

sPBC: Una Extensión Markoviana del Petri Box Calculus

Hermenegilda Macià Soler

Diciembre 2003

Tesis Doctoral presentada al Departamento de Informática de la
Universidad de Castilla-La Mancha para la obtención del grado de
Doctor en Informática

Agradecimientos

Esta primera página de esta tesis (que para mí, es la última que escribo y ... ¡qué ganas tenía!) es para mostrar mi gratitud a todas aquellas personas que han hecho que esto sea posible.

... a mi director de tesis Valentín Valero, por su disposición, por su tiempo, por su dedicación, por su ayuda, por sus consejos siempre sabios, por su paciencia conmigo, por su apoyo, por su amistad, ...

... a Diego Cazorla por su inestimable ayuda con el LaTeX, por estar siempre disponible cada vez que lo he necesitado (y han sido muchísimas veces), por escucharme, por su apoyo, por su amistad, ...

... a Fernando Cuartero por sus consejos, su confianza en mí, por su apoyo, por su amistad, ...

... a aquellas amigas y amigos que han aguantado todas mis neuras, ...

... a mis padres por haberme dado la oportunidad de estudiar y por tantas otras cosas ...

... pero en especial a Joaquín, por quererme, por aguantarme, por soportarme, por cuidar de nuestros hijos (ya que mami siempre tenía "deberes"), por todo.

Por último, quiero dedicar esta tesis a Joaquín, a mis hijos y a mis padres.

Índice General

1	Introducción	1
2	Antecedentes	7
2.1	Modelos formales de concurrencia	7
2.1.1	Redes de Petri	8
2.1.2	Álgebras de Procesos. CCS.	15
2.1.3	Petri Box Calculus	19
2.2	Evaluación de Prestaciones	40
2.2.1	Cadenas de Markov de Tiempo Continuo (CTMC)	41
2.2.2	Redes de Petri estocásticas	43
2.2.3	Álgebras de Procesos estocásticas	50
3	sPBC finito	59
3.1	Sintaxis	59
3.1.1	Notación	59
3.1.2	s-Expresiones Estáticas	61
3.1.3	s-Expresiones Dinámicas	62
3.2	Semántica Operacional	63
3.2.1	Reglas de inacción y equivalencia estructural	63
3.2.2	Reglas de transición estocásticas	66

3.2.3	Sistema de transiciones	73
3.2.4	La política de competencia	80
3.2.5	Ejemplos	82
3.2.6	El proceso estocástico asociado	87
3.3	Semántica denotacional	89
3.3.1	Las s-cajas planas	90
3.3.2	El álgebra de las s-cajas	91
3.3.3	Ejemplos	93
4	Una nueva propuesta para la sincronización	97
4.1	Problemas con la sincronización	97
4.2	La semántica operacional	102
4.3	Semántica denotacional	120
4.4	Algunas observaciones sobre la sincronización	127
4.4.1	Comparación con otras SPAs	130
5	Añadiendo la iteración	135
5.1	Sintaxis	136
5.2	Semántica Operacional	137
5.3	Semántica Denotacional	148
6	Equivalencias	155
6.1	Las nuevas reglas estocásticas	155
6.2	Congruencia	161
6.3	Bisimulación estocástica	206
7	Casos de estudio	213
7.1	Región crítica	213

7.1.1	Enfoque centralizado	214
7.1.2	Anillo lógico	217
7.2	Productor/Consumidor	219
7.2.1	Almacén con capacidad para dos ó más ítems	222
7.3	El protocolo del Bit Alternante	225
8	Conclusiones y Trabajo Futuro	231
8.1	Trabajos publicados	233
8.2	Trabajo Futuro	235

Índice de Tablas

2.1	Semántica operacional de CCS	17
2.2	Reglas de inacción (I) de PBC	25
2.3	Reglas de inacción (II) de PBC	26
2.4	Reglas de transición de la semántica operacional de PBC	27
2.5	Reglas Skp y Rdo de PBC	29
2.6	Semántica operacional de PEPA	55
3.1	Reglas de inacción (I)	64
3.2	Reglas de inacción (II)	64
3.3	Reglas de transición estocásticas (I)	67
3.4	Reglas de transición estocásticas del operador <i>sy</i>	70
4.1	Reglas para la nueva propuesta de sincronización	106
5.1	Reglas de inacción para el operador de iteración	137
5.2	Reglas de transición estocástica del operador $[**]$	138
6.1	Reglas Skp y Rdo de sPBC	157
6.2	Regla de transición <i>Fantasma</i>	159

Índice de Figuras

2.1	Ejemplo de red de Petri marcada	9
2.2	$ts(\overline{E_1})$ y $ts(\overline{F_1})$	28
2.3	$ts(\overline{E_2})$ y $ts(\overline{F_2})$	29
2.4	$ts_2(\overline{E_1})$ y $ts_2(\overline{F_1})$	30
2.5	$ts_2(\overline{E_3})$ y $ts_2(\overline{F_3})$	30
2.6	$ts_2(\overline{E_4})$ y $ts_2(\overline{F_4})$	33
2.7	Definición de Ω y Σ	37
2.8	Ejemplo de $op_\Omega(\Sigma)$. Fase 1: obtención de Θ_1 y Θ_2	37
2.9	Ejemplo de $op_\Omega(\Sigma)$. Fase 2: $\Omega[t_1 \leftarrow \Theta_1] = \Omega'$	37
2.10	Ejemplo de $op_\Omega(\Sigma)$. Fase 2: $\Omega'[t_2 \leftarrow \Theta_2] = op_\Omega(\Sigma)$	38
2.11	cajas operador de PBC finito	38
2.12	caja operador (no segura) para $[**]$ de PBC	38
2.13	caja operador (segura) para $[**]$ de PBC	39
2.14	caja plana de E_5	40
2.15	Red de Petri que modela un multiprocesador de memoria compartida	48
2.16	Grafo de accesibilidad de la red que modela el multiprocesador de memoria compartida	48
2.17	CTMC de la red que modela el multiprocesador de memoria compartida	49
3.1	Sistema de transiciones de $\overline{E_1}$	82
3.2	Sistema de transiciones de $\overline{E_2}$	83

3.3	Sistema de transiciones de $\overline{E_3}$	84
3.4	Sistema de transiciones de $\overline{E_4}$	84
3.5	Sistema de transiciones asociado a $\overline{E_5}$	86
3.6	Diagrama de transición de estados de la CTMC asociada a $\overline{E_1}$	87
3.7	Diagrama de transición de estados de la CTMC asociada a $\overline{E_2}$	88
3.8	Diagrama de transición de estados de la CTMC asociada a $\overline{E_3}$	88
3.9	Diagrama de transición de estados de la CTMC asociada a $\overline{E_4}$	89
3.10	Diagrama de transición de estados de la CTMC asociada a $\overline{E_5}$	89
3.11	s-cajas operador de sPBC finito	92
3.12	s-caja plana de E_1	94
3.13	Grafo de accesibilidad de (Box_{E_1}, M_e)	94
3.14	s-caja plana de E_2	95
3.15	s-caja plana de E_3	95
3.16	s-caja plana asociada de E_4	96
3.17	s-caja plana de E_5	96
4.1	$ts(\overline{(E_1 \parallel F_1) sy a})$	99
4.2	$ts(\overline{(E_2 \parallel F_2) sy a})$ con la primera propuesta para sy	99
4.3	$ts(\overline{(E_2 \parallel F_2) sy a})$ con la nueva propuesta para sy	101
4.4	Sistema de transiciones asociado a G_3	118
4.5	Diagrama de transición de estados de la CTMC asociada a $\overline{(E_2 \parallel F_2) sy a}$	120
4.6	s-caja plana de E_3	125
4.7	s-caja de E_4	125
4.8	s-caja plana de Sys	126
4.9	s-caja plana asociada a E	130
5.1	Sistema de transiciones de $\overline{E_1}$	147

5.2	Sistema de transiciones de $\overline{E_3}$	148
5.3	s-caja operador para $[**]$	149
5.4	s-caja plana de E_1	151
5.5	s-caja plana de E_2	151
5.6	s-caja plana de E_3	152
6.1	$ts(\overline{E_1})$ y $ts(\overline{F_1})$	156
6.2	$ts(\overline{E_2})$ y $ts(\overline{F_2})$	157
6.3	$ts_2(\overline{E_1})$ y $ts_2(\overline{F_1})$	158
6.4	$ts_2(\overline{E_3})$ y $ts_2(\overline{F_3})$	158
6.5	$ts_2(\overline{E_4})$ y $ts_2(\overline{F_4})$	158
6.6	$ts_3(\overline{E_3})$ y $ts_3(\overline{F_3})$	161
6.7	$ts_3(\overline{E_5})$ y $ts_3(\overline{F_5})$	161
6.8	$nts(\overline{E_5})$ y $nts(\overline{F_5})$	169
6.9	$nts(\overline{E_9})$	170
6.10	$nts(\overline{F_9})$	170
6.11	$nts(\overline{E_{10}})$	208
6.12	$ats^{\mathcal{R}}(\overline{E_{10}})$	209
6.13	$ats^{\mathcal{R}}(\overline{E_{11}})$	210
7.1	s-caja plana de P_j	215
7.2	s-caja plana de Q	215
7.3	s-caja plana de P_j (anillo lógico)	219
7.4	s-caja plana de P_1 (anillo lógico)	219
7.5	s-caja plana del <i>Productor</i> .	221
7.6	s-caja plana del <i>Consumidor</i>	221
7.7	s-caja del <i>Almacén</i>	221

7.8	s-caja plana de <i>Almacén₂</i>	223
7.9	s-caja plana de <i>Almacén_n</i>	224
7.10	La estructura del ABP	225
7.11	s-caja plana del <i>Emisor</i>	227
7.12	s-caja plana del <i>Medio</i>	228
7.13	s-caja plana del <i>Receptor</i>	228
7.14	s-caja plana del <i>ackMedio</i>	229

Capítulo 1

Introducción

Con esta tesis se ha pretendido realizar una extensión estocástica (*markoviana*) del Petri Box Calculus (PBC). Para poder entender este trabajo es preciso empezar introduciendo el modelo sobre el cual nos hemos basado: PBC [BDK01b].

PBC es un álgebra de procesos al estilo de CCS [Mil89], pero diseñado con el objetivo de poder disponer a partir de una expresión arbitraria del álgebra de procesos, una traslación fácil y natural a un modelo particular de redes de Petri. Para ello se eligieron los operadores de PBC de una forma especial, teniendo por ejemplo separado el operador paralelo del operador de sincronización. Como la estructura de una red de Petri no cambia a lo largo de su evolución y lo que sí cambia es su marcado, la semántica operacional de PBC se definió con esa misma filosofía: a través de *expresiones estáticas* (que corresponden a redes de Petri sin marcado, esto es, a la estructura de la red) y con *expresiones dinámicas*, que denotan la evolución del proceso (y corresponden a redes de Petri marcadas, con el marcado alcanzado). De esta forma, se tienen dos semánticas equivalentes para una misma sintaxis: una operacional y la otra denotacional, basada esta última en una clase especial de redes de Petri etiquetadas, llamadas *cajas (boxes)*. Con todo ello, se consiguen las ventajas de dos de los modelos más utilizados y estudiados en el diseño de sistemas concurrentes: las álgebras de procesos y las redes de Petri.

A la hora de plantearnos nuestra extensión del modelo base, teníamos en mente mantener la sintaxis del álgebra de procesos (ahora estocástica), y obtener su traducción natural a redes de Petri (ahora también estocásticas). Primero tuvimos que decidir qué modelo de red de Petri estocástica nos resultaba más interesante para nuestros propósitos, eligiendo el modelo de red de Petri estocástica (*markoviana*)

definido en [AMBC⁺95], por ser un modelo ampliamente aceptado, estudiado y utilizado. Estas redes de Petri estocásticas son redes de Petri en las que cada transición tiene asociado un parámetro, que corresponde a la razón de la distribución exponencial negativa asociada a dicha transición. Si una transición está sensibilizada, para que pueda ser ejecutada (disparada) debe sufrir un retraso (*delay*) gobernado por dicha distribución exponencial. Cuando varias transiciones están sensibilizadas a la vez se aplica una *política de competencia*: la más rápida ganará la carrera. El disparo de una transición no conlleva tiempo, y la regla de disparo es la misma que la empleada en el modelo base de redes de Petri. De esta forma, el tiempo de permanencia en cada marcado sigue estando gobernado por una distribución exponencial, y el proceso asociado a su evolución es una *Cadena de Markov de Tiempo Continuo* (CTMC). Como consecuencia de utilizar distribuciones de probabilidad continua, obtenemos que la probabilidad de que dos transiciones se realicen (disparen) a la vez es nula, lo que puede interpretarse como una semántica de concurrencia de *entremezcla* (*interleaving*).

Una vez que tuvimos claro el modelo semántico que queríamos, nos planteamos un álgebra de procesos (markoviana) con las mismas características: cada multiacción, desde que está activada, sufriría un retraso para su ejecución gobernada por una distribución exponencial negativa; asimismo, cuando es posible realizar varias multiacciones a la vez, se aplicará una política de competencia y de nuevo la ejecución de las multiacciones no conllevará ningún tiempo.

De esta forma, una multiacción estocástica de sPBC se denota en la forma $\langle \alpha, r \rangle$, donde α representa la multiacción (como en PBC) y r es el parámetro de su distribución exponencial asociada. Los operadores y la semántica operacional de PBC se intentaron conservar, lo que finalmente se consiguió, salvo en el caso del operador paralelo, ya que dada la naturaleza continua de la distribución exponencial negativa, no se pudo mantener la *conurrencia real* que tiene PBC. Esta es sin dudas la máxima diferencia entre PBC y nuestro modelo sPBC, y el precio que hemos pagado al realizar esta versión estocástica (markoviana). Además, hemos podido mantener la equivalencia entre las dos semánticas: la operacional y la denotacional basada en una clase especial de redes de Petri (*cajas* en PBC y *s-cajas* en sPBC).

El álgebra de procesos estocástica (markoviana) definida es del estilo de las clásicas SPAs (Stochastic Process Algebras) como PEPA [Hil96], EMPA [BG98] o TIPP [HR94]. Hay dos diferencias sustanciales con respecto a estas álgebras de procesos más clásicas: la primera es que en PBC se permiten multiacciones a un

primer nivel, y la segunda es que la sincronización se define de una forma diferente, y separada del paralelo.

A continuación, vamos a ir explicando cómo hemos ido definiendo nuestro modelo, los objetivos planteados, los problemas encontrados y las soluciones que les hemos dado.

El primer problema con que nos tropezamos surgió cuando definimos la sincronización. Nos basamos en la definición que se tiene de dicho operador en PBC plano: dos multiacciones pueden sincronizar sobre a si es posible ejecutarlas en concurrencia y, además una de ellas contiene la acción a , y la otra su correspondiente conjugado \hat{a} ; la nueva multiacción que se origina contiene todas las acciones de ambas multiacciones excepto el par (a, \hat{a}) . Sobre esta idea, la primera dificultad que nos apareció fue cómo detectar la *conurrencia real* en una semántica de *entremezcla*. Para ello, definimos las *bolsas concurrentes* [MVdF01a] sobre *s-expresiones dinámicas operativas*, que nos proporcionarán bolsas de multiacciones que podrían realizarse a la vez (si tuviéramos una *conurrencia real*) en una expresión dinámica donde se han activado todas sus componentes.

La siguiente meta a cubrir fue cómo elegir la razón de la multiacción generada por efecto de una sincronización. Nuestra primera propuesta [MVdF01a] consistió en tomar la menor de las razones involucradas en la sincronización, lo que intuitivamente se interpretaría como *esperarse a la más lenta*. En efecto, decimos intuitivamente, porque esperarse a la más lenta sería en realidad una espera gobernada por una nueva variable aleatoria definida como el máximo de dos variables aleatorias con distribución exponencial, y desafortunadamente, ésta no sigue una distribución exponencial, lo cual nos apartaría del modelo markoviano y complicaría enormemente el modelo.

Nuestro siguiente objetivo fue definir una equivalencia entre expresiones, considerando el criterio de que a nivel funcional dos expresiones equivalentes deben comportarse de la misma forma, y a nivel de prestaciones las CTMCs asociadas han de ser isomorfas. De esta forma, si consideramos:

$$E_1 = \langle \{a\}, 3 \rangle, \quad E_2 = \langle \{a\}, 1 \rangle \square \langle \{a\}, 2 \rangle$$

- E_1 puede interpretarse como un proceso que sufre un retraso para la ejecución de $\{a\}$ gobernado por una distribución exponencial de parámetro 3 (tiempo medio 1/3); esto es, la probabilidad de que E_1 termine su ejecución en un

tiempo menor o igual que t es $1 - e^{-3t}$. Este proceso ejecutará eventualmente el conjunto unitario $\{a\}$, y terminará.

- E_2 también ejecutaría eventualmente una multiacción $\{a\}$, y terminaría. No obstante, el tiempo que le llevaría a E_2 hacerlo seguiría (debido a la política de competencia) una variable aleatoria definida por el mínimo de dos variables aleatorias con distribución exponencial, que en esta ocasión sí es de nuevo una variable aleatoria con distribución exponencial, cuyo parámetro es la suma de los parámetros de ambas distribuciones. De esta forma, de acuerdo con el presente criterio, E_2 sufriría un retraso gobernado por una distribución exponencial de parámetro 3, ejecutaría $\{a\}$, y terminaría.

En consecuencia, E_1 y E_2 se podrían considerar equivalentes.

Una vez que tuvimos claro lo que queríamos reflejar con la equivalencia, nos quedaba definirla formalmente, y comprobar que se conservaba al aplicar los distintos operadores, esto es, que era una congruencia. Desgraciadamente, vimos rápidamente que la decisión de tomar el mínimo como razón de la nueva multiacción obtenida por sincronización, no servía. Para ello nos bastó con considerar el ejemplo siguiente:

$$E_3 = \langle \{\hat{a}\}, 2 \rangle, \quad E_4 = \langle \{\hat{a}\}, 1 \rangle \square \langle \{\hat{a}\}, 1 \rangle$$

E_3 y E_4 son equivalentes según nuestra interpretación, pero al tomar:

$$E_5 = (E_1 \parallel E_3) \text{sy } a, \quad E_6 = (E_2 \parallel E_4) \text{sy } a$$

obtenemos que aunque E_1 y E_2 son equivalentes, al igual que E_3 y E_4 , sin embargo E_5 y E_6 no lo son, ya que E_5 ofrece una única posibilidad de sincronización con razón 2, y E_6 ofrece cuatro posibilidades de sincronización, todas con razón 1.

Para resolver este problema se han definido en esta tesis las llamadas *razones de conflicto* [MVCP03], inspiradas en las *razones aparentes* de PEPA [Hil94a]. Básicamente, la idea consiste en considerar en conflicto dos multiacciones estocásticas en una s-expresión dinámica operativa si ambas están en el ámbito de una elección y además tienen la misma multiacción asociada. Definimos entonces la razón de conflicto de una multiacción como la suma de las razones de las multiacciones que aparecen en un conflicto. Apoyándonos en esta definición particular de razón de conflicto, definimos una nueva sincronización, eligiendo ahora el parámetro que aparece en la sincronización como el mínimo de las razones de conflicto, multiplicado por un factor que representa el peso que tiene cada razón dentro de su razón de

conflicto. Esta nueva propuesta va a proporcionar una fórmula para el cálculo de la razón cuando intervienen varias multiacciones en la sincronización, con la ventaja de que no va a importar el orden en que se sincronicen tales multiacciones estocásticas, teniéndose además que para cualquier bolsa concurrente, cualquier serialización de ella será posible, aunque para ello tendremos que restringirnos a los llamados *términos regulares*. Esta restricción nos obliga a limitar la sintaxis ligeramente, no permitiendo un paralelo dentro de una elección (al más alto nivel).

Para la primera propuesta de sPBC finito presentada en [MVdF01a] teníamos definida una semántica denotacional basada en una clase especial de redes de Petri estocásticas etiquetadas, llamadas *s-cajas*. Con esta nueva definición del operador de sincronización, se tuvo que acoplar la semántica denotacional de redes de Petri para que contemplase los conflictos [MVCP03].

Sin embargo, en nuestro camino para obtener la congruencia aparecieron nuevos problemas. Vimos que teníamos que distinguir entre aquellas expresiones que alcanzan su estado final (en terminología de redes de Petri, las que alcanzan el marcado final) y las que no lo alcanzan. También fue necesario diferenciar entre conductas paralelas y conductas en conflicto, las cuales se perdían al definir los sistemas de transiciones por medio de una semántica de *entremezcla*, por ejemplo, al tomar

$$E_7 = \langle \{a\}, 1 \rangle; \langle \{\hat{a}\}, 2 \rangle \square \langle \{\hat{a}\}, 2 \rangle; \langle \{a\}, 1 \rangle$$

$$E_8 = \langle \{a\}, 1 \rangle \parallel \langle \{\hat{a}\}, 2 \rangle$$

obtenemos que con la semántica de *entremezcla* ambos procesos tienen el mismo comportamiento (a nivel funcional y a nivel de prestaciones), pero si consideramos:

$$E_9 = E_7 \text{ sy } a$$

$$E_{10} = E_8 \text{ sy } a$$

se tiene que E_9 y E_{10} no se comportan de la misma forma, ya que desde E_{10} es posible sincronizar las dos multiacciones estocásticas, con lo que se podría ejecutar $\langle \emptyset, 1 \rangle$, mientras que en E_9 no se da esa posibilidad. Para poder contemplar la situación anterior se aumentó el sistema de transiciones considerando tres nuevas reglas (*Skp*, *Rdo* y *Fan*), se ajustó la definición de equivalencia, tras lo cual se probó que efectivamente se trataba de una congruencia [MVCdF03].

Finalmente, nuestro trabajo más reciente ha sido la incorporación del operador de iteración [MVCC03], y el desarrollo de algunos ejemplos clásicos de sistemas concurrentes, mediante los cuales quedaba reflejada la potencia y flexibilidad de nuestro modelo.

Como trabajo futuro inmediato tenemos previsto introducir en nuestro modelo el operador de recursión. Aunque en el modelo original [BDH92] esto se hace de una forma general dando lugar a redes infinitas, utilizando la teoría del punto fijo, nuestro primer objetivo es introducirlo de una forma restringida, con el fin de que las CTMCs generadas sean ergódicas. También tenemos en mente introducir en nuestro modelo las multiacciones inmediatas, al estilo de EMPA [BG98].

Este ha sido el orden cronológico en el que hemos ido realizando nuestro trabajo. En la redacción final de la tesis, hemos reordenado las ideas con el fin de hacerla más legible. El plan a seguir se describe a continuación. En el segundo capítulo presentamos los antecedentes, haciendo hincapié en las redes de Petri (clásicas y estocásticas), las álgebras de procesos (clásicas y estocásticas) y en el modelo original (PBC). En el tercer capítulo se presenta la primera versión de sPBC finito (sin iteración ni recursión), y con la primera propuesta para la sincronización. En el capítulo cuarto se introduce la nueva propuesta para la sincronización. A continuación, en el capítulo siguiente, se presenta el operador de iteración, siendo en el capítulo sexto donde se estudian las relaciones de equivalencias, dejando para el séptimo algunos casos de estudio, como el de la región crítica, el productor/consumidor o el protocolo del Bit Alternante. Por último, terminaremos con las conclusiones y trabajo futuro, que se encuentran en el capítulo octavo. También se incluye en ese último capítulo un apartado donde se presentan las publicaciones de la doctorando como primer autor, de trabajos relacionados con esta tesis.

Capítulo 2

Antecedentes

2.1 Modelos formales de concurrencia

El término *modelo* se puede utilizar para describir cualquier cosa que represente a un sistema, y que nos sirve para hacer predicciones sobre el comportamiento del mismo. Así, un modelo puede verse como una abstracción del sistema, y según el nivel de abstracción que presente frente al sistema real, las predicciones del modelo serán más o menos acertadas. Como norma general debemos descartar todos los detalles no relevantes, y centrarnos en aquellos aspectos que son esenciales para el comportamiento del sistema. Algunas de las razones que favorecen el estudio sobre modelos frente a la aproximación directa son que esta última suele ser más cara (en términos de recursos del sistema), consume mucho tiempo, a veces es difícil de extrapolar a otros escenarios de operación, e incluso puede que el sistema que queremos estudiar ni siquiera exista todavía. El diseño de un sistema debe estar sujeto a un cierto análisis que asegure que el sistema se comporta correctamente.

En un principio la concurrencia empezó a estudiarse en el campo del diseño de los sistemas operativos, donde aparecen ciertas situaciones donde, por ejemplo, es necesario proteger un recurso compartido de accesos concurrentes indiscriminados. Hoy en día, sus aplicaciones son múltiples lo que hace que sea un campo de intensa investigación. Dentro del marco de los sistemas concurrentes, las redes de Petri [GV03] y las álgebras de procesos [BPS01] son dos de los modelos formales mejor estudiados y más utilizados. A continuación, pasamos a describir brevemente ambos modelos, siguiendo el orden histórico de aparición. En esta primera parte consideraremos los dos modelos básicos sin ninguna consideración temporal y/o probabilística, las que

posteriormente se introducirán en los últimos apartados del capítulo.

2.1.1 Redes de Petri

Las redes de Petri fueron definidas en 1962 por Carl Adam Petri [Pet62]. Desde entonces, se han definido multitud de variantes de las redes, y estudiado su comportamiento desde muy diferentes puntos de vista. En [Net03] puede encontrarse actualizada toda la información referente al mundo de las redes de Petri.

Una red de Petri es un modelo de naturaleza gráfica (concretamente, es un grafo bipartito con dos clases de nodos: lugares y transiciones), aunque con un soporte formal preciso. Ambas características hacen de las redes de Petri un modelo muy útil en el diseño y análisis de sistemas concurrentes: por un lado, su simple representación gráfica permite disponer de modelos de sistemas concurrentes fácilmente legibles y, por otro lado, la descripción formal de las mismas permite analizar de forma sistemática ciertas propiedades del sistema. Hoy en día otro factor a tener en cuenta, es la disponibilidad de herramientas que dan soporte al diseño y análisis de sistemas con redes de Petri, como por ejemplo INA [Sta01].

En esta breve introducción a las Redes de Petri nos limitaremos a comentar algunas definiciones básicas, algunas propiedades a destacar y algunas indicaciones sobre cómo analizar esas propiedades. Un estudio mucho más completo y detallado puede encontrarse en [Sil88], [Bra86], [Pet81] ó [Rei88]. Para hacer esta presentación nos hemos basado en [Val90], [VB01] y [Col01].

Definición 2.1 (Redes de Petri Ordinarias)

Una terna $N = (P, T, F)$, formada por dos conjuntos P y T , y una relación F definida sobre $P \cup T$, se dice que es una *Red de Petri Ordinaria* si satisface:

1. $P \cap T = \emptyset$
2. $F \subseteq (P \times T) \cup (T \times P)$
3. $dom(F) \cup cod(F) = P \cup T$

Al conjunto P se le llama *conjunto de lugares*, al conjunto T *conjunto de transiciones* y a F *relación de flujo*. Gráficamente, los lugares se representan con círculos, mientras que las transiciones se representan con rectángulos o líneas. F relaciona lugares con transiciones por medio de arcos dirigidos entre lugares y transiciones, y

relaciona transiciones con lugares en forma de arcos dirigidos entre transiciones y lugares.

Siendo $X = P \cup T$, para cada $x \in X$ se definen los conjuntos siguientes:

$$\bullet x = \{y \in X \mid (y, x) \in F\} \quad (\text{preconjunto o precondición de } x),$$

$$x \bullet = \{y \in X \mid (x, y) \in F\} \quad (\text{postconjunto o postcondición de } x).$$

□

Definición 2.2 (Marcados de Redes de Petri Ordinarias)

Sea $N = (P, T, F)$ una Red de Petri Ordinaria. Una función $M : P \rightarrow \mathbb{N}$ se llama un *marcado de N* . Entonces, a toda la tupla (P, T, F, M) se le llama *Red de Petri Ordinaria Marcada*. □

Los marcados de las redes de Petri se representan gráficamente incluyendo en cada lugar tantos puntos como tokens le corresponden, o bien anotando cada lugar con el número de tokens asociado al mismo (ver figura 2.1). Dado que los marcados son funciones que toman valores en \mathbb{N} , se pueden representar como vectores de \mathbb{N}^n , siendo $n = |P|$; otras veces se representan en la forma $c_1p_1 + c_2p_2 + \dots + c_np_n$, donde c_i indica el número de tokens del lugar p_i , $i = 1, \dots, n$.

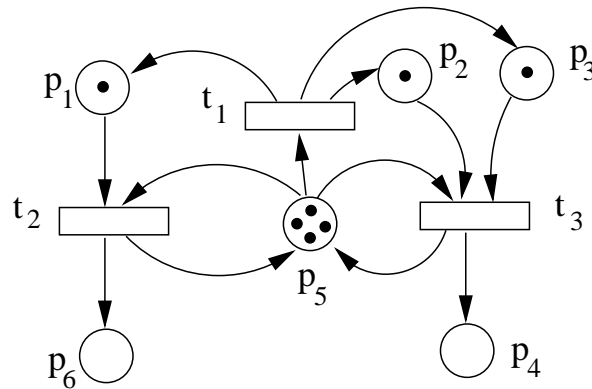


Figura 2.1: Ejemplo de red de Petri marcada

La semántica de una Red de Petri se define mediante la siguiente regla de disparo, que define el marcado alcanzado tras el disparo de una transición.

Definición 2.3 (Regla de Disparo)

Sea $N = (P, T, F, M)$ una Red de Petri Ordinaria Marcada. Una transición $t \in T$

se dice que *está permitida (activada o sensibilizada) bajo el marcado M* , lo que se denota por $M[t\rangle$, si para todo lugar $p \in P$ tal que $(p, t) \in F$, se verifica $M(p) > 0$.

El disparo de una transición t permitida bajo el marcado M da lugar a un nuevo marcado de la red, M' , definido en la forma siguiente:

$$M'(p) = M(p) - W_f(p, t) + W_f(t, p) \quad \forall p \in P$$

donde para cada $x \in (T \times P) \cup (P \times T)$:

$$W_f(x) = \begin{cases} 1 & \text{si } x \in F \\ 0 & \text{si } x \notin F \end{cases}$$

Ello se denota en la forma $M[t\rangle M'$. □

Definición 2.4 (Activación Concurrente de Transiciones)

Sea $N = (P, T, F, M)$ una Red de Petri Ordinaria Marcada, y $R \in \mathcal{B}(T)$ un multiconjunto de transiciones de la red. Se dice que R está permitido bajo M , lo que se denota por $M[R\rangle$, si y sólo si $M(p) \geq \sum_{t \in T} W_f(p, t) \cdot R(t)$, $\forall p \in P$.

El disparo de un multiconjunto de transiciones R bajo el marcado M conduce la red a un nuevo marcado M' , dado por:

$$M'(p) = M(p) - \sum_{t \in T} (W_f(p, t) - W_f(t, p)) \cdot R(t)$$

Esta evolución de la red en un solo paso la denotaremos por $M[R\rangle M'$. □

Definición 2.5 (Redes de Petri Generalizadas)

Una Red de Petri Generalizada es una tupla $N = (P, T, F, K, W)$, donde:

1. (P, T, F) es una Red de Petri Ordinaria.
2. $K : P \longrightarrow \mathbb{N} \cup \{\infty\}$ es una función que indica el número máximo de tokens que puede haber en cada lugar (*función de capacidad*).
3. $W : F \longrightarrow \mathbb{N}^+$ es una función que indica la multiplicidad de cada arco (*pesos de los arcos*).

La función de capacidad se suele omitir cuando ésta es infinita para todos los lugares. Además, habitualmente extenderemos la definición de W a todo el universo de posibles arcos, haciéndola nula para los pares (p, t) ó (t, p) que no estén en F . \square

Algunas clases especiales de redes que nos van a interesar son las siguientes :

Red pura: En ellas se impone la restricción de que no hay lugares que son simultáneamente precondition y postcondición de alguna transición:

$$\forall t \in T, \forall p \in P : W(t, p) \cdot W(p, t) = 0$$

Redes T-restrictivas: Son aquéllas para las cuales todas las transiciones deben tener al menos un lugar precondition y un lugar postcondición:

$$\forall t \in T : \bullet t \neq \emptyset, t \bullet \neq \emptyset$$

Definición 2.6 (Regla de Disparo para Redes de Petri Generalizadas)

Sea $N = (P, T, F, K, W)$ una Red de Petri Generalizada.

1. Una función $M : P \longrightarrow \mathbb{N}$ se dice que es un *marcado* de N si y sólo si $M(p) \leq K(p)$, para todo $p \in P$.
2. Un multiconjunto de transiciones R está permitido bajo un marcado M , lo que se denota por $M[R\rangle$, si y sólo si para todo $p \in P$ se cumplen las dos condiciones siguientes :

$$(a) \quad M(p) \geq \sum_{t \in T} W(p, t) \cdot R(t)$$

$$(b) \quad M(p) - \sum_{t \in T} (W(p, t) - W(t, p)) \cdot R(t) \leq K(p)$$

El disparo de R genera un nuevo marcado M' , dado por:

$$M'(p) = M(p) - \sum_{t \in T} (W(p, t) - W(t, p)) \cdot R(t), \quad \forall p \in P$$

lo que se denota por $M[R\rangle M'$. \square

Definición 2.7 Sea $N = (P, T, F, K, W, M_0)$ una Red de Petri Marcada.

1. Se dice que $\sigma = M_0 t_1 M_1 \dots t_n M_n$ es una secuencia de ocurrencia finita de N si $\forall i \in \{1, \dots, n\}$, $M_{i-1} [t_i] M_i$. Observemos que dado el marcado inicial M_0 , y la secuencia de transiciones $t_1 \dots t_n$, es posible reconstruir la secuencia de ocurrencia completa. Por ello, en ocasiones éstas se denotan simplemente en la forma $M_0[\sigma]M_n$, siendo $\sigma = t_1 \dots t_n$.

Asimismo, se denota por $[M_0]$ al conjunto de marcados que pueden ser alcanzados a partir de M_0 .

2. Se dice que $\sigma = M_0 R_1 M_1 \dots R_n M_n$ es una secuencia de pasos finita de N si $\forall i \in \{1, \dots, n\}$, $M_{i-1} [R_i] M_i$. La notación habitual se extiende de nuevo a secuencias de pasos: $M_0[\sigma]M_n$. □

Ejemplo 2.1 Para la red mostrada en la figura 2.1, tenemos que para el marcado inicial $M_0 = p_1 + p_2 + p_3 + 4p_5$, el multiconjunto de transiciones $R = \{2.t_1, 1.t_2, 1.t_3\}$ está permitido, y su disparo genera el marcado $M' = 2p_1 + 2p_2 + 2p_3 + p_4 + 2p_5 + p_6$. □

La verificación de propiedades sobre redes de Petri tiene un doble objetivo, por un lado el poder demostrar que el diseño realizado es correcto, es decir, que se cumplen las especificaciones de las que se partió, y por otro poder extraer propiedades sobre el comportamiento del sistema.

Las propiedades objeto de interés se relacionan con la conducta del sistema modelado, aunque básicamente pueden englobarse dentro de las siguientes categorías:

1. *Alcanzabilidad*: Un marcado M es alcanzable si existe una secuencia de ocurrencia σ tal que $M_0[\sigma]M$.
2. *Seguridad*: Una red de Petri N es n -segura si $\forall M \in [M_0]$, se cumple que $M(p) \leq n$, $\forall p \in P$. Un caso especial son las redes 1-seguras, en las cuales cada lugar sólo puede contener a lo sumo un token. Sobre este tipo de redes es posible abordar con éxito la verificación de muchas de las propiedades sobre redes de Petri, por lo que, son muy utilizadas en el modelado de sistemas, aunque por la simplicidad del modelo se paga el coste de tener que generar redes muy voluminosas, incluso para sistemas pequeños.

3. *Acotación*: Se dice que un lugar p es acotado si existe un $n_p \in \mathbb{N}$ tal que $M(p) \leq n_p, \forall M \in [M_0]$. Cuando todos los lugares de una red son acotados se dice que ésta es acotada o limitada.
4. *Vivacidad*: Se dice que una transición t es viva si para todo marcado alcanzable $M \in [M_0]$ existe otro marcado $M' \in [M]$ tal que t está permitida bajo M : $M[t]$. Si todas las transiciones de N son vivas, se dice que N es viva.

La vivacidad de una transición garantiza que con independencia del estado actual que el sistema haya alcanzado, éste siempre puede evolucionar hacia un estado que permite disparar la transición.

5. *Ausencia de bloqueos*: Diremos que una red de Petri está libre de bloqueos si ningún marcado alcanzable es un marcado *muerto*. Un marcado M se dice que está *muerto* si no permite disparar ninguna transición.

Esta propiedad nos permitirá comprobar que el sistema nunca llega a una situación de bloqueo (*deadlock*) en la que no puede evolucionar de ninguna forma, ni siquiera de forma parcial, por parte de alguna de sus componentes.

6. *Reversibilidad*: Una red de Petri se dice que es reversible si para cualquier marcado alcanzable $M \in [M_0]$ existe una secuencia de ocurrencia σ tal que $M[\sigma]M_0$.

Esta propiedad nos permite comprobar que con independencia del estado actual alcanzado por el sistema, siempre podemos volver a la situación inicial.

Las técnicas de análisis de redes de Petri se clasifican en: Enumeración, Transformación y Análisis estructural.

1. *Enumeración*. Se requiere la construcción del *grafo de alcanzabilidad* (RG) que representa los marcados de la red y la ocurrencia de transiciones entre ellos. Si la red es limitada, el grafo de alcanzabilidad es finito y se pueden verificar las diferentes propiedades de interés sobre él.
2. *Transformación*. Consisten en transformar la red (N, M_0) en una red (N', M'_0) donde el conjunto de propiedades Π a verificar se conserven, de manera que (N', M'_0) satisface las propiedades Π si y sólo si (N, M_0) las satisface. El objetivo es verificar las propiedades en (N', M'_0) asumiendo que será más

fácil que hacerlo en (N, M_0) . Puede que el espacio de estados (marcados alcanzables) de (N', M'_0) sea más grande que el de (N, M_0) , pero que (N', M'_0) pertenezca a una subclase de redes de Petri para la que se puedan analizar las propiedades sin tener que enumerar los estados. Pero si hacemos transformaciones donde cada vez el espacio de estados es más pequeño, tendríamos lo que se conoce como *métodos de reducción*.

3. *Análisis estructural*. Investigan las relaciones entre el comportamiento de una red y su estructura, independientemente del marcado inicial.

A continuación, pasamos a describir brevemente la verificación basada en el grafo de alcanzabilidad, ya que es a la que haremos referencia en esta tesis.

Definición 2.8 Dada una red, $\mathcal{S} = (N, M_0)$, su grafo de alcanzabilidad es un grafo dirigido, $RG(\mathcal{S}) = (V, E)$, donde:

- $V = \{M \mid M \in [M_0]\}$ (conjunto de marcados alcanzables), que se suele representar en la forma $V = RS(\mathcal{S})$.
- $E = \{(M, t, M') \mid M, M' \in V \text{ y } M[t]M'\}$ (conjunto de arcos).

□

Si $\mathcal{S} = (N, M_0)$ es limitada, $RG(\mathcal{S})$ es finito y se puede construir, por ejemplo, mediante el algoritmo descrito en [Col01]. Para verificar una propiedad sobre una red de este tipo, se necesita un *procedimiento de decisión* que usa el grafo de alcanzabilidad para determinar si la red cumple o no la propiedad. A continuación, definimos lo que se conoce como *predicado de marcado*, que será la forma en que enunciaremos las propiedades a verificar.

Definición 2.9 Se llamará *predicado de marcado* a una proposición lógica cuyos átomos son desigualdades de la forma:

$$\sum_{p \in A} k_p M(p) \leq k$$

donde A es un subconjunto de lugares, y k_p y k son constantes racionales. □

Veamos ahora cómo expresar mediante un predicado de marcado las propiedades expuestas anteriormente.

Cuando una propiedad se expresa como un predicado de marcado, que debe ser satisfecho por todos los marcados alcanzables, se dice que es una *propiedad invariante de marcado*. Ejemplos de propiedades de este tipo son:

- *k-limitación de un lugar p*: $\forall M \in RS(\mathcal{S}), M(p) \leq k$.
- *Exclusión mútua de marcado entre p y p'*: $\forall M \in RS(\mathcal{S}), M(p) = 0 \vee M(p') = 0$.
- *Ausencia de bloqueos*: $\forall M \in RS(\mathcal{S}), \exists t \in T : M[t]$.

También se puede hablar de *propiedades invariantes de vivacidad*: para cada marcado alcanzable, existe por lo menos un marcado alcanzable desde él que verifica la propiedad. Ejemplos de este tipo de propiedades son:

- *Vivacidad de t*: $\forall M \in RS(\mathcal{S}), \exists M' \in RS(\mathcal{S}')$ con $\mathcal{S}' = (N, M)$ tal que $M'[t]$.
- *M_H estado de acogida*: $\forall M \in RS(\mathcal{S}), \exists M' \in RS(\mathcal{S}')$ con $\mathcal{S}' = (N, M)$ tal que $M' = M_H$.
- *Reversibilidad*: $\forall M \in RS(\mathcal{S}), \exists M' \in RS(\mathcal{S}')$ con $\mathcal{S}' = (N, M)$ tal que $M' = M_0$ (marcado inicial).

En [Col01] se pueden encontrar algoritmos de verificación de propiedades, tanto para los invariantes de marcado, como para los invariantes de vivacidad.

2.1.2 Álgebras de Procesos. CCS.

Las álgebras de procesos son modelos formales algebraicos que aparecen en la década de los 80, siendo dos de sus máximos exponentes CSP [Hoa85] y CCS [Mil80, Mil89]. Desde entonces, se han definido multitud de variantes y estudiado su comportamiento desde muy diferentes puntos de vista. Son un mecanismo muy apropiado para la especificación formal de sistemas concurrentes y cabe destacar sobre todo su composicionalidad.

El modelo algebraico Petri Box Calculus se basa en el álgebra de procesos CCS, por lo que es la que pasamos a comentar a continuación. Para esta breve introducción nos hemos basado en [Cua93] para la descripción de los operadores, y en [Núñ01] para las equivalencias. Otros modelos algebraicos y un estudio más completo y

detallado sobre las álgebras de procesos se pueden encontrar, por ejemplo, en [BW90] y [BPS01].

Como para todo lenguaje formal, de un álgebra de procesos hay que definir su sintaxis y su(s) semántica(s). En CCS se considera un conjunto de acciones atómicas Λ , que puede ser partido en dos subconjuntos, los cuales pueden ser aplicados biyectivamente el uno sobre el otro, por medio de una función $(\bar{\cdot})$; así, si la acción $a \in \Lambda$ entonces su correspondiente *conjugada* $\bar{a} \in \Lambda$. A partir de este conjunto, se define el conjunto *Act* en la forma siguiente: $Act = \Lambda \cup \{\tau\}$, siendo τ una acción distinguida, que representará la ejecución no visible de una acción y recibe el nombre de *acción interna*. Dicha acción aparecerá explícitamente en los cómputos, aunque se asume que su ejecución no es observable directamente desde el exterior.

La sintaxis de CCS en notación BNF es la siguiente:

$$P ::= NIL \mid a.P \mid P + P \mid P|P \mid P \setminus a \mid \mu X.P$$

A continuación, hacemos una descripción intuitiva de cada uno de los operadores del lenguaje:

NIL: Representa una máquina parada, incapaz de realizar ninguna acción (ni siquiera interna).

PREFIJO ($a.P$): Dada una acción $a \in Act$ y un proceso P , $a.P$ representa el proceso que puede ejecutar la acción a , y después se comporta como el proceso P .

ELECCIÓN ($+$): Dados dos procesos P, Q , el proceso $P + Q$ representa una elección entre ambos, con lo que el proceso se comporta como P ó como Q , dependiendo de cual de ellos pueda ejecutar la acción ofrecida (o bien la acción interna). Cuando las dos pueden ejecutar la acción visible ofrecida, entonces el sistema elige arbitrariamente cuál de ellos la ejecuta, siguiéndose con el resto de la ejecución del mismo.

COMPOSICIÓN PARALELA ($|$): Dados dos procesos P, Q , el proceso $P|Q$ representa su composición en paralelo. Con la composición, cada uno de ellos puede ejecutar sus acciones de forma independiente del otro, o ambos pueden sincronizar sobre acciones complementarias (vía la función $\bar{\cdot}$), lo que resulta en una única acción interna τ . En consecuencia, el operador paralelo combina entremezcla (*interleaving*) con sincronización, sin imponer ésta jamás. A diferencia de la sincronización de

CSP [Hoa85], que de manera natural permite, e incluso exige, sincronizar simultáneamente varios procesos por medio de una única acción, en CCS la sincronización es intrínsecamente binaria.

RESTRICCIÓN ($P \setminus a$): Dada una acción $a \in \Lambda$ y un proceso P , el proceso $P \setminus a$ se comporta como P , si bien la ejecución de la acción a ni la de \bar{a} están ya permitidas. Su aplicación tras un operador paralelo nos servirá para imponer sincronización sobre las acciones suprimidas.

RECUSIÓN ($\mu X.P$): Representa un proceso obtenido como solución de la ecuación recursiva $X = P$.

La semántica operacional de CCS es la definida mediante las reglas de la tabla 2.1.

1) $\frac{}{a.P \xrightarrow{a} P}$	2) $\frac{P \xrightarrow{a} P'}{P + Q \xrightarrow{a} P'} \quad a \in Act$
3) $\frac{Q \xrightarrow{a} Q'}{P + Q \xrightarrow{a} Q'} \quad a \in Act$	4) $\frac{P \xrightarrow{a} P'}{P Q \xrightarrow{a} P' Q} \quad a \in Act$
5) $\frac{Q \xrightarrow{a} Q'}{P Q \xrightarrow{a} P Q'} \quad a \in Act$	6) $\frac{P \xrightarrow{a} P', Q \xrightarrow{\bar{a}} Q'}{P Q \xrightarrow{\tau} P' Q'} \quad a \in \Lambda$
7) $\frac{P \xrightarrow{a} P'}{P \setminus L \xrightarrow{a} P' \setminus L} \quad a, \bar{a} \notin L$	8) $\frac{P\{\mu X.P/X\} \xrightarrow{a} P'}{\mu X.P \xrightarrow{a} P'} \quad a \in Act$

Tabla 2.1: Semántica operacional de CCS

Interesa tener una relación de equivalencia que permita identificar aquellos procesos que se comporten de la misma forma. Cuando la equivalencia entre dos procesos se refiera a que el primer proceso pueda simular de una cierta forma el comportamiento del segundo, y viceversa, se hablaría de una relación de *bisimulación*.

Definición 2.10 Una relación binaria \mathcal{R} definida sobre el conjunto de procesos es una *bisimulación fuerte* si para todo par de procesos P y Q se cumple $(P, Q) \in \mathcal{R}$ implica que para toda acción $a \in Act$ se tiene:

- $P \xrightarrow{a} P'$ implica que existe un proceso Q' tal que $Q \xrightarrow{a} Q'$ y $(P', Q') \in \mathcal{R}$.
- $Q \xrightarrow{a} Q'$ implica que existe un proceso P' tal que $P \xrightarrow{a} P'$ y $(P', Q') \in \mathcal{R}$.

Dos procesos P y Q se dicen que son fuertemente bisimilares, si existe una bisimulación fuerte \mathcal{R} tal que $(P, Q) \in \mathcal{R}$. \square

Ejemplo 2.2 Si consideramos los procesos:

$$\begin{aligned} P_1 &= a.b.NIL \\ P_2 &= a.NIL + b.NIL \end{aligned}$$

se tiene que no son fuertemente bisimilares, ya que:

$$\begin{aligned} P_1 &\xrightarrow{a} b.NIL \\ P_2 &\xrightarrow{a} NIL \end{aligned}$$

y, obviamente $b.NIL$ y NIL no son equivalentes, pues el primero puede hacer la acción b , mientras que el segundo no.

Sin embargo, se puede comprobar fácilmente que:

$$\begin{aligned} P_3 &= a.NIL \mid b.NIL \\ P_4 &= a.b.NIL + b.a.NIL \end{aligned}$$

sí son fuertemente bisimilares. \square

Esta bisimulación fuerte es una congruencia, esto es, dados dos procesos P y Q fuertemente bisimilares se pueden intercambiar en cualquier ámbito, o lo que es lo mismo, también serían bisimilares fuertes: $a.P$ y $a.Q$, $P + R$ y $Q + R$, $R + P$ y $R + Q$, $P \mid R$ y $Q \mid R$, $R \mid P$ y $R \mid Q$, $P \setminus a$ y $Q \setminus a$, y $\mu X.P$ y $\mu X.Q$, $\forall a \in Act$ y R un proceso cualquiera.

Desde el punto de vista de un observador externo los procesos:

$$\begin{aligned} P_5 &= \tau.a.NIL \\ P_6 &= a.NIL \end{aligned}$$

se comportan de la misma forma (realizan la acción a y terminan). Sin embargo, su equivalencia no está contemplada con la bisimulación fuerte, siendo necesario introducir una nueva relación que se conoce como *bisimulación débil*. Antes de dar paso a la definición de esta nueva equivalencia necesitamos una previa.

Definición 2.11 Sean P y Q dos procesos, diremos que P puede evolucionar de forma interna a Q , y lo denotaremos por $P \xrightarrow{\tau} Q$, si existen $n \geq 0$ y procesos P_1, P_2, \dots, P_n tales que:

$$P \xrightarrow{\tau} P_1 \xrightarrow{\tau} P_2 \xrightarrow{\tau} \dots P_n \xrightarrow{\tau} Q$$

Diremos que P puede evolucionar mediante una transición externa generalizada etiquetada con $a \in \Lambda$ al proceso Q , y lo denotaremos por $P \xrightarrow{a} Q$, si existen procesos P_1 y P_2 tales que:

$$P \xrightarrow{\tau} P_1 \xrightarrow{a} P_2 \xrightarrow{\tau} Q$$

□

Definición 2.12 Una relación binaria \mathcal{R} definida sobre el conjunto de procesos es una *bisimulación débil* si para todo par de procesos P y Q se cumple que $(P, Q) \in \mathcal{R}$ implica que para toda acción $a \in Act$ se tiene:

- $P \xrightarrow{a} P'$ implica que existe un proceso Q' tal que $Q \xrightarrow{a} Q'$ y $(P', Q') \in \mathcal{R}$.
- $Q \xrightarrow{a} Q'$ implica que existe un proceso P' tal que $P \xrightarrow{a} P'$ y $(P', Q') \in \mathcal{R}$.

Dos procesos P y Q se dicen que son débilmente bisimilares, si existe una bisimulación débil \mathcal{R} tal que $(P, Q) \in \mathcal{R}$. □

La bisimulación débil es una congruencia salvo para el operador de elección.

Dos procesos P y Q que son fuertemente bisimilares también son débilmente bisimilares.

2.1.3 Petri Box Calculus

El Petri Box Calculus (PBC) fue introducido por primera vez a principios de los 90 por Best, Devillers y Hall [BDH92]. Su principal objetivo era juntar en el mismo modelo las ventajas de las álgebras de procesos y de las redes de Petri. Aunque la idea de asociar redes de Petri con expresiones de álgebras de procesos no era nueva [DdNM88, Gol88, Tau89, GL91, Old91, BC91], con PBC se pretendía tener una traslación 'natural' de una expresión de un álgebra de procesos a una clase especial de redes de Petri con una semántica de concurrencia real. Para ello, los operadores del álgebra se eligieron de una forma particular, basándose en CCS [Mil89], pero dejando separado el operador paralelo del operador de sincronización, además de permitir multiacciones (multiconjuntos de acciones) en vez de acciones para el operador base. Best, Devillers, Koutny y Esparza han seguido trabajando en el formalismo, generalizándolo con el álgebra de cajas ó álgebra de redes de Petri (*Petri Net Algebra, PNA*) [BDK98, KB99, BDK01a, BDK01b, BDK02], del que PBC [BDE93, KEB94, BK95] es un caso particular. En todos los casos se proporciona, para una misma sintaxis,

una semántica operacional y una denotacional, esta última basada en una clase especial de redes de Petri etiquetadas llamadas *cajas*. La semántica de redes de Petri es composicional y se obtiene combinando refinamiento y reetiquetación, a partir de las *cajas operador* y de las *cajas planas*. Con PNA se estudia cómo deben ser estas cajas operador para obtener la equivalencia entre las dos semánticas, y en este marco general, además de PBC, otras álgebras de procesos como CCS [Mil89], CSP [Hoa85] o COSY (CONcurrent SYstem notation, [JL92]) pueden considerarse como casos particulares [BDK01b].

Así otro de los objetivos de PBC era proporcionar una semántica composicional con redes de Petri (seguras) de lenguajes de programación concurrentes. Con este fin se diseñó el lenguaje $B(PN)^2$ (a Basic Petri Net Programming Notation, [BH93]), sobre el que se han ido introduciendo nuevas mejoras [Mel94, BFF⁺95, FG97] y una herramienta PEP (Programming Enviroment based on Petri nets, [BF94, Gra95, GB96, Gra97a, Gra97b, Gra98]), con una verificación basada en [McM92, Esp93, ERV95, Bes96] y con interfaces a otras herramientas [GMA95, FG98, Sta01]. Una breve introducción al $B(PN)^2$ y al PEP puede encontrarse en [Mac00, MV01].

En la literatura se pueden encontrar dos extensiones temporales de PBC [Kou00, MdF01], y la extensión estocástica sPBC [Mvdf01a, MVCP03, MvCdF03] que se presenta en esta tesis.

A continuación, pasamos a introducir brevemente el modelo Petri Box Calculus. Esta parte del trabajo está basada fundamentalmente en [BK95], [BDK98], [Mar98] y [Mac99].

Sintaxis

Consideraremos la notación siguiente:

- \mathcal{A} como un conjunto numerable de nombres de acciones. Además, para todo $a \in \mathcal{A}$, existe su *conjugado* (que aquí se denota con \hat{a}) $\hat{a} \in \mathcal{A}$, tal que $a \neq \hat{a}$ y $\hat{\hat{a}} = a$, como en CCS. Utilizaremos las letras a, b, \hat{a}, \dots para denotar los elementos de \mathcal{A} .
- $\mathcal{L} = \mathcal{B}(\mathcal{A})$, donde $\mathcal{B}(\mathcal{A})$ representa el conjunto de multiconjuntos finitos de \mathcal{A} . Si $\alpha \in \mathcal{L}$, $\alpha(a)$ indicará el número de veces que aparece a en α .

En particular, tendremos que $\emptyset \in \mathcal{L}$, que se utiliza para especificar la eje-

cución de una multiacción que no contiene acciones visibles. La llamaremos multiacción no observable, o también siguiendo la terminología de CCS, silenciosa o interna.

- El *alfabeto* de una multiacción $\alpha \in \mathcal{L}$ se define como:

$$A(\alpha) =_{def} \{a \in \mathcal{A} \mid \alpha(a) > 0\}$$

- Funciones de renombramiento $f : \mathcal{A} \rightarrow \mathcal{A}$, que conservan los conjugados:

$$\forall a \in \mathcal{A}, f(\widehat{a}) = \widehat{f(a)}$$

A continuación, introducimos algunas operaciones sobre multiacciones:

- **suma**

$$\alpha + \beta =_{def} \delta, \text{ donde } \delta(a) =_{def} \alpha(a) + \beta(a)$$

- **sincronización**

$$\alpha \oplus_a \beta =_{def} \gamma \text{ donde:}$$

$$\gamma(b) =_{def} \begin{cases} \alpha(b) + \beta(b) - 1 & \text{si } b = a \text{ ó } b = \widehat{a} \\ \alpha(b) + \beta(b) & \text{en otro caso} \end{cases}$$

y sólo es aplicable si $a \in A(\alpha)$ y $\widehat{a} \in A(\beta)$, o viceversa.

- **renombramiento**

$$f(\alpha) =_{def} \sum_{a \in \mathcal{A}} \alpha(a) f(a)$$

Expresiones Estáticas

Las *expresiones estáticas* se utilizan para describir la estructura de un sistema concurrente, y corresponderán a redes de Petri no marcadas. Una *expresión estática* de PBC, es un término generado por la expresión BNF siguiente:

$$E ::= \alpha \mid E; E \mid E \square E \mid E[f] \mid E \parallel E \mid E \text{ sy } a \mid E \text{ rs } a \mid [a : E] \mid [E * E * E] \mid X$$

A continuación, hacemos una descripción intuitiva de cada uno de los operadores del lenguaje.

MULTIACCIÓN (α): Una expresión estática de la forma α se llama *multiacción* ó *acción básica*; representa la ejecución simultánea de todas las acciones que aparecen

en α . Un caso especial es \emptyset , que representa un movimiento silencioso sin acciones visibles.

SECUENCIA (;): Dadas dos expresiones E y F , la expresión $E;F$ representa el proceso que primero realiza el proceso E , y luego pasa a ejecutar F .

ELECCIÓN (\square): La expresión $E\square F$ representa la elección entre dos procesos. Por ejemplo,

$$(\{a\}; \{b, c\})\square \{d\}$$

representa a un proceso que primero puede hacer el movimiento $\{a\}$, seguido de un movimiento $\{b, c\}$ y acabar; o bien, un solo movimiento $\{d\}$ y terminar.

RENOMBRAMIENTO ($[f]$): El proceso $E[f]$ puede realizar todos los movimientos que podía hacer E , pero renombrados por la función f .

PARALELO (\parallel): El proceso $E\parallel F$ representa la ejecución independiente de los dos procesos en paralelo, pudiéndose ejecutar sus pasos a la vez o entremezclados. Así,

$$\{a\} \parallel \{b\}$$

puede realizar un movimiento $\{a\}$ seguido de un movimiento $\{b\}$, y terminar; o bien, un movimiento $\{b\}$ seguido de un movimiento $\{a\}$, y terminar; pero también puede realizar el movimiento que consiste en la ejecución concurrente de $\{a\}$ y $\{b\}$, para terminar de inmediato.

SINCRONIZACIÓN (sy): $E sy a$ (con $a \in \mathcal{A}$) puede realizar los mismos movimientos que puede hacer E , y además los resultantes de la sincronización sobre un par de acciones (a, \hat{a}) . Así,

$$(\{a\} \parallel \{\hat{a}\}) sy a$$

puede realizar todos los movimientos de $\{a\} \parallel \{\hat{a}\}$, y también un movimiento vacío (\emptyset) que corresponde a la sincronización de sus dos movimientos atómicos. De igual forma,

$$(\{a\} \parallel \{\hat{a}, \hat{a}\} \parallel \{a\}) sy a$$

puede realizar los movimientos de $\{a\} \parallel \{\hat{a}, \hat{a}\} \parallel \{a\}$, y además una serie de sincronizaciones; por ejemplo, podría hacer un movimiento $\{\hat{a}\}$ (que correspondería a sincronizar las dos primeras componentes, o la segunda con la tercera) en paralelo con $\{a\}$, o bien un solo movimiento \emptyset (que correspondería a sincronizar las tres componentes).

RESTRICCIÓN (*rs*): Dado $a \in \mathcal{A}$, $E \text{ rs } a$ representa un proceso que puede ejecutar los movimientos de E , siempre y cuando en el alfabeto de la multiacción correspondiente no aparezca a ni \hat{a} . Por ejemplo,

$$(\{a\} \parallel \{b\}) \text{ rs } b$$

puede hacer un movimiento $\{a\}$, pero no puede hacer un movimiento $\{b\}$, ni terminar.

SCOPING ($[a :]$): La expresión $[a : E]$ es una expresión derivada que se define como sigue:

$$[a : E] = (E \text{ sy } a) \text{ rs } a$$

Se utiliza para obligar a sincronizar, y se emplea asiduamente en el modelado de sistemas concurrentes.

ITERACIÓN ($[**]$): La expresión $[E_1 * E_2 * E_3]$ representa un proceso que empieza con la ejecución (una sola vez) de la expresión E_1 , seguida de un número arbitrario de veces (en particular puede que ninguna) de E_2 , para terminar con la ejecución (una sola vez) de E_3 , para ya así concluir el proceso. De nuevo, resulta muy útil, en este caso para modelar sistemas concurrentes con conductas repetitivas.

RECURSIÓN (X): X es la variable utilizada en la recursión; aparecerá en la parte izquierda de la ecuación, así como en la parte derecha, en posiciones donde se permitiese una multiacción. Con la recursión se permiten comportamientos infinitos, por ejemplo:

$$X = \{a\}; X$$

define un proceso que puede realizar infinitos movimientos $\{a\}$, y no terminaría nunca. En cambio, el proceso:

$$X = \{a\} \parallel X$$

admite una secuencia finita o infinita de un número finito de movimientos $\{a\}$ concurrentes, y tampoco terminaría nunca.

La recursión de PBC [BDE93, BDK01b] requiere un tratamiento especial, bastante sofisticado, que queda fuera del alcance de esta tesis, por lo que no pasaremos a estudiar su semántica operacional ni denotacional. Sin embargo, es uno de los objetivos a abordar como trabajo futuro, para extender sPBC con este operador.

Expresiones Dinámicas

La semántica operacional de PBC se define a partir de las *expresiones dinámicas*, las cuales se derivan de las expresiones estáticas, añadiendo en ellas barrados y sub-barrados que indicarán las componentes activas, es decir, el estado actual del sistema, y corresponderán a redes de Petri marcadas.

De esta forma, las *expresiones dinámicas* de PBC (sin considerar el operador de recursión) son las definidas de acuerdo a la notación BNF siguiente:

$$G ::= \overline{E} \mid \underline{E} \mid G; E \mid E; G \mid G \square E \mid E \square G \mid G[f] \mid G \parallel G \mid G \text{ sy } a \mid \\ G \text{ r s } a \mid [a : G] \mid [G * E * E] \mid [E * G * E] \mid [E * E * G]$$

donde \overline{E} representa el estado inicial de E , en el que todavía no se ha ejecutado ningún movimiento, pudiendo empezar a ejecutarlos a partir de ahora; mientras que \underline{E} representa el estado final de E , que ya habrá terminado de ejecutar sus multiacciones y no puede realizar más movimientos. Obsérvese que en el caso del operador de secuencia sólo una de las componentes puede ser dinámica, al igual que sucede con el operador de elección, mientras que para el paralelo, ambas componentes tienen que ser dinámicas.

Semántica Operacional

En primer lugar se introducen las transiciones de inacción, las cuales nos permiten establecer las componentes activas de una expresión dinámica. Éstas se pueden aplicar las veces que sea preciso, para finalmente proporcionar expresiones dinámicas en las cuales queda perfectamente especificado (con los barrados) qué multiacciones se pueden realizar. Las reglas de inacción tienen el formato:

$$G \xrightarrow{\emptyset} G'$$

Por ejemplo, una regla de inacción para el operador secuencia es:

$$\overline{E}; \overline{F} \xrightarrow{\emptyset} \overline{E}; F$$

la cual indica que para realizar una secuencia, hay que empezar realizando la primera de las componentes. Otra regla de inacción para la secuencia es:

$$\underline{E}; F \xrightarrow{\emptyset} E; \overline{F}$$

la cual indica ahora que cuando ya ha terminado la primera de las componentes, puede empezar a ejecutarse la segunda. Nos quedaría una última regla de inacción para la secuencia:

$$E; \underline{F} \xrightarrow{\emptyset} E; \underline{F}$$

que indica que si la segunda componente ha terminado, entonces el proceso completo también.

El conjunto completo de reglas de inacción puede encontrarse en las tablas 2.2 y 2.3.

$\overline{E}; \overline{F} \xrightarrow{\emptyset} \overline{E}; F$	$\underline{E}; F \xrightarrow{\emptyset} E; \overline{F}$
$E; \underline{F} \xrightarrow{\emptyset} E; \underline{F}$	
$\overline{E \square F} \xrightarrow{\emptyset} \overline{E \square F}$	$\overline{E \square F} \xrightarrow{\emptyset} E \square \overline{F}$
$\underline{E \square F} \xrightarrow{\emptyset} \underline{E \square F}$	$E \square \underline{F} \xrightarrow{\emptyset} E \square \underline{F}$
$\overline{E \parallel F} \xrightarrow{\emptyset} \overline{E \parallel F}$	$\underline{E \parallel F} \xrightarrow{\emptyset} \underline{E \parallel F}$
$\overline{E[f]} \xrightarrow{\emptyset} \overline{E[f]}$	$\underline{E[f]} \xrightarrow{\emptyset} \underline{E[f]}$
$\overline{E \text{ sy } a} \xrightarrow{\emptyset} \overline{E \text{ sy } a}$	$\underline{E \text{ sy } a} \xrightarrow{\emptyset} \underline{E \text{ sy } a}$
$\overline{E \text{ rs } a} \xrightarrow{\emptyset} \overline{E \text{ rs } a}$	$\underline{E \text{ rs } a} \xrightarrow{\emptyset} \underline{E \text{ rs } a}$
$\overline{[E * F * E']} \xrightarrow{\emptyset} [\overline{E} * F * E']$	$[\underline{E} * F * E'] \xrightarrow{\emptyset} [E * \overline{F} * E']$
$[E * \underline{F} * E'] \xrightarrow{\emptyset} [E * F * \overline{E}']$	$[E * \underline{F} * E'] \xrightarrow{\emptyset} [E * \overline{F} * E']$
$[E * F * \underline{E}'] \xrightarrow{\emptyset} [E * F * E']$	

Tabla 2.2: Reglas de inacción (I) de PBC

A partir de las reglas de inacción, y considerando también las transiciones en sentido inverso, se define la relación de equivalencia estructural entre las expresiones dinámicas:

$$\equiv =_{def} (\xrightarrow{\emptyset} \cup \xleftarrow{\emptyset})^*$$

Como es habitual, denotaremos por $[G]_{\equiv}$ a la clase de equivalencia que contiene a G .

$\frac{\forall op \in \{;, \square\}, G \xrightarrow{\emptyset} G'}{op(G, E) \xrightarrow{\emptyset} op(G', E)}$ $\frac{G_1 \xrightarrow{\emptyset} G'_1}{G_1 \parallel G_2 \xrightarrow{\emptyset} G'_1 \parallel G_2}$ $\frac{\forall op \in \{sy, rs\}, G \xrightarrow{\emptyset} G'}{op(G, a) \xrightarrow{\emptyset} op(G', a)}$ $\frac{G \xrightarrow{\emptyset} G'}{[G * F * E'] \xrightarrow{\emptyset} [G' * F * E']}$ $\frac{G \xrightarrow{\emptyset} G'}{[E * F * G] \xrightarrow{\emptyset} [E * F * G']}$	$\frac{\forall op \in \{;, \square\}, G \xrightarrow{\emptyset} G'}{op(E, G) \xrightarrow{\emptyset} op(E, G')}$ $\frac{G_2 \xrightarrow{\emptyset} G'_2}{G_1 \parallel G_2 \xrightarrow{\emptyset} G_1 \parallel G'_2}$ $\frac{G \xrightarrow{\emptyset} G'}{G[f] \xrightarrow{\emptyset} G'[f]}$ $\frac{G \xrightarrow{\emptyset} G'}{[E * G * E'] \xrightarrow{\emptyset} [E * G' * E']}$
---	--

Tabla 2.3: Reglas de inacción (II) de PBC

La semántica operacional que se define tiene la propiedad de que dadas dos expresiones dinámicas estructuralmente equivalentes, éstas pueden realizar los mismos pasos, por lo que deberán tener más tarde la misma red de Petri marcada asociada.

Además de las reglas de inacción, se define una semántica de secuencia de pasos al estilo de [Plo81], de acuerdo al siguiente formato:

$$G \xrightarrow{\gamma} G', \text{ donde } \gamma \in \mathcal{B}(\mathcal{L}),$$

que se interpreta de la forma siguiente:

La expresión dinámica G se transforma mediante el paso γ , es decir ejecutando simultánea y concurrentemente todas las multiacciones que aparecen en γ , en la expresión dinámica G' .

Las reglas de transición de la semántica operacional son las mostradas en la tabla 2.4, donde el símbolo $+$ denota la unión de multiconjuntos.

Definición 2.13 Siendo G y H expresiones dinámicas tales que

$$G \equiv G' \xrightarrow{\gamma_1} G_1 \equiv G'_1 \xrightarrow{\gamma_2} \dots G_{n-1} \equiv G'_{n-1} \xrightarrow{\gamma_n} H' \equiv H, \quad \text{con } n \geq 0$$

diremos que H es una expresión dinámica *derivada* de G .

Denotaremos por $[G]$, y lo llamaremos *alcance* de G , al conjunto de todas las expresiones dinámicas derivadas de G . □

AR	$\bar{\alpha} \xrightarrow{\{\alpha\}} \underline{\alpha}$
PAR	$\frac{G \xrightarrow{\Gamma} G', H \xrightarrow{\Delta} H'}{G \parallel H \xrightarrow{\Gamma+\Delta} G' \parallel H'}$
CL	$\frac{G \xrightarrow{\Gamma} G'}{G \square F \xrightarrow{\Gamma} G' \square F}$
CR	$\frac{H \xrightarrow{\Delta} H'}{E \square H \xrightarrow{\Delta} E \square H'}$
SL	$\frac{G \xrightarrow{\Gamma} G'}{G; F \xrightarrow{\Gamma} G'; F}$
SR	$\frac{H \xrightarrow{\Delta} H'}{E; H \xrightarrow{\Delta} E; H'}$
SY1	$\frac{G \xrightarrow{\Gamma} G'}{G \text{ sy } a \xrightarrow{\Gamma} G' \text{ sy } a}$
SY2	$\frac{G \text{ sy } a \xrightarrow{\{\alpha+\{a\}\}+\{\beta+\{\hat{a}\}\}+\Gamma} G' \text{ sy } a}{G \text{ sy } a \xrightarrow{\{\alpha+\beta\}+\Gamma} G' \text{ sy } a}$
RR	$\frac{G \xrightarrow{\Gamma} G'}{G[f] \xrightarrow{f(\Gamma)} G'[f]}$
RS	$\frac{G \xrightarrow{\Gamma} G'}{G \text{ rs } a \xrightarrow{\Gamma} G' \text{ rs } a} \quad \forall \alpha \in \Gamma : a, \hat{a} \notin A(\alpha)$
IT1	$\frac{G \xrightarrow{\Gamma} G'}{[G * F * E] \xrightarrow{\Gamma} [G' * F * E]}$
IT2	$\frac{G \xrightarrow{\Gamma} G'}{[E * G * E'] \xrightarrow{\Gamma} [E * G' * E']}$
IT3	$\frac{G \xrightarrow{\Gamma} G'}{[E * F * G] \xrightarrow{\Gamma} [E * F * G']}$

Tabla 2.4: Reglas de transición de la semántica operacional de PBC

Definición 2.14 Dada una expresión dinámica G , definimos el *(multi)sistema de transiciones* asociado a G , como $ts(G) = (V, A, v_0)$, donde:

- $V = \{ [H]_{\equiv} \mid H \in [G] \}$ es el conjunto de estados,
- $v_0 = [G]_{\equiv}$ es el estado inicial,
- $A = \{ ([H]_{\equiv}, \gamma, [J]_{\equiv}) \mid H \in [G] \wedge H \xrightarrow{\gamma} J \}$ es el multiconjunto de transiciones ó arcos. \square

A continuación recordaremos la definición de sistemas de transiciones isomorfos.

Definición 2.15 Dados dos sistemas de transiciones (V^1, A^1, v_0^1) y (V^2, A^2, v_0^2) , diremos que son isomorfos si existe una biyección $h : V^1 \rightarrow V^2$ tal que

- $h(v_0^1) = v_0^2$
- $\forall v, w \in V^1$, y $l \in \mathcal{B}(\mathcal{L})$ se tiene que $A^1(v, l, w) = A^2(h(v), l, h(w))$. \square

Sin embargo, la simple noción de isomorfía entre los correspondientes sistemas de transiciones a partir de expresiones iniciales, no es suficiente para obtener una congruencia sobre el conjunto de expresiones estáticas, como se muestra en el ejemplo siguiente.

Ejemplo 2.3 Sean las expresiones estáticas:

$$E_1 = \{a\} \quad F_1 = \{a\}; \{b\}rs b$$

Los sistemas de transiciones $ts(\overline{E_1})$ y $ts(\overline{F_1})$ se muestran en la figura 2.2.



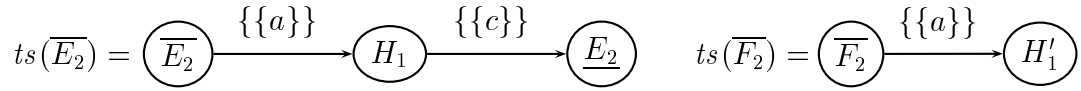
Figura 2.2: $ts(\overline{E_1})$ y $ts(\overline{F_1})$

Obviamente, ambos sistemas son isomorfos, pero si consideramos

$$E_2 = E_1; \{c\} \quad F_2 = F_1; \{c\}$$

tenemos que sus sistemas de transiciones son los mostrados en la figura 2.3,

y es evidente que $ts(\overline{E_2})$ y $ts(\overline{F_2})$ no son isomorfos. \square

Figura 2.3: $ts(\overline{E_2})$ y $ts(\overline{F_2})$

Para resolver este problema se necesita distinguir entre expresiones dinámicas que alcanzan su estado final (como E_1 en la figura 2.2), y aquéllas que nunca lo alcanzan (como F_1 en la figura 2.2). Esto se puede conseguir extendiendo el sistema de transiciones con las dos nuevas transiciones mostradas en la tabla 2.5.

(Skp)	$\overline{E} \xrightarrow{\{skip\}} \underline{E}$
(Rdo)	$\underline{E} \xrightarrow{\{redo\}} \overline{E}$

Tabla 2.5: Reglas Skp y Rdo de PBC

Definición 2.16 Sea E una expresión estática, definimos un nuevo sistema de transiciones $ts_2(\overline{E}) = (V, A, v_0)$, en la forma:

- $V = \{ [H]_{\equiv} \mid H \in [\overline{E}] \} \cup \{ [\underline{E}]_{\equiv} \}$
- $v_0 = [\overline{E}]_{\equiv}$
- $A = \{ ([H]_{\equiv}, \gamma, [J]_{\equiv}) \mid H \in [G] \wedge H \xrightarrow{\gamma} J \} \cup \{ ([\overline{E}]_{\equiv}, \{skip\}, [\underline{E}]_{\equiv}), ([\underline{E}]_{\equiv}, \{redo\}, [\overline{E}]_{\equiv}) \}$

□

Definición 2.17 Dadas dos expresiones estáticas E_1 y E_2 , con

$$ts_2(\overline{E_1}) = (V^{E_1}, A^{E_1}, v_0^{E_1})$$

$$ts_2(\overline{E_2}) = (V^{E_2}, A^{E_2}, v_0^{E_2})$$

diremos que son *ts-isomorfias*, y escribiremos $E_1 \cong E_2$, si existe una función biyectiva:

$$h : V^{E_1} \cup \{ [\underline{E_1}]_{\equiv} \} \rightarrow V^{E_2} \cup \{ [\underline{E_2}]_{\equiv} \}$$

tal que:

- $h(v_0^{E_1}) = v_0^{E_2}$,
- $\forall v^{E_1}, w^{E_1} \in V^{E_1}$, se tiene que $A^{E_1}(v^{E_1}, l, w^{E_1}) = A^{E_2}(h(v^{E_1}), l, h(w^{E_1}))$

□

Ejemplo 2.4 Consideremos de nuevo las expresiones estáticas del ejemplo 2.3:

$$E_1 = \{a\}, \quad F_1 = \{a\}; \{b\}rs b$$

Se tiene que $E_1 \not\cong F_1$, como se puede apreciar en la figura 2.4, donde se presentan $ts_2(\overline{E_1})$ y $ts_2(\overline{F_1})$.

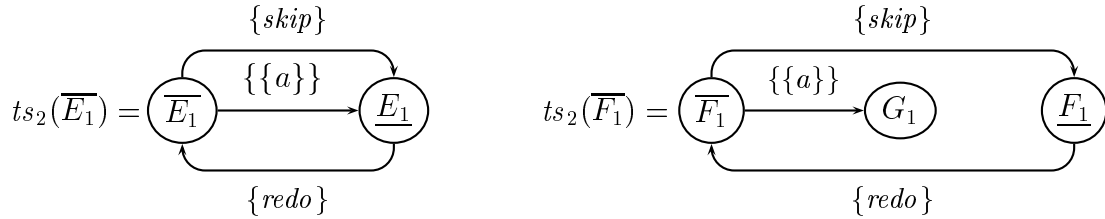


Figura 2.4: $ts_2(\overline{E_1})$ y $ts_2(\overline{F_1})$ □

Ejemplo 2.5 Consideremos ahora

$$E_3 = \{a\} \parallel \{\hat{a}\}$$

$$F_3 = (\{a\}; \{\hat{a}\}) \square (\{\hat{a}\}; \{a\})$$

En la figura 2.5 se muestran $ts_2(\overline{E_3})$ y $ts_2(\overline{F_3})$, pudiéndose apreciar claramente que $E_3 \not\cong F_3$.

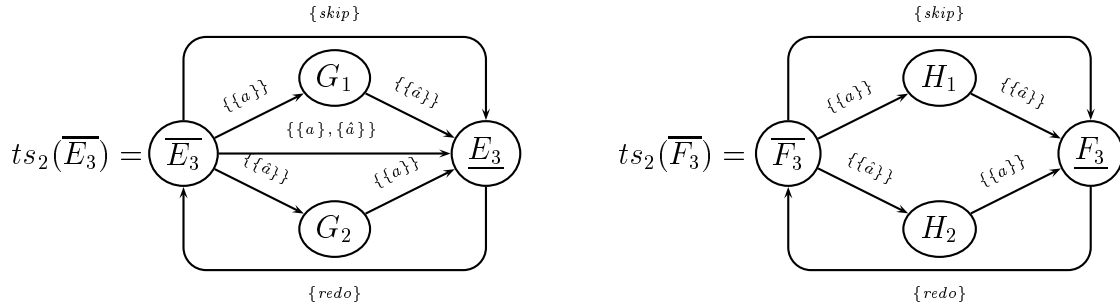


Figura 2.5: $ts_2(\overline{E_3})$ y $ts_2(\overline{F_3})$ □

Para la noción de equivalencia \cong se tienen las siguientes propiedades.

Proposición 2.1 Sean E, F, E' expresiones estáticas cualesquiera, para todo $a, b \in \mathcal{A}$, se tiene:

- (i) $E \parallel F \cong F \parallel E$
- (ii) $E \parallel (F \parallel E') \cong (E \parallel F) \parallel E'$

- (iii) $E \square F \cong F \square E$
- (iv) $E \square (F \square E') \cong (E \square F) \square E'$
- (v) $E ; (F ; E') \cong (E ; F) ; E'$
- (vi) $E \text{ sy } a \text{ sy } b \cong E \text{ sy } b \text{ sy } a$
- (vii) $E \text{ sy } a \text{ sy } a \cong E \text{ sy } a$
- (viii) $E \text{ sy } \hat{a} \cong E \text{ sy } a$
- (ix) $(E ; F) \text{ sy } a \cong (E \text{ sy } a) ; (F \text{ sy } a)$
- (x) $(E \square F) \text{ sy } a \cong (E \text{ sy } a) \square (F \text{ sy } a)$
- (xi) $(E \parallel F) \text{ sy } a \cong ((E \text{ sy } a) \parallel (F \text{ sy } a)) \text{ sy } a$
- (xii) $E \text{ rs } a \text{ rs } b \cong E \text{ rs } b \text{ rs } a$
- (xiii) $E \text{ rs } a \text{ rs } a \cong E \text{ rs } a$
- (xiv) $E \text{ rs } \hat{a} \cong E \text{ rs } a$
- (xv) $(E ; F) \text{ rs } a \cong (E \text{ rs } a) ; (F \text{ rs } a)$
- (xvi) $(E \square F) \text{ rs } a \cong (E \text{ rs } a) \square (F \text{ rs } a)$
- (xvii) $(E \parallel F) \text{ rs } a \cong (E \text{ rs } a) \parallel (F \text{ rs } a)$
- (xviii) Si $a \notin \{b, \hat{b}\} \Rightarrow (E \text{ sy } a) \text{ rs } b \cong (E \text{ rs } b) \text{ sy } a$
- (xix) $(E \text{ rs } a)[f] \cong (E[f]) \text{ rs } f(a)$
- (xx) $[a : [b : E]] \cong [b : [a : E]]$
- (xxi) $[a : [a : E]] \cong [a : E]$
- (xxii) $[\hat{a} : E] \cong [a : E]$
- (xxiii) $[a : (E ; F)] \cong [a : E] ; [a : F]$
- (xxiv) $[a : (E \square F)] \cong [a : E] \square [a : F]$
- (xxv) Si $a \notin \{b, \hat{b}\} \Rightarrow [a : E] \text{ rs } b \cong [a : (E \text{ rs } b)]$
- (xxvi) $[a : E][f] = [f(a) : E[f]]$

- (xxvii) Si $a \notin \{b, \hat{b}\} \Rightarrow [a : E] sy b \cong [a : (E sy b)]$
- (xxviii) $E[f][g] \cong E[g \circ f]$
- (xxix) $E[id] \cong E$
- (xxx) $(E ; F)[f] \cong (E[f]) ; (F[f])$
- (xxxii) $(E \parallel F)[f] \cong (E[f]) \parallel (F[f])$
- (xxxiii) $(E \square F)[f] \cong (E[f]) \square (F[f])$
- (xxxiiii) $(E sy a)[f] \cong E[f] sy (f(a))$
- (xxxv) $[E * F * E'] sy a \cong [E sy a * F sy a * E' sy a]$
- (xxxvi) $[E * F * E'] rs a \cong [E rs a * F rs a * E' rs a]$
- (xxxvii) $[a : [E * F * E']] \cong [[a : E] * [a : F] * [a : E']]$

□

A continuación, se define la noción de expresiones *fuertemente equivalentes*, la cual está inspirada en el concepto de bisimulación fuerte.

Definición 2.18 Dadas dos expresiones estáticas E_1 y E_2 , con

$$ts_2(\overline{E_1}) = (V^{E_1}, A^{E_1}, v_0^{E_1})$$

$$ts_2(\overline{E_2}) = (V^{E_2}, A^{E_2}, v_0^{E_2})$$

diremos que son *fuertemente equivalentes*, y escribiremos $E_1 \approx E_2$, si existe una relación $\mathcal{SB} \subseteq V^{E_1} \times V^{E_2}$, que diremos es una *bisimulación fuerte*, tal que:

- $(v_0^{E_1}, v_0^{E_2}) \in \mathcal{SB}$
- Si $(v^{E_1}, v^{E_2}) \in \mathcal{SB}$ entonces
 - $(v^{E_1}, l, w^{E_1}) \in A^{E_1} \Rightarrow \exists w^{E_2} \in V^{E_2}$ tal que $(v^{E_2}, l, w^{E_2}) \in A^{E_2} \wedge (w^{E_1}, w^{E_2}) \in \mathcal{SB}$.
 - $(v^{E_2}, l, w^{E_2}) \in A^{E_2} \Rightarrow \exists w^{E_1} \in V^{E_1}$ tal que $(v^{E_1}, l, w^{E_1}) \in A^{E_1} \wedge (w^{E_1}, w^{E_2}) \in \mathcal{SB}$.

□

Ejemplo 2.6 Consideremos las expresiones estáticas

$$E_4 = (\{a\}; \{b\}) \square (\{a\}; \{c\})$$

$$F_4 = \{a\}; (\{b\} \square \{c\})$$

Sus correspondientes sistemas de transiciones, $ts_2(\overline{E_4})$ y $ts_2(\overline{F_4})$ son los mostrados en la figura 2.6. Considerando ahora $G_1, G_2 \in \overline{E_4}$ y $H_1 \in \overline{F_4}$, dados por

$$G_1 = (\{a\}; \overline{\{b\}}) \square (\{a\}; \{c\})$$

$$G_2 = (\{a\}; \{b\}) \square (\{a\}; \overline{\{c\}})$$

$$H_1 = \{a\}; (\overline{\{b\}} \square \overline{\{c\}})$$

tenemos que:

$$\mathcal{SB} = \{(\overline{E_4}, \overline{F_4}), (G_1, H_1), (G_2, H_1), (\underline{E_4}, \underline{F_4})\}$$

es una bisimulación fuerte, de modo que $E_4 \approx F_4$.

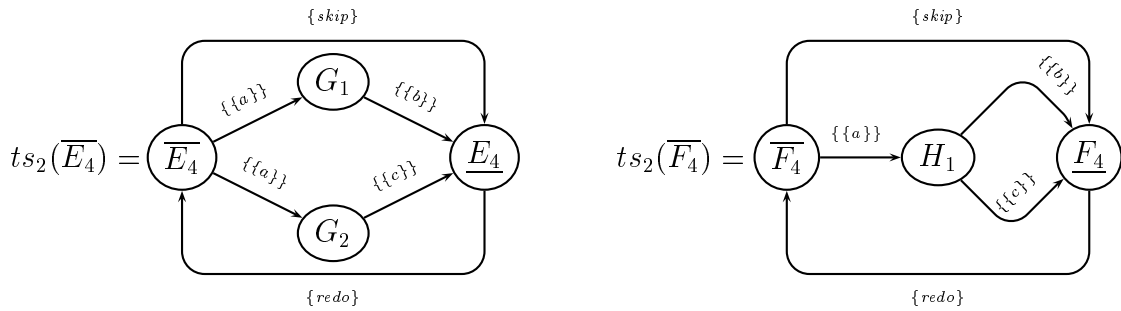


Figura 2.6: $ts_2(\overline{E_4})$ y $ts_2(\overline{F_4})$

□

Proposición 2.2 Dadas dos expresiones estáticas cualesquiera, E_1 y E_2 , si $E_1 \cong E_2$ entonces $E_1 \approx E_2$ □

Observemos que aunque se tiene $\alpha \square \alpha \cong \alpha$, en general no se cumple $E \square E \cong E$, aunque sí se cumple que $E \square E \approx E$.

Otra propiedad interesante es la siguiente:

$$[E * F * E'] \approx (E; X), \quad \text{donde } X =_{def} E' \square (F; X)$$

Semántica Denotacional

Los objetos semánticos que asociaremos a toda expresión se denominan *cajas*, las cuales se definen como clases de redes de Petri etiquetadas con unas ciertas propiedades. En primer lugar definiremos lo que en este contexto se entiende por red de Petri etiquetada.

Definición 2.19 Una red de Petri etiquetada es una tupla $\Sigma = (S, T, W, \lambda)$, donde (S, T, W) es una red de Petri ($S \cap T = \emptyset$, $W : (S \times T) \cup (T \times S) \rightarrow \mathbb{N}$) y λ es una función sobre $S \cup T$ tal que:

- $\lambda|_S : S \rightarrow \{e, i, x\}$
- $\lambda|_T : T \rightarrow \mathcal{B}(\mathcal{L}) \times \mathcal{L}$

Los lugares s tales que $\lambda(s) = e$ se llaman lugares de entrada, y el conjunto de los mismos se denotará por ${}^\circ\Sigma$: ${}^\circ\Sigma = \{s \in S : \lambda(s) = e\}$. A los lugares s tales que $\lambda(s) = x$ les llamaremos lugares de salida, y al conjunto de lugares de salida lo denotaremos por Σ° : $\Sigma^\circ = \{s \in S : \lambda(s) = x\}$. Al resto de lugares los llamaremos internos. Finalmente, si para $t \in T$, $\lambda(t) = \rho_t \subseteq \mathcal{B}(\mathcal{L}) \times \mathcal{L}$, se conocerá como *función de reetiquetado*. Entonces, si $(\Gamma, \alpha) \in \rho_t$, el significado intuitivo es que si tomamos un conjunto finito de transiciones cuyas etiquetas forman Γ , el resultado será una transición con etiqueta visible α . Algunos casos particulares que nos serán de utilidad posteriormente son:

- $\rho_{id} = \{(\{\alpha\}, \alpha) \mid \alpha \in \mathcal{L}\}$ (reetiquetado identidad).

Su significado intuitivo es que no se hace nada, dejando las cosas como estaban.

- $\rho_\alpha = \{(\{\beta\}, \alpha) \mid \beta \in \mathcal{L}\}$ (reetiquetado constante).

A todo elemento de \mathcal{L} le corresponde la constante α . Se puede representar mediante el simple valor α .

- $\rho_{[f]} = \{(\{\alpha\}, f(\alpha)) \mid \alpha \in \mathcal{L}\}$ (reetiquetado de renombramiento).

Modifica la etiqueta de todas las transiciones aplicando la función f .

- $\rho_{\mathcal{L}'} = \{(\{\alpha\}, \alpha) \mid \alpha \in \mathcal{L}'\}$ (reetiquetado de proyección, $\mathcal{L}' \subseteq \mathcal{L}$)

Sólo conserva los elementos de \mathcal{L}' . Un caso particular de proyección se da para la restricción.

- $\rho_{rsa} = \rho_{\mathcal{L}'}$ con $\mathcal{L}' = \mathcal{B}(\mathcal{A} \setminus \{a, \hat{a}\})$ (reetiquetado de restricción)

Sólo conserva las transiciones que no contienen a ni \hat{a} .

- ρ_{sya} (reetiquetado de sincronización) es la menor función de reetiquetado que contiene a ρ_{id} y tal que si $(\Gamma, \alpha + \{a\}) \in \rho_{sya}$ y $(\Delta, \beta + \{\hat{a}\}) \in \rho_{sya}$ entonces $(\Gamma + \Delta, \alpha + \beta) \in \rho_{sya}$

- $\rho_{[a]} = \{(\Gamma, \alpha) \in \rho_{sya} \mid \alpha(a) = 0 = \alpha(\hat{a})\}$ (re Etiquetado de scoping).
Primero hace la sincronización, y después se eliminan aquellas transiciones que contengan a ó \hat{a} . □

Definición 2.20 Una caja de Petri es una red de Petri ordinaria etiquetada $\Sigma = (S, T, W, \lambda)$ que cumple:

- (o) $\forall t \in T, \bullet t \neq \emptyset \neq t \bullet$ (a todos los lugares llegan arcos, y de todos los lugares salen arcos).
- (i) ${}^\circ \Sigma \neq \emptyset$ (tiene al menos un lugar de entrada).
- (ii) $\Sigma^\circ \neq \emptyset$ (tiene al menos un lugar de salida).
- (iii) $\forall s \in {}^\circ \Sigma, \forall t \in T, W(t, s) = 0$ (a los lugares de entrada no llegan arcos).
- (iv) $\forall s \in \Sigma^\circ, \forall t \in T, W(s, t) = 0$ (de los lugares de salida no salen arcos).
- (v) $\forall t \in T, \bullet t \cap t \bullet = \emptyset$ (no hay arcos de ida y vuelta). □

Nos interesarán dos tipos particulares de cajas: las cajas planas y las cajas operador. Las cajas planas van a ser los objetos semánticos básicos, mientras que las cajas operador son las encargadas de combinarlas entre sí, obteniendo otras planas, correspondientes a la traducción de expresiones más complejas. La idea es que dada una caja operador Ω y una tupla de cajas planas Σ (con tantos elementos como transiciones tenga Ω) se obtendrá una nueva caja plana, refinando cada transición de Ω con su caja correspondiente de Σ . En nuestro contexto de PBC, las definiciones de estos dos tipos de cajas serían las siguientes.

Definición 2.21 Una caja plana es una caja en la que cada transición tiene asociada una función de re Etiquetado constante. □

Definición 2.22 Una caja operador es una caja en la que cada transición tiene asociada una función de re Etiquetado distinta de la constante. □

A continuación se describe el mecanismo general que nos permite asociar a cada expresión estática una red, combinando los procesos de refinamiento y re Etiquetado. Se va a realizar en dos fases: en la primera se realiza el cambio de interfaz (por medio de la función de re Etiquetado), y en la segunda se procede al refinamiento.

Definición 2.23 Sea $\Omega = (S, T, W, \lambda)$ una caja operador, la operación de composición que especifica Ω se aplica sobre cualquier conjunto de cajas planas indexadas por las transiciones de Ω , $\Sigma = \{\Sigma_v \mid v \in T\}$, donde $\Sigma_v = (S_v, T_v, W_v, \lambda_v)$, $\forall v \in T$. Se realiza en dos fases:

Fase 1 (cambio de interfaz, reetiquetación) : Sea $v_1 \in T$, con $\lambda(v_1) = \rho_{v_1}$.

Entonces la función de reetiquetado ρ_{v_1} se aplica a Σ_1 , obteniéndose una caja intermedia Θ_1 ; en ella aparecerán nuevas transiciones correspondientes a las nuevas etiquetas, con las conectivas heredadas de las transiciones que las originaron. Se repetirá este mismo procedimiento con todas las transiciones $v_i \in T$.

Fase 2 (Refinamiento) : Para $v_1 \in T$, se realiza el refinamiento por Θ_1 de la forma siguiente:

$$\Omega[v_1 \leftarrow \Theta_1] = (\Omega \sqcup \Theta_1) \oplus (\bullet v_1 \otimes \circ \Theta_1, l_1) \oplus (v_1^\bullet \otimes \Theta_1^\circ, l_2) \ominus (\bullet v_1 \cup \circ \Theta_1) \ominus (v_1^\bullet \cup \Theta_1^\circ)$$

donde \sqcup representa la unión de redes de Petri (lugares y transiciones, con sus respectivas etiquetas y conectivas), \oplus la adición de lugares a la red (con sus correspondientes arcos), \otimes el producto cartesiano de lugares, l_1 es la función de etiquetado que aplica la etiqueta (lugar de entrada, salida o interno) que tenía $\bullet v_1$, l_2 es la función de etiquetado que aplica la etiqueta (lugar de entrada, salida o interno) que tenía v_1^\bullet , y \ominus la eliminación de lugares (con sus correspondientes arcos).

Se repite este procedimiento para todos los restantes $v_i \in T$. □

Puede comprobarse que el refinamiento es independiente del orden en que se elijan las transiciones a refinar de Ω , lo que equivale a realizar todos los refinamientos al tiempo.

Como era de esperar, se tiene la siguiente propiedad:

Proposición 2.3 La red $op_\Omega(\Sigma)$ es una caja plana. □

A continuación, ilustraremos con un ejemplo las dos fases.

Ejemplo 2.7 Sean Ω y $\Sigma = (\Sigma_1, \Sigma_2)$ las redes mostradas en la figura 2.7, donde $\rho_1 = \rho_{id}$ y $\rho_2 = \rho_{id} \cup \{(\{\alpha, \delta\}, \varepsilon)\}$

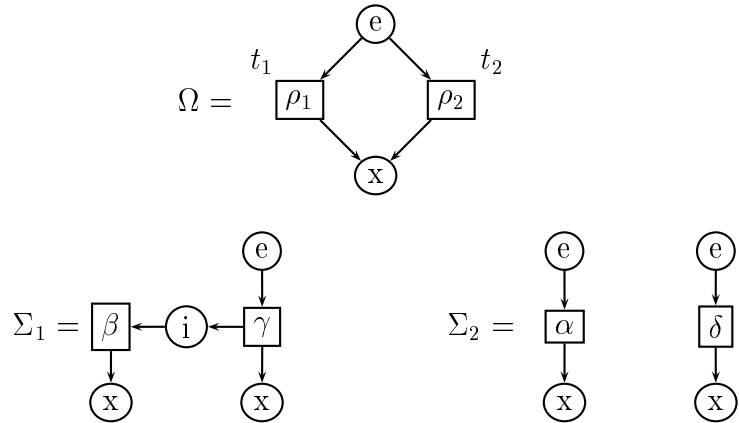


Figura 2.7: Definición de Ω y Σ

Para obtener $op_{\Omega}(\Sigma)$ se procede de la forma siguiente :

Fase 1: Se aplica ρ_1 a Σ_1 y ρ_2 a Σ_2 , obteniéndose Θ_1 y Θ_2 , respectivamente, (ver figura 2.8).

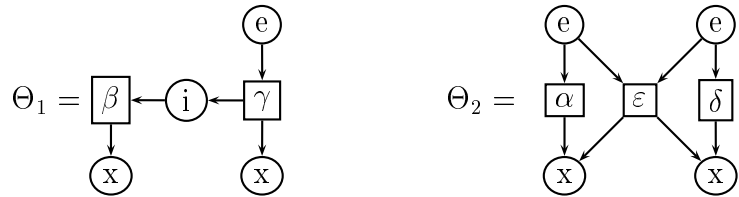


Figura 2.8: Ejemplo de $op_{\Omega}(\Sigma)$. Fase 1: obtención de Θ_1 y Θ_2

Fase 2: Se refina t_1 por Θ_1 (figura 2.9).

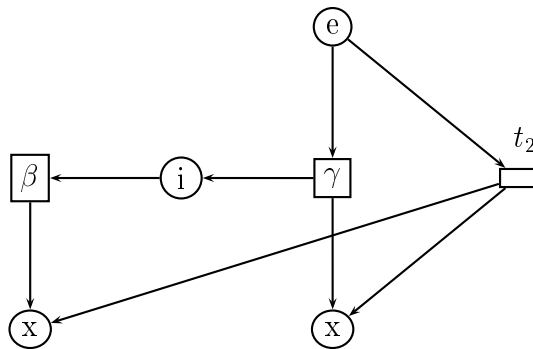


Figura 2.9: Ejemplo de $op_{\Omega}(\Sigma)$. Fase 2: $\Omega[t_1 \leftarrow \Theta_1] = \Omega'$

y luego se refina t_2 por Θ_2 (figura 2.10), obteniéndose así el resultado final $op_{\Omega}(\Sigma)$.

□

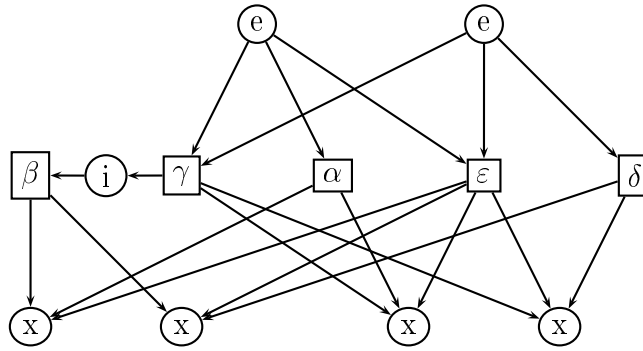


Figura 2.10: Ejemplo de $op_{\Omega}(\Sigma)$. Fase 2: $\Omega'[t_2 \leftarrow \Theta_2] = op_{\Omega}(\Sigma)$

Considerando las cajas operador que aparecen en las figuras 2.11 y 2.12, la semántica denotacional se define formalmente mediante el homomorfismo Box , que asocia a cada expresión estática una caja plana:

$$Box(\alpha) = N_{\alpha}$$

$$Box(op(E_1, \dots, E_n)) = \Omega_{op}(Box(E_1), \dots, Box(E_n))$$

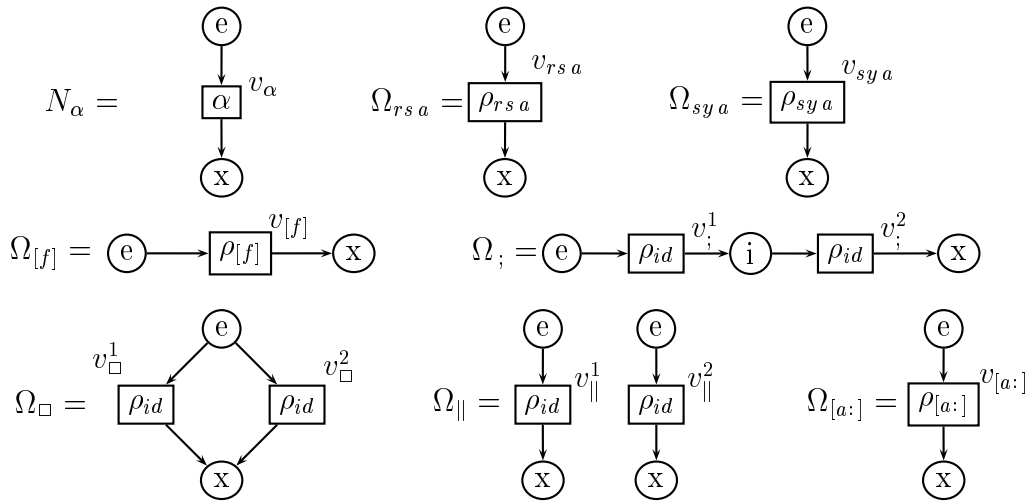


Figura 2.11: cajas operador de PBC finito

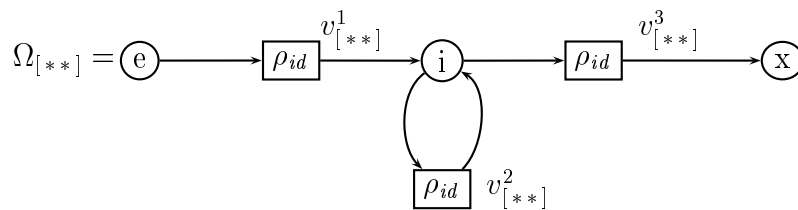


Figura 2.12: caja operador (no segura) para $[**]$ de PBC

Para el caso de la iteración se dispone de dos opciones [BDK01b] al determinar la caja operador. Una de ellas proporciona siempre cajas 1-seguras (ver figura 2.13), pero su caja operador tiene seis transiciones. La otra opción de caja operador es una versión más simple, con tres transiciones (figura 2.12), pero tiene el inconveniente de que se pueden generar redes de Petri 2-seguras, aunque esto sólo es posible cuando en el cuerpo de la iteración aparece un comportamiento paralelo al más alto nivel. En nuestro modelo (sPBC) impondremos (por otras razones) una restricción sintáctica, como consecuencia de la cual ese caso no se podrá dar, por lo que optaremos por considerar la versión más simple. Sin embargo, esta caja operador contiene transiciones con condiciones laterales (arcos de ida y vuelta), por lo cual eliminaremos la restricción $\bullet t \cap t^\bullet = \emptyset$ introducida en la definición de caja, que originalmente sí aparece cuando se considera la versión 1-segura de la iteración, y que sólo se introduce a la hora de definir la caja operador en general, con la intención de simplificar el modelo [BK95].

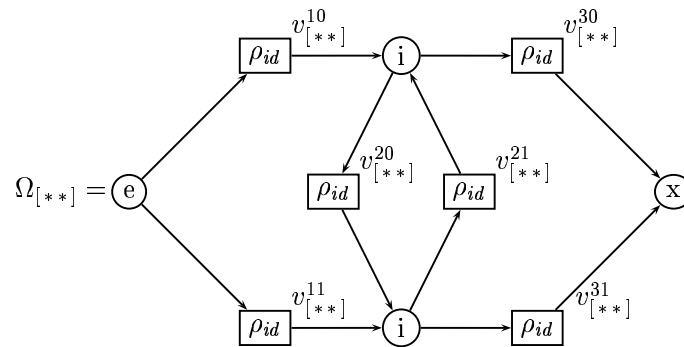


Figura 2.13: caja operador (segura) para $[**]$ de PBC

Asimismo, tomamos M_e como marcado inicial para la caja plana así generada, proporcionando siempre cajas planas seguras y limpias (una caja Σ segura es limpia si todos los marcados alcanzables en (Σ, M_e) que satisfacen $\Sigma^\circ \subseteq M$ ó ${}^\circ\Sigma \subseteq M$, entonces $\Sigma^\circ = M$ ó ${}^\circ\Sigma \subseteq M$, respectivamente).

A continuación, mostramos que para toda expresión estática E , la semántica operacional de \overline{E} y la semántica de la correspondiente caja plana son isomorfas. La interpretación de la evolución de la red de Petri es bajo concurrencia real.

Teorema 2.1 Para toda expresión estática E de PBC, el sistema de transiciones $ts(\overline{E})$ de \overline{E} , y el grafo de accesibilidad de la red de Petri marcada $(Box(E), M_e)$ son isomorfos. \square

Ejemplo 2.8 Para $E_5 = (\{a\} \parallel \{\hat{a}, \hat{a}\} \parallel \{a\})$ sy a la caja plana asociada es la descrita en la figura 2.14.

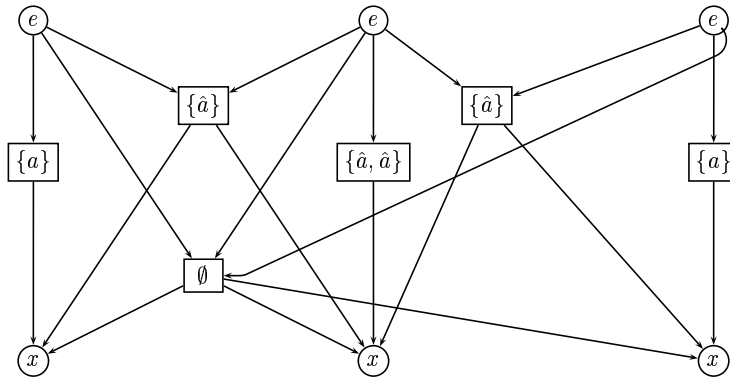


Figura 2.14: caja plana de E_5

□

2.2 Evaluación de Prestaciones

Como ya hemos visto, con el análisis cualitativo o validación de propiedades lógicas de un sistema concurrente, podremos responder a preguntas sobre *corrección*, *ausencia de bloqueos*, *vivacidad*, etc. Con la *evaluación de prestaciones* [Jai91] pretendemos responder a preguntas del estilo: ¿cómo de rápido se realizan las tareas?, ¿con qué aprovechamiento se utilizan los recursos?, ¿con qué probabilidad puede fallar un sistema?, ¿cuánto tiempo tarda en fallar?, etc. Estas cuestiones de tipo cuantitativo se concretan en el cálculo de ciertos *índices de rendimiento* (o de *prestaciones*), que son descriptores cuantificables usados para representar algún aspecto del rendimiento del sistema, que permiten representar medidas de *eficiencia*, de *utilización*, de *fiabilidad*, de *disponibilidad*, etc.

En el marco de esta tesis, nos interesan las técnicas basadas en modelos, en los que se maneja una visión abstracta y simplificada del sistema real a estudiar. Un clásico de la teoría de colas con aplicaciones a los sistemas informáticos se puede encontrar en [Kle75, Kle76]. Más en particular, nos interesan los formalismos matemáticos orientados a la evaluación de prestaciones, junto con el análisis cualitativo. Nos basaremos en las extensiones estocásticas de las redes de Petri (redes de Petri estocásticas, [AMBC⁺95]) y de las álgebras de procesos (álgebras de procesos estocásticas, [HHK02]). En ambos se puede recurrir a su Cadena de Markov de Tiempo

Continuo (CTMC, Continuous Time Markovian Chain [Cin75]) asociada a la evolución del modelo para el estudio de prestaciones. En [Hav98] podemos encontrar una descripción de la panorámica actual de las técnicas disponibles para la evaluación de prestaciones de los sistemas concurrentes.

A continuación, pasamos a introducir los formalismos matemáticos que utilizaremos a lo largo de esta tesis: las cadenas de Markov, las redes de Petri estocásticas y las álgebras de procesos estocásticas.

2.2.1 Cadenas de Markov de Tiempo Continuo (CTMC)

Un *proceso estocástico* [Ros83] es un conjunto de variables aleatorias $\{X(t), t \in T\}$ donde T es un conjunto índice que suele representar el tiempo. El *espacio de estados* del proceso es el conjunto de todos los posibles valores que las variables aleatorias $X(t)$ pueden tomar.

Un *proceso markoviano* es un proceso estocástico en el que el conjunto de las variables aleatorias $\{X(t)\}$ tiene la *propiedad markoviana*, también conocida como *pérdida de memoria*: dado un valor $X(t)$ con $t \in T$, los valores futuros $X(s)$, con $s > t$ no dependen de la historia pasada, o dicho de otra forma, la probabilidad de que un evento futuro ocurra, dado cualquier evento pasado y el estado presente, es independiente del evento pasado y sólo depende del estado actual. Así, en el caso de que el espacio de estados sea discreto y el tiempo continuo, se conoce como Cadena de Markov de Tiempo Continuo (CTMC, Continuous Time Markovian Chain).

Definición 2.24 El proceso estocástico $\{X(t)\}$ es una CTMC si para cualquier secuencia t_1, t_2, \dots, t_{n+1} tal que $t_1 < t_2 < \dots < t_{n+1}$ se tiene:

$$\begin{aligned} Pr\{X(t_{n+1}) = j \mid X(t_1) = i_1, X(t_2) = i_2, \dots, X(t_n) = i_n\} = \\ Pr\{X(t_{n+1}) = j \mid X(t_n) = i_n\} \quad \square \end{aligned}$$

De esta forma el tiempo de estancia en los estados no tiene memoria (bastaría con considerar en la definición anterior $i_1 = i_2 = \dots = i_n$), y la única función de probabilidad continua que satisface esta propiedad es la *exponencial (negativa)*. Por tanto, el tiempo de permanencia en los estados sigue una distribución exponencial.

La función de distribución de una exponencial negativa viene caracterizada por su razón (o parámetro) r , y es de la forma:

$$F(x) = 1 - e^{-rx} \quad \text{con } x \geq 0$$

La esperanza matemática de una distribución exponencial de parámetro r es $1/r$. Además de la propiedad enunciada de pérdida de memoria de la distribución exponencial, también cumple que si X_1 y X_2 son dos variables aleatorias independientes distribuidas exponencialmente, de parámetros r_1 y r_2 , respectivamente, la distribución que sigue la variable aleatoria definida como el mínimo de ambas, $Y = \min\{X_1, X_2\}$, es también una distribución exponencial, en este caso de parámetro $r_1 + r_2$.

Una CTMC se puede caracterizar a través de su *diagrama de transición de estados*, o también a partir de su *matriz generadora infinitesimal*.

El diagrama de transición de estados es un grafo dirigido etiquetado, cuyos nodos son los estados de la CTMC, y cada arco se etiqueta con la razón de la distribución exponencial asociada con la transición que hace moverse de un estado a otro (*velocidad o razón de transición* de estados). Cada transición está gobernada por una distribución exponencial.

La matriz generadora infinitesimal \mathbf{Q} es una matriz cuadrada, cuyos elementos, fuera de la diagonal principal, son las razones de la distribución exponencial asociada con la transición de un estado a otro. Los elementos de la diagonal principal son aquellos que hacen que la suma de cada fila sea 0.

Si denotamos por $\Pi_i(t)$ a la probabilidad de que estemos en el estado i en el instante de tiempo t ,

$$\Pi_i(t) = Pr\{X(t) = i\}$$

estas probabilidades se podrían calcular a partir del sistema de ecuaciones diferenciales:

$$\underline{\Pi}'(t) = \underline{\Pi}(t)\mathbf{Q}, \quad \text{conocido } \underline{\Pi}(0)$$

donde $\underline{\Pi}(t) = (\Pi_0(t), \Pi_1(t), \Pi_2(t), \dots)$

Normalmente nos interesa estudiar lo que ocurre en el *estado de equilibrio*, por lo cual definimos:

$$\Pi_i = \lim_{t \rightarrow \infty} \Pi_i(t)$$

Bajo ciertas condiciones (ergodicidad), estas probabilidades existen y son únicas. Para su cálculo bastaría con resolver el sistema de ecuaciones lineales:

$$\left. \begin{array}{l} \underline{\Pi}\mathbf{Q} = 0 \\ \sum_{i \in S} \Pi_i = 1 \end{array} \right\}$$

donde $\underline{\Pi} = (\Pi_0, \Pi_1, \Pi_2, \dots)$

En [Ste94] pueden encontrarse las técnicas que permiten resolver numéricamente las cadenas de Markov de tiempo continuo.

También estamos interesados en el concepto de *lumpabilidad* [KS60]. La idea es definir una relación de equivalencia sobre el espacio de estados de la CTMC, la cual proporcionará una partición sobre dicho espacio. Llamaremos *proceso agregado* a aquel constituido por los macroestados (clases de equivalencia), de forma que si el espacio de estados original era $\{X_1, X_2, \dots, X_n\}$, ahora el espacio de estados del proceso agregado sería $\{X_{[1]}, X_{[2]}, \dots, X_{[N]}\}$, con $N \leq n$ (idealmente, $N \ll n$). Si las razones de transición del proceso original venían dadas por $q(X_i, X_k)$, entonces la razón de un estado a un macroestado se define de la forma siguiente:

$$q(X_i, X_{[j]}) = \sum_{k \in [j]} q(X_i, X_k)$$

Cuando el proceso agregado sigue siendo un proceso de Markov se dice que se tiene la propiedad de *lumpabilidad*.

Definición 2.25 Un proceso de Markov es lumpable con respecto a una partición $\mathcal{X} = \{X_{[i]}\}$ si para cada distribución inicial el proceso agregado es un proceso de Markov. \square

Teorema 2.2 [KS60] Un proceso de Markov es lumpable con respecto a la partición $\mathcal{X} = \{X_{[i]}\}$ si y sólo si $\forall X_{[k]}, X_{[l]} \in \mathcal{X}, X_i, X_j \in X_{[k]}$

$$q(X_i, X_{[l]}) = q(X_j, X_{[l]}) \quad \square$$

En [Sch84] podemos encontrar una panorámica de las técnicas de agregación para cadenas de Markov.

2.2.2 Redes de Petri estocásticas

La incorporación de aspectos temporales en las redes de Petri se ha realizado de formas muy diversas. En [Bow96] se puede encontrar un estudio sobre las diferentes aproximaciones existentes. Uno de los modelos más utilizados es el propuesto por Merlin [Mer74] en los años 70, donde a cada transición se le asocian dos números

enteros, representando el primero de ellos el primer instante en el que la transición puede ser disparada, y el segundo el instante final en el que ésta debe ser disparada. Estos instantes son relativos al instante en que la transición comenzó a estar permitida. El disparo de las transiciones no conlleva tiempo, y en el mismo instante en que han sido disparadas producen nuevos tokens en los lugares de salida. En otros modelos la información temporal viene dada en los lugares [WDF85], en los tokens [vdA93], en los arcos [Wal83, dFVM00], o incluso las restricciones temporales pueden ser intervalos asociados estáticamente a los lugares, transiciones y arcos [CMS99].

Las primeras referencias de redes de Petri estocásticas se deben a Molloy [Mol81] y a Florin y Natkin [FN85] en los años 80. Desde entonces el modelo ha ido madurando dando lugar a múltiples estudios. En la actualidad contamos además con herramientas como el GreatSPN [CFGR95] de la Universidad de Turín, basado en su modelo GSPN (Generalized Stochastic Petri Nets) [AM89, AMBD98, AMBC⁺95]. Para hacer la breve introducción a las redes de Petri estocásticas que sigue nos hemos basado en [Cam01] y en [Bal01].

Por una red de Petri estocástica entendemos una red de Petri en la que cada transición tiene asociada una espera o retraso (*delay*) de disparo, también conocido como *tiempo de servicio*, que está gobernado por una distribución exponencial negativa. Esto significa que desde que una transición está activada hasta que se completa (dispara), pasa un tiempo aleatorio caracterizado por la indicada distribución exponencial negativa. Cuando varias transiciones están activadas a la vez, se aplica una *política de competencia* (*race condition*): la más rápida gana la carrera; de esta forma se acaba con el no determinismo inherente en el modelo básico de redes de Petri. Cuando una de las transiciones gana la carrera y se dispara (el disparo no conlleva tiempo), se debe decidir qué hacer con el trabajo parcialmente realizado por el resto de transiciones que estaban activadas, que han sido interrumpidas por el disparo de la ganadora, motivando posiblemente un cambio de marcado. Como los tiempos de servicio de las transiciones siguen una distribución exponencial, como consecuencia de la propiedad de *pérdida de memoria* de dicha distribución se hacen indistinguibles las siguientes políticas:

- *remuestreo*: reiniciar el trabajo la próxima vez que la transición se active,
- *memoria de activación*: al dispararse una transición, la que no ha ganado continúa su trabajo si no ha sido desactivada con la ejecución de la mencionada transición,

- *memoria del pasado*: si la transición deja de estar activada bajo el nuevo marcado, su trabajo se guarda, y la próxima vez que se active continuará donde se quedó.

En resumen, en el caso exponencial, la política que decide qué hacer con el trabajo interrumpido es irrelevante. Se tendrá también que el tiempo de permanencia en cada marcado sigue una distribución exponencial, cuyo parámetro es la suma de los parámetros de las transiciones que están activadas bajo ese marcado (por ser el mínimo de varias variables aleatorias con distribución exponencial), y así el proceso estocástico asociado será una CTMC, lo que convierte a estas redes de Petri markovianas en un modelo muy adecuado para el estudio de prestaciones. Si consideramos la posibilidad de tener lugares con más de un token, la situación puede ser más compleja, pues cuando el lugar de entrada contiene varios tokens, hay que definir además la *política de servicio*, pues se podría dar lugar a transiciones con grado de activación mayor que uno, y sería necesario definir el máximo paralelismo de ejecución permitido (*semántica de un solo servidor, de múltiples servidores o de infinitos servidores*), y la *política de cola* asumida con respecto a los tokens que residen en el lugar de entrada de una transición. Como nosotros trabajaremos con redes 1-seguras, no nos será necesario hacer esas distinciones.

El modelo sobre el que nos hemos apoyado es GSPN (Generalized Stochastic Petri Net, [AMBC⁺95]), donde también aparecen arcos inhibidores y transiciones inmediatas. De esta forma, en una GSPN aparecen dos tipos de transiciones: transiciones *temporizadas* con tiempo de servicio exponencial (utilizadas para modelar esperas aleatorias asociadas a la ejecución de actividades), y transiciones *inmediatas*, que no llevan ninguna espera asociada (utilizadas para modelar la ejecución de acciones lógicas, que no conllevan tiempo). De esta forma, las transiciones inmediatas permiten introducir una ramificación probabilística, que es independiente de las especificaciones del tiempo. Cuando hay activadas a la vez transiciones temporizadas e inmediatas, siempre se disparan primero las inmediatas. También existe la posibilidad de definir distintos valores de prioridad sobre el conjunto de las transiciones inmediatas. Por consiguiente, en las GSPNs aparecen dos tipos de marcados: *tangibles* (donde las únicas transiciones activadas son temporizadas) y *fugaces* (donde hay alguna transición inmediata activada). Una comparativa entre SPN y GSPN puede encontrarse en [CDF91].

A continuación, pasamos a definir formalmente el modelo de red de Petri estocástica (markoviana) sobre el que nos hemos apoyado y que, como ya hemos comen-

tado, corresponde al modelo GSPN sin arcos inhibidores ni transiciones inmediatas, y donde siempre tendremos redes 1-seguras.

Definición 2.26 Una red de Petri estocástica es una 4-tupla de la forma

$$\Sigma = (S, T, W, \mu)$$

donde (S, T, W) es una red de Petri ordinaria y $\mu : T \rightarrow \mathbb{R}^+$ es la función que a cada transición le asocia un número real positivo, que corresponde al parámetro de la distribución exponencial negativa asociada a su tiempo de servicio ¹. De esta forma dicho parámetro es la inversa del tiempo medio de servicio de la transición. □

La regla de disparo es la misma que para el modelo básico de redes de Petri y, cuando dos o más transiciones están activadas se aplica una política de competencia. Como utilizamos funciones de probabilidad continua, tendremos que la probabilidad de que se disparen a la vez dos o más transiciones es nula, con lo que la interpretación de la evolución de la red corresponde a una semántica de *entremezcla*. El proceso estocástico generado es una CTMC. A partir del grafo de alcanzabilidad de la red de Petri estocástica se puede generar fácilmente el diagrama de transición de estados de la CTMC, donde cada marcado alcanzable será un estado, y cada par de estados estará unido por un arco etiquetado con las sumas de las razones de las transiciones que conectaban ambos marcados.

A partir de la CTMC se realiza el análisis cuantitativo del comportamiento de la red de Petri estocástica. Nos interesará sobre todo la distribución en el estado estacionario Π , y el análisis de prestaciones se realizará a través de los *índices de prestaciones*. Tales índices pueden expresarse de manera general como funciones de ganancia definidas sobre los marcados de la red, y a partir de ellas se calcula una ganancia media usando la distribución en estado estacionario. Si $r(M)$ representa una de estas funciones de ganancia, la ganancia media puede calcularse a partir de la suma ponderada:

$$R = \sum_{M_i \in RS(M_0)} r(M_i) \cdot \Pi_i$$

Por ejemplo, para calcular la probabilidad de que se dé una cierta propiedad en la red de Petri estocástica, se tiene que ver primero qué marcados cumplen esa

¹En realidad habría que considerar $\mu(t, M)$ para cada marcado alcanzable M , pero en nuestro modelo no es necesario considerar razones dependientes del marcado, por lo que hemos optado por esta definición más simple.

propiedad (llamaremos A a ese conjunto de marcados). Por consiguiente, la función de ganancia quedaría de la forma:

$$r(M) = \begin{cases} 1 & \text{si } M \in A \\ 0 & \text{en otro caso} \end{cases}$$

y la probabilidad deseada se calcularía como:

$$\sum_{M_i \in RS(M_0)} r(M_i) \cdot \Pi_i = \sum_{M_i \in A} \Pi_i$$

Otro ejemplo es el cálculo del *throughput* de una transición t_j (número medio de disparos por unidad de tiempo). Como una transición sólo puede dispararse si está activada, la función de ganancia es la siguiente, donde hemos considerado como A_M el conjunto de transiciones activadas en el marcado M :

$$r(M) = \begin{cases} \mu(t_j) & \text{si } t_j \in A_M \\ 0 & \text{en otro caso} \end{cases}$$

por tanto, el *throughput* de t_j sería:

$$\chi_j = \sum_{M_i \in RS(M_0)} r(M_i) \cdot \Pi_i = \sum_{M_i \in A_M} \mu(t_j) \cdot \Pi_i$$

Ejemplo 2.9 Consideremos un sistema simple de multiprocesador de memoria compartida [AMBC86], en el cual dos procesos deben acceder ocasionalmente a la memoria común compartida. Se asume que los dos procesos tienen igual comportamiento: una secuencia cíclica de actividades locales, seguida de una petición de acceso a la memoria compartida, terminando con un acceso a la memoria común. Suponemos también que todas esas acciones requieren un tiempo de ejecución distribuido exponencialmente. En la figura 2.15 se muestra la red de Petri estocástica que modela dicha arquitectura, donde consideraremos $\mu(t_i) = r_i$, $i = 1, \dots, 6$ como las razones asociadas a cada transición. El marcado de p_1 (p_2) indica que el primer (segundo) procesador está activo.

Un posible comportamiento de la red sería el siguiente: El procesador primero (segundo) trabaja localmente en un tiempo distribuido exponencialmente de parámetro r_1 (r_2) y, a continuación solicita acceder a la memoria común. Cuando se dispara t_1 (t_2), el token de p_1 (p_2) pasa a p_3 (p_4), lo que significa que el primer (segundo) procesador está esperando para acceder a la memoria común. Si p_7 está marcado tendríamos que la memoria común está libre y, la petición de la memoria

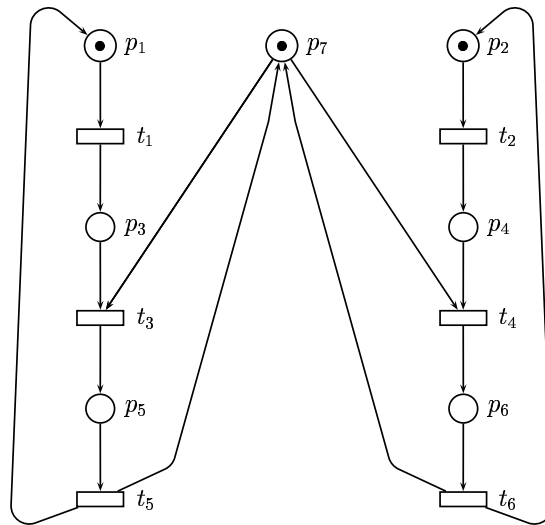


Figura 2.15: Red de Petri que modela un multiprocesador de memoria compartida

común del primer (segundo) procesador que modela t_3 (t_4), que sigue una distribución exponencial de parámetro r_3 (r_4), puede ser atendida. Cuando se dispara t_3 (t_4), el token de p_3 (p_4) pasa a p_5 (p_6), lo que significa que el primer (segundo) procesador está accediendo a la memoria común, donde se permanece un tiempo medio $1/r_5$ ($1/r_6$). Cuando ese tiempo transcurre, se dispara t_5 (t_6) y se vuelve al marcado inicial.

De esta forma, los ciclos de actividad de los dos procesadores se entrelazan, quedando descritos mediante el grafo de alcanzabilidad de la red mostrado en la figura 2.16.

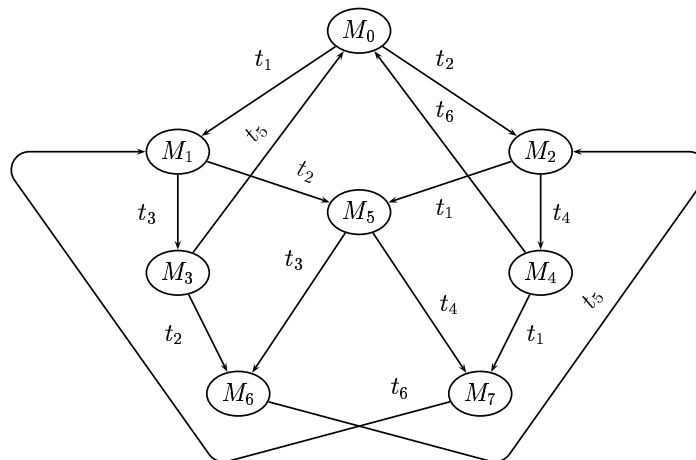


Figura 2.16: Grafo de accesibilidad de la red que modela el multiprocesador de memoria compartida

donde

$$\begin{aligned}
 M_0 &= p_1 + p_2 + p_7 & M_1 &= p_3 + p_2 + p_7 \\
 M_2 &= p_1 + p_4 + p_7 & M_3 &= p_5 + p_2 \\
 M_4 &= p_1 + p_6 & M_5 &= p_3 + p_4 + p_7 \\
 M_6 &= p_5 + p_4 & M_7 &= p_3 + p_6
 \end{aligned}$$

Cuando se alcanza el mercado M_5 , en el cual los dos procesadores quieren acceder a la memoria común, se aplicará una política de competencia, con lo que la probabilidad de que el primer (segundo) procesador acceda a la memoria común, esto es, gane la carrera, es $\frac{r_3}{r_3+r_4}$ ($\frac{r_4}{r_3+r_4}$). Cuando las dos transiciones t_3 y t_4 están activadas a la vez, la velocidad con la que el sistema sale del mercado M_5 es $r_3 + r_4$, esto es, permanece en ese mercado un tiempo medio de $\frac{1}{r_3+r_4}$. Su correspondiente CTMC es la mostrada en la figura 2.17.

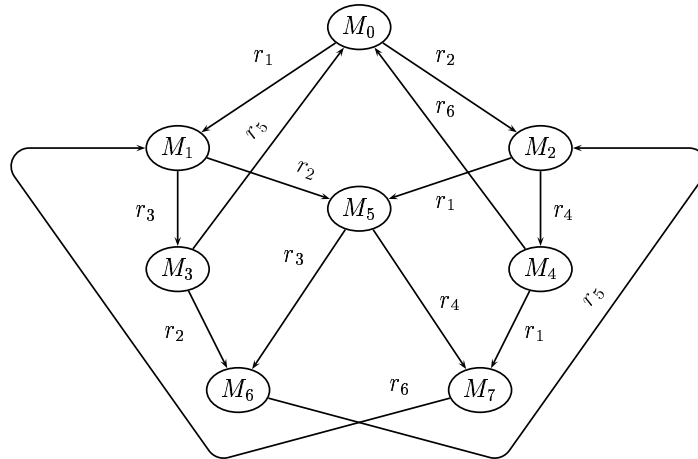


Figura 2.17: CTMC de la red que modela el multiprocesador de memoria compartida

Su matriz generadora infinitesimal es:

$$Q = \begin{pmatrix}
 -(r_1 + r_2) & r_1 & r_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & -(r_2 + r_3) & 0 & r_3 & 0 & r_2 & 0 & 0 & 0 \\
 0 & 0 & -(r_1 + r_4) & 0 & r_4 & r_1 & 0 & 0 & 0 \\
 r_5 & 0 & 0 & -(r_2 + r_5) & 0 & 0 & r_2 & 0 & 0 \\
 r_6 & 0 & 0 & 0 & -(r_1 + r_6) & 0 & 0 & 0 & r_1 \\
 0 & 0 & 0 & 0 & 0 & -(r_3 + r_4) & r_3 & r_4 & 0 \\
 0 & 0 & r_5 & 0 & 0 & 0 & -r_5 & 0 & 0 \\
 0 & r_6 & 0 & 0 & 0 & 0 & 0 & 0 & -r_6
 \end{pmatrix}$$

Como la red de Petri es limitada y tiene un estado de acogida, la CTMC asociada es ergódica, y la distribución de estados estacionario se calcularía resolviendo el

sistema de ecuaciones lineales

$$\begin{cases} (\Pi_0, \Pi_1, \dots, \Pi_7) Q &= (0, 0, \dots, 0) \\ \Pi_0 + \Pi_1 + \dots + \Pi_7 &= 1 \end{cases}$$

De esta forma, si se quiere conocer, por ejemplo, la *potencia de procesamiento* P del sistema, esto es, el número medio de procesadores trabajando localmente (sin acceder a la memoria común), considerando $A = \{M_0, M_1, M_2, M_3, M_4\}$ la función de ganancia se puede definir como:

$$\begin{aligned} r(M_0) &= 2, \quad r(M_1) = r(M_2) = r(M_3) = r(M_4) = 1 \\ r(M_i) &= 0 \quad \text{si } M_i \notin A \end{aligned}$$

y así:

$$P = 2 \cdot \Pi_0 + \Pi_1 + \Pi_2 + \Pi_3 + \Pi_4$$

□

2.2.3 Álgebras de Procesos estocásticas

La introducción de aspectos temporales en las álgebras de procesos se ha realizado de diversas formas, tanto considerando el tiempo continuo (álgebras con *tiempo real*) o discreto. También la ejecución de las acciones y el paso del tiempo pueden considerarse juntas o separadas. A finales de los 80 y en los 90 aparecieron las principales extensiones temporales de las álgebras de procesos. Algunas de las más conocidas son las extensiones de las álgebras CSP [RR83, NS91, Sch95, LdFN96], CCS [MT90, Yi90, HR91], ACP [BB93] y LOTOS [QdFML94, LL97]. También aparecieron las álgebras de procesos probabilísticas, en las que la idea consistió en extender el operador de elección incorporando algún tipo de información probabilística. Algunos trabajos sobre modelos probabilísticos pueden encontrarse en [GJS90, vGSST90, NdFL95, CFV96, CFV97] y con elecciones probabilísticas y no deterministas en [YL92, MMSS96, And99, Caz01, CCFP03]. Pero también tenemos los modelos que consideran tiempos deterministas y elecciones probabilísticas, como [Han91].

Más recientemente, la necesidad de modelar sistemas en los que las acciones conllevan retrasos cuya duración está definida de un modo probabilístico ha provocado la aparición de las álgebras de procesos estocásticas. En este marco, la primera referencia que se conoce es la de Götz, Herzog y Rettelbach en [GHR92]. Considerando ya que el tiempo de retraso es una variable aleatoria que se rige por una distribución exponencial negativa, aparecen las álgebras de procesos markovianas, conocidas

también como álgebras de procesos estocásticas (SPA, Stochastic Process Algebra). Podemos citar TIPP [HR94] (TImed Process and Performance analysis) como continuación del primer trabajo; el modelo de Hillston, PEPA [Hil94a] (Performance Evaluation Process Algebra), sobre el cual nos hemos inspirado principalmente al desarrollar sPBC; MPA (Markovian Process Algebra) [Buc94] de Bucholz, o modelos más sofisticados como EMPA (Extended Markovian Process Algebra)[BGD94] de Bernardo, o ROSA [PCVC00] de L. Pelayo. En todos los modelos anteriores la información estocástica está incluida en las acciones; así, una expresión de la forma $(a, r); P$ indica que la probabilidad de ejecutar a antes de que transcurra un tiempo t se rige por una distribución de probabilidad exponencial de parámetro r (salvo para MPA, donde cada acción tiene asignado siempre el mismo parámetro y r indica las ocurrencias concurrentes de a). Después de ejecutar a el proceso pasa a comportarse como P . La principal diferencia entre ellas es la forma de considerar la sincronización. En [Hil94b] se estudian y comparan las diferentes alternativas al respecto.

Existen otros modelos alternativos como [Her02] de Hermanns, donde la información estocástica (markoviana) del tiempo viene separada del resto de acciones. Así, una expresión de la forma $r; P$ indica que el proceso se retrasa de acuerdo a una distribución exponencial de parámetro r y luego pasa a comportarse como P .

En todos los casos citados se adopta la *política de competencia* (*race condition*): en el caso de que sea posible ejecutar varias acciones estocásticas, ganará la carrera la más rápida. Para el estudio cuantitativo del proceso generado se puede recurrir a la CTMC que tendrá asociada la evolución de dicho proceso.

Un estudio teórico sobre los fundamentos de las álgebras de procesos markovianas puede encontrarse en [HHK02]. Últimamente también han comenzado a estudiarse álgebras de procesos no markovianas, como puede verse en [Lóp03].

Aunque existen trabajos previos que consideran la traslación de las álgebras de procesos estocásticas a redes de Petri estocásticas [Rib95, BDG95], e incluso al contrario [HRRS01], nuestro punto de partida es distinto, ya que pretendemos obtener una extensión markoviana de PBC, que conserve las principales características del modelo original, teniendo así multiacciones a un primer nivel y un operador de sincronización independiente del operador paralelo.

A continuación, presentaremos con un poco más de detalle el lenguaje PEPA [Hil96], que es el más afín al modelo propuesto en esta tesis. Después se pasará a

comentar brevemente EMPA [BG98] y TIPP [HR94].

PEPA

En este álgebra de procesos estocástica un sistema se define como un conjunto de componentes, donde cada componente puede realizar un conjunto de *actividades*. Cada actividad a se representa por un par (α, r) , donde α es el *tipo* de la actividad (que corresponde a una sola acción) y r es el parámetro de la exponencial negativa que gobierna su duración. Dicho parámetro puede quedar sin especificar, denotándose mediante el símbolo \top , ($r \in \mathbb{R}^+ \cup \{\top\}$). También el tipo de acción puede quedar sin especificar (se considera desconocida o irrelevante), lo que se indica con el símbolo τ .

La sintaxis de PEPA es la definida a continuación:

$$P ::= (\alpha, r).P \mid P \underset{L}{\bowtie} Q \mid P + Q \mid P/L \mid X \mid A$$

La interpretación intuitiva de los operadores es la siguiente:

PREFIJO (\cdot): $(\alpha, r).P$ indica que el tipo de acción α se puede ejecutar después de un retraso (*delay*) gobernado por una distribución exponencial de parámetro r (de media $1/r$). La componente pasa a comportarse posteriormente como P . Si $a = (\alpha, r)$, la componente $(\alpha, r).P$ puede escribirse como $a.P$.

ELECCIÓN ($+$): La componente $P + Q$ representa un sistema que puede comportarse como P o como Q . $P + Q$ habilita todas las actividades actuales de P y todas las actividades actuales de Q . La primera actividad que se complete permite distinguir (elegir) una de los componentes de la elección. La otra componente es descartada. La naturaleza continua de las distribuciones de probabilidad asegura que la probabilidad de que P y Q completen una actividad al mismo tiempo es cero. El sistema se comportará posteriormente como P' o Q' , donde P' es la componente que resulta de completar una actividad de P , y Q' es la componente que resulta de completar una actividad de Q .

COOPERACIÓN ($\underset{L}{\bowtie}$): El operador de cooperación es en realidad una familia indexada de operadores, uno por cada posible conjunto L de tipos de acciones. A L se le conoce como el *conjunto de cooperación*, y define los tipos de acciones sobre los que los procesos argumento deben sincronizar o cooperar, es decir, determina la

interacción entre componentes. El tipo de acción desconocido no puede aparecer en L . Todas las actividades de los procesos argumento que no aparecen en el conjunto de cooperación L reciben el nombre de *individuales*, y no se ven afectadas por la cooperación. Las actividades que aparecen en L reciben el nombre de *compartidas* y sólo pueden llevarse a cabo cuando están habilitadas en ambas componentes. Por lo tanto, una componente puede verse bloqueada hasta que la otra esté preparada para participar (*rendez-vous*). De la cooperación de ambas componentes nace una nueva actividad compartida, cuyo tipo es el mismo que el de las dos actividades que sincronizan, y la nueva razón de su correspondiente distribución exponencial atiende a un mínimo ponderado de sus *razones aparentes*, que intuitivamente puede interpretarse como que se espera a la más lenta. Decimos intuitivamente porque, si realmente esperásemos a la más lenta, tendríamos el máximo de dos variables aleatorias con distribución exponencial, que no es una variable aleatoria con distribución exponencial, con lo que no tendríamos un modelo markoviano.

Si una actividad tiene una razón no especificada en una componente, se dice que la componente es *pasiva* con respecto a ese tipo de acción. Esto significa que, aunque la cooperación del componente es necesaria para realizar la acción, la componente no contribuye al trabajo realizado.

OCULTAMIENTO ($/$): La componente P/L se comporta como P salvo que los tipos de acciones del conjunto L están *ocultos*. Por lo tanto, estos tipos de acciones se reemplazan por el tipo desconocido τ , lo que puede ser interpretado como un retraso interno de la componente.

La duración de una acción no se ve afectada si es ocultada. Sin embargo, una acción ocultada no puede utilizarse para cooperar con otra componente. En realidad, el tipo de acción τ no es visible para un observador o componente externo.

VARIABLE (X): Si E es una componente que contiene la variable X , entonces $E[P/X]$ representa la componente resultado de reemplazar en E cada ocurrencia de X por la componente P .

CONSTANTE ($A \stackrel{def}{=} P$): Una constante es una componente cuyo significado está dado mediante una definición del tipo $A \stackrel{def}{=} P$, que asigna a la constante A el comportamiento de la componente P . Permite asignar nombres a componentes (comportamientos).

La semántica operacional del lenguaje se muestra en la tabla 2.6. En una de las

reglas relativas al operador de cooperación aparece el término $r_\alpha(E)$, que es la razón aparente del tipo de acción α en E ; este valor se calcula como la suma de todas las acciones de tipo α que están habilitadas en un instante en P . Formalmente, la definición sería la siguiente.

Definición 2.27 La *razón aparente* de un tipo de acción α en una componente P , denotada por $r_\alpha(P)$, se define como:

1. $r_\alpha((\beta, r).P) = \begin{cases} r & \text{si } \beta = \alpha \\ 0 & \text{si } \beta \neq \alpha \end{cases}$
2. $r_\alpha(P + Q) = r_\alpha(P) + r_\alpha(Q)$
3. $r_\alpha(P/L) = \begin{cases} r_\alpha(P) & \text{si } \alpha \notin L \\ 0 & \text{si } \alpha \in L \end{cases}$
4. $r_\alpha(P \underset{L}{\bowtie} Q) = \begin{cases} \min(r_\alpha(P), r_\alpha(Q)) & \text{si } \alpha \in L \\ r_\alpha(P) + r_\alpha(Q) & \text{si } \alpha \notin L \end{cases}$

□

PEPA dispone de una herramienta propia: PEPA Workbench [GH94]. En [PRC01] puede encontrarse una breve referencia al álgebra y a la herramienta.

EMPA

En este álgebra de procesos estocástica se definen varias clases de acciones, según el tiempo asignado a las mismas. Así se tienen:

- Acciones de *tiempo exponencial*, denotadas por (a, r) , donde r es el parámetro de la distribución exponencial asociada al retraso que sigue la ejecución de la acción.
- Acciones *inmediatas*, denotadas por $(a, \infty_{l,w})$, donde l representa el nivel de prioridad y w el peso asociado a la misma. Estas acciones se consideran urgentes, en el sentido de que se deben ejecutar tan pronto como estén disponibles.
- Acciones *pasivas*, denotadas por $(a, *)$. Estas acciones pueden esperar tanto como sea necesario antes de ejecutarse. Su duración se fija a través de una sincronización con una acción $(a, \infty_{l,w})$ ó (a, r) .

Prefijo

$$\overline{(\alpha, r).E \xrightarrow{(\alpha, r)} E}$$

Cooperación

$$\frac{E \xrightarrow{(\alpha, r)} E'}{E \underset{L}{\bowtie} F \xrightarrow{(\alpha, r)} E' \underset{L}{\bowtie} F} \quad (\alpha \notin L)$$

$$\frac{F \xrightarrow{(\alpha, r)} F'}{E \underset{L}{\bowtie} F \xrightarrow{(\alpha, r)} E \underset{L}{\bowtie} F'} \quad (\alpha \notin L)$$

$$\frac{E \xrightarrow{(\alpha, r_1)} E' \quad F \xrightarrow{(\alpha, r_2)} F'}{E \underset{L}{\bowtie} F \xrightarrow{(\alpha, R)} E' \underset{L}{\bowtie} F'} \quad (\alpha \in L)$$

$$\text{donde } R = \frac{r_1}{r_\alpha(E)} \frac{r_2}{r_\alpha(F)} \min(r_\alpha(E), r_\alpha(F))$$

Elección

$$\frac{E \xrightarrow{(\alpha, r)} E'}{E + F \xrightarrow{(\alpha, r)} E'}$$

$$\frac{F \xrightarrow{(\alpha, r)} F'}{E + F \xrightarrow{(\alpha, r)} F'}$$

Ocultamiento

$$\frac{E \xrightarrow{(\alpha, r)} E'}{E/L \xrightarrow{(\alpha, r)} E'/L} \quad (a \notin L)$$

$$\frac{E \xrightarrow{(\alpha, r)} E'}{E/L \xrightarrow{(\tau, r)} E'/L} \quad (a \in L)$$

Constante

$$\frac{E \xrightarrow{(\alpha, r)} E'}{A \xrightarrow{(\alpha, r)} E'} \quad (A \stackrel{def}{=} E)$$

Tabla 2.6: Semántica operacional de PEPA

Las elecciones se resuelven de la forma siguiente:

- Si se tiene una elección entre acciones exponenciales, se atiende a la política de competencia: gana la más rápida.
- Cuando tenemos una elección entre acciones inmediatas con distinto nivel de prioridad, se ejecutará la que tenga asignada un mayor nivel de prioridad.
- Cuando tenemos una elección entre acciones inmediatas con el mismo nivel de prioridad, entonces la elección se resuelve probabilísticamente, teniendo en cuenta los pesos.
- Si se tiene una elección entre acciones pasivas, ésta será no determinista.
- Para el resto de elecciones, las inmediatas ganan frente a las exponenciales, y éstas sobre las pasivas.

Los demás operadores son del estilo de PEPA, salvo el operador de sincronización. Para definir la semántica de dicho operador se impone la condición de que una de las dos acciones debe ser pasiva. Así, se puede sincronizar una pasiva $(a, *)$ con una inmediata $(a, \infty_{l,w})$, proporcionando $(a, \infty_{l,w})$; o bien, una pasiva $(a, *)$ con una exponencial (a, r) , generando (a, r) .

Two Towers [BCSS98] es la herramienta de este álgebra de procesos.

TIPP

Este álgebra de procesos estocástica es muy similar a PEPA. Así la sintaxis de TIPP es la mostrada a continuación:

$$P ::= 0 \mid (\alpha, r).P \mid P \parallel_L Q \mid P + Q \mid X \mid A$$

donde $\alpha \in Act = Com \cup \{\tau\}$, representando Com el conjunto de acciones visibles y τ la acción no visible; el valor $r \in \mathbb{R}^+$, denota la razón de la distribución exponencial asociada con el retraso de la ejecución de (α, r) ; $L \in Com$ representa el conjunto de acciones sobre los que se sincronizan P y Q en $P \parallel_L Q$. En esta sencilla álgebra de procesos, cabe resaltar que dos acciones (a, r_1) y (a, r_2) se pueden sincronizar originando una nueva acción: $(a, r_1 \cdot r_2)$. De esta forma, aquellas acciones con razón 1 se pueden considerar pasivas. Los demás operadores tienen una interpretación análoga a la vista en PEPA.

También se dispone aquí de herramienta propia [HHMS00].

Al final del capítulo 4, en el apartado 4.4.1, se hace una comparación entre nuestra propuesta de sincronización y las propuestas ofrecidas por estas álgebras de procesos estocásticas clásicas.

Capítulo 3

sPBC finito. Una primera aproximación

En este capítulo presentaremos una primera versión de sPBC finito. Esta primera aproximación se mejorará con una nueva propuesta del operador de sincronización, la cual se estudiará en el capítulo siguiente, y con un nuevo operador, el de iteración, que será presentado en el capítulo 5. De esta forma, la sintaxis, semántica operacional y denotacional de nuestro modelo quedan completamente descritas a lo largo de dichos tres capítulos.

3.1 Sintaxis

En este apartado introduciremos la sintaxis del álgebra de procesos estocástica en el caso finito, esto es, sin iteración ni recursión. Nos seguiremos refiriendo a ella en capítulos posteriores.

3.1.1 Notación

Consideraremos:

- \mathcal{A} como un conjunto numerable de nombres de acciones. Además, para todo $a \in \mathcal{A}$, existe su *conjugado*, $\hat{a} \in \mathcal{A}$, verificando $a \neq \hat{a}$ y $\widehat{\hat{a}} = a$, como en CCS [Mil89]. Utilizaremos las letras a, b, \hat{a}, \dots para denotar los elementos de \mathcal{A} .

- $\mathcal{L} = \mathcal{B}(\mathcal{A})$, donde $\mathcal{B}(\mathcal{A})$ representa el conjunto de multiconjuntos finitos de \mathcal{A} . Si $\alpha \in \mathcal{L}$, $\alpha(a)$ indicará el número de veces que aparece a en α . Cuando α esté formado por un sólo elemento a de \mathcal{A} , podremos utilizar la notación simplificada de modo que en vez de denotar α por $\{a\}$ lo haremos simplemente por a . En particular, tendremos que $\emptyset \in \mathcal{L}$, que se utiliza para especificar la ejecución de una multiacción que no contiene nombres de acciones visibles. La llamaremos multiacción no observable, o también en la terminología de CCS, silenciosa o interna.
- El *alfabeto* de una multiacción $\alpha \in \mathcal{L}$ se define como:

$$A(\alpha) =_{def} \{a \in \mathcal{A} \mid \alpha(a) > 0\}$$

- Funciones de renombramiento $f : \mathcal{A} \rightarrow \mathcal{A}$, que conservan los conjugados:

$$\forall a \in \mathcal{A}, f(\widehat{a}) = \widehat{f(a)}$$

- Definimos el conjunto de multiacciones estocásticas como:

$$\mathcal{SL} =_{def} \{\langle \alpha, r \rangle \mid \alpha \in \mathcal{L} \quad y \quad r \in \mathbb{R}^+\}$$

Se permite que una multiacción $\alpha \in \mathcal{L}$ tenga razones diferentes en la misma especificación.

- Definimos el alfabeto de una multiacción estocástica como el alfabeto de su correspondiente multiacción:

$$A(\langle \alpha, r \rangle) =_{def} A(\alpha)$$

Esta definición puede ser fácilmente generalizada a los elementos de $\mathcal{B}(\mathcal{SL})$: dada $\gamma = \{ \langle \alpha_1, r_1 \rangle, \dots, \langle \alpha_n, r_n \rangle \} \in \mathcal{B}(\mathcal{SL})$, se define el alfabeto de γ como:

$$A(\gamma) =_{def} \cup_{\langle \alpha_i, r_i \rangle \in \gamma} A(\langle \alpha_i, r_i \rangle)$$

A continuación, introducimos algunas operaciones sobre multiacciones:

- **suma**

– *de multiacciones:*

$$\alpha + \beta =_{def} \delta, \text{ donde } \delta(a) =_{def} \alpha(a) + \beta(a)$$

– de multiconjuntos de multiacciones estocásticas:

$$\begin{aligned} & \{ \langle \alpha_1, r_1 \rangle, \dots, \langle \alpha_n, r_n \rangle \} + \{ \langle \beta_1, s_1 \rangle, \dots, \langle \beta_n, s_n \rangle \} =_{def} \\ & \{ \langle \alpha_1, r_1 \rangle, \dots, \langle \alpha_n, r_n \rangle, \langle \beta_1, s_1 \rangle, \dots, \langle \beta_n, s_n \rangle \} \end{aligned}$$

• **sincronización**

– de multiacciones:

$\alpha \oplus_a \beta =_{def} \gamma$, donde:

$$\gamma(b) =_{def} \begin{cases} \alpha(b) + \beta(b) - 1 & \text{si } b = a \text{ ó } b = \hat{a} \\ \alpha(b) + \beta(b) & \text{en otro caso} \end{cases}$$

y sólo es aplicable si $a \in A(\alpha)$ y $\hat{a} \in A(\beta)$, o viceversa.

– de multiacciones estocásticas:

$$\langle \alpha, r_1 \rangle \oplus_a \langle \beta, r_2 \rangle =_{def} \langle \alpha \oplus_a \beta, R \rangle$$

donde, en esta primera aproximación, $R = \min\{r_1, r_2\}$. En todo caso, sólo es aplicable si $a \in A(\alpha)$ y $\hat{a} \in A(\beta)$, o viceversa.

• **renombramiento**

– de multiacciones: $f(\alpha) =_{def} \sum_{a \in \mathcal{A}} \alpha(a) f(a)$

– de multiconjuntos de multiacciones estocásticas:

$$f(\{ \langle \alpha_1, r_1 \rangle, \dots, \langle \alpha_n, r_n \rangle \}) =_{def}$$

$$\{ \langle f(\alpha_1), r_1 \rangle, \dots, \langle f(\alpha_n), r_n \rangle \}$$

El conjunto de operadores de sPBC finito vendrá dado por:

$$op =_{def} \{ ;, \square, \parallel \} \cup \{ [f] \} \cup \{ sy a, rs a, [a :] \}$$

3.1.2 s-Expresiones Estáticas

Las s-expresiones estáticas se utilizarán para describir la estructura de un sistema concurrente, mientras que las s-expresiones dinámicas describirán el estado actual del sistema. Al igual que en PBC plano, las primeras corresponderán más tarde a redes de Petri sin marcar y las segundas a redes de Petri marcadas. Como un sistema evoluciona ejecutando multiacciones, las s-expresiones dinámicas capturarán sus

cambios de estados por medio de los barrados y sub-barrados sobre las s-expresiones estáticas.

Las *s-expresiones estáticas* de sPBC finito son aquéllas definidas por medio de la siguiente notación BNF:

$$E ::= \langle \alpha, r \rangle \mid E; E \mid E \square E \mid E \parallel E \mid E[f] \mid E \text{ sy } a \mid E \text{ rs } a \mid [a : E]$$

donde $[a : E] = (E \text{ sy } a) \text{ rs } a$.

La única novedad con respecto a PBC plano es que se utiliza un tipo diferente de expresión constante $\langle \alpha, r \rangle$, que llamaremos multiacción estocástica (básica), donde $\alpha \in \mathcal{L}$ (esto es, α es una multiacción de PBC), y $r \in \mathbb{R}^+$. La ejecución de $\langle \alpha, r \rangle$ representa la ejecución simultánea de todas las acciones de α , después de un retraso (*delay*) que sigue una distribución exponencial negativa con parámetro r .

Para denotar las s-expresiones estáticas utilizaremos las letras: E, F, E_i, \dots . Como ya hemos comentado, las s-expresiones estáticas describen las características estructurales de los sistemas concurrentes. Su semántica operacional se define utilizando s-expresiones dinámicas, las cuales se definen a continuación.

3.1.3 s-Expresiones Dinámicas

La semántica operacional de sPBC se define por medio de las s-expresiones dinámicas, las cuales se derivan de las s-expresiones estáticas, anotando las mismas con barrados y sub-barrados, para indicar las *componentes activas*. Así, las *s-expresiones dinámicas* de sPBC finito son aquéllas definidas por medio de la siguiente notación BNF:

$$G ::= \overline{E} \mid \underline{E} \mid G; E \mid E; G \mid G \square E \mid E \square G \mid G \parallel G \mid G[f] \mid G \text{ sy } a \mid G \text{ rs } a \mid [a : G]$$

La interpretación informal es exactamente la misma que en PBC:

- \overline{E} indicará que la s-expresión estática E acaba de ser activada y puede comenzar a ejecutar sus multiacciones estocásticas. Representa el estado inicial de E .
- \underline{E} indicará que la s-expresión estática E ha ejecutado todas las multiacciones estocásticas que le correspondían, de modo que no puede ejecutar más mul-

tiaciones estocásticas. La llamaremos final y representará un posible estado final de E .

- $[a : G] = (G \text{ sy } a) \text{ rs } a$.

Para denotar las s-expresiones dinámicas utilizaremos las letras: G, H, G_i, \dots , y el conjunto de s-expresiones dinámicas lo llamaremos *DynExpr*.

3.2 Semántica Operacional

La semántica operacional de sPBC finito se define de una forma muy similar a la semántica operacional de PBC [BDK01a, BDK98, BDK01b, BK95], pero existen algunas diferencias en lo referente a las reglas de transición de multiacciones (estocásticas), siendo la más importante la que se deriva de tener ahora una semántica de orden total, con lo que satisface la Suposición de Orden Total (*Total Orden Assumption, TOA*) [Bae92]:

" Toda ejecución de secuencias de multiacciones observables están totalmente ordenadas por precedencia ",

al contrario de lo que pasaba en PBC, donde se tenía una semántica de orden parcial (*true concurrency*).

3.2.1 Reglas de inacción y equivalencia estructural

Las transiciones de inacción se introducen en PBC para llegar a establecer las componentes activas de una expresión dinámica. Por ejemplo, una regla de inacción para el operador paralelo de PBC es:

$$\overline{E} \parallel \overline{F} \xrightarrow{\emptyset} \overline{E} \parallel \overline{F}$$

que significa que para activar la composición paralela se tiene que activar cada uno de sus componentes. Estas transiciones no corresponden a ningún cambio dinámico de estado, y se consideran reversibles, lo cual se captura introduciendo la transición inversa, $\xleftarrow{\emptyset}$.

Las reglas de inacción para sPBC son las definidas en las tablas 3.1 y 3.2.

$\overline{E}; \overline{F} \xrightarrow{\emptyset} \overline{E}; F$	$\underline{E}; F \xrightarrow{\emptyset} E; \overline{F}$	$E; \underline{F} \xrightarrow{\emptyset} \underline{E}; F$
$\overline{E} \square \overline{F} \xrightarrow{\emptyset} \overline{E} \square F$	$\overline{E} \square \overline{F} \xrightarrow{\emptyset} E \square \overline{F}$	$\underline{E} \square F \xrightarrow{\emptyset} \underline{E} \square F$
$E \square \underline{F} \xrightarrow{\emptyset} \underline{E} \square F$	$\overline{E} \parallel \overline{F} \xrightarrow{\emptyset} \overline{E} \parallel \overline{F}$	$\underline{E} \parallel \underline{F} \xrightarrow{\emptyset} \underline{E} \parallel F$
$\overline{E}[f] \xrightarrow{\emptyset} \overline{E}[f]$	$\underline{E}[f] \xrightarrow{\emptyset} \underline{E}[f]$	$\overline{E} \text{ sy } a \xrightarrow{\emptyset} \overline{E} \text{ sy } a$
$\underline{E} \text{ sy } a \xrightarrow{\emptyset} \underline{E} \text{ sy } a$	$\overline{E} \text{ rs } a \xrightarrow{\emptyset} \overline{E} \text{ rs } a$	$\underline{E} \text{ rs } a \xrightarrow{\emptyset} \underline{E} \text{ rs } a$

Tabla 3.1: Reglas de inacción (I)

$\frac{\forall op \in \{;, \square\}, G \xrightarrow{\emptyset} G'}{op(G, E) \xrightarrow{\emptyset} op(G', E)}$	$\frac{\forall op \in \{;, \square\}, G \xrightarrow{\emptyset} G'}{op(E, G) \xrightarrow{\emptyset} op(E, G')}$
$\frac{G_1 \xrightarrow{\emptyset} G'_1}{G_1 \parallel G_2 \xrightarrow{\emptyset} G'_1 \parallel G_2}$	$\frac{G_2 \xrightarrow{\emptyset} G'_2}{G_1 \parallel G_2 \xrightarrow{\emptyset} G_1 \parallel G'_2}$
$\frac{\forall op \in \{\text{sy}, \text{rs}\}, G \xrightarrow{\emptyset} G'}{op(G, a) \xrightarrow{\emptyset} op(G', a)}$	$\frac{G \xrightarrow{\emptyset} G'}{G[f] \xrightarrow{\emptyset} G'[f]}$

Tabla 3.2: Reglas de inacción (II)

Es de destacar que en nuestro caso la dirección de la flecha resultará importante, pues nos permitirá definir una clase de s-expresiones dinámicas "canónicas", que llamaremos *s-expresiones dinámicas operativas*. Como vamos a definir una semántica de orden total, si nos apoyáramos sólo en las s-expresiones dinámicas nos encontraríamos con algunas dificultades técnicas a la hora de definir la semántica de la sincronización. Por ejemplo:

- $\overline{E} = \overline{\langle a, r_1 \rangle \square \langle \hat{a}, r_2 \rangle}$ puede realizar, en un sólo paso, la multiacción estocástica $\langle a, r_1 \rangle$, ó la multiacción estocástica $\langle \hat{a}, r_2 \rangle$, y terminar tras ello.
- $\overline{F} = \overline{\langle a, r_1 \rangle \parallel \langle \hat{a}, r_2 \rangle}$ también puede realizar, en un sólo paso, la multiacción estocástica $\langle a, r_1 \rangle$, ó la multiacción estocástica $\langle \hat{a}, r_2 \rangle$, pero luego podría realizar la otra y terminaría.

Obsérvese que $\overline{E} \text{ sy } a$ no puede hacer nada que no pudiese hacer \overline{E} , sin embargo $\overline{F} \text{ sy } a$ puede, además de hacer lo mismo que \overline{F} , realizar la multiacción estocástica

$\langle \emptyset, R \rangle$. Para distinguir estos comportamientos, introducimos las bolsas concurrentes (capturan aquellas bolsas de multiacciones que en PBC ya teníamos de forma directa al considerar una semántica de *conurrencia real*). De esta forma, para \overline{F} tendremos la bolsa concurrente $\{\langle a, r_1 \rangle, \langle \hat{a}, r_2 \rangle\}$, pero para \overline{E} habría que distinguir la multiacción que está activada (según la regla de inacción que se hubiera aplicado). Por lo tanto, las bolsas concurrentes no se pueden definir directamente sobre las s-expresiones dinámicas, y habrá que hacerlo sobre las s-expresiones dinámicas llamadas *operativas* (aquéllas en las que ya no es posible aplicar más reglas de inacción). En consecuencia, nos limitaremos a estas s-expresiones dinámicas operativas, sin que ello suponga ninguna restricción, ya que cada clase de s-expresiones dinámicas que consideraremos equivalentes, contendrá al menos una s-expresión dinámica operativa.

Definición 3.1 Sean G y G' dos s-expresiones dinámicas cualesquiera, de forma que $G \xrightarrow{\emptyset} G'$, entonces decimos que G' es una *derivación inmediata* (en un paso) de G . De forma general, si $G \xrightarrow{\emptyset} G_1 \xrightarrow{\emptyset} G_2 \xrightarrow{\emptyset} \dots \xrightarrow{\emptyset} G'$ diremos que G' es una *derivación* de G . \square

Definición 3.2 Una s-expresión dinámica G diremos que es *operativa* si no tiene ninguna derivación.

Al conjunto de todas las s-expresiones dinámicas operativas lo denotaremos por $OpDynExpr$. \square

A partir de una s-expresión dinámica G no operativa, siempre se puede obtener una derivación suya operativa, que no tiene porqué ser única. Veamos un ejemplo:

Ejemplo 3.1 Sea $G_1 = \overline{\langle \{a\}, r_1 \rangle \square \langle \{b\}, r_2 \rangle}$. A partir de ella podemos obtener dos derivaciones inmediatas operativas de G_1 :

$$G_1 \xrightarrow{\emptyset} \overline{\langle \{a\}, r_1 \rangle} \square \langle \{b\}, r_2 \rangle = G_{11}$$

$$G_1 \xrightarrow{\emptyset} \langle \{a\}, r_1 \rangle \square \overline{\langle \{b\}, r_2 \rangle} = G_{12}$$

Por lo tanto, G_{11} y G_{12} son operativas, pues desde ellas no se puede aplicar ninguna regla de inacción. \square

Aunque la dirección de la flecha juega un papel importante para obtener las operativas, esto no significa que las s-expresiones dinámicas que se relacionan mediante

las reglas de inacción dejen de ser equivalentes. Por ello, consideramos la dirección inversa $\xleftarrow{\emptyset}$ para definir la relación de equivalencia estructural \equiv sobre s-expresiones dinámicas de sPBC finito, que tiene por objetivo capturar los comportamientos operacionales iguales a partir de representaciones sintácticas distintas, manteniendo la relación de equivalencia estructural de PBC [BDK01a] y de tPBC [Kou00].

Definición 3.3 Entre las s-expresiones dinámicas de PBC finito, se define la relación de equivalencia:

$$\equiv =_{def} (\xrightarrow{\emptyset} \cup \xleftarrow{\emptyset})^*$$

que llamaremos relación de equivalencia estructural.

Por $[G]_{\equiv}$ entenderemos la clase de equivalencia a la que pertenece G. □

3.2.2 Reglas de transición estocásticas

Se define la semántica operacional por medio de reglas, de las cuales se infieren transiciones de la forma:

$$G \xrightarrow{\langle \alpha, r \rangle} G' \quad \text{con } \langle \alpha, r \rangle \in \mathcal{SL},$$

que capturan todas las posibles evoluciones de cada s-expresión dinámica operativa. Corresponden a la ejecución de una multiacción α , cuya espera (*delay*) hasta ejecutarse es aleatoria, estando definida mediante una variable aleatoria que sigue una distribución exponencial (negativa) de parámetro r . Además, como veremos después, la probabilidad de que se ejecute dicha transición vendrá determinada por una condición de competencia.

En consecuencia, aplicando repetidamente estas reglas, se obtendrán secuencias de transiciones, que corresponderán a las secuencias de transiciones de la red de Petri estocástica que posteriormente asociaremos al proceso de sPBC. Esto representa una diferencia notable con respecto PBC y a tPBC, donde cada regla corresponde a la ejecución de un multiconjunto de multiacciones que pueden realizarse concurrentemente, dando lugar a una secuencia de pasos en la red de Petri asociada. Es decir, PBC y tPBC soportan *conurrencia real*, mientras que sPBC no. Esto es debido a que la probabilidad de ejecutar dos ó más multiacciones al mismo tiempo es nula. En consecuencia, nuestra semántica operacional de orden total (de *entremezcla*) captura adecuadamente la semántica natural del lenguaje, lo que luego se repetirá en la red de Petri estocástica asociada a cada proceso. Además, la propiedad de *pérdida de*

memoria de la distribución exponencial (negativa) juega un papel importante, ya que el tiempo que ha transcurrido hasta que una primera componente evoluciona ejecutando una multiacción no afecta al tiempo de ejecución de las componentes restantes.

Supondremos que todas las multiacciones estocásticas tienen asociada una razón que es independiente del tiempo, y que éstas son homogéneas respecto del tiempo.

En la tabla 3.3 se definen formalmente las reglas para los diversos operadores del lenguaje, salvo para la sincronización, que se verá posteriormente con más detalle. En estas reglas supondremos que G y H son s-expresiones dinámicas operativas.

<p>(B) $\frac{\frac{\frac{\quad}{\langle \alpha, r \rangle} \langle \alpha, r \rangle}{\langle \alpha, r \rangle}}{\langle \alpha, r \rangle}$</p>	
<p>(S1) $\frac{G \xrightarrow{\langle \alpha, r \rangle} G'}{G; F \xrightarrow{\langle \alpha, r \rangle} G'; F}$</p>	<p>(S2) $\frac{H \xrightarrow{\langle \alpha, r \rangle} H'}{E; H \xrightarrow{\langle \alpha, r \rangle} E; H'}$</p>
<p>(Rs) $\frac{G \xrightarrow{\langle \alpha, r \rangle} G'}{G \text{ rs } a \xrightarrow{\langle \alpha, r \rangle} G' \text{ rs } a} \quad a, \hat{a} \notin A(\alpha)$</p>	<p>(Re) $\frac{G \xrightarrow{\langle \alpha, r \rangle} G'}{G[f] \xrightarrow{\langle f(\alpha), r \rangle} G'[f]}$</p>
<p>(E1) $\frac{G \xrightarrow{\langle \alpha, r \rangle} G'}{G \square F \xrightarrow{\langle \alpha, r \rangle} G' \square F}$</p>	<p>(E2) $\frac{H \xrightarrow{\langle \alpha, r \rangle} H'}{E \square H \xrightarrow{\langle \alpha, r \rangle} E \square H'}$</p>
<p>(C1) $\frac{G \xrightarrow{\langle \alpha, r \rangle} G'}{G \parallel H \xrightarrow{\langle \alpha, r \rangle} G' \parallel H}$</p>	<p>(C2) $\frac{H \xrightarrow{\langle \alpha, r \rangle} H'}{G \parallel H \xrightarrow{\langle \alpha, r \rangle} G \parallel H'}$</p>

Tabla 3.3: Reglas de transición estocásticas (I)

4.2.1 Sincronización. Primera aproximación

De las reglas C1 y C2 de la tabla 3.3 para el operador \parallel , o más bien del hecho que no tengamos ninguna más para el mismo, se deduce que no es posible la *concurrentia real*, esto es, no es posible realizar simultáneamente dos o más multiacciones estocásticas, con lo que se tiene una semántica de orden total. Así, si G y H son operativas, y tenemos las transiciones estocásticas:

$$G \xrightarrow{\langle \alpha, r \rangle} G', \quad H \xrightarrow{\langle \beta, s \rangle} H'$$

de acuerdo a las reglas de la tabla 3.3 podremos obtener las secuencias de transiciones:

$$\begin{aligned} G \parallel H &\xrightarrow{\langle \alpha, r \rangle} G' \parallel H \xrightarrow{(\emptyset)^*} J_1^* \xrightarrow{\langle \beta, s \rangle} G_1 \parallel H_1 \\ G \parallel H &\xrightarrow{\langle \beta, s \rangle} G \parallel H' \xrightarrow{(\emptyset)^*} J_2^* \xrightarrow{\langle \alpha, r \rangle} G_2 \parallel H_2 \end{aligned}$$

donde $G_1 \parallel H_1 \equiv G_2 \parallel H_2 \equiv G' \parallel H'$ y, J_1^* y J_2^* son s-expresiones dinámicas operativas.

En definitiva, la naturaleza estocástica del modelo hace que la probabilidad de que dos multiacciones se ejecuten a la vez es nula. Sin embargo, para poder definir correctamente la sincronización necesitamos saber qué multiacciones estocásticas podrían ejecutarse en cada instante, lo que se corresponde con los posibles conjuntos de bolsas de multiacciones estocásticas que podrían realizarse simultáneamente en PBC. En nuestro ejemplo, tendremos que $\{ \langle \alpha, r \rangle, \langle \beta, s \rangle \}$ será una *bolsa concurrente* de $G \parallel H$.

A continuación, definimos el conjunto de bolsas concurrentes para cada s-expresión dinámica operativa. Cada bolsa concurrente será un multiconjunto de multiacciones estocásticas, y el conjunto de las bolsas concurrentes de G lo denotaremos por $BC(G)$.

Definición 3.4 Definimos la función:

$$BC : OpDynExpr \longrightarrow \mathcal{P}(\mathcal{B}(SL))$$

en la forma siguiente:

- Si $G \in OpDynExpr$ es final (es decir $G = \underline{E}$), se toma $BC(G) = \emptyset$.
- Si $G \in OpDynExpr$ no es final:
 - $BC(\overline{\langle \alpha, r \rangle}) = \{ \{ \langle \alpha, r \rangle \} \}$.
 - Para cada $\gamma \in BC(G)$, tenemos:
 - $\gamma \in BC(G; E)$
 - $\gamma \in BC(E; G)$
 - $\gamma \in BC(G \square E)$
 - $\gamma \in BC(E \square G)$
 - $\gamma \in BC(G \text{ rs } a)$, si $a, \hat{a} \notin A(\gamma)$
 - $\gamma \in BC(G \text{ sy } a)$
 - $f(\gamma) \in BC(G[f])$
 - $\gamma \in BC(G \parallel H)$
 - $\gamma \in BC(H \parallel G)$
 - Si $\gamma_1 \in BC(G)$, $\gamma_2 \in BC(H)$, entonces $\gamma_1 + \gamma_2 \in BC(G \parallel H)$.

- Si $\gamma \in BC(G \text{ sy } a)$, y $\langle \alpha, r_1 \rangle, \langle \beta, r_2 \rangle \in \gamma$, (con $\langle \alpha, r_1 \rangle \neq \langle \beta, r_2 \rangle$, o bien son instancias diferentes de la misma multiacción estocástica en γ) con $a \in A(\alpha)$, $\hat{a} \in A(\beta)$, ó viceversa, entonces $\gamma' \in BC(G \text{ sy } a)$, donde:

$$\gamma' = (\gamma + \{\langle \alpha, r_1 \rangle \oplus_a \langle \beta, r_2 \rangle\}) \setminus \{\langle \alpha, r_1 \rangle, \langle \beta, r_2 \rangle\} \quad \square$$

Nótese que la definición anterior contempla todos los casos posibles de s-expresiones dinámicas operativas, y por lo tanto está bien definida.

Además, a partir de la definición de bolsa concurrente se puede concluir la siguiente propiedad.

Propiedad 3.1 *Para toda $G \in OpDynExpr$, si $\gamma \in BC(G)$ entonces para todo submulticonjunto no vacío γ_i de γ se tiene que $\gamma_i \in BC(G)$. Es decir, $BC(G)$ es cerrado con respecto a la inclusión de multiconjuntos.*

Demostración: Inmediata. □

Veamos un ejemplo para ilustrar la definición anterior en el caso concreto de la sincronización.

Ejemplo 3.2 Sea G_2 la s-expresión dinámica operativa siguiente:

$$G_2 = \overline{\langle \{a\}, 1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, 2 \rangle} \parallel \overline{\langle \{a\}, 3 \rangle}$$

Para obtener $BC(G_2 \text{ sy } a)$ necesitamos primero conocer $BC(G_2)$:

$$\begin{aligned} BC(G_2) = \{ & \langle \{a\}, 1 \rangle, \langle \{\hat{a}, \hat{a}\}, 2 \rangle, \langle \{a\}, 3 \rangle, \\ & \langle \{a\}, 1 \rangle, \langle \{\hat{a}, \hat{a}\}, 2 \rangle, \\ & \langle \{\hat{a}, \hat{a}\}, 2 \rangle, \langle \{a\}, 3 \rangle, \\ & \langle \{a\}, 1 \rangle, \langle \{a\}, 3 \rangle, \langle \{a\}, 1 \rangle, \\ & \langle \{\hat{a}, \hat{a}\}, 2 \rangle, \langle \{a\}, 3 \rangle \} \end{aligned}$$

Entonces, obtenemos las nuevas bolsas de $BC(G_2 \text{ sy } a)$ sincronizando sobre pares de acciones conjugadas (a, \hat{a}) :

$$\begin{aligned} BC(G_2 \text{ sy } a) = BC(G_2) \cup \{ & \langle \{\hat{a}\}, 1 \rangle, \langle \{a\}, 3 \rangle, \\ & \langle \{a\}, 1 \rangle, \langle \{\hat{a}\}, 2 \rangle, \langle \emptyset, 1 \rangle, \\ & \langle \{\hat{a}\}, 1 \rangle, \langle \{\hat{a}\}, 2 \rangle \} \end{aligned}$$

□

Una vez definido BC , estamos en condiciones de presentar las reglas estocásticas para la sincronización (tabla 3.4). Observemos que en la regla $Sy2$ la razón de la nueva multiacción estocástica será la menor de las razones de las multiacciones estocásticas sincronizadas, lo que intuitivamente refleja que esperamos a la más lenta.

La regla $Sy1$ captura el hecho de que la sincronización conserva el comportamiento de G . La regla $Sy2$ captura la sincronización de dos multiacciones $\langle \alpha_1, r_1 \rangle$, $\langle \alpha_2, r_2 \rangle$, las cuales se ejecutan una tras otra, aunque posiblemente haya que ejecutar algunas reglas de inacción entre ambas, para obtener s-expresiones dinámicas operativas a las que poder aplicar las reglas de transición estocásticas. La nueva multiacción estocástica generada por sincronización es:

$$\langle \alpha_1, r_1 \rangle \oplus_a \langle \alpha_2, r_2 \rangle = \langle \alpha_1 \oplus_a \alpha_2, R \rangle$$

donde $R = \min\{r_1, r_2\}$ y \oplus_a es el operador definido en el apartado 3.1.1 de este mismo capítulo.

<p>(Sy1) $\frac{G \xrightarrow{\langle \alpha, r \rangle} H}{G \text{ sy } a \xrightarrow{\langle \alpha, r \rangle} H \text{ sy } a}$</p> <p>(Sy2) Con $\{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle\} \in BC(G_1 \text{ sy } a)$, $a \in A(\alpha_1)$, $\hat{a} \in A(\alpha_2)$:</p> $\frac{G_1 \text{ sy } a \xrightarrow{\langle \alpha_1, r_1 \rangle} G_2 \text{ sy } a \xrightarrow{(-\hat{a})^*} G_2^* \text{ sy } a \xrightarrow{\langle \alpha_2, r_2 \rangle} G_3 \text{ sy } a}{G_1 \text{ sy } a \xrightarrow{\langle \alpha_1 \oplus_a \alpha_2, \min\{r_1, r_2\} \rangle} G_3 \text{ sy } a}$

Tabla 3.4: Reglas de transición estocásticas del operador sy

Definición 3.5 Dada una s-expresión dinámica operativa G , llamaremos conjunto *ejecutable de G* al conjunto de las multiacciones estocásticas que se pueden realizar a partir de G , aplicando alguna de las reglas de las tablas 3.3 y 3.4, y lo denotaremos por $ejec(G)$. \square

Observemos que dada una s-expresión dinámica operativa cualquiera G y $\langle \alpha, r \rangle \in ejec(G)$, entonces $\{\langle \alpha, r \rangle\} \in BC(G)$, y viceversa.

En la proposición siguiente vamos a probar que la definición de BC junto con las reglas de la tabla 3.4 capturan adecuadamente lo que pretendemos en relación a

las multiacciones estocásticas en concurrencia; es decir, siempre que tengamos una bolsa concurrente de G , cualquier serialización de ella es posible como evolución de G , garantizándose así que se cumple la primera parte de *Sy2*. Además, todas las serializaciones de una misma bolsa conducen a s-expresiones dinámicas equivalentes, garantizándose así que no importa el orden en que las coloquemos al sincronizar, ya que:

$$\alpha_1 \oplus_a \alpha_2 = \alpha_2 \oplus_a \alpha_1$$

$$\min\{r_1, r_2\} = \min\{r_2, r_1\}$$

Posteriormente, tendremos que tener cuidado para tener en cuenta una sola vez cada bolsa.

Proposición 3.1 Dada una s-expresión dinámica operativa cualquiera G , y $\gamma \in BC(G)$, para toda serialización de las multiacciones estocásticas de γ :

$$\langle \alpha_1, r_1 \rangle . \langle \alpha_2, r_2 \rangle \dots \langle \alpha_n, r_n \rangle$$

existe una secuencia de transiciones:

$$G \xrightarrow{\langle \alpha_1, r_1 \rangle} G_1 \xrightarrow{(\emptyset)}^* G_1^* \xrightarrow{\langle \alpha_2, r_2 \rangle} \dots \xrightarrow{(\emptyset)}^* G_{n-1}^* \xrightarrow{\langle \alpha_n, r_n \rangle} G'$$

Además, todas las s-expresiones dinámicas G' obtenidas por serialización de γ son equivalentes respecto a \equiv .

Demostración:

Por inducción estructural sobre la sintaxis de G .

- Caso base: $G = \overline{\langle \alpha_1, r_1 \rangle}$. Trivial.
- Una simple aplicación de la hipótesis de inducción resuelve los casos siguientes:

$$G; E \quad , \quad E; G \quad , \quad E \square G \quad , \quad G \square E \quad , \quad G \text{ r s a } y \quad G[f]$$

- Para el caso del operador paralelo, $G_1 \parallel G_2$, si $\gamma \in BC(G_1)$ (ó $\gamma \in BC(G_2)$), también se aplicaría la hipótesis de inducción y se concluiría. Pero si consideramos $\gamma = \gamma_1 + \gamma_2$, con $\gamma_1 \in BC(G_1)$ y $\gamma_2 \in BC(G_2)$, en este caso tendremos que aplicar la hipótesis de inducción sobre ambos, para conseguir secuencias de transiciones que pueden ser combinadas de la forma adecuada, ya que las evoluciones de ambas componentes son completamente independientes:

- El caso base sería:

$$\gamma = \{ \langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle \}$$

donde $\langle \alpha_1, r_1 \rangle \in ejec(G_1)$ y $\langle \alpha_2, r_2 \rangle \in ejec(G_2)$.

Como las evoluciones de G_1 y G_2 son totalmente independientes, tendríamos:

$$\begin{aligned} G_1 \parallel G_2 \xrightarrow{\langle \alpha_1, r_1 \rangle} G_1' \parallel G_2 \xrightarrow{(-\emptyset)}^* G_1'^* \parallel G_2 \xrightarrow{\langle \alpha_2, r_2 \rangle} G_1'^* \parallel G_2' \\ G_1 \parallel G_2 \xrightarrow{\langle \alpha_2, r_2 \rangle} G_1 \parallel G_2' \xrightarrow{(-\emptyset)}^* G_1 \parallel G_2'^* \xrightarrow{\langle \alpha_1, r_1 \rangle} G_1' \parallel G_2'^* \end{aligned}$$

De esta forma las dos serializaciones de γ :

$$\langle \alpha_1, r_1 \rangle . \langle \alpha_2, r_2 \rangle \quad \text{y} \quad \langle \alpha_2, r_2 \rangle . \langle \alpha_1, r_1 \rangle$$

son posibles desde G , y además, obviamente

$$G_1'^* \parallel G_2' \equiv G_1' \parallel G_2'^*$$

- El caso general nos lleva a considerar:

$$\gamma = \{ \langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle, \dots, \langle \alpha_n, r_n \rangle \}$$

donde

$$\begin{aligned} \gamma_1 &= \{ \langle \alpha_1, r_1 \rangle, \dots, \langle \alpha_k, r_k \rangle \} \in BC(G_1) \\ \gamma_2 &= \{ \langle \alpha_{k+1}, r_{k+1} \rangle, \dots, \langle \alpha_n, r_n \rangle \} \in BC(G_2) \end{aligned}$$

Entonces podremos aplicar la hipótesis de inducción sobre ambas evoluciones para, con un razonamiento similar al del caso, base concluir la propiedad.

Con este razonamiento inductivo sobre el número de multiacciones estocásticas en γ podemos concluir que todas las s-expresiones dinámicas obtenidas por serialización de γ son equivalentes respecto a \equiv .

- Para el caso del operador sy , si tenemos $G = G_1 sy a$ necesitamos distinguir los dos casos siguientes:
 - Si $\gamma \in BC(G_1)$, entonces el resultado es inmediato, bastando con aplicar la hipótesis de inducción.
 - Si $\gamma = (\gamma_1 + \{ \langle \alpha_1, r_1 \rangle \oplus_a \langle \alpha_2, r_2 \rangle \}) \setminus \{ \langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle \}$, con $\gamma_1 \in BC(G)$, $a \in A(\alpha_1)$, $\hat{a} \in A(\alpha_2)$, entonces podría darse el caso de que $\gamma_1 \in BC(G_1)$, ó γ_1 podría ser obtenida de la misma forma, sincronizando algunas multiacciones; pero este procedimiento no puede ser aplicado

indefinidamente, de modo que aplicaremos un nuevo razonamiento inductivo sobre el número de veces que ha sido aplicada la sincronización de multiacciones. Debido a ello, nuestro caso base es $\gamma_1 \in BC(G_1)$, para el cual podemos aplicar la hipótesis de la primera inducción, considerando una serialización en la que $\langle \alpha_1, r_1 \rangle$ y $\langle \alpha_2, r_2 \rangle$ aparecen seguidas. Aunque esta secuencia empieza en G_1 , se puede obtener la misma empezando en $G_1 \text{ sy } a$ (regla *Sy1*), para luego aplicar la regla *Sy2*, obteniendo así una secuencia de transiciones que corresponde a la serialización dada de γ . Podemos observar que el orden de las restantes multiacciones estocásticas de γ_1 no es importante, y de esta forma, para cualquier serialización de γ podemos obtener la correspondiente secuencia de transiciones. Además, con esta secuencia de transiciones alcanzamos la misma s-expresión dinámica que alcanzábamos con γ_1 . En consecuencia, aplicando la hipótesis de inducción estructural podemos concluir que todas las s-expresiones dinámicas alcanzadas por serialización de γ son equivalentes respecto a \equiv . Para el caso general tenemos ahora, como hipótesis, que la propiedad se cumple para $\gamma_1 \in BC(G_1 \text{ sy } a)$, lo que nos permite razonar de la misma forma que en el caso base, considerando una serialización de γ_1 . La única diferencia es que en este caso la secuencia de reglas que obtenemos empieza directamente desde $G_1 \text{ sy } a$, en lugar de desde G_1 . □

3.2.3 Sistema de transiciones

En este apartado definimos el (multi)sistema de transiciones que define la semántica operacional de sPBC, cuyos estados son clases de equivalencias de s-expresiones dinámicas, con respecto a la relación \equiv . Veamos primero algunas definiciones que utilizaremos después.

Definición 3.6 Sean G, G', H y H' s-expresiones dinámicas, tales que

$$G' \xrightarrow{\langle \alpha, r \rangle} H', \quad \text{con } G' \in [G]_{\equiv} \quad \text{y} \quad H' \in [H]_{\equiv}$$

entonces diremos que H y H' son ambas s-expresiones dinámicas *derivadas inmediatas* (en un paso) de G . □

Ahora pasamos a definir el multiconjunto de multiacciones estocásticas que se pueden ejecutar en un solo paso desde una clase de equivalencia.

Definición 3.7 Dada la clase $[G]_{\equiv}$, denotaremos por $act(G)$ al multiconjunto de multiacciones estocásticas que se pueden ejecutar en un solo paso desde cualquier s-expresión dinámica operativa de la clase. De modo que tenemos,

$$act(G) = \{ \langle \alpha, r \rangle \mid \exists G' \in [G]_{\equiv} \wedge \exists H \in DynExpr \text{ con } G' \xrightarrow{\langle \alpha, r \rangle} H \}$$

Cada multiacción estocástica de $act(G)$ diremos que está *activada* en G . \square

Con el objetivo de definir el número de veces que cada multiacción estocástica aparece en $act(G)$, tendremos en cuenta las diferentes formas *significativas* en que se ha podido obtener tal transición estocástica. Por ejemplo, para la siguiente s-expresión dinámica no operativa:

$$G = \overline{\langle \alpha, r \rangle \square \langle \alpha, r \rangle}$$

tenemos dos formas *significativamente* diferentes de obtener $\langle \alpha, r \rangle$, según qué s-expresión dinámica operativa consideremos, aplicando entonces la regla $E1$ ó $E2$, según corresponda. Así, en este caso:

$$act(G) = \{ \langle \alpha, r \rangle, \langle \alpha, r \rangle \}$$

Pero en el caso de la sincronización, consideraremos que la aplicación de la regla $Sy2$ en los dos sentidos posibles, según el orden en que tomemos las dos multiacciones estocásticas de la bolsa concurrente, genera transiciones *no significativamente diferentes*. Por ejemplo, si consideramos la s-expresión dinámica:

$$G_1 = \overline{(\langle \{a\}, r \rangle \parallel \langle \{\hat{a}\}, r \rangle) \text{ sy } a}$$

entonces sólo consideraremos la ejecución de $\langle \emptyset, r \rangle$ una vez, obteniendo entonces:

$$act(G_1) = \{ \langle \{a\}, r \rangle, \langle \{\hat{a}\}, r \rangle, \langle \emptyset, r \rangle \}$$

Esto puede generalizarse para la sincronización de n multiacciones estocásticas en la forma siguiente:

Observación 3.1 Al generar el sistema de transiciones iremos quedándonos con cierta información para aquellas transiciones que han sido obtenidas al aplicar la regla ($Sy2$); en concreto, con la historia pasada de las sincronizaciones realizadas hasta obtener esta transición. De modo que si detectamos que con una nueva aplicación de dicha regla ($Sy2$) se generaría una nueva transición cuya historia coincidiría con

la de alguna transición ya generada anteriormente, entonces dicha transición no es considerada. Para quedarnos con la historia de cada transición originada al aplicar la regla (Sy2) seguimos el criterio siguiente:

Se etiquetan numéricamente las multiacciones estocásticas de la s-expresión dinámica operativa de izquierda a derecha (por orden de escritura) y, al aplicar la regla (Sy2) obtenemos nuevas transiciones con nuevas multiacciones estocásticas, que irán etiquetadas con la numeración obtenida con la concatenación de las historias de las componentes involucradas en la sincronización. \square

Ejemplo 3.3 Para $G_3 \text{ sy } a = \overline{\langle \{a\}, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel \overline{\langle \{a\}, r_3 \rangle} \text{ sy } a$ la numeración sería:

- 1 para $\langle \{a\}, r_1 \rangle$
- 2 para $\langle \{\hat{a}, \hat{a}\}, r_2 \rangle$
- 3 para $\langle \{a\}, r_3 \rangle$

Denotando:

$$G_1 \text{ sy } a = \overline{\langle \{a\}, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel \overline{\langle \{a\}, r_3 \rangle} \text{ sy } a$$

$$G_2 \text{ sy } a = \overline{\langle \{a\}, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel \overline{\langle \{a\}, r_3 \rangle} \text{ sy } a$$

$$G_3 \text{ sy } a = \overline{\langle \{a\}, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel \overline{\langle \{a\}, r_3 \rangle} \text{ sy } a$$

$$G_{12} \text{ sy } a = \overline{\langle \{a\}, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel \overline{\langle \{a\}, r_3 \rangle} \text{ sy } a$$

$$G_{23} \text{ sy } a = \overline{\langle \{a\}, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel \overline{\langle \{a\}, r_3 \rangle} \text{ sy } a$$

$$G_{123} \text{ sy } a = \overline{\langle \{a\}, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel \overline{\langle \{a\}, r_3 \rangle} \text{ sy } a$$

El paso de $G \text{ sy } a$ a $G_{12} \text{ sy } a$ aplicando la regla (Sy2) se puede hacer de dos formas:

Con $\{\langle \{a\}, r_1 \rangle, \langle \{\hat{a}, \hat{a}\}, r_2 \rangle\} \in BC(G \text{ sy } a)$, $a \in A(\{a\})$, $\hat{a} \in A(\{\hat{a}, \hat{a}\})$:

$$\frac{G \text{ sy } a \xrightarrow{\langle \{a\}, r_1 \rangle}_1 G_1 \text{ sy } a \xrightarrow{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle}_2 G_{12} \text{ sy } a}{G \text{ sy } a \xrightarrow{\langle \{\hat{a}\}, r_{12} \rangle}_{1,2} G_{12} \text{ sy } a}$$

donde la transición obtenida tiene como subíndice 1.2, reflejando las dos multiacciones estocásticas que han intervenido en la sincronización. Ahora, a la instancia de

la nueva multiacción estocástica así obtenida, $\langle \{\hat{a}\}, r_{12} \rangle$, se la etiquetará con esa numeración.

Aplicando las reglas en orden inverso, habríamos obtenido:

$$\frac{G \text{ sy } a \xrightarrow{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} {}_2 G_2 \text{ sy } a \xrightarrow{\langle \{a\}, r_1 \rangle} {}_1 G_{12} \text{ sy } a}{G \text{ sy } a \xrightarrow{\langle \{\hat{a}\}, r_{12} \rangle} {}_{2.1} G_{12} \text{ sy } a}$$

que según el criterio indicado no sería incluida, pues tiene la misma historia que la anterior.

Por otra parte, tenemos:

$\{\langle \{a\}, r_3 \rangle, \langle \{\hat{a}, \hat{a}\}, r_2 \rangle\} \in BC(G \text{ sy } a)$, $a \in A(\{a\})$, $\hat{a} \in A(\{\hat{a}, \hat{a}\})$, con lo que

$$\frac{G \text{ sy } a \xrightarrow{\langle \{a\}, r_3 \rangle} {}_3 G_3 \text{ sy } a \xrightarrow{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} {}_2 G_{23} \text{ sy } a}{G \text{ sy } a \xrightarrow{\langle \{\hat{a}\}, r_{23} \rangle} {}_{2.3} G_{23} \text{ sy } a}$$

De modo que desde $G \text{ sy } a$ se puede aplicar $Sy2$ sobre $\{\langle a, r_3 \rangle, \langle \hat{a}, r_{12} \rangle\}$ o sobre $\{\langle \{a\}, r_1 \rangle, \langle \hat{a}, r_{23} \rangle\}$, y únicamente se apuntaría una de ellas, ya que tienen la misma historia ($\{3\}.\{1, 2\}$ y $\{1\}.\{2, 3\} = \{1, 2, 3\}$):

- $\{\langle \{a\}, r_3 \rangle, \langle \{\hat{a}\}, r_{12} \rangle\} \in BC(G \text{ sy } a)$, $a \in A(\{a\})$, $\hat{a} \in A(\{\hat{a}\})$:

$$\frac{G \text{ sy } a \xrightarrow{\langle \{a\}, r_3 \rangle} {}_3 G_3 \text{ sy } a \xrightarrow{\langle \{\hat{a}\}, r_{12} \rangle} {}_{1.2} G_{123} \text{ sy } a}{G \text{ sy } a \xrightarrow{\langle \emptyset, r_{123} \rangle} {}_{3.1.2} G_{123} \text{ sy } a}$$

- $\{\langle \{a\}, r_1 \rangle, \langle \{\hat{a}\}, r_{23} \rangle\} \in BC(G \text{ sy } a)$, $a \in A(\{a\})$, $\hat{a} \in A(\{\hat{a}\})$:

$$\frac{G \text{ sy } a \xrightarrow{\langle \{a\}, r_1 \rangle} {}_1 G_1 \text{ sy } a \xrightarrow{\langle \{\hat{a}\}, r_{23} \rangle} {}_{2.3} G_{123} \text{ sy } a}{G \text{ sy } a \xrightarrow{\langle \emptyset, r_{123} \rangle} {}_{1.2.3} G_{123} \text{ sy } a}$$

□

Ejemplo 3.4 Consideremos la s-expresión dinámica siguiente:

$$G_4 = \overline{((\langle a, 1 \rangle \square \langle a, 1 \rangle) \parallel \langle \{\hat{a}, \hat{a}, b\}, 1 \rangle \parallel \langle a, 1 \rangle) \text{ sy } a}$$

G_4 no es operativa y a partir de ella se pueden obtener las s-expresiones dinámicas operativas siguientes:

$$G'_4 = \overline{((\overline{\langle a, 1 \rangle} \square \langle a, 1 \rangle) \parallel \overline{\langle \{\hat{a}, \hat{a}, b\}, 1 \rangle} \parallel \overline{\langle a, 1 \rangle}) \text{ sy } a}$$

$$G_4'' = ((\langle a, 1 \rangle \square \overline{\langle a, 1 \rangle}) \parallel \overline{\langle \hat{a}, \hat{a}, b \rangle, 1 \rangle} \parallel \overline{\langle a, 1 \rangle}) \text{ sy } a$$

La numeración de cada multiacción sería la siguiente:

$$G_4' = ((\overline{\langle a, 1 \rangle_1} \square \langle a, 1 \rangle_2) \parallel \overline{\langle \hat{a}, \hat{a}, b \rangle, 1 \rangle_3} \parallel \overline{\langle a, 1 \rangle_4}) \text{ sy } a$$

$$G_4'' = ((\langle a, 1 \rangle_1 \square \overline{\langle a, 1 \rangle_2}) \parallel \overline{\langle \hat{a}, \hat{a}, b \rangle, 1 \rangle_3} \parallel \overline{\langle a, 1 \rangle_4}) \text{ sy } a$$

De esta forma:

$$\text{act}(G_4) = \{\langle a, 1 \rangle_1, \langle a, 1 \rangle_2, \langle \hat{a}, \hat{a}, b \rangle, 1 \rangle_3, \langle a, 1 \rangle_4\}$$

$$\text{ejec}(G_4') = \{\langle a, 1 \rangle_1, \langle \hat{a}, \hat{a}, b \rangle, 1 \rangle_3, \langle a, 1 \rangle_4\}$$

$$\text{ejec}(G_4'') = \{\langle a, 1 \rangle_2, \langle \hat{a}, \hat{a}, b \rangle, 1 \rangle_3, \langle a, 1 \rangle_4\}$$

$$\begin{aligned} BC(G_4') = \{ & \{\langle a, 1 \rangle_1, \langle \hat{a}, \hat{a}, b \rangle, 1 \rangle_3, \langle a, 1 \rangle_4\} \\ & \{\langle \hat{a}, b \rangle, 1 \rangle_{1.3}, \langle a, 1 \rangle_4\}, \{\langle a, 1 \rangle_1, \langle \hat{a}, b \rangle, 1 \rangle_{3.4}\}, \\ & \{\langle a, 1 \rangle_1, \langle \hat{a}, \hat{a}, b \rangle, 1 \rangle_3\}, \{\langle a, 1 \rangle_1, \langle a, 1 \rangle_4\}, \\ & \{\langle \hat{a}, \hat{a}, b \rangle, 1 \rangle_3, \langle a, 1 \rangle_4\}, \{\langle a, 1 \rangle_1\}, \\ & \{\langle \hat{a}, \hat{a}, b \rangle, 1 \rangle_3\}, \{\langle a, 1 \rangle_4\} \\ & \{\langle \hat{a}, b \rangle, 1 \rangle_{1.3}\}, \{\langle \hat{a}, b \rangle, 1 \rangle_{3.4}\}, \\ & \{\langle b, 1 \rangle_{1.3.4}\} \} \end{aligned}$$

$$\begin{aligned} BC(G_4'') = \{ & \{\langle a, 1 \rangle_2, \langle \hat{a}, \hat{a}, b \rangle, 1 \rangle_3, \langle a, 1 \rangle_4\} \\ & \{\langle \hat{a}, b \rangle, 1 \rangle_{2.3}, \langle a, 1 \rangle_4\}, \{\langle a, 1 \rangle_2, \langle \hat{a}, b \rangle, 1 \rangle_{3.4}\}, \\ & \{\langle a, 1 \rangle_2, \langle \hat{a}, \hat{a}, b \rangle, 1 \rangle_3\}, \{\langle a, 1 \rangle_2, \langle a, 1 \rangle_4\}, \\ & \{\langle \hat{a}, \hat{a}, b \rangle, 1 \rangle_3, \langle a, 1 \rangle_4\}, \{\langle a, 1 \rangle_2\}, \\ & \{\langle \hat{a}, \hat{a}, b \rangle, 1 \rangle_3\}, \{\langle a, 1 \rangle_4\} \\ & \{\langle \hat{a}, b \rangle, 1 \rangle_{2.3}\}, \{\langle \hat{a}, b \rangle, 1 \rangle_{3.4}\}, \\ & \{\langle b, 1 \rangle_{2.3.4}\} \} \end{aligned}$$

Tendremos en cuenta lo anterior cuando en el ejemplo 3.9 construyamos el sistema de transiciones asociado a $\overline{E_5}$. \square

Los dos lemas siguientes justifican nuestra decisión de no contabilizar estas transiciones. El primero de ellos establece que da igual el orden que sigamos (siempre y cuando sean posibles las sincronizaciones), ya que siempre obtendremos la misma multiacción. El segundo lema nos garantiza que si partimos de la misma s-expresión dinámica operativa, independientemente del orden que hayamos seguido para sincronizar (siempre que sea factible), llegaremos a s-expresiones dinámicas equivalentes, y la multiacción estocástica originada por todas las sincronizaciones es siempre la misma.

Lema 3.1 Sean $\alpha_1, \dots, \alpha_n \in \mathcal{L}$, $\{i_1, \dots, i_n\}$ y $\{j_1, \dots, j_n\}$ dos permutaciones del conjunto $\{1, \dots, n\}$.

(i) Si es factible realizar las operaciones \oplus_a (con $a \in \mathcal{A}$) que aparecen en las dos expresiones siguientes:

$$e_1 = (\dots((\alpha_{i_1} \oplus_a \alpha_{i_2}) \oplus_a \alpha_{i_3}) \oplus_a \dots \oplus_a \alpha_{i_n})$$

$$e_2 = (\dots((\alpha_{j_1} \oplus_a \alpha_{j_2}) \oplus_a \alpha_{j_3}) \oplus_a \dots \oplus_a \alpha_{j_n})$$

Entonces $e_1 = e_2$.

(ii) Si es factible realizar las operaciones que aparecen en los operadores \oplus_{a_s} en las siguientes expresiones para $a_s \in \{a_1, \dots, a_{n-1}\} \in \mathcal{A}$, y siendo $\{k_1, \dots, k_{n-1}\}$ y $\{l_1, \dots, l_{n-1}\}$ dos permutaciones del conjunto $\{1, \dots, n-1\}$:

$$e_1 = (\dots((\alpha_{i_1} \oplus_{a_{k_1}} \alpha_{i_2}) \oplus_{a_{k_2}} \alpha_{i_3}) \oplus_{a_{k_3}} \dots \oplus_{a_{k_{n-1}}} \alpha_{i_n})$$

$$e_2 = (\dots((\alpha_{j_1} \oplus_{a_{l_1}} \alpha_{j_2}) \oplus_{a_{l_2}} \alpha_{j_3}) \oplus_{a_{l_3}} \dots \oplus_{a_{l_{n-1}}} \alpha_{j_n})$$

Entonces $e_1 = e_2$.

Demostración:

(i) En ambas expresiones se realiza básicamente una unión de multiconjuntos, pero cada vez que se sincroniza se quita un par $\{a, \hat{a}\}$. Como esto se hace $(n-1)$ veces, quedaría en ambos casos:

$$(\alpha_1 + \alpha_2 + \dots + \alpha_n) \setminus (n-1) \cdot \{a, \hat{a}\}$$

(ii) De nuevo se realiza básicamente una unión de multiconjuntos, pero cada vez que se sincroniza con respecto de a_k se quita un par $\{a_k, \hat{a}_k\}$. Como esto se hace para $k = 1, \dots, (n-1)$, quedaría en ambos casos:

$$(\alpha_1 + \alpha_2 + \dots + \alpha_n) \setminus \{a_1, \hat{a}_1, \dots, a_{n-1}, \hat{a}_{n-1}\}$$

□

Lema 3.2 Sea G una s-expresión dinámica operativa, y sean $\gamma = \{\langle \alpha_1, r_1 \rangle, \dots, \langle \alpha_n, r_n \rangle\}$ y dos permutaciones del conjunto $\{1, \dots, n\}$: $\{i_1, \dots, i_n\}$ y $\{j_1, \dots, j_n\}$. Supongamos que existen dos serializaciones:

$$G \xrightarrow{\langle \alpha_{i_1}, r_{i_1} \rangle} G_1 \xrightarrow{\langle \alpha_{i_2}, r_{i_2} \rangle} \dots \xrightarrow{\langle \alpha_{i_n}, r_{i_n} \rangle} G_n$$

$$G \xrightarrow{\langle \alpha_{j_1}, r_{j_1} \rangle} G'_1 \xrightarrow{\langle \alpha_{j_2}, r_{j_2} \rangle} \dots \xrightarrow{\langle \alpha_{j_n}, r_{j_n} \rangle} G'_n$$

en las que la regla (*Sy2*) puede ser aplicada en ambos casos $n-1$ veces (para las mismas acciones a_1, a_2, \dots, a_k , posiblemente repetidas, pero el mismo número de veces en cada caso), obteniéndose por tanto en ambos casos una única transición:

$$G \xrightarrow{\langle \beta_i, R_i \rangle} G_n$$

$$G \xrightarrow{\langle \beta_j, R_j \rangle} G'_n$$

Entonces ha de ocurrir que $G_n \equiv G'_n$ y $\langle \beta_i, R_i \rangle = \langle \beta_j, R_j \rangle$.

Demostración:

Es inmediato concluir que γ debe ser una bolsa concurrente de una s-expresión dinámica operativa de la forma general $G = G' \text{ sy } a_1 \dots \text{ sy } a_j$, (con $j \leq n-1$), por lo que se puede aplicar la proposición 3.1, con lo que $G_n \equiv G'_n$.

Además, $R_i = R_j$, dado que vamos tomando los valores mínimos de las razones:

$$R_i = \min\{r_{i_1}, \dots, r_{i_n}\} = \min\{r_{j_1}, \dots, r_{j_n}\} = R_j$$

Finalmente, aplicando el lema anterior concluimos que $\beta_i = \beta_j$. □

Definición 3.8 Denotaremos por $r(G)$ a la suma de todas las razones de las multiacciones estocásticas de $\text{act}(G)$:

$$r(G) =_{\text{def}} \sum_{\langle \alpha_i, r_i \rangle \in \text{act}(G)} r_i \cdot n_i$$

donde n_i es el número de veces que aparece $\langle \alpha_i, r_i \rangle$ en $\text{act}(G)$, teniendo en cuenta la Observación 3.1. □

Definición 3.9 Sean G y H son s-expresiones dinámicas tales que

$$G \equiv G' \xrightarrow{\langle \alpha_1, r_1 \rangle} G_1 \equiv G'_1 \xrightarrow{\langle \alpha_2, r_2 \rangle} \dots G_{n-1} \equiv G'_{n-1} \xrightarrow{\langle \alpha_n, r_n \rangle} H' \equiv H, \text{ con } n \geq 0$$

entonces diremos que H es una s-expresión dinámica *derivada* de G .

Denotaremos por $[G]$, y lo llamaremos *alcance de G* , al conjunto de todas las s-expresiones dinámicas derivadas de G . □

Observemos que si H es una s-expresión dinámica derivada de G , entonces $\forall G_i \equiv G, \forall H_j \equiv H$, se tiene que H_j es una s-expresión dinámica derivada de G_i .

Estamos interesados en aquellas s-expresiones dinámicas que pueden ser derivadas a partir de una s-expresión dinámica inicial \overline{E} y, nos referiremos a ellas como las s-expresiones alcanzables desde la s-expresión estática E .

Definición 3.10 Dada una s-expresión estática E , diremos que una s-expresión dinámica G es *alcanzable* desde E , si $G \in [\overline{E}]$. \square

Definición 3.11 Dada una s-expresión dinámica G , definimos el *(multi)sistema de transiciones* asociado a G , como $ts(G) = (V, A, v_0)$, donde:

- $V = \{ [H]_{\equiv} \mid H \in [G] \}$ es el conjunto de estados,
- $v_0 = [G]_{\equiv}$ es el estado inicial,
- $A = \{ ([H]_{\equiv}, \langle \alpha, r \rangle, [J]_{\equiv}) \mid H \in [G] \wedge H \xrightarrow{\langle \alpha, r \rangle} J \}$ es el multiconjunto de transiciones ó arcos.

A la hora de construir el multiconjunto A tendremos en cuenta la Observación 3.1, de modo que no repetiremos aquellos arcos que correspondan a sincronizaciones que tengan la misma historia. \square

3.2.4 La política de competencia

Una condición *race* o de *competencia* gobernará el comportamiento dinámico del modelo cuando sea posible ejecutar más de una multiacción estocástica. Esto ocurre en dos casos diferentes: cuando estamos en el ámbito de una elección o en el de un operador paralelo. Cuando varias multiacciones estocásticas son posibles, todas empezarán a tratar de ejecutarse, pero la que antes termine (se complete) será la que determine el cambio de estado, esto es, la más rápida ganará la carrera. En consecuencia, la multiacción estocástica con mayor razón asociada (de todas las activadas) tendrá mayor probabilidad de ganar la carrera, ya que su tiempo de espera (*delay*) medio asociado es menor. De esta forma la condición de competencia tiene el efecto de reemplazar el no-determinismo que se tiene en las álgebras de procesos clásicas, tales como CCS [Mil89] y PBC [BDK01b], por una elección

probabilística, como en las álgebras de procesos Markovianas (como PEPA [Hil96] o EMPA [BG98]). Debido a esta política, la probabilidad de que una determinada multiacción estocástica se ejecute primero (gane la carrera) es la razón asociada a dicha multiacción estocástica, dividida por la suma de las razones asociadas a las otras multiacciones estocásticas que puedan ejecutarse en ese momento.

Observación 3.2 Dada una s-expresión dinámica G cualquiera, la probabilidad de que la multiacción estocástica $\langle \alpha_i, r_i \rangle \in act(G)$ gane la carrera estando en el estado $[G]_{\equiv}$ es la siguiente:

$$P(G, \langle \alpha_i, r_i \rangle) = r_i \cdot n_i / r(G)$$

siendo n_i el número de veces que $\langle \alpha_i, r_i \rangle$ aparece en $act(G)$.

Si distinguimos las diferentes instancias de cada multiacción estocástica, entonces la probabilidad de que la multiacción estocástica de $\langle \alpha_i, r_i \rangle$ etiquetada con k , es decir, la probabilidad de que $\langle \alpha_i, r_i \rangle_k$ gane la carrera estando en el estado $[G]_{\equiv}$ es:

$$P'(G, \langle \alpha_i, r_i \rangle_k) = r_i / r(G)$$

□

Cuando una multiacción estocástica termina su ejecución (se completa), el comportamiento del modelo puede cambiar, obteniéndose una nueva s-expresión dinámica. El tiempo de permanencia en $[G]_{\equiv}$, para $G \in [\overline{E}]$ es una variable aleatoria que sigue una distribución exponencial de razón $r(G)$, por ser ésta el mínimo de las variables aleatorias que miden los retrasos de cada multiacción estocástica de $act(G)$.

Proposición 3.2 Sea E una s-expresión estática. El tiempo de permanencia en $[G]_{\equiv}$, para cada $G \in [\overline{E}]$, sigue una distribución exponencial negativa de parámetro $r(G)$.

Demostración:

La demostración es inmediata al tratarse de distribuciones exponenciales independientes y considerarse la política de competencia, ya que el tiempo de permanencia en $[G]_{\equiv}$ es el mínimo de las citadas distribuciones exponenciales, cuya razón es la suma de las razones involucradas. □

Obsérvese que en caso de conflicto, las multiacciones estocásticas que se podían realizar, pero que en el transcurso de la competición no se han completado al haber perdido ésta, se abortan o también podríamos decir que se interrumpen, si se pueden

realizar en la nueva s-expresión dinámica. La propiedad de *pérdida de memoria* de la distribución exponencial hace innecesario tener en cuenta *cuánto tiempo ha transcurrido desde que se encuentra activada*, gracias a lo cual no hace falta especificar qué política de memoria se sigue (*resampling, enabling memory o age memory*).

3.2.5 Ejemplos

En los ejemplos siguientes mostraremos los sistemas de transiciones asociados a las s-expresiones dinámicas \overline{E}_i que iremos presentando, y a partir de cada uno de ellos se deducirán los posibles comportamientos de las mismas (secuencias de multiacciones estocásticas). Para no recargar la notación, etiquetaremos los nodos de los sistemas de transiciones con un representante de su clase.

Ejemplo 3.5 Consideremos la s-expresión estática siguiente:

$$E_1 = (\langle \{a\}, r_1 \rangle ; \langle \{b\}, r_2 \rangle) \square \langle \{c\}, r_3 \rangle$$

Entonces, el sistema de transiciones de \overline{E}_1 sería el que mostramos en la figura 3.1, donde G_1 viene dado por

$$G_1 = (\underline{\langle \{a\}, r_1 \rangle} ; \langle \{b\}, r_2 \rangle) \square \langle \{c\}, r_3 \rangle$$

o cualquier s-expresión dinámica equivalente estructuralmente a ella.

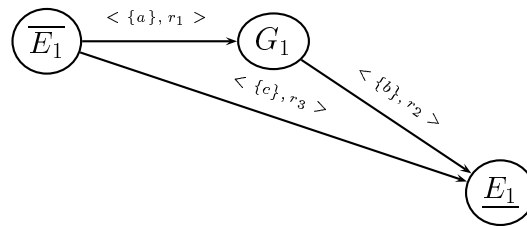


Figura 3.1: Sistema de transiciones de \overline{E}_1

Para E_1 los comportamientos posibles (camino de \overline{E}_1 a \underline{E}_1) son:

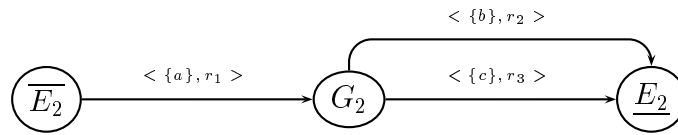
$\langle \{a\}, r_1 \rangle \cdot \langle \{b\}, r_2 \rangle$ con probabilidad $\frac{r_1}{r_1+r_3}$, y

$\langle \{c\}, r_3 \rangle$ con probabilidad $\frac{r_3}{r_1+r_3}$. □

Ejemplo 3.6 Consideremos ahora:

$$E_2 = \langle \{a\}, r_1 \rangle ; (\langle \{b\}, r_2 \rangle \square \langle \{c\}, r_3 \rangle)$$

su correspondiente sistema de transiciones es el mostrado en la figura 3.2.

Figura 3.2: Sistema de transiciones de $\overline{E_2}$

donde G_2 viene dado por

$$G_2 = \langle \{a\}, r_1 \rangle; (\overline{\langle \{b\}, r_2 \rangle} \square \overline{\langle \{c\}, r_3 \rangle})$$

o cualquier s-expresión dinámica equivalente estructuralmente a ella.

Para E_2 , los comportamientos posibles serían:

$$\begin{aligned} &\langle \{a\}, r_1 \rangle \cdot \langle \{b\}, r_2 \rangle \text{ con probabilidad } \frac{r_2}{r_2+r_3}, \text{ y} \\ &\langle \{a\}, r_1 \rangle \cdot \langle \{c\}, r_3 \rangle \text{ con probabilidad } \frac{r_3}{r_2+r_3}. \end{aligned}$$

□

Ejemplo 3.7 Dado ahora:

$$E_3 = \langle \{a\}, r_1 \rangle \parallel \langle \{\hat{a}, \hat{a}\}, r_2 \rangle \parallel \langle \{a\}, r_3 \rangle$$

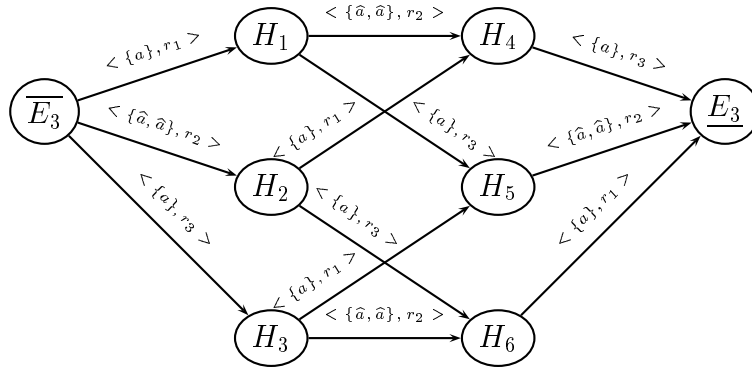
su correspondiente sistema de transiciones es el mostrado en la figura 3.3, donde los H_i son de la forma:

$$\begin{aligned} H_1 &= \overline{\langle \{a\}, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel \overline{\langle \{a\}, r_3 \rangle} \\ H_2 &= \overline{\langle \{a\}, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel \overline{\langle \{a\}, r_3 \rangle} \\ H_3 &= \overline{\langle \{a\}, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel \overline{\langle \{a\}, r_3 \rangle} \\ H_4 &= \overline{\langle \{a\}, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel \overline{\langle \{a\}, r_3 \rangle} \\ H_5 &= \overline{\langle \{a\}, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel \overline{\langle \{a\}, r_3 \rangle} \\ H_6 &= \overline{\langle \{a\}, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel \overline{\langle \{a\}, r_3 \rangle} \end{aligned}$$

Considerando:

$$\gamma_1 = \{ \langle \{a\}, r_1 \rangle, \langle \{\hat{a}, \hat{a}\}, r_2 \rangle, \langle \{a\}, r_3 \rangle \} \in BC(G_3)$$

con $G_3 = \overline{\langle \{a\}, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel \overline{\langle \{a\}, r_3 \rangle}$, se tiene que para E_3 los 6 caminos posibles que empiezan en $\overline{E_3}$ y terminan en $\underline{E_3}$ corresponden a la ejecución de las tres multiacciones estocásticas de γ_1 en cualquier orden :

Figura 3.3: Sistema de transiciones de $\overline{E_3}$

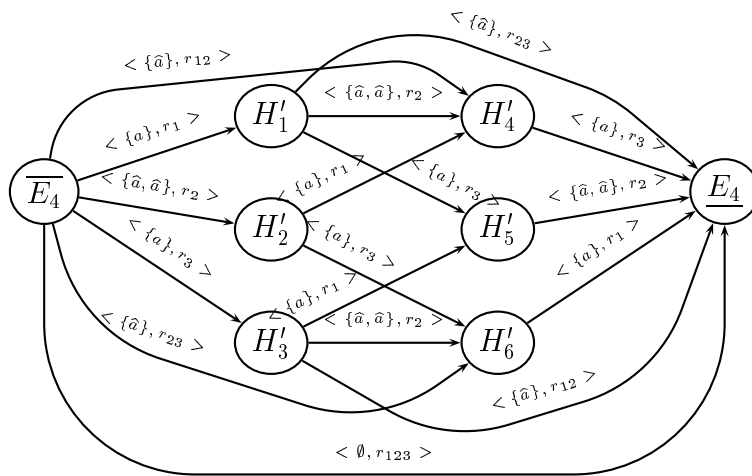
$\langle \{a\}, r_1 \rangle \cdot \langle \{\widehat{a}, \widehat{a}\}, r_2 \rangle \cdot \langle \{a\}, r_3 \rangle$ con probabilidad $\frac{r_1}{r_1+r_2+r_3} \cdot \frac{r_2}{r_2+r_3}$
 $\langle \{a\}, r_1 \rangle \cdot \langle \{a\}, r_3 \rangle \cdot \langle \{\widehat{a}, \widehat{a}\}, r_2 \rangle$ con probabilidad $\frac{r_1}{r_1+r_2+r_3} \cdot \frac{r_3}{r_2+r_3}$
 $\langle \{\widehat{a}, \widehat{a}\}, r_2 \rangle \cdot \langle \{a\}, r_1 \rangle \cdot \langle \{a\}, r_3 \rangle$ con probabilidad $\frac{r_2}{r_1+r_2+r_3} \cdot \frac{r_1}{r_1+r_3}$
 $\langle \{\widehat{a}, \widehat{a}\}, r_2 \rangle \cdot \langle \{a\}, r_3 \rangle \cdot \langle \{a\}, r_1 \rangle$ con probabilidad $\frac{r_2}{r_1+r_2+r_3} \cdot \frac{r_3}{r_1+r_3}$
 $\langle \{a\}, r_3 \rangle \cdot \langle \{a\}, r_1 \rangle \cdot \langle \{\widehat{a}, \widehat{a}\}, r_2 \rangle$ con probabilidad $\frac{r_3}{r_1+r_2+r_3} \cdot \frac{r_1}{r_1+r_2}$
 $\langle \{a\}, r_3 \rangle \cdot \langle \{\widehat{a}, \widehat{a}\}, r_2 \rangle \cdot \langle \{a\}, r_1 \rangle$ con probabilidad $\frac{r_3}{r_1+r_2+r_3} \cdot \frac{r_2}{r_1+r_2}$.

□

Ejemplo 3.8 Consideremos ahora:

$$E_4 = (\langle \{a\}, r_1 \rangle \parallel \langle \{\widehat{a}, \widehat{a}\}, r_2 \rangle \parallel \langle \{a\}, r_3 \rangle) \text{ sy } a = E_3 \text{ sy } a$$

cuyo sistema de transiciones correspondiente es el mostrado en la figura 3.4,

Figura 3.4: Sistema de transiciones de $\overline{E_4}$

donde $H'_i \equiv H_i$ sy a , siendo los H_i los mismos que en el ejemplo anterior.

Para E_4 sería posible realizar también los 6 posibles comportamientos de E_3 (aunque con diferente probabilidad), correspondientes a la ejecución de las tres multiacciones estocásticas de γ_1 , donde γ_1 es una bolsa concurrente de G_3 , con:

$$G_3 = \overline{\langle a, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel \overline{\langle a, r_3 \rangle}$$

y, por tanto, γ_1 es una bolsa concurrente de G_3 *sy a*.

Como resultado de las sincronizaciones obtenemos además las siguientes nuevas bolsas concurrentes de G_3 *sy a*:

$$\begin{aligned} \gamma_2 &= \{ \langle \{\hat{a}\}, r_{12} \rangle, \langle \{a\}, r_3 \rangle \} \quad \text{con} \quad r_{12} = \min\{r_1, r_2\} \\ \gamma_3 &= \{ \langle \{\hat{a}\}, r_{23} \rangle, \langle \{a\}, r_1 \rangle \} \quad \text{con} \quad r_{23} = \min\{r_2, r_3\} \\ \gamma_4 &= \{ \langle \emptyset, r_{123} \rangle \} \quad \text{con} \quad r_{123} = \min\{r_1, r_2, r_3\} \end{aligned}$$

En total, E_4 tendrá $3!+2!+2!+1=11$ comportamientos posibles.

Por ejemplo:

$$\langle \{a\}, r_1 \rangle \cdot \langle \{\hat{a}, \hat{a}\}, r_2 \rangle \cdot \langle \{a\}, r_3 \rangle$$

con probabilidad

$$\frac{r_1}{r_1 + r_2 + r_3 + r_{12} + r_{23} + r_{123}} \cdot \frac{r_2}{r_2 + r_3 + r_{23}}$$

O también sería posible el camino:

$$\langle \{\hat{a}\}, r_{12} \rangle \cdot \langle \{a\}, r_3 \rangle$$

con probabilidad:

$$\frac{r_{12}}{r_1 + r_2 + r_3 + r_{12} + r_{23} + r_{123}}$$

También existiría la posibilidad de realizar una única multiacción estocástica:

$$\langle \emptyset, r_{123} \rangle$$

en este caso, con probabilidad:

$$\frac{r_{123}}{r_1 + r_2 + r_3 + r_{12} + r_{23} + r_{123}}$$

□

Ejemplo 3.9 Sea ahora

$$E_5 = ((\langle a, 1 \rangle \square \langle a, 1 \rangle) \parallel \langle \{\hat{a}, \hat{a}, b\}, 1 \rangle \parallel \langle a, 1 \rangle) \text{ sy } a$$

3.2.6 El proceso estocástico asociado

El proceso estocástico asociado a la evolución de una s-expresión dinámica \overline{E} es un proceso de Markov, en particular, una Cadena de Markov de Tiempo Continuo (CTMC).

Proposición 3.3 Dada una s-expresión estática E , definimos el proceso estocástico asociado $X(t)$, mediante $X(t) = [G]_{\Xi}$, para cada evolución posible en tiempo t de la s-expresión dinámica \overline{E} en una s-expresión dinámica en la clase $[G]_{\Xi}$, con $G \in [\overline{E}]$. Entonces, $X(t)$ es un proceso de Markov.

Demostración:

Es inmediata, al tratarse de distribuciones exponenciales y manejarse la política de competencia, teniéndose así que el tiempo de permanencia en cada estado también está distribuido exponencialmente (proposición 3.2). \square

La obtención del diagrama de transiciones de la CTMC es inmediata a partir del sistema de transiciones $ts(\overline{E})$: los nodos y los arcos se conservan, y si sólo hay un arco entre dos nodos, éste se etiqueta con la razón asociada a la multiacción estocástica que etiquetaba el arco en $ts(\overline{E})$, mientras que si hay dos o más arcos entre dos nodos en $ts(\overline{E})$, los unimos en uno único, etiquetado con la suma de las razones.

Ejemplo 3.10 Para la s-expresión estática siguiente:

$$E_1 = (\langle \{a\}, r_1 \rangle; \langle \{b\}, r_2 \rangle) \square \langle \{c\}, r_3 \rangle$$

el diagrama de transición de estados correspondiente a su CTMC puede verse en la figura 3.6.

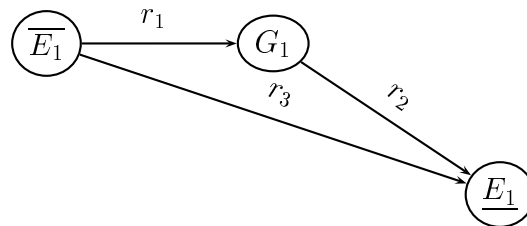


Figura 3.6: Diagrama de transición de estados de la CTMC asociada a \overline{E}_1

\square

Ejemplo 3.11 Para la s-expresión estática siguiente:

$$E_2 = \langle \{a\}, r_1 \rangle ; (\langle \{b\}, r_2 \rangle \square \langle \{c\}, r_3 \rangle)$$

el diagrama de transición de estados correspondiente a su CTMC puede verse en la figura 3.7.

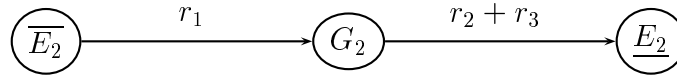


Figura 3.7: Diagrama de transición de estados de la CTMC asociada a $\overline{E_2}$

□

Ejemplo 3.12 Para la s-expresión estática siguiente:

$$E_3 = \langle \{a\}, r_1 \rangle \parallel \langle \{\hat{a}, \hat{a}\}, r_2 \rangle \parallel \langle \{a\}, r_3 \rangle$$

el diagrama de transición de estados correspondiente a su CTMC puede verse en la figura 3.8.

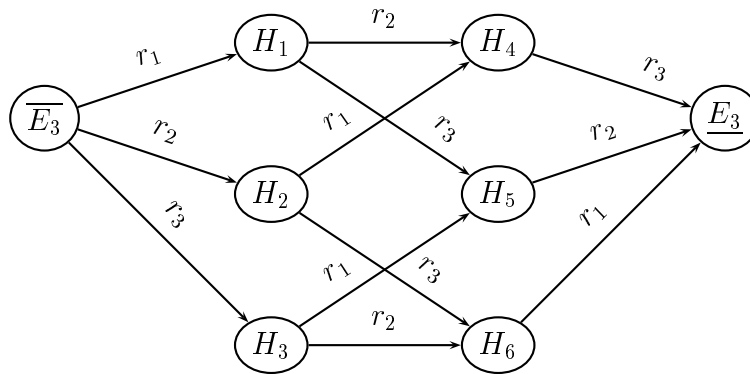


Figura 3.8: Diagrama de transición de estados de la CTMC asociada a $\overline{E_3}$

□

Ejemplo 3.13 Para la s-expresión estática siguiente:

$$E_4 = (\langle \{a\}, r_1 \rangle \parallel \langle \{\hat{a}, \hat{a}\}, r_2 \rangle \parallel \langle \{a\}, r_3 \rangle) \text{ sy } a = E_3 \text{ sy } a$$

el diagrama de transición de estados correspondiente a su CTMC puede verse en la figura 3.9.

□

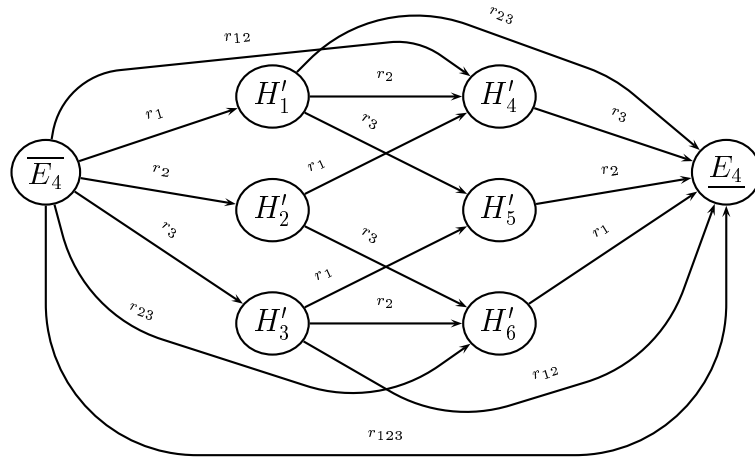


Figura 3.9: Diagrama de transición de estados de la CTMC asociada a $\overline{E_4}$

Ejemplo 3.14 Finalmente, si tomamos:

$$E_5 = ((\langle a, 1 \rangle \square \langle a, 1 \rangle) \parallel \langle \{\hat{a}, \hat{a}, b\}, 1 \rangle \parallel \langle a, 1 \rangle) \text{ sy } a$$

el diagrama de transición de estados correspondiente a su CTMC puede verse en la figura 3.10.

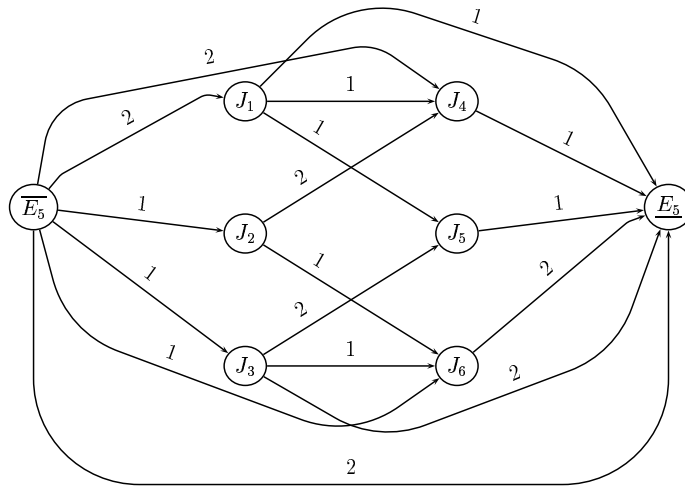


Figura 3.10: Diagrama de transición de estados de la CTMC asociada a $\overline{E_5}$

□

3.3 Semántica denotacional

En este apartado vamos a definir una semántica denotacional de redes de Petri estocásticas para las s-expresiones. Para ello introducimos una clase especial de redes

de Petri, a partir de las redes de Petri estocásticas [AM89, AMBC⁺95, AMBD98] y las cajas de Petri [BDK01a, BDK98, BDK01b, BK95]. Entonces, los objetos semánticos que utilizaremos los llamaremos cajas de Petri estocásticas o *s-cajas*. Toman la estructura de las cajas de Petri, tomando como soporte las redes de Petri estocásticas.

3.3.1 Las s-cajas planas

Recordemos primero la definición de caja en PBC:

Observación 3.3 $\Sigma = (S, T, W, \lambda)$ es una caja si (S, T, W) es una red de Petri, y λ es una función de reetiquetación, definida como sigue:

- $\lambda : S \longrightarrow \{e, i, x\}$

donde por convenio llamaremos

$${}^{\circ}\Sigma = \{s \in S \quad : \quad \lambda(s) = e\}$$

$$\Sigma^{\circ} = \{s \in S \quad : \quad \lambda(s) = x\}$$

al conjunto de lugares de entrada y salida, respectivamente.

- $\lambda : T \longrightarrow \mathcal{B}(\mathcal{L}) \times \mathcal{L}$

$\lambda(t)$ es una relación que asocia elementos de \mathcal{L} a bolsas de multiacciones. De hecho diremos que una caja es plana si $\forall t \in T$, $\lambda(t)$ es una relación constante, es decir, un elemento de \mathcal{L} .

Las cajas se definen como redes de Petri etiquetadas con las siguientes restricciones:

- ${}^{\circ}\Sigma \neq \emptyset \neq \Sigma^{\circ}$ (al menos hay un lugar de entrada y uno de salida).
- $\bullet({}^{\circ}\Sigma) = \emptyset = (\Sigma^{\circ})\bullet$ (ni llegan arcos a los lugares de entrada, ni salen arcos de los lugares de salida).
- $\forall t \in T : \bullet t \neq \emptyset \neq t\bullet \wedge \bullet t \cap t\bullet = \emptyset$ (la red es T-restrictiva y pura).
- $W(s, t) \in \{0, 1\} \wedge W(t, s) \in \{0, 1\} \quad \forall s \in S, \forall t \in T$ (red de Petri ordinaria).

□

Definición 3.12 Una caja plana estocástica, o *s-caja plana*, es una tupla:

$$\Sigma = (S, T, W, \lambda, \mu)$$

donde $\Sigma_1 = (S, T, W, \lambda)$ es una caja plana y $\mu : T \rightarrow \mathbb{R}^+$ es la función que asocia a cada transición la razón de una distribución exponencial (negativa) que lleva asociada, esto es, $\Sigma_2 = (S, T, W, \mu)$ es una red de Petri estocástica. \square

Una s-caja plana puede estar marcada o no. Al igual que en PBC, el marcado inicial se denotará por M_e y representará el marcado en el que sólo los lugares de entrada están marcados, cada uno con un token; y el marcado final, denotado por M_x , representará el marcado en el que sólo los lugares de salida están marcados, cada uno con un token.

Así, una s-caja plana es una clase especial de red de Petri estocástica etiquetada, cuyo comportamiento sigue la clásica *regla de disparo* de las redes de Petri estocásticas, en cuanto a transiciones activadas, disparo y marcado alcanzado. En consecuencia, a nivel funcional es como la red de Petri asociada (con semántica de *entremezcla*), y en cuanto a evaluación de prestaciones, se estudiará la CTMC asociada a la red de Petri estocástica (ver [AMBC⁺95]).

3.3.2 El álgebra de las s-cajas

Para poder definir el álgebra de las s-cajas introducimos en la figura 3.11 una colección de *s-cajas operadores*, apoyándonos en las cajas operadores clásicas de PBC.

Las relaciones de reetiquetado $\rho_{op} \subseteq \mathcal{B}(\mathcal{SL}) \times \mathcal{SL}$ que aparecen en la figura 3.11 se definen como sigue:

- $\rho_{id} = \{(\{\langle \alpha, r \rangle\}, \langle \alpha, r \rangle) \mid \langle \alpha, r \rangle \in \mathcal{SL}\}$
- $\rho_{[f]} = \{(\{\langle \alpha, r \rangle\}, \langle f(\alpha), r \rangle) \mid \langle \alpha, r \rangle \in \mathcal{SL}\}$
- $\rho_{rs\ a} = \{(\{\langle \alpha, r \rangle\}, \langle \alpha, r \rangle) \mid \langle \alpha, r \rangle \in \mathcal{SL} \wedge a, \hat{a} \notin A(\alpha)\}$
- $\rho_{sy\ a}$ es la menor relación de reetiquetado que contiene a ρ_{id} , tal que:

$$\begin{aligned} (\Gamma, \langle \alpha + \{a\}, r_1 \rangle) \in \rho_{sy\ a} \wedge (\Delta, \langle \beta + \{\hat{a}\}, r_2 \rangle) \in \rho_{sy\ a} \\ \Rightarrow (\Gamma + \Delta, \langle \alpha + \beta, \min\{r_1, r_2\} \rangle) \in \rho_{sy\ a} \end{aligned}$$
- $\rho_{[a:\]} = \{(\Gamma, \langle \alpha, r \rangle) \in \rho_{sy\ a} \mid a, \hat{a} \notin A(\alpha)\}$

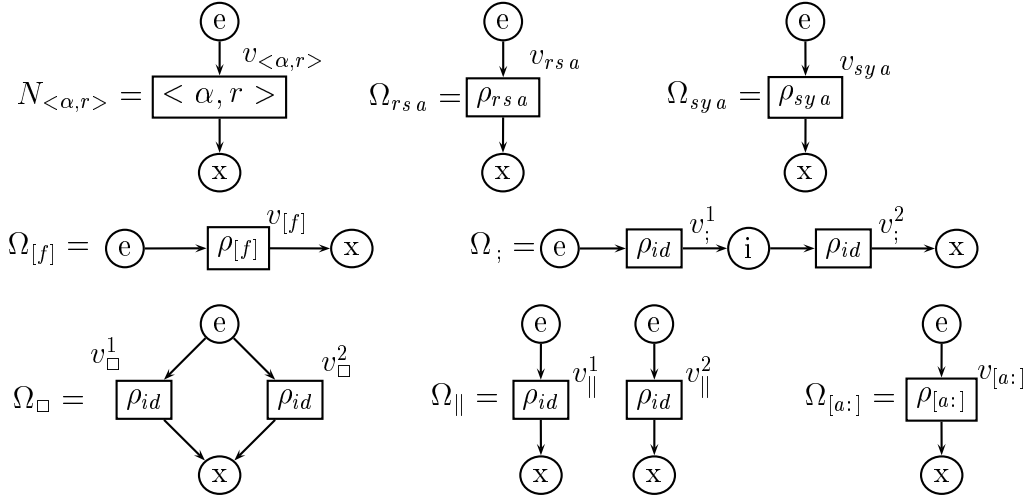


Figura 3.11: s-cajas operador de sPBC finito

En ρ_{sya} tendremos en cuenta observaciones similares a las incorporadas en la Observación 3.1; es decir, a la hora de generar nuevas transiciones por efecto de las sincronizaciones, tomaremos una única copia en aquellos casos en las que las transiciones involucradas sean las mismas, para determinar lo cual basta con numerar las transiciones de partida. Esta observación no era necesaria en PBC, ni en la semántica operacional, ni en la semántica de cajas, debido a que la duplicación de transiciones no nos afectaba en términos semánticos, pero en el contexto de las redes de Petri estocásticas es evidente que hay que ser más cuidadoso al respecto.

Para obtener las cajas planas, utilizamos el mismo mecanismo que en PBC, combinando el refinamiento y el reetiquetado (ver apartado 2.1.3 del capítulo 2). De esta forma, como usamos s-cajas planas que tienen exactamente la misma estructura que las cajas planas de PBC, obtendremos también s-cajas seguras y limpias.

La semántica denotacional se define formalmente mediante el siguiente homomorfismo, que asocia una s-caja plana a cada s-expresión estática:

$$\begin{aligned} \text{Box}_s(\langle \alpha, r \rangle) &= N_{\langle \alpha, r \rangle} \\ \text{Box}_s(\text{op}(E_1, \dots, E_n)) &= \Omega_{\text{op}}(\text{Box}_s(E_1), \dots, \text{Box}_s(E_n)) \end{aligned}$$

Asimismo, tomamos M_e como marcado inicial para la s-caja plana así generada.

A continuación, mostramos que para toda una s-expresión estática E , la semántica operacional de \overline{E} y la semántica de la correspondiente s-caja plana son isomorfas.

Teorema 3.1 Para toda s-expresión estática E de sPBC finito, el sistema de transiciones $ts(\overline{E})$ y el grafo de accesibilidad de la red de Petri estocástica marcada $(Box_s(E), M_e)$ son isomorfos.

Demostración:

Si prescindimos de la información estocástica tanto en el modelo algebraico, como en el modelo de redes, obtenemos el mismo isomorfismo que en PBC, pues en ambos casos estamos considerando únicamente la semántica de orden total (*semántica de entremezcla*). Por lo tanto, la estructura del sistema de transiciones de \overline{E} y la del grafo de accesibilidad de la red, a nivel funcional, es exactamente la misma, pues en ambos casos hemos aplicado el mismo criterio para eliminar las evoluciones "redundantes". Podemos establecer una asociación entre las transiciones que quitamos en $ts(\overline{E})$ y en el grafo de accesibilidad de la red, observando la numeración realizada en el álgebra (etiquetando las multiacciones de izquierda a derecha, según el orden de escritura y concatenando al realizar la sincronización), y la realizada en la red, etiquetando cada transición de la red exactamente con el mismo número que empleamos en el álgebra para etiquetar la multiacción correspondiente. De todo ello concluimos que los dos sistemas de transiciones son isomorfos.

Asimismo, dado que en ambos casos (álgebra y redes) los conflictos se resuelven de la misma forma, esto es, aplicando la política de competencia (*race condition*), y dado que las multiacciones estocásticas obtenidas en el álgebra y las que etiquetan la correspondiente transición en la red son exactamente iguales, obtenemos que ambos sistemas de transiciones se comportan de la misma forma. \square

Corolario 3.1 La CTMC asociada a $ts(\overline{E})$ y la CTMC asociada a $(Box_s(E), M_e)$ son isomorfos. \square

3.3.3 Ejemplos

A continuación, mostramos las s-cajas planas asociadas a los ejemplos 3.5 a 3.9.

Ejemplo 3.15 Si consideremos el término

$$E_1 = (\langle \{a\}, r_1 \rangle; \langle \{b\}, r_2 \rangle) \square \langle \{c\}, r_3 \rangle$$

su s-caja plana correspondiente, $Box_s(E_1)$, es la mostrada en la figura 3.12.

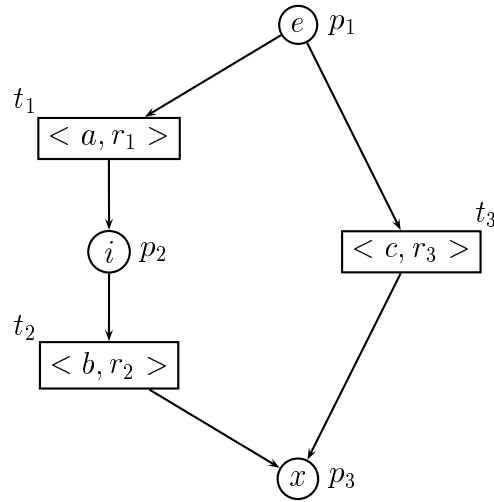


Figura 3.12: s-caja plana de E_1

Además el grafo de accesibilidad de $(Box_s(E_1), M_e)$ puede observarse en la figura 3.13, donde hemos considerado:

$$M_e = p_1, \quad M_1 = p_2, \quad M_x = p_3$$

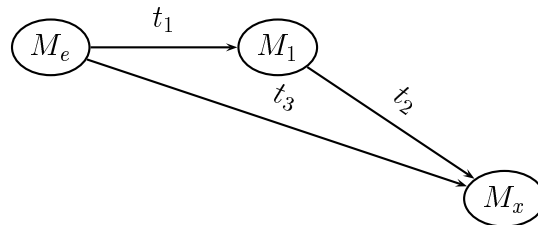


Figura 3.13: Grafo de accesibilidad de (Box_{E_1}, M_e)

Puede constatarse que dicho grafo de accesibilidad es isomorfo al sistema de transiciones de $\overline{E_1}$ (ver figura 3.1). □

Ejemplo 3.16 Si consideramos el término

$$E_2 = \langle \{a\}, r_1 \rangle ; (\langle \{b\}, r_2 \rangle \square \langle \{c\}, r_3 \rangle)$$

su s-caja plana asociada, $Box_s(E_2)$, es la mostrada en la figura 3.14. De nuevo, el grafo de accesibilidad de $(Box_s E_2, M_e)$ es isomorfo al sistema de transiciones de $\overline{E_2}$. □

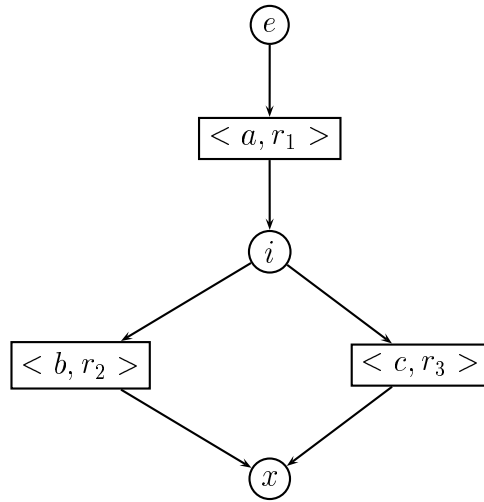


Figura 3.14: s-caja plana de E_2

Ejemplo 3.17 Siendo $E_3 = \langle \{a\}, r_1 \rangle \parallel \langle \{\hat{a}, \hat{a}\}, r_2 \rangle \parallel \langle \{a\}, r_3 \rangle$, su s-caja plana asociada, $Box_s(E_3)$, es la mostrada en la figura 3.15. Una vez más, el grafo de accesibilidad de $(Box_s E_3, M_e)$ es isomorfo al sistema de transiciones de $\overline{E_3}$.

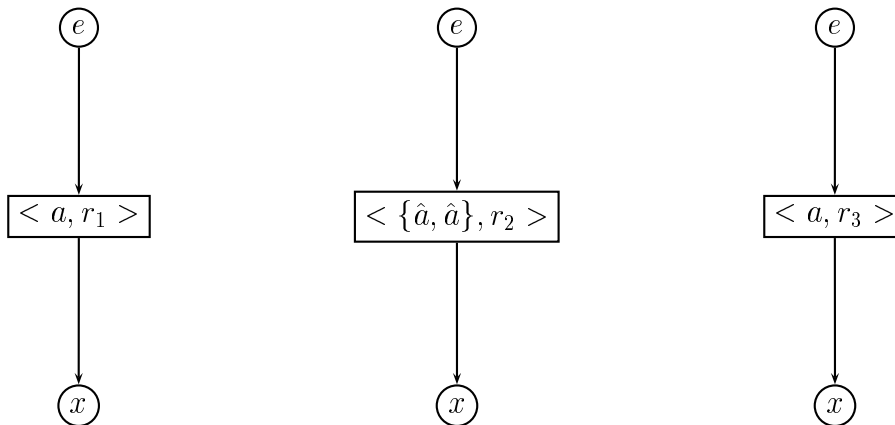


Figura 3.15: s-caja plana de E_3

□

Ejemplo 3.18 Siendo

$$E_4 = (\langle \{a\}, r_1 \rangle \parallel \langle \{\hat{a}, \hat{a}\}, r_2 \rangle \parallel \langle \{a\}, r_3 \rangle) \text{ sy } a = E_3 \text{ sy } a$$

su s-caja plana asociada, $Box_s(E_4)$, es la mostrada en la figura 3.16, donde $r_{12} = \min\{r_1, r_2\}$, $r_{23} = \min\{r_2, r_3\}$ y $r_{123} = \min\{r_1, r_2, r_3\}$. El grafo de accesibilidad de $(Box_s(E_4), M_e)$ es isomorfo al sistema de transiciones asociado a $\overline{E_4}$.

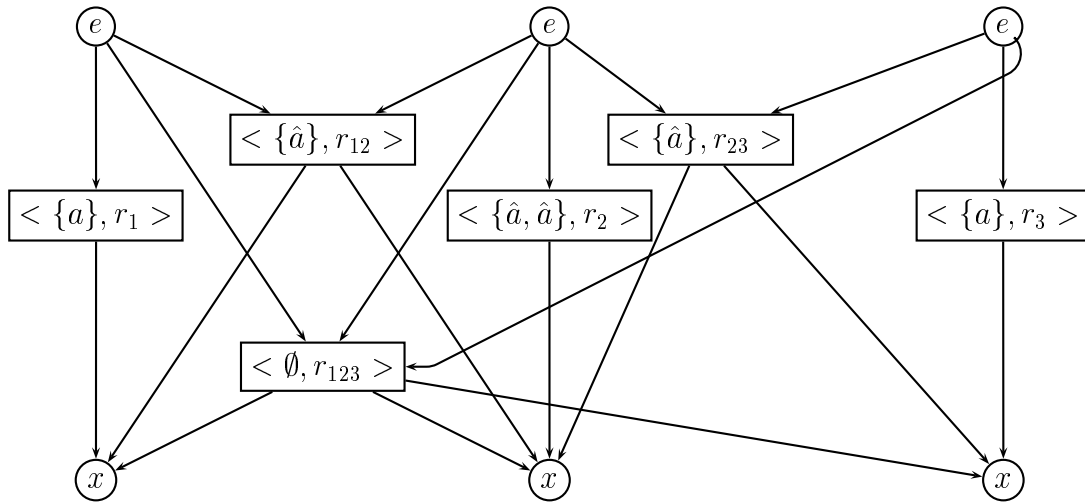


Figura 3.16: s-caja plana asociada de E_4

□

Ejemplo 3.19 Finalmente para

$$E_5 = ((\langle a, 1 \rangle \square \langle a, 1 \rangle) \parallel \langle \hat{a}, \hat{a}, b \rangle, 1 \parallel \langle a, 1 \rangle) \text{ sy } a$$

su s-caja plana asociada, $Box_s(E_5)$ es la red mostrada en la figura 3.17. De nuevo, el grafo de accesibilidad de $(Box_s(E_5), M_e)$ es isomorfo al sistema de transiciones de $\overline{E_5}$.

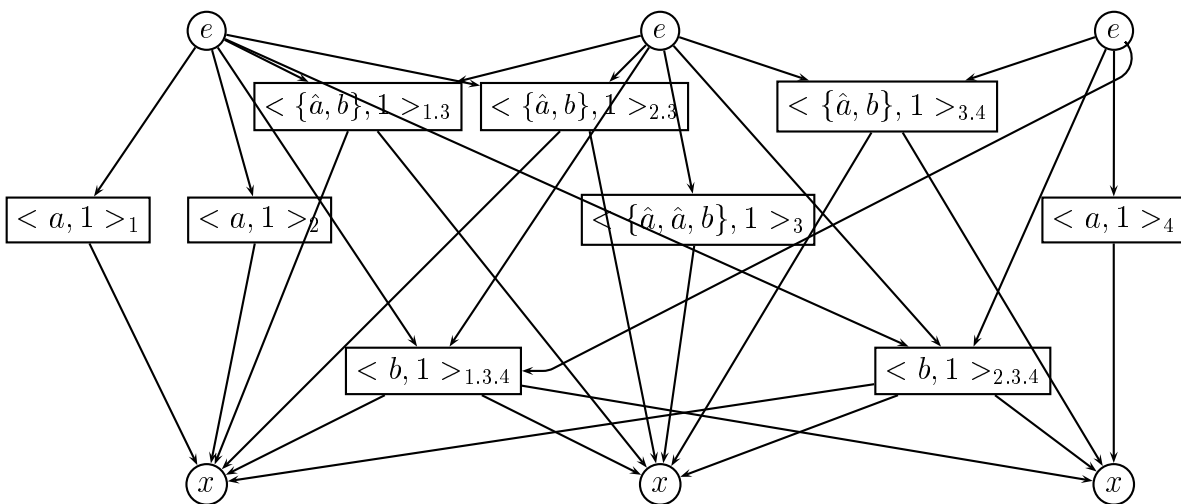


Figura 3.17: s-caja plana de E_5

□

Capítulo 4

Una nueva propuesta para la sincronización

Estudiando con detalle el modelo de sPBC introducido en el capítulo 3, encontramos ciertos problemas con la sincronización a la hora de obtener una equivalencia estocástica que fuese una congruencia (ver capítulo 6). Por ello nos planteamos una propuesta alternativa para la sincronización, que conservara en lo posible el *espíritu* de tomar el retraso de la multiacción más lenta al realizar una sincronización. En el primer apartado de este capítulo plantearemos los problemas que provocaron el cambio en la semántica del operador de sincronización. En el segundo apartado presentaremos formalmente la semántica operacional con la nueva propuesta, y en el siguiente, la semántica denotacional, así como la equivalencia entre las dos semánticas. Finalmente, haremos una comparativa con otras álgebras de procesos estocásticos y expondremos las limitaciones de nuestra propuesta.

4.1 Problemas con la sincronización

En este apartado introduciremos algunos ejemplos que motivaron nuestra decisión de modificar la semántica de la sincronización, y plantearemos la forma intuitiva de resolverlos.

Nuestra primera idea fue definir una equivalencia estocástica en la que dos s-expresiones fueran equivalentes si, a nivel funcional pueden realizar las mismas multiacciones estocásticas y, a nivel de estudio de prestaciones, tienen asociadas

cadenas de Markov isomorfas. Así, denotando la equivalencia por \sim , perseguimos que:

$$\langle \alpha, r \rangle \sim \langle \alpha, r_1 \rangle \square \langle \alpha, r_2 \rangle \square \dots \square \langle \alpha, r_n \rangle$$

siendo $r = r_1 + r_2 + \dots + r_n$, ya que a nivel funcional la s-expresión dinámica inicial correspondiente a la de la parte izquierda sólo podría realizar la multiacción α , al igual que la s-expresión dinámica inicial correspondiente a la de la parte derecha. Además, como $r = r_1 + r_2 + \dots + r_n$, entonces la CTMC asociada a las s-expresión de la parte izquierda es isomorfa a la CTMC asociada a la s-expresión de la parte derecha. Por otra parte, pretendemos que \sim sea una congruencia.

Informalmente, diremos que dos s-expresiones estáticas E y F son equivalentes, denotado por $E \sim F$, si los sistemas de transiciones asociados a \overline{E} y \overline{F} tienen el mismo número de nodos, pudiéndose establecer un isomorfismo entre los nodos, y además, los arcos que conectan nodos isomorfos en ambos sistemas de transiciones llevan asociadas las mismas multiacciones, siendo idénticas las CTMCs asociadas a ambos sistemas de transiciones (salvo el isomorfismo sobre el conjunto de los estados). Sin embargo, veremos que la relación de equivalencia así definida no es aún una congruencia, por lo que tendremos que enriquecer la definición para conseguir obtener una relación que sí sea una congruencia (ver capítulo 6). No obstante, esta relación es un razonable punto de partida de cara a lograr la equivalencia adecuada.

Ejemplo 4.1 Consideremos:

$$\begin{aligned} E_1 &= \langle \{a\}, 2 \rangle & , & & E_2 &= \langle \{a\}, 1 \rangle \square \langle \{a\}, 1 \rangle \\ F_1 &= \langle \{\hat{a}\}, 3 \rangle & , & & F_2 &= \langle \{\hat{a}\}, 1 \rangle \square \langle \{\hat{a}\}, 2 \rangle \end{aligned}$$

Entonces, de acuerdo con la definición (informal) anterior: $E_1 \sim E_2$ y $F_1 \sim F_2$, y también $E_1 \parallel F_1 \sim E_2 \parallel F_2$.

Sin embargo, si sincronizamos, tomando la menor de las razones para la multiacción resultado de la sincronización, obtenemos:

$$(E_1 \parallel F_1) \text{ sy } a \not\sim (E_2 \parallel F_2) \text{ sy } a$$

ya que en $(E_1 \parallel F_1) \text{ sy } a$ hay una nueva multiacción estocástica, $\langle \emptyset, 2 \rangle$. El sistema de transiciones de $\overline{(E_1 \parallel F_1) \text{ sy } a}$ se muestra en la figura 4.1.

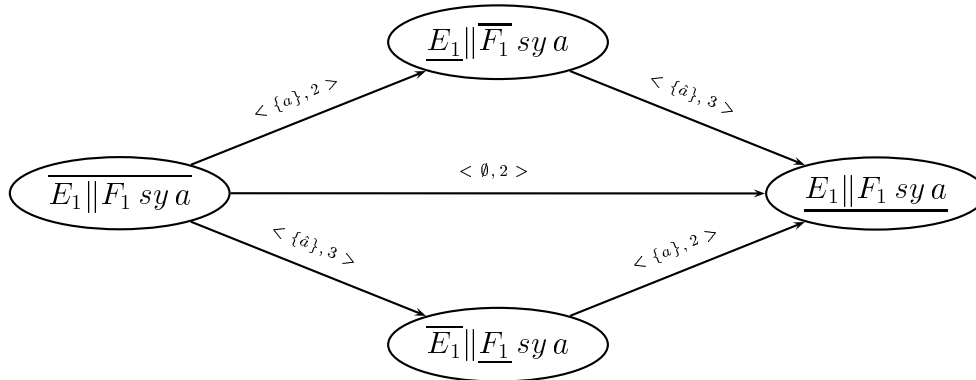


Figura 4.1: $ts(\overline{(E_1 \parallel F_1) sy a})$

Sin embargo, al considerar $(E_2 \parallel F_2) sy a$, cuando sincronizamos se originan cuatro nuevas multiacciones estocásticas, cuya suma de razones no es 2:

- $\langle \emptyset, 1 \rangle$ al sincronizar la primera componente de E_2 con la primera de F_2 ,
- $\langle \emptyset, 1 \rangle$ al sincronizar la primera componente de E_2 con la segunda de F_2 ,
- $\langle \emptyset, 1 \rangle$ al sincronizar la segunda componente de E_2 con la primera de F_2 ,
- $\langle \emptyset, 1 \rangle$ al sincronizar la segunda componente de E_2 con la segunda de F_2 .

El sistema de transiciones de $\overline{(E_2 \parallel F_2) sy a}$ se muestra en la figura 4.2. Claramente se ve que las CTMCs asociadas a los dos sistemas de transiciones no son isomorfas.

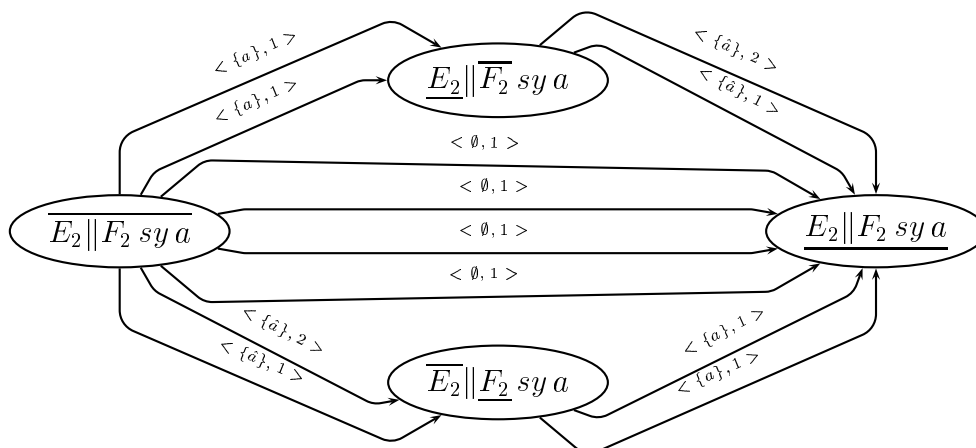


Figura 4.2: $ts(\overline{(E_2 \parallel F_2) sy a})$ con la primera propuesta para sy

□

El punto de referencia del que partiremos después corresponde con las *razones aparentes* de PEPA [Hil94a], que consisten en sumar las razones de las acciones estocásticas que tienen asociada el mismo tipo de acción (entre las que puede realizar el proceso, tanto si están en el ámbito de un operador paralelo, como si lo está en el de una elección). A partir de esta idea, nosotros consideraremos las *razones de conflicto*, en las que sumaremos todas las razones de aquellas multiacciones coincidentes que estén involucradas en una elección. En consecuencia, sólo sumaremos las razones involucradas en las elecciones, no las involucradas en operadores paralelos. Al sincronizar, como razón de la nueva multiacción tomaremos la mínima de las razones de conflicto, pero ponderada con el peso que tiene cada una de las multiacciones estocásticas envueltas en la sincronización dentro de su razón de conflicto.

De esta forma, volviendo al ejemplo anterior, en E_2 las dos multiacciones estocásticas $\langle \{a\}, 1 \rangle$ están en conflicto (con razón de conflicto 2, que es la suma de ambas razones), al igual que las dos multiacciones estocásticas $\langle \{\hat{a}\}, 1 \rangle$ y $\langle \{\hat{a}\}, 2 \rangle$ de F_2 (con razón de conflicto 3). Así, al sincronizar:

- La primera componente de E_2 con la primera de F_2 , tendremos que la razón asociada será: $\frac{1}{2} \cdot \frac{1}{3} \cdot \min\{2, 3\}$, originando la multiacción estocástica $\langle \emptyset, \frac{1}{3} \rangle$.
- La primera componente de E_2 con la segunda de F_2 , tendremos que la razón asociada será: $\frac{1}{2} \cdot \frac{2}{3} \cdot \min\{2, 3\}$, originando la multiacción estocástica $\langle \emptyset, \frac{2}{3} \rangle$.
- La segunda componente de E_2 con la primera de F_2 , tendremos que la razón asociada será: $\frac{1}{2} \cdot \frac{1}{3} \cdot \min\{2, 3\}$, originando la multiacción estocástica $\langle \emptyset, \frac{1}{3} \rangle$.
- La segunda componente de E_2 con la segunda de F_2 , tendremos que la razón asociada será: $\frac{1}{2} \cdot \frac{2}{3} \cdot \min\{2, 3\}$, originando la multiacción estocástica $\langle \emptyset, \frac{2}{3} \rangle$.

En consecuencia, la suma de estas cuatro multiacciones estocásticas es 2, coincidiendo, ahora sí, con la obtenida en $(E_1 \parallel F_1) \text{ sy } a$. El sistema de transiciones de $\overline{(E_2 \parallel F_2) \text{ sy } a}$ con la nueva propuesta de sincronización es el mostrado en la figura 4.3, donde ahora se puede apreciar que las CTMCs asociadas a $\overline{(E_1 \parallel F_1) \text{ sy } a}$ y a $\overline{(E_2 \parallel F_2) \text{ sy } a}$ son isomorfas.

Antes de definir formalmente las razones de conflicto y la nueva semántica (operacional y denotacional) para la sincronización, revisemos un ejemplo que se plantea en [Rib95]. A través del mismo mostraremos el principal inconveniente que tendríamos si utilizáramos sin más las *razones aparentes* de PEPA. El problema consiste en que

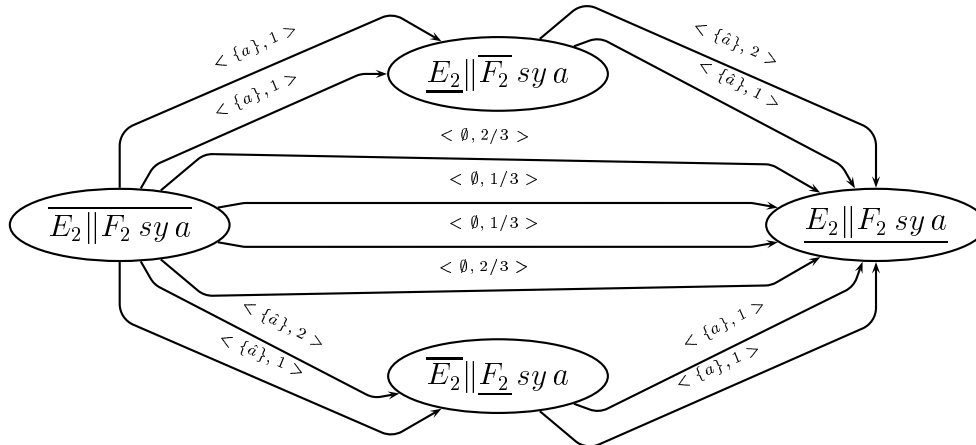


Figura 4.3: $ts(\overline{E_2} \parallel \overline{F_2} \text{ sy } a)$ con la nueva propuesta para sy

al definir una semántica de redes de Petri, las razones de las transiciones podrían depender en ocasiones del marcado de la red. Más tarde volveremos a este ejemplo, planteándolo en términos de sPBC, y veremos cómo ha quedado resuelto el problema.

Ejemplo 4.2 Hemos tomado de PEPA [Rib95] este ejemplo, en el que el proceso *Sys* se define mediante la composición paralela de tres componentes: dos de ellos sin sincronización alguna, $Comp_1$ y $Comp_2$, es decir, independientes, componiéndose el resultado en paralelo con $Comp_3$, sincronizando sobre las acciones de tipo α :

$$Sys = (Comp_1 \parallel Comp_2) \underset{\{\alpha\}}{\bowtie} Comp_3$$

$$Comp_1 = (\beta, \lambda_1).(\alpha, \lambda_2)$$

$$Comp_2 = (\alpha, \lambda_3) + (\alpha, \lambda_4)$$

$$Comp_3 = (\alpha, \lambda_5).(\alpha, \lambda_6)$$

Observemos que después de ejecutar (β, λ_1) , se podría ejecutar (α, λ_2) , de modo que al ser posible esta ejecución, su razón asociada debe tenerse en cuenta al calcular la razón aparente correspondiente a (α, λ_3) . Por lo tanto, si sincronizamos (α, λ_3) con (α, λ_5) , obtenemos como razón asociada para esta sincronización:

$$R'_{35} = \frac{\lambda_3}{\lambda_2 + \lambda_3 + \lambda_4} \cdot \frac{\lambda_5}{\lambda_5} \cdot \min\{\lambda_2 + \lambda_3 + \lambda_4, \lambda_5\}$$

De modo que en la red de Petri asociada habría que tomar esa razón para la transición correspondiente a la sincronización, pero sólo en el caso de haber disparado la transición correspondiente a (β, λ_1) . Sin embargo, si no se hubiese ejecutado (β, λ_1) , la razón obtenida habría sido:

$$R_{35} = \frac{\lambda_3}{\lambda_3 + \lambda_4} \cdot \frac{\lambda_5}{\lambda_5} \cdot \min\{\lambda_3 + \lambda_4, \lambda_5\}$$

□

De este ejemplo podemos concluir que en la semántica de red de Petri del modelo PEPA pueden aparecer transiciones cuyas razones dependen del marcado, lo que sin duda representa un serio inconveniente a la hora de realizar la traducción. Por ello pretendemos evitar este hecho en nuestro modelo. Para ello lo que haremos es distinguir las multiacciones en paralelo de aquellas que están en conflicto, afectadas por una elección.

Una vez introducidos los problemas que aparecerían, y diseñada informalmente una posible solución, nos queda formalizarla. Tenemos, entonces que redefinir la semántica operacional para el operador *sy*, redefinir su semántica de cajas y comprobar que los resultados de equivalencia del teorema 3.1 se mantienen.

4.2 La semántica operacional

Seguiremos manteniendo la sintaxis introducida en el apartado 3.1 del capítulo anterior, pero ahora sólo consideraremos funciones de renombramiento f que, además de conservar los conjugados, sean biyectivas. Pero para poder conseguir que el operador de sincronización tenga un buen comportamiento, en el sentido de que no importe el orden en que se aplica la regla de la sincronización, (siempre que se obtenga la misma multiacción estocástica), es necesario introducir una restricción sintáctica. Esta restricción sintáctica consistirá en no permitir un paralelo (al más alto nivel) en los parámetros de una elección. En el ejemplo 4.4 se podrá apreciar la necesidad de esta restricción, para poder garantizar el buen comportamiento del operador de sincronización.

Los términos que cumplan dicha restricción se llamarán *regulares*. De ahora en adelante siempre consideraremos términos regulares. Más formalmente, las s-expresiones estáticas regulares E son aquellas s-expresiones de sPBC generadas mediante el siguiente par de expresiones regulares:

$$\begin{aligned} D &::= \langle \alpha, r \rangle \mid D; E \mid D \text{ sy } a \mid D \text{ rs } a \mid D[f] \mid [a : D] \mid D \square D \\ E &::= \langle \alpha, r \rangle \mid E; E \mid E \text{ sy } a \mid E \text{ rs } a \mid E[f] \mid [a : E] \mid E \parallel E \mid D \square D \end{aligned}$$

donde llamamos *deterministas* a los D -términos.

Una s-expresión dinámica es regular si su correspondiente s-expresión estática lo es, y el conjunto de las s-expresiones dinámicas regulares operativas será denotado por *OpReDynExpr*.

Para redefinir la semántica operacional de sPBC, mantendremos las reglas de inacción de las tablas 3.1 y 3.2, las reglas de las transiciones estocásticas de la tabla 3.3, y la regla (*Sy1*) de la tabla 3.4, pero como después mostraremos, cambiaremos (*Sy2*) de la tabla 3.4 por (*Sy2'*) de la tabla 4.1. Insistimos en que siempre tendremos en cuenta que sólo consideraremos términos regulares. Para poder modificar la regla de la sincronización, primero necesitamos adaptar la definición de bolsa concurrente a la nueva propuesta de sincronización. Veamos por tanto la nueva definición de bolsas concurrentes de un término de *OpReDynExpr*.

Definición 4.1 Definimos $BC : OpReDynExpr \longrightarrow \mathcal{P}(\mathcal{B}(SL))$, de la forma siguiente:

- Si $G \in OpReDynExpr$ es final, esto es, $G = \underline{E}$, tomamos $BC(G) = \emptyset$.
- Si $G \in OpReDynExpr$ no es final, atendiendo a la estructura sintáctica de los términos de *OpReDynExpr*, distinguimos los siguientes casos:

- $BC(\overline{\langle \alpha, r \rangle}) = \{\{\langle \alpha, r \rangle\}\}$
- Si $\gamma \in BC(G)$, entonces: $\gamma \in BC(G; E)$, $\gamma \in BC(E; G)$, $\gamma \in BC(E \square G)$, $\gamma \in BC(G \square E)$, $\gamma \in BC(G \text{ rs } a)$ (cuando $a, \hat{a} \notin A(\gamma)$), $\gamma \in BC(G \text{ sy } a)$, y $f(\gamma) \in BC(G[f])$.
- Si $\gamma_1 \in BC(G)$, entonces $\gamma_1 \in BC(G \parallel H)$.
- Si $\gamma_2 \in BC(H)$, entonces $\gamma_2 \in BC(G \parallel H)$.
- Si $\gamma_1 \in BC(G)$, $\gamma_2 \in BC(H)$, entonces $\gamma_1 + \gamma_2 \in BC(G \parallel H)$.
- Si $\gamma \in BC(G \text{ sy } a)$, y $\langle \alpha, r_1 \rangle, \langle \beta, r_2 \rangle \in \gamma$, (con $\langle \alpha, r_1 \rangle \neq \langle \beta, r_2 \rangle$ ó bien son instancias diferentes de la misma multiacción estocástica en γ) con $a \in A(\alpha)$, y $\hat{a} \in A(\beta)$, entonces $\gamma' \in BC(G \text{ sy } a)$, donde:

$$\gamma' = (\gamma + \{\langle \alpha \oplus_a \beta, R \rangle\}) \setminus \{\langle \alpha, r_1 \rangle, \langle \beta, r_2 \rangle\}$$

y R es la razón de la nueva multiacción estocástica, la cual será definida más tarde (ver Regla *Sy2'* de la tabla 4.1). □

Como consecuencia de la restricción sintáctica introducida, podemos obtener la propiedad siguiente, que nos garantiza la ausencia de comportamientos paralelos dentro de una elección.

Proposición 4.1 Sea $G \in OpReDynExpr$, de forma que existen dos s-expresiones estáticas regulares E_1, E_2 , tales que:

$$G \equiv \overline{E_1 \square E_2}$$

Entonces, no es posible que existan $\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle$ tales que

$$\{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle\} \in BC(G)$$

Demostración:

Como G es regular tenemos que $G \equiv \overline{D_1 \square D_2}$, donde D_1 y D_2 son deterministas. Entonces podemos considerar $G \equiv \overline{D_1} \square D_2$ (respectivamente, $G \equiv D_1 \square \overline{D_2}$). De esta forma, se puede aplicar inducción estructural sobre D_1 (D_2), y con la definición de bolsa concurrente obtener la propiedad deseada:

- El caso base corresponde a $D_1 = \langle \alpha, r \rangle$, y es trivial, ya que consideramos $G = \overline{D_1} \square D_2$ y $BC(G) = \{\langle \alpha, r \rangle\}$.
- Para D_1 de la forma $D; E$, $D \text{ sy } a$, $D \text{ rs } a$, $D[f]$, y $[a : D]$ sólo necesitamos aplicar la hipótesis de inducción y la definición de bolsa concurrente.
- Para D_1 de la forma $D_{i_1} \square D_{i_2}$ podemos repetir el razonamiento, utilizando la hipótesis de inducción, para así llegar a la propiedad deseada. \square

Para poder definir las razones de las multiacciones estocásticas generadas en una sincronización, es necesario identificar las situaciones de conflicto. En concreto, para cada s-expresión dinámica operativa regular G , definiremos el multiconjunto de conflictos asociados para cada instancia de una multiacción estocástica $\langle \alpha, r \rangle_i$ que se puede ejecutar desde G , el cual se denotará por $Conflict(G, \langle \alpha, r \rangle_i)$, teniendo sólo en cuenta aquellas multiacciones estocásticas que están en elección con ella con la misma multiacción asociada α . Denotaremos este multiconjunto de conflictos por $Conflict(G, \langle \alpha, r \rangle_i)$, aunque omitiremos el subíndice i si está claro a qué instancia de $\langle \alpha, r \rangle$ nos estamos refiriendo.

Definición 4.2 Definimos la siguiente función parcial:

$$Conflict : OpReDynExpr \times \mathcal{SL} \longrightarrow \mathcal{B}(\mathcal{SL})$$

la cual para cada instancia i de la multiacción estocástica $\langle \alpha, r \rangle$ ejecutable desde G nos proporciona el multiconjunto de multiacciones estocásticas $\langle \alpha, r' \rangle$ en *conflicto* con ella. La función se define por inducción estructural¹:

¹donde omitimos el subíndice en $\langle \alpha, r \rangle_i$ para no cargar la notación.

1. $Conflict(\overline{\langle \alpha, r \rangle}, \langle \alpha, r \rangle) = \{\langle \alpha, r \rangle\}$
2. Si $\langle \alpha, r \rangle$ es ejecutable en G , y $C = Conflict(G, \langle \alpha, r \rangle)$, entonces:
 - (a) $Conflict(G; E, \langle \alpha, r \rangle) = Conflict(E; G, \langle \alpha, r \rangle) = C$,
 - (b) $Conflict(G||H, \langle \alpha, r \rangle) = Conflict(H||G, \langle \alpha, r \rangle) = C$,
 - (c) Si $a, \hat{a} \notin A(\alpha)$, entonces $Conflict(G \text{ rs } a, \langle \alpha, r \rangle) = C$,
 - (d) Para cada función biyectiva f , $Conflict(G[f], \langle f(\alpha), r \rangle) = f(C)$,
 - (e) Para el operador de elección necesitamos distinguir los dos casos siguientes:
 - Si $G \not\equiv \overline{E}$:
 $Conflict(G \square F, \langle \alpha, r \rangle) = Conflict(F \square G, \langle \alpha, r \rangle) = C$
 - Si $G \equiv \overline{E}$:
 $Conflict(G \square F, \langle \alpha, r \rangle) = Conflict(F \square G, \langle \alpha, r \rangle) = C +$
 $\{ \langle \alpha, r_j \rangle \mid \exists H_i \in OpReDynExpr, H_i \equiv \overline{F} \text{ y } H_i \xrightarrow{\langle \alpha, r_j \rangle} H'_i \}$
 - (f) $Conflict(G \text{ sy } a, \langle \alpha, r \rangle) = C$,

3. Sea $\{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle\} \in BC(G \text{ sy } a)$, $a \in A(\alpha_1)$, $\hat{a} \in A(\alpha_2)$ y $G \text{ sy } a \xrightarrow{\langle \alpha_1 \oplus_a \alpha_2, R_{12} \rangle} G' \text{ sy } a$ obtenida por aplicación de la regla $Sy2'$.

Entonces:

$$Conflict(G \text{ sy } a, \langle \alpha_1 \oplus_a \alpha_2, R_{12} \rangle) =$$

$$\{ \langle \alpha_1 \oplus_a \alpha_2, R_{ij} \rangle \mid \langle \alpha_1, r_i \rangle \in C_1, \langle \alpha_2, r_j \rangle \in C_2, \text{ donde}$$

$$R_{ij} = \frac{r_i}{cr(G \text{ sy } a, \langle \alpha_1, r_1 \rangle)} \frac{r_j}{cr(G \text{ sy } a, \langle \alpha_2, r_2 \rangle)} \cdot \min_{k=1,2} \{ cr(G \text{ sy } a, \langle \alpha_k, r_k \rangle) \} \}$$

con $C_k = Conflict(G \text{ sy } a, \langle \alpha_k, r_k \rangle)$, $k = 1, 2$, y $cr(G, \langle \alpha, r \rangle_k)$, llamado *razón de conflicto* para G y $\langle \alpha, r \rangle_k$, está definido como la suma de todas las razones de las multiacciones estocásticas que aparecen en el multiconjunto $Conflict(G, \langle \alpha, r \rangle_k)$:

$$cr(G, \langle \alpha, r \rangle_k) = \sum_{\langle \alpha, r_j \rangle \in Conflict(G, \langle \alpha, r \rangle_k)} r_j \cdot n_j$$

donde n_j es el número de veces que aparece $\langle \alpha, r_j \rangle$ en $Conflict(G, \langle \alpha, r \rangle_k)$. □

Obsérvese que $Conflict(G, \langle \alpha, r \rangle_i)$ es una función parcial, que está bien definida para cada multiacción estocástica $\langle \alpha, r \rangle_i$ ejecutable en G . El único caso que

requiere alguna explicación es cuando tenemos $G \text{ sy } a$ y una multiacción estocástica obtenida desde una sincronización: $\langle \alpha_1 \oplus_a \alpha_2, R_{12} \rangle$, ya que para ello necesitamos conocer $cr(G \text{ sy } a, \langle \alpha_i, r_i \rangle)$, para $i = 1, 2$. Entonces, ambas funciones son mutuamente recursivas, y no es difícil comprobar que se trata de una recursión bien fundada, ya que siempre acabamos llegando al caso de $(G \text{ sy } a, \langle \alpha_i, r_i \rangle)$, donde $\langle \alpha_i, r_i \rangle$ ya es ejecutable desde G .

Ya estamos en condiciones de definir las dos reglas de la semántica operacional para el operador de sincronización de acuerdo con la nueva propuesta. Estas se muestran en la tabla 4.1. Únicamente hemos modificado la regla *Sy2*, en ella podemos observar que tomamos como razón de la multiacción estocástica generada el mínimo de las razones de conflicto de $\langle \alpha_1, r_1 \rangle$, $\langle \alpha_2, r_2 \rangle$, ponderado por un factor, el cual se introduce para garantizar que la relación de equivalencia, que formalmente se introducirá en la definición 6.7 del capítulo 6, es una congruencia (ver teorema 6.1).

<p>(Sy1) $\frac{G \xrightarrow{\langle \alpha, r \rangle} H}{G \text{ sy } a \xrightarrow{\langle \alpha, r \rangle} H \text{ sy } a}$</p> <p>(Sy2') Sea $\{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle\} \in BC(G \text{ sy } a)$, $a \in A(\alpha_1)$, $\hat{a} \in A(\alpha_2)$, entonces</p> $\frac{G \text{ sy } a \xrightarrow{\langle \alpha_1, r_1 \rangle} G_1 \text{ sy } a \xrightarrow{(\emptyset)}^* G_1^* \text{ sy } a \xrightarrow{\langle \alpha_2, r_2 \rangle} G_{12} \text{ sy } a}{G \text{ sy } a \xrightarrow{\langle \alpha_1 \oplus_a \alpha_2, R \rangle} G_{12} \text{ sy } a}$ <p>$R = \frac{r_1}{cr(G \text{ sy } a, \langle \alpha_1, r_1 \rangle)} \cdot \frac{r_2}{cr(G \text{ sy } a, \langle \alpha_2, r_2 \rangle)} \cdot \min_{i=1,2} \{cr(G \text{ sy } a, \langle \alpha_i, r_i \rangle)\}$</p>

Tabla 4.1: Reglas para la nueva propuesta de sincronización

Para abreviar, y abusando en cierta forma de la notación, denotaremos a la multiacción estocástica obtenida por sincronización de las multiacciones estocásticas $\langle \alpha_1, r_1 \rangle$ y $\langle \alpha_2, r_2 \rangle$ mediante $\langle \alpha_1, r_1 \rangle \oplus_a \langle \alpha_2, r_2 \rangle$.

Presentamos a continuación un ejemplo para clarificar las últimas definiciones.

Ejemplo 4.3 Consideremos la siguiente s-expresión dinámica operativa regular:

$$G_1 = ((\overline{\langle \{a, \hat{a}\}, r_1 \rangle} \square \langle \{a, \hat{a}\}, r_2 \rangle) \| (\overline{\langle \{a, \hat{a}\}, r_1 \rangle} \square \langle \{b\}, r_3 \rangle)) \text{ sy } a$$

donde, de acuerdo a la propuesta de enumeración que establecimos en el capítulo anterior, numeramos las multiacciones estocásticas de izquierda a derecha, según el

orden de escritura. Obtenemos entonces:

$$G_1 = ((\overline{\langle \{a, \hat{a}\}, r_1 \rangle_1} \square \langle \{a, \hat{a}\}, r_2 \rangle_2) \| (\overline{\langle \{a, \hat{a}\}, r_1 \rangle_3} \square \langle \{b\}, r_3 \rangle_4)) \text{ sy } a$$

de forma que obtenemos:

$$\begin{aligned} \text{Conflict}(G_1, \langle \{a, \hat{a}\}, r_1 \rangle_1) &= \{ \langle \{a, \hat{a}\}, r_1 \rangle_1, \langle \{a, \hat{a}\}, r_2 \rangle_2 \} \\ \text{Conflict}(G_1, \langle \{a, \hat{a}\}, r_1 \rangle_3) &= \{ \langle \{a, \hat{a}\}, r_1 \rangle_3 \} \end{aligned}$$

Por lo tanto,

$$\begin{aligned} cr(G_1, \langle \{a, \hat{a}\}, r_1 \rangle_1) &= r_1 + r_2 \\ cr(G_1, \langle \{a, \hat{a}\}, r_1 \rangle_3) &= r_1 \end{aligned}$$

Las nuevas multiacciones estocásticas que aparecen al sincronizar se numeran concatenando las etiquetas de las expresiones argumento, de modo que:

$$\begin{aligned} \langle \{a, \hat{a}\}, r_1 \rangle_1 \oplus_a \langle \{a, \hat{a}\}, r_1 \rangle_3 &= \langle \{a, \hat{a}\}, R_1 \rangle_{1.3} \\ \langle \{a, \hat{a}\}, r_2 \rangle_2 \oplus_a \langle \{a, \hat{a}\}, r_1 \rangle_3 &= \langle \{a, \hat{a}\}, R_2 \rangle_{2.3} \end{aligned}$$

$$\text{siendo } R_1 = \frac{r_1}{r_1+r_2} \cdot \frac{r_1}{r_1} \cdot \min\{r_1+r_2, r_1\} = \frac{r_1^2}{r_1+r_2}$$

$$\text{y } R_2 = \frac{r_2}{r_1+r_2} \cdot \frac{r_1}{r_1} \cdot \min\{r_1+r_2, r_1\} = \frac{r_1 \cdot r_2}{r_1+r_2}$$

Finalmente, obtenemos:

$$\begin{aligned} \text{Conflict}(G_1, \langle \{a, \hat{a}\}, R_1 \rangle_{1.3}) &= \{ \langle \{a, \hat{a}\}, R_1 \rangle_{1.3}, \langle \{a, \hat{a}\}, R_2 \rangle_{2.3} \} \\ cr(G_1, \langle \{a, \hat{a}\}, R_1 \rangle_{1.3}) &= R_1 + R_2 = r_1 \end{aligned}$$

Podemos observar que $\langle \{a, \hat{a}\}, R_2 \rangle_{2.3}$ no es ejecutable desde la s-expresión operativa G_1 , pero sí lo será desde G'_1 , con $G_1 \equiv G'_1$, siendo

$$G'_1 = ((\langle \{a, \hat{a}\}, r_1 \rangle_1 \square \overline{\langle \{a, \hat{a}\}, r_2 \rangle_2}) \| (\overline{\langle \{a, \hat{a}\}, r_1 \rangle_3} \square \langle \{b\}, r_3 \rangle_4)) \text{ sy } a$$

□

Recordemos ahora la definición de conjunto de alcance de una s-expresión dinámica regular.

Definición 4.3 Para cada $G \in \text{ReDynExpr}$ definimos el conjunto de las s-expresiones dinámicas que pueden ser derivadas a partir de $[G]_{\equiv}$ de la forma siguiente:

$$\begin{aligned} [G] &= \{ G' \mid G' \in [G]_{\equiv} \} \cup \{ H' \in \text{ReDynExpr} \mid \exists \langle \alpha_1, r_1 \rangle, \dots, \langle \alpha_n, r_n \rangle \in \mathcal{SL} \\ &\quad \text{con } G \equiv G' \xrightarrow{\langle \alpha_1, r_1 \rangle} G_1 \equiv G'_1 \xrightarrow{\langle \alpha_2, r_2 \rangle} \dots G_{n-1} \equiv G'_{n-1} \xrightarrow{\langle \alpha_n, r_n \rangle} H \equiv H' \} \end{aligned}$$

Llamaremos *alcance de G* a este conjunto $[G]$.

□

Obsérvese que si G es regular, cualquier $H \in [G]$ es también regular.

En primer lugar veremos que dadas dos multiacciones estocásticas de una bolsa concurrente de G , los conflictos se conservan a lo largo de la ejecución. Este lema se utilizará para demostrar que en una sincronización no importa el orden con que se aplique la regla *Sy2'* (corolario 4.2).

Lema 4.1 Sea $G \in OpReDynExpr$ y $\gamma = \{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle\} \in BC(G)$, con

$$G \xrightarrow{\langle \alpha_1, r_1 \rangle} H \xrightarrow{\emptyset} H^* \xrightarrow{\langle \alpha_2, r_2 \rangle} J$$

entonces $Conflict(G, \langle \alpha_2, r_2 \rangle) = Conflict(H^*, \langle \alpha_2, r_2 \rangle)$.

Demostración: Por inducción estructural sobre la sintaxis de G .

- **Caso base.** Nuestro caso base será $G = G_1 \parallel G_2$, con $G_1, G_2 \in OpReDynExpr$, $\langle \alpha_1, r_1 \rangle$ es ejecutable en G_1 y $\langle \alpha_2, r_2 \rangle$ es ejecutable en G_2 , ya que partimos de $\gamma = \{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle\} \in BC(G)$. Entonces:

$$G = G_1 \parallel G_2 \xrightarrow{\langle \alpha_1, r_1 \rangle} G'_1 \parallel G_2 \xrightarrow{\emptyset} G'_1 \parallel G_2 \xrightarrow{\langle \alpha_2, r_2 \rangle} G'_1 \parallel G'_2$$

Y tomando $H = G'_1 \parallel G_2$, $H^* = G'_1 \parallel G_2$ y $J = G'_1 \parallel G'_2$, llegamos a:

$$\begin{aligned} Conflict(G_1 \parallel G_2, \langle \alpha_2, r_2 \rangle) &= Conflict(G_2, \langle \alpha_2, r_2 \rangle) = \\ &= Conflict(G'_1 \parallel G_2, \langle \alpha_2, r_2 \rangle) = Conflict(H^*, \langle \alpha_2, r_2 \rangle) \end{aligned}$$

con lo cual habríamos resuelto el caso base al llegar a:

$$Conflict(G, \langle \alpha_2, r_2 \rangle) = Conflict(H^*, \langle \alpha_2, r_2 \rangle)$$

- **Caso general:** Una simple aplicación de la hipótesis de inducción sobre G' resuelve los casos de la composición secuencial ($G = G'; E$), la restricción ($G = G' \text{ rs } a$) y el renombramiento ($G = G'[f]$).

Para el operador de elección podemos tener $G = G' \square F$ ó $G = F \square G'$. Debido a la proposición 4.1 no puede ocurrir que $G' \equiv \bar{E}$, para ninguna s-expresión estática E , es decir, G' no puede ser inicial, ni equivalente a ninguna expresión inicial (con respecto a \equiv). Por ello sólo necesitamos aplicar la hipótesis de inducción sobre G' y la definición de *Conflict*, ya que $Conflict(G, \langle \alpha_2, r_2 \rangle) = Conflict(G', \langle \alpha_2, r_2 \rangle)$.

Consideremos ahora el caso de la sincronización $G = G' \text{ sy } a$, el cual es un poco más complicado. Necesitaremos distinguir dos subcasos:

- Si $\gamma \in BC(G')$, el resultado es inmediato, ya que bastaría aplicar la hipótesis de inducción y la definición de *Conflict*, ya que en este caso $Conflict(G, \langle \alpha_2, r_2 \rangle) = Conflict(G', \langle \alpha_2, r_2 \rangle)$.
- Si $\gamma \notin BC(G')$, entonces al menos una de las multiacciones estocásticas de la bolsa ha sido obtenida por sincronización, es decir se ha conseguido aplicando la Regla *Sy2'*. Entonces podemos proceder por inducción sobre el número de veces que la regla ha sido utilizada. Por simplicidad, consideraremos el caso en que $\langle \alpha_2, r_2 \rangle$ ha sido obtenida al aplicar la Regla *Sy2'*, (para $\langle \alpha_1, r_1 \rangle$ se podría razonar de la misma forma, y si las dos se han obtenido aplicando la Regla *Sy2'*, entonces deberíamos utilizar una inducción doble).

Caso base. Consideraremos $\langle \alpha_2, r_2 \rangle = \langle \alpha'_2, r'_2 \rangle \oplus_a \langle \alpha''_2, r''_2 \rangle$.

Entonces, $\{\langle \alpha_1, r_1 \rangle, \langle \alpha'_2, r'_2 \rangle, \langle \alpha''_2, r''_2 \rangle\} \in BC(G')$, y

$$G = G' \text{ sy } a \xrightarrow{\langle \alpha_1, r_1 \rangle} H' \text{ sy } a \xrightarrow{(\emptyset)} H'^* \text{ sy } a = H^* \xrightarrow{\langle \alpha'_2, r'_2 \rangle \oplus_a \langle \alpha''_2, r''_2 \rangle} J' \text{ sy } a = J$$

Entonces, aplicando la hipótesis de inducción sobre G' para $\{\langle \alpha_1, r_1 \rangle, \langle \alpha'_2, r'_2 \rangle\}$ y $\{\langle \alpha_1, r_1 \rangle, \langle \alpha''_2, r''_2 \rangle\}$, tenemos que:

$$\begin{aligned} Conflict(G, \langle \alpha'_2, r'_2 \rangle) &= Conflict(G', \langle \alpha'_2, r'_2 \rangle) = Conflict(H'^*, \langle \alpha'_2, r'_2 \rangle) \\ Conflict(G, \langle \alpha''_2, r''_2 \rangle) &= Conflict(G', \langle \alpha''_2, r''_2 \rangle) = Conflict(H'^*, \langle \alpha''_2, r''_2 \rangle) \end{aligned}$$

Y por definición de *Conflict*,

$$Conflict(G, \langle \alpha'_2, r'_2 \rangle \oplus_a \langle \alpha''_2, r''_2 \rangle) = Conflict(H^*, \langle \alpha'_2, r'_2 \rangle \oplus_a \langle \alpha''_2, r''_2 \rangle)$$

con lo que tendríamos que la propiedad se cumple.

Caso general. Suponemos ahora que:

$$\langle \alpha_2, r_2 \rangle = \langle \alpha'_2, r'_2 \rangle \oplus_a \langle \alpha''_2, r''_2 \rangle$$

donde $\langle \alpha''_2, r''_2 \rangle$ se ha obtenido aplicando n-1 veces la Regla *Sy2'*.

Considerando $\{\langle \alpha_1, r_1 \rangle, \langle \alpha'_2, r'_2 \rangle, \langle \alpha''_2, r''_2 \rangle\} \in BC(G)$ y

$$G = G' \text{ sy } a \xrightarrow{\langle \alpha_1, r_1 \rangle} H' \text{ sy } a \xrightarrow{(\emptyset)} H'^* \text{ sy } a = H^* \xrightarrow{\langle \alpha'_2, r'_2 \rangle \oplus_a \langle \alpha''_2, r''_2 \rangle} J' \text{ sy } a = J$$

Entonces, por la hipótesis de la inducción interna:

$$Conflict(G, \langle \alpha''_2, r''_2 \rangle) = Conflict(H^*, \langle \alpha''_2, r''_2 \rangle)$$

y aplicando la hipótesis de la inducción estructural tenemos que:

$$\begin{aligned} \text{Conflict}(G, \langle \alpha'_2, r'_2 \rangle) &= \text{Conflict}(G', \langle \alpha'_2, r'_2 \rangle) = \\ \text{Conflict}(H'^*, \langle \alpha'_2, r'_2 \rangle) &= \text{Conflict}(H^*, \langle \alpha'_2, r'_2 \rangle) \end{aligned}$$

Así, obtenemos:

$$\begin{aligned} \text{Conflict}(G, \langle \alpha'_2, r'_2 \rangle \oplus_a \langle \alpha''_2, r''_2 \rangle) &= \\ \text{Conflict}(H^*, \langle \alpha'_2, r'_2 \rangle \oplus_a \langle \alpha''_2, r''_2 \rangle) &= \end{aligned}$$

Por lo que concluimos que en todos los casos se cumple la propiedad deseada. \square

Veamos a continuación cómo calcular la razón de la nueva multiacción estocástica obtenida después de aplicar varias veces la sincronización.

Proposición 4.2 Sea $G \in \text{OpReDynExpr}$, $\gamma = \{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle, \dots, \langle \alpha_n, r_n \rangle\} \in BC(G)$, y una serialización de γ :

$$G \xrightarrow{\langle \alpha_1, r_1 \rangle} G_1 \xrightarrow{(\emptyset)} G_1^* \xrightarrow{\langle \alpha_2, r_2 \rangle} \dots \xrightarrow{\langle \alpha_n, r_n \rangle} G_n$$

sobre la cual podemos aplicar $n - 1$ veces la Regla *Sy2'* para obtener una única transición de la forma $G \xrightarrow{\langle \beta, R \rangle} G_n$.

Entonces tenemos que:

- $R = \left(\prod_{k=1}^n \frac{r_k}{cr(G, \langle \alpha_k, r_k \rangle)} \right) \cdot \min_{k=1, \dots, n} \{cr(G, \langle \alpha_k, r_k \rangle)\}$
- $cr(G, \langle \beta, R \rangle) = \min_{k=1, \dots, n} \{cr(G, \langle \alpha_k, r_k \rangle)\}$

Demostración:

Procederemos por inducción sobre la cantidad n (número de multiacciones estocásticas en la bolsa, contabilizando repeticiones).

- **Caso base:** ($n = 2$)

Sólo tenemos que aplicar la Regla *Sy2'* una vez. Partimos de

$$\{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle\} \in BC(G)$$

y

$$G \xrightarrow{\langle \alpha_1, r_1 \rangle} G_1 \xrightarrow{(\emptyset)} G_1^* \xrightarrow{\langle \alpha_2, r_2 \rangle} G_{12}$$

Entonces, por definición de R ,

$$R = \frac{r_1}{cr(G, \langle \alpha_1, r_1 \rangle)} \cdot \frac{r_2}{cr(G, \langle \alpha_2, r_2 \rangle)} \cdot \min_{i=1,2} \{cr(G, \langle \alpha_i, r_i \rangle)\}$$

Además, aplicando la definición de *Conflict* para el operador de sincronización, llegamos a que $cr(G, \langle \beta, R \rangle) = \min_{i=1,2} \{cr(G, \langle \alpha_i, r_i \rangle)\}$.

- **Caso general:** Si la Regla *Sy2'* se ha aplicado $n - 1$ veces ($n > 2$), entonces tenemos:

$$\langle \beta, R \rangle = \langle \beta_1, R_1 \rangle \oplus_a \langle \beta_2, R_2 \rangle$$

donde $\langle \beta_1, R_1 \rangle$ se ha obtenido después de aplicar $k - 2$ sincronizaciones, y la otra multiacción estocástica, $\langle \beta_2, R_2 \rangle$, después de $n - k$ sincronizaciones. De esta forma, podemos aplicar la hipótesis de inducción en ambos casos y obtener:

$$R_1 = \prod_{i=1}^{k-1} \frac{r_i}{cr(G, \langle \alpha_i, r_i \rangle)} \cdot \min_{i=1, \dots, k-1} \{cr(G, \langle \alpha_i, r_i \rangle)\}$$

$$R_2 = \prod_{i=k}^n \frac{r_i}{cr(G, \langle \alpha_i, r_i \rangle)} \cdot \min_{i=k, \dots, n} \{cr(G, \langle \alpha_i, r_i \rangle)\}$$

y

$$cr(G, \langle \beta_1, R_1 \rangle) = \min_{i=1, \dots, k-1} \{cr(G, \langle \alpha_i, r_i \rangle)\}$$

$$cr(G, \langle \beta_2, R_2 \rangle) = \min_{i=k, \dots, n} \{cr(G, \langle \alpha_i, r_i \rangle)\}$$

Aplicando la definición de sincronización para ambas multiacciones estocásticas, tenemos que la nueva razón es de la forma:

$$R = \frac{R_1}{cr(G, \langle \beta_1, R_1 \rangle)} \cdot \frac{R_2}{cr(G, \langle \beta_2, R_2 \rangle)} \cdot \min_{i=1,2} \{cr(G, \langle \beta_i, R_i \rangle)\}$$

con lo cual,

$$R = \left(\prod_{k=1}^n \frac{r_k}{cr(G, \langle \alpha_k, r_k \rangle)} \right) \cdot \min_{k=1, \dots, n} \{cr(G, \langle \alpha_k, r_k \rangle)\}$$

$$\text{y } cr(G, \langle \beta, R \rangle) = \min_{k=1, \dots, n} \{cr(G, \langle \alpha_k, r_k \rangle)\}.$$

□

La siguiente proposición, que ya habíamos enunciado para la primera propuesta de sincronización, muestra que para cualquier s-expresión dinámica operativa regular G de sPBC finito, y cualquier bolsa concurrente de multiacciones estocásticas de G , cualquier serialización de la bolsa puede ejecutarse a partir de G .

Proposición 4.3 Dada una s-expresión dinámica operativa regular G , $\gamma \in BC(G)$, y una serialización de las multiacciones estocásticas de γ : $\langle \alpha_1, r_1 \rangle \dots \langle \alpha_n, r_n \rangle$, entonces existe una secuencia de transiciones de la forma:

$$G \xrightarrow{\langle \alpha_1, r_1 \rangle} G_1 \xrightarrow{(\emptyset)} G_1^* \xrightarrow{\langle \alpha_2, r_2 \rangle} G_2 \xrightarrow{(\emptyset)} G_2^* \xrightarrow{\langle \alpha_3, r_3 \rangle} \dots \xrightarrow{(\emptyset)} G_{n-1}^* \xrightarrow{\langle \alpha_n, r_n \rangle} G'$$

Además, todas las s-expresiones G' obtenidas por la serialización de γ son equivalentes con respecto a \equiv .

Demostración:

Es el mismo razonamiento que ya hicimos cuando demostramos esta proposición en el capítulo anterior para la propuesta primera de sincronización (ver proposición 3.1). Tan sólo es necesario precisar que al sincronizar las razones no varían con el orden en que se hagan las sincronizaciones, según establece la proposición 4.2. \square

Teniendo en cuenta la proposición anterior tendremos el siguiente corolario.

Corolario 4.1 Sea $G \in OpReDynExpr$, $\gamma \in BC(G)$, y una serialización de las multiacciones estocásticas de γ de la forma: $\langle \alpha_1, r_1 \rangle \dots \langle \alpha_n, r_n \rangle$. Entonces existe una secuencia de transiciones:

$$G \xrightarrow{\langle \alpha_1, r_1 \rangle} G_1 \xrightarrow{(\emptyset)} G_1^* \xrightarrow{\langle \alpha_2, r_2 \rangle} G_2 \xrightarrow{(\emptyset)} G_2^* \xrightarrow{\langle \alpha_3, r_3 \rangle} \dots \xrightarrow{(\emptyset)} G_{n-1}^* \xrightarrow{\langle \alpha_n, r_n \rangle} G'$$

con $Conflict(G, \langle \alpha_i, r_i \rangle) = Conflict(G_{i-1}^*, \langle \alpha_i, r_i \rangle)$, para $i = 2, \dots, n$.

Además, todas las s-expresiones obtenidas por la serialización de γ son equivalentes con respecto a \equiv .

Demostración:

Es una consecuencia inmediata de la proposición anterior y del lema 4.1. \square

Además, la sincronización se puede generalizar cuando se sincroniza sobre diferentes acciones, obteniéndose así el siguiente resultado.

Corolario 4.2 Sea $G \in OpReDynExpr$, $\gamma = \{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle, \dots, \langle \alpha_n, r_n \rangle\} \in BC(G)$, y dos permutaciones del conjunto $\{1, \dots, n\}$: $\{i_1, \dots, i_n\}$ y $\{j_1, \dots, j_n\}$. Suponiendo que hay dos serializaciones de γ :

$$\begin{aligned} G &\xrightarrow{\langle \alpha_{i_1}, r_{i_1} \rangle} G_1 \xrightarrow{(\emptyset)} G_1^* \xrightarrow{\langle \alpha_{i_2}, r_{i_2} \rangle} \dots \xrightarrow{\langle \alpha_{i_n}, r_{i_n} \rangle} G_n \\ G &\xrightarrow{\langle \alpha_{j_1}, r_{j_1} \rangle} G'_1 \xrightarrow{(\emptyset)} G'_1^* \xrightarrow{\langle \alpha_{j_2}, r_{j_2} \rangle} \dots \xrightarrow{\langle \alpha_{j_n}, r_{j_n} \rangle} G'_n \end{aligned}$$

sobre las cuales podemos aplicar $n - 1$ veces la Regla *Sy2'* (para las mismas acciones a_1, \dots, a_{n-1} , tal vez repetidas, pero el mismo número de veces en los dos casos), obteniéndose en cada caso una única transición $G \xrightarrow{\langle \beta_i, R_i \rangle} G_n$ y $G \xrightarrow{\langle \beta_j, R_j \rangle} G'_n$, respectivamente. Entonces:

$$G_n \equiv G'_n \quad \text{y} \quad \langle \beta_i, R_i \rangle = \langle \beta_j, R_j \rangle$$

Demostración:

A partir del corolario 4.1 tenemos que $G_n \equiv G'_n$. Por otro lado, tenemos $\beta_i = \beta_j$, de la definición de la sincronización de multiacciones. Finalmente, $R_i = R_j$ se obtiene aplicando la fórmula descrita en la proposición 4.2 para el cálculo de la razón de la multiacción obtenida por efecto de una sincronización. \square

El lema siguiente, de carácter técnico, nos será de utilidad en el capítulo 6, donde probaremos que una cierta relación de equivalencia que allí introduciremos es realmente una congruencia.

Lema 4.2 Sea $G \in OpReDynExpr$, y $\langle \alpha, r \rangle$ ejecutable en G , con

$$Conflict(G, \langle \alpha, r \rangle) = \{ \langle \alpha, r \rangle, \langle \alpha, r_1 \rangle, \dots, \langle \alpha, r_n \rangle \}$$

Entonces, para cada $\langle \alpha, r_i \rangle \in Conflict(G, \langle \alpha, r \rangle)$ existe $G_i \in OpReDynExpr$, con $G_i \equiv G$, $\langle \alpha, r_i \rangle$ ejecutable en G_i , y $Conflict(G_i, \langle \alpha, r_i \rangle) = Conflict(G, \langle \alpha, r \rangle)$.

Demostración:

Procederemos por inducción estructural sobre la sintaxis de G .

- **Caso base.** Consideraremos $G = \overline{\langle \alpha, r \rangle}$.
Este caso es inmediato, ya que $Conflict(G, \langle \alpha, r \rangle) = \{ \langle \alpha, r \rangle \}$.
- **Caso general.** Aplicando la hipótesis de inducción se resuelven los casos correspondientes a los operadores de composición secuencial, restricción, renombramiento y paralelo.

Para el operador de la elección, ($G = G' \square F$ ó $G = F \square G'$), necesitamos distinguir dos casos:

- Si $G' \not\equiv \overline{E}$, para toda s-expresión estática regular E , entonces sólo tenemos que aplicar la hipótesis de inducción sobre G' y la definición de *Conflict*, para obtener la propiedad.
- Si $G' \equiv \overline{E}$ para alguna s-expresión estática E , entonces también necesitamos diferenciar dos subcasos:
 - Si $\langle \alpha, r_i \rangle \in \text{Conflict}(G', \langle \alpha, r \rangle)$, entonces basta aplicar la hipótesis de inducción para obtener la propiedad.
 - Si $\langle \alpha, r_i \rangle \notin \text{Conflict}(G', \langle \alpha, r \rangle)$, entonces debido a la proposición 4.1, existe alguna s-expresión dinámica operativa H_i , $H_i \equiv \overline{F}$, tal que $\langle \alpha, r_i \rangle$ es la única multiacción estocástica ejecutable desde H_i . Si ahora consideramos $G_i = E \square H_i$ (ó $G_i = H_i \square E$ en el caso simétrico), se tiene que G_i es operativa, $G_i \equiv G$ y además $\text{Conflict}(G, \langle \alpha, r \rangle) = \text{Conflict}(G_i, \langle \alpha, r_i \rangle)$.

Para el operador de sincronización, $G = G' \text{ sy } a$, el razonamiento es algo más complicado, y tenemos que diferenciar de nuevo los diferentes casos posibles:

- Si $\langle \alpha, r \rangle$ es ejecutable en G' , entonces bastaría con aplicar la hipótesis de inducción para obtener la propiedad deseada.
- En otro caso, $\langle \alpha, r \rangle$ se ha obtenido al aplicar la Regla *Sy2'*. Aplicamos entonces un razonamiento inductivo sobre el número de veces que *Sy2'* se ha aplicado hasta obtener $\langle \alpha, r \rangle$:
 - **Caso base.** Suponemos que $\langle \alpha, r \rangle$ se ha obtenido tras aplicar *Sy2'* una sola vez. Entonces podemos tomar:

$$\langle \alpha, r \rangle = \langle \alpha_1, r' \rangle \oplus_a \langle \alpha_2, r'' \rangle$$

de lo cual se sigue que $\{\langle \alpha_1, r' \rangle, \langle \alpha_2, r'' \rangle\} \in BC(G' \text{ sy } a)$, y entonces:

$$G' \text{ sy } a \xrightarrow{\langle \alpha_1, r' \rangle} G'^{11} \text{ sy } a \xrightarrow{\emptyset} G'^{1*} \text{ sy } a \xrightarrow{\langle \alpha_2, r'' \rangle} G'^{12} \text{ sy } a$$

Además,

$$C_1 = \text{Conflict}(G' \text{ sy } a, \langle \alpha_1, r' \rangle) = \text{Conflict}(G', \langle \alpha_1, r' \rangle)$$

y por el corolario 4.1:

$$C_2 = \text{Conflict}(G' \text{ sy } a, \langle \alpha_2, r'' \rangle) = \text{Conflict}(G'^{1*} \text{ sy } a, \langle \alpha_2, r'' \rangle) = \text{Conflict}(G'^{1*}, \langle \alpha_2, r'' \rangle) = \text{Conflict}(G', \langle \alpha_2, r'' \rangle)$$

Si $\langle \alpha, r_i \rangle \in \text{Conflict}(G, \langle \alpha, r \rangle)$ entonces

$$\langle \alpha, r_i \rangle = \langle \alpha_1, r'_i \rangle \oplus_a \langle \alpha_2, r''_i \rangle,$$

con $\langle \alpha_1, r'_i \rangle \in C_1$ y $\langle \alpha_2, r''_i \rangle \in C_2$. Como en G' debemos de tener un operador paralelo, entonces podríamos aplicar la hipótesis de inducción sobre cada componente del mismo para así concluir que existen algunas s-expresiones dinámicas operativas G_{1_i} , con $G_{1_i} \equiv G'$, de forma que $\{\langle \alpha_1, r'_i \rangle, \langle \alpha_2, r''_i \rangle\} \in BC(G_{1_i})$, y:

$$\text{Conflict}(G_{1_i}, \langle \alpha_1, r'_i \rangle) = C_1, \quad \text{Conflict}(G_{1_i}, \langle \alpha_2, r''_i \rangle) = C_2$$

Entonces, tomando $G_i = G_{1_i}$ *sy a*, obtenemos que G_i es operativa, $G_i \equiv G$, $\langle \alpha, r_i \rangle$ es ejecutable en G_i , y:

$$\text{Conflict}(G_i, \langle \alpha, r_i \rangle) = \text{Conflict}(G, \langle \alpha, r \rangle)$$

- **Caso general.** Suponemos que $\langle \alpha, r \rangle$ se ha obtenido después de aplicar n veces la Regla *Sy2'*. Esto es:

$$\langle \alpha, r \rangle = \langle \alpha_1, r' \rangle \oplus_a \langle \alpha_2, r'' \rangle$$

donde $\langle \alpha_1, r' \rangle$ se ha obtenido después de aplicar k veces la Regla *Sy2'*, y $\langle \alpha_2, r'' \rangle$ se ha obtenido después de aplicar $n-k$ veces la Regla *Sy2'*. Ahora podríamos aplicar la hipótesis de inducción sobre $\langle \alpha_1, r' \rangle$ y sobre $\langle \alpha_2, r'' \rangle$, y proceder como en el caso base para concluir la propiedad buscada. \square

A continuación presentamos una serie de ejemplos. El primero de ellos pone de manifiesto la necesidad de la restricción sintáctica que hemos introducido. Consideraremos una s-expresión dinámica no regular, en la cual aparece un paralelo dentro de una elección y veremos los problemas que surgen.

Ejemplo 4.4 Consideremos la siguiente s-expresión dinámica operativa y *no regular*:

$$G_2 = (((\overline{\langle \{a\}, r_1 \rangle} \parallel \overline{\langle \{a\}, r_2 \rangle}) \square \langle \{a\}, r_3 \rangle) \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_4 \rangle}) \text{ sy } a$$

desde la cual se sigue que:

$$\{\langle \{a\}, r_1 \rangle, \langle \{a\}, r_2 \rangle, \langle \{\hat{a}, \hat{a}\}, r_4 \rangle\} \in BC(G_2)$$

Entonces, de acuerdo a la definición de bolsa concurrente basada sobre las razones de conflicto, tenemos que:

$$\gamma = \{ \langle \{a\}, r_1 \rangle, \langle \{\hat{a}\}, R_{24} \rangle \} \in BC(G_2)$$

donde $R_{24} = \frac{r_2}{r_2+r_3} \cdot \min\{r_2 + r_3, r_4\}$.

Sin embargo, no toda serialización de γ es ahora posible desde G_2 , ya que $\langle \{\hat{a}\}, R_{24} \rangle$ no puede ser ejecutada desde G'_2 , donde:

$$G_2 \xrightarrow{\langle \{a\}, r_1 \rangle} G'_2$$

$$G'_2 = (((\langle \{a\}, r_1 \rangle \parallel \overline{\langle \{a\}, r_2 \rangle}) \square \langle \{a\}, r_3 \rangle) \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_4 \rangle}) \text{ sy } a$$

Lo que sí podemos ejecutar desde G'_2 es $\langle \{\hat{a}\}, R'_{24} \rangle$, con $R'_{24} = \min\{r_2, r_4\}$, pero en general tenemos, sin embargo, que $R_{24} \neq R'_{24}$. \square

El siguiente ejemplo muestra que si cumplimos la restricción sintáctica introducida, entonces no importa el orden en que se realicen las sincronizaciones.

Ejemplo 4.5 Consideremos la siguiente s-expresión dinámica operativa:

$$G_3 = \overline{\langle a, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel (\overline{\langle a, r_3 \rangle} \square \langle a, r_4 \rangle) \text{ sy } a$$

con $\gamma = \{ \langle a, r_1 \rangle, \langle \{\hat{a}, \hat{a}\}, r_2 \rangle, \langle a, r_3 \rangle \} \in BC(G_3)$. Tenemos:

$$\text{Conflict}(G_3, \langle a, r_1 \rangle) = \{ \langle a, r_1 \rangle \}, \quad cr(G_3, \langle a, r_1 \rangle) = r_1$$

$$\text{Conflict}(G_3, \langle \{\hat{a}, \hat{a}\}, r_2 \rangle) = \{ \langle \{\hat{a}, \hat{a}\}, r_2 \rangle \}, \quad cr(G_3, \langle \{\hat{a}, \hat{a}\}, r_2 \rangle) = r_2$$

$$\text{Conflict}(G_3, \langle a, r_3 \rangle) = \{ \langle a, r_3 \rangle, \langle a, r_4 \rangle \}, \quad cr(G_3, \langle a, r_3 \rangle) = r_3 + r_4$$

Llamando:

$$H_1 = \overline{\langle a, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel (\overline{\langle a, r_3 \rangle} \square \langle a, r_4 \rangle) \text{ sy } a,$$

$$H_{12} = \overline{\langle a, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel (\overline{\langle a, r_3 \rangle} \square \langle a, r_4 \rangle) \text{ sy } a,$$

$$H_{123} = \overline{\langle a, r_1 \rangle} \parallel \overline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel (\overline{\langle a, r_3 \rangle} \square \langle a, r_4 \rangle) \text{ sy } a$$

Aplicando ($Sy2'$) tendríamos:

$$\frac{G_3 \xrightarrow{\langle a, r_1 \rangle} H_1 \xrightarrow{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} H_{12}}{G_3 \xrightarrow{\langle \hat{a}, R_{12} \rangle} H_{12}}$$

con $R_{12} = \min\{r_1, r_2\}$ y $\text{Conflict}(G_3, \langle \hat{a}, R_{12} \rangle) = \{ \langle \hat{a}, R_{12} \rangle \}$.

Volviendo a aplicar ($Sy2'$) obtenemos

$$\frac{G_3 \xrightarrow{\langle \hat{a}, R_{12} \rangle} H_{12} \xrightarrow{\langle a, r_3 \rangle} H_{123}}{H \xrightarrow{\langle \emptyset, R_{12,3} \rangle} H_{123}}$$

con

$$R_{12,3} = \frac{R_{12}}{R_{12}} \cdot \frac{r_3}{r_3 + r_4} \cdot \min\{R_{12}, r_3 + r_4\} = \frac{r_3}{r_3 + r_4} \cdot \min\{r_1, r_2, r_3 + r_4\}$$

Es fácil comprobar que en los demás casos en los que se pueda aplicar dos veces la Regla $Sy2'$ a partir de una serialización diferente de γ , las razones coinciden con $R_{12,3}$, es decir:

$$R_{3,12} = R_{1,23} = R_{23,1} = R_{12,3} \quad \square$$

El (multi)sistema de transiciones asociado a una s-expresión dinámica que vimos en el capítulo anterior (definición 3.11) sigue siendo válido en esta propuesta, teniendo en cuenta la nueva regla para la sincronización. Además seguiremos aplicando el criterio expuesto en la Observación 3.1, a la hora de generar dicho sistema de transiciones. Así, en el ejemplo anterior sólo se computará un único arco de la forma:

$$(G_3, \langle \emptyset, R_{1,2,3} \rangle, H_{123})$$

El sistema de transiciones asociado a G_3 se muestra en la figura 4.4.

En el sistema de transiciones asociado a G_3 las s-expresiones dinámicas que nos faltaban por definir son:

$$\begin{aligned} H_2 &= \overline{\langle a, r_1 \rangle} \parallel \underline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel (\overline{\langle a, r_3 \rangle} \square \overline{\langle a, r_4 \rangle}) \text{ sy } a, \\ H_3 &= \overline{\langle a, r_1 \rangle} \parallel \underline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel (\underline{\langle a, r_3 \rangle} \square \underline{\langle a, r_4 \rangle}) \text{ sy } a, \\ H_{13} &= \underline{\langle a, r_1 \rangle} \parallel \underline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel (\underline{\langle a, r_3 \rangle} \square \underline{\langle a, r_4 \rangle}) \text{ sy } a, \\ H_{23} &= \overline{\langle a, r_1 \rangle} \parallel \underline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel (\underline{\langle a, r_3 \rangle} \square \underline{\langle a, r_4 \rangle}) \text{ sy } a, \\ H_{123} &= \underline{\langle a, r_1 \rangle} \parallel \underline{\langle \{\hat{a}, \hat{a}\}, r_2 \rangle} \parallel (\underline{\langle a, r_3 \rangle} \square \underline{\langle a, r_4 \rangle}) \text{ sy } a \end{aligned}$$

y las nuevas razones originadas al sincronizar:

$$\begin{aligned} R_{1,2} &= \min\{r_1, r_2\}, \\ R_{2,3} &= \frac{r_3}{r_3+r_4} \cdot \min\{r_2, r_3 + r_4\}, & R_{2,4} &= \frac{r_4}{r_3+r_4} \cdot \min\{r_2, r_3 + r_4\}, \\ R_{1,2,3} &= \frac{r_3}{r_3+r_4} \cdot \min\{r_1, r_2, r_3 + r_4\}, & R_{1,2,4} &= \frac{r_4}{r_3+r_4} \cdot \min\{r_1, r_2, r_3 + r_4\} \end{aligned}$$

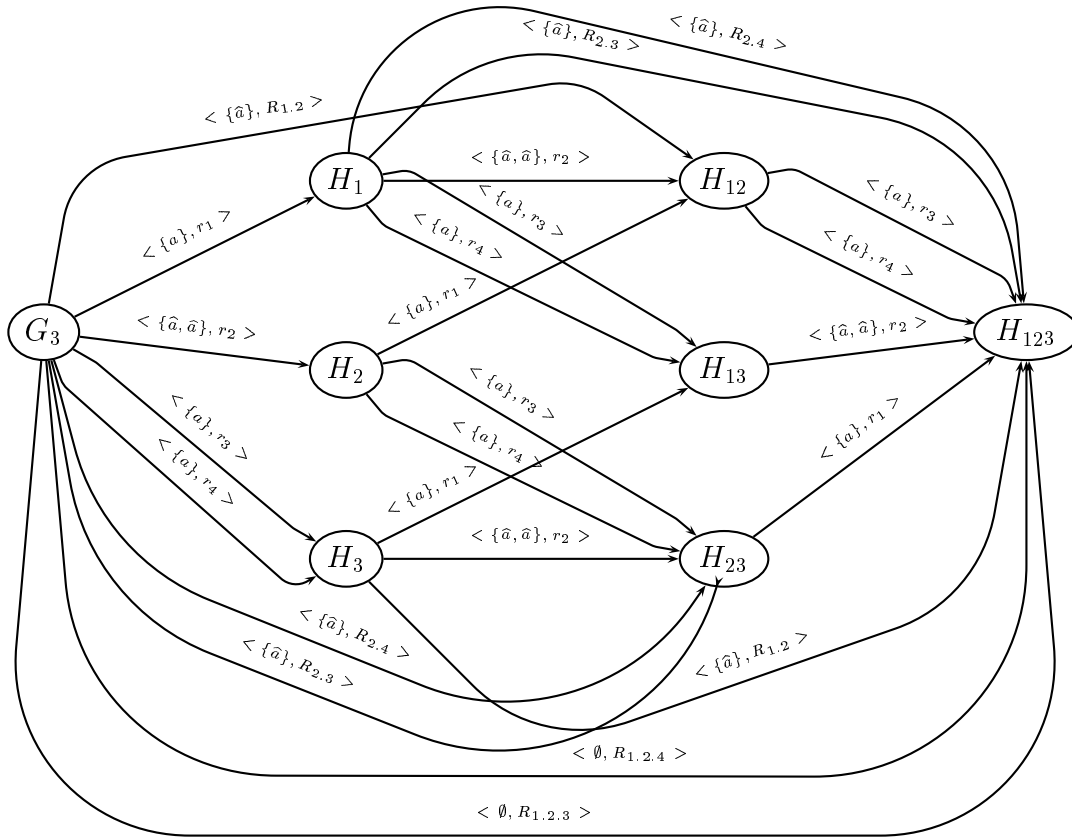


Figura 4.4: Sistema de transiciones asociado a G_3

Se sigue utilizando la política de competencia expuesta en el capítulo anterior, de modo que el proceso estocástico asociado a una s-expresión dinámica regular \overline{E} sigue siendo una cadena de Markov de tiempo continuo. La CTMC asociada a \overline{E} se obtiene a partir de $ts(\overline{E})$ de la misma forma que expusimos en el capítulo anterior: los nodos y los arcos se conservan, y si sólo hay un arco entre dos nodos, éste se etiqueta con la razón asociada a la multiacción estocástica que etiquetaba el arco en $ts(\overline{E})$, mientras que si hay dos o más arcos entre dos nodos en $ts(\overline{E})$, los unimos en uno único, etiquetado con la suma de las razones.

Ejemplo 4.6 Retomemos nuestro ejemplo introductorio (ejemplo 4.1). En él considerábamos:

$$E_1 = \langle \{a\}, 2 \rangle, \quad E_2 = \langle \{a\}, 1 \rangle \square \langle \{a\}, 1 \rangle$$

$$F_1 = \langle \{\hat{a}\}, 3 \rangle, \quad F_2 = \langle \{\hat{a}\}, 1 \rangle \square \langle \{\hat{a}\}, 2 \rangle$$

Es fácil comprobar que el sistema de transiciones de $\overline{(E_2 \parallel F_2)}$ *sy a* es el mostrado en la figura 4.3, ya que a partir de $\overline{(E_2 \parallel F_2)}$ *sy a* se pueden obtener las siguientes cuatro s-expresiones dinámicas operativas, en las que hemos enumerado las cuatro

multiacciones estocásticas de acuerdo a nuestro criterio habitual (de izquierda a derecha, según el orden de escritura):

$$J_1 = \overline{\langle \{a\}, 1 \rangle_1} \square \langle \{a\}, 1 \rangle_2 \parallel \overline{\langle \{\hat{a}\}, 1 \rangle_3} \square \langle \{\hat{a}\}, 2 \rangle_4$$

$$J_2 = \overline{\langle \{a\}, 1 \rangle_1} \square \langle \{a\}, 1 \rangle_2 \parallel \langle \{\hat{a}\}, 1 \rangle_3 \square \overline{\langle \{\hat{a}\}, 2 \rangle_4}$$

$$J_3 = \langle \{a\}, 1 \rangle_1 \square \overline{\langle \{a\}, 1 \rangle_2} \parallel \overline{\langle \{\hat{a}\}, 1 \rangle_3} \square \langle \{\hat{a}\}, 2 \rangle_4$$

$$J_4 = \langle \{a\}, 1 \rangle_1 \square \overline{\langle \{a\}, 1 \rangle_2} \parallel \langle \{\hat{a}\}, 1 \rangle_3 \square \overline{\langle \{\hat{a}\}, 2 \rangle_4}$$

con

$$\{\langle a, 1 \rangle_1, \langle \hat{a}, 1 \rangle_3\} \in BC(J_1)$$

$$\{\langle a, 1 \rangle_1, \langle \hat{a}, 2 \rangle_4\} \in BC(J_2)$$

$$\{\langle a, 1 \rangle_2, \langle \hat{a}, 1 \rangle_3\} \in BC(J_3)$$

$$\{\langle a, 1 \rangle_2, \langle \hat{a}, 2 \rangle_4\} \in BC(J_4)$$

$$Conflict(J_1, \langle a, 1 \rangle_1) = \{\langle a, 1 \rangle_1, \langle a, 1 \rangle_2\}, \quad cr(J_1, \langle a, 1 \rangle_1) = 2$$

$$Conflict(J_1, \langle \hat{a}, 1 \rangle_3) = \{\langle \hat{a}, 1 \rangle_3, \langle \hat{a}, 2 \rangle_4\}, \quad cr(J_1, \langle \hat{a}, 1 \rangle_3) = 3$$

$$Conflict(J_2, \langle a, 1 \rangle_1) = \{\langle a, 1 \rangle_1, \langle a, 1 \rangle_2\}, \quad cr(J_2, \langle a, 1 \rangle_1) = 2$$

$$Conflict(J_2, \langle \hat{a}, 2 \rangle_4) = \{\langle \hat{a}, 1 \rangle_3, \langle \hat{a}, 2 \rangle_4\}, \quad cr(J_2, \langle \hat{a}, 2 \rangle_4) = 3$$

$$Conflict(J_3, \langle a, 1 \rangle_2) = \{\langle a, 1 \rangle_1, \langle a, 1 \rangle_2\}, \quad cr(J_3, \langle a, 1 \rangle_2) = 2$$

$$Conflict(J_3, \langle \hat{a}, 1 \rangle_3) = \{\langle \hat{a}, 1 \rangle_3, \langle \hat{a}, 2 \rangle_4\}, \quad cr(J_3, \langle \hat{a}, 1 \rangle_3) = 3$$

$$Conflict(J_4, \langle a, 1 \rangle_2) = \{\langle a, 1 \rangle_1, \langle a, 1 \rangle_2\}, \quad cr(J_4, \langle a, 1 \rangle_2) = 2$$

$$Conflict(J_4, \langle \hat{a}, 1 \rangle_4) = \{\langle \hat{a}, 1 \rangle_3, \langle \hat{a}, 2 \rangle_4\}, \quad cr(J_4, \langle \hat{a}, 1 \rangle_4) = 3$$

Y así:

- A partir de J_1 se podría aplicar la Regla $Sy2'$ obteniendo la nueva multiacción estocástica: $\langle \emptyset, \frac{1}{3} \rangle_{1,3}$,
- A partir de J_2 se podría aplicar la Regla $Sy2'$ obteniendo la nueva multiacción estocástica: $\langle \emptyset, \frac{2}{3} \rangle_{1,4}$,
- A partir de J_3 se podría aplicar la Regla $Sy2'$ obteniendo la nueva multiacción estocástica: $\langle \emptyset, \frac{1}{3} \rangle_{2,3}$,
- A partir de J_4 se podría aplicar la Regla $Sy2'$ obteniendo la nueva multiacción estocástica: $\langle \emptyset, \frac{2}{3} \rangle_{2,4}$.

De esta forma se consiguen cuatro arcos de la forma:

$$(\overline{(E_2 \parallel F_2) sy a}, \langle \emptyset, r_i \rangle, \underline{(E_2 \parallel F_2) sy a})$$

cuya suma de razones es 2. El diagrama de transición de estados de la CTMC asociada a $\overline{(E_2 \parallel F_2)} sy a$ es el mostrado en la figura 4.5.

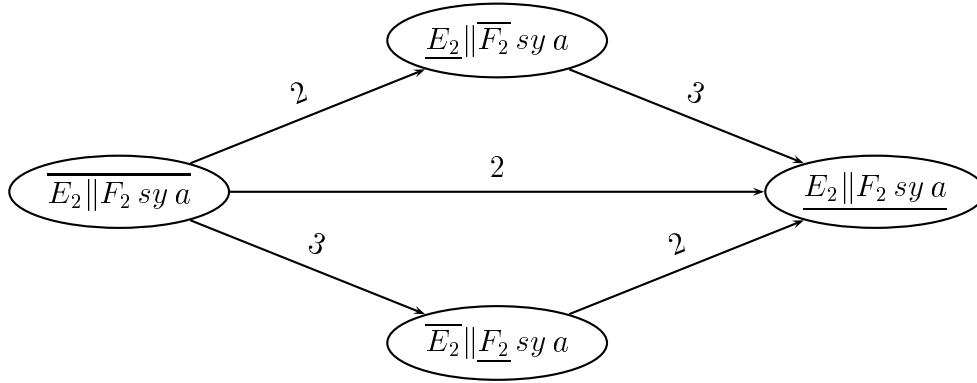


Figura 4.5: Diagrama de transición de estados de la CTMC asociada a $\overline{(E_2 \parallel F_2)} sy a$

□

Ejemplo 4.7 Volviendo a considerar el ejemplo tomado de PEPA que fue expuesto en el ejemplo 4.2, utilizando ahora la sintaxis de sPBC y cambiando α por a , tendríamos:

$$Sys = [a : (Comp_1 \parallel Comp_2 \parallel Comp_3)]$$

$$Comp_1 = \langle \beta, \lambda_1 \rangle; \langle \{a\}, \lambda_2 \rangle$$

$$Comp_2 = \langle \{a\}, \lambda_3 \rangle \square \langle \{a\}, \lambda_4 \rangle$$

$$Comp_3 = \langle \{\hat{a}\}, \lambda_5 \rangle; \langle \{\hat{a}\}, \lambda_6 \rangle$$

Podemos sincronizar $\langle \{a\}, \lambda_3 \rangle$ con $\langle \{\hat{a}\}, \lambda_5 \rangle$, obteniendo como razón asociada para la multiacción de sincronización:

$$R_{35} = \frac{\lambda_3}{\lambda_3 + \lambda_4} \cdot \frac{\lambda_5}{\lambda_5} \cdot \min\{\lambda_3 + \lambda_4, \lambda_5\}$$

Obsérvese que esta razón es independiente de que ya se haya ejecutado o no previamente la multiacción estocástica $\langle \beta, \lambda_1 \rangle$, ya que $\langle a, \lambda_3 \rangle$ y $\langle a, \lambda_2 \rangle$ no estarán nunca en conflicto. □

4.3 Semántica denotacional

Ahora tenemos que adaptar la semántica denotacional introducida en el capítulo anterior a la nueva propuesta de sincronización. Conservaremos los objetos semánticos (las s-cajas) que habíamos definido allí, pero ahora necesitamos más información

asociada. En concreto, para cada transición obtenida composicionalmente necesitamos conocer qué transiciones están en *conflicto* con ella, para así poder calcular su *razón de conflicto*. Seguimos la estrategia ya expuesta con anterioridad, basada en la enumeración de multiacciones: enumeramos las multiacciones estocásticas de izquierda a derecha, según aparezcan en la sintaxis de la s-expresión estática regular, y esta numeración se conservará en la correspondiente transición de la red de Petri que tenga asociada. Solamente aparecen nuevas transiciones en la red de Petri que no llevan asociada una multiacción estocástica en la sintaxis, en el caso de la sincronización. Estas nuevas transiciones se enumerarán con la concatenación de las enumeraciones de las transiciones (multiacciones estocásticas) argumento de la sincronización.

Para poder definir la función semántica que asocia una s-caja plana a cada término regular de sPBC finito, consideraremos las funciones siguientes:

$$\eta : T \longrightarrow \mathbb{N}^* \quad \text{y} \quad \kappa : T \longrightarrow \mathcal{P}(\mathbb{N}^*)$$

donde para cada transición $t \in T$, $\eta(t)$ nos dará la enumeración de t de acuerdo a nuestro criterio, y $\kappa(t)$ representará el conjunto de transiciones que están en conflicto con t . Esas funciones se van a definir de forma estructural, a la vez que construimos la correspondiente s-caja plana.

Además, para cada $t \in T$, necesitaremos definir también su correspondiente *razón de conflicto*, que denotaremos por $cr(t)$:

$$cr(t) = \sum_{\eta(t_j) \in \kappa(t)} \mu(t_j)$$

La estructura de la red se obtiene como en PBC, y de la misma forma que expusimos en el capítulo anterior, combinando refinamiento y reetiquetado. En consecuencia, las s-cajas así obtenidas seguirán siendo seguras y limpias, cuando partimos del marcado inicial M_e . Así, la semántica denotacional para una s-expresión regular estática se define formalmente mediante el siguiente homomorfismo:

$$\begin{aligned} \text{Box}_s(\langle \alpha, r \rangle_i) &= N_{\langle \alpha, r \rangle_i} \\ \text{Box}_s(\text{op}(E_1, \dots, E_n)) &= \Omega_{\text{op}}(\text{Box}_s(E_1), \dots, \text{Box}_s(E_n)) \end{aligned}$$

Pero debemos definir a la vez los valores de η y κ para cada transición obtenida de la red de Petri.

$$\bullet \text{Box}_s(\langle \alpha, r \rangle_i) = N_{\langle \alpha, r \rangle_i} = \begin{array}{c} \textcircled{e} \longrightarrow \boxed{\langle \alpha, r \rangle} \xrightarrow{t_i} \textcircled{x} \end{array}$$

donde $\eta(t_i) = i$ y $\kappa(t_i) = \{i\}$.

Para los restantes operadores de sPBC finito, las s-cajas operadores son las mismas ya descritas en el capítulo anterior (figura 3.11), y las funciones de reetiquetado $\rho_{op} \subseteq \mathcal{B}(\mathcal{SL}) \times \mathcal{SL}$ que aparecen en dicha figura siguen siendo las mismas, salvo en el caso del operador de sincronización, que posteriormente expondremos. Por lo tanto:

- $\rho_{id} = \{(\{\langle \alpha, r \rangle\}, \langle \alpha, r \rangle) \mid \langle \alpha, r \rangle \in \mathcal{SL}\}$
- $\rho_{[f]} = \{(\{\langle \alpha, r \rangle\}, \langle f(\alpha), r \rangle) \mid \langle \alpha, r \rangle \in \mathcal{SL}\}$
- $\rho_{rsa} = \{(\{\langle \alpha, r \rangle\}, \langle \alpha, r \rangle) \mid \langle \alpha, r \rangle \in \mathcal{SL} \wedge a, \hat{a} \notin A(\alpha)\}$

De esta forma, las correspondientes funciones semánticas se definen como se muestra a continuación, donde consideraremos que $\text{Box}_s(E_i) = (S_i, T_i, W_i, \lambda_i, \mu_i)$ es la s-caja plana correspondiente a E_i y η_i, κ_i son, respectivamente, las funciones de enumeración y conflicto para $T_i, i = 1, 2$.

$$\bullet \text{Box}_s(E_1; E_2) = \Omega_*(\text{Box}_s(E_1), \text{Box}_s(E_2))$$

Se puede observar que en esta definición no se introducen nuevas transiciones, ni nuevos conflictos; por lo tanto conservaremos la enumeración y conflictos que ya teníamos. Formalmente,

$$\eta(t) = \begin{cases} \eta_1(t) & \text{si } t \in T_1 \\ \eta_2(t) & \text{si } t \in T_2 \end{cases} \quad \kappa(t) = \begin{cases} \kappa_1(t) & \text{si } t \in T_1 \\ \kappa_2(t) & \text{si } t \in T_2 \end{cases}$$

$$\bullet \text{Box}_s(E_1 \parallel E_2) = \Omega_{\parallel}(\text{Box}_s(E_1), \text{Box}_s(E_2))$$

Estamos en el mismo caso que antes, no se introducen nuevas transiciones, ni nuevos conflictos, por lo que η y κ se definen exactamente igual que antes:

$$\eta(t) = \begin{cases} \eta_1(t) & \text{si } t \in T_1 \\ \eta_2(t) & \text{si } t \in T_2 \end{cases} \quad \kappa(t) = \begin{cases} \kappa_1(t) & \text{si } t \in T_1 \\ \kappa_2(t) & \text{si } t \in T_2 \end{cases}$$

$$\bullet \text{Box}_s(E_1[f]) = \Omega_{[f]}(\text{Box}_s(E_1))$$

Sólo cambiamos la etiqueta de algunas multiacciones, y como sólo consideramos funciones biyectivas, tenemos que se conservan los conflictos y la enumeración. Formalmente:

$$\eta(t) = \eta_1(t), \quad \forall t \in T_1 \quad \text{y} \quad \kappa(t) = \kappa_1(t), \quad \forall t \in T_1$$

- $Box_s(E_1 \square E_2) = \Omega_{\square}(Box_s(E_1), Box_s(E_2))$

Con este operador no se introducen nuevas transiciones, por lo que se conservará la enumeración de las transiciones. Sin embargo, pueden introducirse nuevos conflictos. En concreto, aquellas transiciones de T_1 con precondition en ${}^\circ Box_s(E_1)$ entrarán en conflicto con aquellas transiciones de T_2 con precondition en ${}^\circ Box_s(E_2)$. Como trabajamos con términos regulares, $Box_s(E_1)$ y $Box_s(E_2)$ tendrán un único lugar de entrada, y de esta forma $Box_s(E_1 \square E_2)$ tendrá también un único lugar de entrada. Formalmente:

$$\eta(t) = \begin{cases} \eta_1(t) & \text{si } t \in T_1 \\ \eta_2(t) & \text{si } t \in T_2 \end{cases}$$

$$\kappa(t) = \begin{cases} \kappa_1(t) \cup \kappa_2(t') & \text{si } t \in T_1, \bullet t \in {}^\circ Box_s(E_1), \exists t' \in T_2, \bullet t' \in {}^\circ Box_s(E_2), \lambda(t) = \lambda(t') \\ \kappa_1(t) & \text{si } t \in T_1, \bullet t \in {}^\circ Box_s(E_1), \nexists t' \in T_2, \bullet t' \in {}^\circ Box_s(E_2), \lambda(t) = \lambda(t') \\ \kappa_1(t) & \text{si } t \in T_1, \bullet t \notin {}^\circ Box_s(E_1) \\ \kappa_2(t) \cup \kappa_1(t') & \text{si } t \in T_2, \bullet t \in {}^\circ Box_s(E_2), \exists t' \in T_1, \bullet t' \in {}^\circ Box_s(E_1), \lambda(t) = \lambda(t') \\ \kappa_2(t) & \text{si } t \in T_2, \bullet t \in {}^\circ Box_s(E_2), \nexists t' \in T_1, \bullet t' \in {}^\circ Box_s(E_1), \lambda(t) = \lambda(t') \\ \kappa_2(t) & \text{si } t \in T_2, \bullet t \notin {}^\circ Box_s(E_2) \end{cases}$$

Se puede observar que $\kappa(t)$ está bien definida en el primer y cuarto caso, debido a que los valores de $\kappa_2(t')$ con $t' \in T_2, \bullet t' \in {}^\circ Box_s(E_2), \lambda(t) = \lambda(t')$, son todos iguales, lo que sucede de forma análoga en el otro caso.

- $Box_s(E_1 \text{ rs } a) = \Omega_{\text{rs } a}(Box_s(E_1))$

Con este operador eliminamos todas las transiciones etiquetadas con una multiacción que contenga a la acción a (ó \hat{a}). Esto no va a afectar a la enumeración de las transiciones no eliminadas, que siguen conservando la que ya tenían. Tampoco va a afectar a los conflictos de las transiciones no eliminadas, ya que recordemos que en $\kappa(t)$ todos los elementos tienen la misma multiacción asociada, así, si se elimina uno de ellos, se eliminan todos.

$$\eta(t) = \eta_1(t), t \in T_1, a, \hat{a} \notin \lambda(t) \quad \text{y} \quad \kappa(t) = \kappa_1(t), t \in T_1, a, \hat{a} \notin \lambda(t)$$

- $Box_s(E_1 \text{ sy } a) = \Omega_{\text{sy } a}(Box_s(E_1))$

Tomamos ahora la siguiente relación para la sincronización: $\rho_{\text{sy } a} \subseteq \mathcal{B}(\mathcal{L}) \times \mathcal{L}$ es la menor relación de reetiquetado que contiene a ρ_{id} y además cumple que:

$$\left((\Gamma, \alpha + \{a\}) \in \rho_{\text{sy } a} \quad \wedge \quad (\Delta, \beta + \{\hat{a}\}) \in \rho_{\text{sy } a} \right) \Rightarrow (\Gamma + \Delta, \alpha + \beta) \in \rho_{\text{sy } a}$$

De esta forma, $\rho_{\text{sy } a}$ permite obtener la estructura de la red (donde se han podido originar nuevas transiciones), así como las multiacciones con que se

etiquetan dichas transiciones. Si $t_1 \in \text{Box}_s(E_1)$ entonces $t_1 \in \text{Box}_s(E_1 \text{ sy } a)$, conservando su multiacción, razón, enumeración y conflictos. Esto es,

$$\lambda(t_1) = \lambda_1(t_1), \mu(t_1) = \mu_1(t_1), \eta(t_1) = \eta_1(t_1), \kappa(t_1) = \kappa_1(t_1)$$

Ahora, para cada $t_1, t_2 \in T_1$, con $\lambda_1(t_1) = \alpha + \{a\}$, $\lambda_1(t_2) = \beta + \{\hat{a}\}$, se origina por sincronización una nueva transición t , cuya multiacción asociada, $\lambda(t)$, es

$$\lambda(t) = \alpha + \beta$$

y su razón, $\mu(t)$, se calcula en la forma siguiente:

$$\mu(t) = \frac{\mu_1(t_1)}{cr(t_1)} \cdot \frac{\mu_1(t_2)}{cr(t_2)} \cdot \min\{cr(t_1), cr(t_2)\}$$

Además, para definir η y κ hemos de tener en cuenta nuestro criterio de enumeración por concatenación:

$$\begin{aligned} \eta(t) &= \eta_1(t_1) \cdot \eta_1(t_2) \\ \kappa(t) &= \kappa_1(t_1) \otimes \kappa_1(t_2) = \{n_1 \cdot n_2 \mid n_1 \in \kappa_1(t_1), n_2 \in \kappa_1(t_2)\} \end{aligned}$$

Para no introducir transiciones redundantes, que a nivel de prestaciones cambiarían la CTMC asociada, sólo consideraremos en la s-caja plana una de las posibles transiciones originadas por sincronización (con diferente orden) sobre el mismo conjunto de transiciones; es decir no repetiremos transiciones que tengan la misma enumeración, aunque en diferente orden.

Veamos ahora algunos ejemplos, con sus respectivas traducciones a s-cajas.

Ejemplo 4.8 Consideremos de nuevo el ejemplo 4.5:

$$E_3 = (\langle \{a\}, r_1 \rangle \parallel \langle \{\hat{a}, \hat{a}\}, r_2 \rangle \parallel (\langle \{a\}, r_3 \rangle \square \langle \{a\}, r_4 \rangle)) \text{ sy } a$$

La s-caja plana correspondiente es la mostrada en la figura 4.6, donde:

$$\begin{aligned} \eta(t_1) &= 1, & \kappa(t_1) &= \{1\}, & \eta(t_2) &= 2, & \kappa(t_2) &= \{2\}, \\ \eta(t_3) &= 3, & \kappa(t_3) &= \{3, 4\}, & \eta(t_4) &= 4, & \kappa(t_4) &= \{3, 4\}, \\ \eta(t_{1.2}) &= 1.2, & \kappa(t_{1.2}) &= \{1.2\}, & R_{12} &= \min\{r_1, r_2\} \\ \eta(t_{2.3}) &= 2.3, & \kappa(t_{2.3}) &= \{2.3, 2.4\}, & R_{23} &= \frac{r_3}{r_3+r_4} \cdot \min\{r_2, r_3 + r_4\} \\ \eta(t_{2.4}) &= 2.4, & \kappa(t_{2.4}) &= \{2.3, 2.4\}, & R_{24} &= \frac{r_4}{r_3+r_4} \cdot \min\{r_2, r_3 + r_4\} \\ \eta(t_{1.2.3}) &= 1.2.3, & \kappa(t_{1.2.3}) &= \{1.2.3, 1.2.4\}, & R_{123} &= \frac{r_3}{r_3+r_4} \cdot \min\{r_1, r_2, r_3 + r_4\} \\ \eta(t_{1.2.4}) &= 1.2.4, & \kappa(t_{1.2.4}) &= \{1.2.3, 1.2.4\}, & R_{124} &= \frac{r_4}{r_3+r_4} \cdot \min\{r_1, r_2, r_3 + r_4\} \end{aligned}$$

□

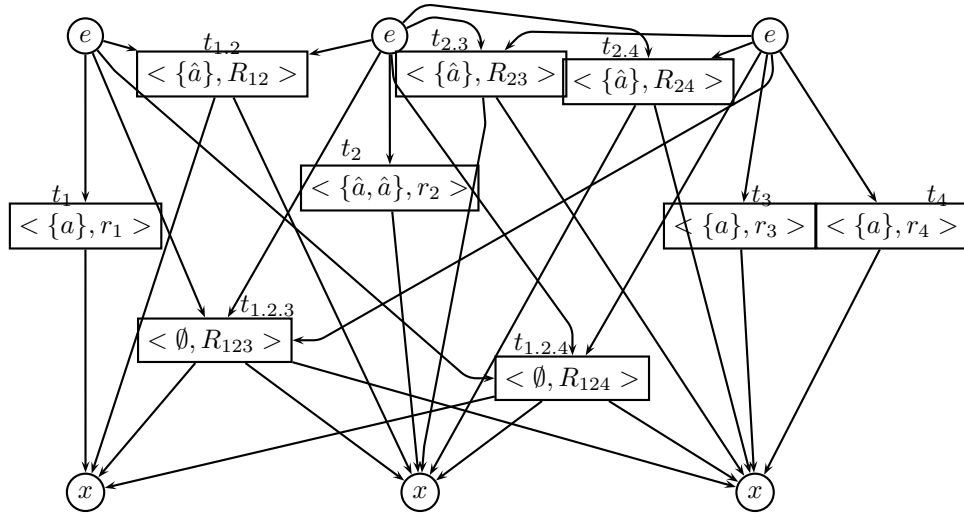


Figura 4.6: s-caja plana de E_3

Ejemplo 4.9 Si consideramos ahora la s-expresión estática regular:

$$E_4 = ((\langle \{a, \hat{a}\}, r_1 \rangle \square \langle \{a, \hat{a}\}, r_2 \rangle) \parallel (\langle \{a, \hat{a}\}, r_1 \rangle \square \langle \{b\}, r_3 \rangle)) \text{ sy } a$$

que correspondería a la s-expresión estática asociada a la s-expresión dinámica operativa G_1 del ejemplo 4.3, su s-caja plana asociada es la mostrada en la figura 4.7,

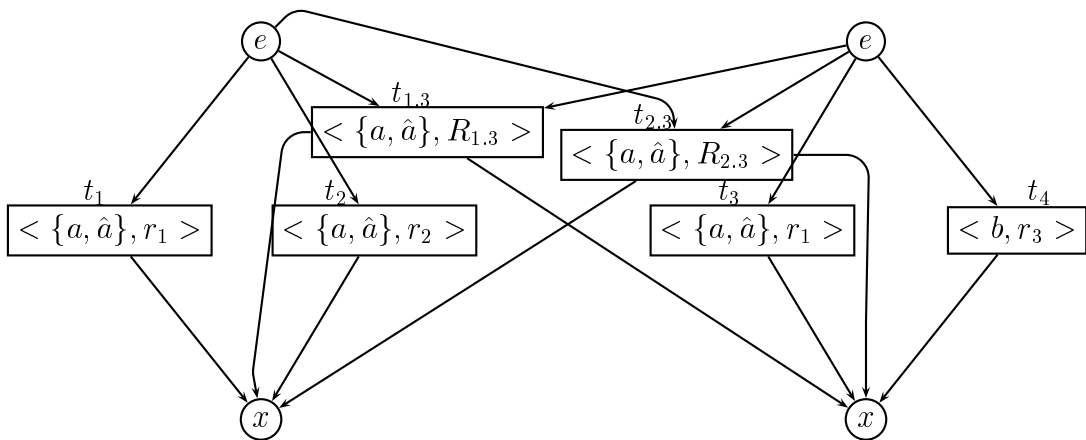


Figura 4.7: s-caja de E_4

donde:

$$\begin{aligned}
 \eta(t_1) &= 1, & \kappa(t_1) &= \{1, 2\}, \\
 \eta(t_2) &= 2, & \kappa(t_2) &= \{1, 2\}, \\
 \eta(t_3) &= 3, & \kappa(t_3) &= \{3\}, \\
 \eta(t_4) &= 4, & \kappa(t_4) &= \{4\}, \\
 \eta(t_{1.3}) &= 1.3, & \kappa(t_{1.3}) &= \{1.3, 2.3\}, \\
 R_{1.3} &= \frac{r_1}{r_1+r_2} \cdot \min\{r_1 + r_2, r_1\} = \frac{r_1^2}{r_1+r_2}, \\
 \eta(t_{2.3}) &= 2.3, & \kappa(t_{2.3}) &= \{1.3, 2.3\}, \\
 R_{2.3} &= \frac{r_2}{r_1+r_2} \cdot \min\{r_1 + r_2, r_1\} = \frac{r_1 r_2}{r_1+r_2}.
 \end{aligned}$$

□

Retomemos de nuevo el ejemplo tomado de PEPA (ejemplo 4.2),

Ejemplo 4.10

$$\begin{aligned}
 Sys &= [a : (Comp_1 \parallel Comp_2 \parallel Comp_3)] \\
 Comp_1 &= \langle \beta, \lambda_1 \rangle ; \langle \{a\}, \lambda_2 \rangle \\
 Comp_2 &= \langle \{a\}, \lambda_3 \rangle \square \langle \{a\}, \lambda_4 \rangle \\
 Comp_3 &= \langle \{\hat{a}\}, \lambda_5 \rangle ; \langle \{\hat{a}\}, \lambda_6 \rangle
 \end{aligned}$$

Su s-caja plana asociada es la mostrada en la figura 4.8,

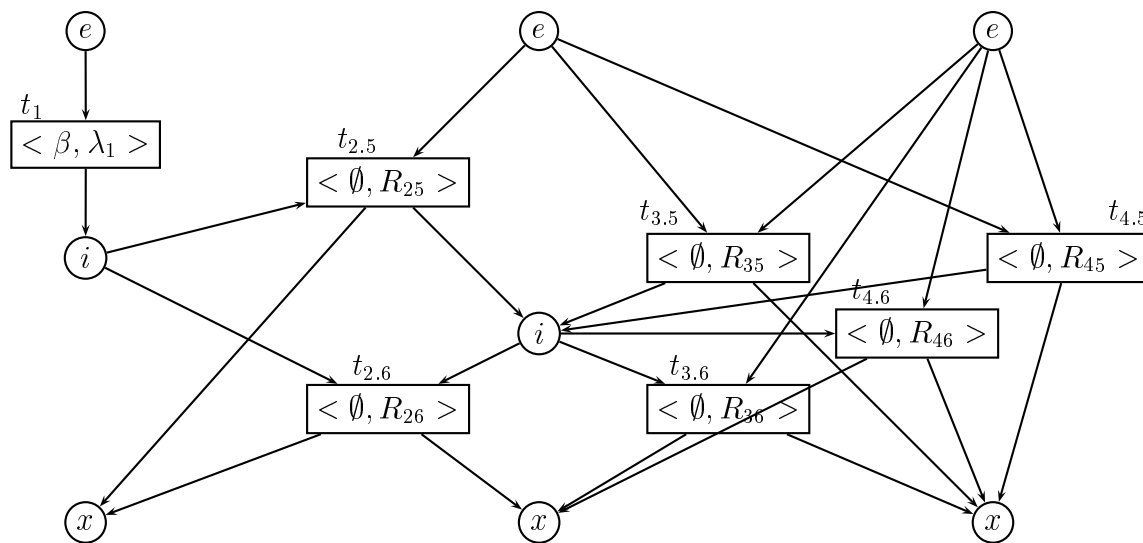


Figura 4.8: s-caja plana de *Sys*

donde

$$\begin{aligned}
\eta(t_1) &= 1, & \kappa(t_1) &= \{1\} \\
\eta(t_{2.5}) &= 2.5, & \kappa(t_{2.5}) &= \{2.5\} & R_{2.5} &= \min\{\lambda_2, \lambda_5\} \\
\eta(t_{2.6}) &= 2.6, & \kappa(t_{2.6}) &= \{2.6\} & R_{2.6} &= \min\{\lambda_2, \lambda_6\} \\
\eta(t_{3.5}) &= 3.5, & \kappa(t_{3.5}) &= \{3.5, 4.5\} & R_{3.5} &= \frac{\lambda_3}{\lambda_3 + \lambda_4} \min\{\lambda_3 + \lambda_4, \lambda_5\} \\
\eta(t_{3.6}) &= 3.6, & \kappa(t_{3.6}) &= \{3.6, 4.6\} & R_{3.6} &= \frac{\lambda_3}{\lambda_3 + \lambda_4} \min\{\lambda_3 + \lambda_4, \lambda_6\} \\
\eta(t_{4.5}) &= 4.5, & \kappa(t_{4.5}) &= \{3.5, 4.5\} & R_{4.5} &= \frac{\lambda_4}{\lambda_3 + \lambda_4} \min\{\lambda_3 + \lambda_4, \lambda_5\} \\
\eta(t_{4.6}) &= 4.6, & \kappa(t_{4.6}) &= \{3.6, 4.6\} & R_{4.6} &= \frac{\lambda_4}{\lambda_3 + \lambda_4} \min\{\lambda_3 + \lambda_4, \lambda_6\}
\end{aligned}$$

□

Finalmente, observemos que seguimos teniendo el mismo teorema enunciado en el capítulo anterior (teorema 3.1), que nos garantiza la equivalencia entre las dos semánticas.

Teorema 4.1 Para cada s-expresión estática E de sPBC finito, el sistema de transiciones $ts(\overline{E})$ y el grafo de accesibilidad de la red de Petri estocástica marcada $(Box_s(E), M_e)$ son isomorfos.

Demostración:

Es inmediato, dado que lo único que hemos variado es la razón de las nuevas transiciones originadas por sincronización, y dado que hemos tomado el mismo criterio en la semántica operacional y en la semántica de cajas, entonces se sigue manteniendo el isomorfismo entre el grafo de accesibilidad de la s-caja, considerando el marcado inicial y el mencionado sistema de transiciones. □

4.4 Algunas observaciones sobre la sincronización

En nuestra propuesta hemos considerado el multiconjunto $Conflict(G, \langle \alpha, r \rangle_i)$, en el cual G es operativa y $\langle \alpha, r \rangle_i$ es ejecutable en G , de modo que en dicho multiconjunto tendremos aquellas multiacciones estocásticas, con multiacción asociada α , envueltas en elección con $\langle \alpha, r \rangle_i$. Ciertamente es que, nuestra primera idea no fue evidentemente esa, sino considerar en conflicto con respecto a G y $\langle \alpha, r \rangle_i$ a toda multiacción estocástica que estuviera en elección con $\langle \alpha, r \rangle_i$, tuviera o no la misma multiacción asociada. Veamos los problemas que esa definición origina; para ello llamaremos $Conflict2$ a la función así definida.

Ejemplo 4.11 Consideremos la s-expresión estática siguiente:

$$E = ((\langle \{\hat{a}, a\}, r_1 \rangle \square \langle a, r_2 \rangle) \parallel \langle a, r_3 \rangle \parallel \langle \{\hat{a}, a\}, r_4 \rangle) \text{ sy } a$$

y la s-expresión dinámica operativa G , $G \equiv \overline{E}$, definida como sigue:

$$G = ((\overline{\langle \{\hat{a}, a\}, r_1 \rangle} \square \langle a, r_2 \rangle) \parallel \overline{\langle a, r_3 \rangle} \parallel \overline{\langle \{\hat{a}, a\}, r_4 \rangle}) \text{ sy } a$$

Entonces,

$$\text{Conflict2}(G, \langle \{\hat{a}, a\}, r_1 \rangle) = \{\langle \{\hat{a}, a\}, r_1 \rangle, \langle a, r_2 \rangle\}$$

$$\text{Conflict2}(G, \langle a, r_3 \rangle) = \{\langle a, r_3 \rangle\}$$

$$\text{Conflict2}(G, \langle \{\hat{a}, a\}, r_4 \rangle) = \{\langle \{a, \hat{a}\}, r_4 \rangle\}$$

Obteniéndose:

$$\langle \{\hat{a}, a\}, r_1 \rangle \oplus_a \langle a, r_3 \rangle = \langle a, R_{1.3} \rangle$$

siendo $R_{1.3} = \frac{r_1}{r_1+r_2} \min\{r_1 + r_2, r_3\}$.

Ahora tendríamos que definir el conflicto con respecto a una multiacción estocástica obtenida por sincronización. Nuestra idea fue definirla a partir del producto cartesiano de los conflictos de cada multiacción estocástica que interviene en la sincronización, siempre y cuando fuese posible la misma. Así, en nuestro ejemplo tendríamos:

$$\text{Conflict2}(G, \langle a, R_{1.3} \rangle) = \{\langle a, R_{1.3} \rangle\}$$

ya que no es posible sincronizar $\langle a, r_2 \rangle \in \text{Conflict2}(G, \langle \{\hat{a}, a\}, r_1 \rangle)$ con $\langle a, r_3 \rangle$.

Ahora podemos sincronizar:

$$\langle a, R_{1.3} \rangle \oplus_a \langle \{\hat{a}, a\}, r_4 \rangle = \langle a, R_{13.4} \rangle$$

donde $R_{13.4} = \min\{R_{1.3}, r_4\}$.

Sin embargo, si la sincronización de las tres multiacciones estocásticas se hubiese realizado en otro orden, no tendríamos garantizado llegar a la misma razón. Por ejemplo:

$$\langle a, r_3 \rangle \oplus_a \langle \{\hat{a}, a\}, r_4 \rangle = \langle a, R'_{3.4} \rangle$$

donde $R'_{3.4} = \min\{r_3, r_4\}$ y $\text{Conflict2}(G, \langle a, R'_{3.4} \rangle) = \{\langle a, R'_{3.4} \rangle\}$.

Si ahora consideramos:

$$\langle a, R'_{3.4} \rangle \oplus_a \langle \{\hat{a}, a\}, r_1 \rangle = \langle a, R'_{34.1} \rangle$$

tenemos $R'_{34.1} = \frac{r_1}{r_1+r_2} \min\{R'_{3.4}, r_1 + r_2\} = \frac{r_1}{r_1+r_2} \min\{r_1 + r_2, r_3, r_4\}$,

y desde luego $R_{13.4}$ y $R'_{34.1}$ no siempre son iguales (bastaría considerar $r_1 = 1, r_2 = 2, r_3 = 1, r_4 = 0.1$ para comprobar que pueden ser distintos). \square

La conclusión que extraemos del ejemplo anterior es que esta definición de *Conflict*, aparentemente más intuitiva, no sirve para obtener una definición consistente de la semántica de la sincronización, que sea independiente del orden en el que se realizan las sincronizaciones.

Otra opción que manejamos fue considerar el producto cartesiano de las multiacciones estocásticas en conflicto con las multiacciones estocásticas que intervienen en la sincronización, fuera o no posible la misma. En este caso los problemas vienen con el lema 4.2, el cual dejaría de cumplirse, y éste resulta necesario para probar que la relación de equivalencia allí definida (definición 6.7) es una congruencia (teorema 6.1).

A la vista de lo anterior, llegamos a la conclusión de que la propuesta que hemos presentado anteriormente era la más acertada, aunque sea menos intuitiva y no esté tampoco libre de *pegas*. Una de ellas es la restricción sintáctica que nos hemos visto obligados a imponer. Además, esta noción de conflicto no coincide exactamente con la noción clásica de conflicto en el marco de las redes de Petri. Veamos un ejemplo al respecto.

Ejemplo 4.12 Consideremos la s-expresión estática siguiente:

$$E = ((\langle \{a, \hat{a}\}, r_1 \rangle \square \langle \{a, a\}, r_2 \rangle) \parallel (\langle \{a, \hat{a}\}, r_3 \rangle \square \langle \{\hat{a}, \hat{a}\}, r_4 \rangle)) \text{ sy } a$$

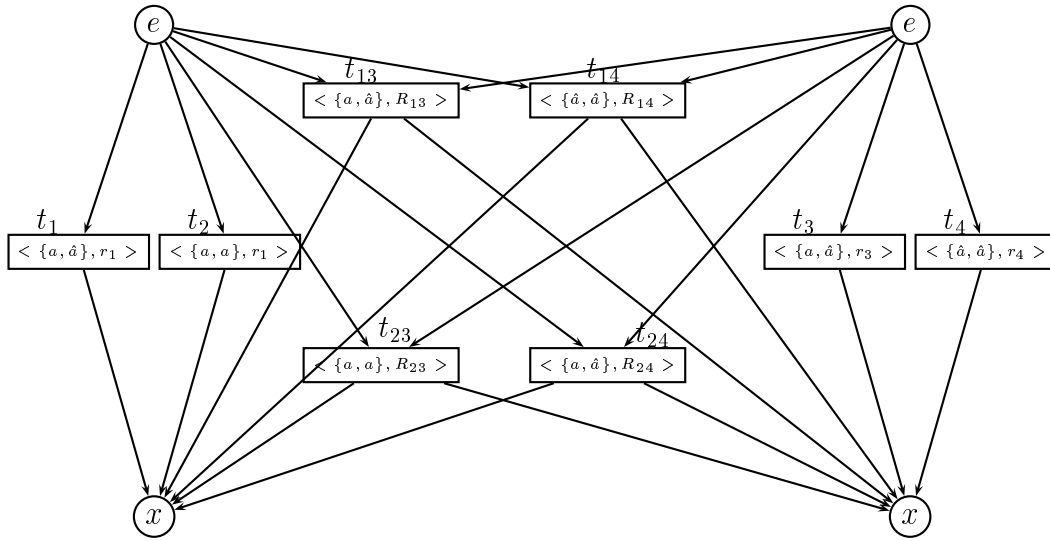
La s-caja asociada se muestra en la figura 4.9, donde se puede apreciar que:

$$\bullet t_{13} = \bullet t_{24}, \quad \lambda(t_{13}) = \lambda(t_{24}) = \{a, \hat{a}\}$$

Sin embargo, de acuerdo con nuestra definición: $\kappa(t_{13}) = \{1.3\} \neq \kappa(t_{24}) = \{2.4\}$.

\square

Afortunadamente, este *desbarajuste* sólo puede producirse cuando consideramos el conflicto de una multiacción estocástica (transición, en la red de Petri) obtenida por sincronización, y no se va a extender más allá, ya que aunque hagamos más elecciones, la que origina el problema debe ir precedida por una secuencia, debido

Figura 4.9: s-caja plana asociada a E

a la restricción sintáctica impuesta. No obstante, esta situación requiere tan sólo reconsiderar nuestra noción de conflicto en las redes de Petri, ajustándola a la definición realizada.

A continuación, pasamos brevemente a exponer qué hacen otras álgebras de procesos estocásticas con el operador de sincronización.

4.4.1 Comparación con otras SPAs

En la literatura podemos encontrar un interesante trabajo de Hillston [Hil94b], en el cual se hace una comparación entre las diferentes definiciones del operador de sincronización en distintas álgebras de procesos estocásticas (en el sentido de Markovianas). También en [HHK02] se pueden encontrar los principales desarrollos teóricos en el campo de las álgebras de procesos estocásticas, y en particular, con respecto a la semántica operacional del operador de sincronización. Basándonos en estos dos trabajos, y en los específicos de cada álgebra de procesos estocástica que vamos a considerar, pasamos a hacer nuestro estudio comparado. Nos centraremos en tres álgebras de procesos estocásticas que consideramos las más afines a nuestro modelo: TIPP [HR94], EMPA [BG98] y PEPA [Hil94a]. La estrategia a seguir es describir cómo considera cada una de ellas la sincronización y en cursiva iremos anotando nuestro enfoque en sPBC.

Las tres álgebras de procesos estocásticas se basan en los mismos operadores:

prefijo, elección, ocultamiento y composición paralela (donde se incluye la sincronización) basada en el operador de concurrencia de CSP [Hoa85].

En sPBC no se contempla el ocultamiento (sí la restricción). Hemos tomado los mismos operadores que en PBC plano, de esta forma nuestro operador de sincronización se basa en el mismo operador de CCS [Mil89], y además el operador paralelo (\parallel) está separado del operador sincronización (sy).

En las tres álgebras de procesos, el operador básico corresponde a una acción estocástica que se representa en la forma (a, r) , donde a es la acción y r es la razón de la exponencial negativa asociada a su retraso (*delay*).

En sPBC permitimos a un primer nivel multiacciones estocásticas, es decir, nuestro caso básico es de la forma $\langle \alpha, r \rangle$, donde α es un multiconjunto de acciones y r representa aquí también la razón de la exponencial negativa asociada a su retraso.

En todas las álgebras de procesos consideradas (*incluida sPBC*), a lo largo del proceso diferentes instancias de la misma acción (*multiacción*) pueden tener asociadas diferentes razones.

El comportamiento del operador de sincronización sobre un conjunto de acciones S , que representaremos aquí como \parallel_S , unificando las distintas notaciones, se define mediante reglas de la forma:

$$\begin{aligned}
 \text{(R1)} \quad & \frac{E \xrightarrow{(a,r)} E'}{E \parallel_S F \xrightarrow{(a,r)} E' \parallel_S F} \quad \text{si } a \notin S \\
 \text{(R2)} \quad & \frac{F \xrightarrow{(a,r)} F'}{E \parallel_S F \xrightarrow{(a,r)} E \parallel_S F'} \quad \text{si } a \notin S \\
 \text{(R3)} \quad & \frac{E \xrightarrow{(a,r_1)} E', F \xrightarrow{(a,r_2)} F'}{E \parallel_S F \xrightarrow{(a,R)} E' \parallel_S F} \quad \text{si } a \in S \quad (\text{Cond})
 \end{aligned}$$

En sPBC, como tenemos el operador \parallel separado del operador sy , las reglas (R1) y (R2) mostradas arriba se corresponderían con (C1) y (C2) de la tabla 3.3, mientras que (R3) se correspondería con $Sy2'$ de la tabla 4.1. Además, como en sPBC (ni en PBC) se obliga a sincronizar (a menos que lo imponamos con el operador $[a :]$) tenemos también la regla $Sy1$ de esa misma tabla 4.1.

Las definiciones de la sincronización varían según la forma de asignar la razón de sincronización R que aparece en (R3), y según la condición *Cond* que se impon-

ga. Consideraremos que en el cálculo de R intervienen las razones de las acciones envueltas en la sincronización, con lo que tendríamos

$$R = r_1 \odot r_2$$

para una cierta operación \odot , que se desea que cumpla las siguientes tres propiedades algebraicas:

- (P1)** $r_1 \odot r_2 = r_2 \odot r_1$, (de esta forma no importa el orden en que se consideren las acciones cuando se sincroniza),
- (P2)** $(r_1 \odot r_2) \odot r_3 = r_1 \odot (r_2 \odot r_3)$, (tampoco va a importar la forma de agrupar las sincronizaciones),
- (P3)** $(r_1 + r_2) \odot r_3 = (r_1 \odot r_3) + (r_2 \odot r_3)$, (se exige para que se tenga una congruencia: si se sustituye $(a, r_1 + r_2)$ por una elección entre (a, r_1) con (a, r_2) , no se tiene que ver afectada la razón total de la sincronización).

Ahora analizamos lo que se tiene en cada una de las álgebras de procesos estocásticas en estudio:

- (TIPP)** Considera $R = r_1 \cdot r_2$, sin imponer ninguna condición ($Cond = \emptyset$).
Por supuesto que con esa propuesta se cumplen las tres propiedades algebraicas de \odot pero, en nuestra opinión, no hay ninguna clara interpretación intuitiva y por eso la desestimamos para nuestro modelo (aunque era muy tentadora porque conllevaba poco esfuerzo). Por ejemplo:

$$(a, 2) \parallel_a (a, 2)$$

originaría una acción $(a, 4)$, mucho más rápida que las que intervienen en la sincronización, y sin embargo:

$$(a, 1/2) \parallel_a (a, 1/2)$$

originaría una acción $(a, 1/4)$, mucho más lenta que las que intervienen en la sincronización.

- (EMPA)** Considera $R = \max\{r_1, r_2\}$ e impone como $Cond$ que $\min\{r_1, r_2\} = 0$, es decir que al menos una de las acciones envuelta en la sincronización tenga razón 0, lo que se conoce en el argot como que es 'pasiva'.

Tampoco nos convenció el tener que utilizar acciones pasivas; no las utiliza PBC plano y tampoco tiene mucho sentido que las utilice su extensión markoviana. Además, en nuestra opinión, pensamos que el lenguaje es más flexible si no se obliga a manejar este tipo de acciones.

(PEPA) Considera $R = \frac{r_1}{r_a(E)} \cdot \frac{r_2}{r_a(F)} \min\{r_a(E), r_a(F)\}$, sin imponer ninguna condición para sincronizar ($Cond = \emptyset$).

Se basa en la idea de 'cooperación': ambos participantes en la cooperación deben completar algo de trabajo antes de que acabe la interacción. La idea es que la acción que se obtiene al sincronizar se 'espere a la más lenta', tomando el mínimo de las razones, pero si se adopta esta opción, no se cumple la tercera de las propiedades algebraicas exigidas a \odot . Por ello se coge el mínimo con respecto a la capacidad total que tiene cada participante de realizar la acción a , lo que se llama 'razón aparente' de a en cada componente, que se denota como $r_a(E)$ y $r_a(F)$, respectivamente. Sin embargo, no es suficiente con tomar ese mínimo, es necesario normalizarlo sobre el peso que tiene la instancia de la acción que interviene en la sincronización, con respecto a su razón aparente en cada componente $\frac{r_1}{r_a(E)}$ y $\frac{r_2}{r_a(F)}$, respectivamente.

Esta propuesta, aunque a primera vista sea demasiado sofisticada es la que más nos convenció, por su interpretación 'intuitiva' de esperarse a la más lenta. El inconveniente que presenta es que no siempre proporciona traducciones estáticas a redes de Petri estocásticas, pues en ocasiones las razones de las transiciones dependen del marcado de la red, tal y como vimos en el ejemplo 4.2, lo cual es algo no deseable, por lo que decidimos buscar la forma de evitarlo.

En consecuencia, en sPBC partimos de la idea de las razones aparentes de PEPA, pero considerando las razones de conflicto. De esta forma, se mantiene la interpretación intuitiva, y además se consiguen siempre traducciones estáticas a redes de Petri estocásticas, en el sentido de que las razones asociadas a las transiciones no dependen del marcado de la red. Uno de los principales objetivos de PBC, y por extensión también de sPBC, es obtener una traducción 'natural' a redes, así que entendíamos que esta traducción no debe tener razones asociadas a las transiciones que dependan del marcado. Además, en sPBC la forma de calcular R cuando se aplica varias veces la regla Sy2' es la descrita en la proposición 4.2, la cual es fácilmente implementable.

Existen también otras alternativas en el marco de las álgebras de procesos estocásticas, como MPA [Buc94] ó IMC [Her02], pero se alejan más de nuestro modelo. Brevemente pasamos a resumirlas.

MPA considera que cada acción tiene siempre la misma razón asociada y su caso base (a, r) representa que la acción a (con razón asociada μ_a) tiene r instancias concurrentes. La operación $r_1 \odot r_2$ la considera como $r_1 \cdot r_2$, que da lugar a una acción a , con razón asociada μ_a , que tiene $r_1 \cdot r_2$ instancias concurrentes activadas.

La otra alternativa, IMC, considera las duraciones estocásticas ortogonales a las acciones, así las duraciones λ pasan a ser un elemento más del álgebra de procesos. De esta forma los prefijos pueden ser de la forma $a; P$, donde a es una acción (al estilo de las álgebras de procesos clásicas), o también podemos tener prefijos de la forma $\lambda; P$, cuyo significado es que el proceso P sufre un retraso que viene determinado por una distribución exponencial de parámetro λ . En nuestra opinión, el mayor inconveniente que presenta este modelo es que no se puede sincronizar sobre acciones estocásticas.

Capítulo 5

Añadiendo la iteración

Hasta ahora todos los operadores definidos originaban procesos con un comportamiento finito, dando lugar a lo que hemos llamado sPBC finito. En este capítulo nos vamos a ocupar de añadir un operador más a nuestra álgebra de procesos: el operador de iteración. Está claro que con este nuevo operador no se logran todos los comportamientos infinitos, pero al combinarlo con el operador de restricción, sí que abrimos una puerta para generar una colección más amplia de procesos, como de hecho pondremos de manifiesto en el capítulo 7 con los casos de estudio.

El operador de recursión necesitaría en general un tratamiento especial, tal y como ocurre en PBC plano, ya que puede dar lugar a redes infinitas. Además, y aunque en esta tesis no hemos entrado a fondo en cuestiones relativas a la evaluación de prestaciones, con estas redes infinitas habría problemas a la hora de obtener resultados sobre prestaciones. Por todo ello, y dado que hemos considerado que el trabajo ya realizado era suficiente para completar un proyecto de tesis doctoral, hemos dejado la definición y estudio del operador de recursión como trabajo futuro.

Así pues, el plan para este capítulo es introducir primero la sintaxis del operador de iteración, pasando luego a estudiar su semántica operacional, y finalmente en el último apartado su semántica denotacional, así como la equivalencia entre las dos semánticas para nuestro modelo sPBC.

5.1 Sintaxis

Seguiremos utilizando la sintaxis introducida en el capítulo 3, aunque necesitamos introducir una restricción adicional a las funciones de renombramiento, tal y como hicimos en el capítulo 4; concretamente exigiremos que además de conservar los conjugados también han de ser biyectivas.

Añadimos por tanto el operador de iteración $[**]$, con lo cual nuestro lenguaje quedaría definitivamente definido por la siguiente expresión BNF:

$$E ::= \langle \alpha, r \rangle \mid E; E \mid E \square E \mid E \parallel E \mid E[f] \mid E \text{ sy } a \mid E \text{ rs } a \mid [a : E] \mid [E * E * E]$$

donde todos los operadores, salvo el último, han sido ya estudiados en los capítulos anteriores. Recordemos que

$$[E_1 * E_2 * E_3]$$

representa un proceso que realiza primero E_1 , después pasa a realizar un número indefinido de veces (puede que ninguna) E_2 , para terminar realizando E_3 . Si combinamos de forma adecuada este operador de iteración con el operador de restricción, es posible obtener comportamientos infinitos. Por ejemplo:

$$[\langle \{a\}, r_1 \rangle * \langle \{b\}, r_2 \rangle * \langle \{c\}, r_3 \rangle] \text{ rs } c$$

representa el proceso que puede ejecutar una vez $\langle \{a\}, r_1 \rangle$, seguido de infinitas ejecuciones de $\langle \{b\}, r_2 \rangle$. Obsérvese que este proceso no termina nunca, al no poder alcanzar la tercera componente.

Para asegurarnos de que el operador de sincronización funciona de forma adecuada necesitaremos restringir de nuevo la sintaxis a aquellos términos en los cuales un comportamiento paralelo no aparece en el nivel más alto de una elección (ver capítulo 4); pero ahora debemos eliminar además de la sintaxis aquellos términos en los que aparezca un operador paralelo en el más alto nivel en los dos últimos argumentos de la iteración. Los términos que cumplan esta restricción se llamarán de nuevo *términos regulares*, y la semántica operacional y denotacional se definirá sobre ellos. En el ejemplo 5.1 ilustraremos lo que puede ocurrir si no respetamos esta restricción.

Formalmente, una s-expresión estática regular E de sPBC será toda aquella generable mediante el siguiente par de expresiones BNF:

$$\begin{aligned} D & ::= \langle \alpha, r \rangle \mid D; E \mid D \text{ sy } a \mid D \text{ rs } a \mid D[f] \mid [a : D] \mid D \square D \mid [D * D * D] \\ E & ::= \langle \alpha, r \rangle \mid E; E \mid E \text{ sy } a \mid E \text{ rs } a \mid E[f] \mid [a : E] \mid E \parallel E \mid D \square D \mid [E * D * D] \end{aligned}$$

donde llamaremos a los D -términos *deterministas*.

Como hasta ahora, la semántica operacional de sPBC se apoya en las s-expresiones dinámicas, las cuales se obtienen de las s-expresiones estáticas anotando en ellas los barrados y sub-barrados, para indicar las componentes activas. Así, una s-expresión dinámica G de sPBC será de la forma:

$$G ::= \overline{E} | \underline{E} | G; E | E; G | G \square E | E \square G | G || G | G[f] | G \text{ sy } a | G \text{ rs } a | \\ [a : G] | [G * E * E] | [E * G * E] | [E * E * G]$$

donde \overline{E} denota el estado inicial de E , y \underline{E} su estado final. Diremos que una s-expresión dinámica es regular si su correspondiente s-expresión estática lo es. Manteniendo la notación anterior, al conjunto de las s-expresiones dinámicas regulares lo denotaremos por *ReDynExpr*.

5.2 Semántica Operacional

Para definir la semántica operacional de sPBC con iteración tan solo tenemos que introducir nuevas reglas de inacción (ver tabla 5.1) y nuevas reglas de transición estocástica (ver tabla 5.2). Estas reglas de inacción ponen de manifiesto lo siguiente: para iniciar una iteración es necesario activar primero el inicio de la misma; si el inicio de la iteración ya ha terminado, entonces se activa el cuerpo; si el cuerpo ya ha finalizado entonces, o bien se puede volver a activar, o se activa la tercera de las componentes; una iteración se da por finalizada si la tercera de las componentes ya lo ha hecho; y cualquier regla de inacción aplicada a cualquiera de las componentes se transmite con el operador de iteración.

$[\overline{E} * F * E'] \xrightarrow{\emptyset} [\overline{E} * F * E']$	$[\underline{E} * F * E'] \xrightarrow{\emptyset} [E * \overline{F} * E']$
$[E * \underline{F} * E'] \xrightarrow{\emptyset} [E * \overline{F} * E']$	$[E * F * E'] \xrightarrow{\emptyset} [E * F * \overline{E}']$
$[E * F * \underline{E}'] \xrightarrow{\emptyset} [\underline{E} * F * E']$	$\frac{G \xrightarrow{\emptyset} G'}{[G * E * F] \xrightarrow{\emptyset} [G' * E * F]}$
$\frac{G \xrightarrow{\emptyset} G'}{[E * G * F] \xrightarrow{\emptyset} [E * G' * F]}$	$\frac{G \xrightarrow{\emptyset} G'}{[E * F * G] \xrightarrow{\emptyset} [E * F * G']}$

Tabla 5.1: Reglas de inacción para el operador de iteración

Las definiciones realizadas en los dos capítulos anteriores se mantienen, así, seguiremos diciendo que una s-expresión dinámica regular G es operativa si no pueden aplicarse reglas de inacción a partir de ella. Al conjunto de las s-expresiones dinámicas regulares operativas lo seguiremos denotando por $OpReDynExpr$. También se mantiene la relación de equivalencia estructural definida sobre las s-expresiones dinámicas regulares:

$$\equiv =_{def} (\xrightarrow{\emptyset} \cup \xleftarrow{\emptyset})^*$$

En la tabla 5.2 se muestran las reglas de transición estocástica para el operador de iteración, donde suponemos que todas las s-expresiones dinámicas que aparecen en la parte izquierda de dichas reglas son operativas y regulares. Dichas reglas reflejan que la iteración conserva los movimientos que cada componente puede realizar.

(It1)	$\frac{G \xrightarrow{\langle \alpha, r \rangle} G'}{[G * E * F] \xrightarrow{\langle \alpha, r \rangle} [G' * E * F]}$
(It2)	$\frac{G \xrightarrow{\langle \alpha, r \rangle} G'}{[E * G * F] \xrightarrow{\langle \alpha, r \rangle} [E * G' * F]}$
(It3)	$\frac{G \xrightarrow{\langle \alpha, r \rangle} G'}{[E * F * G] \xrightarrow{\langle \alpha, r \rangle} [E * F * G']}$

Tabla 5.2: Reglas de transición estocástica del operador $[**]$

Obsérvese que hemos de extender la definición de bolsa concurrente para tener en cuenta este nuevo operador y de esa forma, las reglas de la tabla 4.1 podrán ser aplicadas sobre el nuevo lenguaje.

Definición 5.1 Definimos $BC : OpReDynExpr \rightarrow \mathcal{P}(\mathcal{B}(SL))$, en la forma:

- Si $G \in OpReDynExpr$ es final, esto es, $G = \underline{E}$, tomamos $BC(G) = \emptyset$.
- Si $G \in OpReDynExpr$ no es final, distinguimos los casos siguientes:
 - $BC(\overline{\langle \alpha, r \rangle}) = \{\{\langle \alpha, r \rangle\}\}$
 - Si $\gamma \in BC(G)$, entonces: $\gamma \in BC(G; E)$, $\gamma \in BC(E; G)$, $\gamma \in BC(E \square G)$, $\gamma \in BC(G \square E)$, $\gamma \in BC(G \text{ rs } a)$ (cuando $a, \hat{a} \notin A(\gamma)$), $\gamma \in BC(G \text{ sy } a)$, $f(\gamma) \in BC(G[f])$, $\gamma \in BC([G * E * F])$, $\gamma \in BC([E * G * F])$, $\gamma \in BC([E * F * G])$.

- Si $\gamma_1 \in BC(G)$, entonces $\gamma_1 \in BC(G\|H)$.
- Si $\gamma_2 \in BC(H)$, entonces $\gamma_2 \in BC(G\|H)$.
- Si $\gamma_1 \in BC(G)$, $\gamma_2 \in BC(H)$, entonces $\gamma_1 + \gamma_2 \in BC(G\|H)$.
- Si $\gamma \in BC(G \text{ sy } a)$, y $\langle \alpha, r_1 \rangle, \langle \beta, r_2 \rangle \in \gamma$, (con $\langle \alpha, r_1 \rangle \neq \langle \beta, r_2 \rangle$ ó bien son instancias diferentes de la misma multiacción estocástica en γ) con $a \in A(\alpha)$, y $\hat{a} \in A(\beta)$, entonces $\gamma' \in BC(G \text{ sy } a)$, donde:

$$\gamma' = (\gamma + \{\langle \alpha \oplus_a \beta, R \rangle\}) \setminus \{\langle \alpha, r_1 \rangle, \langle \beta, r_2 \rangle\}$$

y R es la razón de la nueva multiacción estocástica, definida en la Regla $Sy2'$ de la tabla 4.1. \square

Naturalmente, la propiedad presentada en la proposición 4.1 se mantiene, extendida al considerar el cuerpo y la terminación de la iteración en la definición de los términos regulares.

Proposición 5.1 Sea $G \in OpReDynExpr$,

- (i) Si existen dos s-expresiones estáticas regulares E_1, E_2 , con

$$G \equiv \overline{E_1 \square E_2}$$

Entonces, no es posible que $\{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle\} \in BC(G)$.

- (ii) Si existen tres s-expresiones estáticas regulares E, E_1, E_2 , con

$$G \equiv [E * \overline{E_1} * E_2]$$

Entonces, no es posible que $\{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle\} \in BC(G)$.

Demostración:

- (i) Vale la misma demostración que para la proposición 4.1, pero faltaría considerar el caso en el que D_1 es de la forma $[D * D' * D'']$, el cual se resolvería fácilmente, aplicando la hipótesis de inducción sobre D , con la nueva definición de bolsa concurrente.

(ii) Ya que:

$$[E * \overline{E_1} * E_2] \equiv [E * E_1 * \overline{E_2}]$$

sin pérdida de generalidad, tomaremos el primero de los casos.

Como G es regular, tenemos que $G \equiv [E * \overline{D_1} * D_2]$ y bastaría con aplicar inducción estructural sobre D_1 .

- El caso base es $D_1 = \langle \alpha, r \rangle$, y entonces es trivial que se cumple la propiedad, ya que tenemos $G = [E * \overline{D_1} * D_2]$ y $BC(G) = \{\langle \alpha, r \rangle\}$.
- Para D_1 de la forma $D; E, D \text{ sy } a, D \text{ rs } a, D[f], [a : D]$ y $[D * D_{i_1} * D_{i_2}]$ sólo necesitamos aplicar la hipótesis de inducción sobre D , así como la definición de bolsa concurrente.
- Para D_1 de la forma $D_{i_1} \square D_{i_2}$ podemos utilizar la hipótesis de inducción y el caso (i), para concluir la propiedad. \square

Naturalmente, también es necesario extender la definición de la función *Conflict*, para poder calcular adecuadamente la razón de la multiacción generada por efecto de una sincronización. Es importante observar que en una iteración puede considerarse como conflicto la situación que tenemos al inicio de cada pasada de la segunda componente, pues o bien puede realizarse la misma, o bien puede saltarse a la tercera componente.

Definición 5.2 Definimos la siguiente función parcial:

$$\text{Conflict} : \text{OpReDynExpr} \times \mathcal{SL} \longrightarrow \mathcal{B}(\mathcal{SL})$$

la cual para cada instancia i de la multiacción estocástica $\langle \alpha, r \rangle$ ejecutable desde G nos proporciona el multiconjunto de multiacciones estocásticas $\langle \alpha, r' \rangle$ en *conflicto* con ella. Definimos la función por inducción estructural¹:

1. $\text{Conflict}(\overline{\langle \alpha, r \rangle}, \langle \alpha, r \rangle) = \{\langle \alpha, r \rangle\}$.
2. Si $\langle \alpha, r \rangle$ es ejecutable en G , y $C = \text{Conflict}(G, \langle \alpha, r \rangle)$, entonces:
 - (a) $\text{Conflict}(G; E, \langle \alpha, r \rangle) = \text{Conflict}(E; G, \langle \alpha, r \rangle) = C$,
 - (b) $\text{Conflict}(G \parallel H, \langle \alpha, r \rangle) = \text{Conflict}(H \parallel G, \langle \alpha, r \rangle) = C$,

¹donde omitimos el subíndice en $\langle \alpha, r \rangle_i$ para no sobrecargar la notación.

- (c) Si $a, \hat{a} \notin A(\alpha)$, entonces $Conflict(G \text{ rs } a, \langle \alpha, r \rangle) = C$,
- (d) Para cada función biyectiva f , $Conflict(G[f], \langle f(\alpha), r \rangle) = f(C)$,
- (e) Para el operador de elección necesitamos distinguir los dos casos siguientes:
- Si $G \not\equiv \bar{E}$:
 $Conflict(G \square F, \langle \alpha, r \rangle) = Conflict(F \square G, \langle \alpha, r \rangle) = C$
 - Si $G \equiv \bar{E}$:
 $Conflict(G \square F, \langle \alpha, r \rangle) = Conflict(F \square G, \langle \alpha, r \rangle) = C +$
 $\{ \langle \alpha, r_j \rangle \mid \exists H_i \in OpReDynExpr, H_i \equiv \bar{F} \text{ y } H_i \xrightarrow{\langle \alpha, r_j \rangle} H'_i \}$
- (f) Para el operador de iteración tenemos:
- $Conflict([G * E * F], \langle \alpha, r \rangle) = C$
 - Para los dos últimos argumentos de una iteración tenemos:
 - Si $G \not\equiv \bar{E}'$:
 $Conflict([E * G * F], \langle \alpha, r \rangle) = Conflict([E * F * G], \langle \alpha, r \rangle) = C$
 - Si $G \equiv \bar{E}'$:
 $Conflict([E * G * F], \langle \alpha, r \rangle) = Conflict([E * F * G], \langle \alpha, r \rangle) =$
 $C + \{ \langle \alpha, r_j \rangle \mid \exists H_i \in OpReDynExpr, H_i \equiv \bar{F} \text{ y } H_i \xrightarrow{\langle \alpha, r_j \rangle} H'_i \}$
- (g) $Conflict(G \text{ sy } a, \langle \alpha, r \rangle) = C$.

3. Sea $\{ \langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle \} \in BC(G \text{ sy } a)$, $a \in A(\alpha_1)$, $\hat{a} \in A(\alpha_2)$ y $G \text{ sy } a \xrightarrow{\langle \alpha_1 \oplus_a \alpha_2, R_{12} \rangle} G' \text{ sy } a$ obtenida por aplicación de la Regla $Sy2'$.

Entonces:

$$Conflict(G \text{ sy } a, \langle \alpha_1 \oplus_a \alpha_2, R_{12} \rangle) =$$

$$\{ \langle \alpha_1 \oplus_a \alpha_2, R_{ij} \rangle \mid \langle \alpha_1, r_i \rangle \in C_1, \langle \alpha_2, r_j \rangle \in C_2, \text{ donde}$$

$$R_{ij} = \frac{r_i}{cr(G \text{ sy } a, \langle \alpha_1, r_1 \rangle)} \frac{r_j}{cr(G \text{ sy } a, \langle \alpha_2, r_2 \rangle)} \cdot \min_{k=1,2} \{ cr(G \text{ sy } a, \langle \alpha_k, r_k \rangle) \} \}$$

con $C_k = Conflict(G \text{ sy } a, \langle \alpha_k, r_k \rangle)$, $k = 1, 2$, y $cr(G, \langle \alpha, r \rangle_k)$, que llamaremos *razón de conflicto* para G y $\langle \alpha, r \rangle_k$, definida como:

$$cr(G, \langle \alpha, r \rangle_k) = \sum_{\langle \alpha, r_j \rangle \in Conflict(G, \langle \alpha, r \rangle_k)} r_j \cdot n_j$$

donde n_j es el número de veces que aparece $\langle \alpha, r_j \rangle$ en $Conflict(G, \langle \alpha, r \rangle_k)$. □

La definición de conjunto de alcance sigue siendo la misma que la realizada en definición 4.3 y seguimos teniendo idénticos resultados que los presentados en el

apartado 4.2 del capítulo anterior. Estos se exponen a continuación, haciendo las consideraciones oportunas en las demostraciones para adaptarlas al incorporar la iteración.

Lema 5.1 Sea $G \in OpReDynExpr$ y $\gamma = \{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle\} \in BC(G)$, con:

$$G \xrightarrow{\langle \alpha_1, r_1 \rangle} H \xrightarrow{\emptyset} H^* \xrightarrow{\langle \alpha_2, r_2 \rangle} J.$$

Entonces $Conflict(G, \langle \alpha_2, r_2 \rangle) = Conflict(H^*, \langle \alpha_2, r_2 \rangle)$.

Demostración:

La demostración es la realizada en el lema 4.1 a la que hay que añadir el nuevo operador de iteración. Se hará por inducción estructural sobre la sintaxis de G .

- **Caso base.** $G = G_1 \parallel G_2$, donde $G_1, G_2 \in OpReDynExpr$, $\langle \alpha_1, r_1 \rangle$ es ejecutable en G_1 y $\langle \alpha_2, r_2 \rangle$ es ejecutable en G_2 , ya que partimos de $\gamma = \{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle\} \in BC(G)$. Entonces:

$$G = G_1 \parallel G_2 \xrightarrow{\langle \alpha_1, r_1 \rangle} G'_1 \parallel G_2 \xrightarrow{\emptyset} G'^*_1 \parallel G_2 \xrightarrow{\langle \alpha_2, r_2 \rangle} G'^*_1 \parallel G'_2$$

Y tomando $H = G'_1 \parallel G_2$, $H^* = G'^*_1 \parallel G_2$ y $J = G'^*_1 \parallel G'_2$, llegamos a:

$$\begin{aligned} Conflict(G_1 \parallel G_2, \langle \alpha_2, r_2 \rangle) &= Conflict(G_2, \langle \alpha_2, r_2 \rangle) = \\ &Conflict(G'^*_1 \parallel G_2, \langle \alpha_2, r_2 \rangle) = Conflict(H^*, \langle \alpha_2, r_2 \rangle) \end{aligned}$$

con lo cual habríamos resuelto el caso base al llegar a:

$$Conflict(G, \langle \alpha_2, r_2 \rangle) = Conflict(H^*, \langle \alpha_2, r_2 \rangle)$$

- **Caso general:**

- Una simple aplicación de la hipótesis de inducción sobre G' resuelve los casos de la composición secuencial ($G = G'; E$), la restricción ($G = G' rs a$) y el renombramiento ($G = G'[f]$).
- El mismo razonamiento seguido en la demostración del lema 4.1 para el operador de elección se seguiría aquí, y resolvería $G = G' \square F$ y $G = F \square G'$.
- Para el operador de iteración el razonamiento es similar al expuesto para el caso de la elección. Nos podemos encontrar en los casos:

- $G = [G' * E_1 * E_2]$, donde bastaría con aplicar la hipótesis de inducción sobre G' y se concluiría la propiedad deseada.
 - $G = [E_1 * G' * E_2]$ ó $G = [E_1 * E_2 * G']$, donde en ambos casos se tiene que $G' \not\equiv \bar{F}$ para todo F , debido a que tenemos un comportamiento paralelo en G , y por la propiedad semántica deducida de la proposición 5.1, bastaría con aplicar la hipótesis de inducción sobre G' , llegando al resultado deseado.
- Para el caso del operador de sincronización $G = G' \text{ sy } a$, se repetiría el razonamiento expuesto en el lema 4.1. □

En la proposición siguiente veremos que la razón de la multiacción estocástica que obtenemos cuando aplicamos varias veces la regla $Sy2'$ se obtiene según la fórmula vista en la proposición 4.2.

Proposición 5.2 Sea $G \in OpReDynExpr$, $\gamma = \{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle, \dots, \langle \alpha_n, r_n \rangle\}$, $\gamma \in BC(G)$, y una serialización de γ con:

$$G \xrightarrow{\langle \alpha_1, r_1 \rangle} G_1 \xrightarrow{(\emptyset)} G_1^* \xrightarrow{\langle \alpha_2, r_2 \rangle} \dots \xrightarrow{\langle \alpha_n, r_n \rangle} G_n$$

sobre la cual podemos aplicar $n - 1$ veces la Regla $Sy2'$, obteniendo una única transición estocástica de la forma $G \xrightarrow{\langle \beta, R \rangle} G_n$.

Entonces tenemos que:

$$R = \left(\prod_{k=1}^n \frac{r_k}{cr(G, \langle \alpha_k, r_k \rangle)} \right) \cdot cr(G, \langle \beta, R \rangle)$$

$$cr(G, \langle \beta, R \rangle) = \min_{k=1, \dots, n} \{cr(G, \langle \alpha_k, r_k \rangle)\}$$

Demostración:

Se procede por inducción sobre n siguiendo el mismo razonamiento que en la proposición 4.2. □

Proposición 5.3 Dada una s-expresión dinámica operativa regular G , $\gamma \in BC(G)$, y una serialización de las multiacciones estocásticas de γ : $\langle \alpha_1, r_1 \rangle \dots \langle \alpha_n, r_n \rangle$, entonces existe una secuencia de transiciones de la forma:

$$G \xrightarrow{\langle \alpha_1, r_1 \rangle} G_1 \xrightarrow{(\emptyset)} G_1^* \xrightarrow{\langle \alpha_2, r_2 \rangle} G_2 \xrightarrow{(\emptyset)} G_2^* \xrightarrow{\langle \alpha_3, r_3 \rangle} \dots \xrightarrow{(\emptyset)} G_{n-1}^* \xrightarrow{\langle \alpha_n, r_n \rangle} G'$$

Además, todas las s-expresiones G' obtenidas por la serialización de γ a partir de G son equivalentes con respecto a \equiv .

Demostración:

Se seguiría el mismo razonamiento que en la proposición 4.3, teniendo en cuenta que en la inducción estructural sobre la sintaxis de G habría que considerar el caso de la iteración, que se resolvería con una simple aplicación de la hipótesis de inducción. \square

A partir de todo lo anterior tenemos los corolarios siguientes.

Corolario 5.1 Sean $G \in OpReDynExpr$, $\gamma \in BC(G)$, y una serialización de las multiacciones estocásticas de γ de la forma: $\langle \alpha_1, r_1 \rangle . \langle \alpha_2, r_2 \rangle \dots \langle \alpha_n, r_n \rangle .$ Entonces existe una secuencia de transiciones:

$$G \xrightarrow{\langle \alpha_1, r_1 \rangle} G_1 \xrightarrow{(\emptyset)} G_1^* \xrightarrow{\langle \alpha_2, r_2 \rangle} G_2 \xrightarrow{(\emptyset)} G_2^* \xrightarrow{\langle \alpha_3, r_3 \rangle} \dots \xrightarrow{(\emptyset)} G_{n-1}^* \xrightarrow{\langle \alpha_n, r_n \rangle} G'$$

con $Conflict(G, \langle \alpha_i, r_i \rangle) = Conflict(G_{i-1}^*, \langle \alpha_i, r_i \rangle)$, para $i = 2, \dots, n$.

Además, todas las s-expresiones obtenidas por la serialización de γ a partir de G son equivalentes con respecto a \equiv .

Demostración:

Es una consecuencia inmediata de la proposición anterior y del lema 5.1. \square

Corolario 5.2 Sean $G \in OpReDynExpr$, $\gamma = \{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle, \dots, \langle \alpha_n, r_n \rangle\} \in BC(G)$, y dos permutaciones del conjunto $\{1, \dots, n\}$: $\{i_1, \dots, i_n\}$ y $\{j_1, \dots, j_n\}$. Suponiendo que tenemos dos serializaciones:

$$\begin{aligned} G &\xrightarrow{\langle \alpha_{i_1}, r_{i_1} \rangle} G_1 \xrightarrow{(\emptyset)} G_1^* \xrightarrow{\langle \alpha_{i_2}, r_{i_2} \rangle} \dots \xrightarrow{\langle \alpha_{i_n}, r_{i_n} \rangle} G_n \\ G &\xrightarrow{\langle \alpha_{j_1}, r_{j_1} \rangle} G'_1 \xrightarrow{(\emptyset)} G'_1 \xrightarrow{\langle \alpha_{j_2}, r_{j_2} \rangle} \dots \xrightarrow{\langle \alpha_{j_n}, r_{j_n} \rangle} G'_n \end{aligned}$$

sobre las cuales podemos aplicar $n - 1$ veces la Regla $Sy2'$ (para las mismas acciones a_1, \dots, a_{n-1} , puede que repetidas, pero el mismo número de veces en los dos casos), obteniéndose en cada caso una única transición $G \xrightarrow{\langle \beta_i, R_i \rangle} G_n$ y $G \xrightarrow{\langle \beta_j, R_j \rangle} G'_n$, respectivamente, entonces:

$$G_n \equiv G'_n \quad \text{y} \quad \langle \beta_i, R_i \rangle = \langle \beta_j, R_j \rangle$$

Demostración:

A partir del corolario 5.1 tenemos que $G_n \equiv G'_n$. Por otro lado tenemos $\beta_i = \beta_j$, con sólo aplicar la definición de la sincronización de multiacciones. Finalmente, $R_i = R_j$ se obtiene aplicando la razón de sincronización utilizando la fórmula descrita en la proposición 5.2. \square

Lema 5.2 Sea $G \in OpReDynExpr$, y $\langle \alpha, r \rangle$ ejecutable en G , con

$$Conflict(G, \langle \alpha, r \rangle) = \{ \langle \alpha, r \rangle, \langle \alpha, r_1 \rangle, \dots, \langle \alpha, r_n \rangle \},$$

entonces para cada $\langle \alpha, r_i \rangle \in Conflict(G, \langle \alpha, r \rangle)$ existe $G_i \in OpReDynExpr$, con $G_i \equiv G$, $\langle \alpha, r_i \rangle$ ejecutable en G_i , y $Conflict(G_i, \langle \alpha, r_i \rangle) = Conflict(G, \langle \alpha, r \rangle)$.

Demostración:

Procederemos por inducción estructural sobre la sintaxis de G .

- **Caso base.** Consideraremos $G = \overline{\langle \alpha, r \rangle}$.
Este caso es inmediato, ya que $Conflict(G, \langle \alpha, r \rangle) = \{ \langle \alpha, r \rangle \}$
- **Caso general.** Aplicando la hipótesis de inducción se resuelven los casos de la composición secuencial, restricción, renombramiento y paralelo.

Para el operador de elección, ($G = G' \square F$ o $G = F \square G'$), se sigue exactamente el mismo razonamiento que expusimos en la demostración del lema 4.2.

Para el operador de iteración, el razonamiento es similar al caso del operador de elección. Tendríamos que diferenciar tres casos:

- $G = [G_1 * E' * F]$, bastaría con aplicar la hipótesis de inducción sobre G_1 .
- $G = [E' * G_1 * F]$, necesitamos distinguir dos subcasos:
 - Si $G_1 \not\equiv \overline{E}$ para toda s-expresión estática E , entonces bastaría con aplicar la hipótesis de inducción, ya que los conflictos de G_1 y G coinciden.
 - Si $G_1 \equiv \overline{E}$ para alguna s-expresión estática regular E , entonces de nuevo tenemos que distinguir dos subcasos:
 - Si $\langle \alpha, r_i \rangle \in Conflict(G_1, \langle \alpha, r \rangle)$, entonces bastaría con aplicar la hipótesis de inducción para concluir la propiedad.

- Si $\langle \alpha, r_i \rangle \notin \text{Conflict}(G_1, \langle \alpha, r \rangle)$, entonces aplicando la proposición 5.1, existe alguna s-expresión dinámica operativa H_i , con $H_i \equiv \overline{F}$, en donde $\langle \alpha, r_i \rangle$ es la única multiacción estocástica ejecutable desde H_i . Si ahora consideramos $G_i = [E' * E * H_i]$, se sigue que G_i es operativa, $G_i \equiv G$ y $\text{Conflict}(G_i, \langle \alpha, r_i \rangle) = \text{Conflict}(G, \langle \alpha, r \rangle)$
- $G = [E' * F * G_1]$, se seguiría el mismo razonamiento que en el caso anterior.

Para el operador de sincronización, $G = G' \text{ sy } a$, la demostración es la misma que la desarrollada en el lema 4.2. \square

Mostramos a continuación un ejemplo que pone de manifiesto la necesidad de la restricción sintáctica introducida para el caso de la iteración. Consideraremos un comportamiento paralelo al más alto nivel dentro del cuerpo de la iteración.

Ejemplo 5.1 Sea G la siguiente s-expresión dinámica operativa y *no regular*:

$$G = ([\langle \{a\}, r_1 \rangle * (\overline{\langle \{b\}, r_2 \rangle} \parallel \overline{\langle \{b\}, r_3 \rangle} \parallel \overline{\langle \{\hat{b}, \hat{b}\}, r_4 \rangle}) * \langle \{b\}, r_5 \rangle]) \text{ sy } a$$

Tenemos entonces que

$$\gamma = \{\langle \{b\}, r_2 \rangle, \langle \{b\}, r_3 \rangle, \langle \{\hat{b}, \hat{b}\}, r_4 \rangle\} \in BC(G)$$

y, de acuerdo con la definición de BC , tenemos también que

$$\gamma_1 = \{\langle \{b\}, r_2 \rangle, \langle \{\hat{b}\}, R_{34} \rangle\} \in BC(G)$$

con

$$R_{34} = \frac{r_3}{r_3 + r_5} \cdot \min\{r_3 + r_5, r_4\}$$

Sin embargo, no toda serialización de γ_1 es posible desde G , ya que $\langle \{\hat{b}\}, R_{34} \rangle$ no puede ejecutarse en G' , donde $G \xrightarrow{\langle \{b\}, r_2 \rangle} G'$ y

$$G' = ([\langle \{a\}, r_1 \rangle * (\overline{\langle \{b\}, r_2 \rangle} \parallel \overline{\langle \{b\}, r_3 \rangle} \parallel \overline{\langle \{\hat{b}, \hat{b}\}, r_4 \rangle}) * \langle \{b\}, r_5 \rangle]) \text{ sy } a$$

De hecho, se puede ejecutar $\langle \{\hat{b}\}, R'_{34} \rangle$ desde G' , con $R'_{34} = \min\{r_3, r_4\}$, y en general $R_{34} \neq R'_{34}$. \square

La definición del (multi)sistema de transiciones que considerábamos en la definición 3.11 sigue manteniéndose, teniendo en cuenta las tablas completas de reglas que definen la semántica operacional de sPBC. También seguimos manteniendo el criterio expuesto en la Observación 3.1 (numerando las multiacciones estocásticas) a la hora de no repetir transiciones redundantes originadas por la sincronización, en el conjunto de arcos del sistema de transiciones. Además, se mantiene la política de competencia como forma de resolver los conflictos, y el proceso estocástico asociado a la evolución de una s-expresión dinámica regular G de sPBC seguirá siendo una CTMC (con el mismo criterio para obtener el diagrama de transición de estados asociado a la CTMC de G a partir de $ts(G)$).

A continuación, mostramos un par de ejemplos.

Ejemplo 5.2 Sea la s-expresión estática regular siguiente:

$$E_1 = [\langle a, r_1 \rangle * \langle b, r_2 \rangle ; \langle \{c, d\}, r_3 \rangle * \langle e, r_4 \rangle] rs e$$

El sistema de transiciones de $\overline{E_1}$ es el mostrado en la figura 5.1,

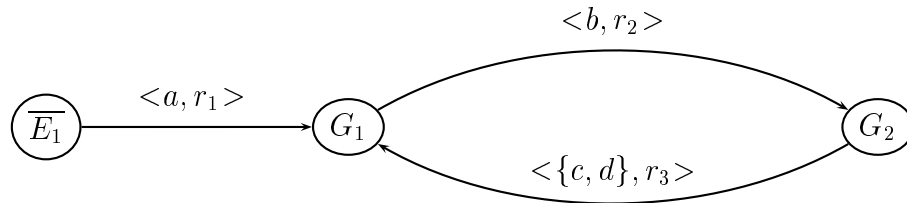


Figura 5.1: Sistema de transiciones de $\overline{E_1}$

donde:

$$G_1 = [\langle a, r_1 \rangle * \overline{\langle b, r_2 \rangle} ; \langle \{c, d\}, r_3 \rangle * \langle e, r_4 \rangle] rs e$$

$$G_2 = [\langle a, r_1 \rangle * \langle b, r_2 \rangle ; \overline{\langle \{c, d\}, r_3 \rangle} * \langle e, r_4 \rangle] rs e$$

□

Ejemplo 5.3 Sea ahora

$$E_2 = [\langle \hat{a}, s_1 \rangle * \langle f, s_2 \rangle ; \langle \hat{c}, s_3 \rangle * \langle e, s_4 \rangle] rs e$$

y consideremos

$$E_3 = [\{a, c\} : (E_1 \parallel E_2)]$$

El sistema de transiciones de $\overline{E_3}$ es el mostrado en la figura 5.2, donde considerando los G_1 y G_2 definidos en el ejemplo anterior y

$$G'_1 = [\langle \hat{a}, s_1 \rangle * \overline{\langle f, s_2 \rangle}; \langle \hat{c}, s_3 \rangle * \langle e, s_4 \rangle] \text{rs } e$$

$$G'_2 = [\langle \hat{a}, s_1 \rangle * \langle f, s_2 \rangle; \overline{\langle \hat{c}, s_3 \rangle} * \langle e, s_4 \rangle] \text{rs } e$$

se tiene:

$$H_1 = [\{a, c\} : (G_1 \parallel G'_1)]$$

$$H_2 = [\{a, c\} : (G_2 \parallel G'_1)]$$

$$H_3 = [\{a, c\} : (G_1 \parallel G'_2)]$$

$$H_4 = [\{a, c\} : (G_2 \parallel G'_2)]$$

Y las razones obtenidas en la sincronización son:

$$R_1 = \min\{r_1, s_1\}, \quad R_3 = \min\{r_3, s_3\}$$

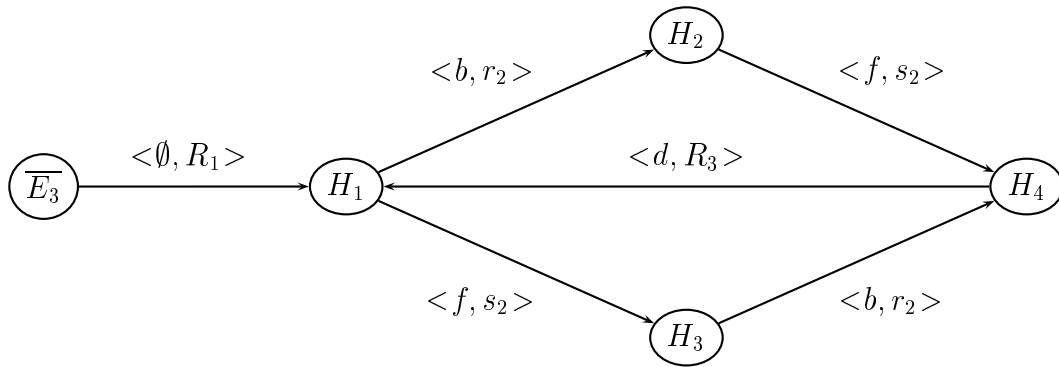


Figura 5.2: Sistema de transiciones de $\overline{E_3}$

□

5.3 Semántica Denotacional

En este apartado necesitamos ampliar la semántica denotacional que expusimos en el capítulo anterior para el nuevo operador de iteración. Necesitamos definir su caja operador, y para cada transición que obtengamos composicionalmente tendremos que saber qué transiciones están en conflicto con ella para poder obtener las *razones de conflicto*, de la misma forma que para los restantes operadores de SPBC. Esto lo haremos por medio de las funciones η y κ que consideramos en el apartado 4.3 del capítulo anterior.

Para seleccionar la caja operador de la iteración en PBC plano [BDK01b] disponemos de dos opciones, tal y como comentamos en el capítulo 2 (apartado 2.1.3). La que proporciona siempre cajas 1-seguras tiene 6 transiciones. La otra opción de caja operador tiene tres transiciones, pero tiene el inconveniente de que se pueden generar redes de Petri 2-seguras, aunque esto último sólo es posible cuando en el cuerpo de la iteración aparece un comportamiento paralelo al más alto nivel. Con la restricción sintáctica introducida este caso no puede darse, así que hemos optado por la versión más simple, que siempre nos proporcionará en nuestro modelo redes 1-seguras. En esta versión tendremos que admitir transiciones con condiciones laterales (arcos de ida y vuelta), de modo que quitaremos de la definición de caja la restricción de que para toda $t \in T : \bullet t \cap t^\bullet = \emptyset$. Por lo demás, mantenemos la definición de s-caja (y s-caja plana) ya establecida (ver apartado 3.3 del capítulo 3).

En el apartado 4.3 del capítulo anterior se definieron las funciones:

$$\eta : T \longrightarrow \mathbb{N}^* \quad \text{y} \quad \kappa : T \longrightarrow \mathcal{P}(\mathbb{N}^*)$$

donde $\eta(t)$ se utilizaba para la numeración de la transición t de acuerdo a nuestro criterio (enumeración de izquierda a derecha, y concatenación en caso de conflicto), y $\kappa(t)$ se utilizaba para identificar el conjunto de conflictos con t . Esas funciones se definían de forma estructural, a la vez que se obtenía la correspondiente s-caja plana.

Entonces, para todos los operadores de sPBC, la estructura de la red se obtiene combinando refinamiento y reetiquetación, obteniéndose siempre s-cajas seguras y limpias. La semántica denotacional vendría dada por el homomorfismo:

$$\begin{aligned} \text{Box}_s(\langle \alpha, r \rangle_i) &= N_{\langle \alpha, r \rangle_i} \\ \text{Box}_s(\text{op}(E_1, \dots, E_n)) &= \Omega_{\text{op}}(\text{Box}_s(E_1), \dots, \text{Box}_s(E_n)) \end{aligned}$$

donde ahora nos quedaría por definir el operador de iteración, siendo su s-caja operador la mostrada en la figura 5.3, con:

$$\rho_{id} = \{(\{\langle \alpha, r \rangle\}, \langle \alpha, r \rangle) \mid \langle \alpha, r \rangle \in \mathcal{SL}\}$$

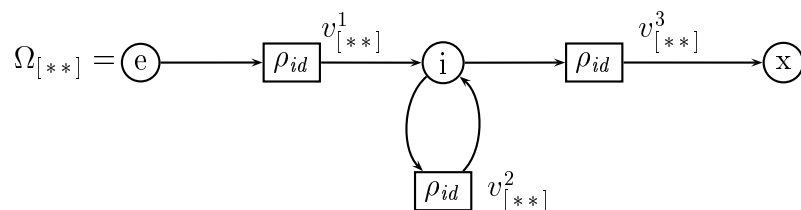


Figura 5.3: s-caja operador para [**]

Considerando que $Box_s(E_i) = (S_i, T_i, W_i, \lambda_i, \mu_i)$ es la s-caja plana correspondiente a E_i , y η_i, κ_i las funciones de numeración y conflicto, respectivamente, para T_i , $i = 1, 2, 3$, se define la función semántica para la iteración como sigue:

$$Box_s([E_1 * E_2 * E_3]) = \Omega_{[**]}(Box_s(E_1), Box_s(E_2), Box_s(E_3)).$$

No se introducen nuevas transiciones con este operador, así que podemos conservar la numeración que ya teníamos en E_1, E_2 y E_3 . Sin embargo, este operador sí introducirá nuevos conflictos. En particular, las transiciones de T_2 que tienen sus precondiciones en ${}^\circ Box_s(E_2)$ están en conflicto con aquellas transiciones con la misma multiacción asociada de T_3 cuyas precondiciones estén en ${}^\circ Box_s(E_3)$. Como consecuencia de trabajar con términos regulares, $Box_s(E_2)$ y $Box_s(E_3)$ tendrán un solo lugar de entrada. Formalmente,

$$\eta(t) = \begin{cases} \eta_1(t) & \text{si } t \in T_1 \\ \eta_2(t) & \text{si } t \in T_2 \\ \eta_3(t) & \text{si } t \in T_3 \end{cases}$$

$$\kappa(t) = \begin{cases} \kappa_1(t) & \text{si } t \in T_1 \\ \kappa_2(t) \cup \kappa_3(t') & \text{si } t \in T_2, \bullet t \in {}^\circ Box_s(E_2), \exists t' \in T_3, \bullet t' \in {}^\circ Box_s(E_3), \lambda(t) = \lambda(t') \\ \kappa_2(t) & \text{si } t \in T_2, \bullet t \in {}^\circ Box_s(E_2), \nexists t' \in T_3, \bullet t' \in {}^\circ Box_s(E_3), \lambda(t) = \lambda(t') \\ \kappa_2(t) & \text{si } t \in T_2, \bullet t \notin {}^\circ Box_s(E_2) \\ \kappa_3(t) \cup \kappa_2(t') & \text{si } t \in T_3, \bullet t \in {}^\circ Box_s(E_3), \exists t' \in T_2, \bullet t' \in {}^\circ Box_s(E_2), \lambda(t) = \lambda(t') \\ \kappa_3(t) & \text{si } t \in T_3, \bullet t \in {}^\circ Box_s(E_3), \nexists t' \in T_2, \bullet t' \in {}^\circ Box_s(E_2), \lambda(t) = \lambda(t') \\ \kappa_3(t) & \text{si } t \in T_3, \bullet t \notin {}^\circ Box_s(E_3) \end{cases}$$

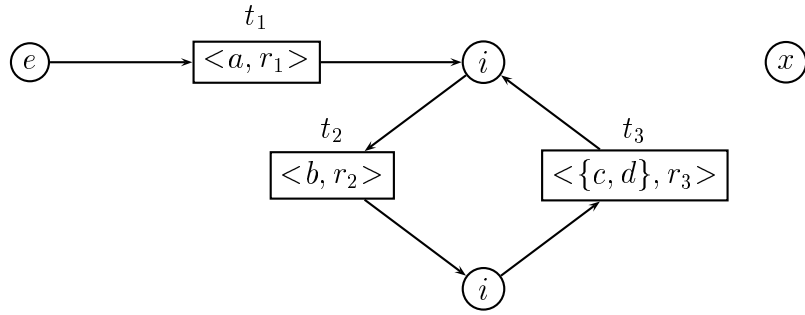
Obsérvese que en el segundo y quinto caso, la función $\kappa(t)$ está bien definida. Por ejemplo en el segundo tenemos que los valores $\kappa_3(t')$ con $t' \in T_3, \bullet t' \in {}^\circ Box_s(E_3)$, $\lambda(t) = \lambda(t')$, son todos iguales; mientras que en el quinto caso, los valores de $\kappa_2(t')$ también coinciden para todos los t' que se pueden utilizar como posibles argumentos.

A continuación, mostramos algunos ejemplos de traducciones de s-expresiones estáticas regulares a s-cajas planas.

Ejemplo 5.4 Sea la siguiente s-expresión estática regular tomada del ejemplo 5.2:

$$E_1 = [\langle a, r_1 \rangle * \langle b, r_2 \rangle ; \langle \{c, d\}, r_3 \rangle * \langle e, r_4 \rangle] rs e$$

Su s-caja plana asociada es la mostrada en la figura 5.4,

Figura 5.4: s-caja plana de E_1

donde:

$$\eta(t_1) = 1, \quad \kappa(t_1) = \{1\},$$

$$\eta(t_2) = 2, \quad \kappa(t_2) = \{2\},$$

$$\eta(t_3) = 3, \quad \kappa(t_3) = \{3\}.$$

□

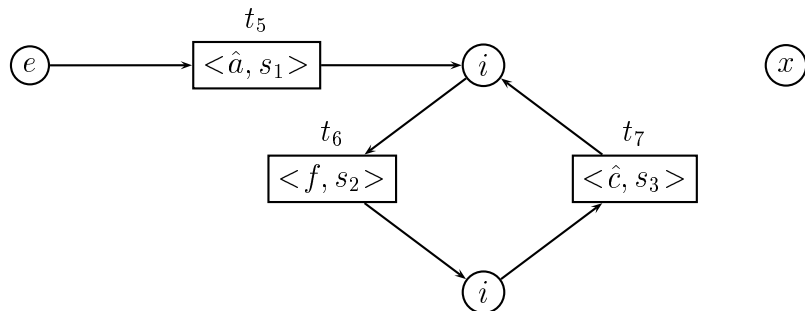
Ejemplo 5.5 Consideremos la s-expresión estática E_3 definida en el ejemplo 5.3:

$$E_3 = [\{a, c\} : (E_1 \parallel E_2)]$$

donde

$$E_2 = [\langle \hat{a}, s_1 \rangle * \langle f, s_2 \rangle ; \langle \hat{c}, s_3 \rangle * \langle e, s_4 \rangle] rs e$$

Obtengamos primero la traducción de E_2 , obtenida cuando en la enumeración de las transiciones conservamos el orden que ocupan dentro de la sintaxis de E_3 . Su s-caja plana asociada es la mostrada en la figura 5.5.

Figura 5.5: s-caja plana de E_2

donde:

$$\eta(t_5) = 5, \quad \kappa(t_5) = \{5\},$$

$$\eta(t_6) = 6, \quad \kappa(t_6) = \{6\},$$

$$\eta(t_7) = 7, \quad \kappa(t_7) = \{7\}.$$

A partir de las s-cajas, de E_1 y E_2 podemos obtener la s-caja asociada a E_3 , que mostramos en la figura 5.6,

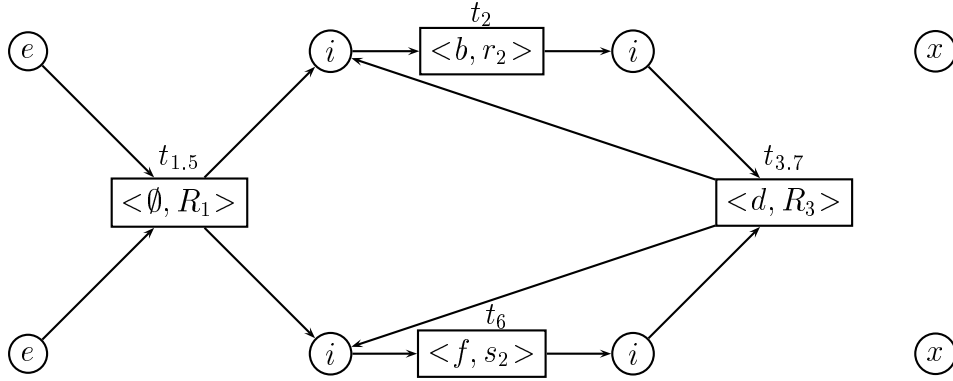


Figura 5.6: s-caja plana de E_3

donde:

$$\begin{aligned} \eta(t_2) &= 2, & \kappa(t_2) &= \{2\}, \\ \eta(t_6) &= 6, & \kappa(t_6) &= \{6\}, \\ \eta(t_{1.5}) &= 1.5, & \kappa(t_{1.5}) &= \{1.5\}, \\ \eta(t_{3.7}) &= 3.7, & \kappa(t_{3.7}) &= \{3.7\}. \end{aligned}$$

□

Finalmente, veamos que se mantiene la equivalencia entre las dos semánticas sobre sPBC completo, tal y como ya teníamos para sPBC finito (teorema 4.1).

Teorema 5.1 Para cada s-expresión estática E de sPBC, el sistema de transiciones $ts(\overline{E})$ asociado a \overline{E} , y el grafo de accesibilidad de la red de Petri estocástica marcada $(Box_s(E), M_e)$ son isomorfos.

Demostración:

Se procede aplicando el mismo razonamiento desarrollado en los capítulos anteriores, a partir del mismo resultado en PBC plano. Esto es, si prescindimos de la información estocástica tanto en el modelo algebraico, como en el modelo de redes, obtenemos el mismo isomorfismo que en PBC, pues en ambos casos estamos considerando únicamente la semántica de orden total (*semántica de entremezcla*). Por lo tanto, la estructura del sistema de transiciones de \overline{E} y la del grafo de accesibilidad de la red, a nivel funcional, es exactamente la misma, pues en ambos casos hemos

aplicado el mismo criterio para eliminar las evoluciones "redundantes" (de acuerdo al criterio de numeración descrito).

Como en ambos casos (álgebra y redes) los conflictos se resuelven de la misma forma, esto es, aplicando la política de competencia, y dado que las multiacciones estocásticas obtenidas en el álgebra y las que etiquetan la correspondiente transición en la red son exactamente iguales (tienen la misma multiacción y la misma razón), obtenemos que ambos sistemas de transiciones se comportan de la misma forma.

Por lo que podremos concluir que existe un isomorfismo entre el grafo de accesibilidad de la s-caja, considerando el marcado inicial y el mencionado sistema de transiciones.

□

Capítulo 6

Equivalencias

En este capítulo nuestro objetivo será definir una relación de equivalencia que capture las ideas expuestas en el capítulo anterior, en el sentido de que dos s-expresiones sean equivalentes si tanto a nivel funcional como a nivel de prestaciones, se comportan de la *misma forma*. Perseguiamos además que dicha relación de equivalencia sea una congruencia.

En el primer apartado introduciremos nuevas reglas estocásticas en la semántica operacional de sPBC, necesarias para establecer nuestra equivalencia. En el segundo apartado se definirá formalmente la relación de equivalencia estocástica y se comprobará que, efectivamente, se trata de una congruencia. Finalmente, el tercer y último apartado de este capítulo lo dedicaremos a presentar algunas observaciones sobre la relación introducida y sus posibles extensiones.

6.1 Las nuevas reglas estocásticas

En primer lugar recordemos la definición de sistema de transiciones, que fue introducido en el capítulo 3:

Definición 6.1 Dada una s-expresión dinámica G , definimos el *(multi)sistema de transiciones* de G , como $ts(G) = (V, A, v_0)$, donde:

- $V = \{ [H]_{\equiv} \mid H \in [G] \}$ es el conjunto de estados,
- $v_0 = [G]_{\equiv}$ es el estado inicial,

- $A = \{ ([H]_{\equiv}, \langle \alpha, r \rangle, [J]_{\equiv}) \mid H \in [G] \wedge H \xrightarrow{\langle \alpha, r \rangle} J \}$
es el multiconjunto de transiciones.

A la hora de construir el conjunto A tendremos en cuenta la Observación 3.1, de modo que no repetiremos aquellos arcos provenientes de sincronizaciones que tengan la misma historia. \square

Recordemos que una consecuencia de utilizar distribuciones exponenciales y de la política de competencia (*race condition*) es que el proceso estocástico asociado a la evolución de una s -expresión dinámica regular de la forma \overline{E} es una CTMC (proposición 3.3), la cual puede obtenerse fácilmente a partir de $ts(\overline{E})$.

Comenzaremos nuestro camino hasta obtener la congruencia definiendo un isomorfismo entre sistemas de transiciones, y veremos que esta noción resulta insuficiente para nuestros propósitos, como de hecho ya ocurre en PBC.

Definición 6.2 Dados dos sistemas de transiciones (V^1, A^1, v_0^1) y (V^2, A^2, v_0^2) , diremos que son isomorfos si existe una biyección $h : V^1 \rightarrow V^2$ tal que

- $h(v_0^1) = v_0^2$
- $\forall v, w \in V^1, y l \in \mathcal{SL}$ se tiene que $A^1(v, l, w) = A^2(h(v), l, h(w))$.

\square

Veamos con un ejemplo que efectivamente este isomorfismo no es realmente una congruencia.

Ejemplo 6.1 Sean

$$E_1 = \langle \{a\}, r_1 \rangle$$

$$F_1 = \langle \{a\}, r_1 \rangle; \langle \{b\}, r_2 \rangle \text{ } rs \text{ } b$$

Los sistemas de transiciones $ts(\overline{E}_1)$ y $ts(\overline{F}_1)$ se muestran en la figura 6.1.



Figura 6.1: $ts(\overline{E}_1)$ y $ts(\overline{F}_1)$

Obviamente, ambos sistemas de transiciones son isomorfos, pero si consideramos:

$$E_2 = E_1; \langle \{c\}, r_3 \rangle$$

$$F_2 = F_1; \langle \{c\}, r_3 \rangle$$

tenemos que sus sistemas de transiciones son los mostrados en la figura 6.2.

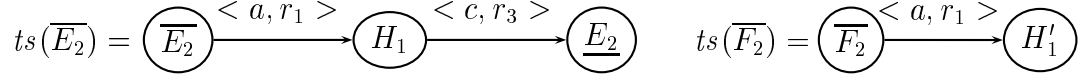


Figura 6.2: $ts(\overline{E_2})$ y $ts(\overline{F_2})$

y obviamente $ts(\overline{E_2})$ y $ts(\overline{F_2})$ no son isomorfos. □

Para resolver este problema necesitamos distinguir entre s-expresiones dinámicas que alcanzan su estado final (como E_1), y aquéllas que nunca alcanzarán su estado final (como F_1). Para ello extenderemos el sistema de transiciones con dos nuevas transiciones (reglas Skp y Rdo en tabla 6.1). Las razones que figuran en las multiacciones que aparecen en estas transiciones tienen básicamente un significado simbólico, ya que dichas transiciones no se tendrán en cuenta para obtener el proceso estocástico asociado. Sólo introducimos dichas transiciones en aras de definir una equivalencia estocástica. Se toma razón 0 para la multiacción *skip*, para poner de manifiesto la interpretación intuitiva de que nunca va a poderse ejecutar, mientras que tomamos ∞ como razón para *redo*, lo cual viene a decirnos que siempre que esté activada, se ejecutará inmediatamente y el proceso se reiniciará.

(Skp)	$\overline{E} \xrightarrow{\langle skip, 0 \rangle} \underline{E}$
(Rdo)	$\underline{E} \xrightarrow{\langle redo, \infty \rangle} \overline{E}$

Tabla 6.1: Reglas Skp y Rdo de sPBC

Denotaremos por $ts_2(\overline{E})$ al sistema de transiciones obtenido tras añadir estas dos nuevas reglas. Los sistemas de transiciones correspondientes para E_1 y F_1 del ejemplo 6.1 son los mostrados en la figura 6.3.

Sin embargo, aunque hemos resuelto un primer problema, aún no tenemos una relación de congruencia. Uno de los problemas viene del hecho de trabajar con una semántica de orden total, que nos impide discernir entre elecciones y paralelos:

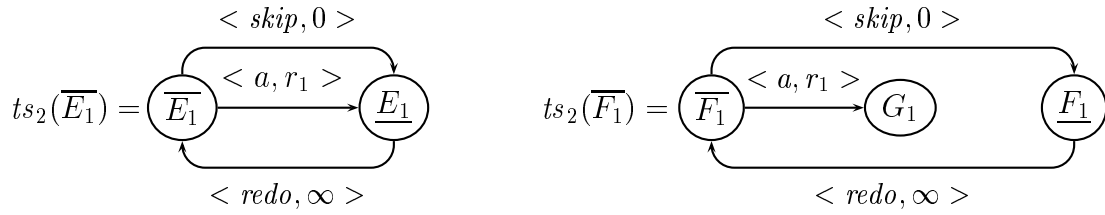


Figura 6.3: $ts_2(\overline{E}_1)$ y $ts_2(\overline{F}_1)$

Ejemplo 6.2 Sean

$$E_3 = \langle \{a\}, r \rangle \parallel \langle \{\hat{a}\}, r \rangle$$

$$F_3 = \langle \{a\}, r \rangle; \langle \{\hat{a}\}, r \rangle \square \langle \{\hat{a}\}, r \rangle; \langle \{a\}, r \rangle$$

entonces $ts_2(\overline{E}_3)$ y $ts_2(\overline{F}_3)$ (mostrados en la figura 6.4) son isomorfos.

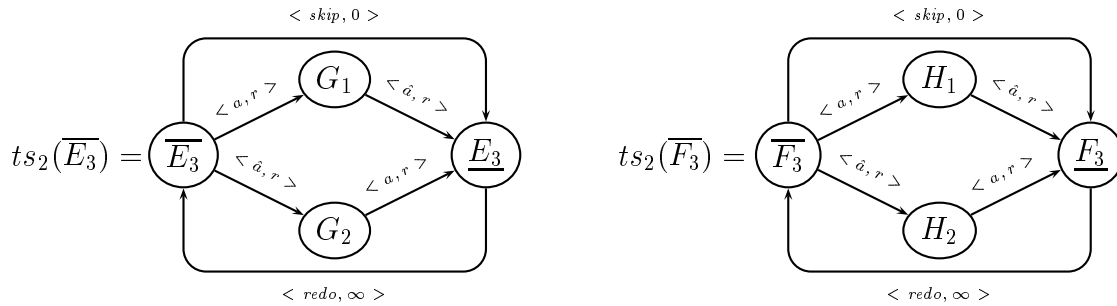


Figura 6.4: $ts_2(\overline{E}_3)$ y $ts_2(\overline{F}_3)$

Pero si ahora consideramos:

$$E_4 = E_3 \text{ sy } a$$

$$F_4 = F_3 \text{ sy } a$$

entonces obtenemos que $ts_2(\overline{E}_4)$ y $ts_2(\overline{F}_4)$ (mostrados en la figura 6.5) no son isomorfos.

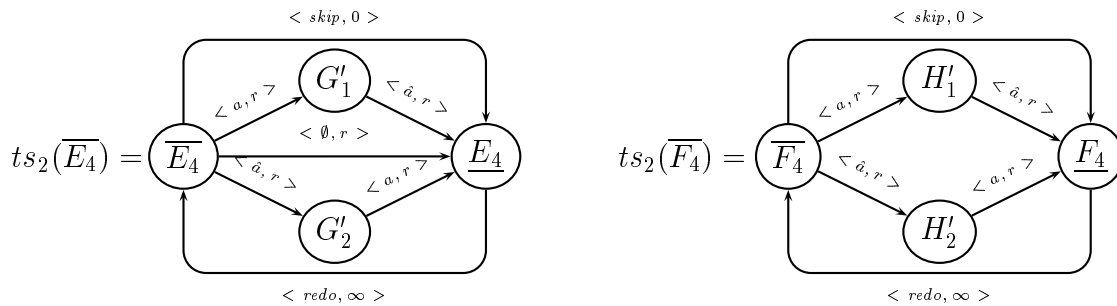


Figura 6.5: $ts_2(\overline{E}_4)$ y $ts_2(\overline{F}_4)$

□

En consecuencia, necesitamos más información en el sistema de transiciones, de forma que seamos capaces de diferenciar las conductas paralelas de las elecciones.

Introducimos al efecto una nueva clase de transiciones, que llamaremos *fantasmas*, queriendo poner de manifiesto así que no van a intervenir en la evolución del sistema (ni a nivel funcional, ni a nivel de prestaciones), sólo se tendrán en cuenta para definir nuestra equivalencia estocástica.

Estas transiciones representarían las ejecuciones potenciales de pares de multiacciones estocásticas en paralelo. Solamente necesitaremos capturar la posible ejecución de dos multiacciones estocásticas en paralelo, y distinguiremos las transiciones fantasmas de las otras, denotándolas de la forma:

$$G_1 \xrightarrow{\langle \alpha_1, r_1 \rangle \parallel \langle \alpha_2, r_2 \rangle} G_2$$

En la tabla 6.2 se introduce la regla que genera esta nueva clase de transiciones. Podemos observar que por cada bolsa concurrente $\{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle\} \in BC(G)$ podríamos obtener dos transiciones diferentes:

$$G \xrightarrow{\langle \alpha_1, r_1 \rangle \parallel \langle \alpha_2, r_2 \rangle} G_2 \quad \text{y} \quad G \xrightarrow{\langle \alpha_2, r_2 \rangle \parallel \langle \alpha_1, r_1 \rangle} G_2$$

de modo que una vez más incluiremos en el sistema de transiciones sólo una de ellas, al igual que hicimos al tratar las sincronizaciones.

<p>(Fan) Sea $\{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle\} \in BC(G)$</p> $\frac{G \xrightarrow{\langle \alpha_1, r_1 \rangle} G_1 \xrightarrow{\emptyset} G_1^* \xrightarrow{\langle \alpha_2, r_2 \rangle} G_2}{G \xrightarrow{\langle \alpha_1, r_1 \rangle \parallel \langle \alpha_2, r_2 \rangle} G_2}$

Tabla 6.2: Regla de transición *Fantasma*

Definimos ahora un tercer sistema de transiciones de \overline{E} , para cada s-expresión estática regular E , teniendo en cuenta las nuevas transiciones generadas por las reglas *Skp*, *Rdo* y *Fan*.

Definición 6.3 Siendo E una s-expresión estática regular, se define $ts_3(\overline{E})$, como

$$ts_3(\overline{E}) = (V \cup \{[E]_{\equiv}\}, A \cup A_{sr} \cup A_g, v_0)$$

donde $ts(\overline{E}) = (V, A, v_0)$, y:

- $A_{sr} = \{ ([\overline{E}]_{\equiv}, \langle skip, 0 \rangle, [E]_{\equiv}), ([E]_{\equiv}, \langle redo, \infty \rangle, [\overline{E}]_{\equiv}) \}$

- $A_g = \{ ([H]_{\equiv}, \langle \alpha, r \rangle \parallel \langle \beta, s \rangle, [J]_{\equiv}) \mid H \in [\overline{E}], H \xrightarrow{\langle \alpha, r \rangle \parallel \langle \beta, s \rangle} J \}$

□

Utilizando este nuevo sistema de transiciones, introducimos un nuevo isomorfismo (\cong_1) entre s-expresiones estáticas regulares.

Definición 6.4 Dado un par cualquiera de s-expresiones estáticas regulares E_1 y E_2 , con:

$$\begin{aligned} ts_3(\overline{E_1}) &= (V^{E_1} \cup \{[E_1]_{\equiv}\}, A^{E_1} \cup A_{sr}^{E_1} \cup A_g^{E_1}, v_0^{E_1}), \text{ con } ts(\overline{E_1}) = (V^{E_1}, A^{E_1}, v_0^{E_1}) \\ ts_3(\overline{E_2}) &= (V^{E_2} \cup \{[E_2]_{\equiv}\}, A^{E_2} \cup A_{sr}^{E_2} \cup A_g^{E_2}, v_0^{E_2}), \text{ con } ts(\overline{E_2}) = (V^{E_2}, A^{E_2}, v_0^{E_2}) \end{aligned}$$

Escribiremos $E_1 \cong_1 E_2$, si existe una función biyectiva:

$$h : V^{E_1} \cup \{[E_1]_{\equiv}\} \rightarrow V^{E_2} \cup \{[E_2]_{\equiv}\}$$

tal que

- $h(v_0^{E_1}) = v_0^{E_2}, h([E_1]_{\equiv}) = [E_2]_{\equiv}$
- $\forall v^{E_1}, w^{E_1} \in V^{E_1}$:
 - $A^{E_1}(v^{E_1}, \langle \alpha, r \rangle, w^{E_1}) = A^{E_2}(h(v^{E_1}), \langle \alpha, r \rangle, h(w^{E_1}))$,
 - $A_g^{E_1}(v^{E_1}, \langle \alpha, r \rangle \parallel \langle \beta, s \rangle, w^{E_1}) = A_g^{E_2}(h(v^{E_1}), \langle \alpha, r \rangle \parallel \langle \beta, s \rangle, h(w^{E_1}))$

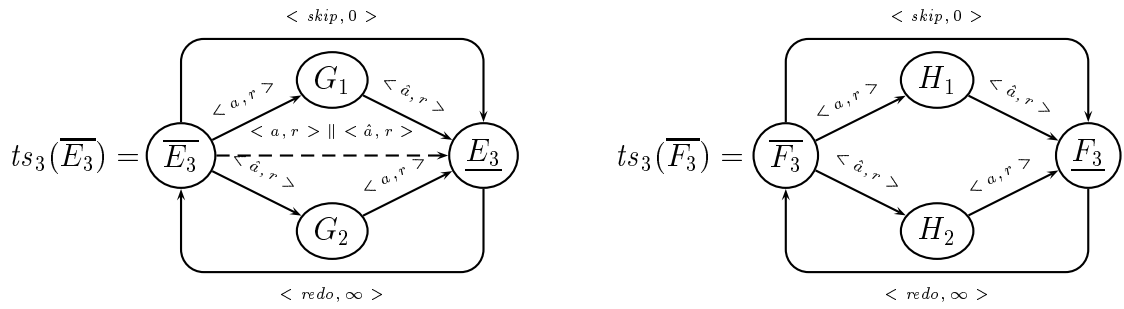
□

Es trivial comprobar que se trata de una relación de equivalencia, es decir, satisface las propiedades reflexiva, simétrica y transitiva.

Ejemplo 6.3 Volvamos a considerar las s-expresiones del ejemplo 6.2:

$$\begin{aligned} E_3 &= \langle \{a\}, r \rangle \parallel \langle \{\hat{a}\}, r \rangle \\ F_3 &= \langle \{a\}, r \rangle; \langle \{\hat{a}\}, r \rangle \quad \square \quad \langle \{\hat{a}\}, r \rangle; \langle \{a\}, r \rangle \end{aligned}$$

$ts_3(\overline{E_3})$ y $ts_3(\overline{F_3})$ se muestran en la figura 6.6, donde se puede apreciar la presencia de la transición fantasma en $ts_3(\overline{E_3})$, de la cual podemos concluir inmediatamente que $E_3 \not\cong_1 F_3$. □

Figura 6.6: $ts_3(\overline{E}_3)$ y $ts_3(\overline{F}_3)$

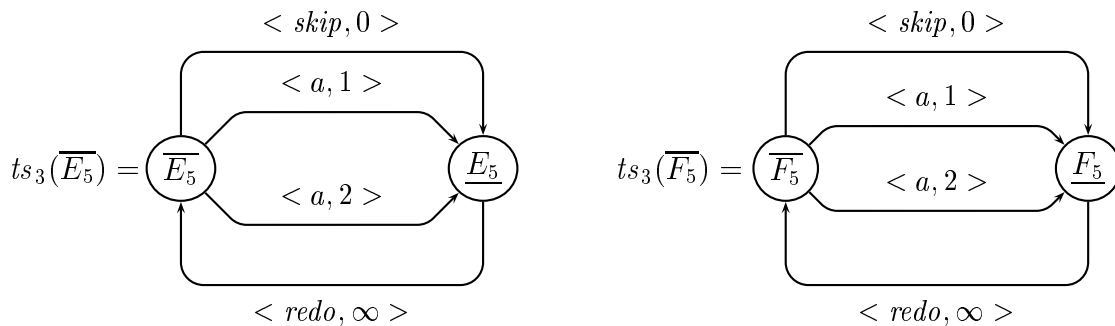
6.2 Congruencia

La relación \cong_1 es básicamente la relación \cong que se tenía en PBC plano. Sin embargo, para sPBC esta relación aún no nos satisface, ya que no es una congruencia, como se pone de manifiesto en el ejemplo siguiente.

Ejemplo 6.4 Consideremos las siguientes s-expresiones estáticas:

$$E_5 = [f : ((\langle \{a, f\}, 1 \rangle \square \langle \{\hat{f}\}, 3 \rangle) \parallel (\langle \{a, f\}, 2 \rangle \square \langle \{\hat{f}\}, 3 \rangle))] \\ F_5 = \langle \{a\}, 1 \rangle \square \langle \{a\}, 2 \rangle$$

Los sistemas de transiciones correspondientes, $ts_3(\overline{E}_5)$ y $ts_3(\overline{F}_5)$, son los mostrados en la figura 6.7.

Figura 6.7: $ts_3(\overline{E}_5)$ y $ts_3(\overline{F}_5)$

Por tanto, son isomorfos. Sin embargo, si tomamos:

$$E_6 = (E_5 \parallel \langle \{\hat{a}\}, 1 \rangle) sy a \\ F_6 = (F_5 \parallel \langle \{\hat{a}\}, 1 \rangle) sy a$$

se puede comprobar que $E_6 \not\cong_1 F_6$, ya que en $ts(\overline{E}_6)$ hay dos arcos conectando \overline{E}_6 con \underline{E}_6 , ambos etiquetados con $\langle \emptyset, 1 \rangle$; mientras que en $ts(\overline{F}_6)$ también hay dos

arcos conectando $\overline{F_6}$ con $\underline{F_6}$, pero ahora uno etiquetado con $\langle \emptyset, 1/3 \rangle$ y el otro con $\langle \emptyset, 2/3 \rangle$. En consecuencia, ni siquiera las CTMCs asociadas son isomorfas. \square

Pretendemos pues definir una relación de equivalencia, que resuelva los problemas anteriores, en la cual dos s-expresiones estáticas regulares serán consideradas equivalentes si, a nivel funcional (cuando eliminamos toda información estocástica) tienen el mismo comportamiento cualitativo, y a nivel de prestaciones generan CTMCs isomorfas. Por ejemplo, con esta nueva equivalencia seremos capaces de identificar las dos s-expresiones siguientes:

$$\langle \alpha, \sum_{i=1}^n r_i \rangle \quad y \quad \langle \alpha, r_1 \rangle \square \dots \square \langle \alpha, r_n \rangle$$

Pero además pretendemos que esta equivalencia se conserve en el ámbito de cualquier operador, esto es, que sea una congruencia.

Por ello, introducimos a continuación una nueva relación de equivalencia, que llamamos *equivalencia estocástica* (definición 6.7). Para ello es necesario anotar cierta información adicional en los arcos del sistema de transiciones, en concreto la información de los conflictos (definición 6.5). Sin embargo es importante precisar que esta información adicional no modifica la interpretación ya conocida de los arcos, es decir, que tan solo anotaremos esta información en aras de poder decidir con exactitud qué arcos están en un mismo conflicto. Asimismo, junto al conjunto de conflictos, anotaremos el número (o la concatenación correspondiente) de la multiacción que aparece etiquetando el arco. Pero, antes que nada, veamos que si una multiacción estocástica $\langle \alpha, r \rangle$ es ejecutable desde una operativa G , entonces su conjunto de conflictos es invariante con respecto a cualquier operativa G' , con $G' \equiv G$, si de ella también es posible ejecutar dicha multiacción estocástica.

Proposición 6.1 Sea $G \in OpReDynExpr$ y $\langle \alpha, r \rangle_n$ una instancia de $\langle \alpha, r \rangle$ ejecutable desde G , con $Conflict(G, \langle \alpha, r \rangle_n) = C$.

Entonces, si $\exists G' \in OpReDynExpr$, con $G' \equiv G$ y $\langle \alpha, r \rangle_n$ también ejecutable desde G' , se tiene que $Conflict(G', \langle \alpha, r \rangle_n) = C$.

Demostración: Por inducción estructural.

Caso base. $G = \overline{\langle \alpha, r \rangle}$. Trivial.

Caso general. Casos posibles:

1. (a) $G = G_1; E$. Si $\langle \alpha, r \rangle_n$ es ejecutable desde G , lo será desde G_1 . Además si G' es operativa, con $G' \equiv G$, y $\langle \alpha, r \rangle_n$ ejecutable desde G' , necesariamente $G' = G'_1; E$, con G'_1 operativa, $G'_1 \equiv G_1$, y $\langle \alpha, r \rangle_n$ ejecutable desde G'_1 , por lo que podremos aplicar la hipótesis de inducción sobre G_1 . Además, como los conflictos no varían, fácilmente se deduce la propiedad:

$$\begin{aligned} \text{Conflict}(G', \langle \alpha, r \rangle_n) &= \text{Conflict}(G'_1, \langle \alpha, r \rangle_n) = \\ \text{Conflict}(G_1, \langle \alpha, r \rangle_n) &= \text{Conflict}(G, \langle \alpha, r \rangle_n) \end{aligned}$$

- (b) $G = E; G_1$. Análogo al subcaso anterior.

2. (a) $G = G_1 \square E$. Habrá que considerar si G_1 es o no equivalente estructuralmente a una s-expresión dinámica inicial (ya que sí lo es, el conjunto de conflictos puede variar).
- i. Si $G_1 \not\equiv \overline{F}$, $\forall F$ estática. Entonces los conflictos no varían, y bastaría con aplicar la hipótesis de inducción sobre G_1 .
 - ii. Si $\exists F$ estática tal que $G_1 \equiv \overline{F}$. Entonces si $\langle \alpha, r \rangle_n$ es ejecutable en G , lo será también en G_1 y puede que otra multiacción estocástica $\langle \alpha, r' \rangle_{n'}$ lo sea en H , donde $H \equiv \overline{E}$, aumentándose así su conjunto de conflictos. Cualquier operativa $G' \equiv G$, desde donde se pueda ejecutar $\langle \alpha, r \rangle_n$ será de la forma $G'_1 \square E$ (con $G'_1 \equiv G_1$). En cualquier caso, aplicando la hipótesis de inducción sobre G_1 y H , y aplicando la definición de conflicto:

$$\begin{aligned} \text{Conflict}(G', \langle \alpha, r \rangle_n) &= \\ \text{Conflict}(G'_1, \langle \alpha, r \rangle_n) \cup \text{Conflict}(H, \langle \alpha, r \rangle) &= \\ \text{Conflict}(G_1, \langle \alpha, r \rangle_n) \cup \text{Conflict}(H, \langle \alpha, r \rangle) &= \\ \text{Conflict}(G, \langle \alpha, r \rangle_n) & \end{aligned}$$

donde hemos considerado que $\langle \alpha, r \rangle_n$ es ejecutable desde G'_1 .

- (b) $G = E \square G_1$. Análogo al subcaso anterior para la elección.

3. (a) $G = G_1 \parallel H$. Se sigue con un razonamiento similar al de la secuencia.

- (b) $G = H \parallel G_1$. Análogo al subcaso anterior.

4. $G = G_1[f]$. Si $\langle \alpha, r \rangle_n$ es ejecutable desde G es debido a que $\langle f^{-1}(\alpha), r \rangle_n$ lo es desde G_1 . De esta forma cualquier operativa G' , $G' \equiv G$, sería de la forma $G'_1[f]$, con $G'_1 \equiv G_1$. Si $\langle \alpha, r \rangle_n$ es ejecutable desde G' es porque $\langle f^{-1}(\alpha), r \rangle_n$ lo es desde G'_1 , por lo que bastaría con aplicar la hipótesis de inducción para comprobar que los conflictos no varían.

5. $G = G_1 \text{ rs } a$. Si $\langle \alpha, r \rangle_n$ es ejecutable G , es porque lo es desde G_1 y $a, \hat{a} \notin A(\alpha)$. Por tanto han de coincidir los conflictos de G y de G_1 para $\langle \alpha, r \rangle_n$. Además, cualquier operativa $G' \equiv G$, tal que $\langle \alpha, r \rangle_n$ es ejecutable desde G' , será de la forma $G' = G'_1 \text{ rs } a$, con $\langle \alpha, r \rangle_n$ ejecutable desde G'_1 y $G'_1 \equiv G_1$, y ya sólo bastaría con aplicar la hipótesis de inducción.
6. (a) $G = [G_1 * E_1 * E_2]$. Se hace un razonamiento similar al de la secuencia y el paralelo.
- (b) $G = [E_1 * G_1 * E_2]$. Se hace un razonamiento similar al caso de la elección, diferenciando si G_1 es inicial o no.
- (c) $G = [E_1 * E_2 * G_1]$. Se hace un razonamiento similar al subcaso anterior.
7. $G = G_1 \text{ sy } a$
- (a) Si $\langle \alpha, r \rangle_n$ es ejecutable desde G_1 , como los conflictos de G_1 se mantienen en G , bastaría con aplicar la hipótesis de inducción ¹.
- (b) Si $\langle \alpha, r \rangle_n$ no es ejecutable desde G_1 , y sí lo es desde G al aplicar la regla $\text{Sy}\mathcal{Z}'$, entonces procederemos sobre el número de veces que se ha aplicado dicha regla para obtener $\langle \alpha, r \rangle_n$.

Caso base. $\langle \alpha, r \rangle_n$ se ha obtenido al aplicar la regla $\text{Sy}\mathcal{Z}'$ una sola vez. Entonces:

$$\langle \alpha, r \rangle_n = \langle \alpha_1, r_1 \rangle_{n_1} \oplus_a \langle \alpha_2, r_2 \rangle_{n_2}, \quad \text{siendo } n = n_1.n_2 \text{ (concatenación)}$$

donde $\langle \alpha_1, r_1 \rangle_{n_1}$ y $\langle \alpha_2, r_2 \rangle_{n_2}$ son ejecutables desde G_1 .

Además, cualquier operativa $G' \equiv G$, con $\langle \alpha, r \rangle_n$ ejecutable desde G' , cumple que es de la forma $G' = G'_1 \text{ sy } a$, con $G'_1 \equiv G_1$, y como $\langle \alpha, r \rangle_n$ se ha obtenido de una sincronización se tendrá que $\langle \alpha_1, r_1 \rangle_{n_1}$ y $\langle \alpha_2, r_2 \rangle_{n_2}$ son ejecutables desde G'_1 . Así, aplicando la hipótesis de inducción:

$$\begin{aligned} \text{Conflict}(G'_1, \langle \alpha_1, r_1 \rangle_{n_1}) &= \text{Conflict}(G_1, \langle \alpha_1, r_1 \rangle_{n_1}) \\ \text{Conflict}(G'_1, \langle \alpha_2, r_2 \rangle_{n_2}) &= \text{Conflict}(G_1, \langle \alpha_2, r_2 \rangle_{n_2}) \end{aligned}$$

¹Nótese que no puede haber nuevas multiacciones estocásticas $\langle \alpha, r' \rangle$ provenientes de sincronización que estén en conflicto con $\langle \alpha, r \rangle_n$, según nuestra definición de conjunto de conflictos.

por lo que aplicando la definición de conflicto en la sincronización se llega a que:

$$\begin{aligned} \text{Conflict}(G', \langle \alpha, r \rangle_n) &= \text{Conflict}(G'_1 \text{ sy } a, \langle \alpha_1, r_1 \rangle_{n_1} \oplus_a \langle \alpha_2, r_2 \rangle_{n_2}) = \\ &= \text{Conflict}(G_1 \text{ sy } a, \langle \alpha_1, r_1 \rangle_{n_1} \oplus_a \langle \alpha_2, r_2 \rangle_{n_2}) = \text{Conflict}(G, \langle \alpha, r \rangle_n) \end{aligned}$$

Caso general. $\langle \alpha, r \rangle_n$ se ha obtenido al aplicar la regla $\text{Sy}2'$ $k-1$ veces, con $k > 2$. Entonces:

$$\langle \alpha, r \rangle_n = \langle \alpha_1, r_1 \rangle_{n_1} \oplus_a \langle \alpha_2, r_2 \rangle_{n_2} \oplus_a \dots \oplus_a \langle \alpha_n, r_k \rangle_{n_k}$$

Con un razonamiento análogo al caso base y aplicando la hipótesis de inducción, llegaríamos a la propiedad deseada, ya que los conflictos en la sincronización se obtienen a partir del producto cartesiano de las multiacciones estocásticas que intervienen en dicha sincronización. □

La proposición anterior nos permite definir:

$$\text{Conflict}([G]_{\equiv}, \langle \alpha, r \rangle_n) = \text{Conflict}(G', \langle \alpha, r \rangle_n)$$

para todo $G \in \text{ReDynExpr}$, tomando cualquier G' operativa, con $\langle \alpha, r \rangle_n$ ejecutable desde G' y $G' \equiv G$.

Ya estamos en condiciones de definir el que llamaremos *nuevo sistema de transiciones*, anotando en los arcos junto a la multiacción estocástica correspondiente su numeración y su conjunto de conflictos. Reiteramos de nuevo el hecho de que ello no afecta a la interpretación que hacemos de los arcos, ni condiciona la posible evolución del sistema por medio de la ejecución de las correspondientes multiacciones estocásticas. Como ya distinguimos (con la numeración) todas las multiacciones estocásticas, tendremos un sistema en lugar de un multi-sistema.

Definición 6.5 Dada E una s-expresión estática regular, definimos el nuevo sistema de transiciones de \overline{E} , al que denotaremos por $\text{nts}(\overline{E})$, de la forma:

$$\text{nts}(\overline{E}) = (V \cup \{[E]_{\equiv}\}, A_a \cup A_{sr} \cup A_g, v_0)$$

donde

- $V = \{[H]_{\equiv} \mid H \in \overline{E}\}$
- $v_0 = [E]_{\equiv}$

- $A_a = \{ ([H]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J]_{\equiv}) \mid H \in [\overline{E}], H \xrightarrow{\langle \alpha, r \rangle^n} J \text{ y } C = \text{Conflict}([H]_{\equiv}, \langle \alpha, r \rangle_n) \}$
- $A_{sr} = \{ ([\overline{E}]_{\equiv}, \langle \text{skip}, 0 \rangle, [E]_{\equiv}), ([E]_{\equiv}, \langle \text{redo}, \infty \rangle, [\overline{E}]_{\equiv}) \}$
- $A_g = \{ ([H]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J]_{\equiv}) \mid H \in [\overline{E}], H \xrightarrow{\langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}} J \}$ □

Observemos que en los casos particulares de *skip* y *redo* no necesitamos introducir ni la numeración ni los conflictos, y por ello, las mantenemos tal como estaban.

Para definir la equivalencia necesitamos esta definición previa:

Definición 6.6 Dados $nts(\overline{E})$ y $(v, \langle \alpha, r \rangle_{n,C}, w) \in A_a$, entonces llamaremos *razón de paso entre los nodos v y w con respecto a la multiacción α y al conflicto C* , a

$$r(v, \alpha, C, w) = \sum \{ r_j \mid (v, \langle \alpha, r_j \rangle_{n',C}, w) \in A_a \}$$

Será por tanto la suma de las razones de las multiacciones estocásticas que van de v a w , con multiacción asociada α y que tienen el mismo conjunto de conflictos C .

La *razón de conflicto de C* se define como:

$$rc(C) = \sum \{ r_j \mid \langle \alpha, r_j \rangle_n \in C \}$$

Al conjunto de conflictos que pueden aparecer en un nodo con respecto a una multiacción lo denotaremos por $\mathcal{C}(v, \alpha)$:

$$\mathcal{C}(v, \alpha) = \{ C_j \mid (v, \langle \alpha, r_i \rangle_{n',C_j}, w_k) \in A_a \}$$

esto es, $\mathcal{C}(v, \alpha)$ es el conjunto de los conjuntos conflictos C_j tales que $\langle \alpha, r_i \rangle_{n',C_j}$ es la etiqueta de un arco que sale de $v' \equiv v$.

También se puede definir la *razón de paso entre los nodos v y w con respecto a la multiacción α* , como:

$$r(v, \alpha, w) = \sum_k \{ r(v, \alpha, C_k, w) \mid (v, \langle \alpha, r_j \rangle_{n',C_k}, w) \in A_a \}$$

□

Podemos finalmente definir nuestra relación de equivalencia (\sim):

Definición 6.7 Dadas dos s-expresiones estáticas regulares E_1, E_2 , con:

$$\begin{aligned} nts(\overline{E_1}) &= (V^{E_1} \cup \{[E_1]_{\equiv}\}, A_a^{E_1} \cup A_{sr}^{E_1} \cup A_g^{E_1}, v_0^{E_1}) \\ nts(\overline{E_2}) &= (V^{E_2} \cup \{[E_2]_{\equiv}\}, A_a^{E_2} \cup A_{sr}^{E_2} \cup A_g^{E_2}, v_0^{E_2}) \end{aligned}$$

Diremos que E_1 y E_2 son *estocásticamente equivalentes*, y lo denotaremos por $E_1 \sim E_2$, si existe una función biyectiva

$$\phi : V^{E_1} \cup \{[E_1]_{\equiv}\} \rightarrow V^{E_2} \cup \{[E_2]_{\equiv}\}$$

y para todo $v \in V^{E_1}$ y para todo α tal que $\mathcal{C}(v, \alpha) \neq \emptyset$, existe una biyección

$$h_{v,\alpha} : \mathcal{C}(v, \alpha) \rightarrow \mathcal{C}(\phi(v), \alpha), \quad \text{tal que } rc(C_j) = rc(h_{v,\alpha}(C_j)), \quad \forall C_j \in \mathcal{C}(\phi(v), \alpha)$$

$h_{v,\alpha}$ es una función biyectiva que asocia a cada conjunto de conflictos en $\mathcal{C}(\phi(v), \alpha)$ otro conjunto de conflictos con respecto a la misma multiacción en la imagen del nodo, tal que se conserva la razón de conflicto. Además, han de cumplirse los puntos siguientes:

- $\phi(\overline{[E_1]_{\equiv}}) = \overline{[E_2]_{\equiv}}$ y $\phi([E_1]_{\equiv}) = [E_2]_{\equiv}$
- Consideremos $\phi(v^1) = v^2 \in V^{E_2}$, $\phi(w^1) = w^2 \in V^{E_2}$, $h_{v^1,\alpha}(C_1) = C'_1$ y $h_{v^1,\beta}(C_2) = C'_2$ entonces:
 - Si $(v^1, \langle \alpha, r_1 \rangle_{n_1, C_1}, w^1) \in A_a^{E_1}$, entonces $\exists s_1, n'_1$ tal que $(v^2, \langle \alpha, s_1 \rangle_{n'_1, C'_1}, w^2) \in A_a^{E_2}$, con $r(v^1, \alpha, C_1, w^1) = r(v^2, \alpha, C'_1, w^2)$.
 - Si $(v^2, \langle \alpha, s_1 \rangle_{n'_1, C'_1}, w^2) \in A_a^{E_2}$, entonces $\exists r_1, n_1$ tal que $(v^1, \langle \alpha, r_1 \rangle_{n_1, C_1}, w^1) \in A_a^{E_1}$, con $r(v^2, \alpha, C'_1, w^2) = r(v^1, \alpha, C_1, w^1)$.
 - Si $(v^1, \langle \alpha, r_1 \rangle_{n_1} \parallel \langle \beta, r_2 \rangle_{n_2}, w^1) \in A_g^{E_1}$, entonces
 - ▶ $\exists (v^1, \langle \alpha, r_1 \rangle_{n_1, C_1}, x^1) \in A_a^{E_1}$,
 - ▶ $\exists (v^1, \langle \beta, r_2 \rangle_{n_2, C_2}, y^1) \in A_a^{E_1}$,
 - ▶ $\exists s_1, s_2, n'_1, n'_2$ tal que $(v^2, \langle \alpha, s_1 \rangle_{n'_1} \parallel \langle \beta, s_2 \rangle_{n'_2}, w^2) \in A_g^{E_2}$, con $(v^2, \langle \alpha, s_1 \rangle_{n'_1, C'_1}, \phi(x^1)) \in A_a^{E_2} \wedge (v^2, \langle \beta, s_2 \rangle_{n'_2, C'_2}, \phi(y^1)) \in A_a^{E_2}$.
 - Si $(v^2, \langle \alpha, s_1 \rangle_{n'_1} \parallel \langle \beta, s_2 \rangle_{n'_2}, w^2) \in A_g^{E_2}$ entonces:
 - ▶ $\exists (v^2, \langle \alpha, s_1 \rangle_{n'_1, C'_1}, x^2) \in A_a^{E_2}$
 - ▶ $\exists (v^2, \langle \beta, s_2 \rangle_{n'_2, C'_2}, y^2) \in A_a^{E_2}$
 - ▶ $\exists r_1, r_2, n_1, n_2$ tal que $(v^1, \langle \alpha, r_1 \rangle_{n_1} \parallel \langle \beta, r_2 \rangle_{n_2}, w^1) \in A_g^{E_1}$, con $(v^1, \langle \alpha, r_1 \rangle_{n_1, C_1}, \phi^{-1}(x^2)) \in A_a^{E_1} \wedge (v^1, \langle \beta, r_2 \rangle_{n_2, C_2}, \phi^{-1}(y^2)) \in A_a^{E_1}$.

□

La interpretación intuitiva de la definición anterior es la siguiente:

Existe una correspondencia biyectiva entre los nodos de los dos nuevos sistemas de transiciones, de forma que:

- Hay una correspondencia entre las s-expresiones iniciales y finales.
- En cada nodo, para cada conjunto de conflictos hay uno asociado en el nodo imagen (con respecto a la misma multiacción), conservándose las razones de conflicto, y las razones de paso con respecto a la multiacción y su conflicto .
- La presencia de un arco fantasma en uno de los lados implica la presencia de otro arco fantasma en el otro lado, con las mismas multiacciones, aunque posiblemente con razones distintas. Además el arco fantasma implica la presencia de arcos de $A_a^{E_1}$ ($A_a^{E_2}$) correspondientes a las multiacciones estocásticas que lo etiquetan, así como su contrapartida.

Corolario 6.1 Los pares de s-expresiones estáticas regulares (E, F) que verifican $E \sim F$ tienen CTMCs isomorfas.

Demostración:

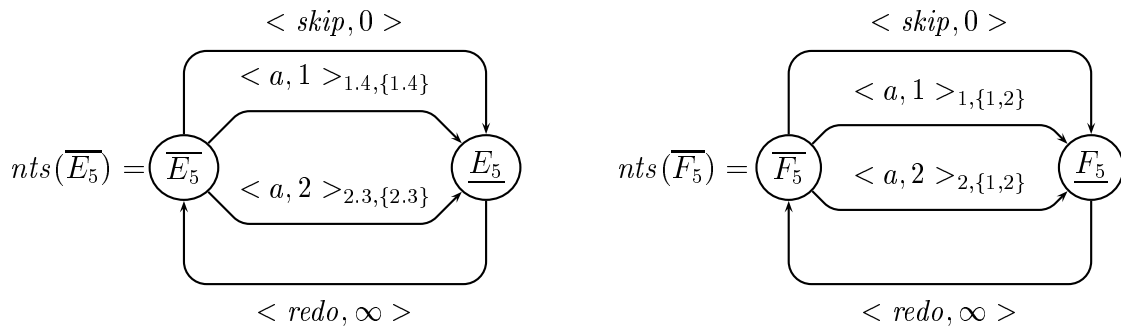
Según la definición de \sim , para nodos equivalentes, tomando cada multiacción estocástica, tenemos que la razón de paso respecto a ella y a su conjunto de conflicto coinciden en $ts(\overline{E})$ y en $ts(\overline{F})$. Por tanto, como la correspondencia de conflictos es unívoca, se tendrá que también se conserva la razón de paso entre dos nodos respecto a una multiacción. En consecuencia, la suma de todas las razones de los arcos que enlazan dos nodos de $ts(\overline{E})$ es la misma en sus correspondientes nodos imagen (en $ts(\overline{F})$), teniéndose entonces que sus correspondientes CTMCs son isomorfas. \square

Ejemplo 6.5 Consideremos de nuevo las s-expresiones estáticas del ejemplo 6.4:

$$E_5 = [f : ((\langle \{a, f\}, 1 \rangle_1 \square \langle \{\hat{f}\}, 3 \rangle_2) \parallel (\langle \{a, f\}, 2 \rangle_3 \square \langle \{\hat{f}\}, 3 \rangle_4))] \\ F_5 = \langle \{a\}, 1 \rangle_1 \square \langle \{a\}, 2 \rangle_2$$

$nts(\overline{E}_5)$ y $nts(\overline{F}_5)$ son los mostrados en la figura 6.8.

Claramente se puede apreciar que $E_5 \not\sim F_5$, ya que no es posible encontrar la biyección entre los conjuntos de conflictos en los dos sistemas de transiciones. \square

Figura 6.8: $nts(\overline{E}_5)$ y $nts(\overline{F}_5)$

Ejemplo 6.6 Los siguientes pares de s-expresiones estáticas regulares son estocásticamente equivalentes:

$$\begin{aligned} E_7 = \langle \{a\}, 2 \rangle &\sim E_8 = \langle \{a\}, 1 \rangle \square \langle \{a\}, 1 \rangle \\ F_7 = \langle \{\hat{a}\}, 3 \rangle &\sim F_8 = \langle \{\hat{a}\}, 1 \rangle \square \langle \{\hat{a}\}, 2 \rangle \end{aligned}$$

Además, tenemos que

$$E_7 \parallel F_7 \sim E_8 \parallel F_8$$

y considerando:

$$\begin{aligned} E_9 &= (E_7 \parallel F_7) \text{ sy } a \\ F_9 &= (E_8 \parallel F_8) \text{ sy } a \end{aligned}$$

se tiene también que

$$E_9 \sim F_9$$

lo que puede comprobarse fácilmente a partir de $nts(\overline{E}_9)$ y $nts(\overline{F}_9)$, los cuales se muestran en las figuras 6.9 y 6.10, respectivamente. \square

Para poder comprobar que realmente \sim es una congruencia necesitamos unos resultados previos.

Lema 6.1 Sea $G \in OpReDynExpr$, $\{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle, \dots, \langle \alpha_n, r_n \rangle\} \in BC(G)$, con $n \geq 2$, y dado $\langle \alpha_1, s_1 \rangle \in Conflict(G, \langle \alpha_1, r_1 \rangle)$. Entonces $\exists G' \equiv G$, operativa, tal que $\{\langle \alpha_1, s_1 \rangle, \langle \alpha_2, r_2 \rangle, \dots, \langle \alpha_n, r_n \rangle\} \in BC(G')$.

Demostración:

Por inducción estructural.

- **Caso base.** Este caso debe ser necesariamente un paralelo, en el que n multiacciones estocásticas estén activadas a la vez:

$$G = G_1 \parallel G_2 \parallel \dots \parallel G_n$$

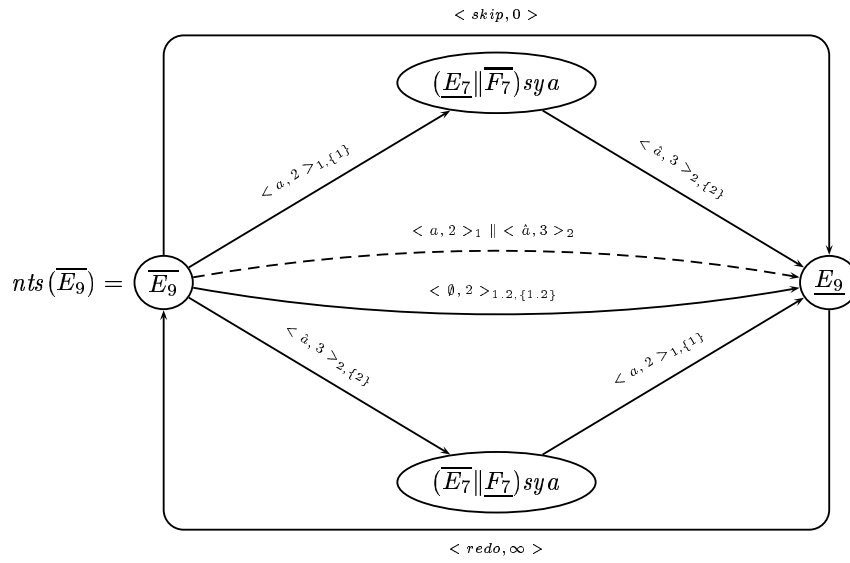


Figura 6.9: $nts(\overline{E}_9)$

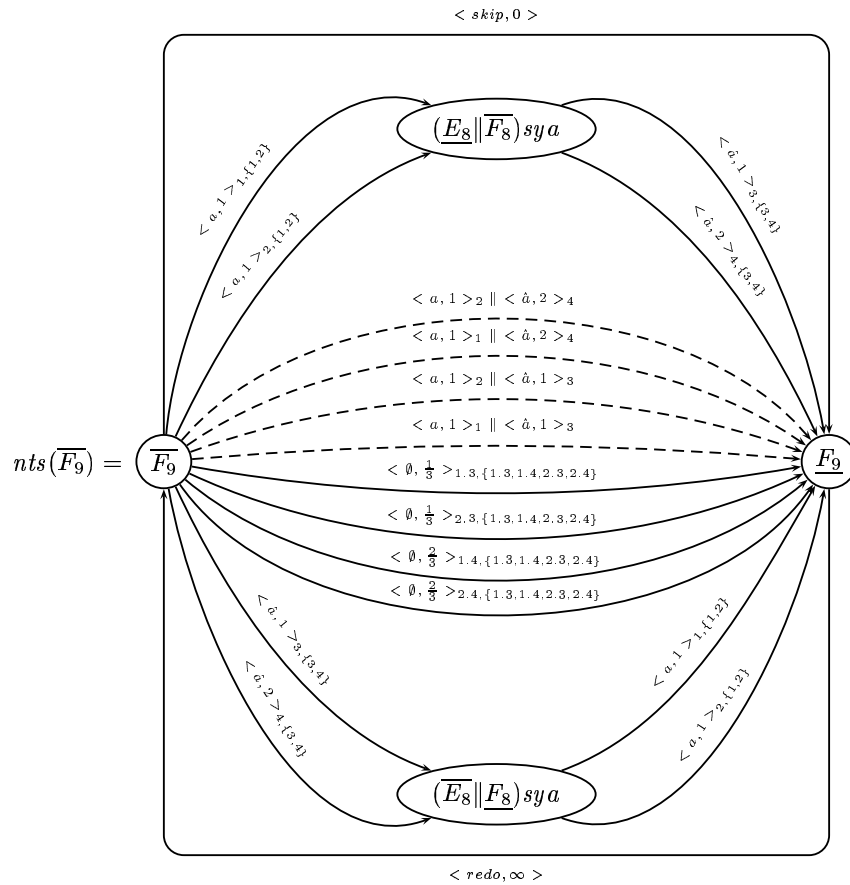


Figura 6.10: $nts(\overline{F}_9)$

donde cada G_i es operativa, y $\langle \alpha_i, r_i \rangle$ es ejecutable desde G_i . Como existe una operativa G'_1 , con $G'_1 \equiv G_1$, donde $\langle \alpha_1, s_1 \rangle$ es ejecutable (lema 5.2), entonces con

$$G' = G'_1 \parallel G_2 \parallel \dots \parallel G_n$$

se tiene que G' operativa, $G' \equiv G$ y $\{\langle \alpha_1, s_1 \rangle, \langle \alpha_2, r_2 \rangle, \dots, \langle \alpha_n, r_n \rangle\} \in BC(G')$.

- **Caso general.** Supongamos que G_1 cumple la hipótesis de inducción. De esta forma, según la sintaxis de las operativas, G será de la forma:

1. $G = G_1; E$ ó $G = E; G_1$, donde E es una s-expresión estática regular cualquiera.

Este caso se resuelve, aplicando la hipótesis de inducción.

2. $G = G_1 \square E$ ó $G = E \square G_1$, donde E es una s-expresión estática regular cualquiera.

Por la restricción sintáctica adoptada, no es posible que G_1 sea inicial, esto es, que exista F estática con $G_1 \equiv \overline{F}$ (ya que supondría tener un paralelo al más alto nivel dentro de una elección). Por lo tanto, los conflictos de G son los de G_1 , así como las multiacciones ejecutables y las bolsas concurrentes, con lo que bastaría con aplicar la hipótesis de inducción.

3. $G = G_1 rs a$. Si $\{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle, \dots, \langle \alpha_n, r_n \rangle\} \in BC(G)$, entonces $a, \hat{a} \notin A(\alpha_i)$, $i = 1, \dots, n$, siendo también una bolsa concurrente de G_1 , con lo que bastaría con aplicar la hipótesis de inducción.

4. $G = G_1[f]$. Si $\{\langle \alpha_1, r_1 \rangle, \langle \alpha_2, r_2 \rangle, \dots, \langle \alpha_n, r_n \rangle\} \in BC(G)$, entonces $\{\langle f^{-1}(\alpha_1), r_1 \rangle, \langle f^{-1}(\alpha_2), r_2 \rangle, \dots, \langle f^{-1}(\alpha_n), r_n \rangle\} \in BC(G_1)$, y si $\langle \alpha_1, s_1 \rangle \in Conflict(G, \langle \alpha_1, r_1 \rangle)$, es porque $\langle f^{-1}(\alpha_1), s_1 \rangle \in Conflict(G_1, \langle f^{-1}(\alpha_1), r_1 \rangle)$, con lo que se reduciría a aplicar la hipótesis de inducción.

5. $G = [G_1 * E * F]$, $G = [E * G_1 * F]$ ó $G = [E * F * G_1]$, donde E y F son s-expresiones estáticas regulares cualquiera.

El primer caso se resolvería fácilmente aplicando la hipótesis de inducción.

Para el segundo y tercer caso, por la restricción sintáctica, no es posible que G_1 sea inicial, teniéndose en G las multiacciones estocásticas, las bolsas concurrentes y los conflictos de G_1 , con lo que se resolvería de nuevo aplicando la hipótesis de inducción.

6. $G = G_1 \text{ sy } a$.

- (a) Si $\langle \alpha_1, r_1 \rangle$ no proviene de sincronizar en a , entonces $\langle \alpha_1, s_1 \rangle$ tampoco (debido a la restricción sintáctica y a la definición de *Conflict*). Entonces podemos suponer que en la bolsa concurrente algunas multiacciones estocásticas podrían provenir de sincronizar. Sin pérdida de generalidad, supondremos que son $\langle \alpha_i, r_i \rangle$, $i = 2, \dots, m$ (con $m \leq n$), que serían de la forma:

$$\langle \alpha_i, r_i \rangle = \langle \alpha_{i_1}, r_{i_1} \rangle \oplus_a \langle \alpha_{i_2}, r_{i_2} \rangle$$

donde $\{\langle \alpha_{i_1}, r_{i_1} \rangle, \langle \alpha_{i_2}, r_{i_2} \rangle\} \in BC(G_1)$. Con lo cual tendremos que:

$$\{\langle \alpha_1, r_1 \rangle, \langle \alpha_{2_1}, r_{2_1} \rangle, \langle \alpha_{2_2}, r_{2_2} \rangle \dots, \langle \alpha_{m_1}, r_{m_1} \rangle, \langle \alpha_{m_2}, r_{m_2} \rangle, \langle \alpha_{m+1}, r_{m+1} \rangle, \dots, \langle \alpha_n, r_n \rangle\} \in BC(G_1)$$

Como $\langle \alpha_1, s_1 \rangle \in \text{Conflict}(G_1, \langle \alpha_1, r_1 \rangle)$ (ya que no proviene de sincronizar), entonces, por la hipótesis de inducción tenemos que existe G'_1 operativa, con $G'_1 \equiv G_1$ y

$$\{\langle \alpha_1, s_1 \rangle, \langle \alpha_{2_1}, r_{2_1} \rangle, \langle \alpha_{2_2}, r_{2_2} \rangle \dots, \langle \alpha_{m_1}, r_{m_1} \rangle, \langle \alpha_{m_2}, r_{m_2} \rangle, \langle \alpha_{m+1}, r_{m+1} \rangle, \dots, \langle \alpha_n, r_n \rangle\} \in BC(G'_1)$$

y sincronizando en la bolsa, se tiene que:

$$\{\langle \alpha_1, s_1 \rangle, \langle \alpha_2, r_2 \rangle, \dots, \langle \alpha_n, r_n \rangle\} \in BC(G'_1 \text{ sy } a)$$

con lo cual, tomando $G' = G'_1 \text{ sy } a$ concluimos que G' es operativa, $G' \equiv G$ y $\{\langle \alpha_1, s_1 \rangle, \langle \alpha_2, r_2 \rangle, \dots, \langle \alpha_n, r_n \rangle\} \in BC(G')$.

- (b) Si $\langle \alpha_1, r_1 \rangle$ proviene de sincronizar en a , entonces $\langle \alpha_1, s_1 \rangle$ también (debido a la restricción sintáctica y a la definición de *Conflict*). Supongamos que son de la forma:

$$\begin{aligned} \langle \alpha_1, r_1 \rangle &= \langle \alpha_{1_1}, r_{1_1} \rangle \oplus_a \langle \alpha_{1_2}, r_{1_2} \rangle \\ \langle \alpha_1, s_1 \rangle &= \langle \alpha_{1_1}, s_{1_1} \rangle \oplus_a \langle \alpha_{1_2}, s_{1_2} \rangle \end{aligned}$$

Y al igual que en el subcaso anterior podemos suponer que $\langle \alpha_i, r_i \rangle$, con $i = 2, \dots, m$ ($m \leq n$) vienen también de sincronizar, siendo de la forma:

$$\langle \alpha_i, r_i \rangle = \langle \alpha_{i_1}, r_{i_1} \rangle \oplus_a \langle \alpha_{i_2}, r_{i_2} \rangle \quad i = 2, \dots, m$$

De ahí:

$$\{\langle \alpha_{1_1}, r_{1_1} \rangle, \langle \alpha_{1_2}, r_{1_2} \rangle, \dots, \langle \alpha_{m_1}, r_{m_1} \rangle, \langle \alpha_{m_2}, r_{m_2} \rangle, \langle \alpha_{m+1}, r_{m+1} \rangle, \dots, \langle \alpha_n, r_n \rangle\} \in BC(G_1)$$

Por definición de *Conflict* sabemos que $\langle \alpha_{1_1}, r_{1_1} \rangle$ y $\langle \alpha_{1_1}, s_{1_1} \rangle$ están en el mismo conjunto de conflicto, y análogamente $\langle \alpha_{1_2}, r_{1_2} \rangle$ y $\langle \alpha_{1_2}, s_{1_2} \rangle$. Obsérvese que, de nuevo, esto es una consecuencia de la restricción sintáctica introducida. Entonces aplicando dos veces la hipótesis de inducción, obtendremos una s-expresión operativa $G'_1 \equiv G_1$, tal que:

$$\{\langle \alpha_{1_1}, s_{1_1} \rangle, \langle \alpha_{1_2}, s_{1_2} \rangle, \langle \alpha_{2_1}, r_{2_1} \rangle, \langle \alpha_{2_2}, r_{2_2} \rangle, \dots, \langle \alpha_{m_1}, r_{m_1} \rangle, \langle \alpha_{m_2}, r_{m_2} \rangle, \langle \alpha_{m+1}, r_{m+1} \rangle, \dots, \langle \alpha_n, r_n \rangle\} \in BC(G'_1)$$

Sincronizando de nuevo en a , y considerando $G' = G'_1$ *sy a*, donde G' es operativa y $G' \equiv G$, se tiene que:

$$\{\langle \alpha_1, s_1 \rangle, \langle \alpha_2, r_2 \rangle, \dots, \langle \alpha_n, r_n \rangle\} \in BC(G')$$

□

Lema 6.2 Sean E_1 y E_2 dos s-expresiones estáticas regulares con $E_1 \sim E_2$, $H_1, H'_2 \in OpReDynExpr$, tales que $H_1 \in [\overline{E_1}]$, $H_2 \in [\overline{E_2}]$, $\phi([H_1]_{\equiv}) = [H'_2]_{\equiv}$, y $\langle \alpha, r \rangle_n$ ejecutable desde H_1 . Entonces, existe algún $\langle \alpha, s \rangle_{n'}$ ejecutable desde $H_2 \equiv H'_2$, tal que:

$$cr(H_1, \langle \alpha, r \rangle_n) = cr(H_2, \langle \alpha, s \rangle_{n'})$$

Demostración:

Consideremos:

$$Conflict(H_1, \langle \alpha, r \rangle_n) = C, h_{[H_1]_{\equiv}, \alpha}(C) = C', \phi([H_1]_{\equiv}) = [H'_2]_{\equiv}, \phi([J_1]_{\equiv}) = [J_2]_{\equiv}$$

entonces si $([H_1]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_1]_{\equiv}) \in A_a^{E_1}$ se tiene que existen n', s tales que

$$([H'_2]_{\equiv}, \langle \alpha, s \rangle_{n',C'}, [J_2]_{\equiv}) \in A_a^{E_2}, \text{ con } rc(C) = rc(C') \text{ (por definición de } h_{[H_1]_{\equiv}, \alpha}),$$

por lo que podremos concluir que existe una operativa $H_2 \equiv H'_2$, desde la cual $\langle \alpha, s \rangle_{n'}$ es ejecutable, tal que $Conflict(H_2, \langle \alpha, s \rangle_{n'}) = C'$ y de ahí:

$$cr(H_1, \langle \alpha, r \rangle_n) = cr(H_2, \langle \alpha, s \rangle_{n'})$$

□

Recordemos que las transiciones fantasmas se han introducido sólo para pares de multiacciones estocásticas regulares, en lugar de hacerlo para cualquier bolsa posible. En la proposición siguiente se pone de manifiesto que efectivamente esta información es suficiente para identificar todo el paralelismo potencial en una s-expresión.

Proposición 6.2 Sean E_1, E_2 dos s-expresiones estáticas regulares, tales que $E_1 \sim E_2$, y $H_1 \in OpReDynExpr$ verificando $H_1 \in [\overline{E_1}]$. Entonces, para cualquier $\gamma_1 = \{\langle \alpha_i, r_i \rangle_{m_i}\}_{i=1}^n \in BC(H_1)$, con $n \geq 2$, existe una s-expresión dinámica operativa regular $H_2 \in \phi([H_1]_{\equiv})$, con alguna bolsa $\gamma_2 = \{\langle \alpha_i, r'_i \rangle_{m'_i}\}_{i=1}^n \in BC(H_2)$.

Además, si $([H_1]_{\equiv}, \langle \alpha_i, r_i \rangle_{m_i, C_i}, [H_{1,i}]_{\equiv}) \in A_a^{E_1}$, para $i = 1, \dots, n$, entonces

$$([H_2]_{\equiv}, \langle \alpha_i, r'_i \rangle_{m'_i, C'_i}, \phi([H_{1,i}]_{\equiv})) \in A_a^{E_2}, \text{ con } h_{[H_1]_{\equiv}, \alpha}(C_i) = C'_i \text{ y } rc(C_i) = rc(C'_i),$$

siendo $h_{[H_1]_{\equiv}, \alpha}$ la biyección que nos proporciona la equivalencia \sim .

Demostración:

Por inducción sobre n .

- **Caso base:** ($n = 2$).

A partir de $\{\langle \alpha_1, r_1 \rangle_{m_1}, \langle \alpha_2, r_2 \rangle_{m_2}\} \in BC(H_1)$ podemos deducir que existe una transición fantasma $([H_1]_{\equiv}, \langle \alpha_1, r_1 \rangle_{m_1} \parallel \langle \alpha_2, r_2 \rangle_{m_2}, [H'_1]_{\equiv})$ en $nts(\overline{E_1})$, con $([H_1]_{\equiv}, \langle \alpha_i, r_i \rangle_{m_i, C_i}, [H_{1,i}]_{\equiv}) \in A_a^{E_1}$ para $i = 1, 2$.

Entonces, aplicando la definición 6.7, obtenemos la correspondiente transición fantasma en $nts(\overline{E_2})$:

$$(\phi([H_1]_{\equiv}), \langle \alpha_1, r'_1 \rangle_{m'_1} \parallel \langle \alpha_2, r'_2 \rangle_{m'_2}, \phi([H'_1]_{\equiv}))$$

con $(\phi([H_1]_{\equiv}), \langle \alpha_i, r'_i \rangle_{m'_i, C'_i}, \phi([H_{1,i}]_{\equiv})) \in A_a^{E_2}$, con $C'_i = h_{[H_1]_{\equiv}, \alpha_i}(C_i)$, y en consecuencia $rc(C_i) = rc(C'_i)$, para $i = 1, 2$.

Con lo que $\gamma_2 = \{\langle \alpha_1, r'_1 \rangle_{m'_1}, \langle \alpha_2, r'_2 \rangle_{m'_2}\} \in BC(H_2)$, para algún $H_2 \in \phi([H_1]_{\equiv})$.

- **Caso general:** ($n > 2$)

Partimos de $\gamma_1 = \gamma_{1,1} \cup \{\langle \alpha_n, r_n \rangle_{m_n}\}$ donde $\gamma_{1,1} = \{\langle \alpha_i, r_i \rangle_{m_i}\}_{i=1}^{n-1}$.

Por la hipótesis de inducción, podemos encontrar una s-expresión dinámica operativa $H_2 \in \phi([H_1]_{\equiv})$ y $\gamma_{1,2} = \{\langle \alpha_i, r'_i \rangle_{m'_i}\}_{i=1}^{n-1} \in BC(H_2)$, tal que

$$\begin{aligned} &\text{si } ([H_1]_{\equiv}, \langle \alpha_i, r_i \rangle_{m_i, C_i}, [H_{1,i}]_{\equiv}) \in A_a^{E_1} \text{ entonces} \\ &([H_2]_{\equiv}, \langle \alpha_i, r'_i \rangle_{m'_i, C'_i}, \phi([H_{1,i}]_{\equiv})) \in A_a^{E_2}, \text{ con } h_{[H_1]_{\equiv}, \alpha_i}(C_i) = C'_i \\ &\text{y } rc(C_i) = rc(C'_i), \text{ para } i = 1, \dots, n-1 \end{aligned}$$

Entonces, considerando que $\{\langle \alpha_i, r_i \rangle_{m_i}, \langle \alpha_n, r_n \rangle_{m_n}\} \in BC(H_1)$ con $i = 1, \dots, n-1$, y utilizando de nuevo la definición 6.7, concluimos que existe $\{\langle \alpha_i, r'_i \rangle_{p_i}, \langle \alpha_n, r'_{n_i} \rangle_{p_{n_i}}\} \in BC(H_{2i})$ donde $H_{2i} \equiv H_2$ y además $\langle \alpha_i, r'_i \rangle_{p_i} \in C'_i$.

Además, sabemos que existe un arco de la forma

$$([H_2]_{\equiv}, \langle \alpha_n, r'_n \rangle_{m'_n, C'_n}, \phi([H_{1,n}]_{\equiv})) \in A_a^{E_2}$$

con $h_{[H_1]_{\equiv}, \alpha_n}(C_n) = C'_n$. De hecho, sabemos que $\langle \alpha_n, r'_{n_i} \rangle_{p_{n_i}} \in C'_n$. De esta forma, por el lema 6.1, obtenemos que $\forall i \in \{1, \dots, n\}$, $\exists H'_{2i} \equiv H_{2i}$ tal que

$$\{\langle \alpha_i, r'_i \rangle_{m'_i}, \langle \alpha_n, r'_n \rangle_{m'_n}\} \in BC(H'_{2i})$$

y de ahí teniendo en cuenta que ya teníamos:

$$\{\langle \alpha_1, r'_1 \rangle_{m'_1}, \langle \alpha_2, r'_2 \rangle_{m'_2}, \dots, \langle \alpha_{n-1}, r'_{n-1} \rangle_{m'_{n-1}}\} \in BC(H_2)$$

como $H_2 \equiv H_{2i}$, y debido a la restricción sintáctica impuesta la única posibilidad para que eso ocurra es que dentro de H_2 exista un paralelo entre la expresión que permite ejecutar las $n-1$ multiacciones estocásticas de $\{\langle \alpha_1, r'_1 \rangle_{m'_1}, \langle \alpha_2, r'_2 \rangle_{m'_2}, \dots, \langle \alpha_{n-1}, r'_{n-1} \rangle_{m'_{n-1}}\}$ y la que nos permite ejecutar $\langle \alpha_n, r'_n \rangle_{m'_n}$. Por tanto,

$$\{\langle \alpha_1, r'_1 \rangle_{m'_1}, \langle \alpha_2, r'_2 \rangle_{m'_2}, \dots, \langle \alpha_{n-1}, r'_{n-1} \rangle_{m'_{n-1}}, \langle \alpha_n, r'_n \rangle_{m'_n}\} \in BC(H'_2)$$

con $H'_2 \equiv H_2$.

□

Finalmente, estamos ya en condiciones de demostrar que la relación de equivalencia \sim es realmente una congruencia.

Teorema 6.1 Sean E_1, E_2, E, E' s-expresiones estáticas regulares tales que

$$E_1 \sim E_2$$

Entonces, tenemos:

- (i) $E_1; E \sim E_2; E$ y $E; E_1 \sim E; E_2$
- (ii) $E_1 \parallel E \sim E_2 \parallel E$ y $E \parallel E_1 \sim E \parallel E_2$
- (iii) $E_1 \square E \sim E_2 \square E$ y $E \square E_1 \sim E \square E_2$
- (iv) $E_1[f] \sim E_2[f]$, para cualquier función de renombramiento biyectiva f
- (v) $E_1 sy a \sim E_2 sy a$, para todo $a \in \mathcal{A}$
- (vi) $E_1 rs a \sim E_2 rs a$, para todo $a \in \mathcal{A}$
- (vii) $[E_1 * E' * E] \sim [E_2 * E' * E]$
- (viii) $[E' * E_1 * E] \sim [E' * E_2 * E]$
- (ix) $[E' * E * E_1] \sim [E' * E * E_2]$

Demostración:

Consideremos:

$$\begin{aligned}
 nts(\overline{E}) &= (V^E \cup \{[\underline{E}]_{\equiv}\}, A_a^E \cup A_{sr}^E \cup A_g^E, v_0^E), \\
 nts(\overline{E}') &= (V^{E'} \cup \{[\underline{E}']_{\equiv}\}, A_a^{E'} \cup A_{sr}^{E'} \cup A_g^{E'}, v_0^{E'}), \\
 nts(\overline{E}_1) &= (V^{E_1} \cup \{[\underline{E}_1]_{\equiv}\}, A_a^{E_1} \cup A_{sr}^{E_1} \cup A_g^{E_1}, v_0^{E_1}), \\
 nts(\overline{E}_2) &= (V^{E_2} \cup \{[\underline{E}_2]_{\equiv}\}, A_a^{E_2} \cup A_{sr}^{E_2} \cup A_g^{E_2}, v_0^{E_2})
 \end{aligned}$$

ϕ la función biyectiva entre los nodos obtenida a partir de $E_1 \sim E_2$, esto es:

$$\phi : V^{E_1} \cup \{[\underline{E}_1]_{\equiv}\} \rightarrow V^{E_2} \cup \{[\underline{E}_2]_{\equiv}\}$$

y para cada par (v, α) , con $v \in V^{E_1}$, tal que $\mathcal{C}(v, \alpha) \neq \emptyset$ tenemos la biyección $h_{v, \alpha}$:

$$h_{v, \alpha} : \mathcal{C}(v, \alpha) \rightarrow \mathcal{C}(\phi(v), \alpha), \quad \text{que cumple} \quad rc(C_j) = rc(h_{v, \alpha}(C_j)), \quad \forall C_j \in \mathcal{C}(v, \alpha)$$

(i) Consideremos

$$F_1 = E_1; E \quad \text{y} \quad F_2 = E_2; E$$

$nts(\overline{F}_1)$ y $nts(\overline{F}_2)$ se obtienen como sigue:

$$nts(\overline{F}_1) = (V^{F_1} \cup \{[\underline{F}_1]_{\equiv}\}, A_a^{F_1} \cup A_{sr}^{F_1} \cup A_g^{F_1}, v_0^{F_1})$$

donde ²

$$V^{F_1} = \{ [H_1; E]_{\equiv} \mid H_1 \in \overline{[E_1]} \} \cup \{ [E_1; G]_{\equiv} \mid G \in \overline{[E]} \}$$

$$A_a^{F_1} = \{ ([H_1; E]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_1; E]_{\equiv}) \mid ([H_1]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_1]_{\equiv}) \in A_a^{E_1} \} \cup \\ \{ ([E_1; G]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [E_1; J]_{\equiv}) \mid ([G]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J]_{\equiv}) \in A_a^E \}$$

$$A_{sr}^{F_1} = \{ (\overline{[F_1]}_{\equiv}, \langle skip, 0 \rangle, [F_1]_{\equiv}), ([F_1]_{\equiv}, \langle redo, \infty \rangle, \overline{[F_1]}_{\equiv}) \}$$

$$A_g^{F_1} = \{ ([H_1; E]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1; E]_{\equiv}) \mid \\ ([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1]_{\equiv}) \in A_g^{E_1} \} \cup \\ \{ ([E_1; G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [E_1; J]_{\equiv}) \mid \\ ([G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J]_{\equiv}) \in A_g^E \}$$

$$v_0^{F_1} = \overline{[F_1]}_{\equiv}$$

Podemos observar que en la definición de V^{F_1} tenemos que:

- $[E_1; E]_{\equiv} = [E_1; \overline{E}]_{\equiv}$ ya que $\underline{E_1}; E \equiv E_1; \overline{E}$
- $\overline{[F_1]}_{\equiv} = \overline{[E_1; E]}_{\equiv}$ ya que $\overline{E_1}; E \equiv \overline{E_1}; \overline{E}$
- $[F_1]_{\equiv} = [E_1; \underline{E}]_{\equiv}$ ya que $E_1; \underline{E} \equiv \underline{E_1}; E$

Análogamente,

$$nts(\overline{[F_2]}) = (V^{F_2} \cup \{ [F_2]_{\equiv} \}, A_a^{F_2} \cup A_{sr}^{F_2} \cup A_g^{F_2}, v_0^{F_2})$$

donde

$$V^{F_2} = \{ [H_2; E]_{\equiv} \mid H_2 \in \overline{[E_2]} \} \cup \{ [E_2; G]_{\equiv} \mid G \in \overline{[E]} \}$$

$$A_a^{F_2} = \{ ([H_2; E]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_2; E]_{\equiv}) \mid ([H_2]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_2]_{\equiv}) \in A_a^{E_2} \} \cup \\ \{ ([E_2; G]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [E_2; J]_{\equiv}) \mid ([G]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J]_{\equiv}) \in A_a^E \}$$

$$A_{sr}^{F_2} = \{ (\overline{[F_2]}_{\equiv}, \langle skip, 0 \rangle, [F_2]_{\equiv}), ([F_2]_{\equiv}, \langle redo, \infty \rangle, \overline{[F_2]}_{\equiv}) \}$$

²Para no sobrecargar la notación, vamos a suponer a lo largo de toda la demostración que las multiacciones estocásticas se numeran independientemente en cada subexpresión, y que esa numeración se conserva en las expresiones donde estén incluidas.

$$A_g^{F_2} = \{ ([H_2; E]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_2; E]_{\equiv}) \mid \\ ([H_2]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_2]_{\equiv}) \in A_g^{E_2} \} \cup \\ \{ ([E_2; G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [E_2; J]_{\equiv}) \mid \\ ([G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J]_{\equiv}) \in A_g^E \}$$

$$v_0^{F_2} = [\overline{F_2}]_{\equiv}$$

$$\text{con } [E_2; E]_{\equiv} = [E_2; \overline{E}]_{\equiv}, \quad [\overline{F_2}]_{\equiv} = [\overline{E_2}; E]_{\equiv} \text{ y } [F_2]_{\equiv} = [E_2; \underline{E}]_{\equiv}.$$

Entonces, definimos $\varphi : V^{F_1} \cup \{[F_1]_{\equiv}\} \rightarrow V^{F_2} \cup \{[F_2]_{\equiv}\}$, de la forma siguiente:

- Si $[H_1]_{\equiv} \in V^{E_1}$ y $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$, entonces $\varphi([H_1; E]_{\equiv}) = [H_2; E]_{\equiv}$
- $\varphi([E_1; G]_{\equiv}) = [E_2; G]_{\equiv}$, para cada $G \in [\overline{E}]$
- $\varphi([F_1]_{\equiv}) = [F_2]_{\equiv}$

Es inmediato comprobar que φ está bien definida, y que además es una biyección que verifica $\varphi([\overline{F_1}]_{\equiv}) = [\overline{F_2}]_{\equiv}$.

Asimismo para cada $v \in V^{F_1}$ y cada α tal que $\mathcal{C}(v, \alpha) \neq \emptyset$, definiremos una biyección $g_{v, \alpha} : \mathcal{C}(v, \alpha) \rightarrow \mathcal{C}(\varphi(v), \alpha)$, lo cual haremos a lo largo de la siguiente distinción de casos.

- Consideremos un arco cualquiera e de $A_a^{F_1}$. Podemos encontrarnos en uno de los casos siguientes:

- $e = ([H_1; E]_{\equiv}, \langle \alpha, r \rangle_{n, C}, [J_1; E]_{\equiv})$, con $([H_1]_{\equiv}, \langle \alpha, r \rangle_{n, C}, [J_1]_{\equiv}) \in A_a^{E_1}$, ya que los conflictos de $\langle \alpha, r \rangle_n$ computados desde una operativa derivada de $\overline{E_1}$, coinciden con los correspondientes a dicha multiacción estocástica, pero computados desde una operativa derivada de $\overline{F_1}$.

Tomando $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$, $\phi([J_1]_{\equiv}) = [J_2]_{\equiv}$ y $h_{[H_1]_{\equiv}, \alpha}(C) = C'$, tenemos que

$$\varphi([H_1; E]_{\equiv}) = [H_2; E]_{\equiv} \quad \text{y} \quad \varphi([J_1; E]_{\equiv}) = [J_2; E]_{\equiv}$$

Definimos entonces $g_{[H_1; E]_{\equiv}, \alpha}(C) = C'$, teniéndose por tanto que $rc(C) = rc(C')$.

De esta forma, como $E_1 \sim E_2$, $\exists n', s$ tal que

$$([H_2]_{\equiv}, \langle \alpha, s \rangle_{n', C'}, [J_2]_{\equiv}) \in A_a^{E_2}$$

y así:

$$([H_2; E]_{\equiv}, \langle \alpha, s \rangle_{n', C'}, [J_2; E]_{\equiv}) \in A_a^{F_2}$$

cumpléndose que

$$\begin{aligned} r([H_1; E]_{\equiv}, \alpha, C, [J_1; E]_{\equiv}) &= r([H_1]_{\equiv}, \alpha, C, [J_1]_{\equiv}) = \\ r([H_2]_{\equiv}, \alpha, C', [J_2]_{\equiv}) &= r([H_2; E]_{\equiv}, \alpha, C', [J_2; E]_{\equiv}) \end{aligned}$$

- $e = ([E_1; G]_{\equiv}, \langle \alpha, r \rangle_{n, C}, [E_1; J]_{\equiv})$ con $([G]_{\equiv}, \langle \alpha, r \rangle_{n, C}, [J]_{\equiv}) \in A_a^E$.

En este caso tenemos que:

$$\varphi([E_1; G]_{\equiv}) = [E_2; G]_{\equiv}, \quad \varphi([E_1; J]_{\equiv}) = [E_2; J]_{\equiv}$$

y definimos $g_{[E_1; G]_{\equiv}, \alpha}(C) = C$.

De esta forma

$$([E_2; G]_{\equiv}, \langle \alpha, r \rangle_{n, C}, [E_2; J]_{\equiv}) \in A_a^{F_2}$$

con

$$\begin{aligned} r([E_1; G]_{\equiv}, \alpha, C, [E_1; J]_{\equiv}) &= r([G]_{\equiv}, \alpha, C, [J]_{\equiv}) = \\ r([E_2; G]_{\equiv}, \alpha, C, [E_2; J]_{\equiv}) & \end{aligned}$$

- Consideremos ahora un arco cualquiera e de $A_g^{F_1}$. Distinguiamos de nuevo dos casos:

- $e = ([H_1; E]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1; E]_{\equiv})$ con

$$([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1]_{\equiv}) \in A_g^{E_1}.$$

Tomando $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$, $\phi([J_1]_{\equiv}) = [J_2]_{\equiv}$, $h_{[H_1]_{\equiv}, \alpha}(C_1) = C'_1$ y $h_{[H_1]_{\equiv}, \beta}(C_2) = C'_2$ tenemos que

$$\varphi([H_1; E]_{\equiv}) = [H_2; E]_{\equiv} \quad \text{y} \quad \varphi([J_1; E]_{\equiv}) = [J_2; E]_{\equiv}$$

y definimos $g_{[H_1; E]_{\equiv}, \alpha}(C_1) = C'_1$ y $g_{[H_1; E]_{\equiv}, \beta}(C_2) = C'_2$

Además, si $([H_1; E]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1; E]_{\equiv}) \in A_g^{F_1}$ con $([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1]_{\equiv}) \in A_g^{E_1}$, entonces, como $E_1 \sim E_2$ llegamos a:

1. $\exists([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1, C_1}, [H'_1]_{\equiv}) \in A_a^{E_1}$
2. $\exists([H_1]_{\equiv}, \langle \beta, s \rangle_{n_2, C_2}, [H''_1]_{\equiv}) \in A_a^{E_1}$
3. $\exists r', s', n'_1, n'_2$ tal que $([H_2]_{\equiv}, \langle \alpha, r' \rangle_{n'_1} \parallel \langle \beta, s' \rangle_{n'_2}, [J_2]_{\equiv}) \in A_g^{E_2}$,
con:
 $([H_2]_{\equiv}, \langle \alpha, r' \rangle_{n'_1, C'_1}, \phi([H'_1]_{\equiv})) \in A_a^{E_2}$ y
 $([H_2]_{\equiv}, \langle \beta, s' \rangle_{n'_2, C'_2}, \phi([H''_1]_{\equiv})) \in A_a^{E_2}$.

Con lo que concluimos que:

1. $\exists([H_1; E]_{\equiv}, \langle \alpha, r \rangle_{n_1, C_1}, [H'_1; E]_{\equiv}) \in A_a^{F_1}$
 2. $\exists([H_1; E]_{\equiv}, \langle \beta, s \rangle_{n_2, C_2}, [H''_1; E]_{\equiv}) \in A_a^{F_1}$
 3. $\exists r', s', n'_1, n'_2$ tal que $([H_2; E]_{\equiv}, \langle \alpha, r' \rangle_{n'_1} \parallel \langle \beta, s' \rangle_{n'_2}, [J_2; E]_{\equiv}) \in A_g^{F_2}$, con:
 $([H_2; E]_{\equiv}, \langle \alpha, r' \rangle_{n'_1, C'_1}, \varphi([H'_1; E]_{\equiv})) \in A_a^{F_2}$, y
 $([H_2; E]_{\equiv}, \langle \beta, s' \rangle_{n'_2, C'_2}, \varphi([H''_1; E]_{\equiv})) \in A_a^{F_2}$.
- $e = ([E_1; G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [E_1; J]_{\equiv})$
con $([G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J]_{\equiv}) \in A_g^E$.

Este caso es simétrico al anterior.

La demostración para los arcos de $A_a^{F_2}$ y $A_g^{F_2}$ es similar ³.

Asimismo, la demostración para $E; E_1 \sim E; E_2$ es análoga.

(ii) Consideremos

$$F_1 = E_1 \parallel E \quad \text{y} \quad F_2 = E_2 \parallel E$$

En este caso tenemos que $nts(\overline{F_1})$ y $nts(\overline{F_2})$ son de la forma:

$$nts(\overline{F_1}) = (V^{F_1} \cup \{[F_1]_{\equiv}\}, A_a^{F_1} \cup A_{sr}^{F_1} \cup A_g^{F_1}, v_0^{F_1})$$

³Esto mismo ocurre para el resto de operadores, de modo que en los restantes casos sólo haremos las demostraciones para los arcos de $A_a^{F_1}$ y $A_g^{F_1}$.

donde:

$$V^{F_1} = \{ [H_1 \| G]_{\equiv} \mid H_1 \in [\overline{E_1}] \text{ y } G \in [\overline{E}] \}$$

$$A_a^{F_1} = \{ ([H_1 \| G]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_1 \| G]_{\equiv}) \mid ([H_1]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_1]_{\equiv}) \in A_a^{E_1} \} \cup \\ \{ ([H_1 \| G]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [H_1 \| J]_{\equiv}) \mid ([G]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J]_{\equiv}) \in A_a^E \}$$

$$A_{sr}^{F_1} = \{ ([\overline{F_1}]_{\equiv}, \langle skip, 0 \rangle, [F_1]_{\equiv}), ([F_1]_{\equiv}, \langle redo, \infty \rangle, [\overline{F_1}]_{\equiv}) \}$$

$$A_g^{F_1} = \{ ([H_1 \| G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1 \| G]_{\equiv}) \mid \\ ([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1]_{\equiv}) \in A_g^{E_1} \} \cup \\ \{ ([H_1 \| G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [H_1 \| J]_{\equiv}) \mid \\ ([G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J]_{\equiv}) \in A_g^E \} \cup \\ \{ ([H_1 \| G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1 \| J]_{\equiv}) \mid \\ ([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1, C_1}, [J_1]_{\equiv}) \in A_a^{E_1} \text{ y } ([G]_{\equiv}, \langle \beta, s \rangle_{n_2, C_2}, [J]_{\equiv}) \in A_a^E \}$$

$$v_0^{F_1} = [\overline{F_1}]_{\equiv}$$

donde $[\overline{F_1}]_{\equiv} = [\overline{E_1} \| \overline{E}]_{\equiv}$, y $[F_1]_{\equiv} = [E_1 \| E]_{\equiv}$.

$$nts(\overline{F_2}) = (V^{F_2} \cup \{ [F_2]_{\equiv} \}, A_a^{F_2} \cup A_{sr}^{F_2} \cup A_g^{F_2}, v_0^{F_2})$$

donde:

$$V^{F_2} = \{ [H_2 \| G]_{\equiv} \mid H_2 \in [\overline{E_2}] \text{ y } G \in [\overline{E}] \}$$

$$A_a^{F_2} = \{ ([H_2 \| G]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_2 \| G]_{\equiv}) \mid ([H_2]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_2]_{\equiv}) \in A_a^{E_2} \} \cup \\ \{ ([H_2 \| G]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [H_2 \| J]_{\equiv}) \mid ([G]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J]_{\equiv}) \in A_a^E \}$$

$$A_{sr}^{F_2} = \{ ([\overline{F_2}]_{\equiv}, \langle skip, 0 \rangle, [F_2]_{\equiv}), ([F_2]_{\equiv}, \langle redo, \infty \rangle, [\overline{F_2}]_{\equiv}) \}$$

$$A_g^{F_2} = \{ ([H_2 \| G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_2 \| G]_{\equiv}) \mid \\ ([H_2]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_2]_{\equiv}) \in A_g^{E_2} \} \cup \\ \{ ([H_2 \| G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [H_2 \| J]_{\equiv}) \mid \\ ([G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J]_{\equiv}) \in A_g^E \} \cup \\ \{ ([H_2 \| G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_2 \| J]_{\equiv}) \mid \\ ([H_2]_{\equiv}, \langle \alpha, r \rangle_{n_1, C_1}, [J_2]_{\equiv}) \in A_a^{E_2} \text{ y } ([G]_{\equiv}, \langle \beta, s \rangle_{n_2, C_2}, [J]_{\equiv}) \in A_a^E \}$$

$$v_0^{F_2} = [\overline{F_2}]_{\equiv}$$

donde $[\overline{F_2}]_{\equiv} = [\overline{E_2} \parallel \overline{E}]_{\equiv}$, y $[F_2]_{\equiv} = [E_2 \parallel E]_{\equiv}$.

Entonces definimos

$$\varphi : V^{F_1} \cup \{[F_1]_{\equiv}\} \rightarrow V^{F_2} \cup \{[F_2]_{\equiv}\}$$

de la forma siguiente:

- $\forall [H_1]_{\equiv} \in V^{E_1}$, $\varphi([H_1 \parallel G]_{\equiv}) = [H_2 \parallel G]_{\equiv}$, donde $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$.
- $\varphi([F_1]_{\equiv}) = [F_2]_{\equiv}$.

Fácilmente se comprueba que φ está bien definida, es una biyección, y $\varphi([\overline{F_1}]_{\equiv}) = [\overline{F_2}]_{\equiv}$.

Asimismo, para cada $v \in V^{F_1}$ y cada α tal que $\mathcal{C}(v, \alpha) \neq \emptyset$, iremos definiendo una biyección $g_{v, \alpha} : \mathcal{C}(v, \alpha) \rightarrow \mathcal{C}(\varphi(v), \alpha)$, en los casos siguientes.

- Tenemos otra vez dos posibles casos para los arcos de $A_a^{F_1}$, y en cada uno de ellos podemos repetir un razonamiento similar al efectuado en el caso (i):

- $e = ([H_1 \parallel G]_{\equiv}, \langle \alpha, r \rangle_{n, C}, [J_1 \parallel G]_{\equiv})$, con $([H_1]_{\equiv}, \langle \alpha, r \rangle_{n, C}, [J_1]_{\equiv}) \in A_a^{E_1}$, ya que los conflictos de $\langle \alpha, r \rangle_n$ tomados desde una operativa derivada de $\overline{E_1}$, coinciden con los obtenidos para la misma multiacción pero tomados desde una operativa derivada de $\overline{F_1}$.

Tomando $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$, $\phi([J_1]_{\equiv}) = [J_2]_{\equiv}$, y $h_{[H_1]_{\equiv}, \alpha}(C) = C'$ tenemos que

$$\varphi([H_1 \parallel G]_{\equiv}) = [H_2 \parallel G]_{\equiv} \quad \text{y} \quad \varphi([J_1 \parallel G]_{\equiv}) = [J_2 \parallel G]_{\equiv}$$

y definimos $g_{[H_1 \parallel G]_{\equiv}, \alpha}(C) = C'$, teniéndose por tanto que $rc(C) = rc(C')$.

De esta forma, como $E_1 \sim E_2$, $\exists n', s$ tal que

$$([H_2]_{\equiv}, \langle \alpha, s \rangle_{n', C'}, [J_2]_{\equiv}) \in A_a^{E_2}$$

y así:

$$([H_2 \parallel G]_{\equiv}, \langle \alpha, s \rangle_{n', C'}, [J_2 \parallel G]_{\equiv}) \in A_a^{F_2}$$

cumpléndose que

$$\begin{aligned} r([H_1 \parallel G]_{\equiv}, \alpha, C, [J_1 \parallel G]_{\equiv}) &= r([H_1]_{\equiv}, \alpha, C, [J_1]_{\equiv}) = \\ r([H_2]_{\equiv}, \alpha, C', [J_2]_{\equiv}) &= r([H_2 \parallel G]_{\equiv}, \alpha, C', [J_2 \parallel G]_{\equiv}) \end{aligned}$$

- $e = ([H_1 \parallel G]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [H_1 \parallel J]_{\equiv})$ con $([G]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J]_{\equiv}) \in A_a^E$.
En este caso tenemos que si $\phi([H_1]_{\equiv} = [H_2]_{\equiv})$, entonces :

$$\varphi([H_1 \parallel G]_{\equiv}) = [H_2; G]_{\equiv}, \quad \varphi([H_1 \parallel J]_{\equiv}) = [H_2 \parallel J]_{\equiv}$$

y definimos $g_{[H_1 \parallel G]_{\equiv}, \alpha}(C) = C$.

De esta forma

$$([H_2 \parallel G]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [H_2 \parallel J]_{\equiv}) \in A_a^{F_2}$$

con

$$\begin{aligned} r([H_1 \parallel G]_{\equiv}, \alpha, C, [H_1 \parallel J]_{\equiv}) &= r([G]_{\equiv}, \alpha, C, [J]_{\equiv}) = \\ r([H_2 \parallel G]_{\equiv}, \alpha, C, [H_2 \parallel J]_{\equiv}) \end{aligned}$$

- Para los arcos de $A_g^{F_1}$ tenemos ahora tres casos:

- $([H_1 \parallel G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1 \parallel G]_{\equiv})$,
con $([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1]_{\equiv}) \in A_g^{E_1}$.
- $([H_1 \parallel G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [H_1 \parallel J]_{\equiv})$,
con $([G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J]_{\equiv}) \in A_g^E$.
- $([H_1 \parallel G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1 \parallel J]_{\equiv})$,
con $([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1, C_1}, [J_1]_{\equiv}) \in A_a^{E_1}$ y $([G]_{\equiv}, \langle \beta, s \rangle_{n_2, C_2}, [J]_{\equiv}) \in A_a^E$.

Los dos primeros casos son análogos al caso (i).

Para el tercer caso, como $E_1 \sim E_2$ podemos encontrar algunos r', n'_1 , y considerando $h_{[H_1]_{\equiv}, \alpha}(C_1) = C'_1$, definimos $g_{[H_1 \parallel G]_{\equiv}, \alpha}(C_1) = C'_1$, tal que

$$([H_2]_{\equiv}, \langle \alpha, r' \rangle_{n'_1, C'_1}, [J_2]_{\equiv}) \in A_a^{E_2}, \text{ y así:}$$

$$([H_2 \parallel G]_{\equiv}, \langle \alpha, r' \rangle_{n'_1} \parallel \langle \beta, s \rangle_{n_2}, [J_2 \parallel J]_{\equiv}) \in A_g^{F_2}.$$

(iii) Consideremos

$$F_1 = E_1 \square E \quad \text{y} \quad F_2 = E_2 \square E$$

Entonces tenemos:

$$nts(\overline{F_1}) = (V^{F_1} \cup \{[\underline{F_1}]_{\equiv}\}, A_a^{F_1} \cup A_{sr}^{F_1} \cup A_g^{F_1}, v_0^{F_1})$$

donde:

$$V^{F_1} = \{[H_1 \square E]_{\equiv} \mid H_1 \in [\overline{E_1}]\} \cup \{[E_1 \square G]_{\equiv} \mid G \in [\overline{E}]\}$$

$$A_a^{F_1} = \{ ([H_1 \sqcap E]_{\equiv}, \langle \alpha, r \rangle_{n, C_1}, [J_1 \sqcap E]_{\equiv}) \mid ([H_1]_{\equiv}, \langle \alpha, r \rangle_{n, C_{11}}, [J_1]_{\equiv}) \in A_a^{E_1} \} \cup \\ \{ ([E_1 \sqcap G]_{\equiv}, \langle \alpha, r \rangle_{n, C_2}, [E_1 \sqcap J]_{\equiv}) \mid ([G]_{\equiv}, \langle \alpha, r \rangle_{n, C_{21}}, [J]_{\equiv}) \in A_a^E \}$$

siendo C_1 y C_2 de la forma siguiente:

- Si $H_1 \not\equiv \overline{E_1}$, entonces $C_1 = C_{11}$.
- Si $H_1 \equiv \overline{E_1}$, entonces:
 $C_1 = C_{11} \cup \{ \langle \alpha, r_j \rangle_{n_j} \mid ([\overline{E}]_{\equiv}, \langle \alpha, r_j \rangle_{n_j, C}, [K]_{\equiv}) \in A_a^E \}$
- Si $G \not\equiv \overline{E}$, entonces $C_2 = C_{21}$.
- Si $G \equiv \overline{E}$, entonces:
 $C_2 = C_{21} \cup \{ \langle \alpha, r_j \rangle_{n_j} \mid ([\overline{E_1}]_{\equiv}, \langle \alpha, r_j \rangle_{n_j, C}, [K_1]_{\equiv}) \in A_a^{E_1} \}$

$$A_{sr}^{F_1} = \{ ([\overline{F_1}]_{\equiv}, \langle skip, 0 \rangle, [F_1]_{\equiv}), ([F_1]_{\equiv}, \langle redo, \infty \rangle, [\overline{F_1}]_{\equiv}) \}$$

$$A_g^{F_1} = \{ ([H_1 \sqcap E]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1 \sqcap E]_{\equiv}) \mid \\ ([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1]_{\equiv}) \in A_g^{E_1} \} \cup \\ \{ ([E_1 \sqcap G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [E_1 \sqcap J]_{\equiv}) \mid \\ ([G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J]_{\equiv}) \in A_g^E \}$$

$$v_0^{F_1} = [\overline{F_1}]_{\equiv}$$

donde en V^{F_1} tenemos:

- $[\overline{F_1}]_{\equiv} = [\overline{E_1} \sqcap E]_{\equiv} = [E_1 \sqcap \overline{E}]_{\equiv}$
- $[F_1]_{\equiv} = [E_1 \sqcap E]_{\equiv} = [E_1 \sqcap \underline{E}]_{\equiv}$

$$nts(\overline{F_2}) = (V^{F_2} \cup \{ [F_2]_{\equiv} \}, A_a^{F_2} \cup A_{sr}^{F_2} \cup A_g^{F_2}, v_0^{F_2})$$

donde:

$$V^{F_2} = \{ [H_2 \sqcap E]_{\equiv} \mid H_2 \in [\overline{E_2}] \} \cup \{ [E_2 \sqcap G]_{\equiv} \mid G \in [\overline{E}] \}$$

$$A_a^{F_2} = \{ ([H_2 \sqcap E]_{\equiv}, \langle \alpha, r \rangle_{n, C_1}, [J_2 \sqcap E]_{\equiv}) \mid ([H_2]_{\equiv}, \langle \alpha, r \rangle_{n, C_{11}}, [J_2]_{\equiv}) \in A_a^{E_2} \} \cup \\ \{ ([E_2 \sqcap G]_{\equiv}, \langle \alpha, r \rangle_{n, C_2}, [E_2 \sqcap J]_{\equiv}) \mid ([G]_{\equiv}, \langle \alpha, r \rangle_{n, C_{21}}, [J]_{\equiv}) \in A_a^E \}$$

siendo C_1 y C_2 de la forma:

- Si $H_2 \not\equiv \overline{E_2}$, entonces $C_1 = C_{11}$.
- Si $H_2 \equiv \overline{E_2}$, entonces:

$$C_1 = C_{11} \cup \{ \langle \alpha, r_j \rangle_{n_j} \mid ([\overline{E}]_{\equiv}, \langle \alpha, r_j \rangle_{n_j, C}, [K]_{\equiv}) \in A_a^E \}$$
- Si $G \not\equiv \overline{E}$, entonces $C_2 = C_{21}$.
- Si $G \equiv \overline{E}$, entonces:

$$C_2 = C_{21} \cup \{ \langle \alpha, r_j \rangle_{n_j} \mid ([\overline{E_2}]_{\equiv}, \langle \alpha, r_j \rangle_{n_j, C}, [K_2]_{\equiv}) \in A_a^{E_2} \}$$

$$A_{sr}^{F_2} = \{ ([\overline{F_2}]_{\equiv}, \langle skip, 0 \rangle, [F_2]_{\equiv}), ([\underline{F_2}]_{\equiv}, \langle redo, \infty \rangle, [\overline{F_2}]_{\equiv}) \}$$

$$A_g^{F_2} = \{ ([H_2 \square E]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_2 \square E]_{\equiv}) \mid ([H_2]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_2]_{\equiv}) \in A_g^{E_2} \} \cup \{ ([E_2 \square G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [E_2 \square J]_{\equiv}) \mid ([G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J]_{\equiv}) \in A_g^E \}$$

$$v_0^{F_2} = [\overline{F_2}]_{\equiv}$$

$$\text{con } [\overline{F_2}]_{\equiv} = [\overline{E_2} \square E]_{\equiv} = [E_2 \square \overline{E}]_{\equiv}, \quad \text{y} \quad [F_2]_{\equiv} = [\underline{E_2} \square E]_{\equiv} = [E_2 \square \underline{E}]_{\equiv}.$$

Entonces definimos

$$\varphi : V^{F_1} \cup \{ [F_1]_{\equiv} \} \rightarrow V^{F_2} \cup \{ [F_2]_{\equiv} \}$$

de la forma siguiente:

- $\forall [H_1]_{\equiv} \in V^{E_1}, \varphi([H_1 \square E]_{\equiv}) = [H_2 \square E]_{\equiv}$, donde $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$.
- $\varphi([E_1 \square G]_{\equiv}) = [E_2 \square G]_{\equiv}$.
- $\varphi([F_1]_{\equiv}) = [F_2]_{\equiv}$.

Se comprueba fácilmente que φ está bien definida, es una biyección y $\varphi([\overline{F_1}]_{\equiv}) = [\overline{F_2}]_{\equiv}$.

Asimismo, para cada $v \in V^{F_1}$ y cada α tal que $\mathcal{C}(v, \alpha) \neq \emptyset$, iremos definiendo una biyección $g_{v, \alpha} : \mathcal{C}(v, \alpha) \rightarrow \mathcal{C}(\varphi(v), \alpha)$, y lo haremos según el tipo de arco considerado.

- Consideremos un arco cualquiera e de $A_a^{F_1}$. Podemos encontrarnos en uno de los seis casos siguientes:

- $e = ([H_1 \sqcap E]_{\equiv}, \langle \alpha, r \rangle_{n, C_1}, [J_1 \sqcap E]_{\equiv})$, con $H_1 \not\equiv \overline{E_1}$ y $([H_1]_{\equiv}, \langle \alpha, r \rangle_{n, C_{11}}, [J_1]_{\equiv}) \in A_a^{E_1}$. Entonces sabemos que $C_1 = C_{11}$. Tomando $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$, $\phi([J_1]_{\equiv}) = [J_2]_{\equiv}$, $h_{[H_1]_{\equiv}, \alpha}(C_1) = C'_1$ tenemos que

$$\varphi([H_1 \sqcap E]_{\equiv}) = [H_2 \sqcap E]_{\equiv} \quad \text{y} \quad \varphi([J_1; E]_{\equiv}) = [J_2; E]_{\equiv}$$

y definimos $g_{[H_1 \sqcap E]_{\equiv}, \alpha}(C_1) = C'_1$, teniéndose asimismo que $rc(C_1) = rc(C'_1)$.

De esta forma, como $E_1 \sim E_2$, $\exists n', s$ tal que

$$([H_2]_{\equiv}, \langle \alpha, s \rangle_{n', C'_1}, [J_2]_{\equiv}) \in A_a^{E_2}$$

y así:

$$([H_2 \sqcap E]_{\equiv}, \langle \alpha, s \rangle_{n', C'_1}, [J_2 \sqcap E]_{\equiv}) \in A_a^{F_2}$$

cumpléndose que

$$\begin{aligned} r([H_1 \sqcap E]_{\equiv}, \alpha, C_1, [J_1 \sqcap E]_{\equiv}) &= r([H_1]_{\equiv}, \alpha, C_1, [J_1]_{\equiv}) = \\ r([H_2]_{\equiv}, \alpha, C'_1, [J_2]_{\equiv}) &= r([H_2 \sqcap E]_{\equiv}, \alpha, C'_1, [J_2 \sqcap E]_{\equiv}) \end{aligned}$$

ya que $H_2 \not\equiv \overline{E_2}$ (por ser ϕ una biyección tal que $\phi(\overline{[E_1]_{\equiv}}) = \overline{[E_2]_{\equiv}}$)

- $e = ([H_1 \sqcap E]_{\equiv}, \langle \alpha, r \rangle_{n, C_1}, [J_1 \sqcap E]_{\equiv})$, con $H_1 \equiv \overline{E_1} \wedge J_1 \not\equiv \underline{E_1}$, y $([H_1]_{\equiv}, \langle \alpha, r \rangle_{n, C_{11}}, [J_1]_{\equiv}) \in A_a^{E_1}$, teniéndose que $C_1 = C_{11} \cup C$, donde $C = \{ \langle \alpha, r_j \rangle_{n_j} \mid (\overline{[E]_{\equiv}}, \langle \alpha, r_j \rangle_{n_j, C}, [K]_{\equiv}) \in A_a^E \}$. Obsérvese que por la restricción sintáctica impuesta, si C es distinto de \emptyset , entonces C es el único conjunto de conflictos de $\mathcal{C}(\overline{[E]_{\equiv}}, \alpha)$.

Tomando $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$, $\phi([J_1]_{\equiv}) = [J_2]_{\equiv}$, $h_{[H_1]_{\equiv}, \alpha}(C_{11}) = C'_{11}$ tenemos que

$$\varphi([H_1 \sqcap E]_{\equiv}) = [H_2 \sqcap E]_{\equiv} \quad \text{y} \quad \varphi([J_1; E]_{\equiv}) = [J_2; E]_{\equiv}$$

y definimos $g_{[H_1 \sqcap E]_{\equiv}, \alpha}(C_1) = C'_{11} \cup C = C'_1$, teniéndose que $rc(C_1) = rc(C'_1)$.

De esta forma, como $E_1 \sim E_2$, $\exists n', s$ tal que

$$([H_2]_{\equiv}, \langle \alpha, s \rangle_{n', C'_{11}}, [J_2]_{\equiv}) \in A_a^{E_2}$$

y así:

$$([H_2 \sqcap E]_{\equiv}, \langle \alpha, s \rangle_{n', C'_{11}}, [J_2 \sqcap E]_{\equiv}) \in A_a^{F_2}$$

cumpléndose que

$$\begin{aligned} r([H_1 \square E]_{\equiv}, \alpha, C_1, [J_1 \square E]_{\equiv}) &= r([H_1]_{\equiv}, \alpha, C_{11}, [J_1]_{\equiv}) = \\ r([H_2]_{\equiv}, \alpha, C'_{11}, [J_2]_{\equiv}) &= r([H_2 \square E]_{\equiv}, \alpha, C'_{11}, [J_2 \square E]_{\equiv}) \end{aligned}$$

ya que $H_2 \equiv \overline{E_2}$ y $J_2 \not\equiv \underline{E_2}$ (por ser ϕ una biyección tal que $\phi([\overline{E_1}]_{\equiv}) = [\overline{E_2}]_{\equiv}$ y $\phi([\underline{E_1}]_{\equiv}) = [\underline{E_2}]_{\equiv}$).

- $e = ([E_1 \square G]_{\equiv}, \langle \alpha, r \rangle_{n, C_1}, [E_1 \square J]_{\equiv})$, con $G \not\equiv \overline{E}$ y $([G]_{\equiv}, \langle \alpha, r \rangle_{n, C_1}, [J]_{\equiv}) \in A_a^E$.

Se razona de forma similar al primer caso considerado.

- $e = ([E_1 \square G]_{\equiv}, \langle \alpha, r \rangle_{n, C_1}, [E_1 \square J]_{\equiv})$, con $G \equiv \overline{E} \wedge J \not\equiv \underline{E}$ y $([G]_{\equiv}, \langle \alpha, r \rangle_{n, C_{11}}, [J]_{\equiv}) \in A_a^{E_1}$

Este caso se razona de forma análoga al segundo caso considerado.

- $e = ([H_1 \square E]_{\equiv}, \langle \alpha, r \rangle_{n, C_1}, [J_1 \square E]_{\equiv})$, con $H_1 \equiv \overline{E_1}$, y $J_1 \equiv \underline{E_1}$ y $([H_1]_{\equiv}, \langle \alpha, r \rangle_{n, C_{11}}, [J_1]_{\equiv}) \in A_a^{E_1}$ teniéndose que $C_1 = C_{11} \cup C$, donde $C = \{\langle \alpha, r_j \rangle_{n_j} \mid ([\overline{E}]_{\equiv}, \langle \alpha, r_j \rangle_{n_j, C} [K]_{\equiv}) \in A_a^E\} \in A_a^E$. Obsérvese que por la restricción sintáctica impuesta, si C es distinto de \emptyset , entonces C es el único conjunto de conflictos de $\mathcal{C}([\overline{E}]_{\equiv}, \alpha)$.

Tomando $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$, $\phi([J_1]_{\equiv}) = [J_2]_{\equiv}$, $h_{[H_1]_{\equiv}, \alpha}(C_{11}) = C'_{11}$ tenemos que

$$\varphi([H_1 \square E]_{\equiv}) = [H_2 \square E]_{\equiv} \quad \text{y} \quad \varphi([J_1; E]_{\equiv}) = [J_2; E]_{\equiv}$$

y definimos $g_{[H_1 \square E]_{\equiv}, \alpha}(C_1) = C'_{11} \cup C = C'_1$, teniéndose que $rc(C_1) = rc(C'_1)$.

De esta forma, como $E_1 \sim E_2$, $\exists n', s$ tal que

$$([H_2]_{\equiv}, \langle \alpha, s \rangle_{n', C'_{11}}, [J_2]_{\equiv}) \in A_a^{E_2}$$

y así:

$$([H_2 \square E]_{\equiv}, \langle \alpha, s \rangle_{n', C'_1}, [J_2 \square E]_{\equiv}) \in A_a^{F_2}$$

cumpléndose que

$$\begin{aligned} r([H_1 \square E]_{\equiv}, \alpha, C_1, [J_1 \square E]_{\equiv}) &= \\ r([H_1]_{\equiv}, \alpha, C_{11}, [J_1]_{\equiv}) + r([\overline{E}]_{\equiv}, \alpha, C, [\underline{E}]_{\equiv}) &= \\ r([H_2]_{\equiv}, \alpha, C'_{11}, [J_2]_{\equiv}) + r([\overline{E}]_{\equiv}, \alpha, C, [\underline{E}]_{\equiv}) &= \\ r([H_2 \square E]_{\equiv}, \alpha, C'_1, [J_2 \square E]_{\equiv}) & \end{aligned}$$

- $e = ([E_1 \sqcap G]_{\equiv}, \langle \alpha, r \rangle_{n, C_1}, [E_1 \sqcap J]_{\equiv})$, con $G \equiv \overline{E}$, y $J \equiv \underline{E}$ y
 $([G]_{\equiv}, \langle \alpha, r \rangle_{n, C_1}, [J]_{\equiv}) \in A_a^E$.
 Análogo al caso anterior.

• Para los arcos de $A_g^{F_1}$ tenemos otra vez dos casos posibles:

- $([H_1 \sqcap E]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1 \sqcap G]_{\equiv})$,
 con $([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1]_{\equiv}) \in A_g^{E_1}$.
- $([E_1 \sqcap G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [E_1 \sqcap J]_{\equiv})$,
 con $([G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J]_{\equiv}) \in A_g^E$

Ambos son inmediatos, y pueden demostrarse con un razonamiento similar al seguido en (i) y (ii).

(iv) Consideremos

$$F_1 = E_1 [f] \quad \text{y} \quad F_2 = E_2 [f]$$

En este caso $nts(\overline{F_1})$ y $nts(\overline{F_2})$ son de la forma

$$nts(\overline{F_1}) = (V^{F_1} \cup \{[F_1]_{\equiv}\}, A_a^{F_1} \cup A_{sr}^{F_1} \cup A_g^{F_1}, v_0^{F_1})$$

donde:

$$V^{F_1} = \{ [H_1 [f]]_{\equiv} \mid H_1 \in [\overline{E_1}] \}$$

$$A_a^{F_1} = \{ ([H_1 [f]]_{\equiv}, \langle f(\alpha), r \rangle_{n, C}, [J_1 [f]]_{\equiv}) \mid ([H_1]_{\equiv}, \langle \alpha, r \rangle_{n, C_1}, [J_1]_{\equiv}) \in A_a^{E_1} \}$$

donde $C = \{ \langle f(\alpha), r \rangle_n \mid \langle \alpha, r \rangle_n \in C_1 \}$

$$A_{sr}^{F_1} = \{ ([\overline{F_1}]_{\equiv}, \langle skip, 0 \rangle, [F_1]_{\equiv}), ([F_1]_{\equiv}, \langle redo, \infty \rangle, [\overline{F_1}]_{\equiv}) \}$$

$$A_g^{F_1} = \{ ([H_1 [f]]_{\equiv}, \langle f(\alpha), r \rangle_{n_1} \parallel \langle f(\beta), s \rangle_{n_2}, [J_1 [f]]_{\equiv}) \mid ([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1]_{\equiv}) \in A_g^{E_1} \}$$

$$v_0^{F_1} = [\overline{F_1}]_{\equiv}$$

Se puede observar que en V^{F_1} tenemos:

- $[\overline{F_1}]_{\equiv} = [\overline{E_1} [f]]_{\equiv}$

$$- \underline{[F_1]}_{\equiv} = \underline{[E_1 [f]]}_{\equiv}$$

$$nts(F_2) = (V^{F_2} \cup \{[\underline{F_2}]_{\equiv}\}, A^{F_2} \cup A_{sr}^{F_2} \cup A_g^{F_2}, v_0^{F_2})$$

donde:

$$V^{F_2} = \{ [H_2 [f]]_{\equiv} \mid H_2 \in \overline{[E_2]} \}$$

$$A_a^{F_2} = \{ ([H_2 [f]]_{\equiv}, \langle f(\alpha), r \rangle_{n,C}, [J_2 [f]]_{\equiv}) \mid ([H_2]_{\equiv}, \langle \alpha, r \rangle_{n,C_1}, [J_2]_{\equiv}) \in A_a^{E_2} \}$$

donde $C = \{ \langle f(\alpha), r \rangle_n \mid \langle \alpha, r \rangle_n \in C_1 \}$

$$A_{sr}^{F_2} = \{ ([\overline{F_2}]_{\equiv}, \langle skip, 0 \rangle, [\underline{F_2}]_{\equiv}), ([\underline{F_2}]_{\equiv}, \langle redo, \infty \rangle, [\overline{F_2}]_{\equiv}) \}$$

$$A_g^{F_2} = \{ ([H_2 [f]]_{\equiv}, \langle f(\alpha), r \rangle_{n_1} \parallel \langle f(\beta), s \rangle_{n_2}, [J_2 [f]]_{\equiv}) \mid ([H_2]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_2]_{\equiv}) \in A_g^{E_2} \}$$

$$v_0^{F_2} = [\overline{F_2}]_{\equiv}$$

$$\text{con } [\overline{F_2}]_{\equiv} = \overline{[E_2 [f]]}_{\equiv}, \text{ y } [\underline{F_2}]_{\equiv} = \underline{[E_2 [f]]}_{\equiv}.$$

Entonces definimos

$$\varphi : V^{F_1} \cup \{[\underline{F_1}]_{\equiv}\} \rightarrow V^{F_2} \cup \{[\underline{F_2}]_{\equiv}\}$$

de la forma siguiente:

- $\forall [H_1]_{\equiv} \in V^{E_1}, \varphi([H_1 [f]]_{\equiv}) = [H_2 [f]]_{\equiv}$, donde $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$.
- $\varphi([\underline{F_1}]_{\equiv}) = [\underline{F_2}]_{\equiv}$

Es inmediato comprobar que φ está bien definida, es una biyección, y $\varphi([\overline{F_1}]_{\equiv}) = [\overline{F_2}]_{\equiv}$.

Además, para cada $v \in V^{F_1}$ y cada α tal que $\mathcal{C}(v, \alpha) \neq \emptyset$, iremos definiendo una biyección $g_{v,\alpha} : \mathcal{C}(v, \alpha) \rightarrow \mathcal{C}(\varphi(v), \alpha)$, en función de los distintos casos que vamos a considerar.

- Consideremos un arco cualquiera de $A_a^{F_1}$:

$$([H_1 [f]]_{\equiv}, \langle f(\alpha), r \rangle_{n,C}, [J_1 [f]]_{\equiv}), \text{ donde } ([H_1]_{\equiv}, \langle \alpha, r \rangle_{n,C_1}, [J_1]_{\equiv}) \in A_a^{E_1},$$

con $C = \{ \langle f(\alpha), r \rangle_n \mid \langle \alpha, r \rangle_n \in C_1 \}$.

Tomando $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$, $\phi([J_1]_{\equiv}) = [J_2]_{\equiv}$, $h_{[H_1]_{\equiv}, \alpha}(C_1) = C'_1$ tenemos que

$$\varphi([H_1[f]]_{\equiv}) = [H_2[f]]_{\equiv} \quad y \quad \varphi([J_1[f]]_{\equiv}) = [J_2[f]]_{\equiv}$$

y definimos $g_{[H_1[f]]_{\equiv}, f(\alpha)}(C) = C'$, donde $C' = \{ \langle f(\alpha), r \rangle_n \mid \langle \alpha, r \rangle_n \in C'_1 \}$ teniéndose que $rc(C) = rc(C')$.

De esta forma, como $E_1 \sim E_2$, $\exists n', s$ tal que

$$([H_2]_{\equiv}, \langle \alpha, s \rangle_{n', C'_1}, [J_2]_{\equiv}) \in A_a^{E_2}$$

y así:

$$([H_2[f]]_{\equiv}, \langle f(\alpha), s \rangle_{n', C'}, [J_2[f]]_{\equiv}) \in A_a^{F_2}$$

cumpléndose que

$$\begin{aligned} r([H_1[f]]_{\equiv}, f(\alpha), C, [J_1[f]]_{\equiv}) &= r([H_1]_{\equiv}, \alpha, C_1, [J_1]_{\equiv}) = \\ r([H_2]_{\equiv}, \alpha, C'_1, [J_2]_{\equiv}) &= r([H_2[f]]_{\equiv}, f(\alpha), C', [J_2[f]]_{\equiv}) \end{aligned}$$

- Los arcos de $A_g^{F_1}$ deben ser de la forma:

$$([H_1[f]]_{\equiv}, \langle f(\alpha), r \rangle_{n_1} \parallel \langle f(\beta), s \rangle_{n_2}, [J_1[f]]_{\equiv})$$

donde $([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1]_{\equiv}) \in A_g^{E_1}$.

Entonces, como $E_1 \sim E_2$, y teniendo en cuenta que f es biyectiva, podemos llegar a la conclusión de que se cumple la propiedad deseada para $E_1[f]$ y $E_2[f]$.

(v) Consideremos $a \in \mathcal{A}$, cualquiera y:

$$F_1 = E_1 \text{ sy } a \quad y \quad F_2 = E_2 \text{ sy } a$$

Tenemos:

$$nts(\overline{F_1}) = (V^{F_1} \cup \{[F_1]_{\equiv}\}, A_a^{F_1} \cup A_{sr}^{F_1} \cup A_g^{F_1}, v_0^{F_1})$$

donde

$$V^{F_1} = \{ [H_1 \text{ sy } a]_{\equiv} \mid H_1 \in [\overline{E_1}] \}$$

$$\begin{aligned}
A_a^{F_1} = & \{ ([H_1 \text{ sy } a]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_1 \text{ sy } a]_{\equiv}) \mid ([H_1]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_1]_{\equiv}) \in A_a^{E_1} \} \cup \\
& \{ ([H_1 \text{ sy } a]_{\equiv}, \langle \alpha_1 \oplus_a \alpha_2, R_{12} \rangle_{n_1.n_2, C_{12}}, [J_1 \text{ sy } a]_{\equiv}) \mid a \in A(\alpha_1), \hat{a} \in A(\alpha_2), \\
& \{ \langle \alpha_1, r_1 \rangle_{n_1}, \langle \alpha_2, r_2 \rangle_{n_2} \} \in BC(H_1 \text{ sy } a) \text{ con } H_1 \text{ sy } a \in [\overline{E_1 \text{ sy } a}] \text{ y} \\
& H_1 \text{ sy } a \xrightarrow{\langle \alpha_1, r_1 \rangle_{n_1, C_1}} G_1 \text{ sy } a \equiv G_1^* \text{ sy } a \xrightarrow{\langle \alpha_2, r_2 \rangle_{n_2, C_2}} J_1 \text{ sy } a \}
\end{aligned}$$

$$A_{sr}^{F_1} = \{ ([\overline{F_1}]_{\equiv}, \langle \text{skip}, 0 \rangle, [F_1]_{\equiv}), ([F_1]_{\equiv}, \langle \text{redo}, \infty \rangle, [\overline{F_1}]_{\equiv}) \}$$

$$\begin{aligned}
A_g^{F_1} = & \{ ([H_1 \text{ sy } a]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_1}, [J_1 \text{ sy } a]_{\equiv}) \mid \\
& ([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1]_{\equiv}) \in A_g^{E_1} \} \cup \\
& \{ ([H_1 \text{ sy } a]_{\equiv}, \langle \alpha_1 \oplus_a \alpha_2, R_{12} \rangle_{n_1.n_2} \parallel \langle \beta, s \rangle_{n_3}, [J_1 \text{ sy } a]_{\equiv}) \mid \\
& a \in A(\alpha_1), \hat{a} \in A(\alpha_2), \{ \langle \alpha_1, r_1 \rangle_{n_1}, \langle \alpha_2, r_2 \rangle_{n_2}, \langle \beta, s \rangle_{n_3} \} \in BC(H_1 \text{ sy } a) \text{ con} \\
& H_1 \text{ sy } a \xrightarrow{\langle \alpha_1, r_1 \rangle_{n_1, C_1}} G_1 \text{ sy } a \equiv G_1^* \text{ sy } a \xrightarrow{\langle \alpha_2, r_2 \rangle_{n_2, C_2}} G_2 \text{ sy } a \equiv G_2^* \text{ sy } a \xrightarrow{\langle \beta, s \rangle_{n_3, C_3}} J_1 \text{ sy } a \}
\end{aligned}$$

donde

$$\begin{aligned}
R_{12} &= \frac{r_1}{cr(H_1 \text{ sy } a, \langle \alpha_1, r_1 \rangle_{n_1})} \cdot \frac{r_2}{cr(H_1 \text{ sy } a, \langle \alpha_2, r_2 \rangle_{n_2})} \cdot \min_{i=1,2} \{ cr(H_1 \text{ sy } a, \langle \alpha_i, r_i \rangle) \} = \\
&= \frac{r_1}{cr(C_1)} \cdot \frac{r_2}{cr(C_2)} \cdot \min \{ cr(C_1), cr(C_2) \}
\end{aligned}$$

$$v_0^{F_1} = [\overline{F_1}]_{\equiv}$$

Podemos observar que en V^{F_1} tenemos: $[\overline{F_1}]_{\equiv} = [\overline{E_1 \text{ sy } a}]_{\equiv}$ y $[F_1]_{\equiv} = [E_1 \text{ sy } a]_{\equiv}$

$$nts(\overline{F_2}) = (V^{F_2} \cup \{ [F_2]_{\equiv} \}, A_a^{F_2} \cup A_{sr}^{F_2} \cup A_g^{F_2}, v_0^{F_2})$$

donde

$$V^{F_2} = \{ [H_2 \text{ sy } a]_{\equiv} \mid H_2 \in [\overline{E_2}] \}$$

$$\begin{aligned}
A_a^{F_2} = & \{ ([H_2 \text{ sy } a]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_2 \text{ sy } a]_{\equiv}) \mid ([H_2]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_2]_{\equiv}) \in A_a^{E_2} \} \cup \\
& \{ ([H_2 \text{ sy } a]_{\equiv}, \langle \alpha_1 \oplus_a \alpha_2, R_{12} \rangle_{n_1.n_2, C_{12}}, [J_2 \text{ sy } a]_{\equiv}) \mid a \in A(\alpha_1), \hat{a} \in A(\alpha_2), \\
& \{ \langle \alpha_1, r_1 \rangle_{n_1}, \langle \alpha_2, r_2 \rangle_{n_2} \} \in BC(H_2 \text{ sy } a) \text{ con } H_2 \text{ sy } a \in [\overline{E_2 \text{ sy } a}] \text{ y} \\
& H_2 \text{ sy } a \xrightarrow{\langle \alpha_1, r_1 \rangle_{n_1, C_1}} G_2 \text{ sy } a \equiv G_2^* \text{ sy } a \xrightarrow{\langle \alpha_2, r_2 \rangle_{n_2, C_2}} J_2 \text{ sy } a \}
\end{aligned}$$

$$A_{sr}^{F_2} = \{ ([\overline{F_2}]_{\equiv}, \langle \text{skip}, 0 \rangle, [F_2]_{\equiv}), ([F_2]_{\equiv}, \langle \text{redo}, \infty \rangle, [\overline{F_2}]_{\equiv}) \}$$

$$\begin{aligned}
A_g^{F_2} = & \{ ([H_2 \text{ sy } a]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_2 \text{ sy } a]_{\equiv}) \mid \\
& ([H_2]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_2]_{\equiv}) \in A_g^{E_2} \} \cup \\
& \{ ([H_2 \text{ sy } a]_{\equiv}, \langle \alpha_1 \oplus_a \alpha_2, R_{12} \rangle_{n_1 \cdot n_2} \parallel \langle \beta, s \rangle_{n_3}, [J_2 \text{ sy } a]_{\equiv}) \mid \\
& a \in A(\alpha_1), \hat{a} \in A(\alpha_2), \{ \langle \alpha_1, r_1 \rangle_{n_1}, \langle \alpha_2, r_2 \rangle_{n_2}, \langle \beta, s \rangle_{n_3} \} \in BC(H_2 \text{ sy } a) \text{ con} \\
& H_2 \text{ sy } a \xrightarrow{\langle \alpha_1, r_1 \rangle_{n_1}, C_1} G_1' \text{ sy } a \equiv G_1'^* \text{ sy } a \xrightarrow{\langle \alpha_2, r_2 \rangle_{n_2}, C_2} G_2' \text{ sy } a \equiv G_2'^* \text{ sy } a \xrightarrow{\langle \beta, s \rangle_{n_3}, C_3} J_2 \text{ sy } a \}
\end{aligned}$$

donde

$$\begin{aligned}
R_{12} &= \frac{r_1}{cr(H_1 \text{ sy } a, \langle \alpha_1, r_1 \rangle_{n_1})} \cdot \frac{r_2}{cr(H_1 \text{ sy } a, \langle \alpha_2, r_2 \rangle_{n_2})} \cdot \min_{i=1,2} \{ cr(H_1 \text{ sy } a, \langle \alpha_i, r_i \rangle) \} = \\
&= \frac{r_1}{cr(C_1)} \cdot \frac{r_2}{cr(C_2)} \cdot \min \{ cr(C_1), cr(C_2) \}
\end{aligned}$$

$$v_0^{F_2} = [\overline{F_2}]_{\equiv}$$

Podemos observar que:

- $[\overline{F_2}]_{\equiv} = [\overline{E_2 \text{ sy } a}]_{\equiv},$
- $[F_2]_{\equiv} = [E_2 \text{ sy } a]_{\equiv}.$

Entonces definimos

$$\varphi : V^{F_1} \cup \{ [F_1]_{\equiv} \} \rightarrow V^{F_2} \cup \{ [F_2]_{\equiv} \}$$

de la forma siguiente:

- Para cada $[H_1]_{\equiv} \in V^{E_1}$, tomamos $\varphi([H_1 \text{ sy } a]_{\equiv}) = [H_2 \text{ sy } a]_{\equiv}$ donde $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}.$
- $\varphi([F_1]_{\equiv}) = [F_2]_{\equiv}.$

Es fácil ver que φ está bien definida, es una biyección y $\varphi([\overline{F_1}]_{\equiv}) = [\overline{F_2}]_{\equiv}.$

Asimismo para cada $v \in V^{F_1}$ y cada α tal que $\mathcal{C}(v, \alpha) \neq \emptyset$, iremos definiendo una biyección $g_{v, \alpha} : \mathcal{C}(v, \alpha) \rightarrow \mathcal{C}(\varphi(v), \alpha)$ en función de los distintos casos que vamos a considerar.

- Consideremos un arco cualquiera e de $A_a^{F_1}$. Tenemos otra vez dos posibles casos: el primero correspondería a una multiacción estocástica ejecutable desde $[H_1]_{\equiv}$, y el segundo a una multiacción estocástica obtenida por sincronización. Sin embargo, como consecuencia de la restricción sintáctica impuesta, no podemos tener dos arcos conectando el mismo par de nodos, uno correspondiente

al primer caso y otro al segundo (eso sólo sería posible si hubiese un paralelo dentro de una elección). Así pues, sólo son posibles los dos casos por separado:

- $e = ([H_1 \text{ sy } a]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_1 \text{ sy } a]_{\equiv})$ con
 $([H_1]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_1]_{\equiv}) \in A_a^{E_1}$.

Tomando $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$, $\phi([J_1]_{\equiv}) = [J_2]_{\equiv}$ y $h(C) = C'$, tenemos que:

$$\varphi([H_1 \text{ sy } a]_{\equiv}) = [H_2 \text{ sy } a]_{\equiv} \text{ y } \varphi([J_1 \text{ sy } a]_{\equiv}) = [J_2 \text{ sy } a]_{\equiv}$$

y definimos $g_{[H_1]_{\equiv}, \alpha}(C) = C'$, teniéndose que $rc(C) = rc(C')$.

De esta forma, como $E_1 \sim E_2$, $\exists n', s$ tal que

$$([H_2]_{\equiv}, \langle \alpha, s \rangle_{n',C'}, [J_2]_{\equiv}) \in A_a^{E_2}$$

y así:

$$([H_2 \text{ sy } a]_{\equiv}, \langle \alpha, s \rangle_{n',C'}, [J_2 \text{ sy } a]_{\equiv}) \in A_a^{E_2}$$

cumpliéndose que

$$\begin{aligned} r([H_1 \text{ sy } a]_{\equiv}, \alpha, C, [J_1 \text{ sy } a]_{\equiv}) &= r([H_1]_{\equiv}, \alpha, C, [J_1]_{\equiv}) = \\ r([H_2]_{\equiv}, \alpha, C', [J_2]_{\equiv}) &= r([H_2 \text{ sy } a]_{\equiv}, \alpha, C', [J_2 \text{ sy } a]_{\equiv}) \end{aligned}$$

- $e = ([H_1 \text{ sy } a]_{\equiv}, \langle \alpha_1 \oplus_a \alpha_2, R_{12} \rangle_{n_1, m_1, C}, [J_1 \text{ sy } a]_{\equiv})$ con
 $H_1 \in [\overline{E_1}]$, $a \in A(\alpha_1)$, $\hat{a} \in A(\alpha_2)$, $\{\langle \alpha_1, r_1 \rangle_{n_1}, \langle \alpha_2, s_1 \rangle_{m_1}\} \in BC(H_1 \text{ sy } a)$ y donde

$$\begin{aligned} H_1 \text{ sy } a \xrightarrow{\langle \alpha_1, r_1 \rangle_{n_1, C_1}} G_1 \text{ sy } a &\equiv G_1^* \text{ sy } a \xrightarrow{\langle \alpha_2, s_1 \rangle_{m_1, C_2}} J_1 \text{ sy } a \\ H_1 \text{ sy } a \xrightarrow{\langle \alpha_2, s_1 \rangle_{m_1, C_2}} G'_1 \text{ sy } a &\equiv G_1'^* \text{ sy } a \xrightarrow{\langle \alpha_1, r_1 \rangle_{n_1, C_1}} J'_1 \text{ sy } a \end{aligned}$$

con $J_1 \equiv J'_1$.

Vamos a proceder por inducción sobre el número de veces que se ha aplicado la Regla *Sy2'* para obtener la transición etiquetada con $\langle \alpha_1 \oplus_a \alpha_2, R_{12} \rangle_{n_1, m_1}$.

Caso base. Se ha aplicado la Regla *Sy2'* una sola vez.

En tal caso: $\{\langle \alpha_1, r_1 \rangle_{n_1}, \langle \alpha_2, s_1 \rangle_{m_1}\} \in BC(H_1)$.

Supongamos que además tenemos:

$$\begin{aligned} \langle \alpha_1, r_i \rangle_{n_i, C_1} \text{ tal que } ([H_1]_{\equiv}, \langle \alpha_1, r_i \rangle_{n_i, C_1}, [G_1]_{\equiv}) &\in A_a^{E_1}, \quad i = 1, \dots, n_1 \\ \langle \alpha_2, s_j \rangle_{m_j, C_2} \text{ tal que } ([H_1]_{\equiv}, \langle \alpha_2, s_j \rangle_{m_j, C_2}, [G'_1]_{\equiv}) &\in A_a^{E_1}, \quad j = 1, \dots, m_2 \end{aligned}$$

Tomando $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$, $\phi([J_1]_{\equiv}) = [J_2]_{\equiv}$, $\phi([G_1]_{\equiv}) = [G_2]_{\equiv}$,
 $\phi([G'_1]_{\equiv}) = [G'_2]_{\equiv}$, $h_{[H_1]_{\equiv}, \alpha_1}(C_1) = C'_1$ y $h_{[H_1]_{\equiv}, \alpha_2}(C_2) = C'_2$ tenemos
que:

$$\begin{aligned}\varphi([H_1 sy a]_{\equiv}) &= [H_2 sy a]_{\equiv}, & \varphi([J_1 sy a]_{\equiv}) &= [J_2 sy a]_{\equiv} \\ \varphi([G_1 sy a]_{\equiv}) &= [G_2 sy a]_{\equiv}, & \varphi([G'_1 sy a]_{\equiv}) &= [G'_2 sy a]_{\equiv}\end{aligned}$$

y definimos $g_{[H_1 sy a]_{\equiv}, \alpha_1 \oplus_a \alpha_2}(C) = C'$ donde

$$C' = \{ \langle \alpha_1, r'_i \rangle_{n'_i} \oplus_a \langle \alpha_2, s'_j \rangle_{m'_j} \mid \langle \alpha_1, r'_i \rangle_{n'_i} \in C'_1, \langle \alpha_2, s'_j \rangle_{m'_j} \in C'_2 \}$$

teniéndose que $rc(C) = rc(C')$, ya que por la proposición 5.2, se tiene
que: $cr(C) = \min\{cr(C_1), cr(C_2)\} = \min\{cr(C'_1), cr(C'_2)\} = cr(C')$.

Aplicando el lema 6.1 podemos generar $n_1 \cdot m_2$ sincronizaciones posi-
bles, por medio de los pares $(\langle \alpha_1, r_i \rangle_{n_i}, \langle \alpha_2, s_j \rangle_{m_j})$, para $i = 1, \dots, n_1$,
 $j = 1, \dots, m_2$, todas con el mismo conflicto C (por el lema 5.2), siendo

$$C = \{ \langle \alpha_1 \oplus_a \alpha_2, R_{ij} \rangle_{n_i, m_j} \mid \langle \alpha_1, r_i \rangle_{n_i} \in C_1, \langle \alpha_2, s_j \rangle_{m_j} \in C_2 \}$$

Para cada uno de los pares $(\langle \alpha_1, r_i \rangle_{n_i}, \langle \alpha_2, s_j \rangle_{m_j})$ podemos ob-
tener una transición $([H_1 sy a]_{\equiv}, \langle \alpha_1 \oplus_a \alpha_2, R_{ij} \rangle_{n_i, m_j, C}, [J_1 sy a]_{\equiv}) \in A_a^{F_1}$,
donde utilizando la proposición 5.2 el valor R_{ij} puede ser escrito de la
forma:

$$R_{ij} = \frac{r_i}{cr1} \cdot \frac{s_j}{cr2} \cdot \min\{cr1, cr2\}$$

donde

$$\begin{aligned}cr1 &= cr(H_1 sy a, \langle \alpha_1, r_1 \rangle) = cr(C_1), \\ cr2 &= cr(H_1 sy a, \langle \alpha_2, s_1 \rangle) = cr(C_2)\end{aligned}$$

y como no hay más arcos de $[H_1 sy a]_{\equiv}$ a $[J_1 sy a]_{\equiv}$ con el mismo conflicto
 C , entonces:

$$\begin{aligned}r([H_1 sy a]_{\equiv}, \alpha_1 \oplus_a \alpha_2, C, [J_1 sy a]_{\equiv}) &= \sum_{\substack{j=1, \dots, m_2 \\ i=1, \dots, n_1}} R_{ij} = \\ &= \frac{\min\{cr(C_1), cr(C_2)\}}{cr(C_1) \cdot cr(C_2)} \cdot \sum_{\substack{j=1, \dots, m_2 \\ i=1, \dots, n_1}} r_i \cdot s_j = \\ &= \frac{\min\{cr(C_1), cr(C_2)\}}{cr(C_1) \cdot cr(C_2)} \cdot r([H_1]_{\equiv}, \alpha_1, C_1, [G_1]_{\equiv}) \cdot r([H_1]_{\equiv}, \alpha_2, C_2, [G'_1]_{\equiv})\end{aligned}$$

Como $E_1 \sim E_2$, por definición de \sim , en E_2 nos encontraremos con que
existen n'_i, r'_i, s'_j, m'_j tal que:

$$\begin{aligned}([H_2]_{\equiv}, \langle \alpha_1, r'_i \rangle_{n'_i, C'_1}, [G_2]_{\equiv}) &\in A_a^{E_2}, & i &= 1, \dots, p_1 & \text{para algún } p_1 \geq 1 \\ ([H_2]_{\equiv}, \langle \alpha_2, s'_j \rangle_{m'_j, C'_2}, [G'_2]_{\equiv}) &\in A_a^{E_2}, & j &= 1, \dots, q_2 & \text{para algún } q_2 \geq 1\end{aligned}$$

Entonces por la proposición 6.2 y lema 6.1, podemos asegurar que

$$([H_2]_{\equiv}, \langle \alpha_1, r'_i \rangle_{n'_i} \parallel \langle \alpha_2, s'_j \rangle_{m'_j}, [J_2]_{\equiv}) \in A_g^{E_2}$$

para $i = 1, \dots, p_1$, $j = 1, \dots, q_2$, con lo cual serán posibles todas esas sincronizaciones, obteniéndose así $p_1 \cdot q_2$ arcos, todos con la misma multiacción $\alpha_1 \oplus_a \alpha_2$ y el mismo conjunto de conflictos C' (por el lema 5.2). Asimismo $\langle \alpha_1, r'_i \rangle_{n'_i} \oplus_a \langle \alpha_2, s'_j \rangle_{m'_j} = \langle \alpha_1 \oplus_a \alpha_2, R'_{ij} \rangle_{n'_i, m'_j}$ con

$$R'_{ij} = \frac{r'_i}{cr(C'_1)} \cdot \frac{s'_j}{cr(C'_2)} \cdot \min\{cr(C'_1), cr(C'_2)\}$$

Obsérvese que no hay más arcos de $[H_2 \text{ sy } a]_{\equiv}$ a $[J_2 \text{ sy } a]_{\equiv}$ con el mismo conflicto. Así:

$$\begin{aligned} r([H_2 \text{ sy } a]_{\equiv}, \alpha_1 \oplus_a \alpha_2, C', [J_2 \text{ sy } a]_{\equiv}) &= \sum_{\substack{j=1, \dots, q_2 \\ i=1, \dots, p_1}} R'_{ij} = \\ &= \frac{\min\{cr(C'_1), cr(C'_2)\}}{cr(C'_1) \cdot cr(C'_2)} \cdot \sum_{\substack{j=1, \dots, q_2 \\ i=1, \dots, p_1}} r'_i \cdot s'_j = \\ &= \frac{\min\{cr(C'_1), cr(C'_2)\}}{cr(C'_1) \cdot cr(C'_2)} \cdot r([H_2]_{\equiv}, \alpha_1, C'_1, [G_2]_{\equiv}) \cdot r([H_2]_{\equiv}, \alpha_2, C'_2, [G'_2]_{\equiv}) = \\ &= \frac{\min\{cr(C_1), cr(C_2)\}}{cr(C_1) \cdot cr(C_2)} \cdot r([H_1]_{\equiv}, \alpha_1, C_1, [G_1]_{\equiv}) \cdot r([H_1]_{\equiv}, \alpha_2, C_2, [G'_1]_{\equiv}) = \\ &= r([H_1 \text{ sy } a]_{\equiv}, \alpha_1 \oplus_a \alpha_2, C, [J_1 \text{ sy } a]_{\equiv}) \end{aligned}$$

Caso general. Suponemos ahora que e ha sido obtenido después de aplicar n veces la Regla $Sy2'$, esto es, la multiacción estocástica se obtiene de la forma:

$$(\langle \beta_1, r_1 \rangle \oplus_a \dots \oplus_a \langle \beta_k, r_k \rangle) \oplus_a (\langle \beta_{k+1}, r_{k+1} \rangle \oplus_a \dots \oplus_a \langle \beta_{n+1}, r_{n+1} \rangle)$$

Entonces, podemos aplicar la hipótesis de inducción a ambas partes, y por medio de un razonamiento muy similar al seguido en el caso base obtenemos la propiedad deseada.

- Consideremos ahora un arco cualquiera e de $A_g^{E_1}$. Tenemos otra vez dos casos:

$$\begin{aligned} - e &= ([H_1 \text{ sy } a]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1 \text{ sy } a]_{\equiv}), \\ &\text{con } ([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1]_{\equiv}) \in A_g^{E_1}. \end{aligned}$$

Es inmediato, teniendo en cuenta que $E_1 \sim E_2$.

$$\begin{aligned} - e &= ([H_1 \text{ sy } a]_{\equiv}, \langle \alpha, r \rangle_p \parallel \langle \beta, s \rangle_q, [J_1 \text{ sy } a]_{\equiv}), \\ &\text{donde } \langle \alpha, r \rangle_p \text{ se ha obtenido por la sincronización de las multiacciones} \end{aligned}$$

estocásticas $\{\langle \alpha_i, r_i \rangle_{p_i}\}_{i=1}^n$, y $\langle \beta, s \rangle_q$ se ha obtenido por la sincronización de las multiacciones estocásticas $\{\langle \beta_j, s_j \rangle_{q_j}\}_{j=1}^m$, con $n + m > 1$. Entonces:

$$\{\langle \alpha_1, r_1 \rangle_{p_1}, \dots, \langle \alpha_n, r_n \rangle_{p_n}, \langle \beta_1, s_1 \rangle_{q_1}, \dots, \langle \beta_m, s_m \rangle_{q_m}\} \in BC(H'_1)$$

donde $H'_1 \equiv H_1$, y así, a partir de la proposición 6.2 podemos obtener:

$$\{\langle \alpha_1, r'_1 \rangle_{p'_1}, \dots, \langle \alpha_n, r'_n \rangle_{p'_n}, \langle \beta_1, s'_1 \rangle_{q'_1}, \dots, \langle \beta_m, s'_m \rangle_{q'_m}\} \in BC(H'_2)$$

para algún $H'_2 \in \phi([H_1]_{\equiv})$. Tomando ahora la multiacción estocástica $\langle \alpha, r' \rangle_{p'}$ que se obtiene sincronizando todas las multiacciones estocásticas de $\{\langle \alpha_i, r'_i \rangle_{p'_i}\}_{i=1}^n$, y como quiera que $\langle \beta, s' \rangle_{q'}$ ha sido obtenido sincronizando $\{\langle \beta_j, s'_j \rangle_{q'_j}\}_{j=1}^m$, tenemos $\{\langle \alpha, r' \rangle_{p'}, \langle \beta, s' \rangle_{q'}\} \in BC(H'_2 \text{ sy } a)$, por lo que habrá una transición fantasma

$$([H'_2 \text{ sy } a]_{\equiv}, \langle \alpha, r' \rangle_{p'} \parallel \langle \beta, s' \rangle_{q'}, [J_2 \text{ sy } a]_{\equiv})$$

cumpliendo así las condiciones de la definición 6.7 para algún $J_2 \in \phi([J_1]_{\equiv})$.

(vi) Consideremos un $a \in \mathcal{A}$, cualquiera, y:

$$F_1 = E_1 \text{ rs } a \quad \text{y} \quad F_2 = E_2 \text{ rs } a$$

Entonces, de nuevo obtenemos $nts(\overline{F_1})$ y $nts(\underline{F_1})$:

$$nts(\overline{F_1}) = (V^{F_1} \cup \{[\underline{F_1}]_{\equiv}\}, A_a^{F_1} \cup A_{sr}^{F_1} \cup A_g^{F_1}, v_0^{F_1})$$

donde

$$V^{F_1} = \{[\overline{F_1}]_{\equiv}\} \cup \{[H_1 \text{ rs } a]_{\equiv} \mid H_1 \in [\overline{E_1}] \text{ y } \exists H \equiv \overline{E_1} \text{ tal que} \\ H \xrightarrow{\langle \alpha_1, r'_1 \rangle} \dots \xrightarrow{\langle \alpha_n, r'_n \rangle} H_1 \text{ con } a, \hat{a} \notin A(\alpha_i), i = 1, \dots, n\}$$

$$A_a^{F_1} = \{([H_1 \text{ rs } a]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_1 \text{ rs } a]_{\equiv}) \mid ([H_1]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_1]_{\equiv}) \in A_a^{E_1}, \\ [H_1 \text{ rs } a]_{\equiv}, [J_1 \text{ rs } a]_{\equiv} \in V^{F_1}, \text{ y } a, \hat{a} \notin A(\alpha)\}$$

$$A_{sr}^{F_1} = \{([\overline{F_1}]_{\equiv}, \langle skip, 0 \rangle, [\underline{F_1}]_{\equiv}), ([\underline{F_1}]_{\equiv}, \langle redo, \infty \rangle, [\overline{F_1}]_{\equiv})\}$$

$$A_g^{F_1} = \{([H_1 \text{ rs } a]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1 \text{ rs } a]_{\equiv}) \mid \\ ([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1]_{\equiv}) \in A_g^{E_1}, [H_1 \text{ rs } a]_{\equiv}, [J_1 \text{ rs } a]_{\equiv} \in V^{F_1}, \\ \text{y } a, \hat{a} \notin A(\alpha) \cup A(\beta)\}$$

$$v_0^{F_1} = [\overline{F_1}]_{\equiv}$$

Se puede observar que en V^{F_1} tenemos:

$$- \overline{[F_1]}_{\equiv} = \overline{[E_1 rs a]}_{\equiv}$$

$$- [F_1]_{\equiv} = [E_1 rs a]_{\equiv}$$

$$nts(\overline{F_2}) = (V^{F_2} \cup \{[F_2]_{\equiv}\}, A_a^{F_2} \cup A_{sr}^{F_2} \cup A_g^{F_2}, v_0^{F_2})$$

donde

$$V^{F_2} = \{ \overline{[F_2]}_{\equiv} \} \cup \{ [H_2 rs a]_{\equiv} \mid H_2 \in \overline{[E_2]} \text{ y } \exists H \equiv \overline{E_2} \text{ tal que} \\ H \xrightarrow{\langle \alpha_1, r_1 \rangle} \dots \xrightarrow{\langle \alpha_n, r_n \rangle} H_2 \text{ con } a, \hat{a} \notin A(\alpha_i), i = 1, \dots, n \}$$

$$A_a^{F_2} = \{ ([H_2 rs a]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_2 rs a]_{\equiv}) \mid ([H_2]_{\equiv}, \langle \alpha, r \rangle_{n,C}, [J_2]_{\equiv}) \in A_a^{E_2}, \\ [H_2 rs a]_{\equiv}, [J_2 rs a]_{\equiv} \in V^{F_2} \text{ y } a, \hat{a} \notin A(\alpha) \}$$

$$A_{sr}^{F_2} = \{ (\overline{[F_2]}_{\equiv}, \langle skip, 0 \rangle, [F_2]_{\equiv}), ([F_2]_{\equiv}, \langle redo, \infty \rangle, \overline{[F_2]}_{\equiv}) \}$$

$$A_g^{F_2} = \{ ([H_2 rs a]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_2 rs a]_{\equiv}) \mid \\ ([H_2]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_2]_{\equiv}) \in A_g^{E_2}, [H_2 rs a]_{\equiv}, [J_2 rs a]_{\equiv} \in V^{F_2} \\ \text{ y } a, \hat{a} \notin A(\alpha) \cup A(\beta) \}$$

$$v_0^{F_2} = \overline{[F_2]}_{\equiv}$$

$$\text{con } \overline{[F_2]}_{\equiv} = \overline{[E_2 rs a]}_{\equiv} \text{ y } [F_2]_{\equiv} = [E_2 rs a]_{\equiv}.$$

Entonces definimos

$$\varphi : V^{F_1} \cup \{[F_1]_{\equiv}\} \rightarrow V^{F_2} \cup \{[F_2]_{\equiv}\}$$

de la forma siguiente:

- Para cada $[H_1 rs a]_{\equiv} \in V^{F_1}$, tomamos $\varphi([H_1 rs a]_{\equiv}) = [H_2 rs a]_{\equiv}$, con $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$.
- $\varphi([F_1]_{\equiv}) = [F_2]_{\equiv}$.

Podemos observar que φ está bien definida, ya que $[H_2 rs a]_{\equiv} \in V^{F_2}$, lo cual es una consecuencia inmediata de $E_1 \sim E_2$, ya que podemos ejecutar en E_2 la misma

cadena de multiacciones que en E_1 , posiblemente con diferentes razones. Además φ es una biyección y $\varphi([\overline{F_1}]_{\equiv}) = [\overline{F_2}]_{\equiv}$.

Además, para cada $v \in V^{E_1}$ y α tal que $\mathcal{C}(v, \alpha) \neq \emptyset$, iremos definiendo una biyección $g_{v, \alpha} : \mathcal{C}(v, \alpha) \rightarrow \mathcal{C}(\varphi(v), \alpha)$, en función de los distintos casos que vamos a considerar.

- Consideremos un arco cualquiera e de $A_a^{E_1}$:

$$e = ([H_1 \text{ rs } a]_{\equiv}, \langle \alpha, r \rangle_{n, C}, [J_1 \text{ rs } a]_{\equiv})$$

con $([H_1]_{\equiv}, \langle \alpha, r \rangle_{n, C}, [J_1]_{\equiv}) \in A^{E_1}$, $a, \hat{a} \notin A(\alpha)$.

Tomando $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$, $\phi([J_1]_{\equiv}) = [J_2]_{\equiv}$, $h_{[H_1]_{\equiv}, \alpha}(C) = C'$ tenemos que

$$\varphi([H_1 \text{ rs } a]_{\equiv}) = [H_2 \text{ rs } a]_{\equiv} \quad \text{y} \quad \varphi([J_1 \text{ rs } a]_{\equiv}) = [J_2 \text{ rs } a]_{\equiv}$$

y definimos $g_{[H_1 \text{ rs } a]_{\equiv}, \alpha}(C) = C'$, teniéndose que $rc(C) = rc(C')$.

Entonces

$$\begin{aligned} r([H_1 \text{ rs } a]_{\equiv}, \alpha, C, [J_1 \text{ rs } a]_{\equiv}) &= r([H_1]_{\equiv}, \alpha, C, [J_1]_{\equiv}) = \\ r([H_2]_{\equiv}, \alpha, C', [J_2]_{\equiv}) &= r([H_2 \text{ rs } a]_{\equiv}, \alpha, C', [J_2 \text{ rs } a]_{\equiv}) \end{aligned}$$

- Consideremos un arco cualquiera e de $A_g^{E_1}$:

$$e = ([H_1 \text{ rs } a]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1 \text{ rs } a]_{\equiv})$$

con $([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1]_{\equiv}) \in A_g^{E_1}$, $a, \hat{a} \notin A(\alpha) \cup A(\beta)$ y $[H_1 \text{ rs } a]_{\equiv} \in V^{E_1}$.

A partir de $E_1 \sim E_2$, considerando $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$, $\phi([J_1]_{\equiv}) = [J_2]_{\equiv}$, $h_{[H_1]_{\equiv}, \alpha}(C_1) = C'_1$ y $h_{[H_1]_{\equiv}, \beta}(C_2) = C'_2$, obtenemos que :

1. $\exists([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1, C_1}, [H'_1]_{\equiv}) \in A_a^{E_1}$

2. $\exists([H_1]_{\equiv}, \langle \beta, s \rangle_{n_2, C_2}, [H''_1]_{\equiv}) \in A_a^{E_1}$

3. $\exists r', n'_1, s', n'_2$ tales que

$$([H_2]_{\equiv}, \langle \alpha, r' \rangle_{n'_1} \parallel \langle \beta, s' \rangle_{n'_2}, [J_2]_{\equiv}) \in A_g^{E_2} \text{ con:}$$

$$([H_2]_{\equiv}, \langle \alpha, r' \rangle_{n'_1, C'_1}, \phi([H'_1]_{\equiv})) \in A_a^{E_2}$$

$$([H_2]_{\equiv}, \langle \beta, s' \rangle_{n'_2, C'_2}, \phi([H''_1]_{\equiv})) \in A_a^{E_2}$$

Entonces, definiendo $g_{[H_1 \text{ rs } a]_{\equiv}, \alpha}(C_1) = C'_1$ y $g_{[H_1 \text{ rs } a]_{\equiv}, \beta}(C_2) = C'_2$, tenemos que si $([H_1 \text{ rs } a]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1 \text{ rs } a]_{\equiv}) \in A_g^{E_1}$, entonces:

1. $\exists([H_1 rs a]_{\equiv}, \langle \alpha, r \rangle_{n_1, C_1}, [H'_1 rs a]_{\equiv}) \in A_a^{F_1}$
2. $\exists([H_1 rs a]_{\equiv}, \langle \beta, s \rangle_{n_2, C_2}, [H''_1 rs a]_{\equiv}) \in A_a^{F_1}$
3. $\exists r', n'_1, s', n'_2$ tales que
 $([H_2 rs a]_{\equiv}, \langle \alpha, r' \rangle_{n'_1} \parallel \langle \beta, s' \rangle_{n'_2}, [J_2 rs a]_{\equiv}) \in A_g^{F_2}$ con:
 $([H_2 rs a]_{\equiv}, \langle \alpha, r' \rangle_{n'_1, C'_1}, \varphi([H'_1 rs a]_{\equiv})) \in A_a^{F_2}$
 $([H_2 rs a]_{\equiv}, \langle \beta, s' \rangle_{n'_2, C'_2}, \varphi([H''_1 rs a]_{\equiv})) \in A_a^{F_2}$

cumpléndose así las condiciones de la definición 6.7.

(vii) Consideremos

$$F_1 = [E_1 * E' * E] \quad \text{y} \quad F_2 = [E_2 * E' * E]$$

$nts(\overline{F_1})$ y $nts(\overline{F_2})$ pueden obtenerse como sigue:

$$nts(\overline{F_1}) = (V^{F_1} \cup \{[F_1]_{\equiv}\}, A_a^{F_1} \cup A_{sr}^{F_1} \cup A_g^{F_1}, v_0^{F_1})$$

donde

$$V^{F_1} = \{([H_1 * E' * E]_{\equiv} \mid H_1 \in [\overline{E_1}]) \cup \{([E_1 * G' * E]_{\equiv} \mid G' \in [\overline{E'}]) \cup \{([E_1 * E' * G]_{\equiv} \mid G \in [\overline{E}])\}$$

$$A_a^{F_1} = \{([H_1 * E' * E]_{\equiv}, \langle \alpha, r \rangle_{n, C}, [J_1 * E' * E]_{\equiv}) \mid ([H_1]_{\equiv}, \langle \alpha, r \rangle_{n, C}, [J_1]_{\equiv}) \in A_a^{E_1}\} \cup \{([E_1 * G' * E]_{\equiv}, \langle \alpha, r \rangle_{n, C_1}, [E_1 * J' * E]_{\equiv}) \mid ([G']_{\equiv}, \langle \alpha, r \rangle_{n, C_{11}}, [J']_{\equiv}) \in A_a^{E'}\} \cup \{([E_1 * E' * G]_{\equiv}, \langle \alpha, r \rangle_{n, C_2}, [E_1 * E' * J]_{\equiv}) \mid ([G]_{\equiv}, \langle \alpha, r \rangle_{n, C_{21}}, [J]_{\equiv}) \in A_a^E\}$$

siendo C_1 y C_2 de la forma:

- Si $G' \not\equiv \overline{E'}$, entonces $C_1 = C_{11}$.
- Si $G' \equiv \overline{E'}$, entonces:
 $C_1 = C_{11} \cup \{\langle \alpha, r_j \rangle_{n_j} \mid ([\overline{E}]_{\equiv}, \langle \alpha, r_j \rangle_{n_j, C}, [K]_{\equiv}) \in A_a^E\}$
- Si $G \not\equiv \overline{E}$, entonces $C_2 = C_{21}$.

- Si $G \equiv \overline{E}$, entonces:

$$C_2 = C_{21} \cup \{ \langle \alpha, r_j \rangle_{n_j} \mid ([\overline{E'}]_{\equiv}, \langle \alpha, r_j \rangle_{n_j, C}, [K']_{\equiv}) \in A_a^{E'} \}$$

$$A_{sr}^{F_1} = \{ ([\overline{F_1}]_{\equiv}, \langle skip, 0 \rangle, [F_1]_{\equiv}), ([F_1]_{\equiv}, \langle redo, \infty \rangle, [\overline{F_1}]_{\equiv}) \}$$

$$\begin{aligned} A_g^{F_1} = & \{ ([H_1 * E' * E]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1 * E' * E]_{\equiv}) \mid \\ & ([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1]_{\equiv}) \in A_g^{E_1} \} \cup \\ & \{ ([E_1 * G' * E]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [E_1 * J' * E]_{\equiv}) \mid \\ & ([G']_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J']_{\equiv}) \in A_g^{E'} \} \cup \\ & \{ ([E_1 * E' * G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [E_1 * E' * J]_{\equiv}) \mid \\ & ([G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J]_{\equiv}) \in A_g^E \} \end{aligned}$$

$$v_0^{F_1} = [\overline{F_1}]_{\equiv}$$

Podemos observar en la definición de V^{F_1} tenemos que:

- $[[\underline{E_1} * E' * E]_{\equiv}] = [[E_1 * \overline{E'} * E]_{\equiv}] = [[E_1 * \underline{E'} * E]_{\equiv}] = [[E_1 * E' * \overline{E}]_{\equiv}]$,
- $[\overline{F_1}]_{\equiv} = [[\overline{E_1 * E' * E}]_{\equiv}] = [[\overline{E_1} * E' * E]_{\equiv}]$
- $[F_1]_{\equiv} = [[E_1 * E' * E]_{\equiv}] = [[E_1 * E' * \underline{E}]_{\equiv}]$

Análogamente,

$$nts(\overline{F_2}) = (V^{F_2} \cup \{[F_2]_{\equiv}\}, A^{F_2} \cup A_{sr}^{F_2} \cup A_g^{F_2}, v_0^{F_2})$$

donde

$$\begin{aligned} V^{F_2} = & \{ [[H_2 * E' * E]_{\equiv} \mid H_2 \in [\overline{E_2}] \} \cup \{ [[E_2 * G' * E]_{\equiv} \mid G' \in [\overline{E'}] \} \cup \\ & \{ [[E_2 * E' * G]_{\equiv} \mid G \in [\overline{E}] \} \end{aligned}$$

$$\begin{aligned} A_a^{F_2} = & \{ ([H_2 * E' * E]_{\equiv}, \langle \alpha, r \rangle_{n, C}, [J_2 * E' * E]_{\equiv}) \mid \\ & ([H_2]_{\equiv}, \langle \alpha, r \rangle_{n, C}, [J_2]_{\equiv}) \in A^{E_2} \} \cup \\ & \{ ([E_2 * G' * E]_{\equiv}, \langle \alpha, r \rangle_{n, C_1}, [E_2 * J' * E]_{\equiv}) \mid \\ & ([G']_{\equiv}, \langle \alpha, r \rangle_{n, C_{11}}, [J']_{\equiv}) \in A^{E'} \} \cup \\ & \{ ([E_2 * E' * G]_{\equiv}, \langle \alpha, r \rangle_{n, C_2}, [E_2 * E' * J]_{\equiv}) \mid \\ & ([G]_{\equiv}, \langle \alpha, r \rangle_{n, C_{21}}, [J]_{\equiv}) \in A^E \} \end{aligned}$$

siendo C_1 y C_2 de la forma:

- Si $G' \not\equiv \overline{E'}$, entonces $C_1 = C_{11}$.
- Si $G' \equiv \overline{E'}$, entonces:
 $C_1 = C_{11} \cup \{ \langle \alpha, r_j \rangle_{n_j} \mid ([\overline{E'}]_{\equiv}, \langle \alpha, r_j \rangle_{n_j, C}, [K]_{\equiv}) \in A_a^E \}$
- Si $G \not\equiv \overline{E}$, entonces $C_2 = C_{21}$.
- Si $G \equiv \overline{E}$, entonces:
 $C_2 = C_{21} \cup \{ \langle \alpha, r_j \rangle_{n_j} \mid ([\overline{E'}]_{\equiv}, \langle \alpha, r_j \rangle_{n_j, C}, [K']_{\equiv}) \in A_a^{E'} \}$

$$A_{sr}^{F_2} = \{ ([\overline{F_2}]_{\equiv}, \langle skip, 0 \rangle, [F_2]_{\equiv}), ([F_2]_{\equiv}, \langle redo, \infty \rangle, [\overline{F_2}]_{\equiv}) \}$$

$$A_g^{F_2} = \{ ([H_2 * E' * E]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_2 * E' * E]_{\equiv}) \mid ([H_2]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_2]_{\equiv}) \in A_g^{E_2} \} \cup \{ ([E_2 * G' * E]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [E_2 * J' * E]_{\equiv}) \mid ([G']_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J']_{\equiv}) \in A_g^{E'} \} \cup \{ ([E_2 * E' * G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [E_2 * E' * J]_{\equiv}) \mid ([G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J]_{\equiv}) \in A_g^E \}$$

$$v_0^{F_2} = [\overline{F_2}]_{\equiv}$$

Por otra parte, en la definición de V^{F_2} tenemos que:

- $[[E_2 * E' * E]_{\equiv}] = [[E_2 * \overline{E'} * E]_{\equiv}] = [[E_2 * \underline{E'} * E]_{\equiv}] = [[E_2 * E' * \overline{E}]_{\equiv}]$,
- $[\overline{F_2}]_{\equiv} = [\overline{[E_2 * E' * E]}]_{\equiv} = [\overline{E_2} * E' * E]_{\equiv}$
- $[F_2]_{\equiv} = [E_2 * E' * \underline{E}]_{\equiv} = [E_2 * E' * E]_{\equiv}$

Entonces, definimos

$$\varphi : V^{F_1} \cup \{[F_1]_{\equiv}\} \rightarrow V^{F_2} \cup \{[F_2]_{\equiv}\}$$

de la forma siguiente:

- Si $[H_1]_{\equiv} \in V^{E_1}$ y $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$, entonces $\varphi([H_1 * E' * E]_{\equiv}) = [H_2 * E' * E]_{\equiv}$
- $\varphi([E_1 * G' * E]_{\equiv}) = [E_2 * G' * E]_{\equiv}$, para cada $G' \in [\overline{E'}]$,
- $\varphi([E_1 * E' * G]_{\equiv}) = [E_2 * E' * G]_{\equiv}$, para cada $G \in [\overline{E}]$,

$$- \varphi([F_1]_{\equiv}) = [F_2]_{\equiv}$$

Es inmediato comprobar que φ está bien definida, y que además es una biyección que verifica $\varphi([\overline{F_1}]_{\equiv}) = [\overline{F_2}]_{\equiv}$.

Además para cada $v \in V^{F_1}$ y α tal que $\mathcal{C}(v, \alpha) \neq \emptyset$, iremos definiendo una biyección $g_{v, \alpha} : \mathcal{C}(v, \alpha) \rightarrow \mathcal{C}(\varphi(v), \alpha)$, en función de los distintos casos que vamos a considerar.

- Consideremos ahora un arco cualquiera e de $A_a^{F_1}$. Nos encontraremos en alguno de los casos siguientes:

$$- e = ([H_1 * E' * E]_{\equiv}, \langle \alpha, r \rangle_{n, C}, [J_1 * E' * E]_{\equiv}),$$

$$\text{con } ([H_1]_{\equiv}, \langle \alpha, r \rangle_{n, C}, [J_1]_{\equiv}) \in A_a^{E_1}.$$

Tomando $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$, $\phi([J_1]_{\equiv}) = [J_2]_{\equiv}$ y $h_{[H_1]_{\equiv}, \alpha}(C) = C'$ tenemos que

$$\varphi([H_1 * E' * E]_{\equiv}) = [H_2 * E' * E]_{\equiv}$$

$$\varphi([J_1 * E' * E]_{\equiv}) = [J_2 * E' * E]_{\equiv}$$

y definimos $g_{[[H_1 * E' * E]_{\equiv}, \alpha]}(C) = C'$, teniéndose que $rc(C) = rc(C')$.

Como $E_1 \sim E_2$, obtenemos:

$$r([H_1 * E' * E]_{\equiv}, \alpha, C, [J_1 * E' * E]_{\equiv}) = r([H_1]_{\equiv}, \alpha, C, [J_1]_{\equiv}) =$$

$$r([H_2]_{\equiv}, \alpha, C', [J_2]_{\equiv}) = r([H_2 * E' * E]_{\equiv}, \alpha, C', [J_2 * E' * E]_{\equiv})$$

$$- e = ([E_1 * G' * E]_{\equiv}, \langle \alpha, r \rangle_{n, C_1}, [E_1 * J' * E]_{\equiv})$$

$$\text{con } ([G']_{\equiv}, \langle \alpha, r \rangle_{n, C_{11}}, [J']_{\equiv}) \in A^{E'} \text{ y } G' \neq \overline{E'}.$$

El razonamiento es similar al considerado en el caso anterior, pues en este caso tenemos que $C_1 = C_{11}$. Considerando

$$\varphi([E_1 * G' * E]_{\equiv}) = [E_2 * G' * E]_{\equiv}$$

$$\varphi([E_1 * J' * E]_{\equiv}) = [E_2 * J' * E]_{\equiv}$$

y definiendo $g_{[[E_1 * G' * E]_{\equiv}, \alpha]}(C_1) = C_1$ (la identidad), llegamos a que:

$$r([E_1 * G' * E]_{\equiv}, \alpha, C_1, [E_1 * J' * E]_{\equiv}) = r([G']_{\equiv}, \alpha, C_{11}, [J']_{\equiv}) =$$

$$r([E_2 * G' * E]_{\equiv}, \alpha, C_1, [E_2 * J' * E]_{\equiv})$$

- $e = ([E_1 * G' * E]_{\equiv}, \langle \alpha, r \rangle_{n, C_1}, [E_1 * J' * E]_{\equiv})$
con $([G']_{\equiv}, \langle \alpha, r \rangle_{n, C_{11}}, [J']_{\equiv}) \in A^{E'}$ y $G' \equiv \overline{E'}$.

donde considerando

$$C = \{ \langle \alpha, r_j \rangle_{n_j} \mid ([\overline{E}]_{\equiv}, \langle \alpha, r_j \rangle_{n_j, C}, [K]_{\equiv}) \in A_a^E \}$$

se tiene que $C_1 = C \cup C_{11}$. Obsérvese que por la restricción sintáctica impuesta, si C es distinto de \emptyset , entonces C es el único conjunto de conflictos de $\mathcal{C}([\overline{E}]_{\equiv}, \alpha)$.

El razonamiento es similar al considerado en el caso anterior. Considerando:

$$\begin{aligned} \varphi([E_1 * G' * E]_{\equiv}) &= [E_2 * G' * E]_{\equiv} \\ \varphi([E_1 * J' * E]_{\equiv}) &= [E_2 * J' * E]_{\equiv} \end{aligned}$$

y definiendo $g_{[E_1 * G' * E]_{\equiv}, \alpha}(C_1) = C_1$ (la identidad), tenemos que:

$$\begin{aligned} r([E_1 * G' * E]_{\equiv}, \alpha, C_1, [E_1 * J' * E]_{\equiv}) &= r([G']_{\equiv}, \alpha, C_{11}, [J']_{\equiv}) = \\ r([E_2 * G' * E]_{\equiv}, \alpha, C_1, [E_2 * J' * E]_{\equiv}) & \end{aligned}$$

donde en la igualdad anterior se ha tenido en cuenta que

$$\nexists \langle \alpha, r_j \rangle_{n_j} \in C$$

de forma que

$$([E_1 * G' * E]_{\equiv}, \langle \alpha, r_j \rangle_{n_j, C}, [E_1 * J' * E]_{\equiv}) \in A_a^{E_1}$$

pues ello implicaría necesariamente

$$([\overline{E}]_{\equiv}, \langle \alpha, r_j \rangle_{n_j, C}, [\overline{E}]_{\equiv}) \in A_a^E$$

lo cual es imposible de acuerdo a nuestra semántica operacional, por lo cual $r([E_1 * G' * E]_{\equiv}, \alpha, C_1, [E_1 * J' * E]_{\equiv}) = r([G']_{\equiv}, \alpha, C_{11}, [J']_{\equiv})$.

- $e = ([E_1 * E' * G]_{\equiv}, \langle \alpha, r \rangle_{n, C_2}, [E_1 * E' * J]_{\equiv})$
con $([G]_{\equiv}, \langle \alpha, r \rangle_{n, C_{21}}, [J]_{\equiv}) \in A_a^E$ y $G \not\equiv \overline{E}$.

El razonamiento es similar al segundo de los casos considerados, ya que $C_2 = C_{21}$. Tomando:

$$\begin{aligned} \varphi([E_1 * E' * G]_{\equiv}) &= [E_2 * E' * G]_{\equiv} \\ \varphi([E_1 * E' * J]_{\equiv}) &= [E_2 * E' * J]_{\equiv} \end{aligned}$$

y definiendo $g_{[[E_1 * E' * G]]_{\equiv}, \alpha}(C_2) = C_2$, tenemos que:

$$r([[E_1 * E' * G]]_{\equiv}, \alpha, C_2, [[E_1 * E' * J]]_{\equiv}) = r([G]_{\equiv}, \alpha, C_{21}, [J]_{\equiv}) = r([[E_2 * E' * G]]_{\equiv}, \alpha, C_2, [[E_2 * E' * J]]_{\equiv})$$

- $e = ([[E_1 * E' * G]]_{\equiv}, \langle \alpha, r \rangle_{n, C_2}, [[E_1 * E' * J]]_{\equiv})$
con $([G]_{\equiv}, \langle \alpha, r \rangle_{n, C_{21}}, [J]_{\equiv}) \in A_a^E$ y $G \equiv \bar{E}$.

donde considerando

$$C = \{ \langle \alpha, r_j \rangle_{n_j} \mid ([\bar{E}']_{\equiv}, \langle \alpha, r_j \rangle_{n_j, C}, [K']_{\equiv}) \in A_a^{E'} \}$$

se tiene que $C_2 = C \cup C_{21}$. Obsérvese que por la restricción sintáctica impuesta, si C es distinto de \emptyset , entonces C es el único conjunto de conflictos de $\mathcal{C}([\bar{E}']_{\equiv}, \alpha)$.

El razonamiento es similar al tercero de los casos considerados. Tomando:

$$\begin{aligned} \varphi([[E_1 * E' * G]]_{\equiv}) &= [[E_2 * E' * G]]_{\equiv} \\ \varphi([[E_1 * E' * J]]_{\equiv}) &= [[E_2 * E' * J]]_{\equiv} \end{aligned}$$

y definiendo $g_{[[E_1 * E' * G]]_{\equiv}, \alpha}(C_2) = C_2$, tenemos que:

$$r([[E_1 * E' * G]]_{\equiv}, \alpha, C_2, [[E_1 * E' * J]]_{\equiv}) = r([G]_{\equiv}, \alpha, C_{21}, [J]_{\equiv}) = r([[E_2 * E' * G]]_{\equiv}, \alpha, C_2, [[E_2 * E' * J]]_{\equiv})$$

- Consideremos ahora un arco cualquiera e de $A_g^{F_1}$. De nuevo tenemos tres casos:

- $e = ([[H_1 * E' * E]]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [[J_1 * E' * E]]_{\equiv})$
con $([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1]_{\equiv}) \in A_g^{E_1}$.

Tomando $\phi([H_1]_{\equiv}) = [H_2]_{\equiv}$, $\phi([J_1]_{\equiv}) = [J_2]_{\equiv}$, $h_{[H_1]_{\equiv}, \alpha}(C_1) = C'_1$ y $h_{[H_1]_{\equiv}, \beta}(C_2) = C'_2$ tenemos que

$$\begin{aligned} \varphi([[H_1 * E' * E]]_{\equiv}) &= [[H_2 * E' * E]]_{\equiv} \\ \varphi([[J_1 * E' * E]]_{\equiv}) &= [[J_2 * E' * E]]_{\equiv} \end{aligned}$$

y definimos $g_{[[H_1 * E' * E]]_{\equiv}, \alpha}(C_1) = C'_1$ y $g_{[[H_1 * E' * E]]_{\equiv}, \beta}(C_2) = C'_2$.

Además, si $([[H_1 * E' * E]]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [[J_1 * E' * E]]_{\equiv}) \in A_g^{F_1}$ con $([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J_1]_{\equiv}) \in A_g^{E_1}$, entonces, como $E_1 \sim E_2$, llegamos a:

1. $\exists([H_1]_{\equiv}, \langle \alpha, r \rangle_{n_1, C_1}, [H'_1]_{\equiv}) \in A_a^{E_1}$

2. $\exists([H_1]_{\equiv}, \langle \beta, s \rangle_{n_2, C_2}, [H_1'']_{\equiv}) \in A_a^{E_1}$
3. $\exists r', s', n'_1, n'_2$ tales que $([H_2]_{\equiv}, \langle \alpha, r' \rangle_{n'_1} \parallel \langle \beta, s' \rangle_{n'_2}, [J_2]_{\equiv}) \in A_g^{E_2}$,
con $([H_2]_{\equiv}, \langle \alpha, r' \rangle_{n'_1, C'_1}, \phi([H_1']_{\equiv})) \in A_a^{E_2}$ y
 $([H_2]_{\equiv}, \langle \beta, s' \rangle_{n'_2, C'_2}, \phi([H_1'']_{\equiv})) \in A_a^{E_2}$.

Entonces, concluimos que:

1. $\exists([H_1 * E' * E]_{\equiv}, \langle \alpha, r \rangle_{n_1, C_1}, [[H_1' * E' * E]_{\equiv}]) \in A_a^{F_1}$
 2. $\exists([H_1 * E' * E]_{\equiv}, \langle \beta, s \rangle_{n_2, C_2}, [[H_1'' * E' * E]_{\equiv}]) \in A_a^{F_1}$
 3. $\exists r', s', n'_1, n'_2$ tales que
 $([H_2 * E' * E]_{\equiv}, \langle \alpha, r' \rangle_{n'_1} \parallel \langle \beta, s' \rangle_{n'_2}, [[J_2 * E' * E]_{\equiv}]) \in A_g^{F_2}$,
con $([H_2 * E' * E]_{\equiv}, \langle \alpha, r' \rangle_{n'_1, C'_1}, \varphi([H_1' * E' * E]_{\equiv})) \in A_a^{F_2}$, y
 $([H_2 * E' * E]_{\equiv}, \langle \beta, s' \rangle_{n'_2, C'_2}, \varphi([H_1'' * E' * E]_{\equiv})) \in A_a^{F_2}$.
- $e = ([E_1 * G' * E]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [[E_1 * J' * E]_{\equiv}])$
con $([G']_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J']_{\equiv}) \in A_g^{E'}$.

Este caso es análogo al anterior.

- $e = ([E_1 * E' * G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [[E_1 * E' * J]_{\equiv}])$
con $([G]_{\equiv}, \langle \alpha, r \rangle_{n_1} \parallel \langle \beta, s \rangle_{n_2}, [J]_{\equiv}) \in A_g^E$.

Este caso es también análogo a los dos anteriores.

(viii) Consideremos

$$F_1 = [E' * E_1 * E] \quad \text{y} \quad F_2 = [E' * E_2 * E]$$

La comprobación de que $F_1 \sim F_2$ es similar al caso anterior.

(ix) Consideremos

$$F_1 = [E' * E * E_1] \quad \text{y} \quad F_2 = [E' * E * E_2]$$

La comprobación de que $F_1 \sim F_2$ es similar a los dos casos anteriores.

□

6.3 Bisimulación estocástica

En este apartado tan solo planteamos las líneas de trabajo futuro que se desarrollarán tras la presentación de esta tesis, referidas al campo de las bisimulaciones estocásticas. Se parte de la propuesta de bisimulación probabilística de Larsen y Skou ([LS91]). La idea es definir una relación de equivalencia sobre los nodos del nuevo sistema de transiciones que hemos definido. Dicha relación de equivalencia nos proporcionará una partición sobre dicho conjunto de nodos. A la relación de equivalencia le exigiremos que el proceso estocástico asociado sea también una CTMC, es decir buscaremos la condición de *lumpabilidad* [KS60]. Una vez obtenida la partición, definiremos el sistema de transiciones agregado asociado a la s-expresión con respecto a dicha relación.

Finalmente, dadas dos s-expresiones estáticas diremos que son estocásticamente bisimilares si podemos obtener en cada una sendas relaciones de equivalencia, de forma que los respectivos sistemas de transiciones agregados sean isomorfos. Como trabajo futuro está el probar que esta bisimulación estocástica sería una congruencia.

Definición 6.8 Sea E una s-expresión estática regular, con

$$nts(\overline{E}) = (V^E, A_a^E \cup A_g^E \cup A_{sr}^E, v_0^E)$$

Consideramos una partición *lumpable* sobre el conjunto de nodos, en el sentido de que $\forall \alpha \in \mathcal{L}$, si para cualesquiera V_1, V_2 clases originadas por la partición, y $[G_1]_{\equiv}, [G_2]_{\equiv} \in V_1$, entonces existe una biyección:

$$\psi : \mathcal{C}([G_1]_{\equiv}, \alpha) \rightarrow \mathcal{C}([G_2]_{\equiv}, \alpha)$$

tal que $cr(C_1) = cr(\psi(C_1))$, $\forall C_1 \in \mathcal{C}([G_1]_{\equiv}, \alpha)$, de forma que:

$$r([G_1]_{\equiv}, \alpha, C_1, V_2) = r([G_2]_{\equiv}, \alpha, \psi(C_1), V_2)$$

donde hemos considerado que dados $[G]_{\equiv} \in V^E$, C un conjunto de conflictos y V una clase originada por la partición se define:

$$r([G]_{\equiv}, \alpha, C, V) = \sum_{[G_k]_{\equiv} \in V} r([G]_{\equiv}, \alpha, C, [G_k]_{\equiv})$$

Esta partición da lugar a una relación de equivalencia \mathcal{R} (aunque también hubiéramos podido partir primero de la relación de equivalencia, que hubiera generado dicha partición).

En el *sistema de transiciones agregado* de \overline{E} obtenido a partir de la relación \mathcal{R} y de $nts(\overline{E})$, tendremos como nodos las clases de equivalencia originadas por la partición, y en los arcos se anotarán las multiacciones, las razones de paso y las razones de conflicto. Lo denotaremos por $ats^{\mathcal{R}}(\overline{E})$, y se define de la forma:

$$ats^{\mathcal{R}}(\overline{E}) = (V^{E_{\mathcal{R}}}, A_a^{E_{\mathcal{R}}} \cup A_g^{E_{\mathcal{R}}} \cup A_{sr}^{E_{\mathcal{R}}}, v_0^{E_{\mathcal{R}}})$$

donde:

- $V^{E_{\mathcal{R}}} = V^E / \mathcal{R}$
- Para definir $A_a^{E_{\mathcal{R}}}$ tomaremos de cada clase $V_i \in V^{E_{\mathcal{R}}}$ un representante, que llamaremos $[G_i]_{\equiv}$, de modo que para cada par de nodos $V_1, V_2 \in V^{E_{\mathcal{R}}}$, para cada α , y cada conjunto de conflictos $C_1 \in \mathcal{C}([G_1]_{\equiv}, \alpha)$, diremos que

$$(V_1, \langle \alpha, R_1 \rangle_{cr_1}, V_2) \in A_a^{E_{\mathcal{R}}}$$

si existe un arco $([G_1]_{\equiv}, \langle \alpha, r_1 \rangle_{n_1, C_1}, [J_1]_{\equiv}) \in A_a^E$, con $[G_1]_{\equiv} \in V_1$, para algún $[J_1]_{\equiv} \in V_2$, de modo que:

- $R_1 = r([G_1]_{\equiv}, \alpha, C_1, V_2)$
- $cr_1 = cr(C_1)$.

Obsérvese que $[G_1]_{\equiv}$ es el representante elegido de V_1 , y que debido a la definición de partición lumpable considerada, cualquier representante elegido nos habría dado el mismo resultado, es decir, si hubiésemos tomado $[G'_1]_{\equiv} \in V_1$, y un arco $([G'_1]_{\equiv}, \langle \alpha, r'_1 \rangle_{n'_1, C'_1}, [J'_1]_{\equiv}) \in A_a^E$, entonces tendríamos que $\psi^{-1}(C'_1) \in \mathcal{C}([G_1]_{\equiv}, \alpha)$, siendo ψ la biyección que nos proporciona la lumpabilidad, y además:

- $r([G_1]_{\equiv}, \alpha, \psi^{-1}(C'_1), V_2) = r([G'_1]_{\equiv}, \alpha, C'_1, V_2)$
- $cr(C'_1) = cr(\psi^{-1}(C'_1))$.

En conclusión, podemos elegir cualquier representante para esta definición.

- Para definir $A_g^{E_{\mathcal{R}}}$ consideraremos de nuevo un representante $[G_i]_{\equiv}$ de cada clase $V_i \in V^{E_{\mathcal{R}}}$, de modo que si para $[G_1]_{\equiv} \in V_1$ existe un arco

$$([G_1]_{\equiv}, \langle \alpha_1, r_1 \rangle_{n_1} \parallel \langle \alpha_2, r_2 \rangle_{n_2}, [J_2]_{\equiv}) \in A_g^E,$$

para algún $[J_2]_{\equiv} \in V^2$, con

1. $([G_1]_{\equiv}, \langle \alpha_1, r_1 \rangle_{n_1, C_1}, [G'_i]_{\equiv}) \in A_a^E$
2. $([G_1]_{\equiv}, \langle \alpha_2, r_2 \rangle_{n_2, C_2}, [G''_i]_{\equiv}) \in A_a^E$

entonces tomaremos un arco (de nuevo asociado al par (C_1, C_2)):

$$(V_1, \langle \alpha_1, R_1 \rangle_{cr_1} \parallel \langle \alpha_2, R_2 \rangle_{cr_2}, V_2) \in A_a^{E_{\mathcal{R}}}$$

con

$$\begin{aligned} R_1 &= r([G_1]_{\equiv}, \alpha_1, C_1, V'_i) \text{ donde } [G'_i]_{\equiv} \in V'_i, \quad cr_1 = cr(C_1) \\ R_2 &= r([G_1]_{\equiv}, \alpha_2, C_2, V''_i) \text{ donde } [G''_i]_{\equiv} \in V''_i, \quad cr_2 = cr(C_2) \end{aligned}$$

- $A_{sr}^{E_{\mathcal{R}}} = \{(V_0, \langle skip, 0 \rangle, V_x), (V_x, \langle redo, \infty \rangle, V_0) \mid [\overline{E}]_{\equiv} \in V_0, [E]_{\equiv} \in V_x\}$
- $v_0^{E_{\mathcal{R}}} = \{V_0 \mid [\overline{E}]_{\equiv} \in V_0\}$ □

Definición 6.9 Dadas dos s-expresiones estáticas regulares cualesquiera E y F , diremos que son *estocásticamente bisimilares*, y lo denotaremos por $E \approx F$, si es posible obtener sendos sistemas de transiciones agregados $ats^{\mathcal{R}}(\overline{E})$ y $ats^{\mathcal{R}'}(\overline{F})$ que sean isomorfos. □

Ejemplo 6.7 Consideremos las s-expresiones estáticas:

$$\begin{aligned} E_{10} &= \langle a, 1 \rangle; \langle b, 3 \rangle \square \langle a, 2 \rangle; \langle b, 3 \rangle \\ F_{10} &= \langle a, 3 \rangle; \langle b, 3 \rangle \end{aligned}$$

El $nts(\overline{E}_{10})$ es el mostrado en la figura 6.11,

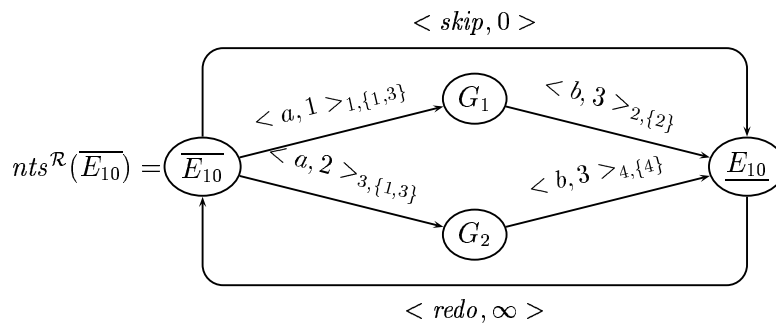


Figura 6.11: $nts(\overline{E}_{10})$

donde hemos considerado:

$$\begin{aligned} G_1 &= \langle a, 1 \rangle_1; \overline{\langle b, 3 \rangle}_2 \square \langle a, 2 \rangle_3; \langle b, 3 \rangle_4 \\ G_2 &= \langle a, 1 \rangle_1; \langle b, 3 \rangle_2 \square \langle a, 2 \rangle_3; \overline{\langle b, 3 \rangle}_4 \end{aligned}$$

Si consideramos la partición dada por

$$V_0 = \{[\overline{E_{10}}]_{\equiv}\}, \quad V_1 = \{[G_1]_{\equiv}, [G_2]_{\equiv}\}, \quad V_x = \{[\underline{E_{10}}]_{\equiv}\}$$

genera la relación \mathcal{R} con

$$V^{E_{10}}/\mathcal{R} = \{V_0, V_1, V_x\},$$

que cumple las condiciones de la definición 6.8, por lo que el sistema de transiciones agregado para E_{10} según esa partición es el mostrado en la figura 6.12.

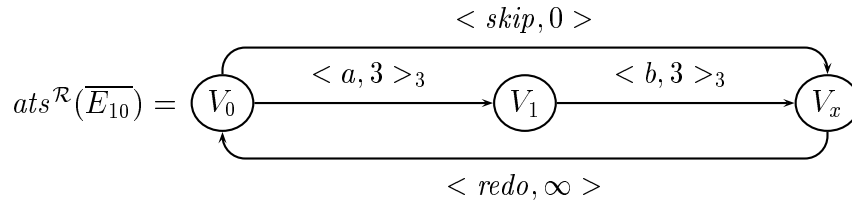


Figura 6.12: $ats^{\mathcal{R}}(\overline{E_{10}})$

Por su parte, para F_{10} si tomamos

$$H_1 = \langle a, 3 \rangle; \overline{\langle b, 3 \rangle}$$

y la partición trivial dada por

$$V'_0 = \{[\overline{F_{10}}]_{\equiv}\}, \quad V'_1 = \{[H_1]_{\equiv}\}, \quad V'_x = \{[\underline{F_{10}}]_{\equiv}\}$$

se genera ahora una relación \mathcal{R}' con

$$V^{F_{10}}/\mathcal{R}' = \{V'_0, V'_1, V'_x\},$$

Es fácil ver que el nuevo sistema de transiciones agregado de F_{10} es isomorfo al de E_{10} , ya que coincide con él una vez sustituidos cada nodo V_i por el correspondiente V'_i , para $i = 0, 1, x$, de modo que claramente son isomorfos, concluyéndose $E_{10} \approx F_{10}$. \square

Ejemplo 6.8 Consideremos las s-expresiones estáticas:

$$\begin{aligned}
 E_{11} &= \langle a, 1 \rangle; (\langle b, 3 \rangle \parallel \langle \widehat{b}, 2 \rangle) \square \langle a, 2 \rangle; (\langle b, 3 \rangle \parallel \langle \widehat{b}, 2 \rangle) \\
 F_{11} &= \langle a, 3 \rangle; ((\langle b, 1 \rangle \square \langle b, 2 \rangle) \parallel \langle \widehat{b}, 2 \rangle)
 \end{aligned}$$

Entonces, tomando para E_{11} :

$$\begin{aligned} G_1 &= \langle a, 1 \rangle; (\overline{\langle b, 3 \rangle} \parallel \overline{\langle \hat{b}, 2 \rangle}) \square \langle a, 2 \rangle; (\langle b, 3 \rangle \parallel \langle \hat{b}, 2 \rangle) \\ G_2 &= \langle a, 1 \rangle; (\langle b, 3 \rangle \parallel \langle \hat{b}, 2 \rangle) \square \langle a, 2 \rangle; (\overline{\langle b, 3 \rangle} \parallel \overline{\langle \hat{b}, 2 \rangle}) \\ G_3 &= \langle a, 1 \rangle; (\overline{\langle b, 3 \rangle} \parallel \overline{\langle \hat{b}, 2 \rangle}) \square \langle a, 2 \rangle; (\langle b, 3 \rangle \parallel \langle \hat{b}, 2 \rangle) \\ G_4 &= \langle a, 1 \rangle; (\overline{\langle b, 3 \rangle} \parallel \langle \hat{b}, 2 \rangle) \square \langle a, 2 \rangle; (\langle b, 3 \rangle \parallel \langle \hat{b}, 2 \rangle) \\ G_5 &= \langle a, 1 \rangle; (\langle b, 3 \rangle \parallel \langle \hat{b}, 2 \rangle) \square \langle a, 2 \rangle; (\overline{\langle b, 3 \rangle} \parallel \overline{\langle \hat{b}, 2 \rangle}) \\ G_6 &= \langle a, 1 \rangle; (\langle b, 3 \rangle \parallel \langle \hat{b}, 2 \rangle) \square \langle a, 2 \rangle; (\overline{\langle b, 3 \rangle} \parallel \overline{\langle \hat{b}, 2 \rangle}) \end{aligned}$$

la partición dada por:

$$\begin{aligned} V_0 &= \{[\overline{E_{11}}]_{\equiv}\}, & V_1 &= \{[G_1]_{\equiv}, [G_2]_{\equiv}\}, & V_2 &= \{[G_3]_{\equiv}, [G_5]_{\equiv}\} \\ V_3 &= \{[G_4]_{\equiv}, [G_6]_{\equiv}\}, & V_x &= \{[E_{11}]_{\equiv}\}, \end{aligned}$$

genera la relación \mathcal{R} con

$$V^{E_{11}}/\mathcal{R} = \{V_0, V_1, V_2, V_3, V_x\},$$

la cual cumple las condiciones de la definición 6.8, y el sistema de transiciones agregado para E_{11} según esa partición, es el mostrado en la figura 6.13.

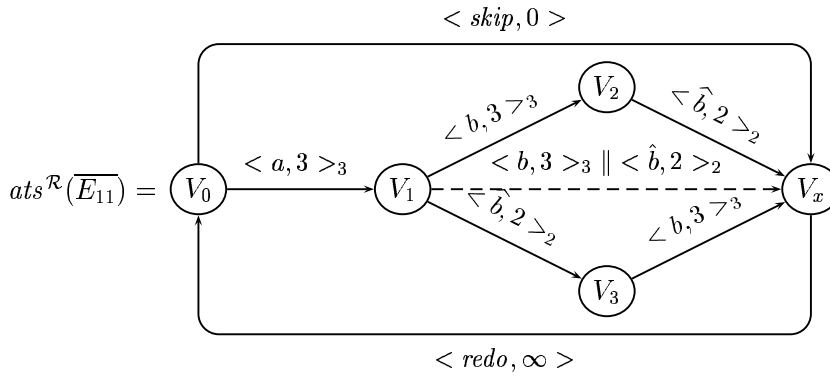


Figura 6.13: $ats^{\mathcal{R}}(\overline{E_{11}})$

Para F_{11} tomando:

$$\begin{aligned} H_1 &= \langle a, 3 \rangle; ((\overline{\langle b, 1 \rangle} \square \langle b, 2 \rangle) \parallel \overline{\langle \hat{b}, 2 \rangle}) \\ H_2 &= \langle a, 3 \rangle; ((\langle b, 1 \rangle \square \langle b, 2 \rangle) \parallel \overline{\langle \hat{b}, 2 \rangle}) \\ H_3 &= \langle a, 3 \rangle; ((\overline{\langle b, 1 \rangle} \square \langle b, 2 \rangle) \parallel \langle \hat{b}, 2 \rangle) \end{aligned}$$

la partición sería la obtenida a partir de la relación identidad,

$$V'_0 = \{[\overline{F_{11}}]_{\equiv}\}, \quad V'_1 = \{[H_1]_{\equiv}\}, \quad V'_2 = \{[H_2]_{\equiv}\}, \quad V'_3 = \{[H_3]_{\equiv}\}, \quad V'_x = \{[F_{11}]_{\equiv}\}$$

y así se generaría

$$V^{F_{11}}/\mathcal{R}' = \{V'_0, V'_1, V'_2, V'_3, V'_x\}$$

Es fácil ver que $ats^{\mathcal{R}'}(\overline{F_{11}})$ es el mismo $ats^{\mathcal{R}}(\overline{E_{11}})$, con sólo cambiar cada V_i por la correspondiente V'_i , para $i = 0, 1, 2, 3, x$, con lo cual queda probado que $E_{11} \approx F_{11}$. \square

Proposición 6.3 Para todas las s-expresiones estáticas regulares E, F , se tiene

$$E \sim F \Rightarrow E \approx F$$

Demostración: Es inmediata; basta considerar en cada caso la relación de identidad. \square

Somos conscientes de que esta bisimulación estocástica está todavía en "pañales" y necesita madurar bastante, sobre todo a la hora de encontrar la relación que nos proporcione el sistema de transiciones agregado que nos interese (claramente existen varias posibilidades, y en particular siempre se tiene la relación identidad). Por ejemplo, sería interesante obtener un algoritmo que generase el sistema de transiciones agregado con el menor número de nodos, que además de facilitar el estudio de la equivalencia, nos abriría una puerta para abordar el problema de la explosión de estados. Nuestro primer objetivo como trabajo futuro más inmediato, será encontrar estos sistemas de transiciones agregados *minimales*, y a partir de ellos considerar el isomorfismo entre sistemas de transiciones para llegar a \approx , y demostrar que \approx es conservada por todos los operadores, es decir que es una congruencia.

Capítulo 7

Casos de estudio

Este capítulo está dedicado al estudio de tres casos *clásicos* en el campo del diseño de sistemas concurrentes, según la perspectiva de nuestro modelo. Con ello se pretende poner de manifiesto la potencia, elegancia y flexibilidad del mismo.

El primer caso a tratar será el de *la región crítica*, para el cual plantearemos dos enfoques: el centralizado y el anillo lógico. En el siguiente apartado estudiaremos el problema del *productor/consumidor*, considerando primero un almacén con capacidad para un solo ítem, para luego generalizarlo a capacidad n . Por último, terminaremos con el *protocolo del Bit Alternante*.

En todos los casos vamos a suponer que nos encontramos bajo una perspectiva markoviana, así que cada multiacción α tendrá asociada una espera (*delay*) para ejecutarse, que seguirá una distribución exponencial negativa de parámetro λ . Dentro de una misma especificación admitiremos tener que una misma multiacción tenga asociados diferentes parámetros en las instancias de la misma.

7.1 Región crítica

Como ya hemos comentado, éste es un ejemplo clásico en el estudio de sistemas concurrentes. Su descripción, sin tener en cuenta tiempos puede encontrarse en [PS89], y una introducción detallada, así como su especificación utilizando un álgebra de procesos con probabilidades, puede encontrarse en [Núñ96]; sobre esta última nos hemos basado para hacer nuestra descripción:

Se tiene un conjunto de procesos. Cada uno de ellos puede ejecutar una tarea propia, o una tarea correspondiente a su región crítica, de modo que sólo uno de los procesos puede ejecutar al mismo tiempo una de éstas. Si dos ó más procesos solicitan entrar en la región crítica a la vez, el sistema tendrá que decidir qué proceso entra, garantizando siempre la deseada exclusión mutua.

7.1.1 Enfoque centralizado

Consideraremos un sistema que cuenta con n procesos ejecutándose en paralelo. Estos procesos se encuentran realizando independientemente su cometido, pero en ocasiones tienen que entrar en su región crítica. Con el enfoque centralizado, se tendrá un proceso auxiliar central que se encargará de otorgar los permisos para entrar en la región crítica, a los procesos que los soliciten. Consideraremos:

- P_i con $i = 1, 2, \dots, n$, los procesos que se encargan de hacer las tareas.
- Q el proceso (centralizador), que se encarga de coordinar la entrada a la región crítica.

Los convenios que se siguen son los siguientes:

1. Las tareas que realiza cada proceso P_i fuera de su región crítica vendrán descritas mediante la ejecución de $\langle t_i, \lambda_{t_i} \rangle$.
2. $\langle r_i, \lambda_{r_i} \rangle$ indicará el trabajo que cada proceso P_i realiza en la región crítica.
3. El centralizador no tiene ninguna preferencia por ninguno de los procesos, a la hora de permitirles el acceso a la región crítica.
4. El proceso P_i realiza ciclos en los que en cada ocasión tendrá la posibilidad de realizar una tarea propia $\langle t_i, \lambda_{t_i} \rangle$ y volver al punto de partida, o bien solicitará permiso para entrar en su región crítica, $\langle p_i, \lambda_{p_i} \rangle$. Cuando obtenga el permiso, entrará en la región crítica $\langle e, \lambda_{e_i} \rangle$, realizará (una sola vez) su trabajo, $\langle r_i, \lambda_{r_i} \rangle$, y luego saldrá de la región crítica, $\langle s, \lambda_{s_i} \rangle$, volviendo al inicio del ciclo.
5. El proceso centralizador Q realiza ciclos que consisten en controlar la entrada en la región crítica $\langle \hat{e}, \lambda_{\hat{e}} \rangle$, y la salida $\langle \hat{s}, \lambda_{\hat{s}} \rangle$. Estas acciones tienen

que sincronizarse con las correspondientes de los procesos P_j , para asegurar la estancia de un único proceso en la región crítica.

6. Los procesos P_j y Q realizan antes de empezar sus ciclos unas tareas de iniciación $\langle i_{P_j}, \lambda_{i_{P_j}} \rangle$ y $\langle i_Q, \lambda_{i_Q} \rangle$, respectivamente. Esas tareas son independientes y no necesitan sincronizarse.
7. El sistema describe un proceso de comportamiento infinito.

Podemos modelar con sPBC el sistema de la forma siguiente:

$$P_j = [\langle i_{P_j}, \lambda_{i_{P_j}} \rangle * \langle t_j, \lambda_{t_j} \rangle \square (\langle p_j, \lambda_{p_j} \rangle ; \langle e, \lambda_{e_j} \rangle ; \langle r_j, \lambda_{r_j} \rangle ; \langle s, \lambda_{s_j} \rangle) * \langle f, \lambda_f \rangle] rs f$$

para $j = 1, \dots, n$.

Con lo que la s-caja plana para cada P_j sería la mostrada en la figura 7.1.

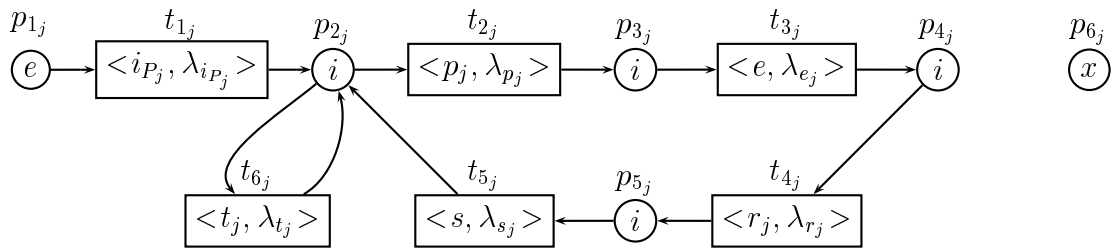


Figura 7.1: s-caja plana de P_j

La especificación de Q sería:

$$Q = [\langle i_Q, \lambda_{i_Q} \rangle * (\langle \hat{e}, \lambda_e \rangle ; \langle \hat{s}, \lambda_s \rangle) * \langle f, \lambda_f \rangle] rs f$$

con lo que la s-caja plana de Q sería la mostrada en la figura 7.2.

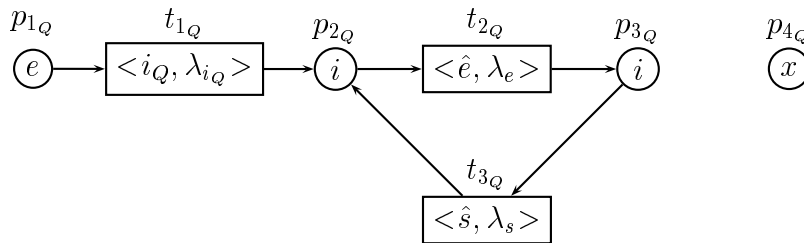


Figura 7.2: s-caja plana de Q

Finalmente, la especificación del *Sistema* es la siguiente:

$$\text{Sistema} = [\{e, s\} : (P_1 \parallel P_2 \parallel \dots \parallel P_n \parallel Q)]$$

Potencia y flexibilidad de sPBC

En [DHR95] podemos encontrar una comparación de los mecanismos para modelar con álgebras de procesos estocásticas con los de redes de Petri estocásticas. Una de las principales ventajas de nuestro modelo es su composicionalidad, de modo que podríamos modelar las componentes más pequeñas de acuerdo a un álgebra de procesos y verificar su comportamiento traduciéndolas a s-cajas planas. Una vez obtenida la s-caja plana asociada al sistema, podríamos preguntarnos sobre propiedades estructurales. Nuestro modelo da lugar siempre a redes 1-seguras, lo que puede facilitar el estudio. Por ejemplo, en el caso expuesto anteriormente, podríamos preguntarnos si se cumple la propiedad de exclusión mutua:

¿Existe algún mercado alcanzable desde M_e en el que estén marcados $(p_{4_j} \vee p_{5_j}) \wedge (p_{4_k} \vee p_{5_k})$ para $j \neq k$?

Además, no es difícil ver que en el caso expuesto la CTMC asociada es ergódica (ya que la red de Petri es 1-segura y tiene como estado de acogida $M = p_{2_1} + p_{2_2} + \dots + p_{2_n} + p_{2_Q}$), así que será posible calcular la distribución de los marcados en estado estacionario $\underline{\Pi}$ a partir de la matriz generadora \mathbf{Q} , resolviendo el sistema:

$$\begin{cases} \underline{\Pi}\mathbf{Q} = \mathbf{0} \\ \sum_i \Pi_i = 1 \end{cases}$$

Una vez conocido $\underline{\Pi}$ será posible calcular los índices de rendimiento, a partir de la noción de *función ganancia* [AMBC⁺95]. Por ejemplo, podemos preguntarnos:

¿cuánto tiempo se está utilizando la región crítica?

para ello consideraríamos:

$$R = \sum_{M_i \in RS(M_e)} r(M_i)\pi_i$$

donde

$$r(M_i) = \begin{cases} 1 & \text{si } p_{3_Q} \text{ está marcado en } M_i \\ 0 & \text{en otro caso} \end{cases}$$

Otra ventaja de nuestro modelo (como álgebra de procesos) es poder definir una congruencia (\sim) (capítulo 6), que nos permite establecer, por ejemplo, que

$$\langle t_j, \lambda_{t_j} \rangle \sim \langle t_j, \lambda'_{t_j} \rangle \square \langle t_j, \lambda''_{t_j} \rangle$$

si $\lambda_{t_j} = \lambda'_{t_j} + \lambda''_{t_j}$. A partir de esta equivalencia se podría hacer la correspondiente sustitución en un sistema, obteniendo un sistema equivalente al original, tanto a nivel funcional como a nivel de estudio de prestaciones. Así pues, con nuestro modelo combinamos la potencia de las álgebras de procesos estocásticas clásicas y de las redes de Petri estocásticas.

Otra ventaja adicional proviene del uso de las multiacciones, el cual proporciona una mayor flexibilidad a la hora de modelar. Por ejemplo, supongamos que el caso modelado se refiere a una línea de producción industrial y Q representa la utilización de un lugar común, en el que los procesos entrarán de dos en dos (supongamos que se trata de un elevador con dos plazas y que por motivos de equilibrio, sólo es posible que funcione si las dos plazas están ocupadas). En tal caso, cada P_i se seguiría modelando de la misma forma y el elevador sería:

$$E = [\langle i_E, \lambda_{i_E} \rangle * (\langle \{\hat{e}, \hat{e}\}, \lambda_e \rangle ; \langle \{\hat{s}, \hat{s}\}, \lambda_s \rangle) * \langle f, \lambda_f \rangle] rs f$$

$$Sistema_{elevador} = [\{e, s\} : (P_1 \parallel P_2 \parallel \dots \parallel P_n \parallel E)]$$

7.1.2 Anillo lógico

Se considera ahora un sistema constituido por n procesos P_i , con $i = 1, \dots, n$, ejecutándose en paralelo sobre un anillo lógico, de manera que cada proceso tiene un predecesor y un sucesor. Los procesos se encuentran realizando independientemente una tarea, teniendo que entrar, de vez en cuando, en una región crítica. Para conseguir la exclusión mutua se tiene un token circulando por el anillo, de modo que sólo el proceso que disponga del token podrá entrar en la región crítica, si así lo desea. Para mostrar la flexibilidad de nuestro modelo vamos a partir de unas hipótesis un poco diferentes de las que corresponden al caso centralizado; así, los convenios que se siguen ahora son:

1. Supondremos que el sistema describe un proceso de comportamiento infinito, pero cada proceso componente P_j realiza una tarea de iniciación $\langle i_{P_j}, \lambda_{i_{P_j}} \rangle$, para $j = 1, 2, \dots, n$, tras lo cual pasará a realizar el siguiente ciclo (con $j = 2, \dots, n$):
 - (a) Se prepara para sus tareas $\langle i_{t_j}, \lambda_{i_{t_j}} \rangle$, y luego pasa a realizar sus tareas fuera de la región crítica, $\langle t_j, \lambda_{t_j} \rangle$, las veces necesarias (eventualmente

ninguna). Cuando acabe de realizarlas, se reseteará vía $\langle f_{t_j}, \lambda_{f_{t_j}} \rangle$. Esta parte se puede modelar de la forma:

$$T_j = [\langle i_{t_j}, \lambda_{i_{t_j}} \rangle * \langle t_j, \lambda_{t_j} \rangle * \langle f_{t_j}, \lambda_{f_{t_j}} \rangle]$$

- (b) Después está en disposición de atrapar el token $\langle a_j, \lambda_{a_j} \rangle$ (cuando se lo pase su predecesor).
- (c) Una vez que ya tiene el token, se prepara para realizar sus tareas $\langle i_{r_j}, \lambda_{i_{r_j}} \rangle$, y luego puede realizar sus tareas dentro de la región crítica, $\langle r_j, \lambda_{r_j} \rangle$, las veces necesarias (en particular ninguna). Cuando acabe de realizarlas, se reseteará $\langle f_{r_j}, \lambda_{f_{r_j}} \rangle$. Esta parte se puede modelar de la forma:

$$R_j = [\langle i_{r_j}, \lambda_{i_{r_j}} \rangle * \langle r_j, \lambda_{r_j} \rangle * \langle f_{r_j}, \lambda_{f_{r_j}} \rangle]$$

- (d) A continuación, le pasará el token al proceso siguiente $\langle \hat{a}_{i+1}, \lambda_{\hat{a}_{i+1}} \rangle$ (tomaríamos $n + 1 = 1$) y volverá al punto de partida.

2. El primer proceso del anillo P_1 posee el token al comenzar el funcionamiento del sistema. Su punto de partida en el ciclo será el ítem (c) del caso previo.
3. El token no se puede perder.

Podemos modelar con sPBC cada componente de la forma siguiente:

$$P_j = [\langle i_{P_j}, \lambda_{i_{P_j}} \rangle * (T_j ; \langle a_j, \lambda_{a_j} \rangle ; R_j ; \langle \hat{a}_{j+1}, \lambda_{\hat{a}_{j+1}} \rangle) * \langle f, \lambda_f \rangle]rsf$$

para $j = 2, \dots, n$.

La s-caja plana de P_j con $j \geq 2$ es la descrita en la figura 7.3.

Y para el primer proceso, que parte teniendo el token tendríamos:

$$P_1 = [\langle i_{P_1}, \lambda_{i_{P_1}} \rangle * (R_1 ; \langle \hat{a}_2, \lambda_{\hat{a}_2} \rangle ; T_1 ; \langle a_1, \lambda_{a_1} \rangle) * \langle f, \lambda_f \rangle]rsf$$

La s-caja plana de P_1 es la mostrada en la figura 7.4.

Finalmente, el sistema completo quedaría modelado de la forma siguiente:

$$Anillo = [A : (P_1 \parallel P_2 \parallel \dots \parallel P_n)]$$

donde $A = \{a_1, a_2, \dots, a_n\}$

Productor:

- Se prepara para iniciar el proceso $\langle i, \lambda_{i_P} \rangle$.
- A continuación, realiza ciclos que consisten siempre en la secuencia de producir un ítem $\langle p, \lambda_p \rangle$ seguida de depositar el ítem en el almacén $\langle a, \lambda_a \rangle$. El número de ciclos que realiza es arbitrario (incluso puede que ninguno).
- Después se prepara para finalizar el proceso $\langle f, \lambda_{f_P} \rangle$.

Consumidor:

- Se prepara para iniciar el proceso $\langle i, \lambda_{i_C} \rangle$.
- A continuación, realiza ciclos que consisten siempre en la secuencia de tomar un ítem del almacén $\langle t, \lambda_t \rangle$ seguida de consumir el ítem $\langle c, \lambda_c \rangle$. El número de ciclos que realiza es arbitrario (incluso puede que ninguno).
- Después se prepara para finalizar el proceso $\langle f, \lambda_{f_C} \rangle$.

Almacén:

- Se prepara para iniciar el proceso $\langle \{\hat{i}, \hat{i}\}, \lambda_{i_A} \rangle$.
- A continuación, (recordemos que el almacén tiene capacidad 1), realiza ciclos que consisten en:
 - primero permitir que se ponga un ítem en el almacén $\langle \hat{a}, \lambda_{\hat{a}} \rangle$, tras lo cual permite que se tome un ítem del almacén $\langle \hat{t}, \lambda_{\hat{t}} \rangle$
- Después se prepara para finalizar el proceso $\langle \{\hat{f}, \hat{f}\}, \lambda_{f_A} \rangle$.

Así, la especificación para la componente *Productor* sería:

$$Productor = [\langle i, \lambda_{i_P} \rangle * (\langle p, \lambda_p \rangle; \langle a, \lambda_a \rangle) * \langle f, \lambda_{f_P} \rangle]$$

La s-caja plana del *Productor* es la descrita en la figura 7.5.

Análogamente, la especificación para el *Consumidor* sería:

$$Consumidor = [\langle i, \lambda_{i_C} \rangle * (\langle t, \lambda_t \rangle; \langle c, \lambda_c \rangle) * \langle f, \lambda_{f_C} \rangle]$$

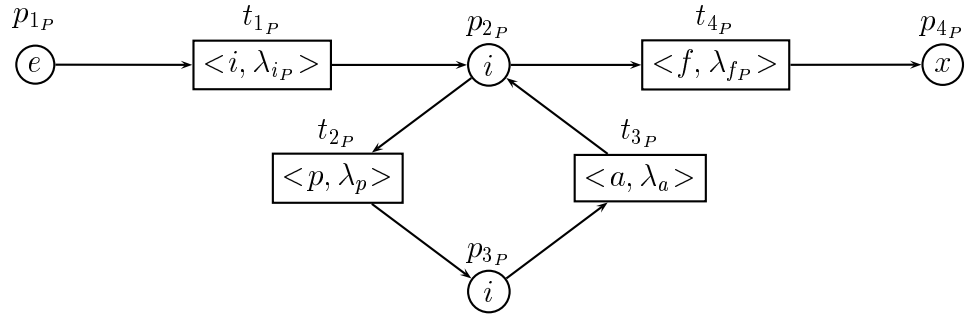


Figura 7.5: s-caja plana del *Productor*.

La s-caja plana del *Consumidor* se muestra en la figura 7.6.

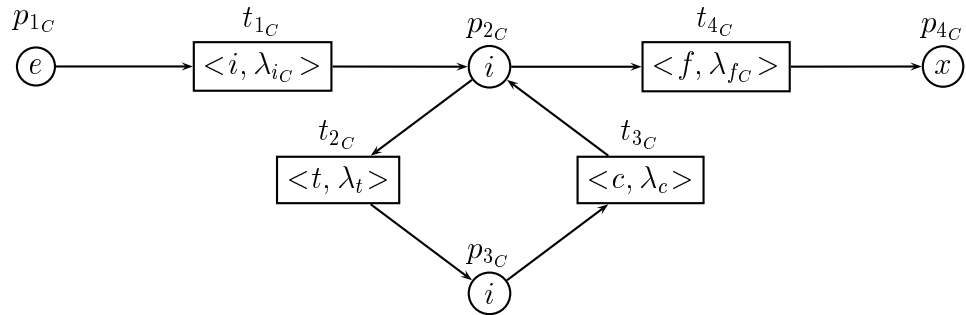


Figura 7.6: s-caja plana del *Consumidor*

Finalmente, para el *Almacén* tendríamos:

$$\text{Almacén} = [\langle \{\hat{i}, \hat{i}\}, \lambda_{i_A} \rangle * \langle \hat{a}, \lambda_{\hat{a}} \rangle ; \langle \hat{t}, \lambda_{\hat{t}} \rangle * \langle \{\hat{f}, \hat{f}\}, \lambda_{f_A} \rangle]$$

La s-caja plana del *Almacén* es la mostrada en la figura 7.7.

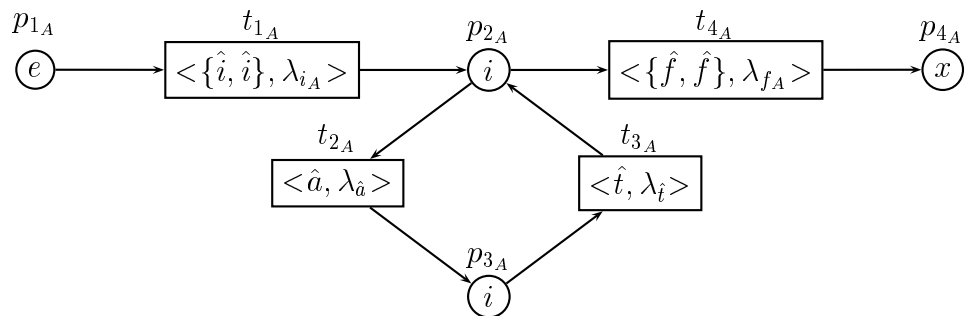


Figura 7.7: s-caja del *Almacén*

De esta forma, la especificación del *Sistema* sería de la forma:

$$\text{Sistema} = [A : (\text{Productor} \parallel \text{Consumidor} \parallel \text{Almacén})]$$

donde $A = \{i, f, a, t\}$

La CTMC asociada no es ergódica, de modo que el análisis puede hacerse a partir de la matriz generadora \mathbf{Q} , con

$$\underline{p}'(t) = \underline{p}(t)\mathbf{Q}, \quad \text{dado } \underline{p}(0)$$

donde $p_i(t)$ representa la probabilidad de que el sistema se encuentre en el estado i (marcado M_i) en el instante de tiempo t . A partir de ahí es posible obtener los índices de rendimiento mediante el concepto de *función ganancia* [Hav98]. Por ejemplo, podríamos estar interesados en conocer:

¿qué probabilidad se tiene de que en el instante de tiempo t se esté esperando a almacenar un ítem que ya se ha producido?

Para ello consideraríamos:

$$E[X(t)] = \sum_{M_i \in RS(M_e)} r(M_i) p_i(t)$$

donde

$$r(M_i) = \begin{cases} 1 & \text{si } p_{3P} \text{ está marcado en } M_i \\ 0 & \text{en otro caso} \end{cases}$$

O también nos puede interesar el tiempo que el sistema ha estado en cualquier estado hasta el instante t . Esto se conoce como medidas acumulativas (*cumulative measures*) [Hav98], y vienen definidas por el vector $\underline{l}(t)$,

$$\underline{l}(t) = \int_0^t \underline{p}(s) ds$$

que denota el tiempo total pasado en el estado i durante el intervalo $[0, t)$. De esta forma, utilizando la misma función ganancia que anteriormente,

$$Y(t) = \sum_{M_i \in RS(M_e)} r(M_i) l_i(t)$$

expresará la cantidad de tiempo que se ha estado esperando para almacenar un ítem que ya se había producido en el período $[0, t)$.

7.2.1 Almacén con capacidad para dos ó más ítems

Si hubiésemos necesitado modelar un almacén con capacidad dos, la descripción del mismo sería ahora la siguiente.

Almacén₂:

- Se prepara para iniciar el proceso $\langle \{\hat{i}, \hat{i}\}, \lambda_{i_A} \rangle$.
- A continuación, realiza ciclos que consisten en:
 - primero permitir que se introduzca un ítem en el almacén $\langle \hat{a}, \lambda_{\hat{a}} \rangle$,
 - a continuación, realiza ciclos (puede que ninguno) que consisten en la secuencia de permitir depositar un ítem en el almacén $\langle \hat{a}, \lambda_{\hat{a}} \rangle$, tras lo que se permite que se tome un ítem del almacén $\langle \hat{t}, \lambda_{\hat{t}} \rangle$.
 - a continuación, se permitirá que se tome el ítem que había almacenado $\langle \hat{t}, \lambda_{\hat{t}} \rangle$.
- Después se prepara para finalizar el proceso $\langle \{\hat{f}, \hat{f}\}, \lambda_{f_A} \rangle$.

La especificación algebraica del *Almacén* con capacidad dos será:

$$\text{Almacén}_2 = [\langle \{\hat{i}, \hat{i}\}, \lambda_{i_A} \rangle * [\langle \hat{a}, \lambda_{\hat{a}} \rangle * (\langle \hat{a}, \lambda_{\hat{a}} \rangle ; \langle \hat{t}, \lambda_{\hat{t}} \rangle) * \langle \hat{t}, \lambda_{\hat{t}} \rangle] * \langle \{\hat{f}, \hat{f}\}, \lambda_{f_A} \rangle]$$

La s-caja plana de *Almacén₂* es la mostrada en la figura 7.8.

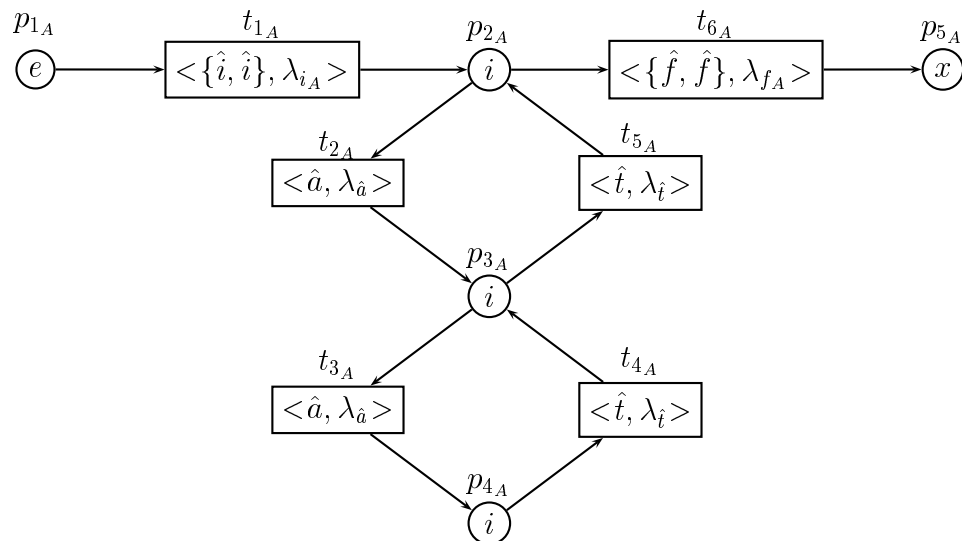


Figura 7.8: s-caja plana de *Almacén₂*

De esta forma, la especificación completa del *Sistema* es:

$$\text{Sistema}_2 = [A : (\text{Productor} \parallel \text{Consumidor} \parallel \text{Almacén}_2)]$$

donde $A = \{i, f, a, t\}$.

Si siguiendo la misma estrategia, si queremos modelar un *Almacén* con capacidad para n ítems, la especificación en sPBC podría ser de la forma:

$$I_1 = (\langle \hat{a}, \lambda_{\hat{a}} \rangle; \langle \hat{t}, \lambda_{\hat{t}} \rangle)$$

$$I_n = [\langle \hat{a}, \lambda_{\hat{a}} \rangle * I_{n-1} * \langle \hat{t}, \lambda_{\hat{t}} \rangle], \quad n \geq 2$$

$$Almacén_n = [\langle \{\hat{i}, \hat{i}\}, \lambda_{i_A} \rangle * I_n * \langle \{\hat{f}, \hat{f}\}, \lambda_{f_A} \rangle]$$

La s-caja plana correspondiente a $Almacén_n$ es la mostrada en la figura 7.9.

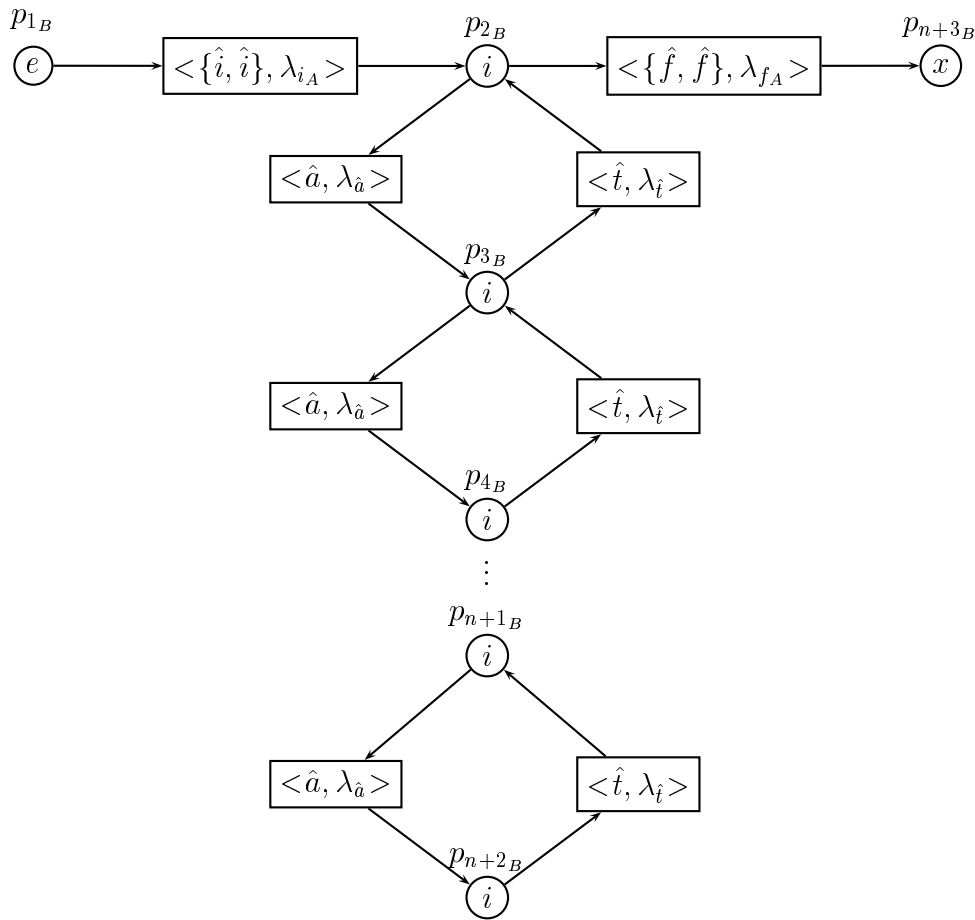


Figura 7.9: s-caja plana de $Almacén_n$

Finalmente, la especificación del *Sistema* en sPBC sería:

$$Sistema_n = [A : (Productor \parallel Consumidor \parallel Almacén_n)]$$

donde $A = \{i, f, a, t\}$.

7.3 El protocolo del Bit Alternante

ABP (Alternating Bit Protocol) [BSW69], es un protocolo simple de comunicación que proporciona una comunicación libre de errores sobre un medio que puede perder mensajes.

En nuestra descripción (basada en la que aparece en [Han91]) intervienen cuatro entidades: un emisor, dos canales y un receptor. La estructura general del sistema de comunicación viene esbozada en la figura 7.10.

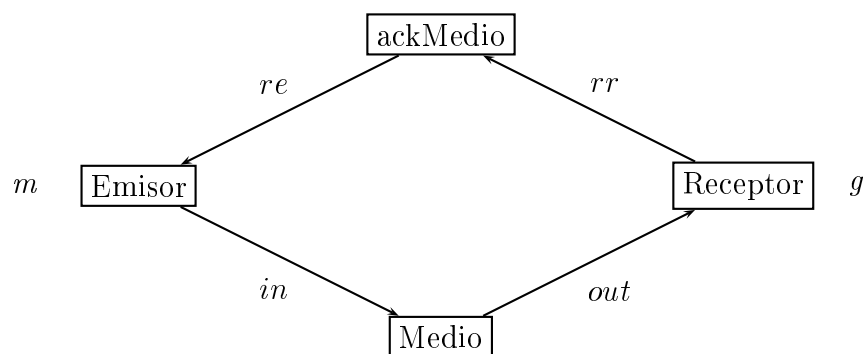


Figura 7.10: La estructura del ABP

A grandes rasgos, el funcionamiento del sistema es como sigue: el *Emisor* produce mensajes que envía al *Receptor* a través de un medio con dos líneas: la que va del *Emisor* al *Receptor*, que denominaremos *Medio*, y que va del *Receptor* al *Emisor*, llamada *ackMedio*, que se encargará de transmitir los acuses de recibo correspondientes. Estas líneas pueden perder tanto los mensajes como los acuses de recibo. Suponemos que las cuatro entidades tienen que sincronizarse para iniciar el sistema, tras lo que cada una de ellas tiene el comportamiento siguiente:

- Emisor:

1. El *Emisor* empieza a transmitir un mensaje $\langle m, \lambda_m \rangle$, que estará etiquetado con 0 ó con 1 (para que el receptor sepa si es un mensaje nuevo o antiguo). La correspondiente multiacción será $\langle \hat{in}_0, \lambda_{\hat{in}_0} \rangle$ ó $\langle \hat{in}_1, \lambda_{\hat{in}_1} \rangle$, según el caso.
2. A continuación repite el proceso (puede que ninguna vez): si no ha recibido el acuse de recibo correspondiente a la etiqueta que le puso al mensaje $\langle nre_i, \lambda_{nre_i} \rangle$, lo volverá a enviar, y la multiacción en este caso será $\langle \hat{in}_i, \lambda_{\hat{in}_i} \rangle$.

3. El paso anterior lo acabará cuando reciba el acuse de recibo correspondiente $\langle re_i, \lambda_{re_i} \rangle$.
 4. Volverá a repetir los pasos anteriores, pero cambiando la etiqueta.
- Medio:
 1. Puede recibir mensajes etiquetados con 0 ó 1 ($\langle in_i, \lambda_{in_i} \rangle$ con $i = 0, 1$ según el caso). A continuación puede perder el mensaje ($\langle p_M, \lambda_{p_M} \rangle$), o bien mandarlo correctamente al receptor ($\langle out_i, \lambda_{out_i} \rangle$ con $i = 0, 1$, según corresponda con el mensaje recibido).
 2. Repite el paso anterior.
 - Receptor:
 1. Al *Receptor* le llega un mensaje que registra $\langle \hat{g}, \lambda_{\hat{g}} \rangle$, lo etiquetará con 0 ó con un 1 (según la etiqueta con la que le llegó), y lo transmitirá con el acuse de recibo $\langle \hat{r}r_0, \lambda_{\hat{r}r_0} \rangle$ ó $\langle \hat{r}r_1, \lambda_{\hat{r}r_1} \rangle$, según el caso.
 2. A continuación repite el proceso (puede que ninguna vez): si le siguen llegando mensajes del medio con la misma etiqueta que ya tenía $\langle out_i, \lambda_{out_i} \rangle$, volverá a enviar el mismo acuse de recibo $\langle \hat{r}r_i, \lambda_{\hat{r}r_i} \rangle$.
 3. El paso anterior lo acabará cuando reciba un mensaje con diferente etiqueta $\langle out_{i+1}, \lambda_{out_{i+1}} \rangle$, tomando $1 + 1 = 0$.
 4. Volverá a repetir los pasos anteriores, pero cambiando la referencia a $i + 1$.
 - ackMedio:
 1. Puede llegar un acuse de recibo del receptor etiquetado con 0 ó 1 ($\langle rr_i, \lambda_{rr_i} \rangle$ con $i = 0, 1$ según el caso), que podrá perder $\langle p_a, \lambda_{p_a} \rangle$, o bien lo mandará al Emisor ($\langle \hat{r}e_i, \lambda_{\hat{r}e_i} \rangle$ con $i = 0, 1$, según corresponda, en función de la etiqueta del acuse de recibo).
 2. Repite el paso anterior.
 - El primer mensaje mandado por el Emisor estará etiquetado con 0.

La especificación del *Emisor* en sPBC será entonces la siguiente:

$$\begin{aligned}
 Emisor = & [\langle \{i, i, i\}, \lambda_{i_E} \rangle * \\
 & (\langle m, \lambda_m \rangle; [\langle \hat{in}_0, \lambda_{\hat{in}_0} \rangle * (\langle nre_0, \lambda_{nre_0} \rangle; \langle \hat{in}_0, \lambda_{\hat{in}_0} \rangle) * \langle re_0, \lambda_{re_0} \rangle]; \\
 & \langle m, \lambda_m \rangle; [\langle \hat{in}_1, \lambda_{\hat{in}_1} \rangle * (\langle nre_1, \lambda_{nre_1} \rangle; \langle \hat{in}_1, \lambda_{\hat{in}_1} \rangle) * \langle re_1, \lambda_{re_1} \rangle]) * \\
 & \langle f, \lambda_f \rangle] rs f
 \end{aligned}$$

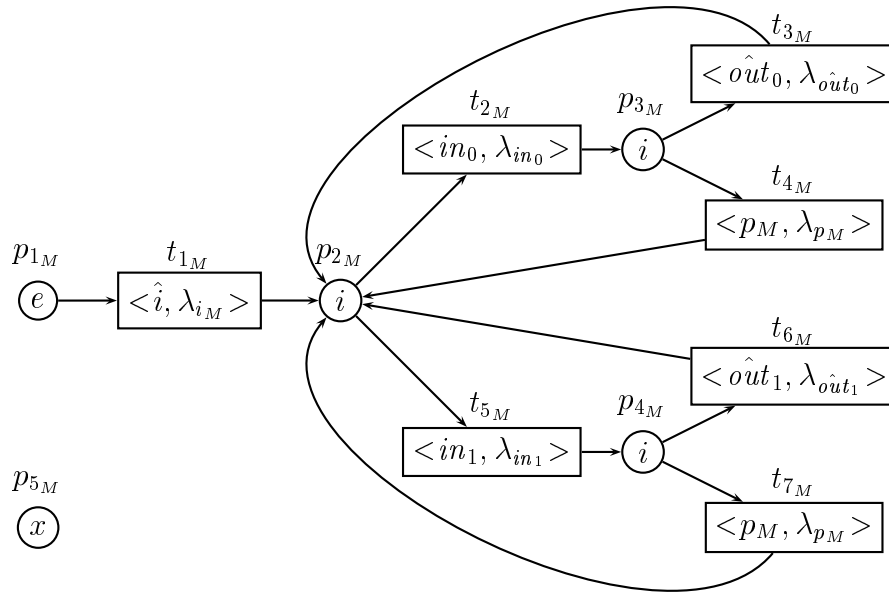


Figura 7.12: s-caja plana del *Medio*

La s-caja plana del *Receptor* es la mostrada en la figura 7.13.

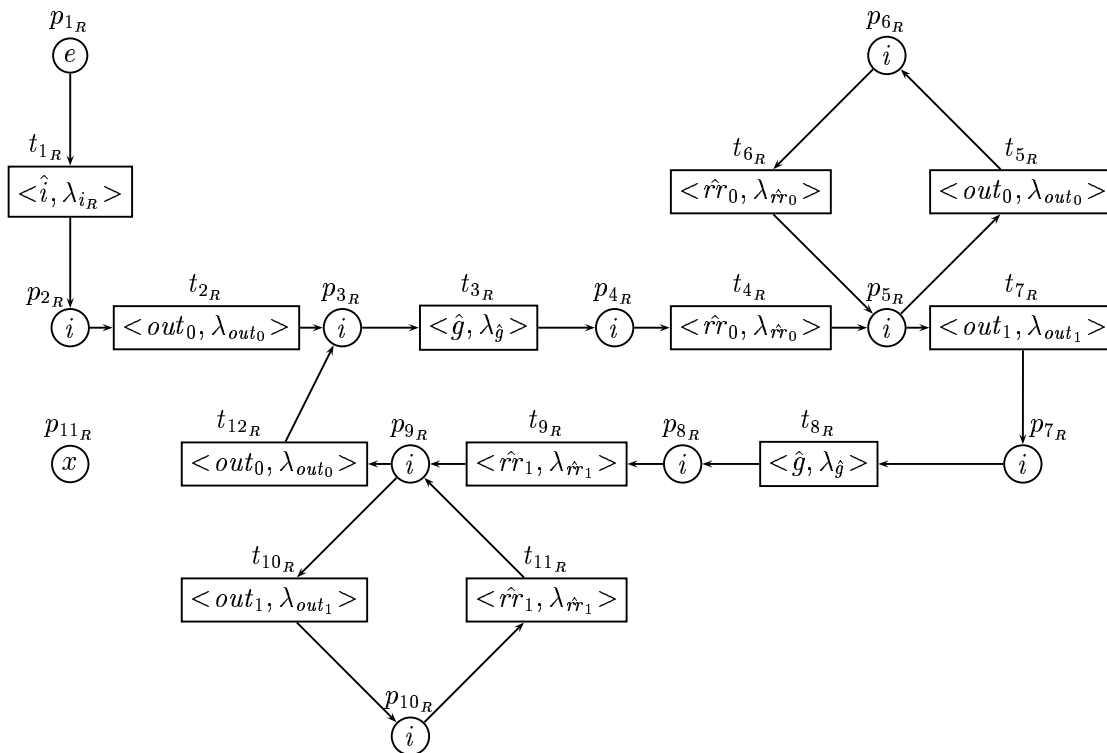


Figura 7.13: s-caja plana del *Receptor*

Por último, la especificación del *ackMedio* será:

$$ackMedio = [\langle \hat{i}, \lambda_{i_{aM}} \rangle * (\langle rr_0, \lambda_{rr_0} \rangle; (\langle \hat{r}e_0, \lambda_{\hat{r}e_0} \rangle \square \langle p_{aM}, \lambda_{p_{aM}} \rangle)) \square (\langle rr_1, \lambda_{rr_1} \rangle; (\langle \hat{r}e_1, \lambda_{\hat{r}e_1} \rangle \square \langle p_{aM}, \lambda_{p_{aM}} \rangle)) * \langle f, \lambda_f \rangle] rs f$$

La s-caja plana del *ackMedio* es la mostrada en la figura 7.14.

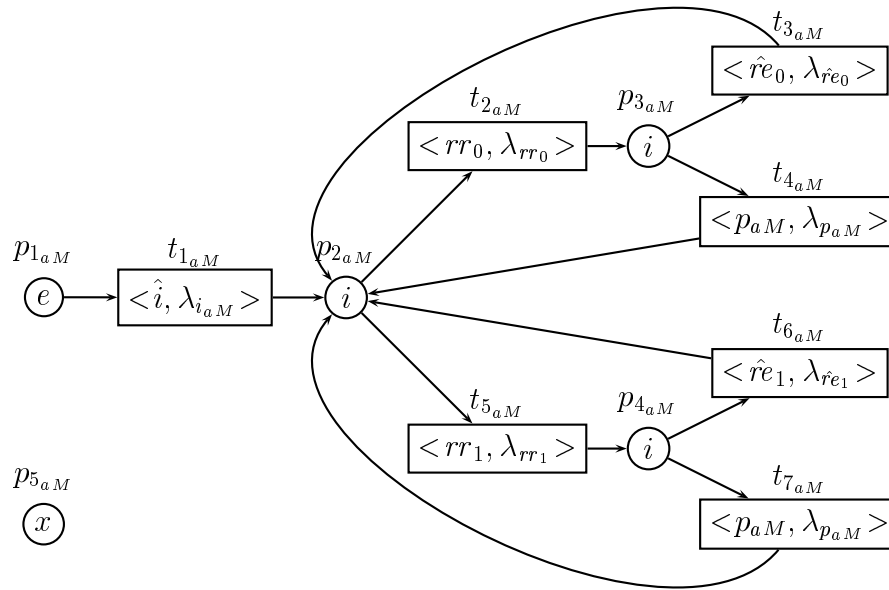


Figura 7.14: s-caja plana del *ackMedio*

En definitiva, la especificación del *Sistema* completo es la siguiente:

$$Sistema = [A : (Emisor || Medio || Receptor || ackMedio)]$$

donde $A = \{i, in_0, in_1, out_0, out_1, re_0, re_1, rr_0, rr_1\}$

Capítulo 8

Conclusiones y Trabajo Futuro

En esta tesis hemos presentado una extensión markoviana del modelo PBC. Por medio de ella, además de tener integrado en el mismo modelo aspectos cualitativos y cuantitativos, se tienen simultáneamente las ventajas de las álgebras de procesos (por ejemplo, la composicionalidad) y de las redes de Petri (por ejemplo, su representación gráfica y los algoritmos de análisis). Todo ello hace que bajo nuestro punto de vista, se trate de un modelo muy adecuado para el diseño, análisis y evaluación de prestaciones de sistemas concurrentes, en los que aparezcan esperas probabilísticas (en particular, gobernadas por una distribución exponencial).

De esta forma hemos definido un álgebra de procesos markoviana, dando para el lenguaje, en primer lugar, su sintaxis y su semántica operacional. Aunque no se contempla el caso de la recursión, sí se pueden obtener comportamientos infinitos, utilizando el operador de iteración. El caso de la sincronización ha sido el más delicado, al estar separado del operador paralelo, y que sepamos, no contemplado así en ninguna otra álgebra de procesos estocástica. Finalmente, conseguimos adaptar la semántica de dicho operador, capturando el significado intuitivo de esperar a la ejecución de la multiacción más lenta de las dos sincronizadas, y con buenas propiedades algebraicas que garantizan, entre otras cosas, una fórmula sistemática para el cálculo de la razón cuando en la sincronización intervienen varias multiacciones estocásticas, la propiedad conmutativa y asociativa de las multiacciones al sincronizar y la obtención de una congruencia al definir una equivalencia estocástica adecuada. A nivel técnico para lograrlo ha sido necesario considerar las razones de conflicto y también una restricción sintáctica, considerando los llamados términos regulares. Además, hemos introducido nuevas transiciones en la semántica operacional, para

poder definir con ellas una equivalencia adecuada que contemple la equivalencia cualitativa (a nivel funcional) y la cuantitativa (a nivel de prestaciones), y que la misma se conserve tras la aplicación de todos los operadores (esto es, que sea una congruencia).

Se ha definido también una semántica denotacional, basada en una clase especial de redes de Petri estocásticas (markovianas) etiquetadas. Se muestra la equivalencia entre las dos semánticas, como se tenía en el modelo base PBC. Como la aceptación de las redes de Petri estocásticas (markovianas) y el trabajo realizado sobre ellas son muy amplios, al tener como objetos semánticos este tipo de redes, heredamos todas las ventajas de ellas, incluidas las correspondientes a la evaluación de prestaciones.

En PBC el operador base es la multiacción α , cuya ejecución no conlleva consumo de tiempo ni retraso alguno, mientras que en sPBC tenemos la multiacción estocástica $\langle \alpha, r \rangle$, cuya interpretación es que desde el momento en que se permite su ejecución hasta que es posible ejecutarla (su ejecución no conlleva tiempo) sufre un retraso o espera gobernado por una distribución exponencial negativa de parámetro r . Cuando varias multiacciones estocásticas (transiciones en la red de Petri) están activadas a la vez, se aplica una política de competencia. La principal diferencia entre sPBC y PBC es que en el modelo base se parte de una concurrencia de orden parcial, mientras que en nuestro modelo es de orden total, ya que al utilizar funciones de probabilidades continuas se tiene que la probabilidad de ejecutar dos multiacciones estocásticas (transiciones, si nos referimos a las redes) a la vez es nula.

De esta forma, sPBC es un álgebra de procesos estocástica (markoviana) del estilo de PEPA, TIPP o EMPA, pero pensada (al igual que PBC) para tener una fácil y natural traslación a redes de Petri. Sus principales diferencias con ellas están originadas por ese punto de partida: permitir multiacciones a un primer nivel, tener separado el operador paralelo del operador de sincronización y tener una semántica operacional basada en s-expresiones dinámicas, en las cuales con los barrados y sub-barrados se refleja la evolución del proceso (correspondiendo al marcado en la red). Estas diferencias generan asimismo sus principales ventajas: siempre se tienen traslaciones a redes de Petri estocásticas donde las razones de las transiciones son independientes del marcado, a diferencia de lo que sucede en PEPA, donde en algunos casos las razones pueden depender del marcado; tiene una aceptable interpretación intuitiva de la nueva razón obtenida al sincronizar (esperarse a la más lenta), a diferencia de EMPA, donde la sincronización siempre impone que uno de los argumentos sea pasivo, o de TIPP, donde la nueva razón es el producto de las

razones que intervienen en la sincronización. Además, se permiten multiacciones, con lo cual se facilita una cierta concurrencia real, aunque sea sólo a un primer nivel.

8.1 Trabajos publicados

A continuación presentamos los trabajos y publicaciones de la doctorando relacionados con los contenidos de la presente tesis:

1. [Mac99] H. Macià. *Una introducción al álgebra de cajas*. Trabajo de doctorado. Dpto. de Informática. Universidad de Castilla - La Mancha, Septiembre 1999.
2. [Mac00] H. Macià. *Análisis de especificaciones algebraicas mediante PBC*. Trabajo de doctorado. Dpto. de Informática. Universidad de Castilla - La Mancha, Septiembre 2000.
3. [MV01] H. Macià, V. Valero. *Pep. Una herramienta para traducir y analizar especificaciones de PBC con redes de Petri*. Proc. XI Escuela de Verano. Sistemas Distribuidos: Modelos y Aplicaciones. Albacete, pp. 187-202, Julio 2001.
4. [MVdF01b] H. Macià, V. Valero, D. de Frutos. *sPBC: A Markovian Extension of finite Petri Box Calculus*. Proc. IX Jornadas de Concurrencia, Sitges, pp. 243-356, Junio 2001.
5. [MVdF01a] H. Macià, V. Valero, D. de Frutos. *sPBC: A Markovian Extension of Finite Petri Box Calculus*. Proc. 9th IEEE Int. Workshop on Petri Nets and Performance Models (PNPM'01), IEEE Computer Society Press, pp. 207-216, Septiembre 2001.
6. [MVdFC01] H. Macià, V. Valero, D. de Frutos, F. Cuartero. *sPBC: Extending PBC with Markovian Multiactions*. Proc. XXVII Conferencia Latinoamericana de Informática (CLEI'01), Mérida (Venezuela), Septiembre 2001.
7. [MVCP02b] H. Macià, V. Valero, F. Cuartero, F.L. Pelayo. *A new proposal for the synchronization in sPBC*. Technical Report, DIAB-02-01-26, Departamento de Informática, Universidad de Castilla-La Mancha, Junio 2002. Disponible en <http://www.info-ab.uclm.es/trep.php>

8. [MVCP02a] H. Macià, V. Valero, F. Cuartero, F.L. Pelayo. *Improving the synchronization in Stochastic Petri Box Calculus*. Proc. II Jornadas sobre Programación y Lenguajes (PROLE'02), El Escorial, Noviembre 2002.
9. [MVCP03] H. Macià, V. Valero, F. Cuartero, F.L. Pelayo. *A new proposal for the synchronization in sPBC*. Proc. Third IEEE Int. Conference on Application of Concurrent to System Design (ACSD'03), IEEE Computer Society Press, pp. 216-225, Junio 2003.
10. [MVC02] H. Macià, V. Valero, F. Cuartero. *A Congruence relation in finite sPBC*. Technical Report, DIAB-02-01-31, Departamento de Informática, Universidad de Castilla-La Mancha, Octubre 2002. Disponible en <http://www.info-ab.uclm.es/trep.php>
11. [MVCdF03] H. Macià, V. Valero, F. Cuartero, D. de Frutos. *A Congruence relation in finite sPBC*. Submitted for publication, 2003.
12. [MVCC03] H. Macià, V. Valero, D. Cazorla, F. Cuartero. *Introducing the iteration in sPBC*. Technical Report, DIAB-03-01-37, Departamento de Informática, Universidad de Castilla-La Mancha, Septiembre 2003. Disponible en <http://www.info-ab.uclm.es/trep.php>

Las cuales se pueden resumir de la forma siguiente:

- En el trabajo de doctorado (1), se presentó el modelo PBC, sobre todo centrándonos en el primer trabajo sobre él [BDH92]. En el siguiente y último trabajo de doctorado (2), se hizo hincapié en la utilización del modelo para modelar sistemas concurrentes, utilizando el lenguaje de programación $B(PN)^2$ y su herramienta PEP. En (3), con motivo del curso de verano *Sistemas distribuidos: Modelos y Aplicaciones* organizado por el Departamento de Informática de la Universidad de Castilla-La Mancha, se presentó una panorámica de la herramienta PEP.
- En (4), (5) y (6) se presentó la primera versión de sPBC finito, que se corresponde con lo expuesto en el capítulo 3 de esta tesis.
- En (7), (8) y (9) se presentó la nueva propuesta de la sincronización (capítulo 4).
- Las equivalencias (capítulo 6) fueron estudiadas para el caso finito en (10) y (11).

- Finalmente, la incorporación del operador de iteración (capítulo 5), manteniendo las equivalencias definidas con anterioridad (capítulo 6), e incorporando algunos ejemplos (capítulo 7) se presentó en (12).

El trabajo realizado en esta tesis, así como el trabajo futuro más inmediato, ha sido desarrollado (o va a ser desarrollado) en el marco de los siguientes proyectos de investigación:

- ★ *Desarrollo formal de Sistemas Distribuidos (TIC97-0669-C03-02)*. Dicho proyecto fue desarrollado en colaboración con la Universidad Complutense de Madrid y la Universidad de Vigo. Tuvo lugar entre 1997 y 2000, con una financiación de 4.500.000 ptas.
- ★ *Evaluación de Rendimientos de Sistemas Distribuidos (TIC2000-0701-C02-02)*. Corresponde a un subproyecto, dentro del proyecto *Desarrollo Formal de Sistemas Basados en Agentes Móviles y Evaluación de Rendimientos*, llevado en colaboración con la Universidad Complutense de Madrid, cuya duración es desde Enero de 2001 hasta finales de 2003, y con una financiación de 6.160.000 ptas.
- ★ *Diseño e Implementación de Sistemas Multimedia con uso de Técnicas Formales (Junta de Comunidades, Consejería de Ciencia y Tecnología, PAC-03001)*. Su duración es de Junio 2003 hasta Junio 2006, con una financiación de 96.000 euros.
- ★ *Descripción y Evaluación de Sistemas Distribuidos y Aplicaciones a Sistemas Multimedia (TIC2003-07848-C02-02)*. Dicho proyecto es de reciente concesión, con una duración de tres años y una financiación de 71.200 euros.

8.2 Trabajo Futuro

Como trabajos futuros más inmediatos, con el objetivo de conseguir un lenguaje más potente y flexible, tenemos previsto:

- Incluir en nuestro modelo multiacciones inmediatas, por ejemplo al estilo de las que se manejan en EMPA [BG98].

- Considerar el operador de recursión, en primer lugar en el caso guardado y, más adelante, en general.
- Definir formalmente una equivalencia basada en el concepto de bisimulación probabilística [LS91], y probar que es una congruencia.

También como trabajo futuro, más ambicioso y a más largo plazo, estaría el obtener una herramienta propia de este modelo, que combinase las herramientas PEP del modelo base PBC y GreatSPN de las redes de Petri estocásticas.

Bibliografía

- [AM89] M. Ajmone Marsan. Stochastic Petri Nets: An Elementary Introduction. *Lecture Notes in Computer Science; Advances in Petri Nets, 424:1-29*, Springer, 1989.
- [AMBC86] M. Ajmone Marsan, A. Balbo, and G. Conte. *Performance Models of Multiprocessors Systems*. MIT Press, Cambridge, USA, 1986.
- [AMBC⁺95] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. Wiley, 1995.
- [AMBD98] M. Ajmone Marsan, A. Bobbio, and S. Donatelli. Petri Nets in Performance Analysis: An Introduction. In *Lecture Notes in Computer Science: Lectures on Petri Nets I: Basic Models, 1491*, pages 211-256, Springer, 1998.
- [And99] S. Andova. Process algebra with a probabilistic choice. In *ARTS'99, LNCS 1601*, 1999.
- [Bae92] J.C.M. Baeten. The Total Order Assumption. In *Proc. Workshop What good is partial order*, Sheffield (E. Best, ed.), Hildesheimer Informatik-Berichte 13/92, Universitat Hildesheim, pages 1-11, 1992.
- [Bal01] G. Balbo. Introduction to Stochastic Petri Nets. In the book *Lecture on Formal Methods and Performance Analysis, (First EEF/Euro Summer School on Trends in Computer Science)*, E. Brinksma, H. Hermanns and J.P. Katoen (Eds.), LNCS 2090, Springer, pp. 84-155, 2001.
- [BB93] J.C.M. Baeten and J.A. Begstra. Real time process algebra. *Formal Aspects of Computing, vol.3*, pp. 142-188, 1993.

- [BC91] G. Boudol and I. Castellani. Flows models of distributed computations: Even structures and nets. Technical Report RR-1484, INRIA, Sophia Antipolis, 1991.
- [BCSS98] M. Bernardo, W.R. Cleaveland, S.T. Sims, and W.T. Stewart. Two Towers: A Tool Integrating Functional and Performance Analysis of Concurrent Systems. In *Proc. of the IFIP Joint Int. Conf. on Formal Description Techniques for Distributed Systems and Communication Protocols and Protocol Specification, Testing and Verification (FORTE/PSTV'98)*, Kluwer, pp. 457-467, 1998.
- [BDE93] E. Best, R. Devillers, and J. Esparza. General Refinement and Recursion Operators for the Petri Box Calculus. In *10th Annual Symposium on Theoretical Aspects of Computer Science. (STACS 93)*. Enjalbert P., Finkel A., Wagner K.W. (Eds.), LNCS 665, Springer, pp. 130-140, 1993.
- [BDG95] M. Bernardo, L. Donatiello, and R. Gorrieri. Giving a Net Semantics to Markovian Process Algebra. In *Proc. of 6th International Workshop on Petri Nets and Performance Models (PNPM'95)*, IEEE-CS Press, pp. 169-178, 1995.
- [BDH92] E. Best, R. Devillers, and J. Hall. The Box Calculus: A New Causal Algebra with Multi-label Communication. In *Advances in Petri Nets*, G. Rozenberg (Eds.), LNCS 609, Springer, pages 21-69, 1992.
- [BDK98] E. Best, R. Devillers, and M. Koutny. Petri Nets, Process Algebras and Programming Languages. In *Lectures on Petri Nets II: Applications*, W. Reisig and Rozenberg (eds.), Advances in Petri Nets, Volume 1492, Springer-Verlag, pp. 1-84,, 1998.
- [BDK01a] E. Best, R. Devillers, and M. Koutny. A Consistent Model for Nets and Process Algebras. In the book *The Handbook on Process Algebras*, J.A. Bergstra, A. Ponse and S.S. Smolka (Eds.), North Holland, Chapter 14, pages 873-944, 2001.
- [BDK01b] E. Best, R. Devillers, and M. Koutny. *Petri Net Algebra*. EATC, Springer, 2001.

- [BDK02] E. Best, R. Devillers, and M. Koutny. *The Box Algebra = Petri Nets + Process Expressions*. Information and Computation. Academic Press, 178:44–100, 2002.
- [Bes96] E. Best. Partial Order Verification with PEP . In *Proc. of POMIV'96*, 1996.
- [BF94] E. Best and H. Fleischhack. *PEP - Programming Enviroment Based on Petri Nets (project supported by the Deutche Forschungsmeinsschaft)*, 1993-1994.
- [BFF⁺95] E. Best, H. Fleischhack, W. Fraczak, R.P. Hopkins, H. Klaudel, and E. Pelz. An M-net Semantics of $B(PN)^2$. In *Proc. of STRICT'95 (Berlin)*, Springer, 1995.
- [BG98] M. Bernardo and R. Gorrieri. A Tutorial on EMPA : A Theory of Concurrent Process with Nondeterminism, Priorities, Probabilities and Time. *Theoretical Computer Science*, 202:1-54, 1998.
- [BGD94] M. Bernardo, R. Gorrieri, and L. Donatello. MPA: A Stochastic Process Algebra. Technical Report, Laboratory of Computer Science, University of Bologna, 1994.
- [BH93] E. Best and R.P. Hopkins. $B(PN)^2$ - a Basic Petri Net Programming Notation. In *Proc. of PARLE'93*, A. Bode, M. Reeve and G. Wolf (Eds.), LNCS 694, Springer, pages 379-390, 1993.
- [BK95] E. Best and M. Koutny. A Refined View of the Box Algebra. In *Proc. Application and Theory of Petri Nets*. LNCS 935, Springer, pp. 1-20, 1995.
- [Bow96] F.D. Bowden. Modelling time in Petri nets. In *Proc. Second Australia-Japan Workshop on Stochastic Models*, 1996.
- [BPS01] J.A. Bergstra, A. Ponse, and S.A. Smolka. (Editors). *Handbook of process algebra*. Elsevier, 2001.
- [Bra86] G.W. Braams. *Las Redes de Petri. Teoría y práctica*. Masson, 1986.
- [BSW69] K. Bartlett, R. Scantlebury, and P. Wilkinson. A note on reliable full-duplex transmissions over half duplex lines. *Communications of the ACM*, 2(5):260-261, 1969.

- [Buc94] P. Buchholz. Compositional Analysis of a Markovian Process Algebra. In *Proc. of 2nd Process Algebra and Performance Modelling Workshop*, 1994.
- [BW90] J.C.M. Baeten and W.P. Weijiland. *Process Algebra*. Cambridge Tracts in Computer Science 18. Cambridge University Press, 1990.
- [Cam01] J. Campos. Evaluación de prestaciones de sistemas concurrentes modelados con redes de Petri. In *Proc. XI Escuela de Verano. Sistemas Distribuidos: Modelos y Aplicaciones*. Albacete, pp. 141-156, 2001.
- [Caz01] D. Cazorla. *PNAL: Un Modelo Algebraico para Procesos Probabilísticos y no Deterministas*. PhD thesis, Departamento de Informática, Universidad de Castilla-La Mancha, 2001.
- [CCFP03] D. Cazorla, F. Cuartero, Pelayo F.L., and J.J. Pardo. Algebraic Theory of Probabilistic and Nondeterministic Processes. *The Journal of Logic and Algebraic Programming*, Vol. 55, pp. 57-103, Elsevier Science B.V., 2003.
- [CDF91] G. Chiola, S. Donatelli, and G. Franceschinis. GSPNs versus SPNs: What is the actual role of immediate transitions? In *Proc. of the 4th IEEE International Workshop on Petri Nets and Performance Models*, pp.20-31, IEEE Computer Society Press, 1991.
- [CFGR95] G. Chiola, G. Franceschinis, R. Gaeta, and M. Ribaud. GreatSPN 1.7: GRaphical Editor and Analyzer for Timed and Stochastic Petri Nets. *Performance Evaluation*, 24:47-68, 1995.
- [CFV96] F. Cuartero, D. de Frutos, and V. Valero. PCSP: A denotational model for probabilistic processes. In *Proc. Third AMAST Workshop on Real-Time Systems*, 1996.
- [CFV97] F. Cuartero, D. de Frutos, and V. Valero. A sound and complete proof system for probabilistic processes. In *ARTS'96, LNCS 1231*, 1997.
- [Cin75] E. Cinlar. *Introduction to Stochastic Processes*. Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [CMS99] A. Cerone and A. Maggiolo-Schettini. Timed-Based Expressivity of time Petri Nets for System Specification. *Theoretical Computer Science*, 216(1-2):1-53, 1999.

- [Col01] J.M. Colom. Verificación de Propiedades en Redes de Petri. In *Proc. XI Escuela de Verano. Sistemas Distribuidos: Modelos y Aplicaciones*. Albacete, pp. 99-114, 2001.
- [Cua93] F. Cuartero. *CSP Probabilístico (PCSP). Un modelo probabilístico de procesos concurrentes*. PhD thesis, Departamento de Informática y Automática. Facultad de Ciencias Matemáticas de la Universidad Complutense de Madrid, 1993.
- [DdNM88] P. Degano, R. de Nicola, and U. Montanari. A Distributed Operational Semantics for CCS Based on C/E Systems. *Acta Informatica* 26, 59-91, 1988.
- [dFVM00] D. de Frutos, V. Valero, and O. Marroquín. Decidability of Properties of Timed-Arc Petri Nets. In *International Conference on Application and Theory of Petri Nets*, LNCS 1825, Springer, pp. 187-206, 2000.
- [DHR95] S. Donatelli, J. Hillston, and M. Ribaud. A Comparison of Performance Evaluation Process Algebra and Generalized Stochastic Petri Nets. In *Proc. 6th International Workshop on Petri Nets and Performance Models, (PNPM'95)*, Durham, North Carolina, 1995.
- [ERV95] J. Esparza, S. Römer, and W. Vogler. An Improvement of McMillan's Unfolding Algorithm. In *Proc. of TACAS'96 (Oldenburg)*, LNCS 1055, Springer, 1995.
- [Esp93] J. Esparza. Model Checking Using Nets Unfoldings . In *Proc. of TAPOFT'93* . LNCS 668, Springer, 1993.
- [FG97] H. Fleischhack and B. Grahlmann. A Petri Net Semantics for $B(PN)^2$ with Procedures. In *Proc. of PDSE'97*, 1997.
- [FG98] H. Fleischhack and B. Grahlmann. A Compositional Petri Net Semantics for SDL. In *Proc. of ATPN'98*, 1998.
- [FN85] G. Florin and S. Natkin. Les reseaux de Petri stochastiques. *Technique et Science Informatiques*, 4(1), 1985.
- [GB96] B. Grahlmann and E. Best. PEP - More than a Petri Net Tool. In *Proc. of TACAS'96 (Passau)*, 1996.

- [GH94] S. Gilmore and J. Hillston. The PEPA Workbench: A tool to support a Process Algebra-based Approach to Performance Modelling. In *Proc. Seventh International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, vol 794, LNCS, Springer, pp. 353-365, 1994.
- [GHR92] N. Götz, U. Herzog, and M. Rettelbach. TIPP - A Language for Timed Processes and Performance Evaluation. Technical Report, University of Erlangen-Nürnberg, Germany, 1992.
- [GJS90] A. Giacalone, C. Jou, and S.A. Smolka. Algebraic Reasoning for Probabilistic Concurrent Systems. In *Proc. Working Conference on Programming Concepts and Methods*, IFIP TC2, 1990.
- [GL91] U. Goltz and R. Loogen. A Non-Interleaving Semantic Model for Non-deterministic Concurrent Processes. *Fundamentae Informaticae 14*, 39-73, 1991.
- [GMA95] B. Grahlmann, M. Moeller, and U. Anhalt. A new Interface for the PEP tool - Parallel Finite Automata. In *Proc. of 2. AWPN'95 (Oldenburg)*, 1995.
- [Gol88] U. Goltz. On Representing CCS Programs by Finite Petri Nets. In *Proc. of MFCS'88*, M.P. Chytil, L. Janiga and V. Koubek (Eds.). LNCS 324, Springer, pp. 339-350, 1988.
- [Gra95] B. Grahlmann. PEP: A Programming Environment based on Petri Nets. In *Application and Theory of Petri Nets'95 - Tool Presentation (Torino)*, 1995.
- [Gra97a] B. Grahlmann. The PEP Tool. In *CAV'97 (Haifa)*, 1997.
- [Gra97b] B. Grahlmann. The Reference Component of PEP. In *Proc. of TACAS'97 (Enschede)*, LNCS 1217, Springer, 1997.
- [Gra98] B. Grahlmann. The state of PEP. In *AMAST'98 (Algebraic Methodology and Software Technology, Manaus Brazil)*, 1998.
- [GV03] C. Girault and R. Valk. *Petri Nets for Systems Engineering. A Guide to Modelling, Verification, and Applications*. Springer, 2003.

- [Han91] H. Hansson. *Time and Probability in Formal Design of Distributed Systems*. PhD thesis, Department of Computer Systems, Uppsala University, 1991.
- [Hav98] R.B. Haverkort. *Performance of Computer Communication Systems. A Model-Based Approach*. John Wiley and Sons, 1998.
- [Her02] H. Hermanns. *Interactive Markov Chains*. PhD thesis. LNCS 2428, Springer, 2002.
- [HHK02] H. Hermanns, U. Herzog, and J.P. Katoen. Process Algebras for Performance Evaluation. In *Theoretical Computer Science, Vol. 274 (1-2)*, pages 43-78, 2002.
- [HHMS00] H. Hermanns, U. Herzog, V. Mertsiotakis, and M. Siegle. Compositional performance modelling with the TIPPTool. In *Performance Evaluation (39)*, pp.5-35, 2000.
- [Hil94a] J. Hillston. *A Compositional Approach to Performance Modelling*. PhD thesis, Dep. of Computer Science, University of Edinburgh, April 1994.
- [Hil94b] J. Hillston. The nature of the synchronization. In *Int. Workshop on Process Algebra and Performance Modelling*, 1994.
- [Hil96] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.
- [Hoa85] C.A.R. Hoare. *Communicating Sequential Process*. Prentice Hall, 1985.
- [HR91] M. Hennesy and T. Regan. A process algebra for timed systems. Technical Report CS0591, University of Sussex, 1991.
- [HR94] H. Hermanns and M. Rettelbach. Syntax, Semantics, Equivalences and Axioms for MTIPP. In *Proc. of the 2nd Workshop on Process Algebra and Performance Modelling*, U. Herzog and M. Rettelbach, (Eds.) Regensburg/Erlangen, pp.71-88, 1994.
- [HRRS01] J. Hillston, L. Recalde, M. Ribaudó, and M. Silva. A Comparison of the Expressiveness of SPA and Bounded SPN Models. In *Proc. 9th IEEE Int. Workshop on Petri Nets and Performance Models, (PNPM'01)*, IEEE Computer Society Press, 2001.

- [Jai91] R. Jain. *The Art of Computer Systems Performance Analysis. Techniques for Experimental Design, Measures, Simulation, and Modelling*. John Wiley and Sons, 1991.
- [JL92] R. Janicki and P.E. Laurer. *Specification and Analysis of Concurrent Systems - the COSY Approach*. EATCS Monographs in Theoretical Computer Science, Springer, 1992.
- [KB99] M. Koutny and E. Best. Fundamental Study: Operational and Denotational Semantics for the Box Algebra. In *Theoretical Computer Science, 211*, pages 1–83, 1999.
- [KEB94] M. Koutny, J. Esparza, and E. Best. Operational Semantics for the Petri Box Calculus. In *Proc. of CONCUR'94*, B. Jonsson and J. Parrow (Eds.), LNCS 836, Springer, pp. 210-225, 1994.
- [Kle75] L. Kleinrock. *Queueing Systems. Volume I: Theory*. John Wiley and sons, 1975.
- [Kle76] L. Kleinrock. *Queueing Systems. Volume II: Computer Applications*. John Wiley and sons, 1976.
- [Kou00] M. Koutny. A Compositional Model of Time Petri Nets. *International Conference on Theory and Application of Petri Nets, 2000, LNCS 1825*, pp. 303-322, Springer, 2000.
- [KS60] J.G. Kemeny and J.L. Snell. *Finite Markov Chains*. Van Nostrand, 1960.
- [LdFN96] L. Llana, D. de Frutos, and M. Núñez. Testing semantics for urgent timed algebras. In *3rd AMAST Workshop on Real-Time Systems*, pp. 33-45, 1996.
- [LL97] L. Léonard and G. Leduc. An introduction to ET-LOTOS for the description of time-sensitive systems. *Computer Networks and ISDN Systems, 29*, pp. 271-292, 1997.
- [Lóp03] N. López. *Semántica para álgebras de procesos estocásticos no markovianos*. PhD thesis, Universidad Complutense de Madrid, 2003.
- [LS91] K. Larsen and A. Skou. Bisimulation through Probabilistic Testing. *Information and Computation, 94(1):1-28*, 1991.

- [Mac99] H. Macià. Una introducción al álgebra de cajas. Trabajo de doctorado. Dpto. de Informática. Universidad de Castilla - La Mancha, 1999.
- [Mac00] H. Macià. Análisis de especificaciones algebraicas mediante PBC. Trabajo de doctorado. Dpto. de Informática. Universidad de Castilla - La Mancha, 2000.
- [Mar98] O. Marroquín. PBC: Petri Box Calculus (el Cálculo de Cajas de Petri). Trabajo de doctorado. Dpto. de Sistemas Informáticos y Programación. Universidad Complutense de Madrid, 1998.
- [McM92] K.L. McMillan. Using Unfoldings to Avoid the State Explosion Problem in the Verification of Asynchronous Circuits. In *Proc. 4th Workshop on Computer Aided Verification*, 164-174, 1992.
- [MdF01] O. Marroquín and D. de Frutos. Extending the Petri Box Calculus with Time. In *Proc. Int. Conf. on Theory and Application of Petri Nets 2001. LNCS 2075, Springer*, pp. 195-207, 2001.
- [Mel94] S. Melzer. Design and Implementation of a C-Code Generator for $B(PN)^2$. In *Proc. of Concurrency, Specification and Programming - Berlin*, 1994.
- [Mer74] P. Merlin. *A Study of the Recoverability of Communication Protocols*. PhD thesis, Dep. of Computer Science, University California, 1974.
- [Mil80] R. Milner. *A Calculus of Communicating Systems*. LNCS 92, Springer-Verlag, 1980.
- [Mil89] R. Milner. *Communication and Concurrency*. Prentice-Hall International, 1989.
- [MMSS96] C. Morgan, K. McIver, K. Seidel, and J.W. Sanders. Refinement-oriented probability for CSP. In *Formal Aspects of Computing 8(6)*, pp. 617-647, 1996.
- [Mol81] M.K. Molloy. *On the Integration of Delay and Throughput Measures in Distributed Processing Models*. PhD thesis, UCLA, Los Angeles, CA, 1981.
- [MT90] F. Moller and C. Tofts. A temporal calculus of communicating systems. In *CONCUR'90, LNCS 458, Springer*, pp. 401-415, 1990.

- [MV01] H. Macià and V. Valero. Pep. Una herramienta para traducir y analizar especificaciones de PBC con redes de Petri. In *Proc. XI Escuela de Verano. Sistemas Distribuidos: Modelos y Aplicaciones*. Albacete, pp. 187-202, 2001.
- [MVC02] H. Macià, V. Valero, and F. Cuartero. A Congruence relation in finite sPBC. Technical Report, DIAB-02-01-31, Department of Computer Science, University of Castilla-La Mancha, October 2002.
- [MVCC03] H. Macià, V. Valero, D. Cazorla, and F. Cuartero. Introducing the iteration in sPBC. Technical Report, DIAB-03-01-37, Department of Computer Science, University of Castilla-La Mancha, September 2003.
- [MVCdF03] H. Macià, V. Valero, F. Cuartero, and D. de Frutos. A Congruence relation in finite sPBC. *Submitted for publication*, 2003.
- [MVCP02a] H. Macià, V. Valero, F. Cuartero, and F.L. Pelayo. Improving the synchronization in Stochastic Petri Box Calculus. In *Proc. II Jornadas sobre Programación y Lenguajes (PROLE'02)*, pp. 7-22, El Escorial, 2002.
- [MVCP02b] H. Macià, V. Valero, F. Cuartero, and F.L. Pelayo. A new proposal for the synchronization in sPBC. Technical Report, DIAB-02-01-26, Department of Computer Science, University of Castilla-La Mancha, June 2002.
- [MVCP03] H. Macià, V. Valero, F. Cuartero, and F.L. Pelayo. A new proposal for the synchronization in sPBC. In *Proc. Third IEEE Int. Conference on Application of Concurrent to System Design (ACSD'03)*, pp. 216-225, IEEE Computer Society Press, 2003.
- [MVdF01a] H. Macià, V. Valero, and D. de Frutos. sPBC: A Markovian Extension of Finite Petri Box Calculus. In *Proc. 9th IEEE Int. Workshop on Petri Nets and Performance Models (PNPM'01)*, pp. 207-216. IEEE Computer Society Press, 2001.
- [MVdF01b] H. Macià, V. Valero, and D. de Frutos. sPBC: A Markovian Extension of finite Petri Box Calculus. In *IX Jornadas de Concurrencia, Sitges*, pp. 243-256, 2001.

- [MVdFC01] H. Macià, V. Valero, D. de Frutos, and F. Cuartero. Extending PBC with Markovian Multiactions. In *Proc. XXVII Conferencia Latinoamericana de Informática (CLEI'01), Mérida (Venezuela)*, 2001.
- [NdFL95] M. Núñez, D. de Frutos, and L. Llana. Acceptance trees for probabilistic processes. In *CONCUR'95, LNCS 962*, pp. 249-263, 1995.
- [Net03] Petri Nets. *Petri Nets World. CPN Group University of Aarhus (Denmark)*. <http://www.daimi.au.dk/PetriNets/>, 2003.
- [NS91] X. Nicollin and J. Sifakis. An overview and synthesis on Timed Process Algebras. In *Computer Aided Verification'91, LNCS 575*, Springer, pp. 376-398, 1991.
- [Núñ96] M. Núñez. *Semántica de Pruebas para Álgebras de Procesos Probabilísticas*. PhD thesis, Universidad Complutense de Madrid, 1996.
- [Núñ01] M. Núñez. Modelización de procesos estocásticos markovianos mediante álgebras de procesos. In *Proc. XI Escuela de Verano. Sistemas Distribuidos: Modelos y Aplicaciones*. Albacete, pp. 35-50, 2001.
- [Old91] E.R. Olderog. *Nets, Terms and Formulas*. Cambridge Tracts in Theoretical Computer Science 23, Cambridge University Press, 1991.
- [PCVC00] F Pelayo, F Cuartero, V. Valero, and D. Cazorla. An example of performance evaluation by using the stochastic process algebra ROSA. In *7th International Conference on Real-Time Systems and Applications pp. 271-278, IEEE Computer Society Press*, 2000.
- [Pet62] C.A. Petri. *Kommunikation mit Automaten*. PhD thesis, Schriften des IMN, Institut für Instrumentelle Mathematic, Bonn, 1962.
- [Pet81] J.L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, 1981.
- [Plo81] G. D. Plotkin. A Structural Approach to Operational Semantics. Technical Report DAIMI FN-19, Computer Science Department, University of Aarhus, 1981.
- [PRC01] F.L. Pelayo, M.C. Ruiz, and D. Cazorla. PEPA Workbench: Una herramienta para la evaluación de prestaciones basada en álgebras de

- procesos. In *Proc. XI Escuela de Verano. Sistemas Distribuidos: Modelos y Aplicaciones*. Albacete, pp. 171-186, 2001.
- [PS89] J.L. Peterson and A. Silberschatz. *Sistemas Operativos. Conceptos Fundamentales*. Reverté, 1989.
- [QdFML94] J. Quemada, D. de Frutos, C. Miguel, and L. Llana. A timed LOTOS extension. In *Theories and Experiences for Real-Time System Development, Amat Series in Computing*, World Scientific, pp. 239-263, 1994.
- [Rei88] W. Reising. *Petri Nets: an Introduction*. EATCS Monographs on Theoretical Computer Science Vol.4. Springer - Verlag, 1988.
- [Rib95] M. Ribaudó. Stochastic Petri Net Semantics for Stochastic Process Algebra. In *Proc. 6th Int. Workshop on Petri Nets and Performance Models, (PNPM'95)*, Durham, 1995.
- [Ros83] S.M. Ross. *Stochastic Processes*. John Wiley and Sons, 1983.
- [RR83] G.M. Reed and A.W. Roscoe. A Timed Model for Communicating Sequential Processes. *Theoretical Computer Science*, 58, pp. 249-261, 1983.
- [Sch84] P. Schweitzer. *Aggregation Methods for Large Markov Chains*. In G. Iazeolla, P.L. Courtois, and A. Hordijk, editors, *Mathematical Computer Performance and Reachability*. North Holland, 1984.
- [Sch95] S. Schneider. An operational semantics for timed CSP. *Information and Computation*, 116, pp. 193-213, 1995.
- [Sil88] M. Silva. *Las Redes de Petri: en la Automática y la Informática*. AC, 1988.
- [Sta01] P.H. Startke. INA: Integrated Net Analyzer. In <http://www.informatik.hu-berlin.de/starke/ina.html>, 2001.
- [Ste94] W.J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, NJ, 1994.
- [Tau89] D. Taubner. Finite Representation of CCS and TCSP Programs by Automata and Petri Nets. *LNCS 369*, Springer, 1989.

- [Val90] V. Valero. *Introducción a las Redes de Petri*. Departamento de Informática y Automática. Facultad de Ciencias Matemáticas de la Universidad Complutense de Madrid, 1990.
- [VB01] V. Valero and A. Bueno. Modelado de sistemas concurrentes con redes de Petri. In *Proc. XI Escuela de Verano. Sistemas Distribuidos: Modelos y Aplicaciones*. Albacete, pp. 83-98, 2001.
- [vdA93] W.M.P. van der Aalst. Interval Timed Coloured Petri nets and their Analysis. *Lecture Notes in Computer Science*, 691:451-472, 1993.
- [vGSST90] R. van Glabbeek, S.A. Smolka, B.U. Steffen, and C.M.N. Tofts. Reactive, generative and stratified models of probabilistic processes. In *Proc. 5th Annual Symposium on Logic in Computer Science*, pp. 130-141, 1990.
- [Wal83] B. Walter. Timed Petri-Nets for Modelling and Analysis Protocols with Real-Time Characteristics. In *Proc. 3rd IFIP Workshop on Protocol Specification, Testing and Verification*, North-Holland, 1983.
- [WDF85] C.Y. Wong, T.S. Dillon, and K.E. Forward. Timed places Petri nets with stochastic representation of place time. In *Proc. Intern. Workshop on Timed Petri Nets*, 1985.
- [Yi90] W. Yi. Real time behavior of asynchronous agents. In *CONCUR'90*, LNCS 458, Springer, pp. 502-520, 1990.
- [YL92] W. Yi and K. Larsen. Testing probabilistic and nondeterministic processes. In *Protocol Specification, Testing and Verification XII*, pp. 47-61, 1992.