Prioridades en un modelo de verdadero paralelismo

Pedro R. D'Argenio[†] Juan V. Echagüe[†‡] Leticia Ramos[*]

pedro@info.unlp.edu.ar echague@fing.edu.uy lramos@info.unlp.edu.ar

[†] *LIFIA, Depto. de Informática, Fac. de Cs. Exactas, Universidad Nacional de La Plata*

[‡] *INCO, Fac. de Ingeniería, Universidad de república*

[*] *Depto. de Informática, Fac. de Cs. Exactas, Universidad Nacional de La Plata*

Resumen

En este trabajos se desarrolla un estudio sobre prioridades en sistemas de verdadero paralelismo mediante la introducción del operador de prioridades en sistemas de transición que admiten la representación de ST estados. El estudio introduce cuatro definiciones alternativa. Se muestra que ellas preservan ciertas propiedades y en particular que una de ellas es la extensión natural del operador de prioridades definido en modelos de interleaving.

1 Introducción

En este trabajo nos interesamos en la *semántica de sistemas reactivos* [Pnu85]. Estos sistemas (cuyos ejemplos clásicos son los sistemas de control industrial, los manejadores de redes de computadoras y los componentes de los sistemas operativos) tienen como vocación el mantener con el medio una cierta interacción.

No es adecuado entonces considerarlos simplemente como mecanizaciones del cálculo de una función o una relación, como es el caso de otros sistemas, que por ello los llamamos *funcionales* o *relacionales*. Los ejemplos típicos de estos últimos sistemas son los programas de cálculo numérico, los compiladores y todos aquellos cuya interacción con el medio se reduce a un único punto de entrada de datos y a uno de salida de resultados.

Los modelos más comunes para la semántica de sistemas reactivos están basados en los *Sistemas de Transiciones Etiquetados* (STE) [Kel76], que son grafos dirigidos donde los nodos representan los estados del sistema, y las flechas las transiciones entre los estados, etiquetadas por acciones en las que se basa la interacción del sistema con el medio.

La popularidad de los STE es debida a su simplicidad, la posibilidad de representarlos gráficamente, la sencillez con que pueden definirse sobre ellos los constructores usuales de sistemas reactivos (composición secuencial, no determinista, paralela, operadores de comunicación y de prioridad, entre otros) y a la existencia de una técnica simple y elegante, la *Semántica Operacional Estructurada* [Plo81] para dar semántica sobre los STE a los lenguajes de programación (inclusive los no reactivos, ver [Hen90]). Muchos lenguajes simples, llamados *lenguajes de descripción de procesos* han sido propuestos para los sistemas reactivos, por ejemplo CCS [Mil80, Mil89], CSP [BHR84, Hoa85], ACP [BK85, BW90] y ATP [NRSV90, NS94]. La gran mayoría de ellos disponen de semántica sobre los STE.

En los STE, la ejecución de un programa puede verse como una sucesión de estados, y de flechas etiquetadas que unen esos estados, simplemente un camino en el grafo que parte del estado inicial:

$$i \overset{a_1}{\dashrightarrow} s_1 \overset{a_2}{\dashrightarrow} s_2 \overset{a_3}{\dashrightarrow} \ldots \overset{a_n}{\dashrightarrow} s_n$$

Este es el origen de la principal limitación en el uso de los ST: no podemos representar aquí dos acciones desarrollándose al mismo tiempo, ya que se debe llegar a un estado (terminando una acción) para poder iniciar una nueva acción. Decimos por esto que los ST no pueden representar directamente el paralelismo. Aunque sí puede hacerlo indirectamente, representando la ejecución de a en paralelo con b como la secuencia a, b o bien la secuencia b, a. Por razones obvias llamamos de entrelazamiento (en ingles *interleaving*) a esta representación del paralelismo en los STE.

La principal limitación entonces de los STE como modelos de sistemas reactivos es que el paralelismo no puede representarse directamente, sino mediante entrelazamiento.

Hay muchos modelos que permiten representar el paralelismo sin utilizar el entrelazamiento, que son llamados en general *sistemas verdaderamente paralelos* (en inglés *truly concurrent*). Algunos de ellos son las Redes de Petri [Pet62, Rei85], las Estructuras de Eventos [Win87] y los Sistemas de Trancisiones Asíncronos [Bed87, Shi85]. En este trabajo nos interesamos en un modelo propuesto recientemente por *Gorrieri y Laneve* [GL95] que llamaremos aqui STTS.

Una de las principales ventajas de los STTS es que son una variante sencilla de los ST. En realidad, en [GL95] y [BGG94], los autores manejan estos modelos como simples ST con ciertas propiedades sobre sus etiquetas, que son garantizadas a partir de una manera especial de construirlos. Estas propiedades les permiten representar el paralelismo sin necesidad de recurrir al entrelazamiento.

A fin de clarificar nuestra exposición, en este trabajo nosotros preferimos considerarlos como un modelo diferente de los STE, con propiedades diferentes.

La idea esencial es que un STTS nos permite expresar la noción elemental de *duración*, donde varias fases de una acción pueden ser ejecutadas entre su inicio y su final. Para ello cada acción sera representada por su *inicio* y su *finalización* correspondiente. En ese sentido un STTS puede verse como un STE donde las etiquetas de las flechas no son acciones $(a, b \ldots)$, sino inicios de acciones (que anotaremos como las acciones, $a, b \ldots$) y los finales correspondientes (que anotaremos con superíndices $a^i, b^j \ldots$, donde i y j representan cual es exactamente la acción que está terminando entre las acciones a y b iniciadas en el pasado).

Esto nos permite representar fácilmente ejecuciones donde las acciones a y b son ejecutadas en paralelo. Una de ellos es, por ejemplo

$$i \xrightarrow{a} s_1 \xrightarrow{b} s_2 \xrightarrow{b^1} s_3 \xrightarrow{a^3} s_4$$

Al llegar al estado s_2, se han ejecutado los inicios de las acciones a y b, pero no el final de ninguna de ellas. Por tanto, *ambas acciones se están ejecutando en paralelo*. La tercera transición, etiquetada b^1 es el final de la acción b iniciada un paso atrás, mientras la transición etiquetada con a^3 es el fin de la acción a iniciada 3 pasos atrás. En este ejemplo los superíndices no parecen estrictamente necesarios, bastaría con poder diferenciar inicios de finales. Sin embargo, en el caso en que haya varias acciones a iniciadas, es imprescindible, cuando ocurre el final de una de ellas, saber de quien se trata: en ese caso el superíndice brinda la información necesaria.

Los operadores de prioridad forman una familia interesante entre los operadores sobre los STE. Se construyen a partir de un cierto orden en las acciones, y su aplicacion provoca la poda de las alternativas de "menor prioridad" entre las transiciones que salen de cada estado.

Estos operadores fueron propuestos por primera vez para modelos de los sistemas reactivos en [BBK86] sobre STE. Formalmente cada operador de prioridad está asociado a un orden parcial $\geq$ en el conjunto de las acciones. Cuando es aplicado a un STE, en cada estado son consideradas las etiquetas de todas las transiciones que parten de allí, y sólo aquellas etiquetadas por acciones maximales según $\geq$ son preservadas.

Debido a que el paralelismo no puede representarse directamente en los STE, cuando se aplica prioridad entre dos acciones que se están ejecutando en forma paralela, el resultado será que la menos prioritaria deberá esperar a que la otra acción termine para poder iniciar su ejecucion. Esto limita el espectro de representación de prioridad en un sistema verdaderamente paralelo. Por ejemplo: Dos personas desean ser atendidas en una farmacia que cuenta con dos empleados. Una de ellas tiene el número de orden uno, y la otra el número dos. El empleado que primero se libere, comenzará a atender a la persona con el número uno, pues tiene prioridad sobre la otra. Si mientras el cliente con el número uno está siendo atendido el otro empleado se libera, podrá atender a la persona con el número dos, no siendo necesario esperar a que la otra termine de ser atendida. Incluso la segunda persona puede terminar primero, si realiza su compra más rápido.

Esta situación no es representable con las definiciones de prioridad conocidas hasta el momento. En este trabajo se presenta un concepto de prioridad que captutra la idea descripta en el ejemplo, definiendo un operador adecuado sobre el modelo STTS. Lo que se hará será definir el orden parcial sobre el inicio o el final de las acciones o sobre ambos. Esto permitirá introducir distintas alternativas en la definición de prioridad.

A pesar de que el operador de prioridad de los STE no sirve para representar ciertas situaciones en los sistemas verdaderamente paralelos, cumple ciertas propiedades que sería deseable poder verificar en el operador de prioridad definido para los STTS. Esto pude hacerse de dos maneras: Enunciar la propiedad que se desea verificar y probar que se cumple, o tratar de establecer una compatibilidad entre ambos operadores de prioridad. En este trabajo se eligió la segunda alternativa. Para esto se define una función que permite traducir acciones divididas en inicio y final (acciones de los STTS) en acciones indivisibles (acciones de los STE). Utilizando esta definición, se prueba que los operadores de prioridad conmutan. Es decir, es lo mismo aplicar prioridad en los STTS y luego traducir a STE, que traducir a STE y luego aplicar prioridad. Pero esta conmutatividad no siempre es posible y depende de cómo se defina el orden parcial. En este trabajo se concluye que cuando el orden parcial se define sobre los finales de las acciones, la conmutatividad no es posible.

Existe una propiedad deseable en los sistemas verdaderamente paralelos que no se puede obtener de los STE debido a la imposibilidad de representar verdadero paralelismo. Esta propiedad expresa que toda acción que comienza puede ser terminada. En este trabajo se prueba que esto siempre es posible. Cuando se analiza esta propiedad para el operador de prioridad, en algunos casos se podrá decir que las acciones pueden ser finalizadas siempre y en cualquier momento. En otros, sólo que pueden ser terminadas sin poder determinar el momento exacto. Esta situación también dependerá del conjunto de acciones sobre el que se defina el orden parcial, llegando a la conclusión que cuando se define sobre el final de las acciones, no se puede determinar el momento exacto en el que la acción finaliza.

El único estudio de fenómenos de prioridad en modelos verdaderamente paralelos que hemos encontrado en la literatura es [?]. Allí los autores muestran una variación enriquecida de las estructuras de eventos, donde un preorden entre eventos representa la prioridad. Creemos que todo lo representable en dicho modelo, lo es también en el primero de los que aquí mostramos.

El trabajo está organizado como sigue: la sección 2 introduce los conceptos básicos de sistemas de transición etiquetados y define el operador de prioridades en este marco. En las sección 3 se definen los sistemas de transición ST y se da una primera definición del operador de prioridad en este contexto. Tres definiciones alternativas del operador de prioridades sobre los STTS son definidas en la sección 4. Finalmente se establece una conclusión del trabajo realizado.

2 Sistemas de transiciones etiquetados

Definición 2.1 (Sistema de transición etiquetado) Un *sistema de transicion etiquetado* (STE), es una estructura $G = (S, i, \mathbf{A}, \dashrightarrow)$ donde

- $S = \{i, u, s, \ldots\}$ es el conjunto de *estados* donde i es el *estado inicial*,
- $\mathbf{A} = \{a, b, e, \surd, \ldots\}$ es el conjunto de *eventos* o *acciones*, con una accion destacada $\surd$ que indica terminación satisfactoria.
- $\dashrightarrow \subseteq S \times \mathbf{A} \times S$, es la *relación de transición* (escribimos $s \overset{e}{\dashrightarrow} s_1$, por $(s, e, s_1) \in \dashrightarrow$)

∎

2.1 Prioridad en los STE

Definición 2.2 (Operador de prioridad en los STE) Dado un STE $G = (S, i, \mathbf{A}, \dashrightarrow)$, para cada orden parcial $\geq$ en el conjunto de acciones $\mathbf{A}$, se define al operador de prioridad $\theta : G \to G$ como

$$\text{PRI} \quad \frac{p \overset{a}{\dashrightarrow} q \quad \forall b > a . p \overset{b}{\not\dashrightarrow}}{\theta(p) \overset{a}{\dashrightarrow} \theta(q)} \qquad\qquad \text{PRI}' \quad \frac{p \overset{\surd}{\dashrightarrow} \delta}{\theta(p) \overset{\surd}{\dashrightarrow} \delta}$$

Formalmente deberíamos haber escrito $\theta_{\geq}$ en lugar de θ pero en general omitiremos el subíndice $\geq$ cuando quede claro en el contexto. ∎

3 Sistemas de transición ST y el operador de prioridad

La prioridad provee un mecanismo de decisión entre 2 o mas eventos en conflicto, imponiendo un orden en el tiempo cuando éstos se ejecutan en forma concurrente. Comenzamos nuestra investigación recordando la definición de los STTS y proponiendo una definición de prioridad para ellos.

El concepto de prioridad conocido dice que si se tienen dos procesos ejecutandose en forma concurrente y uno es mas prioritario, éste se ejecutará primero. El otro sólo prodrá ejecutarse una vez que el primero haya terminado. Pero cuando tenemos acciones cuya evolución puede ser observada en el transcurso de su ejecución, resulta muy estricto y poco útil exigir que la acción más prioritaria termine para que la otra pueda comenzar. Por ejemplo: Dos personas desean ser atendidas en una farmacia que cuenta con dos empleados. Una de ellas tiene el número de orden 1 y la otra el número 2. El empleado que primero se libere comenzará a atender a la persona con el número uno, pues tiene prioridad sobre la otra. Si mientras el cliente con el número uno esta siendo atendido el otro empleado se libera, podra atender a la persona con el número 2 sin esperar a que la otra termine de ser atendida. Incluso la segunda persona puede terminar primero si realiza su compra mas rápido.

El objetivo de esta sección es definir un operador de prioridad que atrape esta intuición.

3.1 Sistemas de Transición ST

En los sistemas de transición ST, cambia la noción de atomicidad permitiendo que una acción pueda ser observada en la mitad de su evolución. No se asume mas que cada acción es atómica y por lo tanto no observable en su evolución. Esto se logra dividiendo la acción en dos partes : principio y final. Cada final estará asociado a un único inicio y esto se determina con funciones auxiliares que asignan a cada final un superíndice $i > 0$ que indica cuantos pasos atrás se inicio la acción. Por ejemplo, a^3 indica que se está terminando una acción que comenzó 3 pasos atrás. Definimos formalmente un sistema de transición ST como:

Definición 3.1 (Sistemas de Transición ST) Un *sistema de transición ST* (STTS) es una estructura $G = (S, i, \mathbf{A}, \longrightarrow)$

- $S = \{i, u, s, \ldots\}$ es el conjunto de *estados* donde i es el *estado inicial*,
- $A = \{a, b, e, \ldots\}$ es el conjunto de *comienzo de acciones*, $A' = \{a^n | a \in A, n > 0\}$ es el conjunto de *terminacion de acciones*; ambos determinan el conjunto de *eventos* $\mathbf{A} = A \cup A' \cup \{\surd\}$ con un evento destacado $\surd$ que indica terminación satisfactoria.
- $\longrightarrow \subseteq S \times \mathbf{A} \times S$, es la *relación de transición*.

■

A los efectos de este trabajo, consideraremos los STE y los STTS que pueden representarse mediante un lenguaje de descripción de procesos simple que llamamos $\mathcal{L}$, donde cada término se define de acuerdo a:

$$t ::= a \mid t \cdot t' \mid t + t' \mid t ||_S t' \mid \varepsilon \mid \delta$$

donde $a \in A$ y $S \subseteq A$. Los operadores $\cdot$ y $+$ son la composición secuencial y alternativa de ACP[BW90]. Con el fin de poder definir el sistema de transición, se consideran dos tipos de operadores adicionales según la siguiente extensión del lenguaje previo:

$$t ::= \ldots \mid a^1 \mid [n]t$$

donde $a \in A$ y n es un entero positivo. Siguiendo la técnica de Plotkin [Plo81], el conjunto de reglas que define el comportamiento de los operadores está dado en la tabla 1 donde $\alpha \in A \cup A'$ y

$$\begin{array}{lcl}
name(a) & = & a \\
name(a^n) & = & a
\end{array}
\qquad\qquad
\begin{array}{lcll}
inc(a,n) & = & a & \\
inc(a^n,m) & = & a^n & \text{si } n > m \\
inc(a^n,m) & = & a^{n+1} & \text{si } n \leq m
\end{array}$$

$$a \xrightarrow{a} a^1 \qquad \frac{p \xrightarrow{\alpha} p'}{p+q \xrightarrow{\alpha} p' \quad q+p \xrightarrow{\alpha} p'} \qquad \frac{p \xrightarrow{\alpha} p'}{p \cdot q \xrightarrow{\alpha} p' \cdot q}$$

$$a^1 \xrightarrow{a^1} \epsilon \qquad \frac{p \xrightarrow{\surd} \delta}{p+q \xrightarrow{\surd} \delta \quad q+p \xrightarrow{\surd} \delta} \qquad \frac{p \xrightarrow{\surd} \delta \quad q \xrightarrow{\alpha} q'}{p \cdot q \xrightarrow{\alpha} q'}$$

$$\epsilon \xrightarrow{\surd} \delta \qquad \frac{p \xrightarrow{\surd} \delta \quad q \xrightarrow{\surd} \delta}{p \cdot q \xrightarrow{\surd} \delta}$$

$$\frac{p \xrightarrow{\alpha} p'}{p||_S q \xrightarrow{\alpha} p'||_S [1]q \quad q||_S p \xrightarrow{\alpha} [1]q||_S p'} \; name(\alpha) \notin S \qquad \frac{p \xrightarrow{\alpha} p'}{[n]p \xrightarrow{inc(\alpha,n)} [n+1]p'}$$

$$\frac{p \xrightarrow{\alpha} p' \quad q \xrightarrow{\alpha} q'}{p||_S q \xrightarrow{\alpha} p'||_S q'} \; name(\alpha) \in S \qquad \frac{p \xrightarrow{\surd} \delta}{[n]p \xrightarrow{\surd} \delta}$$

$$\frac{p \xrightarrow{\surd} \delta \quad q \xrightarrow{\surd} \delta}{p||_S q \xrightarrow{\surd} \delta}$$

Tabla 1

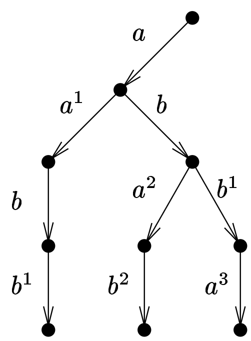
3.2 Un operador de prioridad para STTS

Definición 3.2 (El operador de prioridad θ en STTS) Dado un STTS $G = (S, i, \mathbf{A}, \longrightarrow)$, para cada orden parcial $\geq$ en el conjunto de acciones $\mathbf{A}$, se define el operador de prioridad $\theta : G \to G$ de acuerdo a las reglas

PRI I	$\dfrac{p \xrightarrow{a} q \quad \forall b > a. p \stackrel{b}{\not\longrightarrow}}{\theta(p) \xrightarrow{a} \theta(q)}$	PRI II	$\dfrac{p \xrightarrow{a^i} q}{\theta(p) \xrightarrow{a^i} \theta(q)}$	PRI III	$\dfrac{p \xrightarrow{\surd} \delta}{\theta(p) \xrightarrow{\surd} \delta}$

$\blacksquare$

Ejemplo 3.1 $\theta(a||_\emptyset b)$ con $a > b$



Cuando a y b se inician, como $a > b$, tiene prioridad el inicio de a. Entonces se anula la rama donde b puede comenzar. En el estado 2 se puede elegir entre finalizar a o comenzar b pues el final de una acción no compite con el inicio de otra. La misma situación ocurre en el estado 3 pues los finales no compiten entre si. $\blacksquare$

3.3 Toda acción que comienza termina

Cuando se presantaron los STTS al comienzo de esta sección, se dijo que cada inicio tiene asociado un único final. Lo que no se dijo es que en los STTS definidos por $\mathcal{L}$ este final siempre es alcanzable y exactamente en el momento que se desee. En el gráfico de ejemplo 3.1 se puede observar que por cualquier rama que tome, todas las acciones que comenzaron llegan al final y cada rama provee distintos momentos para terminar cada acción, lo que permite poder decidir en qué momento se desea terminar.

Formalmente este concepto se puede expresar de la siguiente manera:

Teorema 3.1

$$p \xrightarrow{a} p' \implies (\forall \sigma : p' \xrightarrow{\sigma} r \ y \ \forall i \in \{1 \ldots |\sigma|\}. \sigma(i) \neq a^i \implies r \xrightarrow{a^{|\sigma|+1}})$$

Prueba. Por inducción en la estructura de p. Se mostrará sólo el caso

$$\theta(p) \xrightarrow{a} \theta(p') \implies (\forall \sigma : \theta(p') \xrightarrow{\sigma} \theta(r) \text{ y } \forall i \in \{1 \ldots |\sigma|\}. \sigma(i) \neq a^i \implies \theta(r) \xrightarrow{a^{|\sigma|+1}})$$

Por PRI I,

$$p \xrightarrow{a} p' \text{ y } \forall b > a : p \stackrel{b}{\not\longrightarrow}$$

y entonces por inducción

$$p \xrightarrow{a} p' \xrightarrow{\sigma} r \text{ con } \forall i \in \{1 \ldots |\sigma|\}. \sigma(i) \neq a^i)$$

Como la hipótesis vale para todo σ, en los σ tales que $p' \xrightarrow{\sigma}$ quedan incluidos los que están formados por acciones maximales. Entonces

$$
\begin{array}{cccccc}
\theta(p) & \xrightarrow{a} & \theta(p') & \xrightarrow{\sigma} & \theta(r) & \xrightarrow{a^{|\sigma|+1}} \\
 & \Big\downarrow \begin{array}{l}\text{PRI I}\end{array} & & \Big\downarrow \begin{array}{l}\text{PRI I}\\ \text{PRI II}\end{array} & & \Big\uparrow \begin{array}{l}\text{PRI II}\end{array} \\
p & \xrightarrow{a} & p' & \xrightarrow{\sigma} & r & \xrightarrow{a^{|\sigma|+1}}
\end{array}
$$

■

3.4 La prioridad conmuta con la transformación natural de STTS en STE

Se ha presentado una definición para un operador de prioridad sobre los STTS, inspirado en uno ya definido sobre los STE. Es deseable que, por ejemplo, en aquellos STTS que no contienen combinadores paralelos, y que son, por lo tanto, esencialmente STE, la aplicación de ambos operadores devuelva el mismo resultado. El objetivo de esta sección es presentar un resultado formal que nos asegura propiedades como esta y justifica por lo tanto la utilización del término "prioridad" para designar el operador definido.

Este resultado se basa en la posibilidad de establecer una función, aqui llamada MapSt, que devuelva para cada STTS aquel STE que se comporta de una manera similar, pero sin tomar provecho del paralelismo. Esta está definida por las siguientes reglas:

$$
\text{MapSt} \quad \frac{p \xrightarrow{a} q \quad q \xrightarrow{a^1} r}{p \overset{a}{\dashrightarrow} q} \qquad\qquad \text{MapSt}' \quad \frac{p \xrightarrow{\surd} \delta}{p \overset{\surd}{\dashrightarrow} \delta}
$$

Como vemos, en MapSt los estados se mantienen y también aquellas acciones cuyo final se ejecuta inmediatamente después de su inicio.

Para formalizar la noción de que ambos operadores de prioridad representen la misma intuición, mostraremos que el operador de prioridad conmuta con MapSt

Es decir que aplicando prioridad a un STTS y luego mapeando a STE mediante MapSt, se obtiene el mismo resultado que mapeando primero el STTS y luego aplicando el operador de prioridad de STE. Esto se demuestra formalmente con el siguiente teorema.

Teorema 3.2 $\text{MapSt}(\theta(p)) = \theta(\text{MapSt}(p))$

Prueba. Teninendo en cuenta el teorema 3.1 sabemos que para todo p, existen q y r tal que $p \xrightarrow{a} q \xrightarrow{a^1} r$. Entonces, por un lado tenemos que

$$
\text{MapSt}\,\frac{\text{PRI I}\,\dfrac{p \xrightarrow{a} q \quad \forall b > a. p \overset{b}{\not\rightarrow}}{\theta(p) \xrightarrow{a} \theta(q)} \quad \text{PRI II}\,\dfrac{q \xrightarrow{a^1} r}{\theta(q) \xrightarrow{a^i} \theta(r)}}{\theta(p) \overset{a}{\dashrightarrow} \theta(q)}
$$

y por el otro

$$\text{PRI}\,\dfrac{\text{MapSt}\,\dfrac{p \xrightarrow{a} q \quad q \xrightarrow{a^1} r}{p \overset{a}{\dashrightarrow} q} \quad \text{MapSt}\,\dfrac{\forall b > a.p \overset{b}{\not\longrightarrow}}{\forall b > a.p \overset{b}{\not\dashrightarrow}}}{\theta(p) \overset{a}{\dashrightarrow} \theta(q)}$$

El caso en que $\theta(p) \xrightarrow{\surd} \delta$ es trivial. ∎

4 Otras definiciones para el operador de prioridades

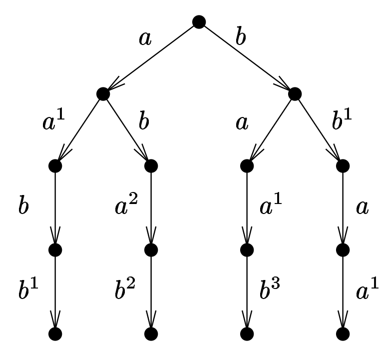
Hasta ahora fue analizado el comportamiento del operador de prioridad aplicando el orden parcial sobre el inicio de las acciones. Pero también se podría querer aplicar prioridad en el momento que dos acciones están finalizando. Por ejemplo: Se cuenta con un sistema compuesto por dos computadoras y una impresora. Cada computadora tiene un editor de textos. En determinado momento se deben imprimir dos textos. El proceso de impresión se puede ver como escribir el texto en el editor e imprimir el texto. Como cada computadora tiene un editor, los textos pueden ser escritos al mismo tiempo. Pero si ambos se mandan a imprimir en el mismo momento, uno deberá ser elegido para ser impreso primero. Por lo tanto se necesita un criterio de prioridad sobre el final de las acciones.

Esta situación se formaliza de la siguiente manera:

4.1 Reglas de prioridad cuando sólo compiten finales

$$\text{PRI I}\quad \dfrac{p \xrightarrow{a} q}{\theta(p) \xrightarrow{a} \theta(q)} \qquad \text{PRI II}\quad \dfrac{p \xrightarrow{a^j} q \quad \forall b > a.p \overset{b^i}{\not\longrightarrow} \quad i > 0}{\theta(p) \xrightarrow{a^j} \theta(q)} \qquad \text{PRI III}\quad \dfrac{p \xrightarrow{\surd} \delta}{\theta(p) \xrightarrow{\surd} \delta}$$

Ejemplo 4.1 $\theta(a||_{\emptyset}b)$ con $a > b$



∎

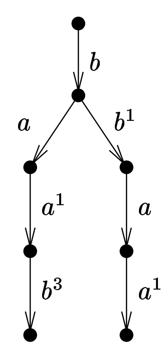
Otra posibilidad para aplicar prioridad es hacerlo sobre los inicios y también sobre los finales de las acciones. En el ejemplo anterior esto sería necesario si sólo se tuviera un editor de textos. En el ejemplo de la farmacia, sería necesario si sólo hubiera un cajero para realizar el cobro de la compra. Por más que un cliente haya comenzado a ser atendido primero, ambos pueden terminar la compra al mismo tiempo, por lo que se necesitaría un criterio de decisión para elegir a quien cobrar primero. Ese criterio puede no ser el mismo que el usado para evaluar el inicio de la compra. Ello implica que pueden usarse dos ordenes parciales diferentes sobre las acciones, uno para los inicios y otro para los finales.

Esta situación se formaliza de la siguiente manera:

4.2 Reglas de prioridad cuando compiten inicios entre sí y finales entre sí

PRI I	$\dfrac{p \xrightarrow{a} q \quad \forall b > a.p \not\xrightarrow{b}}{\theta(p) \xrightarrow{a} \theta(q)}$
PRI II	$\dfrac{p \xrightarrow{a^j} q \quad \forall b \succ a.p \not\xrightarrow{b^i} \quad i > 0}{\theta(p) \xrightarrow{a^j} \theta(q)}$
PRI III	$\dfrac{p \xrightarrow{\surd} \delta}{\theta(p) \xrightarrow{\surd} \delta}$

Ejemplo 4.2 $\theta(a||_{\emptyset}b)$ con $b > a$ y $a \succ b$



■

Como se puede ver en PRI II, los ordenes parciales son distintos.

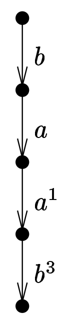
Aún se puede analizar otra posibilidad y es la competencia entre un inicio y un final. Supongamos que en la farmacia del ejemplo, la persona que atiende es la misma que cobra. En el momento de cobrar al cliente que está siendo atendido, llega otro cliente a la farmacia con una herida que debe ser curada de inmediato. La persona que atiende deberá establecer un criterio de prioridad entre el cobro y la atencion del nuevo cliente. Ello implica que pueden usarse tres órdenes parciales diferentes sobre las acciones, uno para los inicios, otro para los finales y un tercero para comparar los inicios con lo finales.

Esta situación se formaliza de la siguiente manera:

4.3 Reglas de prioridad cuando compiten inicios y finales

$$
\begin{array}{ll}
\text{PRI I} & \dfrac{p \xrightarrow{a} q \quad \forall b > a.p \stackrel{b}{\not\longrightarrow} \quad \forall c \succ a.p \stackrel{c^i}{\not\longrightarrow} \quad i > 0}{\theta(p) \xrightarrow{a} \theta(q)} \\[4ex]
\text{PRI II} & \dfrac{p \xrightarrow{a^j} q \quad \forall b \gg a.p \stackrel{b^i}{\not\longrightarrow} \quad i > 0 \quad \forall c \succ a.p \stackrel{c}{\not\longrightarrow}}{\theta(p) \xrightarrow{a^j} \theta(q)} \\[4ex]
\text{PRI III} & \dfrac{p \xrightarrow{\surd} \delta}{\theta(p) \xrightarrow{\surd} \delta}
\end{array}
$$

Ejemplo 4.3 $\theta(a||_{\emptyset}b)$ con $b > a$, $a \succ b$ y $a \gg b$



∎

Si se observan los ejemplos con antención, se puede ver que cuando compiten finales, ya no se puede afirmar que una acción iniciada, puede ser terminada en el momento que se desee. Esto es muy claro en el ejemplo previo, donde la acción b no puede ser terminada inmediatamente después que fue comenzada.

Lo que sí se puede afirmar cuando compiten finales, es que siempre existe algún camino que llevara al final de cualquier acción comenzada, pero el final no será en el momento que se elija, sino en algún momento. Esto se formaliza con el teorema 4.2.

Lema 4.1 *Para todo término p, y para todo σ, $p \xrightarrow{\sigma}$ implica $|\sigma| < f(p)$, donde $f(p)$ está dado por el doble de la cantidad de acciones que ocurren en p.*

Teorema 4.2 *Si $p \xrightarrow{a} p'$ entonces $\forall\sigma.\exists\rho.p' \xrightarrow{\sigma} p'' \xrightarrow{\rho} r$ y $\forall i \in \{1..|\sigma|\}.\sigma(i) \neq a^i$ y $\forall j \in \{1..|\rho|\}.\rho(i) \neq a^{j+|\sigma|}$ implica $r \xrightarrow{a^{|\sigma\rho|+1}} r'$.*

Prueba. Por inducción en la estructura de p. Consideraremos sólo el caso $p \equiv \theta(q)$.

$$\theta(q) \xrightarrow{a} \theta(q') \xrightarrow{\sigma} \theta(q'')$$

implica, por PRI I y PRI II

$$q \xrightarrow{a} q' \xrightarrow{\sigma} q''$$

y por hipótesis inductiva existe σ_1

$$q \xrightarrow{a} q' \xrightarrow{\sigma} q'' \xrightarrow{\sigma_1} r_1 \xrightarrow{a^{n_1}}$$

con $n_1 = |\sigma\sigma_1| + 1$.

Aquí se presentan dos casos:

Caso 1. Si $q'' \xrightarrow{\sigma_1} r_1 \xrightarrow{a^{n_1}}$ es un camino maximal, entonces usando PRI I y PRI II, inmediatamente se obtiene,

$$\theta(q'') \xrightarrow{\sigma_1} \theta(r_1) \xrightarrow{a^{n_1}}$$

Caso 2. Si σ_1 no es maximal, entonces consideramos el prefijo σ_1^{ini} mas largo de σ_1 que este formado por acciones maximales. Podemos escribir, entonces, $\sigma_1 a^{n_1} = \sigma_1^{ini}\sigma_1^{fin}a^{n_1}$, y considerar

$$q \xrightarrow{a} q' \xrightarrow{\sigma} q'' \xrightarrow{\sigma_1^{ini}} \xrightarrow{b_{max_1}} p_1''$$

donde b_{max_1} es una acción maximal, justamente una de aquellas que hace que σ_1^{ini} sea el prefijo maximal más largo. Por hipótesis inductiva

$$q \xrightarrow{a} q' \xrightarrow{\sigma\sigma_1^{ini}b_{max_1}} p_1'' \xrightarrow{\sigma_2} r_2 \xrightarrow{a^{n_2}}$$

con $n_2 = |\sigma\sigma_1^{ini}b_{max_1}\sigma_2| + 1$, y entonces el mismo razonamiento se vuelve a aplicar.

Notar que $\sigma\sigma_1^{ini}b_{max_1}$ es estrictamente mas largo que σ ya que, si bien σ_1^{ini} puede ser vacío, b_{max_1} siempre es una acción

Eso basta para garantizar que en alguna iteración se usará el caso 1. Aumentando el largo del camino maximal en cada paso se llegará eventualmente al caso donde

$$|\sigma\sigma_1^{ini}b_{max_1}\sigma_2^{ini}b_{max_2}\ldots\sigma_n^{ini}b_{max_n}|$$

mida $f(n) - 1$, es decir uno menos que el camino más largo. Este camino debe poder completarse inmediatamente con a^{n+1}, ya que en otro caso la hipótesis inductiva no se cumpliría. Entonces ese camino puede completarse con a^{n+1} que es la acción maximal (y única en este caso). Por lo tanto ese camino genera uno de la forma

$$\theta(q) \xrightarrow{a} \theta(q') \xrightarrow{\sigma} \theta(q'') \longrightarrow \ldots \xrightarrow{a^{n_1}}$$

■

Se concluye de esta propiedad que no existe conmutatividad con STE ya que la función de MapST solo mapea aquellas acciones que finalizan ni bien fueron iniciadas y como ya se dijo, esto no siempre pasa cuando compiten los finales.

5 Conclusiones

En este trabajo se ha explorado la definición de operadores de prioridad en un modelo verdaderamente paralelo. Los operadores de prioridad son bien conocidos en la literatura, y en los STE se representan como la eliminación, entre las transiciones que salen de cada estado, de aquellas cuyas etiquetas no son maximales en el orden parcial de prioridad entre las acciones.

En ese sentido, las transiciones "compiten", y sólo aquellas que tienen etiquetas maximales de acuerdo al preorden dado son preservadas.

Se ha trabajado aquí sobre los STTS, un modelo muy cercano a los STE, pero donde las acciones están divididas en inicio y final. Se propuso cuatro definiciones diferentes del operador de prioridad, según la competencia entre las transiciones involucre los inicios y/o los finales de las transiciones.

Se ha prestado especial interés a la más simple entre ellas, donde sólo los inicios de las acciones compiten entre ellos. Se mostró allí que se cumple la propiedad esencial de los STTS, aquella que dice que todo las transiciones que se inician, para cualquier evolución del sistema, terminan en algún momento. La propiedad es inclusive algo más fuerte: toda acción no terminada puede finalizar en cualquier momento que se decida. Esto permite dar una transformación natural de los STTS en los STE. Se ha probado que la definición de prioridad conmuta con esta transformación. Eso asegura en algun sentido que la operación de prioridad definida representa la misma intuición que la definida en los STE.

Se ha estudiado también con cuidado el caso donde sólo los finales compiten. En este caso, la propiedad que se encuentra es algo mas complicada: para toda transición iniciada, toda evolución ulterior del sistema que no la termine puede completarse con una evolución (que eventualmente involucra varios pasos) de manera que la transición pueda finalizarse. En este sentido este operador es mas complejo que el anterior. Nos permite llegar a STTS cuya traducción en terminos de STE no es directa.

Los estudios presentados en este trabajo recién están iniciandose. Si son muchas las cuestiones que quedan abiertas, hay dos que nos interesa explorar en un futuro cercano. La primera es estudiar las propiedades de compatibilidad de los operadores presentados respecto a equivalencias semánticas que consideren el verdadero paralelismo, en particular con la ST-bisimulacion [GV87]. La segunda está motivada por la dificultad para obtener una traducción entre los STTS y los STE en el caso de aplicación de algunos operadores. A nuestro entender, esto parece indicar que este modelo es verdaderamente rico y la técnica de definición por reglas potente. Vale la pena entonces buscar representar en ese marco otras operaciones y propiedades.

Referencias

[BBK86] J.C.M. Baeten, J.A. Bergstra, and J.W. Klop. Syntax and defining equations for an interrupt mechanism in process algebra. *Fundamentae Informaticae*, IX(2):127–168, 1986.

[Bed87] M.A. Bednarczyc. *Categories of asynchronous systems.* PhD thesis, University of Sussex, 1987.

[BGG94] N Busi, R. van Glabbeek, and R. Gorrieri. Axiomatising ST-bisimulation equivalence. In E.-R. Olderog, editor, *Proceedings of PROCOMET'94, IFIP 2 Working Conference*, San Miniato, 1994. North-Holland.

[BHR84] S.D. Brookes, C.A.R. Hoare, and A.W. Roscoe. A theory of communicating sequential processes. *Journal of the ACM*, 31(3):560–599, July 1984.

[BK85] J.A. Bergstra and J.W. Klop. Algebra of communicating processes with abstraction. *Theoretical Computer Science*, 37:77–121, 1985.

[BW90] J.C.M. Baeten and W.P. Weijland. *Process algebra.* Cambridge University Press, 1990.

[GL95] R Gorrieri and C Laneve. Split and st bisimulation semantcs. *Information and Computation*, 1995. To appear.

[GV87] R. van Glabbeek and F. Vaandrager. Petri nets models for algebraic theories of concurrency. In J.W. de Bakker, A.J. Nijman, and P. Treleaven, editors, *Proceedings of PARLE conference*, volume II, pages 224–242, Eindhoven, 1987. LNCS 259, Springer-Verlag.

[Hen90] Mathew Hennessy. *The semantics of programming languages: an elementary introduction using structural operational semantics.* Jhon Wiley & Sons, Sussex, 1990.

[Hoa85] C.A.R. Hoare. *Communicating sequential process.* Prentice Hall, 1985.

[Kel76] R.M. Keller. Formal verification of parallel programs. *Communications of the ACM*, 8(19):371–384, 1976.

[Mil80] Robin Milner. *A calculus of communicating systems*, volume 92 of *LNCS*. Springer-Verlag, 1980.

[Mil89] Robin Milner. *Communication and concurrency.* Prentice Hall, 1989.

[NRSV90] X. Nicollin, J.-L. Richier, J. Sifakis, and J. Voiron. ATP: an algebra for timed processes. In M. Broy and C.B. Jones, editors, *Proceedings of the IFIP TC 2 Working Conference on Programming Concepts and Methods*, Sea of Gallilee,Israel, April 1990. North-Holland.

[NS94] X. Nicollin and J. Sifakis. The algebra of timed processes, ATP: theory and application. *Information and Computation*, 114:131–178, 1994.

[Pet62] C.A. Petri. *Kommunikation mit Automaten.* PhD thesis, University of Bonn, 1962.

[Plo81] G.D. Plotkin. A structural approach to operational semantics. Report DAIMI-FN-19, Computer Science Department, University of Århus, 1981.

[Pnu85] Amir Pnueli. Linear and branching structures in the semantic and logic of the reactive systems. In W. Brauer, editor, *Proceedings 12th. ICALP*, pages 15–32, Nafplion, 1985. LNCS 194, Springer-Verlag.

[Rei85] W. Reissig. *Petri Nets, An Introduction.* Springer-Verlag, EATCS Monograph on Thoeretical Computer Sciences, 1985.

[Shi85] V.R. Shields. Concurrent machines. *The Computer Journal*, 28(5):449–465, 1985.

[Win87] Glynn Winskel. Event structures. In W. Brauer, W. Reisig, and G. Rozemberg, editors, *Proceedings of Advances in Petri Nets 1986 (Part II)*, pages 325–392, Bad Honnef, 1987. LNCS 255, Springer-Verlag.