Semánticas de Pruebas para Álgebras de Procesos Probabilísticos

Manuel Núñez García

7 de Mayo de 1996

Tesis Doctoral presentada al Dpto. de Informática y Automática de la Universidad Complutense de Madrid para la obtención del grado de Doctor en Ciencias Matemáticas.

Prefacio

Al terminar mis estudios de Licenciatura en la Facultad de Ciencias Matemáticas en Julio de 1992 ya me había planteado la posibilidad de realizar unos estudios de doctorado. Lo que naturalmente no tenía nada claro era el *tema* de una posible *tesis*, cosa habitual pues al terminar sus estudios de Licenciatura los estudiantes[1] suelen tener una idea bastante vaga sobre lo que representa la investigación en el campo de sus estudios, y más específicamente sobre los temas de investigación que se cultivan en el Departamento en el que desean realizar sus estudios de doctorado. Después de los dos primeros meses de cursos de doctorado, las cosas seguían sin estar nada claras. Sospecho que en buena parte ello se debía a lo difícil que resulta integrarse (bajo mi punto de vista) en el mundo de la investigación, si no se tiene una vinculación más o menos formal con la Universidad. La situación cambió radicalmente en el momento en el que conseguí una plaza de ayudante de Escuela Universitaria, lo que sucedió en Diciembre de 1992. Tras las vacaciones de Navidad empecé a *integrarme* en el Departamento. A finales de Enero, David me comentó que acababa de recibir una tesis Doctoral (en concreto, la tesis de Ivan Christoff) en la que se estudiaba un modelo de procesos concurrentes en los que se incluía información probabilística. En aquellos momentos mis conocimientos de concurrencia eran bastante limitados. De hecho se puede decir que se reducían a una breve introducción a las álgebras de

[1] Muy probablemente ello es especialmente cierto para los alumnos de la especialidad de Ciencias de la Computación, que por limitaciones en el tiempo que han dedicado al estudio de la Informática, y por estar dirigida dicha especialidad a una formación general en temas de programación, apenas han tenido contacto con las cuestiones relacionadas con la Teoría de la Programación que de una u otra forma constituyen el núcleo de la investigación del Departamento con componentes matemáticas.

procesos CCS y CSP, y a un par de artículos sobre semánticas para CSP. Sobre el tema de las *probabilidades* en procesos concurrentes mi conocimiento se reducía al conjunto vacío !! Quedamos en el compromiso de echarle un vistazo a esta tesis y si finalmente no entendía nada pasar a otras cosas. En consecuencia comencé a trabajar en la tesis de Christoff y el hecho fue que el tema me pareció muy interesante. A continuación seguí mi estudio con otros trabajos sobre concurrencia y probabilidades. Por supuesto que el trabajo que técnicamente más influyó en mi investigación posterior fue el correspondiente a la tesis de Fernando Cuartero, que Fernando estaba terminando entonces bajo la dirección de David.

Tengo que agradecer sinceramente mi interés por los modelos concurrentes probabilísticos a estas dos tesis, pues su calidad y claridad fue en buena medida la causa de que continuara trabajando en el tema y de que finalmente pudiera escribir la presente tesis. Posiblemente, si no hubiera tenido la suerte de encontrarme con estos dos trabajos en aquellos momentos, esta tesis nunca se habría realizado y yo habría dirigido mis pasos a otro tema de investigación.

Hemos llegado a Agosto de 1993. Yo me encontraba en Moscú por vacaciones y se me ocurrió la idea de intentar realizar una representación del modelo de Fernando utilizando sistemas probabilísticos con transiciones etiquetadas. Es decir, pretendía juntar los dos modelos probabilísticos de los que más conocimiento tenía: el de Ivan Christoff y el de Fernando Cuartero. Estas ideas quedaron plasmadas en [Núñ93], donde se entremezclan, de forma bastante caótica, resultados *científicos* con otros temas que no tienen nada que ver con mi investigación. Como quiera que se trataba de mi primer trabajo, y el mismo fue realizado totalmente en solitario, los resultados fueron bastante decepcionantes. El principal problema que se me planteaba era el de representar de forma finita procesos en los cuales aparecía el operador paralelo en el ámbito de una recursión. Como es bien sabido esto no puede hacerse, dado que la recursión en el ámbito de un operador paralelo puede dar lugar a infinitos estados (claro que en aquellos momentos yo no era de los que lo sabía !!). Sin embargo, conseguí definir un complejo formalismo que *parecía* funcionar en parte. La clave consistía en llevar cuenta de las veces que se había *desplegado* la recursión en el ámbito de un operador paralelo, para lo cual había que aumentar la potencia de los

estados de los sistemas de transiciones. Cuando mis *mayores* me informaron de que estaba intentando hacer algo que en general resultaría imposible abandoné esta vía, aunque creo que con los adecuados retoques formales se podrían conseguir resultados muy interesantes sobre la base de este primer trabajo. Quien sabe si en el futuro tendré tiempo y oportunidad para retomar esta vía de investigación.

El siguiente punto de inflexión se produce durante la lectura de la tesis de Fernando Cuartero en Octubre de 1993. Yo llevaba tiempo pidiéndole a David que pensara en algún trabajo de doctorado para mi. En la lectura, una de las *pegas* que algún miembro del tribunal planteó a Fernando fue la falta de ejemplos de aplicación de su modelo. A la salida de la lectura David me dijo: "Pues bien, si te atreves ya tienes trabajo de doctorado". Lo que parecía un trabajo *sencillo*, consistente en desarrollar una serie de programas, se complicó notablemente. Ello fue así pues buena parte de los ejemplos que me interesaba abordar precisaban un repetido uso del operador paralelo, y más exactamente de la composición en paralelo de n procesos. Pero por razones técnicas el operador paralelo del modelo de Fernando no es en general asociativo, por lo que tuve que comenzar por investigar propiedades de conjuntos de procesos compuestos en paralelo que aseguraran la *asociatividad* del operador sobre ellos, y ello constituyó a la postre la parte más creativa y laboriosa del trabajo. Finalmente, el trabajo [Núñ94] pudo terminarse y presentarse cumpliendo las expectativas al respecto.

Una vez acabado dicho trabajo había que empezar a pensar en un tema para mi tesis, y de nuevo la mejor fuente de inspiración la constituyeron los trabajos conocidos. Aunque el tratamiento matemático en el trabajo de Fernando era realmente excelente, había algunas decisiones de *diseño* que no me convencían en absoluto. Mis objeciones principales eran dos: la *pérdida* parcial de la información probabilística en el operador de elección externa y la forma no habitual en que se define la semántica de pruebas. Yo ya tenía conocimiento del modelo generativo y me parecía que el mismo era bastante más intuitivo que el modelo *reactivo*, que era el utilizado por Fernando con unas ligeras modificaciones debidas al hecho de utilizarse un lenguaje basado en CSP.

En consecuencia mis primeros pasos se dirigieron al estudio de cuáles serían las consecuencias de considerar una interpretación generativa de las probabilidades. Respecto a la semántica de pruebas, Fernando la definía de una forma composicional,

lejos por tanto de la definición habitual basada en la adecuada semántica operacional. Poniendo todo junto afronté la definición de un modelo probabilístico con una interpretación generativa de las probabilidades basado en una semántica de pruebas definida de forma operacional. Tras seis meses de duro trabajo, obtuvimos la deseada semántica de pruebas y dos caracterizaciones alternativas de la misma. Ello constituyó el núcleo de un artículo que fue enviado al congreso CONCUR'95. El hecho de que el mismo fuera aceptado fue un gran estimulante para seguir adelante con el trabajo, amén de una *bendición* a las ideas en que estaba basado. De hecho, la parte central de esta tesis es una versión extendida (a unas 70 páginas) del citado artículo[2].

Durante el resto del año 95 continuó el trabajo en este modelo, que por mi parte alterné con actividades en otros campos de investigación, como la Programación Funcional. En concreto se desarrolló una semántica axiomática para el modelo generativo, se hizo una interpretación reactiva de las probabilidades, y se analizaron una serie de ejemplos de aplicación del nuevo lenguaje a la especificación de sistemas concurrentes que dependieran de una u otra forma de información probabilística.

La conjunción de todo este trabajo ha dado lugar a lo que finalmente es la presente tesis.

[2]Un *referee* dijo que sería oportuno incluir en el artículo la demostración completa de alguno de los teoremas. El problema era que la demostración completa de uno de los teoremas centrales del artículo ocupa más de 15 páginas en esta tesis !!

Agradecimientos

Me gustaría agradecer a

... mis padres por todo el apoyo que me han dado durante mis estudios y especialmente en el momento en el que decidí quedarme en la Universidad y dedicarme a escribir una tesis doctoral en lugar de optar por otras ocupaciones posiblemente más lucrativas.

... mi director de tesis David de Frutos por todo el trabajo desarrollado en esta tesis desde que este proyecto empezó y por haber soportado estoicamente mis *enfados*.

... los miembros del tribunal de esta tesis Joaquim Gabarró, Albert Llamosí, Ricardo Peña, Mario Rodríguez y Scott A. Smolka por haber aceptado participar en este tribunal y haber realizado el esfuerzo de leer más de 200 páginas de un tema poco divertido.

... mis compañeros del Departamento de Informática y Automática por haberme ayudado en estos casi cuatro años siempre que lo he solicitado. En especial a mis compañeros de despacho Carlos y Pedro por aguantar mis humos (los del tabaco y también los malos) todos los días, y a Jesús por su ayuda en la realización de nuestros *seminarios*.

... mis demás amigos y familiares por ayudarme a olvidar por algún momento todo este trabajo y por recordarme que de vez en cuando hay cosas más importantes que escribir una tesis o un artículo. Merecen una mención especial Marco por sus recorridos *turísticos* por esa ciudad tan maravillosa que es New York, y Cordi por acompañarme y ayudarme en los lugares más variopintos (desde comprar algún сувенир en el ГУМ hasta comer unas pizzas deliciosas en Slovenia) и так далее...

Sumario

Indice de figuras

Capítulo 1

Introducción

El presente trabajo está dedicado al estudio de modelos formales de procesos concurrentes en los que se ha introducido una cierta información probabilística. Usualmente, en el primer capítulo de una tesis se comienza por justificar el desarrollo teórico que vendrá a continuación. En nuestro caso se trataría de describir la utilidad de los modelos de procesos concurrentes para describir la *realidad*. A continuación se concretaría más, explicando posiblemente de un modo *informal*, la forma de incluir información probabilística en modelos concurrentes para conseguir una forma ajustada de describir los procesos *reales*, y finalmente se daría un resumen de los contenidos del trabajo. En nuestro caso nos centraremos en el último de estos puntos, puesto que consideramos que en trabajos anteriores ya ha sido amplia y ajustadamente comentada la utilidad de los procesos concurrentes en los cuales se incluye información probabilística (e.g. [Chr90b, Han91, Sei92, Chr93, Low93, Cua93]). En lo que se refiere a una descripción informal del uso de las probabilidades, consideramos magnífica la introducción de la tesis de Linda Christoff [Chr93] donde se presenta una *Overview for Mom*. Tan excelente nos parece, que al respecto creemos que lo mejor que hubiéramos podido hacer sería una mera traducción de lo escrito allí, que en consecuencia hemos decidido omitir.

1.1 Estado del Arte

En esta sección presentamos un recorrido por los distintos modelos que han sido propuestos para tratar sistemas concurrentes en los cuales ha sido añadida una cierta información probabilística, centrándonos especialmente en los más relacionados con las álgebras de procesos. Este estudio ha pretendido ser lo más amplio posible, siendo nuestro objetivo el de tratar especialmente las propuestas de extensiones probabilísticas de modelos de especificación relacionados directamente con las álgebras de procesos, para dar lugar así a una buena guía de referencia para investigadores en este área. Empezaremos reseñando los primeros trabajos en los que se estudiaron sistemas con probabilidades. Después pasaremos a estudiar las extensiones probabilísticas de las álgebras de procesos clásicas (CCS, CSP, ACP, LOTOS), y continuaremos con otros trabajos realizados en el ámbito de los procesos probabilísticos (es decir la extensión con probabilidades de los sistemas de transiciones etiquetadas) y algunos trabajos recientes sobre autómatas probabilísticos. Además, se repasan algunos de los trabajos que se han realizado sobre el tema, bastante relacionado con las probabilidades, de las prioridades. Esta última parte no es, ni lo pretende, ni mucho menos *completa*, dado que no conocemos dicho campo con la profundidad que se requería para ello.

1.1.1 Primeros trabajos

La inclusión de información probabilística en modelos no es una cosa reciente. Ya en 1963, Rabin [Rab63] modifica la noción clásica de *autómatas* incluyendo información probabilística. Hemos de avanzar hasta la década de los 80 para encontrar diversos estudios sobre la verificación de programas probabilísticos y la definición de lógicas temporales con información probabilística (e.g. [FH82, Koz83, HS84, Var85, JP89, HJ89]), pero su marco de trabajo está muy alejado del nuestro.

No es hasta 1989 cuando aparecen los primeros trabajos en los cuales las probabilidades se estudian en un marco cercano al de las álgebras de procesos. En [LS89, LS91]

se extiende la noción de sistemas de transiciones etiquetadas con información probabilística, considerando una interpretación *reactiva* de las probabilidades. Con ello se consigue caracterizar la bisimulación [Mil80, Par81] usando una noción de pruebas probabilísticas[1], en el sentido de que con una probabilidad arbitrariamente cercana a 1, existe una prueba que distingue a dos procesos que no son equivalentes bajo bisimulación. Finalmente, extienden la noción de bisimulación, definiendo una nueva relación de equivalencia entre procesos probabilísticos: la bisimulación probabilística.

En [BM89] se extiende el estudio realizado en [LS89], probando que a dos sistemas de transiciones etiquetadas (en principio sin información probabilística) se les puede asignar *pesos* de forma que los nuevos sistemas obtenidos son indistinguibles para una noción muy general de pruebas probabilísticas si y solamente si estos sistemas son bisimilares. Además, se muestra un par de procesos computables bisimilares los cuales no son equivalentes bajo la noción de pruebas probabilísticas para ninguna asignación computable de *pesos* a sus transiciones.

De todas formas, no es hasta 1990 cuando empiezan a aparecer los primeros trabajos en el ámbito de las álgebras de procesos probabilísticas.

1.1.2 Modelos basados en CCS

CCS [Mil80, Mil89] es, junto con CSP, el álgebra de procesos más utilizada para la especificación y verificación de propiedades en sistemas concurrentes. La forma comúnmente utilizada para dar semántica a lenguajes basados en CCS es mediante una semántica operacional [Plo81], a partir de la cual se introduce la noción de bisimulación. Siguiendo con dicho hábito, los modelos probabilísticos de CCS utilizan normalmente semánticas operacionales en lugar de semánticas denotacionales.

El primer trabajo en el ámbito de las álgebras de procesos probabilísticas aparece

[1] La bisimulación se puede caracterizar mediante una semántica de pruebas no probabilística [Abr87], pero para ello hay que ampliar notablemente la capacidad de las pruebas, permitiendo usar múltiples copias del proceso que está siendo *probado*, para que la prueba pueda experimentar con una copia cada vez.

en 1990. En [GJS90] se presenta una versión probabilística de SCCS [Mil83] (Synchronous CCS) a la que llaman PCCS. En este trabajo, la suma no determinista de procesos pasa a ser una suma probabilística de la forma $\sum_{i\in I}[p_i]E_i$, donde $p_i \in (0,1]$ y $\sum p_i = 1$. El proceso así obtenido ofrece una elección probabilística entre los procesos E_i, de modo que cuando una acción se puede ejecutar por más de uno de estos procesos, se elige entre ellos proporcionalmente a las probabilidades p_i de cada proceso E_i.

Al estar basado en CCS, no se distingue entre la elección interna y la elección externa, por lo que para simular una elección interna entre dos procesos se ha de utilizar la acción τ como prefijo, con lo que se representa el proceso $P \oplus_p Q$ (según la notación utilizada en esta tesis) en la forma $[p]\tau.P + [1-p]\tau.Q$.

En [vGSST90, vGSS95], utilizando PCCS, se definen los modelos *reactivo*, *generativo* y *estratificado* para procesos probabilísticos.

Definen el modelo *reactivo* [LS89] como áquel donde el entorno sólo puede ofrecer una acción cada vez. La semántica que dan es para un subconjunto de PCCS en el que las *sumas* (operador de elección) deben estar *guardadas* por probabilidades y acciones, y se elimina la función de renombramiento.

Si el proceso puede ejecutar la acción ofrecida por el entorno, entonces hace una transición interna de acuerdo a una cierta distribución de probabilidad. Por ejemplo, consideremos el proceso $S = \frac{1}{4}a.P + \frac{3}{4}a.Q + 1b.R$, y supongamos que el entorno ofrece la acción a. En tal caso, S ejecutará la acción a y pasará a comportarse como P con probabilidad igual a $\frac{1}{4}$, mientras que con probabilidad $\frac{3}{4}$ ejecutará a y pasará a comportarse como Q. Si en cambio el entorno ofreciera la acción b, entonces S ejecutaría dicha acción pasando a comportarse como R con probabilidad 1.

Para este lenguaje se da una semántica operacional donde $P \xrightarrow{\alpha[p]} P'$ denota que el proceso P puede ejecutar la acción α y con probabilidad p convertirse en P'. Por ejemplo, la semántica del operador de elección viene dada por

$$\sum_{i\in I}[p_i]\alpha_i.E_i \xrightarrow{\alpha_i[p_i/r_i]}_k E_i \quad \text{donde} \quad r_i = \sum_{\substack{j\in I \\ \alpha_j=\alpha_i}} p_j$$

El índice k sirve para distinguir entre dos transiciones iguales (por ejemplo, para distinguir entre las dos transiciones iguales que produce el proceso $\frac{1}{2}a.nil + \frac{1}{2}a.nil$).

En el modelo *generativo*, el entorno puede ofrecer una elección entre dos o más acciones y el proceso realiza la elección de acuerdo a alguna distribución de probabilidad. Por ejemplo, si el entorno ofrece una a o una b al proceso $\frac{1}{6}a.P + \frac{1}{3}a.Q + \frac{1}{2}b.R$, entonces este proceso ejecutará la acción a con probabilidad $\frac{1}{2}$ y la acción b con probabilidad $\frac{1}{2}$. Si se escoge la acción a, entonces el proceso pasa a comportarse como P con probabilidad $\frac{1}{3}$ y como Q con probabilidad $\frac{2}{3}$. Si sólo se hubiese ofrecido una a, el proceso la ejecutaría, pasando a comportarse como P con probabilidad $\frac{1}{3}$ y como Q con probabilidad $\frac{2}{3}$.

Se define una semántica operacional cuyas transiciones tienen la forma $P \xrightarrow{\alpha[p]} P'$, lo que significa que con probabilidad p el proceso P ejecutará la acción α y se convertirá en P'. La regla para la elección es:

$$E_j \xrightarrow{\alpha[q]}_k E' \implies \sum_{i \in I} [p_i]E_i \xrightarrow{\alpha[p_j \cdot q]}_{j \cdot k} E' \ \ (j \in I)$$

Para definir la semántica del operador de restricción se introduce una función que define la probabilidad con la que el proceso E ejecutará una acción del conjunto A.

$$\nu_G(E, A) = \sum \{\!|\, p_i \,|\!\} \exists \alpha, E_i : \ E \xrightarrow{\alpha[p_i]}_i E_i \ \wedge \ \alpha \in A \, \}\!\}$$

Entonces la restricción se define mediante la siguiente regla:

$$E \xrightarrow{\alpha[p]}_i E' \implies E \restriction A \xrightarrow{\alpha[p/r]}_i E' \restriction A \quad \text{donde } \alpha \in A \ \wedge \ r = \nu_G(E, A)$$

El modelo *estratificado* permite mayor control sobre las elecciones probabilísticas. Supongamos que queremos especificar un proceso que debe elegir, de forma *externa*, entre ejecutar la acción a con probabilidad $\frac{1}{3}$, eligiéndose en otro caso entre b y c con igual probabilidad. Si definimos este proceso en la forma $\frac{1}{3}a + \frac{1}{3}b + \frac{1}{3}c$ y suponemos que el entorno no puede ofrecer la c, entonces (en el modelo generativo) el proceso elegiría entre a y b con igual probabilidad (es decir con probabilidad $\frac{1}{2}$) en lugar de elegir con probabilidades $\frac{1}{3}$ y $\frac{2}{3}$ como se quería especificar. Para solucionar este problema deberíamos haber especificado el proceso en la forma $\frac{1}{3}a + \frac{2}{3}(\frac{1}{2}b + \frac{1}{2}c)$ sin

realizar el *producto* de las probabilidades. En tal caso, si el entorno no ofrece la acción c, entonces con probabilidad $\frac{1}{3}$ se ejecutará la acción a y con probabilidad $\frac{2}{3}$ la acción b. Por tanto, en el modelo estratificado se permiten elecciones entre procesos arbitrarios. La semántica operacional se define mediante dos relaciones:

- Una relación de *transición de acciones*, que se denota por $P \xrightarrow{\alpha} Q$, la cual tiene la definición normal excepto que no existe una regla para la elección.
- Una relación de *transición de probabilidades*, que se denota por $P \stackrel{p}{\longmapsto} Q$, la cual indica que con probabilidad p, el proceso P pasará a comportarse como el proceso Q.

La regla para el operador de elección es

$$\sum_{i \in I} [p_i] E_i \stackrel{p_i}{\longmapsto}_i E_i$$

Para definir la semántica del operador de restrición se introduce una función que nos da la suma de las probabilidades de las transiciones correspondientes a cada conjunto A:

$$\nu_S(E, A) = \begin{cases} 1 & \text{si } E \xrightarrow{\alpha} \wedge\ \alpha \in A \\ 0 & \text{si } E \xrightarrow{\beta} \wedge\ \beta \notin A \\ \sum \{\!\{\, p_i \mid E \stackrel{p_i}{\longmapsto}_i E_i \ \wedge\ \nu_S(E_i, A) \neq 0 \,\}\!\} & \text{e.o.c.} \end{cases}$$

Entonces se define la restricción en la forma

$$E \stackrel{p}{\longmapsto}_i E' \ \wedge\ \nu_S(E', A) \neq 0 \ \Rightarrow\ E \restriction A \stackrel{p/r}{\longmapsto}_i E' \restriction A \quad \text{donde } r = \nu_S(E, A)$$

Para cada modelo se define el correspondiente concepto de bisimulación fuerte inducido por cada semántica operacional. Finalmente, se dan funciones de abstracción que permiten pasar de un modelo a otro.

Partiendo de este trabajo los modelos reactivo y generativo se han estudiado en profundidad, mientras que el modelo estratificado ha quedado un poco más olvidado

(lo cual puede deberse a la dificultad conceptual que implica el hecho de no poder agrupar sucesivas elecciones probabilísticas). De hecho, casi todos los trabajos posteriores están basados bien en el modelo reactivo o en el generativo, no habiendo aparecido prácticamente ninguna interpretación alternativa del significado de las probabilidades (una excepción vendría dada por nuestro modelo *generativo limitado* [NdF95c], el cual explicaremos más adelante).

En [JS90] se consideran las extensiones probabilísticas de varias nociones de equivalencia definidas para procesos no probabilísticos. Se consideran las extensiones de las nociones de *traza* [Hoa85], *traza maximal* [BW82], *fallos* [BR85], *fallos maximales*, *menú* [OH83] y *bisimulación* [Par81, Mil89] al caso probabilístico. Estas nociones se estudian primero en el ámbito de *sistemas probabilísticos de transiciones etiquetadas*, siendo traducidas después al marco del lenguaje PCCS.

Demuestran que, al contrario de lo que ocurría en el marco no probabilístico, la *maximalidad* de trazas o fallos no aumenta la capacidad de distinguir procesos con respecto a las equivalencias de trazas y fallos respectivamente. Además, en el caso probabilístico, la equivalencia de fallos y de menús coinciden. También demuestran que la equivalencia de trazas y la equivalencia de fallos no son congruencias, mientras que la bisimulación probabilística [LS89] sí lo es. Finalizan dando una axiomatización correcta y completa de la bisimulación probabilística.

En [Tof90] se define una versión con *pesos* de SCCS. Por ejemplo, el significado del proceso $mP + nQ$ $(m, n \in \mathbb{N})$ es que el proceso escogerá m veces a P por cada n veces que escoja a Q. La ventaja del uso de pesos, sobre el uso de probabilidades, es que no se necesita hacer reajustes de las mismas. Por ejemplo, la regla que se da para la restricción es:

$$\frac{E \stackrel{p}{\longmapsto}_i E' \ \wedge \ does_A(E')}{E \restriction A \stackrel{p}{\longmapsto} E' \restriction A}$$

donde el predicado $does_A(E')$ es cierto si E' puede ejecutar una acción del conjunto A. La definición formal de $does_A(E')$ es

$$\frac{a \in A \ \wedge \ E \xrightarrow{a} E'}{does_A(E)} \qquad \frac{E \overset{p}{\longmapsto} E' \ \wedge \ does_A(E')}{does_A(E)}$$

En un trabajo posterior [Tof94], se extiende el modelo anterior con una noción de tiempo, presentándose varios ejemplos que muestran la utilidad del modelo para la especificación de sistemas concurrentes.

En [HJ90, Han91] se define un modelo basado en CCS con probabilidades y con tiempo discreto (es decir, con rango los naturales). Los procesos alternan entre estados *probabilísticos* y estados de *acciones*. En los estados de acciones el proceso ofrece al entorno una elección entre un conjunto de acciones. Tras ejecutar una acción, el proceso pasa a un estado probabilístico. El entorno sólo puede ofrecer una acción en cada momento, tratándose por tanto de un modelo reactivo según [vGSST90]. Cuando el proceso se encuentra en un estado probabilístico, el proceso elige entre diversos estados de acciones de acuerdo a la distribución de probabilidad que se indica.

En este modelo se hace una diferencia entre la elección externa (que se denota por $\sum_{i \in I} \alpha_i P_i$) y la elección probabilística (que se denota por $\sum_{i \in I} [p_i] N_i$), la cual podemos entender como una elección interna generalizada con probabilidades.

En el lenguaje sin tiempo se denota por E_P a los estados probabilísticos y por E_N a los estados de acciones. Después se definen dos relaciones:

- $N \overset{\alpha}{\hookrightarrow} P$, que significa que N puede ejecutar la acción α y convertirse en N.
- $P \overset{p}{\mapsto} N$, que significa que P pasa a comportarse como N con probabilidad p.

La regla que se da para el operador de elección probabilística es

$$\sum_{i \in I} [p_i] N_i \overset{p}{\mapsto} N_i \quad \text{donde} \quad p = \sum_{\substack{N_j \equiv N_i \\ j \in I}} p_j$$

Como su elección probabilística es interna, las reglas para el operador de restricción son muy sencillas:

$$\frac{P \overset{p}{\mapsto} N}{P \backslash a \overset{p}{\mapsto} N \backslash a} \qquad \frac{N \overset{\alpha}{\hookrightarrow} P}{N \backslash a \overset{\alpha}{\hookrightarrow} P \backslash a} \ [\alpha, \bar{\alpha} \neq a]$$

En [HJ94] se extiende la lógica temporal definida en [CES86] de forma que se pueda demostrar la correción de sistemas que contengan información temporal y probabilística.

En [YL92] se extiende CCS con un operador probabilístico de elección interna. Basándose en la teoría clásica de semántica de pruebas [dNH84, Hen88], definen una noción de pruebas probabilísticas donde un proceso *pasa* una prueba con un conjunto de probabilidades. De esta forma, extienden las nociones de preorden *may* y *must* de forma que un proceso *puede pasar* una prueba si el supremo del conjunto de probabilidades con las cuales el proceso pasa la prueba es mayor que 1, mientras que un proceso *debe pasar* la prueba cuando el ínfimo de dicho conjunto es igual a 1. Además, muestran que cuando se restringen a una clase de procesos probabilísticos, al olvidar la información probabilística se recuperan los preórdenes clásicos sin probabilidades.

En [JHSY94, JY95] se dan caracterizaciones alternativas para los preordenes definidos en [YL92]. Las caracterizaciones están basadas en la noción de traza (para el caso del preorden *may*) y en la noción de fallo (para el caso del preorden *must*) de CSP.

En [LS92] se define un cálculo de probabilidades basado en SCCS, y se estudia la verificación composicional de procesos probabilísticos con una intepretación reactiva de las probabilidades. Además, definen una lógica probabilística (basada en la lógica de Hennessy-Milner [HM85]) la cual es utilizada para realizar la descomposición de procesos, y es lo suficientemente expresiva como para identificar procesos equivalentes bajo bisimulación probabilística. Finalmente, definen axiomatizaciones completas para el cálculo y para la lógica.

1.1.3 Modelos basados en CSP

En 1978, Hoare presenta el lenguaje CSP [Hoa78]. Se trataba de un lenguaje de programación concurrente, pero con un notable fundamento formal. A partir del

mismo se desarrolló más tarde el modelo TCSP[2] [Bro83, BHR84, Hoa85]. Si bien el salto cualitativo conceptual entre ambos lenguajes es notable, la primera versión fue de gran utilidad, y más adelante serviría en particular para basar en el mismo la definición del lenguaje Occam [Lim84].

Aunque la mayoría de las álgebras de procesos probabílisticas están basadas en CCS, en el ámbito de CSP hay también diversas propuestas interesantes, entre las que destacaremos los trabajos de Seidel, Lowe y Cuartero. En ellas veremos que, en contraste con lo ocurrido en las versiones probabilísticas de CCS, las semánticas para las extensiones probabilísticas de CSP suelen estar basadas en modelos denotacionales (normalmente en el modelo de fallos [BR85], o en el modelo de árboles de aceptación [Hen85]).

Karen Seidel [Sei92, Sei95] desarrolla un modelo probabilístico de CSP, definiendo una semántica en términos de medidas de probabilidad sobre el espacio de trazas infinitas. En su modelo, denota por $[\![P]\!]A$ a la probabilidad con la que el proceso P ejecuta una traza del conjunto A.

Los operadores se definen como transformaciones sobre las medidas de probabilidad. La definición esquemática de cada uno de sus operadores es la siguiente:

- El proceso $a \rightarrow P$ puede ejecutar una traza del conjunto A si P puede ejecutar una traza del conjunto A', donde $A' = \{s | \langle a \rangle s \in A\}$.
- La probabilidad con la que el proceso $P \ {}_p\sqcap \ Q$ ejecuta una traza del conjunto A es igual a p multiplicado por la probabilidad con la que el proceso P ejecuta una traza del conjunto A, más $1-p$ multiplicado por la probabilidad con la que el proceso Q ejecuta una traza del conjunto A.
- El proceso $P \parallel Q$ puede ejecutar una traza $w \in A$ si P y Q pueden ejecutar trazas u y v tales que $u = v = w$ ó u y v son iguales hasta una cierta posición y w es igual al prefijo común a u y v seguido por acciones τ.

[2]Originalmente se introdujo la T por "Theoretical" para diferenciar al modelo algebraico del lenguaje previo, pero posteriormente dicha inicial cayó en desuso. Más tarde reaparece en [RR86, RR88] pero ahora para expresar "Timed" en una extensión temporizada del modelo CSP.

Pero como resulta esperable se presentan problemas a la hora de definir la elección externa, pues con la definición propuesta, no es posible precisar la probabilidad con la que el proceso $a \rightarrow STOP \,\Box\, b \rightarrow STOP$ ejecuta la acción a.

En consecuencia, para definir la semántica de la elección externa se introducen medidas de probabilidad *condicionales*. Para cada proceso P, conjunto de trazas A, y traza y, la expresión $(\!|P|\!)(A, y)$ representa la probabilidad de que el proceso P ejecute una traza de A, suponiendo que el entorno ofrece la traza y (y nada más).

A partir de dicho concepto se define un operador de elección externa sin probabilidades. Informalmente, el proceso $P \,{}_S\Box\, Q$ funciona como P cuando se ofrece una traza de S, y como Q cuando la traza no está en S. Incluso en el marco de este complejo modelo, se ve obligada a limitar la definición al conjunto de trazas que verifican una cierta propiedad, quedando sin definir sobre el resto.

Los problemas continúan, pues sobre dicho modelo no se puede dar una definición semántica razonable para el operador de ocultamiento, dado que para cualquier traza y ofrecida por el entorno al proceso $P\backslash X$, la correspondiente traza y' que se ofrece a P no es una sola, sino cualquier traza y' tal que $y'\backslash X = y$. En consecuencia, deberíamos de partir de un modelo más complejo en el que se nos informe sobre el comportamiento del proceso P cuando se le ofrece un conjunto de trazas en lugar de una sola.

Gavin Lowe [Low93, Low95] define dos lenguajes basados en CSP temporal [RR86, RR88, DS95]: uno con prioridades, y otro con probabilidades. En la versión con prioridades de CSP temporal, para dar semántica a los términos sintácticos se calcula el conjunto de *comportamientos* que un proceso puede tener, donde la representación de un comportamiento incluye un registro de las prioridades dadas a cada acción. Cuando pasa al lenguaje con probabilidades, la semántica recoge las probabilidades con las que pueden suceder distintos comportamientos.

Pasando al estudio en detalle de los lenguajes propuestos, comienza introduciendo un operador de elección externa con prioridades. El proceso $P \,\boxdot\, Q$ funcionará como P si el entorno ofrece cualquiera de las acciones que P puede ejecutar en su primer paso, y como Q si el entorno ofrece un conjunto de acciones de entre las cuales Q

puede ejecutar alguna en su primer paso, sin que P pueda ejecutar ninguna de las acciones del conjunto en su primer paso.

De igual forma se definen operadores con prioridades asociados a los operadores paralelos. Dependiendo de la forma de sincronización entre los procesos, y de cual de las dos componentes de la composición paralela tiene mayor prioridad, se distinguen cuatro tipos de operadores para la composición paralela.

Después se pasa a definir el lenguaje con probabilidades. En dicho lenguaje se incluyen probabilidades en la elección interna en la forma usual, es decir, el proceso $P \,{}_p\sqcap_q\, Q$ funcionará como P con probabilidad p y como Q con probabilidad q (donde $p+q=1$).

La elección externa probabilística se define como un operador derivado de la elección interna y de la elección externa con prioridades; en concreto

$$P \,{}_p\Box_q Q \stackrel{\text{def}}{=} (P \,\mathbb{\Box}\, Q) \,{}_p\sqcap_q\, (Q \,\mathbb{\Box}\, P)$$

Sin embargo, no añade probabilidades para los operadores de composición paralela, de modo que su semántica permanece similar a la que tenían en el modelo con prioridades.

Fernando Cuartero [Cua93, CdFV96] presenta una variante de CSP con probabilidades, en la cual éstas son interpretadas de una forma *reactiva* (en términos de [vGSST90]). Por lo tanto, en la elección externa las probabilidades sólo tienen sentido cuando los dos procesos compuestos pueden ejecutar acciones comunes en su primer paso. Si los procesos no tienen acciones comunes en su primer paso, las probabilidades no influyen en la definición de la elección externa. Por ejemplo, si el entorno ofrece la acción a al proceso $P = [\frac{1}{3}]a \to P_1 \Box [\frac{2}{3}]a \to P_2$, entonces P ejecutará la acción a y pasará a comportarse como P_1 con probabilidad $\frac{1}{3}$ y como P_2 con probabilidad $\frac{2}{3}$. Por otra parte, los procesos $Q_1 = [\frac{1}{3}]a \to P_1 \Box [\frac{2}{3}]b \to P_2$ y $Q_2 = [\frac{1}{2}]a \to P_1 \Box [\frac{1}{2}]b \to P_2$ se considerarán equivalentes.

Se estudian tres semánticas para el nuevo lenguaje: de pruebas, denotacional y operacional, las cuales se prueban equivalentes (es decir, identifican a los mismos procesos). Además, se da una serie de reglas y axiomas, los cuales son correctos con

respecto a las semánticas anteriores, formando un conjunto completo.

De entre los comentados, este último modelo es el que se encuentra más cercano al nuestro. De hecho, como veremos, nuestra semántica axiomática está inspirada en la de Cuartero. La principal diferencia (la cual complica bastante nuestro modelo) viene del hecho de que nosotros consideramos una interpretación generativa de las probabilidades, con lo cual los procesos Q_1 y Q_2 descritos anteriormente pasan a ser distintos en nuestro marco semántico.

1.1.4 Otras álgebras de procesos

Para el caso de ACP [BK84, BW90] sólo conocemos una extensión probabilística. En [BBS92] se definen axiomatizaciones completas para algunos lenguajes basados en ACP donde se introducen probabilidades en los operadores adecuados, considerándose una interpretación generativa de las probabilidades. Con ello se sigue el camino usual para dar semántica a lenguajes basados en ACP, que consiste en hacerlo algebraicamente, es decir, por medio de axiomas.

LOTOS [LOT88] es un álgebra de procesos en la que se combina el operador de elección de CCS con los operadores paralelo y de ocultamiento de CSP. Entre las extensiones probabilísticas de LOTOS podemos citar las siguientes.

En [MFV93] se propone una extensión probabilística que es compatible *hacia arriba* con LOTOS. Se define una semántica operacional y a partir de ella una semántica de pruebas para el lenguaje. Sin embargo, la extensión se limita a la introducción de un operador probabilístico de elección interna, de modo que se combina una elección no probabilística con la elección interna probabilística (es decir, algo similar a lo que ocurría en [YL92]). El trabajo concluye con la presentación de la especificación de un sistema de comunicaciones utilizando el nuevo lenguaje.

En [KLL94] se considera una extensión probabilística de un subconjunto de LOTOS. Se define una semántica del nuevo lenguaje en la que se considera *concurrencia real.* Para hacerlo se utilizan *estructuras de eventos* [Win87, Win89], usando una variante probabilística de la definida en [Lan93]. Pero al igual que en el caso anterior,

la extensión probabilística del lenguaje se sigue reduciendo a la introducción de una elección interna probabilística.

A partir de la semántica de pruebas definida en [dFNQ95], en [NdF95c] los operadores de elección y paralelo de LOTOS son extendidos con una probabilidad. Se da una semántica operacional del nuevo lenguaje, y a partir de ésta se definen interpretaciones reactivas y generativas de las probabilidades. La principal ventaja de estas caracterizaciones, frente a las descritas en [vGSST90], proviene del hecho de que en nuestro caso utilizamos una única semántica operacional, en contraste con lo que se hace en [vGSST90]. Las diferencias entre los modelos provienen de los diferentes conjuntos de pruebas que se consideran en cada caso. Como en otros trabajos, al *olvidar* las probabilidades se pueden recuperar las nociones de *must* y *may*, al menos para una clase adecuada de procesos. Además se define una nueva interpretación de las probabilidades: el modelo *generativo limitado*. En este modelo sólo se admiten las pruebas deterministas tales que las distribuciones de probabilidad son consideradas uniformes sobre los conjuntos de acciones que se ofrecen en cada momento. Finalmente, se presenta un conjunto de *leyes* las cuales son correctas con respecto a los modelos anteriormente definidos.

1.1.5 Modelos para procesos probabilísticos

A partir de la noción de *sistemas de transiciones etiquetadas* [Plo81] han habido varias propuestas de extensiones probabilísticas, las cuales etiquetan con probabilidades las transiciones.

En [Chr90a, Chr90b] se definen procesos probabilísticos, y se dan cuatro órdenes parciales entre procesos probabilísticos.

1. $s \leq_{tr} s'$ significa que interaccionando con cualquier prueba secuencial, s' tiene una probabilidad menor o igual de bloqueo (*deadlock*) que s. Esto corresponde a la noción de modelo reactivo de [vGSST90].

2. $s \leq_{wte} s'$ significa que interaccionando con cualquier prueba, después de ejecutar

cualquier cadena de acciones observables (es decir, sin incluir la acción τ), la probabilidad de bloqueo de s' es menor o igual que la de s, con respecto a las acciones observables que pueden ser ejecutadas en el siguiente paso.

3. $s \leq_{te} s'$ significa que interaccionando con cualquier prueba, la probabilidad de bloqueo de s' es menor o igual que la de s, al ejecutar cualquier cadena de acciones observables finalizada por cualquiera de las acciones que pueden ser ejecutadas en el último paso.

4. $s \leq_{ste} s'$ significa que interaccionando con cualquier prueba, la probabilidad de bloqueo de s' es menor o igual que la de s, al ejecutar cualquier cadena de acciones observables.

De la forma en que están definidos estos órdenes parciales, se verifica que:

$$\text{i)} \quad (s \leq_{ste} s') \Rightarrow (s \leq_{te} s') \Rightarrow (s \leq_{tr} s')$$
$$\text{ii)} \quad (s \leq_{ste} s') \Rightarrow (s \leq_{wte} s') \Rightarrow (s \leq_{tr} s')$$

Sin embargo no existe una relación de inclusión entre $\leq_{te}$ y $\leq_{wte}$.

Se presenta una representación matricial para los procesos probabilísticos y usando resultados de teoría de autómatas con probabilidades [Paz71], se consigue caracterizar las equivalencias entre procesos inducidas por los órdenes parciales $\leq_{tr}$ y $\leq_{ste}$, examinando el comportamiento de los procesos para un número finito de cadenas de elementos observables.

Finalmente se presenta un lenguaje basado en CSP, para el cual se define una semántica operacional en términos de operaciones entre procesos probabilísticos.

En [HT92] se hace un estudio de varias nociones de equivalencia para procesos probabilísticos que resultan ser muy similares a las estudiadas en [JS90] por lo que en esencia se llega a los mismos resultados. En concreto demuestran que las equivalencias para menús, fallos, trazas, trazas maximales y trazas finitas son decidibles en tiempo polinomial para sistemas probabilísticos finitos de transiciones etiquetadas,

extendiendo los resultados de [PT87] al marco probabilístico[3]. Para finalizar, consiguen un algoritmo que, en tiempo polinomial, decide si dos sistemas probabilísticos finitos de transiciones etiquetadas son bisimilares.

En [CSZ92] se presenta una semántica de pruebas para sistemas probabilísticos finitos de transiciones etiquetadas, con una interpretación generativa de las probabilidades. Al igual que en el caso no probabilístico, los procesos y las pruebas son esencialmente iguales, salvo que en las pruebas se incorporan una serie de estados de aceptación. Se muestra que los preórdenes clásicos de pruebas se pueden recuperar, simplemente *olvidando* las probabilidades. Finalmente, se define un operador para componer en paralelo dos de estos sistemas, y se presenta un ejemplo de aplicación de este nuevo operador y del lenguaje definido.

En [YCDS94] se presenta una caracterización alternativa del preorden anterior, mediante la definición de un conjunto de pruebas *esenciales* (llamadas trazas probabilísticas) que tiene el mismo poder de distinción entre procesos que la familia completa de pruebas. Para hacerlo se estudia primero el caso en el que las pruebas no tienen acciones ocultas, y después se pasa al caso general. Además, en el caso general, el preorden resulta ser de hecho una relación de equivalencia, de modo que $p \leq p' \Rightarrow p' \leq p$ ($\Rightarrow p \equiv p'$).

En [Chr93] se definen distintos órdenes parciales para procesos probabilísticos en función del tipo de entorno en el que se encuentran los procesos. Se consideran tres tipos de entornos:

1. *de traza*, cuando el entorno sólo ofrece una acción en cada paso.

2. *de escoba* (*broom*), cuando el entorno ofrece en todos los pasos un conjunto de acciones, salvo en el último paso en el que ofrece una sola acción.

3. *barbado* (*barbed*), cuando el entorno en cualquier paso ofrece un conjunto de acciones.

[3]Estudios sobre la complejidad de la decidibilidad de equivalencias entre procesos no probabilísticos se encuentran por ejemplo en [PT87, KS90, HT91].

A partir de estos tipos de entornos se definen tres órdenes parciales:

1. $p \preceq_{tr} p'$ significa que al interaccionar con un entorno *de traza*, la probabilidad de bloqueo de p' es menor o igual que la de p. Este orden parcial es equivalente al orden parcial $\leq_{tr}$ definido en [Chr90b].

2. $p \preceq_{br} p'$ significa que al interaccionar con un entorno *de escoba*, la probabilidad de bloqueo de p' es menor o igual que la de p.

3. $p \preceq_{ba} p'$ significa que al interaccionar con un entorno *barbado*, la probabilidad de bloqueo de p' es menor o igual que la de p. Este orden parcial es equivalente al orden parcial $\leq_{ste}$ definido en [Chr90b].

Después se da una representación matricial (similar a la dada en [Chr90b]) para los procesos probabilísticos, caracterizándose la equivalencia entre procesos para las tres relaciones inducidas por los órdenes parciales anteriormente expuestos.

Además, se presentan algoritmos para verificar las equivalencias entre procesos generativos (es decir que pueden interaccionar con los tres tipos de entornos anteriormente definidos), que resultan tener complejidad polinómica (mientras que la verificación utilizando matrices es de orden exponencial). Estos algoritmos están inspirados en el trabajo de Tzeng [Tze85, Tze92].

En [CC92] se definen tres lógicas para formalizar las semánticas anteriores. Además, utilizando predicados en estas lógicas expresan propiedades de seguridad (*safety*) y vivacidad (*liveness*).

1.1.6 Autómatas Probabilísticos

Recientemente se han presentado diversos trabajos en los que se extiende el modelo de *Autómatas de Entrada/Salida* [LT87] con probabilidades. Esencialmente, un autómata de entrada/salida es una sistema de transiciones etiquetadas donde las acciones son divididas en dos conjuntos: las controlables localmente (acciones de *entrada*) y las no controlables. Estas últimas se dividen a su vez en dos tipos: las acciones de *salida* y las *internas*. Además, las acciones de entrada siempre tienen que estar *disponibles*; es

decir, cualquier autómata tiene que poder ejecutar una acción de entrada si el entorno se la ofrece.

En [SL94] se estudian varias extensiones de las relaciones de bisimulación y simulación, utilizando como modelo una extensión con probabilidades de los autómatas de entrada/salida con una interpretación reactiva de las probabilidades. Para poder estudiar propiedades sobre el modelo, se hace una adaptación de la lógica TPCTL descrita en [Han91, HJ94] olvidando la información temporal.

En [Seg95a] se define una semántica basada en trazas para el modelo de autómatas probabilísticos definido en [SL94], y se extiende el método de simulación de [LV91] al nuevo modelo.

En [Seg95b] se extiende el modelo de [SL94] con información temporal y se estudian varias semánticas, en particular una similar a la de [Seg95a] pero considerando información temporal.

En [WSS94] se extienden los autómatas de [LT87] con probabilidades, dándose una interpretación generativa de las probabilidades. Cuando restringen los autómatas a aquellos que no tienen acciones ocultas, encuentran una semántica *totalmente abstracta* con respecto a una definición natural de pruebas probabilísticas.

1.1.7 Modelos con prioridades

En esta sección vamos a considerar algunos trabajos que extienden álgebras de procesos con una noción de prioridad. A la hora de construir álgebras de procesos con prioridades, se han considerado fundamentalmente dos aproximaciones: asociar prioridad con eventos o asociarla con operadores de elección.

Asociar prioridad con eventos

Esta prioridad se puede asignar de una forma *global* o *local*.

- *Prioridad global.*

En [BBK86] se define una extensión con prioridades de ACP, en la que se considera un orden parcial entre las acciones. Por lo tanto, se trata de una asignación global de prioridades, dado que si $a < b$ en un estado, no puede ocurrir que $b < a$ en otro estado. Este orden parcial expresa prioridad *potencial*, en el sentido de que en cada estado existen dos posibilidades:

- La relación de prioridad es la expresada por medio del orden parcial sobre el conjunto de acciones, lo cual ocurre si el estado está en el ámbito de un operador de prioridad θ.
- No existe ninguna estructura de prioridad entre las acciones.

En [Jef92] se considera un lenguaje que toma operadores de CCS y CSP, extendiéndolos con tiempo y prioridades, de modo que la asignación de prioridades se realiza mediante un orden total entre las acciones, salvo para las acciones ocultas, a las que se les asigna prioridad localmente. Presenta una axiomatización para una noción de bisimulación fuerte, y muestra como transformar un proceso temporal en uno no temporal mediante el etiquetado de las acciones con tiempo.

- *Prioridad Local.*

  En [CH90] se extiende CCS con una noción de prioridad en la que para cada acción $a \in Act$ se definen una versión con prioridad $\underline{a}$ y una versión sin prioridad a. Además, definen una equivalencia observacional fuerte y dan una axiomatización completa para los términos finitos.

  En [BGLG93] se define un álgebra de procesos con una extensión temporal (sobre un dominio denso) y una noción de prioridad. La asignación de prioridad se hace localmente, asociando en cada momento una prioridad a cada acción.

  En [NCCC94] se estudia un lenguaje similar al de [CH90] sobre el que se define una equivalencia observacional débil y se da una axiomatización completa para los términos finitos.

Asociar prioridad con el operador de elección

En [SS90], se extiende con una probabilidad el operador de elección de CCS. En particular, esta probabilidad puede ser igual a 1, lo cual representa una forma de tratar la prioridad. Para poder tratar estos casos *extremos*, se considera una interpretación estratificada de las probabilidades. Se define una noción de bisimulación probabilística la cual resulta ser una congruencia sobre el cálculo definido.

En [CW95], se extiende CCS con un operador de elección con prioridad. Se define una noción de bisimulación fuerte, que resulta ser una congruencia, y se da un conjunto de axiomas, el cual es completo con respecto a la bisimulación fuerte. Además, los axiomas siguen satisfaciéndose si consideramos los procesos ordinarios de CCS sin este operador de elección con prioridad.

1.2 Resumen de la Tesis

En esta sección haremos un breve resumen del resto de este trabajo. En todo caso, en la introducción de cada uno de los capítulos se explican con mayor profundidad los contenidos y objetivos del correspondiente capítulo.

En el Capítulo 2 presentamos nuestro lenguaje, al que denominamos PPA. Damos una interpretación intuitiva de cada uno de los operadores del lenguaje, y a continuación definimos una semántica operacional. A partir de la semántica operacional definimos una semántica de pruebas en el sentido clásico, y aprovechamos para discutir la utilidad de los *factores de prenormalización* para conseguir una interpretación más intuitiva de las probabilidades con las que los procesos pasan las pruebas. Finalizamos el capítulo con una explicación de una de las decisiones de *diseño* más discutibles de las tomadas a lo largo de esta tesis: la separación entre transiciones internas y externas en la semántica operacional; y en general con una descripción de los problemas que acarrea el hecho de tener acciones ocultas junto con probabilidades.

En el Capítulo 3 estudiamos una interpretación *reactiva* de las probabilidades. Bajo dicha interpretación las pruebas que resultan pertinentes son simplemente las trazas de acciones. Definimos una caracterización alternativa de la semántica de

pruebas utilizando trazas probabilísticas, que no son otra cosa que trazas ordinarias junto con la probabilidad de que el proceso ejecute la misma al ser ofrecida por el entorno. Mostramos que los operadores de elección externa no son congruentes para este modelo, y definimos una semántica denotacional basada también en el concepto de trazas probabilísticas que resulta ser completamente abstracta para el sublenguaje en el que se omite el operador de elección externa.

El Capítulo 4 constituye la parte central de esta tesis. En el mismo estudiamos la interpretación *generativa* de la semántica de pruebas para PPA. En este caso, las pruebas pueden ser tan complejas como los procesos. Como en el caso anterior definimos una caracterización alternativa para la semántica de pruebas. La misma estará basada en una variante de los *conjuntos de aceptación* en los que se incluye la información probabilística adecuada. Así, los estados no seguirán siendo meros conjuntos de acciones, sino conjuntos de pares (acción, probabilidad). Además, en lugar de utilizar secuencias de acciones para caracterizar los estados de un proceso, utilizaremos secuencias de pares (estado, acción). Finalmente, un conjunto de aceptación no será un conjunto de estados, sino un conjunto de estados con una probabilidad asociada. Para probar que efectivamente se trata de una caracterización alternativa, haremos una transformación de los procesos a una clase de *formas normales*, aunque hablando con propiedad no deberíamos utilizar dicha denominación, al tratarse de *procesos* sintácticos que en ocasiones resultan ser infinitos. A partir de la demostración obtenemos además como subproducto dos resultados muy interesantes:

- El conjunto de pruebas precisas para caracterizar la semántica se puede reducir notablemente, limitándonos a considerar las que denominamos *barbas probabilísticas*.

- El hecho de utilizar factores de prenormalización no influirá en la equivalencia inducida por la semántica de pruebas.

A continuación se define una semántica denotacional para PPA. La misma está basada en el concepto de *árboles de aceptación*, efectuando de nuevo las oportunas modificaciones para adecuarlas al marco probabilístico. Estas modificaciones serán

similares a las comentadas anteriormente para el caso de la adaptación de los conjuntos de aceptación. Finalmente, se demuestra que esta semántica es completamente abstracta con respecto a la semántica de pruebas, obteniendo en suma la equivalencia entre las tres semánticas.

En el Capítulo 5 estudiamos un sistema axiomático correcto y completo respecto de la semántica de pruebas para el modelo generativo. Empezamos el estudio con los procesos finitos, esto es, procesos en los que no aparecen ni recursiones ni procesos divergentes. Para dicho subconjunto mostramos un sistema de axiomas correctos. Para realizar la correspondiente prueba de completitud, introducimos *formas normales* para los procesos finitos. Estas formas normales coinciden con las descritas en el Capítulo 4, si nos restringimos a las formas normales finitas. A continuación se estudia el sistema axiomático para el lenguaje completo. Damos primero una serie de axiomas que caracterizan el comportamiento del proceso divergente. Pero ahora hemos de axiomatizar una relación de orden, en lugar de una equivalencia, pues hemos de tratar con aproximaciones finitas para definir la semántica de los procesos recursivos. Entre los axiomas aparece uno, en principio un tanto extraño, que resultará de gran utilidad a la hora de demostrar la completitud del nuevo sistema de axiomas. Como veremos, el citado axioma resulta necesario por estarse trabajando con un dominio que no es ω-algebraico.

En el Capítulo 6 se presentan una serie de ejemplos que muestran la utilidad del nuevo lenguaje para especificar procesos concurrentes donde la información probabilística juega un papel fundamental. En estos ejemplos se utiliza frecuentemente un operador de composición paralela que será definido en el Apéndice A.

En el Capítulo 7 presentamos nuestras conclusiones y damos una serie de líneas para trabajo futuro.

Para finalizar, en el Apéndice A estudiamos la inclusión en nuestro modelo de un operador paralelo y de un operador de restricción (ocultamiento). El hecho de que hayamos relegado a un apéndice el estudio de tales cuestiones proviene del notable esfuerzo, especialmente de orden técnico, que conlleva su estudio. Además, al desgajar el estudio de estas cuestiones del núcleo de la tesis nos concedemos mucha más

libertad para tratarlos de un modo más informal y especulativo. Así, para el caso del operador paralelo consideramos tres posibilidades: sin probabilidades asociadas, con una probabilidad, y con dos probabilidades. Una vez definida la semántica operacional de cada uno de los nuevos operadores, pasamos a hacer un estudio semántico más profundo del operador que incorpora una única probabilidad. Al igual que ocurría con la elección externa, el nuevo operador no es congruente para la semántica de pruebas del modelo reactivo. En el caso del modelo generativo, definimos la semántica denotacional del operador paralelo, y extendemos el conjunto de axiomas definido en el Capítulo 5, incorporando una serie de axiomas para el nuevo operador, mediante las cuales podemos probar que el operador paralelo puede ser considerado como operador derivado (al igual que ocurre en los modelos no probabilísticos) dado que podemos eliminar sus apariciones en *cabeza*. En consecuencia podemos eliminar todas las apariciones en los procesos finitos, mientras que ello sólo sería posible en el caso infinito por medio de expansiones infinitas que nos conducirían a *procesos* sintácticos infinitos.

En el caso del operador de restricción los resultados no son tan halagüeños. Mostramos que la inclusión de un operador tal nos conduciría a una serie de dificultades para las que hemos encontrado posibles *soluciones* que no acaban de satisfacernos totalmente. Una posible solución es la inclusión de prioridades en el modelo, lo cual complica notablemente todo el desarrollo ya realizado. Otra posible solución sería admitir al operador de restricción como operador *primitivo*. Esto nos conduciría a unas formas normales mucho más complejas que las que tenemos, pues no resulta en absoluto claro cual sería la forma adecuada de incorporar la información sobre apariciones del operador de ocultamiento en las nuevas formas normales.

Capítulo 2

PPA: Álgebra de Procesos Probabilística

En este capítulo vamos a presentar nuestra álgebra de procesos probabilística, a la que llamaremos PPA[1]. Este lenguaje está basado en un subconjunto del lenguaje presentado en [Hen88] (este lenguaje apareció por primera vez en [dNH87], siendo su principal característica que se sustituye el operador de elección de CCS por dos operadores de elección, similares a los presentes en CSP), donde los operadores de elección han sido *etiquetados* con una probabilidad. Por lo tanto, tendremos simultáneamente una elección externa probabilística y una elección interna probabilística.

En principio, trabajaremos con un sublenguaje en el que no tenemos ni operadores de composición paralela ni operador de ocultamiento o restricción. En el Apéndice A veremos como se puede añadir un operador de composición paralela sin variar las semánticas definidas para el lenguaje, mientras que el hecho de añadir un operador de ocultamiento o restricción nos llevaría a la necesidad de modificar la semántica para introducir una cierta noción de *prioridad.* En este trabajo los valores *extremos* de las probabilidades (es decir 0 y 1) no serán considerados, dado que nos obligarían a extender los modelos con información sobre prioridades (de forma similar a lo que

[1]Hemos decidido mantener las iniciales del nombre en inglés, *Probabilistic Process Algebra*, que es el que utilizamos en [NdFL95].

nos sucede al tratar un operador de ocultamiento o restricción).

El resto del capítulo está estructurado como sigue. En la Sección 2.1 definimos la sintaxis de nuestra álgebra de procesos y damos una explicación intuitiva de los operadores. En la Sección 2.2 definimos una semántica operacional para PPA. En la Sección 2.3 definimos una semántica de pruebas (parametrizada por un conjunto de pruebas) inducida por la semántica operacional definida en la sección anterior. Finalmente, en la Sección 2.4 comentaremos algunos problemas que aparecen al intentar definir una nueva semántica operacional en la que se combinen transiciones ocultas y observables.

2.1 Sintaxis de PPA

Definición 2.1 Dado un conjunto[2] de acciones Act y un conjunto de identificadores Id, el conjunto de procesos que constituyen PPA se define por medio de la siguiente expresión BNF:

$$P ::= Nil \mid \Omega \mid X \mid a; P \mid P \oplus_p P \mid P +_p P \mid recX.P$$

donde $p \in (0,1)$, $a \in Act$ y $X \in Id$.

Dado un proceso P, definimos inductivamente el *alfabeto de* P, que notaremos por $\alpha(P)$, en la forma:

- $\alpha(Nil) = \alpha(\Omega) = \alpha(X) = \emptyset$.
- $\alpha(a; P') = \{a\} \cup \alpha(P')$.
- $\alpha(P_1 +_p P_2) = \alpha(P_1 \oplus_p P_2) = \alpha(P_1) \cup \alpha(P_2)$.

[2] En [NdFL95] pedíamos que el conjunto de acciones fuera finito, dado que en las demostraciones de ciertos teoremas (p.e. de abstracción completa) considerábamos ciertas pruebas que tenían que ofrecer *todas* las acciones que pertenezcan al conjunto de acciones. Sin embargo, esta restricción no es de hecho necesaria, pues en cada caso nos podemos restringir a las acciones que aparezcan en los procesos, con lo que trabajaríamos siempre con conjuntos finitos de acciones, sin mas que restringirnos a las acciones que aparezcan en dichos conjuntos finitos.

- $\alpha(recX.P') = \alpha(P')$.

Nótese que para todo proceso $P \in$ PPA, su alfabeto es finito. □

De ahora en adelante nos restringiremos a procesos de PPA en los cuales no haya apariciones libres de identificadores (es decir, si un identificador X aparece en un proceso P, deberá estar en el ámbito de una declaración del tipo $recX.P'$ donde P es un subproceso de P'). Además, usualmente omitiremos las apariciones del proceso *Nil* cuando su aparición se pueda deducir del entorno (así expresaremos el proceso $a; b; Nil$ simplemente como $a; b$). A continuación damos una explicación intuitiva de cada uno de los operadores:

- *Nil* es un proceso que está bloqueado, es decir no puede ejecutar ninguna acción.
- Ω es un proceso *divergente*, es decir un proceso que realiza continuamente computaciones internas.
- $a; P$ es un proceso que primero ejecuta la acción a, y después pasa a comportarse como P.
- $P \oplus_p Q$ es un proceso que se comporta como P con probabilidad igual a p, y como Q con probabilidad igual a $1-p$. Esta elección se realiza de forma interna y de forma no determinista, es decir sin interacción alguna con el entorno.
- $P +_p Q$ es un proceso que puede funcionar como P o como Q, pero esta elección depende del entorno. Ello significa que si el entorno ofrece acciones que sólamente uno de los dos procesos puede ejecutar, entonces el correspondiente proceso será elegido. Por el contrario, si ambos procesos pueden ejecutar acciones de las ofrecidas por el entorno, entonces la elección se efectuará teniendo en cuenta las probabilidades con las cuales ambos procesos pueden ejecutar las acciones ofrecidas, las probabilidades con las que el entorno ofrece las acciones, y la probabilidad p que aparece en la elección externa.

  Por tanto, como en el caso no probabilístico, la elección externa tiene también un grado de *no determinismo*, el cual, excluyendo el debido a elecciones internas

presentes en los procesos componentes de la elección externa, se deberá a uno de los dos factores siguientes:

1. Los procesos componentes tienen acciones comunes de entre las que pueden ejecutar en su primer paso.
2. El entorno ofrece distintas acciones que pueden ser ejecutadas unas u otras por los dos procesos en su primer paso, si bien cada una de dichas acciones no tiene porqué ser necesariamente ejecutable por los dos procesos.

Obviamente, ambos factores de no determinismo pueden aparecer simultáneamente. En los ejemplos 2.2, 2.3 y 2.4 se ilustran las distintas situaciones posibles.

- $recX.P$ es utilizado para definir procesos recursivos.

Ejemplo 2.2 Consideremos el proceso $P_1 = a; P_1' +_{\frac{1}{3}} a; P_1''$. Si el entorno ofrece una a, posiblemente junto con más acciones, el proceso P_1 ejecutará a (con probabilidad 1) y pasará a comportarse como P_1' con probabilidad igual a $\frac{1}{3}$ y como P_1'' con probabilidad igual a $\frac{2}{3}$. Este es por tanto un ejemplo de no determinismo del primer tipo descrito. □

Ejemplo 2.3 Consideremos el proceso $P_2 = a; P_2' +_{\frac{1}{4}} b; P_2''$. Si el entorno ofrece la acción a con probabilidad $\frac{1}{6}$, la acción b con probabilidad $\frac{1}{3}$, y otras acciones con probabilidad total $\frac{1}{2}$, entonces P_2 ejecutaría la acción a con probabilidad proporcional a $\frac{1}{4}\cdot\frac{1}{6}$, pasando a comportarse como P_2', y ejecutaría b con probabilidad proporcional a $\frac{3}{4}\cdot\frac{1}{3}$, pasando a comportarse como P_2''. Pero la suma de dichas probabilidades es menor que uno; en concreto vale $\frac{1}{4}\cdot\frac{1}{6}+\frac{3}{4}\cdot\frac{1}{3}=\frac{7}{24}$, por lo que habrá que *normalizarlas* (este concepto quedará más claro cuando definamos la composición entre un proceso y una prueba). La forma de hacerlo consiste en dividir cada una de dichas probabilidades por su suma. Tras ello, P_2 ejecutará la acción a con probabilidad $\frac{\frac{1}{4}\cdot\frac{1}{6}}{\frac{1}{4}\cdot\frac{1}{6}+\frac{3}{4}\cdot\frac{1}{3}}=\frac{1}{7}$, mientras que ejecutará la acción b con probabilidad $\frac{\frac{3}{4}\cdot\frac{1}{3}}{\frac{1}{4}\cdot\frac{1}{6}+\frac{3}{4}\cdot\frac{1}{3}}=\frac{6}{7}$. Este es un ejemplo del segundo tipo de no determinismo. □

Ejemplo 2.4 Consideremos el proceso $P = (a; P_1 +_{\frac{1}{4}} b; P_2) +_{\frac{1}{3}} (a; P_3 +_{\frac{1}{2}} b; P_4)$. Supongamos que el entorno ofrece la acción a con probabilidad $\frac{1}{5}$, la acción b con probabilidad $\frac{2}{5}$ y otras acciones con probabilidad total $\frac{2}{5}$. Entonces, la probabilidad *total* (antes de normalizar) con la que P ejecutaría acciones ante dicha oferta es igual a $\frac{1}{3} \cdot (\frac{1}{4} \cdot \frac{1}{5} + \frac{3}{4} \cdot \frac{2}{5}) + \frac{2}{3} \cdot (\frac{1}{2} \cdot \frac{1}{5} + \frac{1}{2} \cdot \frac{2}{5}) = \frac{19}{60}$.

Tras normalizar, el proceso P ejecutará la acción a con una probabilidad igual a $\frac{\frac{1}{3}\cdot\frac{1}{4}\cdot\frac{1}{5}+\frac{2}{3}\cdot\frac{1}{2}\cdot\frac{1}{5}}{\frac{19}{60}} = \frac{\frac{5}{60}}{\frac{19}{60}} = \frac{5}{19}$, y la acción b con probabilidad $\frac{\frac{1}{3}\cdot\frac{3}{4}\cdot\frac{2}{5}+\frac{2}{3}\cdot\frac{1}{2}\cdot\frac{2}{5}}{\frac{19}{60}} = \frac{\frac{14}{60}}{\frac{19}{60}} = \frac{14}{19}$.

Las distintas *continuaciones* del proceso P tras ejecutar una acción son:

- El proceso P ejecutará la acción a y pasará a comportarse como P_1 con probabilidad $\frac{\frac{1}{3}\cdot\frac{1}{4}\cdot\frac{1}{5}}{\frac{19}{60}} = \frac{1}{19}$.
- El proceso P ejecutará la acción a y pasará a comportarse como P_3 con probabilidad $\frac{\frac{2}{3}\cdot\frac{1}{2}\cdot\frac{1}{5}}{\frac{19}{60}} = \frac{4}{19}$.
- El proceso P ejecutará la acción b y pasará a comportarse como P_2 con probabilidad $\frac{\frac{1}{3}\cdot\frac{3}{4}\cdot\frac{2}{5}}{\frac{19}{60}} = \frac{6}{19}$.
- El proceso P ejecutará la acción b y pasará a comportarse como P_4 con probabilidad $\frac{\frac{2}{3}\cdot\frac{1}{2}\cdot\frac{2}{5}}{\frac{19}{60}} = \frac{8}{19}$.

En este ejemplo se combinan los dos tipos de no determinismo *externo* que se pueden producir en la elección externa. □

Para finalizar esta sección, generalizaremos los operadores de elección de forma que puedan tener un número arbitrario de argumentos (en lugar de dos). Estos operadores generalizados serán utilizados frecuentemente a lo largo de esta tesis (e.g. para definir formas normales, para definir pruebas generalizadas, etc.). Para el caso de la elección externa, exigiremos que los procesos estén *guardados* mediante prefijo y que no se produzca no determinismo por causa de dos prefijos iguales.

Definición 2.5 Sean $P_1, P_2, \ldots, P_n$ procesos de PPA, sean $a_1, a_2, \ldots, a_n \in Act$ acciones diferentes, y sean $p_1, p_2, \ldots, p_n > 0$ tales que $\sum p_i = 1$. Definimos inductivamente la *elección externa generalizada* en la forma:

1. $\sum_{i=1}^{1}[1]\, a_1; P_1 = a_1; P_1$

2. $\sum_{i=1}^{n}[p_i]\, a_i; P_i = (a_1; P_1) +_{p_1} \sum_{i=1}^{n-1}[\frac{p_{i+1}}{1-p_1}]\, (a_{i+1}; P_{i+1})$

□

Definición 2.6 Sean $P_1, P_2, \ldots, P_n$ procesos de PPA, y sean $p_1, p_2, \ldots, p_n > 0$ tales que $\sum p_i \leq 1$. Definimos inductivamente la *elección interna generalizada* en la forma:

1. $\bigoplus_{i=1}^{0}[p_i]\, P_i = \Omega$

2. $\bigoplus_{i=1}^{1}[1]\, P_1 = P_1$

3. $\bigoplus_{i=1}^{n}[p_i]\, P_i = P_1 \oplus_{p_1} \bigoplus_{i=1}^{n-1}[\frac{p_{i+1}}{1-p_1}]\, P_{i+1}$ [si $\sum p_i = 1 \ \wedge \ n > 1$]

4. $\bigoplus_{i=1}^{n}[p_i]\, P_i = \bigoplus_{i=1}^{n}[\frac{p_i}{p}]\, P_i \oplus_p \Omega$ [siendo $p = \sum p_i < 1 \ \wedge \ n > 0$]

□

Nótese que en la elección interna generalizada la suma de las probabilidades puede ser menor que 1. La diferencia entre 1 y dicha suma indica la probabilidad de *divergencia*[3]. Por ejemplo, $P = \bigoplus_{i=1}^{2}([\frac{1}{3}]\, P_1)\ ([\frac{1}{3}]\, P_2) = (P_1 \oplus_{\frac{1}{2}} P_2) \oplus_{\frac{2}{3}} \Omega$, es decir la probabilidad de que el proceso P diverja en su primer paso es igual a $\frac{1}{3}$ (junto con las probabilidades aportadas por P_1 y P_2).

2.2 Semántica Operacional de PPA

En esta sección introduciremos una semántica operacional para nuestra álgebra de procesos. En dicha semántica operacional tendremos dos tipos de transiciones:

[3]Naturalmente, a dicha probabilidad habría que añadir la que aportasen cada uno de los argumentos de la suma, multiplicadas por la probabilidad de cada argumento.

- El significado intuitivo de una transición del tipo $P \xrightarrow{a}_p Q$ es que si el entorno ofreciera *todas* las acciones que constituyen el conjunto de acciones Act, con una distribución equiprobable, entonces la probabilidad de que el proceso P ejecute la acción a y pase a comportarse como el proceso Q es igual a p.

- El significado intuitivo de una transición del tipo $P \rightarrowtail_p Q$ es que el proceso P pasa a comportarse como el proceso Q con probabilidad igual a p, sin interaccionar con el entorno (es decir, se trata de una transición interna).

En nuestro modelo, las transiciones internas se efectúan *lo antes posible*, de modo que un proceso no puede realizar transiciones observables hasta que ha realizado todas sus transiciones internas pendientes. Esto quedaría plenamente justificado si introducimos una componente *temporal* en nuestro lenguaje. En la mayoría de las propuestas para extender con tiempo las álgebras de procesos, las acciones ocultas se consideran *urgentes*; es decir, un proceso no puede dejar pasar el tiempo mientras pueda *evolucionar* mediante una transición interna. Un ejemplo de esto se puede ver en [NJ91, Sch95, LdFN96].

Para evitar una notación más complicada, hemos evitado la introducción de índices en la definición de las transiciones. Usualmente (a partir de [GJS90]), tales índices se utilizan para distinguir entre diferentes apariciones de una misma transición probabilística. En lugar de utilizar dicha solución, nosotros consideraremos que si una transición se pueda derivar más de una vez, entonces cada derivación genera una ocurrencia diferente de esta transición, lo cual puede ser formalizado usando multiconjuntos de transiciones en lugar de utilizar conjuntos de transiciones. Otra forma de evitar estos problemas consiste en aumentar el número de reglas, como se hace por ejemplo en [LS92].

Ejemplo 2.7 Sea $P = a; Nil +_{\frac{1}{2}} a; Nil$. Entonces, P tendrá la transición $P \xrightarrow{a}_{\frac{1}{2}} Nil$ dos veces. Consideremos ahora el proceso $Q = Q' \oplus_{\frac{1}{2}} Q'$. De nuevo, la transición $Q \rightarrowtail_{\frac{1}{2}} Q'$ aparece dos veces. □

En el resto de este trabajo utilizaremos la siguiente notación:

- $P \rightarrowtail \stackrel{\text{not.}}{=\!=} \exists p, P' : P \rightarrowtail_p P'$
- $P \xrightarrow{a} \stackrel{\text{not.}}{=\!=} \exists p, P' : P \xrightarrow{a}_p P'$
- $P \longrightarrow \stackrel{\text{not.}}{=\!=} \exists a, p, P' : P \xrightarrow{a}_p P'$
- $P \not\rightarrowtail \stackrel{\text{not.}}{=\!=} \nexists p, P' : P \rightarrowtail_p P'$
- $P \not\xrightarrow{a} \stackrel{\text{not.}}{=\!=} \nexists p, P' : P \xrightarrow{a}_p P'$
- $P \not\longrightarrow \stackrel{\text{not.}}{=\!=} \nexists a, p, P' : P \xrightarrow{a}_p P'$

Al igual que en el caso no probabilístico, podemos extender la relación inducida por las transiciones internas, $\rightarrowtail$, considerando una relación $P \rightarrowtail^*_p P'$ que indica que el proceso P ha evolucionado al proceso P' tras ejecutar una secuencia de transiciones internas, tal que el producto de las probabilidades asociadas a las mismas es igual a p. En nuestro caso pediremos además que el proceso P' no pueda ejecutar transiciones internas.

Definición 2.8 Sean P, P' dos procesos. Diremos que P evoluciona a P' por medio de una *transición interna generalizada* con probabilidad p si $P \rightarrowtail^*_p P'$ se puede derivar aplicando las siguientes reglas:

$$P \rightarrowtail^*_1 P \text{ si } P \text{ es estable} \quad (\text{i.e. } P \not\rightarrowtail)$$
$$P \rightarrowtail^*_p P' \text{ si } \exists q, q', Q : P \rightarrowtail_q Q \rightarrowtail^*_{q'} P' \wedge p = q \cdot q'$$

□

Al igual que en el caso de las transiciones de la forma $\rightarrowtail$ tendremos que considerar las posibles repeticiones que se produzcan entre transiciones; así la nueva relación induce un multiconjunto de transiciones en lugar de un conjunto. Por ejemplo, si tenemos dos veces la transición $P \rightarrowtail_p Q$, y podemos generar la transición generalizada $Q \rightarrowtail^*_p P'$ tres veces, tendremos la transición generalizada $P \rightarrowtail^*_{p \cdot q} P'$ seis veces.

Tras esta introducción nos encontramos ya en condiciones de enunciar las reglas que permiten definir las transiciones de cada uno de los operadores.

2.2.1 *Nil*, Ω y Prefijo

El proceso *Nil* es un proceso *inactivo*, es decir un proceso que no puede producir ninguna transición. El proceso Ω es un proceso que sólo tiene actividad *interna*, por

lo tanto para él sólo tendremos una regla que nos indica que, con probabilidad 1, Ω evoluciona de forma interna a sí mismo:

$$(DIV)\frac{}{\Omega \rightarrowtail_1 \Omega}$$

Para el caso del operador de prefijo tendremos también una única regla, indicando que un proceso de la forma $a;P$ ejecuta la acción a, con probabilidad 1, y pasa a comportarse como P; es decir:

$$(PRE)\frac{}{a;P \xrightarrow{a}_1 P}$$

2.2.2 Elección Interna

Para la elección interna tendremos dos reglas que indican que un proceso de la forma $P \oplus_p Q$ pasa a comportarse de forma interna (y por lo tanto no determinista) como P con probabilidad p y como Q con probabilidad $1-p$.

$$(INT1)\frac{}{P \oplus_p Q \rightarrowtail_p P} \qquad (INT2)\frac{}{P \oplus_p Q \rightarrowtail_{1-p} Q}$$

2.2.3 Elección Externa

Para el caso de la elección externa tendremos dos grupos de reglas. En el primer grupo trataremos el caso en el que alguno de los procesos que componen la elección externa puede ejecutar transiciones internas:

$$(EXT1)\frac{P \rightarrowtail_q P' \wedge Q_{\oplus} = 0}{P +_p Q \rightarrowtail_q P' +_p Q} \qquad (EXT2)\frac{Q \rightarrowtail_q Q' \wedge P_{\oplus} = 0}{P +_p Q \rightarrowtail_q P +_p Q'}$$

$$(EXT3)\frac{P \rightarrowtail_{q_1} P' \wedge Q \rightarrowtail_{q_2} Q'}{P +_p Q \rightarrowtail_{q_1 \cdot q_2} P' +_p Q'}$$

Las dos primeras reglas indican que si sólo uno de los procesos puede evolucionar mediante una transición interna, entonces se puede realizar tal transición, pero sin

por ello resolver la elección interna. La condición $Q_{\oplus} = 0$ (resp. $P_{\oplus} = 0$) que aparece en la primera regla es la que indica que la probabilidad de que el proceso Q (resp. P) realice una transición interna es igual a 0. Formalmente, dicha función se define en la forma:

$$R_{\oplus} = \sum_{R'} \{\!\{ s \mid R \stackrel{s}{\longrightarrow} R' \}\!\}$$

La tercera regla indica que si los dos procesos pueden efectuar transiciones internas, éstas se ejecutarán al mismo tiempo, sin que de nuevo se resuelva la elección. De nuevo, esta decisión es bastante natural, dado que todo el no determinismo interno se tiene que resolver antes de que el proceso pueda empezar a realizar transiciones observables; por lo tanto asumiendo una cierta hipótesis de *máximo progreso* podemos considerar que estas transiciones se realizan a la vez. De todas formas, las tres reglas dadas se pueden convertir en sólo dos, suprimiendo la última, si bien habrá que tener en cuenta en ellas si los dos procesos pueden realizar transiciones internas o no. En tal caso, conservamos básicamente el mismo significado intuitivo de los procesos, pero optamos por una semántica de *entrelazamiento*, de modo que si los dos procesos compuestos en elección externa pueden realizar transiciones internas al mismo tiempo, éstas serán realizadas en alternancia. Las reglas correspondientes a dicha variante serían las siguientes:

$$(EXT1')\frac{P \stackrel{q}{\longrightarrow} P'}{P +_p Q \stackrel{\frac{p \cdot q}{f(P,Q)}}{\longrightarrow} P' +_p Q} \qquad (EXT2')\frac{Q \stackrel{q}{\longrightarrow} Q'}{P +_p Q \stackrel{\frac{(1-p) \cdot q}{f(P,Q)}}{\longrightarrow} P +_p Q'}$$

donde la función $f(P,Q)$ viene definida en la forma:

$$f(P,Q) = \begin{cases} p & \text{si } Q \not\stackrel{\tau}{\longrightarrow} \\ 1-p & \text{si } P \not\stackrel{\tau}{\longrightarrow} \\ 1 & \text{e.o.c} \end{cases}$$

Finalmente, nótese que a pesar de las reglas anteriores $+_p$ no es un operador *estático*. Dichas reglas indican que las *acciones internas* (i.e. las transiciones internas) no *resuelven* las elecciones externas, pero sí lo harán las acciones externas gobernadas por el segundo grupo de reglas para la elección externa. Como ya hemos dicho al

principio de esta sección (ver página 31), las transiciones internas se han de realizar antes que las externas; por lo tanto estas reglas sólo serán aplicables cuando los dos procesos estén *estables*, es decir, no puedan realizar de momento más transiciones internas. Ello se indica por medio de las condiciones incluidas en las reglas.

$$(EXT4)\frac{P \xrightarrow{a}_q P' \wedge Q_{\oplus} = 0}{P +_p Q \xrightarrow{a}_{p \cdot \hat{q}} P'} \qquad (EXT5)\frac{Q \xrightarrow{a}_q Q' \wedge P_{\oplus} = 0}{P +_p Q \xrightarrow{a}_{(1-p) \cdot \hat{q}} Q'}$$

El valor $\hat{q}$ se obtiene a partir de la probabilidad q con la cual la transición observable es realizada, y teniendo en cuenta si ambos procesos pueden realizar transiciones externas. Si uno de los procesos no puede realizar acciones externas, la probabilidad *total* va a parar al otro proceso; por contra, si los dos pueden realizar transiciones externas, entonces se tiene en cuenta la probabilidad p de la elección externa. En concreto, la probabilidad de que un proceso pueda realizar transiciones observables viene dada por la función

$$R_+ = \sum_{R'} \{\!|\ s \mid \exists a : R \xrightarrow{a}_s R' \ |\!\}$$

a partir de la cual el valor de $\hat{q}$ queda definido como

$$\hat{q} = \frac{q}{p \cdot P_+ + (1-p) \cdot Q_+}$$

2.2.4 Recursión

Para el operador de recursión tenemos una única regla que indica que la recursión se *despliega*:

$$(REC)\frac{}{recX.P \rightarrowtail_1 P\{recX.P/X\}}$$

Con esta definición de semántica operacional tenemos que las transiciones internas y observables no se entremezclan; es decir, si un proceso puede realizar transiciones internas, entonces no puede realizar transiciones observables y viceversa. Esto queda expresado en el siguiente lema, cuya demostración por inducción estructural resulta trivial.

$$(PRE)\frac{}{a;P \xrightarrow{a}_1 P} \qquad (INT1)\frac{}{P \oplus_p Q \rightarrowtail_p P} \qquad (INT2)\frac{}{P \oplus_p Q \rightarrowtail_{1-p} Q}$$

$$(EXT1)\frac{P \rightarrowtail_q P' \wedge\ Q_\oplus = 0}{P +_p Q \rightarrowtail_q P' +_p Q} \qquad (EXT2)\frac{Q \rightarrowtail_q Q' \wedge\ P_\oplus = 0}{P +_p Q \rightarrowtail_q P +_p Q'} \qquad (EXT3)\frac{P \rightarrowtail_{q_1} P' \wedge\ Q \rightarrowtail_{q_2} Q'}{P +_p Q \rightarrowtail_{q_1 \cdot q_2} P' +_p Q'}$$

$$(EXT4)\frac{P \xrightarrow{a}_q P' \wedge\ Q_\oplus = 0}{P +_p Q \xrightarrow{a}_{p \cdot \hat{q}} P'} \qquad (EXT5)\frac{Q \xrightarrow{a}_q Q' \wedge\ P_\oplus = 0}{P +_p Q \xrightarrow{a}_{(1-p)\cdot \hat{q}} Q'}$$

$$(REC)\frac{}{rec\ X.P \rightarrowtail_1 P\{rec\ X.P/X\}} \qquad (DIV)\frac{}{\Omega \rightarrowtail_1 \Omega}$$

$R_\oplus = \sum_{R'} \{\!|\ s \mid R \rightarrowtail_s R'\ |\!\}$, $R_+ = \sum_{R'} \{\!|\ s \mid \exists a : R \xrightarrow{a}_s R'\ |\!\}$, $\hat{q} = \frac{q}{p \cdot P_+ + (1-p)\cdot Q_+}$

Figura 2.1: Semántica Operacional de PPA.

Lema 2.9 Sea P un proceso. Entonces, si $P \rightarrowtail$ tenemos $P \not\longrightarrow$. □

Nótese que el otro sentido de la implicación no es cierto, dado que, por ejemplo, *Nil* no puede realizar ningún tipo de transiciones, ni internas, ni externas. Nótese también que la condición del lema anterior es equivalente a que si $P \longrightarrow$ entonces $P \not\rightarrowtail$.

La demostración del siguiente corolario es trivial a partir del lema anterior y de la definición de la semántica operacional.

Corolario 2.1 Sea P un proceso. Entonces se tiene

$$(P_\oplus = 0 \vee P_\oplus = 1) \wedge (P_+ = 0 \vee P_+ = 1) \wedge (P_\oplus \neq 1 \vee P_+ \neq 1)$$

□

Finalmente, en la Figura 2.1 mostramos agrupadas todas las reglas que definen la semántica operacional de PPA.

2.3 Semántica de Pruebas

Una vez que hemos definido la semántica operacional de nuestro lenguaje, vamos a pasar a definir una semántica de pruebas inducida por la misma. Para definir una

semántica de pruebas, lo primero que debemos definir es la noción de *prueba*. Siguiendo la teoría clásica de semántica de pruebas [dNH84, Hen88] consideraremos que las pruebas serán procesos construídos con la sintaxis dada en la Definición 2.1, pero extendiendo el conjunto de acciones con una nueva acción $\omega \notin Act$, que indica la terminación con éxito de la prueba. Este es un punto que representa un cambio drástico con respecto al modelo más cercano al nuestro, que es el presentado en [Cua93]. En el mismo las pruebas no pueden ser procesos cualesquiera, sino que tienen que cumplir unas propiedades bastante restrictivas: ausencia de no determinismo, tanto interno como externo, y continuación de la prueba sólo tras una de las acciones ofrecidas en cada paso. Además la semántica de pruebas no se puede definir a partir de la composición paralela de un proceso y una prueba, sino que tiene que definirse mediante un procedimiento *ad-hoc* que precisa de inducción estructural sobre la sintaxis de los procesos.

Definición 2.10 Dado un conjunto de acciones Act y un conjunto de identificadores Id, el conjunto de las pruebas, que notaremos por $\mathcal{T}est$, se define mediante la siguiente expresión BNF:

$$T ::= Nil \mid \Omega \mid X \mid a;T \mid T \oplus_p T \mid T +_p T \mid recX.T$$

donde $p \in (0,1)$, $a \in Act \cup \omega$, y $X \in Id$. □

La semántica operacional de las pruebas es la misma que la definida en la sección anterior para los procesos, considerando ω como una acción más.

A continuación debemos definir cómo interaccionan una prueba y un proceso. Como en el caso no probabilístico, esta interacción será modelada mediante la composición en paralelo del proceso y la prueba, considerando como conjunto de sincronización el conjunto de acciones Act (obsérvese que la acción ω no pertenece al conjunto de sincronización) y ocultando[4] las acciones del conjunto Act.

[4]Realmente esto no es fundamental, simplemente lo hacemos para mantener una notación uniforme entre las transiciones de la composición paralela que se producen por causa de una transición externa y las que se producen por causa de una transición interna, dado que lo único que nos interesa es la probabilidad con la cual se realizan las transiciones.

Como en el caso de la elección externa tendremos dos grupos de reglas, uno que trata las transiciones internas, tanto las del proceso como las de la prueba, y otro para las acciones observables.

$$(PAR1)\ \frac{P >\!\!\longrightarrow_p P' \wedge T_\oplus = 0}{P \mid T \longmapsto_p P' \mid T} \qquad (PAR2)\ \frac{T >\!\!\longrightarrow_p T' \wedge P_\oplus = 0}{P \mid T \longmapsto_p P \mid T'}$$

$$(PAR3)\ \frac{P >\!\!\longrightarrow_p P' \wedge T >\!\!\longrightarrow_q T'}{P \mid T \longmapsto_{p \cdot q} P' \mid T'}$$

El significado de estas reglas es similar al de las tres primeras reglas de la elección externa que ya explicamos en su momento: si el proceso y/o la prueba pueden realizar transiciones internas, éstas se realizan antes de cualquier transición externa. Además, como también indicamos para el caso de la elección externa, estas tres reglas se podrían reducir a dos si queremos evitar el hecho de que los dos procesos realicen sus transiciones internas al mismo tiempo (ver página 34). De todas formas, el hecho de que las transiciones internas se realicen al mismo tiempo no implica que el cálculo definido sea *síncrono*.

Nos queda por presentar una regla que considere el caso de la sincronización entre el proceso y la prueba, y otra para la ejecución de la acción de terminación con éxito, ω, por parte de la prueba:

$$(PAR4)\ \frac{P \xrightarrow{a}_p P' \wedge T \xrightarrow{a}_q T'}{P \mid T \longmapsto_{r_1} P' \mid T'} \qquad (PAR5)\ \frac{T \xrightarrow{\omega}_p T' \wedge P_\oplus = 0}{P \mid T \xmapsto{\omega}_{r_2} Nil}$$

Cuando ni el proceso ni la prueba pueden realizar transiciones internas, se podrán aplicar estas reglas. Nótese que una vez que la composición entre proceso y prueba realiza una transición etiquetada con la acción especial ω, esta composición no puede realizar más transiciones (lo cual tiene sentido, dado que la prueba ya ha tenido *éxito*). Nótese también que para aplicar la regla tenemos la condición $P_\oplus = 0$, de modo

que si nos encontramos un proceso divergente (es decir, que sea operacionalmente equivalente a Ω), ninguna prueba será pasada, dado que la composición en paralelo entre el proceso y la prueba generará una secuencia infinita de transiciones internas asociadas a la divergencia del proceso, que tendrá probabilidad 1 de realizarse.

Falta por indicar como se definen los valores de las probabilidades r_1 y r_2 asociadas a estas transiciones. Tendremos

$$r_1 = \frac{p \cdot q}{\mu(P,T)} \qquad r_2 = \frac{p}{\mu(P,T)}$$

donde $\mu(P,T)$ es un *factor de normalización*, similar al utilizado en [CSZ92] (considerando que dicho factor se utiliza solamente cuando sólo se pueden ejecutar acciones *visibles*), que nos indica la suma de las probabilidades de las acciones que el par proceso-prueba puede ejecutar en su primer paso cuando tanto el proceso como la prueba están estables. Al dividir por dicho factor, la suma de las probabilidades de las transiciones generadas por transiciones observables que la composición entre el proceso y la prueba pueden realizar se hace igual a 1. La definición formal del factor de normalización es la siguiente:

$$\begin{aligned}\mu(P,T) \;&= \textstyle\sum_a \{\!|\, p \cdot q \mid \exists P', T' :\; P \xrightarrow{a}_p P' \wedge T \xrightarrow{a}_q T' \,|\!\} \\ &+ \textstyle\sum \{\!|\, p \mid \exists T' :\; T \xrightarrow{\omega}_p T' \,|\!\}\end{aligned}$$

En realidad la forma en la que se ha definido el factor de normalización, y en consecuencia los valores de r_1 y r_2, no resulta ser canónica. Otras definiciones alternativas serían por tanto justificables. De hecho somos conscientes de que la opción que hemos tomado no es la más intuitiva, pero hemos preferido optar en esta Tesis, como ya se hizo, conscientemente o no en [CSZ92], por una definición relativamente sencilla que no complique exageradamente los cómputos precisos para manejarla. En concreto, bajo nuestra definición la acción ω podría *robar* probabilidad a las acciones observables ejecutables por el par proceso-prueba, en el caso de que éstas sean ofrecidas por el proceso junto con otras acciones no ofertadas por la prueba.

Para resolver este problema, en lugar de estos valores, podríamos haber introducido unos factores de prenormalización como los que ya utilizamos en [NdF95b], de modo que obtuviéramos

$$r_1 = \frac{f_1^{P,T}(p) \cdot f_2^{P,T}(q)}{\mu'(P,T)} \qquad r_2 = \frac{f_2^{P,T}(p)}{\mu'(P,T)}$$

donde $f_1^{P,T}(p)$ viene dado por el cociente entre p (la probabilidad asociada a la transición observable del proceso) y el *factor de prenormalización* correspondiente al proceso P al ser puesto en paralelo con T.

$$f_1^{P,T}(p) = \frac{p}{\sum_a \{\!|\, r \mid \exists P', T', p' :\ P \xrightarrow{a}_r P' \wedge T \xrightarrow{a}_{p'} T' \,|\!\}}$$

De igual forma, $f_2^{P,T}(q)$ es igual al cociente entre q (la probabilidad asociada a la transición observable de la prueba) y el *factor de prenormalización* correspondiente a la prueba T al actuar sobre P.

$$f_2^{P,T}(q) = \frac{q}{\sum_a \{\!|\, r \mid \exists P', T', p' :\ T \xrightarrow{a}_r T' \wedge P \xrightarrow{a}_{p'} P' \,|\!\} + \sum \{\!|\, r \mid \exists T' :\ T \xrightarrow{\omega}_r T' \,|\!\}}$$

En definitiva, los factores de prenormalización sirven para recalcular la probabilidad de las acciones que *realmente* tanto el proceso como la prueba podrán ejecutar al componer ambos en paralelo.

Una vez definidos los factores de prenormalización, deberíamos definir un nuevo *factor de normalización* $\mu'(P,T)$, que sigue las pautas del definido anteriormente, si bien en este caso hemos de considerar $f_1^{P,T}(p)$ y $f_2^{P,T}(q)$ en lugar de p y q. Por lo tanto:

$$\begin{aligned} \mu'(P,T) \ &= \sum_a \{\!|\, f_1^{P,T}(p) \cdot f_2^{P,T}(q) \mid \exists P', T' :\ P \xrightarrow{a}_p P' \wedge T \xrightarrow{a}_q T' \,|\!\} \\ &\quad + \sum \{\!|\, f_2^{P,T}(p) \mid \exists T' :\ T \xrightarrow{\omega}_p T' \,|\!\} \end{aligned}$$

Veamos a continuación un ejemplo que ilustra las diferencias entre ambas aproximaciones al problema de la normalización.

Ejemplo 2.11 Sean $P = a +_{\frac{1}{5}} b$, $T = a +_{\frac{1}{2}} \omega$, y consideremos $P \mid T$. La composición tiene dos transiciones: $P \mid T \longmapsto_{x_1} Nil \mid Nil$ y $P \mid T \stackrel{\omega}{\longmapsto}_{x_2} Nil$ que obviamente no pueden ser ampliadas. Si utilizaramos los factores de prenormalización para calcular las probabilidades de ambas transiciones, obtendríamos $x_1 = x_2 = \frac{1}{2}$, lo cual es bastante razonable, dado que en el entorno en el que se encuentra el proceso P, la acción b no es ofrecida; en consecuencia la probabilidad con la que el proceso P ofrece de hecho la acción a debería devenir en 1, lo cual se consigue al utilizar los factores de prenormalización. Sin embargo, si no utilizamos factores de prenormalización (o equivalentemente $f_1^{P,T}$ y $f_2^{P,T}$ no fueran otra cosa que la función identidad), como ocurre en este trabajo o en [CSZ92], obtenemos $x_1 = \frac{1}{5}$ mientras que $x_2 = \frac{4}{5}$. Es decir, la presencia de la acción b le ha *quitado* probabilidad a la transición que se realiza por la sincronización en la acción a, yendo a parar a la transición que ejecuta la acción ω. □

En cualquier caso, a pesar de que utilizar los factores de prenormalización resulta más intuitivo, las relaciones de equivalencia inducidas por la semántica de pruebas no cambian, se utilicen dichos factores o no, como se verá en el Capítulo 4, lo que justifica formalmente el obviarlos, si asumimos que nuestro interés a la hora de introducir la semántica de pruebas no es otro que el de tener un mecanismo para definir la equivalencia semántica entre procesos.

En la Figura 2.2, mostramos agrupadas las reglas que definen la interacción entre los procesos y las pruebas.

El siguiente paso de cara a definir la semántica de pruebas consiste en definir el concepto de *computación* de la composición entre un proceso y una prueba, distinguiéndose a continuación las *computaciones con éxito*, que serán aquéllas en las cuales la prueba ejecuta la acción de aceptación ω. Como quiera que las computaciones tendrán asociada una probabilidad, agregando las de las computaciones con éxito obtendremos finalmente la probabilidad con la cual un proceso pasa una prueba.

$$\frac{P \succ\!\!\rightarrow_p P' \wedge T_\oplus = 0}{P \mid T \longmapsto_p P' \mid T} \qquad \frac{T \succ\!\!\rightarrow_p T' \wedge P_\oplus = 0}{P \mid T \longmapsto_p P \mid T'} \qquad \frac{P \succ\!\!\rightarrow_p P' \wedge T \succ\!\!\rightarrow_q T'}{P \mid T \longmapsto_{p\cdot q} P' \mid T'}$$

$$\frac{P \xrightarrow{a}_p P' \wedge T \xrightarrow{a}_q T'}{P \mid T \longmapsto_{r_1} P' \mid T'} \qquad \frac{T \xrightarrow{\omega}_p T' \wedge P_\oplus = 0}{P \mid T \overset{\omega}{\longmapsto}_{r_2} Nil}$$

donde $r_1 = \frac{p\cdot q}{\mu(P,T)}$ y $r_2 = \frac{p}{\mu(P,T)}$

Figura 2.2: Reglas para la composición de procesos y pruebas.

Definición 2.12 Sea P un proceso y T una prueba. Una *computación* C es una secuencia *maximal* de transiciones de la forma

$$C = P \mid T \longmapsto_{p_1} P_1 \mid T_1 \longmapsto_{p_2} \cdots P_{n-1} \mid T_{n-1} \overset{*}{\longmapsto}_{p_n} R$$

donde $*$ denota o a una etiqueta vacía o a la acción especial ω, y por *maximal* entendemos que C no se puede extender más, esto es no existen $p > 0, R'$ tales que $R \overset{*}{\longmapsto}_p R'$.

Cuando la última transición es de la forma $P_{n-1} \mid T_{n-1} \overset{\omega}{\longmapsto}_{p_n} Nil$, diremos que la computación ha tenido *éxito*. Denotaremos por $Ex(P,T)$ al *conjunto* de *computaciones con éxito* a partir de $P \mid T$.

La probabilidad de una computación C, $Pr(C)$, se define inductivamente en la forma:

$$Pr(Nil) = 1$$
$$Pr(P \mid T \overset{*}{\longmapsto}_p C) = p \cdot Pr(C)$$

Finalmente, definimos la función $pass(P\,,\,T)$, que nos indica la probabilidad con la que el proceso P *pasa* la prueba T, en la forma

$$pass(P\,,\,T) = \sum_{C \in Ex(P,T)} Pr(C)$$

□

Ahora, dada una familia de pruebas, podemos definir la noción de *equivalencia de pruebas* asociada a la misma. En principio, según consideremos una u otra familia de pruebas, tendremos diferentes relaciones de equivalencia.

Definición 2.13 Dado un conjunto de pruebas probabilísticas, $\mathcal{T}$, y dos procesos P y Q, definimos la relación de *equivalencia de pruebas* inducida por la familia de pruebas $\mathcal{T}$, $\approx_{\mathcal{T}}$, en la forma

$$P \approx_{\mathcal{T}} Q \text{ sii } \forall\, T \in \mathcal{T} :\ pass(P\,,\,T) = pass(Q\,,\,T)$$

□

Para terminar esta sección damos el siguiente lema que nos indica que tanto las pruebas que contienen elecciones internas, como las que contienen elecciones externas induciendo no determinismo debido a que las dos pruebas compuestas contengan acciones comunes entre las que se ofrecen en su primer paso, pueden ser eliminadas por resultar redundantes.

Lema 2.14 Sea P un proceso, sean T, T' pruebas, y sea $a \in Act$. Entonces,

1. $pass(P\,,\,(T \oplus_p T')) = p \cdot pass(P\,,\,T) + (1-p)\, pass(P\,,\,T')$
2. $pass(P\,,\,(a;T) +_p (a;T')) = pass(P\,,\,a;(T \oplus_p T'))$

Demostración: Para el primer apartado, si el proceso no puede realizar transiciones internas, el resultado es trivial, por aplicación de la regla $(PAR2)$. En caso contrario, *consideremos los siguientes multiconjuntos de procesos y pruebas:*

$$\{\!|\, P_i \mid P >\!\!\longrightarrow^*_{p_i} P_i \,|\!\}\qquad \{\!|\, T_j \mid T >\!\!\longrightarrow^*_{q_j} T_j \,|\!\}\qquad \{\!|\, T'_k \mid T' >\!\!\longrightarrow^*_{r_k} T'_k \,|\!\}$$

Utilizando repetidas veces la regla $(PAR3)$ hasta que la prueba y/o el proceso no puedan realizar más transiciones internas, y utilizando después si ello es necesario las reglas $(PAR1)$ y $(PAR2)$ obtenemos

$$P \mid T \longmapsto_{s_{11}} \cdots \longmapsto_{s_{ij}} P_i \mid T_j$$
$$P \mid T' \longmapsto_{t_{11}} \cdots \longmapsto_{t_{ik}} P_i \mid T'_k$$

donde $s_{11} \cdots s_{ij} = p_i \cdot q_j$ y $t_{11} \cdots t_{ik} = p_i \cdot r_k$.

En consecuencia tenemos $pass(P\ ,\ T) = \sum_{i,j} p_i \cdot q_j \cdot pass(P_i\ ,\ T_j)$ mientras que $pass(P\ ,\ T') = \sum_{i,k} p_i \cdot r_k \cdot pass(P_i\ ,\ T'_k)$.

Consideremos ahora $P \mid (T \oplus_p T')$; utilizando entonces un argumento similar al empleado en el caso anterior, obtenemos las siguientes computaciones

$$P \mid (T \oplus_p T') \longmapsto_{p \cdot p'_i} P'_i \mid T \longmapsto_{s'_{11}} \cdots \longmapsto_{s'_{ij}} P_i \mid T_j$$
$$P \mid (T \oplus_p T') \longmapsto_{(1-p) \cdot p'_i} P'_i \mid T' \longmapsto_{r'_{11}} \cdots \longmapsto_{r'_{ik}} P_i \mid T'_k$$

donde cada proceso P'_i verifica $P \rightarrowtail_{p'_i} P'_i \rightarrowtail^*_{p''_i} P_i$, teniéndose además $p'_i \cdot p''_i = p_i$.

Entonces, utilizando un razonamiento sencillo por inducción, obtenemos que $(p \cdot p'_i) \cdot \prod s'_{ij} = p \cdot p_i \cdot q_j$ y $((1-p) \cdot p'_i) \cdot \prod r'_{ik} = (1-p) \cdot p_i \cdot r_k$, de lo cual se sigue fácilmente el resultado a probar.

La demostración del segundo apartado se basa en la utilización del multiconjunto de los procesos $\{\!|\ P_i \mid P \xrightarrow{a}_{p_i} P_i\ |\!\}$. Si dicho multiconjunto es vacío ya habríamos terminado. En otro caso, tendremos por un lado las computaciones

$$P \mid (a; T +_p a; T') \longmapsto_{\frac{p_i \cdot p}{\sum p_i}} P_i \mid T$$
$$P \mid (a; T +_p a; T') \longmapsto_{\frac{p_i \cdot (1-p)}{\sum p_i}} P_i \mid T'$$

mientras que por el otro tendremos

$$P \mid a; (T \oplus_p T') \longmapsto_{\frac{p_i}{\sum p_i}} P_i \mid (T \oplus_p T')$$

con lo que utilizando el apartado anterior llegamos al resultado que queríamos demostrar. □

2.4 Separación entre transiciones internas y externas

En esta sección vamos a comentar uno de los puntos que más alejan a nuestra semántica operacional de otras propuestas anteriores para lenguajes probabilísticos.

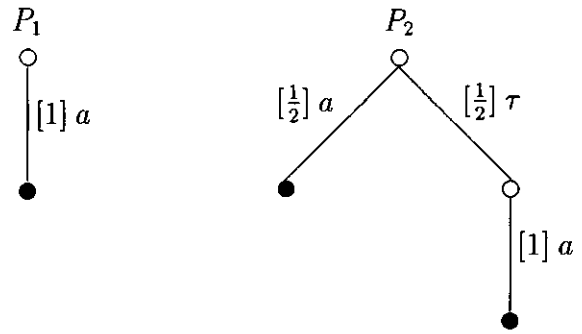
Nos referimos al hecho de que en nuestra semántica operacional no se mezclan las transiciones internas con las externas.

Consideremos el siguiente proceso no probabilístico: $P = a + (b \oplus c)$. En la semántica operacional definida en [dNH87], P tiene las transiciones $P \xrightarrow{a} Nil$, $P \rightarrowtail a + b$ y $P \rightarrowtail a + c$. Sin embargo, y al margen de las probabilidades, en nuestra semántica operacional este mismo proceso sólo tiene las dos últimas transiciones. Los problemas que aparecerían si mezcláramos transiciones internas y externas en nuestra semántica operacional son similares a los que aparecen en el marco de álgebras de procesos probabilísticas basadas en CCS, en las que se maneja una acción oculta τ, cuando se define una semántica de pruebas. Para ilustrar estos problemas, nos centraremos en el trabajo desarrollado en [CSZ92, YCDS94], y veremos dos de los problemas que causa el hecho de mezclar estos dos tipos de transiciones: las τ's son necesarias en las pruebas, o equivalentemente el hecho de quitarlas restaría capacidad a la hora de distinguir entre procesos con el conjunto restante de pruebas (obsérvese que nosotros en cambio podemos eliminar de las pruebas las elecciones internas en virtud del Lema 2.14); y las τ's no se pueden *elevar* en los procesos hasta sustituirlas por elecciones internas entre estados, lo cual impide hablar de una noción de *estado* global en los que las acciones ocultas no aparezcan (cosa que a nosotros no nos ocurre, como quedará probado por nuestra semántica denotacional basada en árboles de aceptación que más adelante desarrollaremos, en cuyos estados no aparecen acciones ocultas [NdFL95]).

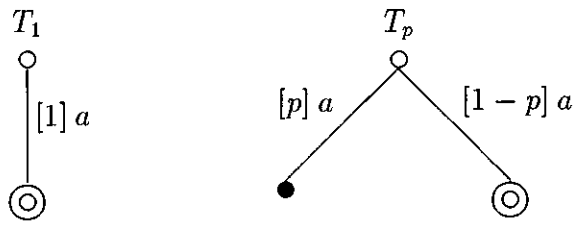
En el resto de la sección, usaremos la representación gráfica de los procesos y de las pruebas considerados como sistemas probabilísticos de transiciones etiquetadas, junto con las definiciones dadas en [CSZ92]. En dicha representación gráfica, un círculo negro denotará un estado bloqueado, mientras que dos círculos concéntricos denotarán un estado de éxito.

2.4.1 Las τ's serían necesarias en las pruebas

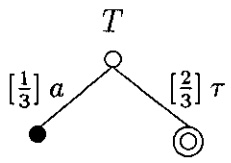
Consideremos los sistemas probabilísticos de transiciones etiquetadas que siguen:



Tenemos que para cualquier prueba T en la cual no aparezcan acciones ocultas, τ's, $pass(P_1, T) = pass(P_2, T)$. Ello es debido a que las únicas *pruebas fundamentales* (denominadas trazas probabilísticas en [YCDS94]) que no contienen τ's que podrían distinguirlos son de la forma



Pero si consideramos la prueba T que sigue

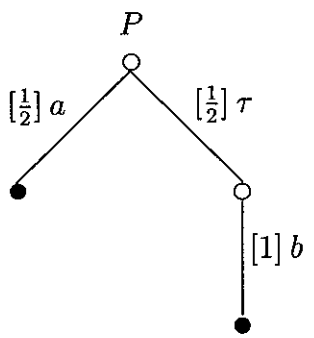


tenemos que $pass(P_1\ ,\ T) = \frac{2}{3}$ mientras que $pass(P_2\ ,\ T) = \frac{7}{9}$. Ello es así dado que $\frac{7}{9} = \frac{1}{6} \cdot \frac{2}{3} + \frac{1}{3} + \frac{1}{3}$, donde el primer sumando indica que P_2 ejecuta τ mientras que T no lo hace, y tras ello se ejecuta la τ de T; el segundo sumando indica que T ejecuta τ y P_2 no; mientras que el último sumando indica que los dos ejecutan τ a la vez.

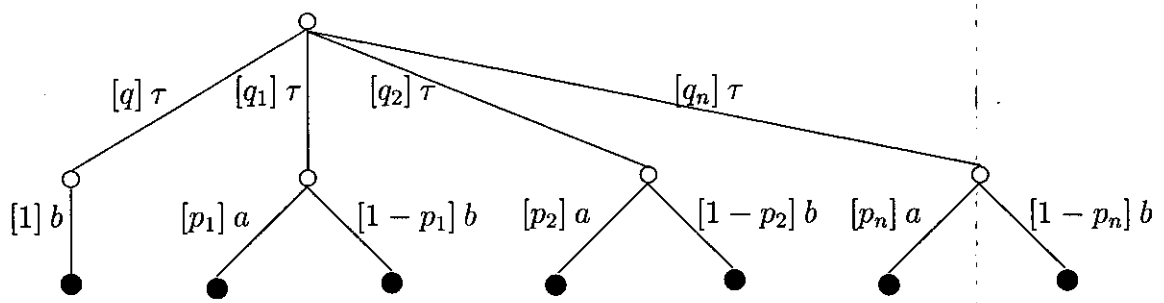
Hay aún casos peores. Por ejemplo, si consideramos los procesos (utilizando una sintaxis *a lo* CCS) $P = a; Nil$ y $Q = \tau; a; Nil$ tenemos que en el marco de pruebas definido en [CSZ92, YCDS94] estos procesos no son equivalentes, lo cual no debería ocurrir en una semántica de pruebas, es decir, en una semántica puramente observacional.

2.4.2 Las τ's no se pueden *elevar* en los procesos probabilísticos

Consideremos el sistema probabilístico de transiciones etiquetadas que sigue:



Si pudiéramos *elevar* las τ's, en el sentido de que existiera una elección interna en cabeza (es decir, algo similar a lo que expresa la ley de CCS $(\tau; b)+\tau; (a+b) \approx a+\tau; b)$, este proceso sería equivalente a un proceso P' de la forma



donde los p_i son todos distintos (si hubiera dos iguales, se podrían agrupar sumando el valor de las correspondientes q_i), y $q + \sum_i q_i = 1$.

Claramente, q debería ser igual a $\frac{1}{2}$, dado que la probabilidad con la cual P pasa la prueba $a;\omega$ es igual a $\frac{1}{2}$, y q es un valor que indica la probabilidad de no poder ejecutar la acción a. También es claro que *debajo* de cada τ debe aparecer la acción b dado que la probabilidad con la cual P pasa la prueba $b;\omega$ es igual a 1.

Consideremos entonces la siguiente tabla, en la que la primera columna denota una serie de pruebas, la segunda indica la probabilidad con la cual el proceso P pasa esa prueba, y la tercera indica la probabilidad con la cual P' pasa la prueba.

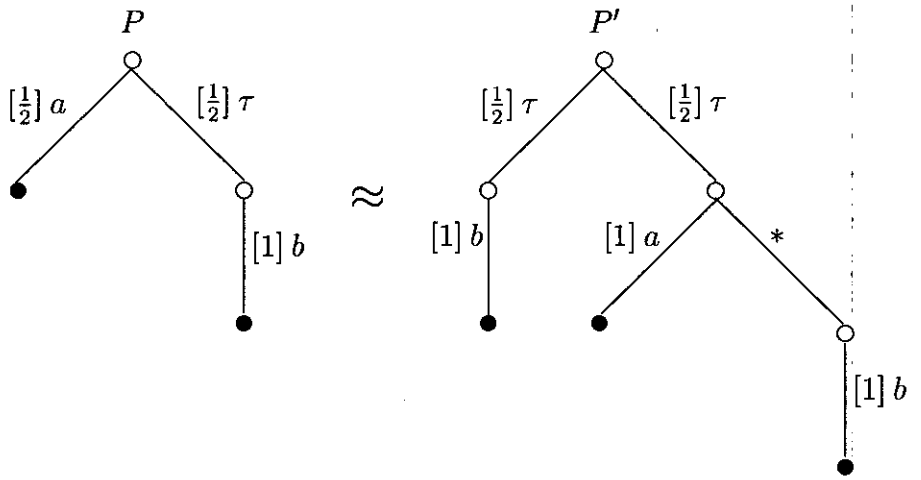
	P	P'	
$a;\omega +_{\frac{1}{2}} b; Nil$	$\frac{1/2\cdot 1/2}{1/4+1/2} = \frac{1}{3}$	$\sum q_i \cdot p_i$	(1)
$a; Nil +_{\frac{1}{2}} b;\omega$	$\frac{1/2}{1/4+1/2} = \frac{2}{3}$	$\frac{1}{2} + \sum q_i \cdot (1-p_i)$	(2)
$\tau; Nil +_{\frac{1}{2}} b;\omega$	$\frac{1/2\cdot 1/2}{1/4+1/2} \cdot \frac{1}{2} = \frac{1}{6}$	$\frac{1}{4} \cdot \frac{1}{2} + \sum \frac{q_i}{2} \cdot \frac{(1-p_i)\cdot 1/2}{1/2+(1-p_i)\cdot 1/2}$	(3)
$\tau; Nil +_{\frac{1}{2}} a;\omega$	$1/4$	$\sum \frac{q_i}{2} \cdot \frac{p_i\cdot 1/2}{1/2+p_i\cdot 1/2}$	(4)

Supongamos que P y P' pasan todas las pruebas con la misma probabilidad. Por (3) tenemos $\sum \frac{q_i}{2} \cdot \frac{(1-p_i)\cdot 1/2}{1/2+(1-p_i)\cdot 1/2} = \frac{1}{24}$. Dividiendo entre $\frac{1}{4}$ en los dos lados de la igualdad anterior, tenemos $\sum q_i \cdot \frac{1-p_i}{1/2+(1-p_i)\cdot 1/2} = \frac{1}{6}$. Por (2) tenemos $\sum q_i \cdot (1-p_i) = \frac{1}{6}$, lo cual implica $\sum q_i \cdot \frac{1-p_i}{1-p_i/2} = \sum q_i \cdot (1-p_i)$, lo cual es obviamente falso, dado que $(1-p_i) < \frac{1-p_i}{1-p_i/2}$, puesto que $p_i > 0$.

Otro razonamiento para mostrar que P y P' no son equivalentes es el siguiente. Por (4) tenemos $\frac{1}{4} = \sum \frac{q_i}{2} \cdot \frac{p_i\cdot 1/2}{1/2+p_i\cdot 1/2} = \sum \frac{q_i}{2} \cdot \frac{p_i}{1+p_i}$. Entonces llegamos a una contradicción dado que $\frac{1}{2} = \sum q_i \cdot \frac{p_i}{1+p_i} < \sum q_i \cdot p_i \overset{(1)}{=} \frac{1}{3}$.

Estos resultados nos llevan a pensar que bajo el modelo definido en [CSZ92, YCDS94] no se podrá conseguir una semántica denotacional basada en una noción de *estado* en los que las acciones ocultas, τ, no aparezcan.

La solución a este problema no es ni mucho menos sencilla. En el marco semántico de [CSZ92, YCDS94], una posible solución podría consistir en introducir una noción de prioridad, y considerar un valor *estático* para las probabilidades de las τ's (ello se podría conseguir parcialmente mediante el uso de prenormalizaciones como en [NdF95b]), de forma que tuviéramos



donde $\xrightarrow{*}$ denota una transición de baja prioridad. Con ello el proceso $a +_{\frac{1}{2}} \tau; b$ sería equivalente al proceso $b \oplus_{\frac{1}{2}} (a +_1 b)$.

Capítulo 3

Semántica de Pruebas para el Modelo Reactivo

En este capítulo daremos una interpretación del modelo *reactivo* [LS89, vGSST90] utilizando una semántica de pruebas, al igual que se hizo en [NdF95c] para una extensión probabilística de LOTOS, si bien aquí lo haremos para el álgebra de procesos definida en el Capítulo 2. Mostraremos que la relación de equivalencia entre procesos inducida por la semántica de pruebas para el modelo reactivo no es una congruencia, pues en concreto los operadores $+_p$ en general no la preservan.

A continuación, daremos una caracterización alternativa de la semántica de pruebas definida para el modelo reactivo basada en el comportamiento operacional de los procesos. Esta caracterización alternativa del modelo reactivo estará basada en una cierta noción de *traza probabilística* [NdF95c] (no confundir con las *trazas probabilísticas* definidas en [YCDS94]). Estas trazas probabilísticas serán pares (*traza, valor*) donde la segunda componente denotará la probabilidad de que el proceso ejecute la traza y llegue a un estado estable.

Finalmente, daremos una semántica denotacional para el subconjunto de PPA en el que no aparecen los operadores de elección externa, la cual es completamente abstracta respecto de la semántica de pruebas para el modelo reactivo. Como en el caso de la caracterización alternativa de la que hemos hablado anteriormente,

esta semántica denotacional estará basada en las trazas de los procesos, es decir la semántica de un proceso vendrá dada por un conjunto de pares (traza, probabilidad).

3.1 Definición del modelo Reactivo

Siguiendo una interpretación intuitiva del modelo reactivo, podemos considerar que en el mismo el entorno, es decir las pruebas, sólo pueden ofrecer una acción en cada momento. Usando la terminología empleada en [Mil80], diríamos que *solamente se puede pulsar un botón a la vez.* Esto da lugar a que al interpretar este modelo en el ámbito de una semántica de pruebas, las pruebas sean simplemente trazas que terminan con una acción de acceptación[1].

Definición 3.1 El conjunto de las *pruebas reactivas*, que denotaremos por $\mathcal{R}$, se define mediante la siguiente expresión BNF:

$$T = \omega \mid a;T \qquad (a \in Act)$$

Escribiremos $P \approx_{\mathcal{R}} Q$ sii $\forall T \in \mathcal{R} : pass(P\,,\,T) = pass(Q\,,\,T)$. □

Si bien es muy cómodo trabajar con el modelo reactivo debido a la sencillez de las pruebas que lo caracterizan, y se puede encontrar una caracterización alternativa bastante *razonable* en términos de trazas probabilísticas (como se verá en la siguiente sección), este modelo presenta dos problemas serios.

El primero de ellos es intrínseco, debido a la sencillez de las pruebas admisibles, consistiendo en que no hay ninguna forma de manejar simultáneamente varias acciones diferentes ofrecidas en una elección externa, por lo que el significado de la probabilidad en el operador de elección externa se *pierde* parcialmente, como probaremos en la Proposición 3.3.

Definición 3.2 Dado un proceso P, definimos el conjunto de sus *acciones iniciales*, que denotaremos por $ini(P)$, en la forma $ini(P) = \{a \mid \exists P', p : \; P \stackrel{*}{\rightarrow\!\!\!\!\!\rightarrow}_p P' \xrightarrow{a}\}$. □

[1] Obviamente, las pruebas que no terminen con una acción de aceptación no tienen interés, dado que la probabilidad de que un proceso pase este tipo de pruebas es igual a cero.

Proposición 3.3 Sean P y P' procesos tales que $ini(P) \cap ini(P') = \emptyset$. Entonces, para cualesquiera $0 < p, q < 1$, se cumple $P +_p P' \approx_{\mathcal{R}} P +_q P'$.

Demostración: La demostración es sencilla teniendo en cuenta que las pruebas sólo pueden ofrecer una acción en cada momento, por lo que teniendo en cuenta que los conjuntos iniciales son disjuntos, una vez que la elección externa es estable, la prueba sólo puede interaccionar por medio de la regla $(PAR4)$ a lo más con uno de los dos procesos (P ó P'), pero en ningún caso con los dos, con lo que el valor asociado a la elección externa resulta irrelevante. Utilizamos también el hecho de que las transiciones internas que $P +_p P'$ y $P +_q P'$ pueden ejecutar son las mismas, y sus probabilidades iguales debido a las reglas $(EXT1)$, $(EXT2)$ y $(EXT3)$. □

Por ejemplo, $(a; P) +_{\frac{1}{2}} (a'; P') \approx_{\mathcal{R}} (a; P) +_{\frac{1}{3}} (a'; P')$. Esta *pérdida* de información probabilística nos llevará a estudiar en un capítulo posterior otro modelo donde podamos controlar de una forma más precisa las probabilidades asociadas a las elecciones externas: el modelo *generativo.*

El segundo de los problemas, aunque de orden técnico, es bastante peor que el anterior, puesto que la relación $\approx_{\mathcal{R}}$ no es una congruencia respecto de los operadores de elección externa, como se ve en el siguiente

Ejemplo 3.4 Sean $P = (a; c) +_{\frac{1}{2}} b$ y $P' = (a; c) +_{\frac{1}{3}} b$. Como se vió anteriormente, $P \approx_{\mathcal{R}} P'$. Consideremos ahora el proceso $Q = a; b$. Tenemos que $P +_{\frac{1}{2}} Q \not\approx_{\mathcal{R}} P' +_{\frac{1}{2}} Q$, dado que $pass(P +_{\frac{1}{2}} Q\ ,\ a; b; \omega) = \frac{2}{3} \neq \frac{3}{4} = pass(P' +_{\frac{1}{2}} Q\ ,\ a; b; \omega)$. □

Para el resto de los operadores sí se podría probar de manera trivial que la relación $\approx_{\mathcal{R}}$ es una congruencia.

Proposición 3.5 Siendo P y P' procesos, se verifica:

- $P \approx_{\mathcal{R}} P' \Longrightarrow \forall a \in Act :\ a; P \approx_{\mathcal{R}} a; P'$.
- $P \approx_{\mathcal{R}} P' \Longrightarrow \forall Q \in \mathrm{PPA}, p \in (0,1) :\ P \oplus_p Q \approx_{\mathcal{R}} P' \oplus_p Q \ \wedge\ Q \oplus_p P \approx_{\mathcal{R}} Q \oplus_p P'$.

□

3.2 Una Caracterización Alternativa

Una vez definido el modelo reactivo en el ámbito de una semántica de pruebas, vamos a presentar en esta sección una caracterización alternativa basada en el comportamiento operacional del proceso. Para ello, lo primero que haremos será extender la relación $\xrightarrow{a}_p$ entre procesos, a secuencias de acciones.

Definición 3.6 Dados dos procesos P y P', diremos que P evoluciona a P' mediante una *transición externa generalizada* con probabilidad p, ejecutando la secuencia de acciones $s = \langle a_1, a_2, \ldots a_n \rangle$, si la transición $P \xrightarrow{s}_p P'$ se puede derivar a partir de las siguientes reglas:

$$P \xrightarrow{\epsilon}_p P' \text{ sii } P \rightarrowtail^*_p P'$$
$$P \xrightarrow{(a)\circ s'}_p P' \text{ sii } \exists P_1, P_2, p_1, p_2, p_3 : P \rightarrowtail^*_{p_1} P_1 \wedge P_1 \xrightarrow{a}_{p_2} P_2 \wedge P_2 \xrightarrow{s'}_{p_3} P'$$

donde $p = p_1 \cdot \frac{p_2}{\sum \{\!\{ q \mid P_1 \xrightarrow{a} q \}\!\}} \cdot p_3$. Escribiremos $P \xrightarrow{s}_0 P'$ si $P \xrightarrow{s}_p P'$ no se puede derivar a partir de las reglas anteriores para ningún $p > 0$. □

Como en el caso de las transiciones definidas anteriormente ($\rightarrowtail_p$, $\xrightarrow{a}_p$, $\rightarrowtail^*_p$) debemos tener cuidado con las repeticiones que se puedan producir al poderse generar una transición $P \xrightarrow{s}_p P'$ de varias formas distintas, de modo que formalmente deberíamos hablar del multiconjunto de transiciones externas generalizadas.

Intuitivamente, $P \xrightarrow{s}_p P'$ si P puede realizar una cierta secuencia de transiciones correspondiente a las acciones que aparecen en s, realizando posiblemente entre medias de dichas transiciones observables una serie de transiciones internas, y llegando tras todas ellas a un proceso estable P'. El valor p representa la probabilidad *global* de la computación derivada, y se obtiene multiplicando las probabilidades de todas las elecciones no deterministas que se resuelven a lo largo de la computación. Es importante notar que además del no determinismo que se corresponda a las transiciones internas generalizadas resueltas durante la computación, debemos considerar el no determinismo inducido por la elección entre diferentes transiciones observables que estén etiquetadas con una misma acción.

En adelante, utilizaremos la notación:

$$\bullet\ P \xrightarrow{s}_p \stackrel{\text{not.}}{=\!=} \exists P' :\ P \xrightarrow{s}_p P' \qquad \bullet\ P \xrightarrow{s} \stackrel{\text{not.}}{=\!=} \exists p, P' :\ P \xrightarrow{s}_p P'$$

A partir de la Definición 3.6, vamos a definir una semántica basada en *trazas probabilísticas* [NdF95c]. Intuitivamente, una traza probabilística es una secuencia de acciones junto con una valor que indica la probabilidad de ejecutar esta secuencia. Sin embargo, estas trazas son distintas de las definidas en [JS90], y no tienen nada que ver con las definidas en [YCDS94] (a éstas últimas las llamaremos *barbas probabilísticas* en este trabajo). La diferencia con el modelo de trazas descrito en [JS90] es que nosotros, a la hora de definir la probabilidad asociada a una traza, sólo consideramos las elecciones no deterministas que se produzcan, mientras que en su caso se consideran todas. Así, el proceso $P = a +_{\frac{1}{2}} b$ tendrá en nuestro caso las trazas $\langle a \rangle$ y $\langle b \rangle$ con probabilidad asociada igual a 1, mientras que en su caso tendrían probabilidad asociada igual a $\frac{1}{2}$. No obstante, cuando consideremos el subconjunto de PPA en el que no aparecen operadores de elección externa (Sección 3.3) sí tendremos un modelo similar al definido en [JS90].

Definición 3.7 Siendo P un proceso, definimos el *conjunto de trazas probabilísticas* del mismo, que denotaremos por $\mathit{trazas}(P)$, en la forma:

$$\mathit{trazas}(P) = \{(s,p) \mid P \xrightarrow{s} \ \wedge\ p = \sum \{\!|\ q \mid P \xrightarrow{s}_q\ |\!\}\ \wedge\ p > 0\}$$

□

A continuación, presentaremos una serie de ejemplos que ilustran esta definición.

Ejemplo 3.8

- $\mathit{trazas}(\mathit{Nil}) = \{(\epsilon, 1)\}$, dado que $\mathit{Nil} >\!\!\longrightarrow_1^* \mathit{Nil}$, y por tanto $\mathit{Nil} \xrightarrow{\epsilon}_1 \mathit{Nil}$, pues al ser *Nil* un proceso estable podemos aplicar la primera regla de la Definición 2.8.
- $\mathit{trazas}(\Omega) = \emptyset$. Esto es una consecuencia directa del hecho de que $\Omega >\!\!\not\longrightarrow_p^*$ para ningún $p > 0$, o lo que es lo mismo $\Omega \xrightarrow{\epsilon}_o$, pues Ω no es estable.

- $trazas(a;\Omega) = trazas(b;\Omega) = \{(\epsilon,1)\}$. Esto sucede dado que dichos procesos, después de ejecutar la acción a ó b no llegan a un proceso estable. Si consideráramos $trazas(a;\Omega) = \{(\epsilon,1),(\langle a\rangle,1)\}$ y $trazas(b;\Omega) = \{(\epsilon,1),(\langle b\rangle,1)\}$, tendríamos que la semántica de trazas no sería equivalente a la de pruebas, dado que no existe ninguna prueba reactiva que distinga a estos procesos. Naturalmente, ambos procesos se pueden distinguir mediante la prueba $T = a +_p \omega$, pero ésta no es una prueba reactiva.

- $trazas(a \oplus_{\frac{1}{2}} b) = \{(\epsilon,1),(\langle a\rangle,\frac{1}{2}),(\langle b\rangle,\frac{1}{2})\}$, mientras que por otro lado tenemos $trazas(a +_{\frac{1}{2}} b) = \{(\epsilon,1),(\langle a\rangle,1),(\langle b\rangle,1)\}$. Es decir, nuestra semántica de trazas probabilísticas es capaz de distinguir entre las elecciones externas y las internas, en contraste con lo que sucede en el caso no probabilístico. Esta distinción resulta del hecho de que en el caso probabilístico juegan un papel las probabilidades asociadas a las elecciones internas, o en general, el no determinismo que aparezca en los procesos ya sea consecuencia de elecciones internas o de no determinismo en elecciones externas, mientras que *olvidamos*, por resultar inútiles, las probabilidades asociadas a las elecciones deterministas.

- Siendo $P = ((a;b) +_{\frac{1}{3}} (a;c)) \oplus_{\frac{1}{2}} b$, tenemos

$$trazas(P) = \{(\epsilon,1),(\langle a\rangle,\frac{1}{2}),(\langle b\rangle,\frac{1}{2}),(\langle a,b\rangle,\frac{1}{6}),(\langle a,c\rangle,\frac{1}{3})\}$$

- Para $P = recX.(a \oplus_p X)$ tenemos $trazas(P) = \{(\epsilon,1),(\langle a\rangle,1)\}$. Ello es así pues las transiciones de P son de la forma:

$$P \rightarrowtail_1 a \oplus_p P \rightarrowtail_p a \xrightarrow{a}_1$$
$$(\text{i.e. } P \xrightarrow{\langle a\rangle}_p)$$

$$P \rightarrowtail_1 a \oplus_p P \rightarrowtail_{1-p} P \rightarrowtail_1 a \oplus_p P \rightarrowtail_p a \xrightarrow{a}_1$$
$$(\text{i.e. } P \xrightarrow{\langle a\rangle}_{(1-p)\cdot p})$$

$$P \rightarrowtail_1 a \oplus_p P \rightarrowtail_{1-p} P \rightarrowtail_1 a \oplus_p P \rightarrowtail_{1-p} P \rightarrowtail_1 a \oplus_p P \rightarrowtail_p a \xrightarrow{a}_1$$
$$(\text{i.e. } P \xrightarrow{\langle a\rangle}_{(1-p)^2\cdot p})$$

... ...

Con lo cual, la probabilidad de la traza $\langle a \rangle$ en el proceso P es igual a

$$p \cdot \sum_{i=0}^{\infty}(1-p)^i = p \cdot \frac{1}{1-(1-p)} = \frac{p}{p} = 1$$

□

La semántica de trazas induce una relación de equivalencia entre procesos de manera canónica.

Definición 3.9 Sean P y P' dos procesos. Diremos que P y P' son *equivalentes bajo trazas probabilísticas*, y escribiremos $P \equiv_{\text{trazas}} P'$, sii $\mathit{trazas}(P) = \mathit{trazas}(P')$. □

Lema 3.10 Para cada proceso P se tiene

$$(s,p) \in \mathit{trazas}(P) \iff \mathit{pass}(P\,,\,\tilde{s}) = p$$

donde $\tilde{s}$ denota a la prueba secuencial formada por las acciones que aparecen en s, terminando con la acción de acceptación ω.

Demostración: Consideraremos el multiconjunto de pares (proceso, probabilidad) que sigue:

$$\tilde{P} = \{\!|\ (P_i, p_i) \mid P \rightarrowtail^*_{p_i} P_i\ |\!\}$$

y realizaremos una demostración por inducción sobre la longitud de la traza s.

Caso Base: $(s = \epsilon,\ \tilde{s} = \omega)$.

Si P es estable el resultado es trivial, dado que $\mathit{pass}(P\,,\,\omega) = 1$, al ser $(PAR5)$ la única regla que puede aplicarse; mientras que $(\epsilon, 1) \in \mathit{trazas}(P)$, dado que $P \rightarrowtail^*_1 P$, y por tanto $P \xrightarrow{\epsilon}_1 P$.

Si P no es estable, tenemos que $(\epsilon, p) \in \mathit{trazas}(P)$ para $p = \sum p_i$. Consideremos entonces $P \mid \omega$. Aplicando reiteradamente la regla $(PAR1)$ y finalmente la regla $(PAR5)$ obtenemos las siguientes computaciones:

$$P \mid \omega \longmapsto_{p_{i1}} P_{i1} \mid \omega \cdots \longmapsto_{p_{in_i}} P_i \mid \omega \xmapsto{\omega}_1 \mathit{Nil}$$

donde $\prod_{j=1}^{n_i} p_{ij} = p_i$, de lo cual se deduce que $\mathit{pass}(P\,,\,\omega) = \sum p_i$ como queríamos demostrar.

Caso Inductivo: $(s = \langle a_1, a_2, \ldots, a_n\rangle = a_1 \circ s',\ \tilde{s} = a_1; a_2; \ldots a_n; \omega)$.

Consideremos el multiconjunto de pares (proceso, probabilidad) que sigue

$$\widetilde{P_{a_1}} = \{\!|\ (R_j, q_j \cdot q'_j) \mid P \rightarrowtail^*_{q_j} Q_j \xrightarrow{a_1}_{q'_j} R_j\ |\!\}$$

y las secuencias de transiciones de la forma

$$P \rightarrowtail^*_{q_j} Q_j \xrightarrow{a_1}_{q'_j} R_j \xrightarrow{s'}_{q''_j} R'_j$$

Si $(s,p) \in \mathit{trazas}(P)$, entonces, aplicando la Definición 3.6, tenemos

$$p = \sum \{\!|\ q_j \cdot \frac{q'_j}{\sum \{\!|\ r \mid Q_j \xrightarrow{a_1}_r\ |\!\}} \cdot q''_j\ |\!\} = \sum \{\!|\ q_j \cdot \frac{q'_j}{\sum \{\!|\ r \mid Q_j \xrightarrow{a_1}_r\ |\!\}} \cdot q_j^{s'}\ |\!\} \qquad (3.1)$$

donde $q_j^{s'} = \sum \{\!|\ q''_j \mid R_j \xrightarrow{s'}_{q''_j}\ |\!\}$, por lo que $(s', q_j^{s'}) \in \mathit{trazas}(R_j)$.

Las computaciones del proceso P en paralelo con la prueba $\tilde{s}$ serán de la forma:

$$P \mid \tilde{s} \longmapsto^*_{q_j} Q_j \mid \tilde{s} \longmapsto_{\widehat{q'_j}} R_j \mid \tilde{s'} \qquad \text{con } \widehat{q'_j} = \frac{q'_j}{\mu(Q_j, \tilde{s})}$$

donde $\longmapsto^*_p$ denota una serie de transiciones derivables por medio de la regla $(PAR1)$, cuyas probabilidades tienen un producto igual a p. En caso de que P fuese estable, tales transiciones no aparecerían, por lo que podríamos tomar $p = 1$. El valor $\widehat{q'_j}$ se obtiene aplicando la regla $(PAR4)$ pues $Q_j \xrightarrow{a_1}_{q'_j} R_j$ y $\tilde{s} \xrightarrow{a_1}_1 \tilde{s'}$. En tal caso, $\mu(Q_j, \tilde{s}) = \sum \{\!|\ r \mid Q_j \xrightarrow{a_1}_{r'}\ |\!\}$, por lo que $\widehat{q'_j} = \frac{q'_j}{\sum \{\!|\ r \mid Q_j \xrightarrow{a_1}_r\ |\!\}}$.

Utilizando todos estos resultados obtenemos

$$\mathit{pass}(P\ ,\ \tilde{s}) = \sum \{\!|\ q_j \cdot \frac{q'_j}{\sum \{\!|\ r \mid Q_j \xrightarrow{a_1}_r\ |\!\}} \cdot \mathit{pass}(R_j\ ,\ \tilde{s'})\ |\!\} \qquad (3.2)$$

Pero por hipótesis de inducción, $\mathit{pass}(R_j\ ,\ \tilde{s'}) = q_j^{s'}$ sii $(s', q_j^{s'}) \in \mathit{trazas}(R_j)$, con lo que obtenemos que el valor p dado por la fórmula (3.1), coincide con el dado por la fórmula (3.2), como queríamos demostrar. □

A partir del lema anterior, derivamos como corolario trivial que las relaciones de equivalencia $\approx_{\mathcal{R}}$ y $\equiv_{\text{trazas}}$ identifican a los mismos procesos.

Teorema 3.1 (Caracterización Alternativa)

Para cualesquiera procesos P y P' se verifica $P \approx_{\mathcal{R}} P' \iff P \equiv_{\text{trazas}} P'$. □

3.3 Semántica Denotacional

En esta sección vamos a definir una semántica denotacional para un subconjunto del lenguaje PPA que será completamente abstracta con respecto a la semántica de pruebas para el modelo reactivo. Dicho sublenguaje, al que llamaremos PPA′, viene definido por medio de la siguiente

Definición 3.11 Dado un conjunto de acciones Act, y un conjunto de identificadores Id, el conjunto de los procesos que pertenecen a PPA′ se define mediante la expresión BNF

$$P ::= \mathit{Nil} \mid \Omega \mid X \mid a;P \mid P \oplus_p P \mid \mathit{rec}X.P$$

donde $p \in (0,1)$, $a \in Act$, y $X \in Id$. □

Como en el caso de PPA, de ahora en adelante nos restringiremos a procesos que no tengan apariciones libres de variables. En el resto de la sección, comenzaremos por definir el dominio semántico sobre el que vamos a trabajar, definiendo a continuación las funciones semánticas correspondientes a cada uno de los operadores sintácticos. Finalmente, mostraremos que nuestra semántica denotacional identifica a los mismos procesos que la semántica de pruebas para el modelo reactivo.

3.3.1 Dominio Semántico

A continuación pasamos a definir el dominio semántico en el que tomará valores la semántica denotacional de nuestro lenguaje. El dominio semántico, al que denotaremos por $\mathbf{TRA}_{\mathrm{Act}}$, será el de los conjuntos *consistentes* de *trazas probabilísticas*, donde, al igual que en la sección anterior, una traza probabilística vendrá dada por una secuencia de acciones $s \in Act^*$, junto con un valor $p \in (0,1]$. Por *consistente* entendemos que en uno de tales conjuntos no puede aparecer la misma traza con dos probabilidades diferentes. Usualmente denotaremos por $R, R_1, \ldots$ a los elementos de $\mathbf{TRA}_{\mathrm{Act}}$. Antes de pasar a definir la preceptiva relación de orden entre los elementos de $\mathbf{TRA}_{\mathrm{Act}}$, introduciremos una función auxiliar por medio de la cual recuperamos la probabilidad asociada a una traza en un conjunto de trazas probabilísticas.

Definición 3.12 Sean $R \in \mathbf{TRA}_{\mathrm{Act}}$ y $s \in Act^*$. Definimos la función $\mathrm{prob}(R, s)$ en la forma:

$$\mathrm{prob}(R, s) = \begin{cases} p & \text{si } (s, p) \in R \\ 0 & \text{e.o.c.} \end{cases}$$

□

Definición 3.13 Sean $R_1, R_2 \in \mathbf{TRA}_{\mathrm{Act}}$. Escribiremos $R_1 \sqsubseteq_{\mathbf{TRA}} R_2$ si para todo $s \in Act^*$ se tiene $\mathrm{prob}(R_1, s) \leq \mathrm{prob}(R_2, s)$. Escribiremos $R_1 =_{\mathbf{TRA}} R_2$ si para todo $s \in Act^*$ se tiene $\mathrm{prob}(R_1, s) = \mathrm{prob}(R_2, s)$. □

Proposición 3.14 La relación $\sqsubseteq_{\mathbf{TRA}}$ es una relación de orden.

Demostración: Es trivial comprobar que la relación $\sqsubseteq_{\mathbf{TRA}}$ cumple las propiedades reflexiva, antisimétrica y transitiva. □

Teorema 3.2 El par $(\mathbf{TRA}_{\mathrm{Act}}, \sqsubseteq_{\mathbf{TRA}})$ es un *orden parcial completo* (cpo).

Demostración: Tendremos que probar la existencia de elemento mínimo y la existencia de cota superior mínima para una cadena.

Existencia de elemento mínimo

Se comprueba fácilmente que el conjunto vacío, que pertenece a $\mathbf{TRA}_{\mathrm{Act}}$, es el elemento mínimo: Si $R \in \mathbf{TRA}_{\mathrm{Act}}$, tenemos

$$\forall s \in Act^* : \mathrm{prob}(\emptyset, s) = 0 \leq \mathrm{prob}(R, s)$$

con lo que se concluye $\emptyset \sqsubseteq_{\mathbf{TRA}} R$.

Existencia de cota superior mínima (lub)

Sea $\{R_n\}_{n \in \mathbb{N}}$ una *cadena* de elementos de $\mathbf{TRA}_{\mathrm{Act}}$. Entonces, el elemento $\sqcup R_n$, queda definido en la forma

$$\sqcup R_n = \{(s, p^s) \mid p^s = \lim_{n \in \mathbb{N}} \mathrm{prob}(R_n, s) \ \wedge \ p^s > 0\}$$

Una vez definido, tenemos que ver que se trata de la cota superior mínima de la cadena:

1. $\sqcup R_n$ *está bien definido.*

   Al formar los elementos R_n una cadena, para cada s los valores $\text{prob}(R_n, s)$ forman una sucesión creciente y acotada por 1. Por lo tanto, existe el límite de esta sucesión, que será menor o igual a 1, y por tanto $\sqcup R_n$ está bien definido.

2. $\sqcup R_n$ *es cota superior de la cadena.*

   Sea R_j un elemento de la cadena. Tenemos que mostrar que para cualquier $s \in Act^*$ se cumple $\text{prob}(R_j, s) \leq \text{prob}(\sqcup R_n, s)$, lo cual es trivial dado que $\text{prob}(\sqcup R_n, s) = \lim_{n \in \mathbb{N}} \text{prob}(R_n, s) \geq \text{prob}(R_j, s)$.

3. $\sqcup R_n$ *es la mínima cota superior.*

   Sea R un elemento de $\mathbf{TRA}_{\text{Act}}$ tal que $\forall\ n \in \mathbb{N} :\ R_n \sqsubseteq_{\mathbf{TRA}} R$. Entonces, tenemos la siguiente cadena de implicaciones:

$$
\begin{array}{rcll}
R_n & \sqsubseteq_{\mathbf{TRA}} & R & \forall\, n \in \mathbb{N} \\
 & \Downarrow & & \\
\text{prob}(R_n, s) & \leq & \underline{\text{prob}(R, s)} & \underline{\forall s \in Act^*} \ \wedge\ \forall\, n \in \mathbb{N} \\
 & \Downarrow & & \\
p^s = \lim_{n \in \mathbb{N}} \text{prob}(R_n, s) & \leq & \underline{\text{prob}(R, s)} & \underline{\forall s \in Act^*} \\
 & \Downarrow & & \\
\text{prob}(\sqcup R_n, s) & \leq & \text{prob}(R, s) & \forall s \in Act^* \\
 & \Downarrow & & \\
\sqcup R_n & \sqsubseteq_{\mathbf{TRA}} & R &
\end{array}
$$

□

3.3.2 Funciones Semánticas

En esta sección vamos a definir una función semántica para cada uno de los operadores de nuestro lenguaje. Además, mostraremos la monotonía y continuidad de estas funciones, para poder aplicar técnicas de punto fijo a la hora de dar semántica a procesos recursivos.

Nil y Ω

Nil sólo tiene una traza probabilística: la traza vacía, y con probabilidad asociada 1. Por lo tanto,

$$[\![Nil]\!] = \{(\epsilon, 1)\}$$

Ω no puede ejecutar ninguna traza, dado que la única transición que puede realizar es $\Omega \rightarrowtail_1 \Omega$, y por tanto es un proceso que no se puede estabilizar. En otras palabras, $pass(\Omega\ ,\ \omega) = 0$, por lo tanto

$$[\![\Omega]\!] = \emptyset$$

Prefijo

Para toda $a \in Act$ definimos la función semántica $a;\_ :: \mathbf{TRA}_{\mathrm{Act}} \longrightarrow \mathbf{TRA}_{\mathrm{Act}}$. El elemento de $\mathbf{TRA}_{\mathrm{Act}}$ $a; R$, tiene las mismas trazas probabilísticas (y con la misma probabilidad) que las de R, pero a todas las trazas se les ha unido como *prefijo* la acción a. Por lo tanto, la definición de $a; R$ es:

$$a; R = \{(\langle a \rangle \circ s, p) \mid (s, p) \in R\} \cup \{(\epsilon, 1)\}$$

Nótese que además de las trazas de la forma $\langle a \rangle \circ s$, hemos añadido la traza vacía. Esto se debe a que el proceso $a; P$ es estable, y por tanto $a; P \rightarrowtail_1^*$ (o equivalentemente, $pass(a; P\ ,\ \omega) = 1$), con lo cual $(\epsilon, 1) \in trazas(a; P)$.

Proposición 3.15 Para cualquier $a \in Act$, la función $a;\_ :: \mathbf{TRA}_{\mathrm{Act}} \longrightarrow \mathbf{TRA}_{\mathrm{Act}}$ es monótona y continua.

Demostración:

Monotonía.

Sean $R, R' \in \mathbf{TRA}_{\mathrm{Act}}$ tales que $R \sqsubseteq_{\mathbf{TRA}} R'$. Tenemos que probar $a; R \sqsubseteq_{\mathbf{TRA}} a; R'$, o equivalentemente, que $\forall s \in Act^* :\ \mathrm{prob}(R, s) \leq \mathrm{prob}(R', s)$. Distinguiremos tres casos:

- $s = \epsilon$. En este caso tenemos $\mathrm{prob}(a; R, \epsilon) = 1 \leq 1 = \mathrm{prob}(a; R', \epsilon)$.

- $s = \langle b \rangle \circ s' \ \wedge \ b \neq a$. En tal caso, dado que las trazas de ambos conjuntos o empiezan por a, o son la traza vacía, tenemos

$$\text{prob}(a; R, s) = 0 \leq 0 = \text{prob}(a; R', s)$$

- $s = \langle a \rangle \circ s'$. Tenemos $\text{prob}(a; R, s) = \text{prob}(R, s') \leq \text{prob}(R', s') = \text{prob}(a; R', s)$.

Continuidad.

Sea $\{R_n\}_{n \in \mathbb{N}}$ una cadena de elementos de $\mathbf{TRA}_{\text{Act}}$. Debido a la monotonía del operador prefijo, tenemos que $\{a; R_n\}_{n \in \mathbb{N}}$ también es una cadena. Hemos de probar que $\forall s \in Act^*$, se cumple

$$\text{prob}(a; \sqcup\{R_n\}_{n \in \mathbb{N}}, s) = \text{prob}(\sqcup\{a; R_n\}_{n \in \mathbb{N}}, s)$$

lo cual se deduce de la siguiente cadena de igualdades

$$\text{prob}(a; \sqcup\{R_n\}_{n \in \mathbb{N}}, s) =$$

$$\begin{cases} 1 & \text{si } s = \epsilon \\ \text{prob}(\sqcup\{R_n\}_{n \in \mathbb{N}}, s') & \text{si } s = \langle a \rangle \circ s' \\ 0 & \text{e.o.c.} \end{cases} = \begin{cases} 1 & \text{si } s = \epsilon \\ \lim\limits_{n \in \mathbb{N}} \text{prob}(R_n, s') & \text{si } s = \langle a \rangle \circ s' \\ 0 & \text{e.o.c.} \end{cases} =$$

$$\lim_{n \in \mathbb{N}} \begin{cases} 1 & \text{si } s = \epsilon \\ \text{prob}(R_n, s') & \text{si } s = \langle a \rangle \circ s' \\ 0 & \text{e.o.c.} \end{cases} = \lim_{n \in \mathbb{N}} \text{prob}(a; R_n, s) = \text{prob}(\sqcup\{a; R_n\}_{n \in \mathbb{N}}, s)$$

□

Elección Interna

Dado un valor $p \in (0, 1)$, la función $\_ \oplus_p \_ :: \mathbf{TRA}_{\text{Act}} \times \mathbf{TRA}_{\text{Act}} \longrightarrow \mathbf{TRA}_{\text{Act}}$ devuelve el conjunto de trazas, unión de las trazas correspondientes a sus argumentos, ponderadas con respecto al factor p. Si una traza pertenece a los dos conjuntos, sumaremos las probabilidades correspondientes. En suma

$$R_1 \oplus_p R_2 = \{(s, q) \mid q = p \cdot \text{prob}(R_1, s) + (1 - p) \cdot \text{prob}(R_2, s) \ \wedge \ q > 0\}$$

Proposición 3.16 Las funciones $\_ \oplus_p \_ :: \mathbf{TRA}_{\mathrm{Act}} \times \mathbf{TRA}_{\mathrm{Act}} \longrightarrow \mathbf{TRA}_{\mathrm{Act}}$ son monótonas y continuas en sus dos argumentos, para cada $p \in (0,1)$.

Demostración: Por simetría es suficiente hacer la demostración para uno de los argumentos. En nuestro caso, la haremos para el primero.

Monotonía.

Sean $R_1, R_2 \in \mathbf{TRA}_{\mathrm{Act}}$ tales que $R_1 \sqsubseteq_{\mathbf{TRA}} R_2$. Tenemos que probar que se cumple $R_1 \oplus_p R \sqsubseteq_{\mathbf{TRA}} R_2 \oplus_p R$, para cualquier $R \in \mathbf{TRA}_{\mathrm{Act}}$, o lo que es lo mismo, que $\forall s \in Act^* :\ \mathrm{prob}(R_1 \oplus_p R, s) \leq \mathrm{prob}(R_1 \oplus_p R, s)$. Pero esto es trivial, dado que

$$\begin{aligned} \mathrm{prob}(R_1 \oplus_p R, s) &= p \cdot \mathrm{prob}(R_1, s) + (1-p) \cdot \mathrm{prob}(R, s) \\ &\leq p \cdot \mathrm{prob}(R_2, s) + (1-p) \cdot \mathrm{prob}(R, s) \\ &= \mathrm{prob}(R_2 \oplus_p R, s) \end{aligned}$$

Continuidad.

Sea $\{R_n\}_{n \in \mathbb{N}}$ una cadena de elementos de $\mathbf{TRA}_{\mathrm{Act}}$. Por la monotonía de la elección interna, tenemos que $\{R_n \oplus_p R\}_{n \in \mathbb{N}}$ también es una cadena. Tenemos que probar que $\forall s \in Act^*$, se cumple

$$\mathrm{prob}(\sqcup\{R_n\}_{n \in \mathbb{N}} \oplus_p R, s) = \mathrm{prob}(\sqcup\{R_n \oplus_p R\}_{n \in \mathbb{N}}, s)$$

lo cual se sigue de la siguiente cadena de igualdades

$$\begin{aligned} \mathrm{prob}(\sqcup\{R_n\}_{n \in \mathbb{N}} \oplus_p R, s) &= p \cdot \mathrm{prob}(\sqcup\{R_n\}_{n \in \mathbb{N}}, s) + (1-p) \cdot \mathrm{prob}(R, s) \\ &= p \cdot (\lim_{n \in \mathbb{N}} \mathrm{prob}(R_n, s)) + (1-p) \cdot \mathrm{prob}(R, s) \\ &= \lim_{n \in \mathbb{N}} (p \cdot \mathrm{prob}(R_n, s) + (1-p) \cdot \mathrm{prob}(R, s)) \\ &= \lim_{n \in \mathbb{N}} \mathrm{prob}(R_n \oplus_p R, s) = \mathrm{prob}(\sqcup\{R_n \oplus_p R\}_{n \in \mathbb{N}}, s) \end{aligned}$$

□

Recursión

Como es usual, cuando se define una semántica denotacional, el significado de los procesos recursivos definidos por expresiones de la forma $recX.P(X)$ se obtiene como el límite de las aproximaciones finitas de la forma

$$P_0 = \Omega, P_1 = P(\Omega), \ldots, P_n = P^n(\Omega)$$

Dado que todos los operadores incluidos en el lenguaje son continuos, éste límite es el menor punto fijo de la ecuación $X = P(X)$. Es decir, definimos

$$[\![recX.\ P(X)]\!] = \bigsqcup_{n=0}^{\infty} [\![P_n]\!]$$

El siguiente lema relaciona la semántica operacional de los procesos recursivos con la de sus aproximaciones finitas. Su demostración es sencilla por inducción sobre el número de veces que se aplica la regla (REC) para desplegar las recursiones a lo largo del cómputo.

Lema 3.17 Sea $P = recX.P(X)$. Entonces, para toda secuencia de acciones s, y para todo $p \in [0,1]$ se verifica que $P \xrightarrow{s}_p P'$ sii $\exists\ n \in \mathbb{N}^+ :\ P_n \xrightarrow{s}_p P_n''$. □

En el lema anterior la correspondencia entre transiciones es uno a uno, en el sentido de que la misma se mantendría con la formalización de la semántica utilizando multiconjuntos de transiciones.

La estrecha relación entre la semántica denotacional de un proceso $P \in \mathrm{PPA}'$ y su funcionamiento operacional, descrito mediante $trazas(P)$, queda reflejada en el siguiente

Lema 3.18 Para cada proceso finito (sin recursión) $P \in \mathrm{PPA}'$, y para toda secuencia de acciones s se verifica:

$$\mathrm{prob}([\![P]\!], s) = \sum \{\!|\ q \mid P \xrightarrow{s}_q\ |\!\}$$

Demostración: La demostración la realizaremos por inducción estructural.

- $P = Nil$. Por definición obtenemos $\mathrm{prob}([\![Nil]\!], \epsilon) = 1$, y $\mathrm{prob}([\![Nil]\!], s) = 0$ si $s \neq \epsilon$; por otra parte, $Nil \xrightarrow{\epsilon}_1 Nil$, y $Nil \xrightarrow{s}_0$ para toda $s \neq \epsilon$, con lo que el resultado se verifica.

- $P = \Omega$. Para cualquier secuencia s, por definición tenemos $\mathrm{prob}([\![\Omega]\!], s) = 0$, mientras que $\Omega \xrightarrow{s}_0$, con lo que el resultado se verifica.

- $P = a; P'$. En primer lugar tenemos $\text{prob}([\![P]\!], \epsilon) = 1$ y $\sum \{\!| \, q \mid P \xrightarrow{\epsilon}_q \, |\!\} = 1$.

  Consideremos el caso de que s no sea una secuencia vacía. Por hipótesis de inducción obtenemos

  $$\text{prob}([\![P']\!], s') = \sum \{\!| \, q \mid P' \xrightarrow{s'}_q \, |\!\}$$

  Si s no empieza por a, entonces $\text{prob}([\![P]\!], s) = 0$, mientras que $P \xrightarrow{s}_0$, con lo que no existe p tal que $(s, p) \in \mathit{trazas}(P)$.

  Si $s = \langle a \rangle \circ s'$, entonces $\text{prob}([\![P]\!], s) = \text{prob}([\![P']\!], s')$. Por hipótesis de inducción obtenemos $\text{prob}([\![P']\!], s') = \sum \{\!| \, q \mid P' \xrightarrow{s'}_q \, |\!\}$, y dado que $P \xrightarrow{s}_q P''$ sii $P' \xrightarrow{s'}_q P''$ obtenemos

  $$\sum \{\!| \, q \mid P' \xrightarrow{s'}_q \, |\!\} = \sum \{\!| \, q \mid P \xrightarrow{s}_q \, |\!\}$$

  de lo cual obtenemos el resultado.

- $P = P_1 \oplus_q P_2$. Por hipótesis de inducción

  $$\text{prob}([\![P_1]\!], s') = \sum \{\!| \, q \mid P_1 \xrightarrow{s'}_q \, |\!\}$$

  $$\text{prob}([\![P_2]\!], s') = \sum \{\!| \, q \mid P_2 \xrightarrow{s'}_q \, |\!\}$$

  Por definición, $\text{prob}([\![P]\!], s) = q \cdot \text{prob}([\![P_1]\!], s) + (1 - q) \cdot \text{prob}([\![P_2]\!], s)$. Por otra parte, $P_1 \xrightarrow{s}_{q_1} P'$ sii $P \xrightarrow{s}_{q \cdot q_1} P'$. De igual forma tenemos $P_2 \xrightarrow{s}_{q_1} P'$ sii $P \xrightarrow{s}_{(1-q) \cdot q_1} P'$, con lo que, aplicando hipótesis de inducción, obtenemos inmediatamente que se verifica el resultado.

□

Teorema 3.3 Para cada proceso $P \in \text{PPA}'$, y para toda secuencia de acciones s se verifica:

$$\text{prob}([\![P]\!], s) = \sum \{\!| \, q \mid P \xrightarrow{s}_q \, |\!\}$$

Demostración: Si P es finito el resultado es cierto en virtud del lema anterior. Consideremos entonces $P = \mathit{rec}X.P'(X)$. Para simplificar la demostración, supongamos que $P'(X)$ es finito (es decir, que no contiene apariciones del operador de recursión).

Por definición tenemos

$$\text{prob}([\![P]\!], s) = \lim_{n \in \mathbb{N}} \text{prob}([\![P_n]\!], s) \tag{3.3}$$

Como quiera que cada P_n es finito tenemos

$$\text{prob}([\![P_n]\!], s) = \sum \{\!| \, q \mid P_n \xrightarrow{s}_q \, |\!\}$$

de lo que se deduce

$$\lim_{n \in \mathbb{N}} \sum \{\!| \, q \mid P_n \xrightarrow{s}_q \, |\!\} = \lim_{n \in \mathbb{N}} \text{prob}([\![P_n]\!], s) \tag{3.4}$$

Nos queda entonces por ver que este último valor coincide con $\sum \{\!| \, q \mid P \xrightarrow{s}_q \, |\!\}$, o lo que es lo mismo que la semántica operacional es continua en un cierto sentido.

Aplicando el Lema 3.17 de derecha a izquierda obtenemos

$$\lim_{n \in \mathbb{N}} \sum \{\!| \, q \mid P_n \xrightarrow{s}_q \, |\!\} \leq \sum \{\!| \, q \mid P \xrightarrow{s}_q \, |\!\} \tag{3.5}$$

Para la otra desigualdad, aplicando la definición de sumatorio, tenemos que $\forall\ \delta > 0$ existe un subconjunto finito de computaciones de la forma $P \xrightarrow{s}_q$, al que llamaremos F_δ, de modo que

$$\sum_{F_\delta} \{\!| \, q \mid P \xrightarrow{s}_q \, |\!\} > \sum \{\!| \, q \mid P \xrightarrow{s}_q \, |\!\} - \delta$$

Dado que F_δ es un conjunto finito de computaciones de la forma $P \xrightarrow{s}_q$, aplicando el Lema 3.17 de izquierda a derecha, obtenemos que debe existir un n tal que las computaciones que pertenecen a F_δ se corresponden con parte de áquellas de la forma $P_n \xrightarrow{s}_q$, por lo que

$$\sum_{F_\delta} \{\!| \, q \mid P \xrightarrow{s}_q \, |\!\} \leq \sum \{\!| \, q \mid P_n \xrightarrow{s}_q \, |\!\}$$

y tomando el límite en n obtenemos

$$\sum \{\!| \, q \mid P \xrightarrow{s}_q \, |\!\} - \delta < \lim_{n \in \mathbb{N}} \sum \{\!| \, q \mid P_n \xrightarrow{s}_q \, |\!\}$$

por lo que haciendo tender δ a cero concluimos

$$\sum \{\!| \, q \mid P \xrightarrow{s}_q \, |\!\} \leq \lim_{n \in \mathbb{N}} \sum \{\!| \, q \mid P_n \xrightarrow{s}_q \, |\!\} \tag{3.6}$$

Finalmente juntando las desigualdades (3.5) y (3.6) obtenemos

$$\sum \{\!|\, q \mid P \xrightarrow{s}_q \,|\!\} = \lim_{n\in\mathbb{N}} \sum \{\!|\, q \mid P_n \xrightarrow{s}_q \,|\!\} \tag{3.7}$$

y en definitiva, conjuntando las ecuaciones (3.3), (3.4) y (3.7), llegamos al resultado deseado

$$\text{prob}([\![P]\!], s) = \sum \{\!|\, q \mid P \xrightarrow{s}_q \,|\!\}$$

El caso general en el que pueden aparecer recursiones dentro de P' se resuelve de forma similar desplegando todas las recursiones al mismo tiempo. □

A partir de este teorema derivamos inmediatamente la equivalencia entre la semántica denotacional y la caraterización alternativa descrita en la Sección 3.2.

Corolario 3.4 Sean $P, Q \in \text{PPA}'$. $[\![P]\!] =_{\text{TRA}} [\![Q]\!]$ sii $\mathit{trazas}(P) \equiv_{\text{trazas}} \mathit{trazas}(Q)$. □

Corolario 3.5 (Abstracción Completa de TRA_{Act})
Sean $P, Q \in \text{PPA}'$. Entonces, $P \approx_{\mathcal{R}} Q$ sii $[\![P]\!] =_{\text{TRA}} [\![Q]\!]$.

Demostración: Inmediata a partir del Teorema 3.1 y del Corolario 3.4. □

Capítulo 4

Semántica de Pruebas para el Modelo Generativo

En este capítulo daremos una interpretación del modelo *generativo* [vGSST90] utilizando una semántica de pruebas, siguiendo los pasos que se dieron en [NdFL95] para nuestra álgebra de procesos PPA.

Como en el capítulo anterior, comenzaremos definiendo una semántica de pruebas para el modelo generativo. Después pasaremos a dar una caracterización alternativa de dicha semántica de pruebas, basándonos en una extensión probabilística de los *conjuntos de aceptación* [Hen88]. Dichos *conjuntos de aceptación probabilísticos* [NdFL95] aunque basados en los correspondientes para el caso no probabilístico, tienen diferencias sustanciales con ellos, más allá de lo que supondría la mera introducción de las probabilidades. Dichas diferencias serán comentadas y justificadas debidamente. Al demostrar que la caracterización alternativa es en efecto equivalente a la semántica de pruebas, es decir, identifica a los mismos procesos sintácticos, obtendremos como resultado colateral un conjunto de pruebas *esenciales* similar al que se obtiene en [YCDS94], si interpretamos sus *trazas probabilísticas*[1] como pruebas probabilísticas.

[1] Aunque el nombre coincide con el que nosotros utilizamos en el Capítulo 3, estas trazas probabilísticas no tienen nada que ver con las nuestras, siendo similares a lo que nosotros denominamos *barbas probabilísticas* en [NdFL95].

Finalizaremos el capítulo definiendo una semántica denotacional completamente abstracta con respecto a la semántica de pruebas para el modelo generativo. Esta semántica estará basada en una extensión probabilística de los *árboles de aceptación* [Hen85]. Además, probaremos que esta semántica denotacional es también equivalente a la caracterización alternativa basada en conjuntos de aceptación probabilísticos que se presentó anteriormente.

4.1 Definición del Modelo Generativo

La interpretación intuitiva del modelo generativo nos dice que el entorno, es decir las pruebas, puede ofrecer más de una acción en cada momento, pudiendo ser además distintas las probabilidades con las que se ofrecen estas acciones. Usando la terminología introducida en [Mil80], podemos decir que *varios botones se pueden pulsar a la vez y con distinta fuerza.* Ello da lugar a que al interpretar este modelo en el ámbito de una semántica de pruebas, las pruebas que debamos considerar sean la totalidad de las consideradas en la Definición 2.10 [2].

Definición 4.1 El conjunto de las *pruebas generativas*, que notaremos por $\mathcal{G}$, se define mediante la expresión BNF

$$T ::= \mathit{Nil} \mid \Omega \mid X \mid a;T \mid T \oplus_p T \mid T +_p T \mid \mathit{rec}X.T$$

donde $p \in (0,1)$, $a \in \mathit{Act} \cup \omega$, y $X \in \mathit{Id}$.

Escribiremos $P \approx_{\mathcal{G}} Q$ sii $\forall T \in \mathcal{G} : \mathit{pass}(P\ ,\ T) = \mathit{pass}(Q\ ,\ T)$. □

4.2 Conjuntos de Aceptación Probabilísticos

En esta sección presentamos una caracterización alternativa de la semántica de pruebas definida en la sección anterior (i.e. $\approx_{\mathcal{G}}$). Al efecto utilizaremos una versión de

[2] Como ya hemos indicado, podríamos reducir el conjunto de pruebas eliminando el no determinismo (utilizando el Lema 2.14) y también las pruebas recursivas, pues si dos procesos son distintos, se les puede distinguir en *tiempo* finito, es decir, usando una prueba finita.

los *conjuntos de aceptación* [Hen88], utilizados allí para dar una caracterización alternativa de la semántica de pruebas *must* en el caso no probabilístico, en los que introduciremos probabilidades para capturar la información inducida por las pruebas probabilísticas.

Los cambios principales con respecto al caso no probabilístico son los siguientes:

- Los estados no serán conjuntos de acciones, sino que serán conjuntos de pares (acción, probabilidad) (ver Ejemplo 4.2).
- En el caso no probabilístico, los conjuntos de aceptación se definen como los estados alcanzables después de que el proceso ejecute una secuencia de acciones. En el marco probabilístico, las secuencias de acciones no son suficientes para determinar unívocamente la continuación subsiguiente. En su lugar, tendremos que utilizar secuencias de pares (estado, acción) donde cada acción pertenece a las contenidas en el estado asociado (ver Ejemplo 4.3).
- La definición de la equivalencia entre procesos debe ser modificada, teniendo en cuenta la información probabilística que aparece en los conjuntos de aceptación probabilísticos. Además, no será necesario utilizar ninguna noción de cierre (cierre bajo unión o cierre convexo) como sucedía en el caso no probabilístico[3] (ver Ejemplo 4.4).

El ejemplo que sigue ilustra la primera de las modificaciones con respecto al caso no probabilístico.

Ejemplo 4.2 Consideremos los procesos $P = a +_{\frac{1}{2}} b$ y $P' = a +_{\frac{1}{3}} b$. En su primer paso, ambos procesos tienen como único estado alcanzable aquél que contiene a las acciones a y b. Si no introducimos ninguna información probabilística en el estado, los procesos serían equivalentes bajo la semántica de conjuntos de aceptación. Sin embargo, los mismos no son equivalentes respecto de la semántica de pruebas para el

[3] Aunque en [Hen88] no se utilizaba de forma explícita ninguna noción de cierre a la hora de definir la equivalencia entre conjuntos de aceptación, la misma quedaba enmascarada bajo la definición de CC.

modelo generativo, pues, por ejemplo, la prueba $T = (a;\omega) +_{\frac{1}{3}} (b; Nil)$ permite distinguirlos. En consecuencia, bajo nuestra definición de conjuntos de aceptación probabilísticos el primer proceso tendrá como estado alcanzable el estado $\{(a, \frac{1}{2}), (b, \frac{1}{2})\}$, mientras que el segundo tendrá como estado alcanzable el estado $\{(a, \frac{1}{3}), (b, \frac{2}{3})\}$. □

En el caso no probabilístico, los estados, es decir, los diferentes conjuntos contenidos en los conjuntos de aceptación, quedan descritos por una secuencia de acciones, dado que las *continuaciones* del proceso después de ejecutar una acción se unifican. En otras palabras, una vez que un proceso ejecuta una acción, no hay posibilidad de distinguir desde que estado se ejecutó la misma, por lo tanto todas las posibles continuaciones de cada proceso después de ejecutar una misma acción se tienen que unir, dando lugar a una continuación única para cada acción. Sin embargo, lo mismo no sucede en el caso probabilístico; en el mismo, podemos distinguir vía pruebas desde que estado se ejecutó la acción, y por tanto las continuaciones no se pueden unir. Ello queda ilustrado con el siguiente

Ejemplo 4.3 Consideremos el proceso no probabilístico $P = (a; d) \oplus ((a; b) + c)$. Tenemos que P es equivalente al proceso $P' = (a; (d \oplus b)) \oplus ((a; (d \oplus b)) + c)$ con respecto a la equivalencia inducida por los conjuntos de aceptación, o lo que es lo mismo respecto de la semántica de pruebas. Vemos pues como las continuaciones después de que el proceso P ejecute la acción a pueden unirse sin variar la semántica del proceso.

Pero como hemos indicado más arriba, lo mismo no ocurre en el caso probabilístico. Si consideramos el proceso $P = (a; d) \oplus_{\frac{1}{2}} ((a; b) +_{\frac{1}{2}} c)$, en caso de existir dos procesos R_1, R_2 tales que P es equivalente al proceso $P' = a; R_1 \oplus_{\frac{1}{2}} ((a; R_1) +_{\frac{1}{2}} c; R_2)$ (lo cual supondría que se mantienen los mismos estados iniciales, pero las continuaciones tras la acción a se unen) con respecto a la semántica de pruebas del modelo generativo, si consideramos la prueba $T = (a; b; \omega) +_{\frac{1}{3}} c$, obtenemos $pass(P\ ,\ T) = \frac{1}{6}$. Por su parte, la probabilidad con la cual el proceso P' pasa la prueba T viene dada por

$$pass(P'\ ,\ T) = \frac{1}{2} \cdot pass(R_1\ ,\ (b;\omega)) + \frac{1}{2} \cdot \frac{1}{3} \cdot pass(R_1\ ,\ (b;\omega))$$

Como hemos supuesto $P \approx_{\mathcal{G}} P'$, deberíamos tener $pass(P' , T) = \frac{1}{6}$, de lo cual se deduce $pass(R_1 , (b;\omega)) = \frac{1}{4}$.

Pero si consideramos ahora la prueba $T' = a;b;\omega$, tenemos $pass(P , T') = \frac{1}{2}$, mientras

$$pass(P' , T') = \frac{1}{2} \cdot pass(R_1 , (b;\omega)) + \frac{1}{2} \cdot pass(R_1 , (b;\omega))$$

y por ser $P \approx_{\mathcal{G}} P'$, obtendríamos $pass(P' , T') = \frac{1}{2}$, de lo cual se deduce $pass(R_1 , (b;\omega)) = \frac{1}{2}$, lo cual se contradice con nuestro resultado anterior. Queda probado por tanto que no puede existir una continuación única R_1 con las características solicitadas. □

En general, las distintas continuaciones tras la misma acción correspondientes a diferentes estados pueden distinguirse por lo que deberemos incluir los estados en las secuencias que definen los conjuntos de aceptación probabilísticos de un proceso.

El siguiente ejemplo muestra la tercera de las diferencias con respecto al caso no probabilístico.

Ejemplo 4.4 Sea P el proceso no probabilístico $a \oplus b$. Si utilizamos la notación introducida en [Hen88] para definir los conjuntos de aceptación de un proceso después de ejecutar cada traza, obtenemos $\mathcal{A}(P, \epsilon) = \{\{a\}, \{b\}\}$, $\mathcal{A}(P, \langle a \rangle) = \mathcal{A}(P, \langle b \rangle) = \{\emptyset\}$ mientras que para cualquier otra traza s, $\mathcal{A}(P, s) = \emptyset$. Este proceso es equivalente respecto a la semántica definida mediante los conjuntos de aceptación al proceso $Q = (a \oplus b) \oplus (a + b)$, pues en general podemos cerrar bajo unión los conjuntos de aceptación.

Si consideramos la versión probabilística de P, $P' = a \oplus_p b$, y suponemos que $p_1, p_2, r > 0$ son tales que P' es equivalente respecto a la semántica de pruebas para el modelo generativo al proceso $Q' = (a \oplus_{p_1} b) \oplus_{p_2} (a +_r b)$, al considerar las pruebas $T_1 = (a;\omega) +_{\frac{1}{2}} (b; Nil)$, $T_2 = (a;\omega) +_{\frac{1}{3}} (b; Nil)$, llegamos respectivamente a

$$\begin{aligned} p &= p_2 \cdot p_1 + (1 - p_2) \cdot r \quad , \text{y} \\ p &= p_2 \cdot p_1 + (1 - p_2) \cdot \tfrac{r}{2-r} \end{aligned}$$

de lo cual se deduce que r sería igual a 0 o igual a 1, ambos valores no válidos[4], con lo que queda probada la imposibilidad de construir un proceso Q tal equivalente a P.

Bajo la semántica de conjuntos de aceptación probabilísticos, P' tendrá como estados alcanzables $\{(a,1)\}$, con probabilidad p, y $\{(b,1)\}$, con probabilidad $1-p$, mientras que ningún estado $\{(a,s),(b,1-s)\}$ es alcanzable en absoluto. En cambio para Q', el estado $\{(a,r),(b,1-r)\}$ es alcanzable con probabilidad $1-p_2$. □

Una vez explicadas las modificaciones con respecto al caso no probabilístico, daremos una serie de definiciones previas que se precisan para definir los conjuntos de aceptación probabilísticos de un proceso después de ejecutar una secuencia de pares (estado, acción). Introduciremos la noción de *estado probabilístico*, la de estado alcanzable (inmediatamente) por un proceso, y finalmente definiremos una relación $P \stackrel{s}{\Longrightarrow}_p P'$ que puede interpretarse como que el proceso P puede evolucionar al proceso P', con probabilidad p, tras ejecutar la serie de acciones indicada en s, atravesando una serie de estados también indicados en dicha secuencia.

Definición 4.5 Sea $A \subseteq Act \times (0,1]$. Definimos el *multiconjunto de acciones* de A como $Act(A) = \{\!|\, a \mid \exists p :\ (a,p) \in A \,|\!\}$. Diremos que A es un *estado* (*probabilístico*) si toda acción $a \in Act$ aparece a lo más una vez en $Act(A)$ (i.e. $Act(A)$ es un conjunto), y se cumple que o bien $\sum \{\!|\, p \mid (a,p) \in A \,|\!\}$ es igual a 1, o bien es igual a 0, o sea $A = \emptyset$.

Dado un estado probabilístico A, definimos la *probabilidad* de a en A, que denotaremos por $pro(a,A)$, por medio de

$$pro(a,A) = \begin{cases} p & \text{si } (a,p) \in A \\ 0 & \text{e.o.c.} \end{cases}$$

Dado un proceso estable P, definimos su *estado probabilístico* (inmediatamente)

[4]En un contexto de prioridad, tales valores extremos serían válidos, pero en nuestro marco de trabajo lo que indicarían dichos valores es que la probabilidad con la que se alcanza el estado en el que aparece dicha elección externa (i.e. $1-p_2$ en nuestro ejemplo) debería ir toda bien a la acción a ó a la acción b, en el caso de que el entorno ofreciera simultáneamente ambas acciones.

alcanzable, que denotaremos por $S(P)$, como el conjunto

$$S(P) = \{(a,p) \mid p = \sum_R \{\!|\, p_i \mid P \xrightarrow{a}_{p_i} R \,|\!\} \wedge p > 0\}$$

□

Usualmente, cuando no haya lugar a confusión, omitiremos el calificativo *probabilístico* al referirnos a estados probabilísticos. En la definición anterior, nótese que sólo definimos estados alcanzables inmediatamente para procesos estables, lo cual resultará suficiente pues para procesos no estables su conjunto de estados alcanzables se definirá a partir de los estados alcanzables por los procesos estables a los cuales el proceso puede evolucionar tras ejecutar una transición interna generalizada. Nótese además que para procesos estables queda justificado el uso del singular, pues tales procesos se encuentran en un (y sólo un) estado.

Definición 4.6 Sean $A_1, \ldots, A_n$ estados ($A_i \neq \emptyset$), y $a_1, \ldots, a_n \in Act$, tales que $a_i \in Act(A_i)$. Dados la secuencia $s = \langle A_1\, a_1, A_2\, a_2, \ldots, A_n\, a_n \rangle$ y $0 < p \leq 1$, definimos de forma inductiva la relación $P \stackrel{s}{\Longrightarrow}_p P'$ en la forma:

$P \stackrel{\epsilon}{\Longrightarrow}_p P'$ sii $P \rightarrowtail^*_p P'$

$P \stackrel{s}{\Longrightarrow}_p P'$ sii $\exists Q_1, P_1, p_1, q_1 \colon\ P \rightarrowtail^*_{p_1} Q_1 \xrightarrow{a_1}_{q_1} P_1 \stackrel{s'}{\Longrightarrow}_{p'} P' \wedge S(Q_1) = A_1 \wedge p = \frac{p' \cdot p_1 \cdot q_1}{r_1}$

donde $s' = \langle A_2\, a_2, \ldots, A_n\, a_n \rangle$ y $r_1 = pro(a_1, A_1)$. Escribiremos $P \stackrel{s}{\not\Longrightarrow}$ si no existen P' y $p > 0$ tales que $P \stackrel{s}{\Longrightarrow}_p P'$. □

Como en el caso de las otras relaciones ya definidas ($\rightarrowtail_p$, $\xrightarrow{a}_p$ y $\rightarrowtail^*_p$) debemos tener cuidado con las repeticiones que se podrían producir a la hora de generar las transiciones de la forma $\stackrel{s}{\Longrightarrow}_p$. Intuitivamente, $P \stackrel{s}{\Longrightarrow}_p P'$ sii P puede ejecutar sucesivamente las acciones a_i atravesando estados correspondientes a una serie de procesos estables Q_i con $S(Q_i) = A_i$, y finalmente evoluciona a P' mediante una transición interna generalizada. El valor p se calcula a partir de las probabilidades con las que se alcanzan los procesos estables Q_i (mediante una transición interna generalizada), a partir de las probabilidades *relativas* de ejecutar las acciones a_i (i.e. los valores $\frac{q_i}{pro(a_i, A_i)}$, que denotan la probabilidad de ejecutar la transición correspondiente, partido por la probabilidad de ejecutar la acción a_i a partir de ese punto), y a

partir de la probabilidad asociada con la última transición interna generalizada que conduce al proceso P'.

Y con todo ello quedamos en condiciones de definir los conjuntos de aceptación probabilísticos de un proceso tras ejecutar una secuencia de pares (estado, acción).

Definición 4.7 Sea P un proceso y s una secuencia de pares (estado, acción). Definimos los *conjuntos de aceptación (probabilísticos)* de P *tras* s como

$$\mathcal{A}(P,s) = \{(A, p_A/q_s) \mid p_A = \textstyle\sum_{P'} \{\!|\, p_i \mid P \stackrel{s}{\Longrightarrow}_{p_i} P' \wedge S(P') = A \,|\!\} \wedge p_A > 0\}$$

donde $q_\epsilon = 1$ y $q_{s'\circ(B\,b)} = \sum_{Q'} \{\!|\, q_i \mid P \stackrel{s'}{\Longrightarrow}_{q_i} Q' \wedge S(Q') = B \,|\!\}$. □

Cuando no haya lugar a confusión, omitiremos el término probabilístico al hablar de los correspondientes conjuntos de aceptación. Intuitivamente, para calcular los conjuntos de aceptación de un proceso P después de ejecutar una secuencia s, primero calculamos los estados alcanzables por P después de ejecutar la secuencia s, y tras ello sumamos las probabilidades de alcanzar dichos estados (cada estado puede ser alcanzable a través de distintos caminos), dividiendo por la probabilidad de que el proceso alcance el último estado de la secuencia s, tras ejecutar las acciones de la misma. Con ello conseguimos que, en cada punto, la suma de las probabilidades de los estados alcanzables sea igual a 1 menos la probabilidad de que el proceso diverja tras ejecutar la secuencia s. Nótese que bajo la definición dada, un proceso divergente no tiene estados alcanzables después de ninguna secuencia, es decir se cumple:

$$\mathcal{A}(\Omega, s) = \emptyset \qquad \text{para cualquier secuencia } s$$

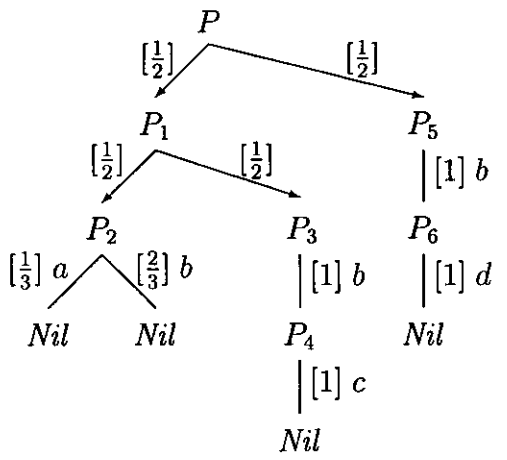
dado que para ningún proceso P y ningún $0 < p \leq 1$ se tiene $\Omega \stackrel{s}{\Longrightarrow}_p P$.

Definición 4.8 (Caracterización Alternativa)

Siendo P, P' procesos, escribiremos $P \cong P'$ si para toda secuencia s de la forma $s = \langle A_1\, a_1, A_2\, a_2, \ldots, A_n\, a_n \rangle$ se verifica:

$$\mathcal{A}(P,s) = \mathcal{A}(P',s)$$

□



$P_1 = (a +_{\frac{1}{3}} b) \oplus_{\frac{1}{2}} (b; c)$
$P_2 = a +_{\frac{1}{3}} b$
$P_3 = b; c$
$P_4 = c$
$P_5 = b; d$
$P_6 = d$

Figura 4.1: Semántica operacional de $P = ((a +_{\frac{1}{3}} b) \oplus_{\frac{1}{2}} (b; c)) \oplus_{\frac{1}{2}} (b; d)$.

A continuación veremos una serie de ejemplos que ilustran cómo se calculan los conjuntos de aceptación tras una secuencia, y que muestran cómo quedan relacionados algunos procesos por medio de la nueva relación $\cong$.

Ejemplo 4.9 Consideremos el proceso $P = ((a +_{\frac{1}{3}} b) \oplus_{\frac{1}{2}} (b; c)) \oplus_{\frac{1}{2}} (b; d)$, cuya semántica operacional aparece representada en la Figura 4.1. Para calcular los conjuntos de aceptación del proceso P tras una secuencia s, calcularemos primero los procesos a los cuáles puede llegar P tras ejecutar la secuencia s.

Comenzando por $\mathcal{A}(P, \epsilon)$, tenemos $P \stackrel{\epsilon}{\Longrightarrow}_{\frac{1}{4}} P_2$, $P \stackrel{\epsilon}{\Longrightarrow}_{\frac{1}{4}} P_3$, $P \stackrel{\epsilon}{\Longrightarrow}_{\frac{1}{2}} P_5$. Además, $S(P_2) = A = \{(a, \frac{1}{3}), (b, \frac{2}{3})\}$, mientras que $S(P_3) = S(P_5) = B = \{(b, 1)\}$, y dado que $q_\epsilon = 1$, tenemos

$$\mathcal{A}(P, \epsilon) = \{\ (A, 1/4), (B, 3/4)\ \}$$

Pasemos ahora a calcular $\mathcal{A}(P, \langle A\, a\rangle)$. Tenemos $P \stackrel{\langle A\, a\rangle}{\Longrightarrow}_{\frac{1}{4}} Nil$. Además, $S(Nil) = \emptyset$, y dado que $q_{\langle A\, a\rangle} = \sum_{Q'} \{\!|\ q_i \mid P \stackrel{\epsilon}{\Longrightarrow}_{q_i} Q' \ \wedge\ S(Q') = A\ |\!\} = \frac{1}{4}$, obtenemos

$$\mathcal{A}(P, \langle A\, a\rangle) = \{\ (\emptyset, 1)\ \}$$

De forma similar se obtiene

$$\mathcal{A}(P, \langle A\, b\rangle) = \{\ (\emptyset, 1)\ \}$$

A la hora de calcular $\mathcal{A}(P, \langle B\, b\rangle)$, tenemos por un lado $P \stackrel{\langle B\, b\rangle}{\Longrightarrow}_{\frac{1}{4}} P_4$ mientras que por otro, $P \stackrel{\langle B\, b\rangle}{\Longrightarrow}_{\frac{1}{2}} P_6$. Dado que $S(P_4) = \{(c,1)\}$ y $S(P_6) = \{(d,1)\}$, y teniendo en cuenta el hecho de que $q_{\langle B\, b\rangle} = \sum_{Q'} \{\!|\, q_i \mid P \stackrel{\epsilon}{\Longrightarrow}_{q_i} Q' \wedge S(Q') = B \,|\!\} = \frac{3}{4}$, obtenemos

$$\mathcal{A}(P, \langle B\, b\rangle) = \{\, (\{(c,1)\}, 1/3), (\{(d,1)\}, 2/3)\,\} \qquad (\tfrac{1/4}{3/4} = \tfrac{1}{3},\ \tfrac{1/2}{3/4} = \tfrac{2}{3})$$

De forma similar a como obtuvimos $\mathcal{A}(P, \langle A\, a\rangle)$, obtenemos

$$\mathcal{A}(P, \langle B\, b, \{(c,1)\}\, c\rangle) = \{\, (\emptyset, 1)\,\}$$
$$\mathcal{A}(P, \langle B\, b, \{(d,1)\}\, d\rangle) = \{\, (\emptyset, 1)\,\}$$

Finalmente, para el resto de las secuencias s, se tiene

$$\mathcal{A}(P, s) = \emptyset$$

Podemos comprobar entonces que, por ejemplo, el proceso P es equivalente al proceso $P' = (a +_{\frac{1}{3}} b) \oplus_{\frac{1}{4}} (b; (c \oplus_{\frac{1}{3}} d))$, es decir $P \cong P'$. De hecho, más adelante tendremos que el proceso P' será la *forma normal* del proceso P. □

Ejemplo 4.10 En el Ejemplo 3.8, mostramos que los procesos $a;\Omega$ y $b;\Omega$ eran equivalentes en el modelo reactivo, si bien comentamos que podrían ser distinguidos por las pruebas de la forma $a +_p \omega$. En consecuencia, la no equivalencia de estos procesos bajo la semántica de pruebas para el modelo generativo debería quedar reflejada en sus conjuntos de aceptación, lo cual en efecto ocurre, dado que

$$\mathcal{A}((a;\Omega), \epsilon) = \{\, (\{(a,1)\}, 1)\,\} \neq \{\, (\{(b,1)\}, 1)\,\} = \mathcal{A}((b;\Omega), \epsilon)$$

En cambio, para el resto de las secuencias, se tiene

$$\mathcal{A}((a;\Omega), s) = \mathcal{A}((b;\Omega), s) = \emptyset \qquad (s \neq \epsilon)$$

□

Ejemplo 4.11 Consideremos el proceso $P = recX.(a \oplus_p X)$. Siguiendo el mismo razonamiento empleado en el Ejemplo 3.8 obtenemos

$$\sum_{P'} \{\!|\, p_i \mid P \stackrel{\epsilon}{\Longrightarrow}_{p_i} P' \wedge S(P') = A = \{(a,1)\} \,|\!\} = 1$$

por lo que

$$\mathcal{A}(P, \epsilon) = \{(A, 1)\}$$

De igual forma, $\sum_{P'} \{\!| p_i \mid P \stackrel{(A\,a)}{\Longrightarrow}_{p_i} P' \wedge S(P') = \emptyset \} \!\} = 1$, con lo cual

$$\mathcal{A}(P, \langle A\,a \rangle) = \{(\emptyset, 1)\}$$

Para el resto de las secuencias obtenemos

$$\mathcal{A}(P, s) = \emptyset \qquad (s \neq \epsilon,\ \langle A\,a \rangle)$$

P resulta así equivalente al proceso $P' = a; Nil$, es decir $P \cong P'$. □

4.3 Teorema de Caracterización

En esta sección probaremos que las dos relaciones dadas por las Definiciones 4.1 y 4.8 coinciden, es decir identifican a los mismos procesos. Para hacerlo comenzaremos por asociar a cada proceso sintáctico lo que denominaremos su *árbol de computaciones*, que aunque relacionadas, no deben confundirse con las *computaciones* introducidas en la Definición 2.12.

En dichos árboles de computaciones se alternan estados *internos* y estados *externos*. De los nodos correspondientes a los estados internos saldrán transiciones internas generalizadas que lleguarán a estados externos, mientras que a partir de los nodos correspondientes a los estados externos saldrán transiciones observables que alcanzarán estados internos. Si asociamos elecciones internas generalizadas a los estados internos (ver Definición 2.6), y elecciones externas generalizadas a los estados externos (ver Definición 2.5), estos *árboles* se pueden ver como procesos sintácticos generalizados, que estarán en *forma normal*. La diferencia con respecto a los procesos ordinarios que constituyen PPA proviene del hecho de que estos nuevos procesos pueden ser (sintácticamente) infinitos, pero resulta trivial extender la definición de la semántica operacional a este tipo de procesos, por lo que también podemos definir una semántica de pruebas para ellos, o más exactamente, la probabilidad con la que cada uno de estos procesos pasa una prueba dada.

Tendremos por tanto una forma de asociar a cada proceso sintáctico otro proceso generalizado que estará *normalizado*, y que será *equivalente* (en un cierto sentido que ya precisaremos) al proceso dado. La definición de dichos procesos normalizados será totalmente operacional, estando basada en los conjuntos de aceptación del proceso tras cada secuencia de pares (estado, acción).

Definición 4.12 Sea P un proceso. Definimos el *proceso normalizado asociado* a P, y lo denotaremos por $\widehat{\mathcal{A}}(P)$, como $\widehat{\mathcal{A}}(P) = \widehat{\mathcal{A}}(P, \epsilon)$, donde

$$\widehat{\mathcal{A}}(P, s) = \bigoplus_{i=1}^{n} [p_i] \sum_{j=1}^{r_i} [p_{i,j}]\, a_{i,j}\,;\, \widehat{\mathcal{A}}(P, s \circ \langle A_i\, a_{i,j}\rangle)$$

siendo $\mathcal{A}(P, s) = \{(A_1, p_1), \ldots, (A_n, p_n)\}$, $A_i = \{(a_{i,1}, p_{i,1}), \ldots, (a_{i,r_i}, p_{i,r_i})\}$ y donde usamos el convenio de que $\sum_{j=1}^{0} P_i$ denota al proceso *Nil*. □

Nótese que la diferencia entre 1 y $\sum p_i$ denotará la probabilidad de divergencia del proceso en cada uno de sus estados internos.

Dada la estrecha relación entre $\widehat{\mathcal{A}}(P)$ y los conjuntos de aceptación de la forma $\mathcal{A}(P, s)$, resulta razonable denominar al proceso $\widehat{\mathcal{A}}(P)$ la *forma normal* del proceso P. El siguiente resultado, cuya prueba resulta trivial a partir de la definición anterior, expresa la noción de equivalencia que existe entre el proceso y su forma normal a la que nos referíamos en la introducción de esta sección.

Lema 4.13 Para cada proceso P se tiene $P \cong \widehat{\mathcal{A}}(P)$. □

Pero podemos ir aún más lejos. De la manera en la que hemos definido las formas normales, se sigue trivialmente que las mismas son únicas dentro de cada clase de equivalencia.

Lema 4.14 $P \cong P' \Leftrightarrow \widehat{\mathcal{A}}(P) = \widehat{\mathcal{A}}(P')$. □

Ejemplo 4.15 El proceso P' que aparecía en el Ejemplo 4.9 está en forma normal, es decir $P' = \widehat{\mathcal{A}}(P')$. El proceso P del Ejemplo 4.9 no está en forma normal, pero naturalmente resulta sencillo comprobar que $\widehat{\mathcal{A}}(P) = P'$. □

El primer paso para demostrar la caracterización alternativa de la semántica de pruebas mediante los conjuntos de aceptación consistirá en probar que un proceso es equivalente con respecto a la semántica de pruebas a su forma normal. De cara a demostrar dicho teorema precisaremos una serie de resultados auxiliares previos.

Lema 4.16 Para cada proceso P, y cada secuencia $s = \langle A_1\, a_1, \ldots, A_n\, a_n \rangle$ se verifica

$$\sum_{P'} \{\!|\, p \mid P \stackrel{s}{\Longrightarrow}_p P' \,|\!\} = \sum_{A} \{\!|\, q \cdot q_s \mid (A, q) \in \mathcal{A}(P, s) \,|\!\}$$

donde $q_\epsilon = 1$ y $q_{s' \circ \langle B\, b \rangle} = \sum_{Q'} \{\!|\, q_i \mid P \stackrel{s'}{\Longrightarrow}_{q_i} Q' \ \wedge\ S(Q') = B \,|\!\}$.

Demostración: A partir de la Definición 4.7 obtenemos

$$(A, q) \in \mathcal{A}(P, s) \iff q = \frac{\sum_{P'} \{\!|\, p \mid P \stackrel{s}{\Longrightarrow}_p P' \ \wedge\ S(P') = A \,|\!\}}{q_s} > 0$$

de lo cual, sustituyendo el valor de cada q en la parte izquierda de la ecuación, se deduce inmediatamente el resultado buscado. □

Lema 4.17 Para todo proceso P, y toda secuencia s, se verifica

$$\sum_{P'} \{\!|\, p \mid P \stackrel{s}{\Longrightarrow}_p P' \,|\!\} = \sum_{P'} \{\!|\, p \mid \widehat{\mathcal{A}}(P) \stackrel{s}{\Longrightarrow}_p P' \,|\!\}$$

Demostración: La demostración la realizaremos por inducción sobre la longitud de la secuencia s.

Caso base: $(s = \epsilon)$

Aplicando el Lema 4.16 obtenemos

$$\sum_{P'} \{\!|\, p \mid P \stackrel{\epsilon}{\Longrightarrow}_p P' \,|\!\} = \sum_{B} \{\!|\, p \mid (B, p) \in \mathcal{A}(P, \epsilon) \,|\!\}$$

Entonces, por el Lema 4.13, al ser $P \cong \widehat{\mathcal{A}}(P)$, tenemos que las probabilidades asociadas a los conjuntos de aceptación del proceso P tras la secuencia vacía son iguales a las correspondientes al proceso $\widehat{\mathcal{A}}(P)$, de lo cual se sigue que el valor anterior es igual a

$$\sum_{B} \{\!|\, p \mid (B, p) \in \mathcal{A}(\widehat{\mathcal{A}}(P), \epsilon) \,|\!\} = \sum_{P'} \{\!|\, p \mid \widehat{\mathcal{A}}(P) \stackrel{\epsilon}{\Longrightarrow}_p P' \,|\!\}$$

Caso Inductivo: $(s = s' \circ \langle B\, b \rangle)$

De nuevo, aplicando el Lema 4.16 obtenemos

$$\sum_{P'} \{\!|\, p \mid P \stackrel{s}{\Longrightarrow}_p P' \,|\!\} = \sum_{A} \{\!|\, q \cdot q' \mid (A, q) \in \mathcal{A}(P, s) \,|\!\} \tag{4.1}$$

donde $q' = \sum_{Q'} \{\!|\, q_i \mid P \stackrel{s'}{\Longrightarrow}_{q_i} Q' \ \wedge\ S(Q') = B \,|\!\}$.

Por hipótesis de inducción, y dado que $P \cong \widehat{\mathcal{A}}(P)$, tenemos

$$q' = \sum_{Q'} \{\!|\, q_i \mid \widehat{\mathcal{A}}(P) \stackrel{s'}{\Longrightarrow}_{q_i} Q' \ \wedge\ S(Q') = B \,|\!\} = q''$$

con lo cual, aplicando de nuevo que $P \cong \widehat{\mathcal{A}}(P)$, obtenemos que el valor dado por la fórmula (4.1) es igual a

$$\sum_{A} \{\!|\, q \cdot q'' \mid (A, q) \in \mathcal{A}(\widehat{\mathcal{A}}(P), s) \,|\!\}$$

y aplicando de nuevo el Lema 4.16 obtemos que el valor de la fórmula anterior es igual a

$$\sum_{P'} \{\!|\, p \mid \widehat{\mathcal{A}}(P) \stackrel{s}{\Longrightarrow}_p P' \,|\!\}$$

como queríamos demostrar. □

Teorema 4.1 Para todo proceso P se verifica $P \approx_{\mathcal{G}} \widehat{\mathcal{A}}(P)$.

Demostración: Tenemos que probar que para cualquier prueba $T \in \mathcal{G}$, la probabilidad con la que el proceso P pasa T es igual a la probabilidad con la cual su forma normal, es decir $\widehat{\mathcal{A}}(P)$, pasa T. En virtud del Lema 2.14, podemos suponer que las pruebas no contienen elecciones internas, y que no hay no determinismo provocado por acciones iguales bajo una elección externa. Por lo tanto, al no contener las pruebas ningún tipo de elección interna, la composición en paralelo del proceso y la prueba resolverá en primer lugar las decisiones internas iniciales del proceso, tras lo que se alcanzará un estado estable en el cual el proceso y la prueba interaccionarán. Pero por el Lema 4.17, la probabilidad de que el proceso P ejecute una secuencia de pares (estado, acción), es decir la de llegar a estados estables después de dicha secuencia, es igual a la probabilidad de que la ejecute su forma normal, y dado que por

el Lema 4.13 la probabilidad asociada con los estados del proceso y de su forma normal son la misma para cualquier secuencia de acciones, tendremos que la interacción entre el proceso y la prueba puede ser simulada, con las mismas probabilidades, por la forma normal interaccionando con la prueba, con lo que la probabilidad de pasar dicha prueba será la misma en ambos casos.

En resumen, al componer en paralelo al proceso con una prueba, primero se ejecutan las elecciones internas hasta que se llega a un estado estable, después se interacciona con la prueba, y tras ello se repite el ciclo hasta llegar o bien a un interbloqueo o a la ejecución de la acción de visto bueno por parte de la prueba, y en virtud de los Lemas 4.13 y 4.17, los mismos cómputos serán posibles para la forma normal del proceso en cuestión. □

Nos disponemos ahora a probar que la noción de equivalencia inducida por la semántica de pruebas y la definida por los conjuntos de aceptación son coincidentes. Al efecto, en virtud del Lema 4.14, resulta suficiente restringirse a procesos de la forma $\widehat{\mathcal{A}}(P)$. Pero antes de nada necesitaremos un resultado técnico de unicidad. Dicho resultado es similar al que aparece en el Lema 9 de [WSS94], pero debido a que nosotros hemos definido las probabilidades de las transiciones de una forma ligeramente diferente, la formulación del lema no es exactamente la misma.

Lema 4.18 Sean f y f' dos funciones racionales de $n \geq 0$ variables $x_1, x_2 \ldots x_n$, definidas como sigue:

$$f = \sum_{i \in I} \frac{c_i}{1 + \sum_{j=1}^{n} d_{j,i} \cdot x_j} \qquad f' = \sum_{i' \in I'} \frac{c'_{i'}}{1 + \sum_{j=1}^{n} d'_{j,i'} \cdot x_j}$$

donde I, I' son conjuntos finitos de índices; $c_i, c'_{i'} > 0$; para cada $r, s \in I$, tales que $r \neq s$, las tuplas $(d_{1,r}, d_{2,r}, \ldots, d_{n,r})$ y $(d_{1,s}, d_{2,s}, \ldots, d_{n,s})$ son distintas; y para cada $r, s \in I'$, tales que $r \neq s$, las tuplas $(d'_{1,r}, d'_{2,r}, \ldots, d'_{n,r})$ y $(d'_{1,s}, d'_{2,s}, \ldots, d'_{n,s})$ son distintas. En tal situación, si $f = f'$ existirá una biyección $h : I \longrightarrow I'$ tal que $d_{j,i} = d'_{j,h(i)}$ y $c_i = c'_{h(i)}$ para todo $i \in I$ y $1 \leq j \leq n$, de modo que las expresiones definiendo f y f' serán idénticas salvo conmutatividad.

Demostración: Asumamos de momento que en la definición de f no aparece ninguna tupla nula, es decir, se tiene $\forall i \in I\ \exists j \in \{1 \ldots n\} :\ d_{j,i} \neq 0$.

Tomemos $p_i = 1 + \sum_{j=1}^{n} d_{j,i} \cdot x_j$ y $p'_{i'} = 1 + \sum_{j=1}^{n} d'_{j,i'} \cdot x_j$, y consideremos el polinomio $P = \prod_{i \in I} p_i \cdot \prod_{i' \in I'} p'_{i'}$. Entonces se tiene $f \cdot P = f' \cdot P$, es decir:

$$\sum_{i \in I} c_i \cdot \prod_{\substack{k \in I \\ k \neq i}} p_k \cdot \prod_{k' \in I'} p'_{k'} = \sum_{i' \in I'} c'_{i'} \cdot \prod_{k \in I} p_k \cdot \prod_{\substack{k' \in I' \\ k' \neq i'}} p'_{k'}$$

Para cada $i \in I$, consideremos el conjunto de raíces $\bar{E}_i \subseteq \mathbb{R}^n$ del polinomio p_i. Entonces, para todo $\bar{e}_i \in \bar{E}_i$ tenemos

$$(f \cdot P)(\bar{e}_i) = c_i \cdot \overbrace{\prod_{\substack{k \in I \\ k \neq i}} p_k(\bar{e}_i) \cdot \prod_{k' \in I'} p'_{k'}(\bar{e}_i)}^{R_i(\bar{e}_i)} = (f' \cdot P)(\bar{e}_i) = 0$$

Como $c_i > 0$, para cada $\bar{e}_i \in \bar{E}_i$ se cumple

$$R_i(\bar{e}_i) = \prod_{\substack{k \in I \\ k \neq i}} p_k \cdot \prod_{k' \in I'} p'_{k'} = 0 \tag{4.2}$$

Geométricamente hablando, los conjuntos $\bar{E}_i$ representan los puntos de un hiperplano en $\mathbb{R}^n$, mientras que la fórmula de la ecuación (4.2), $R_i(\bar{e}_i) = 0$, representa la unión de los hiperplanos correspondientes a cada $k' \in I'$ y a los $k \in I$, $k \neq i$. Pero esta unión es finita, por lo que si $R_i(x) = 0$ para todos los puntos de un hiperplano (en este caso el hiperplano $\bar{E}_i$, o equivalentemente las raíces del polinomio p_i), entonces ha de existir un cierto $k \in I$, $k \neq i$ de modo que el hiperplano asociado a p_k es el hiperplano $\bar{E}_i$, o bien existirá un $k' \in I'$ tal que el hiperplano asociado a $p'_{k'}$ es el hiperplano $\bar{E}_i$. Pero si dos polinomios definen el mismo hiperplano, sus coeficientes deben ser proporcionales, y dado que todos estos polinomios tienen como término independiente el valor 1, dicho coeficiente de proporcionalidad debería ser igual a 1. Ahora bien, el primer caso no se puede producir, pues la hipótesis del lema nos indica que todas las tuplas que aparecen en f son distintas; de modo que debe existir un $k' \in I'$ tal que $p_i = p'_{k'}$.

Como este razonamiento se puede repetir para cada $i \in I$, tendremos una función $h :\ I \longrightarrow I'$ tal que $d_{j,i} = d'_{j,h(i)}$ para cada $i \in I$ y $1 \leq j \leq n$. Dado que todas las

tuplas que aparecen en f y f' son distintas, por la definición de la función h tenemos que es inyectiva. Para simplificar, realizando el ordenamiento oportuno en los índices de I', podemos asumir que $h(i) = i$. Ello nos permite descomponer I' en la forma $I' = I \cup J$ con $I \cap J = \emptyset$, de modo que obtenemos

$$f = \sum_{i \in I} \frac{c_i}{p_i} \qquad f' = \sum_{i \in I} \frac{c'_i}{p_i} + \sum_{j \in J} \frac{c'_j}{p'_j}$$

donde por las hipótesis del lema, los polinomios p'_j serían distintos de los p_i.

Sea entonces $Q = \prod_{i \in I} p_i$ y consideremos $f \cdot Q$ y $f' \cdot Q$, es decir

$$f \cdot Q = \sum_{i \in I} c_i \cdot \prod_{\substack{k \in I \\ k \neq i}} p_k \qquad f' \cdot Q = \sum_{i \in I} c'_i \cdot \prod_{\substack{k \in I \\ k \neq i}} p_k + \sum_{j \in J} \frac{c'_j}{p'_j} \cdot \prod_{i \in I} p_i$$

Para cada $i \in I$, consideramos $\bar{e}_i \in \bar{E}_i$ verificando $p_i(\bar{e}_i) = 0$ y tal que para todo $k \in (I - \{i\}) \cup J$ se tenga $p_k(\bar{e}_i) \neq 0$. Nótese que un punto tal siempre existe, dado que el conjunto de los hiperplanos que aparecen en $(I - \{i\}) \cup J$ es finito, y por tanto la unión de las intersecciones entre estos hiperplanos y el hiperplano $\bar{E}_i$ no puede ser igual a $\bar{E}_i$. Para dicho punto se verifica

$$(f \cdot Q)(\bar{e}_i) = c_i \cdot \prod_{\substack{k \in I \\ k \neq i}} p_k(\bar{e}_i) = (f' \cdot Q)(\bar{e}_i) = c'_i \cdot \prod_{\substack{k \in I \\ k \neq i}} p_k(\bar{e}_i) + \sum_{j \in J} \frac{c'_j}{p'_j(\bar{e}_i)} \cdot \prod_{i \in I} p_i(\bar{e}_i)$$

Si tenemos en cuenta que $c_i, c'_i > 0, \prod_{\substack{k \in I \\ k \neq i}} p_k(\bar{e}_i) \neq 0$, para todo $j \in J$ se tiene $p'_j(\bar{e}_i) \neq 0$, y finalmente que $\prod_{i \in I} p_i(\bar{e}_i) = 0$, obtenemos

$$(f \cdot Q)(\bar{e}_i) = c_i \cdot \prod_{\substack{k \in I \\ k \neq i}} p_k(\bar{e}_i) = (f' \cdot Q)(\bar{e}_i) = c'_i \cdot \prod_{\substack{k \in I \\ k \neq i}} p_k(\bar{e}_i)$$

de donde resulta $c_i = c'_i$ para cada $i \in I$. En consecuencia

$$f = \sum_{i \in I} \frac{c_i}{p_i} \qquad f' = \sum_{i \in I} \frac{c_i}{p_i} + \sum_{j \in J} \frac{c'_j}{p'_j} \tag{4.3}$$

Faltaría probar para concluir la demostración que $J = \emptyset$. Consideremos $f(\bar{0})$ y $f'(\bar{0})$. Sustituyendo en la fórmula (4.3) obtenemos $f(\bar{0}) = \sum_{i \in I} c_i$, mientras que

$f'(\bar{0}) = \sum_{i\in I} c_i + \sum_{j\in J} c'_j$. Como $f = f'$ y los valores c'_j $(j \in J)$ son mayores que cero, debe cumplirse que $\sum_{j\in J} c'_j = 0$, lo cual implica que $J = \emptyset$. De ello se deduce inmediatamente que la la función h introducida previamente es biyectiva.

Afrontemos ahora el caso en el que alguna de las tuplas que aparecen en la definición de f es la tupla nula, es decir $\exists k \in I\ \forall j \in \{1 \ldots n\} :\ d_{k,j} = 0$. Para dicho k no podríamos utilizar el razonamiento anterior consistente en considerar una raíz del polinomio asociado (es decir de p_k), dado que dicho polinomio no tiene raíces. Sin embargo podemos seguir realizando el razonamiento para los demás elementos de I, con lo que en lugar de llegar a la fórmula (4.3), llegaríamos a:

$$f = \sum_{i\in I-\{k\}} \frac{c_i}{p_i} + c_k \qquad f' = \sum_{i\in I-\{k\}} \frac{c_i}{p_i} + \sum_{j\in J} \frac{c'_j}{p'_j} \tag{4.4}$$

donde $I' = (I - \{k\}) \cup J$. Faltaría por probar entonces que $J = \{k'\}$, $c'_{k'} = c_k$ y $p'_{k'} = 1$. Dado que $f = f'$, de la fórmula (4.4) obtenemos $c_k = \sum_{j\in J} \frac{c'_j}{p'_j}$. Como quiera que c_k es una constante, la parte derecha de la igualdad anterior debe ser constante, por lo que todos los polinomios p'_j han de ser iguales[5] al polinomio 1, de modo que sus tuplas asociadas serán la tupla nula, y dado que por las hipótesis del lema no pueden existir tuplas repetidas, sólo aparecerá una tupla tal en J, con lo que queda probado el resultado deseado. □

El siguiente resultado técnico será utilizado también en la demostración del Teorema de Caracterización.

Lema 4.19 Sean $p_1, p_2, \ldots, p_n, p'_1, p'_2, \ldots, p'_n \geq 0$, tales que $\sum_{i=1}^{n} p_i = \sum_{i=1}^{n} p'_i$, y $r, r' > 0$. Entonces,

$$\forall\, 1 \leq i \leq n :\ \frac{p_i}{r} = \frac{p'_i}{r'} \implies \forall\, 1 \leq i \leq n :\ p_i = p'_i \ \wedge\ r = r'$$

[5]Supongamos que existe un polinomio distinto del polinomio 1. Si pasamos todos los términos constantes de la parte derecha a la parte izquierda, y calculamos la suma de las fracciones que quedan en la parte derecha, obtenemos en el denominador un polinomio al menos un grado mayor que el del numerador. Si a continuación pasamos el denominador a la parte izquierda tenemos la igualdad entre dos polinomios de distinto grado, lo cual es una contradicción.

Demostración: Como quiera que $\forall\ 1 \leq i \leq n:\ \frac{p_i}{r} = \frac{p'_i}{r'}$, entonces los sumatorios en i de tales términos serán iguales en los dos casos, es decir

$$\sum_{i=1}^{n} \frac{p_i}{r} = \frac{1}{r} \cdot \sum_{i=1}^{n} p_i = \frac{1}{r'} \cdot \sum_{i=1}^{n} p'_i = \sum_{i=1}^{n} \frac{p'_i}{r'}$$

Por hipótesis, $\sum_{i=1}^{n} p_i = \sum_{i=1}^{n} p'_i$, por lo tanto $r = r'$. De aquí se deduce trivialmente que $\forall\ 1 \leq i \leq n:\ p_i = p'_i$. □

Teorema 4.2 Para cualesquiera procesos P y P' se verifica

$$\widehat{\mathcal{A}}(P) \approx_{\mathcal{G}} \widehat{\mathcal{A}}(P') \iff \widehat{\mathcal{A}}(P) = \widehat{\mathcal{A}}(P')$$

Demostración: La implicación de izquierda a derecha es trivial, pues si dos procesos en forma normal son (sintácticamente) iguales, pasarán cualquier prueba con la misma probabilidad.

Para la implicación de derecha a izquierda, consideremos la unión de los alfabetos de los dos procesos

$$\alpha(P) \cup \alpha(P') = \{a_{i_1}, \ldots, a_{i_n}\} \subseteq Act$$

Nótese que dicha unión es finita. Por simplicidad en la notación, supondremos que $\forall\ 1 \leq j \leq n : i_j = j$, de modo que

$$\alpha(P) \cup \alpha(P') = \{a_1, \ldots, a_n\} \subseteq Act$$

Consideremos ahora, $\widehat{\mathcal{A}}(P)$ y $\widehat{\mathcal{A}}(P')$ que tendrán la forma

$$\widehat{\mathcal{A}}(P) = \bigoplus_{i=1}^{m} [p_i] \sum_{j=1}^{n} [p_{i,j}]\ a_j\ ;\ C_{i,j} \qquad \widehat{\mathcal{A}}(P') = \bigoplus_{i=1}^{m'} [p'_i] \sum_{j=1}^{n} [p'_{i,j}]\ a_j\ ;\ C'_{i,j}$$

donde para simplificar la notación tomamos $p_{i,j} = 0$ cuando la acción a_j no aparecía originalmente en el i-ésimo *sumando* de $\widehat{\mathcal{A}}(P)$ y análogamente para $\widehat{\mathcal{A}}(P')$. Tomaremos $A_i = \{(a_j, p_{i,j}) \mid p_{i,j} > 0\}$ y $A'_i = \{(a_j, p'_{i,j}) \mid p'_{i,j} > 0\}$.

Distinguiremos ahora tres casos dependiendo de la forma que tengan los procesos $\widehat{\mathcal{A}}(P)$ y $\widehat{\mathcal{A}}(P')$:

1. Los dos son finitos.

2. Uno es finito y el otro es infinito.

3. Los dos son infinitos.

Comenzando por el primero de los casos, realizaremos la demostración por inducción completa:

1. (*Caso Base*): Probaremos que si dos procesos en forma normal pasan todas las pruebas con la misma probabilidad, entonces son iguales en su *primer piso*, es decir, se cumple:

$$m = m' \ \wedge \ \forall\, 1 \leq i \leq m : \ (p_i = p'_i \ \wedge \ \forall\, 1 \leq j \leq n : \ p_{i,j} = p'_{i,j})$$

2. (*Caso Inductivo*): Supondremos que las formas normales son iguales en su primer *piso*, pero que alguna de las *continuaciones* es distinta, es decir que existen un estado A_i y una acción $a_j \in Act(A_i)$, tales que $C_{i,j} \neq C'_{i,j}$. Demostraremos entonces que existirá una prueba distinguiendo a las dos formas normales, lo que supone una demostración vía el contrarecíproco del resultado deseado.

Caso base:

Partimos de que las dos formas normales pasan con la misma probabilidad todas las pruebas. En particular, para cada distribución de probabilidad de la forma $\bar{q} = \langle q_1, q_2, \ldots q_n, q_{n+1} \rangle$ con $q_i > 0$, y $\sum q_i = 1$, consideraremos la prueba

$$T^{\bar{q}} = \sum_{j=1}^{n+1} [q_j]\, a_j; Nil$$

donde tomamos $a_{n+1} = \omega$. Al componer una prueba tal con el proceso $\widehat{\mathcal{A}}(P)$ obtenemos

$$pass(\widehat{\mathcal{A}}(P)\,,\, T^{\bar{q}}) \ = \sum_{i=1}^{m} p_i \cdot \frac{q_{n+1}}{q_{n+1} + \sum_{j=1}^{n} p_{i,j} \cdot q_j} \tag{4.5}$$

pues tenemos en primer lugar las elecciones internas del proceso, cada una de ellas con probabilidad p_i, y tras las mismas la prueba sólo se pasará mediante la ejecución inmediata de la acción ω, lo cual sucederá con una probabilidad que viene dada por el cociente entre el valor q_{n+1} asociado a ω y el correspondiente factor de normalización[6].

Dado que $q_{n+1} > 0$, podemos dividir arriba y abajo en tales cocientes por dicho valor, tras lo que la fórmula (4.5) toma la forma

$$\sum_{i=1}^{m} p_i \cdot \frac{1}{1 + \sum_{j=1}^{n} p_{i,j} \cdot \frac{q_j}{q_{n+1}}} \tag{4.6}$$

Realizando el mismo proceso para $\widehat{\mathcal{A}}(P')$ obtenemos

$$pass(\widehat{\mathcal{A}}(P')\,,\,T^{\bar{q}}) = \sum_{i=1}^{m'} p'_i \cdot \frac{1}{1 + \sum_{j=1}^{n} p'_{i,j} \cdot \frac{q_j}{q_{n+1}}} \tag{4.7}$$

Si hacemos un cambio de variable de la forma $q'_j = \frac{q_j}{q_{n+1}}$ e igualamos las expresiones derivadas para $pass(\widehat{\mathcal{A}}(P)\,,\,T^{\bar{q}})$ y $pass(\widehat{\mathcal{A}}(P')\,,\,T^{\bar{q}})$ obtenemos

$$\sum_{i=1}^{m} p_i \cdot \frac{1}{1 + \sum_{j=1}^{n} p_{i,j} \cdot q'_j} = \sum_{i=1}^{m'} p'_i \cdot \frac{1}{1 + \sum_{j=1}^{n} p'_{i,j} \cdot q'_j}$$

Estamos entonces en condiciones de aplicar el Lema 4.18 [7], con lo que obtenemos

$$m = m' \;\wedge\; \forall\, 1 \leq i \leq m :\; (p_i = p'_i \;\wedge\; \forall\, 1 \leq j \leq n :\; p_{i,j} = p'_{i,j})$$

[6] Como quiera que en las pruebas consideradas todas las probabilidades asociadas a las acciones del alfabeto de los procesos son mayores que cero, el resultado de componer el proceso y la prueba sería el mismo si utilizáramos factores de prenormalización (ver página 40).

[7] Esta afirmación necesita una explicación adicional. Las hipótesis del Lema 4.18 exigían que las dos funciones tenían que ser iguales para cualquier valor de las incógnitas (i.e. cualquier tupla de $\mathbb{R}^n$), mientras que nosotros sólo tenemos la igualdad para valores $q'_1, \ldots, q'_n$, tales que $q'_i > 0$ y $\sum q'_i = \frac{1-q_{n+1}}{q_{n+1}}$. Haciendo tender q_{n+1} a cero, podemos conseguir valores arbitrariamente grandes para las variables q'_i, con lo que obtenemos que estas dos funciones son iguales para cualquier valor (estrictamente) positivo de las variables q'_i. Pero por tratarse de funciones racionales, si coinciden sobre todo el cuadrante positivo de $\mathbb{R}^n$, serán iguales sobre cualquier valor de $\mathbb{R}^n$, por lo que podremos aplicar el Lema 4.18.

con lo que queda probado que las dos formas normales coinciden en su primer *piso*.

Caso Inductivo:

Supongamos que las dos formas normales son iguales en su primer piso, pero que existe una continuación que es distinta. De entre aquellos estados iniciales del proceso $\widehat{\mathcal{A}}(P)$ tales que una de sus continuaciones tras una de sus acciones es distinta a la correspondiente continuación del proceso $\widehat{\mathcal{A}}(P')$, consideraremos uno de los que sea *minimal* en su número de acciones, en el siguiente sentido de minimalidad:

> Sea A_j un estado del proceso $\widehat{\mathcal{A}}(P)$ tal que existe $a_k \in Act(A_j)$, con $C_{j,k} \neq C'_{j,k}$ de modo que para todo estado A_r del proceso $\widehat{\mathcal{A}}(P)$ tal que $Act(A_r) \subset Act(A_j)$, se cumple que para todo $a_s \in Act(A_r)$, tenemos $C_{r,s} = C'_{r,s}$.

Obviamente, el estado elegido será distinto del vacío, pues tras el estado vacío no hay continuaciones. Para simplificar la notación, supongamos que

$$Act(A_j) = \{a_1, \ldots, a_k, \ldots, a_r\} \subseteq \{a_1, \ldots, a_n\} = \alpha(P) \cup \alpha(P')$$

Una vez que tenemos fijado el estado A_j, vamos a realizar una partición de los estados alcanzables por los procesos en tres grupos: aquéllos cuyo conjunto de acciones está contenido en las acciones del estado A_j; aquéllos que tienen las mismas acciones que A_j; y el resto, es decir aquéllos que tienen acciones que no pertenecen a las acciones de A_j. Formalmente:

- $A^1 = \{i \mid \exists p, Q : \ \widehat{\mathcal{A}}(P) \stackrel{\epsilon}{\Longrightarrow}_p Q \ \wedge \ S(Q) = A_i \ \wedge \ Act(A_i) \subset Act(A_j)\}$.
- $A^2 = \{i \mid \exists p, Q : \ \widehat{\mathcal{A}}(P) \stackrel{\epsilon}{\Longrightarrow}_p Q \ \wedge \ S(Q) = A_i \ \wedge \ Act(A_i) = Act(A_j)\}$.
- $A^3 = \{i \mid \exists p, Q : \ \widehat{\mathcal{A}}(P) \stackrel{\epsilon}{\Longrightarrow}_p Q \ \wedge \ S(Q) = A_i \ \wedge \ i \notin A^1 \cup A^2\}$.

Como hemos supuesto que $C_{j,k} \neq C'_{j,k}$, por hipótesis de inducción existirá una prueba T que distinga a estas dos formas normales:

$$pass(C_{j,k}\,,\, T) \neq pass(C'_{j,k}\,,\, T) \tag{4.8}$$

Como al menos uno de estos valores ha de ser no nulo, supongamos $pass(C'_{j,k}, T) > 0$ (obviamente, el caso simétrico se trataría igual).

Entonces, para cada distribución de probabilidad de la forma $\bar{q} = \langle q_1, q_2, \ldots q_r \rangle$ tal que $q_i > 0$, y $\sum q_i = 1$, consideramos las pruebas

$$T^{\bar{q}}_{k,\delta} = (\sum_{s=1}^{r} [q_s]\ a_s; T_s) +_\delta (\sum_{s=r+1}^{n} [\frac{1}{n-r}]\ a_s; Nil) \quad \text{donde } T_s = \begin{cases} T & \text{si } s = k \\ Nil & \text{e.o.c.} \end{cases}$$

donde suponemos que si $n = r$ entonces el segundo sumando no aparece.

Al componer tales pruebas con el proceso $\widehat{\mathcal{A}}(P)$ obtenemos:

$$pass(\widehat{\mathcal{A}}(P)\ ,\ T^{\bar{q}}_{k,\delta}) = \sum_{i \in A^1} p_i \cdot \frac{\delta \cdot q_k \cdot p_{i,k} \cdot\ pass(C_{i,k}\ ,\ T)}{\sum_{s=1}^{r} \delta \cdot p_{i,s} \cdot q_s} \tag{4.9}$$

$$+ \sum_{i \in A^2} p_i \cdot \frac{\delta \cdot q_k \cdot p_{i,k} \cdot\ pass(C_{i,k}\ ,\ T)}{\sum_{s=1}^{r} \delta \cdot p_{i,s} \cdot q_s} \tag{4.10}$$

$$+ \sum_{i \in A^3} p_i \cdot \frac{\delta \cdot q_k \cdot p_{i,k} \cdot\ pass(C_{i,k}\ ,\ T)}{\sum_{s=1}^{r} \delta \cdot p_{i,s} \cdot q_s + \sum_{s=r+1}^{n} (1-\delta) \cdot p_{i,s} \cdot \frac{1}{n-r}} \tag{4.11}$$

En cada uno de los casos, si para algún i, $a_k \notin A_i$, aunque la continuación $C_{i,k}$ no exista, tal hecho no producirá ningún problema, pues $p_{i,k}$ sería igual a 0. Si $n = r$, el último sumando no existirá, y las δ's que aparecen en los otros dos sumandos no provocan ningún conflicto dado que se puede eliminar (al ser $\delta > 0$).

De igual forma, para el proceso $\widehat{\mathcal{A}}(P')$ tendremos:

$$pass(\widehat{\mathcal{A}}(P')\ ,\ T^{\bar{q}}_{k,\delta}) = \sum_{i \in A^1} p_i \cdot \frac{\delta \cdot q_k \cdot p_{i,k} \cdot\ pass(C'_{i,k}\ ,\ T)}{\sum_{s=1}^{r} \delta \cdot p_{i,s} \cdot q_s} \tag{4.12}$$

$$+ \sum_{i \in A^2} p_i \cdot \frac{\delta \cdot q_k \cdot p_{i,k} \cdot\ pass(C'_{i,k}\ ,\ T)}{\sum_{s=1}^{r} \delta \cdot p_{i,s} \cdot q_s} \tag{4.13}$$

$$+ \sum_{i \in A^3} p_i \cdot \frac{\delta \cdot q_k \cdot \cancel{p_{i,k} \cdot pass(C'_{i,k}, T)}}{\sum_{s=1}^{r} \delta \cdot p_{i,s} \cdot q_s + \sum_{s=r+1}^{n} (1-\delta) \cdot p_{i,s} \cdot \frac{1}{n-r}} \quad (4.14)$$

Vamos a realizar la demostración por contradicción; es decir supondremos que las dos formas normales pasan con la misma probabilidad todas las pruebas, y llegaremos a que $C_{j,k} = C'_{j,k}$.

Si las formas normales pasan con la misma probabilidad todas las pruebas, en particular pasarán con la misma probabilidad las pruebas de la forma $T^{\bar{q}}_{k,\delta}$, es decir

$$pass(\widehat{\mathcal{A}}(P)\,,\, T^{\bar{q}}_{k,\delta}) \;=\; pass(\widehat{\mathcal{A}}(P')\,,\, T^{\bar{q}}_{k,\delta}) \qquad (\forall\, 0 < \delta < 1 \;\wedge\; \forall\, \bar{q})$$

Como la igualdad es cierta para cualquier $0 < \delta < 1$, se mantendrá si hacemos tender δ a cero, de modo que

$$\lim_{\delta \to 0} pass(\widehat{\mathcal{A}}(P)\,,\, T^{\bar{q}}_{k,\delta}) = \lim_{\delta \to 0} pass(\widehat{\mathcal{A}}(P')\,,\, T^{\bar{q}}_{k,\delta}) \qquad (\forall\, \bar{q}) \qquad (4.15)$$

Si examinamos los sumandos que constituyen ambos lados de la igualdad anterior, observamos que al tender δ a cero, (4.11) y (4.14) también tienden a cero, dado que para cada $i \in A^3$, el numerador tiende a cero, mientras que el denominador tiende a $\sum_{s=r+1}^{n} p_{i,s} \cdot \frac{1}{n-r} > 0$.

Por otra parte, por la elección que hemos hecho del estado A_j, los sumandos (4.9) y (4.12) tendrán el mismo valor, al ser las continuaciones, es decir, los procesos $C_{i,k}$ y $C'_{i,k}$, idénticas para cada $i \in A^1$.

Por lo tanto, si la igualdad expresada por la fórmula (4.15) es cierta, a la vista de los dos párrafos anteriores, para cualquier distribución de probabilidad de la forma $\bar{q} = \langle q_1, q_2, \ldots q_r \rangle$ tal que $q_i > 0$, y $\sum q_i = 1$ se ha de cumplir la igualdad entre las expresiones dadas por (4.10) y (4.13), con lo cual, al dividir cada sumando entre $\delta \cdot q_k \cdot p_{i,k}$, obtenemos:

$$\sum_{i \in A^2} p_i \cdot \frac{pass(C_{i,k}, T)}{1 + \sum_{\substack{s=1 \\ s \neq k}}^{r} \frac{p_{i,s}}{p_{i,k}} \cdot \frac{q_s}{q_k}} = \sum_{i \in A^2} p_i \cdot \frac{pass(C'_{i,k}, T)}{1 + \sum_{\substack{s=1 \\ s \neq k}}^{r} \frac{p_{i,s}}{p_{i,k}} \cdot \frac{q_s}{q_k}}$$

Si eliminamos los sumandos nulos, y consideramos el cambio de variable $q'_s = \frac{q_s}{q_k}$, obtenemos:

$$\sum_{\substack{pass(C_{i,k}\,,\,T)\neq 0 \\ i\in A^2}} p_i \cdot \frac{pass(C_{i,k}\,,\,T)}{1+\sum_{\substack{s=1 \\ s\neq k}}^{r} \frac{p_{i,s}}{p_{i,k}} \cdot q'_s} = \sum_{\substack{pass(C'_{i,k}\,,\,T)\neq 0 \\ i\in A^2}} p_i \cdot \frac{pass(C'_{i,k}\,,\,T)}{1+\sum_{\substack{s=1 \\ s\neq k}}^{r} \frac{p_{i,s}}{p_{i,k}} \cdot q'_s}$$

Podemos entonces aplicar el Lema 4.18, dado que todos los sumandos son distintos de cero y aplicando el Lema 4.19 a los sumandos de la forma $\frac{p_{i,s}}{p_{i,k}}$ tenemos que las correspondientes tuplas son distintas entre sí.

Dado que $pass(C_{j,k}\,,\,T) > 0$, y $j \in A^2$, debe existir algún j' que verifique las siguientes condiciones:

1. $pass(C_{j,k}\,,\,T) = pass(C'_{j',k}\,,\,T)$.
2. $\forall\, 1 \leq s \leq r,\ s \neq k:\ \frac{p_{j,s}}{p_{j,k}} = \frac{p_{j',s}}{p_{j',k}}$.

Pero $\sum_{s=1}^{r} p_{j,s} = 1 = \sum_{s=1}^{r} p_{j',s}$ con lo cual, aplicando de nuevo el Lema 4.19, obtenemos $\forall\, 1 \leq s \leq r:\ p_{j,s} = p_{j',s}$. Al tratarse de estados de un proceso, y no poder haber repeticiones por ello, necesariamente se debe cumplir $j = j'$. Pero si $j = j'$, entonces tenemos $pass(C_{j,k}\,,\,T) = pass(C'_{j,k}\,,\,T)$ lo que contradice la ecuación (4.8). Esto termina la demostración para el caso en el que los dos procesos son finitos.

Si uno de los dos procesos es finito y el otro no, supongamos que el proceso finito tiene *profundidad* n (i.e. todas las secuencias de longitud mayor que n devuelven como conjunto de aceptación $\emptyset$). Entonces, podemos considerar los $n+1$ primeros *pisos* del proceso infinito, y utilizando un razonamiento similar al anterior encontraremos una prueba que los distinga.

Si los dos procesos son infinitos pero distintos, la diferencia entre ellos se alcanzará en tiempo finito, es decir existirá una secuencia finita s tal que $\mathcal{A}(\widehat{\mathcal{A}}(P), s) \neq \mathcal{A}(\widehat{\mathcal{A}}(P'), s)$, y entonces podremos utilizar el razonamiento anterior, considerando los dos procesos hasta profundidad $\mid s \mid$. □

Corolario 4.3 (Teorema de Caracterización)

Para cualesquiera procesos P y P' se verifica $P \approx_{\mathcal{G}} P' \iff P \cong P'$.

Demostración: A partir del Teorema 4.1 obtenemos $P \approx_{\mathcal{G}} P' \Leftrightarrow \hat{\mathcal{A}}(P) \approx_{\mathcal{G}} \hat{\mathcal{A}}(P')$; aplicando el Teorema 4.2 obtenemos $\hat{\mathcal{A}}(P) \approx_{\mathcal{G}} \hat{\mathcal{A}}(P') \Leftrightarrow \hat{\mathcal{A}}(P) = \hat{\mathcal{A}}(P')$; y para finalizar, por el Lema 4.14, tenemos $\hat{\mathcal{A}}(P) = \hat{\mathcal{A}}(P') \Leftrightarrow P \cong P'$. □

La demostración del Teorema 4.2, además de permitirnos demostrar que las relaciones $\approx_{\mathcal{G}}$ y $\cong$ identifican a los mismos procesos, nos da por añadidura un conjunto de pruebas *esenciales*, que resultan suficientes para distinguir cualesquiera procesos no equivalentes. Definiremos a continuación de forma precisa este conjunto de pruebas, a las que llamaremos *barbas probabilísticas*. Como ya comentamos en la introducción de este capítulo, existe un gran parecido entre nuestras barbas probabilísticas y las *trazas probabilísticas* de [YCDS94], si consideramos estas últimas como pruebas probabilísticas.

Definición 4.20 El conjunto de las *barbas probabilísticas*, que notaremos por $\mathcal{PB}$, se define mediante la siguiente expresión BNF:

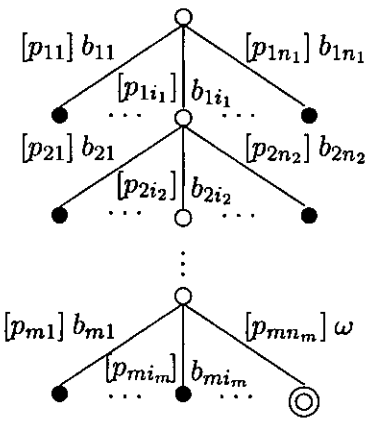
$$T ::= \sum_{i=1}^{s}[p_i]\ (b_i; Nil) +_p \omega \mid \sum_{i=1}^{s}[p_i]\ b_i; T_i \qquad \text{donde } T_i = \begin{cases} T & \text{si } i = s \\ Nil & \text{e.o.c.} \end{cases}$$

siendo $p \in (0,1)$, $\sum p_i = 1$, y $b_i \in Act$.

Escribiremos $P \approx_{\mathcal{PB}} Q$ sii $\forall\, T \in \mathcal{PB} :\ pass(P\ ,\ T)\ =\ pass(Q\ ,\ T)$. □

Intuitivamente, una barba probabilística es una prueba que, o bien es una elección externa generalizada entre una serie de acciones, cuyas continuaciones son el proceso Nil, compuesta en elección externa con la acción ω, o es una elección externa generalizada entre una serie de acciones las cuales todas tienen como continuación el proceso Nil, menos una que tiene como continuación otra barba probabilística. En la Figura 4.2 aparece una representación gráfica de una barba probabilística tipo.

Teorema 4.4 Para cualesquiera procesos P y P' se tiene $P \approx_{\mathcal{G}} P' \iff P \approx_{\mathcal{PB}} P'$.



donde $\forall\, 1 \leq j \leq m : (\sum_{k=1}^{n_j} p_{jk} = 1) \;\wedge\; (i \neq k \Rightarrow b_{ji} \neq b_{jk})$.

Figura 4.2: Barbas Probabilísticas.

Demostración: Es una consecuencia inmediata del Teorema 4.1, que nos indica que dos procesos son equivalentes respecto de la semántica de pruebas sii lo son sus formas normales, y de la demostración del Teorema 4.2, en la cual vemos que para distinguir entre dos procesos en forma normal, es suficiente utilizar barbas probabilísticas. □

Otra consecuencia de las demostraciones de los Teoremas 4.2 y 4.4 es que la equivalencia inducida por la semántica de pruebas en el caso de que consideráramos factores de prenormalización sería la misma que cuando no los consideramos. Tal resultado se sigue del hecho de que si podemos distinguir dos procesos mediante una prueba, podemos encontrar una prueba que distinga a sus formas normales. En la demostración del Teorema 4.2 vimos que al efecto nos podemos restringir a la consideración de barbas probabilísticas, en las cuales siempre se ofrecen todas las acciones, de modo que las probabilidades con las que se pasan tales pruebas sería la misma aunque consideráramos factores de prenormalización. En consecuencia, si denotamos por $\approx_{\mathcal{G}'}$ a la relación de equivalencia inducida por la semántica de pruebas para el modelo generativo considerando factores de prenormalización, obtenemos el siguiente

Corolario 4.5 Para cualesquiera procesos P y P' se tiene $P \approx_{\mathcal{G}} P'$ sii $P \approx_{\mathcal{G}'} P'$.

□

4.4 Una Semántica Denotacional Completamente Abstracta

En esta sección presentaremos una semántica denotacional la cual es completamente abstracta con respecto a la semántica de pruebas para el modelo generativo. Esta interpretación estará basada en los *árboles de aceptación* [Hen85].

Comenzaremos definiendo el dominio semántico sobre el que vamos a trabajar, y tras ello definiremos las funciones semánticas correspondientes a los operadores sintácticos del lenguaje. Finalmente, mostraremos que nuestra semántica denotacional identifica a los mismos procesos que la semántica de pruebas para el modelo generativo.

4.4.1 Dominio Semántico

El dominio semántico sobre el que vamos a trabajar será el de los *árboles de aceptación probabilísticos* (*pat*)[8] *sobre Act*. Denotaremos a este dominio por $\mathbf{PAT}_{\mathrm{Act}}$. Sus elementos serán árboles con dos tipos de nodos: internos (etiquetados por $\oplus$) y externos (etiquetados por $+$), que además han de cumplir las siguientes condiciones:

- La raíz del árbol es un nodo interno.
- Para cada nodo interno del árbol, los arcos que salen a partir de él verifican las siguientes condiciones:
  - Cada uno de ellos está etiquetado con un estado (ver Definición 4.5) junto con una probabilidad, siendo los estados diferentes dos a dos.

[8]Siglas de *probabilistic acceptance trees*.

  - La suma de las probabilidades que etiquetan estos arcos es menor o igual que 1.
  - Conducen a nodos externos.

- Para cada nodo externo del árbol, los arcos que salen a partir de él deben cumplir las siguientes condiciones:
  - Están etiquetados con las acciones pertenecientes al estado que etiqueta el arco que llega a ellos.
  - Para cada acción perteneciente a cada estado existe un único arco etiquetado por la misma.
  - Conducen a nodos internos.

Usualmente denotaremos a los elementos de $\mathbf{PAT}_{\mathrm{Act}}$ por $R, R_1, \ldots$, y omitiremos el término *probabilísticos* al referirnos a los mismos.

Nótese que en un árbol de aceptación pueden coexistir (al igual que ocurría cuando definíamos los conjuntos de aceptación probabilísticos) varios arcos partiendo de un mismo nodo interno etiquetados por estados conteniendo el mismo conjunto de acciones, siempre que las correspondientes distribuciones de probabilidad definidas sobre dichos conjuntos de acciones sean distintas. Por ejemplo, bajo un nodo interno podemos encontrar arcos etiquetados por $\{(a, \frac{1}{2}), (b, \frac{1}{2})\}$ y por $\{(a, \frac{1}{3}), (b, \frac{2}{3})\}$. Nótese también que la suma de las probabilidades asociadas a los arcos que salen de cada nodo interno puede ser menor que uno. La diferencia entre dicha suma y uno indica la probabilidad de divergencia del proceso a partir de dicho punto.

En la Figura 4.3 presentamos diversos ejemplos de procesos junto con su árbol de aceptación. De forma similar a como hacíamos para los conjuntos de aceptación, caracterizaremos los nodos de estos árboles por medio de la secuencia de pares (estado, acción) que nos permite alcanzarlos. En la siguiente definición introducimos los formalismos para poder manejar los estados de un árbol de aceptación que se alcanzan tras una secuencia de pares.

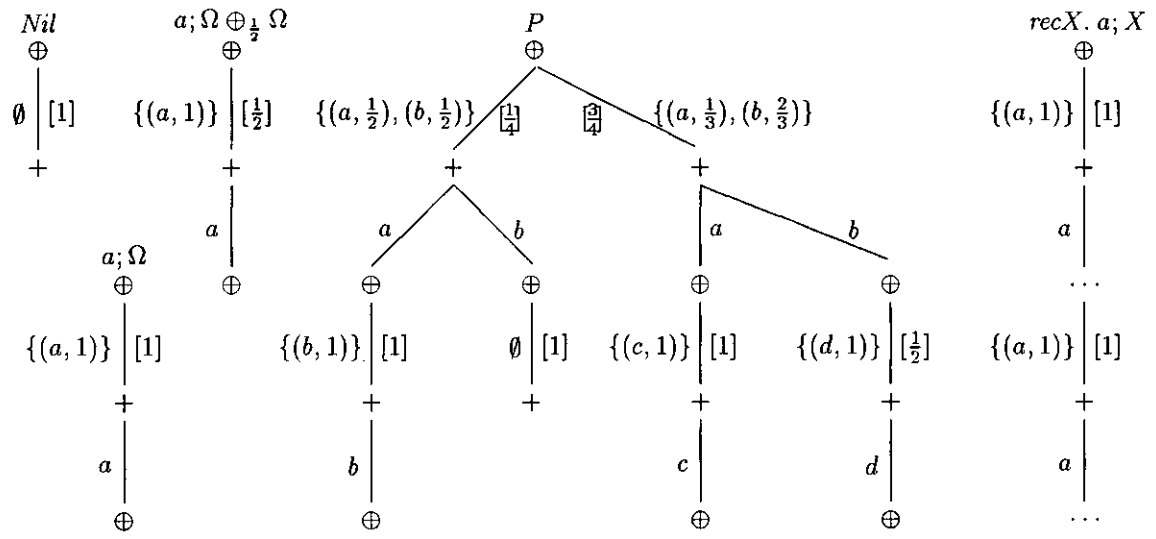


Figura 4.3: Ejemplos de árboles de aceptación probabilísticos.

Definición 4.21 Sea R un *pat* y A un estado. Definimos la *probabilidad* con la que R *alcanza* (inmediatamente) el estado A, y lo denotamos por $p(R, A)$, como p_A si existe un arco que salga de la raíz de R etiquetado por $[p_A]\ A$, y como cero si no existe ningún arco con tales características.

Siendo A un estado tal que $p_A = p(R, A) > 0$, con $a \in act(A)$, definimos la *continuación después de ejecutar la acción* a a *partir de* A, y lo denotamos por $R/(A, a)$, como el árbol cuya raíz es el nodo interno al que se llega mediante el arco etiquetado con a que parte del nodo externo al que llega el arco etiquetado por $[p_A]\ A$.

Sean A_i estados y $a_i \in act(A_i)$ con $i = 1 \ldots n$. Si consideramos la secuencia $s = \langle A_1\, a_1, A_2\, a_2, \ldots, A_n\, a_n \rangle$, definimos la *probabilidad global* con la cual R *alcanza el nodo externo representado por la secuencia* s *y el estado* A, y lo denotamos por $p(R, s, A)$, como

$$\begin{aligned} p(R, \epsilon, A) &= p(R, A) \\ p(R, \langle A_1 a_1 \rangle \circ s, A) &= p(R, A_1) \cdot p(R/(A_1, a_1), s, A) \end{aligned}$$

□

La Figura 4.4 da una visión gráfica de los conceptos introducidos en la definición anterior. Nótese que a partir de los valores de $p(R, s, A)$ podemos reconstruir fácilmente el árbol R. Al efecto, para determinar las probabilidades (locales) asociadas a los estados que etiquetan los arcos que parten de los nodos internos que se alcanzan tras una secuencia de pares (estado, acción) es suficiente dividir por la probabilidad global de alcanzar el último estado de la secuencia. Formalmente, si notamos por $\text{prob}(R, s, X)$ a la probabilidad que etiqueta el arco etiquetado por X que sale del nodo de R que se alcanza tras haber recorrido la secuencia s[9], obtenemos:

$$\text{prob}(R, s, X) = \begin{cases} p(R, X) & \text{si } s = \epsilon \\ \dfrac{p(R, s' \circ \langle A\, a\rangle, X)}{p(R, s', A)} & \text{si } s = s' \circ \langle A\, a\rangle \end{cases}$$

De hecho, en el resto de esta sección utilizaremos de forma implícita esta propiedad en diversas ocasiones a la hora de definir ciertos árboles de aceptación.

A continuación pasaremos a definir la relación de orden que consideramos entre los árboles de aceptación.

Definición 4.22 Sean R_1, R_2 *pat*'s. Escribiremos $R_1 \sqsubseteq_{\mathbf{PAT}} R_2$ sii para toda secuencia s y para todo estado A se cumple $p(R_1, s, A) \leq p(R_2, s, A)$. Escribiremos $R_1 =_{\mathbf{PAT}} R_2$ si $R_1 \sqsubseteq_{\mathbf{PAT}} R_2$ y $R_2 \sqsubseteq_{\mathbf{PAT}} R_1$. □

Esta relación es de orden, y da lugar a un orden parcial completo $(\mathbf{PAT}_{\text{Act}}, \sqsubseteq_{\mathbf{PAT}})$.

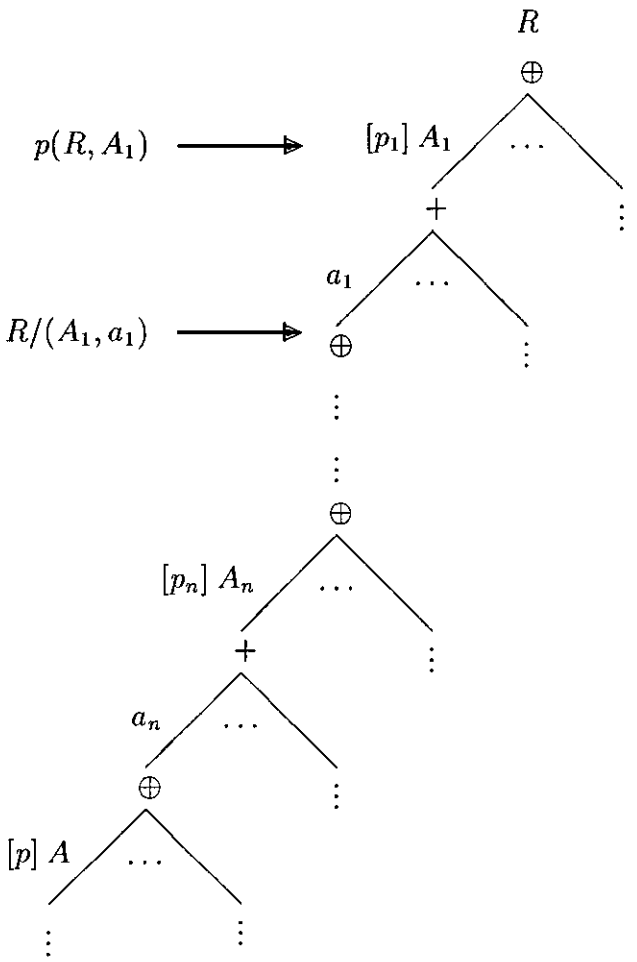
Proposición 4.23 La relación $\sqsubseteq_{\mathbf{PAT}}$ es una relación de orden.

Demostración: Al igual que ocurría en la Proposición 3.14, es trivial comprobar que la relación $\sqsubseteq_{\mathbf{PAT}}$ cumple las propiedades reflexiva, antisimétrica y transitiva. □

Teorema 4.6 El par $(\mathbf{PAT}_{\text{Act}}, \sqsubseteq_{\mathbf{PAT}})$ es un *orden parcial completo* (cpo).

Demostración: Hemos de probar la existencia de un elemento mínimo y la existencia de cota superior mínima para cada cadena.

[9] Es decir, si $s = \langle A_1\, a_1, A_2\, a_2, \ldots, A_n\, a_n\rangle$, consideramos el árbol $R/(A_1, a_1)/(A_2, a_2)\cdots/(A_n, a_n)$.



$$p(R, \langle A_1\ a_1, A_2\ a_2, \ldots, A_n\ a_n \rangle, A) = p_1 \cdot p_2 \cdots p_n \cdot p$$

Figura 4.4: Definición de $p(R, A)$, $R/(A, a)$ y $p(R, s, A)$.

Existencia de elemento mínimo

Veamos que el árbol R definido en la forma $\forall\ s, A:\ p(R, s, A) = 0$, que pertenece a $\mathbf{PAT}_{\mathrm{Act}}$, es el elemento mínimo. En efecto, si $R' \in \mathbf{PAT}_{\mathrm{Act}}$ se tiene

$$\forall s, A:\ p(R, s, A) = 0 \leq p(R', s, A)$$

con lo cual tenemos $R \sqsubseteq_{\mathbf{PAT}} R'$.

Existencia de cota superior mínima (lub)

Sea $\{R_n\}_{n\in\mathbb{N}}$ una *cadena* de elementos de $\mathbf{PAT}_{\mathrm{Act}}$. Entonces, el elemento $\sqcup R_n$ será el dado por

$$p(\sqcup R_n, s, A) = \lim_{n\in\mathbb{N}} p(R_n, s, A)$$

Veamos que $\sqcup R_n$ queda así bien definido. Al formar los elementos R_n una cadena, los valores $p(R_n, s, A)$ forman una sucesión creciente y acotada por 1. Por lo tanto, existe el límite de esta sucesión, que será menor o igual a 1. Además, por definición de límite, si existiese algún nodo interno en el que las probabilidades sumaran $1 + \epsilon$ (con $\epsilon > 0$), debería existir un $j \in \mathbb{N}$ tal que en ese mismo nodo las probabilidades sumaran $1 + \delta$, con $0 < \delta \leq \epsilon$, lo cual no es posible dado que los elementos de la cadena son árboles. Por tanto, $\sqcup R_n$ está bien definido.

Una vez definido, tenemos que ver que se trata de la cota superior mínima de la cadena:

1. $\sqcup R_n$ *es cota superior de la cadena.*

   Sea R_j un elemento de la cadena. Tenemos que mostrar que para cualquier secuencia s y cualquier estado A se cumple $p(R_j, s, A) \leq p(\sqcup R_n, s, A)$, lo cual es trivial dado que $p(\sqcup R_n, s, A) = \lim_{n\in\mathbb{N}} p(R_n, s, A) \geq p(R_j, s, A)$.

2. $\sqcup R_n$ *es la mínima cota superior.*

   Sea R un elemento de $\mathbf{PAT}_{\mathrm{Act}}$ tal que $\forall\ n \in \mathbb{N}:\ R_n \sqsubseteq_{\mathbf{PAT}} R$. Entonces, tenemos la siguiente cadena de implicaciones:

$$
\begin{array}{ccccc}
R_n & \sqsubseteq_{\mathbf{PAT}} & R & \forall\, n \in \mathbb{N} \\
& \Downarrow & & \\
p(R_n, s, A) & \leq & p(R, s, A) & \forall s, A \ \wedge \ \forall\, n \in \mathbb{N} \\
& \Downarrow & & \\
\lim_{n\in\mathbb{N}}\ p(R_n, s, A) & \leq & p(R, s, A) & \forall s, A \\
& \Downarrow & & \\
p(\sqcup R_n, s, A) & \leq & p(R, s, A) & \forall s, A \\
& \Downarrow & & \\
\sqcup R_n & \sqsubseteq_{\mathbf{PAT}} & R &
\end{array}
$$

□

4.4.2 Funciones Semánticas

En esta sección vamos a definir la función semántica asociada a cada uno de los operadores de nuestro lenguaje. Además, mostraremos la monotonía y continuidad de dichas funciones, lo que nos permite aplicar técnicas de punto fijo a la hora de dar semántica a procesos recursivos. Como es usual, denotaremos por $[\![P]\!]$ a la semántica del proceso sintáctico P.

Nil y Ω

Nil tiene un único estado alcanzable (inmediatamente): el estado $\emptyset$, que se alcanza con probabilidad 1. Por lo tanto $[\![Nil]\!]$ será es un árbol con un nodo interno, del cual sale un único arco etiquetado con el estado vacío y probabilidad 1; el mismo va a parar a un nodo externo del que no salen arcos.

$$p([\![Nil]\!], s, A) = \begin{cases} 1 & \text{si } \ s = \epsilon \ \wedge \ A = \emptyset \\ 0 & \text{e.o.c.} \end{cases}$$

Ω no tiene estados alcanzables. Por lo tanto $[\![\Omega]\!]$ es un árbol con un único nodo interno, del cual no parte ningún arco.

$$p([\![\Omega]\!], s, A) = 0 \qquad \forall\, s, A$$

Prefijo

Para cada $a \in Act$ definimos la función semántica $a;\_ :: \mathbf{PAT}_{\mathrm{Act}} \longrightarrow \mathbf{PAT}_{\mathrm{Act}}$. El árbol de aceptación $a; R$ es igual al árbol R pero precedido por un nodo interno y un nodo externo los cuales corresponden a la aparición de la acción a. Es decir, tendrá como raíz un nodo interno del que parte un único arco etiquetado con el estado $\{(a,1)\}$ y con probabilidad asociada 1. Este arco llega a un nodo externo del que sale un único arco etiquetado con a que va a parar a la raíz de R.

$$p(a;R,s,A) = \begin{cases} 1 & \text{si } s = \epsilon \wedge A = \{(a,1)\} \\ p(R,s',A) & \text{si } s = \langle \{(a,1)\}\, a\rangle \circ s' \\ 0 & \text{e.o.c.} \end{cases}$$

Proposición 4.24 Para cada $a \in Act$, la función $a;\_ :: \mathbf{PAT}_{\mathrm{Act}} \longrightarrow \mathbf{PAT}_{\mathrm{Act}}$ es monótona y continua.

Monotonía.

Sean $R, R' \in \mathbf{PAT}_{\mathrm{Act}}$ tales que $R \sqsubseteq_{\mathbf{PAT}} R'$. Tenemos que probar $a; R \sqsubseteq_{\mathbf{PAT}} a; R'$, o lo que es lo mismo, que $\forall\, s, A :\ p(R,s,A) \leq p(R',s,A)$. Distinguiremos tres casos:

- $s = \epsilon \wedge A = \{(a,1)\}$. En este caso tenemos $p(a;R,\epsilon,A) = 1 \leq 1 = p(a;R',\epsilon,A)$.
- $s = \langle \{(a,1)\}\, a\rangle \circ s'$. En este caso, para cada estado A se cumple

$$p(a;R,s,A) = p(R,s',A) \leq p(R',s',A) = p(a;R',s,A)$$

- Para el resto de las secuencias la desigualdad se cumple trivialmente dado que

$$p(a;R,s,A) = 0 \leq 0 = p(a;R',s,A)$$

Continuidad.

Sea $\{R_n\}_{n\in\mathbb{N}}$ una cadena de elementos de $\mathbf{PAT}_{\mathrm{Act}}$. Debido a la monotonía del operador prefijo, tenemos que los árboles $\{a; R_n\}_{n\in\mathbb{N}}$ también forman una cadena. Tenemos que probar que para toda secuencia s y para todo estado A, se cumple

$$p(a; \sqcup\{R_n\}_{n\in\mathbb{N}}, s, A) = p(\sqcup\{a; R_n\}_{n\in\mathbb{N}}, s, A)$$

lo cual se sigue de la siguiente cadena de igualdades

$$p(a; \sqcup\{R_n\}_{n\in\mathbb{N}}, s, A) = \begin{cases} 1 & \text{si } s = \epsilon \wedge A = \{(a,1)\} \\ p(\sqcup\{R_n\}_{n\in\mathbb{N}}, s', A) & \text{si } s = \langle\{(a,1)\}\, a\rangle \circ s' \\ 0 & \text{e.o.c.} \end{cases} =$$

$$\begin{cases} 1 & \text{si } s = \epsilon \wedge A = \{(a,1)\} \\ \lim\limits_{n\in\mathbb{N}} p(R_n, s', A) & \text{si } s = \langle\{(a,1)\}\, a\rangle \circ s' \\ 0 & \text{e.o.c.} \end{cases} =$$

$$\lim_{n\in\mathbb{N}} \begin{cases} 1 & \text{si } s = \epsilon \wedge A = \{(a,1)\} \\ p(R_n, s', A) & \text{si } s = \langle\{(a,1)\}\, a\rangle \circ s' \\ 0 & \text{e.o.c.} \end{cases} =$$

$$\lim_{n\in\mathbb{N}} p(a; R_n, s, A) = p(\sqcup\{a; R_n\}_{n\in\mathbb{N}}, s, A)$$

□

Elección Interna

Dado un valor $p \in (0,1)$, la función $\_ \oplus_p \_ :: \mathbf{PAT}_{\mathrm{Act}} \times \mathbf{PAT}_{\mathrm{Act}} \longrightarrow \mathbf{PAT}_{\mathrm{Act}}$ devuelve un árbol de aceptación, el cual viene dado por la unión ponderada de los árboles R_1 y R_2, considerando la probabilidad p. En términos de estados alcanzables tras una secuencia, un estado será alcanzable en el nuevo proceso tras una secuencia si lo es en alguno de los dos argumentos. En tal caso, la probabilidad con la cual era alcanzable el estado en el primer (resp. segundo) argumento se multiplica por el valor p (resp. $1-p$), obteniéndose así dos sumandos que se sumarán para obtener la correspondiente probabilidad en el árbol $R_1 \oplus_p R_2$.

$$p(R_1 \oplus_p R_2, s, A) = p \cdot p(R_1, s, A) + (1-p) \cdot p(R_2, s, A)$$

Proposición 4.25 Las funciones $\_ \oplus_p \_ :: \mathbf{PAT}_{\mathrm{Act}} \times \mathbf{PAT}_{\mathrm{Act}} \longrightarrow \mathbf{PAT}_{\mathrm{Act}}$ son monótonas y continuas en sus dos argumentos, para cualquier $0 < p < 1$.

Demostración: Por simetría es suficiente hacer la demostración para uno de los argumentos. En nuestro caso, lo haremos para el primero.

Monotonía.

Sean $R_1, R_2 \in \mathbf{PAT}_{\mathrm{Act}}$ tales que $R_1 \sqsubseteq_{\mathbf{PAT}} R_2$. Tenemos que probar que se cumple $R_1 \oplus_p R \sqsubseteq_{\mathbf{PAT}} R_2 \oplus_p R$, para cualquier $R \in \mathbf{PAT}_{\mathrm{Act}}$, o lo que es lo mismo, que para cualquier secuencia s y para cualquier estado A se tiene

$$p(R_1 \oplus_p R, s, A) \leq p(R_2 \oplus_p R, s, A)$$

Pero ello es inmediato, dado que

$$\begin{aligned} p(R_1 \oplus_p R, s, A) &= p \cdot p(R_1, s, A) + (1-p) \cdot p(R, s, A) \\ &\leq p \cdot p(R_2, s, A) + (1-p) \cdot p(R, s, A) \\ &= p(R_2 \oplus_p R, s, A) \end{aligned}$$

Continuidad.

Sea $\{R_n\}_{n \in \mathbb{N}}$ una cadena de elementos de $\mathbf{PAT}_{\mathrm{Act}}$. Por la monotonía de la elección interna, tenemos que los árboles $\{R_n \oplus_p R\}_{n \in \mathbb{N}}$ también forman una cadena. Tenemos que probar que para toda secuencia s y para todo estado A, se cumple

$$p(\sqcup\{R_n\}_{n \in \mathbb{N}} \oplus_p R, s, A) = p(\sqcup\{R_n \oplus_p R\}_{n \in \mathbb{N}}, s, A)$$

lo cual se sigue de la siguiente cadena de igualdades

$$\begin{aligned} p(\sqcup\{R_n\}_{n \in \mathbb{N}} \oplus_p R, s, A) &= p \cdot p(\sqcup\{R_n\}_{n \in \mathbb{N}}, s, A) + (1-p) \cdot \mathrm{prob}(R, s, A) \\ &= p \cdot (\lim_{n \in \mathbb{N}} p(R_n, s, A)) + (1-p) \cdot p(R, s, A) \\ &= \lim_{n \in \mathbb{N}} (p \cdot p(R_n, s, A) + (1-p) \cdot p(R, s, A)) \\ &= \lim_{n \in \mathbb{N}} p(R_n \oplus_p R, s, A) = p(\sqcup\{R_n \oplus_p R\}_{n \in \mathbb{N}}, s, A) \end{aligned}$$

□

Elección Externa

Dado un valor $p \in (0,1)$, la función $\_ +_p \_ :: \mathbf{PAT}_{\mathrm{Act}} \times \mathbf{PAT}_{\mathrm{Act}} \longrightarrow \mathbf{PAT}_{\mathrm{Act}}$ devuelve un árbol de aceptación, el cual representa la *unión externa* ponderada de los árboles R_1 y R_2 respecto de la probabilidad p.

Antes de definir la función semántica asociada al operador $+_p$, definiremos un operador auxiliar que nos permitirá unir dos estados de acuerdo a una cierta probabilidad. Si uno de los estados es vacío, el resultado será el otro estado. Por otra parte, si ninguno es vacío, el resultado será un nuevo estado que contiene como acciones la unión de los conjuntos de acciones de los dos estados, pero donde la probabilidad que cada acción tenía en su estado queda multiplicada por el factor de ponderación asociado.

Definición 4.26 Sean X, Y estados, y $p \in (0,1)$. Definimos la *unión* de los estados X y Y con *probabilidad asociada* p como

$$X \cup_p Y = \begin{cases} X & \text{si } Y = \emptyset \\ Y & \text{si } X = \emptyset \\ \{(a, p \cdot pro(a,X) + (1-p) \cdot pro(a,Y))\} & \text{e.o.c.} \end{cases}$$

□

Como en el caso no probabilístico, a la hora de definir el árbol que se obtiene al componer dos árboles con la función semántica asociada a la elección externa, tendremos que distinguir entre el caso correspondiente a la raíz del nuevo árbol y las continuaciones bajo ella.

• *Descendencia inmediata de la raíz*

Para definir los arcos bajo la raíz del nuevo árbol consideraremos la unión de los estados (iniciales) de los dos árboles que estamos componiendo en elección externa.

$$p(R_1 +_p R_2, \epsilon, A) = \sum_{A = B \cup_p C} p(R_1, \epsilon, B) \cdot p(R_2, \epsilon, C)$$

De modo que de la raíz del nuevo árbol sale un arco etiquetado con el estado A si existen sendos arcos, uno etiquetado con el estado B que sale de la raíz del árbol R_1 y otro etiquetado con el estado C que sale de la raíz del árbol R_2, con $A = B \cup_p C$. La probabilidad que etiqueta dicho arco será igual a la suma de los productos de las probabilidades que etiquetan los arcos correspondientes a cada par de tales estados.

Como consecuencia inmediata de esta definición, tenemos que la función que corresponde a la elección externa será estricta en sus dos argumentos, dado que si un argumento es el árbol asociado a Ω, entonces el resultado coincidirá con $[\![\Omega]\!]$, dado que de la raíz de $[\![\Omega]\!]$ no parte *ningún* arco, es decir $\forall s, A : p([\![\Omega]\!], s, A) = 0$, de modo que

$$[\![P +_p \Omega]\!] = [\![\Omega +_p P]\!] = [\![\Omega]\!] \qquad (\forall\, 0 < p < 1)$$

Además, *Nil* es elemento neutro de esta función:

$$[\![P +_p Nil]\!] = [\![Nil +_p P]\!] = [\![P]\!] \qquad (\forall\, 0 < p < 1)$$

Vamos a presentar una serie de ejemplos que ilustren esta definición.

Ejemplo 4.27 Consideremos los siguientes procesos:

- $P_1 = (a +_{\frac{1}{2}} b) \oplus_{\frac{1}{3}} a$
- $P_2 = b \oplus_{\frac{1}{4}} Nil$
- $P_3 = b \oplus_{\frac{1}{4}} \Omega$
- $P_4 = a \oplus_{\frac{1}{2}} b$

Para cada uno de ellos tenemos:

$$\begin{cases} p([\![P_1]\!], \epsilon, C) = \frac{1}{3} \\ p([\![P_1]\!], \epsilon, A_1) = \frac{2}{3} \\ p([\![P_1]\!], \epsilon, X) = 0 \text{ si } X \neq C, A_1 \end{cases} \qquad \begin{cases} p([\![P_2]\!], \epsilon, B_1) = \frac{1}{4} \\ p([\![P_2]\!], \epsilon, \emptyset) = \frac{3}{4} \\ p([\![P_2]\!], \epsilon, X) = 0 \text{ si } X \neq B_1, \emptyset \end{cases}$$

$$\begin{cases} p([\![P_3]\!], \epsilon, B_1) = \frac{1}{4} \\ p([\![P_3]\!], \epsilon, X) = 0 \text{ si } X \neq B_1 \end{cases} \qquad \begin{cases} p([\![P_4]\!], \epsilon, A_1) = \frac{1}{2} \\ p([\![P_4]\!], \epsilon, B_1) = \frac{1}{2} \\ p([\![P_4]\!], \epsilon, X) = 0 \text{ si } X \neq A_1, B_1 \end{cases}$$

donde $A_1 = \{(a, 1)\}, B_1 = \{(b, 1)\}$ y $C = \{(a, \frac{1}{2}), (b, \frac{1}{2})\}$. Veamos como serían los primeros pisos de algunas de sus composiciones en elección externa:

- La descendencia de la raíz del árbol $R_1 = [\![P_1]\!] +_{\frac{1}{2}} [\![P_2]\!]$ quedaría definida por:

$p(R_1, \epsilon, A_1) = \frac{2}{3} \cdot \frac{3}{4} = \frac{1}{2}$ (unir A_1 de P_1 con $\emptyset$ de P_2)

$p(R_1, \epsilon, \{(a, \frac{1}{4}), (b, \frac{3}{4})\}) = \frac{1}{3} \cdot \frac{1}{4} = \frac{1}{12}$ (unir C de P_1 con B_1 de P_2)

$p(R_1, \epsilon, C) = \frac{1}{3} \cdot \frac{3}{4} + \frac{2}{3} \cdot \frac{1}{4} = \frac{5}{12}$ (unir C de P_1 con $\emptyset$ de P_2 y A_1 de P_1 con B_1 de P_2)

$p(R_1, \epsilon, X) = 0$ si $X \neq A_1, \{(a, \frac{1}{4}), (b, \frac{3}{4})\}, C$

- La descendencia de la raíz del árbol $R_2 = [\![P_1]\!] +_{\frac{1}{3}} [\![P_3]\!]$ quedaría definida por:

$p(R_2, \epsilon, \{(a, \frac{1}{6}), (b, \frac{5}{6})\}) = \frac{1}{3} \cdot \frac{1}{4} = \frac{1}{12}$ (unir C de P_1 con B_1 de P_3)

$p(R_2, \epsilon, \{(a, \frac{1}{3}), (b, \frac{2}{3})\}) = \frac{2}{3} \cdot \frac{1}{4} = \frac{1}{6}$ (unir A_1 de P_1 con B_1 de P_3)

$p(R_2, \epsilon, X) = 0$ si $X \neq \{(a, \frac{1}{6}), (b, \frac{5}{6})\}, \{(a, \frac{1}{3}), (b, \frac{2}{3})\}$

- La descendencia de la raíz del árbol $R_3 = [\![P_2]\!] +_{\frac{1}{2}} [\![P_3]\!]$ quedaría definida por:

$p(R_3, \epsilon, B_1) = \frac{1}{4} \cdot \frac{1}{4} + \frac{3}{4} \cdot \frac{1}{4} = \frac{1}{4}$ (unir B_1 de P_2 con B_1 de P_3 y $\emptyset$ de P_2 con B_1 de P_3)

$p(R_3, \epsilon, X) = 0$ si $X \neq B_1$

Nótese que en este caso el árbol obtenido R_3 será independiente de la probabilidad que consideremos en la elección externa.

- La descendencia de la raíz del árbol $R_4 = [\![P_3]\!] +_{\frac{1}{2}} [\![P_4]\!]$ quedaría definida por:

$p(R_4, \epsilon, C) = \frac{1}{4} \cdot \frac{1}{2} = \frac{1}{8}$ (unir B_1 de P_3 con A_1 de P_4)

$p(R_4, \epsilon, B_1) = \frac{1}{4} \cdot \frac{1}{2} = \frac{1}{8}$ (unir B_1 de P_3 con B_1 de P_4)

$p(R_4, \epsilon, X) = 0$ si $X \neq C, B_1$

□

• *Continuaciones*

Veamos ahora como definiremos el resto del árbol resultado de la composición, es decir como obtendremos $p(R_1 +_p R_2, s, X)$ para cada $s \neq \epsilon$ a partir de los árboles R_1 y R_2. Al efecto estudiaremos como se puede construir el primer estado que aparece en la secuencia s a partir de los estados iniciales de R_1 y R_2, y dependiendo en cada caso de qué estado (el de R_1, el de R_2, o ambos) contenga la acción asociada al primer estado de la secuencia s, la continuación se tomará de R_1, de R_2 o se obtendrá combinando ambas mediante una *elección interna* en la cual intervendrán el parámetro asociado a la elección externa y las respectivas probabilidades asociadas a la acción en los estados combinados de R_1 y R_2. En definitiva obtenemos

$$\begin{aligned} p(R_1 +_p R_2, \langle A\, a\rangle \circ s', X) \;=\; & \sum_{A=B\cup_p C} \frac{p\cdot pro(a,B)}{p\cdot pro(a,B)+(1-p)\cdot pro(a,C)} \cdot p(R_1, s_B, X)\cdot p(R_2, C) \\ & + \sum_{A=B\cup_p C} \frac{(1-p)\cdot pro(a,C)}{p\cdot pro(a,B)+(1-p)\cdot pro(a,C)} \cdot p(R_2, s_C, X)\cdot p(R_1, B) \end{aligned}$$

donde $s_B = \langle B\, a\rangle \circ s'$ y $s_C = \langle C\, a\rangle \circ s'$.

La definición queda ilustrada por el siguiente

Ejemplo 4.28 Consideremos los procesos:

$$\bullet\ P_1 = ((a; Q_1) +_{\frac{1}{2}} (b; Q_1')) \oplus_{\frac{1}{3}} (a; Q_2) \qquad \bullet\ P_2 = (b; Q_2') \oplus_{\frac{1}{4}} Nil$$

En el Ejemplo 4.27 vimos que los arcos que parten de la raíz de $R_1 = [\![P_1]\!] +_{\frac{1}{2}} [\![P_2]\!]$ están etiquetados con los estados $A_1 = \{(a,1)\}$, $A_2 = \{(a,\frac{1}{4}),(b,\frac{3}{4})\}$ y $A_3 = \{(a,\frac{1}{2}),(b,\frac{1}{2})\}$, siendo las probabilidades asociadas $\frac{1}{2}$, $\frac{1}{12}$ y $\frac{5}{12}$ respectivamente. Veamos ahora como será el resto del árbol.

Obviamente, para cualquier secuencia que empiece por un estado distinto de los indicados anteriormente la probabilidad de alcanzar cualquier estado sería 0. Para el resto de las secuencias tenemos:

$$p(R_1, \langle A_1\, a\rangle \circ s', X) = \frac{\frac{1}{2}\cdot 1}{\frac{1}{2}\cdot 1+\frac{1}{2}\cdot 0} \cdot \frac{2}{3} \cdot p(Q_2, s', X) \cdot \frac{3}{4} = \frac{1}{2}\cdot p(Q_2, s', X)$$

$$p(R_1, \langle A_2\, a\rangle \circ s', X) = \frac{\frac{1}{2}\cdot\frac{1}{2}}{\frac{1}{2}\cdot\frac{1}{2}+\frac{1}{2}\cdot 0}\cdot\frac{1}{3}\cdot p(Q_1, s', X)\cdot\frac{1}{4} = \frac{1}{12}\cdot p(Q_1, s', X)$$

$$\begin{aligned} p(R_1, \langle A_2\, b\rangle \circ s', X) &= \frac{\frac{1}{2}\cdot\frac{1}{2}}{\frac{1}{2}\cdot\frac{1}{2}+\frac{1}{2}\cdot 1}\cdot\frac{1}{3}\cdot p(Q'_1, s', X)\cdot\frac{1}{4} + \frac{\frac{1}{2}\cdot 1}{\frac{1}{2}\cdot\frac{1}{2}+\frac{1}{2}\cdot 1}\cdot\frac{1}{4}\cdot p(Q'_2, s', X)\cdot\frac{1}{3} \\ &= \frac{1}{36}\cdot p(Q'_1, s', X) + \frac{1}{18}\cdot p(Q'_2, s', X) \end{aligned}$$

$$\begin{aligned} p(R_1, \langle A_3\, a\rangle \circ s', X) &= \frac{\frac{1}{2}\cdot\frac{1}{2}}{\frac{1}{2}\cdot\frac{1}{2}+\frac{1}{2}\cdot 0}\cdot\frac{1}{3}\cdot p(Q_1, s', X)\cdot\frac{3}{4} + \frac{\frac{1}{2}\cdot 1}{\frac{1}{2}\cdot 1+\frac{1}{2}\cdot 0}\cdot\frac{2}{3}\cdot p(Q_2, s', X)\cdot\frac{1}{4} \\ &= \frac{1}{4}\cdot p(Q_1, s', X) + \frac{1}{6}\cdot p(Q_2, s', X) \end{aligned}$$

$$\begin{aligned} p(R_1, \langle A_3\, b\rangle \circ s', X) &= \frac{\frac{1}{2}\cdot\frac{1}{2}}{\frac{1}{2}\cdot\frac{1}{2}+\frac{1}{2}\cdot 0}\cdot\frac{1}{3}\cdot p(Q'_1, s', X)\cdot\frac{3}{4} + \frac{\frac{1}{2}\cdot 1}{\frac{1}{2}\cdot 0+\frac{1}{2}\cdot 1}\cdot\frac{1}{4}\cdot p(Q'_2, s', X)\cdot\frac{2}{3} \\ &= \frac{1}{4}\cdot p(Q'_1, s', X) + \frac{1}{6}\cdot p(Q'_2, s', X) \end{aligned}$$

□

Proposición 4.29 Las funciones $\_ +_p \_ :: \mathbf{PAT}_{\mathrm{Act}} \times \mathbf{PAT}_{\mathrm{Act}} \longrightarrow \mathbf{PAT}_{\mathrm{Act}}$ son monótonas y continuas en sus dos argumentos, para cualquier $0 < p < 1$.

Demostración: Como de costumbre, es suficiente hacer la demostración para el primer argumento.

Monotonía.

Sean $R_1, R_2 \in \mathbf{PAT}_{\mathrm{Act}}$ tales que $R_1 \sqsubseteq_{\mathbf{PAT}} R_2$. Tenemos que probar que para cualquier $R \in \mathbf{PAT}_{\mathrm{Act}}$ se cumple $R_1 +_p R \sqsubseteq_{\mathbf{PAT}} R_2 +_p R$, o lo que es lo mismo, que para cualquier secuencia s y cualquier estado A se cumple:

$$p(R_1 +_p R, s, A) \leq p(R_2 +_p R, s, A)$$

La demostración la haremos en dos pasos. Primero lo probaremos para la secuencia vacía, y después para el resto de las secuencias.

- Secuencia vacía ($s = \epsilon$).

$$\begin{aligned} p(R_1 +_p R, \epsilon, A) &= \sum_{A = B \cup_p C} p(R_1, \epsilon, B)\cdot p(R, \epsilon, C) \\ &\leq \sum_{A = B \cup_p C} p(R_2, \epsilon, B)\cdot p(R, \epsilon, C) \\ &= p(R_2 +_p R, \epsilon, A) \end{aligned}$$

- Secuencia no vacía ($s = \langle A\, a\rangle \circ s'$, $s_B = \langle B\, a\rangle \circ s'$ y $s_C = \langle C\, a\rangle \circ s'$).

$$\begin{aligned}
p(R_1 +_p R, s, X) &= \sum_{A=B\cup_p C} \frac{p\cdot pro(a,B)}{p\cdot pro(a,B)+(1-p)\cdot pro(a,C)} \cdot p(R_1, s_B, X)\cdot p(R, C) \\
&+ \sum_{A=B\cup_p C} \frac{(1-p)\cdot pro(a,C)}{p\cdot pro(a,B)+(1-p)\cdot pro(a,C)} \cdot p(R, s_C, X)\cdot p(R_1, B) \\
&\leq \sum_{A=B\cup_p C} \frac{p\cdot pro(a,B)}{p\cdot pro(a,B)+(1-p)\cdot pro(a,C)} \cdot p(R_2, s_B, X)\cdot p(R, C) \\
&+ \sum_{A=B\cup_p C} \frac{(1-p)\cdot pro(a,C)}{p\cdot pro(a,B)+(1-p)\cdot pro(a,C)} \cdot p(R, s_C, X)\cdot p(R_2, B) \\
&= p(R_2 +_p R, s, X)
\end{aligned}$$

Continuidad.

Sea $\{R_n\}_{n\in\mathbb{N}}$ una cadena de elementos de $\mathbf{PAT}_{\mathrm{Act}}$. Por la monotonía de la elección externa, tenemos que $\{R_n +_p R\}_{n\in\mathbb{N}}$ también forman una cadena. Tenemos que probar que para toda secuencia s y para todo estado A, se cumple

$$p(\sqcup\{R_n\}_{n\in\mathbb{N}} +_p R, s, A) = p(\sqcup\{R_n +_p R\}_{n\in\mathbb{N}}, s, A)$$

Al igual que antes, haremos la demostración en dos pasos:

- Secuencia vacía ($s = \epsilon$).

$$\begin{aligned}
p(\sqcup\{R_n\}_{n\in\mathbb{N}} +_p R, \epsilon, A) &= \sum_{A=B\cup_p C} p(\sqcup\{R_n\}_{n\in\mathbb{N}}, \epsilon, B)\cdot p(R, \epsilon, C) \\
&= \sum_{A=B\cup_p C} (\lim_{n\in\mathbb{N}} p(R_n, \epsilon, B))\cdot p(R, \epsilon, C) \\
&= \lim_{n\in\mathbb{N}} \sum_{A=B\cup_p C} p(R_n, \epsilon, B)\cdot p(R, \epsilon, C) \\
&= \lim_{n\in\mathbb{N}} p(R_n +_p R, \epsilon, A) = p(\sqcup\{R_n +_p R\}_{n\in\mathbb{N}}, \epsilon, A)
\end{aligned}$$

- Secuencia no vacía ($s = \langle A\, a\rangle \circ s'$, $s_B = \langle B\, a\rangle \circ s'$ y $s_C = \langle C\, a\rangle \circ s'$).

$p(\sqcup\{R_n\}_{n\in\mathbb{N}} +_p R, s, X) =$

$$= \sum_{A=B\cup_p C} \frac{p\cdot pro(a,B)}{p\cdot pro(a,B)+(1-p)\cdot pro(a,C)} \cdot p(\sqcup\{R_n\}_{n\in\mathbb{N}}, s_B, X)\cdot p(R, C)$$
$$+ \sum_{A=B\cup_p C} \frac{(1-p)\cdot pro(a,C)}{p\cdot pro(a,B)+(1-p)\cdot pro(a,C)} \cdot p(R, s_C, X)\cdot p(\sqcup\{R_n\}_{n\in\mathbb{N}}, B)$$

$$= \sum_{A=B\cup_p C} \frac{p\cdot pro(a,B)}{p\cdot pro(a,B)+(1-p)\cdot pro(a,C)} \cdot (\lim_{n\in\mathbb{N}} p(R_n, s_B, X))\cdot p(R, C)$$
$$+ \sum_{A=B\cup_p C} \frac{(1-p)\cdot pro(a,C)}{p\cdot pro(a,B)+(1-p)\cdot pro(a,C)} \cdot p(R, s_C, X)\cdot (\lim_{n\in\mathbb{N}} p(R_n, B))$$

$$= \lim_{n\in\mathbb{N}} \sum_{A=B\cup_p C} \frac{p\cdot pro(a,B)}{p\cdot pro(a,B)+(1-p)\cdot pro(a,C)} \cdot p(R_n, s_B, X)\cdot p(R, C)$$
$$+ \lim_{n\in\mathbb{N}} \sum_{A=B\cup_p C} \frac{(1-p)\cdot pro(a,C)}{p\cdot pro(a,B)+(1-p)\cdot pro(a,C)} \cdot p(R, s_C, X)\cdot p(R_n, B)$$

$$= \lim_{n\in\mathbb{N}} \left(\begin{array}{l} \displaystyle\sum_{A=B\cup_p C} \frac{p\cdot pro(a,B)}{p\cdot pro(a,B)+(1-p)\cdot pro(a,C)} \cdot p(R_n, s_B, X)\cdot p(R, C) \\ + \displaystyle\sum_{A=B\cup_p C} \frac{(1-p)\cdot pro(a,C)}{p\cdot pro(a,B)+(1-p)\cdot pro(a,C)} \cdot p(R, s_C, X)\cdot p(R_n, B) \end{array} \right)$$

$$= \lim_{n\in\mathbb{N}} p(R_n +_p R, s, A) = p(\sqcup\{R_n +_p R\}_{n\in\mathbb{N}}, s, A)$$

□

Recursión

Como es usual cuando se define una semántica denotacional, el significado de los procesos recursivos definidos por expresiones de la forma $recX.P(X)$ se obtiene como el límite de las aproximaciones finitas de la forma

$$P_0 = \Omega, P_1 = P(\Omega), \ldots, P_n = P^n(\Omega)$$

Dado que todos los operadores incluidos en el lenguaje son continuos, este límite es el menor punto fijo de la ecuación $X = P(X)$. Es decir, definimos

$$[\![recX.\ P(X)]\!] = \bigsqcup_{n=0}^{\infty} [\![P_n]\!]$$

4.4.3 Teorema de Abstracción Completa

Una vez que hemos definido operadores semánticos para todos los operadores sintácticos de nuestro lenguaje, vamos a demostrar que la semántica denotacional es equivalente a la semántica de pruebas para el modelo generativo. En lugar de hacer la demostración directamente, probaremos que la semántica denotacional es equivalente a la caracterización basada en conjuntos de aceptación que presentamos en la Sección 4.2. Una vez que tengamos probada dicha equivalencia, por aplicación del Corolario 4.3, obtendremos que las dos semánticas, denotacional y de pruebas, son equivalentes.

Como hicimos para el modelo reactivo, probaremos primero el resultado para procesos finitos (no recursivos), y luego extenderemos el mismo a procesos recursivos, utilizando un lema análogo al Lema 3.17 que nos relacione el comportamiento operacional de dichos procesos con los de sus aproximaciones sintácticas finitas.

Lema 4.30 Para todo proceso *finito* P y toda secuencia s se tiene

$$p([\![P]\!], s, A) = \sum_{P'} \{\!|\, p_i \mid P \stackrel{s}{\Longrightarrow}_{p_i} P' \ \wedge\ S(P') = A \,|\!\}$$

Demostración: La demostración la haremos por inducción estructural y anidadamente por inducción sobre la longitud de la secuencia s.

- $P = Nil$

  Caso Base: $(s = \epsilon)$

  Dado que Nil es un proceso estable y tal que $S(Nil) = \emptyset$, se cumple

  $$\sum_{Q'} \{\!|\, p_i \mid Nil \stackrel{\epsilon}{\Longrightarrow}_{p_i} Q' \ \wedge\ S(Q') = \emptyset \,|\!\} = 1$$

  mientras que

  $$\sum_{Q'} \{\!|\, p_i \mid Nil \stackrel{\epsilon}{\Longrightarrow}_{p_i} Q' \ \wedge\ S(Q') = A \ \wedge\ A \neq \emptyset \,|\!\} = 0$$

  Por otro lado, $p([\![Nil]\!], \epsilon, \emptyset) = 1$, y $p([\![Nil]\!], \epsilon, A) = 0$, para cualquier $A \neq \emptyset$.

Caso Inductivo: $(s = \langle B\, b\rangle \circ s')$

Dado que $Nil \not\stackrel{s}{\Longrightarrow}$, para cada estado A se tiene

$$\sum_{Q'} \{\!|\, p_i \mid Nil \stackrel{s}{\Longrightarrow}_{p_i} Q' \ \wedge\ S(Q') = A \,|\!\} = 0$$

Por otro lado, para $s \neq \epsilon$ se tiene $p([\![Nil]\!], s, A) = 0$ para todo estado A.

- $P = \Omega$

Dado que el proceso Ω no se puede estabilizar (i.e. $\Omega \not\stackrel{s}{\Longrightarrow}$) tenemos que $P \not\stackrel{s}{\Longrightarrow}$ para toda secuencia s, con lo que para todo estado A se tiene

$$\sum_{Q'} \{\!|\, p_i \mid \Omega \stackrel{s}{\Longrightarrow}_{p_i} Q' \ \wedge\ S(Q') = A \,|\!\} = 0$$

Por otra parte, $p([\![\Omega]\!], s, A) = 0$ para cada secuencia s y cada estado A.

- $P = a; P'$

Caso Base: $(s = \epsilon)$

Dado que P es un proceso estable y tal que $S(P) = \{(a, 1)\}$, se cumple

$$\sum_{Q'} \{\!|\, p_i \mid P \stackrel{\epsilon}{\Longrightarrow}_{p_i} Q' \ \wedge\ S(Q') = \{(a,1)\} \,|\!\} = 1$$

mientras que para cualquier otro estado se tiene

$$\sum_{Q'} \{\!|\, p_i \mid P \stackrel{\epsilon}{\Longrightarrow}_{p_i} Q' \ \wedge\ S(Q') = A \ \wedge\ A \neq \{(a,1)\} \,|\!\} = 0$$

Por otro lado, $p([\![P]\!], \epsilon, \{(a,1)\}) = 1$ y $p([\![P]\!], \epsilon, A) = 0$ si $A \neq \{(a,1)\}$.

Caso Inductivo: $(s = \langle B\, b\rangle \circ s')$

Dado que P sólo puede ejecutar la transición externa $P \stackrel{a}{\longrightarrow}_1 P'$, tenemos que si $B \neq \{(a,1)\}$, entonces $P \not\stackrel{s}{\Longrightarrow}$, con lo que para todo estado A se tiene

$$\sum_{Q'} \{\!|\, p_i \mid P \stackrel{s}{\Longrightarrow}_{p_i} Q' \ \wedge\ S(Q') = A \,|\!\} = 0$$

De igual forma, para todo estado A se tiene $p([\![P]\!], s, A) = 0$.

Si $B = \{(a, 1)\}$, dado que $b \in B$, obtendríamos $b = a$. Además $P \stackrel{\langle B\, a\rangle o s'}{\Longrightarrow}_p Q$ sii $P' \stackrel{s'}{\Longrightarrow}_p Q$, con lo que

$$\sum_{Q'} \{\!|\, p_i \mid P \stackrel{s}{\Longrightarrow}_{p_i} Q' \ \wedge\ S(Q') = A \,|\!\} = \sum_{Q'} \{\!|\, p_i \mid P' \stackrel{s'}{\Longrightarrow}_{p_i} Q' \ \wedge\ S(Q') = A \,|\!\}$$

Por hipótesis de inducción tenemos

$$\sum_{Q'} \{\!|\, p_i \mid P' \stackrel{s'}{\Longrightarrow}_{p_i} Q' \ \wedge\ S(Q') = A \,|\!\} = p([\![P']\!], s', A)$$

mientras que por la definición de $[\![a; P']\!]$, obtenemos $p([\![P']\!], s', A) = p([\![P]\!], s, A)$.

- $P = P_1 \oplus_p P_2$

  P puede realizar dos transiciones internas, $P \rightarrowtail_p P_1$ y $P \rightarrowtail_{1-p} P_2$, por lo tanto la probabilidad de que el proceso P ejecute la secuencia s será igual a la suma de las probabilidades con las cuales P_1 y P_2 ejecutan la secuencia s multiplicadas respectivamente por p y $1 - p$. Es decir,

$$\sum_{Q'} \{\!|\, p_i \mid P \stackrel{s}{\Longrightarrow}_{p_i} Q' \wedge S(Q') = A \,|\!\} = p \cdot \sum_{Q'} \{\!|\, p_i \mid P_1 \stackrel{s}{\Longrightarrow}_{p_i} Q' \wedge S(Q') = A \,|\!\} + (1-p) \cdot \sum_{Q'} \{\!|\, p_i \mid P_2 \stackrel{s}{\Longrightarrow}_{p_i} Q' \wedge S(Q') = A \,|\!\}$$

  Por otro lado, por la definición de la función semántica correspondiente a la elección interna, tenemos

$$p([\![P]\!], s, A) = p \cdot p([\![P_1]\!], s, A) + (1-p) \cdot p([\![P_2]\!], s, A)$$

  mientras que por hipótesis de inducción

$$p([\![P_1]\!], s, A) = \sum_{Q'} \{\!|\, p_i \mid P_1 \stackrel{s}{\Longrightarrow}_{p_i} Q' \ \wedge\ S(Q') = A \,|\!\}$$

  y

$$p([\![P_2]\!], s, A) = \sum_{Q'} \{\!|\, p_i \mid P_2 \stackrel{s}{\Longrightarrow}_{p_i} Q' \ \wedge\ S(Q') = A \,|\!\}$$

  con lo que

$$p([\![P]\!], s, A) = \sum_{Q'} \{\!|\, p_i \mid P \stackrel{s}{\Longrightarrow}_{p_i} Q' \ \wedge\ S(Q') = A \,|\!\}$$

- $P = P_1 +_p P_2$

  Si uno de los dos procesos no se puede estabilizar (i.e. $P_1 \not\stackrel{\epsilon}{\Longrightarrow} \vee P_2 \not\stackrel{\epsilon}{\Longrightarrow}$), entonces P no se puede estabilizar, con lo que

  $$\sum_{Q'} \{\!\{ p_i \mid P \stackrel{s}{\Longrightarrow}_{p_i} Q' \wedge S(Q') = A \}\!\} = 0$$

  para cualquier estado A y cualquier secuencia s.

  Por otro lado, por la definición de la elección externa, tenemos $p([\![P]\!], s, A) = 0$ para cualquier estado A y cualquier secuencia s.

  Supongamos entonces que tanto P_1 como P_2 se pueden estabilizar. Realizaremos entonces la correspondiente inducción sobre la longitud de s.

  Caso Base: $(s = \epsilon)$

  El proceso P ejecutará la secuencia vacía una vez que los procesos P_1 y P_2 ejecuten una serie de transiciones internas que les lleven a estabilizarse. Una vez que los dos procesos sean estables, aplicando las reglas $(EXT4)$ y $(EXT5)$, tenemos que las transiciones (observables) que se puedan ejecutar serán aquellas que los procesos hacia los que han evolucionado P_1 y P_2 puedan ejecutar. Consideremos los valores

  $$p_A^{P_1} = \sum_{Q'} \{\!\{ p_i \mid P_1 \stackrel{\epsilon}{\Longrightarrow}_{p_i} Q' \wedge S(Q') = A \}\!\}$$

  y

  $$p_A^{P_2} = \sum_{Q'} \{\!\{ p_i \mid P_2 \stackrel{\epsilon}{\Longrightarrow}_{p_i} Q' \wedge S(Q') = A \}\!\}$$

  Entonces, los estados alcanzables por P tras la secuencia vacía (aplicando las reglas $(EXT1)$, $(EXT2)$ y $(EXT3)$ hasta que los dos procesos sean estables y después aplicando $(EXT4)$ y $(EXT5)$) serán aquellas indicadas en la Definición 4.26, en función de los correspondientes estados alcanzables de P_1 y P_2. Es decir,

  $$\sum_{Q'} \{\!\{ p_i \mid P \stackrel{\epsilon}{\Longrightarrow}_{p_i} Q' \wedge S(Q') = A \}\!\} = \sum_{A_1, A_2} \{\!\{ p_{A_1}^{P_1} \cdot p_{A_2}^{P_2} \mid A = A_1 \cup_p A_2 \}\!\}$$

Por otro lado,

$$p([\![P_1 +_p P_2]\!], \epsilon, A) = \sum_{A=B \cup_p C} p([\![P_1]\!], \epsilon, B) \cdot p([\![P_2]\!], \epsilon, C)$$

Por *hipótesis de inducción* se tiene $p([\![P_1]\!], \epsilon, B) = p_B^{P_1}$ y $p([\![P_2]\!], \epsilon, C) = p_C^{P_2}$, con lo cual obtenemos el resultado deseado:

$$p([\![P_1 +_p P_2]\!], \epsilon, A) = \sum_{Q'} \{\!|\, p_i \mid P \stackrel{\epsilon}{\Longrightarrow}_{p_i} Q' \ \wedge\ S(Q') = A \,|\!\}$$

Caso Inductivo: $(s = \langle B\, b \rangle \circ s')$

Para que el proceso P ejecute la secuencia s se debe cumplir que el proceso P_1 ejecute una transición interna generalizada, con la que llegará a un proceso P_1' que tendrá como estado asociado un cierto B_1, mientras que el proceso P_2 ejecuta una transición interna generalizada, con la que llega a un proceso que tiene como estado asociado un tal B_2, cumpliéndose que $B = B_1 \cup_p B_2$. Es decir, si $P_1 \rightarrowtail^*_{p_1} P_1' \ \wedge\ P_2 \rightarrowtail^*_{p_2} P_2'$, entonces $P \rightarrowtail^*_{p_1 \cdot p_2} P_1' +_p P_2'$ (aplicando las reglas $(EXT1)$, $(EXT2)$ y $(EXT3)$). A partir de este punto, hay tres posibilidades:

$\diamond\ P_1' \xrightarrow{b} \ \wedge\ P_2' \not\xrightarrow{b}$

Entonces el proceso definido por la elección externa ejecuta la acción b a partir de P_1' y después ejecuta el resto de la secuencia s'. Formalmente, la situación sería

- $P_1 \rightarrowtail^*_{p_1} P_1' \xrightarrow{b}_{p_1'} P_1'' \stackrel{s'}{\Longrightarrow}_{p_1''} Q' \ \wedge\ S(P_1') = B_1 \ \wedge\ S(Q') = A$
- $P_2 \rightarrowtail^*_{p_2} P_2' \ \wedge\ S(P_2') = B_2 \ \wedge\ P_2' \not\xrightarrow{b} \ \wedge\ B = B_1 \cup_p B_2$

En tal caso tendremos[10] $P \rightarrowtail^*_{p_1 \cdot p_2} P_1' +_p P_2' \xrightarrow{b}_{p \cdot p_1'} P_1'' \stackrel{s'}{\Longrightarrow}_{p_1''} Q'$, con lo que

$$P \stackrel{s}{\Longrightarrow}_q Q' \ \wedge\ S(Q') = A \iff q = p_1 \cdot p_2 \cdot \frac{p \cdot p_1'}{p \cdot \sum_{Q'} \{\!|\, q_i \mid P_1' \xrightarrow{b}_{q_i} Q' \,|\!\}} \cdot p_1''$$
$$\iff P_1 \stackrel{s_{B_1}}{\Longrightarrow}_{q'} Q' \ \wedge\ P_2 \rightarrowtail^*_{p_2} P_2'$$

[10] Formalmente, en la siguiente fórmula debemos distinguir dos casos: $B_2 = \emptyset$ y $B_2 \neq \emptyset$. En el primer caso, en lugar de tener la transición $\xrightarrow{b}_{p \cdot p_1'}$, tendríamos $\xrightarrow{b}_{p_1'}$, pero esto no produce ningún problema dado que al considerar $P \stackrel{s}{\Longrightarrow}_q$, el valor de p no influye en la probabilidad q en ninguno de los casos. El mismo comentario se aplicaría en el caso siguiente, para $(1-p) \cdot p_2'$.

donde $s_{B_1} = \langle B_1\, b\rangle \circ s'$ y $q' = p_1 \cdot \frac{p_1'}{\sum_{Q'} \{\!|\, q_i \,|\, P_1' \xrightarrow{b} q_i Q' \,|\!\}} \cdot p_1''$. Nótese que, por el hecho de tenerse $S(P_1') = B_1$, se cumple $\sum_{Q'} \{\!|\, q_i \mid P_1' \xrightarrow{b}_{q_i} Q' \,|\!\} = pro(b, B_1)$. Si agrupamos entonces los sumandos de esta forma obtenemos[11]

$$\sum_{\substack{B = B_1 \cup_p B_2 \\ b \in B_1 \ \wedge\ b \notin B_2}} \sum_{Q'} \{\!|\, q_i | P_1 \overset{s_{B_1}}{\Longrightarrow}_{q_i} Q' \wedge S(Q') = A \,|\!\} \cdot \sum_{Q'} \{\!|\, q_i | P_2 \overset{\epsilon}{\Longrightarrow}_{q_i} Q' \wedge S(Q') = B_2 \,|\!\} \tag{4.16}$$

$\diamond\ P_1' \not\xrightarrow{b} \ \wedge\ P_2' \xrightarrow{b}$

La nueva elección externa ejecuta la acción b a partir de P_2' y después ejecuta el resto de la secuencia s'. Formalmente, la situación sería

- $P_1 \rightarrowtail^*_{p_1} P_1' \ \wedge\ S(P_1') = B_1 \ \wedge\ P_1' \not\xrightarrow{b} \ \wedge\ B = B_1 \cup_p B_2$
- $P_2 \rightarrowtail^*_{p_2} P_2' \xrightarrow{b}_{p_2'} P_2'' \overset{s'}{\Longrightarrow}_{p_2''} Q' \ \wedge\ S(P_2') = B_2 \ \wedge\ S(Q') = A$

En tal caso tendremos $P \rightarrowtail^*_{p_1 \cdot p_2} P_1' +_p P_2' \xrightarrow{b}_{(1-p)\cdot p_2'} P_2'' \overset{s'}{\Longrightarrow}_{p_2''} Q'$, con lo que

$$\begin{aligned} P \overset{s}{\Longrightarrow}_q Q' \ \wedge\ S(Q') = A \ &\Longleftrightarrow q = p_1 \cdot p_2 \cdot \frac{p_2'}{\sum_{Q'} \{\!|\, q_i \,|\, P_2' \xrightarrow{b} q_i Q' \,|\!\}} \cdot p_2'' \\ &\Longleftrightarrow P_2 \overset{s_{B_2}}{\Longrightarrow}_{q'} Q' \ \wedge\ P_1 \rightarrowtail^*_{p_1} P_1' \end{aligned}$$

donde $s_{B_2} = \langle B_2\, b\rangle \circ s'$ y $q' = p_2 \cdot \frac{p_2'}{\sum_{Q'} \{\!|\, q_i \,|\, P_2' \xrightarrow{b} q_i Q' \,|\!\}} \cdot p_2''$. Nótese que, por el hecho de tenerse $S(P_2') = B_2$, se cumple $\sum_{Q'} \{\!|\, q_i \mid P_2' \xrightarrow{b}_{q_i} Q' \,|\!\} = pro(b, B_2)$. Si agrupamos los sumandos de esta forma obtenemos

$$\sum_{\substack{B = B_1 \cup_p B_2 \\ b \notin B_1 \ \wedge\ b \in B_2}} \sum_{Q'} \{\!|\, q_i | P_1 \overset{\epsilon}{\Longrightarrow}_{q_i} Q' \wedge S(Q') = B_1 \,|\!\} \cdot \sum_{Q'} \{\!|\, q_i | P_2 \overset{s_{B_2}}{\Longrightarrow}_{q_i} Q' \wedge S(Q') = A \,|\!\} \tag{4.17}$$

$\diamond\ P_1' \xrightarrow{b} \ \wedge\ P_2' \xrightarrow{b}$

El proceso resultante de la elección externa ejecutará una de las transiciones etiquetadas con b de entre las que P_1' puede ejecutar o bien una de las correspondientes a P_2', sopesando en cada caso las probabilidades asociadas a estas

[11] En el resto de la demostración, para simplificar la notación, utilizaremos $b \in X$ como una abreviatura de $pro(b, X) > 0$, y $b \notin X$ como una abreviatura de $pro(b, X) = 0$.

transiciones, por p y $1-p$ respectivamente. Dependiendo de cual de los dos procesos ejecute la acción b tendremos dos casos:

$\diamondsuit$ P_1' ejecuta la acción b.

En tal caso la situación será la siguiente:

- $P_1 \rightarrowtail^*_{p_1} P_1' \xrightarrow{b}_{p_1'} P_1'' \stackrel{s'}{\Longrightarrow}_{p_1''} Q' \ \wedge\ S(P_1') = B_1 \ \wedge\ S(Q') = A$
- $P_2 \rightarrowtail^*_{p_2} P_2' \ \wedge\ S(P_2') = B_2 \ \wedge\ P_2' \xrightarrow{b} \ \wedge\ B = B_1 \cup_p B_2$

De modo que tendremos $P \rightarrowtail^*_{p_1 \cdot p_2} P_1' +_p P_2' \xrightarrow{b}_{p \cdot p_1'} P_1'' \stackrel{s'}{\Longrightarrow}_{p_1''} Q'$, con lo que

$$
\begin{aligned}
& P \stackrel{s}{\Longrightarrow}_q Q' \ \wedge\ S(Q') = A \\
& \quad \Longleftrightarrow q = p_1 \cdot p_2 \cdot \frac{p \cdot p_1'}{p \cdot \sum_{Q'} \{\!|\, q_i \mid P_1' \xrightarrow{b}_{q_i} Q' \,|\!\} + (1-p) \cdot \sum_{Q'} \{\!|\, q_i \mid P_2' \xrightarrow{b}_{q_i} Q' \,|\!\}} \cdot p_1'' \\
& \quad \Longleftrightarrow P_1 \stackrel{s_{B_1}}{\Longrightarrow}_{q'} Q' \ \wedge\ P_2 \rightarrowtail^*_{p_2} P_2' \\
& \qquad \wedge\ q = \frac{p \cdot \sum_{Q'} \{\!|\, q_i \mid P_1' \xrightarrow{b}_{q_i} Q' \,|\!\}}{p \cdot \sum_{Q'} \{\!|\, q_i \mid P_1' \xrightarrow{b}_{q_i} Q' \,|\!\} + (1-p) \cdot \sum_{Q'} \{\!|\, q_i \mid P_2' \xrightarrow{b}_{q_i} Q' \,|\!\}} \cdot q' \cdot p_2
\end{aligned}
$$

donde $s_{B_1} = \langle B_1\, b \rangle \circ s'$ y $q' = p_1 \cdot \frac{p_1'}{\sum_{Q'} \{\!|\, q_i \mid P_1' \xrightarrow{b}_{q_i} Q' \,|\!\}} \cdot p_1''$. Nótese que, por el hecho de tenerse $S(P_1') = B_1$ y $S(P_2') = B_2$, se cumple $\sum_{Q'} \{\!|\, q_i \mid P_1' \xrightarrow{b}_{q_i} Q' \,|\!\} = pro(b, B_1)$, mientras que $\sum_{Q'} \{\!|\, q_i \mid P_2' \xrightarrow{b}_{q_i} Q' \,|\!\} = pro(b, B_2)$.

Finalmente, si agrupamos los sumandos de esta forma obtenemos

$$
\begin{aligned}
\sum_{\substack{B = B_1 \cup_p B_2 \\ b \in B_1 \ \wedge\ b \in B_2}} & \frac{p \cdot pro(b, B_1)}{p \cdot pro(b, B_1) + (1-p) \cdot pro(b, B_2)} \cdot \sum_{Q'} \{\!|\, q_i \mid P_1 \stackrel{s_{B_1}}{\Longrightarrow}_{q_i} Q' \ \wedge\ S(Q') = A \,|\!\} \\
& \cdot \sum_{Q'} \{\!|\, q_i \mid P_2 \stackrel{\epsilon}{\Longrightarrow}_{q_i} Q' \ \wedge\ S(Q') = B_2 \,|\!\} \qquad (4.18)
\end{aligned}
$$

$\diamondsuit$ P_2' ejecuta la acción b.

Por tanto la situación será:

- $P_1 \rightarrowtail^*_{p_1} P_1' \ \wedge\ S(P_1') = B_1 \ \wedge\ P_1' \xrightarrow{b} \ \wedge\ B = B_1 \cup_p B_2$
- $P_2 \rightarrowtail^*_{p_2} P_2' \xrightarrow{b}_{p_2'} P_2'' \stackrel{s'}{\Longrightarrow}_{p_2''} Q' \ \wedge\ S(P_2') = B_2 \ \wedge\ S(Q') = A$

En este caso tendremos $P \rightarrowtail^*_{p_1 \cdot p_2} P'_1 +_p P'_2 \xrightarrow{b}_{(1-p)\cdot p'_2} P''_2 \stackrel{s'}{\Longrightarrow}_{p''_2} Q'$, con lo que

$$\begin{aligned} & P \stackrel{s}{\Longrightarrow}_q Q' \ \wedge \ S(Q') = A \\ & \quad \Longleftrightarrow q = p_1 \cdot p_2 \cdot \frac{(1-p)\cdot p'_2}{p \cdot \sum_{Q'} \{\!|\, q_i \mid P'_1 \xrightarrow{b}_{q_i} Q' \,|\!\}_+ (1-p)\cdot \sum_{Q'} \{\!|\, q_i \mid P'_2 \xrightarrow{b}_{q_i} Q' \,|\!\}} \cdot p''_2 \\ & \quad \Longleftrightarrow P_2 \stackrel{s_{B_2}}{\Longrightarrow}_{q'} Q' \ \wedge \ P_1 \rightarrowtail^*_{p_1} P'_1 \\ & \qquad \wedge \ q = \frac{(1-p)\cdot \sum_{Q'} \{\!|\, q_i \mid P'_2 \xrightarrow{b}_{q_i} Q' \,|\!\}}{p \cdot \sum_{Q'} \{\!|\, q_i \mid P'_1 \xrightarrow{b}_{q_i} Q' \,|\!\}_+ (1-p)\cdot \sum_{Q'} \{\!|\, q_i \mid P'_2 \xrightarrow{b}_{q_i} Q' \,|\!\}} \cdot q' \cdot p_1 \end{aligned}$$

donde $s_{B_2} = \langle B_2\, b\rangle \circ s'$ y $q' = p_2 \cdot \frac{p'_2}{\sum_{Q'} \{\!|\, q_i \mid P'_2 \xrightarrow{b}_{q_i} Q' \,|\!\}} \cdot p''_2$. Nótese que, por el hecho de tenerse $S(P'_1) = B_1$ y $S(P'_2) = B_2$, se cumple $\sum_{Q'} \{\!|\, q_i \mid P'_1 \xrightarrow{b}_{q_i} Q' \,|\!\} = pro(b, B_1)$, mientras que $\sum_{Q'} \{\!|\, q_i \mid P'_2 \xrightarrow{b}_{q_i} Q' \,|\!\} = pro(b, B_2)$.

Si agrupamos los sumandos de esta forma obtenemos

$$\sum_{\substack{B = B_1 \cup_p B_2 \\ b \in B_1 \ \wedge \ b \in B_2}} \frac{(1-p)\cdot pro(b,B_2)}{p\cdot pro(b,B_1)+(1-p)\cdot pro(b,B_2)} \cdot \sum_{Q'} \{\!|\, q_i \mid P_2 \stackrel{s_{B_2}}{\Longrightarrow}_{q_i} Q' \ \wedge \ S(Q') = A \,|\!\}$$
$$\cdot \sum_{Q'} \{\!|\, q_i \mid P_1 \stackrel{\epsilon}{\Longrightarrow}_{q_i} Q' \ \wedge \ S(Q') = B_1 \,|\!\} \quad (4.19)$$

A partir de los cuatro casos anteriores obtenemos

$$\sum_{Q'} \{\!|\, q_i \mid P \stackrel{s}{\Longrightarrow}_{q_i} Q' \ \wedge \ S(Q') = A \,|\!\} = (4.16) + (4.17) + (4.18) + (4.19)$$

Por otra parte,

$$\begin{aligned} & p([\![P_1 +_p P_2]\!], s, A) = \\ & \qquad \sum_{B = B_1 \cup_p B_2} \frac{p\cdot pro(b,B_1)}{p\cdot pro(b,B_1)+(1-p)\cdot pro(b,B_2)} \cdot p([\![P_1]\!], s_{B_1}, A) \cdot p([\![P_2]\!], \epsilon, B_2) \\ & \quad + \sum_{B = B_1 \cup_p B_2} \frac{(1-p)\cdot pro(b,B_2)}{p\cdot pro(b,B_1)+(1-p)\cdot pro(b,B_2)} \cdot p([\![P_2]\!], s_{B_2}, A) \cdot p([\![P_1]\!], \epsilon, B_1) \end{aligned}$$

donde $s_{B_1} = \langle B_1\, b\rangle \circ s'$ y $s_{B_2} = \langle B_2\, b\rangle \circ s'$.

Al igual que antes podemos descomponer esta expresión en cuatro sumatorios, que corresponden a los distintos casos según la acción b corresponda o no a los estados B_1 y/o B_2.

$$
\begin{aligned}
p([\![P_1 +_p P_2]\!], s, A) = \\
&\sum_{\substack{B = B_1 \cup_p B_2 \\ b \in B_1 \ \wedge \ b \notin B_2}} p([\![P_1]\!], s_{B_1}, A) \cdot p([\![P_2]\!], \epsilon, B_2) \\
&+ \sum_{\substack{B = B_1 \cup_p B_2 \\ b \notin B_1 \ \wedge \ b \in B_2}} p([\![P_2]\!], s_{B_2}, A) \cdot p([\![P_1]\!], \epsilon, B_1) \\
&+ \sum_{\substack{B = B_1 \cup_p B_2 \\ b \in B_1 \ \wedge \ b \in B_2}} \frac{p \cdot pro(b, B_1)}{p \cdot pro(b, B_1) + (1-p) \cdot pro(b, B_2)} \cdot p([\![P_1]\!], s_{B_1}, A) \cdot p([\![P_2]\!], \epsilon, B_2) \\
&+ \sum_{\substack{B = B_1 \cup_p B_2 \\ b \in B_1 \ \wedge \ b \in B_2}} \frac{(1-p) \cdot pro(b, B_2)}{p \cdot pro(b, B_1) + (1-p) \cdot pro(b, B_2)} \cdot p([\![P_2]\!], s_{B_2}, A) \cdot p([\![P_1]\!], \epsilon, B_1)
\end{aligned}
$$

donde $s_{B_1} = \langle B_1\, b \rangle \circ s'$ y $s_{B_2} = \langle B_2\, b \rangle \circ s'$. Nótese que en los dos primeros sumatorios los factores $\frac{p \cdot pro(b,B_1)}{p \cdot pro(b,B_1) + (1-p) \cdot pro(b,B_2)}$ y $\frac{(1-p) \cdot pro(b,B_2)}{p \cdot pro(b,B_1) + (1-p) \cdot pro(b,B_2)}$ han desaparecido, dado que en el primer caso $pro(b, B_1) = 0$, mientras que en el segundo $pro(b, B_2) = 0$.

Aplicando la hipótesis de inducción obtenemos

$$
\begin{aligned}
p([\![P_1]\!], s_{B_1}, A) &= \Sigma_{Q'} \{\!|\, q_i \mid P_1 \stackrel{s_{B_1}}{\Longrightarrow}_{q_i} Q' \ \wedge \ S(Q') = A \,|\!\}, \\
p([\![P_1]\!], \epsilon, B_1) &= \Sigma_{Q'} \{\!|\, q_i \mid P_1 \stackrel{\epsilon}{\Longrightarrow}_{q_i} Q' \ \wedge \ S(Q') = B_1 \,|\!\} \\
p([\![P_2]\!], s_{B_2}, A) &= \Sigma_{Q'} \{\!|\, q_i \mid P_2 \stackrel{s_{B_2}}{\Longrightarrow}_{q_i} Q' \ \wedge \ S(Q') = A \,|\!\} \\
p([\![P_2]\!], \epsilon, B_2) &= \Sigma_{Q'} \{\!|\, q_i \mid P_2 \stackrel{\epsilon}{\Longrightarrow}_{q_i} Q' \ \wedge \ S(Q') = B_2 \,|\!\}
\end{aligned}
$$

de lo que se deduce inmediatamente

$$
\sum_{Q'} \{\!|\, q_i \mid P \stackrel{s}{\Longrightarrow}_{q_i} Q' \ \wedge \ S(Q') = A \,|\!\} = p([\![P_1 +_p P_2]\!], s, A)
$$

□

Damos ahora un lema (análogo al Lema 3.17) que nos relaciona el comportamiento operacional de un proceso recursivo con los de sus aproximaciones finitas.

Lema 4.31 Sea $P = recX.P(X)$. Entonces, para toda secuencia de pares (estado, acción) s, y para todo $p \in [0,1]$ se verifica que $P \stackrel{s}{\Longrightarrow}_p P'$ sii $\exists\, n \in \mathbb{N}^+ :\ P_n \stackrel{s}{\Longrightarrow}_p P''_n$.

□

Teorema 4.7 Para todo proceso $P \in \mathrm{PPA}$, y toda secuencia de acciones s se tiene

$$p([\![P]\!], s, A) = \sum_{P'} \{\!|\ p_i \mid P \stackrel{s}{\Longrightarrow}_{p_i} P' \ \wedge\ S(P') = A\ |\!\}$$

Demostración: Análoga a la del Teorema 3.3, basándonos en esta ocasión en los Lemas 4.30 y 4.31. □

Corolario 4.8 Para cada proceso P tenemos $\mathcal{A}(P,s) = \{(A_1,p_1),(A_2,p_2)\ldots(A_n,p_n)\}$ siendo $p_i = \frac{p([\![P]\!],s,A_i)}{p_s^P}$ para cada i, y $p([\![P]\!],s,A) = 0$ para cada A distinto de los A_i, donde $p_\epsilon^P = 1$ y $p_{s'\circ(B\,b)}^P = p([\![P]\!], s', B)$.

Demostración: Trivial a partir del Teorema 4.7, y de la Definición 4.7. □

Corolario 4.9 Para cualesquiera procesos P y Q, $P \cong Q \Leftrightarrow [\![P]\!] =_{\mathrm{PAT}} [\![Q]\!]$. □

Corolario 4.10 (Abstracción Completa para $\mathrm{PAT}_{\mathrm{Act}}$)
Sean P, Q procesos. Entonces, $P \approx_{\mathcal{G}} Q \Leftrightarrow [\![P]\!] =_{\mathrm{PAT}} [\![Q]\!]$.

Demostración: Consecuencia inmediata de los Corolarios 4.3 y 4.9. □

Capítulo 5

Semántica Axiomática para el Modelo Generativo

En este capítulo presentamos un Sistema de Axiomas y Reglas que resulta ser correcto y completo respecto de la semántica de pruebas para el modelo generativo.

Como es habitual, comenzaremos trabajando con el subconjunto de PPA formado por los procesos finitos, es decir, con aquéllos generados por nuestra sintaxis en la cual prescindimos del operador de recursión ($recX.P$), excluyendo también el proceso divergente Ω. Para este subconjunto de nuestra álgebra de procesos probabilística definiremos un conjunto de axiomas y reglas de inferencia que formarán un sistema correcto y completo respecto de la semántica de pruebas para el modelo generativo.

Una vez realizado el estudio para los procesos finitos consideraremos el lenguage completo. Al efecto, a partir del sistema axiomático obtenido para los procesos finitos construiremos un nuevo conjunto que incluirá axiomas y reglas que traten los casos del proceso divergente y de los procesos recursivos. Al igual que ocurría en [Cua93], para poder probar la completitud del nuevo sistema (siempre teniendo en cuenta que hemos incluido reglas infinitarias, con lo cual no podemos hablar de una completitud *real*) además de las reglas infinitarias usuales asociadas a las aproximaciones finitas de los procesos recursivos, necesitaremos una regla adicional que podríamos calificar de orden *técnico*.

5.1 Sistema de Axiomas para procesos finitos

Como ya hemos dicho en la introducción, trabajaremos en principio con el subconjunto de PPA formado por los procesos finitos no divergentes. A este subconjunto lo denotaremos por PPA_{fin}, siendo su definición la siguiente:

Definición 5.1 Dado un conjunto de acciones Act, el conjunto de procesos que pertenecen a PPA_{fin} viene definido mediante la siguiente expresión BNF:

$$P ::= Nil \mid a;P \mid P \oplus_p P \mid P +_p P$$

siendo $p \in (0,1)$ y $a \in Act$. □

Para este sublenguaje definiremos un sistema lógico de reglas y axiomas, el cual induce una relación de equivalencia, a la que notaremos por $\equiv_{\text{gen}}$, entre los términos del lenguaje PPA_{fin}. El sistema incluye axiomas que expresan las propiedades algebraicas básicas de los operadores de nuestro lenguaje, como la idempotencia, conmutatividad, asociatividad, así como relaciones entre los operadores como la distributividad, aunque en algunos casos los axiomas no se ajustarán al correspondiente esquema algebraico puro, pues se precisará un *reequilibrio* de las probabilidades para ajustar las mismas debidamente. También veremos que algunos axiomas que eran correctos en el caso no probabilístico, dejan de serlo al extender el lenguaje con probabilidades. Para cada axioma presentaremos la correspondiente prueba de corrección. En las mismas utilizaremos frecuentemente el hecho de que para probar la equivalencia respecto de la semántica de pruebas para el modelo generativo entre dos procesos podemos restringirnos a barbas probabilísticas (ver Teorema 4.4).

5.1.1 Axiomas del Sistema Probabilístico

- El operador $\oplus_p$ es idempotente.

$$\textbf{(II)} \quad P \oplus_p P \equiv_{\text{gen}} P$$

Demostración: Consideremos una prueba cualquiera $T \in \mathcal{PB}$. Dado que la misma no puede realizar transiciones internas, las únicas transiciones posibles de la composición entre el proceso $R = P \oplus_p P$ y la prueba T son $R \mid T \longmapsto_p P \mid T$ y $R \mid T \longmapsto_{1-p} P \mid T$, con lo cual

$$pass(R\,,\,T) = p \cdot pass(P\,,\,T) + (1-p) \cdot pass(P\,,\,T) = pass(P\,,\,T)$$

• El operador $\oplus_p$ es conmutativo.

$$\textbf{(CI)} \quad P \oplus_p Q \equiv_{\text{gen}} Q \oplus_{1-p} P$$

Demostración: : Siendo $T \in \mathcal{PB}$ una prueba, las únicas transiciones que pueden hacer las composiciones de los procesos involucrados con T son:

$\star\ P \oplus_p Q \mid T \longmapsto_p P \mid T$ $\qquad$ $\star\ P \oplus_p Q \mid T \longmapsto_{1-p} Q \mid T$

$\star\ Q \oplus_{1-p} P \mid T \longmapsto_{1-p} Q \mid T$ $\qquad$ $\star\ Q \oplus_{1-p} P \mid T \longmapsto_p P \mid T$

con lo cual

$$\begin{aligned} pass(P \oplus_p Q\,,\,T) &= p \cdot pass(P\,,\,T) + (1-p) \cdot pass(Q\,,\,T) \\ &= (1-p) \cdot pass(Q\,,\,T) + p \cdot pass(P\,,\,T) \\ &= pass(Q \oplus_{1-p} P\,,\,T) \end{aligned}$$

• El operador $\oplus_p$ es asociativo.

$$\textbf{(AI)} \quad P \oplus_p (Q \oplus_q R) \equiv_{\text{gen}} (P \oplus_{p'} Q) \oplus_{q'} R$$

siendo $q' = p + q - p \cdot q$ y $p' = \frac{p}{q'}$.

Demostración: Sea $T \in \mathcal{PB}$. Las únicas transiciones que pueden hacer las composiciones de los procesos anteriores con T son:

$\star\ P \oplus_p (Q \oplus_q R) \mid T \longmapsto_{1-p} Q \oplus_q R \mid T$ $\qquad$ $\star\ P \oplus_p (Q \oplus_q R) \mid T \longmapsto_p P \mid T$

$\star\ (P \oplus_{p'} Q) \oplus_{q'} R \mid T \longmapsto_{q'} P \oplus_{p'} Q \mid T$ $\qquad$ $\star\ (P \oplus_{p'} Q) \oplus_{q'} R \mid T \longmapsto_{1-q'} R \mid T$

Además, las únicas transiciones que pueden realizar las composiciones de los procesos $Q \oplus_q R$ y $P \oplus_{p'} Q$ con la prueba son:

$\star\ Q \oplus_q R \mid T \longmapsto_q Q \mid T$ $\qquad$ $\star\ Q \oplus_q R \mid T \longmapsto_{1-q} R \mid T$

$\star\ P \oplus_{p'} Q \mid T \longmapsto_{p'} P \mid T$ $\qquad$ $\star\ P \oplus_{p'} Q \mid T \longmapsto_{1-p'} Q \mid T$

De lo cual se deduce que los primeros *pasos* de las composiciones iniciales serán:

$$\star\ P \oplus_p (Q \oplus_q R) \mid T \longmapsto_{1-p} Q \oplus_q R \mid T \longmapsto_q Q \mid T$$
$$\star\ P \oplus_p (Q \oplus_q R) \mid T \longmapsto_{1-p} Q \oplus_q R \mid T \longmapsto_{1-q} R \mid T$$
$$\star\ P \oplus_p (Q \oplus_q R) \mid T \longmapsto_p P \mid T$$
$$\star\ (P \oplus_{p'} Q) \oplus_{q'} R \mid T \longmapsto_{q'} P \oplus_{p'} Q \mid T \longmapsto_{p'} P \mid T$$
$$\star\ (P \oplus_{p'} Q) \oplus_{q'} R \mid T \longmapsto_{q'} P \oplus_{p'} Q \mid T \longmapsto_{1-p'} Q \mid T$$
$$\star\ (P \oplus_{p'} Q) \oplus_{q'} R \mid T \longmapsto_{1-q'} R \mid T$$

con lo cual

$$pass(P \oplus_p (Q \oplus_q R), T) = (1-p)\cdot q \cdot pass(Q,T) + (1-p)\cdot(1-q)\cdot pass(R,T) + p \cdot pass(P,T)$$

mientras que

$$pass((P \oplus_{p'} Q) \oplus_{q'} R, T) = q' \cdot p' \cdot pass(P,T) + q' \cdot (1-p') \cdot pass(Q,T) + (1-q') \cdot pass(R,T)$$

Para terminar la prueba nos queda ver que los factores que multiplican a $pass(P, T)$, $pass(Q, T)$ y $pass(R, T)$ son los mismos en ambos casos. En efecto:

$$\star\ q' \cdot p' = q' \cdot \tfrac{p}{q'} = p$$
$$\star\ q' \cdot (1-p') = q' \cdot (1 - \tfrac{p}{q'}) = q' - p = q - p \cdot q = q \cdot (1-p)$$
$$\star\ 1 - q' = 1 - p - q + p \cdot q = (1-p) \cdot (1-q)$$

Pasemos ahora a enunciar los axiomas relacionados con la elección externa. Mientras que el operador de elección interna era idempotente, asociativo y conmutativo, en el caso de la elección externa sólo se va a cumplir esta última propiedad, mientras que sólo tendremos una forma restringida de la asociatividad y la idempotencia. Además, se verifica que el proceso *Nil* es elemento neutro para la elección externa.

• El operador $+_p$ es conmutativo.

$$\textbf{(CE)} \quad P +_p Q \equiv_{\text{gen}} Q +_{1-p} P$$

Demostración: Sea $T \in \mathcal{PB}$. Aplicando las reglas $(EXT1), (EXT2)$ y $(EXT3)$ vemos que la probabilidad asociada a la elección externa no influye en la probabilidad

con la cual la elección externa ejecuta transiciones internas a partir de los procesos componentes. Es decir, $P +_p Q \rightarrowtail^*_q P' +_p Q' \iff Q +_{1-p} P \rightarrowtail^*_q Q' +_{1-p} P'$. Una vez que el proceso es estable, aplicando las reglas $(EXT4)$ y $(EXT5)$, obtenemos $P' +_p Q' \xrightarrow{a}_q R \iff Q' +_{1-p} P' \xrightarrow{a}_q R$, con lo cual

$$\begin{aligned} pass(P +_p Q\ ,\ T) &= \textstyle\sum_{P',Q'} \{\!|\ q \cdot pass(P' +_p Q'\ ,\ T) \mid P +_p Q \rightarrowtail^*_q P' +_p Q' \ |\!\} \\ &= \textstyle\sum_{P',Q'} \{\!|\ q \cdot pass(Q' +_{1-p} P'\ ,\ T) \mid Q +_{1-p} P \rightarrowtail^*_q Q' +_{1-p} P' \ |\!\} \\ &= pass(Q +_{1-p} P\ ,\ T) \end{aligned}$$

- El proceso Nil es elemento neutro para el operador $+_p$.

$$\textbf{(NE)} \quad P +_p Nil \equiv_{\mathbf{gen}} P$$

Demostración: Sea $T \in \mathcal{PB}$. Aplicando las reglas $(EXT1), (EXT2)$ y $(EXT3)$ tenemos que $P +_p Nil \rightarrowtail^*_q R +_p Nil \iff P \rightarrowtail^*_q R$. Además, por la regla $(EXT4)$, tenemos que $R +_p Nil \xrightarrow{a}_q R' \iff R \xrightarrow{a}_q R'$, de lo cual se deduce $pass(R +_p Nil\ ,\ T) = pass(R\ ,\ T)$, con lo cual

$$\begin{aligned} pass(P +_p Nil\ ,\ T) &= \textstyle\sum_R \{\!|\ q \cdot pass(R +_p Nil\ ,\ T) \mid P +_p Nil \rightarrowtail^*_q R +_p Nil \ |\!\} \\ &= \textstyle\sum_R \{\!|\ q \cdot pass(R +_p Nil\ ,\ T) \mid P \rightarrowtail^*_q R \ |\!\} \\ &= \textstyle\sum_R \{\!|\ q \cdot pass(R\ ,\ T) \mid P \rightarrowtail^*_q R \ |\!\} \\ &= pass(P\ ,\ T) \end{aligned}$$

Como ya dijimos anteriormente, el operador de elección externa no es idempotente, como muestra el siguiente

Ejemplo 5.2 Consideremos los procesos $P = a \oplus_{\frac{1}{2}} b$ y $P' = P +_{\frac{1}{2}} P$, y la prueba $T = a;\omega$. Mientras que $pass(P\ ,\ T) = \frac{1}{2}$, tenemos que $pass(P'\ ,\ T) = \frac{3}{4}$. □

Sin embargo, si el proceso en cuestión no puede realizar transiciones internas el resultado sí se cumple.

Proposición 5.3 Sea P un proceso tal que $P \not\rightarrowtail$. Entonces, $P \approx_{\mathcal{G}} P +_p P$ para todo $p \in (0,1)$.

Demostración: Dado que el proceso P no puede realizar transiciones internas, las únicas reglas que se pueden aplicar para generar transiciones de $P +_p P$ son $(EXT4)$ y $(EXT5)$. Además, si $P \xrightarrow{a}_q P'$, aplicando $(EXT4)$ tenemos $P +_p P \xrightarrow{a}_{p\cdot\hat{q}} P'$, mientras que aplicando $(EXT5)$ tenemos $P +_p P \xrightarrow{a}_{(1-p)\cdot\hat{q}} P'$, donde $\hat{q} = \frac{q}{p\cdot P_+ + (1-p)\cdot P_+} = q$, con lo cual, sumando las probabilidades asociadas a ambas transiciones, y aplicando las reglas $(PAR4)$ y $(PAR5)$, obtenemos el resultado deseado. □

Tampoco es cierta en general la asociatividad del operador de elección externa, ni siquiera si consideramos un reequilibrio de probabilidades como el utilizado en el axioma (**AI**).

Ejemplo 5.4 Consideremos los procesos $P = a +_{\frac{1}{2}} (b +_{\frac{1}{2}} Nil)$ y $P' = (a +_{\frac{2}{3}} b) +_{\frac{3}{4}} Nil$, y la prueba $T = a;\omega +_{\frac{1}{2}} b; Nil$. Se cumple que $pass(P\ ,\ T) = \frac{1}{2}$, mientras que $pass(P'\ ,\ T) = \frac{2}{3}$. Esto es así, dado que $P \approx_{\mathcal{G}} a +_{\frac{1}{2}} b$ mientras que $P' \approx_{\mathcal{G}} a +_{\frac{2}{3}} b$, y claramente $a +_{\frac{1}{2}} b \not\approx_{\mathcal{G}} a +_{\frac{2}{3}} b$. □

La falta de asociatividad del operador de elección externa también aparecía en el modelo estudiado en [Cua93], pero mientras que en aquel modelo la falta de asociatividad provenía de la posibilidad de componer en elección externa dos procesos que puedan ejecutar en su primer paso la misma acción, en nuestro caso el problema viene de la posibilidad de que dicha composición, una vez que los procesos son estables, aparezca el proceso Nil. En nuestro caso el problema tiene una fácil solución, pues mediante el axioma (**NE**) podemos eliminar todas las apariciones del proceso Nil en una composición en elección externa. En definitiva, al igual que ocurría en [Cua93], vamos a tener una forma más restringida de asociatividad para el operador de elección externa, que a la postre será suficiente para conseguir nuestro objetivo de poder transformar cualquier proceso sintáctico en su forma normal.

Proposición 5.5 Sean P_1, P_2, P_3 procesos tales que para todo $i \in \{1 \ldots 3\}$ se cumple $P_i \longrightarrow$, es decir, procesos estables que no sean operacionalmente equivalentes a Nil. Entonces se cumple

$$P_1 +_p (P_2 +_q P_3) \approx_{\mathcal{G}} (P_1 +_{p'} P_2) +_{q'} P_3$$

donde $q' = p + q - p \cdot q$ y $p' = \frac{p}{q'}$.

Demostración: Demostraremos que para toda $a \in Act$ y para todo $Q \in \mathrm{PPA}_{fin}$ se verifica el siguiente resultado

$$\sum \{\!\{ r \mid P_1 +_p (P_2 +_q P_3) \xrightarrow{a}_r Q \}\!\} = \sum \{\!\{ r \mid (P_1 +_{p'} P_2) +_{q'} P_3 \xrightarrow{a}_r Q \}\!\}$$

a partir del cual se sigue inmediatamente la equivalencia que queremos demostrar.

Aplicando las reglas $(EXT4)$ y $(EXT5)$, y teniendo en cuenta el hecho de que por ser todos los procesos estables y distintos de *Nil* se cumple $P_{i+} = 1$, obtenemos

$$\begin{aligned}\sum \{\!\{ r \mid P_1 +_p (P_2 +_q P_3) \xrightarrow{a}_r Q \}\!\} &= \sum \{\!\{ p \cdot r \mid P_1 \xrightarrow{a}_r Q \}\!\} \\ &+ \sum \{\!\{ (1-p) \cdot q \cdot r \mid P_2 \xrightarrow{a}_r Q \}\!\} \\ &+ \sum \{\!\{ (1-p) \cdot (1-q) \cdot r \mid P_3 \xrightarrow{a}_r Q \}\!\}\end{aligned}$$

$$\begin{aligned}\sum \{\!\{ r \mid (P_1 +_{p'} P_2) +_{q'} P_3 \xrightarrow{a}_r Q \}\!\} &= \sum \{\!\{ p' \cdot q' \cdot r \mid P_1 \xrightarrow{a}_r Q \}\!\} \\ &+ \sum \{\!\{ (1-p') \cdot q' \cdot r \mid P_2 \xrightarrow{a}_r Q \}\!\} \\ &+ \sum \{\!\{ (1-q') \cdot r \mid P_3 \xrightarrow{a}_r Q \}\!\}\end{aligned}$$

Pero como quiera que en la demostración de corrección del axioma **(AI)** ya probamos

$\star\ p = q' \cdot p' \qquad \star\ q \cdot (1-p) = q' \cdot (1-p') \qquad \star\ (1-p) \cdot (1-q) = 1 - q'$

obtenemos inmediatamente el resultado deseado. □

A continuación consideraremos un axioma que indica una ley distributiva entre el operador de elección externa y el de elección interna.

- El operador $+_p$ distribuye sobre el operador $\oplus_q$.

$$\textbf{(DEI)} \quad P_1 +_p (P_2 \oplus_q P_3) \equiv_{\mathrm{gen}} (P_1 +_p P_2) \oplus_q (P_1 +_p P_3)$$

Demostración: Sea $T \in \mathcal{PB}$. Aplicando reiteradamente las reglas $(EXT1)$, $(EXT2)$ y $(EXT3)$ obtenemos:

$$P_i \rightarrowtail^*_{r_i} P_i' \ \Rightarrow\ P_1 +_p P_2 \rightarrowtail^*_{r_1 \cdot r_2} P_1' +_p P_2' \ \wedge\ P_1 +_p P_3 \rightarrowtail^*_{r_1 \cdot r_3} P_1' +_p P_3'$$

$$\Rightarrow \begin{cases} (P_1 +_p P_2) \oplus_q (P_1 +_p P_3) \rightarrowtail^*_{q \cdot r_1 \cdot r_2} (P_1' +_p P_2') \\ (P_1 +_p P_2) \oplus_q (P_1 +_p P_3) \rightarrowtail^*_{(1-q) \cdot r_1 \cdot r_3} (P_1' +_p P_3') \end{cases}$$

Por otra parte se tiene:

$$P_i \rightarrowtail^*_{r_i} P_i' \;\Rightarrow\; P_2 \oplus_q P_3 \rightarrowtail^*_{q\cdot r_2} P_2' \;\wedge\; P_2 \oplus_q P_3 \rightarrowtail^*_{(1-q)r_2} P_3'$$

$$\Rightarrow \begin{cases} P_1 +_p (P_2 \oplus_q P_3) \rightarrowtail^*_{r_1\cdot q\cdot r_2} (P_1' +_p P_2') \\ P_1 +_p (P_2 \oplus_q P_3) \rightarrowtail^*_{r_1\cdot(1-q)\cdot r_3} (P_1' +_p P_3') \end{cases}$$

De estos resultados se deduce

$$(P_1 +_p P_2) \oplus_q (P_1 +_p P_3) \rightarrowtail^*_r (P_1' +_p P_2') \iff P_1 +_p (P_2 \oplus_q P_3) \rightarrowtail^*_r (P_1' +_p P_2')$$
$$(P_1 +_p P_2) \oplus_q (P_1 +_p P_3) \rightarrowtail^*_r (P_1' +_p P_3') \iff P_1 +_p (P_2 \oplus_q P_3) \rightarrowtail^*_r (P_1' +_p P_3')$$

Consideremos entonces los siguientes multiconjuntos de pares (proceso, probabilidad):

$$\tilde{P}_1 = \{\!\{ (P_1', r_1) \mid P_1 \rightarrowtail^*_{r_1} P_1' \}\!\}$$
$$\tilde{P}_2 = \{\!\{ (P_2', r_2) \mid P_2 \rightarrowtail^*_{r_2} P_2' \}\!\}$$
$$\tilde{P}_3 = \{\!\{ (P_3', r_3) \mid P_3 \rightarrowtail^*_{r_3} P_3' \}\!\}$$

A partir de los resultados anteriores obtenemos

$$\begin{aligned} & pass((P_1 +_p P_2) \oplus_q (P_1 +_p P_3)\,,\, T) \\ &= \textstyle\sum \{\!\{ q\cdot r_1\cdot r_2\cdot\ pass(P_1' +_p P_2'\,,\, T) \mid (P_1', r_1) \in \tilde{P}_1 \;\wedge\; (P_2', r_2) \in \tilde{P}_2 \}\!\} \\ &+ \textstyle\sum \{\!\{ (1-q)\cdot r_1\cdot r_3\cdot\ pass(P_1' +_p P_3'\,,\, T) \mid (P_1', r_1) \in \tilde{P}_1 \;\wedge\; (P_3', r_3) \in \tilde{P}_3 \}\!\} \\ &= \textstyle\sum \{\!\{ r_1\cdot q\cdot r_2\cdot\ pass(P_1' +_p P_2'\,,\, T) \mid (P_1', r_1) \in \tilde{P}_1 \;\wedge\; (P_2', r_2) \in \tilde{P}_2 \}\!\} \\ &+ \textstyle\sum \{\!\{ r_1\cdot (1-q)\cdot r_3\cdot\ pass(P_1' +_p P_3'\,,\, T) \mid (P_1', r_1) \in \tilde{P}_1 \;\wedge\; (P_3', r_3) \in \tilde{P}_3 \}\!\} \\ &= pass(P_1 +_p (P_2 \oplus_q P_3)\,,\, T) \end{aligned}$$

Este axioma se puede generalizar de forma trivial al caso de una elección interna generalizada.

$$\textbf{(DEIG)} \quad P +_p (\bigoplus_{i=1}^{n} [p_i]\, P_i) \equiv_{\text{gen}} \bigoplus_{i=1}^{n} [p_i]\, (P +_p P_i)$$

La distributividad contraria no se satisface en general en nuestro modelo probabilístico. Esto puede comprobarse mediante el siguiente

Ejemplo 5.6 Consideremos los procesos $P = a \oplus_{\frac{1}{2}} (b +_{\frac{1}{2}} c)$ y $Q = (a \oplus_{\frac{1}{2}} b) +_{\frac{1}{2}} (a \oplus_{\frac{1}{2}} c)$. Mientras que $pass(P\,,\, a; \omega) = \frac{1}{2}$, tenemos que $pass(Q\,,\, a; \omega) = \frac{3}{4}$. □

Como en el caso no probabilístico, el camino para mostrar la completitud del sistema de axiomas será encontrar la noción adecuada de forma normal. Con este objetivo en mente, y dado que en las formas normales aparecerá la elección externa generalizada en lugar de la binaria, daremos un axioma que indica como se comporta un proceso que es la elección externa (binaria) de dos elecciones externas generalizadas, y otro axioma que define el comportamiento de la composición mediante una elección interna de dos elecciones externas generalizadas en las cuales aparecen las mismas acciones y con las mismas probabilidades asociadas. Antes de pasar a enunciar ambos axiomas precisamos una definición auxiliar previa.

Definición 5.7 Sean $A, B \subseteq Act$. Dada una barba probabilística T, consideremos su conjunto de acciones iniciales $\tilde{T} = \{t_1, \ldots, t_u\}$. A partir de A, B, y $\tilde{T}$ construimos los siguientes conjunto de acciones:

$$T_A = \{t_i \mid t_i \in A \cap \tilde{T}\} \qquad T_B = \{t_i \mid t_i \in B \cap \tilde{T}\}$$
$$T_{A-B} = \{t_i \mid t_i \in (A - B) \cap \tilde{T}\} \qquad T_{B-A} = \{t_i \mid t_i \in (B - A) \cap \tilde{T}\}$$
$$T_{A \cap B} = \{t_i \mid t_i \in (A \cap B) \cap \tilde{T}\}$$

□

• **Axioma de expansión de la elección externa.**

Sean $A = \{a_1, \ldots, a_n\} \subseteq Act$ y $B = \{b_1, \ldots, b_m\} \subseteq Act$. Consideremos los procesos $P = \sum_{i=1}^{n} [p_i]\, a_i; P_i$ y $Q = \sum_{j=1}^{m} [q_j]\, b_j; Q_j$. Entonces se tiene

$$\textbf{(EBE)} \quad P +_p Q \equiv_{\text{gen}} R$$

siendo $R = \sum_{k=1}^{l} [r_k]\, c_k; R_k$, $C = \{c_1, \ldots, c_l\} = A \cup B$, donde

$$r_k = \begin{cases} p \cdot p_i & \text{si } c_k = a_i \in A - B \\ (1-p) \cdot q_j & \text{si } c_k = b_j \in B - A \\ p \cdot p_i + (1-p) \cdot q_j & \text{si } c_k = a_i = b_j \in A \cap B \end{cases}$$

$$R_k \;=\; \begin{cases} P_i & \text{si } c_k = a_i \in A - B \\ Q_j & \text{si } c_k = b_j \in B - A \\ P_i \oplus_{p'} Q_j & \text{si } c_k = a_i = b_j \in A \cap B \;\wedge\; p' = \frac{p \cdot p_i}{p \cdot p_i + (1-p) \cdot q_j} \end{cases}$$

Demostración: Para simplificar la notación tomaremos $p(P, a_i) = p_i$ y $p(Q, b_j) = q_j$. Aplicando $(EXT4)$ y $(EXT5)$ tenemos por un lado $P \xrightarrow{a}_q P' \Rightarrow P +_p Q \xrightarrow{a}_{p \cdot q} P'$, mientras que por el otro $Q \xrightarrow{a}_q P' \Rightarrow P +_p Q \xrightarrow{a}_{(1-p) \cdot q} P'$. Veremos entonces que para todo $T \in \mathcal{PB}$ tenemos $pass(P +_p Q\,,\, T) = pass(R\,,\, T)$.

Si T es una barba probabilística de la forma $T = \sum_{i=1}^{u} [s_i]\,(t_i; Nil) +_s \omega$, entonces

$$pass(P +_p Q\,,\, T) =$$

$$\frac{1-s}{(1-s) + \sum\limits_{t_i \in T_A} s \cdot s_i \cdot p \cdot p(P,t_i) \;+ \sum\limits_{t_i \in T_B} s \cdot s_i \cdot (1-p) \cdot p(Q,t_i)} =$$

$$\frac{1-s}{(1-s) + \sum\limits_{t_i \in T_{A-B}} s \cdot s_i \cdot p \cdot p(P,t_i) + \sum\limits_{t_i \in T_{B-A}} s \cdot s_i \cdot (1-p) \cdot p(Q,t_i) + \sum\limits_{t_i \in T_{A \cap B}} s \cdot s_i \cdot (p \cdot p(P,t_i) + (1-p) \cdot p(Q,t_i))} =$$

$$pass(R\,,\, T)$$

Si T es una barba probabilística de la forma $T = \sum_{i=1}^{u} [s_i]\; t_i; T_i$ donde $T_i = T'$ para $i = u$ y $T_i = Nil$ cuando $i \neq u$, distinguiremos cuatro casos:

$\diamond\; \exists\; 1 \leq i \leq n :\; a_i = t_u \in A - B.$

$$pass(P +_p Q\,,\, T) =$$

$$\frac{s_u \cdot p \cdot p_i \cdot pass(P_i\,,\, T')}{\sum\limits_{t_i \in T_A} s_i \cdot p \cdot p(P,t_i) \;+ \sum\limits_{t_i \in T_B} s_i \cdot (1-p) \cdot p(Q,t_i)} =$$

$$\frac{s_u \cdot p \cdot p_i \cdot pass(P_i\,,\, T')}{\sum\limits_{t_i \in T_{A-B}} s_i \cdot p \cdot p(P,t_i) \;+ \sum\limits_{t_i \in T_{B-A}} s_i \cdot (1-p) \cdot p(Q,t_i) \;+ \sum\limits_{t_i \in T_{A \cap B}} s_i \cdot (p \cdot p(P,t_i) + (1-p) \cdot p(Q,t_i))} =$$

$$pass(R\,,\, T)$$

$\diamond\ \exists\ 1 \leq j \leq m:\ b_j = t_u \in B - A$. Simétrico al anterior.

$\diamond\ \exists\ 1 \leq i \leq n,\ 1 \leq j \leq m:\ a_i = b_j = t_u \in A \cap B$.

$$pass(P +_p Q\ ,\ T) =$$

$$\frac{s_u{\cdot}p{\cdot}p_i{\cdot}\ pass(P_i\ ,T') + s_u{\cdot}(1{-}p){\cdot}q_j{\cdot}\ pass(Q_j\ ,T')}{\sum\limits_{t_i \in T_A} s_i{\cdot}p{\cdot}p(P,t_i)\ +\ \sum\limits_{t_i \in T_B} s_i{\cdot}(1{-}p){\cdot}p(Q,t_i)} =$$

$$\frac{s_u{\cdot}(p{\cdot}p_i{+}(1{-}p){\cdot}q_j){\cdot}\ pass(P_i \oplus_{q'} Q_j\ ,T')}{\sum\limits_{t_i \in T_{A-B}} s_i{\cdot}p{\cdot}p(P,t_i)\ +\ \sum\limits_{t_i \in T_{B-A}} s_i{\cdot}(1{-}p){\cdot}p(Q,t_i)\ +\ \sum\limits_{t_i \in T_{A\cap B}} s_i{\cdot}(p{\cdot}p(P,t_i){+}(1{-}p){\cdot}p(Q,t_i))} =$$

$$pass(R\ ,\ T)$$

donde $q' = \frac{p{\cdot}p_i}{p{\cdot}p_i{+}(1{-}p){\cdot}q_j}$.

$\diamond\ t_u \notin A \cup B$. En tal caso $pass(P +_p Q\ ,\ T) = pass(R\ ,\ T) = 0$.

• **Axioma de expansión de la elección interna.**

Sea $A = \{a_1, \ldots, a_n\} \subseteq Act$. Sean $P = \sum\limits_{i=1}^{n} [p_i]\ a_i; P_i$ y $Q = \sum\limits_{i=1}^{n} [p_i]\ a_i; Q_i$.

$$\textbf{(IBE)} \quad P \oplus_p Q \equiv_{\mathbf{gen}} R$$

siendo $R = \sum\limits_{i=1}^{n} [p_i]\ a_i; (P_i \oplus_p Q_i)$.

Demostración: Para simplificar la notación tomaremos $p(a_i) = p_i$. Hemos de ver que para todo $T \in \mathcal{PB}$ tenemos $pass(P \oplus_p Q\ ,\ T) = pass(R\ ,\ T)$.

Si T es una barba probabilística de la forma $T = \sum\limits_{i=1}^{u} [s_i]\ (t_i; Nil) +_s \omega$, entonces

$$\begin{aligned} pass(P \oplus_p Q\ ,\ T) &= p \cdot\ pass(P\ ,\ T)\ + (1-p) \cdot\ pass(Q\ ,\ T) \\ &= \frac{p{\cdot}(1{-}s)}{(1{-}s){+}\sum\limits_{t_i \in \tilde{T} \cap A} s{\cdot}s_i{\cdot}p(t_i)} + \frac{(1{-}p){\cdot}(1{-}s)}{(1{-}s){+}\sum\limits_{t_i \in \tilde{T} \cap A} s{\cdot}s_i{\cdot}p(t_i)} \\ &= \frac{(1{-}s)}{(1{-}s){+}\sum\limits_{t_i \in \tilde{T} \cap A} s{\cdot}s_i{\cdot}p(t_i)} = pass(R\ ,\ T) \end{aligned}$$

Si T es una barba probabilística de la forma $T = \sum_{i=1}^{u}[s_i]\ t_i; T_i$ donde $T_i = T'$ si $i = u$ y $T_i = Nil$ si $i \neq u$, distinguiremos dos casos:

$\diamond\ \exists\ 1 \leq i \leq n :\ a_i = t_u$.

$$pass(P \oplus_p Q\ ,\ T)\ \ p \cdot pass(P\ ,\ T) + (1-p) \cdot pass(Q\ ,\ T) =$$

$$\frac{p \cdot s_u \cdot p_i \cdot pass(P_i\ ,\ T')}{\sum\limits_{t_i \in \tilde{T} \cap A} s_i \cdot p(t_i)} + \frac{(1-p) \cdot s_u \cdot p_i \cdot pass(Q_i\ ,\ T')}{\sum\limits_{t_i \in \tilde{T} \cap A} s_i \cdot p(t_i)} =$$

$$\frac{s_u \cdot p_i \cdot pass(P_i \oplus_p Q_i\ ,\ T')}{\sum\limits_{t_i \in \tilde{T} \cap A} s_i \cdot p(t_i)} = pass(R\ ,\ T)$$

$\diamond\ t_u \notin A$. En tal caso $pass(P \oplus_p Q\ ,\ T) = pass(R\ ,\ T) = 0$.

Como se sigue inmediatamente de su definición, el operador de elección externa generalizada *generaliza* al operador prefijo, dado que el proceso $a_1; P_1$ puede escribirse como $\sum_{i=1}^{1}[1]\ a_i; P_i$. El operador binario $+_p$ también queda parcialmente generalizado, en el sentido de que si $a_1 \neq a_2$ entonces el proceso $(a_1; P_1) +_p (a_2; P_2)$ se reescribiría como $\sum_{i=1}^{2}[p_i]\ a_i; P_i$ donde $p_1 = p$ y $p_2 = 1 - p$.

El siguiente resultado nos muestra que tras las manipulaciones adecuadas el operador binario de elección externa puede ser totalmente eliminado, apareciendo en su lugar elecciones externas generalizadas.

Proposición 5.8 Utilizando los axiomas más arriba introducidos podemos eliminar toda aparición del operador binario $+_p$ en un proceso finito, recurriendo al operador de elección externa generalizada.

Demostración: Realizaremos la prueba por inducción estructural.

- Caso Base. Si $P = Nil$ el resultado es inmediato.
- Casos Inductivos. Sean $P_1, P_2 \in \mathrm{PPA}_{fin}$ tales que en ellos no aparece el operador binario de elección externa.

Si $P = a; P_1$ o bien $P = P_1 \oplus_p P_2$, el resultado es inmediato. En consecuencia, consideremos el caso $P = P_1 +_p P_2$.

Si uno de los procesos es *Nil*, aplicando el axioma (**NE**), tenemos

$$P_1 +_p Nil \equiv_{\mathbf{gen}} P_1$$

Si uno de los procesos es una elección interna generalizada $P_2 = \bigoplus_{i=1}^{n} [q_i]\, Q_i$, aplicando el axioma (**DEIG**), tenemos

$$P_1 +_p (\bigoplus_{i=1}^{n} [q_i]\, Q_i) \equiv_{\mathbf{gen}} \bigoplus_{i=1}^{n} [q_i]\, (P_1 +_p Q_i)$$

por lo que aplicando la hipótesis de inducción, el resultado queda probado.

Por último, si los dos procesos son elecciones externas generalizadas, basta aplicar el axioma (**EBE**) para obtener el resultado buscado.

□

5.1.2 Corrección del Sistema de Axiomas

Además de los axiomas que hemos presentado en la sección anterior necesitaremos un conjunto de reglas que indiquen que la relación $\equiv_{\mathbf{gen}}$ cumple una serie de propiedades *deseables*.

Las reglas de inferencia de nuestro sistema lógico son las presentadas en la Figura 5.1. Las reglas (**E1**), (**E2**) y (**E3**) indican que la igualdad es una relación de equivalencia, cumpliendo en consecuencia las propiedades reflexiva, simétrica y transitiva. Por su parte, las reglas (**C1**), (**C2**) y (**C3**) nos dicen que dicha relación de equivalencia es una congruencia respecto de los operadores básicos del lenguaje. Para probar la corrección de estas reglas tendríamos que demostrar que la relación $\approx_{\mathcal{G}}$ es una relación de equivalencia y una congruencia, pero esto se tiene indirectamente pues por el Corolario 4.10 tenemos que las relaciones $\approx_{\mathcal{G}}$ y $=_{\mathbf{PAT}}$ son equivalentes, y dado que la última de ellas está definida de forma composicional, es trivialmente una relación de equivalencia y una congruencia.

$$\textbf{(E1)}\ \frac{}{P \equiv_{\text{gen}} P} \qquad \textbf{(C1)}\ \frac{P \equiv_{\text{gen}} Q}{a;P \equiv_{\text{gen}} a;Q}$$

$$\textbf{(E2)}\ \frac{P \equiv_{\text{gen}} Q}{Q \equiv_{\text{gen}} P} \qquad \textbf{(C2)}\ \frac{P \equiv_{\text{gen}} Q \ \wedge\ P' \equiv_{\text{gen}} Q'}{P \oplus_p P' \equiv_{\text{gen}} Q \oplus_p Q'}$$

$$\textbf{(E3)}\ \frac{P \equiv_{\text{gen}} Q \ \wedge\ Q \equiv_{\text{gen}} R}{P \equiv_{\text{gen}} R} \qquad \textbf{(C3)}\ \frac{P \equiv_{\text{gen}} Q \ \wedge\ P' \equiv_{\text{gen}} Q'}{P +_p P' \equiv_{\text{gen}} Q +_p Q'}$$

Figura 5.1: Reglas de Inferencia.

Dados dos procesos P y Q escribiremos $\vdash P \equiv_{\text{gen}} Q$ si $P \equiv_{\text{gen}} Q$ se puede derivar a partir de los axiomas descritos en la sección anterior y de las reglas dadas en la Figura 5.1.

Teorema 5.1 (Corrección del Sistema de Axiomas)
Para todo par de términos $P,\ Q \in \text{PPA}_{fin}$ se tiene

$$\vdash P \equiv_{\text{gen}} Q \Longrightarrow P \approx_{\mathcal{G}} Q$$

Demostración: Corolario inmediato del hecho de que cada uno de los axiomas y reglas del sistema son correctos. □

5.1.3 Completitud del Sistema de Axiomas

Con el último resultado de la sección anterior hemos demostrado que si podemos derivar la equivalencia entre dos procesos *finitos* a partir del sistema de axiomas, entonces estos dos procesos son equivalentes respecto de la semántica de pruebas para el modelo generativo.

Pero para que nuestro sistema axiomático fuera realmente *útil* deberíamos ser capaces de garantizar el resultado recíproco; es decir, si tenemos la equivalencia entre dos procesos *finitos* respecto de la semántica de pruebas para el modelo generativo,

deberíamos ser capaces de poder demostrar que la equivalencia de estos dos procesos respecto de la relación $\equiv_{gen}$ se puede derivar a partir del sistema axiomático.

En esta sección demostraremos que nuestro sistema de axiomas cumple dicha propiedad (completitud). Como es habitual en las demostraciones de completitud de sistemas axiomáticos, nos basaremos en el uso de una noción de *forma normal.* Una vez introducida dicha noción, tendremos que demostrar que todo proceso del lenguaje PPA_{fin} se puede transformar mediante la aplicación sucesiva de los axiomas y reglas que forman nuestro sistema axiomático a un proceso que está en forma normal. Además, deberíamos demostrar que dadas dos formas normales distintas, existe una prueba que es pasada por dichas formas normales con probabilidades distintas. En consecuencia, las formas normales son únicas.

Casualmente, nuestras formas normales van a guardar gran similitud con las *formas normales* de la Sección 4.3. En concreto, van a ser elecciones internas generalizadas, de acuerdo a unas ciertas probabilidades, de elecciones externas generalizadas. Las acciones asociadas a las elecciones externas generalizadas *prefijan* a procesos que están en forma normal, con lo cual las formas normales serán procesos en los que se produce una alternancia estricta entre elecciones internas generalizadas y elecciones externas generalizadas. Además, no pueden existir dos elecciones externas generalizadas asociadas a la misma elección interna que tengan asociados los mismos conjuntos de acciones y con la misma distribución de probabilidad. La diferencia con las formas normales definidas en la Sección 4.3 es que, en este caso, las probabilidades asociadas a las elecciones internas generalizadas deben sumar uno, mientras que en aquel caso podían sumar menos que uno (cuando definamos las formas normales para procesos generales de PPA tendremos que la suma vuelve a poder ser menor que uno). Todo ello se formaliza por medio de la siguiente

Definición 5.9 Los procesos de PPA_{fin} que están en *forma normal* quedan definidos mediante la siguiente expresión BNF:

$$N ::= \bigoplus_{i=1}^{n} [p_i] \sum_{j=1}^{r_i} [p_{i,j}]\, a_{i,j} ; N$$

donde se han de cumplir las siguientes restricciones:

- $n \geq 1 \ \wedge \ \sum_{1=1}^{n} p_i = 1$
- $\forall\, 1 \leq i \leq n : p_i > 0 \ \wedge \ r_i \geq 0 \ \wedge \ \sum_{j=1}^{r_i} p_{i,j} = 1 \ \wedge \ \forall\, 1 \leq j \leq r_i : p_{i,j} > 0$
- $\forall\, 1 \leq i \leq n : \forall\, 1 \leq k, l \leq r_i : k \neq l \Rightarrow a_{i,k} \neq a_{i,l}$
- $\forall\, 1 \leq u, v \leq n,\ u \neq v : \{(a_{u,j}, p_{u,j})\}_{j=1}^{r_u} \neq \{(a_{v,j}, p_{v,j})\}_{j=1}^{r_v}$ □

Por abuso de notación, utilizaremos el proceso *Nil* para denotar a las elecciones externas generalizadas que no tengan acciones asociadas (caso $r_i = 0$). De igual forma, utilizaremos la notación prefijo $a; N$ cuando la elección externa generalizada tenga una única acción asociada (caso $r_i = 1$).

Para facilitar la escritura utilizaremos la siguiente notación alternativa para denotar a las formas normales:

$$N ::= \bigoplus_{A \in \mathcal{A}} [p_A] \sum_{(a,p_a) \in A} [p_a]\, a; N_{a,A}$$

donde los $\mathcal{A}$ son subconjuntos finitos pertenecientes a $\mathcal{P}(Act \times (0,1])$ tales que para todo $A \in \mathcal{A}$ se verifica $\sum \{\!|\, p_a \mid (a,p_a) \in A \,|\!\} = 1$.

A continuación veremos una serie de ejemplos en los que se presentan tanto procesos que están en forma normal como procesos que no lo están.

Ejemplo 5.10 Los siguientes procesos están en forma normal:

$a; Nil$ $\qquad$ $(a; Nil) \oplus_{\frac{1}{3}} (b; Nil)$

$a; ((b; Nil) \oplus_{\frac{1}{2}} Nil)$ $\qquad$ $(a; b; Nil) \oplus_{\frac{1}{2}} ((a; Nil) +_{\frac{1}{4}} (b; Nil))$

Por el contrario, los siguientes procesos no están en forma normal:

$(a; Nil) \oplus_{\frac{2}{5}} (a; b; Nil)$ $\qquad$ $(a; Nil) +_{\frac{3}{4}} (a; b; Nil)$ $\qquad$ $(a; Nil) +_{\frac{1}{6}} Nil$

□

Una vez que hemos introducido las formas normales, veamos que dadas dos formas normales distintas, existe una prueba que las distingue. Ello se tiene como consecuencia del Teorema 4.2, pues las formas normales son un caso particular de los procesos dados por la Definición 4.12.

Corolario 5.2 (Unicidad de Formas Normales)

Sean N_1, N_2 dos procesos distintos que están en forma normal. Entonces

$$\exists\, T \in \mathcal{PB} :\ pass(N_1\,,\, T) \neq pass(N_2\,,\, T)$$

donde, naturalmente, la igualdad entre formas normales se debe entender salvo conmutatividad de los operadores de elección. □

Veamos ahora que utilizando el sistema axiomático cualquier proceso se puede transformar en otro que está en forma normal.

Teorema 5.3 Dado un proceso $P \in \mathrm{PPA}_{fin}$ existe un proceso N en forma normal, tal que $\vdash P \equiv_{\mathbf{gen}} N$.

Demostración: La demostración la realizaremos por inducción estructural.

- Si $P = Nil$, entonces P ya está en forma normal sin más que considerar la forma normal correspondiente a $\mathcal{A} = \{\emptyset\}$.
- Si $P = a; P'$, por hipótesis de inducción P' puede ser transformado en una forma normal N', por lo que tomando $\mathcal{A} = \{\{(a,1)\}\}$, $N_{a,A} = N'$ y aplicando la regla (**C1**) obtenemos una forma normal N tal que $P \equiv_{\mathbf{gen}} N$.
- Si $P = P_1 \oplus_p P_2$, por hipótesis de inducción P_1 y P_2 pueden ser transformados en su respectiva forma normal

  $$P_1 \equiv_{\mathbf{gen}} N_1 \ \wedge\ P_2 \equiv_{\mathbf{gen}} N_2$$

  donde $N_1 = \bigoplus_{A\in\mathcal{A}} [p_A] \sum_{(a,p_a)\in A} [p_a]\, a; P_{a,A}$ y $N_2 = \bigoplus_{B\in\mathcal{B}} [q_B] \sum_{(b,q_b)\in B} [q_b]\, b; Q_{b,B}$.

  Aplicando la regla (**C2**) obtenemos $P_1 \oplus_p P_2 \equiv_{\mathbf{gen}} N_1 \oplus_p N_2$. Aplicando el axioma (**IBE**), si ello fuera necesario, dado que toda elección interna generalizada (finita) se puede descomponer en elecciones internas binarias y viceversa, obtenemos la forma normal

  $$N = \bigoplus_{C\in\mathcal{C}} [r_C] \sum_{(c,r_c)\in C} [r_c]\, c; R_{c,C}$$

donde $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$ y $\forall C \in \mathcal{C}$

$$\begin{array}{lll} C = A \in \mathcal{A} - \mathcal{B} & \Rightarrow & r_C = p \cdot p_A \ \wedge \ \forall\, c \in C : \ R_{c,C} = P_{c,A} \\ C = B \in \mathcal{B} - \mathcal{A} & \Rightarrow & r_C = (1-p) \cdot q_B \ \wedge \ \forall\, c \in C : \ R_{c,C} = Q_{c,B} \\ C = A = B \in \mathcal{A} \cap \mathcal{B} & \Rightarrow & r_C = p \cdot p_A + (1-p) \cdot q_B \ \wedge \\ & & \forall\, c \in C : \ R_{c,C} = P_{c,A} \oplus_p Q_{c,B} \end{array}$$

En los dos primeros casos se obtiene una expresión que ya está en forma normal, mientras que en el último podemos aplicar la hipótesis de inducción a los procesos $P_{c,A}$ y $Q_{c,B}$, para obtener la forma normal buscada. En consecuencia, hemos conseguido una forma normal N tal que $N_1 \oplus_p N_2 \equiv_{\mathbf{gen}} N$, con lo cual, aplicando la regla (**E3**), obtenemos $P_1 \oplus_p P_2 \equiv_{\mathbf{gen}} N$.

- Si $P = P_1 +_p P_2$, tenemos de nuevo por hipótesis de inducción

$$P_1 \equiv_{\mathbf{gen}} N_1 \ \wedge \ P_2 \equiv_{\mathbf{gen}} N_2$$

donde $N_1 = \bigoplus_{A \in \mathcal{A}} [p_A] \sum_{(a,p_a) \in A} [p_a]\, a; P_{a,A}$ y $N_2 = \bigoplus_{B \in \mathcal{B}} [q_B] \sum_{(b,q_b) \in B} [q_b]\, b; Q_{b,B}$.

Aplicando la regla (**C3**) obtenemos $P_1 +_p P_2 \equiv_{\mathbf{gen}} N_1 +_p N_2$. Entonces, aplicando reiteradamente el axioma (**DEIG**), obtenemos el proceso

$$P' = \bigoplus_{\substack{A \in \mathcal{A} \\ B \in \mathcal{B}}} [p_A \cdot q_B] \left(\Big(\sum_{(a,p_a) \in A} [p_a]\, a; P_{a,A} \Big) +_p \Big(\sum_{(b,q_b) \in B} [q_b]\, b; Q_{b,B} \Big) \right)$$

que es la elección interna generalizada entre procesos que son una elección externa entre dos procesos. Veamos entonces como es cada uno de los procesos de la forma

$$\Big(\sum_{(a,p_a) \in A} [p_a]\, a; P_{a,A} \Big) +_p \Big(\sum_{(b,q_b) \in B} [q_b]\, b; Q_{b,B} \Big)$$

Si alguno de los dos conjuntos A o B es vacío, es decir, si el correspondiente proceso es Nil, entonces aplicando el axioma (**NE**) obtendríamos como resultado el otro proceso. En caso contrario, aplicando el axioma (**EBE**), obtenemos que el proceso en cuestión es equivalente bajo $\equiv_{\mathbf{gen}}$ al proceso

$$R_{A,B} = \sum_{(c,r_c) \in C} [r_c]\, c; R_{c,A,B}$$

donde $C = \{(c_1, r_1), \ldots, (c_l, r_l) \mid \exists\, q :\ (c_i, q) \in A \cup B\}$ y

$$r_c = \begin{cases} p \cdot q & \text{si } \exists\, q :\ (c,q) \in A \wedge \nexists\, r :\ (c,r) \in B \\ (1-p) \cdot q & \text{si } \exists\, q :\ (c,q) \in B \wedge \nexists\, r :\ (c,r) \in A \\ p \cdot p_1 + (1-p) \cdot p_2 & \text{si } \exists\, p_1, p_2 :\ (c,p_1) \in A \wedge (c,p_2) \in B \end{cases}$$

$$R_{c,A,B} = \begin{cases} P_{c,A} & \text{si } \exists\, q :\ (c,q) \in A \wedge \nexists\, r :\ (c,r) \in B \\ Q_{c,B} & \text{si } \exists\, q :\ (c,q) \in B \wedge \nexists\, r :\ (c,r) \in A \\ P_{c,A} \oplus_{p'} Q_{c,B} & \text{si } \exists\, p_1, p_2 :\ (c,p_1) \in A \wedge (c,p_2) \in B \\ & \wedge\ p' = \frac{p \cdot p_1}{p \cdot p_1 + (1-p) \cdot p_2} \end{cases}$$

Al igual que en caso de la elección interna, los procesos $R_{c,A,B}$ o están en forma normal, lo que sucede en los dos primeros casos, o bien se pueden transformar aplicando la hipótesis de inducción a forma normal (tercer caso).

Pero el proceso obtenido no está todavía necesariamente en forma normal, dado que cuando consideramos los procesos que nos han aparecido al componer dos elecciones externas generalizadas mediante una elección externa, podemos generar de distintas formas un mismo conjunto C apareciendo en la definición del proceso obtendido R. Es decir, dado un conjunto C, pueden existir varios $A \in \mathcal{A}$ y varios $B \in \mathcal{B}$ tales que al combinarlos nos de el mismo C. En tal caso tendremos que agrupar todas las elecciones externas generalizadas que tengan asociadas las mismas acciones con las mismas probabilidades.

A partir de los conjuntos originales, la forma en la cual se calculan las probabilidades asociadas a las acciones que forman parte de las elecciones externas generalizadas viene dada por la función $\cup_p$ dada en la Definición 4.26, con lo que aplicando el axioma (**IBE**), obtenemos

$$P'' = \bigoplus_{C \in \mathcal{C}} [r_C] \sum_{(c,r_c) \in C} [r_c]\, c; \Big(\bigoplus_{\substack{A \in \mathcal{A},\, B \in \mathcal{B} \\ C = A \cup_p B}} [\frac{p_A \cdot q_B}{r_C}]\, R_{c,A,B} \Big)$$

donde $\mathcal{C} = \{A \cup_p B \mid A \in \mathcal{A} \wedge B \in \mathcal{B}\}$, $r_C = \sum_{\substack{A \in \mathcal{A}, B \in \mathcal{B} \\ A \cup_p B = C}} p_A \cdot q_B$, y

$$R_{c,A,B} = \begin{cases} P_{c,A} & \text{si } \exists\, q:\ (c,q) \in A \wedge \nexists\, r:\ (c,r) \in B \\ Q_{c,B} & \text{si } \exists\, q:\ (c,q) \in B \wedge \nexists\, r:\ (c,r) \in A \\ P_{c,A} \oplus_{p'} Q_{c,B} & \text{si } \exists\, p_1, p_2:\ (c,p_1) \in A \wedge (c,p_2) \in B \\ & \quad \wedge\ p' = \frac{p \cdot p_1}{p \cdot p_1 + (1-p) \cdot p_2} \end{cases}$$

Aplicando la hipótesis de inducción, los procesos de la forma

$$\bigoplus_{\substack{A \in \mathcal{A}, B \in \mathcal{B} \\ C = A \cup_p B}} [\frac{p_A \cdot q_B}{r_C}]\, R_{c,A,B}$$

se pueden transformar en su forma normal, obteniéndose a partir de P'' un proceso P''', de modo que aplicando la regla (**E3**), concluimos $P_1 +_p P_2 \equiv_{\mathbf{gen}} P'''$, donde P''' ya está en forma normal.

□

Teorema 5.4 (Completitud del Sistema de Axiomas)
Para todo par de procesos $P, Q \in \mathrm{PPA}_{fin}$ se tiene

$$P \approx_{\mathcal{G}} Q \implies \vdash P \equiv_{\mathbf{gen}} Q$$

Demostración: Inmediata a partir de los Teoremas 5.2 y 5.3. □

Corolario 5.5 Para todo par de procesos $P, Q \in \mathrm{PPA}_{fin}$ se tiene

$$P \approx_{\mathcal{G}} Q \iff \vdash P \equiv_{\mathbf{gen}} Q$$

□

5.2 Sistema de Axiomas para PPA

En esta sección extenderemos los resultados obtenidos en la sección anterior para procesos finitos, a la totalidad de los de PPA. Al igual que hicimos en el caso de la

semántica denotacional para el modelo generativo, los procesos recursivos del lenguaje se tratarán mediante su aproximación por *procesos finitos*, de forma que la semántica de los procesos recursivos vendrá definida por el límite de la cadena que forman los valores semánticos de sus aproximaciones finitas. Como es usual, tales aproximaciones finitas empezarán con el proceso sintáctico cuya semántica es el menor elemento del dominio semántico. Al igual que ocurría en el modelo denotacional para el modelo generativo, este proceso será Ω. La definición formal de las aproximaciones finitas de un proceso viene dada por la siguiente

Definición 5.11 Sea P un proceso definido mediante la expresión recursiva $recX.P'$. Definimos las *aproximaciones finitas* del proceso P en la forma

$$P^0 = \Omega$$
$$P^{n+1} = P'\{P^n/X\}$$

Definimos el conjunto de *aproximaciones finitas* de P como $APX(P) = \{P^n : n \in \mathbb{N}\}$. Para procesos finitos podemos considerar que sus aproximaciones finitas son iguales a él mismo. □

A nivel sintáctico cada aproximación finita de un proceso es un proceso finito, con lo cual podemos utilizar el estudio que hemos realizado en la sección previa para procesos finitos a la hora de razonar sobre las aproximaciones finitas, si bien al efecto hay que tener en cuenta que en las aproximaciones finitas ahora puede aparecer como primitivo el proceso Ω que no admitíamos en el lenguaje PPA_{fin} de procesos finitos.

Como vamos a caracterizar los procesos recursivos mediante sus aproximaciones finitas, en lugar de axiomatizar una relación de equivalencia entre procesos, como hicimos en el caso finito, tendremos que axiomatizar una relación de orden, la cual denotaremos por $\sqsubseteq_{\mathbf{gen}}$. La equivalencia quedará como noción derivada generada por el cierre antisimétrico de la relación de orden con la que trabajaremos. Este hecho conlleva como consecuencia que a la hora de probar la corrección y completitud del nuevo sistema de axiomas con respecto a la semántica de pruebas tendríamos que considerar la relación de orden entre procesos $\preceq_{\mathcal{G}}$ inducida por dicha semántica de

pruebas, la cual viene dada por

$$P \preceq_{\mathcal{G}} Q \Longleftrightarrow \forall\, T \in \mathcal{G} :\ pass(P\,,\,T) \leq pass(Q\,,\,T)$$

Pero esto no va a ser necesario, dado que en relación con la semántica denotacional definida para el modelo generativo teníamos la noción de orden $\sqsubseteq_{\mathbf{PAT}}$ (ver Definición 4.22), de modo que en lugar de estudiar la corrección y completitud del sistema de axiomas respecto de la semántica de pruebas, la estudiaremos con respecto de la semántica denotacional. Es decir, demostraremos que

$$[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!] \iff \vdash P \sqsubseteq_{\mathbf{gen}} Q$$

A partir de dicho resultado se deduce trivialmente

$$[\![P]\!] =_{\mathbf{PAT}} [\![Q]\!] \iff \vdash P \equiv_{\mathbf{gen}} Q$$

con lo cual, por el Teorema 4.10, obtenemos

$$P \approx_{\mathcal{G}} Q \iff \vdash P \equiv_{\mathbf{gen}} Q$$

5.2.1 Nuevos Axiomas y Reglas

En esta sección enumeraremos los axiomas y reglas que pasan a configurar nuestro nuevo sistema axiomático. Respecto de los axiomas, aumentaremos el conjunto de axiomas dado en la sección anterior con una serie de axiomas que tratan al proceso Ω. Además, como punto de partida, necesitaremos el axioma que nos indica que el proceso *Nil* es menor o igual a sí mismo. La demostración de corrección de todos los siguientes axiomas es trivial.

$$\textbf{(D)}\quad \Omega \sqsubseteq_{\mathbf{gen}} P \qquad\qquad \textbf{(DI)}\quad P \oplus_p \Omega \sqsubseteq_{\mathbf{gen}} P$$

$$\textbf{(DE)}\quad P +_p \Omega \equiv_{\mathbf{gen}} \Omega \qquad\qquad \textbf{(N)}\quad \mathit{Nil} \sqsubseteq_{\mathbf{gen}} \mathit{Nil}$$

El nuevo sistema de reglas se presenta en la Figura 5.2. Para construir el mismo sustituimos en el sistema presentado en la Figura 5.1 las reglas **(E1)**, **(E2)** y **(E3)**,

(O1)	$\dfrac{P \sqsubseteq_{\mathbf{gen}} Q \;\wedge\; Q \sqsubseteq_{\mathbf{gen}} P}{P \equiv_{\mathbf{gen}} Q}$	**(O2)**	$\dfrac{P \equiv_{\mathbf{gen}} Q}{P \sqsubseteq_{\mathbf{gen}} Q}$
(O3)	$\dfrac{P \equiv_{\mathbf{gen}} Q}{Q \sqsubseteq_{\mathbf{gen}} P}$	**(O4)**	$\dfrac{P \sqsubseteq_{\mathbf{gen}} Q \;\wedge\; Q \sqsubseteq_{\mathbf{gen}} R}{P \sqsubseteq_{\mathbf{gen}} R}$
(C1')	$\dfrac{P \sqsubseteq_{\mathbf{gen}} Q}{a;P \sqsubseteq_{\mathbf{gen}} a;Q}$	**(C2')**	$\dfrac{P \sqsubseteq_{\mathbf{gen}} Q \;\wedge\; P' \sqsubseteq_{\mathbf{gen}} Q'}{P +_p P' \sqsubseteq_{\mathbf{gen}} Q +_p Q'}$
(C3')	$\dfrac{P \sqsubseteq_{\mathbf{gen}} Q \;\wedge\; P' \sqsubseteq_{\mathbf{gen}} Q'}{P \oplus_p P' \sqsubseteq_{\mathbf{gen}} Q \oplus_p Q'}$	**(OI1)**	$\dfrac{P \sqsubseteq_{\mathbf{gen}} Q}{P \sqsubseteq_{\mathbf{gen}} P \oplus_p Q}$
(OI2)	$\dfrac{P \sqsubseteq_{\mathbf{gen}} Q}{P \oplus_p Q \sqsubseteq_{\mathbf{gen}} Q}$	**(R1)**	$\dfrac{}{P\{recX.P/X\} \sqsubseteq_{\mathbf{gen}} recX.P}$
(R2)	$\dfrac{\forall\, Q \in APX(P):\; Q \sqsubseteq_{\mathbf{gen}} R}{P \sqsubseteq_{\mathbf{gen}} R}$	**(R3)**	$\dfrac{\forall\, n \in \mathbb{N}:\; P \oplus_{\frac{n-1}{n}} \Omega \sqsubseteq_{\mathbf{gen}} R}{P \sqsubseteq_{\mathbf{gen}} R}$

Figura 5.2: Nuevo Sistema de Reglas.

que indicaban que $\equiv_{\mathbf{gen}}$ era una relación de equivalencia, por una serie de reglas que indican que la relación $\sqsubseteq_{\mathbf{gen}}$ es una relación de orden. De igual forma, sustituiremos las reglas **(C1)**, **(C2)** y **(C3)**, que indicaban la congruencia de los operadores respecto de la relación $\equiv_{\mathbf{gen}}$, por otras similares indicando la precongruencia con respecto a la relación $\sqsubseteq_{\mathbf{gen}}$. La prueba de corrección de todas estas reglas es trivial dada la definición composicional de la semántica denotacional.

Las reglas **(OI1)** y **(OI2)** indican que dados dos procesos P y Q que verifiquen $P \sqsubseteq_{\mathbf{gen}} Q$, la elección interna entre ambos procesos, $P \oplus_p Q$, ocupa una posición intermedia entre ambos respecto de la relación de orden $\sqsubseteq_{\mathbf{gen}}$.

Lema 5.12 Las reglas (**OI1**) y (**OI2**) son correctas.

Demostración: Sean P, Q dos procesos tales que $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$. Entonces, para toda secuencia s y todo estado A se verifica $p(P, s, A) \leq p(Q, s, A)$. Aplicando la definición del operador semántico de elección interna obtenemos

$$p(P \oplus_p Q, s, A) = p \cdot p(P, s, A) + (1 - q) \cdot p(Q, s, A)$$

de lo cual se deduce trivialmente que para toda secuencia s, todo estado A, y cada $p \in (0, 1)$ se verifica

$$p(P, s, A) \leq p(P \oplus_p Q, s, A) \leq p(Q, s, A)$$

A partir de este resultado se deriva inmediatamente la corrección de las reglas (**OI1**) y (**OI2**). □

Respecto a las reglas que hacen referencia a la recursión, las dos primeras aparecen de costumbre en los sistemas de axiomas que trabajan sobre lenguajes que incluyen un operador de recursión (e.g. [Hen88]). La demostración de su corrección se sigue de la caracterización de la caracterización de la semántica de un proceso definido de forma recursiva:

$$[\![recX.\ P]\!] = \bigsqcup_{n=0}^{\infty} [\![P^n]\!]$$

En concreto, la corrección de la regla (**R1**) es trivial dado que

$$[\![P\{recX.P/X\}]\!] = \bigsqcup_{n=1}^{\infty} [\![P^n]\!]$$

La corrección de la regla (**R2**) proviene del hecho de que se está trabajando en un *cpo*, en el que ha de verificarse que $[\![P]\!]$ es la cota superior mínima de la cadena $\{[\![P^i]\!]\}_{i=1}^{\infty}$.

La regla (**R3**) se incorpora a nuestro sistema de reglas por razones *técnicas*. Al igual que ocurría en [Cua93], el dominio sobre el que estamos trabajando no es ω-algebraico, con lo cual necesitaremos esta regla para demostrar la completitud del sistema axiomático. Cuando llegue el momento en el que utilicemos esta regla comentaremos con profundidad su significado y consecuencias.

Lema 5.13 La regla (**R3**) es correcta.

Demostración: Supongamos que para todo $n \in \mathbb{N}$ se verifica

$$[\![P \oplus_{\frac{n-1}{n}} \Omega]\!] \sqsubseteq_{\mathbf{PAT}} [\![R]\!]$$

Esto es equivalente a decir que para todo $n \in \mathbb{N}$, para toda secuencia s y para todo estado A se cumple

$$p(P \oplus_{\frac{n-1}{n}} \Omega, s, A) \leq p(R, s, A)$$

A partir de la definición de la función semántica asociada a la elección interna obtenemos que para toda secuencia s y para todo estado A se cumple

$$p(P \oplus_{\frac{n-1}{n}} \Omega, s, A) = \frac{n-1}{n} \cdot p(P, s, A) + \frac{1}{n} \cdot p(\Omega, s, A) = \frac{n-1}{n} \cdot p(P, s, A)$$

Si consideramos los dos resultados anteriores, obtenemos que para toda secuencia s y para todo estado A se cumple

$$p(P, s, A) = \lim_{n \to \infty} \frac{n-1}{n} \cdot p(P, s, A) \leq p(R, s, A)$$

de lo cual se deduce inmediatamente $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![R]\!]$. □

Lema 5.14 Sea $P = recX.P'$, tal que en P' no aparece el operador de recursión. Para toda aproximación finita $P^n \in APX(P)$ se tiene $\vdash P^n \sqsubseteq_{\mathbf{gen}} P$.

Demostración: Procederemos por inducción respecto de n.

Si $n = 0$, aplicando el axioma (**D**), tenemos

$$P^0 = \Omega \sqsubseteq_{\mathbf{gen}} recX.P'$$

Consideremos el caso de la aproximación $n+1$. Por definición de las aproximaciones finitas tenemos $P^{n+1} = P'\{P^n/X\}$. Dado que en el proceso P' la variable X no aparece en el ámbito de un operador de recursión, si en lugar de sustituir en P' la variable X por el proceso P^n la sustituimos por un proceso que sea mayor o igual que él (i.e. un proceso Q tal que $P^n \sqsubseteq_{\mathbf{gen}} Q$), aplicando reiteradamente las reglas (**C1'**), (**C2'**), y (**C3'**) obtendremos $P'\{P^n/X\} \sqsubseteq_{\mathbf{gen}} P'\{Q/X\}$. Por hipótesis de inducción tenemos $P^n \sqsubseteq_{\mathbf{gen}} P = recX.P'$, con lo cual

$$P'\{P^n/X\} \sqsubseteq_{\mathbf{gen}} P'\{recX.P'/X\}$$

y aplicando la regla (**R1**) obtenemos

$$P'\{recX.P'/X\} \sqsubseteq_{\mathbf{gen}} recX.P' = P$$

con lo que encadenando los resultados anteriores llegamos a

$$\vdash P^{n+1} \sqsubseteq_{\mathbf{gen}} P$$

□

El resultado anterior nos indica que dado un proceso recursivo $recX.P$, de modo que en P no aparecen definiciones recursivas anidadas, podemos inferir a partir del sistema de axiomas que cualquiera de sus aproximaciones finitas es *menor* que él. Al igual que en el caso no probabilístico, el lema anterior se puede generalizar a procesos que contengan más de un operador de recursión. Primero tendríamos que extender nuestro sistema axiómatico para que considere términos de PPA que contengan variables libres, dado que en nuestro sistema axiomático sólo trabajamos con procesos (es decir, términos sin variables libres). Para ello es suficiente con extender el axioma (**D**) en la forma $\Omega \sqsubseteq_{\mathbf{gen}} X$ para todo $X \in Id$. A continuación tendremos que demostrar un caso particular de la propiedad llamada en [Hen88] de *sustitución generalizada*, que dice que si $P_1 \sqsubseteq_{\mathbf{gen}} P_2$, entonces para todo P se verifica $P\{P_1/X\} \sqsubseteq_{\mathbf{gen}} P\{P_2/X\}$ (donde P_1, P_2, P pueden contener variables libres y X aparece libre en P). Pero dado que el proceso de adaptar nuestro sistema de axiomas para tratar con términos abiertos para llevar a cabo la correspondiente demostración no difiere en nada con respecto al caso no probabilístico (ver Capítulo 4 de [Hen88]) hemos preferido omitir la demostración de esta extensión.

Nótese que a partir del axioma (**N**), de $\Omega \equiv_{\mathbf{gen}} \Omega$ (derivable a partir de los axiomas (**D**) y (**O1**)), utilizando inducción estructural, se puede inferir $P \equiv_{\mathbf{gen}} P$ para procesos finitos; mientras que utilizando la regla (**R1**) y el lema anterior, junto con la regla (**R2**), podemos demostrar dicha equivalencia para los procesos infinitos.

En la Figura 5.3 presentamos agrupados todos los axiomas que constituyen el nuevo conjunto de axiomas, con lo cual nuestro sistema axiomático quedará formado por las reglas que aparecen en la Figura 5.2 junto con los axiomas que aparecen en la Figura 5.3.

(II)	$P \oplus_p P \equiv_{gen} P$
(CI)	$P \oplus_p Q \equiv_{\mathbf{gen}} Q \oplus_{1-p} P$
(AI)	$P \oplus_p (Q \oplus_q R) \equiv_{\mathbf{gen}} (P \oplus_{p'} Q) \oplus_{q'} R$
(CE)	$P +_p Q \equiv_{\mathbf{gen}} Q +_{1-p} P$
(NE)	$P +_p Nil \equiv_{\mathbf{gen}} P$
(DEIG)	$P +_p (\bigoplus_{i=1}^{n}[p_i]\, P_i) \equiv_{\mathbf{gen}} \bigoplus_{i=1}^{n}[p_i]\, (P +_p P_i)$
(EBE)	$\sum_{i=1}^{n}[p_i]\, a_i; P_i +_p \sum_{j=1}^{m}[q_j]\, b_j; Q_j \equiv_{\mathbf{gen}} \sum_{k=1}^{l}[r_k]\, c_k; R_k$
(IBE)	$\sum_{i=1}^{n}[p_i]\, a_i; P_i \oplus_p \sum_{i=1}^{n}[p_i]\, a_i; Q_i \equiv_{\mathbf{gen}} \sum_{i=1}^{n}[p_i]\, a_i; (P_i \oplus_p Q_i)$
(D)	$\Omega \sqsubseteq_{\mathbf{gen}} P$
(DI)	$P \oplus_p \Omega \sqsubseteq_{\mathbf{gen}} P$
(DE)	$P +_p \Omega \equiv_{\mathbf{gen}} \Omega$
(N)	$Nil \equiv_{\mathbf{gen}} Nil$

Figura 5.3: Conjunto de Axiomas.

Teorema 5.6 (Corrección del Sistema de Axiomas)
Para todo par de términos P, $Q \in$ PPA tenemos

$$\vdash P \sqsubseteq_{\mathbf{gen}} Q \Longrightarrow [\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$$

Demostración: Corolario inmediato del hecho de que todos los axiomas y reglas del sistema son correctos. □

5.2.2 Completitud del nuevo Sistema de Axiomas

Al igual que cuando trabajamos con procesos finitos, para la prueba de completitud del sistema axiomático comenzaremos por introducir la correspondiente noción de forma normal.

La única diferencia entre esta definición de formas normales y la que presentamos en la Definición 5.9 para los procesos finitos, es que en la nueva definición la

suma de las probabilidades asociadas con las elecciones internas generalizadas puede ser menor que 1. En particular, podremos tener elecciones internas generalizadas que no contengan ningún argumento. Al igual que ocurría cuando estudiamos la semántica denotacional, la diferencia entre la suma de las probabilidades asociadas a las elecciones internas generalizadas y 1 denotará la probabilidad de que el proceso diverja en ese preciso punto.

Definición 5.15 Los procesos de PPA que están en *forma normal* quedan definidos mediante la siguiente expresión BNF:

$$N ::= \bigoplus_{i=1}^{n} [p_i] \sum_{j=1}^{r_i} [p_{i,j}]\, a_{i,j}\,;N$$

donde se cumplen las siguientes restricciones:

- $n \geq 0 \ \wedge \ \sum_{1=1}^{n} p_i \leq 1$
- $\forall\, 1 \leq i \leq n : p_i > 0 \ \wedge \ r_i \geq 0 \ \wedge \ \sum_{j=1}^{r_i} p_{i,j} = 1 \ \wedge \ \forall\, 1 \leq j \leq r_i : p_{i,j} > 0$
- $\forall\, 1 \leq i \leq n : \forall\, 1 \leq k, l \leq r_i, k \neq l : a_{i,k} \neq a_{i,l}$
- $\forall\, 1 \leq u, v \leq n, u \neq v : \{(a_{u,j}, p_{u,j})\}_{j=1}^{r_u} \neq \{(a_{v,j}, p_{v,j})\}_{j=1}^{r_v}$ □

Al igual que hicimos en el caso finito, utilizaremos el proceso *Nil* para denotar a las elecciones externas generalizadas que no tengan acciones asociadas (caso $r_i = 0$), mientras que utilizamos el proceso Ω para denotar una elección interna generalizada entre 0 procesos (caso $n = 0$). De igual forma, utilizaremos el prefijo $a;N$ cuando la elección externa generalizada tenga una única acción asociada.

En esta ocasión utilizaremos de nuevo una notación alternativa de la forma:

$$N ::= \bigoplus_{A \in \mathcal{A}} [p_A] \sum_{(a,p_a) \in A} [p_a]\, a;N_{a,A}$$

donde los $\mathcal{A}$ son subconjuntos finitos de $\mathcal{P}(Act \times (0,1])$ tales que para todo $A \in \mathcal{A}$ se verifica $\sum \{\!|\, p_a \mid (a,p_a) \in A \,|\!\} = 1$.

Una vez que hemos introducido las nuevas formas normales, para demostrar la completitud del sistema axiomático, habremos de demostrar en primer lugar que

cualquier proceso *finito* puede ser transformado en su forma normal mediante el uso de las reglas y axiomas que forman el sistema axiomático.

Teorema 5.7 Dado un proceso $P \in$ PPA finito, en el sentido de que en el mismo no aparece el operador de recursión, existe un proceso N que está en forma normal tal que $\vdash P \equiv_{\mathbf{gen}} N$.

Demostración: Análoga a la del Teorema 5.3 para procesos finitos, al estar el proceso Ω en forma normal. □

A continuación probaremos que si dos procesos están relacionados mediante la relación de orden $\sqsubseteq_{\mathbf{PAT}}$, entonces también están relacionados mediante la relación $\sqsubseteq_{\mathbf{gen}}$. La demostración es compleja, debiéndose de distinguir si los procesos a comparar son finitos o no. Por ello la ordenaremos mediante una sucesión de lemas, cada uno de los cuales cubre uno de los casos posibles.

Lema 5.16 Sean $P, Q \in$ PPA procesos finitos. Entonces, $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!] \Longrightarrow P \sqsubseteq_{\mathbf{gen}} Q$.

Demostración: Los procesos dados P y Q pueden ser transformados a procesos en forma normal utilizando el sistema axiomático, por lo que podemos restringirnos al estudio de dichas formas normales equivalentes. En consecuencia, tomemos P y Q de la forma

$$P = \bigoplus_{A \in \mathcal{A}} [p_A] \sum_{(a,p_a) \in A} [p_a]\, a; P_{a,A}$$

$$Q = \bigoplus_{B \in \mathcal{B}} [q_B] \sum_{(b,q_b) \in B} [q_b]\, b; Q_{b,B}$$

donde $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}(\Sigma \times (0,1])$. Nótese que $p(P, \epsilon, A) = p_A \ \wedge \ p(Q, \epsilon, B) = q_B$.

La demostración la realizaremos por inducción respecto de la complejidad de los procesos, donde entendemos por complejidad su profundidad, y en el caso de dos procesos con la misma profundidad consideraremos que un proceso es más complejo que otro si los estados de este último están contenidos en los estados del primero.

En primer lugar podemos suponer $P \neq \Omega$, dado que si $P = \Omega$ el resultado es inmediato por aplicación del axioma (**D**). Si $P \neq \Omega$, se nos presentan tres casos a considerar:

- Los conjuntos $\mathcal{A}$ y $\mathcal{B}$ son distintos.
  Dado que $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$, tenemos que debe existir un conjunto B' tal que $B' \in \mathcal{B} - \mathcal{A}$. Además, el resto de los conjuntos que pertenecen a $\mathcal{B}$ tendrán una probabilidad asociada mayor o igual en el proceso Q de la que tengan en el proceso P, mientras que puesto que $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$, no puede existir ningún $A \in \mathcal{A} - \mathcal{B}$. Esto indica en particular que la probabilidad de divergencia de P en su primer paso será mayor o igual que $q_{B'}$, puesto que tenemos

$$\sum_{A\in\mathcal{A}} p_A \leq \sum_{A\in\mathcal{A}} q_A < \sum_{A\in\mathcal{A}} q_A + q_{B'} \leq \sum_{B\in\mathcal{B}} q_B \leq 1$$

Utilizando el axioma (**AI**) podemos reescribir los procesos P y Q en la forma

$$P \equiv_{\mathbf{gen}} (\bigoplus_{A\in\mathcal{A}} [\tfrac{p_A}{1-q_{B'}}] \sum_{(a,p_a)\in A} [p_a]\, a; P_{a,A}) \oplus_{1-q_{B'}} \Omega$$

$$Q \equiv_{\mathbf{gen}} (\bigoplus_{B\in\mathcal{B}-B'} [\tfrac{q_B}{1-q_{B'}}] \sum_{(b,q_b)\in B} [q_b]\, b; Q_{b,B}) \oplus_{1-q_{B'}} (\sum_{(b',q_{b'})\in B'} [q_{b'}]\, b'; Q_{b',B'})$$

Por otra parte, a partir del axioma (**D**) se tiene

$$\Omega \sqsubseteq_{\mathbf{gen}} \sum_{(b',q_{b'})\in B} [q_{b'}]\, b'; Q_{b',B'}$$

Además, por ser $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$, obtenemos

$$[\![\bigoplus_{A\in\mathcal{A}} [\frac{p_A}{1-q_{B'}}] \sum_{(a,p_a)\in A} [p_a]\, a; P_{a,A}]\!] \sqsubseteq_{\mathbf{PAT}} [\![\bigoplus_{B\in\mathcal{B}-B'} [\frac{q_B}{1-q_{B'}}] \sum_{(b,q_b)\in B} [q_b]\, b; Q_{b,B}]\!]$$

y aplicando la hipótesis de inducción, dado que los estados del proceso de la parte derecha están contenidos en los estados de Q, obtenemos

$$\bigoplus_{A\in\mathcal{A}} [\frac{p_A}{1-q_{B'}}] \sum_{(a,p_a)\in A} [p_a]\, a; P_{a,A} \sqsubseteq_{\mathbf{gen}} \bigoplus_{B\in\mathcal{B}-B'} [\frac{q_B}{1-q_{B'}}] \sum_{(b,q_b)\in B} [q_b]\, b; Q_{b,B}$$

con lo cual, aplicando la regla (**C3'**), obtenemos

$$P \sqsubseteq_{\mathbf{gen}} Q$$

- $\mathcal{A} = \mathcal{B}$ y $\exists C : p_C \neq q_C$.

  Dado que $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$, tenemos que $\forall\, A \in \mathcal{A} :\ p_A \leq q_A$, por lo que se debe cumplir $p_C < q_C$. Realizando una descomposición similar a la utilizada en el caso anterior podemos reescribir nuestros procesos en la forma

$$P \equiv_{\mathbf{gen}} (\bigoplus_{A \in \mathcal{A}-C} [\tfrac{p_A}{1-q_C}] \sum_{(a,p_a)\in A} [p_a]\ a; P_{a,A}) \oplus_{1-q_C} ((\sum_{(c,p_c)\in C} [p_c]\ c; P_{c,C}) \oplus_{\frac{p_C}{q_C}} \Omega)$$

$$Q \equiv_{\mathbf{gen}} (\bigoplus_{A \in \mathcal{A}-C} [\tfrac{q_A}{1-q_C}] \sum_{(a,p_a)\in A} [p_a]\ a; Q_{a,A}) \oplus_{1-q_C} ((\sum_{(c,p_c)\in C} [p_c]\ c; Q_{c,C}) \oplus_{\frac{p_C}{q_C}} (\sum_{(c,p_c)\in C} [p_c]\ c; Q_{c,C}))$$

  Nótese que utilizando las reglas (**OI1**), (**OI2**) y (**O1**) podemos inferir

$$\vdash \sum_{(c,p_c)\in C} [p_c]\ c; Q_{c,C}) \oplus_{\frac{p_C}{q_C}} \sum_{(c,p_c)\in C} [p_c]\ c; Q_{c,C})) \equiv_{\mathbf{gen}} \sum_{(c,p_c)\in C} [p_c]\ c; Q_{c,C}))$$

  A partir del axioma (**D**) se tiene

$$\Omega \sqsubseteq_{\mathbf{gen}} \sum_{(c,p_c)\in C} [p_c]\ c; Q_{c,C}$$

  y dado que $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$ obtenemos

$$[\![\sum_{(c,p_c)\in C} [p_c]\ c; P_{c,C}]\!] \sqsubseteq_{\mathbf{PAT}} [\![\sum_{(c,p_c)\in C} [p_c]\ c; Q_{c,C}]\!]$$

  con lo cual, aplicando la hipótesis de inducción, se deduce

$$\sum_{(c,p_c)\in C} [p_c]\ c; P_{c,C} \sqsubseteq_{\mathbf{gen}} \sum_{(c,p_c)\in C} [p_c]\ c; Q_{c,C}$$

  y aplicando la regla (**C3'**) obtenemos

$$\sum_{(c,p_c)\in C} [p_c]\ c; P_{c,C} \oplus_{\frac{p_C}{q_C}} \Omega \sqsubseteq_{\mathbf{gen}} \sum_{(c,p_c)\in C} [p_c]\ c; Q_{c,C} \oplus_{\frac{p_C}{q_C}} \sum_{(c,p_c)\in C} [p_c]\ c; Q_{c,C}$$

  De nuevo, al ser $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$ obtenemos

$$[\![\bigoplus_{A \in \mathcal{A}-C} [\frac{p_A}{1-q_C}] \sum_{(a,p_a)\in A} [p_a]\ a; P_{a,A}]\!] \sqsubseteq_{\mathbf{PAT}} [\![\bigoplus_{A \in \mathcal{A}-C} [\frac{q_A}{1-q_C}] \sum_{(a,p_a)\in A} [p_a]\ a; Q_{a,A}]\!]$$

  y aplicando la hipótesis de inducción

$$\bigoplus_{A \in \mathcal{A}-C} [\frac{p_A}{1-q_C}] \sum_{(a,p_a)\in A} [p_a]\ a; P_{a,A} \sqsubseteq_{\mathbf{gen}} \bigoplus_{A \in \mathcal{A}-C} [\frac{q_A}{1-q_C}] \sum_{(a,p_a)\in A} [p_a]\ a; Q_{a,A}$$

con lo cual, aplicando la regla (**C3'**) sobre los resultados anteriores, obtenemos

$$P \sqsubseteq_{\mathbf{gen}} Q$$

- $\mathcal{A} = \mathcal{B}$ y $\forall C \in \mathcal{A} : p_C = q_C$.
  Dado que $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$, tenemos que $\forall A \in \mathcal{A}$ y $\forall\ (a, p_a) \in A$ se verifica $[\![P_{a,A}]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q_{a,A}]\!]$, por lo que aplicando la hipótesis de inducción obtenemos

  $$P_{a,A} \sqsubseteq_{\mathbf{gen}} Q_{a,A}$$

  Entonces, por el axioma (**C1'**) obtenemos que para toda acción a y todo conjunto de acciones A se verifica

  $$a; P_{a,A} \sqsubseteq_{\mathbf{gen}} a; Q_{a,A}$$

  por lo que, aplicando reiteradamente el axioma (**C2'**), obtenemos que para todo A se tiene

  $$\sum_{(a,p_a)\in A} [p_a]\, a; P_{a,A} \sqsubseteq_{\mathbf{gen}} \sum_{(a,p_a)\in A} [p_a]\, a; Q_{a,A}$$

  y aplicando de nuevo reiteradamente el axioma (**C3'**) concluimos

  $$P = \bigoplus_{A\in\mathcal{A}} [p_A] \sum_{(a,p_a)\in A} [p_a]\, a; P_{a,A} \sqsubseteq_{\mathbf{gen}} \bigoplus_{A\in\mathcal{A}} [p_A] \sum_{(a,p_a)\in A} [p_a]\, a; Q_{a,A} = Q$$

□

Consideraremos ahora los casos en que alguno de los procesos no sea finito.

Lema 5.17 Sea $P \in PPA$ un proceso que no es finito y $Q \in$ PPA que sí lo es. Entonces, $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!] \Longrightarrow P \sqsubseteq_{\mathbf{gen}} Q$.

Demostración: Dado que las aproximaciones finitas de P forman una cadena, de modo que $[\![P]\!]$ es la cota superior mínima de los valores $[\![P^n]\!]$, se tiene

$$[\![P^0]\!] \sqsubseteq_{\mathbf{PAT}} [\![P^1]\!] \sqsubseteq_{\mathbf{PAT}} \cdots \sqsubseteq_{\mathbf{PAT}} [\![P^n]\!] \cdots \sqsubseteq_{\mathbf{PAT}} [\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$$

Puesto que tanto cada P^n como Q son finitos, podemos aplicar los resultados obtenidos anteriormente para procesos finitos, concluyendo que para todo n se verifica

$$P^n \sqsubseteq_{\mathbf{gen}} Q$$

de donde, aplicando la regla (**R2**), llegamos a

$$P \sqsubseteq_{\mathbf{gen}} Q$$

□

Consideremos ahora el caso en el que P es finito pero Q no lo es, estando definido de forma recursiva. En este caso se nos plantea la misma dificultad que ya aparecía en [Cua93]. Dado que el único medio de dar semántica a los procesos infinitos es utilizando sus aproximaciones finitas, el camino más pausible para llegar a demostrar $P \sqsubseteq_{\mathbf{gen}} Q$ consistiría en garantizar la existencia de un cierto m de manera que la aproximación finita m-ésima del proceso Q cumpla $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q^m]\!]$. Entonces, como quiera que los procesos P y Q^m son finitos, podemos aplicar el resultado del Lema 5.16 para deducir $P \sqsubseteq_{\mathbf{gen}} Q^m$. Además, tendríamos $Q^m \sqsubseteq_{\mathbf{gen}} Q$ (Lema 5.14), con lo cual, aplicando la regla (**O4**), obtenemos $P \sqsubseteq_{\mathbf{gen}} Q$.

Si el dominio semántico sobre el que trabajamos fuese ω-algebraico[1] la existencia de un m tal estaría garantizada, pero desgraciadamente tal no es la situación en nuestro caso, como ilustra el siguiente

Ejemplo 5.18 Consideremos el proceso $P = recX.((a; Nil) \oplus_{\frac{1}{2}} X)$. Es fácil ver que las aproximaciones finitas de dicho proceso vienen dadas por

$$P^n = (a; Nil) \oplus_{1-\frac{1}{2^n}} \Omega$$

Al ser $[\![P]\!] = \sqcup[\![P^n]\!]$, trivialmente tenemos $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} \sqcup[\![P^n]\!]$. Además, $[\![P]\!]$ describe a un proceso finito, pues

$$[\![recX.((a; Nil) \oplus_{\frac{1}{2}} X)]\!] =_{\mathbf{PAT}} [\![a; Nil]\!]$$

[1] Es decir, que para cualquier proceso finito no existen cadenas infinitas estrictamente crecientes de procesos tal que el proceso finito sea supremo de la cadena.

como ya se comentó al describir la semántica denotacional para el modelo generativo. Sin embargo, no existe ningún m tal que $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![P^m]\!]$, pues de existir tendríamos

$$1 = p([\![P]\!], \epsilon, \{(a,1)\}) \leq p([\![P^m]\!], \epsilon, \{(a,1)\}) = 1 - \frac{1}{2^m}$$

lo que evidentemente es imposible.

Con lo cual queda probado que el proceso (finito) $[\![a; Nil]\!]$ es cota superior mínima de la cadena infinita no trivial de procesos $\{[\![P^n]\!]\}_{n=1}^{\infty}$. □

Por lo tanto en general es preciso utilizar otro camino para probar $P \sqsubseteq_{\mathbf{gen}} Q$ a partir de $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$. Esta es la razón por la cual la regla (**R3**) aparece en el sistema axiomático, de modo que es ahora cuando quedará justificada la inclusión de la misma en nuestro sistema axiomático.

Lema 5.19 Sea $P \in$ PPA un proceso finito y $Q \in$ PPA un proceso que no lo es. Entonces, $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!] \Longrightarrow P \sqsubseteq_{\mathbf{gen}} Q$.

Demostración: Partimos de $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!] = \sqcup [\![Q^n]\!]$. De existir un $m \in \mathbb{N}$ tal que $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q^m]\!]$, la demostración se realizaría como hemos indicado anteriormente.

Consideremos entonces el caso en el que no existe un m tal. Dado que la sucesión Q^n forma una cadena, tendremos que para toda secuencia s y todo estado A se cumplirá

$$p(P, s, A) \leq p(Q, s, A) = \lim_n p(Q^n, s, A)$$

Consideremos aquellas secuencias s y aquellos estados A para los cuales se tiene $p(P, s, A) > 0$. Para tales secuencias y estados se verificará para todo $k > 0$ que

$$(1 - \frac{1}{k}) \cdot p(P, s, A) < \lim_n p(Q^n, s, A)$$

Nótese que $(1 - \frac{1}{k}) \cdot p(P, s, A) = p(P \oplus_{1-\frac{1}{k}} \Omega, s, A)$. Dado que P es un proceso finito, el conjunto de pares (s,A) que verifican $p(P, s, A) > 0$ es finito, con lo cual para cada $k \in \mathbb{N}$ existirá un cierto $n_k \in \mathbb{N}$ tal que se verifique

$$p(P \oplus_{1-\frac{1}{k}} \Omega, s, A) \leq p(Q^{n_k}, s, A)$$

para cada una de las secuencias s y estados A verificando $p(P, s, A) > 0$. Obviamente, cuando $p(P, s', A') = 0$ la desigualdad anterior se satisface trivialmente, con lo cual llegamos a

$$[\![P \oplus_{1-\frac{1}{k}} \Omega]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q^{n_k}]\!]$$

y dado que tanto $P \oplus_{1-\frac{1}{k}} \Omega$ como Q^{n_k} son finitos, podemos aplicar el resultado para procesos finitos, obteniéndose que para todo $k \in \mathbb{N}$ se tiene

$$P \oplus_{1-\frac{1}{k}} \Omega \ \sqsubseteq_{\mathbf{gen}} Q^{n_k}$$

Dado que para todo $n \in \mathbb{N}$ se tiene $Q^n \ \sqsubseteq_{\mathbf{gen}} Q$, tenemos que para todo $k \in \mathbb{N}$

$$P \oplus_{1-\frac{1}{k}} \Omega \ \sqsubseteq_{\mathbf{gen}} Q$$

con lo que aplicando la regla (**R3**) obtenemos

$$P \ \sqsubseteq_{\mathbf{gen}} Q$$

□

Teorema 5.8 Sean P, Q procesos en PPA. Entonces, $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!] \Longrightarrow P \ \sqsubseteq_{\mathbf{gen}} Q$.

Demostración: Si bien P o Q son finitos, el resultado ha quedado probado en alguno de los lemas anteriores. Consideremos entonces el caso en el que ambos procesos P y Q no son finitos, y se cumple $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$. Entonces, como consecuencia del Lema 5.19, tenemos que para toda aproximación finita P^n de P se tiene $P^n \ \sqsubseteq_{\mathbf{gen}} Q$, y en tal caso aplicando la regla (**R2**), concluimos $P \ \sqsubseteq_{\mathbf{gen}} Q$. □

Al igual que se sugiere en [Cua93], podríamos haber resuelto el problema que nos ha llevado a la inclusión de la regla (**R3**) considerando en su lugar una definición alternativa de las aproximaciones finitas. Las nuevas aproximaciones finitas incluirían explicitamente la información aportada por dicha regla. En concreto la definición formal de las nuevas aproximaciones finitas sería la siguiente:

$$P'_0 = \Omega$$
$$P'_{n+1} = P^n \oplus_{\frac{n}{n+1}} \Omega$$

donde los procesos P^n son las antiguas aproximaciones finitas definidas en la Definición 5.11.

Con esta definición alternativa de las aproximaciones finitas, la información que aportan las reglas (**R2**) y (**R3**) al sistema axiomático quedaría englobada en la propia regla (**R2**), de modo que ya no sería necesario incluir la regla (**R3**) en el nuevo sistema axiomático.

La demostración de completitud del sistema de axiomas ha sido realizada respecto de la relación de orden inducida por la semántica denotacional para el modelo generativo. Los siguientes resultados indican la completitud del sistema axiomático respecto de las relaciones de equivalencia inducidas por la semántica denotacional y la semántica de pruebas del modelo generativo.

Corolario 5.9 Sean $P, Q \in \text{PPA}$. Entonces, $[\![P]\!] =_{\text{PAT}} [\![Q]\!] \Longrightarrow \vdash P \equiv_{\text{gen}} Q$.

Demostración: Inmediato a partir del Teorema 5.8, teniendo en cuenta el hecho de que las relaciones $\sqsubseteq_{\text{PAT}}$ y $\sqsubseteq_{\text{gen}}$ son relaciones de orden. □

Corolario 5.10 (Completitud del Sistema de Axiomas)
Para cualesquiera procesos P y Q se tiene $P \approx_{\mathcal{G}} Q \Longrightarrow \vdash P \equiv_{\text{gen}} Q$.

Demostración: Inmediata a partir de los Corolarios 4.10 y 5.9. □

Corolario 5.11 Para todo par de procesos $P, Q \in \text{PPA}$ se tiene

$$P \approx_{\mathcal{G}} Q \Longleftrightarrow \vdash P \equiv_{\text{gen}} Q$$

□

Capítulo 6

Ejemplos

En el presente capítulo[1] presentaremos una serie de ejemplos que ilustran la utilidad de nuestro lenguaje para especificar sistemas concurrentes que contienen información probabilística. En los distintos ejemplos que estudiaremos vamos a intentar explotar al máximo las posibilidades que nos ofrece el lenguaje y las ventajas que conlleva el utilizar un modelo probabilístico frente a uno que no lo sea.

Los ejemplos están ordenados en orden creciente de *complejidad*, finalizando con uno de mayor entidad en el cual recopilamos todas las características del modelo PPA que ya habrán quedado ilustradas por los ejemplos anteriores.

Las aplicaciones que presentamos son las siguientes:

- Dos especificaciones para el problema de entrada de procesos en su *región crítica*: siguiendo primero un enfoque *centralizado*, y a continuación con una estructuración en *anillo lógico*.
- Un protocolo de comunicaciones en el que los mensajes se pueden perder: *Alternating Bit Protocol*.
- Dos aplicaciones relacionadas con los *Sistemas Tolerantes a Fallos*: evaluación por redundancia, y por redundancia y escrutinio.

[1] La semántica de los operadores de composición paralela que se utilizan en este capítulo está definida en el Apéndice A.

- El problema de los *filósofos hambrientos.*
- Una ampliación del problema del *barbero durmiente.*

En algunas de las aplicaciones utilizaremos una variante de los contadores introducidos en [BHR84], en la cual utilizamos como índices de los procesos conjuntos de naturales en lugar de simples naturales. Todos los conjuntos que aparecen en las especificaciones son finitos, de modo que tal variante no representa problema alguno, pues podríamos codificar los citados conjuntos por medio de naturales y utilizar entonces los contadores habituales.

Por otra parte, en lugar de utilizar el constructor $recX.P$ utilizaremos identificadores para definir los procesos recursivos que precisemos. En los casos en los que el proceso venga definido mediante un conjunto *finito* de ecuaciones mutuamente recurrentes, la traducción a un proceso *equivalente* con la notación correspondiente al operador de recursión se podría efectuar en la forma usual, ilustrada por el siguiente

Ejemplo 6.1 Consideremos los siguientes procesos mutuamente recursivos P y Q:

$$P = (a;Q) +_{\frac{1}{2}} (b;P) \qquad Q = (b;P) \oplus_{\frac{1}{3}} (c;Q)$$

Entonces, los procesos P' y Q' que siguen serían las *traducciones* de dichos procesos a la sintaxis original de nuestro modelo:

$$\begin{aligned} P' &= recX.(a;(recY.((b;X) \oplus_{\frac{1}{3}} (c;Y))) +_{\frac{1}{2}} (b;X)) \\ Q' &= recX.(b;(recY.((a;X) +_{\frac{1}{2}} (b;Y))) \oplus_{\frac{1}{3}} (c;X)) \end{aligned}$$

□

Obviamente, dicha traducción no podría llevarse a cabo en los casos en los que el proceso viniera definido mediante un conjunto infinito de ecuaciones.

En el ejemplo de los filósofos y en el del barbero durmiente, se nos presenta una situación que no está estrictamente contemplada en nuestra sintaxis, pues precisamos procesos definidos de forma recursiva, los cuales están indexados por un *parámetro* cuyo valor se toma como argumento en un operador del proceso, en concreto en una aparición de la elección interna. El problema no sería tal si tuviéramos un conjunto

finito de ecuaciones, pues podríamos aplicar el método anterior. Pero de hecho algunos de estos índices pueden ser números reales, lo que nos llevaría a tener que trabajar con un conjunto no numerable de identificadores (uno para cada real), lo cual no está permitido en nuestro modelo PPA, entre otras cosas porque ello provocaría serios problemas a la hora de calcular el menor punto fijo de una función. El problema se resuelve parcialmente asumiendo que los índices que aparezcan han de ser números racionales, lo que no representa una gran pérdida al tratarse de un conjunto denso. Esto no exigiría modificar los ejemplos, pues en los dos casos en los que que aparece este problema los citados argumentos vienen producidos por una función que toma una *semilla* para comemzar. Si tomamos como tal un número racional y la función transforma números racionales en números racionales, el problema estará resuelto (de hecho, las dos funciones que utilizamos en los ejemplos donde esto ocurre cumplen dicha propiedad). Pero por supuesto, para ajustarnos estrictamente a nuestra sintaxis, tales funciones deberían tener rango finito.

6.1 Región Crítica

En esta sección presentamos las especificaciones de dos algoritmos que aseguran la exclusión mutua entre n procesos a la hora de entrar en sus regiones críticas. Los algoritmos resuelven respectivamente el problema en el caso de que consideremos un enfoque centralizado, y cuando consideramos un enfoque distribuido en forma de anillo lógico. Estos algoritmos son clásicos en el campo de los sistemas distribuidos y han sido tomados de [PS89].

En esencia, tendremos un conjunto de procesos los cuales ejecutan una tarea. En determinados momentos el entorno indicará a cada proceso que puede pasar a ejecutar la tarea correspondiente a su región crítica (es decir, una tarea que sólo uno de los procesos puede ejecutar al mismo tiempo). Asumimos que el entorno podría dar dicha indicación a más de un proceso al tiempo, en cuyo caso el sistema habrá de decidir que proceso entra en su región crítica, garantizando siempre la exclusión mutua.

6.1.1 Enfoque centralizado

Supondremos un sistema que cuenta con n procesos ejecutándose en paralelo (la especificación del algoritmo es independiente de que el paralelismo sea real o simulado, es decir de si tenemos un procesador o varios). Estos procesos se encuentran realizando independientemente su cometido, pero en ocasiones tienen que entrar en su región crítica, para garantizar la no interferencia entre sus acciones. Como estamos suponiendo un enfoque centralizado, tendremos un proceso auxiliar central que se encargará de otorgar los permisos para entrar en su región crítica a los procesos que lo soliciten.

Especificación del sistema

El sistema viene constituido por $n+1$ procesos que interaccionan entre sí. En primer lugar tenemos los procesos $\{P_i \mid 1 \leq i \leq n\}$ que son los encargados de realizar las tareas. Por otra parte tenemos el proceso Q que se encargará de coordinar la entrada en la región crítica. Los n procesos se comunican con el proceso Q a través de las acciones $entrar, salir$ y $pedir_i$.

Tendremos los siguientes convenios:

- Las tareas que realiza cada proceso fuera de su región crítica serán simuladas mediante la ejecución de una única acción $trabajo_i$.
- Todas las acciones que cada proceso ejecuta en su región crítica vendrán recogidas por una única acción que llamaremos $región_i$.
- El centralizador no tiene preferencia por ningún proceso a la hora de permitirle el acceso a su región crítica.
- En cada paso, el entorno le ofrece a cada proceso la posibilidad de realizar una de sus acciones usuales (i.e. $trabajo_i$) o entrar en su región crítica (i.e. $entrarcrit_i$). Asumimos que el entorno no puede ofrecer a un mismo proceso las dos posibilidades al tiempo.

Pasaremos ahora a especificar las distintas componentes del sistema.

Proceso *i-ésimo*

El funcionamiento de cada uno de estos procesos vendría dado por el siguiente algoritmo:

1. Inicialmente el proceso tiene dos posibilidades:
   - El proceso realiza una acción de su trabajo para la que no necesita entrar en su región crítica. Después de realizar dicha acción vuelve al estado inicial.
   - El proceso quiere entrar en su región crítica; en tal caso pasa al estado 2.
2. El proceso solicita al centralizador permiso para entrar en su región crítica. Espera a que se le conceda dicho permiso, tras lo cual realiza las acciones de su región crítica, indica que ha terminado, y retorna al estado inicial.

En consecuencia, la especificación formal de dichos procesos es la siguiente:

$$\begin{aligned} P_i &= (trabajo_i; P_i) +_p (entrarcrit_i; Crit_i) \\ Crit_i &= pedir_i; entrar; región_i; salir; P_i \end{aligned}$$

Nótese que la probabilidad que nuestra sintaxis nos obliga a introducir en la elección externa entre las acciones $trabajo_i$ y $entrarcrit_i$ es de hecho irrelevante, al haberse asumido que el entorno sólo puede ofrecer una de las dos acciones al tiempo.

Centralizador

El funcionamiento del centralizador viene descrito por el siguiente algoritmo:

- Se tiene un bucle en el cual lo primero que se hace es recibir peticiones de entrada en la región crítica, tras lo cual se da permiso a uno de los procesos que han solicitado entrar en su región crítica, se espera a que el proceso indique que ha terminado, y se retorna al principio del bucle.

Formalmente, la especificación del Centralizador viene dada por:

$$Q = \sum_{i=1}^{n} [\frac{1}{n}] \ (pedir_i; entrar; salir; Q)$$

Si el centralizador quisiera dar mayor prioridad a alguno de los procesos para entrar en su región crítica, bastaría con modificar en consecuencia las probabilidades asociadas a la elección externa generalizada.

Sistema Centralizado

El sistema vendrá dado por la composición en paralelo de los n procesos, sincronizando en el conjunto vacío, con el proceso Q, sincronizando en las acciones $entrar, salir$ y $pedir_i$.

$$Sistema = (\underset{i=1}{\overset{n}{\|_{\emptyset}}} [\tfrac{1}{n}]P_i) \|_A^{\frac{1}{2}} Q$$

donde $A = \{pedir_i, entrar, salir \mid 1 \leq i \leq n\}$.

En la especificación dada hemos asumido que todos los procesos tienen la misma prioridad a la hora de ejecutar sus acciones. Si quisiéramos dar mayor prioridad a alguno de los procesos, bastaría con cambiar las probabilidades que aparecen en la composición paralela generalizada. También hemos supuesto que las acciones que ejecutan los procesos, dentro y fuera de su región crítica, tienen la misma prioridad que las acciones que ejecutan en paralelo con el centralizador para comunicarse con él. Si deseáramos alterar dicha simetría dando, por ejemplo, mayor prioridad a las acciones que ejecutan los procesos sería suficiente aumentar el valor del parámetro de la composición paralela. Tales situaciones se repetirán en la mayoría de los restantes ejemplos que aparecen en el capítulo, por lo que en lo sucesivo evitaremos la reiteración de la presente explicación.

6.1.2 Anillo lógico

Consideremos ahora un sistema constituido por n procesos ejecutándose en paralelo sobre un anillo lógico, de manera que cada proceso tiene un *predecesor* y un *sucesor* (de nuevo el algoritmo es independiente del hecho de que tengamos un procesador o varios, lo que en este caso significa que la conexión que se precisa entre los procesos es lógica, y

no necesariamente física). Los procesos se encuentran realizando independientemente una tarea, debiendo entrar de vez en cuando en su región crítica. Para conseguir la exclusión mutua tendremos un *token* circulando por el anillo, y sólo el proceso que disponga del *token* podrá entrar en la región crítica, si así lo desea.

Especificación del sistema

El sistema vendrá constituido por n procesos $\{P_i | 1 \leq i \leq n\}$ que interaccionan entre sí. Dichos procesos se comunican entre sí por medio de las acciones $\{token_i \mid 1 \leq i \leq n\}$.

Supondremos que se cumplen las siguientes hipótesis:

- Cada proceso ejecuta en principio la acción $trabajo_i$, la cual indica que está realizando acciones que no pertenecen a su región crítica.
- Todas las acciones que un proceso realice en su región crítica las simularemos mediante la ejecución de una única acción que llamaremos $región_i$.
- El *token* no se puede perder.
- El primer proceso del anillo posee el *token* al comenzar el funcionamiento del sistema.
- Las sumas y restas que aparecen en la especificación se hacen módulo n con corrección del dominio, $\{1, \ldots, n\}$ en lugar del habitual $\{0, \ldots, n-1\}$, de modo que $1 - 1 = n$ y $n + 1 = 1$.

Pasamos ahora a especificar las componentes del sistema.

Proceso *i-ésimo*

El funcionamiento de cada proceso P_i viene descrito por el siguiente algoritmo:

1. Inicialmente el proceso tiene dos posibilidades:
   - O bien realiza una acción de su trabajo para la cual no necesita entrar en su región crítica, tras lo cual vuelve al estado inicial.

- O bien recibe el *token* que le pasa el proceso $(i-1)$-*ésimo*, en cuyo caso pasa al estado 2.

2. De nuevo el proceso vuelve a tener dos posibilidades:

   - El proceso decide no entrar en su región crítica; en tal caso pasa el *token* al proceso $(i+1)$-*ésimo* y vuelve a su estado inicial.
   - El proceso decide entrar en su región crítica. Entonces realiza las acciones confinadas a la misma, tras lo cual pasa el *token* al proceso $(i+1)$-*ésimo* y retorna al estado inicial.

   Esta elección será realizada de forma interna por el proceso, asumiéndose que se tendrá la misma probabilidad de entrar o no en la región crítica.

En consecuencia, la especificación de los procesos P_i es la siguiente:

$$\begin{aligned} P_i &= (trabajo_i; P_i) +_{\frac{1}{2}} (token_i; Crit_i) \\ Crit_i &= (región_i; Devolver_i) \oplus_{\frac{1}{2}} Devolver_i \\ Devolver_i &= token_{i+1}; P_i \end{aligned}$$

Sistema en Anillo

La especificación del sistema vendrá dada por la composición en paralelo de los n procesos, pero dado que los conjuntos de sincronización varían dependiendo de los procesos que se traten, no podremos hacer uso del operador paralelo general, de modo que tendremos que limitarnos a utilizar reiteradamente el operador paralelo binario.

Para indicar que el proceso P_1 es quien posee el *token* inicialmente, el estado inicial en el sistema de dicho proceso será $Crit_1$ en lugar de P_1.

En definitiva, la especificación del Sistema es la siguiente:

$$\boxed{Sistema = (Crit_1 \parallel^{\frac{1}{n}}_{T_1} (P_2 \parallel^{\frac{1}{n-1}}_{T_2} (\cdots (P_{n-1} \parallel^{\frac{1}{2}}_{T_{n-1}} P_n) \cdots)))}$$

donde $T_1 = \{token_1, token_2\}$ y $T_i = \{token_{i+1}\}$ para $1 < i \leq n-1$.

6.2 Alternating Bit Protocol

En esta sección presentamos una especificación del protocolo ABP [BSW69]. El ABP es representante simplificado de una clase de protocolos usados en la transmisión de datos en sistemas concurrentes cuyo propósito es el de conseguir una transmisión segura de datos sobre medios que pueden perderlos. Este protocolo ha sido formulado anteriormente utilizando otros modelos de procesos concurrentes. Por ejemplo, en el marco de las álgebras de procesos sin probabilidades lo encontramos en [Mil89] y en [PS88], para CCS y CSP respectivamente, y añadiendo tiempo o probabilidades lo encontramos en [Chr90b, Sch89, Han91] (para procesos probabilísticos, CSP con tiempo y CCS con tiempo y probabilidades respectivamente).

Supondremos que el sistema cuenta con un solo transmisor de mensajes y con un receptor único (ver Figura 6.1). A grandes rasgos, el funcionamiento del sistema es como sigue. Se producen mensajes que son transmitidos a través de un medio no seguro desde el transmisor. Dichos mensajes son recogidos del medio por el receptor, el cual manda el mensaje a su destino, remitiendo un acuse de recibo al transmisor por cada mensaje recibido. Además, tenemos la restricción de que en el medio de transmisión sólo puede haber un mensaje en cada momento. Suponemos que el medio puede perder tanto los mensajes que el transmisor intenta mandar al receptor, como los acuses de recibido remitidos por el receptor al transmisor. El transmisor vuelve a mandar los mensajes que se pierden, o que supone que se han perdido al no recibir contestación del receptor, al igual que los acuses de recibo se vuelven a transmitir por el receptor en el caso de que se hayan perdido. Los mensajes que manda el transmisor irán etiquetados, de modo que el receptor sepa si se trata de un mensaje antiguo o nuevo. Al efecto será suficiente utilizar dos valores: 0 y 1, lo que justifica la denominación del protocolo.

6.2.1 Especificación del sistema

El sistema está constituido por tres componentes que interaccionan entre sí: el transmisor, el medio y el receptor.

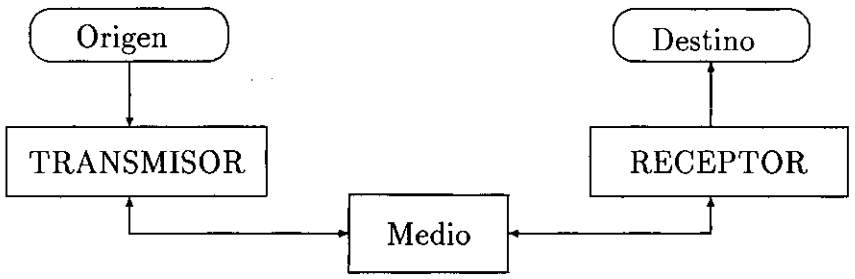


Figura 6.1: Sistema para comunicación de mensajes.

El transmisor interacciona con el entorno (origen) a través de la acción *mensaje*, y con el medio a través de las acciones mm_0 y mm_1 (mandar mensaje) y de las acciones smt_0 y smt_1 (señal de mensaje transmitido). El receptor interacciona con el entorno (destino) a través de la acción *recibido*, y con el medio a través de las acciones mr_0 y mr_1 (mensaje recibido) y de las acciones smr_0 y smr_1 (señal de mensaje recibido).

Consideraremos los siguientes convenios:

- El primer mensaje mandado por el transmisor tiene como indicativo el 0.
- Las señales de mensaje recibido que manda el receptor tienen el mismo indicativo que el último mensaje recibido.
- Cuando alguna de las componentes del sistema puede mandar o recibir varias *señales* al mismo tiempo, supondremos que no existe ninguna preferencia por ninguna de ellas.

Pasamos ahora a especificar las componentes del sistema.

Transmisor

El funcionamiento del transmisor se describe informalmente en la siguiente forma:

1. Inicialmente el transmisor recibe un mensaje desde el origen y pasa al estado 2.
2. El transmisor manda al medio un mensaje como mensaje 0 (mm_0) y pasa al estado 3.

3. Ahora el transmisor tiene tres posibilidades:

   - El transmisor no recibe señal de mensaje transmitido por el medio, en cuyo caso vuelve a mandar el mensaje 0, permaneciendo en el estado 3. Actúa así pues supone que el mensaje se ha perdido.
   - El transmisor recibe una señal de mensaje transmitido con número incorrecto, es decir smt_1, el cual es ignorado, permaneciendo por tanto en el estado 3.
   - El transmisor recibe una señal de mensaje transmitido con número correcto, es decir smt_0, pasando tras ello al estado 4.

4. El transmisor recibe un mensaje desde el origen y pasa al estado 5.

5. El transmisor manda al medio un mensaje como mensaje 1 (mm_1) y pasa al estado 6.

6. Ahora el transmisor tiene tres posibilidades:

   - El transmisor no recibe señal de mensaje transmitido por el medio, por lo que vuelve a mandar el mensaje 1, permaneciendo en el estado 6.
   - El transmisor recibe una señal de mensaje transmitido con número incorrecto, es decir smt_0, el cual es ignorado, permaneciendo por tanto en el estado 6.
   - El transmisor recibe una señal de mensaje transmitido con número correcto, es decir smt_1, tras lo cual retorna al estado inicial.

Formalmente, la especificación del transmisor es la siguiente:

$$
\begin{aligned}
T1 &= mensaje; T2 \\
T2 &= mm_0; T3 \\
T3 &= (mm_0; T3) +_{\frac{1}{3}} ((smt_0; T4) +_{\frac{1}{2}} (smt_1; T3)) \\
T4 &= mensaje; T5 \\
T5 &= mm_1; T6 \\
T6 &= (mm_1; T6) +_{\frac{1}{3}} ((smt_1; T1) +_{\frac{1}{2}} (smt_0; T6))
\end{aligned}
$$

Receptor

El algoritmo que describe el funcionamiento del receptor es el siguiente:

1. Inicialmente el receptor tiene tres posibilidades:

   - El receptor no recibe ningún mensaje. En consecuencia manda, por si acaso, una señal de mensaje recibido 1 y se queda en el estado 1. Tal acción se toma por suponerse que una señal de mensaje recibido remitida anteriormente podría haberse perdido.
   - El receptor recibe un mensaje con número incorrecto, es decir mr_1, por lo cual lo ignora, permaneciendo por tanto en el estado 1.
   - El receptor recibe una señal de mensaje recibido con número correcto, es decir mr_0. Entonces manda el mensaje al destino, mediante la acción *recibido*, tras lo cual manda una señal de mensaje recibido 0, y pasa al estado 2. Dicho estado 2 es análogo, pero correspondiente al indicativo 1.

2. De nuevo el receptor tiene tres posibilidades:

   - El receptor no recibe ningún mensaje. Entonces manda una señal de mensaje recibido 0 y permanece en el estado 2.
   - El receptor recibe un mensaje con número incorrecto, es decir mr_0, por lo cual es ignorado, permaneciendo en el estado 2.
   - El receptor recibe una señal de mensaje recibido con número correcto, es decir mr_1. Entonces manda el mensaje al destino, mediante la acción *recibido*, tras lo cual manda una señal de mensaje recibido 1, y retorna al estado inicial.

Formalmente, la especificación del receptor es la siguiente:

$$\begin{aligned} R1 &= (smr_1; R1) +_{\frac{1}{3}} ((mr_1; R1) +_{\frac{1}{2}} (mr_0; recibido; smr_0; R2)) \\ R2 &= (smr_0; R2) +_{\frac{1}{3}} ((mr_0; R2) +_{\frac{1}{2}} (mr_1; recibido; smr_1; R1)) \end{aligned}$$

Aunque el receptor pueda mandar señales de mensaje recibido sin haber recibido ningún mensaje, lo que en particular podría suceder en el estado inicial, ello no produciría ningún efecto *desagradable*, pues tales señales no serán tomadas en cuenta por el transmisor, al estar etiquetadas con un número incorrecto.

Medio

El algoritmo que describe el funcionamiento del medio es el siguiente:

1. El medio puede recibir tanto mensajes del transmisor como señales de llegada desde el receptor. Una vez que ha recibido cualquier mensaje lo envía y tras ello pueden darse dos casos:

   - El medio manda correctamente el mensaje o señal recibido, lo cual ocurrirá con una probabilidad de $\frac{4}{5}$. Tras ello, el medio está preparado para recibir un nuevo mensaje.
   - El medio pierde el mensaje o señal recibido, lo cual ocurrirá con una probabilidad de $\frac{1}{5}$. Tras ello, el medio vuelve a estar preparado para recibir más mensajes.

La especificación formal del medio es la siguiente:

$$\begin{aligned} Medio = \sum_{i=1}^{4} \; & [\tfrac{1}{4}] \;\; (mm_0; ((mr_0; Medio) \oplus_{\frac{4}{5}} Medio)) \\ & [\tfrac{1}{4}] \;\; (mm_1; ((mr_1; Medio) \oplus_{\frac{4}{5}} Medio)) \\ & [\tfrac{1}{4}] \;\; (smr_0; ((smt_0; Medio) \oplus_{\frac{4}{5}} Medio)) \\ & [\tfrac{1}{4}] \;\; (smr_1; ((smt_1; Medio) \oplus_{\frac{4}{5}} Medio)) \end{aligned}$$

Sistema ABP

Para obtener la especificación del sistema componemos en paralelo el transmisor y el receptor, sincronizando en el conjunto vacío, con el medio, sincronizando en esta ocasión en la unión de los alfabetos de las componentes, exceptuando las acciones *mensaje* y *recibido*. La especificación del sistema es:

$$\boxed{Sistema = ((T1 \parallel_{\emptyset}^{\frac{1}{2}} R1) \parallel_{A}^{\frac{2}{3}} Medio)}$$

donde $A = \{mm_j, smt_j, mr_j, smr_j \mid j = 0, 1\}$.

6.3 Sistemas Tolerantes a Fallos

En esta sección presentamos especificaciones de soluciones a dos problemas que se presentan en la implementación de sistemas tolerantes a fallos. En el primer ejemplo ilustraremos la estrategia de evaluación por *redundancia*: varias unidades se usan para realizar el mismo computo, y se devuelve un resultado en función de los resultados computados por cada unidad. En el segundo ilustramos la estrategia de evaluación por *redundancia y escrutinio*: varias unidades se usan para realizar el mismo computo y después se elige el resultado que se haya producido un mayor número de ocasiones. Estos ejemplos están inspirados en los que se tratan en [Chr90b].

Tendremos una serie de unidades que reciben una misma entrada efectuando el mismo cómputo sobre la misma. Una vez computado el resultado, se comunica al evaluador. Cuando el evaluador dispone de los resultados de todas las unidades (al menos de las que estén activas), genera una salida a partir de los resultados recibidos. Un esquema del sistema aparece en la Figura 6.2.

Las unidades pueden tener fallos, distinguiéndose dos tipos de fallos:

- *Fallos Permanentes*: Un fallo de este tipo lleva a la unidad a un estado a partir del cual no puede hacer más cálculos ni dar resultados al evaluador (es decir la unidad se rompe).
- *Fallos Pasajeros*: Un fallo de este tipo consiste en que una unidad no calcula correctamente el resultado de un cómputo, pero la unidad sigue funcionando.

En el primer ejemplo consideraremos el caso en el que sólo pueden ocurrir fallos permanentes, mientras que en el segundo supondremos que sólo pueden ocurrir fallos pasajeros.

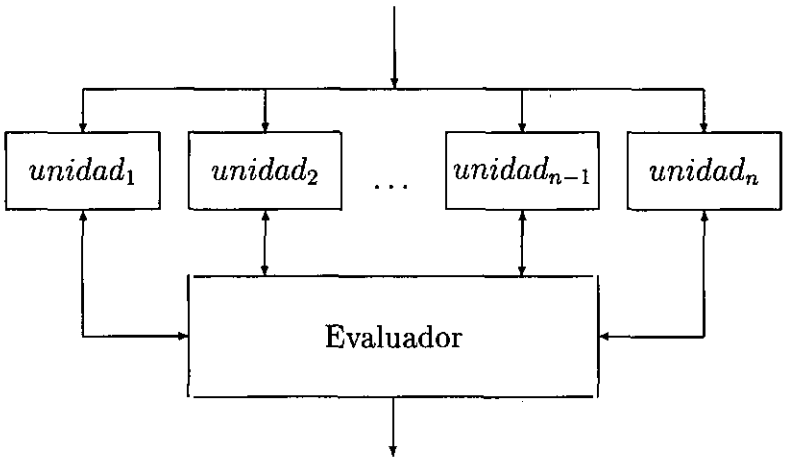


Figura 6.2: Sistema Tolerante a fallos.

6.3.1 Tolerancia a fallos por redundancia

Consideramos un sistema tolerante a fallos en el que sólo pueden ocurrir fallos permanentes. Asumiremos los siguientes hechos:

- Cuando se produce un fallo en alguna unidad, ésta lo indica al evaluador por medio de una acción antes de dejar de funcionar. Una vez que la unidad ha fallado, supondremos que puede seguir recibiendo datos de entrada, aunque al estar *rota* no producirá datos de salida.
- La probabilidad de que una unidad falle es $\frac{1}{10}$.
- Consideraremos que el evaluador manda al entorno el resultado de aplicar una determinada función a los valores recibidos.

Especificación del sistema

El sistema se modela utilizando $n + 1$ componentes que interaccionan entre sí: n unidades iguales y un evaluador. Las unidades interaccionan con el entorno a través de la acción *input*, y con el evaluador a través de las acciones c_i y f_i que indican respectivamente el funcionamiento correcto y el fallo. El evaluador interacciona con

el entorno a través de las acciones *resul* y *fallo* que se utilizan para dar un resultado al entorno o indicarle que todas las unidades han fallado. Damos a continuación la especificación de las unidades, del evaluador y del sistema que se obtiene al componer las distintas componentes en paralelo.

Unidad *i-ésima*

El funcionamiento de una unidad se describe informalmente por medio del siguiente algoritmo:

1. En el estado inicial la unidad recibe un dato de entrada del entorno. Tras ello pasa al estado 2.
2. Ahora la unidad tiene dos posibilidades:
   - O bien calcula un resultado, lo que sucederá con probabilidad $\frac{9}{10}$, y en tal caso manda el resultado al evaluador y retorna al estado 1.
   - O bien falla, lo cual ocurre con probabilidad $\frac{1}{10}$, y en tal caso manda un mensaje de fallo al evaluador y pasa al estado 3.
3. La unidad permanece inactiva en lo sucesivo; puede recibir *inputs* pero los mismos serán ignorados.

La especificación formal es la siguiente:

$$\begin{aligned} Unidad_i &= input; ((c_i; Unidad_i) \oplus_{\frac{9}{10}} (f_i; Fallo_i)) \\ Fallo_i &= input; Fallo_i \end{aligned}$$

Evaluador

El funcionamiento del evaluador queda descrito por medio del siguiente algoritmo:

1. Si todavía tiene que establecer comunicación con más unidades pasa al estado 2, sino pasa al estado 3.

2. Ahora hay dos posibilidades:

   - El evaluador recibe de la unidad i un mensaje de operación correcta. En tal caso la elimina del conjunto de unidades pendientes de respuesta, la incorpora al conjunto de unidades activas, y vuelve al estado 1.
   - El evaluador recibe un mensaje de fallo de la unidad i. En consecuencia la quita también del conjunto de unidades pendientes de respuesta, pero no la incorpora en el conjunto de unidades activas, volviendo al estado 1.

3. De nuevo hay dos posibilidades:

   - El conjunto de unidades activas es no vacío. Entonces comunica el resultado al entorno y retorna al estado 1, manteniendo el conjunto de unidades activas.
   - Todas las unidades están *rotas*. En tal caso se manda un mensaje al entorno para indicarle el fallo del sistema y tras ello para.

La especificación formal del sistema es la siguiente:

$$\begin{aligned} Eval_{\emptyset,\emptyset} &= fallo; Nil \\ Eval_{A,\emptyset} &= resul; Eval_{\emptyset,A} \end{aligned}$$

$$\begin{aligned} Eval_{\{j_1,\ldots,j_s\},\{i_1,\ldots,i_r\}} = & \left(\sum_{i \in \{i_1,\ldots,i_r\}} \left\lfloor \frac{1}{|\{i_1,\ldots,i_r\}|} \right\rfloor (c_i; Eval_{\{j_1,\ldots,j_s,i\},\{i_1,\ldots,i_r\}-\{i\}}) \right) \\ +_{\frac{1}{2}} & \left(\sum_{i \in \{i_1,\ldots,i_r\}} \left\lfloor \frac{1}{|\{i_1,\ldots,i_r\}|} \right\rfloor (f_i; Eval_{\{j_1,\ldots,j_s\},\{i_1,\ldots,i_r\}-\{i\}}) \right) \end{aligned}$$

Obsérvese que una posible modificación del parámetro probabilístico de la elección externa no influiría en el resultado final del sistema, dado que todas las unidades activas han de mandar alguna contestación al evaluador antes de que el sistema pueda evolucionar. Aunque en esta especificación utilizamos como argumento de la elección externa generalizada un conjunto, en lugar de un subrango, ello no genera ningún problema respecto del correcto uso del operador. Usualmente utilizamos un subrango en el operador generalizado, pero el uso de conjuntos como los utilizados, al ser

subconjuntos de un universo finito, se podría simular con subrangos por medio de un renombramiento adecuado de los índices, con lo que queda justificada la utilización de la variante del operador.

Sistema

Una vez más, la especificación del sistema consiste en la composición en paralelo de las unidades, sincronizando en la acción *input*, con el evaluador, sincronizando en este caso en las acciones c_i y f_i.

$$Sistema = (\ \|\|_{i=1}^{n}{}_{\{input\}}\ [\tfrac{1}{n}]Unidad_i)\ \|_B^{\frac{1}{2}}\ Evaluador_{\emptyset,\{1,..,n\}}$$

siendo $\{B = \{c_i, f_i \mid 1 \leq i \leq n\}$.

6.3.2 Tolerancia a fallos por redundancia y escrutinio

Consideremos ahora un sistema tolerante a fallos en el cual sólo pueden ocurrir fallos pasajeros. Asumimos los siguientes convenios:

- La probabilidad de que una unidad calcule erróneamente el resultado es de $\frac{1}{10}$.
- Las unidades sólo pueden recibir como entrada, y dar como resultado, 0 ó 1. También supondremos que en cada momento el entorno ofrece un único valor.

Especificación del sistema

El sistema se modela por medio de $n + 1$ componentes que interaccionan entre sí: n unidades iguales y un evaluador. Las unidades interaccionan con el entorno a través de las acciones $input_0$ y $input_1$, y con el evaluador a través de las acciones mr_i (mandar resultado), r_0 y r_1. Cada r_i supondría la remisión del resultado correcto al haberse recibido como entrada $input_i$, mientras que la remisión de r_{1-i} se produciría en el caso de mal funcionamiento de la componente. El evaluador interacciona con el entorno a través de las acciones s_0 y s_1.

Veamos la especificación de las unidades, del evaluador y del sistema resultante al componer las distintas componentes en paralelo.

Unidad *i-ésima*

El funcionamiento de cada unidad se describe informalmente en la siguiente forma:

1. En el estado inicial la unidad puede recibir dos tipos de entrada, con lo que tiene dos posibilidades:
   - Si se recibe como *input* un 0 se pasa al estado 2.
   - Si se recibe un 1 se pasa al estado 3.
2. La unidad calcula el resultado y sincroniza con el evaluador mediante la acción mr_i. Tras ello tiene dos posibilidades:
   - Si ha calculado bien el resultado, lo que ocurre con probabilidad $\frac{9}{10}$, *manda* al evaluador r_0. Tras ello vuelve al estado inicial.
   - Si ha tenido un fallo, lo que ocurre con probabilidad $\frac{1}{10}$, *manda* al evaluador r_1, y tras ello vuelve también al estado inicial.
3. La unidad calcula el resultado y sincroniza con el evaluador mediante la acción mr_i. Tras ello tiene dos posibilidades:
   - Si ha calculado bien el resultado, lo que ocurre con probabilidad $\frac{9}{10}$, *manda* al evaluador r_1, tras lo cual vuelve al estado inicial.
   - Si ha tenido un fallo, lo que ocurre con probabilidad $\frac{1}{10}$, *manda* al evaluador r_0, y retorna al estado inicial.

En consecuencia, la especificación formal es la siguiente:

$$\begin{aligned} Unidad_i &= (input_1; mr_i; Resul_{i,1}) +_{\frac{1}{2}} (input_0; mr_i; Resul_{i,0}) \\ Resul_{i,0} &= (r_0; Unidad_i) \oplus_{\frac{9}{10}} (r_1; Unidad_i) \\ Resul_{i,1} &= (r_1; Unidad_i) \oplus_{\frac{9}{10}} (r_0; Unidad_i) \end{aligned}$$

Evaluador

El funcionamiento del evaluador se describe informalmente en la forma siguiente:

1. Si no tiene que establecer comunicación con más unidades pasa al estado 6. En caso contrario pasa al estado 2.

2. El evaluador espera un mensaje de la forma mr_i. Cuando lo recibe, quita a i del conjunto de unidades pendientes de mandar el resultado y pasa al estado 3.

3. Si el evaluador recibe un mensaje con resultado 0, pasa al estado 4; si recibe un 1 pasa al estado 5.

4. Hay dos posibilidades:

   - Si el contador de los mensajes recibidos con un 1 es distinto de cero, entonces resta uno a este contador y vuelve al estado inicial.
   - Si el contador de los mensajes recibidos con un 1 es igual a cero, entonces suma uno al contador de los mensajes recibidos con un 0 y vuelve también al estado inicial.

5. Hay dos posibilidades:

   - Si el contador de los mensajes recibidos con un 0 es distinto de cero, entonces resta uno a este contador y retorna al estado inicial.
   - Si el contador de los mensajes recibidos con un 0 es igual a cero, entonces suma uno al contador de los mensajes recibidos con un 1 y vuelve al estado inicial.

6. Hay tres posibilidades:

   - Si el contador de los mensajes recibidos con un 0 es distinto de cero, manda un 0 al entorno y tras ello pasa al estado inicial.
   - Si el contador de los mensajes recibidos con un 1 es distinto de cero, manda un 1 al entorno y vuelve al estado inicial.

- Si el contador de los mensajes recibidos con un 0 y el contador de los mensajes recibidos con un 1 son iguales a cero, entonces se elige de forma equitativa entre mandar un 0 ó un 1.

La especificación formal viene dada por el siguiente sistema recursivo de procesos, en el que $r > 0$:

$$
\begin{aligned}
Eval_{\emptyset,0,0} &= (s_0; Eval_{\{1,\ldots,n\},0,0}) \oplus_{\frac{1}{2}} (s_1; Eval_{\{1,\ldots,n\},0,0}) \\
Eval_{\emptyset,r,0} &= s_0; Eval_{\{1,\ldots,n\},0,0} \\
Eval_{\emptyset,0,r} &= s_1; Eval_{\{1,\ldots,n\},0,0} \\
Eval_{\{i_1,\ldots,i_s\},0,0} &= \sum_{i\in\{i_1,\ldots,i_s\}} [\underbrace{\tfrac{1}{\{i_1,\ldots,i_s\}}}] \, (mr_i; (\quad (r_0; Eval_{\{i_1,\ldots,i_s\}-\{i\},1,0}) \\
&\qquad\qquad +_{\frac{1}{2}} \; (r_1; Eval_{\{i_1,\ldots,i_s\}-\{i\},0,1}))) \\
Eval_{\{i_1,\ldots,i_s\},r,0} &= \sum_{i\in\{i_1,\ldots,i_s\}} [\underbrace{\tfrac{1}{\{i_1,\ldots,i_s\}}}] \, (mr_i; (\quad (r_0; Eval_{\{i_1,\ldots,i_s\}-\{i\},r+1,0}) \\
&\qquad\qquad +_{\frac{1}{2}} \; (r_1; Eval_{\{i_1,\ldots,i_s\}-\{i\},r-1,0}))) \\
Eval_{\{i_1,\ldots,i_s\},0,r} &= \sum_{i\in\{i_1,\ldots,i_s\}} [\underbrace{\tfrac{1}{\{i_1,\ldots,i_s\}}}] \, (mr_i; (\quad (r_0; Eval_{\{i_1,\ldots,i_s\}-\{i\},0,r-1}) \\
&\qquad\qquad +_{\frac{1}{2}} \; (r_1; Eval_{\{i_1,\ldots,i_s\}-\{i\},0,r+1})))
\end{aligned}
$$

Sistema

Para obtener la especificación del sistema componemos en paralelo las unidades, sincronizando en las acciones $input_0$ y $input_1$, con el evaluador, sincronizando en este caso en las acciones mr_i, r_0 y r_1.

$$
\boxed{Sistema = (\, \|_{i=1}^{n}{}_A \; [\tfrac{1}{n}] \; Unidad_i) \; \|_B^{\frac{1}{2}} \; Evaluador_{\{1,\ldots,n\},0,0}}
$$

donde $A = \{input_0, input_1\}$ y $B = \{mr_i, r_0, r_1 \mid 1 \leq i \leq n\}$.

6.4 Los Filósofos Hambrientos

El problema de los filósofos hambrientos se presenta, y resuelve, por primera vez en [Dij65], y posteriormente lo podemos encontrar en [Dij71] (el artículo también se

encuentra recopilado en [HP72]). Se trata de un ejemplo clásico del problema más general que comporta la compartición de recursos por procesos en los sistemas operativos, que acarrea una serie de consecuencias desagradables como el interbloqueo, la inanición, etc. En la solución que presentamos se resuelven los problemas de interbloqueo y de exclusión mutua en la apropiación de recursos. Por su parte, el problema de la inanición es parcialmente resuelto mediante la inclusión de probabilidades a la hora de facilitar los recursos (es decir los palillos) a los filósofos.

6.4.1 Presentación del Problema

Cinco filósofos dedican su vida a realizar dos tareas: pensar y comer. Los filósofos están sentados en una mesa circular que tiene en su centro un tazón de arroz, que asumimos no se acaba nunca. Además, en la mesa hay cinco palillos, de manera que cada filósofo tiene uno a la izquierda y otro a la derecha y cinco platos.

Mientras que un filósofo está pensando, no come, pero naturalmente de vez en cuando los filósofos tienen hambre. Puede tener hambre en cualquier momento, pero la probabilidad aumenta cuanto más tiempo hace que no come. Cuando tiene hambre intenta coger los palillos que tiene a su izquierda y a su derecha, lo que podrá hacer si no los tienen los correspondientes vecinos de mesa. Cada filósofo sólo puede coger un palillo en cada instante. Cuando un filósofo tiene los dos palillos, entonces coge arroz, lo pone en su plato y come hasta que no tiene más hambre. Cuando termina de comer deja los palillos (de nuevo de uno en uno) y vuelve a pensar.

Hemos de dar la especificación de los cinco filósofos y de los cinco palillos, necesitando además otros tantos procesos auxiliares, que funcionarán como semáforos, los cuales se encargan de evitar el interbloqueo del sistema, encargándose de asegurar que si un filósofo coge un palillo, el otro que precisa no será tomado por su vecino.

6.4.2 Especificación del Sistema

El sistema está constituido por 15 componentes que interaccionan entre sí : 5 filósofos, 5 palillos y 5 semáforos. Los filósofos interaccionan con los palillos por medio de las

acciones $t_{i,i}, t_{i+1,i}, t'_{i,i}$ y $t'_{i+1,i}$, donde el índice i corresponde a su izquierda y el $i+1$ a su derecha, y las t indican coger el palillo, mientras que las t' indican soltar el palillo, y con los semáforos a través de las acciones c_i y s_i que representan el permiso para coger y la comunicación de soltar los palillos. En las especificaciones tomamos el convenio que las sumas son módulo 5 con correción a 1, es decir $4+1=5$ y $5+1=1$.

Veamos ahora las componentes del sistema:

Filósofos

El algoritmo que regula la vida de un filósofo es el siguiente:

1. Inicialmente el filósofo puede hacer dos cosas, entre las cuales se elige de forma interna:
   - El filósofo decide seguir pensando. Cuando acaba de pensar vuelve al estado 1, si bien ahora tiene más hambre que antes.
   - El filósofo decide comer. Entonces pasa al estado 2.
2. El filósofo espera hasta que pueda *bloquear* a su semáforo. Cuando bloquea el semáforo, coge el palillo de su izquierda, después el de su derecha, come, devuelve el palillo de la izquierda, después el de la derecha y retorna al estado inicial, ahora ya sin hambre.

En la especificación de cada filósofo utilizamos dos parámetros. El primero es su índice mientras que el segundo nos indica el tiempo que lleva sin comer, mediante un valor en el intervalo $[0,1)$.

En concreto, la especificación de cada filósofo queda como sigue:

$$\begin{aligned} Filo_{i,p} &= (pensar_i; Filo_{i,f(p)}) \oplus_p Comer_i \\ Comer_i &= c_i; t_{i,i}; t_{i+1,i}; arroz_i; t'_{i,i}; t'_{i+1,i}; s_i; Filo_{i,0} \end{aligned}$$

La función $f : [0,1) \longrightarrow [0,1)$ *mide* el tiempo que lleva un filósofo sin comer y ha de ser monótona creciente. Una elección razonable cumpliendo estas condiciones es la siguiente:

$$f(p) = p + \frac{1-p}{2}$$

Semáforos

El algoritmo que regula el funcionamiento de cada semáforo es el siguiente:

1. El semáforo al que acceden los filósofos i e $i+1$ tiene dos posibilidades:
   - El filósofo i pide bloquear el semáforo. Entonces se le da entrada tras lo que se espera a que el filósofo i desbloquee el semáforo. Una vez que ello sucede se vuelve al estado inicial.
   - El filósofo $i+1$ pide bloquear el semáforo. Entonces se le da entrada tras lo que se espera a que el filósofo $i+1$ desbloquee el semáforo, retornando al estado inicial.

En consecuencia, la especificación del susodicho semáforo es la siguiente:

$$S_{i,i+1} = (c_i; s_i; S_{i,i+1}) +_{\frac{1}{2}} (c_{i+1}; s_{i+1}; S_{i,i+1})$$

Para conseguir una sincronización correcta entre los semáforos tendremos que utilizar un razonamiento similar al que utilizamos en la especificación del sistema estructurado en anillo lógico (ver página 166). De modo que al componer los semáforos en paralelo obtenemos:

$$Sem = (((S_{1,2} \parallel_{C_1}^{\frac{1}{2}} S_{2,3}) \parallel_{C_2}^{\frac{2}{3}} S_{3,4}) \parallel_{C_3}^{\frac{3}{4}} S_{4,5}) \parallel_{C_4}^{\frac{4}{5}} S_{5,1}$$

donde $C_i = \{c_{i+1}, s_{i+1}\}$ si $1 \leq i \leq 3$ y $C_4 = \{c_5, c_1, s_5, s_1\}$.

Como usualmente, hemos supuesto que los semáforos no tienen predilección por ninguno de los filósofos, y que todas las acciones que ejecutan los diferentes semáforos tienen la misma prioridad.

Palillos

El funcionamiento de los palillos es similar al de los semáforos. Los filósofos que tienen acceso a ellos cuando tengan hambre intentarán cogerlos y cuando acaben de comer los devolverán.

$$T_i = (t_{i,i}; t'_{i,i}; T_i) +_{\frac{1}{2}} (t_{i,i-1}; t'_{i,i-1}; T_i)$$

Mesa

Para obtener la especificación del sistema componemos en paralelo los palillos con los filósofos, sincronizando en las acciones $t_{i,j}$ y $t'_{i,j}$, y tras ello componemos en paralelo el nuevo proceso obtenido con los semáforos, sincronizando en las acciones c_i y s_i.

$$Mesa = ((\|_{i=1}^{5}{}_{\emptyset}\ [\tfrac{1}{5}]\ Filo_{i,0})\ \|_{A}^{\frac{1}{2}}\ (\|_{i=1}^{5}{}_{\emptyset}\ [\tfrac{1}{5}]\ T_i))\ \|_{B}^{\frac{2}{3}}\ Sem$$

donde $A = \{t_{i,j}, t'_{i,j} \mid 1 \leq i \leq 5 \ \wedge\ i \leq j \leq i+1\}$ y $B = \{c_i, s_i \mid 1 \leq i \leq 5\}$.

6.5 Barbería Automática

En esta sección presentamos una especificación de una variante del problema del barbero durmiente, que encontramos por ejemplo en [BA82]. Este es un caso particular del problema más general conocido como del *productor-consumidor*, correspondiente al caso en el que hay varios consumidores y un solo productor. La solución que presentamos en esta sección es más complicada que la expuesta en [BA82], dado que vamos a dar un algoritmo que hace *más cosas* de las que se pedían en el citado trabajo.

6.5.1 Presentación del problema

Un barbero tiene una barbería en la que hay dos habitaciones: una para afeitar a los clientes y otra que se utiliza como sala de espera. Además, tiene un computador que gestiona los turnos y citas de los clientes.

Para poder ser afeitado en la barbería hay que ser *socio* (la barbería cuenta con N socios), pero los socios tienen que pedir cita previa para poder acudir a la barbería. Tales citas son gestionadas automáticamente por el computador, de modo que cuando el mismo da cita a un socio que lo solicita, éste acude a la barbería. Si hay sitio en la sala de espera, en la que hay n sitios, pasan y se sientan; en caso contrario, esperan

en la calle en cola. Además, los socios conocen el horario de la barbería, por lo tanto no piden citan antes de abrir, ni piden cita después de que haya cerrado. Respecto del barbero, tenemos que afeita como máximo a R clientes al día. Si llega antes la hora de cerrar, sólo atiende a los que ya estén en la sala de espera. Cuando llega por la mañana a la barbería, lo primero que hace es conectar el computador para que empiece a dar citas. Una representación gráfica de la barbería se puede ver en la Figura 6.3.

6.5.2 Especificación de la Barbería

La barbería está constituida por $N + 2$ componentes que interaccionan entre sí : el barbero, el computador y los N socios. Los clientes interaccionan con el barbero a través de la acción $afeitar$, con el computador a través de las acciones $cita_i, sitio_i, turno_i, cerrado_i$ y fin_i, donde i indica el número de socio, y con el entorno a través de las acciones $hora_i, nohora_i$ y $hacercosas_i$ que representan la consulta del reloj y las otras cosas que hacen a lo largo del día además de afeitarse. El barbero interacciona con el computador a través de las acciones $fin, empezar$ y $siguiente$. Por último, el computador interacciona con el entorno a través de las acciones $hora, nohora, hora', nohora'$, que se utilizan para consultar la hora.

Computador

El algoritmo que utiliza el computador es el siguiente:

1. Si R clientes ya han pedido cita, entonces pasa al estado 8; en caso contrario pasa al estado 2.

2. Se plantean dos posibilidades:

   - Si es la hora de cerrar, pasa al estado 11.
   - Si no es la hora, se plantean dos posiblidades:
     - Si alguien pide cita, entonces le dice que puede acudir a la sala de espera, le mete en la lista de espera, suma uno al número de clientes

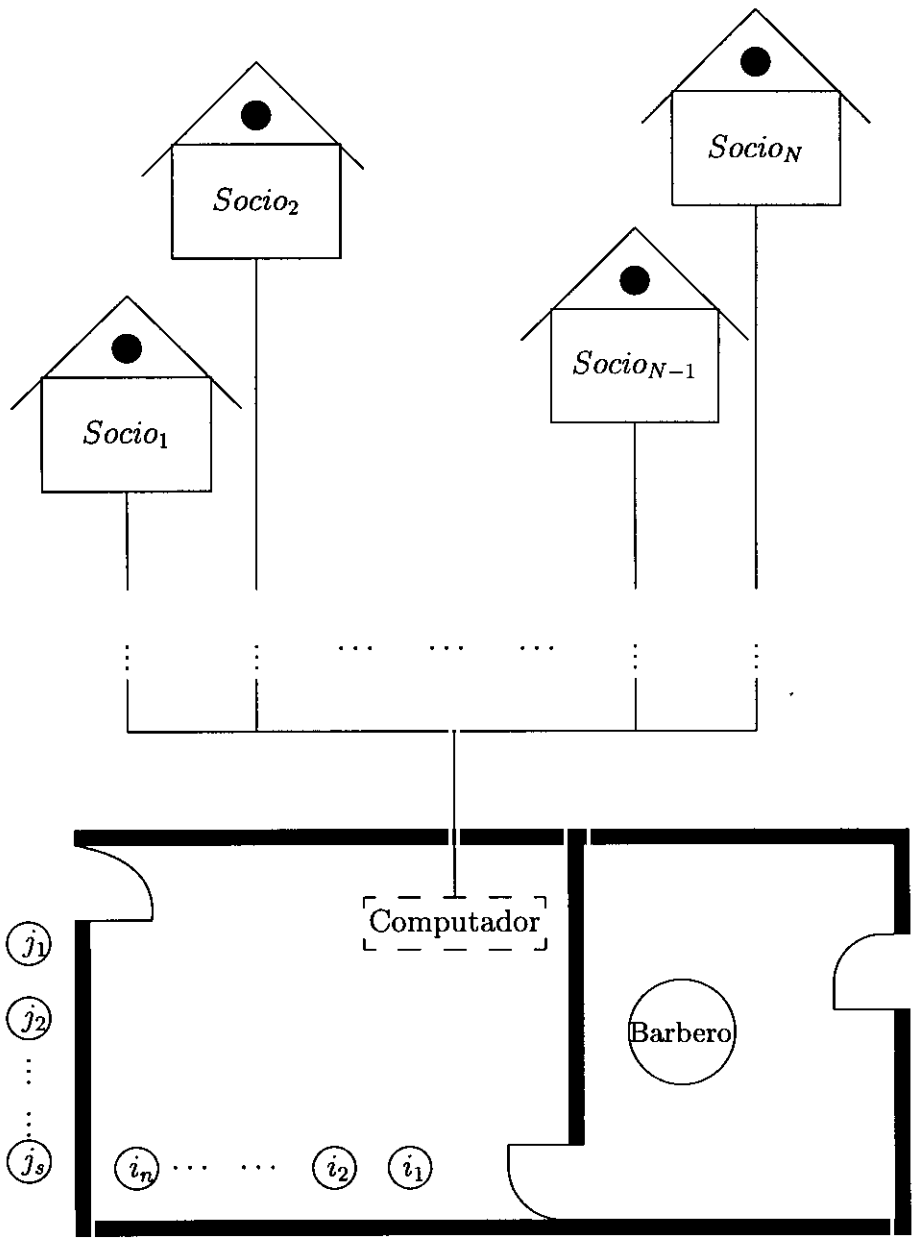


Figura 6.3: Instancia de la Barbería.

citados y pasa al estado 3.

- Si llega la hora de cerrar, pasa al estado 11.

3. Si R clientes han pedido cita, entonces pasa al estado 8; en caso contrario tiene tres posibilidades:

   - Si no tiene clientes en la sala de espera, pasa al estado 2.
   - Si tiene menos de n clientes en la sala de espera, pasa al estado 4.
   - Si tiene n clientes en la sala de espera, pasa al estado 5.

4. Se plantean dos posibilidades:

   - Si es la hora de cerrar, pasa al estado 11.
   - Si no es la hora, se plantean dos posiblidades:
     - Si alguien pide cita, entonces le dice que puede acudir a la sala de espera, le mete en la lista de espera, suma uno al número de clientes citados y pasa al estado 3.
     - Si el barbero le indica que puede pasar el siguiente cliente, entonces le indica al que tiene el primer turno en la sala de espera que puede hacerlo, resta uno al número de clientes en la sala de espera y vuelve al estado 3.

5. Se plantean dos posibilidades:

   - Si es la hora de cerrar, pasa al estado 11.
   - Si no es la hora, se plantean dos posiblidades:
     - Si alguien pide cita, entonces le pone el primero en la cola que hay en la calle, suma uno al número de clientes citados y pasa al estado 6.
     - Si el barbero le indica que puede pasar el siguiente cliente, entonces le indica al que tiene el primer turno en la sala de espera que puede hacerlo, resta uno al número de clientes en la sala de espera y vuelve al estado 3.

6. Si ya R clientes han pedido cita, entonces pasa al estado 8; en caso contrario tiene dos posibilidades:
   - Si no tiene clientes en la cola de la calle, pasa al estado 5.
   - Si tiene clientes en la cola de la calle, pasa al estado 7.
7. Se plantean dos posibilidades:
   - Si es la hora de cerrar, pasa al estado 11.
   - Si no es la hora, se plantean dos posiblidades:
     - Si alguien pide cita, entonces le pone en la cola que hay en la calle, suma uno al número de clientes citados y vuelve al estado 6.
     - Si el barbero le indica que puede pasar el siguiente cliente, entonces le indica al que tiene el primer turno en la sala de espera que puede hacerlo, y le indica al que tiene el primer turno en la cola de la calle que pase a la sala de espera, tras lo que retorna al estado 6.
8. Se plantean tres posibilidades:
   - Si no tiene a nadie en la sala de espera, pasa al estado 14.
   - Si no tiene a nadie esperando en la calle, pasa al estado 9.
   - Si tiene clientes esperando en la calle, pasa al estado 10.
9. Se plantean dos posibilidades:
   - Si es la hora de cerrar, pasa al estado 11.
   - Si no es la hora, entonces espera a que el barbero le indique que puede pasar el siguiente, y le dice al que tiene el primer turno de la sala de espera que puede pasar, pasando al estado 8.
10. Se plantean dos posibilidades:
    - Si es la hora de cerrar, pasa al estado 11.

- Si no es la hora, espera a que el barbero pida un nuevo cliente, y le dice al que tiene el primer turno de la sala de espera que puede pasar, llamando al primero de la cola de la calle para que entre a la sala de espera, tras lo que pasa al estado 8.

11. Se plantean tres posibilidades:

    - Si no tiene a nadie en la sala de espera, pasa al estado 14.
    - Si no tiene a nadie esperando en la calle, pasa al estado 13.
    - Si tiene clientes esperando en la calle, pasa al estado 12.

12. Indica a todos los clientes que están esperando en la calle que pueden volver a casa pues es la hora de cerrar y sólo se afeitará a quienes ya están en la barbería. Después pasa al estado 13.

13. Espera a que el barbero le indique que puede pasar el siguiente, entonces le dice al que tiene el primer turno de la sala de espera que puede pasar y vuelve al estado 11.

14. Le dice al barbero que ya ha terminado, pues bien o ya ha afeitado a R clientes o bien es la hora de cerrar y no queda nadie en la sala de espera, y pasa al estado 15.

15. En este estado el computador funciona como *contestador automático* (para decir a los clientes que hayan llamado pidiendo cita que está cerrado) hasta que pasa un cierto tiempo (es decir cuando se produce la acción *hora'* en el reloj externo). Entonces se *desconecta* hasta que llega el barbero por la mañana y pasa al estado 1.

El hecho de que en los estados 2 y 15 del algoritmo se pregunte dos veces por la hora es debido a que si todavía no es la hora de cerrar, o la de quitar el contestador, el computador quedará esperando a que se produzca una llamada para dar cita, o para contestar que está cerrado. Si no se produce ninguna llamada y llega la hora, el programa podrá detectarlo mediante la acción *hora* o la acción *hora'*. Esta es la

mejor forma que hemos encontrado de simular un *time-out* en el marco de un álgebra de procesos sin tiempo.

Para especificar el comportamiento del programa, el proceso llevará cuatro parámetros. Los dos primeros son dos listas ordenadas: una para los clientes que esperen en la calle y otra para los clientes de la sala de espera; mientras los otros dos son dos contadores que representan el número de socios citados y el número de clientes en la sala de espera. Veamos a continuación la especificación formal del programa, en la que $0 \leq j < R$ y $1 \leq k < n$.

$$Programa = Programa_{\emptyset,\emptyset,0,0}$$

$$\begin{aligned}
&Programa_{\emptyset,\emptyset,j,0} = \\
&\qquad (hora; Cerrar_{\emptyset,\emptyset}) \\
&+_{\frac{1}{2}} \ (nohora; (\quad (\sum_{i=1}^{N} [\tfrac{1}{N}] \ (cita_i; sitio_i; Programa_{\emptyset,\{i\},j+1,1})) \\
&\qquad\qquad +_{\frac{1}{2}} \ (hora; Cerrar_{\emptyset,\emptyset})))
\end{aligned}$$

$$\begin{aligned}
&Programa_{\emptyset,\{i_1,\ldots,i_k\},j,k} = \\
&\qquad (hora; Cerrar_{\emptyset,\{i_1,\ldots,i_k\}}) \\
&+_{\frac{1}{2}} \ (nohora \rightarrow (\quad (\sum_{i=1}^{N} [\tfrac{1}{N}] \ (cita_i; sitio_i; Programa_{\emptyset,\{i_1,\ldots,i_k,i\},j+1,k+1})) \\
&\qquad\qquad +_{\frac{1}{2}} \ (siguiente; turno_{i_1}; Programa_{\emptyset,\{i_2,\ldots,i_k\},j,k-1})))
\end{aligned}$$

$$\begin{aligned}
&Programa_{\emptyset,\{i_1,\ldots,i_n\},j,n} = \\
&\qquad (hora; Cerrar_{\emptyset,\{i_1,\ldots,i_n\}}) \\
&+_{\frac{1}{2}} \ (nohora; (\quad (\sum_{i=1}^{N} [\tfrac{1}{N}] \ (cita_i; Programa_{\{i\},\{i_1,\ldots,i_n\},j+1,n})) \\
&\qquad\qquad +_{\frac{1}{2}} \ (siguiente; turno_{i_1}; Programa_{\emptyset,\{i_2,\ldots,i_n\},j,n-1})))
\end{aligned}$$

$$\begin{aligned}
&Programa_{\{j_1,\ldots,j_s\},\{i_1,\ldots,i_n\},j,n} = \\
&\qquad (hora; Cerrar_{\{j_1,\ldots,j_s\},\{i_1,\ldots,i_n\}}) \\
&+_{\frac{1}{2}} \ (nohora; (\quad (\sum_{i=1}^{N} [\tfrac{1}{N}] \ (cita_i; Programa_{\{j_1,\ldots,j_s,i\},\{i_1,\ldots,i_n,\},j+1,n})) \\
&\qquad\qquad +_{\frac{1}{2}} \ (siguiente; turno_{i_1}; sitio_{j_1}; Programa_{\{j_2,\ldots,j_s\},\{i_2,\ldots,i_n,j_1\},j,n})))
\end{aligned}$$

$$Programa_{\emptyset,\emptyset,R,0} = Acabar$$

$$Programa_{\emptyset,\{i_1,\ldots,i_k\},R,k} = (hora; Cerrar_{\emptyset,\{i_1,\ldots,i_k\}}) +_{\frac{1}{2}} (nohora; siguiente; turno_{i_1}; Programa_{\emptyset,\{i_2,\ldots,i_k\},R,k-1})$$

$$Programa_{\emptyset,\{i_1,\ldots,i_n\},R,n} = (hora; Cerrar_{\emptyset,\{i_1,\ldots,i_n\}}) +_{\frac{1}{2}} (nohora; siguiente; turno_{i_1}; Programa_{\emptyset,\{i_2,\ldots,i_n\},R,n-1})$$

$$Programa_{\{j_1,\ldots,j_s\},\{i_1,\ldots,i_n\},R,n} = (hora; Cerrar_{\{j_1,\ldots,j_s\},\{i_1,\ldots,i_n\}}) +_{\frac{1}{2}} (nohora; siguiente; turno_{i_1}; sitio_{j_1}; Programa_{\{j_2,\ldots,j_s\},\{i_2,\ldots,i_n,j_1\},R,n})$$

$$Cerrar_{\emptyset,\emptyset} = Acabar$$

$$Cerrar_{\emptyset,\{i_1,\ldots,i_k\}} = siguiente; turno_{i_1}; Cerrar_{\emptyset,\{i_2,\ldots,i_k\}}$$

$$Cerrar_{\emptyset,\{i_1,\ldots,i_n\}} = siguiente; turno_{i_1}; Cerrar_{\emptyset,\{i_2,\ldots,i_n\}}$$

$$Cerrar_{\{j_1,\ldots,j_s\},\{i_1,\ldots,i_n\}} = fin_{j_1}; Cerrar_{\{j_2,\ldots,j_s\},\{i_1,\ldots,i_n\}}$$

$$Acabar = fin; Contestador$$

$$Contestador = (hora'; empezar; Programa_{\emptyset,\emptyset,0,0}) +_{\frac{1}{2}} (nohora'; ((\sum_{i=1}^{N} [\tfrac{1}{N}] (cerrado_i; Contestador)) +_{\frac{1}{2}} (hora'; empezar; Programa_{\emptyset,\emptyset,0,0})))$$

Barbero

El algoritmo que sigue el Barbero es el siguiente:

1. El barbero tiene dos posibilidades:

   - Si recibe la señal del computador de fin de la jornada, bien porque sea la hora de cerrar y no quede nadie en la sala de espera, o porque ya haya afeitado a R clientes, pasa al estado 2.

- En caso contrario le dice al computador que llame al siguiente cliente para afeitar y pasa al estado 3.

2. El Barbero se va a descansar. Cuando vuelve al día siguiente enciende el computador encontrándose en el estado inicial.

3. El Barbero afeita al cliente de turno y retorna al estado inicial.

En consecuencia la especificación formal del Barbero es la siguiente:

$$Barbero = (fin; descansar; empezar; Barbero) +_{\frac{1}{2}} (siguiente; afeitar; Barbero)$$

Cliente

El algoritmo que regula el comportamiento de cada socio es el siguiente:

1. El cliente se levanta por la mañana y dependiendo de la barba que tenga decide ir a la barbería, en cuyo caso pasa al estado 2, o no ir, pasando al estado 5.

2. Ahora hay dos posibilidades:

   - Si no es hora de pedir cita, porque es demasiado tarde, pasa al estado 5.
   - Si es hora de pedir cita pasa al estado 3.

3. De nuevo hay dos posibilidades:

   - El computador le indica que está cerrado, de modo que la solicitud del cliente no ha sido atendida. Entonces pasa al estado 5.
   - El computador le da cita y pasa al estado 4.

4. Otra vez al cliente le pueden ocurrir dos cosas:

   - El computador le da sitio en la barbería, entonces espera a que le llame para afeitarse, se afeita y pasa al estado 5.
   - El computador le dice que es la hora de cerrar, lo cual significa que como está todavía en la calle tiene que irse, pasando al estado 5.

5. El cliente pasa el resto del día haciendo sus cosas y al día siguiente la barba le ha crecido de acuerdo con una función que toma como parámetros la barba del día anterior (que si se afeitó será 0) y un parámetro de crecimiento de barba.

Para especificar el comportamiento de cada cliente se considerarán tres parámetros: un número de socio, un parámetro de *barba actual* y un parámetro $p \in (0,1)$ de *crecimiento de barba*. La especificación formal es la siguiente, en la que se tiene $1 \leq i \leq N$.

$$Cliente_{i,r,p} = ((nohora_i; restodia_{i,r,p}) +_{\frac{1}{2}} (hora_i; irbarbero_{i,r,p})) \oplus_r restodia_{i,r,p}$$

$$restodia_{i,r,p} = hacercosas_i; Cliente_{i,f(r,p),p}$$

$$\begin{aligned} irbarbero_{i,r,p} = \; & (cerrado_i; restodia_{i,r,p}) \\ & +_{\frac{1}{2}} (cita_i; ((sitio_i; turno_i; afeitar; restodia_{i,0,p}) +_{\frac{1}{2}} (fin_i; restodia_{i,r,p}))) \end{aligned}$$

La función de crecimiento de barba, que tiene como parámetros la *barba actual* y el *parámetro de crecimiento de barba*, deberá ser monótona no decreciente y tener su imagen en el intervalo $[0,1]$. Una posible función que cumple dichas condiciones es

$$f(r,p) = \begin{cases} p & \text{si } r = 0 \\ r + \frac{1-r}{2} & \text{si } r \neq 0 \end{cases}$$

Barbería

Para obtener la especificación completa de la barbería tenemos que componer en paralelo a todos los clientes, sincronizando en el conjunto vacío, con el programa del computador y con el barbero.

$$Barbería = ((\|_{i=1}^{N}{}_{\emptyset} \; [\tfrac{1}{N}] \; Cliente_{i,r_i,p_i}) \|_A^{\frac{1}{2}} (Programa \|_B^{\frac{1}{2}} Barbero))$$

donde tenemos $A = \{cerrado_i, cita_i, sitio_i, turno_i, fin_i, afeitar \mid 1 \leq i \leq N\}$ mientras que $B = \{fin, empezar, siguiente\}$.

Capítulo 7

Conclusiones

A lo largo de este trabajo hemos estudiado diferentes semánticas de pruebas para un álgebra de procesos probabilística. Empezamos considerando una interpretación *reactiva* de las probabilidades. Para dicho primer modelo hemos definido una semántica de pruebas y una caracterización alternativa. Además, dimos una semántica denotacional que es completamente abstracta para un subconjunto de nuestro lenguaje.

Aunque resulta muy fácil razonar con el modelo reactivo debido a la sencillez de las pruebas consideradas, encontramos diversos problemas, tanto semánticos como intuitivos, en este modelo. A continuación hemos desarrollado una interpretación del modelo generativo en nuestro lenguaje. Además de la correspondiente semántica de pruebas hemos definido una caracterización alternativa basada en conjuntos de aceptación y una semántica denotacional completamente abstracta basada en árboles de aceptación. También hemos dado un conjunto de axiomas y reglas correcto y completo con respecto a la semántica de pruebas.

Para demostrar la utilidad de nuestro lenguaje presentamos la especificación de una serie de ejemplos que tratan protocolos de comunicaciones, sistemas tolerantes a fallos, y problemas clásicos de compartición de recursos.

Finalmente, en el Apéndice A extenderemos nuestro lenguaje con un operador paralelo y discutiremos los problemas que acarrearía la inclusión de un operador de restricción.

Aunque el tratamiento de nuestro lenguaje ha sido bastante completo, quedan sin embargo muchas líneas abiertas para extensiones futuras de los modelos desarrollados.

En primer lugar sería muy interesante el estudio de un lenguaje con prioridades en el cual el operador de restricción *encajara* perfectamente. El estudio de una extensión tal se nos antoja excesivamente complejo a la vista de los resultados obtenidos en esta Tesis y de los obtenidos en un trabajo previo sobre prioridades [NdF95a].

Otra posible línea sería estudiar un modelo probabilístico–temporal. Aunque ya existen propuestas que incluyen tanto información temporal como probabilística, e.g. [Han91, Low93], estos modelos no están basado en una semántica de pruebas, e incluso el marco semántico del primero está muy alejado del nuestro, mientras que el tratamiento de los operadores de elección externa y paralelo en el segundo de ellos no es del todo satisfactorio. En la línea que proponemos existe una primera propuesta recogida en [Gre95], aunque este trabajo se encuentra todavía en una fase preliminar precisando de reflexiones más profundas.

En estos momentos estamos trabajando en la extensión probabilística de la noción de pruebas definida en [Phi87]. Los primeros resultados obtenidos en esta línea nos han parecido bastante satisfactorios, y entendemos que pueden dar lugar a un trabajo más profundo.

Finalmente, estamos estudiando la posibilidad de la inclusión de probabilidades a la hora de resolver el no-determinismo que se produce en ocasiones al trabajar con lenguajes funcionales concurrentes. En tales lenguajes se producen en efecto situaciones en las que resulta imprescindible un cierto no-determinismo, como por ejemplo a la hora de determinar que procesos deben comunicar por medio de uno de sus canales. La inclusión de información probabilística se podría utilizar para dar mayor *prioridad* a un proceso que a otro a la hora de comunicarse con un tercero. Aunque esta línea no esté tan relacionada con este trabajo como lo están las líneas de trabajo anteriores, consideramos que puede resultar interesante investigar en ella, con lo que estableceríamos conexiones entre las dos líneas de investigación a las que el autor del trabajo viene prestando su atención.

Apéndice A

Nuevos Operadores

En este apéndice discutimos la posible inclusión de nuevos operadores en nuestro lenguaje. Si miramos las álgebras de procesos clásicas, echamos en falta dos operadores en nuestro lenguaje: un operador de *composición paralela* y un operador de *ocultamiento* o *restricción*. La utilidad de tales operadores es evidente en un lenguaje utilizado para la especificación de procesos concurrentes. Mientras que el operador paralelo nos permite componer procesos de forma que puedan ejecutarse concurrentemente, un operador de ocultamiento sirve para *abstraer* las partes del sistema que queremos queden ocultas para un observador externo.

Mientras que no existe ningún problema en la inclusión de un operador paralelo, que de hecho se puede definir como operador derivado a partir del resto de operadores del lenguaje, veremos que el hecho de introducir un operador de ocultamiento nos llevaría a tener que ampliar nuestro modelo para tratar no sólo con probabilidades sino también con información sobre prioridades.

En la primera sección de este apéndice presentamos tres posibles variantes de operadores paralelos para nuestro lenguaje. El primero no tendrá ninguna probabilidad asociada, y será similar al que utilizamos para componer los procesos y las pruebas en el Capítulo 2. El segundo tendrá una probabilidad asociada que servirá para dar mayor o menor peso a cada una de los dos componentes de la composición paralela a la hora de ejecutar acciones que no pertenezcan al conjunto de sincronización. Este

operador es similar al considerado en [Cua93]. El último de los operadores tiene dos probabilidades asociadas: una para asignar un peso a cada una de las componentes que se utilizará para regular la ejecución de acciones que no pertenecen al conjunto de sincronización, y otra para indicar el peso que se da a las acciones que no pertenecen al conjunto de sincronización con respecto a las de dicho conjunto. Este operador está inspirado en los operadores de composición paralela descritos en en [BBS92], si bien en nuestro caso colapsamos en un único operador sus operadores $|_{r,s}$ y $\|_{r,s}$.

Definiremos la semántica operacional de cada uno de estos operadores, y dado que el tratamiento semántico es bastante similar para todos ellos, nos centraremos en el operador con un parámetro para realizar un estudio más profundo de la semántica de este operador.

A.1 Tres Propuestas de Operador Paralelo

Como ya hemos comentado en la introducción, vamos a presentar tres variantes probabilísticas de un operador paralelo. Estos operadores paralelos estarán basados en el operador paralelo de CSP [Hoa85], de modo que la sincronización entre los dos procesos se producirá a través de un conjunto de acciones que aparecerá como parámetro del operador paralelo. El tratamiento de un operador paralelo basado en el paradigma utilizado en CCS [Mil80, Mil89], al margen de la consideración de las acciones internas, sería bastante similar.

Como en el caso de la composición entre procesos y pruebas, a la hora de definir la semántica operacional de cada uno de estos operadores precisaremos dos grupos de reglas: uno que trata las transiciones internas de ambos procesos, y otro para las acciones observables. Las transiciones internas se tratarán de forma idéntica en los tres casos, mientras que el tratamiento de las transiciones asociadas a las acciones observables variará teniendo en cuenta los parámetros de cada uno de los operadores.

A la hora de definir la probabilidad asociada a las transiciones correspondientes a acciones observables podemos seguir dos caminos: usar factores de prenormalización o no hacerlo. Como quiera que la utilización de factores de prenormalización fue ya

ampliamente comentada en la Sección 2.3, en el resto de este apéndice omitiremos el tratamiento de estos factores definiendo los valores asociados a las transiciones observables de la composición paralela utilizando un único factor de normalización.

A.1.1 Operador Paralelo sin probabilidades

En este primer caso, el operador paralelo no tendrá probabilidades asociadas, teniendo como parámetro único un conjunto de sincronización.

$$(PAR1)\ \frac{P \succ\!\!\rightarrow_p P' \wedge Q_\oplus = 0}{P \parallel_A Q \succ\!\!\rightarrow_p P' \parallel_A Q} \qquad (PAR2)\ \frac{Q \succ\!\!\rightarrow_p Q' \wedge P_\oplus = 0}{P \parallel_A Q \succ\!\!\rightarrow_p P \parallel_A Q'}$$

$$(PAR3)\ \frac{P \succ\!\!\rightarrow_p P' \wedge Q \succ\!\!\rightarrow_q Q'}{P \parallel_A Q \succ\!\!\rightarrow_{p \cdot q} P' \parallel_A Q'}$$

El significado de estas reglas es similar al ya explicado al considerar las correspondientes a la composición entre procesos y pruebas. A continuación presentamos las reglas que generan las transiciones observables que la composición paralela puede realizar.

$$(PAR4)\ \frac{P \xrightarrow{b}_p P' \wedge Q_\oplus = 0 \wedge b \notin A}{P \parallel_A Q \xrightarrow{b}_{r_1} P' \parallel_A Q} \qquad (PAR5)\ \frac{Q \xrightarrow{b}_p Q' \wedge P_\oplus = 0 \wedge b \notin A}{P \parallel_A Q \xrightarrow{b}_{r_1} P \parallel_A Q'}$$

$$(PAR6)\ \frac{P \xrightarrow{a}_p P' \wedge Q \xrightarrow{a}_q Q' \wedge a \in A}{P \parallel_A Q \xrightarrow{a}_{r_2} P' \parallel_A Q'}$$

Las probabilidades r_1 y r_2 que aparecen en las mismas vienen dadas por

$$r_1 = \frac{p}{\mu(P, Q, A)} \qquad r_2 = \frac{p \cdot q}{\mu(P, Q, A)}$$

donde $\mu(P, Q, A)$ es un *factor de normalización*, en el cual se suman las probabilidades de las acciones que pertenecen al conjunto de sincronización que ambos procesos pueden ejecutar, con las de las acciones que no pertenecen a dicho conjunto y que alguno de los dos procesos puede ejecutar. En el primer caso, dichas probabilidades se obtienen multiplicando las probabilidades con las cuales ambos procesos pueden

$$\frac{P \rightarrowtail_p P' \wedge Q_\oplus = 0}{P \parallel_A Q \rightarrowtail_p P' \parallel_A Q} \qquad \frac{Q \rightarrowtail_p Q' \wedge P_\oplus = 0}{P \parallel_A Q \rightarrowtail_p P \parallel_A Q'} \qquad \frac{P \rightarrowtail_p P' \wedge Q \rightarrowtail_q Q'}{P \parallel_A Q \rightarrowtail_{p \cdot q} P' \parallel_A Q'}$$

$$\frac{P \xrightarrow{b}_p P' \wedge Q_\oplus = 0 \wedge b \notin A}{P \parallel_A Q \xrightarrow{b}_{r_1} P' \parallel_A Q} \qquad \frac{Q \xrightarrow{b}_p Q' \wedge P_\oplus = 0 \wedge b \notin A}{P \parallel_A Q \xrightarrow{b}_{r_1} P \parallel_A Q'}$$

$$\frac{P \xrightarrow{a}_p P' \wedge Q \xrightarrow{a}_q Q' \wedge a \in A}{P \parallel_A Q \xrightarrow{a}_{r_2} P' \parallel_A Q'}$$

donde $r_1 = \frac{p}{\mu(P,Q,A)}$ y $r_2 = \frac{p \cdot q}{\mu(P,Q,A)}$.

Figura A.1: Reglas para el operador $\parallel_A$.

ejecutar dichas acciones. En definitiva

$$\begin{aligned}\mu(P,Q,A) &= \textstyle\sum_{a \in A} \{\!\{ p \cdot q \mid \exists P', Q' :\ P \xrightarrow{a}_p P' \wedge Q \xrightarrow{a}_q Q' \}\!\} \\ &\quad + \textstyle\sum_{a \notin A} \{\!\{ p \mid \exists P' :\ P \xrightarrow{a}_p P' \}\!\} + \sum_{a \notin A} \{\!\{ p \mid \exists Q' :\ Q \xrightarrow{a}_p Q' \}\!\}\end{aligned}$$

En la Figura A.1 quedan recopiladas todas las reglas de este operador.

A.1.2 Operador Paralelo con una probabilidad

En este caso, nuestro operador paralelo tendrá una probabilidad asociada, y un conjunto de sincronización. Este operador será similar al descrito en [Cua93], y es el que hemos utilizado en los ejemplos del Capítulo 6.

Como ya indicamos, las reglas correspondientes a las transiciones internas son las mismas que en el caso anterior. A continuación presentamos las reglas que generan las transiciones observables que la composición paralela puede realizar.

$$(PAR4') \ \frac{P \xrightarrow{b}_p P' \wedge Q_\oplus = 0 \wedge b \notin A}{P \parallel_A^{p_1} Q \xrightarrow{b}_{p_1 \cdot r_1} P' \parallel_A^{p_1} Q} \qquad (PAR5') \ \frac{Q \xrightarrow{b}_p Q' \wedge P_\oplus = 0 \wedge b \notin A}{P \parallel_A^{p_1} Q \xrightarrow{b}_{(1-p_1) \cdot r_1} P \parallel_A^{p_1} Q'}$$

$$(PAR6') \ \frac{P \xrightarrow{a}_p P' \wedge Q \xrightarrow{a}_q Q' \wedge a \in A}{P \parallel_A^{p_1} Q \xrightarrow{a}_{r_2} P' \parallel_A^{p_1} Q'}$$

Las probabilidades r_1 y r_2 que aparecen en ellos se definen ahora por medio de

$$r_1 = \frac{p}{\mu(P,Q,A,p_1)} \qquad r_2 = \frac{p \cdot q}{\mu(P,Q,A,p_1)}$$

donde $\mu(P,Q,A,p_1)$ es un nuevo *factor de normalización*, en el que se tienen en cuenta las probabilidades de las acciones que pertenecen al conjunto de sincronización que ambos procesos pueden ejecutar, y las de las acciones que no pertenecen a dicho conjunto que alguno de los dos procesos puede ejecutar. Respecto de las primeras seguimos considerando los productos de las probabilidades con las que ambos procesos pueden ejecutar dichas acciones. En cambio, para las segundas ponderaremos con p_1 (resp. $1-p_1$) a las probabilidades asociadas a las transiciones observables cuyas acciones no pertenecen al conjunto de sincronización. Por lo tanto,

$$\begin{aligned}\mu(P,Q,A,p_1) &= \textstyle\sum_{a\in A} \{\!|\, p\cdot q \mid \exists P', Q' :\ P \xrightarrow{a}_p P' \wedge Q \xrightarrow{a}_q Q' \,|\!\} \\ &+ p_1 \cdot \textstyle\sum_{a\notin A} \{\!|\, p \mid \exists P' :\ P \xrightarrow{a}_p P' \,|\!\} \\ &+ (1-p_1) \cdot \textstyle\sum_{a\notin A} \{\!|\, p \mid \exists Q' :\ Q \xrightarrow{a}_p Q' \,|\!\}\end{aligned}$$

En el Capítulo 6 hemos utilizado una versión generalizada con un número de argumentos arbitrario del operador paralelo binario. Este operador generalizado viene dado por la siguiente

Definición A.1 Sean $P_1, P_2, \ldots, P_n$ procesos de PPA, y sean $p_1, p_2, \ldots, p_n > 0$ tales que $\sum p_i = 1$. Definimos inductivamente la *composición paralela generalizada* en la forma:

1. $\|_{A}^{1}\ [1]\ P = P$
2. $\|_{A}^{n}\ [p_n]\ P_i = P_1 \ \|_{A}^{p_1}\ \|_{A}^{n-1}\ [\frac{p_{i+1}}{1-p_1}]\ P_{i+1}$

□

En la Figura A.2 presentamos agrupadas todas las reglas del operador $\|_{A}^{p_1}$.

$$\frac{P \succ\!\!\rightarrow_p P' \wedge Q_\oplus = 0}{P \parallel_A^{p_1} Q \succ\!\!\rightarrow_p P' \parallel_A^{p_1} Q} \qquad \frac{Q \succ\!\!\rightarrow_p Q' \wedge P_\oplus = 0}{P \parallel_A^{p_1} Q \succ\!\!\rightarrow_p P \parallel_A^{p_1} Q'} \qquad \frac{P \succ\!\!\rightarrow_p P' \wedge Q \succ\!\!\rightarrow_q Q'}{P \parallel_A^{p_1} Q \succ\!\!\rightarrow_{p\cdot q} P' \parallel_A^{p_1} Q'}$$

$$\frac{P \xrightarrow{b}_p P' \wedge Q_\oplus = 0 \wedge b \notin A}{P \parallel_A^{p_1} Q \xrightarrow{b}_{p_1 \cdot r_1} P' \parallel_A^{p_1} Q} \qquad \frac{Q \xrightarrow{b}_p Q' \wedge P_\oplus = 0 \wedge b \notin A}{P \parallel_A^{p_1} Q \xrightarrow{b}_{(1-p_1)\cdot r_1} P \parallel_A^{p_1} Q'}$$

$$\frac{P \xrightarrow{a}_p P' \wedge Q \xrightarrow{a}_q Q' \wedge a \in A}{P \parallel_A^{p_1} Q \xrightarrow{a}_{r_2} P' \parallel_A^{p_1} Q'}$$

donde $r_1 = \frac{p}{\mu(P,Q,A,p_1)}$ y $r_2 = \frac{p\cdot q}{\mu(P,Q,A,p_1)}$.

Figura A.2: Reglas para el operador $\parallel_A^{p_1}$.

A.1.3 Operador Paralelo con dos probabilidades

En esta ocasión, el operador paralelo tendrá dos probabilidades asociadas, y un conjunto de sincronización.

Las reglas para las transiciones internas son idénticas a las de los casos anteriores. A continuación presentamos las reglas correspondientes a las transiciones observables que la composición paralela puede realizar.

$$(PAR4'') \frac{P \xrightarrow{b}_p P' \wedge Q_\oplus = 0 \wedge b \notin A}{P \parallel_A^{p_1,p_2} Q \xrightarrow{b}_{p_1\cdot p_2\cdot r_1} P' \parallel_A^{p_1,p_2} Q} \qquad (PAR5'') \frac{Q \xrightarrow{b}_p Q' \wedge P_\oplus = 0 \wedge b \notin A}{P \parallel_A^{p_1,p_2} Q \xrightarrow{b}_{(1-p_1)\cdot p_2\cdot r_1} P \parallel_A^{p_1,p_2} Q'}$$

$$(PAR6'') \frac{P \xrightarrow{a}_p P' \wedge Q \xrightarrow{a}_q Q' \wedge a \in A}{P \parallel_A^{p_1,p_2} Q \xrightarrow{a}_{(1-p_2)r_2} P' \parallel_A^{p_1,p_2} Q'}$$

En dichas reglas, la probabilidad de que ocurra una acción de sincronización está ponderada por $(1 - p_2)$, mientras que la probabilidad de que ocurra una acción que no pertenezca al conjunto de sincronización está ponderada por p_2. Además, y como en el caso anterior, la probabilidad de que la primera componente ejecute acciones que no pertenezcan al conjunto de sincronización estará además ponderada por p_1, mientras que la de la segunda componente incorpora el factor $1 - p_1$. Al igual que en los casos anteriores, r_1 y r_2 vienen dados por las expresiones

$$r_1 = \frac{p}{\mu(P,Q,A,p_1,p_2)} \qquad r_2 = \frac{p\cdot q}{\mu(P,Q,A,p_1,p_2)}$$

$$\frac{P \succ\!\!\rightarrow_p P' \wedge Q_\oplus = 0}{P \|_A^{p_1,p_2} Q \succ\!\!\rightarrow_p P' \|_A^{p_1,p_2} Q} \qquad \frac{Q \succ\!\!\rightarrow_p Q' \wedge P_\oplus = 0}{P \|_A^{p_1,p_2} Q \succ\!\!\rightarrow_p P \|_A^{p_1,p_2} Q'} \qquad \frac{P \succ\!\!\rightarrow_p P' \wedge Q \succ\!\!\rightarrow_q Q'}{P \|_A^{p_1,p_2} Q \succ\!\!\rightarrow_{p\cdot q} P' \|_A^{p_1,p_2} Q'}$$

$$\frac{P \xrightarrow{b}_p P' \wedge Q_\oplus = 0 \wedge b \notin A}{P \|_A^{p_1,p_2} Q \xrightarrow{b}_{p_1\cdot p_2\cdot r_1} P' \|_A^{p_1,p_2} Q} \qquad \frac{Q \xrightarrow{b}_p Q' \wedge P_\oplus = 0 \wedge b \notin A}{P \|_A^{p_1,p_2} Q \xrightarrow{b}_{(1-p_1)\cdot p_2\cdot r_1} P \|_A^{p_1,p_2} Q'}$$

$$\frac{P \xrightarrow{a}_p P' \wedge Q \xrightarrow{a}_q Q' \wedge a \in A}{P \|_A^{p_1,p_2} Q \xrightarrow{a}_{(1-p_2)r_2} P' \|_A^{p_1,p_2} Q'}$$

donde $r_1 = \frac{p}{\mu(P,Q,A,p_1,p_2)}$ y $r_2 = \frac{p\cdot q}{\mu(P,Q,A,p_1,p_2)}$.

Figura A.3: Reglas para el operador $\|_A^{p_1,p_2}$.

en los que $\mu(P, Q, A, p_1, p_2)$ es un nuevo *factor de normalización*, que se define de forma análoga a como se hizo en el caso anterior, pero incorporando en los sumandos correspondientes a las acciones que pertenecen al conjunto de sincronización el factor $(1 - p_2)$, mientras que las correspondientes a las transiciones observables quedan multiplicadas por p_2. En definitiva,

$$\begin{aligned}\mu(P,Q,A,p_1,p_2) &= (1-p_2)\cdot \textstyle\sum_{a\in A} \{\!|\, p\cdot q \mid \exists P', Q' :\ P \xrightarrow{a}_p P' \wedge Q \xrightarrow{a}_q Q' \,|\!\} \\ &+ p_1\cdot p_2 \cdot \textstyle\sum_{a\notin A} \{\!|\, p \mid \exists P' :\ P \xrightarrow{a}_p P' \,|\!\} \\ &+ (1-p_1)\cdot p_2 \cdot \textstyle\sum_{a\notin A} \{\!|\, p \mid \exists Q' :\ Q \xrightarrow{a}_p Q' \,|\!\}\end{aligned}$$

En la Figura A.3 presentamos agrupadas todas las reglas de este operador.

A.2 Semántica del operador $\|_A^p$

En esta sección estudiaremos el modo de definir una semántica denotacional para el operador $\|_A^p$ en los dos modelos que hemos considerado en este trabajo.

Al igual que ocurría en el modelo reactivo con la elección externa, la relación $\approx_{\mathcal{R}}$ no es una congruencia respecto de los operadores de composición paralela, como se ve en el siguiente ejemplo. Por ello no podemos aspirar a definir una semántica

denotacional para este operador en el marco de dicho modelo.

Ejemplo A.2 Sean $P = (a;c) +_{\frac{1}{2}} b$ y $P' = (a;c) +_{\frac{1}{3}} b$. Tenemos que $P \approx_{\mathcal{R}} P'$. Consideremos ahora el proceso $Q = a;d$. Tenemos que $P \|_{\emptyset}^{\frac{1}{2}} Q \not\approx_{\mathcal{R}} P' \|_{\emptyset}^{\frac{1}{2}} Q$, dado que $pass(P \|_{\emptyset}^{\frac{1}{2}} Q\ ,\ a;d;\omega) = \frac{3}{4} \neq \frac{2}{3} = pass(P' \|_{\emptyset}^{\frac{1}{2}} Q\ ,\ a;d;\omega)$. □

A.2.1 Semántica denotacional de $\|_A^p$

En contraste con lo que ocurre en el modelo reactivo, en el modelo generativo tenemos que el operador $\|_A^p$ es congruente y por tanto podemos definirlo en el marco de la semántica denotacional. Pasamos a continuación a definir la función semántica asociada al operador $\|_A^p$, que nos permitirá incorporar este operador en el desarrollo semántico para el modelo generativo descrito en la Sección 4.4.

Dado $p \in (0,1)$ y $A \subseteq Act$, la función $\_ \|_A^p \_ :: \mathbf{PAT}_{\mathrm{Act}} \times \mathbf{PAT}_{\mathrm{Act}} \longrightarrow \mathbf{PAT}_{\mathrm{Act}}$ devolverá un árbol de aceptación que represente la *composición paralela sincronizando en* A de los correspondientes árboles, en función de la probabilidad p.

Antes de definir la función semántica asociada al operador $\|_A^p$, introduciremos un operador auxiliar que nos permitirá unir dos estados de acuerdo a una cierta probabilidad y a un cierto conjunto de sincronización. El nuevo estado contendrá aquellas acciones que no perteneciendo al conjunto A lo hagan a alguno de los dos estados, junto con las acciones que pertenezcan a A y a ambos estados al tiempo. La probabilidad asociada con las acciones del nuevo estado se calculará a partir de la que tenían en los estados combinados, multiplicándola por el factor p (resp. $1-p$) si no pertenecen al conjunto de sincronización y pertenecen al primer estado (resp. al segundo). Si la acción pertenece a los dos estados, la probabilidad asociada será una suma *ponderada* de las probabilidades que tenía en los estados combinados. En caso de pertenecer al conjunto de sincronización, se considerará el producto de las probabilidades asociadas a la acción en los estados combinados. Además, a fin de que las probabilidades del nuevo estado sumen 1, se utiliza un factor de normalización similar al utilizado al definir la semántica operacional.

Definición A.3 Sean X, Y estados, $A \subseteq Act$ y $p \in (0,1)$. Definimos la *unión* de los estados X y Y con *probabilidad asociada* p y *conjunto de sincronización* A en la forma siguiente:

$$X \|^p_A Y = \{(a, p_a) \mid a \in (Act(X) \cap Act(Y) \cap A) \cup (Act(X) - A) \cup (Act(Y) - A)\}$$

donde la probabilidad p_a viene dada por

$$p_a = \begin{cases} \dfrac{p \cdot pro(a,X) + (1-p) \cdot pro(a,Y)}{\mu(X,Y,A,p)} & \text{si } a \notin A \\ \dfrac{pro(a,X) \cdot pro(a,Y)}{\mu(X,Y,A,p)} & \text{si } a \in A \end{cases}$$

con $\mu(X,Y,A,p)$ dado por:

$$\begin{aligned} \mu(X,Y,A,p) &= \textstyle\sum_{a \in A} \{\!|\, pro(a,X) \cdot pro(a,Y) \,|\!\} \\ &+ p \cdot \textstyle\sum_{a \notin A} \{\!|\, pro(a,X) \,|\!\} + (1-p) \cdot \sum_{a \notin A} \{\!|\, pro(a,Y) \,|\!\} \end{aligned}$$

□

Nótese la sobrecarga del símbolo $\|^p_A$ que utilizamos tanto para denotar la composición paralela de procesos como la unión de estados, y de la función μ que denota el factor de normalización para procesos sintácticos y el utilizado para *normalizar* la unión de estados en los procesos semánticos.

Al igual que hicimos en el caso de la elección externa tendremos que distinguir entre la definición de la raíz del nuevo árbol y la de las distintas continuaciones bajo ella.

Para definir la raíz del nuevo árbol consideramos la unión, utilizando la función $\|^p_A$, de los estados iniciales de los dos árboles que estamos componiendo en paralelo.

$$p(R_1 \|^p_A R_2, \epsilon, X) = \sum_{X = B \|^p_A C} p(R_1, \epsilon, B) \cdot p(R_2, \epsilon, C)$$

De modo que de la la raíz del nuevo árbol sale un arco etiquetado con el estado X si y sólo si existen un arco que sale de la raíz del árbol R_1 etiquetado con el estado B y un arco que sale de la raíz del árbol R_2 etiquetado con el estado C tales que $X = B \parallel_A^p C$. La probabilidad que etiqueta el arco será igual a la suma del producto de las probabilidades que etiquetan los arcos correspondientes a los distintos pares de estados B y C que se pueden combinar dándonos X.

Al igual que ocurría para la elección externa, tenemos que la función semántica correspondiente al operador paralelo es estricta en sus dos argumentos, dado que si un argumento es $[\![\Omega]\!]$, el resultado seguirá siendo dicho valor, dado que de la raíz de $[\![\Omega]\!]$ no parte ningún arco; es decir $\forall\, s, X :\ p([\![\Omega]\!], s, X) = 0$.

$$[\![P \parallel_A^p \Omega]\!] = [\![\Omega \parallel_A^p P]\!] = [\![\Omega]\!] \qquad (\forall\, A, 0 < p < 1)$$

Además, *Nil* es elemento neutro de dicha función semántica cuando el conjunto de sincronización asociado al operador paralelo es el conjunto vacío

$$[\![P \parallel_\emptyset^p Nil]\!] = [\![Nil \parallel_\emptyset^p P]\!] = [\![P]\!] \qquad (\forall\, 0 < p < 1)$$

Ahora tenemos que definir como será el resto del árbol bajo la raíz, es decir como definiremos $p(R_1 \parallel_A^p R_2, s, X)$ a partir de los árboles R_1 y R_2. En este caso determinaremos como se puede construir el primer estado que aparece en la secuencia s a partir de los estados iniciales de R_1 y R_2, y dependiendo de qué estado (el de R_1, el de R_2, o ambos) contenga la acción asociada al primer estado de la secuencia s, y de que esta acción pertenezca o no al conjunto de sincronización, el árbol que *evolucionará* será R_1, R_2, o ambos al tiempo. Si los dos estados contienen dicha primera acción, y la misma no pertenece al conjunto de sincronización, se realizará una *elección interna* en la cual intervendrán el parámetro asociado al operador paralelo y las respectivas probabilidades asociadas a la acción en los estados correspondientes de R_1 y R_2. Si la acción pertenece al conjunto de sincronización, serán ambos árboles los que evolucionen. Esta evolución de los árboles vendrá dada por la operación de continuación de un árbol después de ejecutar una acción a partir de un estado:

$R/(A,a)$, (ver Definición 4.21). En definitiva

$$p(R_1 \|^p_A R_2, \langle B\, b\rangle \circ s', X) = \sum_{B=C \|^p_A D} p(R_1,\epsilon,C)\cdot p(R_2,\epsilon,D)\cdot$$

$$\begin{cases} q_1 \cdot p(R_1/(C,b) \|^p_A R_2, s', X) + \\ q_2 \cdot p(R_1 \|^p_A R_2/(D,b), s', X) & \text{si } b \in (Act(C)-A)\cup(Act(D)-A) \\ p(R_1/(C,b) \|^p_A R_2/(D,b), s', X) & \text{si } b \in A \cap Act(C) \cap Act(D) \\ 0 & \text{e.o.c.} \end{cases}$$

donde $q_1 = \frac{p\cdot pro(b,C)}{p\cdot pro(b,C)+(1-p)\cdot pro(b,D)}$ y $q_2 = \frac{(1-p)\cdot pro(b,D)}{p\cdot pro(b,C)+(1-p)\cdot pro(b,D)}$.

A continuación probaremos la monotonía y continuidad del operador. Para ello necesitamos el resultado que presentaremos en el Lema A.5, el cual indica que el operador $/(A,a)$ es monótono en un cierto sentido. Nótese que sin embargo este operador no es monótono en general, como muestra el siguiente

Ejemplo A.4 Consideremos los siguientes procesos:

$$P_1 = (a;(b \oplus_{\frac{1}{3}} \Omega)) \oplus_{\frac{1}{3}} \Omega$$
$$P_2 = (a;(b \oplus_{\frac{1}{4}} \Omega)) \oplus_{\frac{1}{2}} \Omega$$

Se verifica que $[\![P_1]\!] \sqsubseteq_{\mathbf{PAT}} [\![P_2]\!]$, pero sin embargo

$$p([\![P_1]\!]/(\{(a,1)\},a),\epsilon,\{(b,1)\}) = \frac{1}{3} > p([\![P_2]\!]/(\{(a,1)\},a),\epsilon,\{(b,1)\}) = \frac{1}{4}$$

□

Lema A.5 Sean $R_1, R_2 \in \mathbf{PAT}_{\mathrm{Act}}$ tales que $R_1 \sqsubseteq_{\mathbf{PAT}} R_2$. Entonces, para todo estado A tal que $p(R_1,A) > 0$, para toda acción $a \in Act(A)$, para toda secuencia s, y para todo estado X, se verifica:

$$p(R_1,\epsilon,A)\cdot p(R_1/(A,a),s,X) \leq p(R_2,\epsilon,A)\cdot p(R_2/(A,a),s,X)$$

Demostración: La misma es francamente sencilla, pues tenemos

$$\begin{aligned} p(R_1, \epsilon, A) \cdot p(R_1/(A,a), s, X) &= p(R_1, \langle A\, a\rangle \circ s, X) \\ &\leq p(R_2, \langle A\, a\rangle \circ s, X) \\ &= p(R_2, \epsilon, A) \cdot p(R_2/(A,a), s, X) \end{aligned}$$

□

Proposición A.6 Las funciones $\_ \|^p_A \_ :: \mathbf{PAT}_{\mathrm{Act}} \times \mathbf{PAT}_{\mathrm{Act}} \longrightarrow \mathbf{PAT}_{\mathrm{Act}}$ son monótonas y continuas en sus dos argumentos, para cualesquiera $0 < p < 1$ y $A \subseteq Act$.

Demostración: Como de costumbre, es suficiente hacer la demostración para uno de los argumentos. En nuestro caso, lo haremos para el primero.

Monotonía.

Sean $R_1, R_2 \in \mathbf{PAT}_{\mathrm{Act}}$ tales que $R_1 \sqsubseteq_{\mathbf{PAT}} R_2$. Tenemos que ver que se cumple $R_1 \|^p_A R \sqsubseteq_{\mathbf{PAT}} R_2 \|^p_A R$, para cualquier $R \in \mathbf{PAT}_{\mathrm{Act}}$, o lo que es lo mismo, que para cualquier secuencia s y para cualquier estado X se tiene:

$$p(R_1 \|^p_A R, s, X) \leq p(R_2 \|^p_A R, s, X)$$

La demostración la haremos en dos pasos. Primero lo probaremos para la secuencia vacía, y después para las secuencias no vacías.

- Secuencia vacía ($s = \epsilon$).

$$\begin{aligned} p(R_1 \|^p_A R, \epsilon, X) &= \sum_{X=B \|^p_A C} p(R_1, \epsilon, B) \cdot p(R, \epsilon, C) \\ &\leq \sum_{X=B \|^p_A C} p(R_2, \epsilon, B) \cdot p(R, \epsilon, C) \\ &= p(R_2 \|^p_A R, \epsilon, X) \end{aligned}$$

- Secuencia no vacía ($s = \langle B\, b\rangle \circ s'$).

$$p(R_1 \|^p_A R, s, X) = \sum_{B=C \|^p_A D} p(R_1, \epsilon, C) \cdot p(R, \epsilon, D) \cdot$$

$$\begin{cases} q_1 \cdot p(R_1/(C,b) \|^p_A R, s', X) + \\ q_2 \cdot p(R_1 \|^p_A R/(D,b), s', X) & \text{si } b \in (Act(C) - A) \cup (Act(D) - A) \\ p(R_1/(C,b) \|^p_A R/(D,b), s', X) & \text{si } b \in A \cap Act(C) \cap Act(D) \\ 0 & \text{e.o.c.} \end{cases}$$

$\leq$ (por hipótesis de inducción y aplicación del Lema A.5)

$$\sum_{B=C \;\|^p_A\; D} p(R_2, \epsilon, C) \cdot p(R, \epsilon, D) \cdot \begin{cases} q_1 \cdot p(R_2/(C,b) \;\|^p_A\; R, s', X) + \\ q_2 \cdot p(R_2 \;\|^p_A\; R/(D,b), s', X) & \text{si } b \in (Act(C) - A) \cup (Act(D) - A) \\ p(R_2/(C,b) \;\|^p_A\; R/(D,b), s', X) & \text{si } b \in A \cap Act(C) \cap Act(D) \\ 0 & \text{e.o.c.} \end{cases}$$

$= p(R_2 \;\|^p_A\; R, s, X)$

Continuidad.

Sea $\{R_n\}_{n\in\mathbb{N}}$ una cadena de elementos de $\mathbf{PAT}_{\mathrm{Act}}$. Por la monotonía del operador paralelo, tenemos que $\{R_n \;\|^p_A\; R\}_{n\in\mathbb{N}}$ también es una cadena. Tenemos que probar que para toda secuencia s y para todo estado X, se cumple

$$p(\sqcup\{R_n\}_{n\in\mathbb{N}} \;\|^p_A\; R, s, X) = p(\sqcup\{R_n \;\|^p_A\; R\}_{n\in\mathbb{N}}, s, X)$$

Al igual que antes, haremos la demostración en dos pasos:

- Secuencia vacía ($s = \epsilon$).

$$\begin{aligned} p(\sqcup\{R_n\}_{n\in\mathbb{N}} \;\|^p_A\; R, \epsilon, X) &= \sum_{X=B \;\|^p_A\; C} p(\sqcup\{R_n\}_{n\in\mathbb{N}}, \epsilon, B) \cdot p(R, \epsilon, C) \\ &= \sum_{X=B \;\|^p_A\; C} (\lim_{n\in\mathbb{N}} p(R_n, \epsilon, B)) \cdot p(R, \epsilon, C) \\ &= \lim_{n\in\mathbb{N}} \sum_{X=B \;\|^p_A\; C} p(R_n, \epsilon, B) \cdot p(R, \epsilon, C) \\ &= \lim_{n\in\mathbb{N}} p(R_n \;\|^p_A\; R, \epsilon, X) = p(\sqcup\{R_n \;\|^p_A\; R\}_{n\in\mathbb{N}}, \epsilon, X) \end{aligned}$$

- Secuencia no vacía ($s = \langle B\, b\rangle \circ s'$).

$$p(\sqcup\{R_n \parallel_A^p R\}_{n\in\mathbb{N}}, s, X) = \lim_{n\in\mathbb{N}} p(R_n \parallel_A^p R, s, X) =$$

$$\lim_{n\in\mathbb{N}} \sum_{B=C \parallel_A^p D} p(R_n, \epsilon, C) \cdot p(R, \epsilon, D) \cdot$$

$$\begin{cases} q_1 \cdot p(R_n/(C,b) \parallel_A^p R, s', X) + \\ q_2 \cdot p(R_n \parallel_A^p R/(D,b), s', X) & \text{si } b \in (Act(C) - A) \cup (Act(D) - A) \\ p(R_n/(C,b) \parallel_A^p R/(D,b), s', X) & \text{si } b \in A \cap Act(C) \cap Act(D) \\ 0 & \text{e.o.c.} \end{cases}$$

$=$

$$\sum_{B=C \parallel_A^p D} \lim_{n\in\mathbb{N}} p(R_n, \epsilon, C) \cdot p(R, \epsilon, D) \cdot$$

$$\begin{cases} q_1 \cdot \lim_{n\in\mathbb{N}} p(R_n/(C,b) \parallel_A^p R, s', X) + \\ q_2 \cdot \lim_{n\in\mathbb{N}} p(R_n \parallel_A^p R/(D,b), s', X) & \text{si } b \in (Act(C) - A) \cup (Act(D) - A) \\ \lim_{n\in\mathbb{N}} p(R_n/(C,b) \parallel_A^p R/(D,b), s', X) & \text{si } b \in A \cap Act(C) \cap Act(D) \\ 0 & \text{e.o.c.} \end{cases}$$

$=$

$$\sum_{B=C \parallel_A^p D} p(\sqcup\{R_n\}_{n\in\mathbb{N}}, \epsilon, C) \cdot p(R, \epsilon, D) \cdot$$

$$\begin{cases} q_1 \cdot p(\sqcup\{R_n\}_{n\in\mathbb{N}}/(C,b) \parallel_A^p R, s', X) + \\ q_2 \cdot p(\sqcup\{R_n\}_{n\in\mathbb{N}} \parallel_A^p R/(D,b), s', X) & \text{si } b \in (Act(C) - A) \cup (Act(D) - A) \\ p(\sqcup\{R_n\}_{n\in\mathbb{N}}/(C,b) \parallel_A^p R/(D,b), s', X) & \text{si } b \in A \cap Act(C) \cap Act(D) \\ 0 & \text{e.o.c.} \end{cases}$$

$= p(\sqcup\{R_n\}_{n\in\mathbb{N}} \parallel_A^p R, s, A)$

Nótese que a lo largo de la demostración hemos hecho uso de la hipótesis de inducción para deducir los siguientes resultados:

$$\begin{array}{ccc} q_1 \cdot \lim\limits_{n\in\mathbb{N}} p(R_n/(C,b) \|_A^p R, s', X) & & q_1 \cdot p(\sqcup\{R_n\}_{n\in\mathbb{N}}/(C,b) \|_A^p R, s', X) \\ + & = & + \\ q_2 \cdot \lim\limits_{n\in\mathbb{N}} p(R_n \|_A^p R/(D,b), s', X) & & q_2 \cdot p(\sqcup\{R_n\}_{n\in\mathbb{N}} \|_A^p R/(D,b), s', X) \end{array}$$

$$\lim_{n\in\mathbb{N}} p(R_n/(C,b) \|_A^p R/(D,b), s', X) = p(\sqcup\{R_n\}_{n\in\mathbb{N}}/(C,b) \|_A^p R/(D,b), s', X)$$

□

A.2.2 Semántica axiomática de $\|_A^p$

En esta sección estudiaremos los axiomas correspondientes al operador $\|_A^p$ en el modelo generativo. Al igual que ocurre en los modelos no probabilísticos en los que no se considera *concurrencia real*, el operador paralelo podrá ser considerado como derivado a partir de los restantes operadores. Esta noción de *derivabilidad* nos permitirá reducir la aplicación del operador paralelo sobre dos procesos en forma normal a una expresión en la que, en general, siguen apareciendo operadores paralelos pero nunca en *cabeza* de la expresión. Por lo tanto, aplicando reiteradamente los axiomas, podemos eliminar completamente los operadores paralelos en aquellos casos en los que el operador paralelo no aparezca en el ámbito de un operador de recursión. Cuando el operador paralelo aparece en el ámbito de una recursión, aunque no los podemos eliminar totalmente, podemos *hundir* las apariciones del operador paralelo tanto como deseemos.

Los axiomas para el operador paralelo indican que el mismo es conmutativo y que distribuye sobre la elección interna. Además, tenemos un axioma de *expansión* similar al que aparece en las álgebras de procesos no probabilísticas, y consideraremos por separado el caso particular en el que uno de los procesos es *Nil*. Por último tenemos un axioma que nos indica que al componer en paralelo un proceso con un procesos divergente el resultado es la divergencia. Tras presentar cada axioma incluiremos la correspondiente demostración de corrección.

- El operador $\|_A^p$ es conmutativo.

$$\textbf{(CP)} \quad P \|_A^p Q \equiv_{\mathbf{gen}} Q \|_A^{1-p} P$$

Demostración: Sea $T \in \mathcal{PB}$. Aplicando las reglas $(PAR1), (PAR2)$ y $(PAR3)$ vemos que la probabilidad asociada a la composición paralela no influye en la probabilidad con la que dicha composición ejecuta transiciones internas a partir de los procesos componentes. Es decir, $P \parallel_A^p Q \succ\!\!\rightarrow_q^* P' \parallel_A^p Q' \iff Q \parallel_A^{1-p} P \succ\!\!\rightarrow_q^* Q' \parallel_A^{1-p} P'$. Una vez que el proceso es estable, aplicando las reglas $(PAR4)$, $(PAR5)$ y $(PAR6)$, obtenemos $P' \parallel_A^p Q' \xrightarrow{a}_q R \iff Q' \parallel_A^{1-p} P' \xrightarrow{a}_q R$, con lo que

$$\begin{aligned} pass(P \parallel_A^p Q\ ,\ T) &= \textstyle\sum_{P',Q'} \{\!\{\ q \cdot pass(P' \parallel_A^p Q'\ ,\ T) \mid P \parallel_A^p Q \succ\!\!\rightarrow_q^* P' \parallel_A^p Q'\ \}\!\} \\ &= \textstyle\sum_{P',Q'} \{\!\{\ q \cdot pass(Q' \parallel_A^{1-p} P'\ ,\ T) \mid Q \parallel_A^{1-p} P \succ\!\!\rightarrow_q^* Q' \parallel_A^{1-p} P'\ \}\!\} \\ &= pass(Q \parallel_A^{1-p} P\ ,\ T) \end{aligned}$$

- El operador $\parallel_A^p$ distribuye sobre el operador $\oplus_q$.

$$\textbf{(DPI)} \quad P_1 \parallel_A^p (P_2 \oplus_q P_3) \equiv_{\mathbf{gen}} (P_1 \parallel_A^p P_2) \oplus_q (P_1 \parallel_A^p P_3)$$

Demostración: Sea $T \in \mathcal{PB}$. Aplicando reiteradamente las reglas $(PAR1), (PAR2)$ y $(PAR3)$ obtenemos

$$P_i \succ\!\!\rightarrow_{r_i}^* P_i' \ \Rightarrow\ P_1 \parallel_A^p P_2 \succ\!\!\rightarrow_{r_1 \cdot r_2}^* P_1' \parallel_A^p P_2' \ \wedge\ P_1 \parallel_A^p P_3 \succ\!\!\rightarrow_{r_1 \cdot r_3}^* P_1' \parallel_A^p P_3'$$

$$\Rightarrow \begin{cases} (P_1 \parallel_A^p P_2) \oplus_q (P_1 \parallel_A^p P_3) \succ\!\!\rightarrow_{q \cdot r_1 \cdot r_2}^* (P_1' \parallel_A^p P_2') \\ (P_1 \parallel_A^p P_2) \oplus_q (P_1 \parallel_A^p P_3) \succ\!\!\rightarrow_{(1-q) \cdot r_1 \cdot r_3}^* (P_1' \parallel_A^p P_3') \end{cases}$$

Por otra parte tenemos

$$P_i \succ\!\!\rightarrow_{r_i}^* P_i' \ \Rightarrow\ P_2 \oplus_q P_3 \succ\!\!\rightarrow_{q \cdot r_2}^* P_2' \ \wedge\ P_2 \oplus_q P_3 \succ\!\!\rightarrow_{(1-q) r_2}^* P_3'$$

$$\Rightarrow \begin{cases} P_1 \parallel_A^p (P_2 \oplus_q P_3) \succ\!\!\rightarrow_{r_1 \cdot q \cdot r_2}^* (P_1' \parallel_A^p P_2') \\ P_1 \parallel_A^p (P_2 \oplus_q P_3) \succ\!\!\rightarrow_{r_1 \cdot (1-q) \cdot r_3}^* (P_1' \parallel_A^p P_3') \end{cases}$$

De ambos resultados se deduce

$$\begin{aligned} (P_1 \parallel_A^p P_2) \oplus_q (P_1 \parallel_A^p P_3) \succ\!\!\rightarrow_r^* (P_1' \parallel_A^p P_2') &\iff P_1 \parallel_A^p (P_2 \oplus_q P_3) \succ\!\!\rightarrow_r^* (P_1' \parallel_A^p P_2') \\ (P_1 \parallel_A^p P_2) \oplus_q (P_1 \parallel_A^p P_3) \succ\!\!\rightarrow_r^* (P_1' \parallel_A^p P_3') &\iff P_1 \parallel_A^p (P_2 \oplus_q P_3) \succ\!\!\rightarrow_r^* (P_1' \parallel_A^p P_3') \end{aligned}$$

Consideramos ahora los siguientes multiconjuntos de pares (proceso, probabilidad):

$$\tilde{P}_1 = \{\!\{ (P_1', r_1) \mid P_1 \rightarrow^*_{r_1} P_1' \}\!\}$$
$$\tilde{P}_2 = \{\!\{ (P_2', r_2) \mid P_2 \rightarrow^*_{r_2} P_2' \}\!\}$$
$$\tilde{P}_3 = \{\!\{ (P_3', r_3) \mid P_3 \rightarrow^*_{r_3} P_3' \}\!\}$$

A partir de los resultados anteriores obtenemos

$$\begin{aligned}
&pass((P_1 \|^p_A P_2) \oplus_q (P_1 \|^p_A P_3), T) \\
&\quad = \sum \{\!\{ q \cdot r_1 \cdot r_2 \cdot pass(P_1' \|^p_A P_2', T) \mid (P_1', r_1) \in \tilde{P}_1 \wedge (P_2', r_2) \in \tilde{P}_2 \}\!\} \\
&\quad + \sum \{\!\{ (1-q) \cdot r_1 \cdot r_3 \cdot pass(P_1' \|^p_A P_3', T) \mid (P_1', r_1) \in \tilde{P}_1 \wedge (P_3', r_3) \in \tilde{P}_3 \}\!\} \\
&\quad = \sum \{\!\{ r_1 \cdot q \cdot r_2 \cdot pass(P_1' \|^p_A P_2', T) \mid (P_1', r_1) \in \tilde{P}_1 \wedge (P_2', r_2) \in \tilde{P}_2 \}\!\} \\
&\quad + \sum \{\!\{ r_1 \cdot (1-q) \cdot r_3 \cdot pass(P_1' \|^p_A P_3', T) \mid (P_1', r_1) \in \tilde{P}_1 \wedge (P_3', r_3) \in \tilde{P}_3 \}\!\} \\
&\quad = pass(P_1 \|^p_A (P_2 \oplus_q P_3), T)
\end{aligned}$$

Este axioma se puede generalizar de forma trivial para cubrir el caso de la elección interna generalizada.

$$\textbf{(DPIG)} \quad P \|^p_A (\bigoplus_{i=1}^{n} [p_i] P_i) \equiv_{\text{gen}} \bigoplus_{i=1}^{n} [p_i] (P \|^p_A P_i)$$

Antes de ver el axioma de expansión precisaremos una definición auxiliar similar a la introducida en la Definición 5.7.

Definición A.7 Sean $A, B, X \subseteq Act$. Dada una barba probabilística T, consideremos su conjunto de acciones iniciales $\tilde{T} = \{t_1, \ldots, t_u\}$. Entonces, a partir de A, B, X, y $\tilde{T}$ construimos los siguientes conjunto de acciones:

$$\begin{aligned}
T_{A-X} &= \{t_i \mid t_i \in (A - X) \cap \tilde{T}\} & T_{B-X} &= \{t_i \mid t_i \in (B - X) \cap \tilde{T}\} \\
T_{A\cap B\cap X} &= \{t_i \mid t_i \in (A \cap B \cap X) \cap \tilde{T}\} & T_{(A\cap B)-X} &= \{t_i \mid t_i \in ((A \cap B) - X) \cap \tilde{T}\} \\
T_{(A-B)-X} &= \{t_i \mid t_i \in T_{A-X} - B\} & T_{(B-A)-X} &= \{t_i \mid t_i \in T_{B-X} - A\}
\end{aligned}$$

□

• Sean $A = \{a_1, \ldots, a_n\} \subseteq Act$ y $B = \{b_1, \ldots, b_m\} \subseteq Act$. Si consideramos los procesos $P = \sum_{i=1}^{n} [p_i]\, a_i; P_i$ y $Q = \sum_{j=1}^{m} [q_j]\, b_j; Q_j$, se tiene

$$\textbf{(EP)} \quad P \parallel_X^p Q \equiv_{\text{gen}} R$$

donde $R = \sum_{k=1}^{l} [\frac{r_k}{\mu(P,Q,X,p)}]\, c_k; R_k$, $C = \{c_1, \ldots, c_l\} = (A \cup B) - X \cup (A \cap B \cap X)$,

$$r_k = \begin{cases} p_i \cdot q_j & \text{si } c_k = a_i = b_j \in X \\ p \cdot p_i & \text{si } c_k = a_i \in (A - B) - X \\ (1-p) \cdot q_j & \text{si } c_k = b_j \in (B - A) - X \\ p \cdot p_i + (1-p) \cdot q_j & \text{si } c_k = a_i = b_j \in (A \cap B) - X \end{cases}$$

y

$$R_k = \begin{cases} P_i \parallel_X^p Q_j & \text{si } c_k = a_i = b_j \in X \\ P_i \parallel_X^p Q & \text{si } c_k = a_i \in (A - B) - X \\ P \parallel_X^p Q_j & \text{si } c_k = b_j \in (B - A) - X \\ (P_i \parallel_X^p Q) \oplus_{p'} (P \parallel_X^p Q_j) & \text{si } c_k = a_i = b_j \in (A \cap B) - X \\ & \text{siendo } p' = \frac{p \cdot p_i}{p \cdot p_i + (1-p) \cdot q_j} \end{cases}$$

Demostración: Para abreviar, utilizaremos la notación $p_i = p(P, a_i)$ y $q_j = p(Q, b_j)$. Si $a \notin X$, entonces aplicando las reglas $(PAR4)$ y $(PAR5)$ obtenemos

$$P \xrightarrow{a}_q P' \Longrightarrow P \parallel_X^p Q \xrightarrow{a}_{p_1} P' \parallel_X^p Q \quad \text{y} \quad Q \xrightarrow{a}_q Q' \Longrightarrow P \parallel_X^p Q \xrightarrow{a}_{p_2} P \parallel_X^p Q'$$

donde $p_1 = \frac{p \cdot q}{\mu(P,Q,X,p)}$ y $p_2 = \frac{(1-p) \cdot q}{\mu(P,Q,X,p)}$. Además, si $a \in X$, entonces aplicando la regla $(PAR6)$ se tiene

$$P \xrightarrow{a}_{p_1} P' \wedge Q \xrightarrow{a}_{p_2} Q' \Longrightarrow P \parallel_X^p Q \xrightarrow{a}_{\frac{p_1 \cdot p_2}{\mu(P,Q,X,p)}} P' \parallel_X^p Q'$$

Demostraremos que para todo $T \in \mathcal{PB}$ se tiene $pass(P \parallel_X^p Q\,,\, T) = pass(R\,,\, T)$.

Si T es una barba probabilística de la forma $T = \sum_{i=1}^{u}[s_i]\ (t_i; Nil) +_s \omega$, entonces podemos deducir

$$pass(P \|^p_X Q\ ,\ T) =$$

$$\frac{1-s}{(1-s)+\sum\limits_{t_i\in T_{A-X}} s\cdot s_i\cdot\frac{p\cdot p(P,t_i)}{\mu(P,Q,X,p)} + \sum\limits_{t_i\in T_{B-X}} s\cdot s_i\cdot\frac{(1-p)\cdot p(Q,t_i)}{\mu(P,Q,X,p)} + s\cdot R_{A\cap B\cap X}} =$$

$$\frac{1-s}{(1-s)+\sum\limits_{t_i\in T_{(A-B)-X}} s\cdot s_i\cdot\frac{p\cdot p(P,t_i)}{\mu(P,Q,X,p)} + \sum\limits_{t_i\in T_{(B-A)-X}} s\cdot s_i\cdot\frac{(1-p)\cdot p(Q,t_i)}{\mu(P,Q,X,p)} + \sum\limits_{t_i\in T_{(A\cap B)-X}} s\cdot s_i\cdot\frac{(p\cdot p(P,t_i)+(1-p)\cdot p(Q,t_i))}{\mu(P,Q,X,p)} + s\cdot R_{A\cap B\cap X}} =$$

$$pass(R\ ,\ T)$$

donde $R_{A\cap B\cap X} = \sum\limits_{t_i\in T_{A\cap B\cap X}} s_i\cdot\frac{p(P,t_i)\cdot p(Q,t_i)}{\mu(P,Q,X,p)}$

Si T es una barba probabilística de la forma $T = \sum_{i=1}^{u}[s_i]\ t_i; T_i$ donde $T_i = T'$ si $i = u$ y $T_i = Nil$ si $i \neq u$, hemos de considerar varios casos dependiendo de los diferentes conjuntos de acciones a los que la acción t_u pueda pertenecer:

$\diamond\ \exists\ 1 \leq i \leq n:\ a_i = t_u \in (A - B) - X.$

$$pass(P \|^p_X Q\ ,\ T) =$$

$$\frac{s_u\cdot\frac{p\cdot p_i}{\mu(P,Q,X,p)}\cdot pass(P_i \|^p_X Q\ ,T')}{\sum\limits_{t_i\in T_{A-X}} s_i\cdot\frac{p\cdot p(P,t_i)}{\mu(P,Q,X,p)} + \sum\limits_{t_i\in T_{B-X}} s_i\cdot\frac{(1-p)\cdot p(Q,t_i)}{\mu(P,Q,X,p)} + R_{A\cap B\cap X}} =$$

$$\frac{s_u\cdot\frac{p\cdot p_i}{\mu(P,Q,X,p)}\cdot pass(P_i \|^p_X Q\ ,T')}{\sum\limits_{t_i\in T_{(A-B)-X}} s_i\cdot\frac{p\cdot p(P,t_i)}{\mu(P,Q,X,p)} + \sum\limits_{t_i\in T_{(B-A)-X}} s_i\cdot\frac{(1-p)\cdot p(Q,t_i)}{\mu(P,Q,X,p)} + \sum\limits_{t_i\in T_{(A\cap B)-X}} s_i\cdot\frac{(p\cdot p(P,t_i)+(1-p)\cdot p(Q,t_i))}{\mu(P,Q,X,p)} + R_{A\cap B\cap X}} =$$

$$pass(R\ ,\ T)$$

$\diamond\ \exists\ 1 \leq j \leq m:\ b_j = t_u \in (B - A) - X.$ Simétrico al anterior.

$\diamond\ \exists\ 1 \leq i \leq n,\ 1 \leq j \leq m :\ a_i = b_j = t_u \in (A \cap B) - X.$

$$pass(P \parallel_X^p Q\ ,\ T) =$$

$$\frac{s_u \cdot \frac{p \cdot p_i}{\mu(P,Q,X,p)} \cdot pass(P_i \parallel_X^p Q\ ,\ T') + s_u \cdot \frac{(1-p) \cdot q_j}{\mu(P,Q,X,p)} \cdot pass(P \parallel_X^p Q_j\ ,\ T')}{\sum\limits_{t_i \in T_{A-X}} s_i \cdot \frac{p \cdot p(P,t_i)}{\mu(P,Q,X,p)} + \sum\limits_{t_i \in T_{B-X}} s_i \cdot \frac{(1-p) \cdot p(Q,t_i)}{\mu(P,Q,X,p)} + R_{A \cap B \cap X}} =$$

$$\frac{s_u \cdot \frac{p \cdot p_i + (1-p) \cdot q_j}{\mu(P,Q,X,p)} \cdot pass((P_i \parallel_X^p Q) \oplus_{q'} (P \parallel_X^p Q_j)\ ,\ T')}{\sum\limits_{t_i \in T_{(A-B)-X}} s_i \cdot \frac{p \cdot p(P,t_i)}{\mu(P,Q,X,p)} + \sum\limits_{t_i \in T_{(B-A)-X}} s_i \cdot \frac{(1-p) \cdot p(Q,t_i)}{\mu(P,Q,X,p)} + \sum\limits_{t_i \in T_{(A \cap B)-X}} s_i \cdot \frac{(p \cdot p(P,t_i) + (1-p) \cdot p(Q,t_i))}{\mu(P,Q,X,p)} + R_{A \cap B \cap X}} =$$

$$pass(R\ ,\ T)$$

donde $q' = \frac{p \cdot p_i}{p \cdot p_i + (1-p) \cdot q_j}$.

$\diamond\ \exists\ 1 \leq i \leq n,\ 1 \leq j \leq m :\ a_i = b_j = t_u \in A \cap B \cap X.$

$$pass(P \parallel_X^p Q\ ,\ T) =$$

$$\frac{s_u \cdot \frac{p_i \cdot q_j}{\mu(P,Q,X,p)} \cdot pass(P_i \parallel_X^p Q_j\ ,\ T')}{\sum\limits_{t_i \in T_{A-X}} s_i \cdot \frac{p \cdot p(P,t_i)}{\mu(P,Q,X,p)} + \sum\limits_{t_i \in T_{B-X}} s_i \cdot \frac{(1-p) \cdot p(Q,t_i)}{\mu(P,Q,X,p)} + R_{A \cap B \cap X}} =$$

$$\frac{s_u \cdot \frac{p_i \cdot q_j}{\mu(P,Q,X,p)} \cdot pass(P_i \parallel_X^p Q_j\ ,\ T')}{\sum\limits_{t_i \in T_{(A-B)-X}} s_i \cdot \frac{p \cdot p(P,t_i)}{\mu(P,Q,X,p)} + \sum\limits_{t_i \in T_{(B-A)-X}} s_i \cdot \frac{(1-p) \cdot p(Q,t_i)}{\mu(P,Q,X,p)} + \sum\limits_{t_i \in T_{(A \cap B)-X}} s_i \cdot \frac{(p \cdot p(P,t_i) + (1-p) \cdot p(Q,t_i))}{\mu(P,Q,X,p)} + R_{A \cap B \cap X}} =$$

$$pass(R\ ,\ T)$$

$\diamond\ t_u \notin ((A \cup B) - X) \cup (A \cap B \cap X)$. En tal caso $pass(P \parallel_A^p Q, T) = 0 = pass(R, T)$.

Un caso particular de este axioma se presenta cuando uno de los operandos es *Nil*. En tal caso tenemos

- Sea $A = \{a_1, \ldots, a_n\} \subseteq Act$. Si Consideramos un proceso $P = \sum\limits_{i=1}^{n} [p_i]\, a_i; P_i$, se tiene

$$\textbf{(EPN)} \quad P \parallel_X^p Nil \equiv_{\text{gen}} \sum_{a_k \in A-X} [\tfrac{p_k}{\mu(P,Nil,X,p)}]\, a_k; P_k$$

(CP)	$P \parallel_A^p Q \equiv_{\mathbf{gen}} Q \parallel_A^{1-p} P$
(DPIG)	$P \parallel_A^p (\bigoplus_{i=1}^{n} [p_i]\, P_i) \equiv_{\mathbf{gen}} \bigoplus_{i=1}^{n} [p_i]\, (P \parallel_A^p P_i)$
(EP)	$(\sum_{i=1}^{n} [p_i]\, a_i; P_i) \parallel_X^p (\sum_{j=1}^{m} [q_j]\, b_j; Q_j) \equiv_{\mathbf{gen}} \sum_{k=1}^{l} [\frac{r_k}{\mu(P,Q,X,p)}]\, c_k; R_k$
(EPN)	$(\sum_{i=1}^{n} [p_i]\, a_i; P_i) \parallel_X^p Nil \equiv_{\mathbf{gen}} \sum_{a_k \in A-X} [\frac{p_k}{\mu(P,Nil,X,p)}]\, a_k; P_k$
(DP)	$P \parallel_A^p \Omega \equiv_{\mathbf{gen}} \Omega$

Figura A.4: Axiomas para el operador $\parallel_A^p$.

Demostración: La demostración de correción de este axioma es un caso particular de la del axioma anterior. Simplemente comentar que

$$\mu(P, Nil, X, p) = \sum_{i=1}^{n} \{\!|\, p_i \mid a_i \notin X \,|\!\}$$

El último de los axiomas indica que el operador $\parallel_A^p$ es estricto, siendo la demostración de su corrección trivial.

$$\textbf{(DP)} \quad P \parallel_A^p \Omega \equiv_{\mathbf{gen}} \Omega$$

En la Figura A.4 presentamos agrupados todos los axiomas asociados al operador paralelo. El nuevo sistema axiomático estaría formado por las reglas que aparecen en la Figura 5.2 junto con los axiomas que aparecen en las Figuras 5.3 y A.4.

A.3 Operador de Ocultamiento

Como hemos visto en la sección anterior, un operador paralelo (incluso con varias variantes) se puede introducir en el modelo desarrollado en este trabajo sin ser necesaria ninguna modificación en todo el desarrollo anterior. Lamentablemente, esto no ocurre con el operador de ocultamiento. A continuación comentaremos porqué tal operador no se puede incluir en nuestro lenguaje sin modificar fuertemente la semántica

que hemos desarrollado hasta el momento. Esta discusión está muy relacionada con la que expusimos en la Sección 2.4.2 al comentar los problemas que acarrea el hecho de tener un lenguaje probabilístico basado en CCS en el que aparezcan acciones ocultas, τ's, en el ámbito de una elección.

Si deseáramos que el operador de ocultamiento fuera derivado, de forma similar a lo que ocurre con el operador paralelo, se tendría que cumplir un cierto axioma de *distributividad* similar al que se tiene en CSP:

$$((a;c)+b)\backslash a \equiv c \oplus (c+b)$$

Vamos a demostrar que un axioma basado en el correspondiente al caso no probabilístico para PPA no es correcto en general, si hacemos una interpretación similar a la que se hace en otros modelos probabilísticos en los que aparecen acciones ocultas, de la forma en la que un proceso en el que aparecen acciones ocultas pasa unas ciertas pruebas.

Consideremos los procesos $P = (a;c) +_{\frac{1}{2}} b$, y $P' = P\backslash a$. Utilizando una sintaxis *a lo* CCS, tendríamos que P' sería equivalente al proceso $(\tau;c) +_{\frac{1}{2}} b$. Demostraremos que no existen r y s, con $0 < r, s < 1$, tales que $P'' = c \oplus_r (c +_s b) \equiv_{\mathbf{gen}} P'$.

Si componemos en paralelo el proceso P' con la prueba $T_1 = b;\omega$ deberíamos tener $pass(P', T_1) = \frac{1}{2}$, al igual que ocurre en otros modelos que utilizan acciones ocultas (e.g. [CSZ92]). Por lo tanto, r debería ser igual a $\frac{1}{2}$.

Consideremos la prueba $T_2 = b +_{\frac{1}{2}} (c;\omega)$. Tenemos $pass(P'', T_2) = \frac{1}{2} + \frac{1}{2} \cdot s$. De nuevo, haciendo una interpretación intuitiva del funcionamiento del operador de ocultamiento, si no consideramos factores de prenormalización tendríamos $pass(P', T_2) = \frac{2}{3}$, con lo que $s = \frac{1}{3}$. Consideremos ahora la prueba $T_3 = b +_{\frac{2}{3}} (c;\omega)$. Tenemos $pass(P'', T_3) = \frac{3}{4}$, mientras que por un razonamiento similar al utilizado anteriormente obtenemos $pass(P', T_3) = \frac{3}{5}$, lo cual hace imposible que P' y P'' sean equivalentes.

Si en los razonamientos anteriores utilizamos una adaptación de los factores de prenormalización presentados en el Capítulo 2 obtenemos un resultado bastante interesante. En este caso la probabilidad con la que P' pasa la prueba T_2 viene dada

por $pass(P', T_2) = \frac{1}{2}$. Esto es así dado que al componer el proceso con la prueba, la acción c que la prueba ofrece no puede ser ejecutada por el proceso P' (visto como el proceso $(\tau; c) +_{\frac{1}{2}} b)$) en su primer paso, con lo que la probabilidad asociada a c en la prueba debe transferirse a b en el primer paso de la composición. Por lo tanto, en este primer paso deberíamos considerar a todos los efectos de cálculo que la prueba ofrece la acción b con probabilidad 1. Pero este resultado nos lleva a que $s = 0$, y dado que en nuestro modelo las probabilidades asociadas a la elección externa deben pertenecer al intervalo $(0, 1)$, se trata de un valor no válido.

Precisamente este último resultado, en principio un tanto extraño, es el que creemos que puede ayudar a resolver el problema que plantea el operador de restricción. En concreto, P' debería ser equivalente al proceso $P'' = c \oplus_{\frac{1}{2}} (c +_0 b)$, donde $+_0$ sería un operador de prioridad similar a los utilizados en modelos como [SS90, Low95]. Pero la inclusión de un operador de prioridad complica (todavía más !!) nuestro modelo. Un primer intento se realizó en [NdF95a], pero el modelo resultó ser tan complejo que resultaba inmanejable. Por ejemplo, la semántica operacional del operador paralelo requería 6 reglas para cubrir los diferentes casos en los que ambos procesos podían efectuar transiciones de prioridad, a las que había que añadir las reglas correspondientes a las transiciones internas y a las transiciones observables.

Otra solución consistiría en incluir el operador de ocultamiento sin pretender que sea un operador derivado, lo cual complicaría también enormemente la tarea de dar semántica al lenguaje, pues en particular las posibles formas normales de los procesos serían sin duda mucho más complejas que en la actualidad, al tener que aparecer en ellas el operador de ocultamiento.

En todo caso, aunque consideramos interesante admitir el manejo de prioridades a la hora de especificar procesos concurrentes, éstas han quedado fuera del ámbito del presente trabajo, si bien confiamos en tener ocasión para estudiarlas en un futuro próximo.

Bibliografía

[Abr87] S. Abramsky. Observational equivalence as a testing equivalence. *Theoretical Computer Science*, 53(3):225–241, 1987.

[BA82] M. Ben-Ari. *Principles of Concurrent Programming.* Prentice Hall, 1982.

[BBK86] J.C.M. Baeten, J.A. Bergstra, and J.W. Klop. Syntax and defining equations for an interrupt mechanism in process algebra. *Fundamenta Informaticae*, 9(2):127–186, 1986.

[BBS92] J.C.M. Baeten, J.A. Bergstra, and S.A. Smolka. Axiomatizing probabilistic processes: ACP with generative probabilities. In *CONCUR'92, LNCS 630*, pages 472–485, 1992.

[BGLG93] P. Brémond-Grégoire, I. Lee, and R. Gerber. ACSR: An algebra of communicating shared resources with dense time and priorities. In *CONCUR'93, LNCS 715*, pages 417–431, 1993.

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

[BK84] J. A. Bergstra and J. W. Klop. Process algebra for synchronous communication. *Information and Control*, 60:109–137, 1984.

[BM89] B. Bloom and A. R. Meyer. A remark on bisimulation between probabilistic processes. In *Logic at Botik'89, LNCS 363*, pages 26–40, 1989.

[BR85] S.D. Brookes and A.W. Roscoe. An improved failures model for communicating processes. In *Seminar on Concurrency, LNCS 197*, pages 281–305, 1985.

[Bro83] S. D. Brookes. *A Model for Communicating Sequential Processes.* PhD thesis, Oxford University, 1983.

[BSW69] K.A. Bartlett, R.A. Scantlebury, and P.T. Wilkinson. A note on reliable full-duplex transmission over half-duplex links. *Communications of the ACM*, 12(5):260–261, 1969.

[BW82] M. Broy and M. Wirsling. On the algebraic specification of finitary infinite communicating sequential processes. In *Working Conference on Formal Description of Programming Concepts II*, 1982.

[BW90] J.C.M. Baeten and W.P. Weijland. *Process Algebra.* Cambridge Tracts in Computer Science 18. Cambridge University Press, 1990.

[CC92] L. Christoff and I. Christoff. Reasoning about safety and liveness properties for probabilistic processes. In *12th Foundations of Software Technology and Theoretical Computer Science, LNCS 652*, pages 342–355, 1992.

[CdFV96] F. Cuartero, D. de Frutos, and V. Valero. PCSP: A denotational model of probabilistic processes. In *3rd AMAST Workshop on Real-Time Systems*, 1996.

[CES86] E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[CH90] R. Cleaveland and M. Hennessy. Priorities in process algebras. *Information and Computation*, 87:58–77, 1990.

[Chr90a] I. Christoff. Testing equivalences and fully abstract models for probabilistic processes. In *CONCUR'90, LNCS 458*, pages 126–140, 1990.

[Chr90b] I. Christoff. *Testing Equivalences for Probabilistic Processes.* PhD thesis, Department of Computer Systems. Uppsala University, 1990.

[Chr93] L. Christoff. *Specification and Verification Methods for Probabilistic Processes.* PhD thesis, Department of Computer Systems. Uppsala University, 1993.

[CSZ92] R. Cleaveland, S.A. Smolka, and A.E. Zwarico. Testing preorders for probabilistic processes. In *19th ICALP, LNCS 623*, pages 708–719, 1992.

[Cua93] F. Cuartero. *CSP probabilístico (PCSP). Un modelo probabilístico de procesos concurrentes.* PhD thesis, Universidad Complutense de Madrid, 1993.

[CW95] J. Camilleri and G. Winskel. CCS with priority choice. *Information and Computation*, 116:26–37, 1995.

[dFNQ95] D. de Frutos, M. Núñez, and J. Quemada. Characterizing termination in LOTOS via testing. In *Protocol Specification, Testing and Verification XV*, pages 225–240. Chapman & Hall, 1995.

[Dij65] E.W. Dijkstra. Cooperating sequential processes. Technical Report EWD-123, Technological University Eindhoven, 1965. pp: 43–112.

[Dij71] E.W. Dijkstra. Hierarchical ordering of sequential processes. *Acta Informatica*, 1(2):115–138, 1971.

[dNH84] R. de Nicola and M.C.B. Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34:83–133, 1984.

[dNH87] R. de Nicola and M. Hennessy. CCS without τ's. In *TAPSOFT'87, LNCS 249*, pages 138–152, 1987.

[DS95] J. Davies and S. Schneider. A brief history of timed CSP. *Theoretical Computer Science*, 138:243–271, 1995.

[FH82] Y.A. Feldman and D. Harel. A probabilistic dinamic logic. In *14th ACM Symposium on Theory of Computing*, pages 181–195, 1982.

[GJS90] A. Giacalone, C.-C. Jou, and S.A. Smolka. Algebraic reasoning for probabilistic concurrent systems. In *Proceedings of Working Conference on Programming Concepts and Methods, IFIP TC 2*, 1990.

[Gre95] C. Gregorio. El tiempo como factor de prioridad en un álgebra probabilística. Trabajo de Doctorado. Dept. Informática y Automática. Universidad Complutense de Madrid, 1995.

[Han91] Hans A. Hansson. *Time and Probability in Formal Design of Distributed Systems.* PhD thesis, Department of Computer Systems. Uppsala University, 1991.

[Hen85] M. Hennessy. Acceptance trees. *Journal of the ACM*, 32(4):896–928, 1985.

[Hen88] M. Hennessy. *Algebraic Theory of Processes.* MIT Press, 1988.

[HJ89] H. Hansson and B. Jonsson. A framework for reasoning about time and realibility. In *10th IEEE Real-Time Systems Symposium*, 1989.

[HJ90] H. Hansson and B. Jonsson. A calculus for communicating systems with time and probabilities. In *11th IEEE Real-Time Systems Symposium*, pages 278–287, 1990.

[HJ94] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6:512–535, 1994.

[HM85] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, 1985.

[Hoa78] C.A.R. Hoare. Communicating sequential processes. *Communications of the ACM*, 21(8):666–677, 1978.

[Hoa85] C.A.R. Hoare. *Communicating Sequential Processes.* Prentice Hall, 1985.

[HP72] C.A.R. Hoare and R.H. Perrott, editors. *Operating Systems Techniques.* Academic Press, 1972.

[HS84] S. Hart and M. Sharir. Probabilistic temporal logics for finite and bounded models. In *16th ACM Symposium on Theory of Computing*, pages 1–13, 1984.

[HT91] D.T. Huynh and L. Tian. Complexity of deciding readiness and failure equivalences for processes. In *3rd IEEE Symposium on Parallel and Distributed Processing*, pages 738–745, 1991.

[HT92] D.T. Huynh and L. Tian. On some equivalence relations for probabilistic processes. *Fundamenta Informaticae*, 17:211–234, 1992.

[Jef92] A. Jeffrey. Translating timed process algebra into prioritized process algebra. In *Formal Techniques in Real Time and Fault-Tolerant Systems, LNCS 571*, pages 493–506, 1992.

[JHSY94] B. Jonsson, C. Ho-Stuart, and W. Yi. Testing and refinement for nondeterministic and probabilistic processes. In *Formal Techniques in Real-Time and Fault-Tolerant Systems, LNCS 863*, pages 418–430, 1994.

[JP89] C. Jones and G.D. Plotkin. A probabilistic powerdomain of evaluations. In *4th IEEE Symposium on Logic In Computer Science*, pages 186–195, 1989.

[JS90] C.-C. Jou and S.A. Smolka. Equivalences, congruences and complete axiomatizations for probabilistic processes. In *CONCUR'90, LNCS 458*, pages 367–383, 1990.

[JY95] B. Jonsson and W. Yi. Compositional testing preorders for probabilistic processes. In *10th IEEE Symposium on Logic In Computer Science*, 1995.

[KLL94] J.P. Katoen, R. Langerak, and D. Latella. Modeling systems by probabilistic process algebra: An event structures approach. In *Formal Description Techniques VI*, 1994.

[Koz83] D. Kozen. A probabilistic PDL. In *15th ACM Symposium on Theory of Computing*, pages 291–297, 1983.

[KS90] P.C. Kanellakis and S.A. Smolka. CCS expressions, finite state processes and three problems for equivalences. *Information and Computation*, 86:367–383, 1990.

[Lan93] R. Langerak. Bundle event structures: a non-interleaving semantics for LOTOS. In *Formal Description Techniques V*, pages 331–346, 1993.

[LdFN96] L. Llana, D. de Frutos, and M. Núñez. Testing semantics for urgent timed algebras. In *3rd AMAST Workshop on Real-Time Systems*. World Scientific, 1996.

[Lim84] INMOS Limited, editor. *Occam Programming Manual*. Prentice Hall, 1984.

[LOT88] LOTOS. A formal description technique based on the temporal ordering of observational behaviour. IS 8807, TC97/SC21, 1988.

[Low93] G. Lowe. *Probabilities and Priorities in Timed CSP*. PhD thesis, Oxford University, 1993.

[Low95] G. Lowe. Probabilistic and prioritized models of timed CSP. *Theoretical Computer Science*, 138:315–352, 1995.

[LS89] K. Larsen and A. Skou. Bisimulation through probabilistic testing. In *16th ACM Simposium on Principles of Programming Languages*, pages 344–352, 1989.

[LS91] K. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, 1991.

[LS92] K.G. Larsen and A. Skou. Compositional verification of probabilistic processes. In *CONCUR'92, LNCS 630*, pages 456–471, 1992.

[LT87] N.A. Lynch and M.R. Tuttle. Hierarchical correctness proofs for distributed algorithms. In *6th ACM Symp. on Principles of Distributed Computing*, pages 137–151, 1987.

[LV91] N.A. Lynch and F.W. Vaandrager. Forward and backward simulations for timing-based systems. In *REX Workshop "Real-Time: Theory in Practice", LNCS 600*, pages 397–446, 1991.

[MFV93] C. Miguel, A. Fernández, and L. Vidaller. LOTOS extended with probabilistic behaviours. *Formal Aspects of Computing*, 5:253–281, 1993.

[Mil80] R. Milner. *A Calculus of Communicating Systems.* LNCS 92, 1980.

[Mil83] R. Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 253:267–310, 1983.

[Mil89] R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.

[NCCC94] V. Natarajan, I. Christoff, L. Christoff, and R. Cleaveland. Priorities and abstraction in process algebra. In *14th Foundations of Software Technology and Theoretical Computer Science, LNCS 880*, pages 217–230, 1994.

[NdF95a] M. Núñez and D. de Frutos. A denotational model for $P_{\mathcal{G}}$CSP. Unpublished Notes, 1995.

[NdF95b] M. Núñez and D. de Frutos. The power of probabilistic tests. In *IV Jornadas Españolas de Concurrencia*, 1995.

[NdF95c] M. Núñez and D. de Frutos. Testing semantics for probabilistic LOTOS. In *Formal Description Techniques VIII.* Chapman & Hall, 1995.

[NdFL95] M. Núñez, D. de Frutos, and L. Llana. Acceptance trees for probabilistic processes. In *CONCUR'95, LNCS 962*, pages 249–263, 1995.

[NJ91] X. Nicollin and J.Sifakis. An overview and synthesis on timed process algebras. In *Computer Aided Verification'91, LNCS 575*, pages 376–398, 1991.

[Núñ93] M. Núñez. Московские научные задачи и другие проблемы. Manuscript. Москва, 1993.

[Núñ94] M. Núñez. Estudio de la asociatividad en el operador paralelo de PCSP. Ejemplos. Trabajo de Doctorado. Dept. Informática y Automática. Universidad Complutense de Madrid, 1994.

[OH83] E. R. Olderog and C.A.R. Hoare. Specification-oriented semantics for communicating processes. In *10th ICALP, LNCS 154*, pages 561–572, 1983.

[Par81] D. Park. Concurrency and automata on infinite sequences. In *5th G.I. Conference, LNCS 104*, pages 167–183, 1981.

[Paz71] A. Paz. *Introduction to Probabilistic Automata.* Academic Press, 1971.

[Phi87] I. Phillips. Refusal testing. *Theoretical Computer Science*, 50(3):241–284, 1987.

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

[PS88] K. Paliwoda and J.W. Sanders. The Sliding-Window protocol in CSP. Technical Report PRG-66, Oxford University Programming Research Group, 1988.

[PS89] J.L. Peterson and A. Silberschatz. *Sistemas Operativos. Conceptos fundamentales.* Editorial Reverté, 1989.

[PT87] R. Paige and R.E. Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, 16:973–989, 1987.

[Rab63] M.O. Rabin. Probabilistic automata. *Information and Control*, 6:230–245, 1963.

[RR86] G.M. Reed and A.W. Roscoe. A timed model for CSP. In *13th ICALP, LNCS 226*, pages 314–323, 1986.

[RR88] G.M. Reed and A.W. Roscoe. A timed model for communicating sequential processes. *Theoretical Computer Science*, 58:249–261, 1988.

[Sch89] Steve Schneider. *Correctness and Communication in Real-Time Systems.* PhD thesis, Oxford University, 1989.

[Sch95] S. Schneider. An operational semantics for timed CSP. *Information and Computation*, 116:193–213, 1995.

[Seg95a] R. Segala. A compositional trace-based semantics for probabilistic automata. In *CONCUR'95, LNCS 962*, pages 234–248, 1995.

[Seg95b] R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems.* PhD thesis, MIT, Dept. of Electrical Engineering and Computer Science, 1995.

[Sei92] K. Seidel. *Probabilistic Communicating Processes.* PhD thesis, Oxford University, 1992.

[Sei95] K. Seidel. Probabilistic communicating processes. *Theoretical Computer Science*, 152:219–249, 1995.

[SL94] R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. In *CONCUR'94, LNCS 836*, pages 481–496, 1994.

[SS90] S.A. Smolka and B. Steffen. Priority as extremal probability. In *CONCUR'90, LNCS 458*, pages 456–466, 1990.

[Tof90] C. Tofts. A synchronous calculus of relative frequency. In *CONCUR'90, LNCS 458*, pages 467–480, 1990.

[Tof94] C. Tofts. Processes with probabilities, priority and time. *Formal Aspects of Computing*, 6:536–564, 1994.

[Tze85] W. Tzeng. The equivalence and learning of probabilistic automata. In *26th IEEE Symposium on Foundations of Computer Science*, pages 268–273, 1985.

[Tze92] W. Tzeng. A polynomial-time algorithm for the equivalence of probabilistic automata. *SIAM Journal on Computing*, 21:216–227, 1992.

[Var85] M.Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *26th IEEE Symposium on Foundations of Computer Science*, pages 327–338, 1985.

[vGSS95] R. van Glabbeek, S.A. Smolka, and B. Steffen. Reactive, generative and stratified models of probabilistic processes. *Information and Computation*, 121:59–80, 1995.

[vGSST90] R. van Glabbeek, S.A. Smolka, B. Steffen, and C.M.N. Tofts. Reactive, generative, and stratified models of probabilistic processes. In *5th IEEE Symposium on Logic In Computer Science*, pages 130–141, 1990.

[Win87] G. Winskel. Event structures. In *Petri Nets: Applications and Relationships to Other Models of Concurrency, LNCS 255*, pages 325–392, 1987.

[Win89] G. Winskel. An introduction to event structures. In *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency, LNCS 354*, pages 364–397, 1989.

[WSS94] S.-H. Wu, S.A. Smolka, and E.W. Stark. Composition and behaviors of probabilistic I/O automata. In *CONCUR'94, LNCS 836*, pages 513–528, 1994.

[YCDS94] S. Yuen, R. Cleaveland, Z. Dayar, and S.A. Smolka. Fully abstract characterizations of testing preorders for probabilistic processes. In *CONCUR'94, LNCS 836*, pages 497–512, 1994.

[YL92] W. Yi and K.G. Larsen. Testing probabilistic and nondeterministic processes. In *Protocol Specification, Testing and Verification XII*, pages 47–61, 1992.