su topicalización, si entre el sujeto y su verbo media otro elemento desplazado; o si el sujeto se desplaza a la margen izquierda desde una oración anidada.

no-fin).

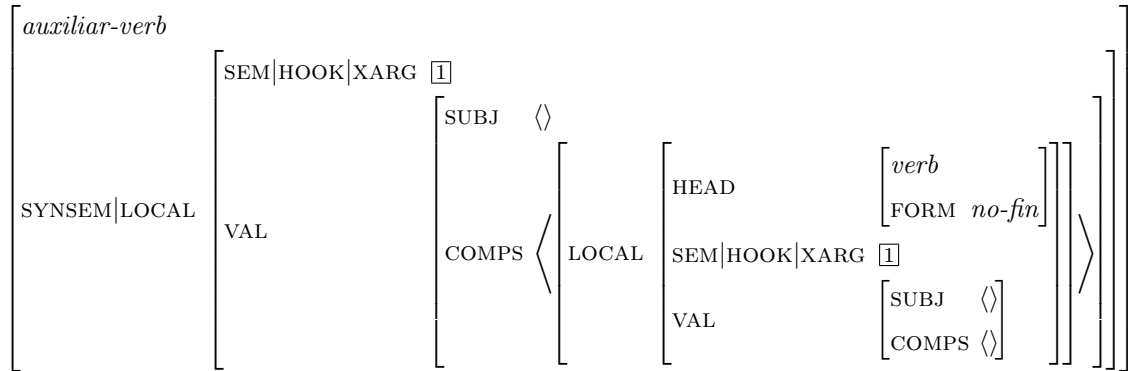


Figura 5.8: Verbo auxiliar en SSG

Por tanto, los sujetos se saturan siempre en el verbo principal (ya sea como sintagmas posverbiales o como huellas de desplazamiento), y no es necesario que asciendan a los auxiliares. El único ascenso que tiene lugar es el del índice semántico de dicho sujeto. Este ascenso se formaliza estableciendo identidad entre el índice XARG del auxiliar y el de su complemento⁸. Una vez el índice del sujeto ha ascendido (en forma de rasgo XARG) hasta un auxiliar finito, en este se establece la concordancia entre el sujeto y la flexión oracional. Para ello, los verbos flexionados se definen en SSG como instancias de *agree-lex-item* —figura 5.9—.

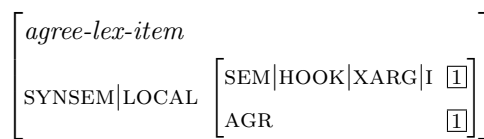


Figura 5.9: *agree-lex-item*

⁸Este índice ha sido identificado en el verbo principal con el índice de su argumento más prominente: el argumento externo de un verbo inergativo o transitivo; o el argumento interno de un verbo inacusativo o pasivo. Es decir, en definitiva, XARG recoge el índice del sujeto del verbo principal.

5.2. *Scrambling*

En 5.1, se explicaron las posibilidades de ordenación lineal del sujeto en español. En este apartado se hace otro tanto con los objetos. De momento se contemplará la existencia de objetos de caso acusativo y dativo. Estos, colocados siempre en COMPS, tienen posición canónica posverbal.

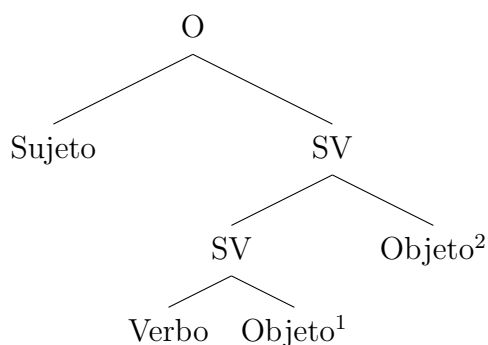


Figura 5.10: Sintagma verbal con dos proyecciones

Como el español es una lengua configuracional, un sintagma verbal con dos objetos habrá de organizarse en varios niveles de proyección, como se muestra en la figura 5.10. Además, dado este carácter configuracional, es esperable una ordenación canónica de los complementos internos al sintagma verbal. Concretamente, la estructura sintáctica de 5.10 impide la aparición del segundo objeto entre el verbo y el primer objeto; pues esta ordenación haría discontinua la linealización de la primera proyección del verbo.

Pero, como ya se vio en 2.1, dentro de tal dominio posverbal, los dos objetos tienen gran libertad de ordenación relativa. Se repiten aquí los ejemplos de aquel apartado. En (5) se observa cómo el objeto de acusativo y el de dativo pueden linealizarse en cualquier orden relativo. De acuerdo con Fernández Soriano (1993), estas diferencias de ordenación lineal no acarrearán diferencias interpretativas.

- (5) a. El botones le ha subido el equipaje al guía
 b. El botones le ha subido al guía el equipaje

Esta autora sugiere que estos datos pueden explicarse como el resultado de un fenómeno de *scrambling* o redistribución: los constituyentes de una determinada estructura sintáctica pueden desordenarse. Esta será, entonces, la solución adoptada en SSG para los constituyentes del sintagma verbal. El *scrambling* se ha modelado en HPSG de diferentes modos. Prácticamente todas las soluciones utilizadas en las gramáticas computacionales consisten en la estipulación en reglas *ad hoc*; y acarrear problemas de ambigüedad y falta de eficiencia computacional. Se explican estas soluciones en 5.2.1. SSG, en cambio, adopta la solución más extendida en los trabajos teóricos en HPSG para la formalización de este fenómeno. Es una solución que, por razones técnicas, no ha sido adoptada habitualmente en las gramáticas computacionales. Su aplicación a una gramática del español es novedosa. Se explicará en 5.2.2.

5.2.1. Soluciones previas

Según Müller (2004), las soluciones que se han dado al fenómeno del *scrambling* en HPSG son las siguientes. Una posibilidad es utilizar reglas sintácticas que saturen en un mismo paso todos los complementos. De este modo, se puede estipular directamente cada orden posible en una regla diferente. Las dos reglas necesarias para dar cuenta de los órdenes posibles de los dos objetos de un verbo en español obedecerían a la caracterización de las figuras 5.11 y 5.12.

$$\left[\begin{array}{ll} 1\text{-flat-comps-rule} & \\ \text{STEM} & \langle \boxed{1}, \boxed{2}, \boxed{3} \rangle \\ \text{SYNSEM|LOCAL|VAL|COMPS} & \langle \rangle \\ \text{HEAD-DTR} & \boxed{1} \left[\text{SYNSEM|LOCAL|COMPS} \langle \boxed{2}, \boxed{3} \rangle \right] \\ \text{NON-HEAD-DTR-1} & \boxed{2} \\ \text{NON-HEAD-DTR-2} & \boxed{3} \end{array} \right]$$

Figura 5.11: *Scrambling* con reglas mononivel (I)

Estas dos reglas saturan al tiempo los dos complementos del núcleo. La única diferencia entre ambas se encuentra en que se presupone una realización

<i>2-flat-comps-rule</i>	
STEM	$\langle \boxed{1}, \boxed{3}, \boxed{2} \rangle$
SYNSEM LOCAL VAL COMPS	$\langle \rangle$
HEAD-DTR	$\boxed{1}[\text{SYNSEM LOCAL COMPS } \langle \boxed{2}, \boxed{3} \rangle]$
NON-HEAD-DTR-1	$\boxed{2}$
NON-HEAD-DTR-2	$\boxed{3}$

Figura 5.12: *Scrambling* con reglas mononivel (II)

fonética distinta en STEM⁹.

Esta solución es muy poco elegante, pues implica crear *ad hoc* una regla para cada orden posible. Es más, con este sistema, habría que estipular nuevas reglas por cada posible número de elementos que pudiera contener una lista. Por ejemplo, las unidades léxicas transitivas y las ditransitivas saturarían sus complementos mediante reglas diferentes. Del mismo modo, si la gramática contempla listas de complementos con tres elementos, serían necesarias cuatro reglas más. Por último, no parece posible, con reglas de este tipo, intercalar modificadores del núcleo entre los complementos.

Por tanto, parece conveniente saturar los elementos de una lista de subcategorización mediante la reiterada aplicación de la misma regla. Para modelar el *scrambling* de este modo es necesario que las reglas puedan saturar un complemento cualquiera, esté este o no en la primera posición de la lista de subcategorización. Esto se puede lograr entendiendo que los complementos se agrupan, no en una lista, sino en un conjunto. La regla puede saturar así elementos en un orden aleatorio. El problema de este planteamiento es que genera ambigüedades indeseables cuando el núcleo se realiza en medio de sus complementos.

⁹En realidad, el orden lineal de constituyentes esperable en la cadena de entrada se establece normalmente en el signo en una lista **ARGS**. Esta es una cuestión técnica que resulta poco intuitiva y complica la discusión. Se presupone, entonces, que la lista **STEM** refleja el orden de constituyentes tal como se encuentra en la cadena de entrada.



Figura 5.13: Ambigüedad indeseable

Imagínese que un núcleo H tiene dos complementos en su conjunto de subcategorización: R y L . Como estos complementos están organizados en un conjunto, no se ha establecido entre ellos un orden de realización. Imagínese que R se realiza a su derecha y L a su izquierda. Esto podría lograrse realizando primero R y después L , o a la inversa —véase la figura 5.13—. Se da, por tanto, una ambigüedad indeseable.

Esta situación tiene lugar en una gramática del español, por ejemplo, cuando uno de los complementos se topicaliza y el otro no. Imagínese que L es el complemento que se va a topicalizar y R el que no. En tal caso, una regla de extracción eliminaría a L del conjunto de subcategorización (para que al final de la derivación pudiera realizarse a la izquierda) y otra realizaría a R a la derecha de H . Estos dos pasos podrían darse en un orden u otro, lo cual da lugar a la conocida ambigüedad.

<i>1-set-comps-rule</i>	
STEM	$\langle \boxed{3}, \boxed{1} \rangle$
SYNSEM LOCAL VAL COMPS	$\langle \boxed{2} \rangle$
HEAD-DTR	$\boxed{3} \left[\text{SYNSEM LOCAL COMPS} \langle \boxed{1} . \boxed{2} \rangle \right]$
NON-HEAD-DTR	$\boxed{1}$

Figura 5.14: *Scrambling* con conjuntos (I)

Además, no es posible implementar un conjunto directamente con la sintaxis utilizada habitualmente para el desarrollo de gramáticas computacionales —TDL—. Lo que hacen las gramáticas computacionales que intentan representar un conjunto es definir este como lista de n posiciones, asociada

a n reglas de saturación. Cada posición i es saturada por la correspondiente regla i . Por ejemplo, para realizar los complementos de un verbo ditransitivo, Matrix utiliza las dos reglas de las figuras 5.14 y 5.15.

$$\left[\begin{array}{l} 2\text{-set-comps-rule} \\ \text{STEM} \quad \langle \boxed{4}, \boxed{2} \rangle \\ \text{SYNSEM|LOCAL|VAL|COMPS} \quad \langle \boxed{1} . \boxed{3} \rangle \\ \text{HEAD-DTR} \quad \boxed{4} \left[\text{SYNSEM|LOCAL|COMPS} \langle \boxed{1}, \boxed{2} . \boxed{3} \rangle \right] \\ \text{NON-HEAD-DTR} \quad \boxed{2} \end{array} \right]$$

Figura 5.15: *Scrambling* con conjuntos (II)

Estas reglas funcionan en buena medida del mismo modo: eliminan de la lista COMPS del núcleo el elemento que se realiza, y dejan en tal lista el resto de elementos. Pero se diferencian en que cada una realiza un elemento de la lista COMPS. Esta solución también pide, por tanto, la estipulación de varias reglas: tantas como número de elementos puedan contener las listas de la gramática. No es, entonces, una solución elegante.

Un tercer modo de afrontar el problema del *scrambling* es usar reglas léxicas. Cada regla léxica daría cuenta de una ordenación posible de los complementos de la palabra. Así pues, un verbo ditransitivo en español pudiera concretarse mediante una regla léxica que colocase su argumento acusativo en primer lugar de la lista COMPS y el dativo en segundo; o mediante otra regla que colocase tales argumentos en orden inverso. La primera tendría el aspecto que se muestra en la figura 5.16 y la segunda el de la figura 5.17.

$$\left[\begin{array}{l} ac\text{-dat-ditr-lex-rule} \\ \text{SYNSEM|LOCAL|VAL|COMPS} \quad \langle \boxed{1}, \boxed{2} \rangle \\ \text{ARG-ST} \quad \langle \textit{synsem}, \boxed{1} [\text{CASE } \textit{acc}], \boxed{2} [\text{CASE } \textit{dat}] \rangle \end{array} \right]$$

Figura 5.16: *Scrambling* con reglas léxicas (I)

En estas reglas se supone un orden de oblicuidad fijo entre los argumentos

de un verbo, que se recoge en la lista ARG-ST. La diferencia entre una regla y otra estriba en el paso de estos argumentos a la lista de subcategorización COMPS. Cada una de estas reglas léxicas da lugar a un patrón de realización del verbo. La aplicación de las reglas sintácticas canónicas sobre cada uno de ellos dará lugar a cada orden posible de complementos.

$$\left[\begin{array}{l} \text{dat-ac-ditr-lex-rule} \\ \text{SYNSEM|LOCAL|VAL|COMPS} \langle \boxed{2}, \boxed{1} \rangle \\ \text{ARG-ST} \quad \langle \text{synsem}, \boxed{1}[\text{CASE } \textit{acc}], \boxed{2}[\text{CASE } \textit{dat}] \rangle \end{array} \right]$$

Figura 5.17: *Scrambling* con reglas léxicas (II)

Este sistema es equivalente al anterior. Implica la misma ambigüedad indeseable. Además, estipula cada orden posible en una regla léxica independiente, lo cual sigue sin ser una solución elegante.

5.2.2. Constituyentes discontinuos

Las versiones clásicas de HPSG no contemplan fenómenos de *scrambling*. Sin embargo, HPSG cuenta con una Teoría de la Linealización. Su formulación original se debe a Reape (1992). La han reformulado después Pollard y otros (1993) y Kathol y Pollard (1995). Se ha utilizado para gramáticas computacionales en el proyecto Babel —Müller (1996)—. Esta propuesta sí da cuenta de los ejemplos de (5) de forma elegante. Se repiten estos en (6).

- (6) a. El botones le ha subido el equipaje al guía
 b. El botones le ha subido al guía el equipaje

Todas estas versiones usan el concepto crucial de constituyente discontinuo. Imagínese que los elementos de la lista COMPS aparecen siempre en un mismo orden. Por ejemplo, imagínese que el complemento acusativo precede siempre en COMPS al dativo. Además, las reglas sintácticas saturan siempre al primer argumento de las listas. De este modo, podría darse cuenta de oraciones como (5-a), donde el acusativo precede al dativo. Pero, si se contem-

pla la existencia de constituyentes discontinuos, las oraciones donde —como en (6-b)— el dativo precede al acusativo también serían posibles. Haciendo uso de tal concepto, tanto en (6-a) como en (6-b), la primera proyección del sintagma verbal la formarían el núcleo desnudo *subido* y el complemento acusativo *el equipaje*. La única diferencia sería que, en (6-a), ese constituyente sería continuo (sus elementos se han linealizado juntos, de forma continua), mientras que en (6-b) el constituyente sería discontinuo (sus elementos se han linealizado separados, de forma discontinua, pues entre ellos aparece *al guía*). Una vez creada, en ambos casos, la primera proyección del sintagma verbal, esta puede unirse al dativo para crear la segunda.

Para implementar esta noción de constituyente discontinuo, es necesario que el *parser* que utilice la gramática contemple la posibilidad de que dos elementos que se unan en el análisis no tengan necesariamente que ser adyacentes. El *parser* en cuestión —siguiendo a Müller (2004)— debe permitir que dos elementos se unan en el análisis siempre que no se superpongan. La cadena de entrada de todo análisis sintáctico está dividida en n unidades atómicas: *tokens*. A cada una de ellas se le asigna una posición. Por ejemplo, si la cadena de entrada es la oración de (6-b), a cada *token* le corresponderá una posición de 0 a 8, según se muestra en (7).

(7) [0El] [1 botones] [2 le ha] [3 subido] [4 a] [5 el] [6 guía] [7 el] [8 equipaje]

En un *parser* tradicional se establece que, para que dos elementos se unan, deben ocupar posiciones contiguas: uno debe terminar en la posición i , y otro debe comenzar en $i + 1$. En cambio, en un *parser* que admita constituyentes discontinuos, para que dos elementos se unan, basta con que ninguna de las posiciones de la cadena que ocupe uno las ocupe el otro. En (7), el verbo *subido* ocupa la posición 3 de la cadena, y el complemento acusativo *el equipaje* ocupa las posiciones 7 y 8. Un *parser* tradicional no admitiría la unión de estos dos elementos, pues sus posiciones no son contiguas. Pero uno que sí contemple constituyentes discontinuos sí permite la unión de *subido* y

el equipaje, pues las posiciones de estos no se superponen. Este último *parser*, una vez creado el constituyente discontinuo *subido ... el equipaje*, lo puede unir al complemento dativo *al guía*, pues de nuevo las posiciones de *subido ... el equipaje* (3, 7 y 8) y las de *al guía* (4, 5 y 6) no se superponen. Con todo ello, se crea el constituyente completo *subido al guía el equipaje*, que ocupa las posiciones 3, 4, 5, 6, 7 y 8.

En SSG se concibe la lista COMPS como una lista cuyos elementos, aun siempre en posición posnuclear, permiten cualquier ordenación relativa. El español sigue entendiéndose como una lengua configuracional, en la que cualquier constituyente debe tener una linealización continua: solo las proyecciones intermedias del sintagma verbal pueden ser discontinuas.

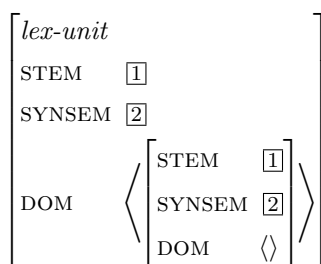


Figura 5.18: Unidad léxica con DOM

Pero, ¿es este concepto de constituyente discontinuo aplicable a una lengua configuracional como el español? Hasta ahora se ha entendido en la discusión que estas lenguas no permiten constituyentes discontinuos. Precisamente, el fenómeno de *scrambling* invita a matizar esta afirmación. La teoría de la linealización de HPSG permite establecer adecuadamente tales matices. De acuerdo con Donohue y Sag (2006), la Teoría de la Linealización de HPSG añade al signo lingüístico un rasgo llamado DOM (dominio). DOM es un conjunto de elementos¹⁰. Cada uno de estos elementos cuenta a su vez con otro dominio DOM anidado. Por ejemplo, la caracterización habitual de una unidad

¹⁰Cada una de las versiones de la Teoría de la Linealización —Reape (1992); Pollard y otros (1993); Kathol y Pollard (1995)— concibe estos elementos como tipos de una u otra clase. Son matices técnicos que no interesan a esta discusión.

léxica en la Teoría de la Linealización es la que se muestra en la figura 5.18. Como se ve, el elemento del dominio DOM de un signo es una réplica de buena parte de la información de ese signo: STEM y SYNSEM. A esa información se añade un nuevo dominio DOM anidado. En el caso de las unidades léxicas, ese dominio anidado está vacío.

En cambio, en el caso de los sintagmas de las lenguas configuracionales, el dominio DOM es un reflejo de las relaciones jerárquicas establecidas entre el sintagma y sus constituyentes. Cada sintagma cuenta con un único elemento DOM. Pero, a su vez, ese elemento cuenta en su rasgo DOM anidado con los elementos correspondientes a sus constituyentes inmediatos. El rasgo DOM original es, por tanto, una estructura con anidación que refleja fielmente la estructura de constituyentes del sintagma. En concreto, en estas lenguas, los sintagmas se definen como instancias del tipo *compacting-phrase* que se muestra en la figura 5.19.

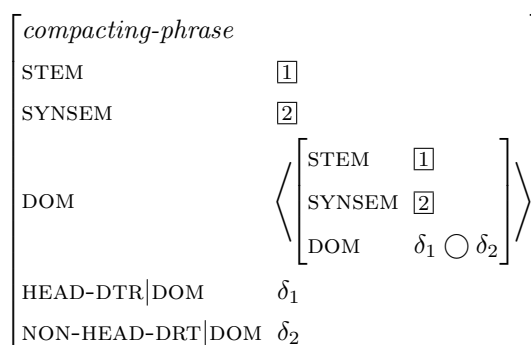


Figura 5.19: Sintagma continuo

Por ejemplo, imagínese una derivación en la que se crean sucesivamente dos proyecciones del núcleo X : primero se une X a su complemento, y después a su especificador —figura 5.20—. Imagínese que X^1 y X^2 se han formado como instancias de *compacting-phrase*. En tal caso, el rasgo DOM de cada uno de estos sintagmas cuenta con un único elemento: el que representa al sintagma en sí. Los elementos correspondientes a sus constituyentes aparecen anidados en el rasgo DOM de ese elemento. Por tanto, la derivación que da

lugar a X^2 crea en el rasgo DOM de este sintagma la estructura de dominios anidados que se observa en la figura 5.21.

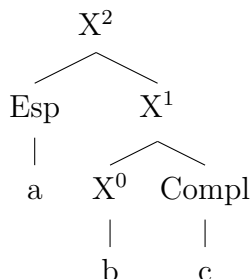


Figura 5.20: Derivación de X^2

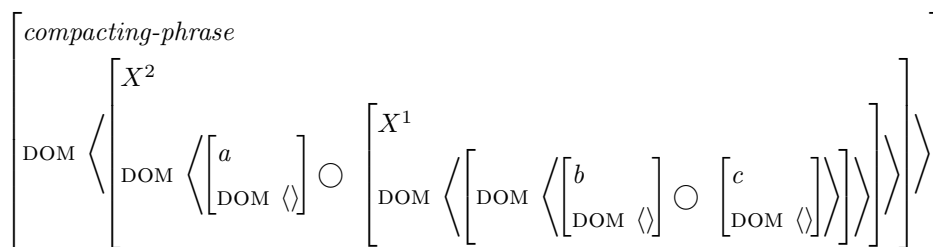


Figura 5.21: DOM en X^2 con compacting-phrase

Además, en la Teoría de la Linealización de HPSG, un rasgo DOM denota un dominio cuyos elementos deben linealizarse de forma continua. Como se ha visto, en un sintagma de tipo *compacting-phrase*, los constituyentes inmediatos cuentan con un dominio propio. Por tanto, en una lengua con este tipo de sintagmas, cada nodo sintagmático debe tener linealización continua. Esta es la característica definitoria de las lenguas configuracionales. Por tanto, en HPSG se formaliza el concepto de lengua configuracional, dada la noción de dominio y sus implicaciones, entendiendo que estas lenguas son aquellas cuyos sintagmas son instancias de *compacting-phrase*.

Por otro lado, los distintos elementos de un mismo dominio se unen mediante el operador \circ —*shuffle*—. Este operador no estipula, en principio, un orden relativo entre estos elementos. Pero a un dominio DOM pueden adscribirse ciertas restricciones de linealización. Por ejemplo, en una lengua de

núcleo inicial, puede establecerse una restricción que pida en los dominios apropiados anteposición del núcleo al complemento. Pueden crearse restricciones vinculadas a criterios discursivos, entonativos, informativos, etc.

¿Cómo se capta en esta teoría el concepto de lengua no configuracional? En las lenguas no configuracionales, los sintagmas son de tipo *liberating-phrase* —figura 5.22—. Como se puede observar, en estos sintagmas, el dominio DOM no refleja la estructura de constituyentes. El rasgo DOM de estos sintagmas es una estructura plana, sin anidación, que recoge directamente la unión de los dominios de los constituyentes. Estos dominios, entonces, no reflejan la relación jerárquica existente entre sintagma y constituyentes.

$$\left[\begin{array}{ll} \textit{liberating-phrase} & \\ \text{DOM} & \delta_1 \circ \delta_2 \\ \text{HEAD-DTR|DOM} & \delta_1 \\ \text{NON-HEAD-DRT|DOM} & \delta_2 \end{array} \right]$$

Figura 5.22: Sintagma discontinuo

Imagínese que, dada la derivación vista en la figura 5.20, X^1 y X^2 son instancias, no de *compacting-phrase*, sino de *liberating-phrase*. En tal caso, el dominio de cada uno de estos sintagmas no crea una estructura anidada, sino que recoge en un único nivel los elementos DOM de sus constituyentes. El resultado es que, en X^2 , DOM cuenta con un elemento por cada uno de los constituyentes terminales de la derivación, y entre ellos no se establecen relaciones jerárquicas: véase en la figura 5.23.

$$\left[\begin{array}{l} \textit{liberating-phrase} \\ \text{DOM} \left\langle \left[\begin{array}{l} a \\ \text{DOM} \langle \rangle \end{array} \right] \circ \left[\begin{array}{l} b \\ \text{DOM} \langle \rangle \end{array} \right] \circ \left[\begin{array}{l} c \\ \text{DOM} \langle \rangle \end{array} \right] \right\rangle \end{array} \right]$$

Figura 5.23: DOM en X^2 con *liberating-phrase*

Dado el concepto de dominio —ámbito en el cual los constituyentes deben linealizarse de forma continua— una lengua cuyos sintagmas sean instan-

cias de *liberating-phrase* permite linealizaciones discontinuas. En este caso, la proyección X^1 no cuenta en X^2 con un dominio propio, sino que sus constituyentes — b y c — están en el mismo dominio de X^2 , junto a a . Esto permite que a se linealice entre b y c , lo cual hace de X^1 un sintagma discontinuo. En definitiva, en HPSG, una lengua no configuracional es aquella en la cual los sintagmas son de tipo *liberating-phrase*.

Solo cabe añadir la formalización del hecho de que la estructura de dominios DOM condiciona la linealización de los constituyentes terminales de un signo. Esta formalización se recogió en Reape (1992) en lo que se llamó *Constituent Ordering Principle*. Este principio establece que existe una correspondencia directa entre el significante de un signo —STEM en SSG— y el significante de los elementos de su dominio: véase en la figura 5.24.

$$\left[\begin{array}{l} \textit{sign} \\ \text{STEM } \phi_1 \oplus \dots \oplus \phi_n \\ \text{DOM } \langle \left[\text{STEM } \phi_1 \right] \oplus \dots \oplus \left[\text{STEM } \phi_n \right] \rangle \end{array} \right]$$

Figura 5.24: *Constituent Ordering Principle*

Cabe esperar que cualquier lengua no configuracional cuente con ciertos contextos en los que la linealización sí debe respetar la estructura sintáctica —véase el estudio del warlpiri de Donohue y Sag (2006)—. Es decir, para entender una lengua como no configuracional basta con que esta haga un uso amplio del patrón *liberating-phrase*. Del mismo modo, el español es claramente una lengua configuracional, dado que, habitualmente, utiliza el patrón *compacting-phrase*. Pero esto no impide que existan además ciertos contextos que admitan la realización discontinua. Por tanto, el fenómeno de *scrambling* que ocupa a este apartado es compatible con su carácter configuracional: en español, lengua configuracional, las proyecciones internas al sintagma verbal no forman un dominio DOM propio. En cambio, la proyección máxima de este sintagma verbal sí tiene un dominio propio.

Aunque SSG asume, desde un punto de vista teórico, esta Teoría de la Linealización, no incluye en su signo lingüístico el rasgo DOM. Este crea una redundancia muy grande en la estructura de rasgos, lo cual acarrea coste de procesamiento. Para dar cuenta de los efectos del *scrambling* basta con concebir, como ya se ha apuntado, la lista COMPS como lista desde la cual el *parser* puede crear constituyentes discontinuos: es posible saturar el primer elemento de la lista COMPS de un núcleo, aunque núcleo y complemento no tengan linealización contigua en la secuencia de entrada.

5.2.3. Problemas

Hasta aquí se han tratado las posibilidades de ordenación de los complementos de acusativo y dativo en posición posverbal. Estos se saturan en tal posición desde la lista COMPS. Por tanto, se ha entendido que esta lista es el dominio dentro del cual es posible el *scrambling* en español. SSG cuenta con que los elementos de COMPS puedan ser saturados en el orden en que están listados, pero quizá por sintagmas no adyacentes al verbo en la secuencia de entrada.

¿Es esto válido para el resto de elementos de COMPS? Como se dijo en 5.1, en SSG el sujeto se recoge también en dicha lista. Por tanto, si el *scrambling* está vinculado a COMPS, el sujeto debe poder realizarse, tras el verbo, en cualquier orden relativo a los objetos. Los siguientes ejemplos, tomados de Fernández Soriano (1993), muestran que, efectivamente, el sujeto *Juan* de (8-b) y (8-c), en posición posverbal, puede ir antepuesto o pospuesto al objeto *el periódico*.

- (8) a. Juan ha comprado el periódico
 b. Ha comprado Juan el periódico
 c. Ha comprado el periódico Juan

Pero, como señala esta autora, la elección de una u otra ordenación acarrea un cierto valor informativo diferente. La ordenación canónica es la de (8-a). Las de (8-b) y (8-c) son ordenaciones derivadas, en las que el sujeto busca

un valor informativo que no tiene en su posición canónica. Los objetos, en cambio, podían variar su posición sin que aquello acarree diferencias interpretativas a la oración. Por tanto, no parece que el principio que regula la libertad de aparición del sujeto posverbal sea el mismo que regulaba la libertad de ordenación de los objetos.

Otro tanto ocurre con las posibilidades de ordenación de otro tipo de complementos de COMPS: los oblicuos y los predicativos. Los siguientes ejemplos de (9), y los juicios de gramaticalidad, son de Fernández Soriano (1993).

- (9) a. Pon el abrigo encima de la silla.
 b. ?Pon encima de la silla el abrigo.
 c. No se debe confundir la creatividad con la perseverancia.
 d. ?*No se debe confundir con la perseverancia la creatividad.

En este trabajo se defiende que, con el debido valor informativo, (9-b) y (9-d) sí son gramaticales. Estos complementos también pueden, por tanto, ordenarse de múltiples modos en posición posverbal; pero, como ocurría con el sujeto, la alteración del orden canónico asigna un valor informativo característico a la oración.

¿En qué consiste este valor informativo asociado a las distintas posibilidades de ordenación posverbal del sujeto y los complementos oblicuos y predicativos? La estructura informativa de una oración enunciativa descansa en el uso de dos elementos: el tema y el rema¹¹. El tema es la parte de la oración que denota una información conocida en el discurso. En cambio, el rema es la parte de la oración que denota una información nueva en el discurso. La idea es que un discurso avanza apoyándose en información conocida (tema) a la que se va añadiendo información nueva (rema). Parece que en español —Contreras (1978), Zubizarreta (1999)— en una oración enunciativa con entonación normal la última sílaba tónica recibe una cierta marca prosódica. En una oración de estas características, se pueden interpretar como

¹¹Zubizarreta (1999).

rema todos aquellos sintagmas que incluyan dicha marca. Por ejemplo, en la oración anterior de (8-a), que se muestra en la figura 5.25, la marca prosódica recae sobre la palabra *periódico*. Así pues, en esta oración el rema puede ser cualquier proyección que incluya *periódico* directa o indirectamente: SN_2 , SV , SF y O . Es decir, el rema pueden ser todos aquellos sintagmas de la oración que incluyen la palabra marcada. En cambio, no pueden ser rema SN_1 , F o V , pues ninguno de estos sintagmas incluyen la marca prosódica de *periódico*.

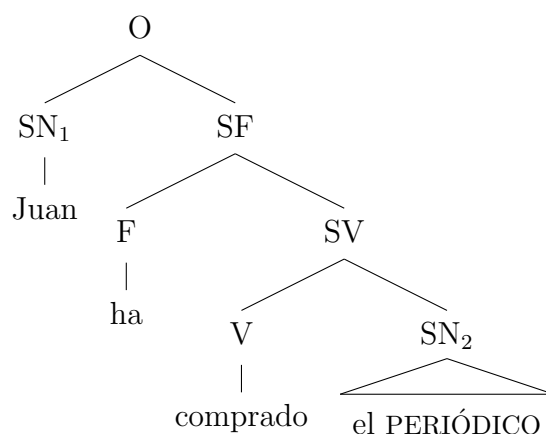


Figura 5.25: Juan ha comprado el periódico

Por tanto, esta oración puede aparecer como respuesta a situaciones en las cuales se sepa que Juan ha comprado algo, pero no se sepa qué; o se sepa que Juan ha hecho algo, pero no se sepa qué; o no se sepa nada, y toda la oración sea información nueva. Pero no es adecuada esta oración en un contexto en el que se sepa que alguien ha comprado el periódico, y se pregunte por el agente de la acción. Esto se debe a que, en tal caso, el rema de la oración debería ser exclusivamente el sujeto. Y, dado que este no incluye la marca prosódica, el sujeto no puede ser rema de la oración.

Se ha dicho que esta marca prosódica recae sobre la última sílaba tónica de la oración. Por tanto, para permitir que el sujeto en cuestión incluya la marca que lo habilitaría como rema de la oración, es necesario un cambio en la ordenación de los constituyentes. La oración de (8-c) es el resultado

de tal cambio. En ella, *Juan*, al aparecer en posición final de la oración, es la palabra que recibe la marca prosódica. Y gracias a ello, es el rema de la oración: (8-c) solo es adecuada en un contexto en el que, efectivamente, se sepa que alguien compró el periódico, pero no se sepa quién lo hizo. En el modelo transformacional que utilizan Contreras (1978) y Zubizarreta (1999), este cambio de orden obedece a un proceso de movimiento de constituyentes. Valga la versión de este de la figura 5.26. Dado este movimiento, el único sintagma que cuenta con la marca prosódica pertinente es SN_1 . Por ello, cabe predecir que (8-c) no admite otra interpretación que aquella en la que el rema es solo el sujeto *Juan*; lo cual parece cierto.

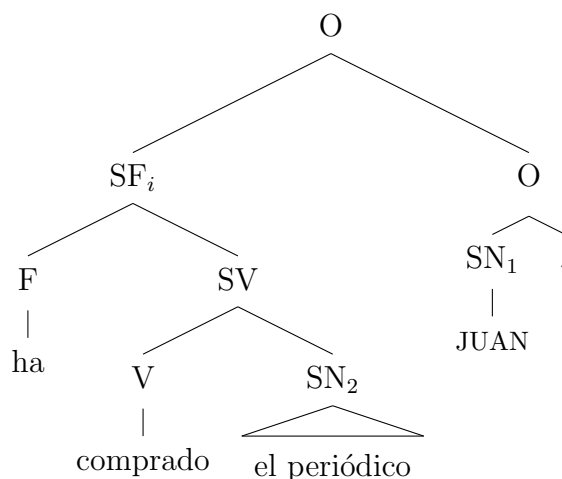


Figura 5.26: Ha comprado el periódico Juan

No se pretende explicar de modo exhaustivo el análisis que estos trabajos harían de las oraciones de (8) y (9). Pero queda claro, gracias a la explicación precedente, que el hecho de que los sujetos puedan obedecer a ordenaciones lineales diferentes se ha entendido como el resultado de la existencia de diferentes configuraciones estructurales adscritas a distintos valores informativos. Cabe pensar que otro tanto ocurre con los complementos de caso oblicuo y los predicativos.

En resumen, parece que todos los argumentos verbales contemplados en

SSG (incluido el sujeto) pueden aparecer, en posición posverbal, en cualquier orden relativo. En general, esos órdenes relativos distintos requieren patrones entonativos distintos y aportan diferencias interpretativas a la oración. Solo los complementos de caso acusativo y dativo pueden permutar sus posiciones sin que tal cambio implique diferencias interpretativas.

Como se dijo en 5.2.2, la Teoría de la Linealización de HPSG puede modelar todas estas posibilidades de ordenación lineal haciendo uso del concepto de constituyente discontinuo. Efectivamente, SSG entenderá que cualquier orden relativo entre los elementos de COMPS es posible gracias a dicho concepto. De acuerdo con la Teoría de la Linealización, todos estos elementos deben listarse en un único dominio DOM. Como se dijo en 5.2.2, a falta de otra restricción, esto implica que todos ellos se linealizan de forma continua, pero en cualquier orden relativo. El hecho de que algunos de estos órdenes esté adscrito a un cierto patrón entonativo e informativo se ha de modelar como restricciones extra sobre la lista DOM en cuestión. No obstante, quedan pendientes de estudiar, explicitar y formalizar cuáles son las restricciones informativas y entonativas que deben vincularse a este dominio.

5.3. Dependencias no acotadas

Se termina el modelado del orden de palabras del español con un apartado dedicado a las dependencias no acotadas. En buena medida, este tipo de fenómenos ya ha sido explicado en apartados previos del trabajo. En 1.1.4.3 se vio que estos son fenómenos no locales que ponen a prueba la adecuación descriptiva y explicativa de las gramáticas. Son fenómenos que, en principio, parecen pedir un tratamiento transformacional. Sin embargo, ya en tal apartado, y con más detalle en 4.1, se explicó que las gramáticas de unificación podían dar cuenta de ellos de un modo satisfactorio.

En concreto, en este apartado se explican tres tipos de desplazamientos en dominio no acotado: desplazamiento de tópicos, de partículas interrogati-

vas y de relativos. Ya desde Gazdar y otros (1985) para GPSG, se ha utilizado el rasgo SLASH —4.1— para dar cuenta de estos desplazamientos. Después, las versiones clásicas de HPSG, así como las gramáticas computacionales —Matrix—, han utilizado sistemas muy similares. SSG, en esencia, utiliza el mismo sistema. Pero los datos del español hacen necesaria cierta complicación de su versión clásica, lo cual acarrea un coste computacional añadido. SSG no cuenta con un análisis de estos fenómenos de desplazamiento capaz de neutralizar dicha complejidad. Solo se aventura, como posible solución, un análisis basado en Balari (1998), cuya aplicación a SSG haría necesario el desarrollo real de un sistema de ligamiento. Como se explicó en 3.1.2, la teoría del ligamiento de HPSG no está lo suficientemente formalizada como para implementarla directamente en una gramática computacional.

En cualquier caso, este apartado tiene como objetivo justificar, en virtud de los datos que se presentan a continuación, el uso de las huellas *topic-trace* y *focus-trace* propuesto en 4.4.2. El que la gramática contemplase dos tipos de huellas diferentes era crucial para el análisis de los clíticos visto en tal apartado. En este apartado se demuestra cómo esa dualidad está justificada por razones independientes: cada tipo de huella desencadena procesos no locales diferentes, imposibles de modelar con el mismo sistema de desplazamiento.

En español son posibles distintos tipos de dependencias no acotadas. Un argumento puede desplazarse a la margen izquierda de la oración para colocarse en una posición donde puede encontrar dos tipos de valor informativo: puede ser tópico —se presenta como tema del que se va a predicar— o foco —se presenta como información que se está corrigiendo o revisando en algún sentido—. *Esos deportes* es ejemplo de tópico en (12-a) y es foco en (10-b).

- (10) a. Esos deportes los puede practicar el turista en la playa
 b. !ESOS DEPORTES puede practicar el turista en la playa, no los otros!

Tópico y foco presentan comportamientos diferentes. Como se vio en 4.2, la topicalización de un argumento cliticizable pide aparición de clítico, mientras que su focalización no. Además, la focalización pide inversión del sujeto y la topicalización no, según se observa en (11).

- (11) a. Esos deportes el turista los puede practicar en la playa
 b. !ESOS DEPORTES *los puede practicar el turista en la playa, no los otros!
 c. *!ESOS DEPORTES el turista puede practicar en la playa, no los otros!

Por último, una oración puede contar con varios tópicos —(12-a)—, mientras que solo es posible un foco —(12-b)—. Sí es posible que convivan tópicos y foco, en cuyo caso los primeros han de preceder linealmente al segundo —(12-c) y (12-d)—.

- (12) a. El turista, esos deportes los puede practicar en la playa
 b. *!EL TURISTA, ESOS DEPORTES puede practicar en la playa, no el guía los otros deportes!
 c. El turista, ¡ESOS DEPORTES puede practicar en la playa, no los otros deportes!
 d. *!EL TURISTA, esos deportes los puede practicar en la playa, no el guía!

Se observa —Zubizarreta (1999)— que el desplazamiento de partículas interrogativas (y en menor medida el de relativos) presenta las mismas características que el desplazamiento de foco. Véase en (13) cómo el movimiento de una partícula interrogativa no admite cliticización y pide inversión de sujeto. Además, no es posible desplazar dos elementos de este tipo, y estos deben linealizarse tras los tópicos, en caso de que estos existan.

- (13) a. ¿Qué deportes *los puede practicar el turista en la playa?
 b. *¿Qué deportes el turista puede practicar en la playa?
 c. *¿Quién, qué deportes puede practicar en la playa?

- d. El turista, ¿qué deportes puede practicar en la playa?
- e. *¿Quién, esos deportes los puede practicar en la playa?

En (14) se observa también que el movimiento de un pronombre relativo no admite cliticización, si bien no pide inversión de sujeto. No se puede observar el orden entre relativos y tópicos, pues estos últimos en estas oraciones no tienen lugar al que desplazarse.

- (14) a. Los deportes que *los puede practicar el turista en la playa
 b. Los deportes que el turista puede practicar en la playa

Por último, téngase en cuenta que es posible extraer elementos mediante topicalización de lugares desde los cuales no se puede extraer un pronombre interrogativo o un relativo: el interior de un sintagma nominal, el interior de un sujeto preverbal o el interior de un adjunto. En (15) se observa cómo no es posible extraer tales elementos desde dentro del sintagma nominal *postales de Madrid*, ni desde el adjunto *cuando abran la tienda*.

- (15) a. Podemos comprar postales de Madrid cuando abran la tienda
 b. *De qué_i podemos comprar [postales t_i] cuando abran la tienda
 c. *La ciudad de la que_i podemos comprar [postales t_i] cuando abran la tienda
 d. *Qué_i podemos comprar postales de Madrid [cuando abran t_i]
 e. *La tienda en la que_i podemos comprar postales de Madrid [cuando abran t_i]

En cambio, sí es posible la topicalización desde el interior de un sintagma nominal: (16).

- (16) De Madrid_i podemos comprar [postales t_i] cuando abran la tienda

Los mismos contrastes se observan cuando se intenta extraer un elemento desde dentro de un sujeto preverbal como *que visitemos el museo* en (17).

- (17) a. Que visitemos el museo es interesante

- b. *¿Qué_i [que visitemos t_i] es interesante?
- c. *El museo que_i [que visitemos t_i] es interesante
- d. El museo_i [que lo visitemos t_i] es interesante

Por tanto, parece que la focalización, el desplazamiento de partículas interrogativas y el de relativos funcionan del mismo modo: obedecen a un mismo sistema de movimiento. De hecho, foco, partícula interrogativa y relativo son incompatibles entre sí. Parecen, entonces, huéspedes alternativos de la misma posición estructural: (18).

- (18)
- a. *¿Quién, ESOS DEPORTES puede practicar en la playa?
 - b. *ESOS DEPORTES, ¿quién puede practicar en la playa?
 - c. *El turista que ¿qué deportes puede practicar en la playa?
 - d. *El turista que ESOS DEPORTES puede practicar en la playa

En cambio, la topicalización, con su comportamiento diferente, obedece a un segundo sistema de movimiento. En cualquier caso, es imposible formalizar en HPSG estos dos tipos de comportamiento con un mismo sistema. La intuición que debe formalizarse es que la posición estructural que ocupan las partículas interrogativas, los relativos y los focos es inmediatamente superior a la oración, y no es iterable; mientras que la posición de los tópicos es superior a esta, y sí es iterable. La lista SLASH formaliza la existencia de una posición de realización de los argumentos externa a la oración. Entiéndase que esta es el lugar de realización de los casos paradigmáticos de desplazamiento a la izquierda: las partículas interrogativas y los relativos. Parece natural contar, además, con una nueva lista TOPIC para la realización de los tópicos. Los tópicos dejarían una huella de tópico, y se desplazarían y realizarían en TOPIC. En cambio, las partículas interrogativas y los relativos dejarían una huella de foco, y se desplazarían y realizarían en SLASH. La lista SLASH solo admitiría un elemento, mientras que TOPIC admitiría varios. Además, SLASH se saturaría antes que TOPIC. Por último, las condiciones de ascenso de información de proyección a proyección serían diferentes entre SLASH y TOPIC, con lo cual se daría cuenta de las distintas condiciones de extracción de estos dos fenómenos. Parece que esta es la solución natural en HPSG a los datos

antes presentados.

Este análisis justifica la existencia de dos tipos de huella distintos, como ya se dijo en 4.4.2: *focus-trace* y *topic-trace*. Se pueden observar sus representaciones en las figuras 5.27 y 5.28. El tipo *focus-trace* representa a las huellas que desencadenan los movimientos de partículas interrogativas y relativos.

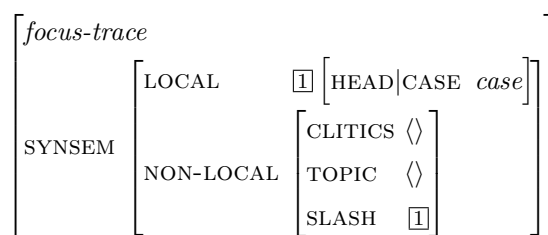


Figura 5.27: *focus-trace*

Por su parte, *topic-trace* es la huella que desencadena las topicalizaciones.

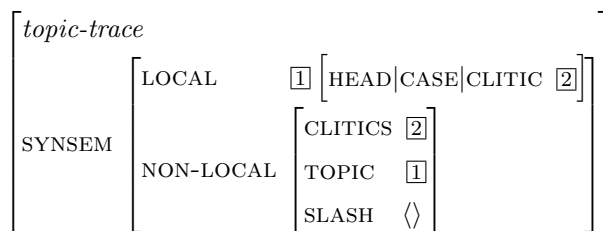


Figura 5.28: *topic-trace*

Ambos son instancias posibles de la posición de complemento de la regla *comps-rule*. Por tanto, ambos saturan una posición argumental de la lista *COMPS* de un verbo. Pero *focus-trace* guarda su información local en su lista no local *SLASH*, mientras que *topic-trace* lo hace en *TOPIC*. En términos locales ambas desempeñan la misma función; pero desencadenan procesos no locales, como se ha visto, diferentes. Como se ve en la figura 5.28, *topic-trace* es una unidad léxica capaz de aportar clítico al verbo al que se une como argumento. La discusión precedente demuestra que su inclusión en la gramática, necesaria para el sistema de clíticos, está justificada por razones independientes

a este.

Queda explicar cómo se saturan en SSG estos elementos desplazados de uno y otro tipo. La lista SLASH se satura mediante la regla *slash-rule*, que se puede observar en la figura 5.29; y TOPIC se satura mediante *topic-rule*: figura 5.30. La superioridad estructural de los tópicos se regula estableciendo que *topic-rule* solo se aplica sobre núcleos con lista SLASH saturada —null—. En cambio, *slash-rule* no altera la lista TOPIC del signo sobre el que se aplica. Además, la iteración de un foco es imposible, pues *slash-rule* satura una lista SLASH de un único elemento.

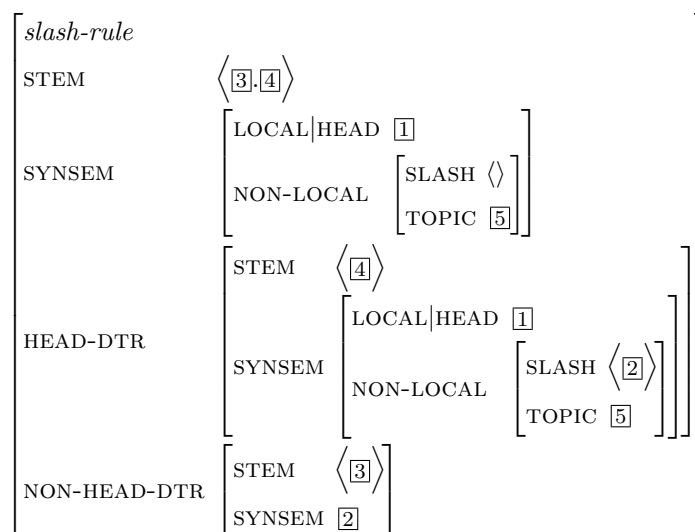


Figura 5.29: Regla de saturación de SLASH

Ha de apuntarse que los tópicos de una oración pueden aparecer en cualquier orden relativo: véanse las oraciones de (19). Por tanto, tal como pasara con COMPS, TOPIC debe denotar un dominio en el cual los elementos pueden realizarse en cualquier orden relativo. El resultado es una gramática conceptualmente sencilla: toda posposición de un argumento se debe a su realización explícita desde COMPS; y casi toda anteposición se debe a un proceso de topicalización, a una realización desde TOPIC. Ambas listas, en principio, admiten saturación a partir de elementos no adyacentes al núcleo.

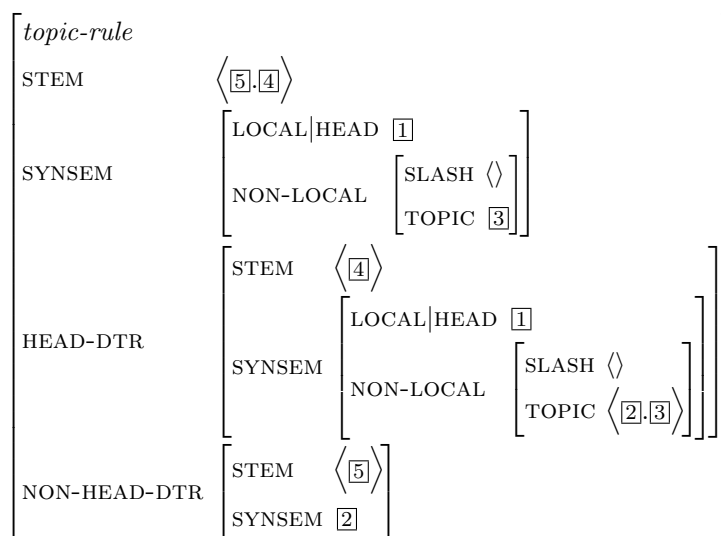


Figura 5.30: Regla de saturación de TOPIC

- (19) a. El turista, esos deportes los puede practicar en la playa
 b. Esos deportes, el turista los puede practicar en la playa

No obstante, la distinción entre SLASH y TOPIC tiene un coste computacional. Para cada argumento desplazado, un *parser* tendría que contemplar a ciegas dos posibilidades: que este dejase huella en SLASH o en TOPIC. Esto generaría dos derivaciones paralelas hasta la aparición del *filler*. Pero es un coste inevitable para toda gramática que quiera dar cuenta de los datos del español. Una posible solución, inspirada en Balari (1998)¹², consistiría en entender

¹²Balari (1998) observa que existen dos grandes grupos de dependencias no acotadas (*Unbounded Dependency Constructions* o *UDCs*). Por un lado, están lo que llama dependencias no acotadas fuertes (*Strong UDCs*). Son los casos prototípicos de movimiento no acotado: el de partículas interrogativas, etc. Por otro, están las dependencias no acotadas débiles (*Weak UDCs*). Estas son la topicalización con cliticización de lenguas como el español, o el italiano (*Clitic Left Dislocation*), los huecos parásitos (*parasitic gaps*) y las construcciones de omisión de objeto (*missing object construction*). *Strong UDCs* y *Weak UDCs* obedecen a comportamientos diferentes. Por ejemplo, las primeras están sujetas a fuertes restricciones a la extracción, que no existen para las segundas. Balari (1998) analiza las primeras como casos reales de movimiento —con una lista SLASH—. En cambio, las segundas las entiende como relaciones de ligamiento. Para él, el elemento de base de esa

que la topicalización no es propiamente un fenómeno de movimiento. Este autor propone que la relación entre el tópic y lo que aquí se ha formalizado con **topic-trace** es, en cambio, una relación de ligamiento. Los tópicos, en su análisis, se unirían a la estructura en virtud de un esquema (llámese **topic-rule**) que no satura ninguna lista no local. En principio, un tópic podría unirse a la estructura libremente, si bien legitimado de algún modo por la aparición en la derivación de un pronombre del cual es antecedente.

SSG podría entonces prescindir de la lista **TOPIC** y la huella **topic-trace** y entender que en la posición de base de la topicalización se coloca un pronombre sin realización fonética. Este tipo de pronombres —como se vio en 4.4.2, formalizados entonces como instancias del tipo **personal-pronoun**— aportan clítico al verbo, igual que lo hacía **topic-trace**. Por tanto, el análisis de los clíticos de SSG sería también válido en esta nueva versión de la gramática; y aun simplificado, pues huellas de topicalización y pronombres omitidos se neutralizarían en el tipo **personal-pronoun**. Pero para la implementación real de esta solución sería necesario formalizar el concepto de ligamiento entre el tópic y el pronombre de base.

5.4. Conclusión

Este capítulo 5 recoge tres cuestiones centrales relativas al orden de palabras del español: la inversión del sujeto, las posibilidades de ordenación relativa entre argumentos posverbiales y los diferentes tipos de dependencias no acotadas. Las soluciones habituales en HPSG para ellas son problemáticas desde un punto de vista de coste computacional. En SSG se ha optado por soluciones que compaginen la motivación lingüística con la economía computacional.

Los datos de inversión del sujeto los cubre SSG entendiendo que la inver-relación de ligamiento puede ser una variable o un pronombre, cada uno de ellos con o sin realización fonética.

sión (la realización pospuesta al verbo) es la opción por defecto: un sujeto, a falta de otro proceso, se realiza en posición posverbal, desde COMPS, como los complementos habituales. La anteposición del sujeto es un proceso de topicalización, como el de cualquier objeto. Falta una justificación teórica rigurosa al respecto, pero se ha mostrado cómo la propuesta es plausible. En cualquier caso, dicha propuesta simplifica enormemente la gramática: en SSG todo argumento posnuclear se realiza desde COMPS y casi todo argumento prenuclear obedece a un proceso de topicalización, desde la lista TOPIC. Además, haciendo uso de la Teoría de la Linealización de HPSG, se ha entendido que tanto la lista de realización pospuesta (COMPS) como la de realización antepuesta (TOPIC) denotan dominios en los cuales los distintos elementos admiten cualquier ordenación relativa. Falta establecer las restricciones informativas y entonativas adscritas a las distintas posibilidades de realización.

Por último, se ha mostrado cómo los datos del español hacen pensar que existen dos tipos distintos de dependencias no acotadas: de entre las contempladas en SSG, por un lado están los movimientos de partículas interrogativas y relativos y por otro, las topicalizaciones. Estos procesos obedecen a comportamientos diferentes, y, por tanto, han de ser modelados como desplazamientos a través de listas distintas. Este modelado justifica la existencia en SSG de distintas huellas. En concreto, justifica la existencia de una huella específica para los procesos de topicalización. La existencia de esta unidad léxica es crucial para el modelado de clíticos visto en el capítulo 4, pues esta es la que aporta al verbo el clítico propio de una topicalización.

Capítulo 6

Conclusión

Este trabajo ha documentado la creación de la gramática SSG: una gramática computacional del español de reducido coste computacional. El modelo gramatical en el que está fundamentada es HPSG, una gramática de unificación. Estas gramáticas, a diferencia del modelo transformacional chomskyano, modelan directamente la estructura superficial de las secuencias. En HPSG, todo objeto de análisis se concibe como un signo lingüístico, y se formaliza como una estructura de rasgos. Es un modelo gramatical radicalmente lexicista: en él, son las unidades léxicas las que cuentan con el grueso de la información, frente a las reglas sintácticas, que son pocas y muy abstractas. En HPSG es necesaria, por tanto, una gestión eficiente de la información léxica. Por ello, estas gramáticas se conciben como una única jerarquía de tipos, donde cada tipo se define como subtipo de uno o varios supertipos, de los cuales hereda la información. Además, una gramática de tipo HPSG cuenta con una serie de reglas léxicas que describen los distintos patrones de comportamiento aplicables a cada palabra.

HPSG nació, frente al modelo chomskyano, como teoría plausible sobre el lenguaje humano, entendido este como problema de procesamiento. De hecho, es el modelo gramatical más aplicado actualmente en Lingüística Computacional. Por ello se ha utilizado para el desarrollo de SSG. Sin embargo, en la práctica, sufre problemas de índole computacional. Como se ha visto en este

trabajo, las posibilidades de realización de cada palabra son, en la práctica, muchas. Dadas las reglas léxicas de una gramática de tipo HPSG, un *parser* calcula todas las hipotéticas posibilidades de realización de cada palabra de la secuencia que analiza. Esas hipótesis desencadenarán después derivaciones parciales, la mayoría de ellas inútiles, a la postre, para la secuencia analizada. La combinación de todas estas posibilidades genera una explosión combinatoria que da lugar a tiempos de análisis muy altos.

SSG es fruto de la reflexión sobre estos problemas computacionales de las gramáticas de tipo HPSG, aplicadas a los fenómenos propios del español. En español, los verbos admiten muchos patrones de realización: transitivo, intransitivo, pasivo, de pasiva refleja, pronominal, con objeto oblicuo, con un complemento predicativo, con un dativo, etc. Los sintagmas que cubren las posiciones argumentales de estos patrones verbales pueden ser de distintos tipos: clíticos, sintagmas nominales, complementantes, huellas que desencadenan desplazamientos, etc. Además, tales argumentos obedecen a distintas posibilidades de ordenación lineal. La combinación de todo ello hace que un verbo normal del español pueda admitir un número realmente grande de posibilidades de realización. Una gramática de tipo HPSG clásica representa cada una de estas posibilidades mediante una regla, normalmente léxica. Es la combinación de estas reglas la que genera los problemas computacionales de los que se ha hablado. Por tanto, es deseable que una gramática de tipo HPSG del español reduzca, en la medida de lo posible, el número de reglas necesarias para dar cuenta de estos datos. SSG quiere ser esa gramática. Cuantas menos sean las reglas, menor será la explosión combinatoria, y mejores los tiempos de análisis.

Así pues, conforme a lo visto en el capítulo 3, SSG da cuenta de las alternancias de diátesis básicas del español: usos intransitivos, transitivos, pasivos, de pasiva refleja, pronominales, con objeto oblicuo, con complemento predicativo, y con dativo. Lo hace con un sistema de nueve reglas léxicas que se han dado en llamar no destructivas. En las versiones clásicas de HPSG, las reglas léxicas podían alterar las unidades sobre las que operaban prácticamente

de cualquier modo: añadían, eliminaban, modificaban argumentos, permutaban sus posiciones, etc. Las reglas léxicas no destructivas de SSG, en cambio, solo efectúan una operación: especifican la estructura de las unidades léxicas con información directamente compatible con ella. Dado este tipo de reglas, se entiende en SSG que no existen patrones de realización verbal canónicos y derivados, sino que todos ellos conviven en pie de igualdad como alternativas. Este estilo de implementación, novedoso en HPSG, es sencillo conceptualmente, y simplifica y facilita el mantenimiento y ampliación de la gramática.

El número de reglas léxicas de SSG es tan reducido gracias a los siguientes avances técnicos y teóricos. En primer lugar, el uso de la lista STOPPER ha eliminado la multiplicación de reglas de añadido de argumentos habituales en las gramáticas clásicas. En SSG, haciendo uso de STOPPER, una única regla puede añadir un argumento dativo, predicativo, etc. a un verbo, con independencia de que dicho verbo contara originariamente con uno u otro número de argumentos. Además, en SSG tampoco se multiplican los patrones antes presentados en función de la naturaleza categorial de sus argumentos. Por ejemplo, los patrones verbales pasivos, el transitivo y el inergativo admiten tanto argumentos internos nominales como verbales. SSG no ha duplicado estos patrones para dar cuenta, mediante reglas diferentes, de uno u otro tipo de argumento interno. La naturaleza categorial de los distintos tipos de complementos subcategorizados por los verbos se restringe mediante el uso de tipos como **argument** (argumento), **external** (argumento externo), **inner** (argumento interno) o **predicative** (complemento predicativo). Cada posición argumental está caracterizada en su verbo mediante estos tipos. Esta caracterización, junto a la correspondiente especificación de caso, basta para restringir la naturaleza categorial posible en cada posición argumental. Efectivamente, el caso de una posición argumental restringe indirectamente las categorías posibles para esta. Los nombres, complementantes y huellas han sido especificados en el léxico con caso estructural (supertipo de nominativo y acusativo). Para que estas unidades léxicas puedan aparecer en posiciones argumentales marcadas con un caso diferente al que establece su caracterización léxica, deben haber sido tomadas como complemento de las pertinentes

preposiciones de marca de caso (de acusativo, dativo u oblicuo).

Cabe señalar, además, que SSG hace un uso crucial de la relación habitual en HPSG entre las listas ARG-ST, COMPS y SUBJ. Los patrones verbales inergativo (*unergative*), inacusativo (*unaccusative*), pasivo (*passive*) y pronominal (*pronominal*) se conciben en SSG como distintos tipos posibles de relación entre tales listas. ARG-ST se ha entendido como lugar en el que se listan los argumentos que, en principio, denota un verbo. Y COMPS y SUBJ son las listas en las cuales deben proyectarse los argumentos de ARG-ST para realizarse de uno u otro modo. Así pues, un patrón pasivo se ha formalizado como aquel que, en el paso de lista a lista, obvia el argumento externo de ARG-ST; y un patrón pronominal es aquel que, en el paso de lista a lista, ha perdido lo que en un antiguo patrón reflexivo era el argumento interno. Estos argumentos, contemplados originariamente por los verbos, pero eliminados por las construcciones pasivas o pronominal, son el lugar natural para la colocación de los clíticos no argumentales del español. Con todo ello, se consigue una formalización sencilla y coherente, fundamentada en unos pocos principios, de lo que otras gramáticas tratan como fenómenos independientes.

Por último, obsérvese que, en SSG, la caracterización léxica de un verbo es fundamentalmente semántica. Los verbos del español se clasifican en esta gramática conforme a la tipología clásica: pueden ser realizaciones (*accomplishment*), logros (*achievement*), actividades (*activity*) o estados (*state*). SSG ha enriquecido el signo lingüístico con información semántica relevante que regula la participación de las unidades léxicas en los patrones sintácticos antes citados. Este sistema deslinda en la gramática los distintos conceptos (sintácticos y semánticos, de uno u otro tipo) y los hace converger adecuadamente, a fin de predecir el comportamiento sintáctico de cada unidad léxica.

El capítulo 4 documenta el sistema de clíticos de SSG. Este sistema da cuenta de los datos del español relativos a tal fenómeno, sin necesidad de añadir a la gramática regla léxica o sintáctica alguna. Esto es novedoso en HPSG, y un avance en la búsqueda de reducción de la explosión combinatoria.

Para ello, SSG concibe los clíticos como afijos que aporta a un verbo una unidad léxica que se une a aquel como argumento. Hay ciertas unidades léxicas que tienen tal propiedad: los pronombres personales, explícitos y omitidos (*personal-pronoun*) y las huellas de topicalización (*topic-trace*). Ese aporte al verbo de un afijo se ha formalizado mediante el sistema habitual en HPSG para dar cuenta de fenómenos no locales (lo cual está justificado, dada la naturaleza no local del fenómeno del salto de clíticos). En SSG, tanto las unidades léxicas sin realización fonética, como aquellas que sí son explícitas, ocupan en las reglas sintácticas (*comps-rule*) la misma posición de complemento. Con ello se evita la necesidad de crear reglas sintácticas específicas que definan la proyección de núcleos a partir de la unión de estos a distintos tipos de unidades léxicas sin realización fonética. Esta simplificación es novedosa entre las gramáticas computacionales de tipo HPSG, y ayuda a que la implementación de SSG sea sencilla, fácil de mantener y de ampliar.

Por último, el capítulo 5 trata el orden de palabras del español. En SSG, fundamentalmente, existen dos listas desde las cuales se realizan los argumentos de un verbo: *COMPS* y *TOPIC*. La realización canónica de todo argumento verbal (incluido el sujeto) es posverbal, desde *COMPS*. Cualquier anteposición (también del sujeto) se debe a un fenómeno de desplazamiento a la izquierda; en principio, a una topicalización. Por tanto, normalmente, los argumentos antepuestos se realizan desde la lista no local *TOPIC*. Este sistema reduce el número de reglas de SSG, y evita ambigüedades indeseables entre los sujetos preverbales canónicos y los sujetos topicalizados. Además, evita la necesidad de dar un tratamiento *ad hoc* al fenómeno de inversión de sujeto. Además, en SSG, las listas *COMPS* y *TOPIC* se conciben como dominios en los cuales los argumentos pueden unirse al núcleo de forma discontinua. Este es el modo habitual de dar cuenta de fenómenos de *scrambling* en la Teoría de la Linealización de HPSG. Gracias a esta posibilidad, SSG da cuenta de que, en español, los complementos posverbales pueden aparecer, tras el verbo, pero en cualquier orden relativo. Del mismo modo, los tópicos del español pueden aparecer, ante el verbo, pero en cualquier orden relativo. Esta libertad relativa en el orden de palabras se ha logrado en SSG sin necesidad de añadido

alguno de regla: ha bastado con confiar en que el *parser* permitirá que los argumentos de las listas COMPS y TOPIC puedan unirse a su núcleo aun sin ser adyacentes a él. Queda pendiente la labor de adscribir buena parte de estas posibilidades de ordenación a determinados patrones entonativos e informativos.

El resultado de todo ello es una gramática conceptualmente sencilla, para la cual ha sido necesaria una implementación relativamente pequeña, fácil de manejar y de ampliar. Además, la reducción sensible del número de reglas es un avance hacia el logro de una gramática eficiente. SSG, en su actual estado de desarrollo, no pretende ser una gramática de cobertura amplia, sino la solución sencilla y eficiente a los problemas centrales a los que debe hacer frente el implementador al desarrollar una gramática del español. Dadas las soluciones centrales de SSG, la implementación de la microestructura de una gramática del español de cobertura amplia debe ser una tarea relativamente sencilla.

No obstante, este trabajo abandona a investigaciones futuras el tratamiento de ciertos problemas centrales del español a los que no se ha dado solución. En primer lugar, sería muy interesante la obtención de un sistema elegante y económico que modelase las relaciones de ligamiento. Si la gramática contase con tal sistema —siguiendo a Balari (1998)— sería posible entender como fenómenos de ligamiento buena parte de las relaciones que ahora mismo se tratan como relaciones de dependencia no acotadas (topicalizaciones). Esto reduciría sensiblemente la ambigüedad de SSG, pues pronombres y huellas de topicalización podrían neutralizarse en un único tipo de unidad léxica capaz de establecer relaciones de ligamiento (recuérdese que ambas aportan clítico al verbo). También sería interesante una teoría sobre el modelado en HPSG de lo que la gramática chomskyana trata asumiendo la existencia de categorías funcionales. Esta teoría debería ayudar a distinguir varios tipos de proyecciones máximas para una misma unidad léxica, distinción útil para regular las posibilidades de aparición de esta. También ayudaría a establecer adecuadamente relaciones de orden de palabras relativas a determinantes,

auxiliares y adverbios con ámbito sobre el sintagma verbal. Por último, falta una reflexión profunda sobre la modificación intersectica, orientada a la eliminación de las ambigüedades que esta genera.

Apéndice A

El *parser* SGP

SGP es un entorno de análisis automático de lenguas naturales conforme a gramáticas de tipo HPSG. Consta de un conjunto de librerías escritas en Perl¹ con orientación a objetos y con acceso a bases de datos alojadas en un servidor MySQL². Perl es un lenguaje de programación interpretado, sencillo y versátil. Por ello, es útil para el desarrollo de prototipos de *parser* destinados a probar distintas estrategias de procesamiento de las lenguas naturales. No es, sin embargo, un lenguaje de programación adecuado para el desarrollo de *parsers* eficientes en sí mismos. SGP debe entenderse, por tanto, como un entorno de *parsing* pensado para la investigación en Lingüística Computacional y no para el desarrollo de herramientas de procesamiento eficientes. De hecho, SGP se ha desarrollado como herramienta de apoyo a la presente investigación. La gramática SSG usa análisis tradicionales en HPSG que, por razones técnicas, no se han aplicado habitualmente a las gramáticas computacionales. SGP solventa esos problemas técnicos.

En la figura A.1 se muestra el funcionamiento básico de SGP. SGP toma como *input* un documento de texto plano. Haciendo uso de la librería `Tokenizer`, convierte este texto plano en texto preprocesado: un texto en el que cada segmento (normalmente cada oración) está identificado como tal, y en el que

¹Wall y Schwartz (1991).

²DuBois (2002).

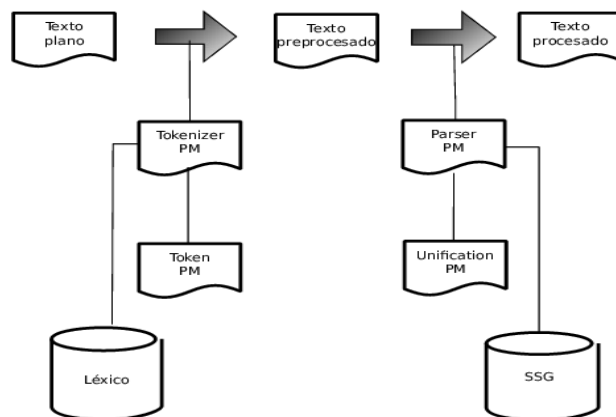


Figura A.1: Flujo principal de SGP

cada *token* (normalmente cada palabra) está también identificado y analizado. A su vez, este texto preprocesado es el *input* para el análisis sintáctico. SGP, con la librería **Parser**, y de acuerdo con una gramática (SSG por ejemplo), toma una lista de segmentos (de oraciones) y para cada uno de ellos ejecuta la función central de análisis sintáctico. El resultado de ello es que para cada segmento se proporcionan 0 , 1 o más análisis sintácticos de los cuales se puede sacar una u otra información: la historia derivativa que ha dado lugar a un análisis, el significado asociado a cada segmento, etc.

Tokenizer y **Parser** son las librerías centrales de SGP. En A.1 se puede ver cómo estas librerías se sirven para funcionar de otras librerías auxiliares, así como de bases de datos. En primer lugar, para la identificación de segmentos y *tokens* es necesaria una gramática que defina qué es un segmento y qué es un *token*. Con tal fin se ha creado la librería **Token**, que cuenta con un conjunto de expresiones regulares que definen los distintos tipos de *tokens* básicos de un texto: palabras separadas por espacios, con mayúscula, sin mayúscula, signos de puntuación de apertura, de cierre, etc. Estos elementos atómicos identificados se utilizan después en **Tokenizer** para

desencadenar transiciones de distintos autómatas de estados finitos. Estos autómatas definen los elementos no atómicos del análisis: segmentos y *tokens* complejos (nombres propios con varias palabras, uniones de clíticos y verbo, etc.). Además, a cada *token* léxico identificado se le da un análisis: se dice de él a qué palabra corresponde (se le asigna un lema), y a qué flexión de esa palabra corresponde (se le asignan unas reglas flexivas). Esta información está guardada en la base de datos *lexicon*, y de ella se extrae para anotar los *tokens*. El resultado de este preprocesamiento es un documento XML³ que representa al texto preprocesado identificando en él sus segmentos y en cada segmento sus *tokens*, anotados con la información léxica pertinente.

Después, SGP, mediante la librería *Parser*, toma el texto preprocesado y analiza por separado cada segmento. *Parser* analiza siempre conforme a una gramática de tipo HPSG que toma de una base de datos⁴. Su primera función es convertir cada *token* del segmento en la palabra correspondiente en la gramática. Después unirá sucesivamente esas palabras para formar constituyentes mayores. Todas estas unidades gramaticales que crea y combina *Parser* son representaciones de estructuras de rasgos previstas por la gramática con la que analiza. En concreto, el sistema codifica estas estructuras de rasgos como objetos de Perl. Cada estructura de rasgos es un objeto de tipo *type* —un tipo de HPSG— cuyos rasgos se recogen como objetos de

³Como referencia general sobre XML, véase Ray (2001). La estructura XML usada en SGP es deudora de la utilizada por el entorno LKB.

⁴Son bases de datos relacionales alojadas en un servidor MySQL. Cada gramática se corresponde con una base de datos que lleva su nombre (para este trabajo se ha usado, por ejemplo, la base de datos *ssg*). Todas tienen el mismo esquema relacional. Cuentan con una tabla *type* que recoge los tipos de la gramática, una tabla *heritage* que recoge las relaciones de herencia inmediata entre los tipos, una tabla *path* que recoge las tuplas que representan la estructura de rasgos de cada tipo, y una tabla *feature* que recoge los rasgos declarados en la gramática: informa del tipo en el que se declaró y del valor que se le asignó. SGP cuenta con una librería *grammar* que gestiona estas bases de datos: sirve de interfaz entre la base de datos y *Parser* y puede crear una base de datos a partir de su declaración en archivos TDL. Para analizar archivos TDL, *grammar* usa un analizador (TDL_Tok) que a su vez se sirve de una gramática de expresiones regulares de la sintaxis TDL (TDL_Gram).

tipo `feature` anidados. A su vez, cada objeto de tipo `feature` tiene anidado un objeto de tipo `type`: su valor⁵. Recuérdese que en HPSG se definen las posibilidades de combinación de constituyentes en términos de unificación. Por tanto, `Parser` permitirá o no la combinación de dos constituyentes (dos objetos de tipo `type`) mediante la librería `Unification`, encargada de comprobar si dos estructuras de rasgos (dos objetos de tipo `type`) unifican o no.

`Parser` usa un algoritmo *bottom-up* con *chart*. Los *parsers bottom-up* operan de abajo hacia arriba: parten de las unidades léxicas de la secuencia e intentan combinarlas hasta alcanzar un axioma⁶. Es un algoritmo clásico, con la salvedad de que admite la creación de constituyentes discontinuos. Todo *parser* admite la creación de sintagmas a partir de la unión de dos constituyentes si la gramática legitima tal unión y si entre ellos se establece una determinada relación de localidad. Para un *parser* habitual, los elementos unidos deben ser adyacentes en la secuencia. En cambio, los *parsers* que permitan la creación de constituyentes discontinuos deben prescindir de este requerimiento: solo deben comprobar que los elementos que unen no se so-

⁵SGP se ajusta a las prácticas habituales de la programación orientada a objetos. Las librerías mencionadas se han diseñado como clases que definen objetos: `Tokenizer` es un objeto que analiza unidades textuales, `Parser` es un objeto que analiza sintácticamente, `Unification` es un objeto que unifica estructuras de rasgos, etc. Además, tanto las unidades textuales del preprocesamiento como las unidades gramaticales del análisis profundo se han modelado como objetos. En el preprocesamiento, un texto como tal es un objeto de tipo `text`, cada segmento es un objeto de tipo `segment`, cada *token* es un objeto de tipo `token`, etc. Un objeto `Tokenizer` puede crear a su vez objetos de estos tipos, anidar convenientemente unos en otros, leer las secuencias de objetos anidadas y modificarlas, escribir un documento de texto XML que represente a un objeto de tipo `text`, etc. Por su parte, un objeto `Parser`, de acuerdo con una gramática, puede crear y combinar unidades gramaticales en forma de estructuras de rasgos. Estas estructuras son, también, objetos de Perl —`type`, `feature`, etc.—. Un objeto de tipo `Parser` hace uso, a su vez, de un objeto de tipo `Unification`. Este objeto es capaz de comparar dos objetos de tipo `type` y decidir si unifican o no, así como de crear el objeto `type` fruto de tal unificación.

⁶Para una explicación pormenorizada del funcionamiento de este tipo de *parsers* véase Gazdar y Mellish (1989).

lapen⁷. `Parser` se ha creado de forma tal que puede configurarse para admitir o no la creación de constituyentes discontinuos.

Una vez el sistema ha calculado, conforme a la gramática, todas las posibilidades de combinación sucesiva de constituyentes, `Parser` comprueba cuáles de los constituyentes construidos son análisis adecuados del segmento en cuestión. Se considera que un constituyente es un análisis legítimo si aglutina a todas las unidades léxicas del segmento, y si unifica con alguno de los axiomas definidos en la gramática. De estos análisis legítimos se puede extraer cierta información. Para el análisis de la *test suite* del apéndice B se extrajo la información que resultó relevante: una representación de la historia derivativa de cada análisis (gracias a la cual se pudo comprobar si el análisis en cuestión era el esperado) y un cálculo del tiempo de procesamiento de cada segmento (gracias al cual se pudo comparar la velocidad de análisis de SSG con la de una gramática tradicional). Para llevar a cabo estas tareas se usaron, respectivamente, las librerías `GraphViz`⁸ y `Benchmark`⁹.

Con todo ello, queda explicado el funcionamiento principal de SGP. Para acabar, se mostrarán ciertos aspectos de SGP que pudieran ser interesantes para la obtención de un sistema de análisis eficiente para determinados propósitos. Se han desarrollado en SGP módulos de preprocesamiento que permiten simplificar la complejidad de uso de las gramáticas. Necesitan ser desarrollados en el futuro, pero permiten ya dar cuenta de oraciones como las de la *test suite* del apéndice B. Efectivamente, se han utilizado en el análisis de dicha *test suite*; se han aplicado en igualdad de condiciones a SSG y a la gramática tradicional con la que se comparó.

En concreto, la librería `Tokenizer` cuenta con las librerías auxiliares siguientes: `clitics`, `np_chunk`, `vp_chunk`, `left_corner` y `silence_pron`. Estas librerías funcionan tras el análisis central de `Tokenizer`. Este análisis central

⁷Hay que evitar que un elemento cree una proyección mediante la unión consigo mismo.

⁸Brocard (2000).

⁹Hietaniemi y Bunce (1994).

identifica las palabras de cada segmento y las anota con la información léxica pertinente, tal como ya se ha explicado. A continuación, gracias a `clitics`, los clíticos y su verbo, que en un primer análisis se codificaron en *tokens* diferentes, se unen en un único *token*. A partir del análisis resultante de este proceso, con la librería `np_chunk`, se identifican las secuencias de *tokens* que forman un sintagma nominal, y se unen también en un único *token* (en realidad se identifican sintagmas nominales, preposicionales, y pronombres de distintos tipos). Del mismo modo, mediante `vp_chunk`, se unen en un *token* las secuencias formadas por un verbo principal y sus auxiliares y modales. Cada uno de estos procesos está regulado por un autómata de estados finitos¹⁰ que va identificando en el *input* (una cadena de *tokens*) las secuencias pertinentes. Para elaborar adecuadamente los *tokens* complejos resultantes es importante que en cada paso se recupere la información relevante. Por ejemplo, un *token* verbal complejo, cuando incorpora clíticos, los codifica en forma de regla flexiva; cuando incorpora al verbo finito, codifica también la regla flexiva correspondiente; y cuando incorpora al verbo principal, especifica el lema. Por su parte, el *token* correspondiente a un sintagma nominal incorpora información sobre el caso (al identificar o no una determinada preposición), sobre la flexión (al identificarla en determinantes, nombres o pronombres), etc.

Las dos librerías que entran en juego a continuación —`left_corner` y `silence_pron`— buscan en el segmento evidencias de la existencia de unidades léxicas sin realización fonética. La estrategia utilizada es buscar elementos desplazados a la margen izquierda, que pudieran indicar la existencia de huellas de desplazamiento. También se comprueba si la flexión verbal (incluidos los clíticos) tiene o no un correlato en forma de sintagmas explicitados. La falta de este correlato puede indicar la presencia de un pronombre correspondiente sin realización fonética. De acuerdo con estas indagaciones, las librerías `left_corner` y `silence_pron` añaden a la secuencia de *tokens* del

¹⁰Todos los autómatas de estados finitos que se utilizan en SGP funcionan gracias a una misma librería `Automata`.

texto preprocesado los *tokens* caracterizadores de las huellas y pronombres omitidos encontrados¹¹. Es decir, los elementos sin realización fonética necesarios para cada oración, aun habiendo sido caracterizados apropiadamente como huellas o pronombres omitidos, sí aparecen explícitamente en el *input* del análisis sintáctico. Gracias a ello, se han podido implementar el *parser* y las gramáticas¹² sin añadir complejidad alguna para permitir proyecciones con este tipo de elementos. Esto supone un gran ahorro computacional, pues el sistema solo permite crear proyecciones a partir de elementos sin realización fonética si estos aparecen en la secuencia de *tokens* del *input* (es decir, si se ha encontrado evidencia de que son necesarios para el análisis).

El resultado de la utilización de estos módulos de preprocesamiento es una gran simplificación del proceso de análisis. Los sintagmas nominales se conciben como elementos atómicos, caracterizados en cuanto a flexión, caso (en función de la presencia o ausencia de determinadas preposiciones), carácter pronominal, interrogativo... Los verbos auxiliares y modales y sus verbos principales forman un único ente atómico también. Las huellas y pronombres omitidos se representan como *tokens* igual que los sintagmas nominales explícitos, si bien desencadenarán (en tanto que huellas y pronombres omitidos) procesos no locales y presencia de clíticos en los verbos a los que se unen. Obviamente, todo ello es una abstracción, útil o no en función del propósito de la herramienta en cuestión. Para esta investigación ha resultado muy útil, pues

¹¹La librería `silence_pron` se encuentra en una versión sencillísima necesitada de un desarrollo futuro. Solo permite la legitimación de pronombres omitidos de primera y segunda persona (cuya referencia siempre se puede recuperar en cualquier contexto). En la *test suite* del apéndice B, los referentes que legitiman la aparición de pronombres de tercera persona se han explicitado de forma trivial como sintagmas encorchetados al comienzo de su oración. Desde ahí `left_corner` los puede identificar como tópicos y crea su correspondiente huella. Es una versión provisional. Esto es posible para SSG pues huella de topicalización y pronombre personal omitido tienen una misma caracterización. La versión ideal de este módulo debería recuperar del contexto de cada oración los referentes posibles para sus pronombres omitidos de tercera persona y calcular *tokens* correspondientes.

¹²Recuérdese que SSG tampoco recogía reglas que permitieran la proyección de un núcleo sin presencia explícita de un complemento.

ha permitido simplificar las gramáticas y los procesos de análisis y así centrar el atención en la demostración de que las mejoras que propone SSG son beneficiosas en términos de eficiencia computacional y elegancia conceptual¹³.

Se acaba este apéndice con la presentación de una última herramienta destinada a recabar información en el preprocesamiento para simplificar el análisis profundo posterior. La librería `vp_chunk` vista anteriormente permite también deducir, a partir de la presencia en los verbos de determinados clíticos o auxiliares, qué patrones de realización están presentes para un verbo determinado en una oración determinada. Por ejemplo, un verbo, en principio, podría admitir los usos transitivo, intransitivo, pasivo y de pasiva refleja. Pero si se observa en su flexión un clítico de acusativo —*lo*— las posibilidades se reducen a una: el uso transitivo. Si, en cambio, se encuentra un auxiliar *ser*, solo será posible el patrón pasivo. Hay, como se ve, mucha información relativa a las unidades léxicas que el preprocesamiento puede dar fácilmente y que restringe enormemente sus posibilidades de realización. SGP explora estas posibilidades.

¹³A fin de ganar en sencillez y rapidez, para el análisis de la *test suite* del apéndice B no se han usado reglas gramaticales de núcleo a la derecha (reglas de sujeto, tópico, etc.). Los sujetos, tópicos y partículas interrogativas de las oraciones de la *test suite* han legitimado, en el preprocesamiento, gracias a la librería `left_corner`, la creación de los *tokens* caracterizadores de sus correspondientes huellas. Es más, se han cambiado los *tokens* de esos argumentos a la izquierda del núcleo por los de sus correspondientes huellas a su derecha. Con ello, podría perderse en la caracterización de las oraciones la información relativa a las relaciones de ámbito de los elementos desplazados. Pero las gramáticas utilizadas en este trabajo no cuentan de momento con tal tipo de información en su signo lingüístico.

Apéndice B

Test suite

En el presente apéndice se muestra la *test suite* mediante la cual se ha depurado SSG. Se persigue con ello un objetivo doble: comprobar la cobertura de la gramática y demostrar que el uso de las estrategias de SSG reduce significativamente los tiempos de análisis de las gramáticas previas.

En cuanto al primer objetivo, la siguiente *test suite* demuestra que SSG tiene la cobertura que se le supone: no genera oraciones que no está previsto que genere, ni deja de generar oraciones que sí debe generar. La *test suite* recoge oraciones que demuestran que los argumentos internos de los verbos que los admiten pueden responder a diversas caracterizaciones en función de su categoría y marca de caso. Esta multiplicidad de posibilidades es interesante, porque SSG la cubre sin necesidad de estipular *ad hoc* cada posibilidad, sino haciendo valer la caracterización dada al tipo que modela lo que puede ser un argumento interno: *inner*, subtipo de *argument*. Véase 3.3. También se recogen las oraciones necesarias para comprobar que los distintos tipos de verbos participan de las alternancias de diátesis adecuadas. Como se explicó en el capítulo 3, SSG regula todo ello con un sencillo sistema de nueve reglas léxicas. A continuación se listan en la *test suite* las oraciones pertinentes para demostrar que SSG da cuenta adecuadamente de los fenómenos de cliticación del español —sin necesidad de uso de reglas ni unidades léxicas, tal como explica el capítulo 4—. Cabe apuntar que se ha optado por

dejar sin análisis a aquellas oraciones que muestran clíticos que, tal como se explicó en 4.5, resultan problemáticos para SSG. Termina la *test suite* con una serie de oraciones que demuestran que SSG cubre los distintos órdenes de palabras posibles en español. Tal como se explica en el capítulo 5, es posible la inversión de sujeto y el fenómeno de *scrambling* entre los complementos posverbales, sin necesidad de añadido de reglas léxicas; y es posible también la topicalización de complementos y la anteposición al verbo de partículas interrogativas.

El segundo objetivo de este apéndice es demostrar que los análisis de SSG ayudan a la elaboración de gramáticas más rápidas que las gramáticas previas. No sería adecuado hacer una comparación de SSG con SRG (referente de gramática del español en la actualidad), pues en la velocidad de procesamiento inciden multitud de factores que distorsionan los resultados (cobertura total de la gramática, requerimientos de preprocesamiento, etc.) En cambio, se ha optado por establecer una comparación entre SSG y lo que se ha decidido llamar NSSG (*No Simplified Spanish Grammar*). NSSG es una gramática prácticamente idéntica a SSG, salvo por el añadido de ciertas reglas léxicas habituales en las gramáticas tradicionales. Estas reglas son las utilizadas habitualmente para modelar las alternancias léxicas, la aparición de clíticos, la inversión de sujeto y el *scrambling*. En realidad, NSSG se beneficia de buena parte de la simplicidad de SSG. Por un lado, comparte su concepción morfológica de los clíticos (y el módulo de preprocesamiento que une clíticos y verbo), lo cual le permite prescindir de una regla sintáctica de saturación de estos¹. También usa el sistema de reglas que añaden argumentos a una lista con independencia de la longitud original de esta (lo cual le evita

¹Los tiempos de análisis del preprocesamiento completo de una oración con SGP son cercanos a 0 segundos. No hay lugar para una reducción significativa de tiempos de análisis mediante la eliminación del módulo que une clíticos y verbo. En cambio, el análisis de esta *test suite* demuestra que el añadido de unas pocas reglas a la gramática sí tiene un impacto significativo en el incremento de tiempo de análisis. No parece, por tanto, que una versión sintactista del fenómeno de los clíticos pueda mejorar, *per se*, los tiempos de análisis del sistema de SSG.

multiplicar innecesariamente las reglas de añadido de dativo, objeto oblicuo y complemento predicativo). El sistema de preprocesamiento y el *parser* de SGP funcionan exactamente igual en el análisis con ambas gramáticas, salvo en lo relativo a la posibilidad de crear constituyentes discontinuos. Efectivamente, a diferencia de SSG, NSSG analiza con el *parser* configurado para no permitir la creación de tales constituyentes, lo cual reduce, en principio, las posibilidades de combinación y los tiempos de análisis. Pero, como se verá, esta reducción no compensa, ni con mucho, el coste de añadir reglas léxicas que modelen el *scrambling*.

A continuación se explican con detalle los cambios que se han efectuado sobre SSG para crear NSSG.

Tal como se vio en 3.3, uno de los logros de SSG ha sido haber modelado las posibilidades de realización de los argumentos de los verbos del español sin necesidad de crear reglas léxicas o unidades léxicas específicas para las distintas posibilidades. Por ejemplo, existen verbos que admiten como objeto tanto un sintagma nominal como un complementante. Y, en función de cuestiones de caso, estos complementos irán precedidos o no de preposición. En *Matrix*, efectivamente, se modelan con tipos distintos el verbo con complemento nominal frente al verbo con complementante. Esta duplicidad se le ha añadido a NSSG. Para aquel verbo que en una oración usara el patrón transitivo único de SSG, en NSSG se han contemplado *v-transitive-np-c* y *v-transitive-comp-c*. Estas reglas se representan en las figuras B.1 y B.2.

$$\left[\begin{array}{l} v\text{-transitive-np-c} \\ \text{ARG-ST} \left\langle \text{SYNSEM}, \left[\begin{array}{l} \text{LOCAL|HEAD} \quad \left[\begin{array}{l} det \\ \text{CASE } acc \end{array} \right] \right] \\ \text{NON-LOCAL|CLITICS} \langle \rangle \end{array} \right] , \dots \right\rangle \end{array} \right]$$

Figura B.1: *v-transitive-np-c*

Del mismo modo, a aquellos verbos que cuentan en una oración con el patrón único de añadido de oblicuo de SSG se les ha supuesto en NSSG dos

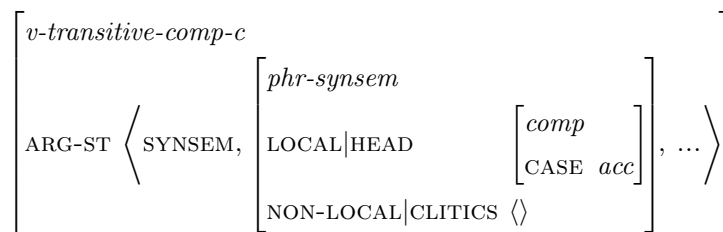


Figura B.2: v-transitive-comp-c

patrones: v-obliq-np-c y v-obliq-comp-c. Estas reglas se representan en las figuras B.3 y B.4².

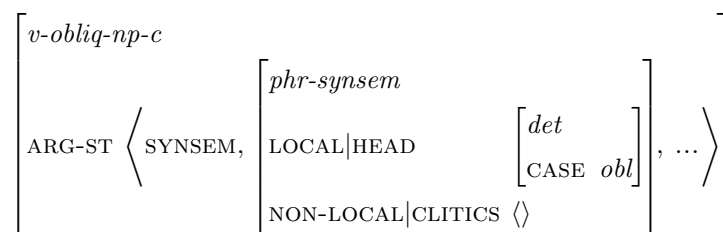


Figura B.3: v-obliq-np-c

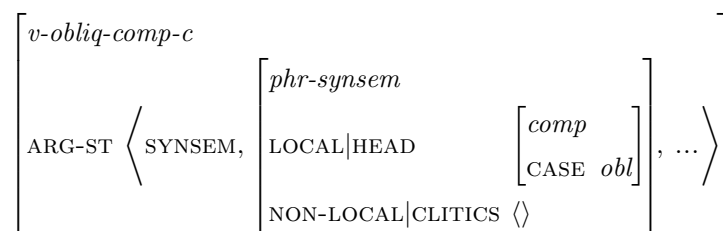


Figura B.4: v-obliq-comp-c

²Obsérvese en estas estructuras de rasgos que las preposiciones de marca de caso no se están tratando como núcleo del sintagma al que marcan —esto es común a las versiones actuales de SSG y NSSG—. Este análisis es posible gracias al módulo de preprocesamiento que crea *chunks* consistentes en sintagmas nominales marcados con uno u otro caso en función de la identificación o no de preposiciones de marca de caso —apéndice A—. Este análisis es una simplificación en relación a lo propuesto desde un punto de vista teórico para SSG —3.3—, pero es compatible con la concepción de las marcas de caso de Pollard y Sag (1994).

Por otro lado, SSG cuenta con un sistema de clíticos que no necesita ni reglas léxicas que conviertan un argumento en clítico —estrategia usada comúnmente en HPSG— ni unidades léxicas cuya existencia no sea ya necesaria en la gramática por motivos independientes. A NSSG, en cambio, sí se le han añadido las reglas que regulan la conversión en clíticos de sus argumentos de acusativo y dativo. En concreto, en NSSG se incluye una regla que hace aparecer el clítico correspondiente a un argumento de acusativo consistente en un pronombre personal —B.5— y la que hace aparecer el clítico cuando el argumento es un elemento elidido, ya sea pronombre o huella de topicalización —B.6—.

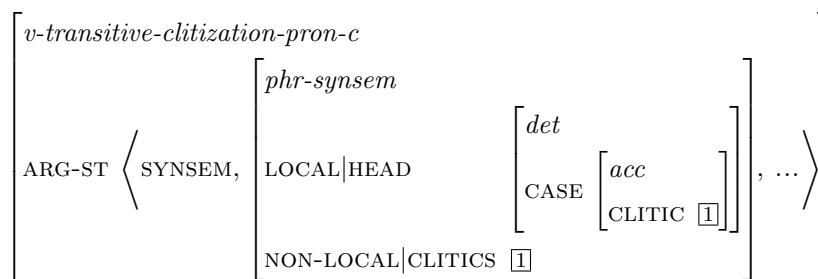


Figura B.5: v-transitive-clitization-pron-c

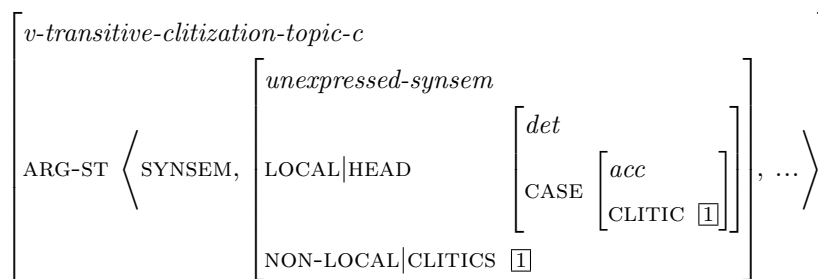


Figura B.6: v-transitive-clitization-topic-c

También se han añadido a NSSG las reglas que regulan la aparición de los clíticos de dativo: en B.7 se muestra la regla que hace aparecer el clítico cuando el dativo es un pronombre personal y en B.8 se muestra la que hace aparecer tal clítico por elisión o topicalización del argumento.

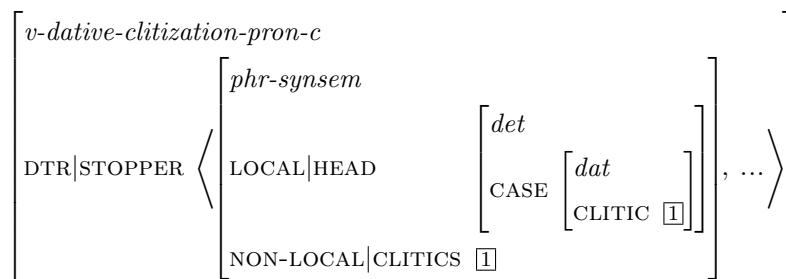


Figura B.7: v-dative-clitization-pron-c

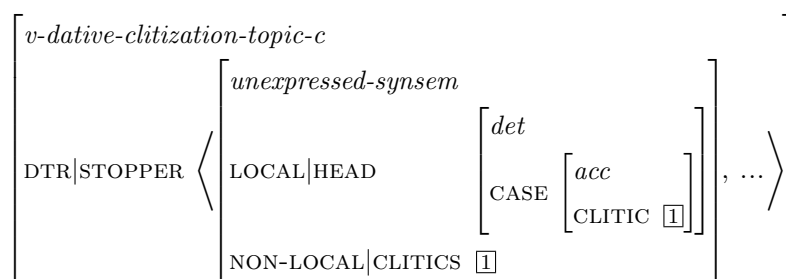


Figura B.8: v-dative-clitization-topic-c

Para permitir la aparición del clítico correspondiente al dativo realizado como sintagma preposicional posverbal o como partícula interrogativa desplazada, sería necesario añadir dos reglas específicas. Se ha optado por no cubrir estas posibilidades. Como se vio en 4.5, estos datos son tan problemáticos para SSG (cuya versión actual tampoco los contempla) como para las gramáticas tradicionales.

Por último, a NSSG se le han añadido reglas que le permiten dar cuenta del orden de palabras del español de acuerdo con los análisis habituales en las gramáticas previas. El tipo *unergative* regula en SSG la relación entre la lista ARG-ST y las listas SUBJ y COMPS para los verbos con argumento externo. En SSG, SUBJ siempre queda vacía, pues incluso los argumentos externos pasan de ARG-ST a COMPS. Con este modelado, como se vio en 5.1, todo sujeto se satura de forma posverbal desde COMPS, y la posición preverbal se entiende como fruto de un desplazamiento. Gracias a ello, SSG puede prescindir de cualquier regla específica para tratar los sujetos: la inversión es fruto de

una saturación canónica como complemento y la anteposición se logra con el sistema general de topicalización. En cambio, este tipo *unergative* se ha reformulado en NSSG de acuerdo con el análisis habitual en HPSG para los sujetos: el primer elemento de ARG-ST (el sujeto) pasa a SUBJ y el resto de la lista (los complementos) pasa a COMPS —B.9—.

$$\left[\begin{array}{l} \textit{unergative} \\ \text{SYNSEM|LOCAL|VAL} \\ \text{ARG-ST} \end{array} \left[\begin{array}{l} \text{SUBJ} \langle \boxed{1} \rangle \\ \text{COMPS} \boxed{2} \\ \langle \boxed{1} . \boxed{2} \rangle \end{array} \right] \right]$$

Figura B.9: *unergative*

Con esta caracterización, todos los argumentos externos de la gramática NSSG deberían saturarse desde SUBJ con una regla específica *subj-rule* como la vista en 1.3.3.2. Además, para dar cuenta de los casos de inversión de sujeto, NSSG necesita una regla *inversion-c*, representada en B.10³.

$$\left[\begin{array}{l} \textit{inversion-c} \\ \text{SYNSEM|LOCAL|VAL} \\ \text{DTR|SYNSEM|LOCAL|VAL} \end{array} \left[\begin{array}{l} \text{SUBJ} \langle \rangle \\ \text{COMPS} \langle \boxed{1} . \boxed{2} \rangle \\ \text{SUBJ} \langle \boxed{1} \rangle \\ \text{COMPS} \langle \boxed{2} \rangle \end{array} \right] \right]$$

Figura B.10: *inversion-c*

El *scrambling*, por su parte, se ha modelado en NSSG mediante uno de los sistemas tradicionales: el uso de reglas léxicas que especifican los distintos órdenes posibles para los elementos de la lista COMPS. En total, se han

³En realidad, se ha permitido a NSSG beneficiarse del preprocesamiento que hace SGP: se identifican los argumentos preverbiales como elementos desplazados y se coloca en posición posverbal la huella correspondiente —apéndice A—. Gracias a ello, y a la regla de inversión, no ha sido necesario añadir a NSSG una regla de saturación de SUBJ, pues los sujetos se han podido encontrar siempre tras su verbo.

creado seis reglas léxicas de *scrambling*: *scrambling-c-one* —B.11— legitima a los verbos con dos complementos en COMPS en un orden inverso al canónico. Por su parte, *scrambling-c-two* —B.12—, *scrambling-c-three*, *scrambling-c-four*, *scrambling-c-five* y *scrambling-c-six* legitiman las distintas ordenaciones posibles de listas de tres elementos necesarias para dar cuenta de las oraciones de la *test suite*⁴.

$$\left[\begin{array}{l} \textit{scrambling-c-one} \\ \text{SYNSEM|LOCAL|VAL} \quad \left[\text{COMPS} \langle \underline{2}, \underline{1} \rangle \right] \\ \text{DTR|SYNSEM|LOCAL|VAL} \quad \left[\text{COMPS} \langle \underline{1}, \underline{2} \rangle \right] \end{array} \right]$$

Figura B.11: *scrambling-c-two*

$$\left[\begin{array}{l} \textit{scrambling-c-two} \\ \text{SYNSEM|LOCAL|VAL} \quad \left[\text{COMPS} \langle \underline{1}, \underline{3}, \underline{2} \rangle \right] \\ \text{DTR|SYNSEM|LOCAL|VAL} \quad \left[\text{COMPS} \langle \underline{1}, \underline{2}, \underline{3} \rangle \right] \end{array} \right]$$

Figura B.12: *scrambling-c-two*

Se muestra a continuación la *test suite* en cuestión, en forma de tabla. En la columna de la izquierda aparecen las oraciones que la componen. Se ha usado la convención habitual de colocar un asterisco ante las oraciones agramaticales. Además, se han colocado entre corchetes —tal como se explicó en el apéndice A— los antecedentes de los pronombres omitidos de tercera persona. En la segunda columna aparece el número de análisis esperable para cada oración. A continuación, en la tercera columna, se encuentra el número de análisis que da SSG a la oración correspondiente, junto al tiempo de CPU empleado en segundos (entre paréntesis). Por último, en la columna de la derecha, se muestran los análisis obtenidos por NSSG junto al tiempo de CPU, también en segundos y entre paréntesis.

⁴En la *test suite* hay también oraciones con cuatro elementos en COMPS. A fin de evitar tiempos de análisis prohibitivos, para estas oraciones no se han añadido reglas de *scrambling*. Quedan, por tanto, sin análisis.

Oración	NºA	SSG	NSSG
Argumentos internos de distinta categoría			
1. Sé cosas.	1	1 (3,35)	1 (97,27)
2. Sé los precios.	1	1 (3,82)	1 (110,22)
3. Sé que han subido los precios.	1	1 (9,94)	1 (180,83)
4. Sé calcular los precios.	1	1 (2,05)	1 (17,05)
5. Me acuerdo de cosas.	1	1 (4,85)	1 (103,65)
6. Me acuerdo del precio.	1	1 (4,93)	1 (105,36)
7. Me acuerdo de que ha subido el precio.	1	1 (12,46)	1 (170,06)
8. ¿Te acuerdas de subir los precios?	1	1 (12,87)	1 (375,65)
Realizaciones y logros			
9. El hotel ha contratado al guía.	1	1 (1,71)	1 (16,72)
10. Ha sido contratado un guía.	1	1 (1,08)	1 (2,77)
11. Se han contratado dos guías.	1	1 (1,08)	1 (2,75)
12. *El hotel ha contratado.	0	0 (1,13)	0 (16,39)
13. *[Ellos] El hotel ha contratado que suban el precio.	0	0 (9,23)	0 (82,55)
14. El botones ha subido el equipaje hasta la habitación.	1	1 (7,02)	1 (68,88)
15. ¿Ellos han convencido al guía de todo?	1	1 (4,17)	1 (110,74)
16. Ellos han comparado el hotel con el resort.	1	1 (4,07)	1 (62,87)
17. El hotel nos ha contratado un guía.	1	1 (6,26)	1 (164,78)
18. Nos ha sido contratado un guía.	1	1 (2,53)	1 (12,16)
19. Se nos han contratado dos guías.	1	1 (2,55)	1 (12,26)
20. El botones os ha subido el equipaje hasta la habitación.	1	1 (28,91)	0 (397,58)
21. ¿[A él] Ellos os lo han convencido de todo?	1	1 (15,08)	0 (556,33)
22. Ellos os han comparado el hotel con el resort.	1	1 (14,93)	0 (349,02)
23. Ha subido el precio.	1	1 (3,21)	1 (66,11)
24. Ha subido el precio hasta cien euros.	1	1 (4,41)	1 (70,25)
25. Nos ha subido el precio hasta cien euros.	2	2 (13,15)	2 (414,67)
26. *Ha subido que suban el equipaje.	0	0 (5,91)	0 (95,91)

 Actividades y estados

27. Tú sabes.	1	1 (3,22)	1 (106,78)
28. [Eso] Tú lo sabes.	1	1 (4,48)	1 (116,05)
29. [Ellos] Sabes si han subido el precio.	1	1 (17,60)	1 (185,21)
30. Se sabe el precio.	1	1 (1,03)	1 (2,46)
31. *Ha sido sabido el precio.	0	0 (2,66)	0 (115,67)
32. Tú sabes de eso.	1	1 (4,35)	1 (119,17)
33. Tienes las entradas.	1	1 (2,26)	1 (17,49)
34. *Tienes de las entradas.	0	0 (2,07)	0 (17,23)
35. Puedo confiar en ti.	1	1 (2,07)	1 (9,80)
36. *Puedo confiar a ti.	0	0 (1,96)	0 (9,41)
37. *Se confía él.	0	0 (1,43)	0 (9,47)
38. El hotel se desentiende de la situación.	1	1 (2,51)	1 (11,45)
39. *El hotel se desentiende la situación.	0	0 (2,83)	0 (12,08)
40. *El hotel desentiende de la situación.	0	0 (2,10)	0 (11,87)
41. *El hotel desentiende la situación.	0	0 (2,11)	0 (11,25)
42. ¿Tú te relacionas con el guía?	1	1 (4,92)	1 (57,04)
43. *¿Tú relacionas con el guía?	0	0 (2,09)	0 (7,66)
44. [Él] Admiraba su belleza.	1	1 (4,37)	1 (121,58)
45. [Él] Se admiraba de su belleza.	1	1 (4,64)	1 (126,38)
46. *[Él] Admiraba de su belleza.	0	0 (4,16)	0 (127,39)
47. *[Él] Se admiraba su belleza.	0	0 (5,01)	0 (128,19)
48. Has secado la toalla.	1	1 (2,40)	1 (20,00)
49. Se ha secado la toalla.	1	1 (1,91)	1 (19,44)
50. [Eso] Tú me lo sabes.	1	1 (12,59)	1 (594,51)
51. [Eso] Tú me lo tienes.	1	1 (6,00)	1 (147,00)
52. Tú me confías en Él.	1	1 (5,72)	1 (87,70)
53. El hotel se te desentiende de la situación.	1	1 (6,82)	1 (104,25)
54. Tú te me debes relacionar con el guía.	1	1 (6,88)	1 (7,78)
55. Te admiras de su belleza.	1	1 (9,73)	1 (655,42)
56. Me has secado la toalla.	1	1 (6,27)	1 (187,36)

Clíticos

57. La entrada puedo comprarla.	1	1 (2,20)	1 (18,42)
58. Al encargado le has dado la entrada.	1	1 (3,98)	1 (167,81)
59. ¿[La entrada] Puedo comprarla?	1	1 (2,10)	1 (18,77)
60. ¿[A ella] Le has dado la entrada?	1	1 (3,89)	1 (165,99)
61. ¿La has encontrado a ella?	1	1 (1,63)	1 (17,69)
62. ¿Le has dado la entrada a él?	1	1 (3,91)	1 (165,32)
63. *¿[La entrada] Puedo comprar?	0	0 (2,02)	0 (18,84)
64. *¿[A ella] has dado la entrada?	0	0 (3,76)	0 (161,77)
65. *¿Has encontrado a ella?	0	0 (1,51)	0 (17,72)
66. *¿Has dado la entrada a él?	0	0 (3,72)	0 (161,31)
67. ¿Has encontrado al guía?	1	1 (1,56)	1 (18,02)
68. *¿Lo has encontrado al guía?	0	0 (1,49)	0 (17,79)
69. ¿Has dado la entrada al encargado?	1	1 (4,20)	1 (158,60)
70. ¿Le has dado la entrada al encargado?	1	0 (4,17)	0 (163,08)
71. ¿Qué puedo comprar?	1	1 (2,15)	1 (18,17)
72. *¿Qué puedo comprarlo?	0	0 (1,98)	0 (18,21)
73. ¿A quién has dado la entrada?	1	1 (4,22)	1 (155,28)
74. ¿A quién le has dado la entrada?	1	0 (4,28)	0 (161,30)
75. [El guía, ellos] dicen que viene.	1	1 (5,18)	1 (23,77)
76. [Tú] Él dice que vienes.	1	1 (5,06)	1 (18,45)
77. [Él] Viene.	1	1 (1,46)	1 (3,74)
78. ¿Tú confías en él?	1	1 (2,04)	1 (9,03)
79. ¿En él confías tú?	1	1 (1,99)	1 (9,05)
80. ¿Tú confías?	1	1 (1,46)	1 (8,84)
81. El hotel lo queremos pagar nosotros.	1	1 (1,97)	1 (17,03)
82. El hotel queremos pagarlo nosotros.	1	1 (1,99)	1 (17,38)
83. *El hotel lo queremos pagarlo nosotros.	0	0 (0,58)	0 (1,30)
84. El hotel ¿sabes si lo podemos pagar nosotros?	1	1 (11,57)	1 (127,11)
85. *El hotel ¿lo sabes si podemos pagar nosotros?	0	0 (11,13)	0 (136,95)
86. [A él] Se le han perdido los billetes.	1	1 (2,52)	1 (11,54)
87. *[A él] Le se han perdido los billetes.	0	0 (0,66)	0 (1,45)
88. *[Ellos] Me me han contratado.	0	0 (0,65)	0 (1,83)
89. *[Ellos] Te te han contratado.	0	0 (0,65)	0 (1,84)
90. *[Ellos] Me te han contratado.	0	0 (0,80)	0 (1,99)
91. [Ellos] Te me han contratado.	1	1 (4,36)	1 (153,92)

92. [Ellos, eso, a él] Se lo han contratado.	1	1 (4,38)	1 (162,59)
93. *[Ellos, eso, a él] Le lo han contratado.	0	0 (0,81)	0 (2,17)
94. *[Ellos, eso, a él] Lo te han contratado.	0	0 (0,77)	0 (1,97)

Orden de palabras

95. ¿Has encontrado tú la toalla?	1	1 (1,58)	1 (17,27)
96. ¿Tú has encontrado la toalla?	1	1 (1,62)	1 (16,80)
97. ¿Le has dado la toalla a él?	1	1 (3,87)	1 (157,18)
98. ¿Le has dado a él la toalla?	1	1 (3,90)	1 (163,89)
99. ¿Le has dado tú a él la toalla?	1	1 (3,87)	1 (166,23)
100. ¿Le has dado a él tú la toalla?	1	1 (3,89)	1 (167,68)
101. ¿Le has dado a él la toalla tú?	1	1 (3,92)	1 (139,18)
102. ¿Le has dado tú la toalla a él?	1	1 (3,88)	1 (156,65)
103. ¿Le has dado la toalla tú a él?	1	1 (3,89)	1 (159,49)
104. ¿Le has dado la toalla a él tú?	1	1 (3,92)	1 (159,78)
105. ¿Has subido tú la toalla hasta la habitación?	1	1 (5,39)	1 (66,60)
106. ¿Has subido tú hasta la habitación la toalla?	1	1 (5,46)	1 (66,88)
107. ¿Has subido la toalla tú hasta la habitación?	1	1 (5,47)	1 (64,07)
108. ¿Has subido hasta la habitación tú la toalla?	1	1 (5,48)	1 (66,89)
109. ¿Has subido la toalla hasta la habitación tú?	1	1 (5,48)	1 (64,41)
110. ¿Has subido hasta la habitación la toalla tú?	1	1 (5,44)	1 (66,62)
111. ¿Qué has subido hasta la habitación?	1	1 (5,50)	1 (64,60)
112. La toalla ¿la has subido hasta la habitación?	1	1 (3,61)	1 (62,18)
113. La toalla ¿a quién se la has subido hasta la habitación?	1	1 (11,89)	0 (320,28)
114. *¿A quién la toalla se la has subido hasta la habitación?	0	0 (6,05)	0 (303,88)
115. La toalla, a él ¿se la has subido hasta la habitación?	1	1 (12,50)	0 (356,36)
116. A él, la toalla ¿se la has subido hasta la habitación?	1	1 (12,75)	0 (361,53)
117. *¿A quién qué has subido hasta la habitación?	0	0 (6,17)	0 (68,29)
118. *¿Qué a quién has subido hasta la habitación?	0	0 (6,02)	0 (69,00)

Todas las oraciones han sido analizadas con cada gramática diez veces. El tiempo de CPU que se muestra es la media aritmética de los valores obtenidos en estas diez iteraciones.

Como puede observarse comparando las columnas segunda, tercera y cuarta de la tabla, la cobertura de SSG y NSSG es básicamente la esperada. Solo se observan las siguientes divergencias entre los resultados obtenidos y el número de análisis esperable para cada oración. Como se ha dicho anteriormente, los clíticos de dativo correspondientes a sintagmas canónicos posverbales y a partículas interrogativas son problemáticos tanto para SSG como para una gramática tradicional (NSSG). Se ha optado por no incluir en las gramáticas una solución provisional. Por tanto, las oraciones con tal tipo de clíticos quedan sin analizar. Son las oraciones 70 y 74. Además, para evitar tiempos de análisis prohibitivos, no se han incluido en NSSG reglas de *scrambling* que legitimen los distintos órdenes de complementos pertinentes para listas COMPS con cuatro elementos. Por ello, las oraciones 20, 21, 22, 115 y 116 han quedado sin análisis para NSSG⁵.

Por último, cabe comentar que los tiempos de análisis utilizados por el *parser* con SSG son mucho menores que los utilizados con NSSG. SGP es un sistema de *parsing* pensado para la prueba y comparación de estrategias de análisis. No es un sistema eficiente en sí mismo. Se observa que el añadido de unas pocas reglas a una gramática dispara la explosión combinatoria y aumenta de forma muy sensible los tiempos de análisis de SGP. SSG usa nueve reglas léxicas y una gramatical. NSSG utiliza trece reglas más⁶. Con una diferencia de reglas de este tipo (se dobla, más o menos, el número de reglas) se ha obtenido un crecimiento exponencial de los tiempos de análisis. Es posible que un *parser* más eficiente sea capaz de reducir sensiblemente

⁵Téngase en cuenta que —de acuerdo con lo visto en el apéndice A— el análisis de estas oraciones en NSSG confía en la existencia de un sujeto invertido. Este sujeto y tres complementos suman los cuatro elementos en cuestión.

⁶Recuérdese —apéndice A— que el preprocesamiento elimina algunas reglas cuando encuentra evidencias de que es posible hacerlo.

este crecimiento exponencial. Pero los resultados obtenidos en este trabajo demuestran que, en lo que a la elaboración de la gramática se refiere, la reducción del número de reglas es esencial para lograr herramientas eficientes. Esta conclusión es la esperable. Sí resulta más interesante observar cómo, a la luz de estos resultados, cabe concluir que, para tratar el fenómeno de *scrambling*, el uso de un *parser* capaz de crear constituyentes discontinuos (característica que incrementa las posibilidades combinatorias y el tiempo de análisis) es preferible con mucho a la multiplicación de reglas léxicas.

Índice general

Agradecimientos	III
Summary	v
Introducción	XIX
1. Marco teórico	1
1.1. La Lingüística Generativa	1
1.1.1. Lenguajes formales y Teoría de la Computación	2
1.1.2. La jerarquía de lenguajes de Chomsky	6
1.1.3. Máquinas abstractas	18
1.1.4. Un formalismo adecuado para el lenguaje natural	31
1.2. La Gramática Transformacional	49
1.2.1. La Teoría Estándar	51
1.2.2. El modelo de Principios y Parámetros	53
1.2.3. El Programa Minimista	56
1.2.4. Conclusión	61
1.3. Las Gramáticas de Unificación. HPSG	62
1.3.1. Características generales	62
1.3.2. Estructuras de rasgos y unificación	65
1.3.3. HPSG	69
1.4. El modelo gramatical de SSG	102
2. HPSG: problemas	105
2.1. Explosión combinatoria	106

2.2. Adecuación descriptiva	110
2.3. Motivación lingüística	112
2.4. Estado de la cuestión	114
2.5. Conclusión	118
3. SSG: alternancias de diátesis	123
3.1. Revisión del signo lingüístico (I)	124
3.1.1. El rasgo HOOK	124
3.1.2. Teoría del ligamiento: ARG-ST	128
3.1.3. La noción de argumento: el tipo <i>argument</i>	131
3.1.4. Relación entre ARG-ST y VAL	138
3.2. Tipos verbales	141
3.3. Argumentos estructurales	148
3.4. Verbos pronominales	155
3.5. Argumentos no estructurales	160
3.5.1. Argumentos con caso nominativo o acusativo	161
3.5.2. Argumentos con caso oblicuo	165
3.5.3. Complementos predicativos	171
3.5.4. Argumentos con caso dativo	174
3.6. Reglas léxicas no destructivas	178
3.7. Alternancias de diátesis con reglas no destructivas	182
3.7.1. Realizaciones y logros	184
3.7.2. Actividades y estados	186
3.7.3. Alternancia acusativo oblicuo	190
3.7.4. Alternancia causativa	191
3.8. Alternancias con verbos pronominales	193
3.9. Conclusión	195
4. SSG: clíticos	199
4.1. Revisión del signo lingüístico (II)	200
4.1.1. El rasgo NON-LOCAL	200
4.1.2. La regla <i>comps-rule</i> con complementos elididos	207
4.2. Casuística	212

<i>ÍNDICE GENERAL</i>	306
4.3. Soluciones previas	215
4.4. El sistema de clíticos de SSG	218
4.4.1. Los clíticos como problema morfológico	218
4.4.2. Los clíticos motivados por la naturaleza del argumento	223
4.4.3. El salto de clíticos	231
4.5. Problemas	236
4.6. Conclusión	238
5. SSG: orden de palabras	241
5.1. El sujeto	244
5.2. <i>Scrambling</i>	253
5.2.1. Soluciones previas	254
5.2.2. Constituyentes discontinuos	258
5.2.3. Problemas	265
5.3. Dependencias no acotadas	269
5.4. Conclusión	277
6. Conclusión	279
A. El <i>parser</i> SGP	287
B. <i>Test suite</i>	295
Bibliografía	313

Bibliografía

- ARANDA, JOAQUÍN; DURO, NATIVIDAD; FERNÁNDEZ, JOSÉ LUIS; JIMÉNEZ, JOSÉ y MORILLA, FERNANDO: *Fundamentos de lógica matemática y computación*. Sanz y Torres, 2006.
- BALARI, SERGIO: «Pronouns, Variables and Extraction». En: *Romance in HPSG*, pp. 151–217. CSLI Publications, 1998.
- BARWISE, J. y PERRY, J.: *Situations and Attitudes*. The MIT Press, Cambridge, Massachusetts, 1983.
- BENDER, EMILY M.; FLICKINGER, DAN y OEPEN, STEPHAN: «The Grammar Matrix: An Open-Source Starter-Kit for the Rapid Development of Cross-Linguistically Consistent Broad-Coverage Precision Grammars», 2002. CSLI, Stanford University.
<http://www.delph-in.net/matrix>
- BRESNAN, J.: *Lexical-Functional Syntax*. Oxford, Basil Blackwell, 2001.
- BROCARD, LEON: «GraphViz», 2000.
<http://search.cpan.org/dist/GraphViz/>
- CARPENTER, BOB: *The Logic of Typed Featured Structures*. Cambridge University Press, 1992.
- : *Typed-Logical Semantics*. The MIT Press, Cambridge, Massachusetts, 1995.
- CHOMSKY, NOAM: «Three Models for the Description of Language». *IRE Transactions PGIT*, 1956, **2**, pp. 113–124.

- : *Syntactic Structures*. Mouton and co., N.Y. publishers, 1957.
- : *Aspects of the Theory of Syntax*. The MIT Press, Cambridge, Massachusetts, 1965.
- : *Barriers*. The MIT Press, Cambridge, Massachusetts, 1986a.
- : *Knowledge of Language: Its Nature, Origins and Use*. Praeger Publishers, N.Y., USA, 1986b.
- : *The Minimalist Program*. Massachusetts Institute of Technology (MIT Press), 1995.
- : *Sobre la naturaleza y el lenguaje*. Cambridge University Press, 2003.
- : «On Phases». En: *Foundational Issues in Linguistic Theory*, pp. 133–166. The MIT Press, Cambridge, Massachusetts, 2008.
- CHURCH, A.: «The calculi of lambda-conversion». *Annals of Mathematical Studies*, 1941, **6**.
- CONTRERAS, HELES: *El orden de palabras en español*. Cátedra, Madrid, 1978.
- COPESTAKE, ANN: *Implementing Typed Feature Structure Grammars*. University of Chicago Press, 2002.
<http://www.delph-in.net/LkbTop>
- COPESTAKE, ANN y FLICKINGER, DAN: «An open-source grammar development environment and broad-coverage English grammar using HPSG». *Proceedings of the Second conference on Language Resources and Evaluation*, 2000, pp. 15–28.
- COPESTAKE, ANN; FLICKINGER, DAN; POLLARD, CARL y SAG, IVAN A.: «Minimal Recursion Semantics: An Introduction». *Research on Language and Computation*, 1998, **3**, pp. 281–332.
- DALRYMPLE, M.; LAMPING, J. y SARASWAT, V.: *Formal Issues in Lexical Functional Grammar*. CSLI Publications, 1995.

- DEVLIN, K.: *Logic and Information*. Cambridge University Press, 1991.
- DONOHUE, CATHRYN y SAG, IVAN A.: «Domains in Warlpiri», 2006. Stanford University.
- DUBOIS, PAUL: *Programming Perl*. O'Reilly Media, 2002.
- EGUREN, LUIS y FERNÁNDEZ SORIANO, OLGA: *Introducción a una sintaxis minimista*. Gredos, 2004.
- FERNÁNDEZ SORIANO, OLGA: «Sobre el orden de palabras en español». *Cuadernos de Filología Hispánica*, 1993, **II**, pp. 113–152.
- : «El pronombre personal. Formas y distribuciones. Pronombres átonos y tónicos». En: Ignacio Bosque y Violeta Demonte (Eds.), *Gramática descriptiva de la lengua española*, volumen I, pp. 1209–1273. Espasa, 1999.
- FEYNMAN, RICHARD P.: *Conferencias sobre computación*. Crítica, 2003.
- FLICKINGER, DAN: «On building a more efficient grammar by exploiting types». *Natural Language Engineering*, 2000, **6**, pp. 15–28.
- GAZDAR, G.; KLEIN, E.; PULLUM, G. y SAG, I.: *Generalized Phrase Structure Grammar*. Harvard University Press, 1985.
- GAZDAR, G. y MELLISH, C. S.: *Natural Language Processing in PROLOG*. Addison-Wesley, 1989.
- GREENBERG, J.: «Some universals of language with particular reference to the order of meaningful elements». En: *Universals of Language*, The MIT Press, Cambridge, Massachusetts, 1963.
- GUTIÉRREZ-BRAVO, RODRIGO: «Topicalization and preverbal subjects in Spanish Wh-interrogatives». En: *Selected Proceedings of the 10th Spanish Linguistics Symposium*, pp. 225–236. Cascadilla Press, 2008.
- GUTIÉRREZ ORDÓÑEZ, SALVADOR: «Los dativos». En: *Gramática descriptiva de la lengua española*, volumen II, pp. 1855–1930. Espasa, 1999.

- GÖDEL, KURT: «Über Formal Unentscheidbare Sätze der Principia Mathematica und Verwandter Systeme. I». *Monatshefte für Mathematik und Physik*, 1931, **38**, pp. 173–198.
- : *On Formally Undecidable Propositions*. New York: Basic Books, 1962.
- HIETANIEMI, JARKKO y BUNCE, TIM: «Benchmark», 1994.
<http://perldoc.perl.org/Benchmark.html/>
- HOPCROFT, JOHN E.; MOTWANI, RAJEEV y ULLMAN, JEFFREY D.: *Introduction to Automata Theory, Languages and Computation*. Pearson Education, 2001.
- IRANZO, PASCUAL JULIÁN: *Lógica simbólica para informáticos*. RA-MA Editorial, 2004.
- JOHNSON, DAVID E. y POSTAL, PAUL M.: *Arc Pair Grammar*. Princeton University Press, 1980.
- JOSHI, ARAVIND K.: «Unification and Some New Grammatical Formalisms». En: Yorick Wilks (Ed.), *Theoretical Issues in Natural Language Processing*, pp. 36–40. Lawrence Erlbaum Associates, 1989.
- KAMP, HANS y REYLE, U.: *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kuwer Academic Publishers, 1993.
- KASPER, ROBERT T. y ROUNDS, WILLIAMS C.: «A logical semantics for feature structures». En: *Proceedings of the 24th Annual Meeting on Association for Computational Linguistics*, pp. 257–266. Association for Computational Linguistics, Morristown, NJ, USA, 1986.
- KATHOL, ANDREAS y POLLARD, CARL J.: «Extraposition via Complex Domains Formation». En: *Proceedings of the Thirty-Third Annual Meeting of the ACL*, Association for Computational Linguistics, Boston, 1995.

- KAY, MARTIN: «Functional Grammar». En: *Proceedings of the Fourth Annual Meeting of the Berkeley Linguistics Society*, Berkeley Linguistics Society, 1979.
- KLEENE, S.C.: «General recursive functions of natural numbers». *Mathematische Annalen*, 1936, **112**, pp. 727–742.
- LEVIN, B. y RAPPAPORT, M.: *Unaccusativity at the Syntax-Lexical Semantics Interface*. Massachusetts Institute of Technology (MIT Press), 1995.
- : «Building Verb Meaning». En: *The Projection of Arguments: Lexical and Compositional Factors*, pp. 96–134. CSLI Publications, 1998.
- MARIMON, MONTSERRAT: «The Spanish DELPH-IN Grammar». *Languages Resources and Evaluation*, 2013, **47**.
<http://www.delph-in.net/wiki/index.php/Grammars>
- MENDIKOETXEA, AMAIA: «Construcciones con se: medias, pasivas e impersonales». En: *Gramática descriptiva de la lengua española*, volumen II, pp. 1631–1722. Espasa, 1999.
- MEURERS, W. DETMAR y MINNER, GUIDO: «A Computational Treatment of Lexical Rules in HPSG as Covariation in Lexical Entries». *Computational Linguistics*, 1997, **23**, pp. 543–568.
- MONACHESI, PAOLA: «Decomposing Italian Clitics». En: *Romance in HPSG*, pp. 305–357. CSLI Publications, 1998.
- MORENO SANDOVAL, ANTONIO: *Lingüística computacional*. Editorial Síntesis, 1998.
- : *Gramáticas de unificación y rasgos*. Antonio Machado Libros, 2001.
- MÜLLER, STEFAN: «The Babel System: A Parser for an HPSG Fragment of German», 1996. [Http://www.dfki.de/stefan/Pub/babel.html](http://www.dfki.de/stefan/Pub/babel.html).
- : «Continuous or Discontinuous Constituents? A Comparison between Syntactic Analyses for Constituents Order and Their Processing System». *Research on Language and Computation*, 2004, **2**, pp. 209–257.

- PERLMUTTER, D.: «Impersonal passives and the Unaccusative Hypothesis». En: *Proceedings of the Fourth Annual Meeting of the Berkeley Linguistics Society*, pp. 157–189. Berkeley Linguistics Society, 1978.
- PETERS, P. STANLEY y RITCHIE, R. W.: «On the Generative Power of Transformational Grammar». *Information Science*, 1973, **6**, pp. 49–83.
- PINEDA, LUIS y MEZA, IVÁN: «The Spanish Pronominal Clitic System». *Procesamiento del lenguaje natural*, 2005, **34**, pp. 67–104.
- POLLARD, CARL y SAG, IVAN A.: *Information-Based Syntax and Semantics*. The University of Chicago Press, 1987.
- : *Head-Driven Phrase Structure Grammar*. The University of Chicago Press, 1994.
- POLLARD, CARL J.; KASPER, ROBERT T. y LEVINE, ROBERT D.: «Studies in Constituent Ordering: Toward a Theory of Linearization in HPSG», 1993. Grant Proposal to the National Science Foundation, Ohio State University.
- POST, E.: «Finite combinatory processes: Formulation i». *J. of Symbolic Logic*, 1936, **1**, pp. 103–105.
- : «A variant of recursively unsolvable problem». *Bull AMS*, 1946, **52**, pp. 264–268.
- RAY, ERIK T.: *Learning XML*. O'Reilly Media, 2001.
- REAPE, MIKE: *A Formal Theory of Word Order: A Case Study in West Germany*. Tesis doctoral, University of Edinburgh, 1992.
- RUSSELL, BERTRAND y WHITEHEAD, ALFRED N.: *Principia Mathematica*. Cambridge University Press, 1992.
- SAG, IVAN A.; WASOW, THOMAS y BENDER, EMILY: *Syntactic Theory: a Formal Introduction*. CSLI Publications, 2002.

- SELLS, PETER: *Lectures on Contemporary Syntactic Theories*. CSLI Publications, 1985.
- SERRANO, SEBASTIÁN: *Elementos de lingüística matemática*. Editorial Anagrama, 1975.
- SHIEBER, STUART M.: *An Introduction to Unification Based Approaches to Grammar*. CSLI Publications, 1986.
- SÁNCHEZ LEÓN, FERNANDO: «Gramáticas y Lenguajes Formales», 2006. Departamento de Lingüística Computacional de la Real Academia Española.
- TORREGO, ESTHER: «Unergative-unaccusative alternations in Spanish». *MIT Working Papers in Linguistics*, 1989, **10**, pp. 253–272.
- TSURUOKA, YOSHIMASA; MIYAO, YUSUKE y TSUJII, JUN'ICHI: «Towards efficient probabilistic HPSG parsing: integrating semantic and syntactic preference to guide the parsing», 2004.
- TURING, A.: «On computable numbers with an application to entscheidungsproblem». *Proc. London Math. Society*, 1936, **42(2)**, pp. 230–265.
- WALL, LARRY y SCHWARTZ, RANDAL L.: *Programming Perl*. O'Reilly Media, 1991.
- ZUBIZARRETA, M^A LUISA: *Prosody, Focus and Word Order*. The MIT Press, Cambridge, Massachusetts, 1998.
- : «Las funciones informativas: tema y foco». En: Ignacio Bosque y Violeta Demonte (Eds.), *Gramática descriptiva de la lengua española*, volumen III, pp. 4217–4242. Espasa, 1999.