





Un modelo semántico de procesos basado en la duración.

Andrea Christoph

<p>TES 97/9 DIF-01973 SALA</p>	<p>UNIVERSIDAD NACIONAL DE LA PLATA FACULTAD DE INFORMÁTICA Biblioteca 50 y 120 La Plata catalogo.info.unip.edu.ar biblioteca@info.unip.edu.ar</p>  <p>DIF-01973</p>
--	--



BIBLIOTECA
FAC. DE INFORMÁTICA
U.N.L.P.

Modelo Semántico de procesos basado en la duración.

Andrea Viviana Christoph

Trabajo de grado presentado al Departamento de Informática de la Universidad Nacional de La Plata como parte de los requisitos para obtener el título de Licenciado en Informática.

Director: Prof. Juan Echagüe.

Departamento de Informática,
Facultad de Ciencias Exactas, UNLP.

La Plata, Febrero de 1997.

DONACION.....

TES

\$.....

97/90j.1

Fecha.....

30-8-05

Inv. E..... Inv. B.....

1973



Resumen

En el campo de la semántica de los lenguajes para descripción de procesos, hay un interés en la equivalencia funcional de los mismos. En muchos casos nos interesa, además de la equivalencia funcional, la información relativa a la eficiencia o a la rapidez de los procesos. Este será el enfoque del presente trabajo, donde estudiaremos relaciones que unen dos procesos equivalentes desde el punto de vista funcional, y entre los cuales uno de los procesos es más rápido, o igualmente de rápido que el otro.

Consideraremos que las acciones que puede ejecutar un sistema tiene asociado una duración no nula. Si un proceso tiene varios subprocesos secuenciales en paralelo, cada uno de ellos tendrá un reloj que se modificará solamente cuando una acción local al subproceso sea ejecutada.

La teoría de procesos con duración propuesta por Gorrieri et al. en [4], apunta a describir cuando un proceso es más rápido que otro a partir de un preorden donde todos los comportamientos de un proceso son idénticos a los de un segundo, pero se ejecutan siempre antes en el tiempo. Sin embargo, no permite establecer una comparación entre los procesos x e y cuando y comienza rápidamente, mientras x , luego de un inicio más lento, acelera su funcionamiento superando a y . Dos procesos como estos, no aparecen relacionados de ninguna manera en la propuesta de Gorrieri. Aquí lo que se propone, es un preorden que permite establecer las mismas cosas que Gorrieri, y que además permite capturar este tipo de relaciones, al cual denominaremos *Preorden Al final*, y lo identificaremos con \sqsubseteq_f .

Dos procesos P y Q están relacionados por el preorden *Al final* ($P \sqsubseteq_f Q$) si son funcionalmente equivalentes y P es, a partir de un momento y hasta el final, al menos tan rápido como Q .

Demostraremos que \sqsubseteq_f es una precongruencia para todos los operadores, excepto para la recursión. Probaremos también que la teoría que proponemos incluye en forma propia a la teoría definida por Gorrieri.

El preorden induce, de la forma usual, a una relación de equivalencia a la que denominaremos *Equivalencia Al final*.

Presentaremos axiomatizaciones del preorden y de la equivalencia para procesos finitos, que son correctas y completas.

Compararemos la teoría propuesta con otras que surgen en la literatura. Demostraremos que es incomparable con el *Causal Preorder* propuesto en [1] y es estrictamente más grueso que el *Location Preorder* de [6].

Parte de los resultados que aquí se presentan se obtuvieron trabajando conjuntamente con Philippe Schnoebelen y Pedro D'Argenio. Algunos resultados obtenidos durante el desarrollo del proyecto fueron presentados en [10].

Las actividades involucradas en la preparación de este trabajo fueron parcialmente financiadas por el proyecto BID-CONICYT 140/94, Localidad y Duración en los Sistemas de Transiciones Asíncronos.



Contenidos.

1	Introducción	6
1.1	Lenguaje	7
1.2	Semántica Operacional	7
1.3	Ejemplo: <i>Máquinas Productoras</i>	8
2	Preorden "Al Final"	10
2.1	Definición del Preorden	10
2.2	Relación con el <i>Performance Preorder</i>	12
2.3	Relación con la Bisimulación Clásica	15
2.4	Propiedades	17
3	Axiomatización	31
3.1	Conjunto de Axiomas	31
3.2	Correctitud y Completitud de C	34
4	Relación con otros preórdenes	38
4.1	Causal Preorder	38
4.2	Location Preorder	41
4.3	Relación entre "más paralelo" y "más rápido"	45
	Bibliografía.	48

Lista de Figuras.

1.1	Reglas de Inferencia	8
1.2	Reglas de Congruencia	9
1.3	Proceso para la máquina Y	9
1.4	Proceso para la máquina X	9
2.1	Sistema de transición de $a.(a a a) \sqsubseteq_f, a.a a.a$	13
2.2	Ejemplo	18
2.3	Precongruencia para el <i>Retardo</i>	26
2.4	Precongruencia para el <i>Prefijo</i>	27
2.5	Precongruencia para el <i>Paralelo</i>	28
2.6	Precongruencia para la <i>Selección</i>	29
2.7	Precongruencia para la <i>Selección</i>	30
3.1	Reglas de Inferencia Auxiliares	32
3.2	Conjunto de Axiomas \mathcal{A}	33
3.3	Axiomas para \sqsubseteq_f	34
4.1	Reglas de Inferencia	39
4.2	Conjunto de Axiomas	39
4.3	a) Procesos no relacionados con \sqsubseteq_c ; b) Relación para \simeq_c	42
4.4	Reglas de Inferencia	43
4.5	Ejemplo de procesos no relacionados con \sqsubseteq_f	44



Capítulo 1

Introducción

Estamos interesados en comparar procesos que hacen lo mismo, con respecto a la velocidad de los mismos. No queremos exigir que un proceso x deba comportarse durante toda su ejecución de manera más rápida que un proceso y , para decir que x es más rápido que y . La condición que imponemos no es tan fuerte, pues diremos que x es más rápido que y , si x es a partir de un momento y hasta el final, más rápido que y .

Dadas dos máquinas productoras, X e Y . La máquina Y desde que se enciende mantiene una producción a velocidad constante. En cambio X es una máquina que comienza a trabajar a una velocidad muy lenta, inferior a la velocidad de Y , y que luego de un tiempo de calentamiento acelera su funcionamiento alcanzando su máxima velocidad, que supera la velocidad de producción de Y . En una situación como la mencionada podríamos preguntarnos:

”Cuál es la máquina que produce a mayor velocidad ? ”

En la literatura aparecen distintas teorías en las cuales se estudian modelos para procesos con tiempo y se establecen relaciones de preorden y equivalencia, pero con ninguna de ellas las máquinas antes mencionadas estarían relacionadas. En este trabajo proponemos una relación que nos permitirá afirmar que la máquina X es más rápida que la máquina Y .

En este capítulo vamos a definir el modelo sobre el cual trabajaremos; así como también algunas hipótesis que tendremos en cuenta a lo largo del desarrollo del trabajo.

Vamos a estudiar la equivalencia funcional de procesos y la noción de ”más rápido que”, con respecto a la velocidad de los mismos. Para ello necesitamos que el tiempo esté representado de alguna manera en el modelo. Los procesos tendrán asociado *relojes*, que no se parecen mucho a los relojes que conocemos, pues no se modifican por el paso de tiempo sino por la ejecución de acciones. Cada subproceso secuencial tendrá asociado un reloj que se modificará solamente cuando se ejecute una acción local al mismo.

Las acciones tienen asociado un tiempo de duración no nulo, que se mantendrá constante durante toda la ejecución. Consideramos que las acciones ocurren tan pronto como sea posible, esto es conocido en la literatura como *Eagerness*.

Asumimos *máximo paralelismo*, es decir siempre que un nuevo subproceso se active, hay un procesador libre para ejecutarlo.

1.1 Lenguaje

El lenguaje que utilizamos es muy simple y permite definir sistemas concurrentes, que no tienen sincronización de acciones, es decir no hay comunicación entre distintos subprocesos que se ejecutan en paralelo.

Denominamos **Act** al conjunto de acciones atómicas que se pueden realizar, $\text{Act} = \{ a, b, c, \dots \}$ y **Var** al conjunto de variables para la definición de procesos, $\text{Var} = \{ x, y, z, \dots \}$.

Los procesos se generan con la siguiente sintaxis:

$$E ::= Nil \mid a.E \mid E + F \mid E|F \mid x \mid \text{rec}x.E$$

donde $x \in \text{Var}$, $a \in \text{Act}$ y E, F son procesos.

El significado intuitivo de los operadores del lenguaje es el siguiente:

- *Nil* Es un proceso que no hace nada.
- $a.E$ Ejecuta la acción a y luego se comporta como E .
- $E + F$ (*Choice*) Elige comportarse como E o como F .
- $E|F$ (*Composición paralela*) Se ejecutan acciones de E y de F en paralelo. Este operador no permite comunicación.
- x Se comporta según la definición de la variable x .
- $\text{rec}x.E$ Es un proceso recursivo, según la definición de x .

1.2 Semántica Operacional

Describiremos el comportamiento de los procesos mediante la semántica operacional. Para ello utilizaremos sistemas de transiciones. Necesitamos que los sistemas reflejen, además de funcionalidad, tiempo. Para representar el tiempo, las transiciones del sistema tendrán etiquetas que describirán las acciones de los procesos y el tiempo asociado a la realización de las mismas.

Definición 1.2.1 Un Sistema de Transición con Etiquetas es una terna $\langle S, M, T \rangle$, donde S es un conjunto de estados, M es un conjunto de etiquetas y $T = \{ \xrightarrow{\mu} \subseteq S \times S \mid \mu \in M \}$ es la relación de transición.

Los estados están generados con una sintaxis similar a la de los procesos, la cual se encuentra enriquecida por *relojes*, que reflejan el retardo de un estado.

La sintaxis para generar los estados del sistema es:

$$S ::= Nil \mid a.E \mid s + t \mid s|t \mid recx.E \mid (n \Rightarrow s)$$

donde $n \in \mathbb{N}^+$, $a \in Act$ y E es un proceso.

Llamaremos Π al conjunto de etiquetas, $\Pi = Act \times \mathbb{N}^+$.

El significado de una etiqueta $\langle a, n \rangle \in \Pi$ en una transición es, la acción a terminó de ejecutarse exactamente n unidades de tiempo después de que la computación comenzó.

La relación de Transición está dada por el conjunto de reglas de inferencia (Fig. 1.1).

La congruencia estructural se puede observar en (Fig. 1.2)

$Act \quad \frac{}{a.E \xrightarrow{\langle a, f(a) \rangle} (f(a) \Rightarrow E)}$	
$Alt1 \quad \frac{s_1 \xrightarrow{\langle a, n \rangle} s'_1}{s_1 + s_2 \xrightarrow{\langle a, n \rangle} s'_1}$	$Par1 \quad \frac{s_1 \xrightarrow{\langle a, n \rangle} s'_1}{s_1 s_2 \xrightarrow{\langle a, n \rangle} s'_1 s_2}$
$Alt2 \quad \frac{s_2 \xrightarrow{\langle a, m \rangle} s'_2}{s_1 + s_2 \xrightarrow{\langle a, m \rangle} s'_2}$	$Par2 \quad \frac{s_2 \xrightarrow{\langle a, m \rangle} s'_2}{s_1 s_2 \xrightarrow{\langle a, m \rangle} s_1 s'_2}$
$Rec \quad \frac{E[rec\ x.E/x] \xrightarrow{\langle a, n \rangle} s}{rec\ x.E \xrightarrow{\langle a, n \rangle} s}$	$Clock \quad \frac{s_1 \xrightarrow{\langle a, n \rangle} s'_1}{(k \Rightarrow s_1) \xrightarrow{\langle a, k+n \rangle} (k \Rightarrow s'_1)}$

Figura 1.1: Reglas de Inferencia

1.3 Ejemplo: Máquinas Productoras

Recordemos el ejemplo mencionado al comienzo del capítulo: Dadas dos máquinas productoras, X y Y . La máquina Y desde que se enciende mantiene una producción a velocidad constante. En cambio X es una máquina que comienza a trabajar a

$$\begin{aligned}
 (n \Rightarrow s_1 | s_2) &= (n \Rightarrow s_1) | (n \Rightarrow s_2) \\
 (n \Rightarrow s_1 + s_2) &= (n \Rightarrow s_1) + (n \Rightarrow s_2) \\
 E &= (0 \Rightarrow E) \\
 (n \Rightarrow (m \Rightarrow s)) &= ((n + m) \Rightarrow s)
 \end{aligned}$$

Figura 1.2: Reglas de Congruencia

$$\begin{aligned}
 Y &= Z_{reg} | Z_{reg} \\
 Z_{reg} &= prod. Z_{reg}
 \end{aligned}$$

Figura 1.3: Proceso para la máquina Y

una velocidad muy lenta, inferior a la velocidad de Y , y que luego de un tiempo de calentamiento acelera su funcionamiento alcanzando su máxima velocidad, que supera la velocidad de producción de Y .

Veamos como podemos escribir con el lenguaje presentado a los procesos que se corresponderían con las máquinas X e Y .

Para la máquina Y que mantine su velocidad de producción constante tenemos el proceso de la Fig. 1.3.

Mientras que la descripción para el funcionamiento para la máquina X , la cual comienza a producir en forma lenta y luego después de un determinado tiempo se comporta de manera más rápida, se corresponde con el proceso de la Fig. 1.4.

$$\begin{aligned}
 X_0 &= prod. X_1 \\
 X_n &= prod. X_{n+1} \quad (n < N) \\
 X_N &= X'_0 | Z_{reg} \\
 X'_0 &= prod. X'_1 \\
 X'_n &= prod. X'_{n+1} \quad (n < N) \\
 X'_N &= Z_{reg} | Z_{reg}
 \end{aligned}$$

Figura 1.4: Proceso para la máquina X

Capítulo 2

Preorden "Al Final"

2.1 Definición del Preorden

En esta sección presentamos la definición formal del preorden *Al final*, en la cual se observan los *caminos maximales* de los procesos.

Definición 2.1.1 Un camino maximal es un camino finito en el cual existe un estado (s_k), a partir del cual no se puede realizar ninguna acción, al cual llamaremos *estado bloqueado*; o bien es un camino infinito.

Para relacionar procesos con respecto a la velocidad proponemos un preorden \sqsubseteq_f sobre P: $E \sqsubseteq_f F$ si sólo si E y F son funcionalmente equivalentes, pero E se comporta a partir de un determinado momento y hasta el final, en forma más rápida que F .

Definición 2.1.2 (Preorden Al final)

1) Sea \mathfrak{R} una relación binaria sobre el conjunto de estados de S, decimos que \mathfrak{R} es una *F-bisimulación* si para $(s_0, t_0) \in \mathfrak{R}$, se cumplen las siguientes propiedades de *transferencia*:

i) Para todo camino *maximal*

$$(s_0 \xrightarrow{\langle a_1, n_1 \rangle} s_1 \xrightarrow{\langle a_2, n_2 \rangle} s_2 \dots \xrightarrow{\langle a_j, n_j \rangle} s_j \xrightarrow{\langle a_{j+1}, n_{j+1} \rangle} s_{j+1} \dots)$$

existe un camino

$$(t_0 \xrightarrow{\langle a_1, m_1 \rangle} t_1 \xrightarrow{\langle a_2, m_2 \rangle} t_2 \dots \xrightarrow{\langle a_j, m_j \rangle} t_j \xrightarrow{\langle a_{j+1}, m_{j+1} \rangle} t_{j+1} \dots)$$

tal que: $\forall i (s_i \mathfrak{R} t_i)$ y $(\exists j) (\forall k \geq j) n_k \leq m_k$

ii) Para todo camino *maximal*

$$(t_0 \xrightarrow{\langle a_1, m_1 \rangle} t_1 \xrightarrow{\langle a_2, m_2 \rangle} t_2 \dots \xrightarrow{\langle a_j, m_j \rangle} t_j \xrightarrow{\langle a_{j+1}, m_{j+1} \rangle} t_{j+1} \dots)$$

existe un camino

$$(s_0 \xrightarrow{\langle a_1, n_1 \rangle} s_1 \xrightarrow{\langle a_2, n_2 \rangle} s_2 \dots \xrightarrow{\langle a_j, n_j \rangle} s_j \xrightarrow{\langle a_{j+1}, n_{j+1} \rangle} s_{j+1} \dots)$$

tal que: $\forall i (s_i \mathcal{R} t_i)$ y $(\exists j) (\forall k \geq j) m_k \geq n_k$.

- 2) Dos estados s_0 y t_0 están relacionados por el preorden *Al final* ($s_0 \sqsubseteq_f t_0$) si y sólo si existe una *F-bisimulación* \mathcal{R} tal que $(s_0, t_0) \in \mathcal{R}$.
- 3) Decimos que $E \sqsubseteq_f F$ si y sólo si $(0 \Rightarrow E) \sqsubseteq_f (0 \Rightarrow F)$.

Algunos ejemplos de procesos relacionados con el preorden son:

- $a.(a|a|a) \sqsubseteq_f a.a|a.a$ (1)
- $a|a|a + a.a.a \sqsubseteq_f a|a|a + a.a|a + a.a.a$
- $a|b \sqsubseteq_f a.b + b.a$
- $a.b + b.a \not\sqsubseteq_f a|b$

Para los procesos de (1) tenemos la relación dada en la (Fig. 2.1).

Proposición 2.1.3 *Propiedades de la F-bisimulación:*

- 1) *La relación identidad es una F-bisimulación.*
- 2) *Si \mathcal{R}_1 y \mathcal{R}_2 son F-bisimulaciones entonces:*
 - a) $\mathcal{R}_1 \circ \mathcal{R}_2$ *es una F-bisimulación.*
 - b) $\mathcal{R}_1 \cup \mathcal{R}_2$ *es una F-bisimulación.*

Dem.

- 1) Es trivial.
- 2) Sean \mathcal{R}_1 y \mathcal{R}_2 *F-bisimulaciones* tales que $(s_0, v_0) \in \mathcal{R}_1$ y $(v_0, t_0) \in \mathcal{R}_2$. Esto es:
 - i) $(\forall i) (s_i \mathcal{R}_1 v_i)$ y $(\exists j_1) (\forall k \geq j_1) (n_k \leq l_k)$ y
 - ii) $(\forall i) (v_i \mathcal{R}_2 t_i)$ y $(\exists j_2) (\forall k \geq j_2) (l_k \leq m_k)$.

Prueba de a) Por la definición de composición de relaciones y las hipótesis vale:

$$(\forall i) (s_i \mathcal{R}_1 \circ \mathcal{R}_2 t_i)$$

y por la transitividad de \leq en N se cumple que:

$$(\forall k \geq \max(j_1, j_2)) (n_k \leq m_k)$$

en consecuencia $\mathfrak{R}_1 \circ \mathfrak{R}_2$ es una F -bisimulación que contiene $(0 \Rightarrow E_1, 0 \Rightarrow E_3)$.

Prueba de b) $\mathfrak{R}_1 \cup \mathfrak{R}_2$ es una F -bisimulación, pues dado un par de estados $(s_0, t_0) \in \mathfrak{R}_1 \cup \mathfrak{R}_2$ ocurre que $(s_0, t_0) \in \mathfrak{R}_1$ o bien $(s_0, t_0) \in \mathfrak{R}_2$, y tanto \mathfrak{R}_1 como \mathfrak{R}_2 son F -bisimulaciones, con lo cual la prueba no resulta difícil. ■

Proposición 2.1.4 \sqsubseteq_f es un preorden.

Dem.

- Reflexividad: vale por Prop. 2.1.3 (1)
- Transitividad: sean E_1, E_2 y E_3 procesos tales que : $E_1 \sqsubseteq_f E_2$ y $E_2 \sqsubseteq_f E_3$. Entonces existen \mathfrak{R}_1 y \mathfrak{R}_2 F -bisimulaciones tales que $(s_0, v_0) \in \mathfrak{R}_1$ y $(v_0, t_0) \in \mathfrak{R}_2$. Definiendo \mathfrak{R} como la composición de \mathfrak{R}_1 y \mathfrak{R}_2 , por el resultado de Prop. 2.1.3(2), \mathfrak{R} es una F -bisimulación que contiene $((0 \Rightarrow E_1), (0 \Rightarrow E_3))$, y en consecuencia $E_1 \sqsubseteq_f E_3$ ■

En forma natural, la intersección del preorden, da una noción de equivalencia, a la cual llamaremos *Equivalencia al final*.

Definición 2.1.5 Dos procesos E_1 y E_2 , son *equivalentes al final*, lo anotaremos $(E_1 \simeq_f E_2)$, si y sólo si $E_1 \sqsubseteq_f E_2$ y $E_2 \sqsubseteq_f E_1$.

2.2 Relación con el *Performance Preorder*

El *preorden al final*, impone condiciones más débiles que el *Performance Preorder* propuesto en [4], pues este último cuando decimos que el proceso x es más rápido que el proceso y pide que en todo momento x se comporte, al menos tan rápidamente como y ; en cambio al preorden al final solamente exige que a partir de un determinado momento y hasta el final, el proceso x sea al menos tan rápido como y , sin tener en cuenta como se comportaban antes de ese momento. Recordaremos primero la definición del preorden de [4] y luego mostraremos que el preorden *Al final* incluye en forma propia al *Performance Preorder*.

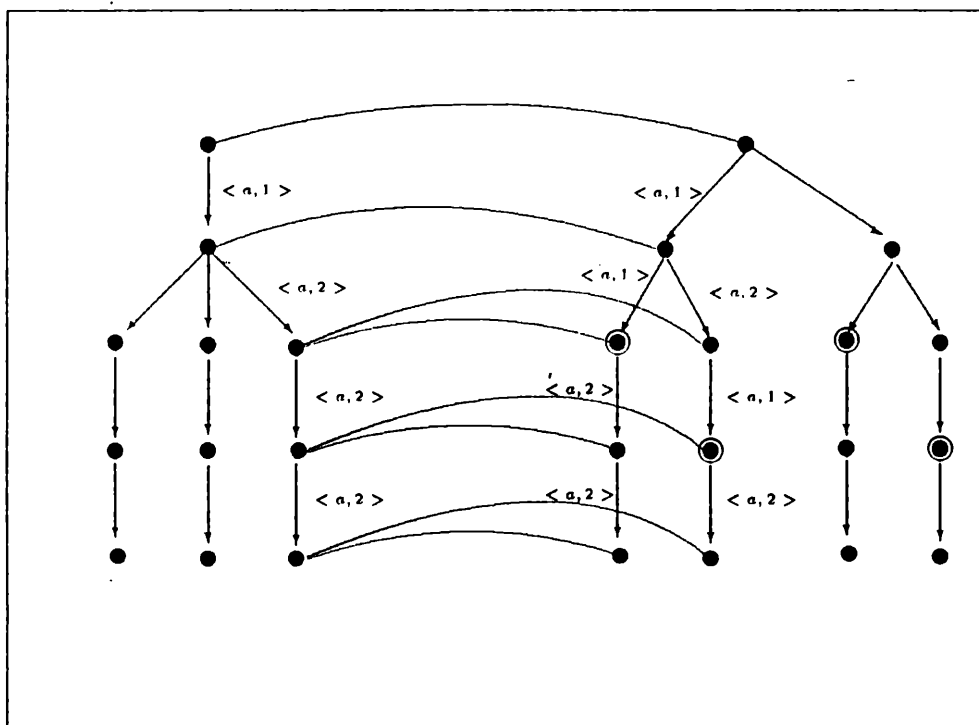


Figura 2.1: Sistema de transición de $a.(a|a|a) \sqsubseteq_f a.a|a.a$

Definición 2.2.1 (Performance Preorder)

- 1) Sea Rel el conjunto de relaciones binarias en S . El funcional $G : Rel \rightarrow Rel$, se define para cada $\mathfrak{R} \in Rel$, de la siguiente forma: $(s_1; s_2) \in G(\mathfrak{R})$, si para $a \in Act$,
 - a) $\forall s_1 \xrightarrow{\langle a_1, n_1 \rangle} s_2, \exists t_1 \xrightarrow{\langle a_1, m_1 \rangle} t_2; n_1 \leq m_1$ y $(s_2, t_2) \in \mathfrak{R}$.
 - b) $\forall t_1 \xrightarrow{\langle a_1, m_1 \rangle} t_2, \exists s_1 \xrightarrow{\langle a_1, n_1 \rangle} s_2; m_1 \geq n_1$ y $(s_2, t_2) \in \mathfrak{R}$.
- 2) Una relación $\mathfrak{R} \in Rel$ es una G -bisimulación si $\mathfrak{R} \subseteq G(\mathfrak{R})$.
- 3) Dos estados s_1 y s_2 están relacionados con el *Performance preorder* ($s_1 \sqsubseteq_p s_2$), si y sólo si existe un G -bisimulación \mathfrak{R} tal que $(s_1, s_2) \in \mathfrak{R}$.
- 4) Decimos que $E_1 \sqsubseteq_p E_2$ si y sólo si $(0 \Rightarrow E_1) \sqsubseteq_p (0 \Rightarrow E_2)$.

Proposición 2.2.2 $\sqsubseteq_p \subset \sqsubseteq_f$.

Dem.

- Veamos que: $\forall E, F: E \sqsubseteq_p F \Rightarrow E \sqsubseteq_f F$. $E \sqsubseteq_p F$ implica que existe una G -bisimulación \mathfrak{R} tal que $(0 \Rightarrow E, 0 \Rightarrow F) \in \mathfrak{R}$.
No es difícil probar si \mathfrak{R} es una G -bisimulación entonces la misma \mathfrak{R} es una F -bisimulación. Pues la relación satisficará las propiedades de transferencia con $j \geq 1$.
- Existen procesos E, F tales que: $E \sqsubseteq_f F \not\Rightarrow E \sqsubseteq_p F$. Sean los procesos E y F los siguientes:

$$E = a.(a|a|a) \text{ y } F = a.a|a.a.$$

Probamos en la sección anterior que E es más rápido al final que F .

Sin embargo no existe una relación que sea una G -bisimulación, pues para la siguiente transición del proceso F :

$$(1 \Rightarrow a)|(0 \Rightarrow a.a) \xrightarrow{\langle a, 1 \rangle} (1 \Rightarrow a) | (1 \Rightarrow a)$$

la única transición que podría imitarla en el proceso E es:

$$(1 \Rightarrow a) | (1 \Rightarrow a) | (1 \Rightarrow a) \xrightarrow{\langle a, 2 \rangle} (2 \Rightarrow a) | (1 \Rightarrow a) | (1 \Rightarrow a)$$

en la cual la acción a demora más tiempo. En consecuencia E no es más rápido que F según el *Performance preorder*. ■

2.3 Relación con la Bisimulación Clásica

La bisimulación es una relación de equivalencia que solamente se interesa en la equivalencia funcional de los procesos, sin tener en cuenta los tiempos.

Definición 2.3.1 (Bisimulación)

Sea \mathfrak{R} una relación binaria sobre el conjunto de estados S , diremos que \mathfrak{R} es una *Bisimulación*, con $(s_1, s_2) \in \mathfrak{R}$ si para $a \in \text{Act}$,

- a) $\forall s_1 \xrightarrow{a} s_2, \exists t_1 \xrightarrow{a} t_2; (s_2, t_2) \in \mathfrak{R}$.
 b) $\forall t_1 \xrightarrow{a} t_2, \exists s_1 \xrightarrow{a} s_2; (s_2, t_2) \in \mathfrak{R}$.

Buscamos una relación entre el preorden *Al final* y la *Bisimulación*, esto se debe a que estamos interesados en relacionar procesos que sean funcionalmente equivalentes. Para poder demostrar los resultados que hallamos, necesitamos introducir la definición de dos nuevas funciones sobre los estados: *cut* y *clocks*. La función *cut* dado un estado de S retorna el agente E que se obtiene quitando todos los tiempos prefijos del estado. Esta función es necesaria para poder abstraernos de los tiempos que mantiene cada proceso y poner interés en las funciones que el mismo realiza.

Definición 2.3.2 Definimos por inducción estructural sobre S ,

$$\begin{aligned} \text{cut}(\text{Nil}) &= \text{Nil} \\ \text{cut}(a.E) &= a.E \\ \text{cut}(s + t) &= \text{cut}(s) + \text{cut}(t) \\ \text{cut}(s|t) &= \text{cut}(s)|\text{cut}(t) \\ \text{cut}(\text{rec } x.F) &= \text{rec } x.\text{cut}(F) \\ \text{cut}(n \Rightarrow s) &= \text{cut}(s) \end{aligned}$$

La función *clocks* retorna el conjunto $\{n_1, \dots, n_k\}$ de todos los valores correspondientes a los relojes que aparecen en el estado.

Definición 2.3.3 Definimos por inducción estructural sobre S ,

$$\begin{aligned} \text{clocks}(\text{Nil}) &= \{0\} \\ \text{clocks}(a.E) &= \{0\} \\ \text{clocks}(s + t) &= \text{clocks}(s) \cup \text{clocks}(t) \\ \text{clocks}(s|t) &= \text{clocks}(s) \cup \text{clocks}(t) \\ \text{clocks}(n \Rightarrow s) &= \{n\} \cup \text{clocks}(s) \end{aligned}$$

El siguiente lema, mencionado en [4], nos será muy útil, pues permite establecer una relación entre las transiciones con tiempo de nuestro modelo y las transiciones unicamente de acciones del modelo utilizado en [1].

Lema 2.3.4 Para todo $s, t \in S$: $\exists n \in N$; $s \xrightarrow{\langle a, n \rangle} t$ sii $cut(s) \xrightarrow{a} cut(t)$.

El preorden *Al final* tiene algunas propiedades interesantes con respecto a la *Bisimulación Clásica*, denotada por \sim , que se resumen en el siguiente teorema. Denominaremos P_{seq} al conjunto de procesos que se generan sin utilizar el operador para la composición paralela.

Teorema 2.3.5 Sean $E_1, E_2 \in P_{seq}$, $E_3, E_4 \in P$ y g una función de duración para las acciones. Entonces:

- 1) $E_3 \sqsubseteq_f^g E_4$ implica $E_3 \sim E_4$.
- 2) $E_1 \sim E_2$ sii $E_1 \sqsubseteq_f^g E_2$.
- 3) $E_3 \sim E_4$ sii $E_3 \sqsubseteq_f^g E_4$ para alguna función de duración g tal que: $g(a) = 0 \forall a \in Act$.
- 4) $E_1 \sim E_3$ sii $E_3 \sqsubseteq_f^g E_1$.

Dem.

- 1) $E_3 \sqsubseteq_f^g E_4 \Rightarrow E_3 \sim E_4$, porque la relación:

$$\mathfrak{R} = \{(cut(s'_3), cut(s'_4)) | s'_3 \sqsubseteq_f^g s'_4\}$$

es una bisimulación que contiene (E_3, E_4) .

- 2) $(\Rightarrow) E_1 \sim E_2 \Rightarrow E_1 \sqsubseteq_f^g E_2$, porque la relación:

$$\mathfrak{R} = \{(n \Rightarrow E'_1, n \Rightarrow E'_2) | n \in N, E'_1 \sim E'_2 \text{ y } E'_1, E'_2 \in P_{seq}\}$$

es una G - bisimulación que contiene $((0 \Rightarrow E_1), (0 \Rightarrow E_2))$ y $\sqsubseteq_p \subset \sqsubseteq_f$.

(\Leftarrow) En el inciso 1) probamos que para todo par de procesos E, F de P : $E \sqsubseteq_f F$ implica que $E \sim F$, entonces vale en particular para procesos secuenciales.

- 3) Trivial, pues los relojes nunca se incrementan.

- 4) $(\Rightarrow) E_1 \sim E_3 \Rightarrow E_3 \sqsubseteq_f^g E_1$, porque la relación:

$$\mathfrak{R} = \{(s, n \Rightarrow E'_1) | cut(s) = E'_3, E'_1 \sim E'_3 \text{ y } n \geq \max(clocks(s))\}$$

es una G - bisimulación que contiene $(0 \Rightarrow E_3, 0 \Rightarrow E_1)$ y $\sqsubseteq_p \subset \sqsubseteq_f$.

(\Leftarrow) Nuevamente como en el inciso 2), es un caso particular de la prueba mencionada en 1).

■

2.4 Propiedades

La elección de la función g para la duración de las acciones influye notablemente en la relación \sqsubseteq_f , pues la relación de dos procesos con el preorden puede variar dependiendo de la función de duración que elegimos. El siguiente ejemplo muestra la dependencia de \sqsubseteq_f con respecto a la función de duración elegida.

Ejemplo 2.4.1 Consideremos el siguiente par de procesos:

$$E_1 = a.a|c \quad \text{y} \quad E_2 = a.a|c + c.(a|a)$$

Podemos demostrar que $E_1 \sqsubseteq_f^g E_2$ si y sólo si $g(a) \leq g(c)$ (Fig. 2.2).

Tomando la ejecución del proceso E_2 , cuando este elige comportarse como el proceso que aparece en el lado derecho:

$$(0 \Rightarrow c.(a|a)) \xrightarrow{\langle c, g(c) \rangle} (g(c) \Rightarrow a|a) \xrightarrow{\langle a, g(c)+g(a) \rangle} (g(c) + g(a) \Rightarrow Nil) \mid (g(c) \Rightarrow a) \xrightarrow{\langle a, g(c)+g(a) \rangle} (g(c) + g(a) \Rightarrow Nil) \mid (g(c) + g(a) \Rightarrow Nil)$$

vemos que la única forma que tiene de imitarlo E_1 es con :

$$(0 \Rightarrow a.a) \mid (0 \Rightarrow c) \xrightarrow{\langle c, g(c) \rangle} (0 \Rightarrow a.a) \mid (g(c) \Rightarrow Nil) \xrightarrow{\langle a, g(a) \rangle} (g(a) \Rightarrow a) \mid (g(c) \Rightarrow Nil) \xrightarrow{\langle a, g(a)+g(a) \rangle} (g(a) + g(a) \Rightarrow Nil) \mid (g(c) \Rightarrow Nil)$$

si no fuera $g(a) \leq g(c)$, vemos que no existe un momento a partir del cual E_2 sea tan lento como E_1 , pues la segunda acción a de E_1 termina luego de $2 \times g(a)$ unidades de tiempo, mientras que la segunda acción a de E_2 luego de $g(c) + g(a)$.

Podemos definir una noción más fina del preorden, en la cual la relación no depende de la función de duración que se elija para las distintas acciones. A esta relación la denominaremos *más rápido al final independiente* y la definimos de la siguiente manera:

Definición 2.4.2 E_1 está relacionado en forma independiente de la función de duración por el preorden *Al final* con E_2 , denotado por $E_1 \sqsubseteq_f^i E_2$, si $E_1 \sqsubseteq_f^g E_2$ para cualquier función de duración g .

Un ejemplo de relación independiente de la función de duración es la que existe entre los procesos $a|b \sqsubseteq_f a.b + b.a$.

La equivalencia *Al final* está definida mediante la intersección del preorden, con lo cual también depende de la función de duración elegida para las acciones.

Ejemplo 2.4.3 Sean los siguientes procesos:

$$E_1 = a.a|c + c.a.a \quad \text{y} \quad E_2 = a.a|c + c.a.a + c.(a|a)$$

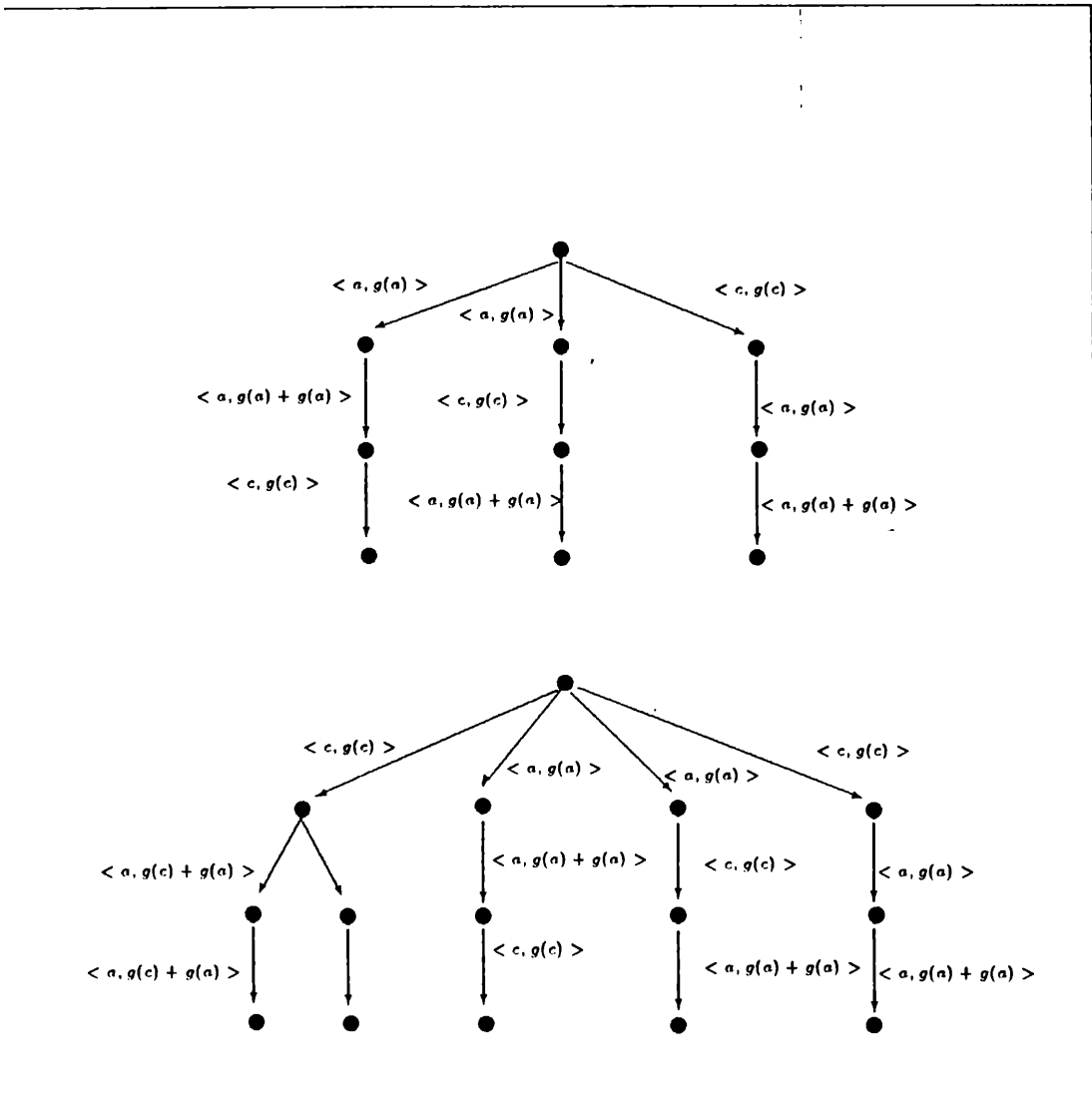


Figura 2.2: Ejemplo

Decimos que $E_1 \simeq_f E_2$ si y sólo si $g(a) \leq g(c)$. Los argumentos aquí son similares a los del Ej. 2.4.1.

Definición 2.4.4 Diremos que E y F son *equivalentes al final en forma independiente* y lo anotaremos $E \simeq_f^i F$, si $E \sqsubseteq_f^g F$ y $F \sqsubseteq_f^g E$ para toda función de duración g .

Hay algunas propiedades interesantes que esperamos del preorden *Al final* respecto de los posibles retardos de los procesos, así como relaciones entre los retardos a partir de la relación del preorden.

Proposición 2.4.5 Para todo proceso E de P y $n \in \mathbb{N}^+$,

$$E \sqsubseteq_f n \Rightarrow E.$$

Dem. Con $\mathfrak{R} = \{ (s, n \Rightarrow s) \}$, $n \geq 0$ por la regla *clock* vale trivialmente, pues al mismo proceso lo retrasamos n unidades de tiempo. ■

Como consecuencia de la proposición anterior surge que no podemos incorporar un retardo a un proceso que termina, y que este sea más rápido que el mismo sin el retardo.

Proposición 2.4.6 Sea E un proceso que debe terminar, lo anotaremos $E \downarrow$,

$$(\neg \exists n > 0) : (n \Rightarrow E) \sqsubseteq_f E.$$

A partir de dos procesos E, F tales que $E \sqsubset_f F$ (E es estrictamente más rápido que F : $E \sqsubseteq_f F$ y $F \not\sqsubseteq_f E$), podemos deducir incorporando retardos a ambos procesos una relación entre los mismos. Si a E lo retrasamos n unidades de tiempo y a F m unidades, y ocurre que el proceso $(m \Rightarrow F)$ es más rápido al final que $(n \Rightarrow E)$, podemos asegurar que el retardo incorporado a F es menor estricto que el que incorporamos a E . Esto se debe al cambio que se observa en la relación de los procesos con el preorden.

Proposición 2.4.7 Sean E, F procesos de P :

$$\text{Si } E \sqsubseteq_f F \text{ y } (m \Rightarrow F) \sqsubseteq_f (n \Rightarrow E) \text{ entonces } m < n.$$

Al componer paralelamente un proceso F que siempre termina con uno E que nunca lo hace, podemos observar que los caminos maximales del proceso correspondiente a la composición son infinitos. Además, a partir del momento en el que F termine, los estados de los caminos serán similares a los de los caminos de E . Si tomamos dos procesos bisimilares, que siempre terminan y componemos cada uno de ellos con un proceso que nunca lo hace, encontraremos que ambos procesos son equivalentes *al final*. Pues, a partir del momento que los procesos que deben terminar lo hagan, los caminos de las composiciones coinciden, pues sólo queda el proceso que no puede terminar. En la siguiente proposición se formaliza esta idea.

Proposición 2.4.8 Sea E un proceso que no puede terminar, lo anotaremos $E \uparrow$, y F, G procesos que siempre deben terminar; denotado por $F \downarrow$ y $G \downarrow$. Se cumple que:

$$F \sim G \Rightarrow E|F \simeq_f E|G$$

Dem. $F \sim G \Rightarrow \exists \mathcal{R}$ tal que: $(s_1, t_1) \in \mathcal{R}$, con \mathcal{R} y \mathcal{R}^{-1} bisimulaciones, y

- a) $\forall s_1 \xrightarrow{\langle a, n \rangle} s_2: \exists t_1 \xrightarrow{\langle a, m \rangle} t_2$, tal que $(s_2, t_2) \in \mathcal{R}$ y
- b) $\forall t_1 \xrightarrow{\langle a, m \rangle} t_2: \exists s_1 \xrightarrow{\langle a, n \rangle} s_2$, tal que $(s_2, t_2) \in \mathcal{R}$.

No es difícil probar que tanto $\mathcal{R}_n = \{(E|F, E|G) | F \mathcal{R} G\}$ como $\mathcal{R}_n^{-1} = \{(E|G, E|F) | F \mathcal{R} G\}$ son F -bisimulaciones. Pues en algún momento, tanto F como G terminarán, y sólo continúa el proceso E , sin tener importancia los tiempos de ejecución de los procesos que terminan, en consecuencia $E|F \simeq_f E|G$. ■

Un caso particular de la proposición anterior es:

$$(\forall n, m \in N^+) E | (m \Rightarrow F) \simeq_f E | (n \Rightarrow F)$$

Luego de ver que se cumplen algunas propiedades del preorden que esperabamos, llegamos a una propiedad que además de ser deseable es muy importante, nos estamos refiriendo a la *precongruencia*. El preorden al final es una precongruencia para todos los operadores del lenguaje que generan el conjunto de procesos finitos (P_{fin}). Sin embargo, cuando trabajamos con procesos infinitos, no lo es.

Teorema 2.4.9 \sqsubseteq_f es una precongruencia para P_{fin} .

Dem.

1) $E \sqsubseteq_f F \Rightarrow (k \Rightarrow E) \sqsubseteq_f (k \Rightarrow F)$.

$E \sqsubseteq_f F$ implica que existe una relación \mathfrak{R} que es una F -bisimulación que contiene $(0 \Rightarrow E, 0 \Rightarrow F)$ (denominaremos $s_0 = (0 \Rightarrow E)$ y $t_0 = (0 \Rightarrow F)$).

Definimos

$$\mathfrak{R}' = \{ (k \Rightarrow E, k \Rightarrow F) \mid k \in N \text{ y } s_0 \mathfrak{R} t_0 \}$$

Sea

$$\begin{aligned} & (k \Rightarrow s_0) \xrightarrow{\langle a_1, n_1+k \rangle} (k \Rightarrow s_1) \xrightarrow{\langle a_2, n_2+k \rangle} (k \Rightarrow s_2) \dots \\ & \dots \xrightarrow{\langle a_j, n_j+k \rangle} (k \Rightarrow s_j) \xrightarrow{\langle a_{j+1}, n_{j+1}+k \rangle} (k \Rightarrow s_{j+1}) \dots \end{aligned}$$

un camino maximal de E , aplicando la regla *clock* en cada transición obtenemos un camino de la forma

$$s_0 \xrightarrow{\langle a_1, n_1 \rangle} s_1 \xrightarrow{\langle a_2, n_2 \rangle} s_2 \dots \xrightarrow{\langle a_j, n_j \rangle} s_j \xrightarrow{\langle a_{j+1}, n_{j+1} \rangle} s_{j+1} \dots$$

Por la hipótesis sabemos $s_0 \mathfrak{R} t_0$ y que \mathfrak{R} es una F -bisimulación, entonces existe un camino

$$t_0 \xrightarrow{\langle a_1, m_1 \rangle} t_1 \xrightarrow{\langle a_2, m_2 \rangle} t_2 \dots \xrightarrow{\langle a_j, m_j \rangle} t_j \xrightarrow{\langle a_{j+1}, m_{j+1} \rangle} t_{j+1} \dots$$

que imita al camino desde s_0 , y que a partir de un momento las acciones en t demoran más tiempo que en s , es decir $(\forall i) s_i \mathfrak{R} t_i$ y $(\exists j)(\forall r \geq j) n_r \leq m_r$.

Aplicando la regla *clock* a cada transición de este camino, obtenemos:

$$\begin{aligned} & (k \Rightarrow t_0) \xrightarrow{\langle a_1, m_1+k \rangle} (k \Rightarrow t_1) \xrightarrow{\langle a_2, m_2+k \rangle} (k \Rightarrow t_2) \dots \\ & \dots \xrightarrow{\langle a_j, m_j+k \rangle} (k \Rightarrow t_j) \xrightarrow{\langle a_{j+1}, m_{j+1}+k \rangle} (k \Rightarrow t_{j+1}) \dots \end{aligned}$$

donde: $(\forall i) (k \Rightarrow s_i) \mathfrak{R}' (k \Rightarrow t_i)$ y $(\exists j)(\forall r \geq j) n_r + k \leq m_r + k$.

Para probar que todo camino desde F puede ser imitado por E de manera más rápida al final, se contruye el camino correspondiente a la imitación en forma similar a la anterior.

En consecuencia, \mathfrak{R}' es una F -bisimulación y $(k \Rightarrow E, k \Rightarrow F)$ pertenecen a \mathfrak{R}' (Fig. 2.3).

2) $E \sqsubseteq_f F \Rightarrow a.E \sqsubseteq_f a.F$

$E \sqsubseteq_f F$ implica que existe una relación \mathfrak{R} que es una F -bisimulación que contiene $(0 \Rightarrow E, 0 \Rightarrow F)$ (denominaremos $s_0 = (0 \Rightarrow E)$ y $t_0 = (0 \Rightarrow F)$).

Definimos

$$\mathfrak{R}'' = \{ (a.E, a.F) \mid (0 \Rightarrow E) \mathfrak{R} (0 \Rightarrow F) \} \cup \mathfrak{R}'$$

aplicando la regla correspondiente al prefijo (*Act*) a los estados $a.E$ y $a.F$, obtenemos $(g(a) \Rightarrow E)$ y $(g(a) \Rightarrow F)$ respectivamente.

Por la prueba del inciso anterior, y ser la relación \mathfrak{R}' una *F-bisimulación*, podemos decir que \mathfrak{R}'' es una *F-bisimulación* (Fig. 2.4).

$$3) \quad E \sqsubseteq_f F \Rightarrow (\forall P) E|P \sqsubseteq_f F|P \quad \text{y} \quad (\forall P) P|E \sqsubseteq_f P|F.$$

$E \sqsubseteq_f F$ implica que existe una relación \mathfrak{R} que es una *F-bisimulación* que contiene $(0 \Rightarrow E, 0 \Rightarrow F)$ (denominaremos $s_0 = (0 \Rightarrow E)$ y $t_0 = (0 \Rightarrow F)$).

Definimos

$$\mathfrak{R}_3 = \{ (s|v, t|v) \mid s \mathfrak{R} t \}$$

La prueba resulta bastante intuitiva y básicamente refleja la idea de intercambiar secuencias no consecutivas de las transiciones, que son el resultado de la aplicación de las reglas *Par1* (o *Par2*), con las transiciones que sabemos que existen por la relación \mathfrak{R} (Fig. 2.5).

i) Sea

$$s_0|v_0 \xrightarrow{\langle a_1, n_1 \rangle} s_1|v_1 \xrightarrow{\langle a_2, n_2 \rangle} s_2|v_2 \dots \xrightarrow{\langle a_j, n_j \rangle} s_j|v_j \xrightarrow{\langle a_{j+1}, n_{j+1} \rangle} s_{j+1}|v_{j+1} \dots$$

un camino del proceso $E|P$. En cada transición podemos saber si se ejecutó un acción del proceso E o P , pues dada una transición

$$s_i|v_i \xrightarrow{\langle a, n \rangle} s_{i+1}|v_{i+1}$$

* Si $s_i = s_{i+1}$ entonces la acción a corresponde al proceso P y se aplicó la regla *Par2*;

* Si $v_i = v_{i+1}$, la acción a es del proceso E y se aplicó la regla *Par1*.

Construiremos el camino correspondiente al proceso $F|P$ que imita las acciones del proceso $E|P$ pero demorando más tiempo. Para construirlo seguiremos el siguiente procedimiento:

- 1) Extraer del camino las transiciones que corresponden a la aplicación de la regla *Par1*.
- 2) Formar una secuencia con los antecedentes de la regla *Par1* correspondientes a las transiciones extraídas en (1).
- 3) Obtener la secuencia de transiciones *F-bisimilares* del proceso F , que sabemos que existe por la relación \mathfrak{R} .
- 4) Aplicar la regla *Par1* al camino obtenido en (3), formando un nuevo camino.

- 5) Rearmar el camino en base al camino original, reemplazando las transiciones que antes fueron extraídas por el nuevo camino obtenido en el paso anterior.

Así obtenemos un camino de la forma

$$t_0|v_0 \xrightarrow{\langle a_1, n_1 \rangle} t_1|v_1 \xrightarrow{\langle a_2, n_2 \rangle} t_2|v_2 \dots \xrightarrow{\langle a_j, n_j \rangle} t_j|v_j \xrightarrow{\langle a_{j+1}, n_{j+1} \rangle} t_{j+1}|v_{j+1} \dots$$

que es el camino que buscábamos.

- ii) Sea el siguiente un camino del proceso $F|P$

$$t_0|v_0 \xrightarrow{\langle a_1, n_1 \rangle} t_1|v_1 \xrightarrow{\langle a_2, n_2 \rangle} t_2|v_2 \dots \xrightarrow{\langle a_j, n_j \rangle} t_j|v_j \xrightarrow{\langle a_{j+1}, n_{j+1} \rangle} t_{j+1}|v_{j+1} \dots$$

Aplicando el mismo procedimiento que en el inciso anterior podemos obtener un camino del proceso $E|P$ que imita las acciones, pero demorando menos tiempo.

En consecuencia, \mathfrak{R}_3 es una F -bisimulación.

La prueba de $E \sqsubseteq_f F \Rightarrow (\forall P) P|E \sqsubseteq_f P|F$, es similar. En lugar de utilizar la regla *Par1*, aplicaremos la regla *Par2* y la relación será :

$$\mathfrak{R}'_3 = \{ (v|s, v|t) \mid s \mathfrak{R} t \}$$

- 4) $E \sqsubseteq_f F \Rightarrow (\forall P) E + P \sqsubseteq_f F + P$ y $(\forall P) P + E \sqsubseteq_f P + F$.

$E \sqsubseteq_f F$ implica que existe una relación \mathfrak{R} que es una F -bisimulación que contiene $(0 \Rightarrow E, 0 \Rightarrow F)$ (denominaremos $s_0 = (0 \Rightarrow E)$ y $t_0 = (0 \Rightarrow F)$).

Definimos

$$\mathfrak{R}_s = \{ (s + v, t + v) \mid s \mathfrak{R} t \} \cup Id \cup \mathfrak{R}$$

Dividiremos la prueba en dos casos que se corresponden con elegir comportarse como el proceso E (Fig. 2.6), o como el proceso P (Fig. 2.7).

- i) Sea

$$s_0 + v_0 \xrightarrow{\langle a_1, n_1 \rangle} s_1 \xrightarrow{\langle a_2, n_2 \rangle} s_2 \dots \xrightarrow{\langle a_j, n_j \rangle} s_j \xrightarrow{\langle a_{j+1}, n_{j+1} \rangle} s_{j+1} \dots$$

un camino del proceso $P + E$. Aplicando la regla *sum1* a la primer transición del camino obtenemos el camino:

$$s_0 \xrightarrow{\langle a_1, n_1 \rangle} s_1 \xrightarrow{\langle a_2, n_2 \rangle} s_2 \dots \xrightarrow{\langle a_j, n_j \rangle} s_j \xrightarrow{\langle a_{j+1}, n_{j+1} \rangle} s_{j+1} \dots,$$

luego por las propiedades de transferencia de la relación \mathfrak{R} , existe un camino

$$t_0 \xrightarrow{\langle a_1, m_1 \rangle} t_1 \xrightarrow{\langle a_2, m_2 \rangle} t_2 \dots \xrightarrow{\langle a_j, m_j \rangle} t_j \xrightarrow{\langle a_{j+1}, m_{j+1} \rangle} t_{j+1} \dots$$

más lento al final para todo camino desde s_0 . Aplicando la regla *sum1* a la primera transición del camino desde t_0 , en sentido contrario, obtenemos un camino como el siguiente:

$$t_0 + v_0 \xrightarrow{\langle a_1, m_1 \rangle} t_1 \xrightarrow{\langle a_2, m_2 \rangle} t_2 \dots \xrightarrow{\langle a_j, m_j \rangle} t_j \xrightarrow{\langle a_{j+1}, m_{j+1} \rangle} t_{j+1} \dots$$

Para probar que para todo camino desde t_0 existe uno desde s_0 , se sigue el mismo procedimiento pero ahora a partir del camino que comienza en el estado $t_0 + v_0$.

ii) Sea el siguiente camino

$$s_0 + v_0 \xrightarrow{\langle a_1, n_1 \rangle} v_1 \xrightarrow{\langle a_2, n_2 \rangle} v_2 \dots \xrightarrow{\langle a_j, n_j \rangle} v_j \xrightarrow{\langle a_{j+1}, n_{j+1} \rangle} v_{j+1} \dots$$

del proceso $P + E$. Aplicando la regla *sum2* a la primer transición del camino obtenemos otro camino:

$$v_0 \xrightarrow{\langle a_1, n_1 \rangle} v_1 \xrightarrow{\langle a_2, n_2 \rangle} v_2 \dots \xrightarrow{\langle a_j, n_j \rangle} v_j \xrightarrow{\langle a_{j+1}, n_{j+1} \rangle} v_{j+1} \dots,$$

por ser la relación Id una F -bisimulación, podemos aplicar a la primer transición del mismo la regla *sum2*, en orden inverso y obtener así un camino

$$t_0 + v_0 \xrightarrow{\langle a_1, n_1 \rangle} v_1 \xrightarrow{\langle a_2, n_2 \rangle} v_2 \dots \xrightarrow{\langle a_j, n_j \rangle} v_j \xrightarrow{\langle a_{j+1}, n_{j+1} \rangle} v_{j+1} \dots$$

En consecuencia \mathfrak{R}_s es una F -bisimulación.

La prueba de $E \sqsubseteq_f F \Rightarrow (\forall P) E + P \sqsubseteq_f F + P$, es simétrica, la relación es la siguiente

$$\mathfrak{R}_s = \{ (v + s, v + t) \mid s \mathfrak{R} t \} \cup Id \cup \mathfrak{R}$$

Para mostrar que la relación del preorden no se mantiene cuando usamos el operador para la recursión (*Rec*), estudiaremos el siguiente ejemplo. ■

Sean los siguientes procesos:

$$G \uparrow$$

$$E[x] = c.(x|1 \Rightarrow a)|G$$

$$F[x] = c.(x|0 \Rightarrow a)|G$$

Veamos que $(\forall x) E[x] \sqsubseteq_f F[x]$

- Si x es un proceso que termina.

Tenemos que

$$c.(x|1 \Rightarrow a) \sim c.(x|0 \Rightarrow a) \text{ y}$$

G es un proceso que no termina

con lo cual podemos aplicar Prop. 2.4.8 obteniendo

$$c.(x|1 \Rightarrow a)|G \simeq_f c.(x|0 \Rightarrow a)|G$$

- Si x es un proceso que no termina.
Por Prop. 2.4.8 sabemos que

$$(x|1 \Rightarrow a) \simeq_f (x|1 \Rightarrow a)$$

luego por ser \sqsubseteq_f una precongruencia para los operadores *Prefijo* y *Paralelo*,

$$c.(x|1 \Rightarrow a)|G \simeq_f c.(x|0 \Rightarrow a)|G$$

Sin embargo, $\text{rec}x.E[x] \not\sqsubseteq_f \text{rec}x.F[x]$. Para comprobar esto basta con observar las trazas:

$$\text{rec}x.F[x]: c:1 \quad a:2 \quad c:2 \quad a:3 \quad c:3 \quad a:4$$

$$\text{rec}x.E[x]: c:1 \quad a:3 \quad c:2 \quad a:4 \quad c:3 \quad a:5$$

Los retardos se mantienen a lo largo de los caminos infinitos, en consecuencia no va a existir un momento a partir del cual $\text{rec}x.E[x]$ sea más rápido que $\text{rec}x.F[x]$. Pues \sqsubseteq_f no tolera retardos infinitos.

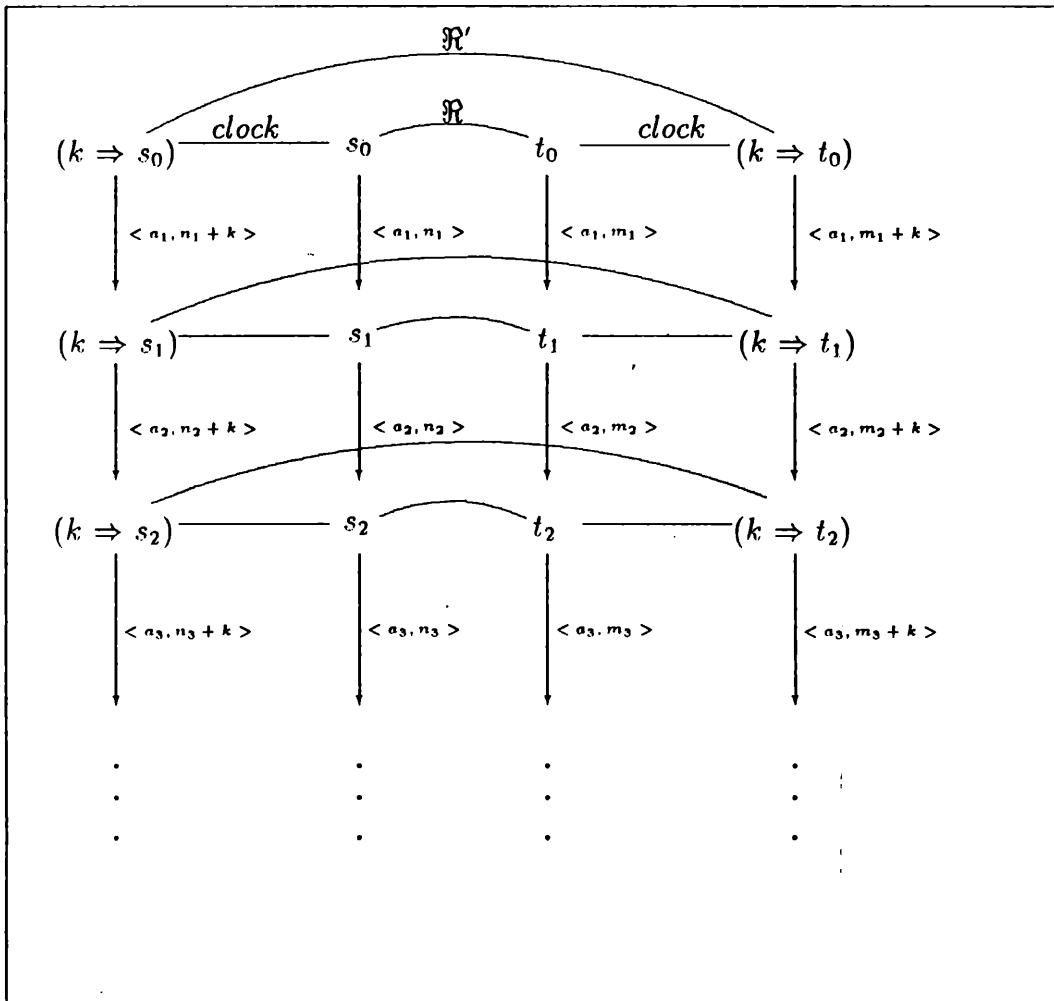


Figura 2.3: Precongruencia para el Retardo

CAPÍTULO 2. PREORDEN "AL FINAL"

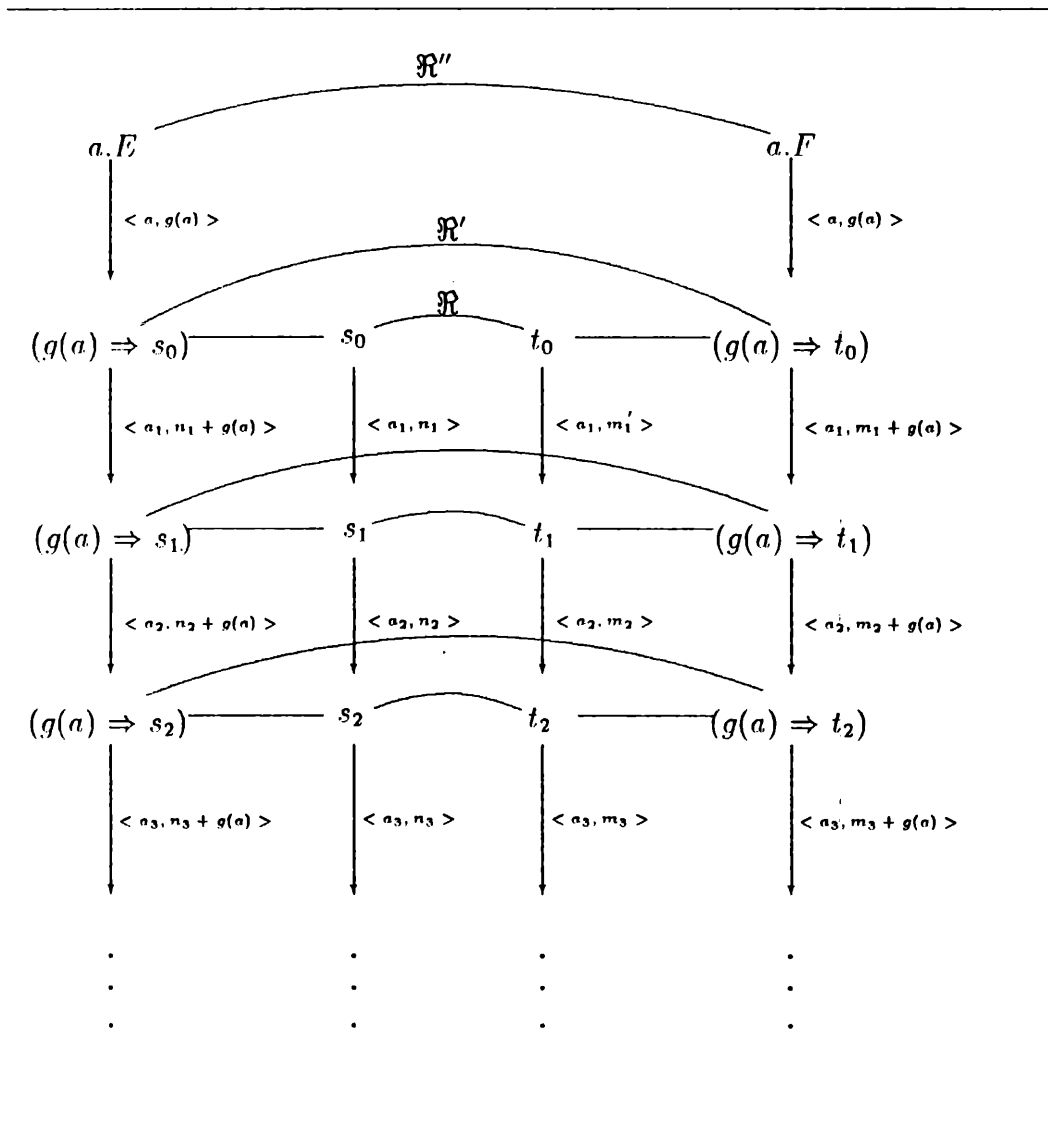


Figura 2.4: Precongruencia para el *Prefixo*

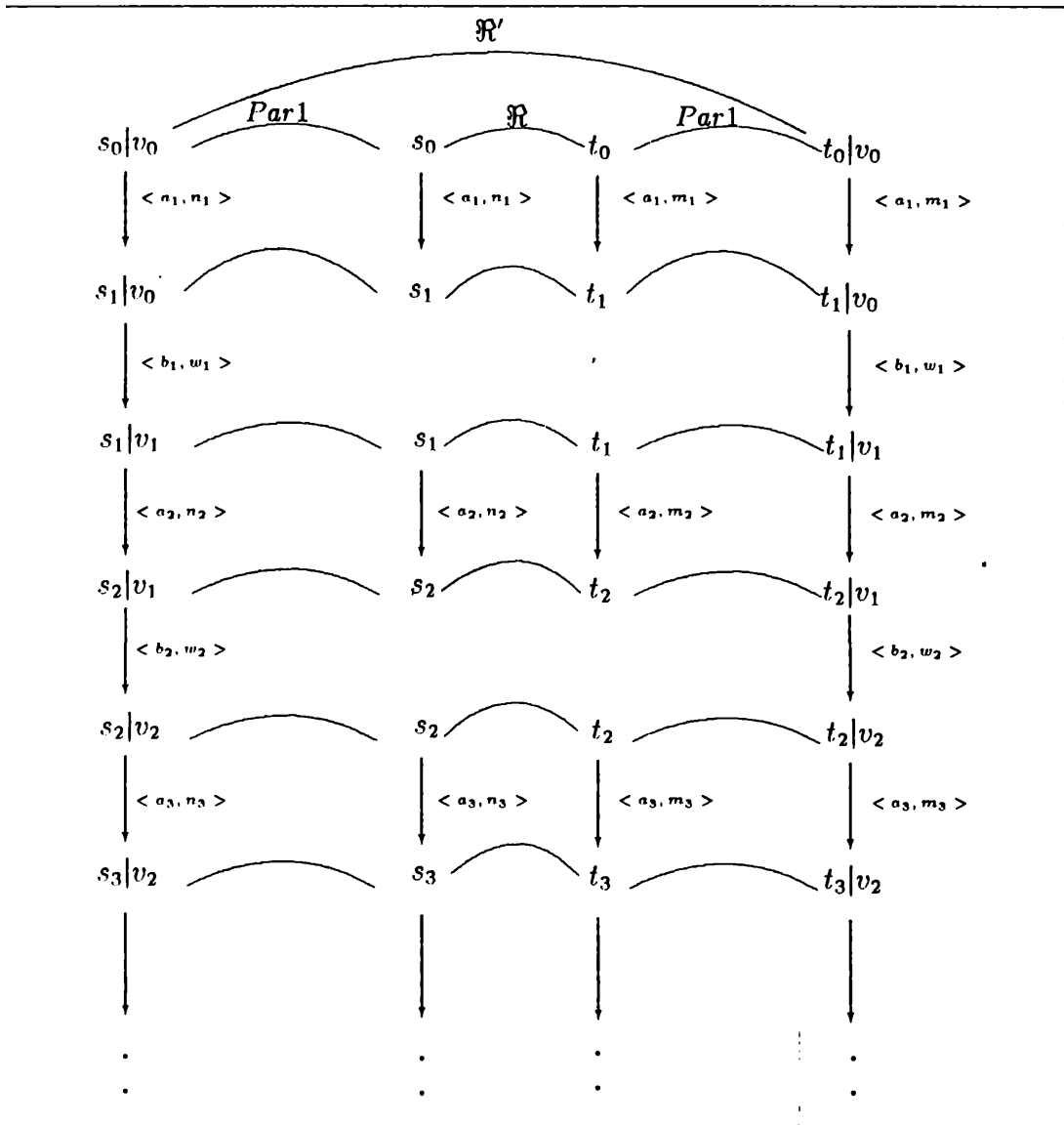


Figura 2.5: Precongruencia para el Paralelo

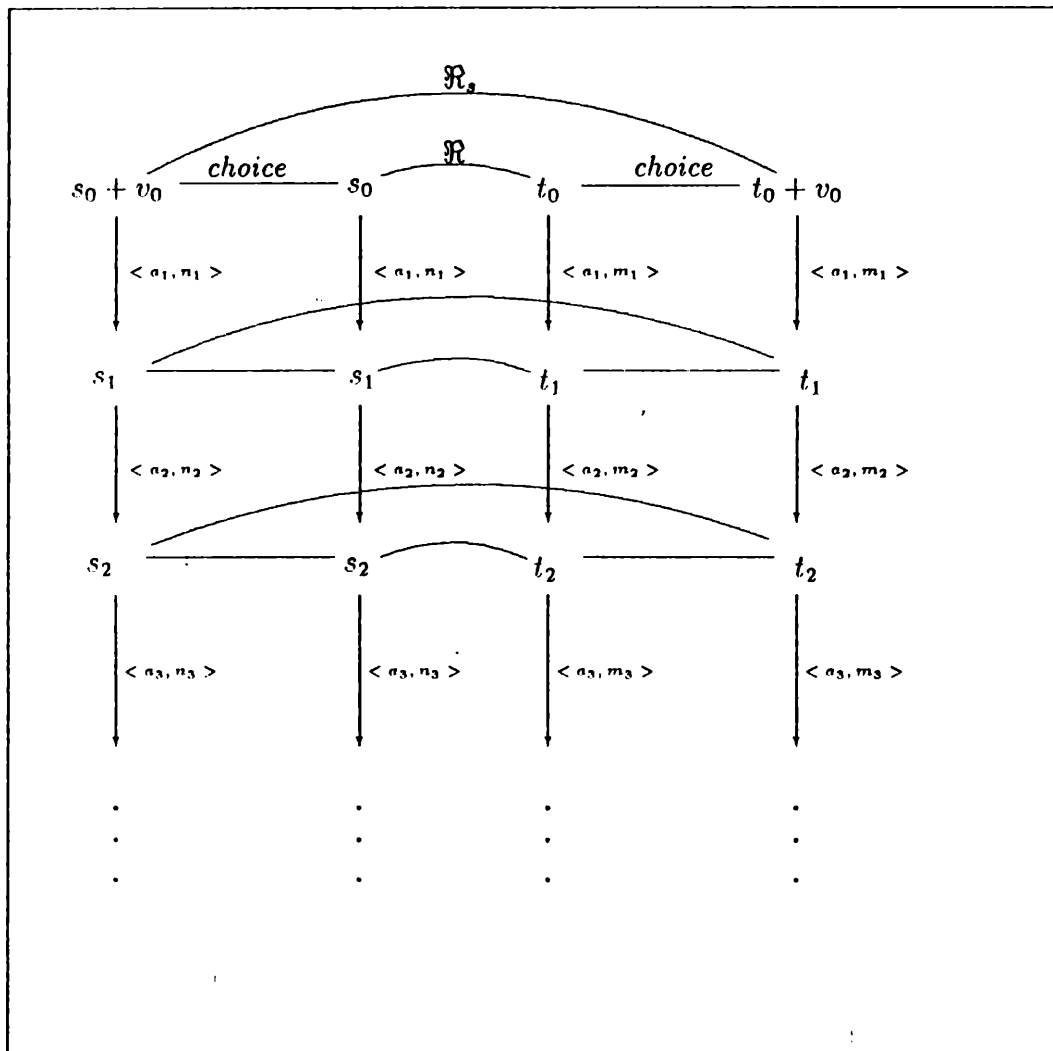


Figura 2.6: Precongruencia para la Selección

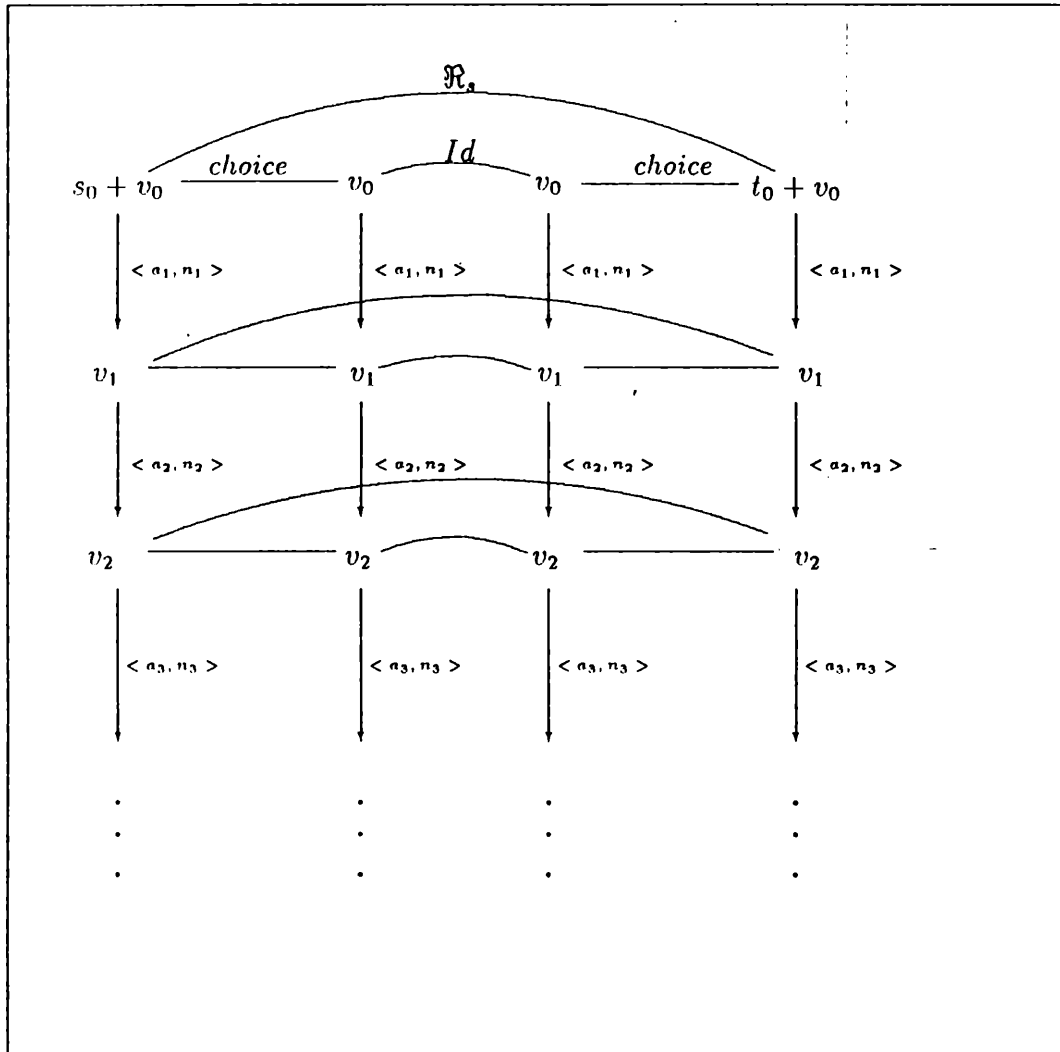


Figura 2.7: Precongruencia para la *Selección*

Capítulo 3

Axiomatización

3.1 Conjunto de Axiomas

El propósito de este capítulo es dar una axiomatización correcta y completa para los procesos finitos del preorden *Al final*, así como también para la relación de equivalencia inducida por el mismo.

Trabajaremos sobre el conjunto de estados finitos S_{fin} , que incluye a los procesos finitos, utilizando la técnica denominada *árboles etiquetados*, en la cual las etiquetas tendrán la forma $\langle a, n \rangle$.

Para obtener un conjunto finito de axiomas necesitamos extender el álgebra propuesto para generar S , con dos operadores auxiliares individualizados como: *Pre* y *Lpar*. Denominaremos T al conjunto de estados que se genera con la sintaxis extendida. La semántica operacional se obtiene extendiendo las reglas de la Fig. 1.1 con las reglas de la Fig. 3.1.

La generalización del *Prefijo* es necesaria para definir la noción de forma normal de un término de T , mientras que el operador *Left-merge*, es necesario para definir un conjunto finito de axiomas reduciendo cada estado de T a su forma normal.

Proposición 3.1.1 \sqsubseteq_f es una precongruencia para los operadores *Pre* y *Lpar*.

Dem.

- 1) $s \sqsubseteq_f t \Rightarrow \langle a, n \rangle: s \sqsubseteq_f \langle a, n \rangle: t$
 $s \sqsubseteq_f t$ implica que existe una *F-bisimulación* \mathfrak{R} tal que $(s, t) \in \mathfrak{R}$. Definimos

$$\mathfrak{R}' = \{ (\langle a, n \rangle: s, \langle a, n \rangle: t) \mid s \mathfrak{R} t \} \cup \mathfrak{R}$$

No es difícil probar que \mathfrak{R}' es una *F-bisimulación*, pues desde el estado $\langle a, n \rangle: s$ podemos aplicar la regla *Pre* pasando al estado s , hacemos lo mismo desde el estado $\langle a, n \rangle: t$, pasando al estado t , sabemos por hipótesis que $(s, t) \in \mathfrak{R}$.

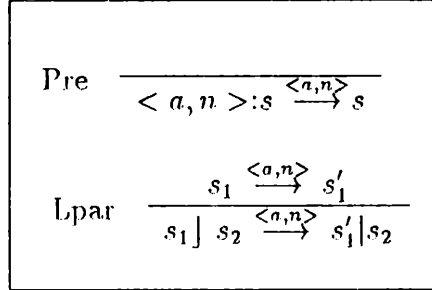


Figura 3.1: Reglas de Inferencia Auxiliares

2) $s \sqsubseteq_f t \Rightarrow (\forall v) s \parallel v \sqsubseteq_f t \parallel v$
 $s \sqsubseteq_f t$ implica que existe una F -bisimulación \mathcal{R} tal que $(s, t) \in \mathcal{R}$. Definimos

$$\mathcal{R}' = \{ (s \parallel v, t \parallel v) \mid s \mathcal{R} t \} \cup \{ (s \parallel v, t \parallel v) \mid s \mathcal{R} t \}$$

No es complicado probar que \mathcal{R}' es una F -bisimulación, pues la regla *Lpar* hace que se realice la primera acción del proceso a la izquierda, y luego se ejecuten en paralelo el proceso obtenido luego de la acción y el proceso de la derecha. Por el Teo. 2.4.9 \sqsubseteq_f es una precongruencia para el operador de composición paralela.

■

Definición 3.1.2 Una forma normal es un término de la forma:

$$\sum_{i \in I} \langle a_i, n_i \rangle : t_i$$

donde $\forall i \in I, t_i$ es una forma normal, $a_i \in Act$ y $n_i \in N$. Por convención la forma normal para *Nil* es con $I = \emptyset$.

Sea A el conjunto de axiomas de Fig. 3.2, el cual forma una axiomatización correcta y completa para la *Equivalencia Competitiva*, propuesta por Gorrieri en [4].

La expresión $A \vdash t_1 = t_2$ significa que se puede probar que los términos t_1 y t_2 son equivalentes utilizando los axiomas del conjunto A y las reglas de inferencia usuales: reflexividad, simetría, transitividad y la sustitución de términos iguales, según la congruencia estructural mencionada en Fig. 1.2.

Para dar una axiomatización de \sqsubseteq_f sobre T , necesitamos la introducción de las desigualdades. Cuando $t_1 = t_2$ es una ecuación en A , entonces $t_1 \leq t_2$ será una desigualdad de A_{\leq} , conjunto de axiomas con desigualdades. Las reglas usuales de inferencia: reflexividad, transitividad y sustitución de \leq se tiene en cuenta mediante lo siguiente:

(<i>sum1</i>)	$t_1 + t_2 = t_2 + t_1$
(<i>sum2</i>)	$(t_1 + t_2) + t_3 = t_1 + (t_2 + t_3)$
(<i>sum3</i>)	$t + t = t$
(<i>sum4</i>)	$t + Nil = t$
(<i>pref</i>)	$a.E = \langle a, f(a) \rangle : (f(a) \Rightarrow E)$
(<i>exp</i>)	$t_1 t_2 = t_1]t_2 + t_2]t_2$
(<i>lpar1</i>)	$Nil]t = Nil$
(<i>lpar2</i>)	$\langle a, k \rangle : t_1]t_2 = \langle a, k \rangle : (t_1 t_2)$
(<i>lpar3</i>)	$(t_1 + t_2)]t_3 = (t_1]t_3) + (t_2]t_3)$
(<i>clock1</i>)	$(n \Rightarrow Nil) = Nil$
(<i>clock2</i>)	$(n \Rightarrow \langle a, k \rangle : t) = \langle a, k + n \rangle : (n \Rightarrow t)$
(<i>clock3</i>)	$(n \Rightarrow (t_1 + t_2)) = (n \Rightarrow t_1) + (n \Rightarrow t_2)$

Figura 3.2: Conjunto de Axiomas A

$(fin1) \quad \langle a, n \rangle: Nil \leq_f \langle a, m \rangle: Nil \quad (n \leq m)$
$(fin2) \quad \langle a, n \rangle: t \leq_f \langle a, m \rangle: t \quad (size(t) \geq 0)$

Figura 3.3: Axiomas para \sqsubseteq_f

$$\frac{t_1 \leq t_2 \text{ y } t_2 \leq t_1}{t_1 = t_2} \qquad \frac{t_1 = t_2}{t_1 \leq t_2 \text{ y } t_2 \leq t_1} \qquad \frac{t_1 = t_2}{t_2 = t_1}$$

Nuestro conjunto de axiomas será A_{\leq} extendido con los axiomas de la Fig. 3.3 y lo denominaremos C .

El axioma $fin2$ tiene una regla equivalente, a la cual denominaremos $fin2'$.

$$(fin2') \quad \frac{t_1 \leq_f t_2}{\langle a, n \rangle: t_1 \leq_f \langle a, m \rangle: t_2} \quad (size(t_1) > 0)$$

Para probar que del axioma se puede deducir la regla, tomamos como hipótesis que $t_1 \leq_f t_2$ y usamos las propiedades de sustitución y transitividad.

Lema 3.1.3 $\langle a, n \rangle: t_1 \sqsubseteq_f \langle a, m \rangle: t_2$ si y sólo si:

$$(1) \quad n \leq m \text{ y } t_1 \simeq_f Nil \simeq_f t_2$$

o

$$(2) \quad t_1 \sqsubseteq_f t_2 \text{ y } t_1 \not\sqsubseteq_f Nil$$

La prueba del lema anterior es trivial, vale por definición.

3.2 Correctitud y Completitud de C

En esta sección vamos a demostrar que el conjunto de axiomas C es correcto y completo. Para demostrar la completitud introduciremos algunas definiciones nuevas y lemas auxiliares, que nos facilitarán la prueba.

Proposición 3.2.1 Para todo $t \in T$, $C \vdash t_1 \leq t_2$ implica $t_1 \sqsubseteq_f t_2$.

Dem. La prueba es estandar. Solamente probaremos la correctitud de los axiomas $(fin1)$ y $(fin2)$. La prueba de los axiomas restantes es una consecuencia de la prueba de correctitud de [4].

- El axioma (*fin1*) es correcto.
Para probar esto es suficiente observar que la relación

$$\mathfrak{R} = \{ (\langle a, n \rangle : Nil, \langle a, m \rangle : Nil) \mid n \leq m \} \cup \{(Nil, Nil)\}$$

es una *F*-bisimulación.

- El axioma (*fin2*) es correcto.
La relación

$$\mathfrak{R} = \{ (\langle a, n \rangle : t, \langle a, m \rangle : t) \mid size(t) \geq 0 \} \cup Id$$

es una *F*-bisimulación por el resultado de Prop. 2.1.3(a), y por ser \sqsubseteq_f una precongruencia para el operador *Pre*. ■

Para probar la completitud del conjunto de axiomas *C*, para el preorden *Al final*, es necesario mostrar que todo estado puede ser reducido a una forma normal.

Proposición 3.2.2 *Para todo $t \in T$ existe una forma normal $nf(t)$ tal que:*

$$A \vdash t = nf(t)$$

Dem. En [4] se prueba que todo término puede ser reducido a una forma normal. En la prueba de esta afirmación se utiliza el conjunto de axiomas *A*, que está incluido en nuestro conjunto de axiomas *C*. La prueba del lema coincide con lo demostrado por Gorrieri en [4]. ■

Definición 3.2.3 El tamaño de un término se define en forma inductiva:

$$\begin{aligned} size(Nil) &= 0 \\ size(a.E) &= 1 + size(E) \\ size(\langle a, n \rangle : t) &= 1 + size(t) \\ size(t_1 + t_2) &= \max(size(t_1), size(t_2)) \\ size(t_1|t_2) &= size(t_1) + size(t_2) \\ size(t_1]t_2) &= size(t_1) + size(t_2) \\ size(n \Rightarrow t) &= size(t) \end{aligned}$$

Lema 3.2.4 *$size(t) = 0$ si y sólo si $A_{\leq} \vdash t = Nil$*

Dem. Por inducción sobre la estructura de *t*.

- 1) $t = Nil$, vale por reflexibilidad. $A_{\leq} \vdash t = Nil$

- 2) $t = a.t_1$, $size(t) = 1 + size(t_1) \neq 0$
- 3) $t = \langle a, n \rangle : t_1$, $size(t) = 1 + size(t_1) \neq 0$
- 4) $t = t_1 + t_2$, $size(t) = 0 = \max(size(t_1), size(t_2))$, si y sólo si $size(t_1) = 0$ y $size(t_2) = 0$. Por hipótesis inductiva $A_{\leq} \vdash t_1 = Nil$ y $A_{\leq} \vdash t_2 = Nil$. Usando el axioma *sum3*, $A_{\leq} \vdash t_1 + t_2 = Nil$.
- 5) $t = t_1 | t_2$, $size(t) = 0 = size(t_1 | t_2)$, si y sólo si $size(t_1) = 0$. Por hipótesis inductiva, $A_{\leq} \vdash t_1 = Nil$ por el axioma *lpar1*, $A_{\leq} \vdash t_1 | t_2 = Nil$
- 6) $t = t_1 | t_2$, por el axioma *exp* $t_1 | t_2 = t_1 | t_2 + t_2 | t_2$, $size(t) = 0 = size(t_1) + size(t_2) = \max(size(t_1 | t_2), size(t_2 | t_2))$, si y sólo si $size(t_1 | t_2) = 0$ y $size(t_2 | t_2) = 0$. Por hipótesis inductiva $A_{\leq} \vdash t_2 | t_2 = Nil$, por el axioma *sum4*: $A_{\leq} \vdash t_1 | t_2 + Nil = Nil$, luego por sustitución $A_{\leq} \vdash t_1 | t_2 + t_2 | t_2 = Nil$. Usando nuevamente el axioma *exp* $A_{\leq} \vdash t_1 | t_2 = Nil$
- 7) $t = (n \Rightarrow t_1)$, $size(t) = 0$ entonces $size(t_1) = 0$. Por hipótesis inductiva, $A_{\leq} \vdash t_1 = Nil$, por el axioma *clock1* $A_{\leq} \vdash (n \Rightarrow t_1) = Nil$

■

Por la correctitud y completitud del conjunto de axiomas A probada por Gorrieri en [4], queda probado también que $size(t) = 0$ si y sólo si $t \simeq_f Nil$.

Proposición 3.2.5 *Para todo $t_1, t_2 \in T$: $t_1 \sqsubseteq_f t_2$ implica $C \vdash t_1 \leq t_2$.*

Dem. Por la proposición Prop. 3.2.2, podemos asumir que t_1, t_2 están en forma normal, a las cuales llamaremos $nf(t_1)$ y $nf(t_2)$ respectivamente.

La prueba se desarrolla por inducción sobre el $\max(size(t_1), size(t_2))$

- **Caso Base:** $\max(size(t_1), size(t_2)) = 0$. Como $(\forall t) size(t) \geq 0$, entonces $size(t_1) = size(t_2) = 0$. Por Lema 3.2.4, $C \vdash t_1 + t_2 = Nil$ y $C \vdash t_2 = Nil$. Entonces $C \vdash t_1 = t_2$. Como $=$ implica \leq , también vale $C \vdash t_1 \leq t_2$.
- **Paso Inductivo:** Suponemos

$$nf(t_1) = \sum_{i \in I} \langle a_i, n_i \rangle : u_i$$

$$nf(t_2) = \sum_{j \in J} \langle b_j, m_j \rangle : v_j$$

Probaremos que: $C \vdash t_1 + t_2 \leq t_2$ y $C \vdash t_1 \leq t_1 + t_2$, luego por la transitividad de \leq , queda probada la proposición.

Como $nf(t_1) \sqsubseteq_f nf(t_2)$:

$$\begin{aligned}
& (\forall j \in J)(\exists i_{(j)} \in I) \text{ tal que: } \langle a_{i_{(j)}}, n_{i_{(j)}} \rangle : u_{i_{(j)}} \sqsubseteq_f \langle b_j, m_j \rangle : v_j \\
& \qquad \qquad \qquad \downarrow_{\text{def.}} \\
& \forall j \in J : \quad 1) a_{i_{(j)}} = b_j, n_{i_{(j)}} \leq m_j \text{ y } u_{i_{(j)}} = Nil \\
& \qquad \qquad \qquad \circ \\
& \qquad \qquad \qquad 2) a_{i_{(j)}} = b_j, u_{i_{(j)}} \sqsubseteq_f v_j \text{ y } u_{i_{(j)}} \neq Nil
\end{aligned}$$

- 1) Por el axioma (*fin1*) $C \vdash \langle a_{i_{(j)}}, n_{i_{(j)}} \rangle : u_{i_{(j)}} \leq \langle b_j, m_j \rangle : v_j$.
- 2) Por el Lema 3.2.4, $t \neq Nil \Rightarrow size(t) > 0$; por hipótesis inductiva $C \vdash u_{i_j} \leq v_j$, aplicando la regla (*fin2'*) obtenemos:

$$C \vdash \langle a_{i_{(j)}}, n_{i_{(j)}} \rangle : u_{i_{(j)}} \leq \langle b_j, m_j \rangle : v_j$$

Haciendo la sumatoria sobre todos los términos tenemos:

$$\sum_{j \in J} \langle a_{i_{(j)}}, n_{i_{(j)}} \rangle : u_{i_{(j)}} \leq \sum_{j \in J} \langle b_j, m_j \rangle : v_j$$

y sumando $nf(t_1)$ a ambos términos, obtenemos:

$$\begin{aligned}
& \sum_{i \in I} \langle a_i, n_i \rangle : u_i + \sum_{j \in J} \langle a_{i_{(j)}}, n_{i_{(j)}} \rangle : u_{i_{(j)}} \leq \\
& \sum_{i \in I} \langle a_i, n_i \rangle : u_i + \sum_{j \in J} \langle b_j, m_j \rangle : v_j
\end{aligned}$$

Usando los axiomas *sum1*, *sum2* y *sum3*, y teniendo en cuenta que $i_{(j)} \in I$, podemos reducir a:

$$\sum_{i \in I} \langle a_i, n_i \rangle : u_i \leq \sum_{i \in I} \langle a_i, n_i \rangle : u_i + \sum_{j \in J} \langle b_j, m_j \rangle : v_j$$

En consecuencia, $t_1 \leq t_1 + t_2$. En forma análoga se prueba: $t_1 + t_2 \leq t_2$. ■

Proposición 3.2.6 Para todo $t_1, t_2 \in T$, $t_1 \simeq_f t_2$ implica $C \vdash t_1 = t_2$.

Dem. Se deduce de la completitud del preorden y de la definición de la *equivalencia al final*. ■

Capítulo 4

Relación con otros preórdenes

En este capítulo estudiaremos otros preórdenes con el fin de encontrar una relación entre cada uno de ellos con el preorden *Al final*.

La noción que utiliza cada preorden es diferente, por ejemplo el *Causal Preorder* de [1] relaciona procesos según el grado de paralelismo que tiene cada proceso. Dice que el proceso x es más causal que el proceso y si x realiza menos acciones en paralelo que y . Nosotros probaremos aquí que un proceso que realiza más acciones en paralelo que otro, no necesariamente es más rápido al final.

Otro preorden estudiado y que tiene una relación con el preorden *Al final* es el *Location Preorder* de [6]. Este preorden relaciona procesos según el grado de distribución espacial de cada uno. Dice que el proceso x está relacionado mediante el preorden con el proceso y , si x está más distribuido espacialmente que y . Mostramos en la segunda sección de este capítulo que si x está más distribuido espacialmente que y , entonces x es más rápido al final que y .

4.1 Causal Preorder

En la teoría de la semántica de los procesos [1], las transiciones son de la forma $p_1 \xrightarrow{u} p_2$, donde p_1, p_2 son procesos, y u es un proceso determinístico, llamado computación.

Definición 4.1.1 El conjunto de computaciones, $Comp$, es generado con la siguiente sintaxis:

$$u ::= Nil \mid a.u \mid u|u$$

donde $a \in Act$.

Para $u \in Comp$, sea \xrightarrow{u} la menor relación binaria en P , que satisface los axiomas y reglas de Fig. 4.1.

$Act1 \quad \frac{}{a.p \xrightarrow{a.Nil} p}$	$Act2 \quad \frac{p \xrightarrow{u} p'}{a.p \xrightarrow{a.u} p'}$
$Alt1 \quad \frac{p \xrightarrow{u} p'}{p + q \xrightarrow{u} p'}$	$Alt2 \quad \frac{p \xrightarrow{u} p'}{q + p \xrightarrow{u} p'}$
$Par1 \quad \frac{p \xrightarrow{u} p'}{p q \xrightarrow{u} p' q}$	$Par2 \quad \frac{p \xrightarrow{u} p'}{q p \xrightarrow{u} q p'}$
$Par3 \quad \frac{p \xrightarrow{u} p', q \xrightarrow{v} q'}{p q \xrightarrow{u v} p' q'}$	

Figura 4.1: Reglas de Inferencia

$(par1) \quad x Nil = x$
$(par2) \quad x y = y x$
$(par3) \quad (x y) z = x (y z)$
$(seq) \quad a.(x y) \leq a.x y$

Figura 4.2: Conjunto de Axiomas

La herramienta para relacionar el no-determinismo y la concurrencia en el comportamiento de los procesos de P , es un preorden entre las computaciones \leq_c . Intuitivamente si u, v son computaciones, diremos que $u \leq_c v$ si u y v ejecutan las mismas acciones, pero en v se realizan en forma más paralela. Formalmente, \leq_c es la menor precongruencia en $Comp$, que satisface el conjunto de axiomas de Fig. 4.2.

El preorden entre procesos de P , se define en base al preorden \leq_c entre las computaciones de $Comp$.

Definición 4.1.2 Sean $p, q \in P$, $p \sqsubseteq_c q$ si y sólo si existe una relación \mathfrak{R} tal que, si $(p, q) \in \mathfrak{R}$:

- 1) Para toda transición $p \xrightarrow{u} p'$, existe $q \xrightarrow{v} q'$, tal que $u \leq_c v$ y $(p', q') \in \mathfrak{R}$.
- 2) Para toda transición $q \xrightarrow{v} q'$, existe $p \xrightarrow{u} p'$, tal que $u \leq_c v$ y $(p', q') \in \mathfrak{R}$.

Proposición 4.1.3 Sean $E_1, E_2 \in P$, entonces:

- 1) $E_1 \sqsubseteq_f^g E_2$ no implica $E_2 \sqsubseteq_c E_1$
- 2) $E_1 \sqsubseteq_f^g E_2$ no implica $E_1 \sqsubseteq_c E_2$
- 3) $E_1 \sqsubseteq_c E_2$ no implica $E_2 \sqsubseteq_f^g E_1$
- 4) $E_1 \sqsubseteq_c E_2$ no implica $E_1 \sqsubseteq_f^g E_2$

Dem.

- 1) $a.a|c \sqsubseteq_f^g a.a|c + c.(a|a)$ con $g(a) \leq g(c)$, sin embargo $a.a|c + c.(a|a) \not\sqsubseteq_c a.a|c$.

En Ej. 2.4.1 mostramos que $a.a|c \sqsubseteq_f^g a.a|c + c.(a|a)$ cuando $g(a) \leq g(c)$.

Observemos (Fig. 4.3 a)) que la transición $c.(a|a)$ del proceso E_2 puede ser imitada únicamente por la transición $c.a.a$ de E_1 , pero $c.(a|a) \not\leq_c c.a.a$.

- 2) $a|b \sqsubseteq_f^g a.b + b.a$ para cualquier función de duración g , sin embargo $a|b \not\sqsubseteq_c a.b + b.a$
- 3) $a.(b + c) + a|b \sqsubseteq_c a.(b + c) + a|b + a.b$, sin embargo $a.(b + c) + a|b + a.b \not\sqsubseteq_f^g a.(b + c) + a|b$

Los procesos son equivalentes con el preorden *Causal*(Fig. 4.3 b)). Cuando el proceso E_2 elige ejecutar el subproceso $a.b$, el proceso E_1 puede imitarlo con transiciones que tienen etiquetas equivalentes, según el preorden para las mismas. Para la etiqueta a y la b tiene las transiciones equivalentes del subproceso $a|b$, mientras que para transición con etiqueta $a.b$ tiene la transición con igual etiqueta correspondiente al subproceso $a.(b + c)$.

Pero $E_2 \not\sqsubseteq_f E_1$, justamente cuando E_2 elige comportarse como el subproceso $a.b$ el proceso E_1 no tiene una forma más lenta de imitarlo, E_1 terminará de ejecutarse en $\max(g(a), g(b))$ mientras que E_2 en $g(a) + g(b)$.

$$4) a.b + b.a \sqsubseteq_c a|b, \text{ y } a.b + b.a \not\sqsubseteq_f^g a|b$$

■

Proposición 4.1.4 \sqsubseteq_f coincide en P_{seq} con \sqsubseteq_c .

Dem. Vale por el Teo. 2.3.5(2) y porque la bisimulación clásica coincide con el preorden causal en P_{seq} , probado en [1]. ■

4.2 Location Preorder

En [6] se define un preorden que relaciona procesos con respecto al grado de distribución espacial. Dados los procesos E_1, E_2 , diremos que $E_1 \sqsubseteq_l E_2$, si E_1 es funcionalmente equivalente a E_2 pero E_1 está más distribuido que E_2 . Se introduce un nuevo sistema de transición que relaciona pares de procesos y secuencias de localidades. Las secuencias de localidades indican donde las acciones son ejecutadas. El sistema de transición es de la siguiente forma:

$$(R, Act, Loc, \{ \xrightarrow{a,u} \mid a \in Act, u \in Loc^* \})$$

donde Loc es un conjunto infinito de nombres de acciones (l_1, l_2, \dots) , R es el conjunto de procesos con localidades, que se obtiene extendiendo la gramática generadora de P con el nuevo operador $l ::$ con $l \in Loc$, llamado prefijo. $l :: E$ significa que el agente E está en la localidad l .

El conjunto de procesos con localidades, R , se genera con la siguiente gramática:

$$r ::= Nil \mid a.E \mid E + E \mid r|r \mid l :: r \mid \text{recx}.E$$

donde $l \in Loc$ y $a.E, \text{recx}.E$ son procesos de P cerrados.

Las reglas para la semántica operacional se pueden observar en la Fig. 4.4.

El preorden está basado en un preorden de las secuencias de localidades. Para $u, v \in Loc^*$, $u \leq v$, significa que u es una subpalabra ordenada de v . Sea $r \in R$, anotaremos el conjunto de los nombres de las localidades atómicas que ocurren en r mediante $loc(r)$. A continuación damos la definición formal del preorden.

Definición 4.2.1 Sean $p, q \in P$, $p \sqsubseteq_l q$ si y sólo si existe una relación \mathfrak{R} tal que, si $(p, q) \in \mathfrak{R}$, para todo $a \in Act$:

- 1) Para toda transición $p \xrightarrow{a,u} p'$ con $u = l_1 \dots l_k.l$ y $l \notin (loc(p) \cup loc(q))$, implica $q \xrightarrow{a,v} q'$ con $v = l'_1 \dots l'_m.l$, tal que $u \leq v$ y $(p', q') \in \mathfrak{R}$.

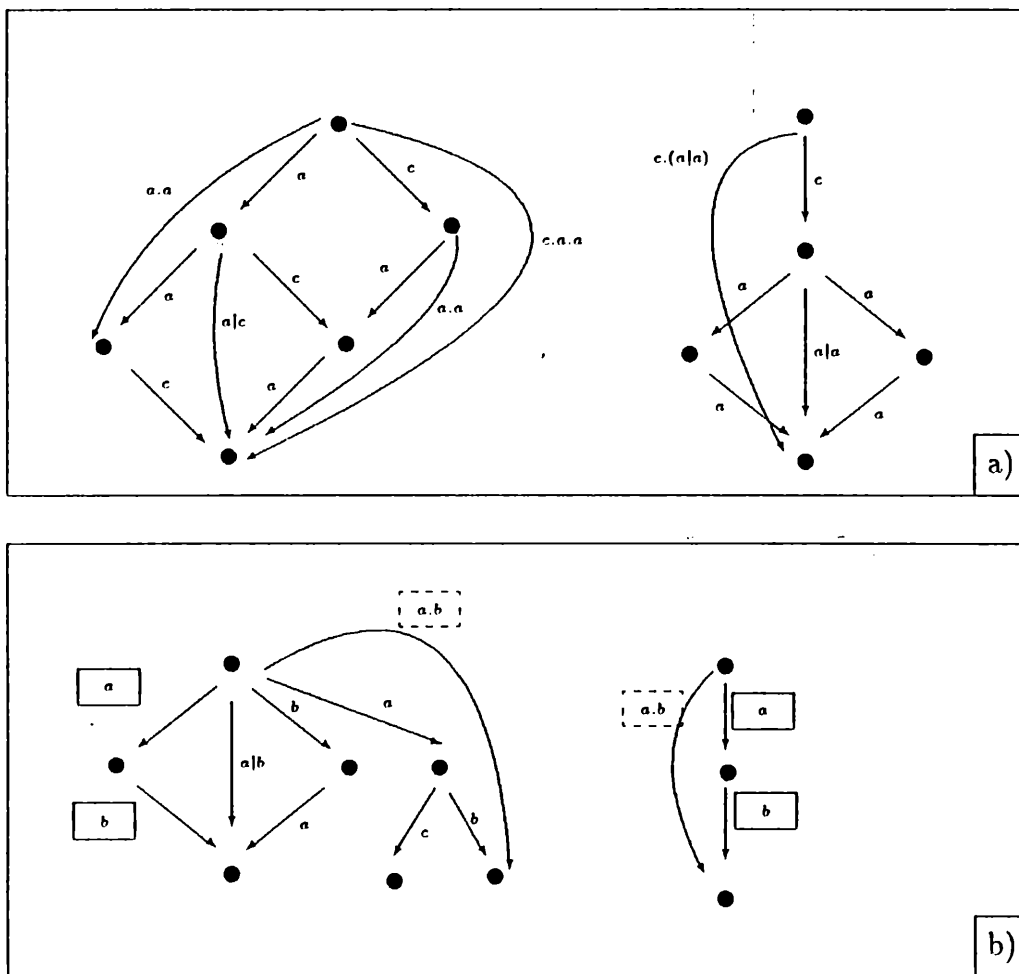


Figura 4.3: a) Procesos no relacionados con \sqsubseteq_c ; b) Relación para \simeq_c

$LT1 \quad \frac{}{a.E \xrightarrow{a,l} l :: E} \quad l \in Loc$	$LT2 \quad \frac{r \xrightarrow{a,u} r'}{k :: r \xrightarrow{a,ku} k :: r'}$
$LT3 \quad \frac{E_1 \xrightarrow{a,u} r'}{E_1 + E_2 \xrightarrow{a,u} r'}$	$LT4 \quad \frac{E_1 \xrightarrow{a,u} r'}{E_2 + E_1 \xrightarrow{a,u} r'}$
$LT5 \quad \frac{r \xrightarrow{a,u} r'}{r s \xrightarrow{a,u} r' s}$	$LT6 \quad \frac{r \xrightarrow{a,u} r'}{s r \xrightarrow{a,u} s r'}$
$LT7 \quad \frac{E[\text{rec } x.E/x] \xrightarrow{a,u} r'}{\text{rec } x.E \xrightarrow{a,u} r'}$	

Figura 4.4: Reglas de Inferencia

2) Para toda transición $q \xrightarrow{a,v} q'$ con $v = l'_1 \dots l'_m.l$ y $l \notin (loc(p) \cup loc(q))$, implica $p \xrightarrow{a,u} p'$ con $u = l_1 \dots l_k.l$, tal que $u \leq v$ y $(p', q') \in \mathcal{R}$.

La definición original del preorden en [6] no requiere que el nombre de la acción sea nuevo. En [4] aparece esta nueva definición, que es equivalente a la original y permite una mejor comparación con el *Performance Preorder*. También con esta definición se facilita la comparación con el preorden *Al Final*.

Gorrieri prueba en [4] que \sqsubseteq implica \sqsubseteq_p , con este resultado y el obtenido en la Prop. 2.2.2, tenemos que si un proceso está más distribuido espacialmente que otro, entonces es más rápido al final que el otro. Mostraremos que no siempre que un proceso sea más rápido que otro, este ha de ser observablemente más distribuido espacialmente. Estos resultados se formalizan en la siguiente proposición.

Proposición 4.2.2 *Sean $E_1, E_2 \in P$, entonces:*

- 1) $E_1 \sqsubseteq_f E_2$ no implica $E_1 \sqsubseteq_l E_2$
- 2) $E_1 \sqsubseteq_l E_2$ implica $E_1 \sqsubseteq_f E_2$

Dem.

1) $a.a|c \sqsubseteq_f a.a|c + c.(a|a)$ con $g(a) \leq g(c)$, sin embargo $a.a|c + c.(a|a) \not\sqsubseteq_l a.a|c$.

Podemos observar en la Fig. 4.5 que no hay forma de asignar localidades a las acciones del subproceso $a.a|c$ cuando ejecuta primero la acción c y luego las acciones a de manera que sea menos distribuido que la ejecución del subproceso $c.(a|a)$.

- 2) En [4] se prueba \sqsubseteq_p implica \sqsubseteq_l , luego por transitividad con la Prop. 2.2.2, \sqsubseteq_f implica \sqsubseteq_f .

La siguiente proposición estudia la relación que existe entre la equivalencia Al Final y la equivalencia a la que induce el *Location Preorder*, definida usualmente como $\simeq_l = \sqsubseteq_l \cap \supseteq_l$.

Proposición 4.2.3 Sean E_1 y E_2 procesos de P .

- 1) $E_1 \simeq_l E_2$ implica $E_1 \simeq_f E_2$;
- 2) Existe una función de duración g tal que $E_1 \simeq_f^g E_2$ no implica $E_1 \simeq_l E_2$.

Dem. Se deriva de la Prop. 4.2.2.

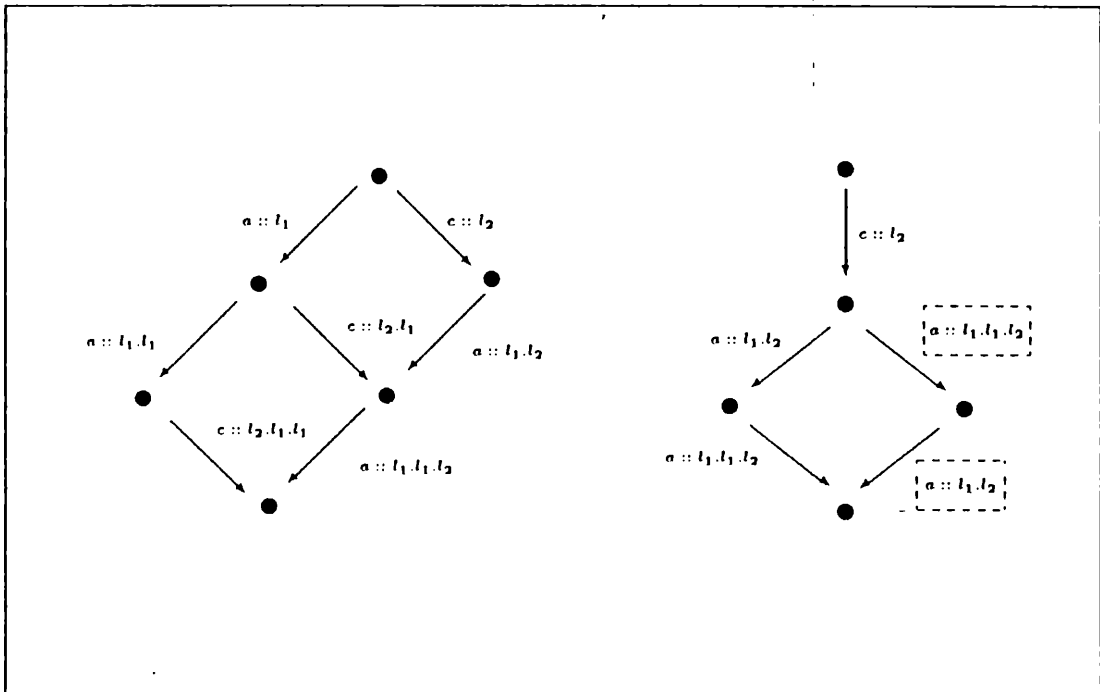


Figura 4.5: Ejemplo de procesos no relacionados con \sqsubseteq_l

4.3 Relación entre "más paralelo" y "más rápido"

Podríamos hacernos preguntas más interesantes aún, como ser "Existe un proceso E que sea a la vez estrictamente más rápido *Al final* que F y estrictamente menos paralelo que F ?" La respuesta a esta pregunta es afirmativa, en la siguiente proposición se enuncia este resultado. La prueba es simplemente un ejemplo, pues queremos demostrar existencia.

Proposición 4.3.1 *Existen procesos E, F tal que: $E \sqsubseteq_f F$ y $E \sqsubseteq_c F$*

Dem. Demostraremos la existencia dando dos procesos E y F que cumplen la proposición, para encontrar dichos procesos haremos:

i) Buscar E_1 y F_1 tales que:

$$E_1 \simeq_f F_1 \text{ y } E_1 \sqsubseteq_c F_1$$

ii) Buscar E_2 y F_2 tales que:

$$E_2 \simeq_c F_2 \text{ y } E_2 \sqsubseteq_f F_2$$

iii) Considerando que :

$$Act(E_1) \cap Act(E_2) = \emptyset$$

$$Act(F_1) \cap Act(F_2) = \emptyset,$$

con $Act(p) =$ Conjunto de acciones del proceso p .

Los procesos que buscamos son $E = E_1 + E_2$ y $F = F_1 + F_2$.

Sean los siguientes procesos:

$$i) \quad E_1 = a.a.a|a.a.a + a.a.a.(a|a|a) + a.a.a.a.a.a$$

$$F_1 = a.a.a|a.a.a + a.a.a.a.a.a$$

Se cumple que:

$$- E_1 \sqsubseteq_f F_1:$$

No es difícil probar que:

$$a.a.a.(a|a|a) \sqsubseteq_f a.a.a.a.a.a$$

pues el segundo proceso es absolutamente secuencial, con lo cual finaliza después de $6.g(a)$ unidades de tiempo mientras que el primero lo hace luego de $4.g(a)$.

$$a.a.a.(a|a|a) + a.a.a.a.a.a \sqsubseteq_f a.a.a.a.a.a$$

por (Teo. 2.4.9) e *Idempotencia*.

$$a.a.a|a.a.a + a.a.a.(a|a|a) + a.a.a.a.a.a \sqsubseteq_f a.a.a|a.a.a + a.a.a.a.a.a$$

por (Teo. 2.4.9).

– $F_1 \sqsubseteq_f E_1$:

$$a.a.a|a.a.a \sqsubseteq_f a.a.a.(a|a|a)$$

vale ya que el primer proceso finaliza luego de $3.g(a)$ unidades de tiempo mientras que el primero lo hace luego de $4.g(a)$ unidades.

$$a.a.a|a.a.a \sqsubseteq_f a.a.a|a.a.a + a.a.a.(a|a|a)$$

por (Teo. 2.4.9) e *Idempotencia*.

$$a.a.a|a.a.a + a.a.a.a.a.a \sqsubseteq_f a.a.a|a.a.a + a.a.a.(a|a|a) + a.a.a.a.a.a$$

por (Teo. 2.4.9).

– $E_1 \sqsubseteq_c F_1$

Se puede probar de una forma intuitiva que el primer proceso es más secuencial que el segundo, pues: $a.a.a.(a|a|a) \leq_c a.a.a|a.a.a$

– $F_1 \not\sqsubseteq_c E_1$

La misma transición de antes nos sirve para probar que F_1 no es más secuencial que E_1 .

ii) $F_2 = d.(b + c) + d|b$

$$F_2 = d.(b + c) + d|b + d.b$$

Se cumple que:

– $E_2 \sqsubseteq_c F_2$

– $F_2 \sqsubseteq_c E_2$

– $E_2 \sqsubseteq_f F_2$

– $F_2 \not\sqsubseteq_f E_2$

Las pruebas son idénticas a las de la primera sección de este capítulo. Necesitamos que las acciones sean otras para que se cumplan las hipótesis para luego hacer la suma de procesos.

■

Conclusión

Hemos definido una teoría para procesos con duración basada en el preorden *Al Final*. El principal objetivo era encontrar un modelo para representar esta clase de procesos, que permitiese establecer relaciones entre éstos, utilizando información relativa a la velocidad o a la rapidéz de cada uno.

Este trabajo se basó en ([4]). Luego de leer su artículo, surgió la pregunta:

”Por qué la teoría basada en el *Performance Preorder* no permite establecer ninguna relación entre dos procesos x e y cuando x comienza su ejecución en forma más lenta que y , pero después de un tiempo acelera su comportamiento superando al proceso y ?”

Esta pregunta nos motivó para comenzar la búsqueda de una nueva relación que considere las situaciones planteadas en la misma. El preorden definido aquí establece una noción de *más rápido al final* que impone condiciones más débiles que la noción de *más rápido* utilizada para el *Performance Preorder*. Para que dos procesos bisimilares, estén relacionados con el preorden *Al final*, uno de ellos debe comportarse, a partir de un determinado momento y hasta el final, de manera más rápida sin tener importancia como se comportaban antes de ese momento.

Los resultados obtenidos mediante el desarrollo se pueden resumir de la siguiente forma: En primer lugar, definimos teoría en base a una relación de preorden para procesos con duración. Probamos que esta teoría es una extensión de la propuesta por Gorrieri et. al en [4]. Luego demostramos que el preorden *Al Final* cumple la propiedad de *precongruencia* para todos los operadores, excepto para el operador de recursión. A continuación presentamos una axiomatización para el preorden, demostrando que es correcta y completa. Comparamos la relación propuesta con otros preórdenes que aparecen en la literatura, encontrando que es incomparable con el *Causal Preorder* de [1] y que es estrictamente más amplio que el *Location Preorder* de [6]. Por último comparamos las nociones de *más paralelo* y *más rápido al final*, probando que existen procesos que pueden ser estrictamente más rápidos y menos paralelos.





Bibliografía

- [1] L. Aceto, *On Relating Concurrency and Nondeterminism*, Technical Report SI/RR-89/06, University of Sussex, 1989.
- [2] L. Aceto, D. Murphy, *Timing and Causality in Process Algebra*, Technical Report 9/93, University of Sussex, 1993.
- [3] S. Arun-Kumar, M. Hennessy, *An Efficiency Preorder for Processes*, Acta Informática **29**, pp.737-760, 1992.
- [4] F. Corradini, R. Gorrieri, M. Rocchetti *Performance Preorder and Competitive Equivalence*, Technical Report UBLCS-95-1, Laboratory of Computer Science, University of Bologna, 1995.
- [5] R. Milner, *Communication and concurrency*, International series on computer science, Prentice Hall International, 1989.
- [6] G. Boudol, I. Castellani, M. Hennessy, A. Kiehn, *A Theory of Processes with Localities*, Formal Aspects of Computing **6**, pp.165-200, 1993.
- [7] M. Hennessy, *The semantics of Programming Languages*, Copyright, 1990.
- [8] F. Moller, C. Tofts, *Relating Processes With Respect to Speed*, Proceedings of CONCUR, LNCS 527, Springer-Verlag, pp. 424-438, 1991.
- [9] R. Alur, C. Courcoubetis, T. Henzinger, *The Observational Power of Clocks*, Proceedings of CONCUR, LNCS 836, Springer-Verlag, pp. 162-177, 1994.
- [10] A. Christoph, J. Echagüe, Ph. Schnoebelen, *Quien ríe último ...*, III Jornadas de Informática e Investigación Operativa, VI Encuentro del Laboratorio de Ciencia de la Computación, Diciembre 1996, Montevideo, Uruguay.

IONACION.....

TES

97/9 ed. 1

Fecha.....

30-8-05

nv

1973

Modelo Semántico de procesos basado en la duración.

Andrea Viviana Christoph

Resumen

En el campo de la semántica de los lenguajes para descripción de procesos, hay un interés en la equivalencia funcional de los mismos. En muchos casos nos interesa, además de la equivalencia funcional, la información relativa a la eficiencia o a la rapidez de los procesos. Este será el enfoque del presente trabajo, donde estudiaremos relaciones que unen dos procesos equivalentes desde el punto de vista funcional, y entre los cuales uno de los procesos es más rápido, o igualmente de rápido que el otro.

Consideraremos que las acciones que puede ejecutar un sistema tiene asociado una duración no nula. Si un proceso tiene varios subprocesos secuenciales en paralelo, cada uno de ellos tendrá un reloj que se modificará solamente cuando una acción local al subproceso sea ejecutada.

La teoría de procesos con duración propuesta por Gorrieri et al. en [4], apunta a describir cuando un proceso es más rápido que otro a partir de un preorden donde todos los comportamientos de un proceso son idénticos a los de un segundo, pero se ejecutan siempre antes en el tiempo. Sin embargo, no permite establecer una comparación entre los procesos x e y cuando y comienza rápidamente, mientras x , luego de un inicio más lento, acelera su funcionamiento superando a y . Dos procesos como estos, no aparecen relacionados de ninguna manera en la propuesta de Gorrieri. Aquí lo que se propone, es un preorden que permite establecer las mismas cosas que Gorrieri, y que además permite capturar este tipo de relaciones, al cual denominaremos *Preorden Al final*, y lo identificaremos con \sqsubseteq_f .

Dos procesos P y Q están relacionados por el preorden *Al final* ($P \sqsubseteq_f Q$) si son funcionalmente equivalentes y P es, a partir de un momento y hasta el final, al menos tan rápido como Q .

Demostraremos que \sqsubseteq_f es una precongruencia para todos los operadores, excepto para la recursión. Probaremos también que la teoría que proponemos incluye en forma propia a la teoría definida por Gorrieri.

El preorden induce, de la forma usual, a una relación de equivalencia a la que denominaremos *Equivalencia Al final*.

Presentaremos axiomatizaciones del preorden y de la equivalencia para procesos finitos, que son correctas y completas.

Compararemos la teoría propuesta con otras que surgen en la literatura. Demostraremos que es incomparable con el *Causal Preorder* propuesto en [1] y es estrictamente más grueso que el *Location Preorder* de [6].

Parte de los resultados que aquí se presentan se obtuvieron trabajando conjuntamente con Philippe Schnoebelen y Pedro D'Argenio. Algunos resultados obtenidos durante el desarrollo del proyecto fueron presentados en [10].

Las actividades involucradas en la preparación de este trabajo fueron parcialmente financiadas por el proyecto BID-CONICYT 140/94, Localidad y Duración en los Sistemas de Transiciones Asíncronas.

TES
97/9
DIF-01973
SALA



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMATICA
Biblioteca
50 y 120 La Plata
catalogo.info.unlp.edu.ar
biblioteca@info.unlp.edu.ar