Jugando con el Tiempo

Semántica de Pruebas
para
Algebras de Procesos Temporizadas

Memoria presentada para la obtención del título de doctor por:

Luis Fernando Llana Díaz

Trabajo dirigido por el doctor: **David de Frutos Escrig**

a Marisa

Agradecimientos

Empezaré agradeciendo a David la dirección del presente trabajo, así como el tiempo dedicado a su revisión y correción, fruto del cual han sido sus siempre bien recibidas sugerencias, que han hecho del mismo gran parte de lo que es en estos momentos.

Mi agradecimiento también a mis queridos compañeros del Departamento de Informática y Automática de la Universidad Complutense de Madrid por el apoyo moral prestado en todo momento. Quisiera dar gracias especialmente a Yolanda, *aventurera de la Cuarta Dimensión*, por sus útiles comentarios durante el desarrollo del presente trabajo.

Quisiera agradecer a mis amigos Javi y Pedro y a mis compañeros (además de amigos) Narciso y Cristóbal por el soporte *gráfico* aportado, gracias a ellos el presente trabajo ha quedado algo más *bonito*.

No puedo olvidarme en este momento de todos mis amigos; de mis padres y toda mi familia, por los ánimos y apoyos que me han mostrado en todo momento.

Por último muchas y sinceras gracias a la persona que quizás más ha sufrido la elaboración del presente trabajo: los nervios, los malos humores, las depresiones, los agobios, ... y a pesar de ello sigue estando a mi lado.

No puedo dejar de dar un agradecimiento *muy especial*, y por este orden, a las siguientes instituciones: Ejército Español, Ministerio de Defensa, Ministerio de Justicia, Secretaría de Estado de Universidades e Investigación y Real Academia Española. Las cuales me *liberaron* unos meses de la tediosa tarea de elaborar el presente trabajo, y consiguieron que el mismo pudiera ser postergado en el *tiempo*.

Indice

Capítulo 1

Introducción

1.1 ¿De qué va tu tesis?

Esta es la pregunta que me hacen la mayoría de mis familiares y amigos; a alguno de ellos le gustaría leerse parte de este trabajo pero, desgraciadamente, el contenido del mismo no está al alcance de una persona ajena al mundo científico. Intentaré dar una explicación lo suficientemente sencilla para que por lo menos puedan entender el contenido de este trabajo. La respuesta exacta a la pregunta de arriba sería: estudiaremos una semántica de pruebas para un álgebra de procesos temporizada en la que las acciones ocultas son urgentes. Probablemente, la mayoría de mis allegados, ante la respuesta en ruso se habrían enterado de lo mismo, así que intentaré explicar lo que significa con palabras llanas cada parte de dicha respuesta.

En primer lugar veamos qué es eso de un *proceso*, en particular un proceso concurrente. Fijémonos en algo que se ha convertido bastante en habitual en nuestra vida: ir a un cajero automático para sacar dinero. Cuando vamos a un cajero automático, lo primero que hacemos es introducir la tarjeta de crédito, a continuación el cajero nos pide que introduzcamos, a través del teclado dispuesto a tal fin, nuestro número secreto. Nosotros (si lo recordamos), obedientes lo introducimos. Si la operación anterior ha tenido éxito, el cajero nos presenta una serie de opciones entre las que está la de sacar dinero. Después de haber seleccionado esta opción nos pide que introduzcamos la cantidad que deseamos. Introducimos la cantidad, y al cabo de un rato (y si tenemos fondos suficientes y dispone de efectivo) el cajero nos da el dinero. ¿Qué ha ocurrido durante intervalo de tiempo? El cajero se ha puesto en comunicación con la computadora que tiene registrados nuestros datos, le ha preguntado si tenía fondos suficientes, etc.... En este ejemplo podemos observar no menos de tres procesos actuando de un modo concurrente: la persona que va a sacar el

-Buenos

dinero, el cajero y la computadora que tiene los datos. Estos tres procesos concurrentes han interactuado entre ellos intercambiándose información: el número secreto (entre la persona y el cajero) y la confirmación de fondos suficientes (entre el cajero y la computadora).

Por su parte, en álgebra, en particular un álgebra de procesos, es un lenguaje matemático con el que se pueden representar los procesos concurrentes de una manera simbólica. Las comunicaciones entre los procesos se representan mediante lo que llamamos *acciones*. Tales acciones pueden ser visibles o invisibles dependiendo del observador de que se trate. Por ejemplo, si hay una persona en el exterior esperando a que acabemos, ésta no puede ver el número secreto (¡más vale!) ni la cantidad de dinero. Para dicha persona esas comunicaciones son ocultas; sin embargo para nosotros que estamos sacando dinero, esas acciones son visibles. En principio, un álgebra de procesos es un ente totalmente sintáctico; permite construir *frases* por medio de un alfabeto y unas reglas sintácticas muy concretas. Dichas reglas sintácticas son esencialmente del mismo estilo que las reglas con las que construimos las oraciones del lenguaje natural, aunque (curiosamente) mucho más sencillas. Si nos quedamos puramente en la sintaxis, el lenguaje que utilizamos todos los días no sirve para nada. Ahora bien, cada frase que utilizamos tiene (al menos) un significado: si tenemos dos frases distintas podemos saber si quieren decir lo mismo, si tienen el mismo significado. Lo mismo hacemos con nuestro lenguaje matemático: dotamos a cada término construido con nuestras reglas de un significado concreto. La semántica es una función que asocia a cada término sintáctico un significado, que indica las acciones que puede realizar un proceso en cada momento. Uno de los tipos de semántica que se utiliza habitualmente para las álgebras de procesos es el que utilizaremos especialmente a lo largo del trabajo: la semántica de pruebas. En grandes líneas, esta semántica dice que dos procesos son iguales, si se comportan de la misma manera cuando interactúan con cualquier otro entorno que no es otra cosa que otro proceso.

Las álgebras de procesos, y sus semánticas, han sido ampliamente estudiadas desde los años 80. Los ejemplos más típicos los constituyen el modelo *CCS* de R. Milner [Mil89] y *CSP* de C. A. R. Hoare [Hoa85]. A lo largo de los años se han ido descubriendo algunas carencias de las álgebras de procesos originales. Una de ellas es que no permiten representar de manera adecuada el tiempo, entendido en su faceta cuantitativa. Para algunos sistemas no es suficiente saber que una comunicación se realizará, si no que es necesario saber cuándo tal cosa sucederá. Para suplir esta carencia surgen las álgebras de procesos temporizadas. A lo largo de estos últimos años han surgido diversas propuestas para definir tales álgebras. El presente trabajo supone un paso más en esta dirección. Las principales características del modelo propuesto son las siguientes:

- Está basado una semántica de pruebas.

- Las acciones ocultas son urgentes, lo que significa que si una de estas acciones se puede realizar en un determinado instante, no puede pasar más tiempo a menos que la acción se realice, o deje de poder realizarse debido a la ejecución de otra acción posible.

Espero que con esta breve introducción, una persona no iniciada pueda tener una ligera idea del contenido del presente trabajo. Las páginas que vienen a continuación están cargadas de formalismos matemáticos, de manera que no serán comprensibles para el público general, así que aprovecho la ocasión para agradecer a todos ellos el interés que han puesto en mi trabajo. Para los valientes que prosigan con la lectura, haremos una breve descripción de lo que se van a encontrar a partir de este momento. En primer lugar, en este mismo capítulo repasaremos algunos de los resultados básicos sobre álgebras de procesos. A continuación, en el capítulo 2 introduciremos conceptos y definiciones básicas que usaremos a lo largo del trabajo. En el capítulo 3 presentaremos un álgebra de procesos básica pero suficientemente expresiva, la dotaremos de una semántica de pruebas y, debido a la complejidad que suelen tener este tipo de semánticas, daremos la obligada caracterización operacional. Después, en el capítulo 4 daremos una semántica denotacional de nuestro lenguaje que será completamente abstracta con respecto a la semántica de pruebas. Esta semántica denotacional será utilizada en el capítulo 5 para generar un conjunto de reglas y axiomas que serán correctos y completos con respecto a la semántica de pruebas. Una vez finalizado el estudio del lenguaje básico, veremos cómo extender nuestro lenguaje con otros operadores característicos de un álgebra de procesos temporizada, esto lo haremos en el capítulo 5. Con ello habremos completado el estudio de un álgebra de procesos tipo *CSP*; en el capítulo 7 indicaremos cómo podrían adaptarse los resultados obtenidos a un álgebra de procesos tipo *CCS*.

1.2 Sintaxis

Empezamos introduciendo el mecanismo que nos va a permitir definir la sintaxis de un álgebra de términos. En particular un álgebra de procesos será un álgebra de términos. Para poder definir un álgebra de términos necesitamos un conjunto de operadores sintácticos Σ, cada uno de ellos con una aridad $n \geq 0$. Consideramos además un conjunto de identificadores (o variables) de proceso X. Entonces tenemos

---

días

Definición 1.2.1 El conjunto $\mathsf{Rec}(\Sigma)$ de términos recursivos sobre Σ viene dado por la siguiente expresión B.N.F.:

$$P ::= op(P_1, \ldots, P_n) \,|\, x |\, \mathtt{REC}\, x.P$$

donde $op \in \Sigma$ tiene aridad n y $x \in X$ es una variable. La construcción $\mathtt{REC}\, x.P$ representa la recursión. Denotaremos por $\mathsf{FRec}(\Sigma)$ el conjunto de términos finitos, que son aquéllos en los que no interviene la recursión.

□

Asociada a la recursión aparecen las nociones de variable libre y ligada.

Definición 1.2.2 Sea $P \in \mathsf{Rec}(\Sigma)$ y $x \in X$:

- x tiene una aparición libre en P, en los siguientes casos
  - $P = x$,
  - $P = op(P_1, \ldots, P_m)$ y x aparece libre en algún P_i,
  - $P = \mathtt{REC}\, y.P_1$, y x aparece libre en P_1 con $x \neq y$.

  Básicamente una aparición de x en P está libre si no está bajo el ámbito de un operador de recursión correspondiente a dicha variable. Denotaremos por $\mathsf{Var}(P)$ el conjunto de variables que aparecen libres en P.
- Una aparición de x es ligada si no es libre.
- Un término es cerrado si no tiene apariciones libres de variables. Denotaremos por $\mathsf{CRec}(\Sigma)$ el conjunto de términos cerrados y por $\mathsf{FCRec}(\Sigma)$ el conjunto de términos finitos cerrados.

□

Un proceso con variables libres nos permite la definición de nuevos procesos sustituyendo alguna (o todas) sus variables libres por otros procesos.

Definición 1.2.3 Siendo $P,\ Q \in \mathsf{Rec}(\Sigma)$, $P[Q/x]$ representa la sustitución de toda aparición libre de la variable x por el término Q:

$$P[Q/x] = \begin{cases} Q & \text{si } P = x \\ y & \text{si } P = y \text{ y } \ x \neq y \\ op(P_1[Q/x], \ldots, P_n[Q/x]) & \text{si } P = op(P_1, \ldots P_n) \\ \mathtt{REC}\, x.P_1 & \text{si } P = \mathtt{REC}\, x.P_1 \\ \mathtt{REC}\, u.P_1[u/y][Q/x] & \text{si } P = \mathtt{REC}\, y.P_1 \text{ y } u \text{ es una variable nueva} \end{cases}$$

$\square$

En las álgebras de procesos tenemos una serie de operadores que aparecen de una forma u otra en todas ellas:

- Operadores de aridad cero, como pueden ser un operador `STOP` que indica el bloqueo, y un operador `DIV` que representa la divergencia.
- Se considera un conjunto de eventos $\mathcal{E}$ que el proceso puede ejecutar lo cual se logra a través del operador de prefijo de aridad 1. Nosotros el prefijo lo denotaremos con a; (evento a seguido de punto y coma).
- Operadores de elección, que clásicamente se dividen en tres grupos:
  - La elección externa $\Box$, en la que los factores que intervienen en la elección son exclusivamente externos.
  - La elección interna $\sqcap$, en la que es el proceso, internamente, quien elige por qué opción inclinarse.
  - La suma $+$ que suele corresponder con una elección mixta de las dos anteriores, más exactamente con una generalización conjunta de ambas.

  Todos estos operadores son, en principio, binarios, pero usualmente se tiene que son asociativos y conmutativos, por lo que se pueden generalizar fácilmente a operadores de aridad arbitraria.
- Operadores derivados, como pueden ser el operador de paralelo, ocultamiento,...

1.3 Semántica

La semántica de un proceso o programa concurrente, como su propio nombre indica, nos define su significado. Hasta aquí todo es fácil, lo realmente difícil se centra en saber qué se entiende por el significado de un proceso. Para programas secuenciales todo el mundo está de acuerdo en que ese significado viene dado por la relación entre la "entrada" y la "salida" que genera el programa. Sin embargo, esta relación ya no es válida para programas concurrentes, ya que en éstos no es el resultado final el que interesa. Es más, muchas veces los programas concurrentes son por naturaleza no terminantes, y si terminan es debido a una situación de error (tal es el caso, por ejemplo, de un sistema operativo). Para dar el significado de un proceso tenemos diversos mecanismos, algunos de los cuales enumeraremos a continuación.

-dijo

1.3.1 Semántica Operacional

Una primera aproximación para definir la semántica de los procesos consiste en caracterizar de algún modo los eventos que en cada momento un proceso puede llevar a cabo. Un evento puede verse como una comunicación entre el proceso y el entorno en el que el mismo se encuentra (en último término el usuario que lo utiliza). Un modo de formalizar la ejecución de dichos eventos la constituyen los sistemas de transiciones etiquetados.

Definición 1.3.1 Un sistema de transiciones etiquetado extendido es una tupla

$$\langle S, s_0, L, \longrightarrow, \rightarrowtail \rangle$$

donde S es un conjunto de estados, $s_0 \in S$, L es un alfabeto de etiquetas, $\longrightarrow$ y $\rightarrowtail$ son las transiciones del sistema:

- $\longrightarrow \subseteq S \times L \times S$. Si $(s, l, t) \in \longrightarrow$ lo denotaremos por $s \xrightarrow{l} t$.
- $\rightarrowtail \subseteq S \times S$. Si $(s, t) \in \rightarrowtail$ lo denotaremos por $s \rightarrowtail t$.

$\square$

En nuestro caso tenemos que S será el conjunto de términos cerrados $\mathsf{CRec}(\Sigma)$ y L el conjunto de evento $\mathcal{E}$. Si tenemos $P \in \mathsf{CRec}(\Sigma)$ su semántica operacional viene dada por el sistema de transiciones asociado:

$$\langle \mathsf{CRec}(\Sigma), P, \mathcal{E}, \longrightarrow, \rightarrowtail \rangle$$

Normalmente la semántica operacional de un procesos es muy intuitiva, indicando lo que puede hacer en cada momento. Pero por su falta de abstracción resulta poco manejable, pues a partir de la misma resulta difícil determinar cuándo dos procesos deben considerarse equivalentes desde un punto de vista razonable. Un trabajo en detalle sobre semánticas operacionales lo podemos encontrar en [Hen90].

1.3.2 Semántica de Pruebas

Quizás esta sea la semántica más intuitiva para un álgebra de procesos. La idea que hay detrás es que dos procesos son iguales si interactúan de la misma manera ante cada entorno posible. Un amplio estudio de la semántica de pruebas para un álgebra de procesos tradicional lo podemos encontrar en [Hen88]. Para dar este tipo de semántica se introduce un nuevo tipo de procesos que llamaremos experimento, prueba o test. La misión de un experimento es comprobar el comportamiento de un proceso. Se hace que interactúen el experimento (como si fuera un proceso cualquiera) y el proceso, y se observa la(s) computación(es) que tiene(n) lugar. Esas computaciones podrán tener éxito o fracaso. Entonces podremos definir los siguientes conceptos:

- Un proceso P pasa una prueba T en sentido *must* si toda computación de P y T tiene éxito.
- Un proceso P pasa una prueba T en sentido *may* si existe una computación de P y T que tiene éxito.

Se dirá entonces que dos procesos son equivalentes en sentido *must* si pasan las mismas pruebas en sentido *must*. Y análogamente diremos que dos procesos son equivalentes en sentido *may* si pasan las mismas pruebas en sentido *may*. En este trabajo nos centraremos en el estudio de la semántica *must* para el álgebra que presentamos y es por ello que contra lo que es usual la hemos mencionado en primer lugar. Utilizaremos la siguiente notación:

- $P \underset{\sim}{\sqsubseteq} Q$ indica que toda prueba que pasa (en sentido *must*) el proceso P, es también pasada por el proceso Q.
- $P \underset{\sim}{=} Q$ indica que P y Q pasan las mismas pruebas, también en sentido *must*.

1.3.3 Semántica Denotacional

Bajo una semántica denotacional se asocia a cada término sintáctico un valor en un cierto *dominio* semántico $\mathcal{D}$. Ello se hace de forma composicional, lo que representa una característica fundamental de tales semánticas. Un *dominio* es un conjunto de objetos matemáticos dotado de un orden $\leq_{\mathcal{D}}$ sobre sus elementos, es decir, una relación binaria que sea reflexiva, transitiva y antisimétrica, con propiedades adecuadas. Un estudio amplio sobre semántica denotacional en general se encuentra en [Sch88].

La dificultad principal a la hora de definir la semántica es la presencia de términos recursivos, por no adecuarse los mismos al mero tratamiento composicional. Por ello necesitamos que dicha estructura sea adecuada, para dotarles de significado de una manera sistemática. Ello se logra por medio de la técnica de puntos fijos. Para ello necesitamos que los dominios *órdenes parciales completos* y que las funciones que traducen el significado de cada operación sintáctica, sean continuas; todo lo cual se formaliza a continuación.

Definición 1.3.2 Siendo $(\mathcal{D}, \leq_{\mathcal{D}})$ un conjunto ordenado, una cadena sobre el mismo es una sucesión $X = \{a_n \in \mathcal{D} \mid n \in \mathbb{N}\} \subseteq \mathcal{D}$ que verifica $a_i \leq_{\mathcal{D}} a_{i+1}$.

$\square$

el

Definición 1.3.3 Un conjunto ordenado $(\mathcal{D}, \leq_{\mathcal{D}})$ es un *orden parcial completo* (*cpo*) sii verifica:

1. Tiene un elemento mínimo $\perp_{\mathcal{D}}$, esto es $\forall x \in \mathcal{D}: \perp_{\mathcal{D}} \leq_{\mathcal{D}} x$.

2. Toda cadena X de $\mathcal{D}$ tiene límite (cota superior mínima) $\mathsf{lub}(X)$ en $\mathcal{D}$; es decir:

$$\begin{array}{ll} \forall x \in X: & x \leq_{\mathcal{D}} \mathsf{lub}(X) \\ \forall d \in \mathcal{D}: & \Big((\forall x \in X: x \leq_{\mathcal{D}} d) \Rightarrow d \leq_{\mathcal{D}} \mathsf{lub}(X)\Big) \end{array}$$

□

Proposición 1.3.4 Si $\mathcal{D}$ es un *cpo* , $\mathcal{D}^n = \underbrace{\mathcal{D} \times \cdots \times \mathcal{D}}_{n}$ con el orden por componentes dado por $(d_1, \ldots, d_n) \leq_{\mathcal{D}^n} (d'_1, \ldots, d'_n)$ sii $\forall i \in \{1, \ldots n\}\ d_i \leq d'_i$, es también un *cpo* .

Definición 1.3.5 Sean $\mathcal{D}$ y $\mathcal{D}'$ dos *cpos* y $\phi : \mathcal{D} \longmapsto \mathcal{D}'$; decimos que

1. ϕ es **estricta** si conserva el elemento mínimo: $\phi(\perp_{\mathcal{D}}) = \perp_{\mathcal{D}'}$,

2. ϕ es **monótona** si conserva el orden parcial:

$$\forall d_1, d_2 \in \mathcal{D}: \quad \Big(d_1 \leq_{\mathcal{D}} d_2 \Rightarrow \phi(d_1) \leq_{\mathcal{D}'} \phi(d_1)\Big)$$

3. ϕ es **continua** si conserva los límites de las cadenas:

$$\forall X \subseteq \mathcal{D},\ X \text{ cadena}: \quad \phi(\mathsf{lub}(X)) = \mathsf{lub}(\phi(X))$$

□

Proposición 1.3.6 $\phi : \mathcal{D}^n \longmapsto \mathcal{D}$, es continua sii lo es respecto de cada componente.

Proposición 1.3.7 $\phi : \mathcal{D} \longmapsto \mathcal{D}$, es continua sii

1. ϕ es monótona y,

2. Para toda cadena $X \subseteq \mathcal{D}$ se verifica $\phi(\mathsf{lub}(X)) \leq_{\mathcal{D}'} \mathsf{lub}(\phi(X))$.

Teorema 1.3.8 (Punto fijo, Knaster-Tarski) Toda $\phi : \mathcal{D} \longmapsto \mathcal{D}$ monótona tiene un mínimo punto fijo ($\mathsf{fix}(\phi)$) en $\mathcal{D}$. Si además ϕ es continua, se verifica

$$\mathsf{fix}(\phi) = \mathsf{lub}\{\phi^n(\perp_{\mathcal{D}}) \mid n \in \mathbb{N}\}$$

donde para cada $d \in \mathcal{D}$, $\phi^0(d) = d$ y $\phi^{n+1}(d) = \phi(\phi^n(d))$.

□

Con lo visto en esta sección estamos en condiciones de definir lo que es un *modelo denotacional* y de ver cómo se obtiene una semántica denotacional basándose en ellos.

Definición 1.3.9 Un modelo denotacional $\mathcal{M}$ para $\mathsf{CRec}(\Sigma)$ consiste en un *cpo* $\mathcal{D}_{\mathcal{M}}$ y un conjunto $\{\mathcal{M}[\![op]\!] \mid op \in \Sigma\}$ de operadores continuos $\mathcal{M}[\![op]\!] : \mathcal{D}^n_{\mathcal{M}} \longmapsto \mathcal{D}_{\mathcal{M}}$.

□

Definición 1.3.10 Siendo $Ent = \{\rho \mid \rho : \mathsf{Var} \longmapsto \mathcal{D}_M\}$ el conjunto de entornos (asignaciones de valores a variables), la semántica denotacional sobre $\mathsf{Rec}(\Sigma)$ inducida por $\mathcal{M}$ viene dada por

$$\mathcal{M}[\![\cdot]\!] : \mathsf{Rec}(\Sigma) \longmapsto (Ent \longmapsto \mathcal{D}_{\mathcal{M}})$$

definida mediante

1. $\mathcal{M}[\![op(P_1, \ldots, P_n)]\!]_\rho = \mathcal{M}[\![op]\!](\mathcal{M}[\![P_1]\!]_\rho, \ldots, \mathcal{M}[\![P_n]\!]_\rho)$.

2. $\mathcal{M}[\![x]\!]_\rho = \rho(x)$.

3. $\mathcal{M}[\![\mathtt{REC}\, x.P]\!]_\rho = \mathsf{fix}(\lambda d.\ \mathcal{M}[\![P]\!]_\rho[d/x])$.

donde $\rho[d/x]$ es idéntico a ρ salvo sobre x en donde vale d.

□

Hecho 1.3.11 *Para términos cerrados $P \in \mathsf{CRec}(\Sigma)$, el valor $\mathcal{M}[\![P]\!]_\rho$ es independiente de ρ.*

□

Por ello, en tales casos podemos prescindir de las asignaciones de variables, y escribir simplemente $\mathcal{M}[\![P]\!]$.

En particular tendremos una constante $\mathtt{DIV} \in \Sigma$ de aridad cero que representa el elemento mínimo del dominio, es decir:

$$\mathcal{M}[\![\mathtt{DIV}]\!] = \perp_{\mathcal{D}}$$

El teorema de Knaster-Tarski, nos indica que el mínimo punto fijo de una función continua viene dado por el límite de la cadena obtenida al aplicar reiteradamente la función al elemento mínimo del dominio. Aplicando esto a nuestro caso particular, tenemos que para calcular la semántica denotacional de un procesos recursivo $\mathtt{REC}\, x.P$ consideramos la secuencia de procesos

$$P_0 = \mathtt{DIV},\ P_1 = P[P_0/x],\ P_2 = P[P_1/x], \cdots$$

---

principito.

de modo que la semántica del proceso vendrá dada por el límite de la semántica de los procesos de la cadena. Siguiendo dicha idea definimos las aproximaciones sintácticas de un proceso.

Definición 1.3.12 Para cada término $P \in \mathsf{Rec}(\Sigma)$ definiremos las aproximaciones finitas, sea $k \in \mathbb{N}$:

$$\mathsf{ap}(P,k) = \begin{cases} \mathtt{DIV} & \text{si } k = 0 \\ x & \text{si } P = x \in \mathsf{Var} \\ op(\mathsf{ap}(P_1,k),\ldots,\mathsf{ap}(P_n,k)) & \text{si } P = op(P_1,\ldots,P_n) \text{ y } k > 0 \\ \mathsf{ap}(P_1,k-1)[\mathsf{ap}(P,k)/x] & \text{si } P = \mathtt{REC}\,x.P_1 \text{ y } k > 0 \end{cases}$$

□

Obviamente si $P \in \mathsf{CRec}(\Sigma)$ entonces $\mathsf{ap}(P,k) \in \mathsf{FCRec}(\Sigma)$. Además tenemos que el conjunto $\{\mathcal{M}[\![\mathsf{ap}(P,k)]\!] \mid k \in \mathbb{N}\}$ forma una cadena, por lo que podemos hablar de su mínima cota superior, verificándose

$$\mathcal{M}[\![P]\!] = \mathsf{lub}\{\mathcal{M}[\![\mathsf{ap}(P,k)]\!] \mid k \in \mathbb{N}\}$$

Buenas introducciones más profunda a la teoría de dominios pueden encontrarse por ejemplo en [SG90] y [Win93].

A lo largo del trabajo utilizaremos un dominio $\mathcal{D}$ sobre el que sólo tenemos definido un preorden $\leq_{\mathcal{D}}$ (no tiene la propiedad antisimétrica). Sin embargo, todo lo dicho anteriormente es igualmente válido; únicamente hay que tener en cuenta que la igualdad siempre ha de entenderse módulo la relación de equivalencia inducida por $\leq_{\mathcal{D}}$ dada por

$$x \equiv y \iff x \leq_{\mathcal{D}} y \ \wedge \ y \leq_{\mathcal{D}} x$$

La semántica denotacional es más manejable, desde un punto de vista matemático, que la semántica de pruebas. Es más fácil decir si dos procesos son equivalentes trabajando con una semántica denotacional que con una semántica de pruebas. Sin embargo tiene el inconveniente de que es menos intuitiva. Lo que haremos nosotros es definir primero una semántica de pruebas, tras ello una semántica denotacional, comprobando que ambas son equivalentes. Diremos que la semántica denotacional es totalmente abstracta con respecto a una relación de orden $\leq$ (o un preorden) sobre el álgebra de procesos, si se cumple

$$P \leq Q \iff \mathcal{M}[\![P]\!] \leq_{\mathcal{D}} \mathcal{M}[\![Q]\!]$$

Entonces, demostrar que la semántica de pruebas es equivalente con la semántica denotacional, es equivalente a probar que esta última es totalmente abstracta con respecto a la relación inducida por la semántica de pruebas.

1.3.4 Semántica Algebraica

Una vez introducidas los distintos tipos de semánticas, observamos que convendría disponer de algún mecanismo algebraico, esto es, que maneje exclusivamente términos sintácticos, que permita deducir igualdades (o desigualdades) entre procesos. Para ello se presentan una serie de axiomas y reglas que permitan deducir de una manera puramente sintáctica dichas relaciones entre los procesos. Un sistema de ecuaciones diremos que es *correcto* con respecto a una semántica si toda relación deducible por aplicación del mismo es cierta en la semántica correspondiente. Diremos que el sistema es *completo* con respecto a la misma relación si toda relación semántica puede ser deducida mediante las ecuaciones. Un buena referencia para todo lo que se refiere a álgebras ecuacionales es [MG85].

Eventualmente podríamos cambiar el orden en el que desarrollamos el proceso, comenzando por dar los axiomas que nos parezcan que ha de cumplir la noción de equivalencia (u orden) entre procesos. Es por ello que también denominamos semántica algebraica a la definición axiomática de dicha equivalencia (u orden).

Esta página está intencionadamente en blanco.

Capítulo 2

Preliminares

En este capítulo introduciremos una serie de conceptos y notaciones que se utilizarán a lo largo del trabajo. El primer aspecto que vamos a decidir es el dominio temporal en el que nos vamos a mover.

2.1 Dominio de Tiempo

Existen en la literatura dos tipos diferenciados de dominios temporales: continuos y discretos. A primera vista un dominio de tiempo continuo parece más razonable que un dominio de tiempo discreto. Sin embargo la introducción de un dominio de tiempo continuo resulta en general complicada, y en concreto cuando nosotros lo hemos intentado hacer en el presente trabajo han surgido una gran cantidad de inconvenientes, de hecho en muchas partes del trabajo (especialmente en la parte de la semántica denotacional), veremos que para que las cosas sean ciertas es necesario suponer que estamos usando un dominio de tiempo discreto. Por otro lado no parece claro que asumir un dominio de tiempo continuo pueda aportar nada especialmente interesante: por muy veloces que sean las computadoras (al menos aquellas de las que tenemos conocimiento), las operaciones ni son instantáneas, ni su velocidad tiende a cero. Por otra parte todas las computadoras (conocidas) están gobernadas por un reloj discreto, y la duración de las acciones es siempre múltiplo del correspondiente ciclo del reloj.

Por todo ello hemos decidido optar por un dominio de tiempo discreto, infinito y con origen. Una de las ventajas que podría representar tener un dominio de tiempo continuo es que se *podrían escribir* fracciones de la unidad. Para permitirlo, lo que haremos es suponer una mínima fracción de tiempo σ, que se puede ver como la duración de un ciclo de reloj, por lo que podemos suponer que cualquier medida temporal es un múltiplo entero de esta

---

días

cantidad. En particular, podremos suponer que la unidad es un cierto múltiplo de dicha fracción de tiempo. Así tenemos que el dominio de tiempo es discreto, pero sin perder por ello la impresión de estar trabajando en un dominio de tiempo continuo, puesto que la fracción de tiempo σ podemos escogerla tan pequeña como precisemos. Al dominio de tiempo lo denotaremos por $\mathcal{T}$, y pudiéndose asumir que incluye a los números naturales, es decir, $\mathbb{N} \subseteq \mathcal{T}$.

2.2 Transiciones

La semántica operacional de un proceso expresa *lo que un proceso puede hacer* mediante transiciones. En las álgebras sin tiempo las transiciones tienen la forma

$$P \xrightarrow{a} Q$$

donde a es la acción que se ha ejecutado. A lo largo de este trabajo, al conjunto de acciones que un proceso podrá realizar lo llamaremos Act. Conforme avancemos, veremos la necesidad de que el conjunto Act sea finito. Para poder representar el ocultamiento de una acción introduciremos lo que llamaremos la acción oculta $\tau \notin Act$, considerándose el conjunto de eventos $\mathcal{E} = Act \cup \{\tau\}$.

Si estamos considerando un álgebra de procesos temporizada tenemos que introducir el tiempo de alguna forma. En este trabajo vamos a considerar que las acciones que se ejecutan son instantáneas, su ejecución no consume tiempo; el tiempo indicará el instante en el que se ejecutan las acciones. Típicamente hay dos formas de introducir el tiempo en la semántica operacional:

- Tener dos tipos de transiciones: transiciones de acciones

  $$P \xrightarrow{a} Q$$

  indicando que el proceso P ha ejecutado instantáneamente la acción a y se ha transformado en el proceso P; y transiciones de tiempo

  $$P \overset{t}{\rightsquigarrow} Q$$

  cuyo significado es que el proceso P ha dejado pasar t unidades de tiempo transformándose en el proceso Q. Un ejemplo de este tipo de transiciones lo podemos encontrar entre otros en [NS91, Yi91a].

- Tener un único tipo de transiciones

  $$P \xrightarrow{at} Q$$

cuyo significado sería que el proceso ha dejado pasar t unidades de tiempo para después ejecutar la acción a transformándose en Q; o de una manera más compacta P ejecuta la acción a en el instante t y se transforma en Q. Este tipo de transiciones los podemos encontrar [Sch95, QMdL94]

La primera forma obliga a tener dos tipos de transiciones: una para acciones y otra para tiempos. De la segunda sólo precisamos un único tipo de transiciones, si bien en ocasiones (transiciones en *interleaving* del operador paralelo) será preciso indicar de alguna manera el paso de tiempo sin que se haya ejecutado acción alguna. Las dos formas son , en cierto modo, equivalentes, pues

- Si en el primer caso se necesitan transiciones de tiempo del tipo

  $$P \overset{t}{\rightsquigarrow} Q$$

  en el segundo caso se hace necesario definir una función (o algún mecanismo parecido) que indique el paso del tiempo:

  $$\mathsf{Upd}(P, t) = Q$$

  Por otra parte, si en el primer caso se tiene una transición del tipo

  $$P \xrightarrow{a} Q$$

  en el segundo se tiene la transición correspondiente

  $$P \xrightarrow{a0} Q$$

- Recíprocamente, si en el segundo caso se tiene una transición

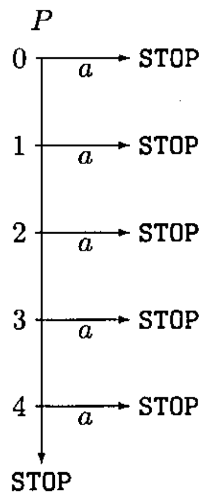
  $$P \xrightarrow{at} Q$$

  en el primer caso existirá un proceso P' tal que

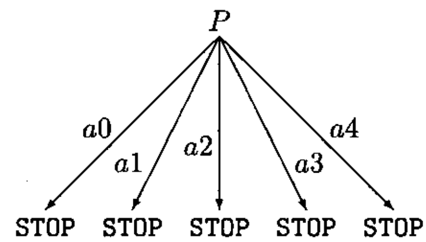
  $$P \overset{t}{\rightsquigarrow} P' \xrightarrow{a} Q$$

La diferencia entre las dos maneras de ver las transiciones se puede ver gráficamente con el proceso $P = a[0..4]\,;\texttt{STOP}$, un proceso que puede realizar la acción a en cualquier instante entre 0 y 4, como sigue:

-dijo



Transiciones con acciones y transiciones temporizadas



Transiciones con acciones temporizadas

Puesto que como queda vista, serían básicamente equivalentes, en principio daría lo mismo optar por una u otra. En este trabajo estamos interesados en dar una semántica basada en estados*, por lo que parece más cómodo trabajar con el segundo tipo de transiciones, pero los resultados habrían sido exactamente los mismos si hubiéramos escogido la primera opción. En cualquiera de las dos formas la ejecución de las acciones es instantánea, el tiempo indica el instante en el que se ejecutan las acciones. Existe también la posibilidad de introducir el tiempo en la duración de las acciones, como se hace en [OM91], la razón de no haber considerado las duraciones de las acciones es porque las álgebras de procesos que estamos interesados en estudiar, en concreto extensiones temporales de *LOTOS* [LOT88, dFLL$^+$95], no consideran esta opción. De todas formas la duración de las transiciones podría ser simulada dividiendo la ejecución de una acción en dos eventos: el de inicialización y el de finalización, de manera análoga a como se hace en [Hen92].

En ocasiones, en las álgebras de procesos, se utiliza también la acción oculta para representar el no determinismo. Por ejemplo, el comportamiento de elección interna se representa operacionalmente como

$$P \sqcap Q \xrightarrow{\tau} Q, \qquad P \sqcap Q \xrightarrow{\tau} P$$

También se utiliza dicha acción para representar el resultado de ocultar alguna acción visible. En el presente trabajo la semántica pretendida de la acción oculta τ será exclusi-

*Un estado contiene información acerca de las acciones que un proceso puede ejecutar, y el instante en el que puede hacerlo.

vamente esta segunda.

En relación con este tema, si analizamos de dónde viene el no determinismo, cuando no está provocado por la ocultación de alguna acción visible, nos encontramos que el mismo resultará de procesos del tipo

$$a\,;P\,\Box\,a\,;Q$$

Para poder representar dichos procesos de una forma canónica, introducimos el operador de elección interna $\sqcap$, con lo que el proceso anterior resulta ser equivalente al proceso

$$a\,;(P\sqcap Q)$$

Nos ha parecido conveniente ser capaces de distinguir entre este tipo de no determinismo, del no determinismo proveniente de las acciones ocultas. La razón para ello es doble:

- Si tenemos en cuenta que estamos en un álgebra de procesos con tiempo, entonces cuando se oculta una acción, ésta no se *ve*, pero sin embargo, se puede observar un efecto de la ejecución: vemos que ha pasado el tiempo.
- Si queremos modelar un operador de suma tipo *CCS*, como haremos en el capítulo 7, las acciones ocultas resolverían la elección, y esto está en contra con la idea de que el no determinismo interno se resuelve de forma autónoma, de manera que el proceso $(P \sqcap Q) + R$ debería evolucionar de manera no determinista hacia $P + R$ o hacia $Q + R$.

Por todo ello creemos conveniente introducir un tercer tipo de transiciones, que representarán la resolución del no determinismo:

$$P \rightarrowtail Q$$

expresando que el proceso P evoluciona de manera no determinista al proceso Q, sin consumir tiempo ni ejecutar acción alguna.

Resumiendo todo lo anterior, en este trabajo tendremos tres tipos transiciones:

- $P \xrightarrow{at} Q$ con $a \in Act$ y $t \in \mathcal{T}$.
- $P \xrightarrow{\tau t} Q$ con $t \in \mathcal{T}$.
- $P \rightarrowtail Q$.

Por comodidad será conveniente usar la notación de transiciones negadas, expresando que ninguna del tipo correspondiente es posible:

$$\begin{array}{lcl} P \not\rightarrowtail & \Longleftrightarrow & \neg\exists Q : P \rightarrowtail Q \\ P \overset{et}{\not\longrightarrow} & \Longleftrightarrow & \neg\exists Q : P \xrightarrow{et} Q. \end{array}$$

el

2.3 Urgencia

En nuestro trabajo presentamos un álgebra de procesos con urgencia. Esto significa que ciertas acciones son ejecutadas en cuanto están disponibles, por lo que los procesos no pueden quedarse parados sin hacer nada, pero dejando pasar el tiempo, si no hay razones poderosas para ello.

Ello es muy razonable para las acciones que un sistema puede ejecutar de una manera autónoma, pero en cuanto aparecen acciones que requieren la colaboración de un sistema externo (por ejemplo que requieren la participación humana en un sistema semiautomático) no parece exigible una respuesta inmediata del mismo.

Por ello la urgencia en este trabajo se referirá exclusivamente a la ejecución de transiciones internas, tanto a las correspondientes a acciones ocultas, como a las de resolución del no determinismo. De modo que la urgencia queda expresada por la siguiente condición

$$P \xrightarrow{et} Q \quad \Rightarrow \quad P \not\rightarrowtail \quad \wedge \quad \forall t' < t \; P \stackrel{\tau t'}{\not\longrightarrow}$$

2.4 Estados

En las secciones anteriores ya hemos introducido el conjunto de acciones *Act*, que ha de ser finito. A partir de este conjunto y el dominio de tiempo, construimos el conjunto de acciones temporizadas *TAct* que serán las acciones a las cuales se les añade una etiqueta de tiempo. Formalmente tenemos que $TAct = Act \times \mathcal{T}$, y si $a \in \mathcal{T}$ y $t \in \mathcal{T}$ diremos que $at \in TAct$.

Por último construimos los estados que son básicamente conjuntos de acciones temporizadas, a los que se les añade cierta información de indefinición (o divergencia). Un estado representa una posible configuración de un proceso elegida de un modo no determinista. En un estado se encuentra el conjunto de acciones que un proceso puede ejecutar y el tiempo en el que las mismas pueden ser ejecutadas: la presencia de at en un estado A de un proceso P indica que en ese estado P puede ejecutar la acción a en el instante t.

Además hemos dicho que los estados tienen una información de indefinición, para introducir dicha información de indefinición consideramos un elemento nuevo $\Omega \notin Act$ considerándose el conjunto $Act_\Omega = Act \cup \{\Omega\}$. Un estado será ahora un cierto conjunto $A \subseteq Act_\Omega \times \mathcal{T}$ (no necesariamente finito). Si Ωt está en un estado A de P, diremos que en ese estado el proceso está indefinido a partir del instante t (también podremos decir que entra en divergencia), es decir, no podremos asegurar absolutamente nada acerca de su comportamiento.

Para fijar conceptos, aunque todavía no hayamos concretado la sintaxis ni la semántica de los procesos, el siguiente ejemplo puede servir de ayuda:

Ejemplo 2.4.1 Consideremos el proceso

$$P = (a1\,;\mathtt{STOP}) \sqcap ((b2\,;\mathtt{STOP}) \,\square\, (\tau 3\,;\mathtt{DIV}))$$

Este proceso *puede elegir* de una manera no determinista entre los procesos

$$a1\,;\mathtt{STOP} \quad \text{y} \quad (b2\,;\mathtt{STOP}) \,\square\, (\tau 3\,;\mathtt{DIV})$$

Si *elige* el primer proceso, podrá ejecutar la acción a en tiempo 1; si *elige* el segundo, podrá ejecutar la acción b en tiempo 2. Además, en el segundo caso, si el proceso no ejecuta la acción b en tiempo 2, y espera 3 unidades de tiempo, entra en un estado indefinido (divergencia). Por ello tendremos que (inicialmente) el proceso P tiene dos estados $\{a1\}$ y $\{b2, \Omega 3\}$.

□

No todo subconjunto de $Act_\Omega \times \mathcal{T}$ será un estado válido. Al respecto tenemos la siguiente

Definición 2.4.2 Un estado A es un subconjunto de $Act_\Omega \times \mathcal{T}$ que verifica las siguientes propiedades:

- Existe como mucho un elemento $\Omega t \in A$; esto es:

$$\Omega t,\ \Omega t' \in A \ \Rightarrow\ t = t'$$

- Si $\Omega t \in A$, todas las acciones temporizadas de A corresponden a un instante menor

$$\Omega t,\ a t' \in A \ \Rightarrow\ t' < t$$

Denotaremos por $\mathcal{ST}$ el conjunto de estados.

□

A continuación introducimos una serie de operaciones sobre estados:

Definición 2.4.3 Siendo que A es un estado:

- Definimos el instante en el que el estado está indefinido por medio de:

$$\mathsf{nd}(A) = \begin{cases} t & \text{si } \Omega t \in A \\ \infty & \text{en otro caso} \end{cases}$$

---

vendedor.

- Si $t \in \mathcal{T}$ definimos los *desplazamientos en el tiempo* del estado A como sigue:

$$A+t=\{a(t+t') \mid at' \in A\}, \quad A-t=\begin{cases} \{a(t'-t) \mid at' \in A,\ t' \geq t\} & \text{si } \mathsf{nd}(A) \geq t \\ \text{indefinido} & \text{en otro caso} \end{cases}$$

  Obsérvese que al no admitirse tiempos negativos, tenemos que en general no será cierta la igualdad $(A-t)+t=A$. En cambio, sí será cierto que $(A+t)-t=A$.

- Definimos la restricción del estado A a las acciones antes de un determinado instante estricto y no estricto:

$$A \upharpoonright t=\{at' \mid at' \in A,\ t'<t\}, \qquad A \upharpoonright\!\!\upharpoonright t=\{at' \mid at' \in A,\ t' \leq t\}$$

- Si $B \subseteq Act$ es un conjunto de acciones, definimos el conjunto de acciones temporizadas de A que están en B y las que no están en B, respectivamente como:

$$A \cap B=\{at \mid at \in A,\ a \in B\}, \qquad A \setminus B=\{at \mid at \in A,\ a \notin B\}$$

- Si $a \in Act$, diremos que $a \in A$ si existe $t \in \mathcal{T}$ tal que $at \in A$.

- Diremos que $A \leq t$ (resp. $A<t$) si todo $a't' \in A$ verifica $t' \leq t$ (resp. $t'<t$).

- Definiremos $TAct(A)=A \upharpoonright \mathsf{nd}(A)$, que nos da por tanto el conjunto de acciones temporizadas de A, olvidando la información sobre divergencia. Se podría definir de manera equivalente $TAct(A)=A \cap Act$.

□

Puesto que los conjuntos de acciones temporizadas son un caso particular de estados ($\mathsf{nd}(A)=\infty$), las definiciones anteriores las aplicaremos también a conjuntos de acciones temporizadas.

A continuación definimos una relación de orden entre estados $\prec$ que será de utilidad en el capítulo 3 a la hora de definir los estados de un proceso, y posteriormente en el capítulo 4 para definir un orden parcial completo en el dominio semántico:

Definición 2.4.4 Sean A_1 y A_2 dos estados, decimos que A_1 es menor que A_2, y lo denotamos por $A_1 \prec A_2$ si y sólo si

$$\mathsf{nd}(A_1) \leq \mathsf{nd}(A_2) \quad \text{y} \quad TAct(A_1)=A_2 \upharpoonright \mathsf{nd}(A_1)$$

□

El significado intuitivo del orden arriba definido es el siguiente: un estado A_1 está menos definido que otro estado A_2 si el instante en el que está indefinido A_1 es menor que el instante en el que está indefinido A_2, y las acciones de A_1 (hasta el instante de indefinición) son las mismas que las de A_2. En particular, si $\mathsf{nd}(A_1) = \infty$ tenemos

$$A_1 \prec A_2 \quad \Longleftrightarrow \quad A_1 = A_2$$

Es fácil comprobar que la relación arriba definida es una relación de orden entre estados.

Definición 2.4.5 Sea $\mathcal{A} = \{A_k \mid k \in \mathbb{N}\}$ una cadena no descendente

$$A_1 \prec A_2 \prec \cdots A_{k-1} \prec A_k \cdots$$

definimos:

$$\mathsf{lub}(\mathcal{A}) = \left(\bigcup_{i=1}^{\infty} TAct(A_i)\right) \bigcup \begin{cases} \{\Omega t\} & \exists l : \forall k > l \ \mathsf{nd}(A_k) = t \\ \varnothing & \text{en otro caso} \end{cases}$$

□

Se puede comprobar fácilmente que $\mathsf{lub}(\mathcal{A})$ es un estado y es la menor cota superior de $\mathcal{A}$ con respecto al orden $\prec$. La razón por la que hemos introducido estas definiciones se encuentra en la necesidad de poder definir los estados para procesos no finitos. Esta definición se hará mediante aproximaciones, de manera que si un estado A está en una determinada aproximación pueden ocurrir los siguientes hechos:

- El estado A está definido en su totalidad; es decir, será ya un estado definitivo del proceso original.
- El estado A no está completamente definido. En este caso el estado A estará completamente definido hasta el instante de indefinición. En sucesivas aproximaciones, el estado A dará lugar a uno o varios estados, pero todos ellos *completando* lo que A deja de *definir*, es decir, aumentado el instante de indefinición e introduciendo nuevas acciones a partir de ese instante.

Para cerrar la cuestión damos la siguiente

Definición 2.4.6 Decimos que un conjunto de estados $\mathcal{A}$ es *temporalmente compacto* si y sólo si para todo estado A se verifica

$$(\forall t \in \mathcal{T}\ \exists A_t \in \mathcal{A} :\ A_t \upharpoonright t = A \upharpoonright t) \quad \Rightarrow \quad A \in \mathcal{A}$$

□

Era

Si nos fijamos en la definición anterior, nos daremos cuenta que si $\mathsf{nd}(A) < \infty$ y se verifica la condición de la definición, entonces podemos tomar $t' > \mathsf{nd}(A)$, por lo que, teniendo en cuenta que A y $A_{t'}$ son estados, tenemos $A = A_{t'}$. Por lo tanto se tiene la

Proposición 2.4.7 Un conjunto de estados $\mathcal{A}$ es temporalmente compacto si y sólo si es cierta la condición de su definición para estados A tales que $\mathsf{nd}(A) = \infty$, es decir,

$$\forall A \in \mathcal{ST}\Big(\mathsf{nd}(A) = \infty \ \wedge\ (\forall t \in \mathcal{T}\ \exists A_t \in \mathcal{A} :\ A_t \upharpoonright t = A \upharpoonright t) \quad \Rightarrow \quad A \in \mathcal{A}\Big)$$

La idea que hay detrás del concepto de un conjunto temporalmente compacto de estados, es que los estados que están en él están totalmente determinados por sus *aproximaciones temporales*. Esta propiedad se puede ver como una cierta continuidad temporal: un conjunto de estados $\mathcal{A}$ será temporalmente compacto si se puede ver como el límite de sus aproximaciones temporales:

$$\mathcal{A}_k = \{A_k \mid A \in \mathcal{A}\} \quad \text{donde} \quad A_k = \begin{cases} A & \text{si } \mathsf{nd}(A) \leq k \\ A \upharpoonright k \cup \{\Omega k\} & \text{si } \mathsf{nd}(A) > k \end{cases}$$

y

$$\mathcal{A} = \left\{ A \;\middle|\; \begin{array}{l} \mathsf{nd}(A) < \infty \ \wedge\ \forall k \in \mathbb{N}\ \exists l \geq k :\ A \in \mathcal{A}_k \\ \mathsf{nd}(A) = \infty \ \wedge\ \forall t \in \mathcal{T}\ \exists l \in \mathbb{N}\ , A_t \in \mathcal{A}_t :\ A_t \upharpoonright t = A \upharpoonright t \end{array} \right\}$$

A continuación demostramos una serie de resultados sobre dicha clase de estados que serán útiles en capítulos sucesivos.

Proposición 2.4.8 Supongamos que $\mathcal{A}$ es un conjunto de estados temporalmente compacto, y A un estado tal que $\mathsf{nd}(A) = \infty$ y para todo $t \in \mathcal{T}$ existe un estado $A_t \in \mathcal{A}$ de manera que $A \upharpoonright t \subseteq A_t$ (resp. $A_t \upharpoonright t \subseteq A$). Entonces existe algún estado $A' \in \mathcal{A}$ tal que $A \subseteq A'$ (resp. $A' \subseteq A$). Además A' lo podemos escoger de manera que:

$$\forall t \in \mathcal{T}\ \exists t' \geq t :\ A_{t'} \upharpoonright t = A' \upharpoonright t$$

Demostración. Puesto que estamos considerando un alfabeto finito y un dominio de tiempo discreto, para un cierto $t \in \mathcal{T}$ sólo puede haber un número finito de $A_{t'} \in \mathcal{A}$ con $t' \geq t$ tales que los conjuntos $A_{t'} \upharpoonright t$ sean distintos.

Por lo tanto podemos encontrar una subsecuencia $\{A_{s_t} \mid t \in \mathcal{T}\}$ tal que $A \upharpoonright t \subseteq A_{s_t}$ y $A_{s_t} \upharpoonright t = A_{s_{t'}} \upharpoonright t$ para $t' \geq t$. Entonces podemos considerar el estado

$$A' = \bigcup_{t \in \mathcal{T}} A_{s_t} \upharpoonright t$$

de modo que para cada $t \in \mathcal{T}$ tenemos que $A' \upharpoonright t = A_{s_t} \upharpoonright t$, por lo cual tenemos:

- Puesto que $\mathcal{A}$ es temporalmente compacto tenemos que $A' \in \mathcal{A}$.
- Puesto que para cada $t \in \mathcal{T}$ tenemos

$$A' \upharpoonright t = A_{s_t} \upharpoonright t \subseteq A \text{ (resp. } A \subseteq A_{s_t} \upharpoonright t = A' \upharpoonright t)$$

  y concluimos que $A' \subseteq A$ (resp. $A \subseteq A'$).

- Tomando $t' = s_t$ tenemos que $A' \upharpoonright t' = A_{t'}$.

□

Proposición 2.4.9 Si $\mathcal{A}_1$ y $\mathcal{A}_2$ son conjuntos de estados temporalmente compactos entonces $\mathcal{A}_1 \cup \mathcal{A}_2$ es un conjunto de estados temporalmente compacto.

Demostración. Para probarlo consideremos un estado A tal que $\mathsf{nd}(A) = \infty$ y supongamos que para cada $t \in \mathcal{T}$ existe $A_t \in \mathcal{A}_1 \cup \mathcal{A}_2$ tal que $A \upharpoonright t = A_t \upharpoonright t$. Entonces tenemos que un número infinito de los A_t están en $\mathcal{A}_1$ o bien un número infinito están en $\mathcal{A}_2$. En consecuencia tenemos dos posibilidades:

- Para cada $t \in \mathcal{T}$ existe $s_t \geq t$ tal que $A_{s_t} \in \mathcal{A}_1$. Entonces tenemos que $A_{s_t} \upharpoonright t = A \upharpoonright t$, y puesto que $\mathcal{A}_1$ es temporalmente compacto, tendremos que $A \in \mathcal{A}_1$
- Para cada $t \in \mathcal{T}$ existe $s_t \geq t$ tal que $A_{s_t} \in \mathcal{A}_2$. Entonces tenemos que $A_{s_t} \upharpoonright t = A \upharpoonright t$, y puesto que $\mathcal{A}_2$ es temporalmente compacto, tendremos que $A \in \mathcal{A}_2$

□

2.5 Barbas y b-trazas

Nosotros hemos adaptado el concepto de barba introducido en [HR95], manteniendo su significado intuitivo. Para explicar lo que es una barba conviene primero explicar el concepto de b-traza, que es una generalización del concepto de traza para las álgebras de procesos tradicionales sin tiempo. Debido a la urgencia, en una *b-traza* se ha de guardar información acerca de las acciones que han sido ofrecidas en cada instante anterior a la ejecución de una determinada acción. Una *b-traza* es una secuencia finita:

$$bs = A_1 a_1 t_1 A_2 a_2 t_2 \cdots A_n a_n t_n$$

donde $n \geq 0$, $A_i \subseteq TAct$, $a_i t_i \in TAct$ y $A_i < t_i$. Hemos dicho que $n \geq 0$, por lo tanto un caso particular corresponde a $n = 0$, en cuyo caso tenemos la b-traza vacía

un

que denotaremos por ϵ. Diremos que n es la longitud de la b-traza, $\mathsf{lon}(bs)$. Más adelante (capítulo 3) diremos cuándo un proceso P puede ejecutar una b-traza y convertirse tras ello en otro proceso P', lo que indicaremos en forma:

$$P \stackrel{bs}{\Longrightarrow} P'$$

La idea que hay detrás es que el proceso puede ejecutar las acciones a_i en instante t_i, teniendo en cuenta que el instante siempre es relativo a la acción anterior; pero además, en cada correspondiente instante de tiempo, se han ido ofreciendo las acciones temporizadas que forman cada conjunto A_i. Como hemos dicho anteriormente, la razón por la cual tenemos esta información es la urgencia: si el proceso hubiera podido interactuar con el entorno (con una prueba) mediante una acción temporizada $at \in A_i$, esta última se tendría que haber ejecutado, haciendo imposible la ejecución de la acción a_i en el instante t_i. Por tanto si esa b-traza ha sido ejecutada, ello significa que el entorno no podía sincronizar en las acciones temporizadas $at \in A_i$.

Por último el concepto de barba es la generalización adecuada del concepto de aceptación. Si $bs = A_1 a_1 t_1 \cdots A_n a_n t_n$ es una b-traza y A un estado tenemos que una barba b viene dada por la concatenación de una b-traza y un estado:

$$b = bs \cdot A = A_1 a_1 t_1 \cdots A_n a_n t_n A$$

Diremos que $b = bs \cdot A$ es una barba de P si existe un proceso P' tal que $P \stackrel{bs}{\Longrightarrow} P'$ y A es un estado de P'. Formalmente tenemos:

Definición 2.5.1

- Definimos el conjunto de *b-trazas* como el menor conjunto $\mathcal{BS}$ que satisface
  - $\epsilon \in \mathcal{BS}$.
  - Si $bs \in \mathcal{BS}$, $at \in TAct$, $A \subseteq TAct$ y $A < t$ entonces $Aat \cdot bs \in \mathcal{BS}$.
- Definimos el conjunto de barbas como $\mathcal{B} = \{bs \cdot A \mid bs \in \mathcal{BS},\ A \in \mathcal{ST}\}$

□

Como caso particular de la definición anterior tenemos las barbas $\epsilon \cdot A$, para simplificar esta notación diremos que A es una barba (omitiendo la b-traza vacía), con lo que podemos considerar los estados como casos particulares de barbas.

A continuación daremos una serie de operaciones que utilizaremos tanto sobre b-trazas sobre con barbas:

Definición 2.5.2

- Operaciones de concatenación, denotadas por un punto $(\cdot)$
  - Si bs y bs' son b-trazas, $bs \cdot bs'$ será la b-traza concatenación de las dos:

$$\begin{aligned} bs &= A_1 a_1 t_1 \cdots A_n a_n t_n \\ bs' &= A'_1 a'_1 t'_1 \cdots A'_m a'_m t'_m \\ bs \cdot bs' &= A_1 a_1 t_1 \cdots A_n a_n t_n A'_1 a'_1 t'_1 \cdots A'_m a'_m t'_m \end{aligned}$$

  - Si bs es una b-traza y b una barba, $bs \cdot b$ será la barba concatenación de bs y b:

$$\begin{aligned} bs &= A_1 a_1 t_1 \cdots A_n a_n t_n \\ b &= A'_1 a'_1 t'_1 \cdots A'_m a'_m t'_m A \\ bs \cdot b &= A_1 a_1 t_1 \cdots A_n a_n t_n A'_1 a'_1 t'_1 \cdots A'_m a'_m t'_m A \end{aligned}$$

- Definimos los desplazamientos temporales de la siguiente manera, si $t \in \mathcal{T}$, bs es una b-traza y b es una barba entonces

$$bs + t = \begin{cases} \epsilon & \text{si } bs = \epsilon \\ (A_1 + t) a_1 (t_1 + t) \cdot bs_1 & \text{si } bs = A_1 a_1 t_1 \cdot bs_1 \end{cases}$$

$$b + t = \begin{cases} A + t & \text{si } b = A \\ (bs + t) \cdot A & \text{si } b = bs \cdot A \ \text{ y } \ bs \neq \epsilon \end{cases}$$

Si $bs = A_1 a_1 t_1 \cdot bs_1$ con $t_1 \geq t$ definimos

$$bs - t = (A_1 - t) a_1 (t_1 - t) \cdot bs_1$$

y entonces para barbas

$$b - t = \begin{cases} (bs - t) \cdot A & \text{si } b = bs \cdot A,\ bs = A_1 a_1 t_1 \cdot bs_1 \text{ con } t_1 \geq t \\ A - t & \text{si } b = A \text{ y } \ \mathsf{nd}(A) \geq t \\ \text{indefinido} & \text{en otro caso} \end{cases}$$

- Si $A \subseteq TAct$, $t \in \mathcal{T}$, $A < t$ y $bs = A_1 a_1 t_1 \cdot bs_1$ es una b-traza no vacía, definimos:

$$(A, t) \sqcup bs = (A \cup (A_1 + t)) a_1 (t_1 + t) \cdot bs_1$$

- Si b es una barba y $t \in \mathcal{T}$, definimos la restricción temporal de b, $b \upharpoonright t$ como sigue

$$b \upharpoonright t = \begin{cases} A \upharpoonright t & \text{si } b = A \\ A_1 a_1 t_1 \cdot (b_1 \upharpoonright (t - t_1)) & \text{si } b = A_1 a_1 t_1 \cdot b_1 \ \text{ y } \ t_1 \leq t \\ A_1 \upharpoonright t & \text{si } b = A_1 a_1 t_1 \cdot b_1 \ \text{ y } \ t_1 > t \end{cases}$$

---

vendedor

- Si b es una barba definimos el instante de indefinición de b, $\mathsf{nd}(b)$, por medio de la siguiente

$$\mathsf{nd}(b) = \begin{cases} \mathsf{nd}(A) & \text{si } b = A \\ t_1 + \mathsf{nd}(b_1) & \text{si } b = A_1 a_1 t_1 \cdot b_1 \end{cases}$$

- Si bs es una b-traza definimos la duración de bs, $\mathsf{t}(bs)$ como sigue

$$\mathsf{t}(bs) = \begin{cases} 0 & \text{si } bs = \epsilon \\ t + \mathsf{t}(b_1) & \text{si } bs = Aat \cdot bs_1 \end{cases}$$

- Si $t \in \mathcal{T}$ y $A \subseteqq TAct$ es tal que $A \leq t$, y b es una barba, definimos

$$(A,t) \sqcup b = \begin{cases} A \cup (A' + t) & \text{si } b = A' \\ \Big((A,t) \sqcup bs\Big) \cdot A' & \text{si } b = bs \cdot A' \ \text{ y } bs \neq \epsilon \end{cases}$$

- Por último, si B es un conjunto de barbas definimos:

$$\mathsf{Btraz}(B) = \{bs \mid \exists b:\ bs \cdot b \in B\}, \quad \mathcal{A}(B) = \{A \in \mathcal{ST} \mid A \in B\}$$

Y para cada $bs \in \mathsf{Btraz}(B)$, definimos

$$\mathsf{Barb}(B, bs) = \{b \mid bs \cdot b \in B\}, \quad \mathcal{A}(B, bs) = \{A \in \mathcal{ST} \mid bs \cdot A \in B\}$$

□

Las barbas nos servirán para caracterizar la semántica de pruebas de un proceso. Para ello primero definiremos un preorden entre conjuntos de barbas. Esta relación inducirá otro preorden entre los procesos que será equivalente a la relación inducida por la semántica de pruebas. Previamente introducimos sendas relaciones de orden entre b-trazas y entre barbas:

Definición 2.5.3 Definimos inductivamente la relación entre b-trazas $\ll$ como sigue:

- $\epsilon \ll \epsilon$.
- Si $A \subseteq A'$ y $bs \ll bs'$, entonces $Aat \cdot bs \ll A'at \cdot bs'$.

A partir de esta relación la extendemos al conjunto de barbas considerando la menor relación $\ll$ que satisface:

- Siendo A y A' son conjuntos de estados tales que $\mathsf{nd}(A') \leq \mathsf{nd}(A)$, $TAct(A') \subseteq A$, y bs y bs' son b-trazas tales que $bs' \ll bs$, entonces $bs' \cdot A' \ll bs \cdot A$.

- Siendo A' es un estado y $Aat \cdot b$ es un barba tales que $\mathsf{nd}(A') \leq t$ y $TAct(A') \subseteq A$, y bs y bs' son b-trazas tales que $bs' \ll bs$, entonces $bs' \cdot A' \ll bs \cdot Aat \cdot b$.

□

Demos una breve explicación de la definición anterior. En primer lugar, tenemos que una b-traza bs' es *peor* respecto $\ll$ que otra b-traza bs si las acciones ejecutadas por ambas $a_i t_i$ son las mismas, pero los conjuntos de acciones ofrecidos en la primera b-traza (A'_i) son más pequeños que los de las segunda (A_i):

$$A'_1 a_1 t_1 A'_2 a_2 t_2 \cdots A'_n a_n t_n \ll A_1 a_1 t_1 A_2 a_2 t_2 \cdots A_n a_n t_n \iff A'_i \subseteq A_i$$

Lo mismo vale para las barbas, si bien en este caso hay que considerar una variante, pues si un estado está indefinido *no nos importa lo que ocurre después*; de ahí la segunda parte de la definición correspondiente a las barbas. Por otro lado observemos que la relación $\ll$ también podemos suponerla definida entre estados, puesto que los estamos considerando casos particulares de barbas:

$$A \ll A' \iff TAct(A) \subseteq A' \wedge \mathsf{nd}(A) \leq \mathsf{nd}(A')$$

Finalmente extendemos esta relación a conjuntos de barbas como sigue:

Definición 2.5.4 Siendo B y B' conjuntos de barbas tendremos

$$B \ll B' \iff \forall b' \in B' \exists b \in B : b \ll b'$$

□

Esta última relación será la que nos servirá para caracterizar la semántica de pruebas.

de

Esta página está intencionadamente en blanco.

Capítulo 3

Lenguaje Básico

En este capítulo introduciremos un álgebra de procesos secuencial básica, pero suficientemente expresiva, pues contiene los elementos básicos de cualquier álgebra de procesos. En concreto tendremos los siguientes operadores:

- Operador de interbloqueo STOP, que representa un sistema que no puede evolucionar.
- Operador de divergencia DIV, representa un sistema que está totalmente fuera de control.
- Operador de prefijo: si $e \in \mathcal{E}^*$ y $t \in \mathcal{T}$ el proceso $et\,;P$ representa un proceso que ejecuta el evento e en tiempo t y después se comporta como P. Por medio de este operador introducimos el tiempo en el álgebra, pues las acciones serán ejecutadas en el instante t que se indica.
- Operador de elección externa ($\Box$). El proceso $P \Box Q$ representa una elección para el entorno; éste elegirá implícitamente entre el proceso P y el proceso Q a través de la primera acción a ejecutarse.
- Operador de elección interna ($\sqcap$). En este caso en el proceso $P \sqcap Q$, es el propio proceso quien elige libremente comportarse bien como P o como Q.

3.1 Sintaxis

Como es usual en las álgebras de procesos, el conjunto de procesos sintácticos será el conjunto de términos sobre una determinada signatura. En este caso se trata de la signatura Σ_{seq} dada por:

*Recordemos que $\mathcal{E} = Act \cup \{\tau\}$.

Definición 3.1.1 Consideramos el conjunto de operadores

$$\Sigma_{\mathsf{seq}} = \{\mathtt{STOP}, \mathtt{DIV}, \Box, \sqcap\} \cup \{et\,;\ \mid e \in \mathcal{E},\ t \in \mathcal{T}\}$$

donde

- STOP y DIV son operadores de aridad cero.
- Si $e \in \mathcal{E}$ y $t \in \mathcal{T}$ el operador et; tiene aridad uno. Este operador lo presentaremos en forma prefija.
- $\Box$ y $\sqcap$ son operadores binarios, que representaremos en forma infija.

El conjunto de procesos será el conjunto de términos cerrados considerando la signatura Σ_{seq}: $\mathsf{CRec}(\Sigma_{\mathsf{seq}})$.

$\Box$

3.2 Semántica Operacional

Una vez definida la sintaxis de los procesos procedemos a dotarles de una semántica operacional. La semántica operacional de un proceso P vendrá dada por la tupla

$$\langle \mathsf{CRec}(\Sigma_{\mathsf{seq}}), P, \longrightarrow, \rightarrowtail \rangle$$

donde las transiciones

- $\longrightarrow \subseteq \mathsf{CRec}(\Sigma_{\mathsf{seq}}) \times (\mathcal{E} \times \mathcal{T}) \times \mathsf{CRec}(\Sigma_{\mathsf{seq}})$,
- $\rightarrowtail \subseteq \mathsf{CRec}(\Sigma_{\mathsf{seq}}) \times \mathsf{CRec}(\Sigma_{\mathsf{seq}})$,

vendrán definidas por las reglas y axiomas que presentaremos más adelante. Para simplificar la notación tendremos:

- Si $(P, et, Q) \in \longrightarrow$ escribiremos $P \xrightarrow{et} Q$.
- Si $(P, Q) \in \rightarrowtail$ escribiremos $P \rightarrowtail Q$.

Para poder definir estas transiciones necesitamos las siguientes funciones auxiliares:

- El predicado estabilidad $\mathsf{stb}(\cdot) : \mathsf{CRec}(\Sigma_{\mathsf{seq}}) \mapsto \mathsf{BOOL}$. Intuitivamente un proceso será estable cuando no pueda hacer transiciones del tipo $\rightarrowtail$. La definición directa es la siguiente:

$$\mathsf{stb}(P) = \begin{cases} \mathsf{false} & \text{si } P = \mathtt{DIV} \\ \mathsf{true} & \text{si } P = \mathtt{STOP} \\ \mathsf{true} & \text{si } P = et\,;P_1 \text{ con } e \in \mathcal{E} \\ \mathsf{stb}(P_1) \wedge \mathsf{stb}(P_2) & \text{si } P = P_1 \,\Box\, P_2 \\ \mathsf{false} & \text{si } P = P_1 \sqcap P_2 \text{ ó } P = \mathtt{REC}\, x.P_1 \end{cases}$$

- Necesitaremos también la función

$$\mathsf{Tiem}(\cdot,\cdot) : \mathsf{CRec}(\Sigma_{\mathsf{seq}}) \times Act \mapsto \mathcal{P}_{\mathrm{fin}}(\mathcal{T}),$$

Si $P \in \mathsf{CRec}(\Sigma_{\mathsf{seq}})$ y $a \in Act$ nos indica los instantes de tiempo en el que el proceso P puede ejecutar la acción a. En concreto

$$\mathsf{Tiem}(P,a) = \begin{cases} \varnothing & \text{si } P = \mathtt{STOP} \text{ ó } P = \mathtt{DIV} \\ \varnothing & \text{si } P = P_1 \sqcap P_2 \text{ ó } P = \mathtt{REC}\, x.P' \\ \{t\} & \text{si } P = at\,;P \\ \varnothing & \text{si } P = et\,;P_1,\ e \in \mathcal{E} \text{ y } a' \neq a \\ \mathsf{Tiem}(P_1,a) \cup \mathsf{Tiem}(P_2,a) & \text{si } P = P_1 \,\Box\, P_2 \end{cases}$$

- El tiempo que un proceso puede permanecer inactivo

$$\mathsf{idle}(\cdot,\cdot) : \mathsf{Rec}(\Sigma_{\mathsf{seq}}) \times \mathcal{P}_{\mathrm{fin}}(Act) \mapsto \mathcal{T} \cup \{\infty\}$$

La función $\mathsf{idle}(P,A)$ indica el instante en el que el proceso P ejecuta la primera acción oculta o bien perteneciente al conjunto A. En concreto tenemos

$$\mathsf{idle}(P,A) = \begin{cases} 0 & \text{si } P = \mathtt{DIV} \text{ ó } P = x \\ \infty & \text{si } P = \mathtt{STOP} \text{ ó } P = at\,;P' \text{ con } a \in Act \setminus A \\ t & \text{si } P = et\,;P' \text{ con } e \in A \cup \{\tau\} \\ \min(\mathsf{idle}(P_1),\mathsf{idle}(P_2)) & \text{si } P = P_1 \,\Box\, P_2 \\ 0 & \text{si } P = P_1 \sqcap P_2 \text{ ó } P = \mathtt{REC}\, x.P_1 \end{cases}$$

Cuando $A = \varnothing$ escribiremos simplemente $\mathsf{idle}(P)$ que indicará el tiempo en el que se ejecuta la primera acción oculta. La razón por la que se ha introducido el conjunto A en la definición es para poder definir más adelante en el capítulo 6 la función para el caso de ocultamiento.

- La función de actualización temporal $\mathsf{Upd}(\cdot,\cdot) : \mathsf{Proc}(X) \times \mathcal{T} \mapsto \mathsf{Proc}(X)$. Intuitivamente $\mathsf{Upd}(P,t)$ es el resultado del paso (sin ejecución de acción alguna) de t

perfeccionadas,

unidades de tiempo sobre el proceso P. En concreto tenemos la siguiente definición recursiva:

$$\mathsf{Upd}(P,t) = \begin{cases} \mathtt{STOP} & \text{si } P = \mathtt{STOP} \\ \mathtt{DIV} & \text{si } P = \mathtt{DIV} \\ e(t'-t)\,;P & \text{si } e \in \mathcal{E},\ P = et'\,;P' \ \text{ y } \ t' \geq t \\ \mathtt{STOP} & \text{si } e \in \mathcal{E},\ P = et'\,;P' \ \text{ y } \ t' < t \\ \mathtt{DIV} & \text{si } P = x \\ \mathsf{Upd}(P_1,t) \,\Box\, \mathsf{Upd}(P_2,t) & \text{si } P = P_1 \,\Box\, P_2 \\ \mathsf{Upd}(P_1,t) \sqcap \mathsf{Upd}(P_2,t) & \text{si } P = P_1 \sqcap P_2 \\ \mathsf{Upd}(P',t)[\mathtt{REC}\, x.P'/x] & \text{si } P = \mathtt{REC}\, x.P' \end{cases}$$

Esta función se precisará para definir la semántica del operador binario de elección externa, y más adelante para el paralelo, cuando un proceso ejecuta una acción y es necesario indicar sobre su *compañero* que el tiempo ha pasado. Debido a las características de la semántica operacional que veremos, esta función nunca se aplicará en los casos en los que un proceso pueda hacer una transición del tipo $\rightarrowtail$, por lo que la definición de $\mathsf{Upd}(P \sqcap Q, t)$ y de $\mathsf{Upd}(\mathtt{REC}\, x.P, t)$ son arbitrarias, puesto que en ninguno de estos casos se aplicará nunca dicha función. Igualmente, debido a las propiedades de la semántica operacional, tendremos que si un proceso P puede realizar una transición del tipo $\xrightarrow{\tau t}$, nunca intentaremos calcular $\mathsf{Upd}(P, t')$ para $t' > t$, por lo que la definición de $\mathsf{Upd}(\tau t; P, t')$ cuando $t' > t$ también de algún modo es arbitraria. En cualquier caso la definición en todos estos casos la hemos hecho de forma que se respete la idea intuitiva de actualización.

Pasemos a continuación, siguiendo las ideas en [Plo81], a dar las reglas y axiomas que permiten deducir las transiciones semánticas, para ello iremos analizando cada uno de los operadores. El cuadro esquemático con las reglas y axiomas que definen las transiciones lo podemos encontrar en la tabla 3.1.

Interbloqueo y divergencia

El interbloqueo STOP y la divergencia DIV son procesos en principio muy similares, puesto que ninguno de los dos puede ejecutar ningún tipo de acciones visibles. La gran diferencia reside en que el interbloqueo se comporta *bien* en compañía de otros procesos mientras que la divergencia no lo hace. Ello se traduce en la semántica operacional: el interbloqueo no realiza ningún tipo de transiciones, y la divergencia realiza transiciones del tipo $\rightarrowtail$; en consecuencia, y puesto que las transiciones $\rightarrowtail$ tienen el máximo nivel de urgencia,

[**DIV**] $\texttt{DIV} \rightarrowtail \texttt{DIV}$	[**PRE**] $et\,;P \xrightarrow{et} P$
[**EXT1**] $\dfrac{P \xrightarrow{at} P', \quad \mathsf{idle}(Q) \geq t, \quad \mathsf{stb}(Q)}{P \,\Box\, Q \xrightarrow{at} P'}$	[**EXT2**] $\dfrac{Q \xrightarrow{at} Q', \quad \mathsf{idle}(P) \geq t, \quad \mathsf{stb}(P)}{P \,\Box\, Q \xrightarrow{at} Q'}$
[**EXT3**] $\dfrac{P \xrightarrow{\tau t} P', \quad \mathsf{idle}(Q) \geq t, \quad \mathsf{stb}(Q)}{P \,\Box\, Q \xrightarrow{\tau t} P' \,\Box\, \mathsf{Upd}(Q,t)}$	[**EXT4**] $\dfrac{Q \xrightarrow{\tau t} Q', \quad \mathsf{idle}(P) \geq t, \quad \mathsf{stb}(P)}{P \,\Box\, Q \xrightarrow{\tau t} \mathsf{Upd}(P,t) \,\Box\, Q'}$
[**EXT5**] $\dfrac{P \rightarrowtail P'}{P \,\Box\, Q \rightarrowtail P' \,\Box\, Q}$	[**EXT6**] $\dfrac{Q \rightarrowtail Q'}{P \,\Box\, Q \rightarrowtail P \,\Box\, Q'}$
[**INT1**] $P \sqcap Q \rightarrowtail P$	[**INT2**] $P \sqcap Q \rightarrowtail Q$
[**REC**] $\texttt{REC}\,x.P \rightarrowtail P[\texttt{REC}\,x.P/x]$	

Tabla 3.1: Semántica operacional de los operadores básicos.

ello dejará progresar al resto de los procesos. Queda en consecuencia justificada la regla

$$[\textbf{DIV}] \quad \texttt{DIV} \rightarrowtail \texttt{DIV}$$

Prefijo

La semántica operacional del prefijo, tanto para acciones visibles, como para acciones ocultas, viene dada por la siguiente regla:

$$[\textbf{PRE}] \quad et\,;P \xrightarrow{et} P$$

donde $e \in \mathcal{E}$ y $t \in \mathcal{T}$.

Elección Externa

La elección externa sólo se *realiza* mediante la ejecución de una acción visible. Puesto que las acciones visibles no son urgentes, sólo se podrán realizar si no hay acciones con una mayor urgencia. Por lo tanto tenemos las siguientes reglas:

$$[\textbf{EXT1}] \quad \frac{P \xrightarrow{at} P', \quad \mathsf{idle}(Q) \geq t, \quad \mathsf{stb}(Q)}{P \,\Box\, Q \xrightarrow{at} P'}$$

$$[\textbf{EXT2}] \quad \frac{Q \xrightarrow{at} Q', \quad \mathsf{idle}(P) \geq t, \quad \mathsf{stb}(P)}{P \,\Box\, Q \xrightarrow{at} Q'}$$

---
de

donde $a \in Act$ y $t \in \mathcal{T}$.

Las acciones pueden ser ejecutadas por cualquier argumento, en la medida que las normas de la urgencia lo permiten. Sin embargo no visibles no *deshacen* la elección externa, por todo lo cual tenemos las siguientes reglas:

$$\textbf{[EXT3]} \quad \frac{P \xrightarrow{\tau t} P', \quad \mathsf{idle}(Q) \geq t, \quad \mathsf{stb}(Q)}{P \mathbin{\Box} Q \xrightarrow{\tau t} P' \mathbin{\Box} \mathsf{Upd}(Q, t)}$$

$$\textbf{[EXT4]} \quad \frac{Q \xrightarrow{\tau t} Q', \quad \mathsf{idle}(P) \geq t, \quad \mathsf{stb}(P)}{P \mathbin{\Box} Q \xrightarrow{\tau t} \mathsf{Upd}(P, t) \mathbin{\Box} Q'}$$

$$\textbf{[EXT5]} \quad \frac{P \rightarrowtail P'}{P \mathbin{\Box} Q \rightarrowtail P' \mathbin{\Box} Q} \qquad \textbf{[EXT6]} \quad \frac{Q \rightarrowtail Q'}{P \mathbin{\Box} Q \rightarrowtail P \mathbin{\Box} Q'}$$

donde $t \in \mathcal{T}$.

Elección Interna

La elección interna representa la combinación no determinista de dos procesos. La resolución de este no determinismo se hace con absoluta urgencia, a través de transiciones $\rightarrowtail$ gobernadas por la siguiente regla:

$$\textbf{[INT1]} \quad P \sqcap Q \rightarrowtail P \qquad \textbf{[INT2]} \quad P \sqcap Q \rightarrowtail Q$$

Recursión

El comportamiento del proceso $\mathtt{REC}\, x.P$ es exactamente el mismo que el del proceso P en el cual la variable x ha sido sustituida por el proceso $\mathtt{REC}\, x.P$. Esto se puede traducir en la semántica operacional por medio de la siguiente regla

$$\textbf{[REC]} \quad \mathtt{REC}\, x.P \rightarrowtail P[\mathtt{REC}\, x.P/x]$$

Para concluir esta sección introducimos una última

Definición 3.2.1 Siendo $P \in \mathsf{CRec}(\Sigma_{\mathsf{seq}})$, definimos el conjunto de *acciones iniciales* de P como el conjunto de acciones temporizadas

$$\mathsf{TA}(P) = \{at \mid \exists P' : P \xrightarrow{at} P' \wedge a \in Act\}$$

$\Box$

3.2.1 Propiedades de la Semántica Operacional

En primer lugar demostraremos que las funciones auxiliares definidas: $\mathsf{idle}(P, A)$, $\mathsf{stb}(P)$ y $\mathsf{Upd}(P, t)$ se comportan como se espera de su definición intuitiva. Ello queda plasmado en la siguiente

Proposición 3.2.2 Siendo $P \in \mathsf{CRec}(\Sigma_{\mathsf{seq}})$ tenemos

- $\mathsf{stb}(P) = \mathsf{true} \iff P \succ\!\!\not\longrightarrow$
- $\mathsf{Tiem}(P, a) = \{t \mid \exists P' : P \xrightarrow{at} P'\}$
- *Suponiendo* $\mathsf{stb}(P)$ *se tiene*

$$\mathsf{idle}(P, A) = \min\left\{t \,\middle|\, \exists P' : P \xrightarrow{et} P' \wedge e \in A \cup \{\tau\}\right\}$$

- Siendo $t', t \in \mathcal{T}$, $t' \geq t$, $e \in \mathcal{E}$, $\mathsf{stb}(P)$ e $\mathsf{idle}(P) \geq t$ tenemos

$$P \xrightarrow{et} P' \iff \mathsf{Upd}(P, t) \xrightarrow{e(t'-t)} P'$$

  Además si P es estable $\mathsf{Upd}(P, t)$ también lo es.

Demostración. La demostración es inmediata por inducción estructural.

□

Como consecuencia de la proposición anterior podemos dar el siguiente

Corolario 3.2.3 Sea $P \in \mathsf{CRec}(\Sigma_{\mathsf{seq}})$ tal que $\mathsf{stb}(P)$ e $\mathsf{idle}(P) \geq t$ entonces

$$\mathsf{TA}(P) = (\mathsf{TA}(P) \upharpoonright t) \cup (\mathsf{TA}(\mathsf{Upd}(P, t)) + t) = \mathsf{TA}(P) \,\wr\, t \cup (\mathsf{TA}(\mathsf{Upd}(P, t)) + t)$$

También como corolario tenemos que la semántica operacional goza de la propiedad de la urgencia, formalizada en la

Proposición 3.2.4 Para cada $P \in \mathsf{CRec}(\Sigma_{\mathsf{seq}})$ se tiene

$$P \xrightarrow{et} P' \quad \Rightarrow \quad P \succ\!\!\not\longrightarrow \wedge\, P \overset{\tau t'}{\not\longrightarrow} \; t' < t$$

Demostración. La demostración resulta trivial por inducción estructural, teniendo en cuenta la proposición 3.2.2.

□

La siguiente propiedad, que podríamos denominar de *no determinismo acotado* o *ramificación finita*, jugará un importante papel a lo largo del trabajo.

---

las

Proposición 3.2.5 Tomemos $P \in \mathsf{CRec}(\Sigma_{\mathrm{seq}})$ entonces

- Existe un número finito de procesos P' tal que $P \rightarrowtail P'$.
- Dados un evento $e \in \mathcal{E}$ y un tiempo $t \in \mathcal{T}$, existe a lo sumo un número finito de procesos P' tal que $P \xrightarrow{et} P'$.

Demostración. La demostración es inmediata por inducción estructural.

□

3.3 Semántica de Pruebas

Como elemento auxiliar previo para poder definir la semántica de pruebas, necesitamos la noción de divergencia. Intuitivamente, diremos que un proceso diverge si puede realizar consecutivamente un número ilimitado de acciones internas, todas ellas en tiempo cero. Para definir con mayor precisión la noción de divergencia, necesitamos en primer lugar la siguiente

Definición 3.3.1 Definimos el predicado de convergencia débil, $P \downarrow$, como el menor predicado que satisface:

- $\mathtt{STOP}\downarrow$, $at\,;P\downarrow$, y $\tau t\,;P\downarrow$.
- si $P\downarrow$ y $Q\downarrow$ entonces $P_1 \sqcap P_2 \downarrow$ y $P_1 \,\Box\, P_2 \downarrow$.
- si $P\downarrow$ entonces $\mathtt{REC}\,x.P\downarrow$.

Diremos que $P\uparrow$ si $P\downarrow$ es falso.

□

El concepto anteriormente definido es puramente auxiliar, pues por ejemplo tenemos $\mathtt{DIV}\uparrow$, pero sin embargo $\tau 0\,;\mathtt{DIV}\downarrow$. En dicha definición se contemplan dos tipos de procesos divergentes (a parte de los construidos a partir de ellos): la recursión no guardada, y el proceso DIV. Tras ella podemos completar la definición formal de convergencia:

Definición 3.3.2 Diremos que un proceso converge (o converge fuertemente), lo que representaremos por $P\Downarrow$, si y sólo si se tiene

$$P\downarrow \quad \wedge \quad \forall P' : ((P \xrightarrow{\tau 0} P' \vee P \rightarrowtail P') \;\Rightarrow\; P'\Downarrow)$$

Diremos que un proceso P diverge, y lo escribiremos $P\Uparrow$, si $P\Downarrow$ es falso.

□

Según la definición anterior un proceso P es divergente, si cumple una de las condiciones siguientes:

- Tras ejecutar un número finito (eventualmente ninguna) de transiciones internas $\rightarrowtail$ ó $\xrightarrow{\tau 0}$, se alcanza un proceso P' tàl que $P' \uparrow$.
- Si puede ejecutar consecutivamente una cantidad infinita de transiciones del tipo $\rightarrowtail$ ó $\xrightarrow{\tau 0}$.

A continuación hemos de introducir el concepto de prueba o test. Esencialmente una prueba será un proceso normal, pero en cuya definición se ha introducido un nuevo proceso básico, OK, que sirve para indicar cuándo un proceso pasa con éxito una prueba. Más formalmente tenemos la

Definición 3.3.3 Consideramos la signatura $\Sigma_{\text{test}} = \Sigma_{\text{seq}} \cup \{\mathtt{OK}\}^{\dagger}$, donde OK es un operador de aridad cero. El conjunto de pruebas será entonces el conjunto de términos recursivos cerrados $\mathsf{CRec}(\Sigma_{\text{test}})$ sobre dicha signatura.

□

En la práctica resultaría absurdo considerar una prueba divergente, de la forma `DIV` ó $\mathtt{REC}\,x.\tau 0\,;x$, pues en tales casos se está *probando* con una prueba *defectuosa* que nunca podrá dar resultados positivos. Más adelante veremos también que es suficiente considerar pruebas finitas, es decir, aquellas que no contengan el operador de recursión.

Es necesario ampliar la semántica operacional de los procesos para extenderla a las pruebas. Al efecto introducimos una acción $\mathtt{OK} \notin \mathcal{E}$ y extendemos las relaciones de transición

- $\rightarrowtail \subseteq \mathsf{CRec}(\Sigma_{\text{test}}) \times \mathsf{CRec}(\Sigma_{\text{test}})$
- $\longrightarrow \subseteq \mathsf{CRec}(\Sigma_{\text{test}}) \times ((\mathcal{E} \times \mathcal{T}) \cup \{\mathtt{OK}\}) \times \mathsf{CRec}(\Sigma_{\text{test}})$

para ello hemos de ampliar la definición de las funciones auxiliares en la siguiente forma:

$$\begin{aligned} \mathsf{Upd}(\mathtt{OK}, t) &= \mathtt{OK} \\ \mathsf{Tiem}(\mathtt{OK}, a) &= \varnothing^{\ddagger} \\ \mathsf{idle}(\mathtt{OK}, A) &= \infty \\ \mathsf{stb}(\mathtt{OK}) &= \mathsf{true} \end{aligned}$$

† Según se vayan introduciendo nuevos operadores en la signatura, podríamos ir considerando a la vez pruebas con un mayor número de operadores. Esto no es necesario debido a que las pruebas con estos operadores tienen ya la suficiente potencia.

‡ Puesto que $\mathtt{OK} \notin \mathcal{E}$ tenemos $\mathtt{OK} \neq a$.

que

Y añadimos las siguientes reglas que definen la semántica operacional de la prueba $\mathtt{OK}$:

$$[\textbf{OK1}] \quad \frac{}{\mathtt{OK} \xrightarrow{\mathtt{OK}} \mathtt{STOP}}$$

$$[\textbf{OK2}] \quad \frac{P \xrightarrow{\mathtt{OK}} P'}{P \,\square\, Q \xrightarrow{\mathtt{OK}} P'} \qquad [\textbf{OK3}] \quad \frac{P \xrightarrow{\mathtt{OK}} P'}{Q \,\square\, P \xrightarrow{\mathtt{OK}} P'}$$

El siguiente paso consiste en definir la composición de un proceso y una prueba $P \mid T$.

$$[\textbf{T1}] \quad \frac{P \rightarrowtail P'}{P \mid T \mapsto P' \mid T} \qquad [\textbf{T2}] \quad \frac{T \rightarrowtail T'}{P \mid T \mapsto P \mid T'}$$

$$[\textbf{T3}] \quad \frac{P \xrightarrow{\tau t} P', \quad \mathsf{stb}(T), \quad \mathsf{idle}(T) \geq t, \quad (\mathsf{TA}(P) \upharpoonright t) \cap (\mathsf{TA}(T) \upharpoonright t) = \varnothing}{P \mid T \mapsto P' \mid \mathsf{Upd}(T, t)}$$

$$[\textbf{T4}] \quad \frac{T \xrightarrow{\tau t} T', \quad \mathsf{stb}(P), \quad \mathsf{idle}(P) \geq t, \quad (\mathsf{TA}(P) \upharpoonright t) \cap (\mathsf{TA}(T) \upharpoonright t) = \varnothing}{P \mid T \mapsto \mathsf{Upd}(P, t) \mid T'}$$

$$[\textbf{T5}] \quad \frac{P \xrightarrow{at} P', \quad T \xrightarrow{at} T', \quad (\mathsf{TA}(P) \upharpoonright t) \cap (\mathsf{TA}(T) \upharpoonright t) = \varnothing}{P \mid T \mapsto P' \mid T'}$$

Tabla 3.2: Reglas de la semántica operacional de $P \mid T$.

Esta composición se podría definir en función del operador paralelo y del ocultamiento que veremos en el capítulo 6 de la siguiente manera:

$$P \mid T = (P \parallel_{Act} Q) \setminus Act$$

donde $\parallel_G$ sería el operador de paralelo, y $\setminus G$ sería el de ocultamiento. Como quiera que de momento no disponemos de dichos operadores definidos, tenemos que definir directamente la semántica operacional de dichas composiciones. Para ello introducimos las transiciones de cómputo de una prueba y un proceso

$$\mapsto \subseteq \Big(\mathsf{CRec}(\Sigma_{\mathrm{seq}}) \times \mathsf{CRec}(\Sigma_{\mathrm{test}})\Big) \times \Big(\mathsf{CRec}(\Sigma_{\mathrm{seq}}) \times \mathsf{CRec}(\Sigma_{\mathrm{test}})\Big)$$

definidas por las reglas que están en la tabla 3.2.

La idea que hay detrás de esta definición es que se ponen el proceso y la prueba a funcionar conjuntamente, se les deja funcionar de manera autónoma, y eventualmente vemos que la prueba da el *visto bueno* al proceso ejecutando la acción $\mathtt{OK}$. Las condiciones laterales en todas las reglas sirven para garantizar la urgencia. En particular la condición

$$(\mathsf{TA}(P) \upharpoonright t) \cap (\mathsf{TA}(T) \upharpoonright t) = \varnothing$$

nos garantiza que el proceso y la prueba no pueden sincronizar en una acción en un instante anterior a t, es decir, estamos exigiendo la urgencia para todas las acciones. Esto viene justificado puesto que dejamos al proceso y a la prueba funcionando como si fueran un sistema independiente, por tanto se justifica el hecho de que todas las acciones sean urgentes.

Definición 3.3.4 Una computación de un proceso P y una prueba T es una secuencia (posiblemente infinita):

$$P \mid T = P_0 \mid T_0 \mapsto P_1 \mid T_1 \cdots P_{k-1} \mid T_{k-1} \mapsto P_k \mid T_k \cdots$$

□

Definición 3.3.5 Sea P un proceso y T una prueba. Una computación de P y T

$$P \mid T = P_0 \mid T_0 \mapsto P_1 \mid T_1 \cdots P_{k-1} \mid T_{k-1} \mapsto P_k \mid T_k \cdots$$

decimos que es:

- *Completa* cuando o bien es finita y está bloqueada, es decir, no puede progresar más, o bien es infinita.
- *Tiene éxito* si existe $k \in \mathbb{N}$ tal que $T_k \xrightarrow{\texttt{OK}}$ y para cada $0 \leq i < k$ se tiene $P_i \Downarrow$.

□

A continuación definimos el concepto de *paso de una prueba*:

Definición 3.3.6

- Decimos que un proceso P *pasa una prueba* T (P must T) si cualquier computación completa de $P \mid T$ tiene éxito.
- Decimos que P es peor que Q respecto a la semántica de pruebas, y escribimos $P \sqsubseteq_{\approx} Q$ sii
$$\forall T \; (P \text{ must } T \;\Rightarrow\; Q \text{ must } T)$$
- Decimos que dos procesos P y Q son equivalentes bajo semántica de pruebas, y escribimos $P \eqsim Q$, sii
$$P \sqsubseteq_{\approx} Q \quad \text{y} \quad Q \sqsubseteq_{\approx} P$$

□

Como comentario final en esta sección observemos que, por definición, todo proceso pasa la prueba T tal que $T \xrightarrow{\texttt{OK}}$, por esta razón a estas pruebas las llamaremos *pruebas triviales*.

apagan

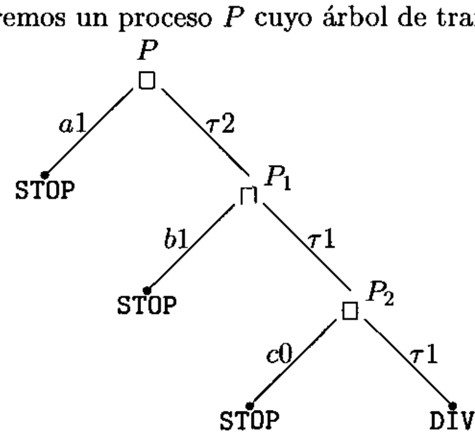
3.4 Caracterización Operacional

La semántica de pruebas introducida en la sección anterior resulta bastante intuitiva: dos procesos son iguales si se comportan igual bajo cada entorno (las pruebas). Sin embargo resulta poco manejable, para ver que dos procesos son equivalentes hay que ver que se comportan igual bajo *todas* las pruebas posibles. Por tanto es conveniente buscar una caracterización más simple. En concreto tenemos una caracterización que depende únicamente de la semántica operacional de cada proceso.

3.4.1 Estados de un Proceso

Como introdujimos en el capítulo 2 cada estado A de un proceso representa una de las posibles configuraciones en las que el mismo puede encontrarse inicialmente, conteniendo la información de cuáles son las acciones que puede realizar y el tiempo en las que son realizadas. Así $at \in A$ implica que el proceso puede ejecutar la acción a en tiempo t. También contiene información sobre el instante en el que el proceso está indefinido o diverge. Si $t = \mathsf{nd}(A) < \infty$ el proceso diverge en el instante t, por lo que a partir de ese instante no pasaría ninguna prueba, salvo las triviales. Para calcular los estados de un proceso veamos el siguiente

Ejemplo 3.4.1 Consideremos un proceso P cuyo árbol de transiciones sea el siguiente



Debido a las condiciones de urgencia, si a dicho proceso le proponemos una prueba que ofrece la acción a en el instante 1, la prueba y el proceso deberán sincronizar obligatoriamente. Si no es así y la prueba permite dejar pasar dos instantes de tiempo, el proceso ejecuta la acción oculta τ en el instante 2 para transformarse en P_1.

El proceso P_1 tiene la posibilidad de ejecutar la acción b en el instante 1 (instante 3 global), pero no la obligación; es decir, si una prueba le propone ejecutar la acción b en dicho instante, P_1 la ejecutará o no de una manera no determinista ya que es el propio proceso quien elige libremente comportarse bien como $b1$; STOP o bien como $\tau 1$; P_2. Si elige comportarse como esta segunda opción y la prueba permite al proceso que pasen dos unidades de tiempo, se ejecutará la acción interna τ en el instante 1 (instante 3 contado globalmente) y se comportará como el proceso P_2.

Este último proceso, por la misma razón que el proceso original, no puede negarse a realizar la acción c en el instante 0 (instante 3 global), pero si se le deja que pase una unidad de tiempo más, se ejecutará la acción interna τ en el instante 1 (instante 4 contado globalmente) y se transformará en el proceso indefinido DIV.

Todo ello se refleja en los estados de los procesos dados, en primer lugar

$$\mathcal{A}(P_2) = \Big\{\{c0, \Omega 1\}\Big\}$$

A continuación los estados de P_1 son, por un lado $\{b_1\}$, y por otro los de P_2 incrementados en una unidad de tiempo:

$$\mathcal{A}(P_1) = \Big\{\{b_1\}, \{c1, \Omega 2\}\Big\}$$

Por último los estados de P son los estados de P_1 incrementados en dos unidades de tiempo a los que se les añade la acción temporizada $\{a1\}$:

$$\mathcal{A}(P) = \Big\{\{a1, b_3\}, \{a1, c3, \Omega 4\}\Big\}$$

□

Puesto que los procesos que contienen en su interior definiciones recursivas podrían hacer un número infinito de acciones internas sin por ello tener que ser divergentes, hemos de definir el conjunto de estados de un proceso por etapas. En primer lugar damos la siguiente definición auxiliar:

Definición 3.4.2 Definimos el conjunto de estados iniciales de un proceso P, *usando k transiciones*, $\mathcal{A}_k(P)$ como el menor conjunto que cumple:

- Si $P \Uparrow$ ó $k = 0$ entonces $\mathcal{A}_k(P) = \Big\{\{\Omega 0\}\Big\}$.
- Si $P \Downarrow$ y $k > 0$ entonces
  - Si $P \rightarrowtail P'$ y $A \in \mathcal{A}_{k-1}(P')$, entonces $A \in \mathcal{A}_k(P)$.
  - Si $P \xrightarrow{\tau t} P'$ y $A \in \mathcal{A}_{k-1}(P)$, entonces $\mathsf{TA}(P) \upharpoonright t \cup (A + t) \in \mathcal{A}_k(P)$.
  - Si $P \not\rightarrowtail$ y $P \not\xrightarrow{\tau t}$, entonces $\mathcal{A}_k(P) = \Big\{\mathsf{TA}(P)\Big\}$.

□

la

Podemos comprobar fácilmente, por inducción sobre k, que los conjuntos $\mathcal{A}_k(P)$ son efectivamente conjuntos de estados. Finalmente, y utilizando la relación de orden entre estados introducida en la definición 2.4.4, tenemos la

Definición 3.4.3 Decimos que A es un estado de P, $A \in \mathcal{A}(P)$, si y sólo si existe una secuencia de estados

$$A_1 \prec A_2 \cdots A_k \prec A_{k+1} \cdots$$

con $A_i \in \mathcal{A}_i(P)$ y $A = \mathsf{lub}\{A_i \mid i \in \mathbb{N}\}$.

□

Puesto que la definición anterior resulta poco manejable, es conveniente buscar alguna caracterización equivalente. Para ello comenzamos dando la siguiente

Definición 3.4.4 Una computación de P

$$P = P_1 \rightarrowtail^* P_1' \xrightarrow{\tau t_1} P_2 \rightarrowtail^* P_2' \xrightarrow{\tau t_2} \cdots P_k \rightarrowtail^* P_k' \xrightarrow{\tau t_k} P_{k+1} \rightarrowtail^* \cdots$$

diremos que es *τ-maximal* sii verifica una de las siguientes condiciones:

- es infinita y $P_i \Downarrow$ para $i \in \mathbb{N}$, o
- es finita (denotando por P_n' el último proceso) $P_i \Downarrow$ para $i < n$ y, bien $\mathsf{stb}(P_n')$ y $\mathsf{idle}(P_n') = \infty$, o bien $P_n' \Uparrow$.[§]

□

Proposición 3.4.5 Siendo $P \in \mathsf{CRec}(\Sigma_{\mathrm{seq}})$, su conjunto de estados $\mathcal{A}(P)$ estará formado por los estados $A \in \mathcal{ST}$ que se pueden generar, como se describe más abajo, a partir de cada una computación τ-maximal

$$P = P_1 \rightarrowtail^* P_1' \xrightarrow{\tau t_1} P_1 \rightarrowtail^* \cdots P_k \rightarrowtail^* P_k' \xrightarrow{\tau t_k} P_{k+1} \rightarrowtail^* \cdots$$

donde tomamos $t^i = \sum_{j=1}^{i-1} t_j$ y denotamos por P_n' el último proceso si la computación es finita.

- En el caso que la computación sea infinita consideramos cada

$$A = \bigcup_{i \in \mathbb{N}} ((\mathsf{TA}(P_i') \upharpoonright t_i) + t^i)$$

[§]Observemos que si $P_i \Downarrow$ entonces $P_i' \Downarrow$, y por tanto en el caso cuando $P_n' \Uparrow$ tenemos necesariamente que $P_n = P_n'$.

- En el caso que la computación sea finita, consideramos cada

$$A = (A_n + t_n) \cup \bigcup_{i<n} ((\mathsf{TA}(P_i') \upharpoonright t_i) + t^i)$$

donde $A_n = \mathsf{TA}(P_n')$ si $P_n \Downarrow$ mientras que en caso contrario tenemos que $A_n = \{\Omega 0\}$.

Demostración. Tomemos en primer lugar $A \in \mathcal{A}(P)$, por la definición existe una sucesión de estados

$$A_1 \prec A_2 \prec \cdots A_k \prec A_{k+1} \prec \cdots$$

tal que $A = \mathsf{lub}\{A_i \mid i \in \mathbb{N}\}$. Si existe A_k tal que $\mathsf{nd}(A_k) = \infty$ entonces, por la definición de la relación $\prec$ tenemos que $A_{k'} = A_k$ para $k' \geq k$, por tanto tenemos que $A = A_k$. Puesto que $A_k \in \mathcal{A}_k(P)$ podemos encontrar fácilmente la computación requerida.

Supongamos entonces que $\mathsf{nd}(A_i) < \infty$ para $i \in \mathbb{N}$. Por un lado tenemos que

$$\mathsf{nd}(A_1) \leq \mathsf{nd}(A_2) \leq \mathsf{nd}(A_3) \cdots$$

Esa sucesión puede estar acotada o no. Si está acotada existe $k \in \mathbb{N}$ tal que para todo $l \geq k$ se verifica que $\mathsf{nd}(A_k) = \mathsf{nd}(A_l)$. Por tanto, por la definición de la relación $\prec$ tenemos que $A = A_k$, e igual que antes, podemos encontrar fácilmente la computación requerida.

Supongamos entonces que cada instante $\mathsf{nd}(A_k)$ es finito, pero no existe ninguna cota superior, en este caso tenemos

$$A = \bigcup_{i \in \mathbb{N}} TAct(A_i)$$

Para cada $k \in \mathbb{N}$ existe un proceso P_k tal que

$$P = P_{1,k} \rightarrowtail^* P_{1,k}' \xrightarrow{\tau t_{1,k}} P_{2,k} \cdots P_{l_k,k} \rightarrowtail^* P_{l_k,k}' \xrightarrow{\tau t_{l_k,k}} P_k$$

de manera que

$$TAct(A_k) = \bigcup_{1 \leq i \leq l_k} (\mathsf{TA}(P_{i,k}') \upharpoonright t_{i,k}) + t^{i,k} \qquad \text{donde} \qquad t^{i,k} = \sum_{j=1}^{i-1} t_{j,k}$$

Puesto que la semántica operacional es finitamente ramificada (propiedad 3.2.5), existe una computación infinita

$$P = P_1 \rightarrowtail^* P_1' \xrightarrow{\tau t_1} P_2 \rightarrowtail^* P_2' \xrightarrow{\tau t_2} \cdots P_k \rightarrowtail^* P_k' \xrightarrow{\tau t_k} P_{k+1} \rightarrowtail^* \cdots$$

tal que para cada $k \in \mathbb{N}$ existe $l \geq k$ tal que

$$TAct(A_l) = \bigcup_{1 \leq i \leq l} (\mathsf{TA}(P_i') \upharpoonright t_i) + t^i \qquad \text{donde} \qquad t^i = \sum_{j=1}^{i-1} t_j$$

sed.

Por tanto tendremos que $A = \bigcup_{i\in\mathbb{N}} (\mathsf{TA}(P_i') \upharpoonright t_i) + t^i$.

Supongamos ahora que el estado A está generado por una computación τ-maximal como la que se indica en el enunciado:

$$P = P_1 \rightarrowtail^* P_1' \xrightarrow{\tau t_1} P_1 \rightarrowtail^* \cdots P_k \rightarrowtail^* P_k' \xrightarrow{\tau t_k} P_{k+1} \rightarrowtail^* \cdots$$

Entonces para cada $k \in \mathbb{N}$ (tomando $k < n$ en el caso que la computación sea finita) tomamos el estado

$$A_k = \{\Omega t^k\} \cup \bigcup_{i\leq k}((\mathsf{TA}(P_i') \upharpoonright t_i) + t^i)$$

Existirá entonces $l_k \geq k$ tal que $A_k \in \mathcal{A}_i(P)$, para $l_k \leq i < l_{k+1}$. Podemos construir la sucesión de estados

$$A_1' \prec A_2' \prec \cdots A_k' \prec A_{k+1}' \prec \cdots$$

de manera que

$$A_i' = \begin{cases} A_{l_k} & \text{si } l_k \leq i < l_{k+1} \\ \{\Omega 0\} & \text{si } i < l_1 \end{cases} \qquad \text{y} \qquad A_i' \in \mathcal{A}_i(P)$$

Si la computación es infinita podemos tomar el estado

$$A = \bigcup_{i\in\mathbb{N}} TAct(A_k) = \bigcup_{i\in\mathbb{N}} TAct(A_k')$$

y puesto que la sucesión de instantes $\{t^k \mid k \in \mathbb{N}\}$ no está acotada tenemos que

$$A = \mathsf{lub}\{A_k' \mid k \in \mathbb{N}\}$$

y por tanto $A \in \mathcal{A}(P)$.

Si la computación es finita (P_n' es el último proceso) tomamos el estado

$$A_n = \left(\bigcup_{i\leq n}(\mathsf{TA}(P_i') \upharpoonright t_i) + t^i\right) \cup \begin{cases} \mathsf{TA}(P_n') & \text{si } P_n' \Downarrow \\ \{\Omega t^n\} & \text{si } P_n' \Uparrow \end{cases}$$

Existe entonces $l_n \in \mathbb{N}$ de manera que $A_n \in \mathcal{A}_i(P)$ para $l_n \leq i$, entonces tomamos para $l_n \leq i$ el estado $A_i' = A_n$. Tenemos entonces que

$$A = \mathsf{lub}\{A_i' \mid i \in \mathbb{N}\}$$

por lo que $A \in \mathcal{A}(P)$.

□

Como consecuencia de esta proposición tenemos el

Corolario 3.4.6 Sea P tal que $P \Downarrow$, entonces

- Si $P \Downarrow$, $P \rightarrowtail P'$ y $A \in \mathcal{A}(P')$ entonces $A \in \mathcal{A}(P)$.
- Si $P \Downarrow$, $P \xrightarrow{\tau t} P'$ y $A \in \mathcal{A}(P')$, entonces $\mathsf{TA}(P) \upharpoonright t \cup (A + t) \in \mathcal{A}(P)$.

La siguiente proposición nos relaciona el conjunto de estados de un cierto proceso P con los de su actualización temporal $\mathsf{Upd}(P, t)$. Debido a la manera en la que está definida la semántica operacional, sólo nos interesa en realidad el caso en el que el proceso P sea estable, y no pueda hacer ninguna acción interna en un instante menor que t. Por otro lado no tiene sentido que intentemos actualizar un proceso que sea divergente. Todo ello queda reflejado en la

Proposición 3.4.7 Sea P un proceso estable verificando $P \Downarrow$, y $t \in \mathcal{T}$ con $\mathsf{idle}(P) \geq t$, se tiene entonces

$$A \in \mathcal{A}(\mathsf{Upd}(P, t)) \quad \Longleftrightarrow \quad (\mathsf{TA}(P) \upharpoonright t) \cup (A + t) \in \mathcal{A}(P)$$

Demostración. Por el corolario 3.2.3 se tiene

$$\mathsf{TA}(P) = \Big(\mathsf{TA}(\mathsf{Upd}(P, t)) + t\Big) \cup (\mathsf{TA}(P) \upharpoonright t)$$

Además por la propiedad 3.2.2 tenemos $\mathsf{stb}(\mathsf{Upd}(P, t))$ y

$$P \xrightarrow{et'} P' \quad \Longleftrightarrow \quad \mathsf{Upd}(P, t) \xrightarrow{t'-t} P' \quad t' \geq t$$

A partir de lo cual basta tener en cuenta la proposición 3.4.5.

□

Finalmente y recordando la definición 2.4.6, tenemos que el conjunto de estados de un proceso verifica la

Proposición 3.4.8 Para cada proceso P, su conjunto de estados $\mathcal{A}(P)$ es temporalmente compacto.

Demostración. Tomemos $\mathcal{A} = \mathcal{A}(P)$ y supongamos que existe A con $\mathsf{nd}(A) = \infty$ tal que para todo $t \in \mathcal{T}$ existe $A_t \in \mathcal{A}$ tal que $A_t \upharpoonright t = A \upharpoonright t$.

Si el conjunto $\{A_t \mid t \in \mathcal{T}\}$ es finito existe $A' \in \mathcal{A}$ tal que $A' \upharpoonright t = A \upharpoonright t$ para todo $t \in \mathcal{T}$, y por tanto tendremos que $A' = A$.

Supongamos entonces que el conjunto $\{A_t \mid t \in \mathcal{T}\}$ es infinito. Tenemos que $\mathsf{nd}(A_t) \geq t$, pero sin pérdida de generalidad podemos suponer que $\mathsf{nd}(A_t) > t$. Entonces, considerando la computación que genera el estado A_t

$$P = P_1 \rightarrowtail^* P_1 \xrightarrow{\tau t_1} P_2 \rightarrowtail^* P_2' \xrightarrow{\tau t_2} \cdots P_{k-1}' \xrightarrow{\tau t_{k-1}} P_k \rightarrowtail^* P_k' \cdots$$

Tomando

y tomando $t^i = \sum_{j=1}^{i-1} t_j$, tenemos que puesto que $\mathsf{nd}(A_t) > t$ existe k tal que $t^k \leq t$, y si $P'_k \xrightarrow{\tau t_k}$ entonces $t^{k+1} > t$. Llamemos P_t al proceso P'_k correspondiente y tenemos

$$A_t \upharpoonright t = ((\mathsf{TA}(P_t) + t^k) \upharpoonright t) \cup \bigcup_{i=1}^{k-1} (\mathsf{TA}(P'_i) + t^i)$$

El conjunto $\{P_t \mid t \in \mathcal{T}\}$ es infinito, por lo que aplicando la proposición 3.2.5 y el lema de König, concluimos que existe una computación

$$P = P_1 \succ\!\!\longrightarrow^* P_1 \xrightarrow{\tau t_1} P_2 \succ\!\!\longrightarrow^* P'_2 \xrightarrow{\tau t_2} \cdots P'_{k-1} \xrightarrow{\tau t_{k-1}} P_k \succ\!\!\longrightarrow^* P'_k \xrightarrow{\tau t_k} \cdots$$

en la que aparecen un número infinito de los P_t, por lo cual obtenemos

$$A = \bigcup_{i \in \mathbb{N}} A_i + t^i \quad \text{y} \quad A \in \mathcal{A}(P)$$

□

A la vista de esta propiedad tenemos que se verificará el siguiente

Corolario 3.4.9 Para cada $P \in \mathsf{CRec}(\Sigma_{\mathrm{seq}})$, se tiene $A \in \mathcal{A}(P)$ sii se verifica alguna de las condiciones siguientes:

- $\mathsf{nd}(A) < \infty$ y $\forall k \in \mathbb{N}\ \exists l \geq k :\ A \in \mathcal{A}_k(P)$.
- $\mathsf{nd}(A) = \infty$ y $\forall t \in \mathcal{T}\ \exists l \in \mathbb{N},\ A_t \in \mathcal{A}_k(P) :\ A \upharpoonright t = A_t \upharpoonright t$.

Demostración.

$\Longrightarrow$ Es obvio por la definición de $\mathcal{A}(P)$.

$\Longleftarrow$ Se deduce de la ramificación finita de la semántica operacional de P y de la proposición anterior.

□

3.5 Barbas de un Proceso

En el capítulo 2 dijimos que una barba era una generalización adecuada de la noción de aceptación. En el caso no temporizado decimos que $s \cdot A$ es una aceptación de P si existe un proceso P' tal que $P \xrightarrow{s} P'$ y A es un estado de P'. En nuestro modelo temporizado procedemos de la manera análoga, pero en este caso con b-trazas.

Definición 3.5.1 Definimos la relación $\Longrightarrow \subseteq \mathsf{CRec}(\Sigma_{\text{seq}}) \times \mathcal{BS} \times \mathsf{CRec}(\Sigma_{\text{seq}})$ como la menor relación que verifica:

- $P \overset{\epsilon}{\Longrightarrow} P$.
- Si $P \rightarrowtail P' \overset{bs}{\Longrightarrow} P_1$ entonces $P \overset{bs}{\Longrightarrow} P_1$.
- Si $P \xrightarrow{\tau t} P' \xrightarrow{bs} P_1$ y $bs \neq \epsilon$, entonces
$$P \overset{bs'}{\Longrightarrow} P_1 \quad \text{donde} \quad bs' = (\mathsf{TA}(P) \upharpoonright t, t) \sqcup (bs + t)$$
- Si $P \xrightarrow{at} P' \overset{bs}{\Longrightarrow} P_1$ entonces
$$P \overset{bs'}{\Longrightarrow} P_1 \quad \text{donde} \quad bs' = (\mathsf{TA}(P) \upharpoonright t)at \cdot bs$$

□

Claramente las b-trazas y los estados están fuertemente relacionados entre sí. En concreto se cumple el siguiente

Lema 3.5.2 Siendo P un proceso se verifica:

- Si $A \in \mathcal{A}(P)$ y $at \in A$ entonces existe un proceso P' tal que
$$P \overset{(A \upharpoonright t)at}{\Longrightarrow} P'$$
- Si $P \overset{Aat}{\Longrightarrow} P'$ entonces existe un estado $A' \in \mathcal{A}(P)$ tal que $TAct(A') \upharpoonright t \subseteq A$.

Demostración. Para demostrar el primer punto tomemos $A \in \mathcal{A}(P)$ y $at \in A$. Consideremos entonces la computación que genera el estado A

$$P = P_1 \rightarrowtail^* P'_1 \xrightarrow{\tau t_1} P_2 \rightarrowtail^* P'_2 \xrightarrow{\tau t_2} \cdots$$

Tomando $t^i = \sum_{j=1}^{i-1} t_j$, existe entonces $k \in \mathbb{N}$ tal que $a(t - t^k) \in \mathsf{TA}(P'_k) \upharpoonright t_k$, de modo que existirá P' tal que

$$P'_k \xrightarrow{a(t-t^k)} P',$$

con lo que tenemos $P \overset{(A \upharpoonright t)at}{\Longrightarrow} P'$.

Supongamos ahora que $P \overset{Aat}{\Longrightarrow} P'$. En tal caso existirá una computación

$$P = P_1 \rightarrowtail^* P'_1 \xrightarrow{\tau t_1} P_2 \rightarrowtail^* P'_2 \xrightarrow{\tau t_2} \cdots P_k \rightarrowtail^* P'_k \xrightarrow{at_k} P'$$

tomemos $t^i = \sum_{j=1}^{i-1} t_j$. Si $P_i \Downarrow$ para todo $i \leq k$ entonces tomando cualquier $A_k \in \mathcal{A}(P'_k)$ se tiene inmediatamente que el estado $A' = (A_k + t^k) \cup A$ verifica las propiedades requeridas. Si existe $i \leq k$ tal que $P_i \Uparrow$ (tomando el menor que lo verifique) se tiene que $A' = \{\Omega t^i\} \cup A \upharpoonright t^i$ es el estado buscado.

□

una

Seguidamente definimos las barbas de un proceso

Definición 3.5.3 Decimos que $b = bs \cdot A$ es una barba de un proceso P, y lo escribiremos $b \in \mathsf{Barb}(P)$, si y sólo si existe un proceso P' tal que

$$P \stackrel{bs}{\Longrightarrow} P' \qquad \text{y} \qquad A \in \mathcal{A}(P')$$

□

A partir de esta última definición y utilizando la relación $\ll$ sobre conjuntos de barbas introducida en el capítulo 2, obtenemos la relación entre procesos que nos servirá para caracterizar la semántica de pruebas.

Definición 3.5.4 Siendo $P, Q \in \mathsf{CRec}(\Sigma_{\mathrm{seq}})$, decimos que

$$P \ll Q \quad \Longleftrightarrow \quad \mathsf{Barb}(P) \ll \mathsf{Barb}(Q)$$

□

3.6 Equivalencia entre la Semántica de Pruebas y la Caracterización Operacional.

En esta sección demostraremos la equivalencia entre los dos tipos de relaciones entre procesos que hemos visto: por un lado la semántica de pruebas $\underset{\sim}{\sqsubset}$ y por otro lado la relación inducida por las barbas $\ll$.

En primer lugar demostraremos

$$P \ll Q \;\Rightarrow\; P \underset{\sim}{\sqsubset} Q$$

Para ello precisamos una serie de resultados previos. En primer lugar veremos una proposición similar a la proposición 3.2.3, que nos relaciona las b-trazas que un proceso puede ejecutar, y las b-trazas que puede ejecutar una actualización temporal suya. Puesto que únicamente hacemos la actualización temporal de procesos que se *dejan*, es decir, procesos estables y procesos que pueden permanecer inactivos; sólo estamos interesados en la actualización de tales procesos.

Lema 3.6.1 Sea $P \in \mathsf{CRec}(\Sigma_{\mathrm{seq}})$ verificando $\mathsf{stb}(P)$, $t \in \mathcal{T}$ con $\mathsf{idle}(P) \geq t$ y $bs \neq \epsilon$; entonces

$$\mathsf{Upd}(P,t) \stackrel{bs}{\Longrightarrow} P' \quad \Longleftrightarrow \quad P \xrightarrow{(\mathsf{TA}(P)\upharpoonright t,t)\sqcup bs} P'$$

Demostración. Es una consecuencia inmediata del corolario 3.2.3.

□

Las demostraciones que vienen a continuación son bastante rutinarias, así que para facilitar la lectura, hemos preferido dar una idea de las demostraciones y colocar la mismas en el apéndice A.

Proposición 3.6.2 Sean $bs = A_1a_1t_1 \cdots A_na_nt_n$ y $bs' = A'_1a_1t_1 \cdots A'_na_nt_n$ con $n > 0$, tal que $A_i \cap A'_i = \varnothing$, T una prueba tal que $T \overset{bs}{\Longrightarrow} T_1$ y P un proceso tal que $P \overset{bs'}{\Longrightarrow} P_1$, entonces existe una computación desde $P \mid T$ hasta $P_1 \mid T_1$.

Demostración. Para entender esta proposición basta tener en cuenta que puesto que los conjuntos $A_i \cap A'_i = \varnothing$, por lo que el proceso y la prueba no pueden sincronizar en acciones distintas a las a_i en el instante t_i. Así que la prueba y el proceso van evolucionando de manera independiente sin posibilidad de sincronizar en ninguna acción, hasta que el proceso y la prueba se *ponen de acuerdo* para ejecutar conjuntamente la acción a_i en instante t_i. Nótese que si en algún momento se tiene que $A_i \cap A'_i \neq \varnothing$, debido a la urgencia, el proceso y la prueba deberían sincronizar (salvo en el caso que alguno de los dos realizase una acción interna en el mismo instante) y entonces la computación no sería posible.

□

La proposición anterior refleja la necesidad de *guardar* la información sobre las acciones que un proceso puede ejecutar *antes* que una determinada acción. Las proposiciones que vienen a continuación son similares a las anteriores, pero para estados. A continuación demostraremos que si un proceso *está* en un estado en el que no puede sincronizar con las acciones que ofrece una prueba, $(A \cap (\mathsf{TA}(T'_i) \upharpoonright t_i + t^i) = \varnothing)$, ambos procesos pueden evolucionar libremente, al menos hasta el instante de indefinición del estado, $(\mathsf{nd}(A) > t_k)$.

Proposición 3.6.3 Sea $A \in \mathcal{A}(P)$ y T una prueba que tiene una computación

$$T = T_0 \rightarrowtail^* T'_0 \xrightarrow{\tau t_1} T_1 \rightarrowtail^* T'_1 \xrightarrow{\tau t_2} T_2 \rightarrowtail^* T'_2 \xrightarrow{\tau t_3} \cdots \xrightarrow{\tau t_k} T_k$$

de modo que, tomando $t^i = \sum_{j=1}^{i-1} t_i$, se verifica:

- $A \cap (\mathsf{TA}(T'_i) \upharpoonright t_i + t^i) = \varnothing$ y
- $\mathsf{nd}(A) > t^k$.

Entonces existe un proceso P_k para el cual se tiene una computación desde $P \mid T$ hasta $P_k \mid T_k$ en la que aparecen todas las pruebas indicadas en la computación de arriba.

---

a

Demostración. La razón por la cual la computación es posible es la misma que en la proposición anterior.

□

Proposición 3.6.4 Sean $A \in \mathcal{A}(P)$, y T una prueba que tiene una computación

$$T = T_1 \rightarrowtail^* T_1' \xrightarrow{\tau t_1} T_2 \rightarrowtail^* T_2' \xrightarrow{\tau t_2} \cdots T_k \rightarrowtail^* T_k'$$

de modo que, tomando $t^i = \sum_{j=1}^{i-1} t_j$, cumple:

- $A \cap \left(\mathsf{TA}(T_i') \upharpoonright t_i + t^i\right) = \varnothing$,
- existe k tal que $\mathsf{nd}(A) \leq t^k$ y $\mathsf{nd}(A) > t^l$ para $l < k$.

Entonces existen P' y T' tales que hay una computación desde $P \mid T$ hasta $P' \mid T'$ con $P' \Uparrow$, y de modo que a lo largo de dicha computación aparecen todas las pruebas de la computación T en el enunciado.

Demostración. Consideremos la computación que genera el estado A:

$$P = P_1 \rightarrowtail^* P_1' \xrightarrow{\tau t_1^P} \cdots P_{n-1} \rightarrowtail^* P_{n-1}' \xrightarrow{\tau t_n^P} P_n$$

Puesto que $\mathsf{nd}(A) < \infty$, la computación es finita y $P_n \Uparrow$. En virtud de la propiedad anterior y la propiedad 3.4.7 podemos suponer $k = 1$; tomemos entonces $t^{i,P} = \sum_{j=1}^{i-1} t_j^P$ y $T = T_1'$. Puesto que $T_1 \rightarrowtail^* T$ tenemos que $P_1 \mid T_1 \rightarrowtail^* P_1 \mid T$. Además, puesto que $A \cap (\mathsf{TA}(T) \upharpoonright t_1) = \varnothing$ tendremos

$$(\mathsf{TA}(P_i') \cap \mathsf{Upd}(T, t^{i,P})) \upharpoonright t_i^P = \varnothing$$

En consecuencia tenemos que la siguiente computación es posible:

$$\begin{array}{l} P_1 \mid T \mapsto^* P_1' \mid T \mapsto P_2 \mid \mathsf{Upd}(T, t^{2,P}) \cdots \\ \cdots P_{n-1} \mid \mathsf{Upd}(T, t^{n-1,P}) \mapsto^* P_{n-1}' \mid \mathsf{Upd}(T, t^{n-1,P}) \mapsto P_n \mid \mathsf{Upd}(T, t^{n-1,P}) \end{array}$$

Y puesto que $P_n \Uparrow$, la proposición queda demostrada.

□

Estamos ya en condiciones de demostrar la primera parte de la equivalencia.

Teorema 3.6.5 Si $P \ll Q$ entonces $P \sqsubseteq Q$.

Demostración. Para probar $P \sqsubseteq Q$ consideremos cada prueba T tal que P must T y demostremos que Q must T. Para ello debemos analizar cada computación completa de $Q \mid T$. Cada una de tales computaciones proviene de una computación de Q y otra de T. Ambas computaciones van dando lugar a una serie de b-trazas

$$Q \xRightarrow{A_1 a_1 t_1 \cdots A_k a_k t_k} Q_k, \qquad T \xRightarrow{A'_1 a_1 t_1 \cdots A'_k a_k t_k} T_k$$

Tomemos $bs_k = A_1 a_1 t_1 \cdots A_k a_k t_k$ y $bs'_k = A'_1 a_1 t_1 \cdots A'_k a_k t_k$. Puesto que las b-trazas vienen de una computación de $P \mid T$ se verificará que $A_i \cap A'_i = \varnothing$. Tomemos entonces un estado cualquiera de Q_k, A^Q_{k+1}. Tendremos entonces que

$$b_k = bs_k A^Q_{k+1} \in \mathsf{Barb}(Q)$$

Puesto que $P \ll Q$, para cada una de tales barbas, existirá una barba

$$b''_k = bs''_k \cdot A^P_{k+1} \in \mathsf{Barb}(P)$$

tal que $b''_k \ll b_k$. Si en algún momento se cumple que $\mathsf{lon}(b''_k) < \mathsf{lon}(b_k)$, por la proposiciones 3.6.2 y 3.6.4, teniendo en cuenta que P must T, podremos obtener fácilmente que la computación de $Q \mid T$ es de éxito.

Supongamos entonces que para cada $k \in \mathbb{N}$ se verifica $\mathsf{lon}(b''_k) = \mathsf{lon}(b_k)$. Si el número de b-trazas que surgen de la computación de $Q \mid T$ es infinito, por la proposición 3.6.2, teniendo en cuenta que la semántica operacional es finitamente ramificada, podemos aplicar el lema de König, para obtener que la computación de $Q \mid T$ es de éxito. Supongamos entonces que se generan un número finito de b-trazas, siendo la última bs_n. Existen entonces una prueba T_n y un proceso Q_n tales que

$$T \xRightarrow{bs'_n} T_n, \quad \text{y} \quad Q \xRightarrow{bs_n} Q_n$$

Ahora la computación que termina en $Q_n \mid T_n$ va a generar un estado A^Q de Q_n tal que la intersección con las acciones que va ofreciendo la prueba es vacía. Puesto que $P \ll Q$ existirá un proceso P_n tal que

$$P \xRightarrow{bs''_n} P_n \quad A^P \in \mathcal{A}(P_n) \quad \text{y} \quad bs''_n \cdot A^P \ll bs_n \cdot A^Q$$

Se puede ver entonces que la computación de $Q|T$ es de éxito gracias a que P must T, a las proposiciones 3.6.2, 3.6.3 y 3.6.4, a que la semántica operacional es finitamente ramificada y el lema de König.

□

la

A continuación hemos de probar la otra parte de la equivalencia

$$P \sqsubseteq\!\!\!\sim Q \;\Rightarrow\; P \ll Q$$

Lo haremos por reducción al absurdo; es decir, demostraremos

$$P \not\ll Q \;\Rightarrow\; P \not\sqsubseteq\!\!\!\sim Q$$

Por tanto si existe $b \in \mathsf{Barb}(Q)$ de modo que no existe $b' \in \mathsf{Barb}(P)$ con $b' \ll b$, entonces podemos encontrar una prueba T tal que P must T y Q ~~must~~ T. Para ilustrar la forma en que encontraremos una prueba tal, veamos el siguiente

Ejemplo 3.6.6 Consideremos los procesos

$$P = (\mathsf{STOP}) \sqcap (a1\,;\mathsf{STOP} \,\Box\, b2\,;\mathsf{STOP})$$

y

$$Q = (\mathsf{STOP}) \sqcap (a1\,;\mathsf{STOP} \,\Box\, b2\,;\mathsf{STOP}) \sqcap (b2\,;\mathsf{STOP})$$

Sus respectivos conjuntos de barbas vienen dados por

$$\mathsf{Barb}(P) = \left\{ \begin{array}{l} \varnothing,\ \{a1, b2\}, \\ \varnothing a1 \varnothing,\ \{a1\}b2\varnothing \end{array} \right\} \qquad \mathsf{Barb}(Q) = \left\{ \begin{array}{l} \varnothing,\ \{b2\}, \{a1, b2\}, \\ \varnothing a1 \varnothing,\ \varnothing b2 \varnothing,\ \{a1\}b2\varnothing \end{array} \right\}$$

Tenemos que $Q \ll P$, pero considerando $b = \varnothing b_2 \varnothing \in \mathsf{Barb}(Q)$ nos encontramos con que no existe $b' \in \mathsf{Barb}(P)$ tal que $b \ll b'$, de modo que $P \not\ll Q$. Veamos porqué ello es así. En el conjunto $\mathsf{Barb}(P)$, la oferta de la acción b en instante 2 siempre está *acompañada* por la de a en el instante 1. Si P pudiera sincronizar con una prueba que pudiera ejecutar la acción a en el instante 1 estarían obligados, ambos prueba y proceso, a ejecutarla. Con lo cual b en el instante 2 no se podría ejecutar. En cambio en el proceso Q, si la prueba está *interesada* en ejecutar b en el instante 2, Q no tendría ningún motivo para negarse. Así que la prueba que pase P pero no Q tendría que ofrecer a en el instante 1 y después dar el visto bueno. En concreto, consideremos la prueba

$$T = (a1\,;\mathsf{OK}) \,\Box\, (\tau 2\,;\mathsf{OK}) \,\Box\, (b2\,;\mathsf{STOP})$$

Es fácil comprobar que en efecto P must T y Q ~~must~~ T, con lo cual quedaría probado $P \not\sqsubseteq\!\!\!\sim Q$.

$\Box$

Antes de continuar, conviene introducir cierta notación adicional con el objetivo de simplificar algo las cosas. Si $A = \{a_1t_1, \ldots, a_nt_n\}$ es un conjunto finito de acciones temporizadas, por medio de

$$[A]\,;\mathtt{OK}$$

denotaremos la prueba

$$a_1t_1\,;\mathtt{OK}\,\square\,(a_2\,;\mathtt{OK}\,\square\,(\cdots(a_{n-1}t_{n-1}\,;\mathtt{OK}\,\square\,a_nt_n\,;\mathtt{OK})\cdots))$$

Obsérvese que en dicha prueba no tiene influencia del orden en el que se hayan colocado las acciones temporizadas pues tenemos

$$[A]\,;\mathtt{OK} \xrightarrow{at} \mathtt{OK} \qquad \forall at \in A$$

Si el conjunto de acciones temporizadas es vacío tomaremos

$$[\varnothing]\,;\mathtt{OK} = \mathtt{STOP}$$

Vamos a generalizar lo obtenido en el ejemplo anterior, con el objetivo de encontrar una prueba de manera que P must T y Q m~~u~~st T para procesos tales que $P \not\ll Q$. Si $P \not\ll Q$, existe una barba $b \in \mathsf{Barb}(Q)$ tal que *no existe* $b' \in \mathsf{Barb}(P)$ *cumpliendo que* $b' \ll b$. Empezamos dando una definición general: definiremos una prueba en función de un conjunto de barbas y una barba, que serán después el conjunto de barbas de un proceso ($\mathsf{Barb}(P)$ en el caso anterior) y una barba de un proceso ($b \in \mathsf{Barb}(Q)$). Más adelante veremos que dicha prueba es en efecto la que estamos buscando.

Definición 3.6.7 Sea B un conjunto de barbas y b una barba, entonces

- Si la barba b es un estado A, consideremos

$$T_1 = \begin{cases} \tau t\,;\mathtt{OK} & \text{si } t = \mathsf{nd}(A) < \infty \\ \mathtt{STOP} & \text{en otro caso} \end{cases}$$

  Y siendo $A' \subseteqq TAct$ un conjunto finito que verifique

  - $A' \cap A = \varnothing$,
  - $A' \cap A'' \neq \varnothing$ para todo $A'' \in \mathcal{A}(B)$ tal que $\mathsf{nd}(A'') \leq \mathsf{nd}(A)$,

  tomamos

$$T_2 = [A']\,;\mathtt{OK}$$

  Decimos entonces que $T = T_1 \,\square\, T_2$ es una prueba bien formada con respecto a B y b.

semana,

- Si $b = Aat \cdot b_1$, tomamos el conjunto de barbas

$$B_1 = \{b' \mid \exists\, A' \subseteq A \text{ t.q. } A'at \cdot b' \in B\}$$

Si $B_1 = \varnothing$ consideremos $T_1 = \texttt{STOP}$, en caso contrario consideremos T_1 una prueba bien formada con respecto a B_1 y b_1. Por otra parte tomemos un conjunto finito $A' \subseteqq TAct$ tal que $A' \cap A = \varnothing$, $A' \cap A'' \neq \varnothing$ para todo A'' cumpliendo una de las siguientes condiciones:

  - Existe una barba b'' tal que $A''at \cdot b'' \in B$ y $b'' \ll b_1$, o bien,
  - $A'' \in B$ y $\mathsf{nd}(A'') \leq t$.

Decimos entonces que la prueba

$$T = (\tau t\, ; \texttt{OK}) \,\Box\, (at\, ; T_1) \,\Box\, [A']\, ; \texttt{OK}$$

está bien formada con respecto a B y b.

□

Para un conjunto arbitrario general de barbas B y una barba cualquiera b, es posible que no exista ninguna prueba bien formada con respecto a B y b.

Ejemplo 3.6.8 Tomemos

$$B = \left\{ \begin{array}{c} \{at\}, \\ \varnothing at \varnothing \end{array} \middle|\, t \in \mathbb{N} \right\} \qquad \text{y} \qquad b = \varnothing$$

Es claro que hay ningún conjunto finito $A' \subseteqq TAct$ que interseque a todos los estados de B, no existe $b' \in B$ tal que $b' \ll b$, y $\mathsf{Barb}(\texttt{STOP}) = \{b\}$. Si damos por bueno que dichas pruebas serán las que servirán para distinguir procesos no relacionados por $\ll$ tenemos que si existiera P tal que $B = \mathsf{Barb}(P)$, para encontrar una prueba T tal que P must T y STOP m~~u~~st T tendría que ser de la forma

$$T = \mathop{\Box}_{t \in \mathbb{N}} at\, ; \texttt{OK}$$

Pero dicha prueba no se puede construir con nuestra sintaxis puesto que se trata de una elección infinita.

□

Sin embargo, debido a la ramificación finita de los procesos sintácticos, no hay conjuntos $\mathsf{Barb}(P)$ con el mal comportamiento del conjunto B del ejemplo anterior. De modo que si $B = \mathsf{Barb}(P)$, $b \in \mathsf{Barb}(Q)$ y no existe $b' \in B$ tal que $b' \ll b$, entonces existe siempre una prueba T bien formada con respecto a $\mathsf{Barb}(P)$ y b. De cara a probarlo damos en primer lugar el siguiente

Lema 3.6.9 Supongamos que $\mathcal{A}$ es un conjunto de estados temporalmente completo y $A \subseteq TAct$ un conjunto de acciones temporizadas tal que todo estado $A' \in \mathcal{A}$ verifica $A' \cap A \neq \varnothing$. Entonces existe un conjunto finito $A_0 \subseteq A$ tal que

$$\forall A' \in \mathcal{A} \quad A' \cap A_0 \neq \varnothing$$

Demostración. Razonemos por inducción al absurdo, y supongamos que existe un conjunto de acciones temporizadas A tal que para cada subconjunto finito $A_0 \subseteq A$ existe un estado $A_0' \in \mathcal{A}$ tal que $A_0 \cap A_0' = \varnothing$. Fijemos $t \in \mathcal{T}$, puesto que el conjunto $A \rceil t$ es finito existe $A_t \in \mathcal{A}$ tal que $A \rceil t \cap A_t = \varnothing$. Además, puesto que el dominio de tiempo es discreto y el alfabeto finito, podemos escoger los A_t verificando

$$A_t \rceil t = A_{t'} \rceil t \quad \text{para} \quad t' \geq t$$

Tomemos entonces el estado

$$A' = \bigcup_{t \in \mathcal{T}} A_t \rceil t$$

y tendremos que, por ser $\mathcal{A}$ temporalmente compacto, $A' \in \mathcal{A}$, y puesto que $A_t \cap A \rceil t = \varnothing$, tenemos $A' \cap A = \varnothing$ lo que contradice nuestra hipótesis.

□

Proposición 3.6.10 Si $B = \mathsf{Barb}(P)$ y b es una barba tal que no existe $b' \in B$ tal que $b' \ll b$, entonces existe una prueba T bien formada respecto a B y b.

Demostración. Sea $b = A_1 a_1 t_1 \cdots A_n a_n t_n A$. Razonemos por inducción sobre n.

n = 0 Tomemos $t = \mathsf{nd}(A)$ y $\mathcal{A} = \mathcal{A}(P)$. Puesto que no existe $A' \in \mathcal{A}$ tal que $A' \ll A$, para cada $A' \in \mathcal{A}$ tal que $\mathsf{nd}(A') \leq t$ se verifica que $TAct(A') \setminus A \neq \varnothing$. Consideremos el conjunto de acciones temporizadas

$$A_1 = \bigcup_{\substack{\mathsf{nd}(A') \leq t \\ A' \in \mathcal{A}}} TAct(A') \setminus A$$

Si $t < \infty$ tenemos que A_1 es finito. Tomaremos entonces $A_0 = A_1$.

Si $t = \infty$ tenemos

- Por ser P un proceso, $\mathcal{A} = \mathcal{A}(P)$ es temporalmente compacto.
- Para cada $A' \in \mathcal{A}$ se tiene $A' \cap A_1 \neq \varnothing$.

Entonces por aplicación del lema anterior tenemos que existe $A_0 \subseteq A_1$ finito tal que $A_0 \cap A' \neq \varnothing$ para todo $A' \in \mathcal{A}$.

Tomamos entonces las pruebas

$$T_1 = \begin{cases} \tau t \,;\mathtt{OK} & \text{si } t \leq \infty \\ \mathtt{STOP} & \text{en otro caso} \end{cases} \qquad T_2 = [A_0]\,;\mathtt{OK}$$

y tenemos que $T = T_1 \,\Box\, T_2$ es en efecto una prueba bien formada con respecto a B y a b.

n > 0 Supongamos ahora que $b = Aat \cdot b_1$. Por la propiedad 3.2.5, existe un número finito de procesos Q tales que $P \stackrel{A'at}{\Longrightarrow} Q$. Consideremos el conjunto

$$\mathcal{Q} = \{Q \mid P \stackrel{A'at}{\Longrightarrow} Q \text{ y } A' \subseteq A\}$$

Si $\mathcal{Q}$ no fuera finito, por la proposición 3.2.5, podríamos encontrar fácilmente un estado $A' \in \mathcal{A}(P)$ tal que $\mathsf{nd}(A') \leq t$ y $A' \upharpoonright t \subseteq A$; con lo cual se verificaría $A' \ll b$, que contradice la hipótesis de partida. Por tanto $\mathcal{Q}$ es un conjunto finito. Si $\mathcal{Q} = \varnothing$, basta tomar $T' = \mathtt{STOP}$, en caso contrario podemos considerar el proceso

$$P' = \prod_{Q \in \mathcal{Q}} Q$$

verificándose

$$B_1 = \mathsf{Barb}(P') = \{b' \mid \exists\, A' \subseteq A \text{ t.q. } A'at \cdot b' \in \mathsf{Barb}(P)\}$$

Puesto que no existe $b_P \in \mathsf{Barb}(P)$ tal que $b_P \ll b$, no tampoco existe $b_{P'} \in \mathsf{Barb}(P')$ tal que $b_{P'} \ll b_1$. Por tanto, por hipótesis de inducción existe una prueba bien formada T' con respecto a B_1 y b_1.

Por otro lado se cumple:

- Si $b'' \ll b_1$ y $b'' = A'at \cdot b'$ entonces $A' \setminus A \neq \varnothing$.
- Si $A' \in \mathsf{Barb}(P)$ es tal que $\mathsf{nd}(A') \leq t$ $\mathit{TAct}(A') \setminus A \neq \varnothing$.

Tomemos entonces el conjunto

$$A_0 = \bigcup \left\{ A' \setminus A \;\middle|\; \begin{array}{c} \exists\, A'' \in \mathsf{Barb}(P) :\ \mathsf{nd}(A'') \leq t \ y\ A' = \mathit{TAct}(A'') \\ \text{ó} \\ \exists\, b' :\ A'at \cdot b' \in \mathsf{Barb}(P) \text{ y } b' \ll b_1 \end{array} \right\}$$

Puesto que estamos considerando un alfabeto finito y un dominio de tiempo discreto tenemos que A_0 es finito. También tenemos que A_0 verifica las condiciones de la definición, de modo que la prueba

$$T = \tau t \,;\, \mathtt{OK} \,\Box\, at \,;\, T' \,\Box\, [A_0] \,;\, \mathtt{OK}$$

está bien formada con respecto a $\mathsf{Barb}(P)$ y a b.

□

Una vez probada la existencia de la prueba bien formada con respecto a $\mathsf{Barb}(P)$ y $b(\in \mathsf{Barb}(Q))$, *hemos de probar que verifica las propiedades requeridas. En primer lugar*

Proposición 3.6.11 Sea B un conjunto de barbas, b una barba tal que no existe $b' \in B$ tal que $b' \ll b$ y T una prueba bien formada con respecto a B y a b. Si $B = \mathsf{Barb}(P)$, entonces P must T.

Demostración. Nos limitaremos a ver aquí un resumen de la demostración, está hecha con todo detalle en el apéndice A. Hemos de demostrar que en cualquier computación de $P \mid T$, acaba dando el visto bueno. Lo probamos por inducción sobre la longitud de la barba b.

b = A. Cualquier estado A' de P verifica

$$\mathsf{nd}(A') \leq \mathsf{nd}(A) \quad \Rightarrow \quad \exists at \in A' : \; at \notin A$$

La prueba T es de la forma

$$T = [A''] \,;\, \mathtt{OK} \,\Box \begin{cases} \tau t \,;\, \mathtt{OK} & \text{si } t = \mathsf{nd}(A) < \infty \\ \mathtt{STOP} & \text{en otro caso} \end{cases}$$

donde el conjunto A'' verifica

$$\forall A' \in \mathcal{A}(P) : \; \mathsf{nd}(A') \leq \mathsf{nd}(A) \; \Rightarrow \; TAct(A') \cap A'' \neq \varnothing$$

Entonces si el proceso evoluciona de manera que genera un estado A' de manera que $A'' \cap A' = \varnothing$ entonces se verifica $\mathsf{nd}(A') > \mathsf{nd}(A)$, y por tanto llegará un momento en el que la prueba ejecute la acción interna (debido a las reglas de la urgencia), conduciéndonos a una computación de éxito. Si P evoluciona hacia un estado A' tal que $A' \leq \mathsf{nd}(A)$, entonces existe $at \in TAct(A') \cap A''$, y por tanto en algún momento se deberá ejecutar la correspondiente sincronización con lo que la prueba dará el visto bueno.

no

$\mathbf{b = Aat \cdot b'}$. En este caso la prueba T es de la forma

$$T = (\tau t\,;\,\mathsf{OK}) \,\Box\, (at\,;T_1) \,\Box\, [A']\,;\mathsf{OK}$$

donde T_1 es una prueba bien formada con respecto al conjunto

$$B_1 = \{b' \mid \exists A' \subseteq A \text{ t.q. } A'at \cdot b' \in B\}$$

y a b'[¶], y A' es un conjunto de acciones temporizadas tal que $A' \cap A'' \neq \varnothing$ para cada conjunto A'' verificando una de las siguientes propiedades

- $A'' \in \mathcal{A}(P)$, $at \notin A''$ y $\mathsf{nd}(A'') < t$.
- $A''at \cdot b'' \in \mathsf{Barb}(P)$ y $b'' \ll b$.

Entonces el proceso evolucionará hacia un estado A'', que verificará o bien $at \notin A''$ o bien $at \in A''$. Si se verifica la primera posibilidad, entonces

- Si $\mathsf{nd}(A) > t$, llegará un instante en el que la prueba T tendrá que hacer la acción interna y dar el visto bueno.
- Si $\mathsf{nd}(A) \leq t$, ha de existir una acción $at \in A'' \cap A'$, por tanto la prueba y el proceso han de sincronizar en dicha acción y con lo que la prueba acaba dando el visto bueno.

Si se verifica que $at \in A''$ tenemos dos posibilidades:

- Existe $a't' \in A'' \cap A'$, entonces el proceso y la prueba sincronizan en la acción y la prueba acaba dando el visto bueno.
- $A''\rceil t \subseteq A$. Entonces existe un proceso P_1 tal que $P \overset{(A''\rceil t)at}{\Longrightarrow} P_1$. Consideremos el proceso

$$Q = \bigsqcap\{P_1 \mid P \overset{A_1 at}{\Longrightarrow} P_1 \text{ y } A_1 \subseteq A\}^{\star}$$

verificándose $\mathsf{Barb}(Q) = B_1$. Entonces, por hipótesis de inducción, Q must T_1. Puesto que $Q \rightarrowtail P'$, cualquier computación de $P_1 \mid T_1$ ha de dar el visto bueno. Puesto que la computación de $P \mid T$ nos conduce hasta $P_1 \mid T_1$, y ésta segunda es de éxito, también lo será la primera.

□

[¶]Si $B_1 = \varnothing$ entonces $T_1 = \mathtt{STOP}$.

[⋆]Si el conjunto de procesos P_1 tales que existe $A_1 \subseteq A$ y $P \overset{A_1 at}{\Longrightarrow} P_1$ no fuera finito, podríamos encontrar un estado $A_1 \in \mathcal{A}(P)$ tal que $\mathsf{nd}(A_1) \leq t$ y $TAct(A_1) \subseteq A$, con lo que llegamos a una contradicción puesto que $A_1 \ll Aat \cdot b'$.

Proposición 3.6.12 Sea B un conjunto de barbas, b una barba tal que no existe $b' \in B$ tal que $b' \ll b$ y T una prueba bien formada con respecto a B y a b. Entonces, si $b \in \mathsf{Barb}(P)$, se tendrá $P \not\text{must}\ T$.

Demostración. Razonamos por inducción sobre la longitud de la barba b:

b = A. Puesto que $A \in \mathsf{Barb}(P)$, en virtud de la proposición 3.4.5 existe una computación

$$P = P_1 \rightarrowtail^* P_1' \xrightarrow[=]{\tau t_1} P_2 \rightarrowtail^* \xrightarrow{\tau t_2} \rightarrowtail^* \cdots$$

que genera dicho estado según la. Tomemos $t^i = \sum_{j=1}^{i-1} t_j$. La prueba T es de la forma

$$T = ([A'] ; \mathtt{OK}) \,\square \begin{cases} \tau t ; \mathtt{OK} & \text{si } t < \infty \\ \mathtt{STOP} & \text{en otro caso} \end{cases}$$

donde $t = \mathsf{nd}(A)$. Supongamos que $t < \infty$, por lo que existe P_k verificando

$$P_k \Uparrow \quad \text{y} \quad t = t^k$$

Puesto que $\mathsf{TA}(P_k') \upharpoonright t_k \cap \mathsf{Upd}(T, t^k) \upharpoonright t_k = \varnothing$ tenemos la computación

$$P \mid T \mapsto^* P_1' \mid T \xrightarrow{\tau t_1} P_2 \mid \mathsf{Upd}(T, t^2) \mapsto^* P_2' \mid \mathsf{Upd}(T, t^2) \mapsto \cdots P_k \mid \mathsf{Upd}(T, t^k)$$

y como quiera que $P_k \Uparrow$ llegamos a $P \not\text{must}\ T$.

Análogamente si, $t = \infty$, puesto que $\mathsf{TA}(P_k') \upharpoonright t_k \cap \mathsf{Upd}(T, t^k) \upharpoonright t_k = \varnothing$, la prueba nunca puede sincronizar con el proceso, y por tanto tenemos que nunca podrá dar el visto bueno, de modo que $P \not\text{must}\ T$.

b = Aat · b₁. En tal caso tenemos que existe P_1 tal que

$$P \xRightarrow{Aat} P_1 \qquad \text{y} \qquad b_1 \in \mathsf{Barb}(P_1)$$

La prueba será de la forma

$$T = ([A'] ; \mathtt{OK}) \,\square\, (at ; T_1) \,\square\, (\tau t ; \mathtt{OK})$$

donde A' es un conjunto de acciones temporizadas verificando $A' \cap A = \varnothing$; y T_1 es una prueba construida a partir del conjunto de barbas

$$B_1 = \{b' \mid \exists A' \subseteq A : \ A'at \cdot b' \in B\}$$

de la siguiente forma:

---

se

- si $B_1 = \varnothing$ entonces $T_1 = \texttt{STOP}$ y evidentemente tendremos $P_1 \ \cancel{\text{must}}\ T_1$,
- si $B_1 \neq \varnothing$ entonces T_1 será una prueba bien formada con respecto a B_1 y a b_1, con lo que por hipótesis de inducción $P_1 \ \cancel{\text{must}}\ T_1$.

Puesto que $T \xRightarrow{A'at} T_1$ y $A' \cap A = \varnothing$, en virtud de la proposición 3.6.2, tenemos que existe una computación desde $P \mid T$ hasta $P_1 \mid T_1$. Por hipótesis de inducción tenemos que $P_1 \ \cancel{\text{must}}\ T_1$, y por tanto $P \ \cancel{\text{must}}\ T$.

□

Finalmente ya podemos demostrar el resultado que buscábamos:

Corolario 3.6.13 Si $P \sqsubseteq Q$, entonces $P \ll Q$.

Demostración. Razonemos por reducción al absurdo, y supongamos por tanto que existe una barba $b \in \mathsf{Barb}(Q)$ tal que no existe $b' \in \mathsf{Barb}(P)$ con $b' \ll b$. En virtud de la proposición 3.6.10 existe una prueba bien formada T con respecto a $\mathsf{Barb}(P)$ y a b. Por la proposición 3.6.11 tenemos que P must T, y aplicando la proposición 3.6.12 tenemos que $Q \ \cancel{\text{must}}\ T$, y lo que $P \not\sqsubseteq Q$, que está en contradicción con nuestras hipótesis.

□

Uniendo este último resultado con el teorema 3.6.5 obtenemos la equivalencia buscada:

Teorema 3.6.14 Si $P, Q \in \mathsf{CRec}(\Sigma_{\text{seq}})$ se tiene

$$P \sqsubseteq Q \qquad \Longleftrightarrow \qquad P \ll Q$$

Capítulo 4

Semántica Denotacional

4.1 Dominio Semántico

En la semántica denotacional se representa cada término sintáctico por medio de un valor en un cierto dominio semántico. En nuestro caso representaremos cada proceso por medio de un *conjunto consistente* de barbas. Tendremos que todo proceso podrá ser representado por un conjunto consistente de barbas, si bien, y como es habitual, el recíproco no será cierto: pueden existir conjuntos consistentes de barbas que no puedan ser generados por ningún proceso sintáctico.

Definición 4.1.1 Un conjunto de barbas B es consistente, lo cual denotamos por $B \in \mathcal{B}_{\text{con}}$, si y sólo si cumple las siguientes condiciones:

- $B \neq \varnothing$.
- *Cerrado bajo prefijo*: Si $A_1 a_1 t_1 \cdots A_{n-1} a_{n-1} t_{n-1} A_n a_n t_n \in \mathsf{Btraz}(B)$, existe un estado A verificando:
  - $A_1 a_1 t_1 \cdots A_{n-1} a_{n-1} t_{n-1} A \in B$ y
  - $A \Uparrow t_n = A_n$.
- *Cerrado bajo continuaciones*: Supongamos que $bs \cdot A \in B$ y $at \in A$, entonces $bs \cdot (A \Uparrow t) at \in \mathsf{Btraz}(B)$.
- *Temporalmente compacto*: Si $bs \in \mathsf{Btraz}(B)$, entonces $\mathcal{A}(B, bs)$ es temporalmente compacto.

□

siente

Exigimos en primer lugar que el conjunto de barbas no sea el vacío. El que los conjuntos deban estar cerrados bajo prefijos está justificado por el hecho de que si una computación ha sido posible, el proceso ha tenido que pasar por todos los estados intermedios. La idea intuitiva tras la segunda condición, *cerrado bajo continuaciones*, es que si después de una computación es posible ejecutar una acción, entonces ha de existir una computaciones que continúan ejecutando tal acción. La última propiedad, *temporalmente compacto*, como ya hemos indicado, está muy relacionada con el hecho de que la semántica operacional sea finitamente ramificada.

Sería deseable que la relación $\ll$ fuera un orden parcial, sin embargo no lo es puesto que no verifica la propiedad antisimétrica. Ello implica inmediatamente que la igualdad entre conjuntos de barbas no puede ser totalmente abstracta con respecto a la equivalencia de pruebas. Sin embargo esto no es totalmente catastrófico pues se se puede probar fácilmente que la relación $\ll$ es un preorden. Entonces para inducir un orden parcial basta considerar la relación de equivalencia

$$B_1 \approx B_2 \quad \Longleftrightarrow \quad B_1 \ll B_2 \wedge B_2 \ll B_1$$

de modo que la relación $\ll$ induce una relación de orden en el conjunto cociente $\mathcal{B}_{\mathrm{comp}} = \mathcal{B}_{\mathrm{con}}/\approx$. Trabajar con el conjunto $\mathcal{B}_{\mathrm{comp}}$ resultaría demasiado engorroso, por lo que es preferible seguir trabajando con el conjunto $\mathcal{B}_{\mathrm{con}}$. A pesar de que a nivel de la igualdad, $\mathcal{B}_{\mathrm{con}}$ no sea totalmente abstracto, demostraremos en cambio que la relación $\ll$ si que lo es, es decir

$$\mathcal{B}_{\mathrm{con}}[\![P]\!] \ll \mathcal{B}_{\mathrm{con}}[\![Q]\!] \quad \Longleftrightarrow \quad P \sqsubseteq\!\!\!\sim Q$$

A partir de ello, se sigue inmediatamente que la igualdad en el conjunto $\mathcal{B}_{\mathrm{comp}}$ es totalmente abstracta con respecto a la semántica de pruebas.

Vemos a continuación una serie de propiedades que resultarán útiles de cara a demostraciones postreras:

Proposición 4.1.2 Si B es un conjunto consistente de barbas y $bs \in \mathsf{Btraz}(B)$, entonces $\mathsf{Barb}(B, bs)$ es un conjunto consistente de barbas.

Demostración. Es fácil comprobar que el conjunto $\mathsf{Barb}(B, bs)$ en efecto cumple las condiciones de la definición 4.1.1.

□

Proposición 4.1.3 Supongamos que B es un conjunto consistente de barbas, A un estado tal que $\mathsf{nd}(A) = \infty$, bs una b-traza tal que para todo $t \in \mathcal{T}$ existe un estado A_t de manera que $bs \cdot A_t \in B$, $A \upharpoonright t \subsetneq A_t$ (resp. $A_t \upharpoonright t \subseteq A_t$) Entonces existe un estado A' tal que $bs \cdot A \in B$ y $A \subseteq A'$ (resp. $A' \subsetneq A$).

Demostración. Puesto que B es completo y $bs \in \mathsf{Btraz}(B)$, tenemos que el conjunto de estados $\mathcal{A}(B, bs)$ es temporalmente compacto. Se obtiene entonces el resultado deseado como consecuencia inmediata de la proposición 2.4.8.

□

4.2 Operadores

En esta sección definiremos el valor semántico de cada uno de los operadores sintácticos de la signatura Σ_{seq}.

4.2.1 Interbloqueo y Divergencia

Tanto el operador de interbloqueo STOP, como el de divergencia DIV, son valores constantes. Para definir su semántica analicemos sus semánticas operacionales y de pruebas.

El operador de interbloqueo, STOP, no es capaz de realizar ningún evento, pero deja que los demás procesos sigan funcionando, Por ello su valor en términos de barbas será:

$$\mathcal{B}_{\text{con}}[\![\mathtt{STOP}]\!] = \{\varnothing\}$$

El operador de divergencia, DIV, al igual que el de interbloqueo, no realiza ningún evento, pero tampoco deja que los demás progresen. Por tanto tendrá un único estado completamente indefinido:

$$\mathcal{B}_{\text{con}}[\![\mathtt{DIV}]\!] = \Big\{\{\Omega 0\}\Big\}$$

Tanto $\mathcal{B}_{\text{con}}[\![\mathtt{STOP}]\!]$ como $\mathcal{B}_{\text{con}}[\![\mathtt{DIV}]\!]$ son conjuntos de barbas consistentes. Además tenemos la siguiente

Proposición 4.2.1

$$\mathsf{Barb}(\mathtt{STOP}) = \{\varnothing\} = \mathcal{B}_{\text{con}}[\![\mathtt{STOP}]\!] \quad \text{y} \quad \mathsf{Barb}(\mathtt{DIV}) = \Big\{\{\Omega 0\}\Big\} = \mathcal{B}_{\text{con}}[\![\mathtt{DIV}]\!]$$

Con lo que la semántica denotacional de estos operadores es acorde con su semántica de pruebas.

necesidad

4.2.2 Prefijo de Acción Oculta

Estudiaremos ahora el caso del prefijo cuando el evento que prefija es la acción oculta: $\tau t;P$. El único efecto *visible* de una acción oculta es el de retrasar la ejecución de las demás acciones. Por tanto, si B es un conjunto de barbas consistente y $t \in \mathcal{T}$, definimos

$$\mathcal{B}_{\mathrm{con}}[\![\tau t;]\!](B) = \{b + t \mid b \in B\}$$

Está claro que si B es un conjunto consistente de barbas, se tiene que $\mathcal{B}_{\mathrm{con}}[\![\tau t;]\!](B)$ será también un conjunto consistente de barbas. Cara a probar la abstracción de la semántica denotacional con respecto a la semántica de pruebas, hemos de demostrar:

Proposición 4.2.2 Siendo B un conjunto consistente de barbas se verifica

- Si $B \ll \mathsf{Barb}(P)$ entonces $\mathcal{B}_{\mathrm{con}}[\![\tau t;]\!](B) \ll \mathsf{Barb}(\tau t\,;P)$.
- Si $\mathsf{Barb}(P) \ll B$ entonces $\mathsf{Barb}(\tau t\,;P) \ll \mathcal{B}_{\mathrm{con}}[\![\tau t;]\!](B)$.

Demostración. Por un lado tenemos que $\mathsf{TA}(\tau t\,;P) = \varnothing$, y puesto que $\tau t\,;P \xrightarrow{\tau t} P$, tenemos que $b \in \mathsf{Barb}(P)$ si y sólo si $b + t \in \mathsf{Barb}(\tau t\,;P)$.

□

Además se cumple

Proposición 4.2.3 Si $B \ll B'$ y $t \in \mathcal{T}$ entonces $\mathcal{B}_{\mathrm{con}}[\![\tau t;]\!](B) \ll \mathcal{B}_{\mathrm{con}}[\![\tau t;]\!](B)$

Demostración. Es consecuencia inmediata del siguiente hecho:

$$b \ll b' \quad \Rightarrow \quad b + t \ll b' + t$$

□

4.2.3 Prefijo de Acciones Externas

Estudiamos ahora el caso en el que el evento que prefija es una acción visible: $at\,;P$, con $a \in Act$. Tenemos que el proceso $at\,;P$ únicamente puede ejecutar la acción a en tiempo t para convertirse tras ello en P. Por tanto para $at \in TAct$ y B, definimos

$$\mathcal{B}_{\mathrm{con}}[\![at;]\!](B) = \{\varnothing at \cdot b \mid b \in B\} \cup \Big\{\{at\}\Big\}$$

Al igual que antes, tenemos que $\mathcal{B}_{\mathrm{con}}[\![at;]\!](B)$ es un conjunto consistente de barbas. Para probar la correspondencia con la semántica de pruebas en este caso basta probar la

Proposición 4.2.4 Siendo B un conjunto consistente de barbas se verifica

- Si $B \ll \mathsf{Barb}(P)$ entonces $\mathcal{B}_{\mathsf{con}}[\![at;]\!](B) \ll \mathsf{Barb}(at\,;P)$.
- Si $\mathsf{Barb}(P) \ll B$ entonces $\mathsf{Barb}(at\,;P) \ll \mathcal{B}_{\mathsf{con}}[\![at;]\!](B)$.

Proposición 4.2.5 Por un lado tenemos que $\mathcal{A}(at\,;P) = \Big\{\{at\}\Big\}$. Por otra parte

$$at\,;P \xrightarrow{at} P \quad \text{y} \quad \mathsf{TA}(at\,;P) = \{at\}$$

Por tanto, $\varnothing at \cdot b \in \mathsf{Barb}(at\,;P)$ si y sólo si $b \in \mathsf{Barb}(P)$.

Además tenemos que este operador es monótono con respecto al preorden $\ll$.

Proposición 4.2.6 Si $B \ll B'$ y $at \in \mathit{TAct}$ entonces $\mathcal{B}_{\mathsf{con}}[\![at;]\!](B) \ll \mathcal{B}_{\mathsf{con}}[\![at;]\!](B')$

Demostración. Es consecuencia inmediata de los siguientes hechos:

- $\mathcal{A}\Big(\mathcal{B}_{\mathsf{con}}[\![at;]\!](B)\Big) = \mathcal{A}\Big(\mathcal{B}_{\mathsf{con}}[\![at;]\!](B')\Big) = \Big\{\{at\}\Big\}$.
- Si $b \ll b'$ entonces $\varnothing at \cdot b \ll \varnothing at \cdot b'$.

□

4.2.4 Elección Interna

A continuación estudiamos el primero de los operadores binarios: la elección interna. Observando la semántica operacional tenemos que $P \sqcap Q$ se puede *transformar* de forma no determinista bien en P o bien en Q. Por tanto, si B_1 y B_2 son conjuntos de barbas consistentes, definimos

$$\mathcal{B}_{\mathsf{con}}[\![\sqcap]\!](B_1, B_2) = B_1 \cup B_2$$

En esta ocasión el probar la consistencia de $\mathcal{B}_{\mathsf{con}}[\![\sqcap]\!](B_1, B_2)$ no es tan inmediato como en los casos anteriores.

Proposición 4.2.7 Si B_1 y B_2 son conjuntos consistentes de barbas, entonces el conjunto de barbas $\mathcal{B}_{con}[\![\sqcap]\!](B_1, B_2)$ es consistente.

Demostración. Tomemos $B = \mathcal{B}_{\mathsf{con}}[\![\sqcap]\!](B_1, B_2) = B_1 \cup B_2$. Probar que $B \neq \varnothing$ y que B es cerrado bajo prefijos y continuidades es inmediato dado que B_1 y B_2 son consistentes. Para probar la compacidad temporal, tomemos $bs \in \mathsf{Btraz}(B)$. Tenemos las siguientes posibilidades:

de

- $bs \in \mathsf{Btraz}(B_1)$ y $bs \notin \mathsf{Btraz}(B_2)$. En tal caso tenemos que $\mathcal{A}(B, bs) = \mathcal{A}(B_1, bs)$, y puesto que el segundo conjunto es temporalmente compacto, el primero lo será también.
- $bs \in \mathsf{Btraz}(B_2)$ y $bs \notin \mathsf{Btraz}(B_1)$. Caso simétrico al anterior.
- $bs \in \mathsf{Btraz}(B_2)$ y $bs \in \mathsf{Btraz}(B_1)$. En este caso tenemos

$$\mathcal{A}(B, bs) = \mathcal{A}(B_1, bs) \cup \mathcal{A}(B_2.bs)$$

y puesto que $\mathcal{A}(B_1, bs)$ y $\mathcal{A}(B_2, bs)$ son temporalmente compactos, por la proposición 2.4.9, también lo es su unión, $\mathcal{A}(B, bs)$.

□

Cara a demostrar la correspondencia de la semántica denotacional con la semántica de pruebas, para este operador, hemos de probar que se verifica la

Proposición 4.2.8 Siendo B_1 y B_2 conjuntos consistentes de barbas, y P_1 y P_2 procesos se tiene

- Si $B_1 \ll \mathsf{Barb}(P_1)$ y $B_2 \ll \mathsf{Barb}(P_2)$, entonces

$$\mathcal{B}_{\text{con}}[\![\sqcap]\!](B_1, B_2) \ll \mathsf{Barb}(P_1 \sqcap P_2)$$

- Si $\mathsf{Barb}(P_1) \ll B_1$ y $\mathsf{Barb}(P_2) \ll B_2$, entonces

$$\mathsf{Barb}(P_1 \sqcap P_2) \approx \mathcal{B}_{\text{con}}[\![\sqcap]\!](B_1, B_2)$$

Demostración. Simplemente hay que tener en cuenta que

$$P_1 \sqcap P_2 \succ\!\!\longrightarrow P_1 \qquad \text{y} \qquad P_1 \sqcap P_2 \succ\!\!\longrightarrow P_2$$

y por tanto $b \in \mathsf{Barb}(P_1 \sqcap P_2)$ si y sólo si $b \in \mathsf{Barb}(P_1)$ o $b \in \mathsf{Barb}(P_2)$.

□

Por último veremos que éste operador es monótono con respecto a la relación $\ll$.

Proposición 4.2.9 Si $B_1 \ll B_1'$ y $B_2 \ll B_2'$ entonces

$$\mathcal{B}_{\text{con}}[\![\sqcap]\!](B_1, B_2) \ll \mathcal{B}_{\text{con}}[\![\sqcap]\!](B_1', B_2')$$

Demostración. Si $b' \in \mathcal{B}_{\text{con}}[\![\sqcap]\!](B_1', B_2')$ tenemos que b' pertenece a B_1' o a B_2'. Supongamos que está en B_1' (el otro caso es simétrico). Puesto que $B_1 \ll B_1'$ existe $b \in B_1$ tal que $b \in B_1$ y $b \ll b'$, por lo que llegamos a $b \in \mathcal{B}_{\text{con}}[\![\sqcap]\!](B_1, B_2)$.

□

4.2.5 Elección Externa

Este operador es el más complicado de los vistos hasta el momento. Observando la definición de su semántica operacional vemos que sólo las transiciones con acciones visibles *resuelven* la elección. Hasta que ello sucede ambos argumentos están presentes, de modo que los estados del proceso $P \,\square\, Q$ serán una combinación de los estados de P y los de Q. En cambio, una vez resuelta la elección, el proceso se comportará como el proceso que haya realizado la acción visible.

Daremos en primer lugar una siguiente definición, que más adelante utilizaremos de nuevo en el capítulo 6 a la hora de la definir el operador paralelo.

Definición 4.2.10 Si A_1 y A_2 son estados y $G \subsetneq Act$ definimos

$$\begin{aligned} A_1 \sqcup_G A_2 = \; & (A_1 \cap A_2 \cap G) \upharpoonright t \\ & \cup((A_1 \setminus G) \upharpoonright t) \cup ((A_2 \setminus G) \upharpoonright t) \\ & \cup \begin{cases} \{\Omega t\} & \text{si } t < \infty \\ \varnothing & \text{en otro caso} \end{cases} \end{aligned}$$

donde $t = min\{\mathsf{nd}(A_1), \mathsf{nd}(A_2)\}$.

□

Observemos que cuando $G = \varnothing$, el operador definido coincide prácticamente la unión conjuntista; únicamente hay que tener en cuenta los tiempos de indefinición; cuando éstos coinciden en ambos estados, $\mathsf{nd}(A_1) = \mathsf{nd}(A_2)$, se trata exactamente de la unión: $A_1 \sqcup_\varnothing A_2 = A_1 \cup A_2$. El conjunto de acciones G que aparece como subíndice, se utilizará en el operador paralelo, representando el alfabeto de sincronización entre sus argumentos. Si nos fijamos en la definición, observamos que las acciones temporizadas de $A_1 \sqcup_G A_2$ son las acciones que podrían ejecutar dos procesos con tales estados sincronizando en el alfabeto G, esto es

- Las acciones que no están en G y pueden ejecutar cada uno de los procesos independientemente, y
- las acciones que están en G y pueden realizar ambos procesos.

Finalmente el operador de elección externa queda definido como sigue: si B_1 y B_2 son conjuntos consistentes de barbas, tomamos

$$\begin{aligned} \mathcal{B}_{\mathsf{con}}[\![\square]\!](B_1, B_2) = & \Big\{ A_1 \sqcup_\varnothing A_2 \;\Big|\; A_1 \in B_1 \text{ y } A_2 \in B_2 \Big\} \\ & \cup \Big\{ (A_1 \cup A_2 \upharpoonright t)at \cdot b \;\Big|\; A_1 at \cdot b \in B_1 \; A_2 \in B_2 \text{ y } \mathsf{nd}(A_2) > t \Big\} \\ & \cup \Big\{ (A_2 \cup A_1 \upharpoonright t)at \cdot b \;\Big|\; A_2 at \cdot b \in B_2 \; A_1 \in B_1 \text{ y } \mathsf{nd}(A_1) > t \Big\} \end{aligned}$$

beber.

En primer lugar hemos de demostrar que se trata de un operador entre conjuntos de barbas consistentes, lo que viene dada por la

Proposición 4.2.11 Si B_1 y B_2 son conjuntos consistentes, entonces

$$B = \mathcal{B}_{\mathrm{con}}[\![\Box]\!](B_1, B_2)$$

es también un conjunto consistente de barbas.

Demostración. De nuevo la única propiedad no trivial es la compacidad temporal. Para demostrarla tomemos un estado A tal que $\mathsf{nd}(A) = \infty$, de modo que para todo $t \in \mathcal{T}$ exista $bs \cdot A_t \in B$ con $A \upharpoonright t = A_t \upharpoonright t$. Distinguiremos dos casos:

bs $= \epsilon$. Entonces para cada $t \in \mathcal{T}$ existen $A_{t,1} \in B_1$ y $A_{t,2} \in B_2$ tales que $A_t = A_{t,1} \sqcup_{\varnothing} A_{t,2}$. Puesto que estamos considerando un dominio de tiempo discreto y un alfabeto finito podemos suponer que*

$$A_{t,1} \upharpoonright t = A_{t',1} \upharpoonright t \quad \text{y} \quad A_{t,2} \upharpoonright t = A_{t',2} \upharpoonright t \qquad \text{para} \quad t' \geq t$$

Tomemos entonces

$$A_1 = \bigcup_{t \in \mathcal{T}} A_{t,1} \quad \text{y} \quad A_2 = \bigcup_{t \in \mathcal{T}} A_{t,2}$$

Puesto que B_1 y B_2 son consistentes, tenemos que $A_1 \in B_1$ y $A_2 \in B_2$, y por tanto $A = A_1 \sqcup_{\varnothing} A_2 \in B$.

$\mathbf{bs} = \mathbf{A_1 a_1 t_1} \cdot \mathbf{bs'}$. Para cada $t \in \mathcal{T}$ existen un conjunto de acciones temporizadas $A^{t,1}$ y un estado $A^{t,2}$ con $\mathsf{nd}(A^{t,2}) > t_1$ de modo que $A_1 = A^{t,1} \cup (A^{t,2} \upharpoonright t_1)$ y se verifica

- $A^{t,1}at \cdot (bs' \cdot A_t) \in B_1$ y $A^{t,2} \in B_2$, o bien
- $A^{t,1}at \cdot (bs' \cdot A_t) \in B_2$ y $A^{t,2} \in B_1$.

Puesto que siempre podemos encontrar una subsucesión que así lo verifique, podemos asumir o bien que se da siempre el primer caso o bien siempre el segundo. Supongamos que estamos en el primer caso. Además, puesto que estamos trabajando con un dominio de tiempo discreto y un alfabeto finito, podemos suponer que†

$$A^{t,1} = A^{t'1} \quad \text{y} \quad A^{t,2} \upharpoonright t_1 = A^{t',2} \upharpoonright t_1 \qquad \forall t, t' \in \mathcal{T}$$

*Si tomamos un estado A, tenemos que $A \upharpoonright t$ es finito, y por tanto podemos encontrar una subsecuencia de instantes $\{s_t \mid t \in \mathcal{T},\ s_t \geq t\}$ tal que $A_{s_t} \upharpoonright t = A_{s_{t'}} \upharpoonright t$ para $t \leq t'$.

†Igual que antes podremos encontrar una subsucesión que lo verifica.

Tomemos $A^1 = A^{t,1}$ y $A^2 = A^{t,2}$ para algún $t \in \mathcal{T}$. Tenemos entonces que para todo $t \in \mathcal{T}$ se tiene $A^1 a_1 t_1 \cdot (bs' \cdot A_t) \in B_1$, de modo que por ser B_1 consistente, tenemos que $A^1 \cdot (bs' \cdot A) \in B_1$, y por tanto

$$bs \cdot A = (A^1 \cup A^2 \upharpoonright t_1) a_1 t_1 \cdot (bs' \cdot A) \in B$$

$\square$

postrero Para probar la correspondencia entre la semántica denotacional y la semántica de pruebas para este operador hemos de ver que

$$\begin{array}{lcl} B_1 \ll \mathsf{Barb}(P) \quad \text{y} \quad B_2 \ll \mathsf{Barb}(Q) & \Rightarrow & \mathcal{B}_{\text{con}}[\![\square]\!](B_1, B_2) \ll \mathsf{Barb}(P \,\square\, Q) \\ \mathsf{Barb}(P) \ll B_1 \quad \text{y} \quad \mathsf{Barb}(Q) \ll B_2 & \Rightarrow & \mathsf{Barb}(P \,\square\, Q) \ll \mathcal{B}_{\text{con}}[\![\square]\!](B_1, B_2) \end{array}$$

Debido a la forma más compleja en la que en este caso están definidos los estados, la demostración es más delicada que en los anteriores. En primer lugar veamos cómo es el conjunto de estados de $P \square Q$. Para facilitar la lectura y puesto que la mayoría de las demostraciones son eminentemente técnicas, éstas se encuentran con detalle en el apéndice A.

Lema 4.2.12 Siendo P y Q procesos estables y convergentes, se verifican las siguientes propiedades:

- Si $\mathsf{idle}(P) = \mathsf{idle}(Q) = \infty$, entonces $\mathsf{TA}(P \,\square\, Q) = \mathsf{TA}(Q \,\square\, P) = \mathsf{TA}(P) \cup \mathsf{TA}(Q)$.
- Si $P \xrightarrow{\tau t} P'$ e $\mathsf{idle}(Q) \geq t$, entonces

$$\mathsf{TA}(P \,\square\, Q) \upharpoonright t = \mathsf{TA}(Q \,\square\, P) \upharpoonright t = \mathsf{TA}(P) \upharpoonright t \cup \mathsf{TA}(Q) \upharpoonright t$$

Demostración. En el primer caso basta tener en cuenta que

$$P \,\square\, Q \xrightarrow{at'} R \quad \Longleftrightarrow \quad P \xrightarrow{at'} R \,\vee\, Q \xrightarrow{at'} R$$

En el segundo caso hemos de utilizar además que

$$P \,\square\, Q \xrightarrow{at'} R \quad \Longleftrightarrow \quad t' \leq t \,\wedge\, (P \xrightarrow{at'} R \,\vee\, Q \xrightarrow{at'} R)$$

$\square$

Lema 4.2.13 Siendo $A \in \mathcal{A}(P \,\square\, Q)$ y $t \in \mathcal{T}$, se tiene que

- Si $\mathsf{nd}(A) < t$ existen $A_P \in \mathcal{A}(P)$ y $A_Q \in \mathcal{A}(Q)$ tales que $A_P \sqcup_\varnothing A_Q \ll A$.

-¿Por

- Si $\mathsf{nd}(A) \geq t$ existen $A_P \in \mathcal{A}(P)$ y $A_Q \in \mathcal{A}(Q)$ tales que $(A_P \sqcup_\varnothing A_Q) \upharpoonright t \subseteq A \upharpoonright t$.

Como consecuencia del lema tenemos la siguiente

Proposición 4.2.14 Sean P y Q dos procesos. Si $A \in \mathcal{A}(P \,\square\, Q)$ entonces existen $A_P \in \mathcal{A}(P)$ y $A_Q \in \mathcal{A}(Q)$ tales que $A_P \sqcup_\varnothing A_Q \ll A$.

Demostración. Para cada estado $A \in \mathcal{A}(P \,\square\, Q)$ existen dos posibilidades:

- Si $\mathsf{nd}(A) < \infty$, si aplicamos el lema anterior con $t > \mathsf{nd}(A)$, tenemos que existen $A_P \in \mathcal{A}(P)$ y $A_Q \in \mathcal{A}(Q)$ de modo que $A_P \sqcup_\varnothing A_Q \ll A$.

- $\mathsf{nd}(A) = \infty$. Aplicando el lema anterior, tenemos que para cada $t \in \mathcal{T}$ existen $A^t_P \in \mathcal{A}(P)$ y $A^t_P \in \mathcal{A}(Q)$ tales que

$$(A^t_P \sqcup_\varnothing A^t_Q) = (A^t_P \upharpoonright t) \cup (A^t_Q \upharpoonright t) \subseteq A$$

  Puesto que $A^t_P \subseteq A$, $A^t_Q \subseteq A$ y tanto $\mathcal{A}(P)$ como $\mathcal{A}(Q)$ son conjuntos de estados temporalmente compactos, por la proposición 2.4.8, existen estados $A_P \in \mathcal{A}(P)$ y $A_Q \in \mathcal{A}(Q)$ tales que $A_P \subseteq A$ y $A_Q \subseteq A$. Finalmente tenemos

$$A_P \sqcup_\varnothing A_Q = A_P \cup A_Q \subseteq A$$

□

Lema 4.2.15 Si $A_P \in \mathcal{A}(P)$, $A_Q \in \mathcal{A}(Q)$ y $t \in \mathcal{T}$, entonces:

- Si $\mathsf{nd}(A_P) < t$ ó $\mathsf{nd}(A_Q) < t$, entonces existe $A \in \mathcal{A}(P \,\square\, Q)$ tal que

$$A \ll A_P \sqcup_\varnothing A_Q$$

- Si $\mathsf{nd}(A_P) \geq t$ y $\mathsf{nd}(A_Q) \geq t$, entonces existe $A \in \mathcal{A}(P \,\square\, Q)$ tal que

$$A \upharpoonright t \subseteq (A_P \sqcup_\varnothing A_Q) \upharpoonright t$$

□

Proposición 4.2.16 Sean P y Q procesos y consideremos dos estados $A_P \in \mathcal{A}(P)$ y $A_Q \in \mathcal{A}(Q)$. Entonces existe un estado $A \in \mathcal{A}(P \,\square\, Q)$ tal que $A \ll A_P \sqcup_\varnothing A_Q$.

Demostración. Dados los estados A_P y A_Q existen dos posibilidades:

- $\mathsf{nd}(A_P) < \infty$ ó $\mathsf{nd}(A_Q) < \infty$. Supongamos que $\mathsf{nd}(A_P) < \infty$, en tal caso tomando $t > \mathsf{nd}(A_P)$, tenemos que por el lema anterior existe $A \in \mathcal{A}(P \Box Q)$ tal que $A \ll A_P \sqcup_\varnothing A_Q$.

- $\mathsf{nd}(A_P) = \infty$ y $\mathsf{nd}(A_Q) = \infty$. Aplicando el lema anterior, para cada $t \in \mathcal{T}$ existe $A_t \in \mathcal{A}(P \Box Q)$ tal que
$$A_t \upharpoonright t \subseteq (A_P \sqcup_\varnothing A_Q) \upharpoonright t$$
Entonces como $\mathcal{A}(P \Box Q)$ es un conjunto de estados temporalmente compacto, por la proposición 2.4.8, existirá un estado $A \in \mathcal{A}(P \Box Q)$ tal que $A \subseteq A_P \sqcup_\varnothing A_Q$.

Por lo que en cualquier caso tenemos $A' \ll A$.

□

A continuación probaremos que cada b-traza de $P \Box Q$ se puede obtener como una combinación de un estado de P y una b-traza de Q, o viceversa, un estado de Q y una b-traza de P.

Proposición 4.2.17 Si $P \Box Q \xRightarrow{Aat \cdot bs} R$ entonces existen una b-traza $A_1 at \cdot bs$ y un estado A_2 verificando $\mathsf{nd}(A_2) > t$, $A_1 \cup (A_2 \upharpoonright t) \subseteq A$ de modo que se tiene una de las siguientes condiciones:

- $P \xRightarrow{A_1 at \cdot bs} R$ y $A_2 \in \mathcal{A}(Q)$, o bien
- $Q \xRightarrow{A_1 at \cdot bs} R$ y $A_2 \in \mathcal{A}(P)$.

□

De manera análoga, toda b-traza de P y todo estado (adecuado) de Q da lugar a una b-traza de $P \Box Q$, y de manera simétrica una b-traza de $Q \Box P$.

Proposición 4.2.18 Si $P \xRightarrow{A_P at \cdot bs} R$, y $A_Q \in \mathcal{A}(Q)$ es tal que $\mathsf{nd}(A_Q) > t$, entonces existe $A \subseteq A_P \cup (A_Q \upharpoonright t)$ verificando las siguientes propiedades:

- $P \Box Q \xRightarrow{Aat \cdot bs} R$.
- $Q \Box P \xRightarrow{Aat \cdot bs} R$.

□

Finalmente, como combinación de los resultados anteriores, obtenemos el resultado que buscábamos.

Teorema 4.2.19 Siendo B_1 y B_2 conjuntos consistentes, entonces se tiene:

---

qué

- Si $\mathsf{Barb}(P) \ll B_2$ y $\mathsf{Barb}(Q) \ll B_2$ entonces $\mathsf{Barb}(P \,\square\, Q) \ll \mathcal{B}_{\mathrm{con}}[\![\square]\!](B_1, B_2)$.
- Si $B_1 \ll \mathsf{Barb}(P)$ y $B_2 \ll \mathsf{Barb}(Q)$ entonces $\mathcal{B}_{\mathrm{con}}[\![\square]\!](B_1, B_2) \ll \mathsf{Barb}(P \,\square\, Q)$.

$\square$

Para concluir esta sección demostraremos que el operador correspondiente a la elección externa es monótono con respecto a la relación $\ll$.

Proposición 4.2.20 Si B_1, B'_1, B_2 y B'_2 conjuntos consistentes de barbas se verifica

$$B_1 \ll B'_1 \ \text{ y } \ B_2 \ll B'_2 \quad \Rightarrow \quad \mathcal{B}_{\mathrm{con}}[\![\square]\!](B_1, B_2) \ll \mathcal{B}_{\mathrm{con}}[\![\square]\!](B'_1, B'_2)$$

Demostración. Sea $b' \in \mathcal{B}_{\mathrm{con}}[\![\square]\!](B'_1, B'_2)$. Consideremos en primer lugar el caso en que b' es un estado A'. Entonces existen estados $A'_1 \in B'_1$ y $A'_2 \in B'_2$ tales que $A' = A'_1 \sqcup_{\varnothing} A'_2$. Además puesto que $B_1 \ll B'_1$ y $B_2 \ll B'_2$ existirán estados $A_1 \in B_1$ y $A_2 \in B_2$ de modo que $A_1 \ll A'_1$ y $A_2 \ll A'_2$. Tomamos entonces el estado $A = A_1 \sqcup_{\varnothing} A_2$ y tenemos que

$$A \ll A' \quad \text{y} \quad A \in \mathcal{B}_{\mathrm{con}}[\![\square]\!](B_1, B_2)$$

Supongamos ahora que $b' = A'at \cdot b''$. Entonces existen un conjunto de acciones temporizadas A'_1 y un estado A'_2 tales que $A' = A'_1 \cup A'_2 \upharpoonright t$, $\mathsf{nd}(A'_2) > t$ y se verifica una de las siguientes posibilidades:

- $A'_1at \cdot b'' \in B'_1$ y $A'_2 \in B_2$, o bien
- $A'_1at \cdot b'' \in B'_2$ y $A'_2 \in B_1$

Supongamos que se da la primera posibilidad (la otra es simétrica). Existen entonces una barba $b_1 \in B_1$ con $b_1 \ll A'_1at \cdot b''$ y un estado $A_2 \in B_2$ verificando $A_2 \ll A'_2$. Construimos entonces la barba b de la siguiente forma:

- Si $b_1 = A$ tomamos $b = A_1 \sqcup_{\varnothing} A_2$.
- Si $b_1 = A_1at \cdot b'_1$ y $\mathsf{nd}(A_2) \leq t$, como quiera que B_1 es consistente existirá un estado $A''_1 \in B_1$ tal que $A''_1 \upharpoonright t \subseteq A_1$. Tomamos entonces $b = A''_1 \sqcup_{\varnothing} A_2$.
- Si $b_1 = A_1at \cdot b'_1$ y $\mathsf{nd}(A_2) > t$, tomamos $b = (A_1 \cup A_2 \upharpoonright t)at \cdot b'_1$.

Vemos inmediatamente que en cualquier caso tenemos $b \ll b'$ y $b \in \mathcal{B}_{\mathrm{con}}[\![\square]\!](B_1, B_2)$.

$\square$

4.3 Semántica para Procesos Finitos

Una vez definidos los operadores semánticos, tenemos inmediatamente definida la semántica denotacional para los procesos finitos. La misma la definiremos para todos los términos finitos en $\mathsf{FRec}(\Sigma_{\mathsf{seq}})$, de modo que los procesos finitos constituirán un caso particular. Para poder definir la semántica de procesos con variables debemos considerar una asignación de valores a las variables, es decir una función

$$\rho : \mathsf{Var} \mapsto \mathcal{B}_{\mathsf{con}}$$

Utilizando las mismas llegamos a la siguiente

Definición 4.3.1 Si $P \in \mathsf{FRec}(\Sigma_{\mathsf{seq}})$ y ρ es una asignación de variables definimos

$$\mathcal{B}_{\mathsf{con}}[\![P]\!]_\rho = \begin{cases} \rho(x) & \text{si } P = x \in \mathsf{Var} \\ \mathcal{B}_{\mathsf{con}}[\![op]\!](\mathcal{B}_{\mathsf{con}}[\![P_1]\!]_\rho, \ldots, \mathcal{B}_{\mathsf{con}}[\![P_n]\!]_\rho) & \text{si } P = op(P_1, \ldots, P_2) \text{ y } op \in \Sigma_{\mathsf{seq}} \end{cases}$$

□

Si una determinada variable de proceso x no aparece en P, la semántica del proceso no dependerá del valor de x en la asignación de variables ρ. En consecuencia, para procesos finitos (términos finitos cerrados), podemos hablar tranquilamente de su semántica sin preocuparnos de ninguna asignación de variables.

Definición 4.3.2 Si $P \in \mathsf{FCRec}(\Sigma_{\mathsf{seq}})$ es un proceso finito definimos

$$\mathcal{B}_{\mathsf{con}}[\![P]\!] = \mathcal{B}_{\mathsf{con}}[\![P]\!]_\rho$$

donde ρ es una asignación cualquiera de variables.

□

Como consecuencia de las propiedades de las funciones semánticas correspondiente a cada operador, podemos probar que, para procesos finitos, la semántica denotacional dada es completamente abstracta con respecto a la semántica de pruebas:

Teorema 4.3.3 Para todo proceso finito $P \in \mathsf{FCRec}(\Sigma_{\mathsf{seq}})$, se tiene $\mathsf{Barb}(P) \approx \mathcal{B}_{\mathsf{con}}[\![P]\!]$.

Demostración. La demostración es inmediata por inducción estructural, gracias a las propiedades demostradas para cada uno los operadores:

- Para `STOP` y `DIV`, utilizamos la proposición 4.2.1.

vendes

- Para el operador de prefijo con acción oculta, la proposición 4.2.2.
- Para el operador de prefijo con acción visible, la proposición 4.2.4.
- Para la elección interna, la proposición 4.2.8.
- Para la elección externa, el teorema 4.2.19.

□

4.4 Procesos Recursivos

En esta sección daremos un valor semántico a los procesos recursivos. Para ello utilizaremos las técnicas clásicas de punto fijo. Desgraciadamente no hemos podido demostrar que el preorden $\ll$ es completo. Sin embargo hemos podido dar con un orden alternativo que sí lo es, conservando una fuerte relación con el orden $\ll$, lo que nos permitirá llevar adelante nuestro trabajo de forma coherente. Para definir dicho este preorden necesitamos primero generalizar la relación $\prec$ entre estados definida en la definición 2.4.4.

Definición 4.4.1 Siendo b y b' dos barbas, diremos que b *está menos definida* que b', y lo escribiremos $b \prec b'$, si y sólo si se verifica una de las siguientes condiciones:

- $b = A\,,\; b' = A'$ y $A' \prec A$.
- $b = Aat \cdot b_1\,,\; b' = A'\,,\; \mathsf{nd}(A') \leq t$ y $TAct(A') = A \upharpoonright \mathsf{nd}(A')$.
- $b = Aat \cdot b_1\,,\; b' = Aat \cdot b_1'$ y $b_1' \prec b_1$.

□

Fijándonos en la definición anterior podemos observar que si tomamos una barba b de modo que $\mathsf{nd}(b) = \infty$, para que se cumpla $b \prec b'$, se ha de cumplir necesariamente que $b = b'$. Sin embargo, si $\mathsf{nd}(b) < \infty$, más allá del instante de indefinición podemos tener cualquier información, que defina la barba *un poco más*. A continuación extendemos la relación $\prec$ sobre conjuntos de barbas.

Definición 4.4.2 Diremos que B_1 está menos definido que B_2, y lo escribiremos $B_1 \prec B_2$, si y sólo si se verifican

- Para cada $b_2 \in B_2$ existe $b_1 \in B_1$ de modo que $b_1 \prec b_2$.
- Para cada $bs \cdot A_1 \in B_1$, existe $bs \cdot A_2 \in B_2$ de modo que $A_1 \prec A_2$.

□

La idea que hay debajo de la definición anterior es la siguiente: cuando $B_1 \prec B_2$ tenemos que

- Si $b_1 \in B_1$, está completamente definida, $\mathsf{nd}(b_1) = \infty$, tendremos que b_1 también habrá de pertenecer a B_2. En cambio si todavía no está completamente definida, bastará con que exista en B_2 otra barba b_2 que *esté algo tanto o más definida* que b_1.
- Cada barba $b_2 \in B_2$ proviene de definir *algo más* alguna barba de B_1.

Es fácil comprobar que la relación $\prec$ verifica las propiedades reflexiva y transitiva. En cambio no verifica la propiedad antisimétrica, por lo que la relación $\prec$ no es una relación de orden entre conjuntos consistentes de barbas, sino sólo un preorden. Esto implica que si trabajamos directamente con esta relación, no podemos esperar en general la unicidad de las mínimas cotas superiores. No obstante, toda la teoría de *cpo*'s puede generalizarse sin problemas al caso de preordenes. En particular podemos probar que $(\mathcal{B}_{\text{con}}, \prec)$ es completo Para ello consideramos una cadena no decreciente $\mathcal{B}$ de conjuntos consistentes de barbas:

$$B_1 \prec B_2 \prec B_3 \cdots$$

Hemos de ver que tiene una mínima cota superior, es decir, que existe un conjunto consistente de barbas $\mathsf{lub}(\mathcal{B})$ tal que

- Para cada $i \in \mathbb{N}$ se verifica $B_i \prec \mathsf{lub}(\mathcal{B})$.
- Si B' es un conjunto consistente de barbas tal que $B_i \prec B'$ para cada $i \in \mathbb{N}$, entonces $\mathsf{lub}(\mathcal{B}) \prec B'$,

Definimos a continuación el conjunto $\mathsf{lub}(\mathcal{B})$ que más tarde comprobaremos que reúne las propiedades deseadas.

Definición 4.4.3 Sea $\mathcal{B} = \{B_i \mid i \in \mathbb{N}\}$ una cadena no decreciente de conjuntos de barbas. Tendremos que $bs \cdot A \in \mathsf{lub}(\mathcal{B})$ si y sólo si se cumple una de las siguientes condiciones:

- Si $\mathsf{nd}(A) < \infty$ y para todo $k \in \mathbb{N}$ existe $l \geq k$ de modo que $bs \cdot A \in B_l$.
- Si $\mathsf{nd}(A) = \infty$, y para todo $t \in \mathcal{T}$ existe $l \in \mathbb{N}$ y un estado A_t verificando

$$bs \cdot A_t \in B_l \quad \text{y} \quad A_t \upharpoonright t = A \upharpoonright t$$

esto?

□

Observemos en el segundo punto de la definición anterior, que el estado A_l que aparece ha de verificar $\mathsf{nd}(A_l) \geq t$.

La siguiente proposición es una caracterización del conjunto que acabamos de definir que resulta más cómoda algunas ocasiones.

Proposición 4.4.4 Sea $\mathcal{B} = \{B_i \mid i \in \mathbb{N}\}$ una cadena no decreciente de conjuntos de barbas. Tendremos que $b \in \mathsf{lub}(\mathcal{B})$ si y sólo si se cumple una de las siguientes condiciones:

- Si $\mathsf{nd}(b) < \infty$ y para todo $k \in \mathbb{N}$ existe $l \geq k$ que $bs \in B_l$.
- Si $\mathsf{nd}(b) = \infty$, y para todo $t \in \mathcal{T}$ existe $l \in \mathbb{N}$ y una barba b_l verificando

$$b_l \in B_l \quad \text{y} \quad b_l \upharpoonright t = b \upharpoonright t$$

Demostración. El resultado es consecuencia inmediata del siguiente hecho:

$$\mathsf{nd}(bs \cdot A) = \mathsf{t}(bs) + \mathsf{nd}(A)$$

y por tanto $\quad \mathsf{nd}(bs \cdot A) < \infty \quad \Longleftrightarrow \quad \mathsf{nd}(A) < \infty.$

□

Demostramos a continuación que $\mathsf{lub}(\mathcal{B})$ es en efecto la menor cota superior de la cadena no decreciente $\mathcal{B}$. Precisamos para ello una serie de lemas auxiliares.

Lema 4.4.5 Dada una secuencia de estados de modo que

$$A_1 \prec A_2 \prec A_3 \cdots A_k \prec A_{k+1} \cdots$$

entonces tomando $A = \mathsf{lub}\{A_i \mid i \in \mathbb{N}\}$ (definición 2.4.5) se verifican:

- Si $\mathsf{nd}(A) = \infty$ y $t \in \mathcal{T}$, existe $l \in \mathbb{N}$ tal que $A_l \upharpoonright t = A \upharpoonright t$.
- Si $\mathsf{nd}(A) < \infty$, existe $l \in \mathbb{N}$ tal que $A = A_k$ para todo $k \geq l$.

Demostración. Observemos en primer lugar que $\mathsf{nd}(A) = sup\{\mathsf{nd}(A_i) \mid i \in \mathbb{N}\}$, y por tanto para cada $t \leq \mathsf{nd}(A)$, habrá de existir $l \in \mathbb{N}$ con $\mathsf{nd}(A_l) \geq t$. Puesto que $A_l \prec A_k$, para $k \geq l$ tendremos entonces que $A_l \upharpoonright t = A_t \upharpoonright t$, y por tanto $A \upharpoonright t = A_l \upharpoonright t$.

En el caso de que $\mathsf{nd}(A) = \infty$, de lo dicho anteriormente se sigue inmediatamente el resultado buscado correspondiente. Supongamos entonces que $t = \mathsf{nd}(A) < \infty$. En tal caso ha de existir $l \in \mathbb{N}$ tal que $\mathsf{nd}(A_l) = t$, y puesto que $A_l \prec A_k$ y $\mathsf{nd}(A_k) \leq t$ para $k \geq l$, llegamos a $A_k = A_l$.

□

Lema 4.4.6 $bs \in \mathsf{Btraz}(\mathsf{lub}(\mathcal{B}))$ si y sólo si existe $l \in \mathbb{N}$ tal que $bs \in \mathsf{Btraz}(B_l)$.

Demostración.

$\boxed{\Rightarrow}$

Sea $bs \in \mathsf{Btraz}(\mathsf{lub}(\mathcal{B}))$; entonces por definición existe, una barba b tal que $bs \cdot b \in \mathcal{B}$. Supongamos que $b = bs' \cdot A'$; distinguimos entonces dos casos:

- $\mathsf{nd}(A') < \infty$, en cuyo caso existe $l \in \mathbb{N}$ tal que $bs \cdot (bs' \cdot A') \in B_l$. Con lo cual $bs \in \mathsf{Btraz}(B_l)$.
- $\mathsf{nd}(A') = \infty$, en este caso existe $l \in \mathbb{N}$ y un estado A_0 tales que

$$bs \cdot (bs' \cdot A_0) \in B_l \quad \text{y} \quad A_0 \upharpoonright 0 = A' \upharpoonright 0$$

con lo cual $bs \in \mathsf{Btraz}(B_l)$.

$\boxed{\Leftarrow}$

Supongamos ahora que existe $l \in \mathbb{N}$ tal que $bs \in \mathsf{Btraz}(B_l)$. Puesto que B_l es consistente, existirá un estado A_l tal que $bs \cdot A_l \in B_l$, y como quiera que $B_k \prec B_{k+1}$, podemos encontrar una cadena

$$A_l \prec A_{l+1} \prec \cdots A_k \prec A_{k+1} \cdots$$

verificando $bs \cdot A_k \in B_k$. Tomamos entonces el estado $A = \mathsf{lub}\{A_k \mid k \geq l\}$ y por el lema 4.4.5 tenemos que $bs \cdot A \in \mathsf{lub}(\mathcal{B})$, con lo que llegamos a $bs \in \mathsf{Btraz}(\mathsf{lub}(\mathcal{B}))$.

□

Veamos ahora que $\mathsf{lub}(\mathcal{B})$ cumple las condiciones necesarias para ser la mínima cota superior. La primera de ellas es que $\mathsf{lub}(\mathcal{B})$, en efecto, es un conjunto consistente de barbas:

Proposición 4.4.7 Siendo $\mathcal{B} = \{B_i \mid i \in \mathbb{N}\}$ una cadena no decreciente de conjuntos consistente de barbas

$$B_1 \prec B_2 \prec \cdots B_{k-1} \prec B_k \cdots$$

tenemos que el conjunto $\mathsf{lub}(\mathcal{B})$ siempre es es un conjunto consistente de barbas

Demostración. Hemos de comprobar que $\mathsf{lub}(\mathcal{B})$ verifica las condiciones de la definición 4.1.1. Para simplificar la notación tomemos $B = \mathsf{lub}(\mathcal{B})$.

-dijo

- $B \neq \varnothing$. Puesto que B_1 es consistente existirá alguna barba $bs \cdot A_1 \in B_1$. Entonces, como $B_k \prec B_{k+1}$, podemos encontrar una secuencia de estados

$$A_1 \prec A_2 \prec \cdots A_k \prec A_{k+1} \cdots$$

de modo que $bs \cdot A_i \in B_i$. Entonces si tomamos el estado $A = \mathsf{lub}\{A_i \mid i \in \mathbb{N}\}$, por el lema 4.4.5 se tiene que $bs \cdot A \in B$.

- *Cerrado bajo prefijos.* Consideremos una b-traza

$$bs = A_1 a_1 t_1 \cdots A_{n-1} a_{n-1} t_{n-1} A_n a_n t_n \in \mathsf{Btraz}(B)$$

Por el lema 4.4.6 existe $l \in \mathbb{N}$ de manera que $bs \in \mathsf{Btraz}(B_l)$. Tenemos entonces que $bs \in \mathsf{Btraz}(B_k)$ para $k \geq l$. Entonces para cada $k \geq l$ existirá un estado A^k de modo que

  - $A_1 a_1 t_1 \cdots A_{n-1} a_{n-1} t_{n-1} \cdot A^k \in B_k$,
  - $A_n = A^k \upharpoonright t_n$.

Combinando dichos estados podemos construir un estado A verificando

  - $A_1 a_1 t_1 \cdots A_{n-1} a_{n-1} t_{n-1} \cdot A \in \mathsf{lub}(\mathcal{B})$,
  - $A_n = A \upharpoonright t_n$.

- *Cerrado bajo continuaciones.* Supongamos ahora que $bs \cdot A \in B$, y que $at \in A$. Según la forma de A dos posibilidades:

  - Si $\mathsf{nd}(A) < \infty$, entonces para cada $k \in \mathbb{N}$ existirá $l \geq k$ de modo que $bs \cdot A \in B_l$. Puesto que cada B_l es consistente se tiene

$$bs \cdot (A \upharpoonright t)at \in \mathsf{Btraz}(B_l)$$

  y por aplicación del lema 4.4.6 tenemos que $bs \cdot (A \upharpoonright t)at \in \mathsf{Btraz}(\mathsf{lub}(\mathcal{B}))$.

  - Si $\mathsf{nd}(A) = \infty$ y tomamos $t' > t$, tenemos que existirá $l \in \mathbb{N}$ y un estado A' tales que

$$bs \cdot A' \in B_{l'} \quad \text{y} \quad A' \upharpoonright t' = A \upharpoonright t$$

  Puesto que B_l es consistente y $A' \upharpoonright t = A \upharpoonright t$ tenemos que

$$bs \cdot (A \upharpoonright t)at \in \mathsf{Btraz}(B_l)$$

  Y al igual que que en el caso anterior, por aplicación del lema 4.4.6, tenemos que $bs \cdot (A \upharpoonright t)at \in \mathsf{Btraz}(\mathsf{lub}(\mathcal{B}))$.

- *Temporalmente compacto.* Sean $bs \in \mathsf{Btraz}(B)$ y $\mathcal{A} = \mathcal{A}(B, bs)$, y tomemos un estado A tal que $\mathsf{nd}(A) = \infty$ y verificando que para todo $t \in \mathcal{T}$ se cumple que existe $A_t \in \mathcal{A}$ tal que $A_t \upharpoonright t = A \upharpoonright t$. Entonces, puesto que $A_t \in \mathcal{A}$, existirá $l \in \mathbb{N}$ tal que

$$bs \cdot A_l \in B_l \quad \text{y} \quad A_l \upharpoonright t = A_t \upharpoonright t = A \upharpoonright t$$

por lo que $bs \cdot A \in B$, y en consecuencia $A \in \mathcal{A}$.

□

El siguiente paso consiste en demostrar que B es *mayor* que todos los conjuntos de barbas de la cadena:

Proposición 4.4.8 Sea $\mathcal{B} = \{B_i \mid i \in \mathbb{N}\}$ una cadena no decreciente de conjuntos consistente de barbas

$$B_1 \prec B_2 \prec \cdots B_{k-1} \prec B_k \cdots$$

entonces $B_i \prec \mathsf{lub}(\mathcal{B})$ para todo $i \in \mathbb{N}$.

Demostración. Para simplificar la notación tomemos $B = \mathsf{lub}(\mathcal{B})$.

- Tomemos en primer lugar $bs \cdot A \in B$. Si $\mathsf{nd}(A) < \infty$, existirá $l \geq i$ tal que $bs \cdot A \in B_l$. como quiera que $B_i \prec B_l$, existirá $b_i \in B_i$ tal que $b_i \prec bs \cdot A$.

  Supongamos ahora que $\mathsf{nd}(A) = \infty$. Entonces para cada $t \in \mathcal{T}$ existen $l_t \in \mathbb{N}$ y un estado A_{l_t} tales que $bs \cdot A_{l_t} \in B_{l_t}$ y $A_{l_t} \upharpoonright t = A \upharpoonright t$. Sin pérdida de generalidad podemos suponer que $l_t > l_{t'}$ para $t > t'$, por lo que ha de existir $t \in \mathcal{T}$ tal que $l_t \geq i$. Entonces para cada cada $t' > t$, teniendo en cuenta que $B_i \prec B_{l_{t'}}$ habrá de existir una barba $bs_{t'} \cdot A_{t'} \in B_i$ tal que

$$bs_{t'} \cdot A_{t'} \prec bs \cdot A_{l_{t'}}$$

  Si algún $bs_{t'}$ verifica $\mathsf{lon}(bs_{t'}) < \mathsf{lon}(bs)$ tenemos directamente que $bs_{t'} \cdot A_{t'} \prec bs \cdot A$. Supongamos entonces que todo $bs_{t'}$ verifica $\mathsf{lon}(bs_{t'}) = \mathsf{lon}(bs)$, es decir, $bs_{t'} = bs$ y $A_{t'} \prec A_{l_{t'}}$. Si existe entonces $t' \geq t$ tal que $\mathsf{nd}(A_{t'}) \leq t'$ tenemos que $A_{t'} \prec A$. Supongamos pues que $\mathsf{nd}(A_{t'}) > t'$ para todo $t' > t$, en cuyo caso o tenemos

$$A_{t'} \upharpoonright t' = A_{l_{t'}} \upharpoonright t' = A \upharpoonright t'$$

  De donde como B_i es un conjunto consistente de barbas, el conjunto de estados $\mathcal{A}(B, bs)$ es temporalmente compacto, y por tanto $bs \cdot A \in B_i$.

---

el

- Sea ahora $bs \cdot A_i \in B_i$. Puesto que $B_i \prec B_j$ para todo $j \geq i$, tenemos que existe $bs \cdot A_j \in B_i$ de modo que $A_i \prec A_j$. Tomemos entonces el estado

$$A = \mathsf{lub}\{A_j \mid j \geq i\}$$

que verifica $bs \cdot A \in B$ y $A_i \prec A$.

□

Una vez comprobado que $\mathsf{lub}(\mathcal{B})$ es una cota superior de la cadena $\mathcal{B}$, hay que probar que es *la menor* (salvo equivalencia) de todas ellas.

Proposición 4.4.9 Sea $\mathcal{B} = \{B_i \mid i \in \mathbb{N}\}$ una cadena no decreciente de conjuntos consistente de barbas

$$B_1 \prec B_2 \prec \cdots \prec B_{k-1} \prec B_k \cdots$$

Si B' es un conjunto consistente de barbas verificando que $B_i \prec B'$ para cada $i \in \mathbb{N}$, entonces $\mathsf{lub}(\mathcal{B}) \prec B'$.

Demostración. De nuevo para simplificar tomaremos $B = \mathsf{lub}(\mathcal{B})$.

- Consideremos en primer lugar tomemos $b' \in B'$. Puesto que para cada $i \in \mathbb{N}$ $B_i \prec B'$, existirá una barba $b_i = bs_i \cdot A_i \in B_i$ tal que $b_i \prec b'$. Sin pérdida de generalidad podemos suponer que existe $k_0 \in \mathbb{N}$ tal que $\mathsf{lon}(b_k) = \mathsf{lon}(b_{k_0})$ para $k \geq k_0$, y por tanto $bs_k = bs_{k_0}$. Distinguimos entonces dos casos:

  - $\mathsf{lon}(b') > \mathsf{lon}(b_k)$. En tal caso, para cada $k \geq k_0$ tenemos que

  $$b' = A_1 a_1 t_1 \cdots A_m a_m t_m \cdot b'' \quad , \quad b_k = A_1 a_1 t_1 \cdots A_m^k \quad \text{y} \quad \mathsf{nd}(A_m^k) \leq t_m$$

  Puesto que estamos considerando un alfabeto finito y un dominio de tiempo discreto, podremos tomar $k_1 \geq k_0$ de modo que $A_m^{k_1} = A_m^k$ para $k \geq k_1$, por lo que tomando $A = A_m^{k_1}$ obtenemos $b = A_1 a_1 t_1 \cdots A \in B$ y $b \prec b'$.

  - $\mathsf{lon}(b') = \mathsf{lon}(b_k)$. Tomemos que $b' = bs \cdot A'$, entonces cada bs_k para $k \geq k_0$ ha de ser igual a bs. De nuevo podemos distinguir dos casos:

    * Existe $t \in \mathcal{T}$ tal que para un número infinito de estados A_k se tiene $\mathsf{nd}(A_k) < t$. En tal caso, puesto que estamos considerando un alfabeto finito y un dominio de tiempo discreto, podremos suponer que existe $k_1 \geq k_0$ tal que $A_k = A_{k_1}$ para $k \geq k_1$. Por lo que tomando $A = A_{k_1}$ tenemos $b = bs \cdot A \in B$ y $b \prec b'$.

* Para cada $t \in \mathcal{T}$ existe $l_t \geq k_0$ tal que $\mathsf{nd}(A_{l_t}) \geq t$. Entonces, puesto que $A^{l_t} \prec A'$ tenemos $A^{k_t} \upharpoonright t = A' \upharpoonright t$, por lo que $bs \cdot A' \in B$.

- Tomemos ahora $b = bs \cdot A \in B$. Según sea el estado A hay dos posibilidades:
  - Si $\mathsf{nd}(A) < \infty$, existirá $l \in \mathbb{N}$ tal que $bs \cdot A \in B_l$, y puesto que $B_l \prec B'$ existirá un estado A' tal que $bs \cdot A' \in B'$ y $A \prec A'$.
  - Si $\mathsf{nd}(A) = \infty$, para cada $t \in \mathcal{T}$ existen $l_t \in \mathbb{N}$ y un estado A_t tales que

$$bs \cdot A_t \in B_{l_t} \quad \text{y} \quad A_t \upharpoonright t = A \upharpoonright t$$

Puesto que $B_{l_t} \prec B'$, para cada $t \in \mathcal{T}$ existe un estado A'_t tal que

$$bs \cdot A'_t \in B' \quad \text{y} \quad A_t \prec A$$

lo cual implica $A'_t \upharpoonright t = A_t \upharpoonright t = A \upharpoonright t$. Finalmente, puesto que B' es temporalmente compacto, obtenemos $bs \cdot A \in B'$.

□

Una vez probado que el preorden $\prec$ es completo, vamos a enunciar su relación con el orden $\ll$. En primer lugar tenemos el

Lema 4.4.10 Si $B \prec B'$ entonces $B \ll B$

Demostración. Observemos que $b \prec b' \Rightarrow b \ll b'$, de donde el resultado se sigue inmediatamente.

□

En virtud del lema anterior, si tenemos una cadena $\mathcal{B}$ con

$$B_1 \prec B_2 \cdots B_k \prec B_{k+1} \prec \cdots$$

tendremos que $\mathsf{lub}(\mathcal{B})$ será también una cota superior con respecto a la relación $\ll$:

$$\forall i \in \mathbb{N} \quad B_i \ll \mathsf{lub}(\mathcal{B})$$

La siguiente proposición va más allá; nos indica que $\mathsf{lub}(\mathcal{B})$ seguirá siendo la menor cota superior con respecto al orden $\ll$:

Proposición 4.4.11 Sea $\mathcal{B} = \{B_i \mid i \in \mathbb{N}\}$ una cadena no decreciente de conjuntos consistentes de barbas

$$B_1 \prec B_2 \prec \cdots B_{k-1} \prec B_k \cdots$$

Si B' es un conjunto consistente de barbas tal que $B_i \ll B'$, se verificará $\mathsf{lub}(\mathcal{B}) \ll B'$.

---

principito.

Demostración. Sea $b' = bs' \cdot A \in B'$. Entonces para cada $i \in \mathbb{N}$ existe una barba

$$b_i = bs_i \cdot A_i \in B_i \quad \text{t.q.} \quad b_i \ll b'$$

Puesto que estamos considerando un alfabeto finito y un dominio de tiempo discreto, podemos encontrar una secuencia de naturales

$$1 \leq k_1 < k_2 < \cdots k_i < k_{i+1} \cdots$$

de modo que $bs_{k_i} = bs_{k_j}$ para todo $i,\ j \in \mathbb{N}$. A fin de simplificar la notación, sin pérdida de generalidad, podemos suponer que existe $k \in \mathbb{N}$ tal que $bs_k = bs_{k'}$ para $k' \geq k$. Tomando $bs = bs_k$, tenemos entonces dos posibilidades:

- $\mathsf{lon}(bs) = \mathsf{lon}(bs')$. En tal caso, hay de nuevo dos casos posibles:

  - Existe $t \in \mathcal{T}$ tal para todo $k' \geq k$ se tiene que $\mathsf{nd}(A_{k'}) \leq t$. Puesto que estamos considerando un alfabeto finito y un dominio de tiempo discreto, podemos suponer que

  $$A_{k'} = A_k \quad \text{para} \quad k' \geq k$$

  Tomando entonces $A = A_k$ tenemos que $b = bs \cdot A \in B$ y $b \ll b'$.

  - Para cada $t \in \mathcal{T}$ existe $l_t \geq k$ tal que $\mathsf{nd}(A_{l_t}) \geq t$. Puesto que estamos suponiendo un dominio de tiempo discreto y un alfabeto finito podemos suponer que

  $$A_{l_t} \upharpoonright t = A_{l_{t'}} \upharpoonright t \quad \text{para} \quad t' \geq t$$

  Tomando entonces como A el estado

  $$A = \bigcup_{t \in \mathcal{T}} A_{l_t} \upharpoonright t$$

  y se verifica que $b = bs \cdot A \in B$ y $b \ll b'$

- $\mathsf{lon}(bs) < \mathsf{lon}(bs')$. Supongamos que $\mathsf{lon}(bs) = n$ y $\mathsf{lon}(bs') = n$, de modo que

  $$bs = A_1 a_1 t_1 \cdots A_l a_l t_l \cdots A_n a_n t_n \quad \text{y} \quad bs = A'_1 a_1 t_1 \cdots A'_l a_l t_l$$

  Tenemos entonces que $\mathsf{nd}(A_{k'}) \leq t_{l+1}$ para $k' \geq k$. Puesto que estamos considerando un dominio de tiempo discreto y un alfabeto finito, podemos suponer que $A_{k'} = A_k$ para $k' \geq k$, y tomando $A = A_k$, tenemos que $b = bs \cdot A \in B$ y $b \ll b'$.

□

4.4.1 Monotonía y Continuidad

En la presente sección probaremos la monotonía y la continuidad de los operadores semánticos con respecto al preorden $\prec$. A fin de no recargar con tecnicismos el desarrollo de la misma algunas de las correspondientes demostraciones se postergan hasta el apéndice A.

4.4.1.1 Prefijo de Acciones Ocultas

Veamos en primer lugar que el prefijo con una acción interna es monótono:

Proposición 4.4.12 Sean B y B' dos conjuntos consistentes de barbas. Si $B \prec B'$ y $t \in \mathcal{T}$ entonces $\mathcal{B}_{\mathrm{con}}[\![\tau t;]\!]B \prec \mathcal{B}_{\mathrm{con}}[\![\tau t;]\!](B')$.

Demostración. Simplemente hay que tener en cuenta que

$$b \prec b' \qquad \Rightarrow \qquad b + t \prec b' + t$$

□

Una vez que tenemos que este operador es monótono podemos pasar a demostrar que es continuo:

Proposición 4.4.13 Siendo $\mathcal{B}$ una cadena de conjuntos de barbas consistentes:

$$B_1 \prec B_2 \prec B_3 \cdots B_k \prec B_{k+1} \prec \cdots$$

se tiene que $\mathcal{B}_{\mathrm{con}}[\![\tau t;]\!](\mathsf{lub}(\mathcal{B})) = \mathsf{lub}\{\mathcal{B}_{\mathrm{con}}[\![\tau t;]\!](B_i) \mid i \in \mathbb{N}\}$.

□

4.4.1.2 Prefijo con Acciones Visibles

Al igual que en el caso anterior, probamos en primer lugar la monotonía del operador con respecto al preorden $\prec$.

Proposición 4.4.14 Sean B y B' dos conjuntos consistentes de barbas tales que $B \prec B'$, para cada $at \in TAct$, se tiene $\mathcal{B}_{\mathrm{con}}[\![at;]\!]B \prec \mathcal{B}_{\mathrm{con}}[\![at;]\!](B')$.

Demostración. Simplemente hay que tener en cuenta que

$$b \prec b' \qquad \Longleftrightarrow \qquad \emptyset at \cdot b \prec \emptyset at \cdot b'$$

□

-Supone

Atacamos a continuación la continuidad del operador de prefijo:

Proposición 4.4.15 Para cada cadena de barbas $\mathcal{B}$:

$$B_1 \prec B_2 \prec \cdots B_i \prec B_{i+1} \prec \cdots$$

se tiene $\mathcal{B}_{\text{con}}[\![at;]\!](\mathsf{lub}(\mathcal{B})) = \mathsf{lub}\{\mathcal{B}_{\text{con}}[\![at;]\!](B_i) \mid i \in \mathbb{N}\}$.

□

4.4.1.3 Elección Interna

De nuevo vemos en primer lugar demostraremos la monotonía de este operador. Como se trata de un operador binario, basta probar la monotonía respecto de cada una de las componentes:

Proposición 4.4.16 Siendo B, B_1 y B_1' conjuntos consistentes de barbas, tenemos que si $B_1 \prec B_1'$, entonces $\mathcal{B}_{\text{con}}[\![\sqcap]\!](B_1, B) \prec \mathcal{B}_{\text{con}}[\![\sqcap]\!](B_1', B)$ y $\mathcal{B}_{\text{con}}[\![\sqcap]\!](B, B_1) \prec \mathcal{B}_{\text{con}}[\![\sqcap]\!](B, B_1')$.

Demostración. Basta observar que este operador no es otra cosa que la unión conjuntista.

□

La continuidad para este operador queda expresada por la siguiente

Proposición 4.4.17 Siendo $\mathcal{B}_1$ y $\mathcal{B}_2$ dos cadenas de conjuntos consistentes de barbas

$$B_{11} \prec B_{12} \prec B_{13} \cdots \quad \text{y} \quad B_{21} \prec B_{22} \prec B_{23} \cdots$$

se tiene $\mathcal{B}_{\text{con}}[\![\sqcap]\!](\mathsf{lub}(\mathcal{B}_1), \mathsf{lub}(\mathcal{B}_2)) = \mathsf{lub}\{\mathcal{B}_{\text{con}}[\![\sqcap]\!](B_{1i}, B_{2i}) \mid i \in \mathbb{N}\}$.

□

Demostración. Pese a que la la demostración es bastante sencilla, preferimos dejarla para el apéndice A.

□

4.4.1.4 Elección Externa

Al igual que en el caso anterior, para demostrar la monotonía, es suficiente probarla con respecto a cada una de las componentes:

Proposición 4.4.18 Siendo B, B_1 y B_1' conjuntos consistentes de barbas con $B_1 \prec B_1'$, se tiene $\mathcal{B}_{\text{con}}[\![\Box]\!](B_1, B) \prec \mathcal{B}_{\text{con}}[\![\Box]\!](B_1', B)$ y $\mathcal{B}_{\text{con}}[\![\Box]\!](B, B_1) \prec \mathcal{B}_{\text{con}}[\![\Box]\!](B, B_1')$.

□

En este caso, teniendo en cuenta que acabamos de demostrar la monotonía, para demostrar la continuidad del operador elección externa basta considerar la siguiente:

Proposición 4.4.19 Siendo $\mathcal{B}_1$ y $\mathcal{B}_2$ dos cadenas de conjuntos consistentes de barbas

$$B_{11} \prec B_{12} \prec B_{13} \cdots \quad \text{y} \quad B_{21} \prec B_{22} \prec B_{23} \cdots$$

tenemos $\mathcal{B}_{\mathrm{con}}[\![\square]\!](\mathsf{lub}(\mathcal{B}_1), \mathsf{lub}(\mathcal{B}_2)) \prec \mathsf{lub}\{\mathcal{B}_{\mathrm{con}}[\![\square]\!](B_{1i}, B_{2i}) \mid i \in \mathbb{N}\}$.

□

4.4.2 Semántica de Procesos Recursivos

La definición de la semántica de los procesos recursivos la haremos con técnicas de punto fijo con respecto al preorden $\prec$. En las secciones anteriores hemos demostrado que los distinto operadores del lenguaje son contínuos con respecto al preorden $\prec$. Puesto que $\prec$ se trata simplemente de un preorden (no es orden parcial), los puntos fijos mínimos no serán únicos, aunque si los son módulo la relación de equivalencia inducida, dada por

$$B_1 \sim B_2 \qquad \Longleftrightarrow \qquad B_1 \prec B_2 \quad \wedge \quad B_2 \prec B_2$$

Tras estas observaciones podemos dar la siguiente

Definición 4.4.20 Siendo $P \in \mathsf{Rec}(\Sigma_{\mathrm{seq}})$ y ρ una asignación de variables, definimos

$$\mathcal{B}_{\mathrm{con}}[\![P]\!]_\rho = \begin{cases} \mathcal{B}_{\mathrm{con}}[\![op]\!](\mathcal{B}_{\mathrm{con}}[\![P_1]\!]_\rho, \ldots, \mathcal{B}_{\mathrm{con}}[\![P_n]\!]_\rho) & \text{si } P = op(P_1, \ldots P_n) \text{ y } op \in \Sigma_{\mathrm{seq}} \\ \rho(x) & \text{si } x \in \mathsf{Var} \\ \mathsf{fix}(\lambda B.\ \mathcal{B}_{\mathrm{con}}[\![P_1]\!]_{\rho[B/x]}) & \text{si } P = \mathtt{REC}\, x.P_1 \end{cases}$$

□

Como dijimos anteriormente el punto fijo mínimo será único, módulo la relación $\sim$. Por otro lado, puesto que para $P \in \mathsf{CRec}(\Sigma_{\mathrm{seq}})$ tenemos que $\mathcal{B}_{\mathrm{con}}[\![P]\!]_\rho$ no depende de ρ, escribiremos simplemente $\mathcal{B}_{\mathrm{con}}[\![P]\!]$. Por último recordamos que el punto fijo mínimo de una función se puede obtener como límite (o cota superior) de sus aproximaciones finitas. En nuestro caso tenemos

$$\mathcal{B}_{\mathrm{con}}[\![P]\!] \sim \mathsf{lub}\{\mathcal{B}_{\mathrm{con}}[\![\mathsf{ap}(P, k)]\!] \mid k \in \mathbb{N}\}$$

---

una

Abstracción

Vamos a demostrar la correspondencia para procesos recursivos entre la denotacional y la semántica de pruebas. Dada la caracterización operacional de la semántica de pruebas es suficiente probar

$$\mathsf{Barb}(P) \approx \mathcal{B}_{\text{con}}[\![P]\!]$$

Hemos visto que si $P = \mathtt{REC}\,x.P_1$, $\mathcal{B}_{\text{con}}[\![P]\!]$ viene dado por el punto fijo mínimo de la función que define al proceso. Observemos que mediante la regla **[REC]** de la semántica operacional estamos logrando prácticamente lo mismo.

Lema 4.4.21 Siendo P un proceso, se verifican:

- Si $b \in \mathsf{Barb}(P)$ y $t \leq \mathsf{nd}(b)$, existen $k \in \mathbb{N}$ y una barba b' tales que

$$b_k \in \mathsf{Barb}(\mathsf{ap}(P,k)) \quad \text{y} \quad b_k \upharpoonright t = b \upharpoonright t$$

- Si $b \in \mathsf{Barb}(P)$, $\mathsf{nd}(b) < \infty$ entonces existe $k \in \mathbb{N}$ tal que

$$\forall k' \geq k : \ b \in \mathsf{Barb}(\mathsf{ap}(P,k'))$$

Demostración. Simplemente hay que tener en cuenta que cada computación finita de P puede ser *simulada* por una aproximación suficientemente avanzada del proceso.

□

Proposición 4.4.22 Para cada proceso P se tiene $\mathcal{B}_{\text{con}}[\![P]\!] \ll \mathsf{Barb}(P)$.

Demostración. Puesto que $\mathsf{ap}(P,k)$ es un proceso finito, en virtud del teorema 4.3.3 se verifica

$$\mathcal{B}_{\text{con}}[\![\mathsf{ap}(P,k)]\!] \approx \mathsf{Barb}(\mathsf{ap}(P,k))$$

Por el lema 4.4.21 tenemos que $\mathsf{Barb}(\mathsf{ap}(P,k)) \ll \mathsf{Barb}(P)$, por lo que

$$\mathcal{B}_{\text{con}}[\![\mathsf{ap}(P,k)]\!] \ll \mathsf{Barb}(P)$$

Y entonces, por la proposición 4.4.11, concluimos $\mathcal{B}_{\text{con}}[\![P]\!] \ll \mathsf{Barb}(P)$.

□

Lema 4.4.23 Para cada proceso P se verifican:

- Si $b \in \mathsf{Barb}(\mathsf{ap}(P,k))$ y $\mathsf{nd}(b) \geq t$, entonces existe $b' \in \mathsf{Barb}(P)$ tal que $b \upharpoonright t = b' \upharpoonright t$.

- Si b es una barba tal que $\mathsf{nd}(b) < \infty$, entonces

$$\big(\forall k \in \mathbb{N}\ \exists l \geq k : b \in \mathsf{Barb}(\mathsf{ap}(P, k'))\big) \ \Rightarrow\ b \in \mathsf{Barb}(P)$$

Demostración. Para la primera parte basta tener en cuenta que cualquier computación de $\mathsf{ap}(P, k)$ puede ser simulada por P. Para la segunda, hemos de tener en cuenta además que la semántica operacional es finitamente ramificada.

□

Proposición 4.4.24 Para todo proceso P se tiene $\mathsf{Barb}(P) \ll \mathcal{B}_{\mathrm{con}}[\![P]\!]$.

Demostración. Puesto que $\mathcal{B}_{\mathrm{con}}[\![P]\!] \sim \mathsf{lub}\{\mathcal{B}_{\mathrm{con}}[\![\mathsf{ap}(P,k)]\!] \mid k \in \mathbb{N}\}$, es suficiente probar

$$\mathsf{Barb}(P) \ll \mathsf{lub}\{\mathcal{B}_{\mathrm{con}}[\![\mathsf{ap}(P,k)]\!] \mid k \in \mathbb{N}\}$$

Tomemos $b \in \mathsf{lub}\{\mathcal{B}_{\mathrm{con}}[\![\mathsf{ap}(P,k)]\!] \mid k \in \mathbb{N}\}$. Detallaremos el caso en el que $\mathsf{nd}(b) = \infty$, el caso en el que $\mathsf{nd}(b) < \infty$ sería muy similar. Por la definición de mínima cota superior, para cada $t \in \mathcal{T}$ existen $l_t \in \mathbb{N}$ y $b_t \in \mathcal{B}_{\mathrm{con}}[\![\mathsf{ap}(P, l_t)]\!]$ de modo que $b_t \upharpoonright t = b \upharpoonright t$. Sin pérdida de generalidad podemos suponer que las longitudes de todas las barbas b_t son iguales, y además que $l_t > l_{t'}$ para $t > t'$. Puesto que $\mathsf{ap}(P, l_t)$ es un proceso finito, por el teorema 4.3.3, para cada b_t existe $b'_t \in \mathsf{Barb}(\mathsf{ap}(P, l_t))$ verificando $b'_t \ll b_t$. Ahora podemos distinguir dos casos

- Existe $t_0 \in \mathcal{T}$ de manera que existe un número infinito de barbas b'_t con $\mathsf{nd}(b'_t) \leq t_0$. Puesto que estamos considerando un alfabeto finito y un dominio de tiempo discreto, podemos suponer que todas ellas son iguales, y entonces tomaremos b' igual a una cualquiera de ellas. Entonces, por la primera parte del lema 4.4.23 tenemos que $b' \in \mathsf{Barb}(P)$. Finalmente, puesto que $b_{t_0} \upharpoonright t = b \upharpoonright t_0$, $b'_t \ll b_t$ y $\mathsf{nd}(b') < t_0$, tenemos $b' \ll b$.

- Existe un número infinito de barbas b'_t de modo que $\mathsf{nd}(b'_t) \geq t$. En tal caso podemos suponer sin pérdida de generalidad, que todas ellas lo cumplen. Sea $b'_t = bs'_t \cdot A'_t$, puesto que estamos considerando un alfabeto finito y un dominio de tiempo discreto, podemos suponer que todas las b-trazas bs'_t son iguales a una cierta que denotaremos por bs', y además que $A'_t \upharpoonright t = A'_{t'} \upharpoonright t$ para $t' \geq t$. Por la segunda parte del lema 4.4.23, para cada $t \in \mathcal{T}$ existe un estado A''_t tal que $A''_t \upharpoonright t = A'_t \upharpoonright t$ y $bs' \cdot A''_t \in \mathsf{Barb}(P)$. Tomamos entonces el estado

$$A' = \bigcup_{t \in \mathcal{T}} A''_t \upharpoonright t$$

puesto que $\mathcal{A}(\mathsf{Barb}(P), bs')$ es temporalmente compacto $A' \in \mathcal{A}(\mathsf{Barb}(P), bs')$, de modo que $b' = bs' \cdot A' \in \mathsf{Barb}(P)$, y por tanto $b' \ll b$.

---

gran

□

Por último como combinación de las proposiciones 4.4.22 y 4.4.24, tenemos el resultado que buscábamos:

Teorema 4.4.25 Para cada proceso P, se tiene $\mathsf{Barb}(P) \approx \mathcal{B}_{\text{con}}[\![P]\!]$.

□

Y entonces como corolario obtenemos el resultado de abstracción de la semántica denotacional respecto a la de pruebas:

Corolario 4.4.26 Para cualesquiera son procesos P y Q, se tiene

$$P \sqsubseteq_{\approx} Q \quad \Longleftrightarrow \quad \mathcal{B}_{\text{con}}[\![P]\!] \ll \mathcal{B}_{\text{con}}[\![Q]\!]$$

□

Capítulo 5

Sistema de Ecuaciones

En el presente capítulo veremos un método puramente algebraico que nos permitirá caracterizar la semántica de pruebas: una axiomatización de la equivalencia obtenida vía un sistema de ecuaciones. Aunque para simplificar hablaremos genéricamente de un sistema de ecuaciones, más exactamente tendremos un sistema de ecuaciones e inecuaciones que denominaremos E. Las inecuaciones resultarán imprescindibles cuando tratemos los procesos recursivos, para procesos finitos bastaría tener un sistema de ecuaciones.

Por tanto tendremos una serie de reglas y axiomas que nos permitirán deducir ecuaciones e inecuaciones del tipo

$$P =_{\mathrm{E}} Q \quad \text{y} \quad P <_{\mathrm{E}} Q$$

Las igualdades y desigualdades que se deduzcan serán correctas con respecto a la semántica de pruebas de modo que

$$\begin{array}{lcl} P <_{\mathrm{E}} Q & \Rightarrow & P \underset{\sim}{\sqsubset} Q \\ P =_{\mathrm{E}} Q & \Rightarrow & P \underset{\sim}{\equiv} Q \end{array}$$

Pero además el sistema formado por dichas reglas será también completo, de modo que

$$\begin{array}{lcl} P \underset{\sim}{\sqsubset} Q & \Rightarrow & P <_{\mathrm{E}} Q \\ P \underset{\sim}{\equiv} Q & \Rightarrow & P =_{\mathrm{E}} Q \end{array}$$

Para desarrollar dicho axiomas nos basaremos en la semántica denotacional estudiada en el capítulo anterior, cuya abstracción respecto de la semántica de pruebas ha quedado allí probada, de modo que

$$\begin{array}{lcl} \mathcal{B}_{\mathrm{con}}[\![P]\!] \ll \mathcal{B}_{\mathrm{con}}[\![Q]\!] & \Longleftrightarrow & P \underset{\sim}{\sqsubset} Q \\ \mathcal{B}_{\mathrm{con}}[\![P]\!] \approx \mathcal{B}_{\mathrm{con}}[\![Q]\!] & \Longleftrightarrow & P \underset{\sim}{\equiv} Q \end{array}$$

---

economía

En consecuencia bastará demostrar que el conjunto de axiomas y reglas es correcto y completo con respecto a dicha semántica denotacional:

$$\begin{array}{lcl} \mathcal{B}_{\text{con}}[\![P]\!] \ll \mathcal{B}_{\text{con}}[\![Q]\!] & \Longleftrightarrow & P <_{\mathrm{E}} Q \\ \mathcal{B}_{\text{con}}[\![P]\!] \approx \mathcal{B}_{\text{con}}[\![Q]\!] & \Longleftrightarrow & P =_{\mathrm{E}} Q \end{array}$$

Para demostrar que el sistema de axiomas es correcto, basta ir viendo que cada una de las reglas y axiomas del sistema son correctos, cosa por lo general bastante sencilla. La demostración de la completitud suele resultar bastante más complicada.

Como es habitual comenzaremos restringiéndonos a los procesos finitos, para posteriormente introducir reglas infinitarias que permitan manejar los procesos infinitos. Para demostrar la completitud del sistema de reglas y axiomas, demostraremos que utilizando los mismos cada proceso se puede pasar a forma normal, siendo relativamente fácil demostrar la completitud para dichas formas normales. En el caso de las álgebras de procesos no temporizadas, las formas normales son del estilo

$$\bigsqcap_{A\in\mathcal{A}} \mathop{\Box}_{a\in A} a\,;P_a$$

donde $\mathcal{A}$ es un conjunto de estados cerrado bajo unión y convexidad. Nosotros, en general, no podremos exigir dicho cierre, si bien necesitamos en su lugar un nuevo tipo de cierre que puede ser visto como una generalización temporal de dicho cierre.

Antes de empezar dando ecuaciones recordemos que estamos manejando una serie de equivalencias, en principio diferentes:

- La igualdad sintáctica que denotamos con el símbolo $=$.
- La igualdad semántica, la inducida por la semántica de pruebas o equivalentemente por semántica denotacional. Como quiera que coinciden, denotamos ambas mediante el símbolo $\approx$.
- La igualdad que se deducirá por medio de las reglas y axiomas que constituyen en sistema de ecuaciones E, que denotaremos por $=_{\mathrm{E}}$.

5.1 Conjuntos Pseudo-Convexos de Estados

Como hemos indicado más arriba, en las álgebras de procesos sin tiempo, se exige que los estados que aparecen en las formas normales sean cerrados bajo convexidad. Ello es así pues por ejemplo, un proceso que tenga los estados $\{a\}$ y $\{b, c\}$, no puede distinguirse de otro proceso que tenga junto a los dos estados anteriores el estado $\{a, b\}$; en concreto los procesos

$$P = a\,;\texttt{STOP} \sqcap (b\,;\texttt{STOP} \,\Box\, c\,;\texttt{STOP}) \quad \text{y} \quad Q = P \sqcap (a\,;\texttt{STOP} \,\Box\, c\,;\texttt{STOP})$$

son indistinguibles en un álgebra de procesos tradicional. Sin embargo, en nuestro modelo temporizado no sucede tal cosa, como ilustra el siguiente ejemplo:

Ejemplo 5.1.1 Consideremos el proceso $P = (a1\,;\texttt{STOP}) \sqcap (b1\,;\texttt{STOP} \,\Box\, c2\,;\texttt{STOP})$. Obviamente tiene dos estados: $\{a1\}$ y $\{b1, c2\}$. Consideremos ahora el proceso

$$Q = P \sqcap (a1\,;\texttt{STOP} \,\Box\, c2\,;\texttt{STOP})$$

Los estados de Q son los estados de P más el estado $\{a1, c2\}$ que pertenece al cierre convexo del conjunto de estados de P. Sin embargo los procesos P y Q pueden distinguirse fácilmente; al efecto basta observar que en el proceso P la acción c en el instante 2 siempre va acompañada de la acción b en el instante 1, sin embargo en Q esto no es cierto. Por lo tanto, si consideramos la prueba

$$T = (b1\,;\texttt{OK}) \,\Box\, (\tau 2\,;\texttt{OK}) \,\Box\, (c2\,;\texttt{STOP})$$

tenemos que P must T pero Q $\not\!\text{must}$ T.

En cambio si consideramos el proceso $Q' = P \sqcap (a1\,;\texttt{STOP} \,\Box\, b1\,;\texttt{STOP})$, éste incorpora el estado $A = \{a1, b1\}$. En este caso, dicha adición sí que no afecta a la semántica. Fijémonos que el razonamiento que antes basamos en la acción c en el instante 2 no producirá los mismos frutos. En concreto, las razones que justifican el que los estados existentes permitan introducir un estado nuevo A sin alterar la semántica son los siguientes:

- Existe en el conjunto de partida un estado A' (en nuestro caso $\{a1\}$) tal que $A' \ll A$.
- Para cada acción temporizada at que aparezca en el estado nuevo A, existe un estado A' en el conjunto original verificando $at \in A'$ y $A' \upharpoonright t \subseteq A$. En concreto en nuestro caso tenemos:
  - para $a1 \in A$ tenemos el estado $A' = \{a1\}$,
  - para $b1 \in A$ tenemos el estado $A' = \{b1, c2\}$.

$\Box$

El ejemplo anterior nos muestra que al *hacer el cierre* adecuado de los estados de un proceso, podemos incluir algunos de los estados pero otros no. En concreto los estados que se pueden incluir constituyen lo que llamamos el cierre *pseudo-convexo* del conjunto de estados en cuestión.

de

Definición 5.1.2 Siendo $\mathcal{A}$ un conjunto de estados diremos que un estado A está en el cierre pseudo-convexo del conjunto $\mathcal{A}$, $A \in \mathsf{pc}(\mathcal{A})$, si y sólo si verifican

1. Existe $A' \in \mathcal{A}$ tal que $A' \ll A$ y no existe $A' \in \mathcal{A}$ tal que $A' \ll A$ y $\mathsf{nd}(A') < \mathsf{nd}(A)$.*

2. Para cada $at \in A$ existe $A' \in \mathcal{A}$ de modo que $at \in A'$ y $A' \rceil t \subseteq A$.

Diremos que $\mathcal{A}$ está cerrado bajo pseudo-convexidad, o es pseudo-convexo, si $\mathcal{A} = \mathsf{pc}(\mathcal{A})$.

□

Observemos en primer lugar que todos los estados que están en el cierre pseudo-convexo, están también en el cierre convexo. También es interesante observar que en presencia de divergencias estados iniciales del conjunto original $\mathcal{A}$ podrían desaparecer del cierre $\mathsf{pc}(\mathcal{A})$. Además nuestra elección a la hora de definir $\mathsf{pc}(\mathcal{A})$ nos conduce a que dicho conjunto no contenga a todos los estados que podría contener manteniendo la noción de equivalencia semántica. Para ilustrar todo esto consideremos el siguiente

Ejemplo 5.1.3 Consideremos los procesos

$$P = \mathtt{DIV} \qquad \text{y} \qquad Q = \mathtt{DIV} \sqcap (a0\,;\mathtt{STOP})$$

Estos procesos son equivalentes bajo la semántica de pruebas. Por otra parte tenemos que

$$\mathcal{A}(P) = \Big\{\{\Omega 0\}\Big\} \qquad \text{y} \qquad \mathcal{A}(Q) = \Big\{\{\Omega 0\}, \{a0\}\Big\}$$

Esto nos indica que a la hora de definir el cierre del conjunto de estados $\mathcal{A}(P)$, somos *libres* en principio de introducir el estado $\{a0\}$, y razonando de manera totalmente análoga cualquier otro. De manera recíproca, al calcular el cierre del conjunto de estados del proceso Q, podemos incluir o no el estado $\{a0\}$ en el cierre.

Nosotros hemos optado por no incluir el estado $\{a0\}$ en ninguno de los casos. En general tenemos que bajo la definición que hemos dado tenemos que la única forma de que un estado A verifique $A \in \mathcal{A}$ y $A \notin \mathsf{pc}(\mathcal{A})$ es que exista $A' \in \mathcal{A}$ de modo que $\mathsf{nd}(A') < \mathsf{nd}(A)$ y $A' \ll A$.

□

Hemos dicho que $\mathsf{pc}(\mathcal{A})$ es un cierre, pero dicha afirmación precisa de la correspondiente demostración.

*Esta condición implica en particular que existe $A' \in \mathcal{A}$ de modo que $A' \ll A$ y $\mathsf{nd}(A') = \mathsf{nd}(A)$.

Proposición 5.1.4 Para todo conjunto de estados $\mathcal{A}$, se tiene

$$\mathsf{pc}(\mathcal{A}) = \mathsf{pc}(\mathsf{pc}(\mathcal{A}))$$

Demostración. Para ver la igualdad de los dos conjuntos veamos por separado cada una de las inclusiones.

$\boxed{\mathsf{pc}(\mathsf{pc}(\mathcal{A})) \subseteq \mathsf{pc}(\mathcal{A})}$

Sea $A \in \mathsf{pc}(\mathsf{pc}(\mathcal{A}))$. Para comprobar que $A \in \mathsf{pc}(\mathcal{A})$, veamos que se cumplen las condiciones de la definición 5.1.2.

1. Puesto que $A \in \mathsf{pc}(\mathsf{pc}(\mathcal{A}))$, existe $A_1 \in \mathsf{pc}(\mathcal{A})$ de modo que $A_1 \ll A$ y $\mathsf{nd}(A_1) = \mathsf{nd}(A)$. Como $A_1 \in \mathsf{pc}(\mathcal{A})$, existirá $A_2 \in \mathcal{A}$ tal que $A_2 \ll A_1$ y no existe $A_2' \in \mathcal{A}$ tal que $\mathsf{nd}(A_2') < \mathsf{nd}(A_1)$. Por lo tanto tenemos que $A_2 \ll A$ y no existe $A_2' \in \mathcal{A}$ tal que $\mathsf{nd}(A_2') < \mathsf{nd}(A)$.
2. Sea ahora $at \in A$. Entonces existe $A_1 \in \mathsf{pc}(\mathcal{A})$ tal que $at \in A_1$ y $A_1 \upharpoonright t \subseteq A$. Puesto que $A_1 \in \mathsf{pc}(\mathcal{A})$ existirá $A_2 \in \mathcal{A}$ tal que $at \in A_2$ y $A_2 \upharpoonright t \subseteq A_1$, y por tanto $A_2 \upharpoonright t \subseteq A$.

$\boxed{\mathsf{pc}(\mathcal{A}) \subseteq \mathsf{pc}(\mathsf{pc}(\mathcal{A}))}$

Supongamos ahora que $A \in \mathsf{pc}(\mathcal{A})$. Si $A \notin \mathsf{pc}(\mathsf{pc}(\mathcal{A}))$ es porque existe $A_1 \in \mathsf{pc}(\mathcal{A})$ verificando $A_1 \ll A$ y $\mathsf{nd}(A_1) < \mathsf{nd}(A)$. Puesto que $A_1 \in \mathsf{pc}(\mathcal{A})$ existirá $A_2 \in \mathcal{A}$ tal que $A_2 \ll A_1$. De modo que $A_2 \ll A$ y $\mathsf{nd}(A_2) < \mathsf{nd}(A)$, y entonces $A \notin \mathsf{pc}(\mathcal{A})$, lo que contradice nuestra hipótesis inicial.

$\square$

Damos a continuación dos propiedades interesantes del cierre pseudo-convexo de un conjunto de estados:

Proposición 5.1.5 Para todo conjunto de estados $\mathcal{A}$ se tienen:

- Si $A \in \mathcal{A}$, existe $A' \in \mathsf{pc}(\mathcal{A})$ tal que $\mathsf{nd}(A') \leq \mathsf{nd}(A)$ y $Act(A') = A \upharpoonright \mathsf{nd}(A')$.
- Si $\mathcal{A}$ es finito, y cada $A \in \mathcal{A}$ también es finito, entonces $\mathsf{pc}(\mathcal{A})$ es finito y cada $A' \in \mathsf{pc}(\mathcal{A})$ también lo es.

Demostración. Para demostrar el primer punto consideremos $A \in \mathcal{A}$. Si $A \notin \mathsf{pc}(\mathcal{A})$ es porque existe $A_1 \in \mathcal{A}$ tal que $A_1 \ll A$ y $\mathsf{nd}(A_1) < \mathsf{nd}(A)$. Lo mismo ocurre con A_1: si

tiempo

$A_1 \notin \mathsf{pc}(\mathcal{A})$ es porque existe $A_2 \in \mathcal{A}$ tal que $A_2 \ll A_1$ y $\mathsf{nd}(A_1)$. Vamos así construyendo una sucesión de estados

$$A = A_0 \ll A_1 \cdots A_{k-1} \ll A_k \ll \cdots$$

de manera que $\mathsf{nd}(A_{k-1}) < \mathsf{nd}(A_k)$. Puesto que trabajamos sobre un dominio de tiempo discreto, la cadena no puede ser infinita y por tanto tiene que existe un $A_k \in \mathcal{A}$ de modo que $A_k \in \mathsf{pc}(\mathcal{A})$. Si consideramos ahora el estado

$$A' = A \upharpoonright \mathsf{nd}(A_k)$$

se puede comprobar fácilmente que $A' \in \mathsf{pc}(\mathcal{A})$ y verifica las condiciones requeridas.

Para demostrar el segundo punto tomamos el conjunto

$$\mathfrak{A} = \bigcup_{A \in \mathcal{A}} A \subsetneq Act_\Omega \times \mathcal{T}$$

Puesto que $\mathcal{A}$ es finito y cada $A \in \mathcal{A}$ es finito, tenemos que $\mathfrak{A}$ es finito. Por otro lado, $\mathsf{pc}(\mathcal{A}) \subseteq \mathcal{P}(\mathfrak{A})^\dagger$, y puesto que $\mathfrak{A}$ es finito, concluimos que $\mathsf{pc}(\mathcal{A})$ también lo es. Además puesto que cada $A' \in \mathsf{pc}(\mathcal{A})$ verifica $A' \subseteq \mathfrak{A}$, deducimos también que A' es finito.

□

5.2 Procesos Finitos

Comenzaremos el estudio del sistema de ecuaciones por el caso más simple, es decir, los procesos finitos. Presentaremos un conjunto de ecuaciones, que demostraremos que es correcto y completo con respecto a la semántica denotacional. Puesto que la semántica denotacional es totalmente abstracta con respecto a la semántica de pruebas, las ecuaciones serán también correctas y completas con respecto a la semántica de pruebas.

En primer lugar, en el cuadro 5.1 tenemos un conjunto de reglas que indican que la relación $\leq$ es una relación de orden y los operadores del lenguaje son monótonos con respecto a ella. En base a ellas tenemos que la relación $=_{\mathrm{E}}$ es una congruencia. La corrección de todas estas reglas es consecuencia inmediata de la monotonía de los operadores con respecto a la relación $\ll$.

A continuación, en la tabla 5.2, tenemos los axiomas de conmutatividad y asociatividad de la elección externa y la elección interna, junto con los de idempotencia y de distributividad. Tal y como se han definido los operadores en la semántica denotacional, es fácil demostrar que todos estos axiomas son correctos. Obsérvese que la distributividad

$^\dagger\mathcal{P}(X)$ denota las partes del conjunto X

[EQ1]	$P <_{\mathrm{E}} P$	**[EQ2]**	$\dfrac{P <_{\mathrm{E}} Q,\ Q <_{\mathrm{E}} P}{P =_{\mathrm{E}} Q}$
[EQ3]	$\dfrac{P <_{\mathrm{E}} Q,\ Q <_{\mathrm{E}} R}{P <_{\mathrm{E}} R}$	**[EQ4]**	$\dfrac{P =_{\mathrm{E}} Q}{P <_{\mathrm{E}} Q,\ Q <_{\mathrm{E}} P}$
[MO1]	$\dfrac{P <_{\mathrm{E}} Q}{et\,;P <_{\mathrm{E}} et\,;Q}\quad e \in \mathcal{E}$		
[MO2]	$\dfrac{P <_{\mathrm{E}} P',\ Q <_{\mathrm{E}} Q'}{P \,\Box\, Q <_{\mathrm{E}} P' \,\Box\, Q'}$	[MO3]	$\dfrac{P <_{\mathrm{E}} P',\ Q <_{\mathrm{E}} Q'}{P \sqcap Q <_{\mathrm{E}} P' \sqcap Q'}$

Tabla 5.1: Relación de equivalencia y monotonía.

entre el operador de elección externa y el de elección interna sólo funciona en un *sentido*; es decir, la ecuación

$$P \sqcap (Q \,\Box\, R) =_{\mathrm{E}} (P \sqcap Q) \,\Box\, (P \sqcap R)$$

no es correcta; como muestra el siguiente

Ejemplo 5.2.1 *Consideremos los siguientes conjuntos consistentes de barbas:*

$$\begin{array}{rcl} B_1 & = & \{\{a1\}, \varnothing a1 \varnothing\} \\ B_2 & = & \{\{b0\}, \varnothing b0 \varnothing\} \\ B_3 & = & \{\{c1\}, \varnothing c1 \varnothing\} \end{array}$$

Tenemos que

$$B = \mathcal{B}_{\text{con}}[\![\sqcap]\!](B_1, \mathcal{B}_{\text{con}}[\![\Box]\!](B_2, B_3)) = \left\{ \begin{array}{l} \{a1\}, \{b0, c1\}, \\ \varnothing a1 \varnothing, \{b0\} c1 \varnothing, \varnothing b0 \varnothing \end{array} \right\}$$

$$B' = \mathcal{B}_{\text{con}}[\![\sqcap]\!](\mathcal{B}_{\text{con}}[\![\Box]\!](B_1, B_2), \mathcal{B}_{\text{con}}[\![\Box]\!](B_1, B_3)) = \left\{ \begin{array}{l} \{b0, a1\}, \{a1, c1\}, \\ \varnothing b0 \varnothing, \{b0\} a1 \varnothing, \varnothing a1 \varnothing, \varnothing c1 \varnothing \end{array} \right\}$$

En consecuencia $b' = \varnothing c1 \varnothing \in B'$, pero no existe ningún $b \in B$ verificando $b \ll b'$, y por tanto $B \not\ll B'$.

$\Box$

A las reglas y axiomas que hemos visto hasta ahora las podríamos calificar de *habituales* en las álgebras de procesos ordinarias. A partir de ahora nos centramos en una serie de reglas más específicas para la extensión temporal que estamos estudiando. Comenzaremos dando una serie de definiciones y notaciones auxiliares:

-dijo

[COM1]	$P \,\Box\, Q =_{\mathrm{E}} Q \,\Box\, P$
[COM2]	$P \sqcap Q =_{\mathrm{E}} Q \sqcap P$
[ASO1]	$P \,\Box\, (Q \,\Box\, R) =_{\mathrm{E}} (P \,\Box\, Q) \,\Box\, R$
[ASO2]	$P \sqcap (Q \sqcap R) =_{\mathrm{E}} (P \sqcap Q) \sqcap R$
[ID1]	$P =_{\mathrm{E}} P \,\Box\, P$
[ID2]	$P =_{\mathrm{E}} P \sqcap P$
[ID3]	$\mathtt{STOP} \,\Box\, P =_{\mathrm{E}} P$
[DIS1]	$at\,; P \,\Box\, at\,; Q =_{\mathrm{E}} at\,; (P \sqcap Q)$
[DIS2]	$at\,; P \sqcap at\,; Q =_{\mathrm{E}} at\,; (P \sqcap Q)$
[DIS3]	$P \,\Box\, (Q \sqcap R) =_{\mathrm{E}} (P \,\Box\, Q) \sqcap (P \,\Box\, R)$

Tabla 5.2: Conmutatividad, asociatividad, idempotencia, distributividad

Definición 5.2.2

- Siendo $t \in \mathcal{T} \cup \{\infty\}$ definimos $\mathtt{DIV}(t) = \begin{cases} \tau t\,; \mathtt{DIV} & \text{si } t < \infty \\ \mathtt{STOP} & \text{si } t = \infty \end{cases}$

- Dada una familia finita de de procesos definimos la elección externa generalizada entre sus miembros, de la forma siguiente:

$$\begin{aligned} \Box_{Q \in \varnothing} Q &= \mathtt{STOP} \\ \Box_{Q \in \{P\} \cup \mathcal{P}} Q &= P \,\Box\, \Box_{Q \in \mathcal{P}} Q \end{aligned}$$

  Análogamente, para conjuntos finitos, pero no vacíos en este caso definimos la elección interna generalizada entre ellos:

$$\begin{aligned} \bigsqcap_{Q \in \{P\}} Q &= P \\ \bigsqcap_{Q \in \{P\} \cup \mathcal{P}} Q &= P \sqcap \bigsqcap_{Q \in \mathcal{P}} Q \end{aligned}$$

  Estas definiciones vienen justificadas por el hecho de que tanto la elección externa como la interna son conmutativas y asociativas.

- Siendo A un estado finito con $t_A = \mathsf{nd}(A)$ su instante de indefinición, si para cada $at \in A$ tenemos definido un proceso asociado P_{at}, entonces tomaremos

$$\Box_{at \in A} P_{at} = \mathtt{DIV}(t_A) \,\Box\, \Box_{at \in A} at\,; P_{at}$$

□

Antes de continuar observemos que, gracias al axioma de idempotencia [**ID2**], el siguiente axioma es correcto

$$\left(\bigsqcap_{P \in \mathcal{P}} P\right) \sqcap \left(\bigsqcap_{Q \in \mathcal{Q}} Q\right) =_{\mathrm{E}} \bigsqcap_{P \in \mathcal{P} \cup \mathcal{Q}} P$$

Ello quiere decir que siempre que hagamos la elección interna de dos procesos que sean a su vez sendas elecciones internas generalizadas, podremos quitar los procesos que aparezcan en la intersección. Para simplificar la notación, siempre que aparezcan elecciones de este estilo, supondremos que estaremos eliminando las posibles repeticiones. Todo lo dicho anteriormente vale exactamente lo mismo para la elección externa, puesto que gracias a los axiomas [**ID1**] y [**ID3**] tenemos

$$\left(\bigsqcup_{P \in \mathcal{P}} P\right) \Box \left(\bigsqcup_{Q \in \mathcal{Q}} Q\right) =_{\mathrm{E}} \bigsqcup_{P \in \mathcal{P} \cup \mathcal{Q}} P$$

[**I1**]	$\tau t\,;(et'\,;P) =_{\mathrm{E}} e(t+t')\,;P$	$e \in \mathcal{E}$
[**I2**]	$\tau t\,;(P \Box Q) =_{\mathrm{E}} (\tau\,;P) \Box (\tau t\,;Q)$	
[**I3**]	$\tau t\,;(P \sqcap Q) =_{\mathrm{E}} (\tau\,;P) \sqcap (\tau t\,;Q)$	
[**I4**]	$\tau t\,;\mathtt{STOP} =_{\mathrm{E}} \mathtt{STOP}$	
[**DIV1**]	$\mathtt{DIV}(t) \Box \mathtt{DIV}(t') =_{\mathrm{E}} \mathtt{DIV}(t)$	si $t \leq t'$
[**DIV2**]	$\mathtt{DIV}(t) \Box \bigsqcup_{at \in A} at\,;P_{at} =_{\mathrm{E}} \mathtt{DIV}(t) \Box \bigsqcup_{at \in A \upharpoonright t} at\,;P_{at}$	

Tabla 5.3: Ecuaciones específicas sencillas.

Axiomas

Para completar el conjunto de axiomas, para el caso de procesos finitos, necesitamos los siguientes:

- Siendo $t, t' \in \mathcal{T}$, tenemos los siguientes axiomas:

$$\begin{array}{ll} [\mathbf{I1}] & \tau t\,;(et'\,;P) =_{\mathrm{E}} e(t+t')\,;P \qquad e \in \mathcal{E} \\ [\mathbf{I2}] & \tau t\,;(P \Box Q) =_{\mathrm{E}} (\tau t\,;P) \Box (\tau t\,;Q) \\ [\mathbf{I3}] & \tau t\,;(P \sqcap Q) =_{\mathrm{E}} (\tau t\,;P) \sqcap (\tau t\,;Q) \\ [\mathbf{I4}] & \tau t\,;\mathtt{STOP} =_{\mathrm{E}} \mathtt{STOP} \end{array}$$

el

[PC1] $(P_A \,\Box\, a_0t_0\,;P) \sqcap (Q_B \,\Box\, a_0t_0\,;Q) =_{\mathrm{E}} (P_A \,\Box\, a_0t_0\,;P) \sqcap (Q_B \,\Box\, a_0t_0\,;(P \sqcap Q))$

donde $P_A = \Box_{at\in A} P_{at}, \quad Q_B = \Box_{at\in B} Q_{at},$
$t_0 < \min(\mathsf{nd}(A), \mathsf{nd}(B)) \quad \text{y} \quad A \upharpoonright t_0 \subseteq B$

[PC2] $\displaystyle \Box_{at\in A} P_{at} \sqcap \Box_{at\in B} Q_{at} =_{\mathrm{E}} \Box_{at\in A} P_{at} \sqcap \Box_{at\in B'} Q_{at}$

donde $A \ll B, \quad t_A = \mathsf{nd}(A) < \infty, \quad B' = B \upharpoonright t_A \cup \{\Omega t_A\}$

[PC3] $\displaystyle \bigsqcap_{A\in\mathcal{A}} P_A =_{\mathrm{E}} \bigsqcap_{A\in\mathcal{A}\cup\{B\}} P_A$

donde $\displaystyle P_A =_{\mathrm{E}} \Box_{at\in A} P^A_{at}$, para cada $A \in \mathcal{A}$
$B \notin \mathcal{A}, \quad \exists A \in \mathcal{A} :\ A \ll B$
$\forall at \in B\ \exists A \in \mathcal{A} :\ at \in A \wedge A \upharpoonright t \subseteq B$
$\displaystyle P_B = \Box_{at\in B} P^B_{at}$ donde $P^B_{at} = \bigsqcap\{P^A_{at} \mid A \in \mathcal{A},\ at \in A,\ A \upharpoonright t \subseteq B\}$

Tabla 5.4: Ecuaciones específicas para el cierre pseudo-convexo.

- Si $t, t' \in \mathcal{T} \cup \{\infty\}$ verifican $t \le t'$, tenemos el axioma

$$\textbf{[DIV1]} \quad \mathtt{DIV}(t) \,\Box\, \mathtt{DIV}(t') =_{\mathrm{E}} \mathtt{DIV}(t)$$

- Para cada conjunto finito de acciones temporizadas $A \subsetneq TAct$, consideramos el axioma

$$\textbf{[DIV2]} \quad \mathtt{DIV}(t) \,\Box \Box_{at\in A} P_{at} =_{\mathrm{E}} \mathtt{DIV}(t) \,\Box \Box_{at\in A\upharpoonright t} P_{at}$$

- Sean ahora dos estados finitos A y B, a los que tenemos asociados los procesos

$$P_A = \Box_{at\in A} P_{at} \qquad \text{y} \qquad Q_B = \Box_{bt\in B} Q_{bt}$$

Si $a_0t_0 \in TAct$ es una acción temporizada verificando

$$t_0 < \min(\mathsf{nd}(A), \mathsf{nd}(B)) \quad \text{y} \quad A \upharpoonright t_0 \subseteq B$$

entonces tenemos el axioma

[PC1] $(P_A \,\Box\, a_0t_0\,;P) \sqcap (Q_B \,\Box\, a_0t_0\,;Q) =_{\mathrm{E}} (P_A \,\Box\, a_0t_0\,;P) \sqcap (Q_B \,\Box\, a_0t_0\,;(P \sqcap Q))$

- Si A, B son dos estados finitos tales que $A \ll B$ y $t_A = \mathsf{nd}(A) < \infty$, entonces considerando el estado
$$B' = B \upharpoonright t_A \cup \{\Omega t_A\}$$
tenemos la ecuación
$$\textbf{[PC2]} \quad \bigsqcap_{at \in A} P_{at} \sqcap \bigsqcap_{at \in B} Q_{bt} =_{\mathrm{E}} \bigsqcap_{at \in A} P_{at} \sqcap \bigsqcap_{at \in B'} Q_{at}$$

- Para definir el último, y más complejo de los axiomas vistos hasta ahora, consideremos un conjunto finito no vacío de estados $\mathcal{A}$ tal que cada $A \in \mathcal{A}$ es finito. Supongamos que para cada $A \in \mathcal{A}$ tenemos asociado un proceso
$$P_A = \bigsqcap_{at \in A} P_{at}^A$$
Sea entonces $B \notin \mathcal{A}$ un estado finito tal que
   - existe $A \in \mathcal{A}$ tal que $A \ll B$ y
   - para todo $bt \in B$ existe $A \in \mathcal{A}$ tal que $bt \in A$ y $A \upharpoonright t \subseteq B$.

  Consideramos entonces el proceso:
$$P_B = \bigsqcap_{at \in B} P_{at}^B$$
donde para cada $at \in B$ tenemos el proceso $P_{at}^B = \bigsqcap\{P_{at}^A \mid A \in \mathcal{A},\ at \in A,\ A \upharpoonright t \subseteq B\}$. Tenemos entonces el axioma:
$$\textbf{[PC3]} \quad \bigsqcap_{A \in \mathcal{A}} P_A =_{\mathrm{E}} \bigsqcap_{A \in \mathcal{A} \cup \{B\}} P_A$$

En la tablas 5.3 y 5.4 tenemos recogidas las ecuaciones específicas para nuestro lenguaje temporizado. La demostración de la corrección de todas ellas es bastante sencilla. Utilizando estos axiomas podremos *pasar* cada proceso a forma normal:

Definición 5.2.3 Un proceso finito está en *forma prenormal* si y sólo si tiene la forma
$$P = \bigsqcap_{A \in \mathcal{A}} \left(\bigsqcap_{at \in A} P_{at}^A \right)$$
donde cada uno de los procesos P_{at}^A está también en forma prenormal. Diremos el proceso está en *forma normal* cuando el conjunto $\mathcal{A}$ sea pseudo-cerrado y cada P_{at}^A esté en forma normal.

□

vendedor-.

Nota: El caso base de las definiciones anteriores corresponde al caso en el que

$$\mathcal{A} = \{A_1, A_2 \cdots A_n\}$$

donde cada $A_i = \{\Omega t_i\}$ o bien $A_i = \varnothing$. En este tal tenemos que

$$P = \prod_{i=1}^{n} \mathtt{DIV}(t_i) \quad \text{donde } t_i = \infty \text{ si } A_i = \varnothing$$

En el caso de las formas normales $\mathcal{A}$ ha de ser pseudo-cerrado, de modo que el caso base para las formas normales corresponde a los procesos P sea de la forma

$$P = \mathtt{DIV}(t) \qquad \text{con } t \in \mathcal{T} \cup \{\infty\}$$

Definición 5.2.4 Si P es un proceso en forma prenormal, definimos la profundidad de P, $\mathsf{prof}(P)$, como sigue:

$$\mathsf{prof}(P) = \begin{cases} 1 & \text{si } P = \prod_{i=1}^{n} \mathtt{DIV}(t_i) \\ \\ 1 + max\{\mathsf{prof}(P_{at}^{A}) \mid A \in \mathcal{A} \ \wedge at \in A\} & \text{si } P = \prod_{A \in \mathcal{A}} \square_{at \in A} P_{at}^{A} \end{cases}$$

□

Para demostrar la completitud del sistema de axiomas seguiremos los siguientes pasos:

- Cualquier proceso se puede transformar mediante las ecuaciones en una forma prenormal.
- Cualquier proceso en forma prenormal se puede transformar usando las ecuaciones en un proceso en forma normal.
- Si dos formas normales son semánticamente equivalentes, entonces se puede deducir la igualdad entre ambas.

Proposición 5.2.5 Si P es un proceso finito, entonces existe Q en forma prenormal tal que $P =_{\mathrm{E}} Q$.

Demostración. Hacemos la demostración por inducción estructural sobre P

Casos Base Tenemos $\mathtt{STOP} = \mathtt{DIV}(\infty)$, y $P = \mathtt{DIV} = \mathtt{DIV}(0)$.

Casos Inductivos Hemos de distinguir los siguientes casos:

Prefijo. Sea $P = et\,; P_1$; por hipótesis de inducción tenemos existe Q_1 en forma prenormal verificando $Q_1 =_{\mathrm{E}} P_1$. Si $e = a \in Act$ entonces $Q = at\,; Q_1$ está en forma prenormal y tenemos que $Q =_{\mathrm{E}} P$. Si por el contrario $e = \tau$, entonces aplicando al proceso $\tau t\,; Q_1$ las reglas **[I1]**, **[I2]**, **[I3]** y **[I4]** nos da un proceso Q que está en forma prenormal.

Elección interna. Sea $P = P_1 \sqcap P_2$. Por hipótesis de inducción tenemos procesos Q_1 y Q_2 en forma prenormal de modo que $P_1 =_{\mathrm{E}} Q_1$ y $P_2 =_{\mathrm{E}} Q_2$. Tomamos entonces el proceso $Q = Q_1 \sqcap Q_2$, es claro que Q está en forma normal y $Q =_{\mathrm{E}} P$.

Elección externa. Sea $P = P_1 \sqcap P_2$. Por hipótesis de inducción existirá procesos en forma prenormal Q_1 y R_1 de modo que $Q_1 =_{\mathrm{E}} P_1$ y $Q_2 =_{\mathrm{E}} P_2$, por tanto $P =_{\mathrm{E}} Q_1 \,\Box\, Q_2$. Siendo entonces

$$Q_1 = \bigsqcap_{A \in \mathcal{A}} \; \Box_{at \in A} \; Q1^A_{at} \quad \text{y} \quad Q_2 = \bigsqcap_{B \in \mathcal{B}} \; \Box_{bt \in B} \; Q2^B_{bt}$$

consideraremos el proceso

$$P' = \bigsqcap_{\substack{A \in \mathcal{A} \\ B \in \mathcal{B}}} \left(\Box_{at \in A} Q1^A_{at} \right) \Box \left(\Box_{bt \in B} Q2^B_{bt} \right)$$

Por el axioma **[DIS2]** tenemos que $P' =_{\mathrm{E}} Q_1 \,\Box\, Q_2$. Entonces para cada $A \in \mathcal{A}$ y $B \in \mathcal{B}$ consideramos el estado $C = A \sqcup_{\varnothing} B$, y para cada $ct \in C$ consideramos el proceso Q^C_{ct} dado por

$$Q^C_{ct} = \begin{cases} Q1^A_{ct} & \text{si } ct \in A \setminus B, \\ Q2^A_{ct} & \text{si } ct \in B \setminus A, \\ Q1^A_{ct} \sqcap Q2^B_{ct} & \text{si } ct \in A \cup B. \end{cases}$$

Tomemos entonces

$$Q = \bigsqcap_{\substack{C = A \sqcup_{\varnothing} B \\ A \in \mathcal{A},\; B \in \mathcal{B}}} \; \Box_{ct \in C} \; Q^C_{ct}$$

y tenemos, por un lado, que Q está en forma prenormal, y por otro que aplicando las ecuaciones **[DIS1]**, **[ID3]**, **[DVI1]** y **[DIV2]** obtenemos que $Q =_{\mathrm{E}} P'$, por lo que $Q =_{\mathrm{E}} P$.

$\Box$

Proposición 5.2.6 Si P está en forma prenormal, existe Q en forma normal de modo que $P =_{\mathrm{E}} Q$, además se cumple que $\mathsf{prof}(P) \leq \mathsf{prof}(Q)$.

Los

Demostración. Haremos la demostración por inducción sobre la profundidad de P:

prof(P) = 1 En tal caso tenemos $P = \bigsqcap_{i=1}^{n} \mathtt{DIV}(t_i)$ con $t_i \in \mathcal{T} \cup \{\infty\}$. Si tomamos $t = \min\{t_1 \mid 1 \leq i \leq n\}$, aplicando el axioma [**PC2**] tenemos $P =_{\mathrm{E}} \mathtt{DIV}(t)$ y claramente $\mathtt{DIV}(t)$ está en forma normal.

prof(P) > 1 Sea P de la forma

$$P = \bigsqcap_{A \in \mathcal{A}} \bigsqcup_{at \in A} P_{at}^{A}$$

Consideramos $\mathcal{B} = \mathsf{pc}(\mathcal{A})$. Aplicando la proposición 5.1.5 tenemos que $\mathcal{B}$ es finito. Partimos del proceso P y lo vamos transformando hasta conseguir un proceso P'' en forma prenormal tal que

$$\mathcal{A}(P'') = \mathcal{B} \quad \text{y} \quad P'' =_{\mathrm{E}} P$$

de la siguiente manera:

- Para cada $B \in \mathcal{B} \setminus \mathcal{A}$ tomamos el proceso

  $$P_B = \bigsqcup_{at \in B} at\,; P_{at}^{B} \quad \text{donde} \quad P_{at}^{B} = \bigsqcap\{P_{at}^{A} \mid at \in A \text{ y } A \upharpoonright t \subseteq B\}$$

  Consideramos entonces el proceso

  $$P' = P \sqcap \bigsqcap_{B \in \mathcal{B} \setminus \mathcal{A}} P_B$$

  Aplicando la ecuación [**PC3**] tenemos que $P' =_{\mathrm{E}} P$ y $\mathsf{prof}(P') = \mathsf{prof}(P)$.

- Para cada $A \in \mathcal{A} \setminus \mathcal{B}$ tenemos que existe $A' \in \mathcal{B}$ tal que se verifica

  $$\mathsf{nd}(A') < \mathsf{nd}(A) \quad \text{y} \quad TAct(A') = A \upharpoonright \mathsf{nd}(A')$$

  Por tanto, si tomamos el proceso

  $$P'' = \bigsqcap_{A \in \mathcal{B}} \bigsqcup_{at \in A} P_{at}^{A}$$

  por el axioma [**PC2**] tenemos que $P'' =_{\mathrm{E}} P'$, además $\mathsf{prof}(P'') \leq \mathsf{prof}(P)$.

Ahora, para cada $A \in \mathcal{B}$ y $at \in A$ tomamos

$$Q_{at}^{A} = \bigsqcap\{P_{at}^{A'} \mid A'\mathcal{B},\ at \in A'\ A' \upharpoonright t \subseteq A\}$$

aplicando el axioma [**PC1**] tenemos

$$P'' =_{\mathrm{E}} \prod_{A\in\mathcal{B}} \coprod_{at\in A} Q^A_{at}$$

Entonces para cada $A \in \mathcal{B}$ y $at \in A$ tenemos que $\mathsf{prof}(Q^A_{at}) < \mathsf{prof}(P)$, de modo que aplicando la hipótesis de inducción tenemos que existe un proceso en forma normal $Q1^A_{at}$ tal que

$$Q1^A_{at} =_{\mathrm{E}} Q^A_{at}$$

Finalmente, tomando

$$Q = \prod_{A\in\mathcal{B}} \coprod_{at\in A} Q1^A_{at}$$

obtenemos $P''{=}_{\mathrm{E}}Q$ y por tanto $P{=}_{\mathrm{E}}Q$. Además, puesto que en virtud de la hipótesis de inducción tenemos que $\mathsf{prof}(Q1^A_{at}) \leq \mathsf{prof}(Q^A_{at})$ para cada $A \in \mathcal{B}$, tendremos que $\mathsf{prof}(Q) \leq \mathsf{prof}(P)$.

□

Proposición 5.2.7 Sean P y Q procesos en forma prenormal

$$P = \prod_{A\in\mathcal{A}} \left(\coprod_{at\in A} P^A_{at}\right) \quad \text{y} \quad Q = \prod_{A\in\mathcal{B}} \left(\coprod_{at\in A} Q^A_{at}\right)$$

tales que $\mathcal{B}_{\mathrm{con}}[\![P]\!] \approx \mathcal{B}_{\mathrm{con}}[\![Q]\!]$. Entonces $\mathsf{pc}(\mathcal{A}) = \mathsf{pc}(\mathcal{B})$ En particular, si P y Q están en forma normal, tendremos que $\mathcal{A} = \mathcal{B}$.

Demostración. Para demostrar la igualdad de los dos conjuntos de estados probaremos la doble inclusión entre ellos. Debido a la simetría de la proposición basta probar cualquiera de ellas, pues la otra sería totalmente simétrica. Consideramos entonces $A \in \mathsf{pc}(\mathcal{A})$. Para ver que $A \in \mathsf{pc}(\mathcal{B})$ veremos que dicho conjunto cumple las condiciones de la definición 5.1.2 con respecto a $\mathcal{B}$. A fin de simplificar la notación tomaremos

$$\begin{aligned} B_1 = \mathcal{B}_{\mathrm{con}}[\![P]\!] &= \{A \mid A \in \mathcal{A}\}\cup \\ &\quad \{(A \upharpoonright t)at\cdot b \mid A \in \mathcal{A},\ at \in A,\ b \in \mathcal{B}_{\mathrm{con}}[\![P^A_{at}]\!]\} \\ B_2 = \mathcal{B}_{\mathrm{con}}[\![Q]\!] &= \{A \mid A \in \mathcal{B}\}\cup \\ &\quad \{(A \upharpoonright t)at\cdot b \mid A \in \mathcal{B},\ at \in B,\ b \in \mathcal{B}_{\mathrm{con}}[\![Q^A_{at}]\!]\} \end{aligned}$$

- Puesto que $A \in \mathsf{pc}(\mathcal{A})$ existirá $A_1 \in \mathcal{A}$ de modo que $A_1 \ll A$. Por otra parte, como $Q \ll P$, existirá $A_2 \in \mathcal{B}$ tal que $A_1 \ll A_2$, y por tanto $A_2 \ll A$.

  Supongamos ahora que existe $A_2 \in \mathcal{B}$ tal que $A_2 \ll A$ y $\mathsf{nd}(A_2) < \mathsf{nd}(A)$. Puesto que $\mathcal{B}_{\mathrm{con}}[\![P]\!] \ll \mathcal{B}_{\mathrm{con}}[\![Q]\!]$ tendremos algún estado $A_1 \in \mathcal{A}$ de modo que $A_1 \ll A_2$, y por

expertos

tanto $A_1 \ll A$. Además, puesto que $\mathsf{nd}(A_1) \leq \mathsf{nd}(A_2)$ tenemos que $\mathsf{nd}(A_1) < \mathsf{nd}(A)$, lo cual está en contradicción con el el hecho de que $A \in \mathsf{pc}(\mathcal{A})$.

Hemos visto por tanto que existe $A_2 \in \mathcal{B}$ tal que $A_2 \ll A$, y además no existe $A_2' \in \mathcal{B}$ tal que $A_2' \ll A$ y $\mathsf{nd}(A_2) < A$.

- Tomemos ahora $at \in A$. En primer lugar observamos que $t < \mathsf{nd}(A)$. Por otro lado, por la definición de conjunto pseudo-convexo, tenemos que existe $A_1 \in \mathcal{A}$ tal que $at \in A_1$ y $A_1 \upharpoonright t \subseteq A$. Puesto que $\mathcal{B}_{\text{con}}[\![P_{at}^{A_1}]\!]$ es un conjunto consiste de barbas existirá al menos un estado $A_1'' \in \mathcal{B}_{\text{con}}[\![P_{at}^{A_1}]\!]$, tomando entonces $A_1' = A_1 \upharpoonright t$ se verifica

$$b_1 = A_1' at A_1'' \in B_1$$

Por otra parte $\mathcal{B}_{\text{con}}[\![Q]\!] \ll \mathcal{B}_{\text{con}}[\![P]\!]$, entonces existe $b_2 \in \mathcal{B}_{\text{con}}[\![Q]\!]$ tal que $b_2 \ll b_1$. Ahora hemos de distinguir dos posibilidades en función de la barba b_2:

$\mathbf{b_2 = A_2'}$ con $A_2' \ll A_1'$ y $\mathsf{nd}(A_2') \leq t$. En tal caso, puesto que $\mathcal{B}_{\text{con}}[\![P]\!] \ll \mathcal{B}_{\text{con}}[\![A]\!]$ existirá $A_1''' \in \mathcal{A}$ de tal que $A_1''' \ll A_2'$. Por tanto tenemos $A_1''' \ll A$ y $\mathsf{nd}(A_1''') \leq t < \mathsf{nd}(A)$, lo que está en contradicción con el hecho de que $A \in \mathsf{pc}(\mathcal{A})$, por lo que este caso queda excluido.

$\mathbf{b_2 = A_2' at A_2''}$ con $A_2' \subseteq A_1'$ y $A_2'' \ll A_1''$. Entonces existirá un estado $A_2 \in \mathcal{B}$ tal que $A_2 \upharpoonright t = A_2'$ y $at \in A_2$, con lo cual se cumpliría la segunda condición de la definición 5.1.2 para que $A \in \mathsf{pc}(\mathcal{B})$.

□

Proposición 5.2.8 Para cualesquiera P y Q, procesos en forma normal se tiene

$$\mathcal{B}_{\text{con}}[\![P]\!] \approx \mathcal{B}_{\text{con}}[\![Q]\!] \quad \Rightarrow \quad P =_{\text{E}} Q$$

Demostración. Haremos la misma por inducción sobre la profundidad de P:

$\mathsf{prof}(\mathbf{P}) = \mathbf{1}$ En tal caso tenemos que $P = \mathtt{DIV}(t)$, y por tanto

$$\mathcal{B}_{\text{con}}[\![P]\!] = \begin{cases} \{\Omega t\} & \text{si } t < \infty \\ \varnothing & \text{si } t = \infty \end{cases}$$

y como quiera que Q está en forma normal tenemos que $Q = \mathtt{DIV}(t)$.

$\mathsf{prof}(\mathbf{P}) > \mathbf{1}$ Tenemos entonces que

$$P = \bigsqcap_{A \in \mathcal{A}} \Box_{at \in A} P_{at}^{A}$$

por tanto

$$\begin{aligned}\mathcal{B}_{\text{con}}[\![P]\!] = & \ \{A \mid A \in \mathcal{A}\} \cup \\ & \cup \{(A \upharpoonright t)at \cdot b \mid A \in \mathcal{A},\ at \in A,\ b \in \mathcal{B}_{\text{con}}[\![P^A_{at}]\!]\}\end{aligned}$$

Puesto que Q está en forma normal ha de ser de la forma:

$$Q = \prod_{B \in \mathcal{B}} \bigsqcup_{bt \in B} Q^B_{bt}$$

y por tanto

$$\begin{aligned}\mathcal{B}_{\text{con}}[\![Q]\!] = & \ \{B \mid A \in \mathcal{B}\} \cup \\ & \cup \{(A \upharpoonright t)bt \cdot b \mid A \in \mathcal{B},\ at \in B,\ b \in \mathcal{B}_{\text{con}}[\![Q^A_{at}]\!]\}\end{aligned}$$

Aplicando la proposición anterior tenemos $\mathcal{A} = \mathcal{B}$. Ahora, para cada $A \in \mathcal{A}$ y para cada $at \in A$ consideramos los procesos

$$P1^A_{at} = \prod \{P^{A'}_{at} \mid A' \in \mathcal{A},\ at \in A' \text{ y } A' \upharpoonright t \subseteq A\}$$

$$Q1^A_{at} = \prod \{Q^{A'}_{at} \mid A' \in \mathcal{A},\ at \in A' \text{ y } A' \upharpoonright t \subseteq A\}$$

Por el axioma **[PC1]** tenemos que

$$P =_{\text{E}} \prod_{A \in \mathcal{A}} \bigsqcup_{at \in A} P1^A_{at} \quad \text{y} \quad Q =_{\text{E}} \prod_{A \in \mathcal{A}} \bigsqcup_{at \in A} Q1^A_{at}$$

Demostremos ahora que $P1^A_{at} \approx Q1^A_{at}$, para ello tomemos $b \in P1^A_{at}$; tenemos entonces que

$$A \upharpoonright t \cdot b \in \mathcal{B}_{\text{con}}[\![P]\!]$$

Por otra parte, puesto que $Q \approx P$ existe $b' \in \mathcal{B}_{\text{con}}[\![Q]\!]$ tal que

$$b' \ll A \upharpoonright t \cdot b$$

Si b' fuera un estado A', tendríamos que $\mathsf{nd}(A') < t$ y $TAct(A') \subseteq A$, con lo que llegaríamos a

$$\mathsf{nd}(A') < \mathsf{nd}(A) \quad \text{y} \quad A' \ll A$$

de modo que $\mathcal{A}$ no sería pseudo-cerrado. Hemos de tener entonces

$$b' = A'at \cdot b'' \quad \text{con} \quad A' \subseteq A \quad \text{y} \quad b'' \ll b$$

Pero en tal caso existirá un estado $A'' \in \mathcal{A}$ tal que

$$at \in A'', \quad A' = A'' \upharpoonright t \quad \text{y} \quad b'' \in \mathcal{B}_{\text{con}}[\![Q1^{A''}_{at}]\!]$$

han

Puesto que $A'' \upharpoonright t = A' \subseteq A$ tendremos que $b'' \in \mathcal{B}_{\mathsf{con}}[\![Q1^A_{at}]\!]$ con lo cual queda demostrado $Q1^A_{at} \ll P1^A_{at}$. Haciendo un razonamiento simétrico obtendríamos $P1^A_{at} \ll Q1^A_{at}$.

Como quiera $P1^A_{at}$ y $Q1^A_{at}$ están en forma prenormal, existirán procesos $P2^A_{at}$ y $Q2^A_{at}$ en forma normal tales que

$$P1^A_{at} =_{\mathrm{E}} P2^A_{at} \quad \text{y} \quad Q1^A_{at} =_{\mathrm{E}} Q2^A_{at}$$

Puesto que las reglas del sistema son correctas tendremos

$$P2^A_{at} \approx Q2^A_{at}$$

y como $\mathsf{prof}(P1^A_{at}) \geq \mathsf{prof}(P2^A_{at})$, por hipótesis de inducción obtenemos

$$P2^A_{at} =_{\mathrm{E}} Q2^A_{at}$$

de modo que

$$\begin{aligned} P \;&=_{\mathrm{E}}\; \bigsqcap_{A \in \mathcal{A}} \mathop{\Box}_{at \in A} at\,;P1^A_{at} \;=_{\mathrm{E}}\; \bigsqcap_{A \in \mathcal{A}} \mathop{\Box}_{at \in A} at\,;P2^A_{at} \;=_{\mathrm{E}} \\ &=_{\mathrm{E}}\; \bigsqcap_{A \in \mathcal{A}} \mathop{\Box}_{at \in A} at\,;Q2^A_{at} \;=_{\mathrm{E}}\; \bigsqcap_{A \in \mathcal{A}} \mathop{\Box}_{at \in A} at\,;Q1^A_{at} \;=_{\mathrm{E}}\; Q \end{aligned}$$

$\square$

Finalmente, como corolario de los anteriores tenemos la completitud del sistema de ecuaciones:

Teorema 5.2.9 Para cualesquiera procesos finitos $P, Q \in \mathsf{FRec}(\Sigma_{\mathsf{seq}})$, se tiene

$$\mathcal{B}_{\mathsf{con}}[\![P]\!] \approx \mathcal{B}_{\mathsf{con}}[\![Q]\!] \quad \Rightarrow \quad P =_{\mathrm{E}} Q$$

Demostración. Aplicando la proposición 5.2.5 tenemos que existen procesos P_1 y Q_1 en forma prenormal tales que

$$P_1 =_{\mathrm{E}} P \quad \text{y} \quad Q_1 =_{\mathrm{E}} Q$$

Por la proposición 5.2.6 existirán entonces procesos P_2 y Q_2 en forma normal, tales que

$$P_1 =_{\mathrm{E}} P_2 \quad \text{y} \quad Q_1 =_{\mathrm{E}} Q_2$$

Puesto que las del sistema ecuaciones son correctas, tenemos

$$\begin{aligned} \mathcal{B}_{\mathsf{con}}[\![P_2]\!] \;&\approx\; \mathcal{B}_{\mathsf{con}}[\![P_1]\!] \;\approx\; \mathcal{B}_{\mathsf{con}}[\![P]\!] \;\approx \\ &\approx\; \mathcal{B}_{\mathsf{con}}[\![Q]\!] \;\approx\; \mathcal{B}_{\mathsf{con}}[\![Q_1]\!] \;\approx\; \mathcal{B}_{\mathsf{con}}[\![Q_2]\!] \end{aligned}$$

De modo que aplicando la proposición 5.2.8 obtenemos $P_2 =_{\mathrm{E}} Q_2$, y por tanto $P =_{\mathrm{E}} Q$.

$\square$

5.3 Procesos Recursivos

Para poder deducir propiedades de los procesos recursivos hemos de poder manejar axiomáticamente la relación de orden entre procesos. Surgen así las inecuaciones, algunas de las cuales ya han aparecido en la tabla 5.1. La primera inecuación que consideramos es la regla más compleja que hemos visto hasta el momento en el sistema. Para introducirla consideremos dos procesos P yQ en forma **prenormal**:

$$P = \prod_{A \in \mathcal{A}} \coprod_{at \in A} P^A_{at} \quad \text{y} \quad Q = \prod_{A \in \mathcal{B}} \coprod_{at \in A} Q^A_{at}$$

Además, para cada $B \in \mathcal{B}$ y $at \in B$ consideremos los conjuntos de estados:

$$\begin{array}{rcl} \mathsf{Per}(\mathcal{A}, A, at) & = & \{A' \mid A' \in \mathcal{A},\ at \in A',\ A' \upharpoonright t \subseteq A\} \\ \mathsf{BM}(\mathcal{A}, A, at) & = & \{A' \mid A' \in \mathcal{A},\ \mathsf{nd}(A') \leq t,\ A' \ll A\} \end{array}$$

Se tiene

$$[\mathbf{ME}] \quad \frac{\forall A \in \mathcal{B} \left(\begin{array}{l} (\exists A' \in \mathcal{A} :\ A' \ll A) \wedge \\ \wedge\ \forall at \in A \left(\begin{array}{l} \mathsf{BM}(\mathcal{A}, A, at) \neq \varnothing \quad \vee \\ \vee \quad \mathsf{Per}(\mathcal{A}, A, at) \neq \varnothing \ \wedge \displaystyle\prod_{A' \in \mathsf{Per}(\mathcal{A}, A, at)} P^{A'}_{bt} <_{\mathrm{E}} Q^A_{bt} \end{array} \right) \end{array} \right)}{P <_{\mathrm{E}} Q}$$

Dado lo complejo de la regla anterior vamos a intentar explicarla. Mediante esta regla pretendemos caracterizar la relación $\ll$ entre procesos que están en forma prenormal; es decir, la situación en la que para cada barba b de Q existe una barba b' de P *peor*. Según sea dicha barba tenemos dos posibilidades:

- El primer caso es que la barba b sea estado $A \in \mathcal{B}$. Entonces ha de existir un estado A' de P tal que $A' \ll A$. O sea

  $$\forall A \in \mathcal{B}\ \exists A' \in \mathcal{A} :\ A' \ll A$$

  que es precisamente la condición de la premisa de la regla [**ME**].

- El otro caso es que la barba b sea de la forma $A_1 at \cdot b_1$. Entonces existirá un estado $A \in \mathcal{B}$ tal que $at \in A$, $A_1 = A \upharpoonright t$ siendo b_1 es una barba del proceso Q^A_{at}. Para que exista una barba menor en P hay dos posibilidades:

  - Se trate de un estado $A' \in \mathcal{A}$ verificando $\mathsf{nd}(A') \leq t$ y $TAct(A') \subseteq A_1$, y por tanto $A' \ll A$. Se verifica entonces

    $$\mathsf{BM}(\mathcal{A}, A, at) \neq \varnothing$$

---

hecho

– La misma sea de la forma $A'_1 bt \cdot b'_1$ con $A'_1 \subseteq A_1$ y $b'_1 \ll b_1$. Entonces existirá un estado P, $A' \in \mathcal{A}$, tal que $A'_1 = A' \upharpoonright t$, $at \in A'$ siendo b'_1 es una barba de $P^{A'}_{at}$. Verificándose es este caso

$$\mathsf{Per}(\mathcal{A}, A, at) \neq \varnothing \;\wedge \prod_{A' \in \mathsf{Per}(\mathcal{A},A,at)} P^{A'}_{bt} <_{\mathrm{E}} Q^{A}_{bt}$$

Puesto que todo lo anterior es preciso para cada barba de la forma $A_1 at \cdot b'$, queda así justificada da como resultado la segunda condición de la premisa:

$$\forall A \in \mathcal{B}\; \forall at \in A \left(\begin{array}{l} \mathsf{BM}(\mathcal{A}, A, at) \neq \varnothing \;\; \vee \\ \vee \;\; \mathsf{Per}(\mathcal{A}, B, at) \neq \varnothing \;\wedge \displaystyle\prod_{A' \in \mathsf{Per}(\mathcal{A},A,at)} P^{A'}_{bt} <_{\mathrm{E}} Q^{A}_{bt} \end{array} \right)$$

[ME1]	$\mathtt{DIV}(t') < \mathtt{DIV}(t)$	$t' \leq t$
[ME2]	$\mathtt{DIV} < P$	
[ME3]	$P \sqcap Q < Q$	

Tabla 5.5: Inecuaciones derivadas

[REC]	$\mathtt{REC}\, x.P =_{\mathrm{E}} P[\mathtt{REC}\, x.P/x]$
[APF]	$\dfrac{\forall k \in \mathbb{N}\ \mathsf{ap}(P,k) < Q}{P < Q}$

Tabla 5.6: Axiomas para la recursión

cuando se verifican ambas condiciones podemos concluir en efecto $P \ll Q$, con lo que queda justificada la corrección de la regla **[ME]**.

A partir de dicha regla anterior pueden derivarse algunas otras que serán de utilidad más específica como las que se presentan en la tabla 5.5. En lo que se refiere a la recursión tenemos inicialmente las habituales que se presentan en la tabla 5.6. En las álgebras de procesos si tiempo habituales, tras incorporara estas reglas adicionales se podría demostrar la completitud del sistema de ecuaciones. Pero en nuestro caso ello no es así, como prueba el siguiente

Ejemplo 5.3.1 Consideremos los procesos $P = \mathtt{STOP}$ y $Q = \mathtt{REC}\, x.\tau 1 \,;\, x$. Claramente se verifica que

$$\mathcal{B}_{\mathrm{con}}[\![P]\!] = \Big\{\{\varnothing\}\Big\} = \mathcal{B}_{\mathrm{con}}[\![Q]\!]$$

Por otro lado tenemos $\mathcal{B}_{\text{con}}[\![\mathsf{ap}(Q,k)]\!] = \Big\{\{\Omega k\}\Big\}$. Entonces aplicando el axioma [**ME1**] obtenemos que $\mathsf{ap}(Q,k) <_{\mathrm{E}} P$ (recordemos que $\mathtt{STOP} = \mathtt{DIV}(\infty)$), con lo cual habríamos deducido que $Q <_{\mathrm{E}} P$. Pero desgraciadamente no tenemos ninguna regla que nos permita deducir que $P <_{\mathrm{E}} Q$, de modo que no se puede deducir tampoco $P =_{\mathrm{E}} Q$.

Para poder derivar dicha equivalencia, necesitaríamos alguna regla que nos indicase que si dos procesos se comportan igual por periodos de tiempo finitos tan largos como se desee, entonces son equivalentes. Esta propiedad se puede interpretar como una cierta continuidad en el tiempo y corrección va a quedar justificada por el hecho de que los estados de un proceso son temporalmente compactos.

□

En consecuencia para poder enunciar la regla requerida necesitamos un operador auxiliar $\mathtt{RED}(P,t)$ con $t \in \mathcal{T}$, cuyo efecto va a ser *limitar hasta el instante (finito) t la la vida* del proceso P. A dicho operador lo denominaremos *reductor*. Como quiera que su manejo tiene lugar al nivel sintáctico, hemos de añadir a la signatura este nuevo operador, obteniendo

$$\Sigma_{\mathsf{seq+}} = \Sigma_{\mathsf{seq}} \cup \{\mathtt{RED}(\cdot,t) \mid t \in \mathcal{T}\}$$

Por su parte la semántica denotacional del operador viene dada por

$$\mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](B) = \{\mathtt{RED}(b,t) \mid b \in B\}$$

donde la función sobre barbas $\mathtt{RED}(b,t)$ viene definida como sigue:

$$\mathtt{RED}(b,t) = \begin{cases} A & \text{si } b = A \ \text{ y } \mathsf{nd}(A) \le t \\ (A \upharpoonright t) \cup \{\Omega t\} & \text{si } b = A \ \text{ y } \mathsf{nd}(A) > t \\ A_1 a_1 t_1 \cdot \mathtt{RED}(b_1, t - t_1) & \text{si } b = A_1 a_1 t_1 \cdot b_1 \ \text{ y } \ t_1 < t \\ (A_1 \upharpoonright t) \cup \{\Omega t\} & \text{si } b = A_1 a_1 t_1 \cdot b_1 \ \text{ y } \ t_1 \ge t \end{cases}$$

Como podemos observar en su definición, el proceso $\mathtt{RED}(P,t)$ se comporta como el proceso P hasta que llega el instante t, a partir de entonces el proceso entra en divergencia, con lo que ha quedado reducida su vida útil.

La semántica operacional de este nuevo operador podría definirse mediante las reglas de la tabla 5.7. La definición de las funciones auxiliares necesarias para completar la semántica operacional la incluimos en la misma tabla.

Veamos a continuación una serie de propiedades interesantes del operador.

Lema 5.3.2 Siendo b una barba, se verifican:

- Si $t > \mathsf{nd}(b)$, entonces $b = \mathtt{RED}(b,t)$.

cálculos.

$$[\textbf{RED1}]\quad \frac{P \xrightarrow{et'} P',\ t' < t,\ t > 0}{\mathtt{RED}(P,t) \xrightarrow{et'} \mathtt{RED}(P', t-t')}\ e \in \mathcal{E} \qquad [\textbf{RED2}]\quad \frac{P \rightarrowtail P',\ t > 0}{\mathtt{RED}(P,t) \rightarrowtail \mathtt{RED}(P',t)}$$

$$[\textbf{RED3}]\quad \mathtt{RED}(P,0) \rightarrowtail \mathtt{DIV}$$

$$\mathsf{stb}(\mathtt{RED}(P,t)) = \begin{cases} \mathsf{stb}(P) & \text{si } t > 0 \\ \mathsf{false} & \text{si } t = 0 \end{cases}$$

$$\begin{aligned} \mathsf{Tiem}(\mathtt{RED}(P,t),a) &= \mathsf{Tiem}(P,a) \uparrow t \\ \mathsf{idle}(\mathtt{RED}(P,t),A) &= \min(\mathsf{idle}(P,A),t) \end{aligned}$$

$$\mathsf{Upd}(\mathtt{RED}(P,t),t') = \begin{cases} \mathtt{RED}(\mathsf{Upd}(P,t'),t-t') & \text{si } t \leq t \\ \mathtt{DIV} & \text{si } t' > t \end{cases}$$

$$P \downarrow\ \wedge\ t > 0 \quad \Rightarrow \quad \mathtt{RED}(P,t) \downarrow$$

Tabla 5.7: Semántica Operacional de $\mathtt{RED}(P,t)$

- Si $\mathsf{nd}(\mathtt{RED}(b,t)) < t$ entonces $b = \mathtt{RED}(b,t)$.
- Si $\mathsf{nd}(b) < t$ entonces $\quad b \in B \iff b \in \mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](B)$.

Demostración. Las dos primeras partes se prueba trivialmente mediante inducción sobre la longitud de la barba. La tercera es consecuencia inmediata de las anteriores.

□

Hemos de demostrar ahora que la semántica denotacional del nuevo operador está bien definida. En primer lugar resulta fácil comprobar que si B es un conjunto consistente de barbas y $t \in \mathcal{T}$, entonces $\mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](B)$ es un conjunto consistente de barbas. Además hemos de demostrar que la función semántica asociada es monótona con respecto a las relaciones $\ll$ y $\prec$ así como su continuidad con respecto a la segunda. Todo ello queda incluido en la siguiente

Proposición 5.3.3

- Para cada barba b y cada $t \in \mathcal{T}$, se tiene $\mathtt{RED}(b,t) \prec b$.

- Si b y b' son barbas tales que $b \prec b'$ entonces $\mathtt{RED}(b,t) \prec \mathtt{RED}(b',t)$.

- Si b y b' son barbas tales que $b \ll b$ entonces $\mathtt{RED}(b,t) \ll \mathtt{RED}(b,t)$.

- Si B y B' son conjuntos de barbas tales que $B \prec B'$ entonces

$$\mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](B) \prec \mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](B')$$

- Si B y B' son conjuntos de barbas tales que $B \ll B'$ entonces

$$\mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](B) \ll \mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](B')$$

- Si $B_1 \prec B_2 \prec \cdots$ es una secuencia no decreciente de conjuntos consistentes de barbas, se tiene

$$\mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](\mathsf{lub}\{B_i \mid i \in \mathbb{N}\}) = \mathsf{lub}\{\mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](B_i) \mid i \in \mathbb{N}\}$$

Demostración. La demostración de las tres primeras partes es sencilla por inducción sobre la longitud de b y b'. Las dos siguientes son consecuencia inmediata de las anteriores. Para demostrar la última parte consideramos los conjuntos

$$B = \mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](\mathsf{lub}\{B_i \mid i \in \mathbb{N}\}) \quad \text{y} \quad B' = \mathsf{lub}\{\mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](B_i) \mid i \in \mathbb{N}\}$$

- Sea en primer lugar $b \in B'$; puesto que todo $b' \in \mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](B_i)$ verifica $\mathsf{nd}(b') \leq t$, tenemos $\mathsf{nd}(b) \leq t < \infty$. Entonces para todo $k \in \mathbb{N}$ existe $l \geq k$ de modo que $b \in \mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](B_l)$. Si $\mathsf{nd}(b) < t$, aplicando el lema anterior deducimos que $b \in B_l$, y entonces $b \in \mathsf{lub}\{B_i \mid i \in \mathbb{N}\}$, con lo que $b \in B$.

  Si $\mathsf{nd}(b) = t$ existirá $b_l \in B_l$ tal que $b = \mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](b_l)$, consideramos en particular el 'l' correspondiente a $k = 1$. Siendo $b_l = bs_l \cdot A_l$, como $B_i \prec B_{i+1}$, podemos encontrar una secuencia de estados

$$A_l \prec A_{l+1} \cdots A_i \prec A_{i+1} \cdots$$

  de modo que $bs_l \cdot A_i \in B_i$. Tomando entonces el estado $A = \mathsf{lub}\{A_i \mid i \geq l\}$ tenemos

$$bs_l \cdot A \in \mathsf{lub}\{B_i \mid i \in \mathbb{N}\}$$

  Finalmente utilizamos el hecho de que tener en cuenta que $\mathtt{RED}(bs_l \cdot A, t) = b$, de donde deducimos $b \in B$.

Se

- Sea ahora $b \in B$; existe entonces $b' \in \mathsf{lub}\{B_i \mid i \in \mathbb{N}\}$ tal que $\mathtt{RED}(b', t) = b$. Si $\mathsf{nd}(b') < \infty$ tenemos que para todo $k \in \mathbb{N}$ existe $l \geq k$ tal que $b' \in B_l$, por tanto

$$b = \mathtt{RED}(b', t) \in \mathcal{B}_{\mathrm{con}}[\![\mathtt{RED}(t)]\!](B_l)$$

de modo que $b \in B'$. Si por contra $\mathsf{nd}(b') = \infty$ y siendo $b = bs \cdot A$, tenemos que para cada $s \in \mathcal{T}$ existen $l_s \in \mathbb{N}$ y un un estado A_s de modo que $bs \cdot A_s \in B_{l_s}$ y $A \upharpoonright s = A_s \upharpoonright s$. Tomemos $s_0 \geq t - \mathsf{t}(bs)$[‡], y se verificará

$$\mathtt{RED}(bs \cdot A_{s_0}, t) = \mathtt{RED}(b', t) = b$$

Como quiera que $B_i \prec B_{i+1}$ podremos encontrar una secuencia de estados

$$A_0 \prec A_1 \prec \cdots A_i \prec A_{i+1} \prec \cdots$$

tal que $bs \cdot A_i \in B_{l_{s_0}+i}$. Tenemos entonces que $b = \mathtt{RED}(bs \cdot A_i, t) \in \mathcal{B}_{\mathrm{con}}[\![\mathtt{RED}(t)]\!](B_i)$ y por tanto $b \in B' = \mathsf{lub}\{\mathcal{B}_{\mathrm{con}}[\![\mathtt{RED}(t)]\!](B_i) \mid i \in \mathbb{N}\}$.

□

[RED1]	$\mathtt{RED}(et\,; P, t') =_{\mathrm{E}} (et\,; \mathtt{RED}(P, t'-t)) \,\Box\, \mathtt{DIV}(t)$	$t' > t$ y $e \in \mathcal{E}$
[RED2]	$\mathtt{RED}(et\,; P, t') =_{\mathrm{E}} \mathtt{DIV}(t')$	$t' \leq t$ y $e \in \mathcal{E}$
[RED3]	$\mathtt{RED}(P \,\Box\, Q, t) =_{\mathrm{E}} \mathtt{RED}(P, t) \,\Box\, \mathtt{RED}(Q, t)$	
[RED4]	$\mathtt{RED}(P \sqcap Q, t) =_{\mathrm{E}} \mathtt{RED}(P, t) \sqcap \mathtt{RED}(Q, t)$	
[RED5]	$\mathtt{RED}(\mathtt{DIV}(t), t') =_{\mathrm{E}} \mathtt{DIV}(t'')$	donde $t'' = \min(t, t')$
[RED6]	$\mathtt{RED}(\mathtt{REC}\, x.P, t) = \mathtt{RED}(P[\mathtt{REC}\, x.P/x], t)$	

Tabla 5.8: Ecuaciones para $\mathtt{RED}(P, t)$

Tal y como hemos definido el operador, es fácil comprobar que las ecuaciones de la tabla 5.8 son correctas. Estos axiomas nos permiten *eliminar* el operador $\mathtt{RED}(\cdot,\cdot)$ de cualquier proceso finito. De modo que sigue siendo cierto que para cualquier proceso finito P existe un proceso Q en forma normal tal que $P =_{\mathrm{E}} Q$.

Llega así el momento de incorporar la regla que precisamos para poder demostrar la completitud del sistema:

$$[\mathbf{APT}] \quad \frac{\forall t \in \mathcal{T}\ \mathtt{RED}(P, t) <_{\mathrm{E}} \mathtt{RED}(Q, t)}{P <_{\mathrm{E}} Q}$$

[‡]Recordemos que $\mathsf{t}(bs)$ es la suma de los tiempos que aparecen en las acciones de la b-traza bs (Definición 2.5.2).

Como ya hemos comentado anteriormente, la regla anterior nos indica que si dos procesos son iguales durante cualquier intervalo finito, entonces serán equivalentes. Su corrección está íntimamente relacionada con la propiedad de compacidad temporal.

Proposición 5.3.4 Supongamos que para todo $t \in \mathcal{T}$ se tiene

$$\mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](B_1) \ll \mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](B_2)$$

entonces $B_1 \ll B_2$.

Demostración. Tomemos $b = bs \cdot A \in B_2$; sabemos que

$$\mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](B_1) \ll \mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](B_2)$$

Entonces para cada $t \in \mathcal{T}$ existirá $b_t \in \mathcal{B}_{\text{con}}[\![\mathtt{RED}(t)]\!](B_1)$ tal que $b_t \ll \mathtt{RED}(b, t)$. Si para algún $t \in \mathcal{T}$ se verifica $\mathsf{nd}(b_t) < t$ entonces tendremos $b_t \in B_1$ y por tanto $b_t \ll b$.

Supongamos entonces que para cada $t \in \mathcal{T}$ se verifica $\mathsf{nd}(b_t) \geq t$. Entonces necesariamente tendremos $\mathsf{nd}(b) = \infty$. Puesto que $b = bs \cdot A$, esto implica que $\mathsf{nd}(A) = \infty$. Tomamos entonces $t \geq \mathsf{t}(bs)$, y necesariamente tendremos que $\mathsf{lon}(b_t) = \mathsf{lon}(b)$. De modo que para cada $t \geq \mathsf{t}(bs)$, b_t es de la forma

$$b_t = bs_t \cdot A_t \quad \text{con} \quad bs_t \ll bs,\ A_t \upharpoonright t \subseteq A \text{ y } \mathsf{nd}(A_t) \geq t - \mathsf{t}(bs)$$

Puesto que estamos considerando un alfabeto finito y un dominio de tiempo discreto sólo puede haber un número finito de b-trazas bs' tales que $bs' \ll bs$, por tanto entre las b-trazas bs_t sólo puede haber un número finito distintas entre sí, de modo que alguna se repetirá un número infinito de veces. En consecuencia podemos suponer que todas ellas son iguales. Tomemos entonces $bs' = bs_{\mathsf{t}(bs)}$, y puesto que el conjunto B_1 es consistente tenemos que el conjunto de estados

$$\mathcal{A}' = \mathcal{A}(B_1, bs')$$

será temporalmente compacto. Además, para cada $t \in \mathcal{T}$ se verifica

$$A_{t+\mathsf{t}(b)} \in \mathcal{A} \qquad A_{t+\mathsf{t}(b)} \upharpoonright t \subseteqq A$$

Puesto que $\mathcal{A}$ es temporalmente compacto, aplicando la proposición 2.4.8, tendremos que existe $A' \in \mathcal{A}$, o lo que es lo mismo $bs' \cdot A' \in B_1$, con $A' \subseteqq A$. Puesto que $\mathsf{nd}(A) = \infty$ entonces $A' \ll A$, y tomando $b' = bs' \cdot A'$ obtenemos

$$b' \in B_1 \quad \text{y} \quad b' \ll b$$

□

---

ahorrarían

Demostraremos ahora la completitud del nuevo sistema de ecuaciones (inecuaciones incluidas) para el caso de procesos finitos, es decir, que si P y Q son procesos finitos se cumple:

$$\mathcal{B}_{\text{con}}[\![P]\!] \ll \mathcal{B}_{\text{con}}[\![Q]\!] \;\Rightarrow\; P <_{\text{E}} Q$$

Puesto que cualquier proceso finito se puede pasar a forma prenormal, bastaría con demostrarlo para procesos en forma prenormal:

Proposición 5.3.5 Para cualesquiera procesos en forma prenormal P y Q, se verifica

$$\mathcal{B}_{\text{con}}[\![P]\!] \ll \mathcal{B}_{\text{con}}[\![Q]\!] \quad \Rightarrow \quad P <_{\text{E}} Q$$

Demostración. Siendo

$$P = \prod_{A \in \mathcal{A}} \bigsqcup_{at \in A} P^A_{at} \quad \text{y} \quad Q = \prod_{A \in \mathcal{B}} \bigsqcup_{at \in A} Q^A_{at}$$

haremos la demostración por inducción sobre la suma de las profundidades de ambos procesos $\mathsf{prof}(P) + \mathsf{prof}(Q)$. de Q.

$\mathsf{prof}(\mathbf{Q}) = \mathbf{1}$ **y** $\mathsf{prof}(\mathbf{P}) = \mathbf{1}$. En tal caso

$$Q = \prod_{i=1}^{n} \mathtt{DIV}(t_1) \quad \text{con} \quad t_i \in \mathcal{T} \cup \{\infty\}$$

tomando $t_Q = \min\{t_1, \ldots, t_n\}$, aplicando las ecuaciones [**PC2**] y [**ID2**], obtenemos $Q =_{\text{E}} \mathtt{DIV}(t_Q)$. Por otro lado

$$P = \prod_{i=1}^{m} \mathtt{DIV}(s_1) \quad \text{con} \quad s_i \in \mathcal{T} \cup \{\infty\}$$

como antes, tomando $s_P = \min\{s_1, \ldots, s_m\}$, tenemos $P =_{\text{E}} \mathtt{DIV}(s_P)$. Entonces $\{\Omega t_Q\} \in \mathcal{B}_{\text{con}}[\![Q]\!]$, y puesto que $\mathcal{B}_{\text{con}}[\![P]\!] \ll \mathcal{B}_{\text{con}}[\![Q]\!]$ existirá un estado $A \in \mathcal{B}_{\text{con}}[\![P]\!]$ verificando $A \ll \{\Omega t_Q\}$. Puesto que A es un estado de P, ha de existir $i \in \{1, \ldots n\}$ tal que $A = \{\Omega s_i\}$, y debido a la manera en la que hemos escogido s_P tendremos $s_P \leq s_i$. Como $s_i \leq t_Q$, aplicando la ecuación [**ME1**], tenemos

$$\mathtt{DIV}(s_P) <_{\text{E}} \mathtt{DIV}(t_Q)$$

con lo que obtenemos $P <_{\text{E}} Q$.

$\mathsf{prof}(\mathbf{Q}) > \mathbf{1}$ **ó** $\mathsf{prof}(\mathbf{P}) > \mathbf{1}$ Tenemos

$$\begin{array}{rcl}
\mathcal{B}_{\text{con}}[\![P]\!] & = & \{A \mid A \in \mathcal{A}\} \cup \\
& & \{(A \uparrow t)at \cdot b \mid A \in \mathcal{A},\ at \in A,\ b \in \mathcal{B}_{\text{con}}[\![P^A_{at}]\!]\} \\
\mathcal{B}_{\text{con}}[\![Q]\!] & = & \{A, \mid A \in \mathcal{B}\} \cup \\
& & \{(A \uparrow t)at \cdot b \mid A \in \mathcal{B},\ at \in B,\ b \in \mathcal{B}_{\text{con}}[\![Q^A_{at}]\!]\}
\end{array}$$

Tomemos $A \in \mathcal{B}$. Existe entonces $A \in \mathcal{A}$ tal que $A \ll B$ con lo que se cumple la primera condición de la premisa de la regla [**ME**]. Para ver que se cumple la segunda, tomemos $at \in A$ y supongamos que $\mathsf{BM}(\mathcal{A}, A, at) = \varnothing$. Hemos de demostrar

$$\mathsf{Per}(\mathcal{A}, A, at) \neq \varnothing \quad \text{y} \quad \bigsqcap_{A' \in \mathsf{Per}(\mathcal{A}, A, at)} P_{at}^{A'} <_{\mathrm{E}} Q_{at}^{A}$$

Siendo $b_1 \in \mathcal{B}_{\mathrm{con}}[\![Q_{at}^A]\!]$ tendremos

$$b = (A \upharpoonright t)at \cdot b_1 \in \mathcal{B}_{\mathrm{con}}[\![Q]\!]$$

Entonces como que $\mathcal{B}_{\mathrm{con}}[\![P]\!] \ll \mathcal{B}_{\mathrm{con}}[\![Q]\!]$ existirá una barba $b' \in \mathcal{B}_{\mathrm{con}}[\![P]\!]$ tal que $b' \ll b$. Puesto que $\mathsf{BM}(\mathcal{A}, A, at) = \varnothing$ tenemos que b' ha de ser de la forma

$$b' = A_1 at \cdot b_1'$$

Existe entonces $A' \in \mathcal{A}$ con $at \in A$, $A_1 = A' \upharpoonright t$ y $A_1 \subseteq A \upharpoonright t$, de donde deducimos $\mathsf{Per}(\mathcal{A}, A, at) \neq \varnothing$. Podemos por tanto considerar el proceso

$$P' = \bigsqcap_{A' \in \mathsf{Per}(\mathcal{A}, A, at)} P_{at}^{A'}$$

Para poder deducir $P <_{\mathrm{E}} Q$ bastaría demostrar $P' <_{\mathrm{E}} Q_{at}^A$. Tenemos que tanto P' como Q_{at}^A están en forma prenormal, Q_{at}^A tiene profundidad menor que Q, y P' tiene menor profundidad que P. Por tanto si demostramos que

$$\mathcal{B}_{\mathrm{con}}[\![P']\!] \ll \mathcal{B}_{\mathrm{con}}[\![Q_{at}^A]\!]$$

por hipótesis de inducción obtendríamos el resultado. Para demostrarlo, observemos que si $b_1 \in \mathcal{B}_{\mathrm{con}}[\![Q_{at}^A]\!]$ entonces $b = (A \upharpoonright t)at \cdot b_1 \in \mathcal{B}_{\mathrm{con}}[\![Q]\!]$. Existirá entonces una barba $b' \in \mathcal{B}_{\mathrm{con}}[\![P]\!]$ tal que $b' \ll b$. Por la misma razón que antes, tenemos que b' ha de ser de la forma

$$b' = A_1' at \cdot b_1' \quad \text{con} \quad A_1' \subseteq A \upharpoonright t \;\; \text{y} \;\; b_1' \ll b_1$$

por lo tanto existe $A' \in \mathcal{A}$ tal que $A_1' = A' \upharpoonright t$ y $b_1' \in \mathcal{B}_{\mathrm{con}}[\![P_{at}^{A'}]\!]$. Puesto que $A_1' \subseteq A' \upharpoonright t$ tenemos que $A \in \mathsf{Per}(\mathcal{A}, A, at)$ con lo que

$$\mathcal{B}_{\mathrm{con}}[\![P_{at}^{A'}]\!] \subseteq \mathcal{B}_{\mathrm{con}}[\![P']\!]$$

y por tanto $b_1' \in \mathcal{B}_{\mathrm{con}}[\![P']\!]$.

□

Como corolario de la proposición anterior obtenemos la completitud del nuevo sistema de ecuaciones con respecto a la relación $\ll$ para procesos finitos.

Teorema 5.3.6 Si P y Q son procesos finitos tales que $\mathcal{B}_{\mathsf{con}}[\![P]\!] \ll \mathcal{B}_{\mathsf{con}}[\![Q]\!]$ entonces $P <_{\mathrm{E}} Q$.

Demostración. Puesto que P y Q son procesos finitos tenemos que existen P_1 y P_2 en forma prenormal con

$$P_1 =_{\mathrm{E}} P \quad \text{y} \quad Q_1 =_{\mathrm{E}} Q$$

Puesto que las reglas y axiomas del sistema son correctos tenemos que

$$\mathcal{B}_{\mathsf{con}}[\![P_1]\!] \approx \mathcal{B}_{\mathsf{con}}[\![P]\!] \quad \text{y} \quad \mathcal{B}_{\mathsf{con}}[\![Q_1]\!] \approx \mathcal{B}_{\mathsf{con}}[\![Q]\!]$$

y por tanto $\mathcal{B}_{\mathsf{con}}[\![P_1]\!] \ll \mathcal{B}_{\mathsf{con}}[\![Q_1]\!]$. Aplicando la proposición anterior obtenemos $P_1 <_{\mathrm{E}} Q_1$, y en definitiva $P =_{\mathrm{E}} P_1 <_{\mathrm{E}} Q_1 =_{\mathrm{E}} Q$.

□

Abordamos ahora la prueba de la completitud del sistema para procesos cualesquiera. Para ello precisamos todavía de una serie de resultados previos.

Proposición 5.3.7 Siendo $P \in \mathsf{FRec}(\Sigma_{\mathsf{seq}})$ proceso finito, y $Q \in \mathsf{CRec}(\Sigma_{\mathsf{seq}})$ un proceso cualquiera, eventualmente recursivo, verificando $\mathcal{B}_{\mathsf{con}}[\![\mathtt{RED}(P,t)]\!] \ll \mathcal{B}_{\mathsf{con}}[\![\mathtt{RED}(Q,t)]\!]$, entonces existe $k \in \mathbb{N}$ de modo que $\mathcal{B}_{\mathsf{con}}[\![\mathtt{RED}(P,t)]\!] \ll \mathcal{B}_{\mathsf{con}}[\![\mathsf{ap}(\mathtt{RED}(Q,t),k)]\!]$.

Demostración. Razonaremos por reducción al absurdo y suponemos que para todo $k \in \mathbb{N}$ se tiene

$$\mathcal{B}_{\mathsf{con}}[\![\mathtt{RED}(P,t)]\!] \not\ll \mathcal{B}_{\mathsf{con}}[\![\mathsf{ap}(\mathtt{RED}(Q,t),k)]\!]$$

de modo que para todo $k \in \mathbb{N}$ existirá una barba $b_k \in \mathcal{B}_{\mathsf{con}}[\![\mathsf{ap}(\mathtt{RED}(Q,t),k)]\!]$ tal que no existe $b \in \mathcal{B}_{\mathsf{con}}[\![\mathtt{RED}(P,t)]\!]$ con $b \ll b_k$. Puesto que P es finito, existe una cota para la longitud de sus barbas:

$$\exists L \in \mathbb{N} : \ \forall b \in \mathcal{B}_{\mathsf{con}}[\![\mathtt{RED}(P,t)]\!] \quad \Rightarrow \quad \mathsf{lon}(b) \leq L$$

Si existiera $k \in \mathbb{N}$ tal que $\mathsf{lon}(b_k) > L$, podríamos encontrar $b' \in \mathcal{B}_{\mathsf{con}}[\![\mathtt{RED}(Q,t)]\!]$ tal que no existiría $b \in \mathcal{B}_{\mathsf{con}}[\![\mathtt{RED}(P,t)]\!]$ con $b \ll b'$. Tenemos por tanto $\mathsf{lon}(b_k) \leq L$ para todo $k \in \mathbb{N}$. Recordemos ahora que

$$\mathsf{ap}(\mathtt{RED}(Q,t),k) = \mathtt{RED}(\mathsf{ap}(Q,k),t)$$

entonces tendremos también que $\mathsf{nd}(b_k) \leq t$. Como quiera que estamos considerando un dominio de tiempo discreto y un alfabeto finito, tenemos que entre las barbas b_k sólo puede haber un número finito diferentes, con lo que podemos suponer que todas ellas son iguales a una dada b'. Tendremos entonces que $b' \in \mathcal{B}_{\mathrm{con}}[\![\mathtt{RED}(Q,t)]\!]$, no existiendo $b \in \mathcal{B}_{\mathrm{con}}[\![\mathtt{RED}(P,t)]\!]$ tal que $b \ll b'$, lo que está en contradicción con nuestras hipótesis.

□

Lema 5.3.8 Para cada proceso P se tiene $\mathsf{ap}(P,k) <_{\mathrm{E}} P$.

Demostración. Por inducción sobre k.

□

Por fin podemos concluir el resultado buscado:

Teorema 5.3.9 Para cualesquiera procesos P y Q se tiene

$$\mathcal{B}_{\mathrm{con}}[\![P]\!] \ll \mathcal{B}_{\mathrm{con}}[\![Q]\!] \quad \Rightarrow \quad P <_{\mathrm{E}} Q$$

Demostración. Puesto que $\mathcal{B}_{\mathrm{con}}[\![P]\!] \ll \mathcal{B}_{\mathrm{con}}[\![Q]\!]$, de r la monotonía del operador reductor se sigue

$$\mathcal{B}_{\mathrm{con}}[\![\mathtt{RED}(P,t)]\!] \ll \mathcal{B}_{\mathrm{con}}[\![\mathtt{RED}(Q,t)]\!]$$

Por tanto tenemos que para todo $k \in \mathbb{N}$

$$\mathcal{B}_{\mathrm{con}}[\![\mathsf{ap}(\mathtt{RED}(P,t),k)]\!] \ll \mathcal{B}_{\mathrm{con}}[\![\mathtt{RED}(Q,t)]\!]$$

En virtud de la proposición anterior, existe $l \in \mathbb{N}$ tal que

$$\mathcal{B}_{\mathrm{con}}[\![\mathsf{ap}(\mathtt{RED}(P,t),k)]\!] \ll \mathcal{B}_{\mathrm{con}}[\![\mathsf{ap}(\mathtt{RED}(Q,t),l)]\!]$$

Puesto que los procesos $\mathtt{RED}(\mathsf{ap}(Q,l),t)$ y $\mathtt{RED}(\mathsf{ap}(P,k),t)$ son finitos, por el teorema 5.3.6

$$\mathsf{ap}(\mathtt{RED}(P,t),k) <_{\mathrm{E}} \mathsf{ap}(\mathtt{RED}(Q,t),l)$$

Aplicando entonces el lema anterior concluimos $\mathsf{ap}(\mathtt{RED}(Q,t),l) <_{\mathrm{E}} \mathtt{RED}(Q,t)$, con lo cual para todo $k \in \mathbb{N}$ se tiene

$$\mathsf{ap}(\mathtt{RED}(P,t),k) <_{\mathrm{E}} \mathtt{RED}(Q,t)$$

Entonces, aplicando la regla [**APF**], obtenemos que para todo $t \in \mathcal{T}$

$$\mathtt{RED}(P,t) <_{\mathrm{E}} \mathtt{RED}(Q,t)$$

de donde la aplicación de la regla [**APT**], nos conduce a $P <_{\mathrm{E}} Q$.

□

y

Finalmente, resumimos en un teorema el resultado de este capítulo, que es consecuencia de la corrección de cada una de las reglas y axiomas, y de la completitud de todo el sistema con respecto a la relación $\ll$.

Teorema 5.3.10 Si P y Q son procesos entonces

$$\mathcal{B}_{\text{con}}[\![P]\!] \approx \mathcal{B}_{\text{con}}[\![Q]\!] \quad \Longleftrightarrow \quad P =_{\text{E}} Q$$

Demostración. Veamos por separado cada una de las implicaciones:

$\boxed{\Rightarrow}$

Puesto que $\mathcal{B}_{\text{con}}[\![P]\!] \approx \mathcal{B}_{\text{con}}[\![Q]\!]$ tenemos que

$$\mathcal{B}_{\text{con}}[\![P]\!] \ll \mathcal{B}_{\text{con}}[\![Q]\!] \quad \text{y} \quad \mathcal{B}_{\text{con}}[\![Q]\!] \ll \mathcal{B}_{\text{con}}[\![P]\!]$$

Por el teorema anterior tenemos que

$$P <_{\text{E}} Q \quad \text{y} \quad Q <_{\text{E}} P$$

Entonces por la ecuación [**EQ2**] tenemos que $P =_{\text{E}} Q$.

$\boxed{\Leftarrow}$

Que es consecuencia inmediata de la corrección de todas y cada una de las reglas y axiomas del sistema.

□

Capítulo 6

Operadores Derivados

En este capítulo enriqueceremos nuestro lenguaje básico con otros operadores característicos de las álgebras de procesos, y veremos la forma adecuada de tratarlos. En concreto, estudiaremos el operador paralelo y el de ocultamiento como operadores muy característicos de las álgebras de procesos. Pero también estudiaremos cómo podemos introducir intervalos de tiempo en el operador de prefijo, como operador característico de las álgebras (aplicables a la práctica) de procesos temporizadas. Estos operadores llevan el calificativo de derivados, porque se *pueden* eliminar de los procesos finitos. Aunque esto último no es cierto cuando se consideran intervalos infinitos en el operador de prefijo, sin embargo no necesitaremos ningún tipo de herramienta adicional a la hora de manejarlo.

Cada una de las secciones de este capítulo sigue el siguiente esquema:

- Se ampliará la sintaxis con un nuevo operador (o bien una familia de operadores) *op* de aridad n:
$$\Sigma_{op} = \Sigma_{\mathrm{seq}+} \cup \{op\}$$
- Al los operador introducido se le dotará de una semántica operacional. Es decir, se dará un conjunto de reglas y axiomas que permitirán deducir las transiciones que un proceso construido con este operador puede hacer. Esta semántica operacional ha de verificar las propiedades de la sección 3.2.1.
- Definiremos a continuación la semántica de pruebas, para ello bastará con extender la relación de convergencia a este nuevo operador, y para ello habremos de extender la definición del predicado de convergencia débil. En principio deberíamos extender también el mundo de las pruebas al poder aparecer en ellas este nuevo operador. Podría pensarse que con ello se podría haber añadido más potencia al mecanismo de prueba, Pero ello no es así, pues si nos fijamos en la forma en la que se define

el paso de pruebas, nos damos cuenta que al efecto no utilizamos para nada la sintaxis de los procesos. Nos limitamos a utilizar las propiedades de la semántica operacional enumeradas en la sección 3.2.1. Pero como quiera que estas propiedades se siguen conservando, la caracterización operacional alternativa sigue siendo válida. Además obtenemos de rebote que la potencia de las pruebas no ha cambiado; es decir, ni ganamos ni perdemos nada si sólo admitimos pruebas construidas con los operadores básicos introducidos en el capítulo 3.

- A continuación se definirá la semántica denotacional del operador:

$$\mathcal{B}_{\text{con}}[\![op]\!] : \underbrace{\mathcal{B}_{\text{con}} \times \cdots \times \mathcal{B}_{\text{con}}}_{n} \longmapsto \mathcal{B}_{\text{con}}$$

que tendrá que ser monótona con respecto a la relación $\ll$ (y por tanto congruente con $\approx$), y además monótona y contínua con respecto a la relación $\prec$. Además, tendremos que comprobar que la denotación que hemos escogido para el operador no es arbitraria, sino que tiene mucho que ver con la semántica de pruebas. En concreto hemos de ver que

$$\begin{array}{lcl} B_i \ll \mathsf{Barb}(P_i) & \Rightarrow & \mathcal{B}_{\text{con}}[\![op]\!](B_1,\ldots,B_n) \ll \mathsf{Barb}(op(P_1,\ldots,P_n)) \\ \mathsf{Barb}(P_i) \ll B_i & \Rightarrow & \mathsf{Barb}(op(P_1,\ldots,P_n)) \ll \mathcal{B}_{\text{con}}[\![op]\!](B_1,\ldots,B_n) \end{array}$$

- Por último daremos un conjunto de ecuaciones que permitirán *eliminar* dicho operador, al menos de los procesos finitos. Si $P_1,\ldots,P_n$ son procesos finitos, podremos pasarlos a forma prenormal $P_1',\ldots,P_n'$, y tendremos que

$$op(P_1,\ldots,P_n) =_{\text{E}} op(P_1',\ldots P_n')$$

A continuación, las reglas que daremos nos permitirán eliminar el operador op del proceso $op(P_1',\ldots P_n')$ obteniendo así un proceso finito $P' \in \mathsf{FCRec}(\Sigma_{\text{seq}})$ que verifique

$$P' =_{\text{E}} op(P_1',\ldots P_n')$$

Por último tendremos que $P' =_{\text{E}} op(P_1,\ldots,P_n)$, y consecuencia podremos seguir razonando como en el capítulo 5.

6.1 Paralelo

Muy probablemente éste es el operador más característico de un álgebra de procesos (concurrente). *Un operador paralelo toma dos procesos y los pone a funcionar simultáneamente* de forma concurrente. En nuestro caso consideramos un conjunto de acciones G en las cuales los dos procesos han de sincronizar, de modo que para el proceso resultante de la composición en paralelo pueda ejecutar dichas acciones es necesaria la colaboración de ambos argumentos. Formalmente tenemos la siguiente familia de operadores de aridad dos:

$$\{\|_G \mid G \subseteq Act\}$$

que representaremos, como habitualmente se hace, en forma infija.

6.1.1 Semántica Operacional

Para dar la semántica operacional de un nuevo operador, hemos de comenzar por extender la definición de las funciones auxiliares que intervienen en las reglas de las transiciones. Para extender la definición de la función $\mathsf{stb}(\cdot)$ no tenemos ningún problema:

$$\mathsf{stb}(P \|_G Q) = \mathsf{stb}(P) \wedge \mathsf{stb}(Q)$$

Para definir la función $\mathsf{Tiem}(\cdot, a)$ tenemos que tener en cuenta si $a \in G$ o no:

$$\mathsf{Tiem}(P \|_G Q, a) = \begin{cases} \mathsf{Tiem}(P,a) \cup \mathsf{Tiem}(Q,a) & \text{si } a \notin G \\ \mathsf{Tiem}(P,a) \cap \mathsf{Tiem}(Q,a) & \text{si } a \in G \end{cases}$$

La función $\mathsf{idle}(P \|_G Q, A)$ resulta algo más compleja; hemos de tener en cuenta los conjuntos $A \cap G$ y $A \setminus G$:

$$\begin{aligned} &\mathsf{idle}(P \|_G Q, A) = \min(\mathsf{idle}(P, A \setminus G), \mathsf{idle}(Q, A \setminus G), t) \\ &\text{donde} \quad t = \min\Big(\bigcup_{a \in A \cap G} \mathsf{Tiem}(P,a) \cap \mathsf{Tiem}(Q,a)\Big)^* \end{aligned}$$

Definimos por último la función de actualización como sigue:

$$\mathsf{Upd}(P \|_G Q) = \mathsf{Upd}(P,t) \|_G \mathsf{Upd}(Q,t)$$

Una vez extendidas las funciones auxiliares podemos dar las reglas que definen las transiciones de la semántica operacional. En primer lugar, si alguno de los procesos puede realizar alguna transición vacía, también la podrá realizar el proceso combinación de los dos:

$$\textbf{[PAR1]} \quad \frac{P \rightarrowtail P'}{P \|_G Q \rightarrowtail P' \|_G Q} \qquad \textbf{[PAR2]} \quad \frac{Q \rightarrowtail Q'}{P \|_G Q \rightarrowtail P \|_G Q'}$$

*Tenemos como mínimo del conjunto vacío es infinito.

minutos

Por otro, lado si uno de los procesos puede ejecutar un evento $e \notin G$, y su compañero puede esperar, entonces la composición paralela también podrá ejecutar dicho evento:

$$[\mathbf{PAR3}]\quad \frac{P \xrightarrow{et} P',\ \mathsf{stb}(Q),\ \mathsf{idle}(Q,t)}{P \parallel_G Q \xrightarrow{et} P' \parallel_G \mathsf{Upd}(Q,t)} \qquad [\mathbf{PAR4}]\quad \frac{Q \xrightarrow{et} Q',\ \mathsf{stb}(P),\ \mathsf{idle}(P,t)}{P \parallel_G Q \xrightarrow{et} \mathsf{Upd}(P,t) \parallel_G Q'}$$

Por último la ejecución de las acciones del conjunto de sincronización requieren la colaboración de ambos procesos:

$$[\mathbf{PAR5}]\quad \frac{P \xrightarrow{at} P',\ Q \xrightarrow{at} Q'}{P \parallel_G Q \xrightarrow{at} P' \parallel_G Q'} \qquad a \in G$$

Para concluir, observamos que se siguen conservando todas las propiedades de la semántica operacional citadas en la sección 3.2.1. En particular se siguen conservando las propiedades de la urgencia y del no determinismo acotado, propiedades que son claves a la hora de caracterizar la semántica de pruebas.

6.1.2 Semántica de Pruebas

Para definir la semántica de pruebas hemos de comenzar extendendiendo la definición de convergencia débil. Para ello añadimos en la definición 3.3.1 la siguiente condición

$$P \downarrow\ \wedge\ Q \downarrow \quad \Rightarrow \quad P \parallel_G Q \downarrow$$

Como dijimos anteriormente definiendo el predicado de convergencia débil, tenemos automáticamente definido el predicado de convergencia, y esto es todo lo que necesitamos para extender la semántica de pruebas.

6.1.3 Semántica Denotacional

Definiremos ahora la semántica denotacional del nuevo operador. Para ello necesitamos la siguiente definición auxiliar (conviene refrescar aquí, la definición de $A_1 \sqcup_G A_2$ en la definición 4.2.10):

Definición 6.1.1 Siendo b_1 y b_2 barbas y $G \subseteq Act$, definimos el conjunto de barbas $b_1 \parallel_G b_2$ como el menor conjunto de barbas que verifique:

$$b_1 = A_1 \ \text{ y } \ b_2 = A_2 \quad \Rightarrow \quad b_1 \parallel_G b_2 = \{A_1 \sqcup_G A_2\}$$

$$\left.\begin{array}{l} b_1 = A_1 at \cdot b_1',\ b_2 = A_2,\ a \notin G, \\ \mathsf{nd}(A_2) \geq t \ \text{ y } \ b' \in b_1' \parallel_G (A_2 - t) \end{array}\right\} \Rightarrow \quad (A_1 \sqcup_G A_2 \upharpoonright t) at \cdot b' \in b_1 \parallel_G b_2$$

$$\left.\begin{array}{l} b_1 = A_1,\ b_2 = A_2at\cdot b_2',\ a \notin G, \\ \mathsf{nd}(A_1) \geq t \ \text{ y } \ b' \in (A_1 - t) \parallel_G b_2' \end{array}\right\} \Rightarrow \quad (A_1 \sqcup_G A_2 \rceil t)at \cdot b' \in b_1 \parallel_G b_2$$

$$\left.\begin{array}{l} b_1 = A_1a_1t_1\cdot b_1', \quad b_2 = A_2a_2t_2\cdot b_2', \\ t_1 \leq t_2, \quad a_1 \notin G \ \text{ y} \\ b' \in b_1' \parallel_G (A_2 - t_1)a_2(t_2 - t_1)\cdot b_2' \end{array}\right\} \Rightarrow \quad (A_1 \sqcup_G A_2 \rceil t_1)a_1t_1 \cdot b' \in b_1 \parallel_G b_2$$

$$\left.\begin{array}{l} b_1 = A_1a_1t_1\cdot b_1', \quad b_2 = A_2a_2t_2\cdot b_2', \\ t_1 \geq t_2, \quad a_2 \notin G \ \text{ y} \\ b' \in (A_1 - t_2)a_1(t_1 - t_2)\cdot b_1' \parallel_G b_2' \end{array}\right\} \Rightarrow \quad (A_1 \rceil t_2 \sqcup_G \cup A_2)a_2t_2 \cdot b' \in b_1 \parallel_G b_2$$

$$\left.\begin{array}{l} b_1 = A_1at\cdot b_1', \quad b_2 = A_2at\cdot b_2', \\ a \in G \ \text{ y } \ b' \in b_1' \parallel_G b_2' \end{array}\right\} \Rightarrow \quad (A_1 \sqcup_G A_2)at \cdot b' \in b_1 \parallel_G b_2$$

□

Obsérvese la simetría de la definición anterior: para cualquier par de barbas b_1 y b_2 se tiene $b_1 \parallel_G b_2 = b_2 \parallel_G b_1$.

Definición 6.1.2 Si B_1 y B_2 son conjuntos consistentes de barbas y $G \subseteq Act$, tomamos

$$\mathcal{B}_{\text{con}}[\![\parallel_G]\!](B_1, B_2) = \Big\{ b \mid \exists\ b_1 \in B_1,\ b_2 \in B_2 :\ b \in b_1 \parallel_G b_2 \Big\}$$

□

Debido a la simetría de la definición de $b_1 \parallel_G b_2$, tendremos trivialmente que este operador será conmutativo: $B_1 \parallel_G B_2 = B_2 \parallel_G B_1$. Comprobamos ahora que la función $\mathcal{B}_{\text{con}}[\![\parallel_G]\!]$ está bien definida. Para ello necesitamos algo más de notación:

Definición 6.1.3 Siendo B un conjunto consistente de barbas, $A \in B$ un estado y $t \in \mathcal{T}$ tal que $t \leq \mathsf{nd}(A)$, tomamos

$$\begin{array}{rl} \mathsf{Barb}(B, A, t) = & \{A - t\} \cup \\ & \Big\{ (A' - t)a'(t' - t)\cdot b' \in B \ \Big|\ A'a't'\cdot b' \in B,\ t' \geq t,\ A \rceil t' = A' \Big\} \end{array}$$

□

Resulta sencillo comprobar que, en las condiciones de la definición anterior, el conjunto de barbas $\mathsf{Barb}(B, A, t)$ es consistente.

---

a

Proposición 6.1.4 Si B_1 y B_2 son conjuntos consistentes y $G \subsetneq Act$, entonces el conjunto de barbas $\mathcal{B}_{\text{con}}[\![\|_G]\!](B_1, B_2)$ es consistente.

Demostración. Para facilitar la notación tomemos $B = \mathcal{B}_{\text{con}}[\![\|_G]\!](B_1, B_2)$. Hemos de probar probar:

$\mathbf{B} \neq \varnothing$.

Observemos al efecto que, como B_1 y B_2 son consistentes, han de existir dos estados $A_1 \in B_1$ y $A_2 \in B_2$. Tomemos entonces el estado $A = A_1 \sqcup_G A_2$ que claramente pertenece al conjunto de barbas B.

Cerrado bajo prefijos.

Consideremos la b-traza

$$bs = A_1 a_1 t_1 \cdots A_{n-1} a_{n-1} t_{n-1} A_n a_n t_n \in \mathsf{Btraz}(B)$$

Por la definición de $\mathsf{Btraz}(B)$ existe una barba b' de modo que la barba $b = bs \cdot b'$ está en el conjunto B. Por la definición del operador paralelo existen barbas $b_1 \in B_1$ y $b_2 \in B_2$ de manera que $b \in b_1 \|_G b_2$. La demostración sigue entonces por inducción sobre la longitud de la b-traza bs ($n = \mathsf{lon}(bs)$).

$\mathbf{n = 1}$. Tenemos las siguientes casos posibles:

- $a_1 \in G$, $b_1 = A^1 a_1 t_1 \cdot b_1'$, $b_2 = A^2 a_1 t_1 \cdot b_2'$, $A_1 = A^1 \sqcup_G A^2$ y $b \in b_1' \|_G b_2'$. Puesto que B_1 y B_2 son consistentes existirán estados $A_1' \in B_1$ y $A_2' \in B_2$ tales que
$$A_1' \upharpoonright t_1 = A^1 \quad \text{y} \quad A_2' \upharpoonright t_1 = A^2$$
Basta entonces tomar $A = A_1' \sqcup_G A_2'$ para obtener el resultado deseado.
- $a_1 \notin G$, $b_1 = A^1 a_1 t_1 \cdot b_1'$ y $b_2 = A^2 a' t' \cdot b_2'$ con $t_1 \leq t'$ o bien $b_2 = A^2$ con $\mathsf{nd}(A^2) \geq t_1$. Puesto que B_1 es consistente existe un estado $A_1' \in B_1$ tal que $A_1' \upharpoonright t_1 = A^1$. Obtenemos el estado A_2' de la siguiente manera:
  - Si $b_2 = A^2 a' t' \cdot b_2'$ existe un estado $A_2' \in B_2$ tal que $A_2' \upharpoonright t' = A^2$.
  - Si $b_2 = A_2$ basta tomar $A_2' = A_2$.

  Es claro entonces que $A = A_1' \sqcup_G A_2'$ nos conduce al resultado.
- $a_1 \notin G$, $b_2 = A^2 a_1 t_1 \cdot b_2'$ y $b_1 = A^1 a' t' \cdot b_1'$ con $t_1 \leq t'$ o bien $b_1 = A^1$ con $\mathsf{nd}(A^1) > t_1$. Que es totalmente simétrico al anterior.

$\mathbf{n > 1}$. Ahora tenemos los siguientes casos:

- $a_1 \in G$, $b_1 = A^1 a_1 t_1 \cdot b_1'$ y $b_2 = A^2 a_1 t_1 \cdot b_2'$. Tomamos $B_1' = \mathsf{Barb}(B_1, A^1 a_1 t_1)$ y $B_2' = \mathsf{Barb}(B_2, A^2 a_1 t_1)$, y tenemos que

$$A_2 a_2 t_2 \cdots A_{n-1} a_{n-1} t_{n-1} \cdots A_n a_n t_n \cdot b' \in B_1' \parallel_G B_2' \subseteq \mathsf{Barb}(B, A_1 a_1 t_1)$$

De modo que el resultado deseado se sigue aplicando la hipótesis de inducción.

- $a_1 \notin G$, $b_1 = A^1 a_1 t_1 \cdot b_1'$ y bien $b_2 = A^2 a' t' \cdot b_2'$ con $t_1 \leq t'$ o bien $b_2 = A^2$ con $\mathsf{nd}(A_2) \geq t_1$. Consideramos por un lado $B_1' = \mathsf{Barb}(B_1, A_1 a_1 t_1)$, y B_2' dependiendo de la barba b_2, definido como sigue:
  - Si $b_2 = A^2 a' t' \cdot b_2'$, puesto que B_2 es cerrado bajo prefijos, existirá un estado $A_2' \in B_2$ tal que $A_2' \upharpoonright t' = A^2$. En este caso tomamos el conjunto $B_2' = \mathsf{Barb}(B_2, A_2', t_1)$.
  - En cambio, si $b_2 = A^2$ tomamos $B_2' = \mathsf{Barb}(B, A^2, t_1)$.

  En cualquier caso tenemos que tanto B_1' y B_2' son conjuntos consistentes de barbas y

$$A_2 a_2 t_2 \cdots A_{n-1} a_{n-1} t_{n-1} \cdots A_n a_n t_n \cdot b' \in B_1' \parallel_G B_2' \subseteq \mathsf{Barb}(B, A_1 a_1 t_1)$$

Al igual que antes, obtenemos el resultado aplicando la hipótesis de inducción.

- $a_1 \notin G$, $b_2 = A^1 a_1 t_1 \cdot b_2'$ y bien $b_1 = A^2 a' t' \cdot b_1'$ con $t' \geq t_1$ o bien $b_1 = A^2$ con $\mathsf{nd}(A^2) > t_1$. Caso totalmente simétrico al anterior.

Cerrado bajo continuaciones.

Sea ahora $b = bs \cdot A \in B$ y $at \in A$. Haremos la demostración por inducción sobre la longitud de la barba b.

b = A. Por la definición de operador paralelo existen dos estados $A_1' \in B_1$ y $A_2' \in B_2$ tales que

$$A = A_1 \sqcup_G A_2$$

Tenemos las siguientes posibilidades:

- $a \in G$, $at \in A_1$ y $at \in A_2$. Puesto que B_1 y B_2 son conjuntos consistentes de barbas existirán barbas

$$b_1 = A_1' at A_1'' \in B_1 \quad \text{y} \quad b_2 = A_2' at A_2'' \quad \text{siendo } A_1 = A_1 \upharpoonright t \ \text{ y } A_2' = A_2 \upharpoonright t$$

Tomando entonces

$$b = (A_1' \sqcup_G A_2') at (A_1'' \sqcup_G A_2'') \in b_1 \parallel_G b_2$$

la

llegamos al resultado deseado.

- $a \notin G$ y $at \in A_1$. Puesto que B_1 es un conjunto consistente de barbas, existe una barba $b_1 = A_1' at A_1'' \in B_1$ tal que $A_1' = A_1 \rceil t$. Tomando entonces

$$b = (A_1' \sqcup_G A_2' \rceil t) at (A_1'' \sqcup_G (A_2'' - t)) \in b_1 \parallel_G A_2$$

obtenemos el resultado.

- $a \notin G$ y $at \in A_1$. Este caso es simétrico al anterior.

$\mathbf{b = Aa't' \cdot b'}$. Existen entonces dos barbas $b_1 \in B_1$ y $b_2 \in B_2$ tales que $b \doteq b_1 \parallel_G b_2$. Tenemos varias posibilidades:

- $a \in G$, $b_1 = A_1 a't' \cdot b_1'$ y $b_2 = A_2 a't' \cdot b_2'$. Tomamos $B_1' = \mathsf{Barb}(B_1, A_1 a't')$ y $B_2' = \mathsf{Barb}(B_2, A_2 a't')$ y se verificará

$$b' \in B_1' \parallel_G B_2' \subseteq \mathsf{Barb}(B, Aa't')$$

Lo que nos conduce al resultado aplicando la hipótesis de inducción.

- $a \in G$, $b_1 = A_1 a't' \cdot b_1'$ y bien b_2 es una barba de la forma $A_2 a_2 t_2 \cdot b_2'$ con $t_2 \geq t'$ o bien es un estado A_2 verificando $\mathsf{nd}(A_2) \geq t'$. Consideramos ahora los conjuntos de barbas $B_1' = \mathsf{Barb}(B_1, A_1 a't')$ y B_2' dependiendo de b_2 definido en la siguiente manera:

  - Si $b_2 = A_2 a_2 t_2 \cdot b_2'$, puesto que B_2, es consistente existe un estado $A_2' \in B_2$ tal que $A_2' \rceil t_2 = A_2$. Tomamos entonces $B_2' = \mathsf{Barb}(B_2, A_2', t')$.
  - Si $b_2 = A_2$ tomamos $B_2' = \mathsf{Barb}(B_2, A_2, t')$.

  Tenemos entonces

$$b' \in B_1' \parallel_G B_2' \subseteq \mathsf{Barb}(B, Aa't')$$

  de donde se sigue el resultado aplicando la hipótesis de inducción.

- $a \in G$, $b_2 = A_2 a't' \cdot b_2'$ y bien $b_1 = A_1 a_1 t_1 \cdot b_1'$ con $t_1 \geq t'$ o bien $b_1 = A_1$ con $\mathsf{nd}(A_1) \geq t'$, que es simétrico al anterior.

Temporalmente compacto. Sea b una barba tal que $\mathsf{nd}(b) = \infty$ y supongamos que para cada $t \in \mathcal{T}$ existe $b_t \in B$ tal que $b \rceil t = b_t \rceil t$. Entonces para cada $t \in \mathcal{T}$ existirán barbas $b_{1,t} \in B_1$ y $b_{2,t} \in B_2$ tales que $b_t \in b_{1,t} \parallel_G b_{2,t}$. Fijado $t \in \mathcal{T}$ sólo puede haber un número finito de barbas $b_{1,t'}$ con $t' \geq t$ tales que las barbas $b_{1,t'} \rceil t$ sean diferentes, y lo mismo ocurre para las barbas $b_{2,t'}$. Podemos suponer por tanto que $b_{1,t} \upharpoonright t = b_{1,t'} \upharpoonright t$ y $b_{2,t} \upharpoonright t = b_{2,t'} \upharpoonright t$ para $t' \geq t$. Existirán entonces $b_1 \in B_1$ y $b_2 \in B_2$ tales que para cada $t \in \mathcal{T}$ se tenga

$$b_1 \upharpoonright t = b_{1,t} \upharpoonright t \ \wedge \ b_2 \upharpoonright t = b_{2,t} \upharpoonright t$$

Por lo que $b \in b_1 \parallel_G b_2$.

□

Hemos de probar además que el operador paralelo respeta la relación $\ll$.

Lema 6.1.5 Si $b \in b_1 \parallel_G b_2$, $b_1' \ll b_1$, y $\mathsf{lon}(b_1') = \mathsf{lon}(b_1)$ entonces existe $b' \in b_1' \parallel_G b_2$ tal que $b' \ll b$.

Demostración. La demostración es inmediata por inducción sobre la longitud de b.

□

Puesto que tenemos que $b \parallel_G b' = b' \parallel_G b$, el lema anterior puede también aplicarse a partir de $b_2' \ll b_2$. Como consecuencia de este lema tenemos la

Proposición 6.1.6 Si B_1, B_2 y B_1' son conjuntos consistentes de barbas, se tiene

$$B_1 \ll B_1' \quad \Rightarrow \quad \mathcal{B}_{\text{con}}[\![\parallel_G]\!](B_1, B) \ll \mathcal{B}_{\text{con}}[\![\parallel_G]\!](B_1', B)$$

Puesto que el operador paralelo es conmutativo tendremos además

$$\mathcal{B}_{\text{con}}[\![\parallel_G]\!](B, B_1) \ll \mathcal{B}_{\text{con}}[\![\parallel_G]\!](B, B_1')$$

Demostración. Sea $b \in \mathcal{B}_{\text{con}}[\![\parallel_G]\!](B_1', B)$; por la definición existen barbas $b_1' \in B_1'$ y $b_2 \in B$ tales que $b \in b_1' \parallel_G b_2$. Puesto que $B_1 \ll B_1'$ existe una barba $b_1 \in B_1$ tal que $b_1 \ll b_1'$. Si $\mathsf{lon}(b_1) = \mathsf{lon}(b_1')$, obtenemos el resultado mediante la aplicación del lema anterior. Si por el contrario $\mathsf{lon}(b_1) < \mathsf{lon}(b_1')$, podemos tomar $t = \mathsf{nd}(b_1) < \infty$, y como tanto B_1 y B_2 son conjuntos consistentes existirán barbas $b_1'' \in B_1$ y $b_2'' \in B_2$ tales que

$$b_1'' \upharpoonright t = b_1' \quad \text{y} \quad b_2'' \upharpoonright t$$

Entonces existirá $b'' \in b_1'' \parallel_G b_2''$ tal que $b'' \upharpoonright t = b$. Puesto que también se verifica $b_1 \ll b_1''$, aplicando el lema anterior existe

$$b''' \in b_1 \parallel_G b_2'' \subseteq \mathcal{B}_{\text{con}}[\![\parallel_G]\!](B_1, B)$$

tal que $b''' \ll b''$. Tenemos entonces que $\mathsf{nd}(b''') = \mathsf{nd}(b_1)$, y con lo que $b''' \ll b$.

□

semana.

Dado que el operador paralelo es conmutativo, esta última proposición implica la monotonía del mismo con respecto a la relación $\ll$, o lo que es lo mismo, la congruencia con respecto a la relación $\approx$. Para acabar con el estudio de la semántica denotacional de este operador hemos de probar la continuidad del mismo.

6.1.3.1 Continuidad

En primer lugar demostraremos la monotonía del operador con respecto a la relación $\prec$. Para ello damos un lema similar al lema 6.1.5

Lema 6.1.7 Si $b \in b_1 \|_G b_2$, $b_1' \prec b_1$ y $\mathsf{lon}(b_1') = \mathsf{lon}(b_1)$ entonces existe $b' \in b_1' \|_G b_2$ tal que $b' \prec b$.

Demostración. La demostración es inmediata por inducción sobre la longitud de b.

□

Puesto que tenemos que $b \|_G b' = b' \|_G b$, el lema anterior es también válido si partimos de $b_2' \prec b_2$. Debido a la conmutatividad del operador, la monotonía está directamente implicada por la siguiente

Proposición 6.1.8 Para cualesquiera conjuntos consistentes de barbas B_1, B_2 y B_1', se tiene

$$B_1 \prec B_1' \quad \Rightarrow \quad B_1 \|_G B \prec B_1' \|_G B$$

Demostración. Esta demostración es prácticamente igual que la de la proposición 6.1.6, pero aplicando este caso el lema anterior.

□

Una vez visto que el operador es monótono con respecto a $\prec$, podemos abordar la prueba de su continuidad. Para ello será suficiente probar la

Proposición 6.1.9 Siendo $\mathcal{B} = \{B_i \mid i \in \mathbb{N}\}$ una cadena no decreciente de conjuntos de barbas consistentes se tiene

$$\mathsf{lub}(\mathcal{B}) \|_G B = \mathsf{lub}\{B_i \|_G B \mid i \in \mathbb{N}\}$$

Demostración. Para demostrar la igualdad entre ambos conjuntos de barbas veamos la doble inclusión.

$\boxed{\supseteq}$

Sea $b = bs \cdot A \in \mathsf{lub}\{B_i \|_G B \mid i \in \mathbb{N}\}$. Supongamos en primer lugar que $\mathsf{nd}(b) < \infty$;

en tal caso tenemos

$$\forall k \in \mathbb{N}\ \exists l \geq k:\ b \in B_l \parallel_G B$$

De modo que para cada l existirá barbas $b_{l,1} \in B_l$ y $b_{l,2} \in B$ tal que $b \in b_{l,1} \parallel_G b_{l,2}$. Puesto que $\mathsf{nd}(b) < \infty$ tenemos que o bien $\mathsf{nd}(b_{l,1}) = \mathsf{nd}(b)$, o bien $\mathsf{nd}(b_{l,2}) = \mathsf{nd}(b)$. En principio no tiene porqué ser cierta siempre la mismo, pero sí podremos encontrar una subsecuencia de modo que ello suceda. Podemos suponer por tanto que es siempre cierta la primera opción, o bien la segunda. En consecuencia tenemos dos posibilidades:

- Para todo l se verifica que $t = \mathsf{nd}(b_{l,1}) = \mathsf{nd}(b)$. Puesto que estamos considerando un dominio de tiempo discreto y un alfabeto finito, podemos suponer que todas las $b_{l,1}$ son iguales. Tomamos entonces como b_1 cualquiera de ellos y tendremos

  $$b_1 \in \mathsf{lub}\{B_i \mid i \in \mathbb{N}\}$$

  En principio las $b_{l,2}$ no tienen porqué ser iguales, pero sólo puede haber un número finito de barbas $b_{l,2} \upharpoonright t$, por lo que podemos suponer que las barbas $b_{l,2} \upharpoonright t$ son iguales. Tomamos como b_2 una cualquiera de las $b_{l,2}$, y tendremos

  $$b \in b_1 \parallel_G b_2 \subseteq \mathsf{lub}\{B_i \mid i \in \mathbb{N}\} \parallel_G B$$

- Para todo l se verifica que $t = \mathsf{nd}(b_{l,2}) = \mathsf{nd}(b)$. Al igual que antes podemos suponer que todas las $b_{l,2}$ son todas iguales, tomemos b_2 cualquiera de ellas. Por otro lado tenemos que sólo puede haber un número finito de barbas $b_{l,1} \upharpoonright t$ diferentes, con lo que podremos suponer que todas ellas son iguales. Tenemos entonces que existirá $b_1 \in \mathsf{lub}\{B_i \mid i \in \mathbb{N}\}$ tal que $b_1 \upharpoonright t = b_{l,1} \upharpoonright t$. Tenemos entonces

  $$b \in b_1 \parallel_G b_1 \subseteq \mathsf{lub}\{B_i \mid i \in \mathbb{N}\} \parallel_G B$$

Supongamos ahora que $\mathsf{nd}(b) = \infty$. En este caso tal se verifica que

$$\forall t \in \mathcal{T}\ \exists l \in \mathbb{N},\ b_t \in B_l \parallel_G B:\ b \upharpoonright t = b_t \upharpoonright t$$

Existen entonces $b_{t,1} \in B_l$ y $b_{t,2} \in B$ tales que $b_t \in b_{t,1} \parallel_G b_{t,2}$. Puesto que estamos considerando un alfabeto finito y un dominio de tiempo discreto, podemos suponer que $b_{t,1} \upharpoonright t = b_{t',1} \upharpoonright t$ y $b_{t,1} \upharpoonright t = b_{t',1} \upharpoonright t$ para todo $t' \geq t$. Existirá entonces una barba $b_1 \in \mathsf{lub}\{B_i \mid i \in \mathbb{N}\}$ tal que

$$\forall t \in \mathcal{T}:\ b_{t,1} \upharpoonright t = b_1 \upharpoonright t$$

-¿Y

Entonces, puesto que B es consistente, existirá $b_2 \in B$ tal que

$$\forall t \in \mathcal{T} : \ b_{t,2} \upharpoonright t = b_2 \upharpoonright t$$

Tendremos entonces que $b \in b_1 \parallel_G b_2 \subseteq \mathsf{lub}\{B_i \mid i \in \mathbb{N}\} \parallel_G B$.

$\boxed{\subseteq}$

Tomemos ahora $b \in \mathsf{lub}(\mathcal{B}) \parallel_G B$; por la definición del operador existirán sendas barbas $b_1 \in \mathsf{lub}(\mathcal{B})$ y $b_2 \in B$ tales que $b \in b_1 \parallel_G b_2$. Para probar

$$b \in \mathsf{lub}\{B_i \parallel_G B \mid i \in \mathbb{N}\}$$

distingamos dos casos en función de la barba b:

- $\mathsf{nd}(b) < \infty$. Tal y como están definidos los operadores tenemos

  $$\mathsf{nd}(b) = \min(\mathsf{nd}(b_1), \mathsf{nd}(b_2))$$

  Por tanto volvemos a tener dos posibilidades:

  - $\mathsf{nd}(b_1) = \mathsf{nd}(b)$. En este caso para cada $k \in \mathbb{N}$ existe $l_k \geq k$ tal que $b_1 \in B_{l_k}$. Por tanto tenemos

    $$b \in B_{l_k} \parallel_G B$$

    y como consecuencia $b \in \mathsf{lub}\{B_i \parallel_G B \mid i \in \mathbb{N}\}$.
  - $\mathsf{nd}(b_2) = \mathsf{nd}(b)$. Tenemos en este caso que $\mathsf{nd}(b_1) \geq \mathsf{nd}(b_2)$. Tomemos $t = \mathsf{nd}(b_2)$; existe entonces $k \in \mathbb{N}$ tal que para cada $l \geq k$ existe una barba b_l verificando $b_1 \upharpoonright t = b \upharpoonright t$. Tenemos entonces que

    $$b \in b_1 \parallel_G b_2 = b_l \parallel_G b_2 \qquad \text{para} \qquad l \geq k$$

    por lo que $b \in \mathsf{lub}\{B_i \parallel_G B \mid i \in \mathbb{N}\}$.
- $\mathsf{nd}(b) = \infty$, por lo que $\mathsf{nd}(b_1) = \infty$. Entonces, por la definición, tenemos que para cada $t \in \mathcal{T}$ existen $l_t \in \mathbb{N}$ y una barba $b_t \in B_{l_t}$ tales que $b_1 \upharpoonright t = b_t \upharpoonright t$. Puesto que también $\mathsf{nd}(b_2) = \infty$ existirá $b'_t \in b_t \parallel_G b_2$ tal que

  $$b \upharpoonright t = b'_t \upharpoonright t$$

  y por tanto $b \in \mathsf{lub}\{B_i \parallel_G B \mid i \in \mathbb{N}\}$.

$\square$

6.1.3.2 Abstracción

En esta sección veremos que la definición de la semántica denotacional está bien relacionada con su semántica de pruebas. Estudiamos en primer lugar la relación entre los respectivos conjuntos de estados. Para ello precisamos de una serie de lemas previos similares a los que vimos en el caso del operador de elección externa ($\Box$). La mayoría de las demostraciones son bastante rutinarias y no entrañan ninguna dificultad especial. Para facilitar la lectura del trabajo postergamos las mismas hasta el apéndice A.

Lema 6.1.10 Sean P y Q procesos estables y convergentes y sea $G \subseteq Act$. Entonces se verifican las siguientes propiedades:

- Si $\mathsf{idle}(P) = \mathsf{idle}(Q) = \infty$ entonces

$$\mathsf{TA}(P\|_G Q) = \mathsf{TA}(Q\|_G P) = \mathsf{TA}(P) \sqcup_G \mathsf{TA}(Q)$$

- Si $P \xrightarrow{\tau t} P'$ e $\mathsf{idle}(Q) \geq t$, entonces

$$\mathsf{TA}(P\|_G Q) \upharpoonleft t = \mathsf{TA}(Q\|_G P) \upharpoonleft t = (\mathsf{TA}(P) \upharpoonleft t) \sqcup_G (\mathsf{TA}(Q) \upharpoonleft t)$$

Demostración. En el primer caso hay que tener en cuenta que siendo $a \notin G$ tenemos

$$P \|_G Q \xrightarrow{at'} R \quad \Longleftrightarrow \quad P \xrightarrow{at'} R \vee Q \xrightarrow{at'} R$$

Mientras que para $a \in G$ tenemos

$$P \|_G Q \xrightarrow{at'} R \quad \Longleftrightarrow \quad P \xrightarrow{at'} R \wedge Q \xrightarrow{at'} R$$

En el segundo caso , si $a \in G$, además se verifica

$$P \|_G Q \xrightarrow{at'} R \quad \Longleftrightarrow \quad t' \leq t \wedge (P \xrightarrow{at'} R \vee Q \xrightarrow{at'} R)$$

y si $a \notin G$ entonces

$$P \|_G Q \xrightarrow{at'} R \quad \Longleftrightarrow \quad t' \leq t \wedge (P \xrightarrow{at'} R \wedge Q \xrightarrow{at'} R)$$

$\Box$

Lema 6.1.11 Siendo $A \in \mathcal{A}(P\|_G Q)$ y $t \in \mathcal{T}$, se tiene

- $\mathsf{nd}(A) < t \quad \Rightarrow \exists A_P \in \mathcal{A}(P),\ A_Q \in \mathcal{A}(Q): \ A_P \sqcup_G A_Q \ll A$

---

qué

- $\mathsf{nd}(A) \geq t \quad \Rightarrow \exists A_P \in \mathcal{A}(P),\ A_Q \in \mathcal{A}(Q) :\ (A_P \sqcup_G A_Q) \upharpoonright t \subseteq A \upharpoonright t.$

$\square$

Proposición 6.1.12 Para cualesquiera procesos P y Q, si $A \in \mathcal{A}(P \|_G Q)$ existirán $A_P \in \mathcal{A}(P)$ y $A_Q \in \mathcal{A}(Q)$ de modo que $A_P \sqcup_G A_Q \ll A$.

$\square$

Lema 6.1.13 Si tenemos $A_P \in \mathcal{A}(P)$, $A_Q \in \mathcal{A}(Q)$ y $t \in \mathcal{T}$, entonces

- $\mathsf{nd}(A_P) < t$ ó $\mathsf{nd}(A_Q) < t \quad \Rightarrow \exists A \in \mathcal{A}(P \|_G Q) :\ A \ll A_P \sqcup_G A_Q,$
- $\mathsf{nd}(A_P) \geq t$ y $\mathsf{nd}(A_Q) \geq t \quad \Rightarrow \exists A \in \mathcal{A}(P \|_G Q) :\ A \upharpoonright t \subseteq (A_P \sqcup_G A_Q) \upharpoonright t.$

$\square$

Proposición 6.1.14 Siendo P y Q son procesos, y A_P y A_Q estados de P y Q respectivamente, entonces existe un estado $A \in \mathcal{A}(P \|_G Q)$ tal que $A \ll A_P \sqcup_G A_Q$.

$\square$

Las demostraciones de los lemas 6.1.11 y 6.1.13 son muy similares a las demostraciones de los lemas 4.2.13 y 4.2.15, mientras que las de las de las proposiciones 6.1.12, y 6.1.14 son respectivamente iguales a las de las proposiciones 4.2.14 y 4.2.16.

Proposición 6.1.15 Para cualesquiera procesos P y Q, se tiene

$$b \in \mathsf{Barb}(P \|_G Q) \quad \Rightarrow \quad \exists b_1 \in \mathsf{Barb}(P),\ b_2 \in \mathsf{Barb}(Q) \text{ y } b' \in b_1 \|_G b_2 :\ b' \ll b$$

Demostración. La demostración es muy parecida a la de la proposición 4.2.17, y puede encontrarse completa en el apéndice A.

$\square$

Proposición 6.1.16 Siendo $b_1 \in \mathsf{Barb}(P)$, $b_2 \in \mathsf{Barb}(Q)$ y $b \in b_1 \|_G b_2$, tenemos que existe $b' \in \mathsf{Barb}(P \|_G Q)$ tal que $b' \ll b$.

Demostración. La demostración es muy parecida a la de la proposición 4.2.18, se encuentra en el apéndice A.

$\square$

Teorema 6.1.17 Para cualesquiera conjuntos consistentes de barbas B_1 y B_2, se tiene

- Si $B_1 \ll \mathsf{Barb}(P)$ y $B_2 \ll \mathsf{Barb}(Q)$ entonces $\mathcal{B}_{\mathsf{con}}[\![\|_G]\!](B_1, B_2) \ll \mathsf{Barb}(P \|_G Q)$.

- Si $\mathsf{Barb}(P) \ll B_1$ y $\mathsf{Barb}(Q) \ll B_2$ entonces $\mathsf{Barb}(P \parallel_G Q) \ll \mathcal{B}_{\mathrm{con}}[\![\parallel_G]\!](B_1, B_2)$.

Demostración. La primera parte es una aplicación de la proposición 6.1.15, mientras que la segunda lo es de la proposición 6.1.16.

□

Como consecuencia de este teorema, podemos extender todos los resultados de abstracción de la semántica denotacional al operador paralelo.

6.2 Ecuaciones

Como es habitual para los operadores derivado, daremos las ecuaciones necesarias para poder *eliminarlo* de cualquier término finito. Tenemos en concreto el axioma de expansión que presentamos en la tabla 6.1.

$$
\textbf{[PAR]} \quad \left(\bigsqcap_{A\in\mathcal{A}} \Box_{at\in A} P^A_{at}\right) \parallel_G \left(\bigsqcap_{B\in\mathcal{B}} \Box_{at\in B} Q^B_{at}\right) =
$$

$$
\bigsqcap_{\substack{A\in\mathcal{A}\\ B\in\mathcal{B}}} \left(\mathtt{DIV}(t_{A,B}) \,\Box \Box_{\substack{at\in A,\ a\notin G\\ t<\mathsf{nd}(B)}} at \,;\, R1^{A,B}_{at} \,\Box \Box_{\substack{at\in B,\ a\notin G\\ t<\mathsf{nd}(A)}} at \,;\, R2^{A,B}_{at} \,\Box \Box_{\substack{at\in B\cap A\\ a\in G}} R4^{A,B}_{at}\right)
$$

$$
\begin{aligned}
\text{donde}\quad R1^{A,B}_{at} &= P^A_{at} \parallel_G \left(\Box_{b(t'-t)\in(B-t)} Q^B_{bt'}\right)\\
R2^{A,B}_{at} &= \left(\Box_{b(t'-t)\in(A-t)} Q^A_{bt'}\right) \parallel_G Q^B_{at}\\
R3^{A,B}_{at} &= P^A_{at} \parallel_G Q^B_{at}\\
t_{A,B} &= \min\{\mathsf{nd}(A), \mathsf{nd}(B)\}
\end{aligned}
$$

Tabla 6.1: Axioma de expansión del operador paralelo.

Para escribir este axioma, lo único que hemos hecho es trasladar al marco algebraico la definición de la semántica denotacional para argumentos del operador paralelo que sean de la forma

$$
P = \bigsqcap_{A\in\mathcal{A}} \Box_{at\in A} P^A_{at} \quad \text{y} \quad Q = \bigsqcap_{A\in\mathcal{B}} \Box_{at\in A} Q^A_{at}
$$

se

siendo $\mathcal{A}$ y $\mathcal{B}$ son conjuntos finitos de estados en los que cada estado $A \in \mathcal{A} \cup \mathcal{B}$ es finito. Observemos que dichos procesos no tienen porque estar en forma normal ni prenormal. La semántica denotacional de los procesos P y Q verifica:

$$\begin{array}{rcl} \mathcal{B}_{\text{con}}[\![P]\!] & = & \{A \mid A \in \mathcal{A}\} \cup \{(A \upharpoonright t)at \cdot b \mid A \in \mathcal{A},\ at \in A \ \text{ y } \ b \in \mathcal{B}_{\text{con}}[\![P^A_{at}]\!]\} \\ \mathcal{B}_{\text{con}}[\![Q]\!] & = & \{A \mid A \in \mathcal{B}\} \cup \{(A \upharpoonright t)at \cdot b \mid A \in \mathcal{B},\ at \in A \ \text{ y } \ b \in \mathcal{B}_{\text{con}}[\![Q^A_{at}]\!]\} \end{array}$$

Por lo tanto la semántica denotacional del proceso $P \parallel_G Q$ vale

$$\mathcal{B}_{\text{con}}[\![P]\!] \parallel_G \mathcal{B}_{\text{con}}[\![Q]\!] =$$

$$\bigcup_{\substack{A \in \mathcal{A} \\ B \in \mathcal{B}}} \left(\begin{array}{l} \{A \sqcup_G B\} \cup \\[2ex] \cup \left\{ ((A \sqcup_G B) \upharpoonright t)at \cdot b \;\middle|\; \begin{array}{l} at \in A,\ a \notin G, \quad t < \mathsf{nd}(B), \quad \text{y} \\ b \in \mathcal{B}_{\text{con}} \left[\!\!\left[P^A_{at} \parallel_G \left(\mathop{\Box}\limits_{b(t'-t) \in (B-t)} Q^B_{bt'} \right) \right]\!\!\right] \end{array} \right\} \\[4ex] \cup \left\{ ((A \sqcup_G B) \upharpoonright t)at \cdot b \;\middle|\; \begin{array}{l} at \in B,\ a \notin G, \quad t < \mathsf{nd}(A) \quad \text{y} \\ b \in \mathcal{B}_{\text{con}} \left[\!\!\left[\left(\mathop{\Box}\limits_{b(t'-t) \in (A-t)} P^B_{bt'} \right) \parallel_G Q^B_{at} \right]\!\!\right] \end{array} \right\} \\[4ex] \cup \left\{ ((A \sqcup_G B) \upharpoonright t)at \cdot b \;\middle|\; at \in A \cap B,\ a \in G \ \text{ y } \ b \in \mathcal{B}_{\text{con}}[\![P^A_{at} \parallel_G Q^B_{at}]\!] \right\} \end{array} \right)$$

de donde se obtiene el lado derecho del axioma de expansión.

6.3 Ocultamiento

Este es otro de los operadores característicos de un álgebra de procesos, cuyo tratamiento resulta usualmente problemático. Al aplicar este operador se *oculta* toda aparición de determinada acción visible convirtiéndola en una acción oculta τ. Formalmente tenemos una familia de operadores de ocultamiento de aridad 1:

$$\{\backslash a \mid a \in Act\}$$

También podríamos haber optado por una notación un poco más general, permitiendo ocultar al tiempo un conjunto (finito) de acciones. Sin embargo, ello no aportaría gran cosa puesto que dicha extensión puede ser simulada con el operador propuesto, pues siendo $G = \{a_1, \ldots, a_n\}$ tendremos que $(\cdots (P \setminus a_1) \cdots) \setminus a_n$ equivaldría a $P \setminus G$.

6.3.1 Semántica Operacional

De nuevo hemos de comenzar extendiendo la definición de las funciones auxiliares que intervienen en las reglas de las transiciones:

$$\begin{array}{lcl}
\mathsf{stb}(P \setminus a) & = & \mathsf{stb}(P) \\
\mathsf{Tiem}(P \setminus a, b) & = & \begin{cases} \mathsf{Tiem}(P,b) \wr t_a & \text{si} a \neq b \ \text{ y } \ t_a = \min\{t \mid t \in \mathsf{Tiem}(P,a)\} \\ \varnothing & \text{si } a = b \end{cases} \\
\mathsf{idle}(P \setminus a, A) & = & \mathsf{idle}(P, A \cup \{a\}) \\
\mathsf{Upd}(P \setminus a, t) & = & \mathsf{Upd}(P,t) \setminus a
\end{array}$$

Podemos ahora dar las reglas que definen la semántica operacional del nuevo operador. En primer lugar las transiciones vacías siguen siendo las mismas antes de aplicar el operador:

$$[\mathbf{OCUL1}] \quad \frac{P \rightarrowtail P'}{P \setminus a \rightarrowtail P' \setminus a}$$

Para el resto de las transiciones correspondientes a acciones distintas de a, tenemos que tener en cuenta que al ocultar la acción a se convierte en oculta, y por tanto es urgente:

$$[\mathbf{OCUL2}] \quad \frac{P \xrightarrow{et} P', \quad \mathsf{idle}(P, \{a\}) \geq t}{P \setminus a \xrightarrow{et} P' \setminus a} \quad e \in \mathcal{E},\ e \neq a$$

Por último tenemos que la acción a debe ser ocultada:

$$[\mathbf{OCUL4}] \quad \frac{P \xrightarrow{at} P', \quad \mathsf{idle}(P, \{a\}) \geq t,}{P \setminus a \xrightarrow{\tau t} P' \setminus a}$$

hace

6.3.2 Semántica de Pruebas

Para definir la semántica de pruebas lo único que hemos de hacer es extender la definición de convergencia débil, para ello tenemos

$$P\downarrow \quad \Rightarrow \quad P \setminus a \downarrow$$

6.3.3 Semántica Denotacional

Para definir la semántica denotacional de este operador, necesitamos primero una serie de definiciones previas:

Definición 6.3.1 Diremos que a es ocultable en una b-traza bs, y lo denotaremos por $\mathsf{ocul}(bs, a)$, cuando lo indique la siguiente definición inductiva:

- $\mathsf{ocul}(\epsilon, a)$, y en tal escribiremos $\epsilon \setminus a = \epsilon$.
- Si $\mathsf{ocul}(bs_1, a)$ y $a \notin A_1$ tendremos $\mathsf{ocul}(A_1 a_1 t_1 \cdot bs_1)$ si se verifica alguna de las condiciones siguientes:
  - $a \neq a_1$. En este caso definimos $(A_1 a_1 \cdot bs_1) \setminus a = A_1 a_1 t_1 \cdot (bs_1 \setminus a)$.
  - $a = a_1$ y $bs_1 \neq \epsilon$. En tal caso definimos $(A_1 a_1 \cdot bs_1) \setminus a = (A_1, t_1) \sqcup (bs_1 \setminus a)$.

Para facilitar la escritura y lectura de estos conceptos, cuando nos permitamos escribir la b-traza $bs \setminus a$, estaremos diciendo implícitamente que se tiene $\mathsf{ocul}(bs, a)$.

□

Observemos que las únicas b-trazas no ocultables son aquellas que *finalizan* con la ejecución de la acción que se ha ocultado.

Definición 6.3.2 Siendo B un conjunto consistente de barbas y $a \in Act$, para cada $k \in \mathbb{N}$ definimos inductivamente el conjunto de estados $\mathsf{Focul}(B, a, k)$, como sigue:

- $\mathsf{Focul}(B, a, 0) = \Big\{\{\Omega 0\}\Big\}$.
- Siendo $k > 0$ tenemos
  - Si $A \in B$ y $a \notin A$ entonces $A \in \mathsf{Focul}(B, a, k)$.
  - Si $Aat \in \mathsf{Btraz}(B)$, $a \notin A$ y $A_1 \in \mathsf{Focul}(\mathsf{Barb}(B, Aat), a, k-1)$, entonces

$$A \cup (A_1 + t) \in \mathsf{Focul}(B, a, k)$$

□

Antes de continuar conviene explicar brevemente la definición anterior. Observemos en primer lugar que si $A \in \mathsf{Focul}(B, a, k)$ existirá una b-traza bs de longitud l con $l \leq k$:

$$bs = A'_1 a t_1 \cdots A'_l a t_l \in \mathsf{Btraz}(B)$$

verificando $a \notin A'_i$. Entonces, tomando $t^i = \sum_{j=1}^{i-1} t_j$, el estado A verifica:

- Si $l < k$ ha de existir un estado A' de modo que

$$bs \cdot A' \in B, \quad a \notin A' \quad \text{y} \quad A = (A' + t^{l+1}) \bigcup_{1 \leq i \leq l} (A'_i + t^i)$$

- Si $l = k$ entonces

$$A = \left\{ \{\Omega t^{l+1}\} \right\} \bigcup_{1 \leq i \leq l} (A'_i + t^i)$$

Entonces la semántica correspondiente al operador de ocultamiento queda como sigue:

Definición 6.3.3 Siendo B un conjunto consistente de barbas y $a \in Act$, diremos que $bs \cdot A \in \mathcal{B}_{\text{con}}[\![\backslash a]\!](B)$ si y sólo si se cumple una de la siguientes posibilidades:

- $\mathsf{nd}(A) < \infty$ y para cada $k \in \mathbb{N}$ existen $l \geq k$ y una b-traza bs' verificando:

$$A \in \mathsf{Focul}(\mathsf{Barb}(B, bs'), a, l) \quad \text{y} \quad bs' \setminus a = bs$$

- $\mathsf{nd}(A) = \infty$ y para cada $t \in \mathcal{T}$ existen $l \in \mathbb{N}$, un estado A_t y una b-traza bs' verificando:

$$A_t \upharpoonright t = A \upharpoonright t, \quad A_t \in \mathsf{Focul}(\mathsf{Barb}(B, bs'), a, l) \quad \text{y} \quad bs' \setminus a = bs$$

$\square$

Antes de seguir adelante más conviene observar algunos hechos que se utilizarán en lo sucesivo en esta sección. En primer lugar si existe un estado $A_l \in \mathsf{Focul}(B, a, l)$ tal que $\mathsf{nd}(A_l) \geq t$ y $l' > l$, entonces existe también un estado $A_{l'} \in \mathsf{Focul}(B, a, l')$ de manera que $A_l \upharpoonright t = A_{l'} \upharpoonright t$. Por lo tanto, en la definición de la semántica del operador $\mathcal{B}_{\text{con}}[\![\backslash a]\!](B)$, cuando $\mathsf{nd}(A) = \infty$ podemos suponer sin ninguna pérdida alguna de generalidad que la secuencia de l's asociada a las t's es creciente. Además, si $A_l \in \mathsf{Focul}(B, a, l)$ y $t \leq \mathsf{nd}(A)$, existe un estado $A \in \mathcal{B}_{\text{con}}[\![\backslash a]\!](B)$ tal que $A_l \upharpoonright t = A \upharpoonright t$, y viceversa: si $A \in \mathcal{B}_{\text{con}}[\![\backslash a]\!](B)$ y $\mathsf{nd}(A) \geq t$, entonces existen $l \in \mathbb{N}$ y un estado $A_l \in \mathsf{Focul}(B, a, l)$ tal que $A_l \upharpoonright t = A \upharpoonright t$.

Comprobamos a continuación que la definición del operador está bien hecha, es decir, que se cumple la

con

Proposición 6.3.4 Si B es un conjunto consistente de barbas, entonces $\mathcal{B}_{\text{con}}[\![\backslash a]\!](B)$ también lo es.

Demostración. Veamos que $\mathcal{B}_{\text{con}}[\![\backslash A]\!](B)$ cumple las condiciones de la definición 4.1.1.

$\mathcal{B}_{\text{con}}[\![\backslash \mathbf{A}]\!](\mathbf{B}) \neq \varnothing$

Puesto que B es consistente existirá un estado $A_1 \in B$. Si $a \notin A_1$ tendríamos $A_1 \in \mathcal{B}_{\text{con}}[\![\backslash A]\!](B)$ con lo y habríamos concluido. Si por el contrario $a \in A_1$, tomando $t_1 = \min\{t \mid at \in A_1\}$ tendremos que existirá un estado A_2 tal que

$$(A_1 \upharpoonright t_1)at_1A_2 \in B$$

Si $a \notin A_2$ entonces

$$A = (A_1 \upharpoonright t_1) \cup (A_2 + t_1) \in \mathcal{B}_{\text{con}}[\![\backslash A]\!](B)$$

Si $a \in A_2$, repetimos el proceso que hicimos con A_1. Iterando este razonamiento tenemos dos posibilidades

- Existe una barba $(A_1 \upharpoonright t_1)a_1t_1 \cdots (A_n \upharpoonright t_n)a_nt_nA_{n+1} \in B$ tal que $a \notin A_i \upharpoonright t_i$ para $i \leq n$ y $a \notin A_{n+1}$. Entonces, tomando

$$A = (A_{n+1} + t^{n+1}) \cup \bigcup_{i \leq n} A_n + t^n \qquad \text{donde} \qquad t^i = \sum_{j=1}^{i-1} t_j$$

  se verifica $A \in \mathcal{B}_{\text{con}}[\![\backslash A]\!](B)$.

- Existe un número infinito de b-trazas $(A_1 \upharpoonright t_1)a_1t_1 \cdots (A_n \upharpoonright t_n)a_nt_n \in \mathsf{Btraz}(B)$ tales que $a \notin A_i \upharpoonright t_i$. Tomamos entonces $t^i = \sum_{j=1}^{i-1} t_j$. Si existe k de manera que para todo $l \geq k$ se verifica $t_l = 0$, tomemos el estado

$$A = \{\Omega t^k\} \cup \bigcup_{i \leq k} (A_i \upharpoonright t_i) + t^i$$

  Si por contra, para todo k existe $l \geq k$ tal que $t_l \neq 0$ tomamos el estado

$$A = \bigcup_{i \in \mathbb{N}} (A_i \upharpoonright t_i) + t^i$$

  Y es fácil ver que en cualquiera de los dos caso tenemos $A \in \mathcal{B}_{\text{con}}[\![\backslash a]\!](B)$.

Cerrado bajo prefijos

Sea $bs' = A_1a_1t_1 \cdots A_na_nt_n \in \mathsf{Btraz}(\mathcal{B}_{\text{con}}[\![\backslash A]\!](B))$, existirá entonces una b-traza $bs \in \mathsf{Btraz}(B)$ tal que y $bs \setminus a = bs'$. La b-traza bs se podrá descomponer en una serie de subtrazas $bs = bs_1 \cdot bs_2 \cdots bs_n$ de manera que cada bs_i verifique:

- $bs_i = A_{i1}at_{i1}\cdots A_{il_i}a_it_{il_i}$ con $l_i \geq 1$
- Tomando $t^{ij} = \sum_{k=1}^{j-1} t_{ij}$ se tiene

$$t_i = t_{il_i} + t^{il_i} \quad \text{y} \quad A_i = \bigcup_{j=1}^{l_i} A_{ij} + t^{ij}$$

Por lo que tenemos

$$(bs_1\cdots bs_{n-1}) \setminus a = A_1a_1t_1\cdots A_{n-1}a_{n-1}t_{n-1}, \quad \text{y} \quad A_n = \bigcup_{i\leq l_n} A_{n\,i} + t^{n\,i}$$

Puesto que B es consistente, existe un estado A^n tal que

$$bs_1\cdots bs_{n-1}A_{n1}at_{n1}\cdots A_{n\,l_n-1}at_{n\,l_n-1}A^n \in B \quad \text{y} \quad A^n \upharpoonright (t_n - t^{nl_n}) = A_{nl_n}$$

Tomamos entonces el estado

$$A' = A^n + t^{n\,l_n} \cup \bigcup_{i<l_n} A_{ni} + t^{ni}$$

y se verificará que $A_n = A' \upharpoonright t_n$. Si ahora $a \notin A'$ entonces tendríamos

$$A_1a_1t_1\cdots A_{n-1}a_{n-1}t_{n-1}A' \in \mathcal{B}_{\text{con}}[\![\setminus A]\!](B)$$

Si por el contrario $a \in A'$, teniendo en cuenta que $a \notin A_n$, y razonando como en el apartado anterior, podremos encontrar un estado A'' tal que

$$A_1a_1t_1\cdots A_{n-1}a_{n-1}t_{n-1}A'' \in \mathcal{B}_{\text{con}}[\![\setminus A]\!](B) \quad \text{y} \quad A'' \upharpoonright t_n = A' \upharpoonright t_n$$

Cerrado bajo continuaciones

Sea $bs \cdot A \in \mathcal{B}_{\text{con}}[\![\setminus A]\!](B)$ y $a't' \in A$. Existen entonces $l \in \mathbb{N}$, una b-traza bs' y un estado A' tales que

$$a't' \in A', \quad bs' \setminus a = bs, \quad A' \in \mathsf{Focul}(\mathsf{Barb}(B, bs'), a, l) \quad \text{y} \quad A' \upharpoonright t' = A \upharpoonright t'$$

En consecuencia existirá una barba

$$A_1at_1\cdots A_nat_nA_{n+1} \in \mathsf{Barb}(B, bs')$$

tal que, tomando $t^i = \sum_{j=1}^{i-1} t_j$, se verifica

$$A' \upharpoonright t' = (A_{n+1} \upharpoonright (t' - t^{n+1})) + t^{n+1} \cup \bigcup_{i\leq n} A_i + t^i$$

Puesto que B es consistente tenemos

$$A_1at_1\cdots A_nat_n(A_{n+1} \upharpoonright (t' - t^{n+1}))a'(t' - t^{n+1}) \in \mathsf{Btraz}(\mathsf{Barb}(B, bs'))$$

Y entonces se verifica $bs \cdot (A' \upharpoonright t')a't' \in \mathsf{Btraz}(\mathcal{B}_{\text{con}}[\![\setminus a]\!](B))$

esos

Temporalmente compacto

Esta condición resulta totalmente trivial, dada la definición del operador.

□

Para comprobar que el operador queda bien definido hemos de ver que el operador es congruente con respecto a la relación $\ll$. Para ello es suficiente ver la

Proposición 6.3.5 Sean B_1, B_2 conjuntos consistentes de barbas tales que $B_1 \ll B_2$. Entonces $\mathcal{B}_{\mathrm{con}}[\![\backslash a]\!](B_1) \ll \mathcal{B}_{\mathrm{con}}[\![\backslash a]\!](B_2)$.

Demostración. Cualquier barba $b_2 = bs_2 \cdot A_2 \in \mathcal{B}_{\mathrm{con}}[\![\backslash a]\!](B_2)$ se genera a partir de una b-traza bs'_2 tal que $bs'_2 \backslash a = bs_2$ y a partir de ella el estado A_2 se genera a partir de estados $A'_2 \in \mathsf{Focul}(\mathsf{Barb}(B_2, bs'_2), a, l)$.

Antes de empezar con la demostración propiamente dicha, comenzamos por estudiar lo que ocurre con dichas b-traza bs'_2 y estados A'_2. Supongamos por tanto que $\mathsf{ocul}(bs'_2, a)$ y $A'_2 \in \mathsf{Focul}(\mathsf{Barb}(B_2, bs'_2), a, l)$. Existe entonces una b-traza

$$bs' = A_1 at_1 \cdots A_m at_m \in \mathsf{Btraz}(\mathsf{Barb}(B_2, bs'_2))$$

que genera el estado A'_2, es decir:

- Si $m < l$ existirá un estado A_{m+1} tal que

$$a \notin A_{m+1}, \quad A'_2 = \bigcup_{i=1}^{m+1} A_i + t^i \quad \text{y} \quad A_1 at_1 \cdots A_m at_m A_{m+1} \in \mathsf{Barb}(B_2, bs'_2)$$

- Si $m = l$ tendremos $A'_2 = \{\Omega t^{l+1}\} \cup \bigcup_{i=1}^{l} A_i + t^i$, por lo que podemos tomar como estado A_{m+1} cualquiera verificando (puesto que B_2 es consistente existirá al menos uno tal)

$$A_1 at_1 \cdots A_m at_m A_{m+1} \in \mathsf{Barb}(B_2, bs'_2)$$

En cualquiera de los dos casos, y puesto que $B_1 \ll B_2$, existirá una barba $b'_1 \in B_1$ tal que

$$b'_1 \ll bs'_2 \cdot (A_1 at_1 \cdots A_m at_m A_{m+1})$$

En función de la forma de dicha barba podemos distinguir tres posibilidades:

1. $\mathsf{lon}(b'_1) < \mathsf{lon}(bs'_2)$. En este caso podemos descomponer la barba b'_1 en dos trozos:

$$b'_1 = bs'_1 \cdot (A^1_1 at^t_1 \cdots A^1_s at^1_s A^1_{s+1}), \quad \mathsf{ocul}(bs'_1, a) \quad \text{y} \quad a \notin A^1_i$$

Entonces podemos tomar

$$bs_1 = bs'_1 \setminus a, \quad A_1 = \bigcup_{i=1}^{s+1}\left(A_i^1 + \sum_{j=1}^{i-1} t_j^1\right) \quad \text{y} \quad b_1 = bs_1 \cdot A_1$$

y tendremos que $b_1 \in \mathcal{B}_{\text{con}}[\![\backslash a]\!](B_1)$. Además, puesto que la longitud de b'_1 es estrictamente menor que la de la barba original, tendremos directamente que $b_1 \ll b_2$.

2. $\mathsf{lon}(bs'_2) \leq \mathsf{lon}(b'_1) < \mathsf{lon}(bs'_2 \cdot (A_1 at_1 \cdots A_m at_m A_{m+1}))$. En este caso podemos descomponer la barba b_1 en dos trozos:

$$\begin{aligned} b'_1 &= bs'_1 \cdot A_1^1 at_1^t \cdots A_s^1 at_s^1 A_{s+1}^1 \\ &\text{donde} \quad bs'_1 \ll bs'_2 \quad \text{y} \\ &\qquad A_1^1 at_1^t \cdots A_s^1 at_s^1 A_{s+1}^1 \ll A_1 at_1 \cdots A_m at_m A_{m+1} \end{aligned}$$

Tomamos entonces

$$bs_1 = bs'_1 \setminus a, \quad A_1 = \bigcup_{i=1}^{s+1}\left(A_i^1 + \sum_{j=1}^{i-1} t_j^1\right) \quad \text{y} \quad b_1 = bs_1 \cdot A_1$$

y tendremos que $b_1 \in \mathcal{B}_{\text{con}}[\![\backslash a]\!](B_1)$, y al igual que antes obtenemos $b_1 \ll b_2$.

3. $\mathsf{lon}(b'_1) = \mathsf{lon}(bs'_2 \cdot (A_1 at_1 \cdots A_m at_m A_{m+1}))$. Este es el caso más complejo. Al igual que antes podemos dividir la barba b_1 en dos trozos:

$$\begin{aligned} b'_1 &= bs'_1 \cdot A_1^1 at_1^t \cdots A_m^1 at_m^1 A_{m+1}^1 \\ &\text{donde} \quad bs'_1 \ll bs'_2 \quad \text{y} \\ &\qquad A_1^1 at_1^t \cdots A_m^1 at_m^1 A_{m+1}^1 \ll A_1 at_1 \cdots A_m at_m A_{m+1} \end{aligned}$$

También como en el caso anterior tomamos

$$bs_1 = bs'_1 \setminus a, \quad \text{y} \quad A'_1 = \bigcup_{i=1}^{m+1}\left(A_i^1 + \sum_{j=1}^{i-1} t_j^1\right)$$

Ahora si $l < m$ se tiene directamente $A'_1 \in \mathsf{Focul}(\mathsf{Barb}(B_1, bs'_1), a, l)$. En el caso que $l = m$ tenemos que si en la unión que conforma el estado A'_1 sustituimos A_{m+1} por $\{\Omega 0\}$ obtenemos un estado que pertenece al conjunto $\mathsf{Focul}(\mathsf{Barb}(B_1, bs'_1), a, l)$. Por abuso de notación seguiremos denotando al mismo por A'_1. En cualquier caso tenemos $A'_1 \ll A'_2$. De momento, en este caso no podemos concluir más hechos, aunque como veremos resultará suficiente.

Volvemos ya a la demostración de la proposición. Sea $bs_2 \cdot A_2 \in \mathcal{B}_{\text{con}}[\![\backslash a]\!](B)$, distinguimos dos casos en función de la forma de A_2:

- $\mathsf{nd}(A_2) < \infty$. En este caso para cada $k \in \mathbb{N}$ existen $l_k \geq k$ y una b-traza bs_{2k} verificando

$$bs_2 = bs_{2k} \setminus a \quad \text{y} \quad A \in \mathsf{Focul}(\mathsf{Barb}(B_2, bs_{2k}), a, l_k)$$

Entonces teniendo en cuenta las propiedades de los elementos que concurren que hemos visto más arriba, tenemos dos posibilidades:

1. Existe l_k de manera que encontramos una barba $b_1 \in \mathcal{B}_{\mathrm{con}}[\![\setminus a]\!](B_1)$ tal que $b_1 \ll bs_2 \cdot A_2$, con lo que hemos concluido.
2. Para cada l_k encontramos una b-traza bs_{l_k} y un estado A_{1l_k} tales que

$$bs_{1l_k} \ll bs_{2k}, \quad A_{1l_k} \in \mathsf{Focul}(\mathsf{Barb}(B_1, b^1_{l_k}), B_1, l_k) \quad \text{y} \quad A_{1l_k} \ll A_2$$

Solamente hay un número finito de estados A' tales que $A' \ll A_2$, por lo que podemos suponer que todos los estados A_{1l_k} que han aparecido son iguales a uno dado que denotaremos por A_1. Algo similar ocurre con las b-trazas; como tenemos que $bs_{1l_k} \ll bs_{2k}$ llegamos a $bs_{1l_k} \setminus a \ll bs_2$, y puesto que sólo puede haber un número finito de b-trazas bs' tales que $bs' \ll bs_2$, podemos suponer que las barbas $bs_{1l_k} \setminus a$ son iguales a una dada bs_1. Entonces se verifican

$$bs_1 \cdot A_1 \in \mathcal{B}_{\mathrm{con}}[\![\setminus a]\!](B_1) \quad \text{y} \quad bs_1 \cdot A \ll bs_2 \cdot A_2$$

- $\mathsf{nd}(A_2) = \infty$. Entonces, para cada $t \in \mathcal{T}$ existen $l_t \in \mathbb{N}$, una b-traza bs_{2t} y un estado A_{2l_t} verificando

$$bs_2 = bs_{2t} \setminus a, \quad A_{2t} \in \mathsf{Focul}(\mathsf{Barb}(B_2, bs_{2t}), a, l_t) \quad \text{y} \quad A_{2t} \upharpoonright t = A_2 \upharpoonright t$$

Entonces, en función de los hechos vistos más arriba, tenemos dos posibilidades:

1. Existe l_t de manera que encontramos una barba $b_1 \in \mathcal{B}_{\mathrm{con}}[\![\setminus a]\!](B_1)$ tal que $b_1 \ll bs_{2t} \cdot A_{2t}$, con lo que hemos concluido.
2. Para cada l_t encontramos una b-traza bs_{1t} y un estado A_{1t} tales que

$$bs_{1t} \ll bs_{2t}, \quad A^1_{1t} \in \mathsf{Focul}(\mathsf{Barb}(B_1, b_{1t}), B_1, l_t) \quad \text{y} \quad A_{1t} \ll A_{2t}$$

Para cada t tenemos que sólo puede haber un número finito de estados A' tales que $A' \upharpoonright t \subseteq A_2 \upharpoonright t$, por lo que podemos suponer que $A_{1t} \upharpoonright t = A_{1t'} \upharpoonright t$ para $t' \geq t$. Algo similar ocurre con las b-trazas, como $bs_{1t} \ll bs_{2t}$ tenemos $bs_{1t} \setminus a \ll bs_2$, puesto que sólo puede haber un número finito de b-trazas bs'

tales que $bs' \ll bs_2$, podemos suponer que las barbas $bs_{1t} \setminus a$ son iguales a una dada bs_1. Tomando entonces

$$A_1 = \bigcup_{t \in \mathcal{T}} A_{1t} \upharpoonright t \quad \text{y} \quad bs_1 = bs_{1t} \setminus a$$

se verifica $bs_1 \cdot A_1 \in \mathcal{B}_{\text{con}}[\![\setminus a]\!](B_1)$ y $bs_1 \cdot A_2 \ll bs_1 \cdot A_2$.

□

6.3.3.1 Continuidad

Para que la semántica denotacional quede bien definida, es necesario también que este nuevo operador sea monótono y continuo con respecto a la relación $\prec$. Comencemos por la monotonía.

Proposición 6.3.6 Si $B_1 \prec B_2$, entonces $\mathcal{B}_{\text{con}}[\![\setminus a]\!](B_1) \prec \mathcal{B}_{\text{con}}[\![\setminus a]\!](B_2)$.

Demostración. Para demostrar que $\mathcal{B}_{\text{con}}[\![\setminus a]\!](B_1) \prec \mathcal{B}_{\text{con}}[\![\setminus a]\!](B_2)$ hemos de probar:

1. Para cada $bs_2 \cdot A_2 \in \mathcal{B}_{\text{con}}[\![\setminus a]\!](B_2)$ existe $b_1 \cdot A_1 \in \mathcal{B}_{\text{con}}[\![\setminus a]\!](B_1)$. Según sea el estado A_2 tenemos dos posibilidades:

   - $\mathsf{nd}(A_2) < \infty$. Por la definición tenemos que para cada $k \in \mathbb{N}$ existen $l_k \geq k$ y una b-traza bs_{2k} tales que

     $$bs_2 = bs_{2k} \setminus a \quad \text{y} \quad A \in \mathsf{Focul}(\mathsf{Barb}(B_2, bs_{2k}), a, l_k)$$

     Razonando como en la proposición 6.3.5 tenemos dos posibilidades:

     – Existe una barba $b_1 \in \mathcal{B}_{\text{con}}[\![\setminus a]\!](B_1)$ tal que $b_1 \prec b_2$, y entonces ya hemos acabado.

     – Para l_k, existe $A_{1k} \in \mathsf{Focul}(\mathsf{Barb}(B_1, bs_{2k}), a, l_k)$ tal que $A_{1k} \prec A$. Puesto que $\mathsf{nd}(A) < \infty$ podemos suponer que todos los estados A_{1k} son iguales a uno dado que denotamos por A_1. Por tanto tendremos que la barba $bs_2 \cdot A_1$ pertenece al conjunto $\mathcal{B}_{\text{con}}[\![\setminus a]\!](B_1)$, con lo cual hemos concluido.

   - $\mathsf{nd}(A_2) = \infty$. En este caso para cada $t \in \mathcal{T}$ existen $l_t \in \mathbb{N}$, una b-traza bs_{2t} y un estado A_{2t} tales que

     $$bs_2 = bs_{2k} \setminus a, \quad A_{2t} \in \mathsf{Focul}(\mathsf{Barb}(B_2, bs_{2t}), a, l_t) \quad \text{y} \quad A \upharpoonright t = A_{2t} \upharpoonright t$$

     Tenemos las dos mismas posibilidades que en el caso anterior:

---

y

- Existe una barba $b_1 \in \mathcal{B}_{\text{con}}[\![\backslash a]\!](B_1)$ tal que $b_1 \prec b_2$, y entonces ya hemos acabado.
- Para l_t, existe $A_{1t} \in \mathsf{Focul}(\mathsf{Barb}(B_1, bs_{2t}), a, l_t)$ tal que $A_{1t} \prec A_t$. Según sean los estados A_{1t} tenemos dos posibilidades:
  * Existe $t_0 \in \mathcal{T}$ tal que $\mathsf{nd}(A_{1t}) \leq t_0$ para todo $t \in \mathcal{T}$. Podemos entonces suponer que la sucesión l_t es creciente, es decir, $l_t < l_{t'}$ para $t < t'$. Por otro lado podemos suponer que todos los estados A_{1t} son iguales a uno dado A_1. Con lo que tendremos $bs_2 \cdot A_1 \in \mathcal{B}_{\text{con}}[\![\backslash a]\!](B_1)$ y $A_1 \prec A_2$.
  * La sucesión de los tiempos de $\mathsf{nd}(A_{1t})$ no está acotada. En tal caso podemos suponer que $\mathsf{nd}(A_{1t}) \geq t$. Entonces tendremos

$$A_1 = \bigcup_{t \in \mathcal{T}} A_{1t} \upharpoonright t$$

  y se verificará que $bs_2 \cdot A_1 \in \mathcal{B}_{\text{con}}[\![\backslash a]\!](B_1)$ y $A_1 \prec A_2$.

2. Para cada $bs \cdot A \in\in \mathcal{B}_{\text{con}}[\![\backslash a]\!](B_1)$, hemos de encontrar un estado A' tal que $A \prec A$ y $bs \cdot A \in \mathcal{B}_{\text{con}}[\![\backslash a]\!](B_1)$. Ello se sigue de los siguientes hechos:
   - Si $bs' \in \mathsf{Btraz}(B_1)$, como $B_1 \prec B_2$ se tiene $bs' \in \mathsf{Btraz}(B_2)$.
   - Para cada $l \in \mathbb{N}$ y cada estado $A_1 \in \mathsf{Focul}(\mathsf{Barb}(B_1, bs'), a, l)$, existe un estado $A_2 \in \mathsf{Focul}(\mathsf{Barb}(B_1, bs'), a, l)$ tal que $A_1 \prec A_2$.

$\square$

Una vez visto que el operador es monótono, podremos abordar la demostración de la continuidad.

Proposición 6.3.7 Dada una secuencia no decreciente $\mathcal{B} = \{B_k \mid k \in \mathbb{N}\}$, se tiene

$$\mathcal{B}_{\text{con}}[\![\backslash a]\!](\mathsf{lub}(\mathcal{B})) \sim \mathsf{lub}(\{\mathcal{B}_{\text{con}}[\![\backslash a]\!](B_k) \mid k \in \mathbb{N}\})$$

Demostración. Puesto que $B_i \prec \mathsf{lub}(\mathcal{B})$, por la monotonía tenemos

$$\mathcal{B}_{\text{con}}[\![\backslash a]\!](B_i) \prec \mathcal{B}_{\text{con}}[\![\backslash a]\!](\mathsf{lub}(\mathcal{B}))$$

Con lo que obtenemos $\mathsf{lub}\{\mathcal{B}_{\text{con}}[\![\backslash a]\!](B_k) \mid k \in \mathbb{N}\} \prec \mathcal{B}_{\text{con}}[\![\backslash a]\!](\mathsf{lub}(\mathcal{B}))$.

Para demostrar la otra desigualdad consideramos en primer lugar

$$b = bs \cdot A \in \mathsf{lub}\{\mathcal{B}_{\text{con}}[\![\backslash a]\!](B_k) \mid k \in \mathbb{N}\}$$

Debemos encontrar $b' \in \mathcal{B}_{\text{con}}[\![\backslash a]\!](\mathsf{lub}(\mathcal{B}))$ tal que $b' \prec b$. Para ello distinguimos los siguientes en función de la forma de A:

- $\mathsf{nd}(A) = \infty$. Para cada $t \in \mathcal{T}$ existen $l_t \in \mathbb{N}$ y un estado A_t tales que

$$bs \cdot A_t \in \mathcal{B}_{\text{con}}[\![\backslash a]\!](B_{l_t}) \quad \text{y} \quad A_t \upharpoonright t = A \upharpoonright t$$

Existen entonces una b-traza $bs' \in \mathsf{Btraz}(B_{l_t})$ y un estado A'_t tales que

$$bs = bs' \setminus a, \quad A'_t \in \mathsf{Focul}(\mathsf{Barb}(B_{l_t}, bs'), a, l_t) \quad \text{y} \quad A'_t \upharpoonright t = A_t \upharpoonright t = A \upharpoonright t$$

Puesto que $B_{l_t} \prec \mathsf{lub}(\mathcal{B})$, podemos asegurar que existe un estado A''_t tal que

$$A''_t \in \mathsf{Focul}(\mathsf{Barb}(\mathsf{lub}(\mathcal{B}), bs'), a, l_t) \quad \text{y} \quad A''_t \upharpoonright t = A'_t \upharpoonright t = A \upharpoonright t$$

Por tanto $bs \cdot A \in \mathcal{B}_{\text{con}}[\![\backslash a]\!](\mathsf{lub}(\mathcal{B}))$.

- $\mathsf{nd}(A) < \infty$. Para cada $k \in \mathbb{N}$ existe $l \geq k$ tal que

$$bs \cdot A \in \mathcal{B}_{\text{con}}[\![\backslash a]\!](B_l)$$

Entonces para cada $k' \in \mathbb{N}$ existen $l' \geq k'$ y una b-traza $bs_{l'} \in \mathsf{Btraz}(B_l)$ tales que

$$bs = bs_{l'} \setminus a, \quad A \in \mathsf{Focul}(\mathsf{Barb}(B_l, bs_{l'}), a, l')$$

Podemos distinguir los siguientes casos:

  - Para cada l existe algún l' de manera que la b-traza que genera el estado A en $\mathsf{Focul}(\mathsf{Barb}(B_l, bs_{l'}), a, l')$ tiene longitud mayor que l'. Entonces tenemos

$$A \in \mathsf{Focul}(\mathsf{Barb}(\mathsf{lub}(\mathcal{B}), bs_{l'}), a, l')$$

  - En caso contrario tenemos que para cada l podemos encontrar una barba

$$b = bs_{1l} \cdot bs_{2l} \cdot A_l \in B_l$$

de manera que $bs_{1l} \setminus a = bs$, y la barba $bs_{2l} \cdot A_l$ genera el estado A. Si el número de barbas de esta forma es finito, habrá alguna de ellas que se repita infinitamente que denotaremos por b, y por entonces $b \in \mathsf{lub}(\mathcal{B})$, por lo que tendremos

$$bs \cdot A \in \mathcal{B}_{\text{con}}[\![\backslash a]\!](\mathsf{lub}(\mathcal{B}))$$

Por el contrario, si el número de barbas distintas es infinito, tendremos que las longitudes de las mismas no puede estar acotadas. En tal caso podremos encontrar una secuencia de barbas que contengan un prefijo común de la forma

$$bs_0 \cdot A_1 at_1 \cdots A_n at_n \underbrace{\varnothing a0 \varnothing a0 \cdots \varnothing a0}_{h \text{ veces}}$$

tres

de modo que $\mathsf{ocul}(bs_0, a)$, y la parte final es arbitrariamente grande. Tomamos entonces $bs' = bs_0 \setminus a$ y el estado

$$A' = \{\Omega t^{n+1}\} \cup \bigcup_{i \leq n} A_i + t^i \qquad \text{donde} \qquad t^i = \sum_{j=1}^{i-1} t_j$$

Es fácil comprobar entonces que $bs' \cdot A' \in \mathcal{B}_{\mathrm{con}}[\![\setminus a]\!](\mathsf{lub}(\mathcal{B}))$ y $bs' \cdot A' \prec bs \cdot A$.

Tomemos ahora $b = bs \cdot A \in \mathcal{B}_{\mathrm{con}}[\![\setminus a]\!](\mathsf{lub}(\mathcal{B}))$. Distinguimos dos casos dependiendo de la forma de A:

- $\mathsf{nd}(A) = \infty$. En tal caso para cada $t \in \mathcal{T}$ existen $l_t \in \mathbb{N}$, una b-traza bs_t y un estado A_t tales que

$$bs_t \setminus a = bs, \quad A_t \in \mathsf{Focul}(\mathsf{Barb}(\mathsf{lub}(\mathcal{B}), bs_t), a, l_t) \quad \text{y} \quad A \upharpoonright t = A_t \upharpoonright t$$

  Existen entonces $m_t \in \mathbb{N}$ y un estado A'_t tales que

$$bs_t \in \mathsf{Btraz}(B_{m_t}), \quad A'_t \in \mathsf{Focul}(\mathsf{Barb}(B_{m_t}, bs_t), a, l_t) \quad \text{y} \quad A'_t \upharpoonright t = A_t \upharpoonright t$$

  y por tanto existe un estado A''_t tal que

$$bs \cdot A''_t \in \mathcal{B}_{\mathrm{con}}[\![\setminus a]\!](B_{m_t}) \quad \text{y} \quad A''_t \upharpoonright t = A \upharpoonright t$$

  Puesto que lo anterior lo hemos hecho para cada $t \in \mathcal{T}$, existirá un estado A' tal que

$$bs \cdot A' \in \mathsf{lub}\{\mathcal{B}_{\mathrm{con}}[\![\setminus a]\!](B_i) \mid i \in \mathbb{N}\} \quad \text{y} \quad \forall t \in \mathcal{T}\ A' \upharpoonright t = A \upharpoonright t$$

  lo cual implica que $bs \cdot A \in \mathsf{lub}\{\mathcal{B}_{\mathrm{con}}[\![\setminus a]\!](B_i) \mid i \in \mathbb{N}\}$.

- $\mathsf{nd}(A) < \infty$. En este caso, para cada $k \in \mathbb{N}$ existen $l \geq k$ y un b-traza bs_k tales que

$$bs_k \setminus a = bs, \quad A \in \mathsf{Focul}(\mathsf{Barb}(\mathsf{lub}(\mathcal{B}), bs_k), a, l)$$

  Entonces para cada $k' \in \mathbb{N}$ existe $l' \geq k'$ tal que

$$bs_k \in \mathsf{Btraz}(B_{l'}) \quad \text{y} \quad A \in \mathsf{Focul}(\mathsf{Barb}(B_{l'}, bs_k), a, l)$$

  Por tanto $bs \cdot A \in \mathcal{B}_{\mathrm{con}}[\![\setminus a]\!](B_{l'})$, y en consecuencia tenemos

$$bs \cdot A \in \mathsf{lub}\{\mathcal{B}_{\mathrm{con}}[\![\setminus a]\!](B_i) \mid i \in \mathbb{N}\}$$

  lo que implica el resultado buscado.

$\square$

6.3.3.2 Abstracción

Probaremos en esta sección la relación entre la definición denotacional del operador de ocultamiento y su definición operacional.

Lema 6.3.8 Si $A \in \mathcal{A}(P)$ es tal que $a \notin A$, entonces $A \in \mathcal{A}(P \setminus a)$.

Demostración. Simplemente hay que tener en cuenta que la computación del proceso P que genera el estado A

$$P = P_1 \rightarrowtail^* P_1' \xrightarrow{\tau t_1} \cdots$$

también es posible en $P \setminus a$ puesto que $a \notin \mathsf{TA}(P_i')$. Puesto que la acción no está disponible en ninguno de los procesos P_i' que aparecen, tendremos $\mathsf{TA}(P_i') \rceil t_i = \mathsf{TA}(P_i' \setminus a) \rceil t_i$.

□

Proposición 6.3.9 Sea P un proceso y $A \in \mathsf{Focul}(\mathsf{Barb}(P), a, k)$, entonces

- $\mathsf{nd}(A) < \infty \quad \Rightarrow \quad \exists l \geq k : \ A \in \mathcal{A}_l(P \setminus a)$.
- $\mathsf{nd}(A) = \infty \quad \Rightarrow \quad A \in \mathcal{A}(P \setminus a)$.

Demostración. Puesto que $A \in \mathsf{Focul}(\mathsf{Barb}(P), a, k)$, existe una b-traza

$$bs = A_1 a t_1 \cdots A_l a t_l \in \mathsf{Btraz}(P)$$

de longitud $l \leq k$, tal que

$$A = \bigcup_{i=1}^{l+1} A_i + t^i \qquad \text{donde} \qquad t^i = \sum_{j=1}^{i-1} t_j$$

donde el estado A_{l+1} se ha obteniendo, una en función de la relación entre l y k como sigue:

- Si $l < k$ entonces $bs \cdot A_{l+1} \in P$ y $a \notin A$.
- Si $l = k$ entonces $A_{k+1} = \{\Omega 0\}$.

Puesto que $bs \in \mathsf{Btraz}(P)$ existirá un proceso P' tal que $P \overset{bs}{\Longrightarrow} P'$, con $A_{l+1} \in \mathcal{A}(P')$ en caso que $l < k$. Haremos la demostración por inducción sobre la longitud de la deducción de $P \overset{bs}{\Longrightarrow} P'$.

- Si la longitud de la deducción es uno, tenemos que $P \overset{\epsilon}{\Longrightarrow} P$. Si $k = 0$ entonces $A = \{\Omega 0\}$, de modo que tomando $l = 0$ se verifica $\{\Omega 0\} \in \mathcal{A}_0(P)$. Si $k \geq 1$ tendremos que $A \in \mathcal{A}(P)$ y $a \notin A$. Por la proposición anterior tenemos que $A \in \mathcal{A}(P \setminus a)$, y

---

minutos?

por tanto en el caso de que $\mathsf{nd}(A) = \infty$ ya habríamos acabado. Supongamos ahora que $\mathsf{nd}(A) < \infty$. Por la definición de $\mathcal{A}(P)$ existe una secuencia de estados

$$A_0 \prec A_1 \prec A_2 \prec \cdots A_n \prec A_{n+1} \prec \cdots$$

de manera que $A_i \in \mathcal{A}_i(P)$ y $A = \mathsf{lub}\{A_k \mid k \in \mathbb{N}\}$. Puesto que $\mathsf{nd}(A) < \infty$ existe $m \in \mathbb{N}$ tal que $A_m = A_{m'}$ para $m' \geq m$. Basta entonces tomar $l = \max(k, m)$ para concluir en el caso de que $\mathsf{nd}(A) < \infty$.

- Si la longitud de la deducción es mayor que uno, tenemos los siguientes casos:
  - $P \rightarrowtail P'' \stackrel{bs'}{\Longrightarrow} P'$. Este caso es totalmente trivial, pues podemos aplica la hipótesis de inducción ya que se tiene

    $$P \setminus a \rightarrowtail P'' \setminus a$$

  - $P \xrightarrow{\tau t} P'' \stackrel{bs'}{\Longrightarrow} P'$, $bs' \neq \epsilon$. En este caso tenemos

    $$bs' = A_1' a(t_1 - t) A_2 a t_2 \cdots A_l a t_l \quad \text{y} \quad A_1 = (\mathsf{TA}(P) \upharpoonright t) \cup (A_1' + t)$$

    Tenemos entonces que el estado $A'' = A - t$ verifica

    $$A'' \in \mathsf{Focul}(\mathsf{Barb}(P''), a, k)$$

    Puesto que $a \notin A_1$ tenemos que $a \notin \mathsf{TA}(P) \upharpoonright t$ y por tanto se verificará $\mathsf{idle}(P, \{a\}) \geq t$. Por tanto la transición

    $$P \setminus a \xrightarrow{\tau t} P'' \setminus a$$

    es posible. Con lo que tenemos el resultado por aplicación de la hipótesis de inducción.
  - $P \xrightarrow{at} P'' \stackrel{bs'}{\Longrightarrow} P'$. En este caso tenemos

    $$bs' = A_2 a t_2 \cdots A_l a t_l \quad \text{y} \quad A_1 = (\mathsf{TA}(P) \upharpoonright t)$$

    de modo que

    $$A'' = A - t \in \mathsf{Focul}(\mathsf{Barb}(P''), a, k-1)$$

    Puesto que $a \notin A_1$ tenemos que $a \notin \mathsf{TA}(P) \upharpoonright t$ y por tanto $\mathsf{idle}(P, \{a\}) \geq t$. Por lo tanto la transición

    $$P \setminus a \xrightarrow{\tau t} P'' \setminus a$$

    es posible. Con lo que tenemos el resultado por aplicación de la hipótesis de inducción.

□

Proposición 6.3.10 Siendo A un estado tal que $A \in \mathcal{A}(P \setminus a)$, se tiene

- Si $\mathsf{nd}(A) = \infty$, para cada $t \in \mathcal{T}$ existen $l_t \in \mathbb{N}$ y $A_t \in \mathsf{Focul}(\mathsf{Barb}(P), a, l_t)$ tales que $A_t \upharpoonright t = A \upharpoonright t$.

- Si $\mathsf{nd}(A) < \infty$, para cada $k \in \mathbb{N}$ existe $l \geq k \in \mathbb{N}$ tal que $A \in \mathsf{Focul}(\mathsf{Barb}(P), a, l)$.

Demostración. Si $A \in \mathcal{A}(P\setminus a)$ existe una computación de $P\setminus a$ que genera el dicho estado. Puesto que $P \setminus a \xrightarrow{it} P' \setminus a$ si y sólo si $P \xrightarrow{it} P'$ o bien $P \xrightarrow{at} P'$, la computación de $P \setminus a$ será de la forma (tomando $P_{11} = P$)

$$\begin{array}{l}
P_{11} \setminus a \rightarrowtail^* P'_{11} \setminus a \xrightarrow{\tau t_{11}} P_{12} \setminus a \rightarrowtail^* \cdots P_{1k_1} \setminus a \rightarrowtail^* P'_{1n_1} \setminus a \xrightarrow{\tau t_{1n_1}} P_{21} \setminus a \cdots \\
\qquad\quad \vdots \qquad\qquad\qquad\qquad \ddots \qquad\qquad\qquad \vdots \\
P_{k1} \setminus a \rightarrowtail^* P'_{k1} \setminus a \xrightarrow{\tau t_{k1}} P_{k2} \setminus a \rightarrowtail^* \cdots P_{kn_k} \setminus a \rightarrowtail^* P'_{kn_k} \setminus a \xrightarrow{\tau t_{kn_k}} P_{(k+1)1} \setminus a \\
\qquad\quad \cdots \qquad\qquad\qquad\qquad \cdots \qquad\qquad\qquad \cdots
\end{array}$$

donde cada transición $P'_{ki} \setminus a \xrightarrow{\tau t_{ki}} P_{k(i+1)} \setminus a$ con $i < n_k$ proviene de una transición $P'_{ki} \xrightarrow{\tau t_{ki}} P_{k(i+1)}$, y cada transición $P'_{kn_k} \setminus a \xrightarrow{\tau t_{kn_k}} P_{(k+1)1} \setminus a$ proviene de una transición del tipo $P'_{kn_k} \xrightarrow{at_{kn_k}} P_{(k+1)1}$. Entonces hay tres posibilidades:

1. La computación es finita. Entonces existe $m \in \mathbb{N}$ de modo que la misma termina con P'_{mn_m}, verificándose:

   - Para $l < m$ se tienen las transiciones:

   $$\begin{array}{lll}
   P'_{li} \xrightarrow{\tau t_{li}} P_{l(i+1)} & \text{para} \quad 1 \leq i < n_l & \text{y} \\
   P'_{ln_l} \xrightarrow{at_{ln_l}} P_{(l+1)1} & &
   \end{array}$$

   - Para $l = m$ se verifica:

   $$\begin{array}{lll}
   P'_{mi} \xrightarrow{\tau t_{m(i+1)}} P_{m(i+1)} & \text{para} \quad 1 \leq i < n_m & \text{y} \\
   P_{mn_m} \Downarrow \quad \Rightarrow & \big(\mathsf{stb}(P'_{mn_m}) \ \wedge \ \mathsf{idle}(P'_{mn_m}, \{a\}) = \infty\big) &
   \end{array}$$

2. La computación es infinita, pero proviene de una computación que ejecuta una cantidad infinita de veces la acción a. Entonces existe $m \in \mathbb{N}$ tal que

$$\begin{array}{l}
P'_{li} \xrightarrow{\tau t_{li}} P_{l(i+1)} \quad 1 \leq i < n_l, \quad P'_{ln_l} \xrightarrow{at_{ln_l}} P_{(l+1)1} \quad \text{y} \\
\forall i \in \mathbb{N} \quad P'_{mi} \xrightarrow{\tau t_{mi}} P_{m(i+1)}
\end{array}$$

-Se

3. La computación es infinita y proviene de una computación que ejecuta una cantidad infinita de veces la acción a. Entonces

$$P'_{li} \xrightarrow{\tau t_{li}} P_{l(i+1)} \quad 1 \leq i < n_l \quad \text{y} \quad P'_{ln_l} \xrightarrow{at_{ln_l}} P_{(l+1)1}$$

Ahora Para cada par de enteros $i, j \in \mathbb{N}$ para los que ello tenga sentido tomamos

$$t^{ij} = \sum_{k=1}^{j-1} t_{ik}, \quad t_i = t^{in_i} + t_{in_i}, \quad t^i = \sum_{j=1}^{i-1} t_j \quad \text{y} \quad A_i = \bigcup_{j=1}^{n_i-1} \left((\mathsf{TA}(P'_{ij}) \upharpoonright t_{ij}) + t^{ij}\right)$$

Puesto que la computación es posible tenemos $a \notin \mathsf{TA}(P'_{ij}) \upharpoonright t_{ij}$, por lo que $a \notin A_i$.

En el primer caso, si la computación es finita, Tenemos ahora dos posibilidades en función del último proceso P_{mm_n}:

- $P_{mn_m} \setminus a \Downarrow$ entonces $P'_{mn_m} \Downarrow$, puesto que $P'_{mn_m} \setminus a \not\rightarrowtail$ y $P'_{mn_m} \setminus a \xrightarrow{\tau t}\!\!\!\!\!/\;$ tenemos que $P'_{mn_m} \not\rightarrowtail$ y $P'_{mn_m} \xrightarrow{\tau t}\!\!\!\!\!/\;$, por lo que tomando

$$A_m = \bigcup_{j=1}^{l} \left((\mathsf{TA}(P'_{mj}) \upharpoonright t_{mj}) + t^{mj}\right)$$

tenemos $A_m \in \mathcal{A}(P_{m1})$ y $a \notin A_m$. Entonces se verifica

$$A_1 a t_1 \cdots A_{m-1} a t_{m-1} A_m \in \mathsf{Barb}(P)$$

Puesto que la computación es la que generaba el estado A, tenemos que

$$A = \bigcup_{i \leq m} A_i + t^i$$

Por otro lado tenemos que $A \in \mathsf{Focul}(\mathsf{Barb}(P), a, k)$ para todo $k > m$, entonces, y puesto que en este caso tenemos $\mathsf{nd}(A) = \infty$, para obtener el resultado basta tomar para cada $t \in \mathcal{T}$ cualquier $l_t \geq k$.

- Supongamos ahora que $P_{mn_m} \setminus a \Uparrow$. En tal tomamos el estado

$$A_m = \{\Omega t^{ml}\} \cup \bigcup_{j<l} \left((\mathsf{TA}(P'_{mj}) \upharpoonright t_{mj}) + t^{mj}\right)$$

verificándose $A = \bigcup_{i \leq m} A_i + t^i$. Tenemos entonces dos posibilidades

  - $P_{mn_m} \Uparrow$. En este caso tenemos

$$A_1 a t_1 \cdots A_{m-1} a t_{m-1} A_m \in \mathsf{Barb}(P)$$

  con lo que $A \in \mathsf{Focul}(\mathsf{Barb}(P), a, l)$ para $l \geq m$.

- $P_{mn_m} \Downarrow$. En este caso, tomando $Q_1 = P_{mn_m}$, ha de existir una computación

$$Q_1 \mapsto_1 Q_2 \cdots Q_{k-1} \mapsto_k Q_k \cdots$$

siendo $\mapsto_l \in \{\xrightarrow{a0}, \xrightarrow{\tau 0}, \rightarrowtail\}$. Esta computación puede ser infinita, o bien conducirnos a un proceso Q_k tal que $Q_k \Uparrow$. Por lo que en este caso, o bien tenemos una secuencia infinita de b-trazas

$$bs_h = A_1 a t_1 \cdots A_{m-1} a t_{m-1} (A_m \upharpoonright t^{ml}) \underbrace{a0 \cdots \varnothing a0}_{h \text{ veces}} \in \mathsf{Btraz}(P)$$

o bien una barba

$$b = A_1 a t_1 \cdots A_{m-1} a t_{m-1} (A_m \upharpoonright t^{ml}) a0 \cdots \varnothing a0 \{\Omega 0\}$$

En cualquier caso tenemos $A \in \mathsf{Focul}(\mathsf{Barb}(P), a, l)$ para $l \geq m$.

Puesto que en cualquier caso tenemos que $A \in \mathsf{Focul}(\mathsf{Barb}(P), a, l)$ para $k \geq m$, para obtener el resultado buscado, teniendo en cuenta $\mathsf{nd}(A) < \infty$, basta tomar para cada $k \in \mathbb{N}$, $l = \min(k, m)$.

En el segundo caso tomamos el estado

$$A_m = \bigcup_{i \in \mathbb{N}} \left((\mathsf{TA}(P'_{mi}) \upharpoonright t_{mi}) + t^{mi} \right)$$

Se verifica que $a \notin A_m$ y $A = (A_m + t^m) \cup \bigcup_{i<m} (A_i + t^i)$. Por otro lado tenemos

$$A_1 a t_1 A_2 a t_2 \cdots A_{m-1} a t_{m-1} A_m \in \mathsf{Barb}(P)$$

Puesto que $a \notin A_m$, tenemos que $A \in \mathsf{Focul}(\mathsf{Barb}(P), a, k)$ para todo $k > m$, y por tanto para obtener en este caso el resultado, teniendo en cuenta $\mathsf{nd}(A) = \infty$, basta tomar para cada $t \in \mathcal{T}$ como l_t cualquier número mayor que m.
En el último caso se verifica

$$A = \bigcup_{i \in \mathbb{N}} A_i + t^i$$

Por otro lado, para cada $i \in \mathbb{N}$ tenemos

$$A_1 a t_1 \cdots A_2 a t_2 \cdots A_i a t_i \in \mathsf{Btraz}(P)$$

Por lo tanto se verifica

$$A^i = (A \upharpoonright t^i) \cup \{\Omega t^i\} \in \mathsf{Focul}(\mathsf{Barb}(P), a, i)$$

hace

Observemos que en este caso se tiene que para cada $t \in \mathcal{T}$ existe $t_i \geq t$, por lo tanto se verificará $A \upharpoonright t = A^i \upharpoonright t$. Entonces, para obtener el resultado, basta tomar l_t cualquier valor i de manera que $t^i \geq t$.

□

Proposición 6.3.11 Siempre que $P \setminus a \xRightarrow{bs} Q \setminus a$ existirá una b-traza bs' tal que $\mathsf{ocul}(bs', a)$, $bs = bs' \setminus a$ y $P \xRightarrow{bs'} Q$.

Demostración. Haremos la demostración por inducción sobre la longitud de la deducción de

$$P \setminus a \xRightarrow{bs} Q \setminus a$$

- Si la longitud de la deducción es uno, tenemos que

$$P \setminus a \xRightarrow{\epsilon} P \setminus a \quad \Rightarrow \quad P \xRightarrow{\epsilon} P$$

- Si la longitud de la deducción es mayor que uno, tenemos los siguientes casos:
  - $P \setminus a \rightarrowtail P_1 \setminus a$ y $P_1 \setminus a \xRightarrow{bs} Q \setminus a$. Por hipótesis de inducción tenemos

    $$P_1 \setminus a \xRightarrow{bs \setminus a} Q \setminus a$$

    La demostración acaba aquí puesto que $P \setminus a \rightarrowtail P_1 \setminus a$.
  - $P \setminus a \xrightarrow{a't} P_1 \setminus a$ y $P_1 \setminus a \xRightarrow{bs_1} Q \setminus a$. En este caso tenemos

    $$bs = (\mathsf{TA}(P) \upharpoonright t)a't \cdot bs_1$$

    Puesto que $P \setminus a \xrightarrow{a't} Q \setminus a$ tenemos que $a \notin \mathsf{TA}(P) \upharpoonright t$ y

    $$P \xrightarrow{a't} P_1$$

    Por hipótesis de inducción existe bs'_1 tal que

    $$\mathsf{ocul}(bs'_1, a), \quad bs_1 = bs_1 \setminus a \quad \text{y} \quad P_1 \setminus a \xRightarrow{bs'_1} Q \setminus a$$

    Entonces la b-traza $bs' = (\mathsf{TA}(P) \upharpoonright t)a't \cdot bs'_1$ verifica las propiedades requeridas.
  - $P \setminus a \xrightarrow{\tau t} P_1 \setminus a$, $P_1 \setminus a \xRightarrow{bs_1} Q \setminus a$ y $bs_1 \neq \epsilon$. En este caso tenemos

    $$bs = (\mathsf{TA}(P) \upharpoonright t) \sqcup bs_1$$

    Puesto que $P \setminus a \xrightarrow{\tau t} Q \setminus a$ tenemos que $a \notin \mathsf{TA}(P) \upharpoonright t$ y

    $$P \xrightarrow{\tau t} P_1 \quad \text{o} \quad P \xrightarrow{at} P_1$$

Por hipótesis de inducción existe bs_1' tal que

$$\mathsf{ocul}(bs_1', a), \quad bs_1 = bs_1 \setminus a \quad \text{y} \quad P_1 \setminus a \stackrel{bs_1'}{\Longrightarrow} Q \setminus a$$

Tomando entonces la b-traza

$$bs' = \begin{cases} (\mathsf{TA}(P) \rceil t, t) \sqcup bs_1' & \text{si } P \xrightarrow{\tau t} P_1 \\ (\mathsf{TA}(P) \rceil t) at \cdot bs_1' & \text{si } P \xrightarrow{at} P_1 \end{cases}$$

tenemos que verifica las propiedades requeridas.

□

Proposición 6.3.12 Siendo bs una b-traza tal que $\mathsf{ocul}(bs, a)$, se tiene

$$P \stackrel{bs}{\Longrightarrow} Q \quad \Rightarrow \quad P \setminus a \stackrel{bs \setminus a}{\Longrightarrow} Q \setminus a$$

Demostración. Haremos la demostración por inducción sobre la longitud deducción de $P \stackrel{bs}{\Longrightarrow} Q$.

- Si la longitud de la deducción es uno, tenemos

$$P \stackrel{\epsilon}{\Longrightarrow} P \quad \Rightarrow \quad P \setminus a \stackrel{\epsilon}{\Longrightarrow} P \setminus a$$

- Si la longitud de la deducción es mayor que uno, tenemos los siguientes casos:

  - $P \rightarrowtail P_1$ y $P_1 \stackrel{bs}{\Longrightarrow} Q$. Por hipótesis de inducción tenemos que

$$P_1 \setminus a \stackrel{bs \setminus a}{\Longrightarrow} Q \setminus a$$

  La demostración acaba aquí puesto que $P \rightarrowtail P_1$.

  - $P \xrightarrow{at} P_1$, $P_1 \stackrel{bs_1}{\Longrightarrow} Q$. En este caso tenemos que

$$bs = (\mathsf{TA}(P) \rceil t) at \cdot bs_1, \quad bs_1 \neq \epsilon \quad \text{y} \quad bs \setminus a = (\mathsf{TA}(P) \rceil t, t) \sqcup (bs_1 \setminus a)$$

  Puesto que tenemos $\mathsf{ocul}(bs, a)$ se sigue $a \notin \mathsf{TA}(P) \rceil t$, por tanto $\mathsf{idle}(P, \{a\}) \geq t$ en consecuencia

$$P \setminus a \xrightarrow{\tau t} P_1 \setminus a$$

  Ahora, por hipótesis de inducción tenemos

$$P_1 \setminus a \stackrel{bs_1 \setminus a}{\Longrightarrow} Q \setminus a$$

  y por tanto tenemos

$$P \setminus a \stackrel{bs \setminus a}{\Longrightarrow} Q \setminus a$$

lo

– $P \xrightarrow{a't} P_1$, $P_1 \stackrel{bs_1}{\Longrightarrow} Q$. En este caso tenemos que

$$bs = (\mathsf{TA}(P) \upharpoonright t)a't \cdot bs_1, \quad \mathsf{ocul}(bs_1, a) \quad \text{y} \quad bs \setminus a = (\mathsf{TA}(P) \upharpoonright t)a't \cdot (bs_1 \setminus a)$$

Puesto que se tiene $\mathsf{ocul}(bs, a)$, deducimos $a \not\in \mathsf{TA}(P) \upharpoonright t$, por lo que

$$P \setminus a \xrightarrow{a't} P_1 \setminus a$$

Ahora, por hipótesis de inducción tenemos

$$P_1 \setminus a \stackrel{bs_1 \setminus a}{\Longrightarrow} Q \setminus a$$

y por tanto

$$P \setminus a \stackrel{bs \setminus a}{\Longrightarrow} Q \setminus a$$

– $P \xrightarrow{\tau t} P_1$, $P' \stackrel{bs_1}{\Longrightarrow} P$, $bs_1 \neq \epsilon$. En este caso tenemos que

$$bs = (\mathsf{TA}(P) \upharpoonright t) \sqcup bs_1 \quad bs_1 \neq \epsilon \quad \text{y} \quad bs \setminus a = (\mathsf{TA}(P) \upharpoonright t, t) \sqcup (bs_1 \setminus a)$$

Puesto que se tiene $\mathsf{ocul}(bs, a)$, derivamos $a \not\in \mathsf{TA}(P) \upharpoonright t$ y por tanto

$$P_1 \setminus a \stackrel{bs_1 \setminus a}{\Longrightarrow} Q \setminus a$$

de donde

$$P \setminus a \stackrel{bs \setminus a}{\Longrightarrow} Q \setminus a \, .$$

□

Proposición 6.3.13 Para cada proceso $P \in \mathsf{CRec}(\Sigma_{\mathsf{seq}})$ se verifican

1. $\mathsf{Barb}(P \setminus a) \ll \mathcal{B}_{\mathrm{con}}[\![\setminus a]\!](\mathsf{Barb}(P))$.
2. $\mathcal{B}_{\mathrm{con}}[\![\setminus a]\!](\mathsf{Barb}(P)) \ll \mathsf{Barb}(P \setminus a)$.

Demostración.

1. Tomemos $b = bs \cdot A \in \mathcal{B}_{\mathrm{con}}[\![\setminus a]\!](\mathsf{Barb}(P))$. Supongamos en primer lugar que el número de b-trazas bs' diferentes entre sí tales que

$$bs' \setminus a = bs \quad \text{y} \quad \exists P' : \; P \stackrel{bs'}{\Longrightarrow} P'$$

es infinito. Entonces la longitud de las b-trazas no puede estar acotada, por lo que podremos encontrar una secuencia de b-trazas que tienen un prefijo común de la forma

$$bs_0 \cdot A_1 a t_1 \cdots A_n a t_n \underbrace{\varnothing a 0 \cdots \varnothing a 0}_{h \text{ veces}}$$

donde la longitud h es arbitrariamente grande y $\mathsf{ocul}(bs_0, a)$. Tomamos entonces el estado

$$A_0 = \{\Omega t^{n+1}\} \cup \bigcup_{i \le n} A_i + t^i \qquad \text{donde} \qquad t^i = \sum_{j=1}^{i-1} t_j$$

Se verifica entonces que $(bs_0 \setminus a) \cdot A_0 \in \mathsf{Barb}(P \setminus a)$ y $(bs_0 \setminus a) \cdot A_0 \ll bs \cdot A$.

Podemos ahora suponer que el número de b-trazas que cumplen la condición anterior es finito, entonces tiene sentido considerar

$$\mathcal{Q} = \{P' \mid P \stackrel{bs'}{\Longrightarrow} P',\ bs' \setminus a = bs\} \quad \text{y} \quad Q = \prod_{P' \in \mathcal{Q}} P'$$

y aplicando la proposición 6.3.12, tenemos que para todo $P \in \mathcal{Q}$

$$P \setminus a \stackrel{bs}{\Longrightarrow} P' \setminus a$$

Además es fácil comprobar que

$$\mathsf{Barb}(\mathsf{Barb}(P \setminus a), bs) = \mathsf{Barb}(Q \setminus a)$$

Distinguimos dos casos en función de la forma A:

- $\mathsf{nd}(A) < \infty$. Entonces para cada $k \in \mathbb{N}$ existen $l \ge k$ y una b-traza $bs_k \in \mathsf{Barb}(P)$ tales que

  $$bs_k \setminus a = bs \quad \text{y} \quad A = \mathsf{Focul}(\mathsf{Barb}(P, bs_k), a, l)$$

  y por tanto $A \in \mathsf{Focul}(\mathsf{Barb}(Q), a, l)$ para cada $k \ge l$. Entonces aplicando la proposición 6.3.9 existe $l' \ge l$ tal que $A \in \mathcal{A}_{l'}(Q \setminus a)$. En consecuencia $A \in \mathcal{A}(Q \setminus a)$, por lo que existe $P' \in \mathcal{Q}$ tal que $A \in \mathcal{A}(P' \setminus a)$ y por tanto $bs \cdot A \in \mathsf{Barb}(P \setminus a)$.

- $\mathsf{nd}(A) = \infty$ para cada $t \in \mathcal{T}$ existen $l_t \in \mathbb{N}$, una b-traza $bs_t \in \mathsf{Barb}(P)$ y un estado A_t tales que

  $$bs_t \setminus a = bs, \quad A_t \upharpoonright t = A \upharpoonright t \quad \text{y} \quad A_t = \mathsf{Focul}(\mathsf{Barb}(P, bs_k), a, l_t)$$

  y por tanto $A_t \in \mathsf{Focul}(\mathsf{Barb}(Q), a, l_t)$. Entonces dos posibilidades según la proposición 6.3.9:

  - Si $\mathsf{nd}(A_t) = \infty$ entonces $A_t \in \mathcal{A}(Q \setminus a)$.
  - Si $\mathsf{nd}(A_t) < \infty$ existe $l'_t \ge l_t$ tal que $A_t \in \mathcal{A}_{l_t}(Q \setminus a)$.

que

En cualquier caso, para cada $t \in \mathcal{T}$ podemos encontrar $A'_t \in \mathcal{A}(Q \setminus a)$ tal que

$$A'_t \upharpoonright t = A_t \upharpoonright t = A \upharpoonright t$$

Puesto que esto es cierto para cada $t \in \mathcal{T}$, y los estados de un proceso son temporalmente compactos, tenemos que $A \in \mathcal{A}(Q \setminus a)$. Entonces existe $P' \in Q$ tal que $A \in \mathcal{A}(P' \setminus a)$ y por tanto $bs \cdot A \in \mathsf{Barb}(P \setminus a)$.

2. Tomemos ahora $bs \cdot A \in \mathsf{Barb}(P \setminus a)$, existe entonces P' tal que

$$P \setminus a \stackrel{bs}{\Longrightarrow} P' \setminus a \quad \text{y} \quad A \in \mathcal{A}(P' \setminus a)$$

En virtud la proposición 6.3.11 existirá una b-traza bs' tal que

$$P \stackrel{bs'}{\Longrightarrow} P' \quad \text{y} \quad bs = bs' \setminus a$$

Tenemos entonces el resultado buscado como consecuencia inmediata de la proposición 6.3.10.

$\square$

Proposición 6.3.14 Si $B \approx \mathsf{Barb}(P)$ entonces $\mathcal{B}_{\text{con}}[\![\setminus a]\!](B) \approx \mathsf{Barb}(P \setminus a)$.

Demostración. Tomando $b \in \mathsf{Barb}(P \setminus a)$, por aplicación de la proposición 6.3.13 tenemos que

$$\exists b' \in \mathcal{B}_{\text{con}}[\![\setminus a]\!](\mathsf{Barb}(P)) : \; b' \ll b$$

Entonces por la aplicación de la proposición 6.3.5 tenemos que existe $b'' \in \mathcal{B}_{\text{con}}[\![\setminus a]\!](B)$ tal que $b'' \ll b'$.

Tomemos ahora $b \in \mathcal{B}_{\text{con}}[\![\setminus a]\!](B)$. Por la proposición 6.3.5 existe $b' \in \mathcal{B}_{\text{con}}[\![\setminus a]\!](P)$ tal que $b' \ll b$, entonces aplicando la proposición 6.3.13 tenemos que existe $b'' \in \mathsf{Barb}(P \setminus a)$ tal que $b'' \ll b'$.

$\square$

Como consecuencia de esta proposición, podemos extender al lenguaje que incluye el operador de ocultamiento todos los resultados referentes a la abstracción de la semántica denotacional vistos en el capítulo 4.

6.3.4 Ecuaciones

Las ecuaciones que nos van a permitir trabajar con este operador son bastante simples. Simplemente hay que tener en cuenta que $\backslash a$ oculta la acción a. Por tanto, será suficiente considerar las siguientes ecuaciones de la tabla 6.2.

La corrección de dichas reglas es bastante obvia a partir de las definiciones dadas. Las reglas que hemos dado, permiten eliminar el operador de ocultamiento de cualquier proceso que esté en forma prenormal, con lo que el conjunto de ecuaciones obtenido seguirá siendo completo con respecto a la semántica de pruebas.

$$
\begin{array}{ll}
\textbf{[OCUL1]} & (et\,;P)\setminus a =_{\mathrm{E}} et\,;(P\setminus a) \qquad \text{si } e\in\mathcal{E},\ e\neq a \\
\textbf{[OCUL2]} & (at\,;P)\setminus a =_{\mathrm{E}} \tau t\,;(P\setminus a) \\
\textbf{[OCUL3]} & (P\sqcap Q)\setminus a =_{\mathrm{E}} (P\setminus a)\sqcap(Q\setminus a) \\
\textbf{[OCUL4]} & \left(\mathop{\Box}_{a't'\in A} P_{a't'}\right)\setminus a =_{\mathrm{E}} \mathop{\Box}_{a't'\in A\upharpoonright t}(P_{a't'}\setminus a)\sqcap \mathop{\Box}_{a't'\in A\not\upharpoonright t}(P_{a't'}\setminus a) \\
 & \text{siendo } A\in\mathcal{ST},\ a\in A \text{ y } t=\min\{t\mid at\in A\} \\
\textbf{[OCUL5]} & \left(\mathop{\Box}_{at\in A} P_{at}\right)\setminus a =_{\mathrm{E}} \mathop{\Box}_{at\in A}(P_{at}\setminus a) \\
 & \text{siendo } A\in\mathcal{ST} \text{ y } a\notin A
\end{array}
$$

Tabla 6.2: Ecuaciones para el ocultamiento.

se

6.4 Intervalos de Tiempo en Acciones Visibles

A la hora de hacer especificaciones de sistemas temporizados, conviene poder introducir intervalos temporales en las acciones que se van a ejecutar, los cuales indican que la acción indicada se puede ejecutar entre los instantes t_1 y t_2 indicados. Para ello introduciremos un nuevo operador

$$a[t_1, t_2]; P$$

cuyo significado es que el proceso comienza pudiendo ejecutar la acción a en el intervalo de tiempo $[t_1, t_2]$ y tras ellos comportarse comportarse como el proceso P. Formalmente introducimos en la signatura del modelo la siguiente familia de operadores.

$$\Big\{a[t_1, t_2]; \mid a \in Act, \quad t_1 \in \mathcal{T}, \quad t_1 \leq t_2 \quad \text{y} \quad t_2 \in \mathcal{T} \cup \{\infty\}\Big\}$$

Antes de seguir adelante vamos a centrar nuestra atención en dos detalles, el primero de ellos es que permitimos que el límite superior del intervalo sea infinito. Esto tiene dos consecuencias inmediatas

- Hemos introducido un proceso que puede realizar un número infinito de acciones temporizadas. A pesar de ello, la semántica operacional sigue siendo finitamente ramificada, en virtud de la definición que hemos dado (definición 3.2.5).

- No se trata de un operador derivado, en el sentido de que no se puede eliminar su presencia en el marco de los procesos finitos. Hasta ahora ningún proceso finito podía realizar un número infinito de acciones. No obstante, no tendremos que modificar las nociones de forma normal ni prenormal puesto que este operador será tratado mediante el operador de reducción:

$$\texttt{RED}(a[0, \infty]; P, t) = \texttt{DIV}(t) \,\Box \bigsqcap_{t' < t} at' \,;\, P$$

En segundo lugar, fijémonos en el hecho de que sólamente hemos permitido poner intervalos en acciones visibles; no permitimos intervalos en acciones internas. La razón que justifica esta limitación es nuestro deseo de para seguir manteniendo la de urgencia de dichas acciones. En tal caso, si permitiéramos intervalos en acciones ocultas, éstas deberían ser ejecutadas en el instante inferior de tiempo, con lo que sería inocua la presencia de los restantes valores del intervalo.

De nuevo, para poder definir la semántica operacional, en primer lugar hemos de comenzar por extender las funciones auxiliares que intervienen en la definición de las reglas. A continuación podremos dar las reglas en sí. Todo ello lo tenemos recogido en

$$[\mathbf{INT}] \quad a[t_1,t_2];P \xrightarrow{at} P \quad t \in \mathcal{T},\ t_1 \le t \le t_2$$

$$\begin{aligned}
\mathsf{stb}(\mathtt{RED}(P,t)) &= \ \mathsf{true} \\
\mathsf{Tiem}(a[t_1,t_2,];P,a') &= \begin{cases} \{t \in \mathcal{T} \mid t_1 \le t \le t_2\} & \text{si } a = a' \\ \varnothing & \text{en otro caso} \end{cases} \\
\mathsf{idle}(a[t_1,t_2];P,A) &= \begin{cases} \infty & \text{si } a \notin A \\ t_1 & \text{en otro caso} \end{cases} \\
\mathsf{Upd}(a[t_1,t_2];P,t) &= \begin{cases} a[t_1-t,t_2-t];P & \text{si } t \le t_1 \\ a[0,t_2-t];P & \text{si } t_1 < t \le t_2 \\ \mathtt{STOP} & \text{si } t_2 < t \end{cases}
\end{aligned}$$

$$a[t_1,t_2];P \downarrow$$

Tabla 6.3: Reglas y funciones que definen la semántica operacional de la operación de prefijo con intervalos de tiempo.

la tabla 6.3. Es fácil comprobar que tras la adición de este nuevo operador se siguen verificando las propiedades de la semántica operacional recogidas en la sección 3.2.1, con lo que también siguen siendo válidos todo los resultados correspondientes a la semántica de pruebas.

6.5 Semántica Denotacional

Extender la semántica denotacional para cubrir a este operador es bastante sencillo. Tomamos

$$\mathcal{B}_{\text{con}}[\![a[t_1,t_2];]\!](B) = \{A_a^{t_1,t_2}\} \cup \left\{(A_a^{t_1,t_2} \upharpoonright t)at \cdot b \mid t \in \mathcal{T},\ t_1 \le t \le t_2 \ \wedge\ b \in B\right\}$$

$$\text{donde} \quad A_a^{t_1,t_2} = \left\{\{at \mid t \in \mathcal{T} \ \wedge\ t_1 \le t \le t_2\}\right\}$$

Es fácil comprobar que este operador cumple todas las propiedades exigibles:

- Si B es un conjunto consistente, $a \in Act$, $t_1 \in \mathcal{T}$, $t_2 \in \mathcal{T} \cup \{\infty\}$ y $t_1 \le t_2$, entonces $\mathcal{B}_{\text{con}}[\![a[t_1,t_2];]\!](B)$ es también un conjunto consistente de barbas.
- El operador es monótono con respecto a la relación $\ll$.
- El operador es monótono y continuo con respecto a $\prec$.

Por tanto podemos extender en efecto la semántica denotacional para cubrir este nuevo operador.

6.6 Ecuaciones

[INT1]	$\mathtt{RED}(a[t_1,t_2];P,t_0) = \displaystyle\mathop{\Box}_{t_1\le t\le t_2} at\,;\mathtt{RED}(P,t_0-t)$	si $t_1 \le t_2 < t_0$
[INT2]	$\mathtt{RED}(a[t_1,t_2];P,t_0) = \displaystyle\mathop{\Box}_{t_1\le t\le t_0} at\,;\mathtt{RED}(P,t_0-t)$	si $t_1 < t_0 \le t_2$
[INT3]	$\mathtt{RED}(a[t_1,t_2];P,t_0) = \mathtt{DIV}(t_0)$	si $t_0 \le t_1$

Tabla 6.4: Axiomas para el operador de intervalos.

Las ecuaciones necesarias para tratar a este operador están recogidas en la tabla 6.4. Observemos que

$$a[t_1,t_2];P =_{\mathrm{E}} \mathop{\Box}_{t_1\le t\le t_2} at\,;P \quad \text{si} \;\; t_1 \le t_2 < \infty$$

es un proceso bien definido puesto que estamos considerando un dominio de tiempo discreto. Los axiomas de la tabla 6.4 **no** nos permiten en general eliminar la presencia del operador en un término finito. No obstante las reglas siguen siendo completas, puesto que que la eliminación sí es posible en el ámbito del operador $\mathtt{RED}(\cdot,\cdot)$. En cierta manera estamos tratando este operador como si fuera infinito. De hecho, podemos observar la igualdad

$$a[0,\infty];P = \mathtt{REC}\,x.(\tau\sigma\,;x)\,\Box\,a0\,;P$$

donde σ representa la unidad de tiempo. Si nos fijamos en la demostración de la completitud del sistema de axiomas (teorema 5.3.6), observamos que nos limitamos a usar el hecho de que los procesos de la forma

$$\mathtt{RED}(\mathsf{ap}(Q,l),t)$$

son finitos, y por tanto si

$$\mathtt{RED}(\mathsf{ap}(Q,l),t) \ll \mathtt{RED}(\mathsf{ap}(P,k),t)$$

podremos deducir

$$\mathtt{RED}(\mathsf{ap}(Q,l),t) <_{\mathrm{E}} \mathtt{RED}(\mathsf{ap}(P,k),t)$$

Esta última demostración se apoya en el hecho que cualquier proceso finito se puede pasar a forma prenormal. Este último resultado, en el caso que manejamos de intervalos infinitos, no es cierto en nuestro caso. Sin embargo el resultado final sigue siendo cierto puesto que es suficiente que los procesos

$$\mathtt{RED}(\mathsf{ap}(Q, l), t)$$

se puedan pasar a forma prenormal, y esto sigue siendo cierto gracias a los axiomas de la tabla 6.4.

Esta página está intencionadamente en blanco.

Capítulo 7

Suma

En los capítulos precedentes hemos trabajado con un álgebra de procesos tipo *CSP*, de modo que contamos con dos tipos de elección totalmente *pura*: la elección interna y la elección externa. En este capítulo indicaremos cómo podríamos extender los resultados a con un álgebra de procesos tipo *CCS*. La principal diferencia estriba en el hecho de que tendríamos ahora un único operador de elección, que llamaremos suma (+). Dicho operador se puede ver como una elección mixta, entre medias de la elección puramente no determinista y la elección puramente externa del *CSP*. Mientras la elección externa de *CSP* se *resolvía* únicamente mediante la ejecución de una acción visible, la suma puede ser *resuelta* mediante la ejecución de una acción oculta. Esto causa un notable problema de operador, pues ninguna semántica que *no vea* las acciones ocultas puede ser una congruencia con respecto al mismo. Es necesario disponer de una semántica algo *más fuerte* que *vea*, al menos, la primera acción oculta cuando esta ocurra.

Habitualmente no se estudia ningún álgebra de procesos que contenga los tres tipos de operadores de elección. De hecho, si consideramos un álgebra en que tenga como operador de elección únicamente a la suma, como se hace en [LdFN96], la caracterización de la congruencia es más sencilla. En este capítulo lo que hemos pretendido es integrar el operador de suma con los ya existentes, con lo que la caracterización de la congruencia se complica.

El contenido este capítulo no pretende recoger una teoría completa y cerrada de la cuestión; sino más bien una indicación a seguir, en el caso de que queramos extender la teoría vista hasta ahora. El completar esta extensión, hacer las demostraciones, completar los detalles que faltan, etc... debe ser visto como parte del trabajo futuro a desarrollar.

-se

7.1 Sintaxis

La sintaxis de la extensión se apoyará en la signatura expandida

$$\Sigma_+ = \Sigma_{\mathsf{seq}} \cup \{+\}$$

7.2 Semántica Operacional

Para definir la semántica operacional de procesos sobre esta nueva signatura, hemos de definir en primer lugar las funciones auxiliares que empleamos para definir la semántica operacional del lenguaje original. Las funciones se siguen definiendo en la misma manera que habíamos visto anteriormente en los que se refiere a lo operadores de Σ_{seq}. Basta por tanto definir las funciones para el operador de suma:

$$\begin{array}{rcl}
\mathsf{stb}(P+Q) & = & \mathsf{stb}(P) \wedge \mathsf{stb}(Q) \\
\mathsf{idle}(P+Q,A) & = & \min\Big(\mathsf{idle}(P,A), \mathsf{idle}(Q,A)\Big) \\
\mathsf{Tiem}(P+Q,a) & = & \Big(\mathsf{Tiem}(P,a) \nmid \mathsf{idle}(Q)\Big) \cup \Big(\mathsf{Tiem}(Q,a) \nmid \mathsf{idle}(P)\Big) \\
\mathsf{Upd}(P+Q,t) & = & \mathsf{Upd}(P,t) + \mathsf{Upd}(Q,t)
\end{array}$$

Por último, añadimos las reglas que permitirán generar sus transiciones:

$$[\text{MAS1}]\quad \frac{P \rightarrowtail P'}{P+Q \rightarrowtail P'+Q} \qquad\qquad [\text{MAS2}]\quad \frac{Q \rightarrowtail Q'}{P+Q \rightarrowtail P+Q'}$$

$$[\text{MAS3}]\quad \frac{P \xrightarrow{et} P',\ \mathsf{idle}(Q) \geq t,\ \mathsf{stb}(Q)}{P+Q \xrightarrow{et} P'} \qquad [\text{MAS4}]\quad \frac{Q \xrightarrow{et} Q',\ \mathsf{idle}(P) \geq t,\ \mathsf{stb}(P)}{P+Q \xrightarrow{et} Q'}$$

7.3 Semántica de Pruebas

Para dar una semántica de pruebas, lo primero es extender el predicado de convergencia para este operador; para ello, lo único que hemos de hacer es extender el predicado de convergencia débil, que por otra parte se hace de manera natural

$$P\downarrow \wedge\, Q\downarrow \quad\Rightarrow\quad P+Q\downarrow$$

Es fácil comprobar que se siguen manteniendo las propiedades de la semántica operacional dadas en la sección 3.2.1. Esto implica en particular que la caracterización operacional de la semántica de pruebas, vista en el capítulo 3, sigue siendo válida si extendemos el álgebra con este operador. Lo que ocurre es que dicha semántica no es conservada al aplicar el nuevo operador, surgen los problemas típicos en relación con la falta de congruencia de la suma en *CCS*, quizás agravados en nuestro caso.

Ejemplo 7.3.1 Consideremos los procesos

$$P = a4\,;\mathtt{STOP} \quad \text{y} \quad Q = \tau 0\,; a4\,;\mathtt{STOP}$$

Evidentemente, no hay diferencia entre P y Q en base a la semántica de pruebas, se puede comprobar fácilmente que se verifica $P \eqsim Q$.

Sin embargo, si consideramos el proceso

$$R = b1\,;\mathtt{STOP}$$

tenemos que $P + R$ puede ejecutar la acción b en el instante 1, cosa que $Q + R$ no puede hacer puesto que se ha de ejecutar antes la acción oculta en el instante 0, decidiendo así la elección e impidiendo la ejecución de la acción b en el instante 1. De hecho si consideramos la prueba $T = b1\,;\mathtt{OK}$ tenemos

$$P + Q \text{ must } T \quad \text{y} \quad Q + R \text{ m}\!\!/\!\text{ust } T$$

y por tanto $P + R \not\eqsim Q + R$.

□

La mayor congruencia contenida en la semántica de pruebas viene dada por la

Definición 7.3.2 Siendo $P, Q \in \mathsf{CRec}(\Sigma_{\mathsf{seq}})$, definimos

$$\begin{array}{lcl} P \underset{\sim}{\sqsubset}^{\mathrm{C}} Q & \Longleftrightarrow & C[P] \underset{\sim}{\sqsubset} C[Q] \\ P \ll^{C} Q & \Longleftrightarrow & C[Q] \ll C[Q] \end{array}$$

donde $C[\cdot]$ es cualquier contexto en el que se puedan poner los procesos.

□

Debido a la caracterización de la semántica de pruebas tendremos la equivalencia

$$P \underset{\sim}{\sqsubset}^{\mathrm{C}} Q \quad \Longleftrightarrow \quad P \ll^{C} Q$$

El resto de la sección lo destinaremos a dar una caracterización más explícita de la relación $\underset{\sim}{\sqsubset}^{\mathrm{C}}$. Para ello introducimos los concepto de t-barba y tb-traza.

Definición 7.3.3

- Una *tb-traza* es un par $tbs = (t, bs)$ donde $t \in \mathcal{T}$ y bs es una b-traza de longitud 1, $bs = A_1 a_1 t_1$, verificando $t \leq t_1$.

- Una t-barba es un par $tb = (t, b)$ donde $t \in \mathcal{T} \cup \{\infty\}$ y b es una barba que verifican:

dijo

  - Si b es un estado A, entonces $t \leq \mathsf{nd}(A)$.
  - Si $b = A_1 a_1 t_1 \cdot b'$, entonces $t \leq t_1$.

- Si TB es un conjunto de t-barbas, definimos su conjunto de tb-trazas por medio de

$$\mathsf{TBtraz}(TB) = \left\{ (t, A_1 a_1 t_1) \,\middle|\, \exists b : \; (t, A_1 a_1 t_1 \cdot b) \in TB \right\}$$

  Y si $(t, A_1 a_1 t_1) \in \mathsf{TBtraz}(TB)$, definimos el conjunto de barbas continuación mediante

$$\mathsf{Barb}(TB, (t, A_1 a_1 t_1)) = \left\{ b \,\middle|\, (t, A_1 a_1 t_1 \cdot b) \in TB \right\}$$

□

La idea que hay detrás del concepto de tb-traza $(t, A_1 a_1 t_1)$ y del de t-barba (t, b), es la de recordar ahora también la información sobre el instante t, en el que ha sido ejecutada la primera acción cuando se ha generado la barba. Por tanto, el conjunto de tb-trazas y el de t-barbas de un proceso se definirán de la siguiente manera:

Definición 7.3.4 Siendo P, Q dos procesos y $(t, A_1 a_1 t_1)$ una tb-traza definimos inductivamente la relación

$$P \xRightarrow{(t, A_1 a_1 t_1)} Q$$

en la siguiente forma:

- Si $P \xrightarrow{at} Q$ entonces $P \xRightarrow{(t, \mathsf{TA}(P) at)} Q$.
- Si $P \rightarrowtail P'$ y $P' \xRightarrow{(t, A_1 a_1 t_1)} Q$ entonces $P \xRightarrow{(t, A_1 a_1 t_1)} Q$.
- Si $P \xrightarrow{\tau t} P'$ $\xRightarrow{(t', A_1 a_1 t_1)} Q$ entonces $P \xRightarrow{(t, A'_1 a_1 t_1)} Q$ donde $A'_1 = \mathsf{TA}(P) \uparrow t \cup (A_1 + t)$.

Siendo P un proceso, definiremos el conjunto de t-barbas de P ($\mathsf{TBarb}(P)$) como el menor conjunto de t-barbas que verifica:

- Si $P \Uparrow$ entonces $\mathsf{TBarb}(P) = \left\{ (0, \{\Omega 0\}) \right\}$.
- Si $P \Downarrow$ se tendrá
  - Si $P \rightarrowtail P'$ y $tb \in \mathsf{TBarb}(P')$ entonces $tb \in \mathsf{TBarb}(P)$.
  - Si $P \xrightarrow{\tau t} P'$ y $b' \in \mathsf{Barb}(P)$ entonces $(t, (\mathsf{TA}(P) \uparrow t, t) \sqcup b') \in \mathsf{TBarb}(P)$.
  - Si $P \xrightarrow{at} P'$ y $b' \in \mathsf{Barb}(P')$ entonces $(t, \mathsf{TA}(P) at \cdot b') \in \mathsf{TBarb}(P)$.
  - Si $\mathsf{stb}(P)$ e $\mathsf{idle}(P) = \infty$, entonces $(\infty, \mathsf{TA}(P)) \in \mathsf{TBarb}(P)$.

□

La idea para caracterizar la relación $\sqsubseteq^{\mathrm{C}}_{\approx}$ es partir de las t-barbas, al igual que hicimos con la relación $\sqsubseteq_{\approx}$ y las barbas. Entre t-barbas y conjuntos de t-barbas definiremos la relación $\ll^T$, que es en cierta forma la extensión natural de la relación $\ll$.

Definición 7.3.5 Siendo TB_1 y TB_2 conjuntos de t-barbas, diremos que $TB_1 \ll^T TB_2$ si y sólo si se verifican las siguientes condiciones:

- Para todo $(t_2, b_2) \in TB_2$ existe $(t_1, b_1) \in TB_1$ tal que $t_1 \leq t_2$ y $b_1 \ll b_2$.
- Para todo $(t_2, A_2) \in TB_2$ y todo $t' \in \mathcal{T}$ tal que $t' \leq t_2$ y $t' < \mathsf{nd}(A_2)$, existe $(t_1, A_1) \in TB_1$ de modo que se verifica una de las siguientes condiciones:
  - $t_1 \geq t'$ y $A_1 \upharpoonright t' \subseteq A_2$, o bien
  - $\mathsf{nd}(A_1) \leq t'$ y $TAct(A_1) \subseteq A_2$.

□

La relación $\ll^T$ induce una relación entre procesos dada por:

$$P \ll^T Q \iff \mathsf{TBarb}(P) \ll^T \mathsf{TBarb}(Q)$$

En lo que resta de sección probaremos la equivalencia

$$P \sqsubseteq^{\mathrm{C}}_{\approx} Q \iff P \ll^T Q$$

El objetivo de la definición7.3.5 es conseguir que para cada $t \in \mathcal{T}$ se verifique

$$P + ct\,;\mathtt{STOP} \ll Q + ct\,;\mathtt{STOP} \iff P \ll^T Q$$

donde c es una acción nueva, que no aparece ni en P ni en Q. Por lo que para probar la caracterización comenzamos por demostrar la

Proposición 7.3.6 Siendo P y Q dos procesos, las siguientes afirmaciones serán equivalentes:

1. $P \sqsubseteq^{\mathrm{C}}_{\approx} Q$.
2. Para todo proceso R se verifica $P + R \sqsubseteq_{\approx} Q + R$.
3. Para todo $t \in \mathcal{T}$ se verifica

   $$P + ct\,;\mathtt{STOP} \sqsubseteq_{\approx} Q + ct\,;\mathtt{STOP}$$

   donde c es una acción nueva, que no aparece ni en P ni en Q.

el

Demostración. Las implicaciones de arriba hacia abajo son triviales. No limitaremos por tanto a probar las implicaciones de abajo a arriba.

2 $\Rightarrow$ **1.** Para demostrar que $P \sqsubseteq^{C}_{\approx} Q$, hemos de considerar un contexto cualquiera $C[\cdot]$ y probar que

$$C[P] \sqsubseteq_{\approx} C[Q]$$

Haremos entonces la demostración por inducción estructural sobre la construcción del contexto. Para los operadores distintos a la suma la demostración consiste simplemente en aplicar el hecho de que son monótonos con respecto a $\sqsubseteq_{\approx}$. En el caso de la suma se tiene por la condición de que se parte.

3 $\Rightarrow$ **2.** Razonaremos por reducción al absurdo. Supongamos que existe un proceso R tal que

$$P + R \not\sqsubseteq_{\approx} Q + R$$

Por la caracterización de $\sqsubseteq_{\approx}$ mediante barbas, existirá

$$b = bs \cdot A \in \mathsf{Barb}(Q + R)$$

de manera que no existe $b' \in \mathsf{Barb}(P + R)$ tal que $b' \ll b$. Sin pérdida de generalidad podemos suponer que tanto Q como R son procesos estables y convergentes, y además $P \Downarrow$. Según sean Q y R tenemos las siguientes posibilidades:

- $Q \xrightarrow{\tau t} Q'$ e $\mathsf{idle}(R) \geq t$. En este caso la barba b será de la forma

$$b = (\mathsf{TA}(Q) \upharpoonright t \cup \mathsf{TA}(R) \upharpoonright t, t) \sqcup b_1$$

Tendremos entonces que

$$b' = (\mathsf{TA}(Q) \upharpoonright t, t) \sqcup b_1 \in \mathsf{Barb}(Q + ct\,; \mathsf{STOP})$$

Puesto que

$$P + ct\,; \mathsf{STOP} \sqsubseteq_{\approx} Q + ct\,; \mathsf{STOP}$$

o lo que es lo mismo $P + ct\,; \mathsf{STOP} \ll Q + ct\,; \mathsf{STOP}$, existirá entonces una barba $b'' \in \mathsf{Barb}(P + ct\,; \mathsf{STOP})$ tal que $b'' \ll b'$. Como quiera que la acción c en el instante t no puede estar presente en la barba, tendremos que la barba b'' podrá ser generada de alguna de las siguientes formas:

  - $P \rightarrowtail^{*} P' \xrightarrow{\tau t'} P''$, $t' \leq t$ y $b'' = (\mathsf{TA}(P') \upharpoonright t', t') \sqcup b_1''$. En este caso tomamos la barba

$$b''' = (\mathsf{TA}(P') \upharpoonright t' \cup \mathsf{TA}(R) \upharpoonright, t') \sqcup b_1'' \; in \mathsf{Barb}(P + R)$$

que verificaría $b''' \ll b$, lo que está en contradicción con la hipótesis.

– $P \succ\!\!\longrightarrow^* P' \xrightarrow{at} P''$ y $b'' = \mathsf{TA}(P)at \cdot b_1''$. En este caso tomamos la barba

$$b''' = (\mathsf{TA}(P') \upharpoonright t' \cup \mathsf{TA}(R)\upharpoonright)at \cdot b_1'' \in \mathsf{Barb}(P + R)$$

que verificaría b''' y $b''' \ll b$, lo que está de nuevo en contradicción con la hipótesis.

- $R \xrightarrow{rt} R'$ e $\mathsf{idle}(Q) \geq t$. En este caso tenemos

$$b = ((\mathsf{TA}(Q) \upharpoonright t) \cup (\mathsf{TA}(R) \upharpoonright t)t), \sqcup b_1 \qquad \text{donde} \qquad b_1 \in \mathsf{Barb}(R)$$

Por otro lado, y puesto que $\mathsf{idle}(Q) \geq t$ tenemos

$$(\mathsf{TA}(Q) \upharpoonright t)ct\varnothing \in \mathsf{Barb}(Q + ct\,;\mathtt{STOP})$$

Y puesto que $P + ct\,;\mathtt{STOP} \sqsubseteq_{\approx} Q + ct\,;\mathtt{STOP}$ podremos encontrar un proceso estable P' tal que

$$P \succ\!\!\longrightarrow^* P', \quad \mathsf{idle}(P') \geq t \quad \text{y} \quad \mathsf{TA}(P) \upharpoonright t \subseteq \mathsf{TA}(Q) \upharpoonright t$$

Ahora, si tomamos la barba

$$b' = ((\mathsf{TA}(P') \upharpoonright t) \cup (\mathsf{TA}(R) \upharpoonright t), t) \sqcup b_1 \in \mathsf{Barb}(P + R)$$

verifica $b' \ll b$, lo que vuelve a estar en contradicción con la hipótesis de partida.

- $\mathsf{idle}(Q) = \mathsf{idle}(R) = \infty$. Tenemos dos posibilidades para la barba b:

  – $b = \mathsf{TA}(P) \cup \mathsf{TA}(Q)$. En este caso tenemos que para cada $t \in \mathcal{T}$ se verifica

$$(\mathsf{TA}(P) \cup \{ct\} \in \mathsf{Barb}(Q + ct\,;\mathtt{STOP})$$

con lo que existe $A_t \in \mathsf{Barb}(P + ct\,;\mathtt{STOP})$ tal que $A_t \ll \mathsf{TA}(P) \cup \{ct\}$. Por la proposición 2.4.8 existe un estado $A \in \mathsf{Barb}(P)$ tal que $A \ll \mathsf{TA}(P)$. Tenemos entonces que existe un proceso P' estable tal que

$$P \succ\!\!\longrightarrow^* P' \quad \text{y} \quad A \in \mathsf{Barb}(P')$$

Tomando entonces el estado $A' \cup \Big(\mathsf{TA}(R) \upharpoonright \mathsf{idle}(P')\Big)$ se tiene

$$A' \ll b \quad \text{y} \quad A' \in \mathsf{Barb}(P + R)$$

lo cual está en contradicción con nuestra hipótesis de partida.

principipito-,

– $b = Aat \cdot b_1$. En este caso tenemos

$$A = (\mathsf{TA}(Q) \upharpoonright t) \cup (\mathsf{TA}(Q) \upharpoonright t)$$

y se verifica una de las siguientes

* $R \xrightarrow{at} R_1$ y $b_1 \in \mathsf{Barb}(R_1)$. Puesto que $\mathsf{idle}(Q) \geq t$ tenemos

$$(\mathsf{TA}(Q) \upharpoonright t)ct\varnothing \in \mathsf{Barb}(Q + ct\,;\mathtt{STOP})$$

Y puesto que $P + ct;\mathtt{STOP} \sqsubseteq Q + ct;\mathtt{STOP}$ podemos encontrar una barba $b' \in \mathsf{Barb}(P + ct\,;\mathtt{STOP})$ tal que

$$b' \ll (\mathsf{TA}(Q) \upharpoonright t)ct\varnothing$$

Entonces hay dos posibilidades dependiendo de b':

· si b' se trata de un estado A', éste ha de verificar $\mathsf{nd}(A') \leq t$. Tomamos entonces el estado $A'' = A' \cup \mathsf{TA}(R) \upharpoonright \mathsf{nd}(A')$ que está en el conjunto $\mathsf{Barb}(P + R)$ y que verifica $A' \ll b$, lo que está en contradicción con la hipótesis.

· $b' = A'at \cdot b'_1$, entonces podremos encontrar un proceso P' estable tal que

$$P \rightarrowtail^{*} P', \quad \mathsf{idle}(P') \geq t \quad \text{y} \quad \mathsf{TA}(P) \upharpoonright t \subseteq \mathsf{TA}(Q) \upharpoonright t$$

Ahora si tomamos la barba

$$b'' = ((\mathsf{TA}(P') \upharpoonright t) \cup (\mathsf{TA}(R) \upharpoonright t))ct\varnothing \in \mathsf{Barb}(P + R)$$

verifica $b' \ll b$, lo que de nuevo está en contradicción con la hipótesis de partida.

* $Q \xrightarrow{at} Q_1$ y $b_1 \in \mathsf{Barb}(Q_1)$. En este caso tenemos

$$(\mathsf{TA}(Q) \upharpoonright t)atb_1 \in \mathsf{Barb}(Q + ct\,;\mathtt{STOP})$$

y puesto que $P + ct\,;\mathtt{STOP} \sqsubseteq Q + ct\,;\mathtt{STOP}$ existe una barba b' en el conjunto $\mathsf{Barb}(P + ct)$ tal que

$$b' \ll (\mathsf{TA}(Q) \upharpoonright t)atb_1$$

podremos encontrar un proceso P' estable tal que

$$P \rightarrowtail^{*} P' \quad \text{y} \quad b' \in \mathsf{Barb}(P')$$

Entonces tomamos la barba

$$b'' = \begin{cases} A' \cup (\mathsf{TA}(R) \upharpoonright \mathsf{nd}(A')) & \text{si } b' = A' \\ (A'_1 \cup (\mathsf{TA}(R) \upharpoonright t))at \cdot b'_1 & \text{si } b' = A'_1 at \cdot b'_1 \end{cases}$$

que verifica

$$b'' \in \mathsf{Barb}(P + R) \quad \text{y} \quad b'' \ll b$$

lo cual está en contradicción con la hipótesis de partida.

Puesto que la existencia de tal barba b nos ha llevado en cada caso a una contradicción, concluimos la no existencia de la misma con lo que para toda barba $b \in \mathsf{Barb}(Q + R)$ existe otra barba $b' \in \mathsf{Barb}(P + R)$ verificando $b' \ll b$. Lo que nos conduce a que $P + R \precsim Q + R$.

□

Abordamos ahora la demostración de la implicación

$$P \ll^T Q \quad \Rightarrow P \precsim^{\mathrm{C}} Q$$

En virtud del lema anterior y la caracterización de la relación $\precsim$, se trata de una consecuencia de la siguiente

Proposición 7.3.7 Si $P \ll^T Q$, entonces para todo $t \in \mathcal{T}$ se verifica

$$P + ct\,;\mathtt{STOP} \ll Q + ct\,;\mathtt{STOP}$$

Demostración. Para demostrarlo tomemos $b \in \mathsf{Barb}(Q + ct\ ;\mathtt{STOP})$, supongamos sin pérdida de generalidad que $\mathsf{stb}(Q)$ y $Q \Downarrow$. Tenemos las siguientes posibilidades:

- $b = A$ donde $ct \notin A$. Si ello sucede es porque existe un instante $t_Q \leq t$ tal que $(t_Q, A) \in \mathsf{TBarb}(Q)$. Puesto que $P \ll^T Q$, existe (t_P, A_P) con $t_P \leq t_Q$ y $A_P \ll A$, y ya que $t_P \leq t_Q \leq t$, tenemos que $A_P \in \mathsf{Barb}(P + ct\,;\mathtt{STOP})$.
- $b = A$ con $ct \in A$. En este caso tenemos que $A_Q \setminus \{ct\} \in \mathcal{A}(Q)$, y por tanto existe $t_Q > t$ tal que $(t_Q, A_Q) \in \mathsf{TBarb}(Q)$. Puesto que $P \ll^T Q$, existe (t_P, A_P) con $t_P \leq t_Q$ y $A_P \ll A$. Ahora tenemos dos posibilidades:
  - $t_P \leq t$ en cuyo caso $A_P \in \mathsf{Barb}(P + ct\,;\mathtt{STOP})$ por lo que tenemos $A_P \ll A$.
  - $t_P > t$ en cuyo caso tomamos $A' = A_P \cup \{ct\}$, verificándose

$$A' \in \mathcal{A}(P + ct\,;\mathtt{STOP}) \quad \text{y} \quad A' \ll A$$

---

si

- $b = A_1 ct\varnothing$. En este caso tenemos que $\mathsf{idle}(Q) \geq t$ y $A_1 = \mathsf{TA}(Q) \upharpoonright t$. Existe entonces $(t_Q, A_Q) \in \mathsf{TBarb}(Q)$ tal que $t_Q \geq t$ y $A_1 = A_Q \upharpoonright t$. Si $\mathsf{nd}(A_Q) = t$, como $P \ll^T Q$ existirá $(t_P, A_P) \in \mathsf{TBarb}(P)$ de modo que $A_P \ll A_Q$, entonces

$$A_P \in \mathsf{Barb}(P + ct\,;\mathtt{STOP}) \quad \text{y} \quad A_P \ll b$$

  Por tanto podemos suponer se verifica $t < \mathsf{nd}(A_Q)$, y como que $t \leq t_Q$ existirá $(t_P, A_P) \in \mathsf{TBarb}(P)$ verificando una de las siguientes posibilidades:

  - $t_P \geq t$ y $A_P \upharpoonright t_Q \subseteq A_Q$, con lo cual tendremos

$$(A_P \upharpoonright t)ct\varnothing \in \mathsf{Barb}(P + ct\,;\mathtt{STOP})$$

  - $\mathsf{nd}(A_P) \leq t$ y $TAct(A_P) \subseteq A_Q$. Tendremos entonces que A_P es un estado de $P + ct\,;\mathtt{STOP}$ y además $A_P \ll b$.

  En ambos casos hemos encontrado $b \in \mathsf{Barb}(P + ct\,;\mathtt{STOP})$ tal que $b \ll A_1 ct\varnothing$.

- $b = A_1 a_1 t_1 \cdot b'$, $c \notin A_1$ y $c \neq a_1$. Entonces existe $t_Q \leq t$ tal que $(t_Q, b) \in \mathsf{TBarb}(Q)$. Puesto que $P \ll^T Q$ existe $tb' = (t_P, b_P) \in \mathsf{TBarb}(P)$ tal que $b_P \ll^T b$ y $t_P \leq t_Q$, y entonces $b_P \in \mathsf{Barb}(P + ct\,;\mathtt{STOP})$.

- $b = A_1 a_1 t_1 \cdot b'$, $ct \in A_1$ y $c \neq a_1$. Tomamos $b_Q = (A_1 \setminus \{ct\}) a_1 t_1 \cdot b'$, entonces existe t_Q tal que $(t_Q, b_Q) \in \mathsf{TBarb}(Q)$. Puesto que $P \ll^T Q$ existe $tb' = (t_P, b_P) \in \mathsf{TBarb}(P)$ tal que $b_P \ll^T b_Q$. Si $t_P \leq t$, tenemos que $b_P \in \mathsf{Barb}(P + ct\,;\mathtt{STOP})$ y $b_P \ll b$. Si $t_P > t$ entonces tomamos como b' la barba resultante de añadir ct al conjunto inicial de b_P, y obtener así $b' \in \mathsf{Barb}(P + ct\,;\mathtt{STOP})$ y $b' \ll b$.

□

Como corolario de estas dos proposiciones tenemos

$$P \ll^T Q \quad \Rightarrow P \lessapprox^{\mathrm{C}} Q$$

Demostramos ahora el recíproco por reducción al absurdo, es decir,

Proposición 7.3.8 Si $P \not\ll^T Q$ entonces existe un cierto contexto $C[\cdot]$ tales que

$$C[P] \not\ll C[Q]$$

Demostración. Si $P \not\ll^T Q$ tenemos dos posibilidades:

- Existe $(t, b) \in \mathsf{TBarb}(Q)$ y no existe $(t'b') \in \mathsf{TBarb}(P)$ con $b' \ll^T b$ y $t' \leq t$. Consideramos en este caso el contexto $C[\cdot] = [\cdot] + ct\,;\mathtt{STOP}$. Se tiene entonces

$$b \in \mathsf{Barb}(Q + ct\,;\mathtt{STOP})$$

Demostraremos que no existe $b' \in \mathsf{Barb}(P + ct\,;\mathtt{STOP})$ tal que $b' \ll b$ en función de la forma de b:

  - $b = A$. Tomemos $A' \in \mathsf{Barb}(P + ct;\mathtt{STOP})$; existe entonces $(t_P, A_P) \in \mathsf{TBarb}(P)$ tal que $A_P = A' \setminus \{ct\}$. Si $A_P \ll A$ tendríamos $t_P > t$, por lo cual $ct \in A'$ y por tanto $A' \not\ll A$. Si por el contrario $A_P \not\ll A$, entonces es imposible que $A' \ll A$. Con este caso está resuelto.
  - $b = A_1 a_1 t_1 \cdot b_1$, en tal caso se verifica $ct \notin A_1$. Sea $b' \in \mathsf{Barb}(P + ct\,;\mathtt{STOP})$ y supongamos que $b' \ll b$, sólo hay dos posibilidades:
    * $b' = A'_1 a_1 t_1 \cdot b'_1$ con $A'_1 \subseteq A_1$ y por tanto $ct \notin A_1$. Por lo que existe $t' \leq t$ tal que $(t', b') \in \mathsf{TBarb}(P)$, que está en contradicción con la hipótesis de partida.
    * $b' = A'$ con $\mathsf{nd}(A) \leq t_1$. Puesto que $A' \ll b$ tenemos que $ct \notin A$ por lo que existe $t' \leq t$ tal que $(t', A') \in \mathsf{TBarb}(P)$, que está en contradicción con la hipótesis de partida.

- Existen $(t, A) \in \mathsf{TBarb}(Q)$ y $t_1 \mathcal{T}$ tales que $t_1 < \mathsf{nd}(A)$, $t_1 \leq t$ y no se cumple la segunda condición de la definición 7.3.5, es decir, se verifican:

  - No existe $(t', A') \in \mathsf{TBarb}(P)$ tal que $t' \geq t_1$ y $A' \upharpoonright t_1 \subseteq A$.
  - No existe $(t', A') \in \mathsf{TBarb}(P)$ tal que $\mathsf{nd}(A') \leq t_1$ y $\mathit{TAct}(A') \subseteq A$.

Consideramos entonces el contexto

$$C = [\cdot] + ct_1\,;\mathtt{STOP}$$

Puesto que $t_1 < \mathsf{nd}(A)$ y $t_1 \leq t$, tenemos que

$$b = (A \upharpoonright t_1) ct_1 \varnothing \in \mathsf{Barb}(Q + ct_1\,;\mathtt{STOP})$$

Tomemos ahora $b' \in \mathsf{Barb}(P + ct_1\,;\mathtt{STOP})$, y veamos que $b' \not\ll b$. Podemos suponer sin pérdida de generalidad que $P \Downarrow$ y $\mathsf{stb}(P)$. Distinguimos entonces los siguientes casos en función de la barba b':

tuviera

- $b' = A'ct_1\varnothing$. En tal caso existe $(t_P, A_P) \in \mathsf{TBarb}(P)$ de modo que $t_P \geq t_1$ y $A_P \rceil t_1 = A'$. Pero al verificarse la primera de las condiciones de arriba, entonces $A_P \rceil t_1 \not\subseteq A$, y por tanto $b' \not\ll b$.
- $b' = A'$. Supongamos que $b' \ll b$, de modos que $\mathsf{nd}(A') \leq t$, y en consecuencia existe $t_P \in \mathcal{T}$ tal que $(t_P, A') \in \mathsf{TBarb}(P)$. Puesto que $b' \ll b$ tenemos $TAct(A') \subseteq A$ y $\mathsf{nd}(A') \leq t_1$, pero esto está en contradicción con nuestra hipótesis de partida.
- $b' = A'at \cdot b''$ con $a \neq c$. En cuyo caso tenemos directamente que $b' \not\ll b$.

□

Como corolario inmediato tenemos el

Teorema 7.3.9 Para cualesquiera procesos P y Q, se tiene

$$P \sqsubseteq^{\mathrm{C}} Q \quad \iff \quad P \ll^{T} Q$$

Demostración. Es consecuencia inmediata de las proposiciones 7.3.6, 7.3.7 y 7.3.8

□

7.4 Semántica Denotacional

Debido al mal papel que juegan las acciones internas en este caso deberemos basar ahora la definición de la semántica denotacional en las t-barbas. Para los operadores primitivos esta semántica es una simple adaptación de la vista en el capítulo 4. Hemos de comenzar por definir el dominio semántico, que vendrá dado por los conjuntos consistentes de t-barbas.

Definición 7.4.1 Un conjunto TB de t-barbas es consistente si verifica las siguientes condiciones:

- Si $(t, A_1a_1t_1) \in \mathsf{TBtraz}(TB)$ entonces el conjunto de barbas $\mathsf{Barb}(TB, (t, A_1a_1t_1))$ ha de ser también consistente.
- Si $(t, A_1a_1t_1) \in \mathsf{TBtraz}(TB)$ y $t' < t$ ha de existir (t'', A'') verificando

$$A_1 = A'' \rceil t_1 \quad \text{y} \quad t' < t''$$

- Si $(t, A) \in TB$ y $a_1t_1 \in A$, se verifica una de las siguientes condiciones:
  - Existe $(t', A') \in TB$ tal que

$$t' \leq t, \quad TAct(A') \subseteq A \quad \text{y} \quad \mathsf{nd}(A') \leq t_1$$

– Existe una tb-traza $(t', A_1 a_1 t_1) \in \mathsf{TBtraz}(TB)$ tal que

$$t' \leq t \quad \text{y} \quad A_1 \subseteq A$$

- Para cada $t \in \mathcal{T}$ los conjuntos de estados

$$\begin{array}{llll} \mathcal{A}_t^+ = \{A \mid \exists t' \in \mathcal{T} : t' > t & \text{y} & (t', A) \in TB\} & y \\ \mathcal{A}_t^- = \{A \mid \exists t' \in \mathcal{T} : t' \leq t & \text{y} & (t', A) \in TB\} & \end{array}$$

son temporalmente compactos.

□

Observemos que la definición anterior es una adaptación directa de lo noción de conjuntos consistentes de barbas:

- La primera condición nos indica que el conjunto de barbas tras una tb-traza es consistente.

- Las otras condiciones son una adaptación de los conceptos de cierre bajo prefijos, cierre bajo continuaciones y compacidad temporal, referidas a los estados iniciales. La razón por las que se exigen estas condiciones se exponen a continuación:

Prefijos. Si $(t, A_1 a_1 t_1)$ es una tb-traza, debe haber un estado A, la condición de conjunto de barbas consistente, que cumpla $A \upharpoonright t_1 = A_1$. Además hemos de tener en cuenta la congruencia. En particular si consideramos un contexto de la forma

$$C[\cdot] = [\cdot] + ct' \,; \mathtt{STOP}$$

donde c es una acción nueva y $t' \in \mathcal{T}$, si ct' se *no cuela* en A_1 es porque $t' < t$. En tal caso debe existir un estado que cumpla la condición de arriba y además permita que en efecto se incorpore ct'; es decir, para cada $t' < t$ ha de existir una t-barba (t'', A'') tal que

$$t' < t'' \quad \text{y} \quad A'' \upharpoonright t_1 = A$$

Continuaciones. Si $a_1 t_1$ está en algún estado A tiene que ser continuada por alguna tb-traza, a no ser que haya algún estado divergente cuyas acciones iniciales estén en el estado que lo *impida*. Hay que tener en cuenta que si algún ct' se cuela en la tb-traza, también ha de *colarse* en el estado correspondiente.

Compacidad temporal. Con esta condición nos aseguramos de que si sumamos $ct\,;$ STOP al proceso que *ha generado* el conjunto de t-barbas, el conjunto de estados sigue siendo temporalmente compacto.

Pasamos ahora a dar la definición semántica de los operadores primitivos (los de Σ_{seq}). Tenemos como punto de partida la definición de los operadores vista en el capítulo 4, hemos de modificar la definición de manera que en una t-barba tengamos información a cerca de la primera acción oculta (si hay alguna) que ha sido ejecutada.

Divergencia:

$$\mathcal{TB}_{\text{con}}[\![\texttt{DIV}]\!] = \Big\{(0, \{\Omega 0\})\Big\}$$

Interbloqueo:

$$\mathcal{TB}_{\text{con}}[\![\texttt{STOP}]\!] = \Big\{(\infty, \varnothing)\Big\}$$

Prefijo mediante acciones visibles:

$$\begin{aligned}\mathcal{TB}_{\text{con}}[\![at;]\!](TB) &= \Big\{(\infty, \{at\})\Big\} \cup \\ &\cup \Big\{(t, \varnothing at \cdot b) \mid \exists t' \in \mathcal{T} :\ (t', b) \in TB\Big\}\end{aligned}$$

Prefijo mediante acciones ocultas:

$$\mathcal{TB}_{\text{con}}[\![\tau t;]\!](TB) = \Big\{(t, b + t) \mid \exists t' \in \mathcal{T} :\ (t', b) \in TB\Big\}$$

Elección interna:

$$\mathcal{TB}_{\text{con}}[\![\sqcap]\!](TB_1, TB_2) = TB_1 \cup TB_2$$

Elección externa:

$$\begin{aligned}\mathcal{TB}_{\text{con}}[\![\Box]\!](TB_1, TB_2) = &\left\{ (t, A_1 \sqcup_{\varnothing} A_2) \;\middle|\; \begin{array}{l} \exists t_1, t_2 \in \mathcal{T} : \\ \quad (t_1, A_1) \in TB_1,\ (t_2, A_2) \in TB_2 \\ \quad \text{y}\ \ t = \min(t_1, t_2) \end{array} \right\} \\ &\bigcup \left\{ (t, (A_1 \cup A_2 \upharpoonright t_1) a_1 t_1 \cdot b) \;\middle|\; \begin{array}{l} \exists t_2, t' \in \mathcal{T} : \\ \quad (t_2, A_2) \in TB_2,\ \mathsf{nd}(A_2) > t_1, \\ \quad (t', A_1 a_1 t_1 \cdot b) \in TB_1 \\ \quad \text{y}\ \ t = \min(t', t_2) \end{array} \right\} \\ &\bigcup \left\{ (t, (A_1 \upharpoonright t_2 \cup A_2) a_2 t_2 \cdot b) \;\middle|\; \begin{array}{l} \exists t_1, t' \in \mathcal{T} : \\ \quad (t_1, A_1) \in TB_1,\ \mathsf{nd}(A_1) > t_2, \\ \quad (t', A_2 a_2 t_2 \cdot b) \in TB_2 \\ \quad \text{y}\ \ t = \min(t', t_1) \end{array} \right\}\end{aligned}$$

La definición de estos operadores es bastante natural, teniendo en cuenta el sentido que hemos dado al instante que acompaña a la t-barba. Para ver que estos operadores están bien definidos deberíamos proceder para cada uno de los operadores como hicimos en el capítulo 4. Las demostraciones serían prácticamente idénticas a las vistas en dicho capítulo.

Damos por último la definición semántica del operador de suma:

$$\mathcal{TB}_{\mathrm{con}}[\![+]\!](TB_1, TB_2) =$$

$$\left\{ (t_1, A_1 \cup (A_2 \upharpoonright t_1)) \;\middle|\; \begin{array}{c} (t_1, A_1) \in TB_1, \quad \exists t_2 \in \mathcal{T}: \ (t_2, A_2) \in TB_2 \\ \text{y} \ \ t_1 \leq t_2 \end{array} \right\}$$

$$\bigcup \left\{ (t_2, (A_1 \upharpoonright t_2) \cup A_2) \;\middle|\; \begin{array}{c} (t_2, A_2) \in TB_2, \quad \exists t_1 \in \mathcal{T}: \ (t_1, A_1) \in TB_1 \\ \text{y} \ \ t_2 \leq t_1 \end{array} \right\}$$

$$\bigcup \left\{ (t_1, (A_1 \cup (A_2 \upharpoonright t_1))at \cdot b) \;\middle|\; \begin{array}{l} (t_1, A_1 at \cdot b) \in TB_1, \ \mathsf{nd}(A_2) > t \\ \exists t_2 \in \mathcal{T}: \ (t_2, A_2) \in TB_2, \\ \quad \text{y} \ \ t_1 \leq t_2 \end{array} \right\}$$

$$\bigcup \left\{ (t_2, ((A_1 \upharpoonright t_2) \cup A_2)at \cdot b) \;\middle|\; \begin{array}{l} (t_2, A_2 at \cdot b) \in TB_2, \ \mathsf{nd}(A_1) > t \\ \exists t_2 \in \mathcal{T}: \ (t_1, A_1) \in TB_1, \\ \quad \text{y} \ \ t_2 \leq t_1 \end{array} \right\}$$

La complejidad de dicha definición viene justificada por el hecho de que las acciones ocultas eligen bajo el contexto de la suma. El instante en el que se realizan las acciones ocultas viene dado por la componente temporal de una t-barba. Podemos observar que el operador suma distribuye con respecto a la elección interna, como muestra el siguiente:

Ejemplo 7.4.2 Consideremos los procesos

$$P + (Q \sqcap R) \quad \text{y} \quad (P + Q) \sqcap (P + R)$$

El primer proceso puede realizar las transiciones

$$P + (Q \sqcap R) \rightarrowtail P + Q \quad \text{y} \quad P + (Q \sqcap R) \rightarrowtail P + R$$

mientras el segundo podrá realizar

$$(P + Q) \sqcap (P + R) \rightarrowtail P + Q \quad \text{y} \quad (P + Q) \sqcap (P + R) \rightarrowtail P + R$$

y

□

Para este operador se debería demostrar (al igual que para los anteriores) que está bien definido, y que su definición denotacional está de acuerdo con su comportamiento operacional. Como indicamos antes, el desarrollo en detalle de estos puntos los dejamos como parte del trabajo pendiente para el futuro.

A continuación, para completar la definición de la semántica denotacional, deberíamos incorporar al dominio una relación de orden que lo haga completo. La idea de nuevo es adaptar la relación que ya teníamos para las barbas:

Definición 7.4.3 Siendo $tb_1 = (t_1, b_1)$ y $tb_2 = (t_2, b_2)$ t-barbas, tendremos que $tb_1 \prec^T tb_2$ si y sólo si $b_1 \prec b_2$ y se cumplen las condiciones siguientes:

- Si $b_1 = bs \cdot A$ con $bs \neq \epsilon$ entonces $t_1 = t_2$
- Si $b_1 = A_1$ entonces $t_1 = \min(\mathsf{nd}(A_1), t_2)$.

□

Y extendemos el orden, para tenerlo definido entre conjuntos consistentes de t-barbas, como sigue:

Definición 7.4.4 Sean TB_1 y TB_2 conjuntos consistentes de t-barbas entonces diremos que $TB_1 \prec^T TB_2$ si y sólo si se verifican las siguientes condiciones:

- Para cada $tb_2 \in TB_2$ existe $tb_1 \in TB_1$ tal que $tb_1 \prec^T tb_2$.
- Para cada $(t, bs \cdot A_1) \in TB_1$ con $bs \neq \epsilon$ existe un estado A_2 tal que $A_1 \prec A_2$ y $(t, bs \cdot A_2) \in TB_2$.
- Para cada $(t_1, A_1) \in TB_1$ existe $(t_2, A_2) \in TB_2$ de manera que $(t_1, A_1) \prec^T (t_2, A_2)$.

□

Ahora habría que demostrar que:

1. Toda cadena no decreciente de conjuntos consistentes de t-barbas
$$TB_1 \prec^T TB_2 \prec^T \cdots TB_k \prec^T TB_{k+1} \prec^T \cdots$$
tiene una mínima cota superior.
2. Los operadores son monótonos y contínuos con respecto al orden introducido.

Observemos que, como en el capítulo 4, el orden definido es simplemente un preorden, por tanto no podremos esperar la unicidad de las mínimas cotas superiores de una cierta cadena.

7.5 Sistema de Ecuaciones

[TEQ1]	$P <_{\mathrm{ET}} P$	**[EQ2]**	$\dfrac{P <_{\mathrm{ET}} Q,\ Q <_{\mathrm{ET}} P}{P =_{\mathrm{ET}} Q}$
[TEQ3]	$\dfrac{P <_{\mathrm{ET}} Q,\ Q <_{\mathrm{ET}} R}{P <_{\mathrm{ET}} R}$	**[TEQ4]**	$\dfrac{P =_{\mathrm{ET}} Q}{P <_{\mathrm{ET}} Q,\ Q <_{\mathrm{ET}} P}$
[TMO1]	$\dfrac{P <_{\mathrm{E}} Q}{et\,;P <_{\mathrm{ET}} et\,;Q} \quad e \in \mathcal{E}$		
[TMO2]	$\dfrac{P <_{\mathrm{ET}} P',\ Q <_{\mathrm{ET}} Q'}{P \mathbin{\square} Q <_{\mathrm{ET}} P' \mathbin{\square} Q'}$	**[TMO3]**	$\dfrac{P <_{\mathrm{ET}} P',\ Q <_{\mathrm{ET}} Q'}{P \sqcap Q <_{\mathrm{ET}} P' \sqcap Q'}$
[TMO4]	$\dfrac{P <_{\mathrm{ET}} P',\ Q <_{\mathrm{ET}} Q'}{P + Q <_{\mathrm{ET}} P' + Q'}$		

Tabla 7.1: Relación de equivalencia y monotonía.

A la hora de desarrollar el correspondiente conjunto de axiomas correctos y completos, hemos de tener en cuenta que las acciones ocultas han de *poderse ver*, aunque sólo se trate de la primera aparición en una traza de un proceso.

Definición 7.5.1

- Si $t \in \mathcal{T} \cup \{\infty\}$ definimos

$$\mathtt{STOP}(t) = \begin{cases} \tau t\,; \mathtt{STOP} & \text{si } t < \infty \\ \mathtt{STOP} & \text{si } t = \infty \end{cases}$$

- Un t-estado es un par $TA = (t, A)$ donde $t \in \mathcal{T}$, A es un estado y $t \leq \mathsf{nd}(A)$. Diremos que TA es finito si A lo es, diremos que $a't' \in TA$ cuando $a't' \in A$.

- Si $TA = (t, A)$ es un t-estado finito y para cada $a't' \in A$ tenemos un proceso asociado $P_{a't'}$, definimos

$$\mathop{\square}_{a't' \in TA} P_{a't'} = \mathtt{STOP}(t) \mathbin{\square} \mathop{\square}_{a't' \in A} P_{a't'}$$

- Diremos que un proceso está en forma t-normal si es de la forma

$$\bigsqcap_{TA \in \mathcal{TA}} \mathop{\square}_{at \in TA} P^{TA}_{at}$$

  donde $\mathcal{TA}$ es un conjunto finito de t-estados, y cada uno de los $TA \in \mathcal{TA}$ es finito.

[TCOM1]	$P \,\square\, Q =_{\mathrm{ET}} Q \,\square\, P$
[TCOM2]	$P \sqcap Q =_{\mathrm{ET}} Q \sqcap P$
[TCOM3]	$P + Q =_{\mathrm{ET}} Q + P$
[TASO1]	$P \,\square\, (Q \,\square\, R) =_{\mathrm{ET}} (P \,\square\, Q) \,\square\, R$
[TASO2]	$P \sqcap (Q \sqcap R) =_{\mathrm{ET}} (P \sqcap Q) \sqcap R$
[TASO3]	$P + (Q + R) =_{\mathrm{ET}} (P + Q) + R$
[TID1]	$P =_{\mathrm{ET}} P \,\square\, P$
[TID2]	$P =_{\mathrm{ET}} P \sqcap P$
[TID4]	$P =_{\mathrm{ET}} P + P$
[TID3]	$\mathtt{STOP} \,\square\, P =_{\mathrm{ET}} P$
[TID5]	$\mathtt{STOP} + P =_{\mathrm{ET}} P$
[TDIS1]	$at\,; P \,\square\, at\,; Q =_{\mathrm{ET}} at\,; (P \sqcap Q)$
[TDIS2]	$P \,\square\, (Q \sqcap R) =_{\mathrm{ET}} (P \,\square\, Q) \sqcap (P \,\square\, R)$
[TDIS3]	$P + (Q \sqcap R) =_{\mathrm{ET}} (P + Q) \sqcap (P + R)$

Tabla 7.2: Conmutatividad, asociatividad, idempotencia, distributividad

□

En la tabla 7.1 tenemos un primer grupo de ecuaciones, que nos indican la monotonía de los distintos operadores. Observemos, que en el caso del prefijo basta considerar el sistema de axiomas que hemos presentado en el capítulo 5. A continuación, en la tabla 7.2, tenemos los axiomas de conmutatividad, idempotencia y distributividad. Ahora, para poder pasar cualquier proceso a forma t-normal y trabajar con ellas necesitamos además los axiomas presentados en la tabla 7.3.

Por último, si pretendemos trabajar con procesos recursivos, tendremos que ser capaces de manejar desigualdades, por ello introducimos la regla **[TME]** presentada en la tabla 7.4 es prácticamente una transcripción de la relación $\ll^T$ para procesos que están en forma t-normal. Debido a la complejidad de esta última regla, se hace necesario una explicación de la misma. Si tomemos dos procesos de la forma

$$P = \bigsqcap_{TA \in \mathcal{TA}} \ \square_{at \in TA} P^{TA}_{at} \quad \text{y} \quad Q = \bigsqcap_{TA \in \mathcal{TB}} \ \square_{at \in TA} Q^{TA}_{at}$$

para ver que $P \ll^T Q$ hemos de comprobar que:

1. Para todo $tb \in \mathcal{TB}_{\mathrm{con}}[\![Q]\!]$ existe $tb' \in \mathcal{TB}_{\mathrm{con}}[\![P]\!]$ tal que $tb' \ll^T tb$. Esto lo hacemos de una manera inductiva. En primer lugar hemos de probar que para cada t-estado

[TPREF1]	$\tau t_1\,;et_2\,;P =_{\mathrm{ET}} \mathtt{STOP}(t_1) \,\Box\, (e(t_1+t_2)\,;P), \quad e\in\mathcal{E}$
[TPREF2]	$et_1\,;\tau t_2\,;P =_{\mathrm{ET}} e(t_1+t_2)\,;P, \quad e\in\mathcal{E}$
[TPREF3]	$\mathtt{DIV}(t) =_{\mathrm{ET}} \mathtt{STOP}(t) \,\Box\, \mathtt{DIV}(t)$
[TPREF4]	$\mathtt{STOP}(t_1) \,\Box\, \mathtt{STOP}(t_2) =_{\mathrm{ET}} \mathtt{STOP}(t), \quad t=\min(t_1,t_2)$
[TEXT]	$\tau t\,;(P \,\Box\, Q) =_{\mathrm{ET}} (\tau t\,;P) \,\Box\, (\tau t\,;Q)$
[TINT]	$\tau t\,;(P \sqcap Q) =_{\mathrm{ET}} (\tau t\,;P) \sqcap (\tau t\,;Q)$
[TMAS1]	$(\mathtt{STOP}(t_1) \,\Box\, Q_1) + ((et_2\,;P) \,\Box\, Q_2) =_{\mathrm{ET}} \quad$ si $t_2 \le t_1$ $=_{\mathrm{ET}} (\mathtt{STOP}(t_1) \,\Box\, (et_2\,;P) \,\Box\, Q_1) + Q_2$
[TMAS2]	$(\mathtt{STOP}(t_1) \,\Box\, Q_1) + ((et_2\,;P) \,\Box\, Q_2) =_{\mathrm{ET}} \quad$ si $t_2 > t_1$ $=_{\mathrm{ET}} (\mathtt{STOP}(t_1) \,\Box\, Q_1) + Q_2$
[TMAS3]	$P + \mathtt{STOP}(\infty) =_{\mathrm{ET}} P$

Tabla 7.3: Ecuaciones para pasar a forma t-normal.

$(t, A) \in \mathcal{TB}_{\mathrm{con}}[\![Q]\!]$ existe $(t', A') \in \mathcal{TB}_{\mathrm{con}}[\![P]\!]$ tal que $(t, A) \ll^{T} (t', A')$. Es decir,

$$\forall\, TA \in \mathcal{TA} \quad \exists TA' \in \mathcal{TB}: \qquad TA' \ll^{T} TA$$

que es la primera parte de la premisa de la regla **[TME]**. A continuación, para cada t-barba $tb = (t, A_1 a_1 t_1 \cdot b) \in \mathcal{TB}_{\mathrm{con}}[\![Q]\!]$, para comprobar que existe $tb' \in \mathcal{TB}_{\mathrm{con}}[\![P]\!]$ tal que $tb' \ll^{T} tb$, tenemos dos posibilidades:

- Comprobar que existe $(t', A') \in \mathcal{TB}_{\mathrm{con}}[\![P]\!]$ tal que

  $$t' \le t, \quad \mathsf{nd}(A') \le t_1 \quad \text{y} \quad TAct(A') \subseteq A_1$$

  es decir,

  $$\mathsf{tBM}(\mathcal{TA}, TA, a_1 t_1) \neq \varnothing$$

- Comprobar que existe $(t', A'_1 a_1 t_1 \cdot b'_1) \in \mathcal{TB}_{\mathrm{con}}[\![P]\!]$ tal que

  $$t' \le t, \quad A'_1 \subseteq A_1 \quad \text{y} \quad b'_1 \ll b_1$$

  Para ello, dado que Puesto que $(t, A_1 a_1 t_1 \cdot b) \in \mathcal{TB}_{\mathrm{con}}[\![Q]\!]$ existirá un proceso Q_1 tal que

  $$Q \xrightarrow{(t, A_1 a_1 t_1)} Q_1 \quad \text{y} \quad b \in \mathsf{Barb}(Q_1)$$

  Para que exista dicha barba es necesario que el conjunto de procesos

  $$\mathcal{P} = \{P' \mid P \xrightarrow{(t', A'_1 a_1 t_1)} P' \quad A'_1 \subseteq A_1 \quad \text{y} \quad t' \le t\}$$

minutos

$$
[\mathbf{TME}]\quad
\frac{
\begin{array}{l}
\forall\, TA = (t,A) \in \mathcal{TB}: \\
\quad \exists\, TA' \in \mathcal{TA}: \quad TA' \ll^{T} TA \\
\quad \wedge \forall\, a_1t_1 \in A: \\
\qquad \mathsf{tBM}(\mathcal{TA}, TA, a_1t_1) \neq \varnothing\ \vee \\
\qquad \vee\ \mathsf{tPer}(\mathcal{TA}, TA, a_1t_1) \neq \varnothing\ \wedge \displaystyle\bigsqcap_{TA_1 \in \mathsf{tPer}(\mathcal{TA}, TA, a_1t_1)} P^{TA_1}_{a_1t_1} <_{\mathrm{E}} Q^{TA}_{at} \\
\quad \wedge \forall\, t' \in \mathcal{T}\ \text{t.q.}\ t' \leq t\ \wedge\ t' < \mathsf{nd}(A)\quad \exists\,(t'',A'') \in \mathcal{TA}: \\
\qquad (t'' \geq t'\ \wedge\ A'' \upharpoonright t' \subseteq A)\ \vee\ (\mathsf{nd}(A'') \leq t'\ \wedge\ TAct(A'') \subseteq A)
\end{array}
}{
\displaystyle\bigsqcap_{TA \in \mathcal{TA}} \bigsqcup_{at \in TA} P^{TA}_{at} <_{\mathrm{ET}} \bigsqcap_{TA \in \mathcal{TB}} \bigsqcup_{at \in TA} Q^{TA}_{at}
}
$$

$$
\begin{array}{llcl}
\text{donde} & \mathsf{tPer}(\mathcal{TA},(t,A),a_1t_1) & = & \{(t',A') \mid (t',A') \in \mathcal{TA},\ a_1t_1 \in A',\ A' \upharpoonright t \subseteq A\ \text{ y }\ t' \leq t\} \\
 & \mathsf{tBM}(\mathcal{TA},(t,A),a_1t_1) & = & \{(t',A') \mid (t',A') \in \mathcal{TA},\ \mathsf{nd}(A') \leq t_1\ \text{ y }\ TAct(A') \subseteq A\}
\end{array}
$$

Tabla 7.4: Regla para formas t-normales.

no sea vacío, pudiendo así tomar el proceso

$$P_1 = \bigsqcap_{P' \in \mathcal{P}} P'$$

Bastaría entonces comprobar que $P_1 \ll Q_1$, es decir,

$$\mathsf{tPer}(\mathcal{TA}, TA, a_1t_1) \neq \varnothing\ \wedge \bigsqcap_{TA_1 \in \mathsf{tPer}(\mathcal{TA}, TA, a_1t_1)} P^{TA_1}_{a_1t_1} <_{\mathrm{E}} Q^{TA}_{at}$$

Resumiendo todo lo anterior hemos de comprobar que

$$
\begin{array}{l}
\forall\, TA = (t,A) \in \mathcal{TA} \quad \forall\, a_1t_1 \in TA: \\
\quad \mathsf{tBM}(\mathcal{TA}, TA, a_1t_1) \neq \varnothing\ \vee \\
\quad \vee\ \mathsf{tPer}(\mathcal{TA}, TA, a_1t_1) \neq \varnothing\ \wedge \displaystyle\bigsqcap_{TA_1 \in \mathsf{tPer}(\mathcal{TA}, TA, a_1t_1)} P^{TA_1}_{a_1t_1} <_{\mathrm{E}} Q^{TA}_{at}
\end{array}
$$

que es precisamente la segunda condición que aparece en la segunda premisa de la regla [**TME**].

2. Por último hemos de comprobar que para todo $(t,A) \in \mathcal{TB}_{\mathsf{con}}[\![Q]\!]$ y para todo $t' \in \mathcal{T}$ verificando que $t' \leq t$ y $t' < \mathsf{nd}(A)$, existe $(t'',A'') \in \mathcal{TB}_{\mathsf{con}}[\![P]\!]$ tal que se verifica una de las siguientes condiciones:

    - $t'' \geq t'$ y $A'' \upharpoonright t' \subseteq A$, o bien

- $\mathsf{nd}(A'') \leq t'$ y $TAct(A'') \subseteqq A$.

Por tanto se ha de verificar

$$\begin{aligned}\forall\,(t,A) \in \mathcal{TB} \quad \forall\, t' \in \mathcal{T} \text{ t.q. } t' \leq t \;\wedge\; t' < \mathsf{nd}(A) \quad \exists\,(t'',A'') \in \mathcal{TA} :\\ (t'' \geq t' \;\wedge\; A'' \upharpoonright t' \subseteq A) \;\vee\; (\mathsf{nd}(A'') \leq t' \;\wedge\; TAct(A'') \subseteq A)\end{aligned}$$

que forma precisamente la última condición de la premisa de la regla

Observemos por último que para los procesos que hay *detrás* de las acciones visibles hemos de pasar a utilizar utilizar el sistema de axiomas del capítulo 5, pues las posibles acciones internas han dejado de tener efectos malignos.

---

para

Esta página está intencionadamente en blanco.

Capítulo 8

Conclusiones y Trabajo Futuro

Comenzamos haciendo una breve recapitulación de los trabajos previos a esta tesis sobre álgebras de procesos temporizadas. Desde un punto de vista meramente sintáctico las álgebras de procesos temporizadas se pueden ver como una extensión de las álgebras de procesos clásicas, *ACP* [BK84, BW90], *CCS* [Mil80, Mil83, Mil89], y *CSP* [BHR84, BR85, Hoa85]. En las distintas aproximaciones al problema *temporizado* se pretende mantener los conceptos centrales del correspondiente modelo original no temporizado en el que nos hayamos basados.

8.1 Extensiones de *ACP*

La aproximación a la concurrencia realizada por los autores del lenguaje *ACP* es eminentemente axiomática. Así la semántica original del lenguaje es la inducida por una serie de ecuaciones e inecuaciones. En la línea indicada más arriba las las extensiones temporizadas de este lenguaje usan también una aproximación axiomática al problema.

En [BB93] podemos encontrar una extensión temporizada del lenguaje *ACP*, que constituye el lenguaje llamado *BPA*. La manera de introducir el tiempo es parecida a la realizada en nuestro trabajo, haciéndose a través del operador de prefijo. Más en concreto, disponen de dos formas distintas de introducir el tiempo: por medio de un operador prefijo en el que el tiempo es global, por medio de un segundo en el que el tiempo es relativo. Un trabajo relacionado con este último es [Klu93] donde se estudian en detalle diversos tipos de semánticas de bisimulación y sus leyes algebraicas.

Otra extensión temporizada del lenguaje *ACP* la podemos encontrar en [NS94], dando lugar a un lenguaje que denominan *ATP*. Los operadores de tiempo que manejan los podemos considerar como de *bajo* nivel: los plazos temporales se controlan contando

---

gastar,

acciones especiales, χ (tic), que representan el paso de un ciclo de reloj. Puesto que esta acción representa el paso del tiempo absoluto, ha de ser una acción sincronizada por todos los procesos. Además, esta acción puede ser usada en el operador de prefijo como si fuera una acción cualquiera. Por último, mencionaremos que disponen tienen un operador de *time-out*, $\lfloor P \rfloor Q$, que cuando el proceso P puede dejar pasar el tiempo se comporta como el proceso Q, y en otro caso como P. Combinando este operador con el prefijo mediante la acción χ, se puede fijar el intervalo de tiempo en el que es posible ejecutar una determinada acción.

8.2 Extensiones de *CCS*

La semántica del lenguaje original no temporizado viene dada por en una semántica de bisimulación: se define una semántica operacional para el lenguaje, y se considera la semántica inducida por el *juego de la bisimulación*. Las extensiones temporizadas de *CCS* conservan también este rasgo; tienen por tanto una semántica basada en bisimulación.

Una primera versión temporizada de *CCS* la debemos a F. Miller y C. Tofts en [MT90]. El tiempo se introduce mediante el operador de prefijo $(t).P$, cuyo significado es el de retrasar la ejecución del proceso P exactamente t unidades de tiempo. Tienen también un operador de interbloqueo (0), que a diferencia del nuestro no permite el paso del tiempo. En este sentido se parece más a nuestro operador de divergencia. Existen dos operadores de elección: los dos eligen al estilo *CCS*, pero mientras uno de ellos deja al proceso en interbloqueo cuando una de sus componentes cae en interbloqueo, el otro descarta al proceso que está en interbloqueo. Todas las acciones (visibles o no) son consideradas urgentes. En consecuencia, para poder retrasar la ejecución de una acción dispone de operador (δ) que así lo permite. No obstante, se puede limitar el tiempo en el que una acción se puede ejecutar, mediante la combinación de todos estos operadores; es decir, se puede introducir un operador de *time-out* como derivado:

$$\delta.a.P + (t).\tau.0$$

Otra extensión temporal del lenguaje *CCS* la podemos encontrar en [Yi91a, Yi90, Yi91b]. El tiempo es introducido de manera parecida a como se hace en el trabajo anterior; es decir, mediante un operador de prefijo ($\epsilon(t).P$). En este trabajo las únicas acciones urgentes son la internas; las acciones visibles son *persistentes*: se pueden ejecutar en cualquier instante a partir del que están disponibles. Al igual que en el caso anterior se puede simular un operador de *time-out*. Este lenguaje ha seguido siendo estudiado en [AJ94] donde se presenta una axiomatización completa para el lenguaje, considerando procesos guardados; y en [Jef92] donde se discute la introducción de prioridades, resultando que el tiempo puede ser expresado utilizando exclusivamente prioridades.

8.3 Extensiones de *CSP*

Al igual que en los casos anteriores, la aproximación al problema de la definición de la semántica por los diversos autores,que han estudiado extensiones *temporizadas* de *CSP*, respeta la usada por los autores de la versión original *no temporizada.* En este caso se trata de una aproximación denotacional.

Empezaremos comentando los primeros trabajos de Reed y Roscoe [RR86, RR87, RR88] en los que se extienden los modelos de fallos al modelo temporizado. El dominio de tiempo que utilizan es contínuo, pero llama la atención el hecho de que tiene una constante δ a la que llaman *system delay constant*, de manera que después de cualquier llamada recursiva, así como después de la ejecución de cualquier acción, el sistema tiene que esperar como mínimo la cantidad indicada por dicha constante. Entonces ¿Para qué sirve el dominio de tiempo contínuo?

En este modelo se introduce en cada observación un instante de estabilidad; de ahí su nombre *timed failures-stability model.* El instante de estabilidad de un proceso después de ejecutar una traza, indica el tiempo máximo que el proceso dedica a desarrollar cierta actividad interna hasta que entra en un estado estable. En cierta manera, esta cantidad juega un papel dual a nuestro *tiempo de indefinición de una barba.* Una de las grandes diferencias con nuestro modelo es provocado por este hecho, puesto que por ejemplo los procesos

$$P = (a \to P) \setminus a \quad \text{y} \quad Q = \mathtt{STOP}$$

no son distinguibles mediante pruebas (una vez que tenemos en cuenta la constante de retraso), y sin embargo en su modelo tienen semánticas diferentes.

Existe un segundo modelo del propio Reed [Ree88], en el que se elimina este *instante de estabilidad.* Esto tiene como consecuencia que se identifique la divergencia con el interbloqueo. Aparentemente esto no tiene consecuencias sobre la continuidad del modelo, puesto que las llamadas recursivas siguen consumiendo tiempo.

Sobre este modelo podemos encontrar abundantes trabajos en la literatura. Nosotros citaremos aquí los que nos han parecido más representativos. En primer lugar mencionaremos [DS95] donde se da un repaso a la evolución del lenguaje *TCSP.* En [Sch95] se presenta una semántica operacional para el lenguaje, a partir de la cual se pueden computar los fallos temporales de un proceso; resulta que los fallos computados son los mismos

---

andaría

que los calculados mediante la semántica denotacional. Además, como trabajos en los que se presentan metodologías para llevar a la práctica los resultados del modelo podemos citar [Sch89] y [Dav91].

Dejando tierras británicas e internándonos en nuestro país, nos encontramos con el trabajo de Yolanda Ortega [OM91, OMdF90]. En el mismo nos aparece otra semántica de fallos diferente para una extensión temporizada de *CSP*. En este modelo el tiempo se introduce mediante la duración de las acciones: cada acción tiene una determinada duración, y el proceso no puede realizar nuevas acciones en secuencia hasta que no acabe su ejecución. No obstante, ello causa problemas con el operador paralelo: no es verdad que si un lado del proceso ejecuta una acción, el otro tiene que esperar a su compañero, entrando así en contradicción con la noción de concurrencia, pues se permite la ejecución de varias acciones solapada e simultánea. Así de forma similar a lo que ocurre (sin tiempo) en [TV89]: los procesos, en lugar de ejecutar secuencias de acciones simples, ejecutan secuencias de multiconjuntos de acciones.

8.4 Extensiones de *LOTOS*

Como una de las aplicaciones más prácticas de las álgebras de procesos nos encontramos con el lenguaje de especificación *LOTOS* [LOT88]. En las extensiones de *LOTOS* con tiempo, el mismo se introduce mediante el operador de prefijo: las acciones deben ejecutarse dentro de unos intervalos fijados. En cuanto a la semántica operacional se han seguidos dos; aproximaciones, por un lado la de [LL94] en la que hay transiciones de tiempo y transiciones con acciones, siguiendo la semántica operacional de *CCS* definida en [Yi91a]; y por otro lado la sugerida en [QMdL94] siguiendo a [QdFA93], en la que hay un único tipo de transiciones con acciones temporizadas del estilo de las utilizadas en el presente trabajo. Estas dos maneras de introducir la semántica operacional son de hecho totalmente equivalentes, como de una manera intuitiva vimos en el capítulo 2, y en concreto, para *LOTOS* temporizado se ha demostrado en [LR96].

Comentaremos por último la semántica denotacional para *LOTOS* temporizado presentada en [SBD95, BDS95] la cual sigue las ideas de las semánticas de fallos para *CSP*. Evidentemente, se han de tener en cuenta el instante en el que se realiza la primera acción interna, de otra manera sería imposible definir una semántica denotacional para un lenguaje con un operador de elección del tipo de la suma de *CCS*.

8.5 Semántica de Pruebas

En general las semánticas de pruebas siguen un camino diferente al recorrido por las semánticas discutidas en las secciones anteriores. Se parte de un álgebra de procesos, pero de hecho lo que nos interesa de ella es su semántica operacional, esto es el sistema de transiciones que induce. A continuación se incorporan las pruebas o test que tienen como misión la de comprobar las respuestas que dan los procesos al aplicárselas. Básicamente, una prueba suele ser un proceso normal en el que introducimos un proceso o una acción especial para señalar el éxito de la aplicación de la prueba. Al estar basada en el correspondiente sistema de transiciones es que en gran manera independiente del álgebra de procesos a la que se aplique. Así en tanto y cuando se mantengan las propiedades de la semántica operacional que tenga las mismas propiedades del álgebra original, podemos extender su definición a variantes de la misma. Así se puede se puede estudiar en un principio un álgebra de procesos sencilla, para después ampliar el lenguaje con operadores más complejos. El primer trabajo sobre semántica de pruebas es [dNH84], que se desarrolla en profundidad en [Hen88]. Otros tipos de semánticas de pruebas aparecen en [Phi87].

En cuanto a lo relativo a álgebras de procesos temporizadas, el único trabajo previo que conocemos es el de Hennessy y Regan [HR95], en el que se estudia la semántica de pruebas para de un lenguaje que denominan *ATP*. Este lenguaje lo podemos considerar como una extensión temporal de *CCS*; tiene los operadores clásicos de este lenguaje: la suma, la restricción (aunque con la misma sintaxis utilizada en *CSP* para el ocultamiento), y un operador paralelo que funciona como lo hace el de *CCS*. El tiempo está introducido de una forma que nos atrevemos a considerar como bastante primitiva, siendo similar a la utilizada en [NS94]. Así, existe una acción especial (σ) que expresa el paso de un ciclo de reloj, y se dispone de un operador de *time-out* $\lfloor P \rfloor Q$ en el que si pasa un ciclo de reloj sin que P ejecute ninguna acción, el proceso Q toma el control. En este trabajo se introducen las barbas, pues al ser las acciones internas urgentes, es necesario guardar la información acerca de las acciones ofrecidas antes de la ejecución de cualquier acción ejecutada.

8.6 ¿Qué hemos conseguido?

El último trabajo citado representa un primer paso de cara a estudiar la semántica de pruebas de álgebras de procesos temporizadas. Sin embargo, debido a la simplicidad con que se ha introdujo el tiempo, resulta poco adecuada como punto de partida a la hora de estudiar la semántica de pruebas de álgebras en las que el tiempo viene expresado de una manera más compleja, como sucede en el caso de *TE-LOTOS* [dFLL+95]. Este es

despacio

por tanto el punto en el que ha comenzado nuestro trabajo. Nuestro objetivo ha sido el estudio de un álgebra de procesos temporizada en la que el tiempo viene expresado de una manera más compleja, similar a la de las extensiones temporizadas de *LOTOS*.

En el presente trabajo hemos conseguido definir y caracterizar una semántica de pruebas de un álgebra de procesos de la que tenemos definida una semántica operacional con las siguientes características:

Urgencia: las acciones internas son urgentes. En concreto, la semántica operacional que consideramos tiene dos tipos de transiciones internas:

- $P \xrightarrow{\tau t} Q$
- $P \rightarrowtail Q$

La urgencia, viene expresada entonces por las condiciones:

- $P \rightarrowtail Q \quad \Rightarrow \quad \forall e \in \mathcal{E}\ \forall t \in \mathcal{T}: \ P \stackrel{et}{\not\longrightarrow}$
- $P \xrightarrow{\tau t} Q \quad \Rightarrow \quad \forall e \in \mathcal{E}\ \forall t' > t: \ P \stackrel{et}{\not\longrightarrow}$

Con respecto a la urgencia de las transiciones $\rightarrowtail$ podemos observar que hemos sido demasiado conservadores, puesto que todos los resultados obtenidos sería exactamente los mismos si cambiáramos la condición de urgencia de arriba por la siguiente

$$P \rightarrowtail Q \quad \Rightarrow \quad \forall e \in \mathcal{E}\ \forall t \in \mathcal{T}: \ (t \geq 0 \ \Rightarrow \ P \stackrel{et}{\not\longrightarrow})$$

No determinismo acotado o ramificación finita: Cada un proceso en cada instante sólo puede ejecutar un número finito de transiciones que nos conducen a otros tantos procesos *destino*.

Entendemos que los resultados presentados en este trabajo ser fácilmente extrapolables a cualquier álgebra de procesos con una semántica operacional con dichas características. Observemos que las transiciones del tipo $\rightarrowtail$ son necesarias únicamente en el caso que queramos modelar un operador de elección tipo *CCS*. Si no es así las transiciones $\rightarrowtail$ se pueden equiparar totalmente a transiciones del tipo $\xrightarrow{\tau 0}$.

Hemos conseguido además definir una semántica denotacional basada en las barbas, que induce una equivalencia entre procesos que resulta ser totalmente abstracta con respecto a la semántica de pruebas. Esta semántica denotacional está basada en un concepto adecuado de estados: cada procesos va ejecutando acciones, atravesando estados intermedios, tras lo cual se alcanza un estado final que describe la configuración actual del proceso. Ello supone en particular que no se *ven* como tales en ningún momento las acciones ocultas. Esto último provocaría los problemas vistos en el capítulo 7 si se pretende extender el modelo con un operador tipo la suma de *CCS*, que sí ha de *ver* la primera acción oculta ejecutada, pues la misma podría resolver una eventual la elección entre dos sumandos.

Finalmente, hemos introducido un sistema de ecuaciones e inecuaciones que resultan ser correctos y completos con respecto a la semántica denotacional, y puesto que ésta última es totalmente abstracta con respecto a la semántica de pruebas, el sistema será también completo y correcto con respecto a la misma. Fijémonos que hemos precisado de dos reglas que no son finitarias: [**APF**] y [**APT**]. La primera de ellas es la clásica para las aproximaciones finitas de un proceso; pero para caracterizar nuestro modelo denotacional necesitamos algo mas, al tratarse de puesto un dominio que no es algebraico. En concreto necesitamos la segunda regla citada, la cual intuitivamente indica que dos procesos son iguales si no se pueden distinguir en tiempo finito. Este última regla está íntimamente ligada con que el hecho de que el no determinismo sea acotado. Por último, observemos que si nos restringimos a procesos en los que antes de ejecutar una llamada recursiva se consuma tiempo, la segunda regla sería suficiente para demostrar la completitud del sistema. Una manera fácil de garantizar esto último, es comprobar que todas las definiciones recursivas están guardadas temporalmente. Naturalmente, ello es fácil garantizarlo por medio de oportunas restricciones sintácticas.

8.7 Trabajo a Desarrollar en un Futuro

8.7.1 Semántica *may*

En el presente trabajo hemos estudiado en profundidad la semántica *must*, que se define considerando que un proceso pasa una prueba si **todas** las computaciones maximales del proceso-prueba tienen éxito. Queda entonces por estudiar la semántica de tipo *may*: un proceso pasa una prueba si existe **una** computación que tiene éxito. Aunque todavía no disponemos de una prueba formal completa, todo apunta a que para, como de ordinarios, caracterizar la semántica *may* basta considerar las b-trazas de un proceso, de modo que se tenga

$$P \sqsubseteq_{\text{may}} Q \quad \Longleftrightarrow \quad \forall bs \in \mathsf{Btraz}(P)\ \exists bs' \in \mathsf{Btraz}(Q) :\ bs \ll bs'$$

Entonces, en ausencia de divergencias, la semántica *may+must* sería equivalente * a la semántica *must*, tal y como ocurría en el caso no temporizado.

*A nivel de la equivalencia inducida; no es así en lo referente al orden entre procesos.

hacia

Es interesante observar que en el caso temporizado la semántica *may* de nuestro lenguaje no es una congruencia, como muestra el siguiente

Ejemplo 8.7.1 Consideremos los procesos

$$P = a0 \,;\, \mathtt{STOP} \sqcap b0 \,;\, \mathtt{STOP} \quad \text{y} \quad Q = a0 \,;\, \mathtt{STOP} \,\Box\, b0 \,;\, \mathtt{STOP}$$

Es fácil comprobar que ambos procesos son equivalentes bajo la semántica *may*; pero si consideramos el proceso $R = c1 \,;\, \mathtt{STOP}$, resulta que los procesos $P \parallel_{\varnothing} R$ y $Q \parallel_{\varnothing} R$ no serían ya equivalentes, ni tampoco lo son los procesos $P \,\Box\, R$ y $Q \,\Box\, R$. Para distinguirlos basta considerar la prueba

$$T = a0 \,;\, \mathtt{STOP} \,\Box\, c1 \,;\, \mathtt{OK}$$

comprobándose fácilmente que

$$P \,\Box\, R \text{ may } T, \quad P \parallel_{\varnothing} R \text{ may } T, \quad Q \,\Box\, R \not\text{may } T \quad \text{y} \quad Q \parallel_{\varnothing} R \not\text{may } T$$

□

Deberíamos entonces estudiar la mayor congruencia contenida en la semántica *may*, que en el caso de que no haya divergencias, todo indica a pensar que es la propia semántica *must* invertida (si consideramos no sólo la equivalencia sino también el orden entre procesos).

8.7.2 Restricciones

En el lenguaje que hemos considerado hemos asumido una seria de restricciones necesarias para que las cosas funcionasen bien. Dichas restricciones consisten en:

- utilizar un dominio de tiempo discreto,
- utilizar un alfabeto finito de acciones y
- no determinismo acotado en la semántica operacional.

La mayor parte de las razones por las cuales hemos introducido dichas restricciones las encontramos en la semántica denotacional. Así, podemos observar que en muchas de las demostraciones hemos utilizado las mencionadas restricciones, las dos primeras explícitamente, y en lo que se refiere a la última hemos utilizado una consecuencia suya: la compacidad temporal de los estados. El eliminar estas restricciones no obligaría a complicar notablemente el modelo; en particular, todo indica a que para definir una la semántica denotacionales sería necesario utilizar dominio no contínuos. Hay diversos trabajos [RB89, Bar91, Ros93, Sch92, MRS95] que estudian la manera de definir semánticas denotacionales sobre dominios no completos, o tales que las funciones semánticas asociados a los operadores no sean contínuos.

Además de los citados problemas, si quisiéramos admitir un dominio de tiempo denso, deberíamos complicar la definición de la noción de convergencia. El lenguaje presentado es demasiado simple poder poner de manifiesto estos problemas, pero añadiendo otros operadores, se pueden definir procesos que diverjan antes, justo en, o después de un determinado instante:

- Supongamos que permitimos definiciones de procesos como la siguiente:

  $$\mathtt{Zenón}(t) = \tau(t/2)\,;\mathtt{Zenón}(t/2)$$

  Este proceso describe perfectamente el argumento de Zenón de Alea sobre la flecha que nunca llega a su destino. Entonces, el proceso $P = \mathtt{Zenón}(1)$ representaría un proceso que diverge *justo antes* del instante 1.

- Evidentemente, el proceso $Q = \tau 1\,;\mathtt{DIV}$ es un proceso que diverge *justo* en el instante 1.

- Supongamos que tenemos un proceso del estilo

  $$\tau(I)\,;P$$

  donde I es un cierto intervalo abierto o cerrado. Este proceso elegiría de manera no determinista un instante t del intervalo tras lo cual se transformaría en el proceso $\tau t\,;P$. Para no vulnerar las reglas de la urgencia ello se podría hacer incorporando la regla

  $$\tau(I)\,;P \rightarrowtail \tau t\,;P \qquad t \in I$$

  Si consideráramos entonces el intervalo abierto $I = (1,2]$, tenemos que entonces el proceso

  $$R = (1,2]\,;\mathtt{DIV}$$

  diverge *justo después* del instante 1.

Tenemos además que estos tres procesos se podrían distinguir fácilmente mediante pruebas. Para ello es necesario (y suficiente) modificar ligeramente la sintaxis de las pruebas de manera que sea ahora OK una nueva acción en lugar de un proceso. Entonces la prueba OK vendría dada por OK0 ; STOP. Entonces la condición que define el paso de una prueba sería $T \xrightarrow{\mathtt{OK0}}$. En toda esta argumentación, a pesar de la citada variante sintáctica, no hemos

una

introducido nada *extraño*. De hecho inicialmente consideramos dicha vía como alternativa posible en el trabajo. Entonces, si consideráramos las pruebas

$$T_1 = \mathtt{OK1}\,;\mathtt{STOP} \quad \text{y} \quad T_2 = \tau 1\,;\mathtt{OK}$$

tenemos

$$\begin{array}{ll} P \,\not{\text{m}}\text{ust}\, T_1 & P \,\not{\text{m}}\text{ust}\, T_2 \\ Q \text{ must } T_1 & Q \,\not{\text{m}}\text{ust}\, T_2 \\ R \text{ must } T_1 & R \text{ must } T_2 \end{array}$$

Con lo cual vemos que en efecto los tres procesos se pueden distinguir entre sí. Si bien nada de lo que hemos hecho lo podríamos hacer con la sintaxis original del lenguaje, sí se trata de extensiones razonables, y de hecho un posible interés para trabajar con dominios de tiempo denso difícilmente quedaría justificado si no se introducen dichas extensiones en la sintaxis del lenguaje. Así en particular los procesos que hemos introducido están presentes en la sintaxis de las extensiones temporizadas de *LOTOS*.

8.7.3 Nuevos Operadores

La introducción de nuevos operadores *razonables* no debería causar, en principio, ningún tipo de problemas, siempre y cuando respeten las características semánticas del lenguaje. En concreto, en lo referente a la semántica de pruebas no debería haber ningún tipo de problemas. La definición de las semántica denotacional, dependiendo del operador, podría complicarse algo más. Sin embargo en la mayoría de los trabajos sobre álgebras de procesos temporizadas se observa que los operadores adicionales que usualmente se consideran se podrían definir en función de los operadores que tenemos definidos en este trabajo.

Existe sin embargo un operador, que aparece en las propuestas de las extensiones temporizadas de *LOTOS*, que no conserva las propiedades exigidas. Se trata del operador de prefijo mediante una acción interna con un intervalo de tiempo

$$\tau[t_1, t_2]\,; P$$

La propuestas para *TE-LOTOS* que existen actualmente provocan que **no** todas las acciones ocultas sean ya urgentes, pues para el citado operador se tiene la regla

$$\tau[t_1, t_2]\,; P \xrightarrow{\tau t} P \quad \text{para} \quad t_1 \le t \le t_2$$

Todas las acciones ocultas que vienen de un ocultamiento siguen siendo urgentes, pero las correspondientes al nuevo operador de prefijo no. Ello tiene consecuencias extrañas. En particular, la equivalencia de bisimulación débil no implica la semántica de pruebas.

Ejemplo 8.7.2 Consideremos los procesos

$$P = \tau[0,5]\,;a[0,\infty]\,;\mathtt{STOP} \quad \text{y} \quad Q = a[0,\infty]\,;\mathtt{STOP}$$

Bajo cualquiera de las nociones de bisimulación débil propuestas hasta el momento, ambos procesos serían equivalentes. Sin embargo son fácilmente distinguibles mediante la prueba

$$T = a0\,;\mathtt{OK}$$

ya que P $\not\text{must}$ T, pero Q must T.

□

Otro efecto de la introducción de este operador (con la semántica operacional que se propone) es que el operador de ocultamiento no es congruente con respecto a la bisimulación débil:

Ejemplo 8.7.3 Los procesos

$$\tau[0,5]\,;a[0,\infty]\,;b0\,;\mathtt{STOP} \quad \text{y} \quad a[0,\infty]\,;b0\,;\mathtt{STOP}$$

son débilmente bisimilares bajo cualquiera de las nociones de bisimulación definidas hasta la fecha, pero sin embargo $P \setminus a$ y $Q \setminus a$ no lo son puesto que el primero tiene como única posibilidad la de ejecutar la acción b en el instante 0, mientras que el segundo puede ejecutar la acción b en cualquier instante entre 0 y 5.

□

Una posibilidad que existe para resolver este problema, consistiría en modificar la definición de la semántica operacional como se indica en [LdFN95], tomando

$$\tau[t_1,t_2]\,;P \rightarrowtail \tau t\,;P \quad \text{para} \quad t_1 \leq t \leq t_2$$

Esta solución conserva todas las propiedades originales del lenguaje, siempre y cuando nos restrinjamos a un dominio de tiempo discreto, y el extremo superior del intervalo t_2 no sea infinito. Para solventar los problemas que causaría el hecho de que el extremo superior fuese infinito, se podría añadir la regla

$$\tau[t_1,\infty]\,;P \rightarrowtail \mathtt{STOP}$$

Nos parece ser que tras añadir esta regla se podrían reconstruir las demostraciones, de modo que las cosas seguirían funcionando bien.

fuente....

8.7.4 Ampliaciones del lenguaje

En este apartado discutimos brevemente la posibilidad de introducir nuevas características al lenguaje. Entre las características que se podrían añadir se cuentan las siguientes

- Introducir duración a las acciones. Esto parece fácil de hacer añadiendo una regla para el prefijo del estilo

$$at\,;P \xrightarrow{at} \tau d(a)\,;P$$

  donde $d(a)$ sería la duración de la acción a.

- Añadir al lenguaje características de paralelismo real, de manera que los procesos

$$a0\,;\texttt{STOP} \parallel_{\varnothing} b0\,;\texttt{STOP} \quad \text{y} \quad (a0\,;b0\,;\texttt{STOP}) \,\Box\, (b0\,;a0\,;\texttt{STOP})$$

  dejen de ser equivalentes. Un resumen de los modelos propuestos para dotar a un álgebra de procesos no temporizada con esta característica lo podemos encontrar en [Lla93]. Otra alternativa no contemplada en este trabajo, la podemos encontrar en [Hen92], en la que se divide la ejecución de cada acción entre su comienzo y su terminación.

- Añadir al lenguaje elecciones probabilísticas y prioritarias. Las álgebra de procesos probabilísticas han sido ampliamente estudiadas en los últimos años, existiendo numerosas propuestas [Chr90, GJS90, NdFL95, CdV96]. Un trabajo introductorio cara a definir una semántica de pruebas en una álgebra de procesos con tiempo y probabidades es [Gre95]. En dicho modelo el tiempo ha sido introducido como en [NS94, HR95].

  También podemos encontrar en la literatura numerosos trabajos acerca de elecciones prioritarias, pudiendo citar [SS90, CH90, CW95]. Relacionando tiempo y prioridades, en [Jef92], se comenta que, trabajando bajo una semántica de bisimulación,el tiempo se puede representar como un factor de prioridad.

  Relacionando los tres temas, tiempo, probabilidades y prioridades, aunque el resultado no lo consideramos suficientemente satisfactorio, nos encontramos con [Low93], donde se extiende el *CSP* temporizado introduciendo también probabilidades y prioridades.

Apéndice A

Demostraciones

En este apéndice están desarrolladas en su las demostraciones que hemos ido dejando pendientes a lo largo del trabajo.

A.1 Demostraciones del Capítulo 3

Demostración de la proposición 3.6.2.
La computación $P \xRightarrow{bs'} P_1$ se puede deducir en un número finito de pasos, pongamos n. Análogamente la computación $T \xRightarrow{bs} T_1$ se deducirá en m pasos. La demostración se hace entonces por inducción sobre $m + n$.

m + n = 0 En tal caso se tiene $P_1 = P$ y $T_1 = T$.

m + n > 0 Distinguiremos varios subcasos según sea la forma de la primera transición de las computaciones generando ambas b-trazas.

- $P \rightarrowtail P' \xRightarrow{bs'} P_1$. Entonces por hipótesis de inducción tenemos que hay una computación desde $P' \mid T$ hasta $P_1 \mid T_1$. También sabemos que $P \mid T \mapsto P' \mid T$, por lo que hay una transición desde $P \mid T$ hasta $P_1 \mid T_1$.
- $T \rightarrowtail T' \xRightarrow{bs} T_1$. Este caso es totalmente simétrico al anterior
- $P \xrightarrow{\tau t} P' \xRightarrow{bs'_1} P_1$, $\mathsf{idle}(T) \geq t$ y $\mathsf{stb}(T)$. En este caso tenemos

$$bs'_1 \neq \epsilon \quad \text{y} \quad bs' = (\mathsf{TA}(P) \upharpoonleft t, t) \sqcup bs'_1$$

Puesto que $\mathsf{stb}(T)$ e $\mathsf{idle}(T) \geq t$, obtenemos

$$bs = (\mathsf{TA}(T) \upharpoonleft t, t) \sqcup bs_1 \quad \text{para} \quad bs_1 = (A_1 - t)a_1(t_1 - t) \cdots A_n a_n t_n$$

El Principito,

Puesto que $A_1 \cap A_1' = \varnothing$ tenemos $\mathsf{TA}(P) \upharpoonright t \cap \mathsf{TA}(T) \upharpoonright t = \varnothing$, y por tanto

$$P \mid T \mapsto P' \mid \mathsf{Upd}(P,t)$$

Ahora por el lema 3.6.1, tenemos que

$$\mathsf{Upd}(T,t) \xRightarrow{bs_1} T_1 \quad \text{y} \quad P' \xRightarrow{\overline{bs_1'}} P_1$$

Las dos b-trazas obtenidas bs_1 y bs_1' cumplen las condiciones del enunciado, por lo que podemos aplicar la hipótesis de inducción a P' y $\mathsf{Upd}(T,t)$ obteniendo una computación desde $P' \mid \mathsf{Upd}(T,t)$ hasta $P_1 \mid T_1$ que genera una transición desde $P \mid T$ hasta $P_1 \mid T_1$.

- $T \xrightarrow{\tau t} T' \xRightarrow{bs_1} T_1$, $\mathsf{idle}(P) \geq \mathsf{idle}(P) = t$ y $P \not\rightarrowtail$. Este caso es totalmente simétrico al anterior.
- $P \xrightarrow{at} P'$ y $T \xRightarrow{at} T'$. Puesto que $A_1 \cap A_1' = \varnothing$ tenemos que la transición $P \mid T \xrightarrow{at} P' \mid T'$ es posible, lo que inmediatamente nos conduce al resultado deseado aplicando inducción.

□

Demostración de la proposición 3.6.3.
Aplicando la proposición 3.4.5 obtenemos la computación

$$P = P_0 \rightarrowtail^* P_0' \xrightarrow{\tau t_1^P} P_1 \rightarrowtail^* \cdots \xrightarrow{\tau t_l^P} P_l \rightarrowtail^* P_l' \cdots$$

que genera el estado A. Puesto que $\mathsf{nd}(A) \geq t^k$ existe un l verificando

$$\sum_{i=1}^{l} t_i^P \leq t^k \quad \wedge \quad \left(P_l \Downarrow \wedge P_l' \xrightarrow{\tau t_{l+1}} \quad \Rightarrow \quad \sum_{i=1}^{l+1} t_i^P > t^k \right) \quad \wedge \quad \forall 1 \leq i \leq l : \ P_i \Downarrow$$

Supongamos que la computación desde T hasta T_k tiene una longitud t, y la de P hasta P_l tiene una longitud p. Demostraremos entonces que hay una computación desde $P \mid T$ hasta $P_k \mid T_k$, por inducción sobre $t + p$.

$\mathbf{t + p = 0}$ En este caso se tiene que $P_1 = P$ y $T_1 = T$.

$\mathbf{t + p > 0}$ Distinguiremos varios subcasos, según sea el primer paso de las respectivas computaciones en juego.

- $P \rightarrowtail P'$. Tenemos que $P \mid T \rightarrowtail P' \mid T$, por lo que se obtiene el resultado deseado aplicando la hipótesis de inducción.

- $T \rightarrowtail T'$. Al igual que en el caso anterior podríamos aplicar de forma inmediata de la hipótesis de inducción.
- $P \xrightarrow{\tau t} P'$, $\mathsf{idle}(T) \geq \mathsf{idle}(P) = t$ y $\mathsf{stb}(T)$. Como $A \cap \mathsf{TA}(T) \upharpoonright t = \varnothing$, tenemos

  $$\mathsf{TA}(P) \upharpoonright t \cap \mathsf{TA}(T) \upharpoonright t = \varnothing$$

  y por tanto

  $$P \mid T \mapsto P' \mid \mathsf{Upd}(T, t)$$

  Entonces para obtener el resultado deseado basta aplicar la hipótesis de inducción.
- $T \xrightarrow{\tau t} T'$, $\mathsf{idle}(P) \geq \mathsf{idle}(T) = t$ y $\mathsf{stb}(P)$. Este caso es simétrico al anterior.

$\square$

Demostración del teorema 3.6.5.
A lo largo de esta demostración usaremos el orden lexicográfico sobre $\mathbb{N} \times \mathbb{N}$ dado por:

$$(n_1, n_2) \leq (m_1, m_2) \quad \Longleftrightarrow \quad (n_1 < m_1 \vee (n_1 = m_1 \wedge n_2 \leq m_2))$$

Supongamos que P must T, para demostrar que Q must T consideremos una computación cualquiera de $Q \mid T$,

$$Q \mid T = Q_1 \mid T_1 \mapsto Q_2 \mid T_2 \mapsto \cdots Q_k \mid T_k \mapsto Q_{k+1} \mid T_{k+1} \cdots$$

Hay que demostrar que la misma tiene éxito. Podemos descomponer dicha computación $Q \mid T$ se puede en una computación de Q y otra de T. Tomando $Q_{11} = Q$ y $T_{11} = T$, tendríamos

$$\begin{array}{l}
Q_{11} \rightarrowtail^* Q'_{11} \xrightarrow{\tau t^Q_{11}} Q_{12} \rightarrowtail^* Q'_{12} \xrightarrow{\tau t^Q_{12}} \cdots \xrightarrow{\tau t^Q_{1\,n_1-1}} Q_{1\,n_1} \rightarrowtail^* Q'_{1\,n_1} \xrightarrow{a_1 t^Q_{1\,n_1}} \\
Q_{21} \rightarrowtail^* Q'_{21} \xrightarrow{\tau t^Q_{21}} Q_{22} \rightarrowtail^* Q'_{s2} \xrightarrow{\tau t^Q_{22}} \cdots \xrightarrow{\tau t^Q_{2\,n_2-1}} Q_{2\,n_2} \rightarrowtail^* Q'_{2\,n_2} \xrightarrow{a_2 t^Q_{2\,n_2}} \\
\qquad\qquad\qquad \vdots \qquad\qquad\qquad \ddots \qquad\qquad\qquad \vdots \\
Q_{k1} \rightarrowtail^* Q'_{k1} \xrightarrow{\tau t^Q_{k1}} Q_{k2} \rightarrowtail^* Q'_{k2} \xrightarrow{\tau t^Q_{k2}} \cdots \xrightarrow{\tau t^Q_{k\,n_k-1}} Q_{k\,n_k} \rightarrowtail^* Q'_{k\,n_k} \xrightarrow{a_k t^Q_{k\,n_k}} \\
\qquad\qquad\qquad \vdots \qquad\qquad\qquad \ddots \qquad\qquad\qquad \vdots \\
\\
T_{11} \rightarrowtail^* T'_{11} \xrightarrow{\tau t^T_{11}} T_{12} \rightarrowtail^* T'_{12} \xrightarrow{\tau t^T_{12}} \cdots \xrightarrow{\tau t^T_{1\,m_1-1}} T_{1\,m_1} \rightarrowtail^* T'_{1\,m_1} \xrightarrow{a_1 t^T_{1\,m_1}} \\
T_{21} \rightarrowtail^* T'_{21} \xrightarrow{\tau t^T_{21}} T_{22} \rightarrowtail^* T'_{22} \xrightarrow{\tau t^T_{22}} \cdots \xrightarrow{\tau t^T_{2\,m_2-1}} T_{2\,m_2} \rightarrowtail^* T'_{2\,m_2} \xrightarrow{a_2 t^T_{2\,m_2}} \\
\qquad\qquad\qquad \vdots \qquad\qquad\qquad \ddots \qquad\qquad\qquad \vdots \\
T_{k1} \rightarrowtail^* T'_{k1} \xrightarrow{\tau t^T_{k1}} T_{k2} \rightarrowtail^* T'_{k2} \xrightarrow{\tau t^T_{k2}} \cdots \xrightarrow{\tau t^T_{k\,m_k-1}} T_{k\,m_k} \rightarrowtail^* T'_{k\,m_k} \xrightarrow{a_k t^T_{k\,m_k}} \\
\qquad\qquad\qquad \vdots \qquad\qquad\qquad \ddots \qquad\qquad\qquad \vdots
\end{array}$$

Puesto que a lo largo de la computación, la prueba y el proceso evolucionan de forma sincronizada, el instante en el que se realice cada acción a_k en el proceso y en la prueba ha de ser el mismo. Por tanto so notamos a dicho instante por t_k, se ha de verificar que para todo $k \geq 1$

$$t_k = \sum_{i=1}^{n_k} t_{ki}^Q = \sum_{i=1}^{m_k} t_{ki}^T$$

Fijado $k \geq 1$, obtenemos entonces una b-traza de Q

$$bs^Q = A_1^Q a_1 t_1 A_2^Q a_2 t_2 \cdots A_k^Q a_k t_k \text{siendo} A_l^Q = \bigcup_{i=1}^{n_l} \left((\mathsf{TA}(Q'_{li}) \uparrow t_{li}^Q) + \sum_{j=1}^{i-1} t_{lj}^Q \right)$$

y, análogamente, otra b-traza de T

$$bs^T = A_1^T a_1 t_1 A_2^T a_2 t_2 \cdots A_k^T a_k t_k \text{con} A_l^T = \bigcup_{i=1}^{m_l} \left((\mathsf{TA}(T'_{li}) \uparrow t_{li}^T) + \sum_{j=1}^{i-1} t_{lj}^T \right)$$

T puesto que la computación es posible, ha de suceder además que los procesos Q's y las pruebas T's no pueden sincronizar en cada ocasión acciones correspondientes a instantes anteriores a los t_i's. Es decir, para todo l se tendrá

$$A_l^Q \cap A_l^T = \varnothing$$

Puesto que $Q \xRightarrow{bs^Q} Q_{k1}$, tomando como A^Q un estado cualquiera de Q_{k1}, tendremos

$$b^Q = bs^Q \cdot A^Q \in \mathsf{Barb}(Q)$$

Entonces, como quiera que $\mathsf{Barb}(P) \ll \mathsf{Barb}(Q)$, existirá $b^P = bs^P \cdot A^P \in \mathsf{Barb}(P)$ tal que $b^P \ll b^Q$. Existe entonces un proceso P' tal que

$$P \xRightarrow{bs^P} P' \quad \text{con} \quad A^P \in \mathcal{A}(P')$$

Supongamos ahora que la longitud de la b-traza bs^P es estrictamente menor que la longitud de la b-traza bs^Q, es decir que se tiene

$$bs^P = A_1^P a_1 t_1 A_s^P a_2 t_2 \cdots A_r a_r t_r \quad \text{con} \quad r < k$$

Puesto que $b^P \ll b^Q$, tenemos $A_i^P \subseteq A_i^Q$ para cada i tal que $1 \leq i \leq r$; además, tomando $t_{r+1}^P = \mathsf{nd}(A^P)$, se verifica

$$t_{r+1}^P \leq t_{r+1}^Q \quad \text{y} \quad TAct(A^P) \subseteq A_{r+1}$$

Ya que $A_i^P \subseteq A_i^Q$ y $A_i^Q \cap A_i^T = \varnothing$, en virtud de la proposición 3.6.2 existirá una computación desde $P \,|\, T$ hasta $P' \,|\, T_{r1}$, y por la proposición 3.6.4 existe una computación desde $P' \,|\, T_{r1}$

hasta $P'' \mid T''$ para ciertos P'' y T'' con $P'' \Uparrow$. Además, todas las pruebas que aparecen en dicha computación se encuentran entre las originales. Puesto que P must T, habrá alguna de dichas pruebas que concede el visto bueno inmediato, y por tanto la computación original tiene éxito. Con lo cual podremos suponer en cada caso sucesivo que la longitud de la b-traza bs^P es la misma que la de la b-traza bs^Q, y por tanto razonando como antes, por aplicación la proposición 3.6.2 existirá una una computación desde $P \mid T$ hasta el correspondiente $P' \mid T_{k1}$.

Entonces, dependiendo de la forma de la computación de Q, podemos distinguir los siguientes casos:

- Existe $(k, i) \in \mathbb{N} \times \mathbb{N}$ de forma que $Q_{ki} \Uparrow$. Tomemos entonces el primer par que lo cumple. Sin pérdida de generalidad podemos suponer que ninguna prueba anterior a T_{k1} da el visto bueno, pues en caso contrario podríamos concluir inmediatamente. habríamos acabado. Consideramos entonces el estado

$$A^Q = \{\Omega t^Q\} \cup \bigcup_{j=1}^{i-1} \left((\mathsf{TA}(Q'_{kj} \upharpoonright t^Q_{kj}) + \sum_{l=1}^{j-1} t^Q_{kl} \right)$$

donde $t^Q = \sum_{j=1}^{i-1} t^Q_{kj}$. Tenemos que $A^Q \in \mathcal{A}(Q_{k1})$, y como quiera que $P \ll Q$, existe una barba $bs^P \cdot A^P \in \mathsf{Barb}(P)$ tal que $bs^P \cdot A^P \ll bs^Q \cdot A^Q$. Repitiendo el razonamiento de antes tenemos que $\mathsf{lon}(bs^P) = \mathsf{lon}(bs^Q)$, con lo que $bs^P \ll bs^Q$ y $A^P \ll A^Q$. Existe entonces un proceso P_{k1}

$$P \xRightarrow{bs^P} P_{k1}, \quad \text{y} \quad A^P \in \mathcal{A}(P_{k1})$$

por lo tanto tenemos que existe una computación desde $P \mid T$ hasta $P_{k1} \mid T_{k1}$. Por la proposición 3.6.3, para cada l tal que $\mathsf{nd}(A^P) > \sum_{j=1}^{l-1} t^T_{kj}$, existe P_{kl} tal que hay una computación desde $P_{k1} | T_{k1}$ hasta $P_{kl} | T_{kl}$. Si existe un T_{kl} con $\mathsf{nd}(A^P) > \sum_{j=1}^{l-1} t^T_{kj}$ tal que $T_{kl} \xrightarrow{\mathtt{OK}}$, puesto que $\mathsf{nd}(A^Q) \geq \mathsf{nd}(A^P)$, se tiene que la computación de $Q | T$ tiene éxito. Supongamos por tanto que ningún T_{kl} con $\mathsf{nd}(A^P) < \sum_{j=1}^{l} t^T_{kj}$ cumple que $T_{kl} \xrightarrow{\mathtt{OK}}$. Si el número de pruebas que cumple dicha condición es finito encontramos fácilmente una computación sin éxito de $P \mid T$, lo cual contradice la hipótesis. Si por el contrario, el cantidad de pruebas que cumple la condición es infinita, por aplicación de la proposición 3.2.5 y el lema de König, encontramos una computación infinita sin éxito de $P_{k1} \mid T_{k1}$ por lo que tendríamos P $\cancel{\text{must}}$ T, en contradicción con nuestra hipótesis. De modo que ha de existir T_{kl} con $\mathsf{nd}(A^P) < \sum_{j=1}^{l} t^T_{kj}$ tal que $T_{kl} \xrightarrow{\mathtt{OK}}$.

Antonie de Saint-Exupery.

- Para todo par $(k,i) \in \mathbb{N} \times \mathbb{N}$ se cumple que $Q_{ki} \Downarrow$, siendo la computación finita. Consideremos el último proceso Q'_{ki} que aparece en la computación. Podemos suponer sin pérdida de generalidad que ninguna prueba anterior a T_{k1} da el visto bueno, pues si no fuera así ya habríamos concluido. Consideremos el estado

$$A^Q = \bigcup_{j=1}^{i-1} \left((\mathsf{TA}(Q'_{kj} \mid t^Q_{kj}) + \sum_{l=1}^{j-1} t^Q_{kl} \right)$$

Al igual que antes, puesto que $P \ll Q$, existen un proceso P_{k1}, una b-traza bs^P y un estado A^P tales que

$$P \xRightarrow{bs^P} P_{k1}, \quad bs^P \ll bs^Q, \quad A^P \in \mathcal{A}(P_{k1}) \quad \text{y} \quad A^P \ll A^Q$$

Por la proposición 3.6.3 para cada T_{kl} tal que $\mathsf{nd}(A^P) > \sum_{j=1}^{l} t^T_{kj}$, existe P_{kl} tal que hay una computación desde $P_{k1} \mid T_{k1}$ hasta $P_{kl} \mid T_{kl}$. Y al igual que en el apartado anterior concluimos que ha que haber una prueba T_{kl} tal que $T_{kl} \xrightarrow{\mathtt{OK}}$.

- Para todo par $(k,i) \in \mathbb{N} \times \mathbb{N}$ se cumple que $Q_{ki} \Downarrow$ siendo la computación infinita. En este caso hay dos posibilidades:

  - Existe un $k \in \mathbb{N}$ de manera que la computación infinita no realiza ninguna acción visible después de ejecutar $a_{k-1}t_{k-1}$. En consecuencia una computación infinita

$$Q_{k1} \rightarrowtail^* Q'_{k1} \xrightarrow{\tau t^Q_{k1}} Q_{k2} \rightarrowtail^* Q'_{k2} \xrightarrow{\tau t^Q_{k2}} \cdots$$
$$\cdots \xrightarrow{\tau t_{k\,l-1}} Q_{k\,l} \rightarrowtail^* Q'_{k\,l} \xrightarrow{\tau t_{k\,l}} \cdots$$

  Sin pérdida de generalidad podemos suponer que ninguna prueba anterior a T_{k1} da el visto bueno, pues si no fuera así ya habríamos concluido. Consideremos entonces el estado

$$A^Q = \bigcup_{j \in \mathbb{N}} \left((\mathsf{TA}(Q'_{kj} \mid t^Q_{kj}) + \sum_{l=1}^{j-1} t^Q_{kl} \right)$$

  Al igual que razonamos anteriormente, puesto que $P \ll Q$, existirán un proceso P_{k1}, una b-traza bs^P y un estado A^P tales que

$$P \xRightarrow{bs^P} P_{k1}, \quad bs^P \ll bs^Q, \quad A^P \in \mathcal{A}(P_{k1}) \quad \text{y} \quad A^P \ll A^Q$$

  Por la proposición 3.6.3 para cada T_{kl} tal que $\mathsf{nd}(A^P) > \sum_{j=1}^{l} t^T_{kj}$, existe P_{kl} tal que hay una computación desde $P_{k1} \mid T_{k1}$ hasta $P_{kl} \mid T_{kl}$. Y al igual que en los apartados obtendríamos una prueba T_{kl} tal que $T_{kl} \xrightarrow{\mathtt{OK}}$.

– La última posibilidad es que exista una secuencia infinita de b-trazas

$$bs_k^Q = A_1^Q a_1 t_1 A_2^Q a_2 t_2 \cdots A_k^Q a_k t_k \in \mathsf{Btraz}(Q), \qquad k \in \mathbb{N}$$

En tal caso, como vimos antes, para cada $k \in \mathbb{N}$ existirán P_k y una b-traza bs_k^P tales que

$$P \xRightarrow{bs_k^P} P_k \quad \text{y} \quad bs_k^P \ll A_1^Q a_1 t_1 A_2^Q a_2 t_2 \cdots A_k^Q a_k t_k$$

Entonces en virtud de la proposición 3.6.2, hay una computación desde $P \mid T$ hasta $P_k \mid T_{k1}$. Si existe un par $(k, i) \in \mathbb{N} \times \mathbb{N}$ tal que $T_{ki} \xrightarrow{\mathtt{OK}}$, entonces tenemos que la computación de $Q \mid T$ también tiene éxito. Por el contrario, si ningún $T_{ki} \xrightarrow{\mathtt{OK}}$, por la proposición 3.2.5 y el lema de König, existe una computación infinita de $P \mid T$ sin éxito, por lo que P m~~u~~st T, lo que contradice nuestra hipótesis.

Puesto que, en todos los casos hemos concluido que la computación de $Q \mid T$ tiene éxito, queda probado que Q must T.

□

Demostración de la proposición 3.6.11.

Hemos de probar que P must T; para ello hemos de demostrar que cualquier computación de $P|T$ es de éxito. Siendo $b = A_1 a_1 t_1 \cdots A_n a_n t_n A$, haremos la demostración por inducción sobre n, la longitud de b.

n = 0 En este tal tenemos que $b = A$ y T es de la forma:

$$T = [A'] \, ; \mathtt{OK} \,\square \begin{cases} \tau t \, ; \mathtt{OK} & \text{si } t = \mathsf{nd}(A) < \infty \\ \mathtt{STOP} & \text{en otro caso} \end{cases}$$

Cualquier computación de $P \mid T$ proviene de una computación de P

$$P = P_1 \rightarrowtail^* P_1' \xrightarrow{et_1} P_2 \rightarrowtail^* P_2 \xrightarrow{et_2} \cdots$$

Si alguno de los eventos que aparecen es un evento visible, necesariamente tiene que sincronizar con la prueba, y entonces ésta dará el visto bueno y por tanto tendremos una computación con éxito. Podemos suponer entonces que todos los eventos corresponden a acciones ocultas. Distinguiremos dos casos

- Existen $k \in \mathbb{N}$ y un proceso P' tales que $P'_{k-1} \xrightarrow{\tau t_{k-1}} P'$ y $P' \Uparrow$. Supongamos que k es el primer entero que cumple esa condición, tomemos el estado A_1 de P

$$A_1 = \{\Omega t^k\} \cup \bigcup_{i=1}^{k-1} ((\mathsf{TA}(P_i') \upharpoonright t_i) + t^i)$$

donde $t^i = \sum_{j=1}^{i-1} t_i$ para $1 \leq i \leq k$. De nuevo distinguiremos dos subcasos:

1. Existe $a_1t_1 \in A' \cap A_1$. Tomaremos a_1t_1 de manera que t_1 sea mínimo. Existirá entonces P_j' tal que $a_1(t_1 - t^j) \in \mathsf{TA}(P_j') \upharpoonright t_j$, y por tanto la tendremos la computación

$$P \mid T \mapsto^* P_1' \mid T \mapsto P_2 \mid \mathsf{Upd}(T, t^2) \cdots$$
$$\cdots P_j \mid \mathsf{Upd}(T, t^j) \mapsto^* P_j' \mid \mathsf{Upd}(T, t^j) \mapsto P'' \mid \mathsf{OK}$$

por lo que la computación original tiene éxito.

2. $A_1 \cap A' = \varnothing$. En este caso tenemos que $\mathsf{nd}(A) > t$ y por tanto $t^k > t$. Ha de existir entonces un entero j con $1 \leq j < k$ tal que $t^j \leq t$ pero $t^{j+1} > t$. Podemos entonces construir una computación desde $P \mid T$ hasta $P_j' \mid \mathsf{Upd}(T, t^j)$. Tenemos que $T \xrightarrow{\tau t} \mathsf{OK}$, y puesto que $t^j \leq t$ se obtiene

$$\mathsf{Upd}(T, t^j) \xrightarrow{\tau(t-t^j)} \mathsf{OK}$$

Como $t^{j+1} > t$, tenemos que $t_j > t - t^j$, y por tanto se tiene necesariamente

$$P_j' \mid \mathsf{Upd}(T, t^j) \mapsto \mathsf{Upd}(P_j', t - t^j) \mid \mathsf{OK}$$

y por tanto obtenemos de nuevo una computación de éxito.

- Supongamos que para todo $k > 1$ y todo P' tal que $P_{k-1} \xrightarrow{\tau t_{k-1}} P'$ verifica $P' \Downarrow$. Tenemos entonces una computación

$$P = P_1 \rightarrowtail^* P_1' \xrightarrow{\tau t_1} P_2 \rightarrowtail^* P_2' \cdots$$

Como antes tomamos $t^i = \sum_{j=1}^{i-1} t_j$. La computación puede ser finita (en cuyo caso notaremos por P_n' su último proceso), o bien infinita. En cualquier caso dará lugar a un estado A_1, cumpliéndose

$$\underset{\text{Si el cómputo es finito}}{A_1 = \mathsf{TA}(P_n') \cup \bigcup_{i=1}^{n-1} ((\mathsf{TA}(P_i) \upharpoonright t_i) + t^i)} \qquad \underset{\text{Si el cómputo es infinito}}{A_1 = \bigcup_{i \in \mathbb{N}} ((\mathsf{TA}(P_i') \upharpoonright t_i) + t^i)}$$

Si existiese $at \in A_1 \cap A'$, se puede razonar como en la primera parte del apartado anterior, para concluir que P must T. Supongamos entonces que $A_1 \cap A = \varnothing$; se tiene entonces que $\mathsf{nd}(A) = \infty$, y ahora podemos razonar como en el segundo caso del apartado anterior, para concluir que P must T.

Puesto que en cualquier caso obtenemos que la computación de $P \mid T$, que es una cualquiera, tiene éxito, podemos concluir que P must T.

n > 0 Sea $b = Aat \cdot b'$. Tomaremos entonces

$$\mathcal{A} = \left\{ A' \middle| \begin{array}{l} A' \in B \text{ y } \mathsf{nd}(A) < t, \text{ o} \\ A'at \cdot b' \in B \text{ y } A' \not\subseteq A \end{array} \right\}$$

$$B_1 = \{b'_1 \mid A'_1 at \cdot b'_1 \in B \text{ y } A'_1 \subseteq A\}$$

$$T = ([A_1]\,;\mathtt{OK}) \,\Box\, (at\,;T_1) \,\Box\, (\tau t\,;\mathtt{OK})$$

donde la prueba T_1 y el conjunto $A_1 \subseteq TAct$ verifican:

- si $B_1 = \varnothing$ entonces $T_1 = \mathtt{STOP}$, y en caso contrario T_1 es una prueba bien formada con respecto a B_1 y b',
- $A_1 \cap A' \neq \varnothing$ para todo $A' \in \mathcal{A}$.

Consideremos el conjunto de procesos

$$\mathcal{Q} = \left\{ Q \middle| P \stackrel{A'at}{\Longrightarrow} Q,\ A' \subseteq A \right\}$$

Si $\mathcal{Q}$ no fuera finito, por la propiedad 3.2.5 y el lema de König, podríamos encontrar un estado $A' \in \mathcal{A}(P)$ tal que $\mathsf{nd}(A') \leq t$ y $A' \upharpoonright t \subseteq A$; con lo que $A' \ll b$ que estaría en contradicción con nuestra hipótesis de partida. Por tanto el conjunto $\mathcal{Q}$ es finito, y en caso de que $\mathcal{Q} \neq \varnothing$ podemos considerar el proceso

$$P_1 = \bigsqcap_{Q \in \mathcal{Q}} Q$$

obteniendo $\mathsf{Barb}(P_1) = B_1$. Por hipótesis de inducción P_1 must T_1, y puesto que para cada $Q \in \mathcal{Q}$ tenemos la transición $P_1 \rightarrowtail Q$, concluimos que

$$\forall Q \in \mathcal{Q} : \ Q \text{ must } T_1$$

Observemos también que

$$\mathcal{Q} = \varnothing \quad \Longleftrightarrow \quad B_1 = \varnothing$$

Cualquier computación de $P \mid T$, viene de una computación de P y otra de T. Fijémonos en la de P:

$$P = P_1 \rightarrowtail^* P'_1 \xrightarrow{e_1 t_1} P_2 \rightarrowtail^* P'_2 \xrightarrow{e_2 t_2} \cdots$$

Tomemos $t^i = \sum_{j=1}^{i-1} t_i$. Podemos distinguir los siguientes casos:

- Existe k tal que $e_k t_k = a't'$ con $a'(t'+t^k) \in A_1$. Si esto es así ha sido así tenido que suceder con la complicidad de la prueba

$$\mathsf{Upd}(T, t^k) \xrightarrow{a't'} \mathtt{OK}$$

que nos ha llevado por tanto a una computación con éxito.

- Existe k tal que $e_k t_k = a(t - t^k)$. Consideramos el conjunto de acciones

$$A' = \bigcup_{i=1}^{k} ((\mathsf{TA}(P_i') \upharpoonright t_i) + t^i)$$

Supongamos que $a't' \in A' \cap A_1$. Tenemos entonces que existe un proceso P_i' tal que $a'(t' - t^i) \in \mathsf{TA}(P_i') \upharpoonright t_i$. En consecuencia $T \xrightarrow{a't'} \mathtt{OK}$ y por tanto $\mathsf{Upd}(T, t^i) \xrightarrow{a(t'-t^i)} \mathtt{OK}$. Puesto que $t' - t^i < t_i$ tenemos que P_k' tendría que haber sincronizado con $\mathsf{Upd}(T, t^i)$ que habría producido una computación con éxito, y por tanto la transición $P_k' \xrightarrow{a(t-t^k)}$ no es posible.

Podemos suponer por tanto que $A' \cap A_1 = \varnothing$, de modo que $A' \subseteq A$. Tenemos pues que $P \xRightarrow{A'at} P_{k+1}$, y por tanto $P_{k+1} \in Q$. En consecuencia la computación de $P|T$ nos ha llevado hasta $P_{k+1}|T_1$, y entonces, puesto que P_{k+1} must T_1, tenemos que la computación es una computación de éxito.

- Existe k tal que $e_l = \tau$ para todo $l < k$ y $P_k \Uparrow$. Entonces la computación da lugar al estado

$$A' = \{\Omega t^k\} \cup \bigcup_{i=1}^{k-1} ((\mathsf{TA}(P_i') \upharpoonright t_i) + t^i)$$

Si $t^k \leq t$ existe $a't' \in A \cap A_1$; tomemos en particular al que corresponda al instante de tiempo menor. Entonces existe $1 \leq i < k$ tal que $a'(t' - t^i) \in \mathsf{TA}(P_i')$. Tenemos que la computación va desde $P \mid T$ hasta $P_i' \mid \mathsf{Upd}(T, t^i)$, y en consecuencia el proceso y la prueba han de sincronizar en la acción $a'(t' - t^i)$, con lo cual la transición $P_i' \xrightarrow{\tau t_i} P_{i+1}$ no sería posible, llegándose a una contradicción. Por tanto, podemos suponer que $t^k > t$. Existe entonces un proceso P_i' tal que $t^i \leq t$ y $t^{i+1} > t$. Tenemos que la computación de $P \mid T$ nos lleva hasta $P_i' \mid \mathsf{Upd}(T, t^i)$; existe entonces T' tal que

$$T \xrightarrow{\tau t} T' \xrightarrow{\mathtt{OK}}$$

de modo que tendremos $\mathsf{Upd}(T, t^i) \xrightarrow{\tau(t-t^i)} T'$, y puesto que $t_i > t - t^i$, tenemos forzosamente la transición

$$P_i' \mid \mathsf{Upd}(T, t^i) \mapsto \mathsf{Upd}(P_k', t - t^i) \mid T'$$

que nos conduce a una computación de éxito.

- Para todo $k \geq 1$ se verifica $P_k \Downarrow$ y $e_k = \tau$. O bien la computación es finita, en cuyo caso tomamos como P'_n su último proceso, o bien es infinita. En ambos casos se genera un estado A', con

$$A' = \mathsf{TA}(P'_n) \cup \bigcup_{i=1}^{n-1} ((\mathsf{TA}(P_i) \upharpoonright t_i) + t^i) \qquad A' = \bigcup_{i \in \mathbb{N}} ((\mathsf{TA}(P_i) \upharpoonright t_i) + t^i)$$

Si el cómputo es finito — Si el cómputo es infinito

Puesto que $\mathsf{nd}(A') = \infty$, se ha de cumplir alguna de las condiciones siguientes:

  - Existe algún $a't' \in A' \cap A_1$. Tomamos entonces el que corresponda a un instante de tiempo menor. Existe entonces j tal que $a(t' - t^j) \in \mathsf{TA}(P'_j) \upharpoonright t_j$. En consecuencia la computación conjunta de la prueba y el proceso nos lleva hasta $P'_j \mid \mathsf{Upd}(T, t^j)$. Puesto que $T \xrightarrow{at'} \mathtt{OK}$ tenemos que $\mathsf{Upd}(T, t^j) \xrightarrow{a(t-t^j)} \mathtt{OK}$. Por tanto, la prueba y el proceso tendrían que sincronizar en la acción $a'(t' - t^j)$, de modo que la transición $P_j \xrightarrow[\ \ ]{\tau t_j} P_{j+1}$ no sería posible, lo cual nos lleva a una contradicción
  - $at \in A'$. Usando el mismo razonamiento que antes llegamos también a una contradicción.
  - Por tanto, el único caso que queda como posible es el correspondiente a que $at \notin A'$ y $A' \cap A_1 = \varnothing$. Puesto que estamos suponiendo que $P_k \Downarrow$ para todo k, existirá $j \in \mathbb{N}$ tal que $t^j < t$ y si $P_j \xrightarrow{\tau t_j}$ entonces $t^{j+1} > t$. Tenemos que la computación de $P \mid T$ nos lleva hasta $P^j \mid \mathsf{Upd}(T, t^j)$. Puesto que existe T' tal que

$$T \xrightarrow{\tau t} T' \xrightarrow{\mathtt{OK}}$$

tenemos $\mathsf{Upd}(T, t^j) \xrightarrow{\tau(t-t^j)} T'$, y como quiera que $t^{j+1} = t^j + t_j > t$, tenemos $t - t^j > t_j$, con lo que

$$P_j \mid \mathsf{Upd}(T, t^j) \xrightarrow{\tau(t-t^j)} \mathsf{Upd}(P_j, t - t^j) \mid T'$$

de donde se deduce que la computación de $P \mid T$ tiene éxito.

Puesto que en cualquier caso hemos obtenido una computación de éxito, tenemos que P must T.

□

A.2 Demostraciones del Capítulo 4

Demostración de los lemas 4.2.13 y 4.2.15.

La demostración en cada caso es prácticamente igual a las de las proposiciones 6.1.11 y 6.1.13 respectivamente. Y esto es así, porque en particular si $\mathsf{stb}(P)$ y $\mathsf{stb}(Q)$ tenemos

$$\mathsf{TA}(P \parallel_{\varnothing} Q) = \mathsf{TA}(P \,\Box\, Q) = \mathsf{TA}(P) \cup \mathsf{TA}(P)$$

$\Box$

Demostración de la proposición 4.2.17.

Haremos la demostración por inducción sobre la longitud de la deducción de la transición generalizada $P \,\Box\, Q \stackrel{Aat\cdot bs}{\Longrightarrow} R$.

Caso base Si la longitud de la deducción es 1 tenemos que $P \,\Box\, Q \stackrel{\epsilon}{\Longrightarrow} P \,\Box\, Q$, y por tanto no habría nada que probar.

Caso inductivo Aquí tenemos varias posibilidades:

- $P \,\Box\, Q \rightarrowtail R$. En tal caso o bien $P \rightarrowtail P'$ y $R = P' \,\Box\, Q$, o bien $Q \rightarrowtail Q'$ y $R = P \,\Box\, R'$. En ambos casos obtenemos el resultado aplicando la hipótesis de inducción.
- $P \Box Q \xrightarrow{at} R' \stackrel{bs}{\Longrightarrow} R$. En tal caso $A = \mathsf{TA}(P \Box Q) \upharpoonright t$ y o bien $P \xrightarrow{at} R'$, o bien $Q \xrightarrow{at} R'$. Supongamos que se da el primer caso. Tenemos entonces $\mathsf{idle}(Q) \geq t$ y $\mathsf{stb}(Q)$, y por tanto todo estado $A_2 \in \mathcal{A}(Q)$ verifica $\mathsf{TA}(Q) \upharpoonright t = A_2 \upharpoonright t$. Tomando entonces $A_1 = \mathsf{TA}(P) \upharpoonright t$, obtenemos que $P \stackrel{A_1 at\cdot bs}{\Longrightarrow} P'$ con lo que queda solventado este caso.
- $P \,\Box\, Q \xrightarrow{\tau t'} R'$ y $R' \stackrel{A'a(t-t')\cdot bs}{\Longrightarrow} R$. En este caso tenemos

  $$A = (\mathsf{TA}(P \,\Box\, Q) \upharpoonright t') \cup (A' + t')$$

  Por otra parte tenemos dos casos posibles:

  $$P \xrightarrow{\tau t'} P',\ \mathsf{idle}(Q) \geq t,\ \mathsf{stb}(Q)\ \text{ y }\ R' = P' \,\Box\, \mathsf{Upd}(Q, t')$$
  ó
  $$Q \xrightarrow{\tau t'} Q',\ \mathsf{idle}(P) \geq t,\ \mathsf{stb}(P)\ \text{ y }\ R' = \mathsf{Upd}(P, t') \,\Box\, Q'$$

  Supongamos que se da el el primer caso; el segundo sería simétrico. Por hipótesis de inducción, existen una b-traza $A_1'a(t-t') \cdot bs$ y un estado A_2' de manera que $\mathsf{nd}(A_2') > t - t'$ y $A' = A_1' \cup (A_2' \upharpoonright t')$; y además se tiene uno de los dos hechos siguientes:

- $P' \xRightarrow{A'_1 a(t-t')\cdot bs} R$ y $A'_2 \in \mathcal{A}(\mathsf{Upd}(Q,t'))$.
- $Q \xRightarrow{A'_1 a(t-t')\cdot bs} R$ y $A'_2 \in \mathcal{A}(P')$.

En ambos casos, tomando

$$A_1 = (\mathsf{TA}(Q) \upharpoonright t') \cup (A'_1 + t') \quad \text{y} \quad A_2 = (\mathsf{TA}(P) \upharpoonright t') \cup (A'_2 + t')$$

se llega a la conclusión deseada.

□

Demostración de la proposición 4.2.18.
Demostraremos que se verifica la primera propiedad; la segunda es totalmente simétrica. Puesto que $A_Q \in \mathcal{A}(Q)$ existe una computación de Q

$$Q = Q_1 \rightarrowtail^* Q'_1 \xrightarrow{\tau t_1} Q_2 \rightarrowtail^* Q'_2 \xrightarrow{\tau t_2} \cdots$$

que genera dicho estado. Tomemos $t^i = \sum_{j=1}^{i-1} t_j$. Puesto que $\mathsf{nd}(A_Q) > t$ existe $l \in \mathbb{N}$ tal que:

- $Q_j \Downarrow$ para $j \leq l$,
- $t \geq t^i$ y si $Q'_i \xrightarrow{\tau t_l}$, tenemos entonces $t < t^{i+1}$.

Por otro lado tenemos que existe R' de manera que $P \xRightarrow{Aat} R' \xRightarrow{bs} R$. Supongamos que la deducción de $P \xRightarrow{A_P at} R'$ tiene una longitud de k etapas, y sea l la longitud de la computación desde Q hasta Q'_i. Demostraremos la existencia del conjunto $A \subseteq TAct$ tal que $P \,\Box\, Q \xRightarrow{Aat} R'$ por inducción sobre $k + l$. Una vez ello demostrado, el enunciado se sigue inmediatamente.

Caso Base. Puesto que $k \leq 1$ y $l \leq 1$, tenemos $Q'_l = Q$ y $P \xrightarrow{at} R$. En consecuencia se verifica:

- $A_P = \mathsf{TA}(P) \upharpoonright t$,
- puesto que Q es estable, convergente e $\mathsf{idle}(Q) \geq t$, cualquier estado A_Q de Q verifica $\mathsf{TA}(Q) \upharpoonright t = A_Q \upharpoonright t$.

Tenemos entonces $P \,\Box\, Q \xrightarrow{at} R'$, y por tanto tenemos $P \,\Box\, Q \xRightarrow{Aat} R'$ donde $A = \mathsf{TA}(P \,\Box\, Q) \upharpoonright t$. Para concluir basta observar que

$$A = \mathsf{TA}(P \,\Box\, Q) \upharpoonright t = (\mathsf{TA}(P) \upharpoonright t) \cup (\mathsf{TA}(Q) \upharpoonright t) = A_P \cup (A_Q \upharpoonright t)$$

Caso Inductivo Tenemos las siguientes posibilidades:

- $P \rightarrowtail P'$. En este caso tenemos que $P' \xRightarrow{A_P at} R'$, y entonces aplicando la hipótesis de inducción existe $A \subseteq A_P \cup (A_Q \upharpoonright t)$ tal que $P' \,\square\, Q \xRightarrow{Aat} R'$ y por tanto tenemos que $P \,\square\, Q \xRightarrow{Aat} R'$.
- $Q \rightarrowtail Q'$, caso simétrico al anterior.
- $P \xrightarrow{\tau t'} P'$, $\mathsf{stb}(Q)$ e $\mathsf{idle}(Q) \geq t'$. En tal caso existe $A_P \subseteq TAct$ tal que

$$P' \xRightarrow{A'_P a(t-t')} R \quad \text{y} \quad A_P = (\mathsf{TA}(P) \upharpoonright t') \cup (A'_P + t')$$

Puesto que $Q \Downarrow$, existe $A'_Q \in \mathcal{A}(\mathsf{Upd}(Q, t'))$ de manera que

$$A_Q = (\mathsf{TA}(Q) \upharpoonright t') \cup (A'_Q + t')$$

Entonces por hipótesis de inducción tenemos que existe $A' \subseteq TAct$ tal que

$$A' \subseteq A'_P \cup (A'_Q \upharpoonright (t - t')) \quad \text{y} \quad P' \,\square\, \mathsf{Upd}(Q, t') \xRightarrow{A'a(t-t')} R'$$

Entonces tomando $A = (\mathsf{TA}(P \,\square\, Q) \upharpoonright t') \cup (A' + t')$ tenemos que $P \,\square\, Q \xRightarrow{Aat} R'$. Para concluir basta observar que $A \subseteq A_P \cup (A_Q \upharpoonright t)$.

- $Q \xrightarrow{\tau t'} Q'$, $\mathsf{stb}(P)$ e $\mathsf{idle}(P) \geq t'$. En este caso existe $A'_Q \in \mathcal{A}(Q')$ tal que $A_Q = A_1 \cup (A'_Q + t_1)$, siendo $A_1 = \mathsf{TA}(Q) \upharpoonright t'$. Por otro lado, existe un conjunto de acciones temporizadas A'_P verificando

$$\mathsf{Upd}(P, t') \xRightarrow{A'_P a(t-t')} R' \quad \text{y} \quad A_P = (\mathsf{TA}(P) \upharpoonright t') \cup (A'_P + t')$$

Aplicando la hipótesis de inducción existe $A' \subseteq A'_P \cup (A'_Q \upharpoonright (t - t'))$ tal que

$$\mathsf{Upd}(P, t_1) \,\square\, Q' \xRightarrow{A'at} R'$$

Tomamos entonces $A = (\mathsf{TA}(P \,\square\, Q) \upharpoonright t_1) \cup (A' + t_1)$ y tenemos

$$P \,\square\, Q \xrightarrow{\tau t'} \mathsf{Upd}(P, t') \,\square\, Q' \xRightarrow{A'a(t-t')} R'$$

de modo que tenemos $P \,\square\, Q \xRightarrow{Aat} R'$, y por otra parte se cumple $A \subseteq A_P \cup (A_Q \upharpoonright t)$, lo que nos permite concluir la demostración.

$\square$

Demostración del teorema 4.2.19.

Tomemos en primer lugar $b \in \mathcal{B}_{\text{con}}[\![\square]\!](B_1, B_2)$. Según sea la forma de b existen dos posibilidades:

$\mathbf{b} = \mathbf{A}$. Por la definición semántica del operador existen entonces $A_1 \in B_1$ y $A_2 \in B_2$ tales que $A = A_1 \sqcup_\varnothing A_2$. Puesto que $\mathsf{Barb}(P_1) \ll B_1$ existirá un estado $A_P \in \mathcal{A}(P)$ tal que $A_P \ll A_1$, y razonando análogamente para A_2 obtenemos un estado $A_Q \in \mathcal{A}(Q)$ tal que $A_Q \ll A_2$. Aplicando la proposición 4.2.16 obtenemos un estado $A' \in \mathcal{A}(P \Box Q)$ tal que $A' \ll A_P \sqcup_\varnothing A_Q$. Por lo que tenemos

$$A' \ll A_P \sqcup_\varnothing A_Q \ll A_1 \sqcup_\varnothing A_2 = A$$

$\mathbf{b} = \mathbf{Aat} \cdot \mathbf{b'}$. Existen entonces un conjunto de acciones temporizadas A_1 y un estado A_2 con $\mathsf{nd}(A_2) > t$, tales que $A = A_1 \cup A_2 \upharpoonright t$ cumpliéndose una de las siguientes aserciones

- $A_1 at \cdot b' \in B_1$ y $A_2 \in B_2$
- $A_1 at \cdot b' \in B_2$ y $A_2 \in B_1$

Puesto que ambas aserciones son simétricas la una de la otra, supondremos que se presenta la primera de ellas. Puesto que $B_2 \ll \mathsf{Barb}(Q)$ existe un estado A_Q tal que $A_Q \ll A_2$, y existe una barba $b_P \in \mathsf{Barb}(P)$ tal que $b_P \ll b$. Tomamos entonces una barba b_1 escogida como se indica a continuación, distinguiendo casos según sean b_Q y A_Q:

- Si b_P es un estado, A_P entonces por aplicación de la proposición 4.2.14 existe un estado $A' \in \mathcal{A}(P \Box Q)$ tal que $A' \ll A_P \sqcup_\varnothing A_Q$. Tomamos en tal caso $b_1 = A'$.
- Si $b_P = A'_1 at \cdot b''$ y $\mathsf{nd}(A_Q) \leq t$ entonces por el lema 3.5.2 existe un estado $A_P \in \mathcal{A}(P)$ tal que $A_P \upharpoonright t \subseteq A'_1$. Al igual que en el caso anterior concluimos que existe $A' \in \mathcal{A}(P \Box Q)$ tal que $A' \ll A_P \sqcup_\varnothing A_Q$. Y de nuevo en este caso tomamos $b_1 = A'$.
- Supongamos finalmente que $b_P = A'_1 at \cdot b''$ y $\mathsf{nd}(A_Q) > t$. Existirán entonces un estado A'', una b-traza bs'' y un proceso R tales que

  $$b'' = bs'' \cdot A'', \quad P \xRightarrow{A'_1 at \cdot bs''} R \quad \text{y} \quad A'' \in \mathcal{A}(R)$$

  Entonces por la proposición 4.2.18 existe un conjunto de acciones temporizadas A' tal que

  $$A' \subseteq A'_1 \cup A_Q \upharpoonright t \quad \text{y} \quad P \Box Q \xRightarrow{A' at \cdot bs''} R$$

  Tomamos entonces $b_1 = A' at \cdot b'' \in \mathsf{Barb}(P \Box Q)$.

Obsérvese que en todos los casos tenemos que $b_1 \ll b$.

Tomemos ahora $b \in \mathsf{Barb}(P \,\Box\, Q)$, y buscaremos $b_1 \in \mathcal{B}_{\mathrm{con}}[\![\Box]\!](B_1, B_2)$ tal que $b_1 \ll b$. Dependiendo de la forma de b tenemos dos posibilidades:

b = A. Entonces por aplicación de la proposición 4.2.14 tenemos que existen dos estados $A_P \in \mathsf{Barb}(P)$ y $A_Q \in \mathsf{Barb}(Q)$ tales que

$$A_P \sqcup_\varnothing A_Q \ll A$$

Puesto que $B_1 \ll \mathsf{Barb}(P)$ y $B_2 \ll \mathsf{Barb}(Q)$ existen estados $A_1 \in B_1$ y $A_2 \in B_2$ tales que $A_1 \ll A_P$ y $A_2 \ll A_Q$. Tomamos entonces $A' = A_1 \sqcup_\varnothing A_2$ y tenemos

$$A' = A_1 \sqcup_\varnothing A_2 \ll A_P \sqcup_\varnothing A_Q \ll A$$

b = Aat · b′. Puesto que $b \in \mathsf{Barb}(P \,\Box\, Q)$ existirán un estado A', una b-traza bs' y un proceso R tales que

$$b' = bs' \cdot A', \quad P \,\Box\, Q \xRightarrow{Aat \cdot bs'} R \quad \text{y} \quad A' \in \mathcal{A}(R)$$

En virtud de la proposición 4.2.17 existen un conjunto de acciones temporizadas A_1 y un estado A_2 tales que se da una de las siguientes aserciones:

- $A_2 \in \mathcal{A}(Q)$ y $P \xRightarrow{A_1 at \cdot bs'} R$ o bien
- $A_2 \in \mathcal{A}(P)$ y $Q \xRightarrow{A_1 at \cdot b'} R$.

Puesto que las dos posibilidades son simétricas la una de la otra, supondremos que es cierta la primera. Por lo tanto $A_1 at \cdot b' \in \mathsf{Barb}(P)$, y puesto que $B_1 \ll \mathsf{Barb}(P)$ y $B_2 \ll \mathsf{Barb}(Q)$ existen un estado $A_2' \in B_2$ y una barba $b_1' \in B_1$ tales que $A_2' \ll A_1$ y $b_1' \ll A_1 at \cdot b'$. Tomamos entonces una barba b_1 elegida como sigue:

- Si b_1' es un estado A_1' tomamos $b_1 = A_1' \sqcup_\varnothing A_2'$.
- $b_1' = A_1' at \cdot b''$ y $\mathsf{nd}(A_2') \leq t$. Puesto que B es consistente existirá un estado $A_1'' \in B_1$ tal que $A_1' = A_1'' \upharpoonright t$; tomaremos entonces $b_1 = A_1'' \sqcup_\varnothing A_2'$.
- Si $b_1' = A_1' at \cdot b''$ y $\mathsf{nd}(A_2') > t$ tomamos $b_1 = (A_1' \cup A_2' \upharpoonright t) at \cdot b''$.

En todos los casos tenemos que $b_1 \in \mathcal{B}_{\mathrm{con}}[\![\Box]\!](B_1, B_2)$ y $b_1 \ll b$.

□

Demostración de la proposición 4.4.13. Continuidad mediante prefijo de acciones internas.

Para demostrar la continuidad basta probar

$$\mathsf{lub}\{\mathcal{B}_{\mathrm{con}}[\![\tau t]\!](B_i) \mid i \in \mathbb{N}\} = \mathcal{B}_{\mathrm{con}}[\![\tau t;]\!](\mathsf{lub}(\mathcal{B}))$$

Veamos cada una de las inclusiones

$\boxed{\subseteq}$

Tomemos $b = bs \cdot A \in \mathsf{lub}\{\mathcal{B}_{\text{con}}[\![\tau t]\!](B_i) \mid i \in \mathbb{N}\}$. Hay dos posibilidades:

- $\mathsf{nd}(A) < \infty$. En este caso tenemos que para todo $k \in \mathbb{N}$ existe $l \geq k$ tal que $bs \cdot A \in \mathcal{B}_{\text{con}}[\![\tau t;]\!](B_l)$. Existe entonces $bs' \cdot A' \in B_l$ tal que $bs \cdot A = (bs' \cdot A') + t$. Por lo tanto $bs' \cdot A' \in \mathsf{lub}(\mathcal{B})$, y entonces $bs \cdot A \in \mathcal{B}_{\text{con}}[\![\tau t;]\!](\mathsf{lub}(\mathcal{B}))$.
- $\mathsf{nd}(A) = \infty$. En este caso tenemos que para cada $t_0 \in \mathcal{T}$ existen $l \in \mathbb{N}$ y un estado A_l tales que $bs \cdot A_l \in \mathcal{B}_{\text{con}}[\![\tau t;]\!](B_l)$ y $A_l \upharpoonright t_0 = A \upharpoonright t_0$. Tenemos dos posibilidades:
  - $bs = \epsilon$ tenemos entonces que existe $A'_l \in B_l$ tal que $A_l = A'_l + t$, por lo que $A \geq t$. Tomamos entonces $A' = A - t$ y tenemos que $A' \in \mathsf{lub}(\mathcal{B})$, por lo que $A \in \mathcal{B}_{\text{con}}[\![\tau t;]\!](\mathsf{lub}(\mathcal{B}))$.
  - $bs = A_1 a_1 t_1 \cdot bs_1$, tomamos $bs' = (A_1 - t) a_1 (t_1 - t) \cdot bs_1$ y tenemos que $bs' \cdot A_l \in B_l$. Por tanto $bs' \cdot A \in \mathsf{lub}(\mathcal{B})$ y en consecuencia

$$bs \cdot A = (bs' \cdot A) + t \in \mathcal{B}_{\text{con}}[\![\tau t;]\!](\mathsf{lub}(\mathcal{B}))$$

$\boxed{\supseteq}$

Tomemos ahora $bs \cdot A \in \mathcal{B}_{\text{con}}[\![\tau t;]\!](\mathsf{lub}(\mathcal{B}))$. Existe entonces una barba $bs' \cdot A' \in \mathsf{lub}(\mathcal{B})$ tal que $bs \cdot A = (bs' \cdot A') + t$. Tenemos de nuevo dos posibilidades:

- $\mathsf{nd}(A') < \infty$. Entonces para cada $k \in \mathbb{N}$ existe $l \geq k$ tal que $bs' \cdot A' \in B_l$. Por tanto $bs \cdot A \in B_l$, y en consecuencia

$$bs \cdot A \in \mathsf{lub}\{\mathcal{B}_{\text{con}}[\![\tau t;]\!](B_i) \mid i \in \mathbb{N}\}$$

- $\mathsf{nd}(A') = \infty$. Tenemos dos posibilidades
  - $bs' = \epsilon$. Tenemos entonces que $A = A' + t$ y $bs = \epsilon$. En consecuencia para cada $t_0 \in \mathcal{T}$ existe $A'_l \in B_l$ tal que $A' \upharpoonright t_0 = A'_l \upharpoonright t_0$. Tomamos entonces $A_l = A'_l + t \in \mathcal{B}_{\text{con}}[\![\tau t;]\!](B_l)$ de modo que se tiene

$$A = A' + t \in \mathsf{lub}\{\mathcal{B}_{\text{con}}[\![\tau t;]\!](B_i) \mid i \in \mathbb{N}\}$$

  - $bs' = A_1 a_1 \cdot bs'_1$. Entonces $bs = (A_1 + t) a (t_1 + t) \cdot bs'_1$. de modo que fijado $t_0 \in \mathcal{T}$ existe un estado A_l tal que $bs' \cdot A_l \in B_l$, $\mathsf{nd}(A_l) \geq t_0$ y $A \upharpoonright t_0 = A_l \upharpoonright t_0$, y por lo tanto

$$bs \cdot A_l = (bs' \cdot A_l) + t \in \mathcal{B}_{\text{con}}[\![\tau t;]\!](B_l)$$

Puesto que ello lo hemos hecho para un $t_0 \in \mathcal{T}$ arbitrario, concluimos

$$bs \cdot A \in \mathsf{lub}\{\mathcal{B}_{\text{con}}[\![\tau t;]\!](B_i) \mid i \in \mathbb{N}\}$$

□

Demostración de la proposición 4.4.15. Continuidad del operador prefijo mediante una acción visible.
Para simplificar la notación tomaremos los conjuntos

$$B = \mathcal{B}_{\text{con}}[\![at;]\!](\mathsf{lub}(\mathcal{B})) \quad \text{y} \quad B' = \mathsf{lub}(\{\mathcal{B}_{\text{con}}[\![at;]\!](B_i) \mid i \in \mathbb{N}\})$$

Para demostrar la continuidad de este operador probar $B = B'$.

$\boxed{\mathbf{B' \subseteq B}}$

Sea $b = bs \cdot A \in B'$. Si $bs = \epsilon$, puesto $\{at\}$ es el único estado después de la traza vacía que hay en cada uno de los conjuntos B_l, tenemos que forzosamente $A = \{at\}$. Dado que $\{at\} \in B$, este caso estaría concluido. Supongamos ahora que $bs \neq \epsilon$; puesto que toda b-traza no vacía de cada uno de los B_l empieza con $\varnothing at$, tenemos que existirá una b-traza bs_1 tal que $bs = \varnothing at \cdot bs_1$. Tenemos entonces dos posibilidades:

- $\mathsf{nd}(A) < \infty$. En tal caso para cada $k \in \mathbb{N}$ existe $l \geq k$ de modo que $bs \cdot A \in \mathcal{B}_{\text{con}}[\![at;]\!](B_l)$. En consecuencia tendremos que $bs_1 \cdot A \in \mathsf{lub}(\mathcal{B})$, y por tanto $bs \cdot A \in B$.
- $\mathsf{nd}(A) = \infty$. Tenemos entonces que para cada $t_0 \in \mathcal{T}$ existe un estado A_l tal que $bs \cdot A_l \in \mathcal{B}_{\text{con}}[\![at;]\!](B_l)$ y $A \upharpoonright t_0 = A_l \upharpoonright t_0$. Concluimos entonces que $bs_1 \cdot A_l \in B_l$, y por tanto $bs_1 \cdot A \in \mathsf{lub}(\mathcal{B})$, de modo que $bs \cdot A \in B$.

$\boxed{\mathbf{B \subseteq B'}}$

Sea ahora $b = bs \cdot A \in B$. Al igual que en el caso anterior supondremos en primer lugar que $bs = \epsilon$. En tal caso ha de ser $A = \{at\}$. Por otro lado tenemos que para cada $i \in \mathbb{N}$ $\{at\} \in \mathcal{B}_{\text{con}}[\![at;]\!](B_i)$, por lo que $\{at\} \in B'$. Supongamos ahora que $bs \neq \epsilon$, entonces existe bs_1 de manera que $bs = \varnothing at \cdot bs_1$, y por lo tanto $bs_1 \cdot A \in \mathsf{lub}(\mathcal{B})$. Tenemos entonces dos posibilidades:

- $\mathsf{nd}(A) < \infty$. Entonces para cada $k \in \mathbb{N}$ existe $l \geq k$ tal que $bs_1 \cdot A \in B_l$, de donde
$$b = bs \cdot A = (\varnothing at \cdot bs_1) \cdot A \in \mathcal{B}_{\text{con}}[\![at;]\!](B_l)$$
lo que permite concluir $b \in B'$.

- $\mathsf{nd}(A) = \infty$. En este caso, para cada $t_0 \in \mathcal{T}$ existe $l \in \mathbb{N}$ tal que existe un estado A_l de manera que $bs_1 \cdot A \in B_l$. Tenemos entonces que

$$bs \cdot A_l = (\varnothing at \cdot bs_1) \cdot A_l \in \mathcal{B}_{\text{con}}[\![at;]\!](B_l)$$

Puesto que ello lo podemos hacer para cada $t_0 \in \mathcal{T}$, concluimos finalmente que $b = bs \cdot A \in B'$.

□

Demostración de la proposición 4.4.17. Continuidad de la elección interna.
Para simplificar la notación, tomaremos

$$B = \mathcal{B}_{\text{con}}[\![\sqcap]\!](\mathsf{lub}(\mathcal{B}_1), \mathsf{lub}(\mathcal{B}_2)) \;\text{ y }\; B' = \mathsf{lub}(\{\mathcal{B}_{\text{con}}[\![\sqcap]\!](B_{1i}, B_{2i}) \mid i \in \mathbb{N}\})$$

$\boxed{\mathbf{B' \subseteq B}}$

Siendo $b = bs \cdot A \in B'$, tenemos dos posibilidades:

- $\mathsf{nd}(A) < \infty$. En este caso para cada $k \in \mathbb{N}$ existe $l \geq k$ tal que

$$bs \cdot A \in \mathcal{B}_{\text{con}}[\![\sqcap]\!](B_{1l}, B_{2l}) = B_{1l} \cup B_{2l}$$

Tenemos que o bien para una cantidad infinita de índices k's la barba $bs \cdot A$ estará en B_{1l}, o bien para una cantidad infinita de índices k's estará en B_{2l}. Tenemos entonces dos casos simétricos:

  - Para cada $k \in \mathbb{N}$ existe $k \geq l$ tal que $bs \cdot A \in B_{1l}$. Entonces tenemos que $bs \cdot A \in \mathcal{B}_1$ y por tanto $bs \cdot A \in \mathcal{B}_{\text{con}}[\![\sqcap]\!](\mathcal{B}_1, \mathcal{B}_2)$.
  - El caso simétrico al anterior: para cada $k \in \mathbb{N}$ existe $k \geq l$, tal que $bs \cdot A \in B_{2l}$.

- $\mathsf{nd}(A) = \infty$, entonces para cada $t \in \mathcal{T}$ existen $l \in \mathbb{N}$ y un estado A_l tales que $A \upharpoonright t = A_l \upharpoonright t$ y

$$bs \cdot A_l \in \mathcal{B}_{\text{con}}[\![\sqcap]\!](B_{1l}, B_{2l}) = B_{1l} \cup B_{2l}$$

Tenemos entonces que o bien la barba $bs \cdot A_l$ estará en B_{1l} para un número infinito de instantes t's, o bien para un número infinito de instantes t's la barba $bs \cdot A_l$ estará en B_{2l}. Tenemos pues dos posibilidades simétricas:

  - Para cada $t \in \mathcal{T}$ existe $t' \geq t$ tal que el l correspondiente a t' cumple que $bs \cdot A_l \in B_{1l}$; y puesto que $t' \geq t$, tendremos que $A_l \upharpoonright t = A_l \upharpoonright t$. Entonces tenemos que $bs \cdot A \in \mathcal{B}_1$ y por tanto $bs \cdot A \in \mathcal{B}_{\text{con}}[\![\sqcap]\!](\mathcal{B}_1, \mathcal{B}_2)$.

– De manera simétrica al caso al anterior: para cada $t \in \mathcal{T}$ existe $t' \geq t$ tal que el índice l correspondiente a t' verifica $bs \cdot A_l \in B_{2l}$.

$\boxed{\mathbf{B} \subseteq \mathbf{B'}}$

Sea ahora $b = bs \cdot A \in B$. Tendremos que o bien $bs \cdot A \in \mathcal{B}_1$, o bien $bs \cdot A \in \mathcal{B}_2$. Consideremos el primer caso; el segundo sería totalmente simétrico. Tenemos dos posibilidades:

- $\mathsf{nd}(A) < \infty$. Entonces para cada $k \in \mathbb{N}$ existe $l \geq k$ tal que $bs \cdot A \in B_{1l}$. Por tanto tenemos que $bs \cdot A \in \mathcal{B}_{\text{con}}[\![\sqcap]\!](B_{1l}, B_{2l})$ por lo que $bs \cdot A \in B'$.
- $\mathsf{nd}(A) = \infty$. Para cada $t \in \mathcal{T}$ existe un estado A_l tal que $A \upharpoonright t = A_l \upharpoonright t$ y $bs \cdot A_l \in B_{1l}$. Entonces tenemos que $bs \cdot A_l \in \mathcal{B}_{\text{con}}[\![\sqcap]\!](B_{1l}, B_{2l})$, y por tanto tenemos que $bs \cdot A \in B'$.

□

Demostración de la proposición 4.4.18. Monotonía de la elección externa.
Debido a la simetría de la definición del operador, basta con que nos concentremos en el primer argumento. El estudio del segundo sería simétrico.

a) Sea $b' \in \mathcal{B}_{\text{con}}[\![\Box]\!](B'_1, B)$. Tenemos dos posibilidades

- $b' = A'$. Existen $A'_1 \in B'_1$ y $A_0 \in B$ tales que $A' = A'_1 \sqcup_\varnothing A_0$. Existe entonces $A_1 \in B_1$ tal que $A_1 \prec A'_1$, de modo que el estado $A = A_1 \sqcup_\varnothing A_0$ cumple que $A \in \mathcal{B}_{\text{con}}[\![\Box]\!](B_1, B)$ y $A \prec A'$.
- $b' = A'at \cdot b'_0$. Existen entonces una barba $A'_1 at \cdot b'_0$ y un estado A'_2 tales que $\mathsf{nd}(A_2) > t$ y $A' = A_1 \cup (A'_2 \upharpoonright t)$. Tenemos dos posibilidades:
  – $A'_1 at \cdot b'_0 \in B'_1$ y $A'_2 \in B$. Entonces existe $b \in B_1$ tal que $b \prec A'_1 at \cdot b'_0$. Si $b = A_1 \in B_1$ entonces $\mathsf{nd}(A_1) \leq t$ y $TAct(A_1) = A'_1 \upharpoonright \mathsf{nd}(A_1)$, y por lo tanto, tomando $A = A_1 \sqcup_\varnothing A'_2$, tenemos que $A \prec b'$ y $A \in \mathcal{B}_{\text{con}}[\![\Box]\!](B_1, B)$. Ahora, si $b = A'_1 at \cdot b_0$ con $b_0 \prec b'_0$, obtenemos $A'at \cdot b'_0 \in \mathcal{B}_{\text{con}}[\![\Box]\!](B_1, B)$.
  – $A'_1 at \cdot b'_0 \in B$ y $A'_2 \in B_1$. Existe entonces $A_2 \in B_1$ tal que $A_2 \prec A'_2$. Supongamos que $\mathsf{nd}(A_2) \leq t$. Puesto que B es consistente, existirá un estado $A_1 \in B_1$ tal que $A'_1 = A_1 \upharpoonright t$, y tomando entonces el estado $A = A_1 \sqcup_\varnothing A_2$ tenemos que $A \prec b$ y $A \in \mathcal{B}_{\text{con}}[\![\Box]\!](B_1, B)$. Si $\mathsf{nd}(A_2) > t$ entonces tendríamos que $A_2 \upharpoonright t = A'_2 \upharpoonright t$, y por tanto $b' \in \mathcal{B}_{\text{con}}[\![\Box]\!](B_1, B)$.

b) Sea ahora $b = bs \cdot A \in \mathcal{B}_{\text{con}}[\![\Box]\!](B_1, B)$. Tenemos dos posibilidades:

- $b = A$, entonces existen $A_1 \in B_1$ y $A_2 \in B$ tales que $A = A_1 \sqcup_\varnothing A_2$. Tenemos entonces $A'_1 \in B'_1$ tal que $A_1 \prec A'_1$, por lo que tomando $A' = A'_1 \sqcup_\varnothing A_2$, tenemos que $A \prec A'$ y $A' \in \mathcal{B}_{\text{con}}[\![\Box]\!](B'_1, A)$.
- $b = Aat \cdot b_0$. Entonces existen una barba $A_1at \cdot b_0$ y un estado A_2 tales que $\mathsf{nd}(A_2) > t$ y $A = A_1 \cup (A_2 \upharpoonright t)$ de modo que se cumple además una de las dos condiciones siguientes:
  - $A_1at \cdot b_0 \in B$, $A_2 \in B_1$. En tal caso existe $A'_2 \in B'_1$ tal que $A_2 \prec A'_2$, de modo que $A'_2 \upharpoonright t = A_2 \upharpoonright t$, y en consecuencia $Aat \cdot b_0 \in \mathcal{B}_{\text{con}}[\![\Box]\!](B'_1, B)$.
  - $A_1at \cdot b_0 \in B_1$ y $A_2 \in B$. Entonces existirán una b-traza bs_0 y un estado A_0 tales que $b_0 = bs_0 \cdot A_0$, y por tanto existirá un estado A' tal que verifica $A_1at \cdot (bs_0 \cdot A') \in B'_1$ y $A_0 \prec A'$, de modo que tendremos
$$b' = Aat \cdot (bs_0 \cdot A') \in \mathcal{B}_{\text{con}}[\![\Box]\!](B'_1, B)$$

□

Demostración de la proposición 4.4.19. Continuidad de la elección externa.
Para simplificar la notación tomaremos los conjuntos

$$B = \mathcal{B}_{\text{con}}[\![\Box]\!](\mathsf{lub}(\mathcal{B}_1), \mathsf{lub}(\mathcal{B}_2)) \ \ \text{y} \ \ B' = \mathsf{lub}(\{\mathcal{B}_{\text{con}}[\![\Box]\!](B_{1i}, B_{2i}) \mid i \in \mathbb{N}\})$$

Hay que demostrar

1. Para cada $b' \in B'$ existe $b \in B$ tal que $b \prec b'$.
2. Para cada $bs \cdot A \in B$ existe un estado A' tal que $A \prec A'$ y $bs \cdot A' \in B'$.

Demostremos entonces los dos apartados de arriba:

1. Sea $b = bs \cdot A \in B'$. Supongamos primero que $bs = \epsilon$. Distinguiremos dos casos:
   - $\mathsf{nd}(A) < \infty$. En tal caso para cada $k \in \mathbb{N}$ existe $l_k \in \mathbb{N}$ de modo que
$$A \in \mathcal{B}_{\text{con}}[\![\Box]\!](B_{1l_k}, B_{2l_k})$$
     Por tanto para cada l_k existen dos estados $A_{1l_k} \in B_{1l_k}$ y $A_{2l_k} \in B_{2l_k}$ tales que $A = A_{1l_k} \sqcup_\varnothing A_{2l_k}$. Entonces se verifica
$$\mathsf{nd}(A) = \min(\mathsf{nd}(A_{1l_k}), \mathsf{nd}(A_{2l_k}))$$
     por lo que o bien $\mathsf{nd}(A) = \mathsf{nd}(A_{1l_k})$, o bien $\mathsf{nd}(A) = \mathsf{nd}(A_{2l_k})$. Sin pérdida de generalidad, podemos suponer que siempre se cumple la misma de dichas

igualdades. Supongamos en concreto que siempre es cierta la primera. Puesto que estamos considerando un alfabeto finito y un dominio de tiempo discreto podemos suponer además que todos los estados A_{1l_k} son iguales. Tomamos entonces $A_1 = A_{1l_1}$ verificándose $A_1 \in \mathcal{B}_1$. Puesto que para $i \leq j$ se verifica $B_{2i} \prec B_{2j}$ existe una secuencia de estados

$$A'_1 \prec A'_2 \prec A'_3 \prec \cdots$$

de manera que se tiene $A'_1 = A_{2l_1}$ y $A'_i \in B_{2l_i}$. Tomando entonces el estado

$$A_2 = \mathsf{lub}\{A'_{k_i} \mid i \in \mathbb{N}\}$$

se verifica que $A_2 \in \mathcal{B}_2$, y por tanto tenemos que $A = A_1 \sqcup_\varnothing A_2$ y por tanto concluimos que $A \in \mathcal{B}_{\text{con}}[\![\Box]\!](\mathcal{B}_1, \mathcal{B}_2)$.

- $\mathsf{nd}(A) = \infty$. Para cada $t \in \mathcal{T}$ existen $l_t \in \mathbb{N}$ y un estado A_t tales que

$$A_t \in \mathcal{B}_{\text{con}}[\![\Box]\!](B_{1l_t}, B_{2l_t}) \quad \text{y} \quad A \upharpoonright t = A_t \upharpoonright t$$

Existen entonces estados $A_{1t} \in B_{1l_t}$ y $A_{2t} \in B_{2l_t}$ tales que $A_{l_t} = A_{1\,l_t} \sqcup_\varnothing A_{2\,l_t}$ Puesto que estamos considerando un dominio de tiempo discreto y un alfabeto podemos suponer que

$$t' \geq t \quad \Rightarrow \quad A_{it} \upharpoonright t = A_{it'} \upharpoonright t \qquad i \in \{1,2\}$$

Tomando entonces

$$A_i = \bigcup_{t \in \mathcal{T}} (A_{it} \upharpoonright t) \qquad i \in \{1,2\}$$

se verifica que $A_i \in \mathcal{B}_i$, y además $A = A_1 \cup A_2 = A_1 \sqcup_\varnothing A_2 \in \mathcal{B}_{\text{con}}[\![\Box]\!](\mathcal{B}_1, \mathcal{B}_2)$.

Supongamos ahora que $b = A_1 a_1 t_1 \cdot (bs' \cdot A)$. Distinguimos dos casos:

- $\mathsf{nd}(A) < \infty$. Entonces se tiene que para todo $k \in \mathbb{N}$ existe $l_k \geq k$ tal que $b \in \mathcal{B}_{\text{con}}[\![\Box]\!](B_{1l_k}, B_{2l_k})$. Existen entonces una barba $b_{l_k} = A_{1l_k} a_1 t_1 \cdot (bs' \cdot A)$ y un estado A_{2l_k} tales que $\mathsf{nd}(A_{2l_k}) > t_1$, $A_1 = A_{1l_k} \cup A_{2l_k} \upharpoonright t_1$ verificándose además una de las siguientes condiciones:
  - $b_{l_k} \in B_{1l_k}$ y $A_{2l_k} \in B_{2l_k}$, o bien
  - $b_{l_k} \in B_{2l_k}$ y $A_{2l_k} \in B_{1l_k}$.

  Podemos suponer, sin pérdida de generalidad, que siempre se la misma de las dos opciones. Supongamos que se trata de la primera. Puesto que estamos considerando un alfabeto finito y un dominio de tiempo discreto, tenemos que el conjunto $\mathcal{A}_1$ es finito. En consecuencia podemos suponer que los conjuntos A_{1l_k} son todos iguales, tomando $A_1' = A_{1l_1}$. Puesto que para $i \leq j$ se tiene $B_{2l_i} \prec B_{2l_j}$, podemos suponer que para $i \leq j$ se tiene $A_{2l_i} \prec A_{2l_j}$. Entonces, tomando

$$A_2 = \mathsf{lub}\{A_{2l_i} \mid i \in \mathbb{N}\}$$

se tiene $A_2 \in \mathcal{B}_2$. Por tanto $A_1 = A_1' \cup (A_2 \upharpoonright t_1)$, de modo que $b \in \mathcal{B}_{\text{con}}[\![\Box]\!](\mathcal{B}_1, \mathcal{B}_2)$.

- $\mathsf{nd}(A) = \infty$. Entonces para todo $t \in \mathcal{T}$ existe $l_t \in \mathcal{T}$ tal que existe un estado A_t tal que $A_t \upharpoonright t = A \upharpoonright t$ y $A_1 a_1 t_1 \cdot (bs' \cdot A_t) \in \mathcal{B}_{\text{con}}[\![\Box]\!](B_{1l_t}, B_{2l_t})$. Existen entonces una barba $b_t = A_{1t} a_1 t_1 \cdot (bs' \cdot A_t)$ y un estado A_{2t} tales que $\mathsf{nd}(A_{2t}) > t_1$, $A_1 = A_{1t} \cup A_{2t} \upharpoonright t_1$, verificándose una de las siguientes condiciones:

  - $b_t \in B_{1l_t}$ y $A_{2t} \in B_{2t}$, o bien
  - $b_t \in B_{2l_t}$ y $A_{2t} \in B_{1t}$.

  Al igual que antes, sin pérdida de generalidad podemos suponer que siempre se da la misma de las dos opciones, pongamos que es la primera. Puesto que estamos considerando un alfabeto finito y un dominio de tiempo discreto, el conjunto $\mathcal{A}_1$ es finito y por tanto podemos suponer que todos los A_{1t} son iguales. Tomemos A_1' igual a uno cualquiera de ellos. Puesto que para $i \leq j$ se tiene $B_{2i} \prec B_{2j}$, obtenemos una secuencia

$$A_0' \prec A_1' \prec A_2' \prec A_3' \prec \cdots A_k' \prec A_{k+1}' \prec \cdots$$

  tal que $A_i' \in B_{2(l_0+i)}$. Tomando $A_2 = \mathsf{lub}\{A_i' \mid i \geq 0\}$ se verifica entonces que $A_2 \in \mathcal{B}_2$, y por tanto $A_1 = A_1' \cup (A_2 \upharpoonright t_1)$, de donde concluimos finalmente $b \in \mathcal{B}_{\text{con}}[\![\Box]\!](\mathcal{B}_1, \mathcal{B}_2)$.

2. Sea ahora $b = bs \cdot A \in B$. Supongamos que $bs = \epsilon$. en tal caso existen $A_1 \in \mathcal{B}_1$ y $A_2 \in \mathcal{B}_2$ tales que $A = A_1 \sqcup_{\varnothing} A_2$. Tenemos cuatro posibilidades:

   (a) $\mathsf{nd}(A_1) < \infty$ y $\mathsf{nd}(A_2) < \infty$. Supongamos sin pérdida de generalidad, que $\mathsf{nd}(A_1) \leq \mathsf{nd}(A_2)$. Para cada $k \in \mathbb{N}$ existen l_k, $m_k \in \mathbb{N}$ tales que $A_1 \in B_{1l_k}$ y $A_2 \in B_{2m_k}$. Entonces podemos distinguir los siguientes casos:

   - Si $l_k = m_k$, tomando $j_k = l_k$, tenemos que

$$A \in \mathcal{B}_{\text{con}}[\![\Box]\!](B_{1j_k}, B_{2j_k})$$

- Si $m_k < l_k$, tomando $j_k = l_k$, tenemos que existe $A'_2 \in B_{1l_k}$ tal que $A_2 \prec A'_2$, de modo que

$$A = A_1 \sqcup_\varnothing A_2 = A_1 \sqcup_\varnothing A'_2 \in \mathcal{B}_{\text{con}}[\![\Box]\!](B_{1j_k}, B_{2j_k})$$

- Si $l_k < m_k$, existe $j_k \geq m_k$ tal que $A_1 \in B_{1j_k}$. Por otra parte existe $A'_2 \in A_{2j_k}$ tal que $A_2 \prec A'_2$, y por tanto obtenemos

$$A = A_1 \sqcup_\varnothing A_2 = A_1 \sqcup_\varnothing A'_2 \in \mathcal{B}_{\text{con}}[\![\Box]\!](B_{1j_k}, B_{2j_k})$$

En consecuencia, cualquier caso se tiene que

$$\forall k \in \mathbb{N}\ \exists j_k \geq k: \ A \in \mathcal{B}_{\text{con}}[\![\Box]\!](B_{1\,j_k}, B_{2\,j_k})$$

de modo que $A \in B'$.

(b) $\mathsf{nd}(A_1) = \mathsf{nd}(A_2) = \infty$. Para cada $t \in \mathcal{T}$ existen índices $m_t, l_t \in \mathbb{N}$ y estados $A_{1m_t} \in B_{1m_t}$ y $A_{2l_t} \in B_{2l_t}$ tales que $A_1 \upharpoonright t = A_{1\,m_t} \upharpoonright t$ y $A_2 \upharpoonright t = A_{2\,l_t} \upharpoonright t$. Tomamos entonces $j_t = max(l_t, m_t)$, y entonces

- Si $l_t = m_t$, tomamos $A_{j_t} = A_{1m_t} \sqcup_\varnothing A_{2l_t}$
- Si $l_t < m_t$. Puesto que $B_{2l_t} \prec B_{2m_t}$ existe $A'_2 \in B_{2m_t}$ tal que $A_{2l_t} \prec A'_2$, y entonces tomamos $A_{j_t} = A_{1\,m_t} \sqcup_\varnothing A'_2$.
- Si $m_t < l_t$. Puesto que $B_{1m_t} \prec B_{1l_t}$ existe $A'_1 \in B_{1l_t}$ tal que $A_{1m_t} \prec A'_1$, y entonces tomamos $A_{j_t} = A_{2l_t} \sqcup_\varnothing A'_1$.

En todos casos tenemos que $A \upharpoonright t = A_{j_t} \upharpoonright t$, y por tanto $A \in B'$.

(c) $t = \mathsf{nd}(A_1) < \infty$, $\mathsf{nd}(A_2) = \infty$. Existen $l \in \mathbb{N}$ y un estado $A_{2l} \in B_{2l}$ tales que $A_2 \upharpoonright t = A_{2l} \upharpoonright t$. Puesto que para $i \leq j$ se tiene $B_{2i} \prec B_{2j}$, para cada $m \geq l$ existe $A_{2m} \in A_{2m}$ tal que $A_{2l} \prec B_{2m}$, y por tanto $A_2 \upharpoonright t = A_{2m} \upharpoonright t$. Por otro lado, para cada $k \geq l$ existe $m_k \geq k$ tal que $A_1 \in B_{1m_k}$. En consecuencia tendremos

$$A = A_1 \sqcup_\varnothing A_2 = A_1 \sqcup_\varnothing A_{2m_k} \in \mathcal{B}_{\text{con}}[\![\Box]\!](B_{1\,m_k}, B_{2\,m_k})$$

y por tanto $A \in B'$.

(d) $t = \mathsf{nd}(A_2) < \infty$, $\mathsf{nd}(A_1) = \infty$, que es simétrico al caso anterior.

Consideremos ahora $b = Aa_1t_1 \cdot (bs' \cdot A) \in B$. Entonces existen una barba b_1 y un estado A_2 tales que

$$b_1 = A_1a_1t_1 \cdot (bs' \cdot A), \quad \mathsf{nd}(A_2) \geq t_1 A = A_1 \cup (A_2 \upharpoonright t_1)$$

y o bien, $b_1 \in \mathcal{B}_1$ y $A_2 \in \mathcal{B}_2$, o bien, $b_1 \in \mathcal{B}_2$ y $A_2 \in \mathcal{B}_1$. Supongamos que se da el primer caso; el segundo sería simétrico. Entonces puesto que $\mathsf{nd}(A_2) \geq t_1$ existe $l \in \mathbb{N}$ de manera que existe un estado $A_{2l} \in B_{2l}$ tal que $A_2 \upharpoonright t_1 = A_{2l} \upharpoonright t_1$. Además, puesto que $B_l \prec B_{l'}$ para $l' \geq l$, existe un estado $A_{2l'} \in B_{2l'}$ tal que $A_2 \upharpoonright t_1 = A_{2l'} \upharpoonright t_1$. Ahora podemos distinguir dos casos:

- $\mathsf{nd}(A) < \infty$, entonces para cada $k \geq l$ existe $m_k \geq k$ tal que $b_1 \in B_{1m_k}$, de modo que tenemos $b \in \mathcal{B}_{\text{con}}[\![\Box]\!](B_{1m_k}, B_{2m_k})$, por lo que $b \in B'$.
- $\mathsf{nd}(A) = \infty$. Fijado $t \in \mathcal{T}$, existen $m_t \in \mathbb{N}$ y un estado A_{m_t} de manera que

$$A_1 a_1 t_1 \cdot (bs' \cdot A_{m_t}) \in B_{1m_t} \quad \text{y} \quad A_{m_t} \upharpoonright t = A \upharpoonright t$$

  Consideramos el índice $j_t \in \mathbb{N}$ y el estado A_{j_t} elegidos de la siguiente manera:

  - Si $l \leq m_t$, tomamos $j_t = m_t$ y $A_{j_t} = A_{m_t}$.
  - Si $m_t < l$, tomamos $j_t = l$. Por otra parte, puesto que $B_{1m_t} \prec B_{1l}$, existe un estado A_l tal que

$$A_1 a_1 t_1 \cdot (bs' \cdot A_l) \in B_{1m_t} \quad \text{y} \quad A_{m_t} \prec A_l$$

    Tomamos entonces $A_{j_t} = A_l$.

  En ambos casos tenemos que

$$A a_1 t_1 \cdot (bs' \cdot A_{1\,j_t}) \in \mathcal{B}_{\text{con}}[\![\Box]\!](B_{1\,j_t}, B_{2\,j_t}) \quad \text{y} \quad A_{j_t} \upharpoonright t = A \upharpoonright t$$

  Puesto que esto lo podemos hacer para cada $t \in \mathcal{T}$, concluimos inmediatamente que $b \in B'$.

$\Box$

A.3 Demostraciones del Capítulo 6

Demostración del lema 6.1.11.

Consideremos la computación que genera el estado A:

$$P \parallel_G Q = R_0 \rightarrowtail^* R_1' \xrightarrow{\tau t_1} R_2 \rightarrowtail^* R_2' \xrightarrow{\tau t_2} \cdots$$

Tomemos $t^i = \sum_{j=1}^{i-1} t_j$. Puesto que hemos tomado $t \leq \mathsf{nd}(A)$, existe $n \in \mathbb{N}$ tal que:

- Para $i < n$ se tiene $R_i \Downarrow$,
- $t^n \leq t$ y
- O bien $R_n \Uparrow$, o bien $R_n \Downarrow$, R_n' es estable y si $R_n' \xrightarrow{\tau t_n} R_{n+1}$, entonces $t^{n+1} > t$.

Haremos la demostración por inducción sobre el número de transiciones necesarias para llegar hasta R_n'.

- Si el número de transiciones es cero, tenemos que $R' = P \parallel_G Q$. Si $R' \Uparrow$ tenemos que $P \Uparrow$ o $Q \Uparrow$. Supondremos que se da el primer caso. Tenemos entonces que $\mathcal{A}(P) = \Big\{\{\Omega 0\}\Big\}$, y tomando como A_Q cualquier estado de Q tenemos el resultado deseado :

  $$\{\Omega 0\} = \{\Omega 0\} \sqcup_G A_Q \quad \text{y} \quad \mathcal{A}(P \parallel_G Q) = \Big\{\{\Omega 0\}\Big\}$$

  Si $R' \Downarrow$ tenemos que $P \Downarrow$ y $Q \Downarrow$, por lo que

  $$A \upharpoonright t = \mathsf{TA}(R) \upharpoonright t = (\mathsf{TA}(P) \upharpoonright t) \sqcup_G (\mathsf{TA}(Q) \upharpoonright t)$$

  Entonces, tomando como A_P cualquier estado de P y como A_Q cualquier estado de Q, obtenemos el resultado deseado puesto que

  $$A_P \upharpoonright t = \mathsf{TA}(P) \upharpoonright t \quad \text{y} \quad A_Q \upharpoonright t = \mathsf{TA}(Q) \upharpoonright t$$

- Si el número de transiciones es mayor que 0, podemos distinguir los siguientes casos, en función de la forma de dependiendo de la primera de dichas transiciones:

  $\mathbf{P \parallel_G Q \rightarrowtail P' \parallel_G Q}$. En este caso podemos suponer que $P' \parallel_G Q \Downarrow$, pues si tuviéramos que $P' \parallel_G Q \Uparrow$ obtendríamos $P \parallel_G Q \Uparrow$, lo cual no es compatible con las propiedades la computación considerada. Tenemos entonces que $A \in \mathcal{A}(P' \parallel_G Q)$, y por hipótesis de inducción existen $A_P \in \mathcal{A}(P')$ y $A_Q \in \mathcal{A}(Q)$ verificando las condiciones del enunciado. de donde se concluye el resultado buscado, teniendo en cuenta que $A_P \in \mathcal{A}(P)$.

$\mathbf{P} \parallel_{\mathbf{G}} \mathbf{Q} \rightarrowtail \mathbf{P} \parallel_{\mathbf{G}} \mathbf{Q'}$. Este caso es totalmente simétrico al anterior.

$\mathbf{P} \parallel_{\mathbf{G}} \mathbf{Q} \xrightarrow{\tau t_1} \mathbf{P'} \parallel_{\mathbf{G}} \mathsf{Upd}(\mathbf{Q}, \mathbf{t_1})$. En este caso tenemos que

$$A = (\mathsf{TA}(R) \upharpoonright t_1) \cup (A' + t_1)$$

siendo A' un estado del proceso $P' \parallel_G \mathsf{Upd}(Q, t_1)$. Entonces por hipótesis de inducción existen $A'_P \in \mathcal{A}(P)$ y $A'_Q \in \mathcal{A}(\mathsf{Upd}(Q, t_1))$ verificando las condiciones del enunciado con respecto al estado A' y al instante $t - t_1$. Esto es:

- $\mathsf{nd}(A') \geq t - t_1 \;\Rightarrow\; (A'_P \upharpoonright (t - t_1)) \sqcup_G (A'_Q \upharpoonright (t - t_1)) \subseteq A' \upharpoonright (t - t_1)$.
- $\mathsf{nd}(A') < t - t_1 \;\Rightarrow\; A'_P \sqcup_G A'_Q \ll A'$.

Tomamos entonces los estado A_P y A_Q definidos de la siguiente forma:

$$A_P = \mathsf{TA}(P) \upharpoonright t_1 \cup (A'_P + t_1) \quad \text{y} \quad A_Q = \mathsf{TA}(Q) \upharpoonright t \cup (A'_Q + t_1)$$

de donde obtenemos el resultado deseado, ya que $A_P \in \mathcal{A}(P)$, $A_Q \in \mathcal{A}(Q)$ y

$$\mathsf{TA}(P \parallel_G Q) \upharpoonright t_1 = (\mathsf{TA}(P) \upharpoonright t_1) \sqcup_G (\mathsf{TA}(Q) \upharpoonright t_1)$$

$\mathbf{P} \parallel_{\mathbf{G}} \mathbf{Q} \xrightarrow{\tau t_1} \mathsf{Upd}(\mathbf{P'}, \mathbf{t_1}) \parallel_{\mathbf{G}} \mathbf{Q}$. Este caso es totalmente simétrico al anterior.

□

Demostración del lema 6.1.13.

Consideremos la computación que genera el estado A_P

$$P = P_1 \rightarrowtail^* P_1 \xrightarrow{\tau t_1^P} P_2 \rightarrowtail^* \cdots$$

y la que genera A_Q

$$Q = Q_1 \rightarrowtail^* Q_1 \xrightarrow{\tau t_1^Q} Q_2 \rightarrowtail^* \cdots$$

Tomemos $t^{P,i} = \sum_{j=1}^{i-1} t_1^P$ y $t^{Q,i} = \sum_{j=1}^{i-1} t_1^Q$. Entonces han de existir m y n tales que:

- Para $i < m$ se verifica que $P_i \Downarrow$ y para cada $j < n$ se verifica $Q_j \Downarrow$.
- O bien $P_m \Uparrow$, o bien $P_m \Downarrow$, $t^{P,m} \leq t$, P'_m es estable y si $P'_m \xrightarrow{\tau t_{m+1}^P} P_{m+1}$ entonces $t^{P,m+1} > t$.
- O bien $Q_n \Uparrow$, o bien $Q_n \Downarrow$, $t^{Q,n} \leq t$, Q'_n es estable y si $Q'_n \xrightarrow{\tau t_{n+1}^Q} Q_{n+1}$ entonces $t^{Q,n+1} > t$.

Haremos la demostración por inducción sobre la suma del número de transiciones necesarias para llegar hasta P'_m y Q'_n, que notaremos por s:

s = 0. Tenemos que $P'_m = P$ y $Q'_n = Q$. Si $P \Uparrow$ o $Q \Uparrow$ tendríamos que $P \|_G Q \Uparrow$. Cualquier estado A' verifica

$$\{\Omega 0\} \sqcup_G A' = \{\Omega 0\} \quad \text{y} \quad \mathcal{A}(P \parallel_G Q) = \Big\{\{\Omega 0\}\Big\}$$

Si $P \parallel_G Q$ es convergente y estable, cualquier $A \in \mathcal{A}(P \parallel_G Q)$ verifica

$$A \upharpoonright t = (\mathsf{TA}(P) \upharpoonright t) \sqcup_G (\mathsf{TA}(Q) \upharpoonright t)$$

y por el otro lado tenemos que $A_P \upharpoonright t = \mathsf{TA}(P) \upharpoonright t$ y $A_Q = \mathsf{TA}(Q) \upharpoonright t$.

s > 0. En tal caso podemos distinguir los casos que siguen, en función de la forma de la primera transición:

$\mathbf{P} \rightarrowtail \mathbf{P'}$. Entonces tenemos que $A_P \in \mathcal{A}(P')$. Por hipótesis de inducción existe $A \in \mathcal{A}(P' \parallel_G Q)$ verificando las condiciones del enunciado. Por último tenemos que $A \in \mathcal{A}(P \parallel_G Q)$.

$\mathbf{Q} \rightarrowtail \mathbf{P'}$. Este caso es simétrico al anterior.

$\mathbf{P} \xrightarrow{\tau t_1} \mathbf{P'}$, $\mathsf{idle}(\mathbf{Q}) \geq \mathbf{t_1}$ y $\mathsf{stb}(Q)$. Tenemos que en virtud de la proposición 3.4.7, existe un estado $A'_Q \in \mathcal{A}(\mathsf{Upd}(Q, t_1))$ tal que

$$A_Q = (\mathsf{TA}(Q) \upharpoonright t) \cup (A'_Q + t_1)$$

Por otro lado, existe $A'_P \in \mathcal{A}(P')$ tal que

$$A_P = (\mathsf{TA}(P) \upharpoonright t_1) \cup (A + t_1)$$

Por hipótesis de inducción existe $A' \in \mathcal{A}(P' \parallel_G \mathsf{Upd}(Q, t_1))$ verificando las condiciones del enunciado:

$$\left.\begin{array}{c} \mathsf{nd}(A'_Q) < t - t_1 \\ \text{ó} \\ \mathsf{nd}(A'_P) < t - t_1 \end{array}\right\} \Rightarrow A' \ll A'_P \sqcup_G A'_P$$

$$\left.\begin{array}{c} \mathsf{nd}(A'_Q) \geq t - t_1 \\ \text{y} \\ \mathsf{nd}(A'_P) \geq t - t_1 \end{array}\right\} \Rightarrow A' \upharpoonright (t - t_1) \subseteq (A'_P \upharpoonright (t - t_1)) \sqcup_G (A'_Q \upharpoonright (t - t_1))$$

Basta entonces tomar $A = \mathsf{TA}(P \parallel_G Q) \upharpoonright t \cup (A' + t)$ de modo que se verifican las condiciones del enunciado.

$\mathbf{Q} \xrightarrow{\tau t_1} \mathbf{Q'}$, $\mathsf{idle}(\mathbf{P}) \geq \mathbf{t_1}$ y $\mathsf{stb}(Q)$. Este caso es totalmente simétrico al anterior.

□

Demostración de la proposición 6.1.15.
Si $b = bs \cdot A \in \mathsf{Barb}(P \parallel_G Q)$ existe un proceso R tal que

$$P \parallel_G Q \stackrel{bs}{\Longrightarrow} R \qquad \text{y} \qquad A \in \mathcal{A}(R)$$

Haremos la demostración por inducción sobre la profundidad l de la deducción de la transición generalizada $P \parallel_G Q \stackrel{bs}{\Longrightarrow} R$.

l = 0. Es decir, tenemos que $R = P \parallel_G Q$, $bs = \epsilon$ y $A \in \mathcal{A}(P \parallel_G Q)$. Entonces aplicando la proposición 6.1.12 tenemos el resultado.

l > 0. En este caso tenemos varias posibilidades:

- $P \parallel_G Q \rightarrowtail P' \parallel_G Q'$ y $P' \parallel_G Q' \stackrel{bs}{\Longrightarrow} R$. Por la forma de las reglas que definen la semántica operacional sabemos que o bien $P \rightarrowtail P'$ y $Q = Q'$, o viceversa $Q \rightarrowtail Q'$ y $P = P'$. En ambos casos el resultado es inmediato por inducción.
- $P \parallel_G Q \xrightarrow{\tau t} P' \parallel_G Q'$, $P' \parallel_G Q' \stackrel{bs'}{\Longrightarrow} R$, $bs' \neq \epsilon$ y $bs = (\mathsf{TA}(P \parallel_G Q) \upharpoonright t, t) \sqcup bs'$.
  Por la forma de las reglas que definen la semántica operacional tenemos los siguientes casos:
  - $P \xrightarrow{\tau t} P'$, $Q' = \mathsf{Upd}(Q, t)$, $\mathsf{stb}(Q)$ e $\mathsf{idle}(Q) \geq t$, o bien
  - $Q \xrightarrow{\tau t} Q'$, $P' = \mathsf{Upd}(P, t)$, $\mathsf{stb}(P)$ e $\mathsf{idle}(P) \geq t$.

  Puesto que ambos casos son simétricos, resolveremos sólamente el primero. Puesto que $P' \parallel_G \mathsf{Upd}(Q, t) \stackrel{bs'}{\Longrightarrow} R$ tenemos que

  $$bs' \cdot A \in \mathsf{Barb}(P' \parallel_G \mathsf{Upd}(Q, t))$$

  y por hipótesis de inducción existen

  $$b_1' \in \mathsf{Barb}(P'), \quad b_2' \in \mathsf{Barb}(\mathsf{Upd}(Q, t)) \quad \text{y} \quad b'' \in b_1' \parallel_G b_2'$$

  tales que $b'' \ll bs' \cdot A$. Tomamos ahora

  $$b_1 = (\mathsf{TA}(P) \upharpoonright t, t) \sqcup b_1' \quad \text{y} \quad b_2 = (\mathsf{TA}(Q) \upharpoonright t, t) \sqcup b_2'$$

  verificándose $b_1 \in \mathsf{Barb}(P)$ y $b_2 \in \mathsf{Barb}(Q)$. Tomamos entonces

  $$b' = (\mathsf{TA}(P) \upharpoonright t \sqcup_G \mathsf{TA}(Q) \upharpoonright t, t) \sqcup b''$$

  y tendremos que $b' \in b_1 \parallel_G b_2$ y $b' \ll b$.

- $P \parallel_G Q \xrightarrow{at} P' \parallel_G Q'$, $P' \parallel_G Q' \overset{bs'}{\Longrightarrow} R$ y $bs = (\mathsf{TA}(P \parallel_G Q) \upharpoonright t, t)at \cdot bs'$. Por la forma de las reglas que definen la semántica operacional tenemos las siguientes posibilidades:
  - $a \notin G$, $P \xrightarrow{at} P'$, $Q' = \mathsf{Upd}(Q, t)$, $\mathsf{stb}(Q)$ e $\mathsf{idle}(Q) \geq t$. Puesto que

$$P' \parallel_G \mathsf{Upd}(Q, t) \overset{bs'}{\Longrightarrow} R$$

    tendremos que $bs' \cdot A \in \mathsf{Barb}(P' \parallel_G \mathsf{Upd}(Q, t))$, y por hipótesis de inducción tendremos que existirán

$$b_1' \in \mathsf{Barb}(P'), \quad b_2' \in \mathsf{Barb}(\mathsf{Upd}(Q, t)) \quad \text{y} \quad b'' \in b_1' \parallel_G b_2'$$

    tales que $b'' \ll bs' \cdot A$. Tomamos entonces las barbas

$$b_1 = (\mathsf{TA}(P) \upharpoonright t)at \cdot b_1' \quad \text{y} \quad b_2 = (\mathsf{TA}(Q) \upharpoonright t, t) \sqcup b_2'$$

    que verifican $b_1 \in \mathsf{Barb}(P)$ y $b_2 \in \mathsf{Barb}(Q)$. Tomemos entonces

$$b' = (\mathsf{TA}(P) \upharpoonright t \sqcup_G \mathsf{TA}(Q) \upharpoonright t)at \cdot b''$$

    y se verificará que $b' \in b_1 \parallel_G b_2$ y $b' \ll b$.
  - $a \notin G$, $Q \xrightarrow{at} Q'$, $P' = \mathsf{Upd}(P, t)$, $\mathsf{stb}(P)$ e $\mathsf{idle}(P) \geq t$. Que se resuelve de manera simétrica al anterior.
  - $a \in G$, $P \xrightarrow{at} P'$ y $Q \xrightarrow{at} Q'$. Puesto que

$$P' \parallel_G Q' \overset{bs'}{\Longrightarrow} R$$

    existirán

$$b_1' \in \mathsf{Barb}(P'), \quad b_2' \in \mathsf{Barb}(Q') \quad \text{y} \quad b'' \in b_1' \parallel_G b_2'$$

    tales que $b'' \ll bs' \cdot A$. Tomemos entonces las barbas

$$b_1 = (\mathsf{TA}(P) \upharpoonright t)at \cdot b_1' \quad \text{y} \quad b_2 = (\mathsf{TA}(Q) \upharpoonright t)at \cdot b_2'$$

    que verifican $b_1 \in \mathsf{Barb}(P)$ y $b_2 \in \mathsf{Barb}(Q)$. Tomemos entonces

$$b' = (\mathsf{TA}(P) \upharpoonright t \sqcup_G \mathsf{TA}(Q) \upharpoonright t)at \cdot b''$$

    y se verificará que $b' \in b_1 \parallel_G b_2$ y $b' \ll b$.

$\square$

Demostración de la proposición 6.1.16.

Supongamos que $b_1 = bs_1 \cdot A_1$ y $b_2 = bs_2 \cdot A_2$. Existirán entonces P_1 y Q_1 tales que

$$\begin{array}{lcl} P \stackrel{bs_1}{\Longrightarrow} P_1, & & A_1 \in \mathcal{A}(P_1) \\ Q \stackrel{bs_2}{\Longrightarrow} P_1 & \text{y} & A_2 \in \mathcal{A}(P_2) \end{array}$$

La demostración la haremos por inducción sobre la suma de las longitudes de la deducciones de $P \stackrel{bs_1}{\Longrightarrow} P_1$ y $Q \stackrel{bs_2}{\Longrightarrow} P_1$, que notaremos por l. El caso base se da cuando $P_1 = P$ y $Q_1 = Q$. En tal caso tendremos que $A_1 \in \mathcal{A}(P)$ y $A_2 \in \mathcal{A}(Q)$, de modo que el resultado se sigue por aplicación de la proposición 6.1.14.

Respecto al caso inductivo tenemos las siguientes posibilidades:

- $P \rightarrowtail P' \stackrel{bs_1}{\Longrightarrow} P_1$. Tenemos entonces que

$$P \parallel_G Q \rightarrowtail P' \parallel_G Q$$

  De modo que el resultado buscado se sigue inmediatamente por aplicación de la hipótesis de inducción.

- El caso en el que $Q \rightarrowtail Q'$ es totalmente simétrico al anterior.

- $P \xrightarrow{\tau t} P' \stackrel{bs'_1}{\Longrightarrow} P_1$, $bs'_1 \neq \epsilon$, $\mathsf{stb}(Q)$ e $\mathsf{idle}(Q) \geq t$, tomemos $b'_1 = bs'_1 \cdot A_1$. Ahora se verificará que

$$P \parallel_G Q \xrightarrow{\tau t} P' \parallel_G \mathsf{Upd}(Q, t)$$

  Tenemos entonces que la barba b_2 será de la forma

$$b_2 = (\mathsf{TA}(Q) \upharpoonright t, t) \sqcup b'_2$$

  con $b'_2 \in \mathsf{Barb}(\mathsf{Upd}(Q, t))$. Entonces la barba b la será de la forma

$$b = (A, t) \sqcup b''$$

  donde $A = (\mathsf{TA}(P) \upharpoonright t \sqcup_G \mathsf{TA}(Q) \upharpoonright t)$ y $b'' \in b'_1 \parallel_G b'_2$. Por hipótesis de inducción existe $b''' \in \mathsf{Barb}(P' \parallel_G \mathsf{Upd}(Q, t))$ tal que $b''' \ll b''$. Tomamos entonces

$$b' = (\mathsf{TA}(P \parallel_G Q) \upharpoonright t, t) \sqcup b'''$$

  y se verificará que $b' \in \mathsf{Barb}(P \parallel_G Q)$ y $b' \ll b$.

- $Q \xrightarrow{\tau t} Q' \stackrel{bs'_2}{\Longrightarrow} Q_1$, $bs'_2 \neq \epsilon$, $\mathsf{stb}(P)$ e $\mathsf{idle}(P) \geq t$, caso totalmente simétrico al anterior.

- $P \xrightarrow{at} P' \xRightarrow{bs'_1} P_1$, $\mathsf{stb}(Q)$, $\mathsf{idle}(Q) \geq t$ y $a \notin G$. En este caso tenemos que

$$b_1 = (\mathsf{TA}(P) \upharpoonleft t)at \cdot b'_1$$

Tenemos además que la barba b_2 será de la forma

$$b_2 = (\mathsf{TA}(Q) \upharpoonleft t, t) \sqcup b'_2$$

con $b'_2 \in \mathsf{Barb}(\mathsf{Upd}(Q, t))$. Podemos suponer sin pérdida de generalidad* que

$$b = (\mathsf{TA}(P) \upharpoonleft t \sqcup_G \mathsf{TA}(Q) \upharpoonleft t)at \cdot b''$$

con $b'' \in b'_1 \parallel_G b'_2$. Por hipótesis de inducción existe entonces

$$b''' \in \mathsf{Barb}(P' \parallel_G \mathsf{Upd}(Q, t))$$

tal que $b''' \ll b''$. Tomamos

$$b' = (\mathsf{TA}(P \parallel_G Q) \upharpoonleft t)at \cdot b'''$$

y tendremos que $b' \ll b$ y $b' \in \mathsf{Barb}(P \parallel_G Q)$.

- $Q \xrightarrow{at} Q' \xRightarrow{bs'_1} Q_1$, $\mathsf{stb}(P)$, $\mathsf{idle}(P) \geq t$ y $a \notin G$, caso simétrico al anterior.

- $P \xrightarrow{at} P' \xRightarrow{bs'_1} P_1$, $\mathsf{stb}(Q)$, $\mathsf{idle}(Q) \geq t$ y $a \in G$. En este caso, si $Q \stackrel{at}{\not\longrightarrow}$ tendremos que $b_1 \parallel_G b_2 = \varnothing$, se ha de tener por tanto

$$Q \xrightarrow{at} Q' \xRightarrow{bs'_2} Q_1$$

Tomemos $b'_1 = bs'_1 \cdot A_1$ y $b'_2 = bs'_2 \cdot A_2$, y tendremos entonces que

$$b = (\mathsf{TA}(P) \upharpoonleft t \sqcup_G \mathsf{TA}(Q) \upharpoonleft t)at \cdot b''$$

con $b'' \in b'_1 \parallel_G b'_2$. Por hipótesis de inducción existe

$$b''' \in \mathsf{Barb}(P' \parallel_G Q')$$

tal que $b''' \ll b''$. Tomamos entonces

$$b' = (\mathsf{TA}(P \parallel_G Q) \upharpoonleft t)at \cdot b'''$$

y tendremos que $b' \ll b$ y $b' \in \mathsf{Barb}(P \parallel_G Q)$.

□

*Podría ser que $b_2 = (\mathsf{TA}(Q) \upharpoonleft t)a't \cdot b''_2$ para algún $a' \in Act$ y $b = (\mathsf{TA}(P) \upharpoonleft t \sqcup_G \mathsf{TA}(Q) \upharpoonleft t)a't \cdot b''$ con $b'' \in (\varnothing a0 \cdot b'_1) \parallel_G b''_2$, pero en tal caso tendríamos que $Q \xrightarrow{a't} Q'$ por lo que se proseguiría como en el último caso.

Bibliografía

[AJ94] L. ACETO Y A. JEFFREY. A complete axiomatization of timed bisimulation for a class of timed regular behaviours. Informe técnico 4/94, University of Sussex (1994).

[Bar91] G. BARRET. The fixed point theory of unbounded non-determinism. *Formal Aspects of Computing* **3**, 110–128 (1991).

[BB93] J. C. M. BAETEN Y J. A. BERGSTRA. Real time process algebra. *Formal Aspects of Computing* **3**, 142–188 (1993).

[BDS95] J. BRYANS, J. DAVIES Y S. SCHNEIDER. Towards a denotational semantics for ET-LOTOS. En "CONCUR '95", páginas 249–263. Springer-Verlag (1995). LNCS 962.

[BHR84] S. D. BROOKES, C. A. R. HOARE Y A. W. ROSCOE. A theory of communicating sequential processes. *Journal of the Association for Computing Machinery* **31**(3), 560–599 (1984).

[BK84] J. A. BERGSTRA Y J. W. KLOP. Process algebra for syncronous communication. *Information and Control* **60**, 109–137 (1984).

[BR85] S. D. BROOKS Y A. W. ROSCOE. An improved failures model for communicating processes. En "Seminar on Concurrency Proceedings", páginas 281–305, Pittsburg, 1984 (1985). Springer-Verlag. LNCS 197.

[BW90] J.C.M. BAETEN Y W.P. WEIJLAND. "Process Algebra". Cambridge Tracts in Computer Science 18. Cambridge University Press (1990).

[CdV96] F. CUARTERO, D. DE FRUTOS Y V. VALERO. A denotational semantics for reactive PCSP. En "Third AMAST Workshop in Real Time Programming" (Marzo 1996). Aparecerá publicado en World Sientific: AMAST Series in Computing.

[CH90] R. CLEAVELAND Y M. HENNESSY. Priorities in process algebras. *Information and Computation* **87**, 58–77 (1990).

[Chr90] I. CHRISTOFF. "Testing Equivalences for Probabilistic Processes". Tesis Doctoral, Department of Computer Systems. Uppsala University (1990).

[CW95] J. CAMILLERI Y G. WINSKEL. CCS with priority choice. *Information and Computation* **116**, 26–37 (1995).

[Dav91] J. DAVIS. "Specification and Proof in Real-Time Systems". Tesis Doctoral, University of Oxford (Enero 1991).

[dFLL+95] D. DE FRUTOS, G. LEDUC, L. LÉONARD, L. F. LLANA-DÍAZ, C. MIGUEL, J. QUEMADA Y G. RABAY. Time Extended LOTOS. En J. QUEMADA, editor, "Working Draft on Enhancements to LOTOS". ISO/IEC JTC1/SC21/WG1 (Noviembre 1995).

[dNH84] R. DE NICOLA Y M. C. B. HENNESSY. Testing equivalences for processes. *Theoretical Computer Science* **34**, 83–133 (1984).

[DS95] J. DAVIES Y S. SCHNEIDER. A brief history of timed CSP. *Theoretical Computer Science* **138**, 243–271 (1995).

[GJS90] A. GIACALONE, C. JOU Y S.A. SMOLKA. Algebraic reasoning for probabilistic concurrent systems. En "Proceedings of Working Conference on Programming Concepts and Methods, IFIP TC 2" (1990).

[Gre95] C. GREGORIO. El tiempo como factor de prioridad en un álgebra probabilística. Trabajo de Doctorado. Dept. Informática y Automática. Universidad Complutense de Madrid (1995).

[Hen88] M. HENNESSY. "Algebraic Theory of Processes". MIT Press (1988).

[Hen90] M. HENNESSY. "The Semantics of Programming Languages". John Wiley and Sons (1990).

[Hen92] M. HENNESSY. Concurrent testing of processes. Informe técnico CS-1191, University of Sussex (Julio 1992).

[Hoa85] C. A. R. HOARE. "Communicating Sequential Processes". Prentice Hall (1985).

[HR95] M. HENNESSY Y T. REGAN. A process algebra for timed systems. *Information and Computation* **117**, 221–239 (1995).

[Jef92] A. JEFFREY. Translating timed process algebra into priorized process algebra. En "Symposium on Real Time and Fault Tolerant Systems". Springer-Verlag (1992). LNCS 581.

[Klu93] A. S. KLUSENER. "Models and Axioms for a Fragment of Real Time Process Algebra". Tesis Doctoral, Technische Universiteit Eindohven (Diciembre 1993).

[LdFN95] L.F. LLANA-DÍAZ, D. DE FRUTOS-ESCRIG Y M. NÚÑEZ. Alternative definition for internal action prefix in TE-LOTOS. Informe técnico 14-95, Universidad Complutense de Madrid (1995).

[LdFN96] L. F. LLANA-DÍAZ, D. DE FRUTOS Y M. NÚÑEZ. Testing semantics for urgent process algebras. En "Third AMAST Workshop in Real Time Programming" (Marzo 1996). Aparecerá publicado en World Sientific: AMAST Series in Computing.

[LL94] G. LEDUC Y L. LÈONARD. An enhanced version of timed LOTOS and its application to a case study. En R. TENNEY, P. AMER Y Ü. UYAR, editores, "FORTE '93". North Holland (1994).

[Lla93] L. F. LLANA-DÍAZ. Semántica de procesos con paralelismo real. Trabajo de Doctorado. Dept. Informática y Automática. Universidad Complutense de Madrid (1993).

[LOT88] LOTOS. A formal description technique based on the temporal ordering of observational behaviour. IS 8807, TC97/SC21 (1988).

[Low93] G. LOWE. "Probabilities and Priorities in Timed CSP". Tesis Doctoral, Oxford University (1993).

[LR96] L. F. LLANA-DÍAZ Y G. RABAY. Defining equivalences between timed/actions graphs and timed actions graphs. Informe técnico 24-96, Universidad Complutense de Madrid (1996).

[MG85] J. MESSEGUER Y J. GOGEN. Initiallity, induction and computability. En M. NIVAT Y J. REYNOLDS, editores, "Algebraic Methods in Semantics". Cambridge University Press (1985).

[Mil80] R. MILNER. "A Calculus of Communicating Systems". Springer-Verlag (1980). LNCS 92.

[Mil83] R. MILNER. Calculi for synchrony and asynchrony. *Theoretical Computer Science* **25**, 267–310 (1983).

[Mil89] R. MILNER. "Communication and Concurrency". Prentice Hall (1989).

[MRS95] M. W. MISLOWE, A. W. ROSCOE Y S. A. SCHNEIDER. Fixed points without completeness. *Theoretical Computer Science* **138**, 273–314 (1995).

[MT90] F. MOLLER Y C. TOFTS. A temporal calculus of communicating systems. En "CONCUR'90", páginas 401–415. Springer-Verlag (1990). LNCS 458.

[NdFL95] M. NÚÑEZ, D. DE FRUTOS Y L. F. LLANA-DÍAZ. Acceptance trees for probabilistic processes. En "CONCUR'95", páginas 249–263. Springer-Verlag (1995). LNCS 962.

[NS91] X. NICOLLIN Y J. SIFAKIS. An overview and synthesis on timed process algebras. En "Computer Aided Design", páginas 376–398 (1991). LNCS 575.

[NS94] X. NICOLLIN Y J. SIFAKIS. The algebra of timed processes, ATP: Theory and application. *Information and Computation* **114**, 131–178 (1994).

[OM91] Y. ORTEGA-MALLÉN. "En Busca del Tiempo Perdido". Tesis Doctoral, Departamento de Informática y Automática. Universidad Complutense de Madrid (1991).

[OMdF90] Y. ORTEGA-MALLÉN Y D. DE FRUTOS. Timed observations: a semantic model for real-time concurrency. En M. BROY Y C. B. JONES, editores, "TC2-Working Conference on Programming Concepts and Methods". Noth-Holland (1990).

[Phi87] I. PHILLIPS. Refusal testing. *Theoretical Computer Science* **50**, 241–284 (1987).

[Plo81] G. D. PLOTKIN. A structural approach to operational semantics. Informe técnico DAIMI FN-19, Computer Science Department, Aarhus University (1981).

[QdFA93] J. QUEMADA, D. DE FRUTOS Y A. AZCORRA. Tic: A timed calculus. *Formal Aspects of Computing* **5**, 224–252 (1993).

[QMdL94] J. QUEMADA, C. MIGUEL, D. DE FRUTOS Y L. F. LLANA-DÍAZ. A timed LOTOS extension. En T. RUS, editor, "Theories and Experiencies fo Real-Time System Develoment", tomo 2 de "AMAST Series in Computing", páginas 239–263. World Scientific (1994).

[RB89] A. W. ROSCOE Y G. BARRET. Unbounded non-determinism in CSP. En "Mathematical Foundations of Programming Semantics". Springer-Verlag (1989). LNCS 442.

[Ree88] G. M. REED. "A Uniform Mathematical Theory for Real-Time Distributed Computig". Tesis Doctoral, Oxford University (1988).

[Ros93] A. W. ROSCOE. Unbounded non-determinism in CSP. *Journal of Logic and Computation* **3**(2), 131–172 (Abril 1993).

[RR86] G. M. REED Y A. W. ROSCOE. A timed model for communicating sequential processes. En "ICALP '86", páginas 314–323. Springer-Verlag (1986). LNCS 226.

[RR87] G. M. REED Y A. W. ROSCOE. Metric spaces as models for real-time concurrency. En "Mathematical Foundations of Programming Language Semantics". Springer-Verlag (1987). LNCS 298.

[RR88] G. M. REED Y A. W. ROSCOE. A timed model for communicating sequential processes. *Theoretical Computer Science* **58**, 249–261 (1988).

[SBD95] S. SCHNEIDER, J. BRYANS Y J. DAVIES. Real-time LOTOS and timed observations. En "FORTE '95" (Octubre 1995).

[Sch88] D. A. SCHMIDT. "Denotational Semantics, a Methodology for Language Develoment". Wm. C. Brown Publishers (1988).

[Sch89] S. SCHNEIDER. "Correctness and Communication in Real-Time Systems". Tesis Doctoral, University of Oxford (1989).

[Sch92] S. SCHNEIDER. Unbounded nondeterminism for real-time processes. Informe técnico TR-13-92, University of Oxford (Julio 1992).

[Sch95] S. SCHNEIDER. An operational semantics for timed CSP. *Information and Computation* **116**(2), 193–213 (1995).

[SG90] D.S. Scott y G.A. Gunter. Semantic domains. En J. van Leewen, editor, "Handbook of Theoretical Computer Science", tomo B, capítulo 12, páginas 634–674. The MIT Press/Elsevier (1990).

[SS90] S.A. Smolka y B. Steffen. Priority as extremal probability. En "CONCUR'90", páginas 456–466. Springer-Verlag (1990). LNCS 458.

[TV89] D. Taubner y W. Vogler. Step failures semantics and a complete proof system. *Acta Informatica* **27**, 125–126 (1989).

[Win93] G. Winskel. "The Formal Semantics of Programming Languages". Foundations of Computing. The MIT Press (1993).

[Yi90] W. Yi. Real-time behavior of asynchrnous agents. En "CONCUR '90". Springer-Verlag (1990). LNCS 458.

[Yi91a] W. Yi. "A Calculus of Real Time Systems". Tesis Doctoral, Department of Computer Science. Chalmers University of Technology (1991).

[Yi91b] W. Yi. CCS+time = an interleaving model for real time systems. En "ICALP '91". Springer-Verlag (1991). LNCS 510.