CSP probabilístico (PCSP)
Un modelo probabilístico de procesos concurrentes

Fernando Cuartero Gómez

Septiembre, 1993

TESIS DOCTORAL
PRESENTADA AL DPTO. DE INFORMATICA Y AUTOMATICA
DE LA UNIVERSIDAD COMPLUTENSE DE MADRID
PARA LA OBTENCION DEL GRADO DE DOCTOR EN
CIENCIAS MATEMATICAS

Trabajo dirigido por Dr. D. David de Frutos Escrig

a Maribel

Todo cuanto existe en el Universo es fruto del azar y de la necesidad.

Demócrito.

Agradecimientos

A la hora de recordar a todos los que han contribuido a que esta tesis vea finalmente la luz, es verdaderamente difícil tener unas palabras para todos ellos.

Cómo no recordar la labor de todos aquellos maestros y profesores que han contribuido a mi formación, pues la educación es, después de la propia existencia el bien más preciado, y uno de los frutos de la misma es la presente tesis.

De entre todos ellos, ciertamente debo destacar a David, quien, después de tres años en que, durante la licenciatura, me enseñó los misterios de la Informática, aún se atrevió a ser mi tutor en el Programa de Doctorado, e incluso a dirigir mi primer Trabajo de Investigación, con el que me inicié en el lenguaje CSP. No obstante lo cual, aún tuvo la moral de dirigir esta tesis. David, sinceramente, gracias.

También recordaré al resto de profesores y compañeros del Departamento de Informática y Automática de la Universidad Complutense, que tan buen trato y predisposición han mostrado siempre.

Mis compañeros (y sin embargo amigos) del Departamento de Informática de la Universidad de Castilla-La Mancha, en Albacete, también han aportado una dosis de esfuerzo, y sobre todo la inestimable ayuda moral de estar embarcados en esa aventura que es la de crear unos estudios, con suficiente calidad, a partir de nuestro trabajo, interés y poco más.

Entre ellos, cómo no destacar a Valentín, compañero de innumerables fatigas, penalidades y viajes interminables en carretera, y a pesar de ello, inmune al desaliento. Es, ciertamente, muy difícil no conseguir el objetivo que uno persigue teniendo compañeros como Valentín. Por otra parte, también ha sido de agradecer la gentileza del MOPT que, a fin de aliviar nuestra tarea, dispuso el trazado de la autovía Madrid-Albacete.

Y tantos otros que algún granito de arena han aportado, como mi buen amigo Paco, que tuvo el humor de aguantar unas exposiciones de mis problemas (enterándose de bien poco, por cierto), lo que me hizo volver a comprobar que una exposición metódica y cuidadosa de un problema es el primer paso que asegura la solución.

Finalizando, cómo no, con un grato recuerdo, reconocimiento y cariño para quién más ha tenido que sufrir con el desarrollo de la tesis, soportando con paciencia (casi siempre) los malos ratos, obsesiones, y la carga de un periodo prolongado de estudio y trabajo, gracias Maribel.

Contenidos

Capítulo 1

Introducción

El objeto último de la presente Tesis Doctoral es el estudio, a nivel formal, de los sistemas concurrentes. Es bien sabido que la aparición de la concurrencia representa un aumento radical en la complejidad de los sistemas. Ello se pone en particular de manifiesto en los modelos teóricos desarrollados para abordar su estudio a un nivel formal. En consecuencia, en los primeros modelos tales se abstraen todos los aspectos de los procesos concurrentes que se consideran no fundamentales de los mismos, de manera que se pueda razonar a nivel formal sobre el comportamiento de los procesos, sin que la complejidad del modelo formal en cuestión oscurezca y dificulte dichos razonamientos. Naturalmente, el coste que se ha de pagar debido a las susodichas simplificaciones, es la perdida de toda aquella información que voluntariamente se ha abstraído en el proceso de modelización.

Todo ello se observa en particular en los distintos modelos formales que inicialmente se desarrollaron, como son las *Redes de Petri* [Pet81, Rei85] y las denominadas *Algebras de Procesos*, como *CSP* [Hoa85], *CCS* [Mil80, Mil89] y *ACP* [BK84]. En todas las versiones básicas de estos modelos, se parte de una forma u otra de la idea de que un sistema concurrente evoluciona mediante la ejecución en cada momento por parte de una de sus componentes (o de todas aquellas que deban sincronizar al efecto para hacerlo posible) de una acción de entre las que en ese momento sea capaz de ejecutar. Ello liga inmediatamente la concurrencia con el no-determinismo, pues si en un momento son varias las acciones que podrían ejecutarse, no se toma partido por ninguna de ellas, sino que se considera

que todas ellas son igualmente posibles, si bien finalmente sólo uno de los posibles funcionamientos se producirá en cada ejecución del sistema.

De hecho, en la práctica se ha comprobado que, una vez abstraídos aspectos como la equidad (*fairness*) en las elecciones, el paralelismo real (posible uso de varios procesadores al tiempo) o la duración de las acciones, en la modelización de dicho no-determinismo se encuentra la clave de cara a contar con una buena modelización de la concurrencia. Un magnífico trabajo, en el que ello se pone de manifiesto, y en el que además se hace una exposición detallada de la denominada *semántica de pruebas*, previamente presentada en [NH84], es [Hen88].

Ahora bien, una vez estudiados en profundidad los modelos básicos, es oportuno el tratar de ampliarlos o modificarlos en lo preciso, de modo que puedan pasar a estudiarse aquellos aspectos, fundamentalmente cuantitativos, que se habían perdido como resultado de las abstracciones en que aquéllos se basaban. En concreto, se cuenta ya con abundantes resultados en temas como el *paralelismo real* [Tau89, Win87, Old87, DNM90], los modelos temporizados [RR87, MT90, RR88, Ort90] o la equidad [CS87, Fra86, RZ92].

Otro de dichos aspectos, en cierta forma relacionado con el último de los anteriormente citados, es precisamente el que constituye el centro de atención del presente trabajo. Se trata de la introducción en el modelo de informaciones probabilísticas que cuantifiquen las decisiones no-deterministas que aparecen en el seno de cada proceso. Es obvio al respecto, que en la práctica evaluaciones cuantitativas del comportamiento de un proceso pueden resultar tanto o más interesantes que evaluaciones meramente cualitativas. Así, la simple posibilidad de un mal comportamiento de un sistema puede resultar irrelevante si la probabilidad de que el mismo se presente es nula o despreciable. En concreto nuestra aproximación se centrará en el estudio del tema en el marco de las álgebras de procesos, y más específicamente en la definición y estudio de una extensión probabilística del modelo CSP.

Aunque por las características técnicas del mismo, consideramos que el presente trabajo es notablemente novedoso, es sin embargo bien cierto, que no se trata en absoluto del primer trabajo dedicado al estudio de las versiones probabilísticas de las álgebras de procesos, si bien no hace apenas cuatro años, más o menos

el momento en que comenzó a gestarse nuestro trabajo, que se publicaron los primeros trabajos de una cierta entidad dedicados al tema. Nos limitaremos aquí a referenciar los primeros de dichos trabajos, que junto con otros más recientes serán comentados brevemente en el transcurso de la próxima sección.

El primero de los trabajos relacionados con la cuestión es [LS89], donde se estudian una clase de sistemas de transición probabilísticos, definiendo para los mismos una extensión probabilística de la noción de **bisimulación** ([Mil80, Mil83]) en la que se basa la semántica usual del lenguaje CCS. El tema es posteriormente tratado de nuevo en [Chr90B, Chr89].

Ya en el marco de las álgebras de procesos el primer trabajo ha sido [GJS90], en donde se describe un lenguaje denominado PCCS, versión probabilística de SCCS [Mil83], que es a su vez una versión síncrona del lenguaje CCS. En dicho trabajo se describe la sintaxis del lenguaje, su semántica operacional y unas leyes algebraicas, todo lo cual queda ilustrado por un ejemplo de cierta envergadura. Posteriormente, en [GSST90] se continúa con el tema, presentándose tres semánticas operacionales del lenguaje, sobre cada una de las cuales se define la correspondiente noción de bisimulación. Dichas semánticas reciben los nombres de *reactiva*, *generativa* y *estratificada*. Por último, en [JS90, SS90] se analizan algunos aspectos adicionales del lenguaje: en el primero se estudian distintas nociones de equivalencia de procesos, mientras que en el segundo se estudia la noción de *prioridad*.

1.1 Antecedentes y estado actual del tema

El no-determinismo como elemento fundamental en la modelización de la concurrencia, tomó auge en la segunda mitad de la década de los setenta, aunque los primeros trabajos se remonten a períodos anteriores. Podemos citar los trabajos de Milner [Mil80] y Hoare [Hoa78] como elementos fundamentales que sentaron las bases de lo que ha sido, en los últimos años, uno de los campos más fructíferos de la Informática Teórica. Sendas versiones revisadas y actualizadas de dichos trabajos [Hoa85, Mil89], se han convertido ya en clásicos de la literatura especializada.

Aparte de sus interesantes aspectos teóricos, alrededor de dichos modelos se han

desarrollado también múltiples aplicaciones prácticas, que constituyen de hecho la justificación última de su existencia. En particular, en el terreno de los lenguajes de programación se cuenta con el lenguaje de programación concurrente OCCAM, basado en CSP, mientras que LOTOS es un lenguaje de especificación, adoptado como estándar ISO para la especificación de protocolos, basado fundamentalmente en CCS.

Entre los trabajos más importantes en los que se han puesto las bases teóricas para el estudio de las álgebras de procesos, hemos de citar los de Brookes [Bro83], que desarrolló el modelo semántico de procesos en el que se basa la semántica denotacional de CSP, y estudió la semántica axiomática del lenguaje, comparándola con la anterior. Dicho modelo fue mejorado posteriormente en [BR85], donde se incorporó el fenómeno de *divergencia*. Por otra parte, citaremos de nuevo [Hen88], donde se estudia en profundidad la semántica de pruebas, y se relaciona con la semántica denotacional y la semántica axiomática.

Y como quiera que, como ya hemos comentado, el campo de los modelos formales de la concurrencia, y en particular el de las álgebras de procesos, ha resultado tremendamente fructífero en los últimos años, resultaría tremendamente costoso tanto en tiempo como en espacio, hacer un recorrido exhaustivo sobre sus diferentes facetas, por lo que nos centraremos en lo sucesivo en el terreno concreto de las versiones probabilísticas de las álgebras de proceso.

Las álgebras de procesos probabilísticos estudiadas hasta la fecha tienen en común la inclusión de, al menos, un operador de elección y otro de composición paralela; sin embargo las mismas nos presentan distintas variantes a la hora de extender probabilísticamente dichas operaciones. Las diferencias fundamentales aparecen en el tipo (o tipos) de elección considerados y en el hecho de si el paralelismo es o no sincronizado. Por otra parte, existen también diferencias en el tipo de equivalencias entre procesos que se consideran, en función de los distintos marcos que se utilizan para definir la semántica de los procesos probabilísticos. En lo sucesivo centraremos la presentación de los modelos que hemos considerado más representativos, fundamentalmente en el comentario de como se concretan en cada uno de ellos dichas variantes.

1.1.1 Sistemas basados en CCS

Comenzaremos nuestro estudio describiendo un par de aproximaciones que representan versiones probabilísticas de CCS. Entre ellos, tenemos en primer lugar PCCS, un lenguaje que fue descrito inicialmente por Giacalone y otros en [GJS90], donde se presentó un modelo operacional del mismo. Como ya indicamos, la base del lenguaje es la versión síncrona de CCS, SCCS [Mil83]. La sintaxis del lenguaje es la siguiente:

$$E ::= 0 \mid X_i \mid \alpha : E \mid \sum_{i \in I} [p_i] E_I \quad \text{Donde } p_i \in (0,1], \sum_{i \in I} p_i = 1 \mid$$

$$E \times F \mid E \uparrow A \mid E[f] \mid fix_i \tilde{X} \tilde{E}$$

El hecho de estar basado en un lenguaje síncrono simplifica notablemente las notaciones y la definición de la semántica del lenguaje; sin embargo ello se consigue a costa de sacrificar potencia descriptiva, pues aquellos sistemas que son asíncronos por naturaleza han de ser sincronizados de una manera artificial para poder ser modelados con el lenguaje. Tal artificiosidad queda por ejemplo de manifiesto, en el mismo ejemplo presentado en dicho artículo, donde se representa una versión del protocolo AUY.

A partir de la semántica operacional definida, se introduce una noción de bisimulación que induce la correspondiente noción de equivalencia entre procesos. A continuación se presentan una serie de leyes algebraicas que se cumplen en el modelo semántico inducido por dicha noción de equivalencia.

El siguiente paso consiste en introducir las nociones básicas que permiten definir una noción de distancia entre los procesos semánticos, la cual dota al conjunto formado por los mismos de la estructura de espacio métrico. Dicha distancia se define a partir de una cuantificación de la noción de bisimulación, denominada *ϵ-bisimulación*, que permite una desviación acotada por ϵ en la simulación de cada paso de los procesos comparados. Más exactamente, la distancia entre dos procesos vendrá dada por el menor valor ϵ, para el cual ambos son ϵ-bisimilares. Así, dos procesos bisimilares se encuentran a distancia 0, y por lo tanto son de hecho uno mismo, de lo cual se sigue que el universo inducido de procesos semánticos coincide con el correspondiente bajo la bisimulación. La definición de este espacio métrico

(el cual de hecho sólo se define sobre el subconjunto de procesos *deterministas*) parece ser el punto de partida para un trabajo más exhaustivo, que se anuncia de hecho en las conclusiones del artículo en cuestión, pero que no nos consta que haya sido divulgado hasta la fecha.

Por el contrario, la semántica operacional del modelo si es estudiada con más detalle por Van Glabbeck y otros, en un trabajo posterior [GSST90], donde se proponen tres posibles versiones de la semántica denominadas *Reactiva, Generativa* y *Estratificada*, de entre las cuales la segunda coincide con la definida en el primer trabajo.

El modelo Reactivo, basado en el sugerido en [Pnu85], entiende que los procesos reaccionan frente al estímulo presentado por un agente exterior, el cual solicita la ejecución de una determinada acción (de forma similar a lo que en [Mil80] representan la *presión de botones*, aunque sin permitir la presión simultánea de varios de ellos), de manera que las probabilidades sólo se corresponden con las distintas formas que se tengan de ejecutar cada acción posible, cada una de las cuales, cuando sea seleccionada por el exterior, será ejecutada con seguridad. Así, en el proceso

$$[\frac{1}{6}]a : E_1 + [\frac{3}{6}]a : E_2 + [\frac{1}{6}]b : E_3$$

se tiene que si la acción seleccionada por el entorno es a, la misma será ejecutada con seguridad, pasando el sistema a comportarse como E_1 con probabilidad $1/4$ y como E_2 con probabilidad $3/4$; mientras que si la acción ejecutada es la b, el sistema pasará a comportarse como E_3 con probabilidad 1.

Por el contrario, en el modelo generativo se entiende que el sistema es autosuficiente para tomar todas las decisiones, sin responder a ningún requerimiento del exterior. De esta forma las probabilidades indicadas serán aplicadas en primer lugar para decidir la acción a ejecutar, y luego la correspondiente continuación.

Finalmente, el modelo estratificado pretende ser una combinación de ambos, pero a la postre es más bien una versión del generativo, en la que las probabilidades vienen jerarquizadas en la forma en que se indica en la sintaxis del proceso, de manera que ante la imposibilidad de ejecutar una acción, su probabilidad no se reparte entre todas las demás, sino sólo entre las que están a su mismo nivel. Así,

en el ejemplo

$$[\tfrac{1}{3}]a : 0 + [\tfrac{2}{3}]([\tfrac{1}{2}]b : 0 + [\tfrac{1}{2}]c : 0)$$

si la acción c no puede ser ejecutada por el entorno en que se encuentra el proceso, la acción a continuará con probabilidad $1/3$, mientras que en el modelo generativo al ser las probabilidades de las tres acciones iguales, si una de ellas no puede intervenir en el sorteo, su probabilidad se reparte entre todas las demás acciones, con lo que la probabilidad de ejecutarlas continuaría siendo la misma para ambas, siendo por tanto igual a $1/2$.

Estas tres versiones son analizadas con cierto detalle, proponiéndose para cada una de ellas una semántica operacional y la correspondiente noción de bisimulación asociada. Todas ellas son relativamente sencillas, lo cual se debe al hecho de que siempre se mantienen separadas las características de los modelos reactivo y generativo. Volveremos al tema en la sección siguiente, en la que presentamos nuestro propio trabajo, pues el deseo de fundir en un único modelo las características diferenciadas de los modelos generativo y reactivo, motivado por la coexistencia en el lenguaje CSP de sus operadores de elección externa e interna es el que justifica el mismo, así como, por desgracia, es el principal culpable de su bastante mayor complejidad.

Jou y Smolka comparan en [JS90] una serie de nociones de equivalencia entre procesos para el caso generativo, presentando además una axiomatización correcta y completa de la equivalencia entre procesos finitos inducida por la bisimulación.

Por otra parte, Hansson y Jonsson ([HJ90, Han91, Han92]) introducen otro lenguaje basado en CCS en el que incorporan cuantificación de tiempo y probabilidad, al que denominan TPCCS. La base probabilística que utilizan tiene naturaleza generativa, pues es el sistema quien decide sobre la ejecución de las acciones. Sin embargo, junto a la elección probabilística, incorporan una elección *no determinista*, si bien la sintaxis del lenguaje impone que ambos tipos de elección deban alternarse de forma estricta.

Como veremos en su momento, este modelo de procesos, que en este caso se impone como punto de partida, se parece en gran medida a aquél al que nosotros hemos llegado como resultado final, en la búsqueda de las formas normales que

caracterizan la equivalencia inducida por las semánticas que hemos desarrollado. Ello es bien probable que no deba achacarse a la mera casualidad, pero no obstante lo que sí es cierto es que los autores de dichos trabajos, en momento alguno han justificado la razón por la que estudian en concreto dicha clase sintáctica de procesos.

Por otra parte, queremos dejar aquí constancia de lo que nos parece un error importante en el primero de dichos artículos [HJ90]. Se trata del hecho de que a pesar de que en principio la sintaxis impone que las acciones entre las que se elige en el caso de las elecciones no-deterministas sean distintas, sin embargo está también permitido que en dicho contexto aparezca una composición paralela, la cual puede dar lugar a elecciones entre acciones idénticas. Ello sucede en particular en el caso de las acciones internas generadas por las sincronizaciones. Esta cuestión es parcialmente corregida en la siguiente versión del trabajo ([Han91]) en la que las acciones internas aparecen etiquetadas con la acción original cuya sincronización las produjo, manteniéndose en este caso una decisión externa (y por tanto no cuantificable probabilísticamente) respecto a qué acción se ejecutará. No obstante, en ambos trabajos se queda sin aclarar el comportamiento de un sistema que debe decidir entre dos procesos que comienzan con la misma acción.

1.1.2 Sistemas basados en CSP

J. J. Zic, en su artículo [Zic90] expone la posibilidad de extender CSP en términos de probabilidad y tiempo. Sin embargo, se limita a justificar la necesidad de la extensión y a dar una orientación de la ruta a seguir para hacerlo, sin entrar en absoluto en los detalles, ni por tanto en las dificultades, de la tarea.

Ofrece en dicho trabajo la idea de plantear un operador de elección externa probabilístico, para el cual las probabilidades dependerían de un *entorno exterior* variable, en función del cual se obtendrían los distintos comportamientos del operador. Sin embargo, arguye como principal dificultad para la realización de la tarea, el hecho de que la asignación de una distribución de probabilidad que gobierne las decisiones que toma un proceso, requeriría un conocimiento del entorno del mismo. En cierta forma nosotros hemos seguido esa idea al definir la semántica

de pruebas de nuestro lenguaje, en la que los tests juegan el papel de medio exterior que interacciona con el proceso; pero como quiera que dicha semántica se define considerando todos los tests posibles, se tiene que a la postre no se hace ninguna suposición sobre el medio externo concreto subyacente.

Por último, Karen Seidel presenta en [Sei92] una versión probabilística de CSP con una sintaxis similar a la por nosotros considerada. La sintaxis de su lenguaje, también denominado PCSP, es la siguiente:

$$P ::= STOP \mid SKIP \mid X \mid a \rightarrow P \mid P \;{}_p\sqcap\; P \mid P \;{}_p\Box\; P \mid P\backslash B \mid f(P) \mid$$

$$P \parallel P \mid P \,;\, Q \mid P \;{}_p\; ||| \; P \mid \mu X.P \mid \langle X_i = P_i \rangle$$

Como se verá en su momento, la notación $P \;{}_p\sqcap\; P$ aquí utilizada, es similar a la nuestra $[p_1]P \sqcap [p_2]P$, siendo también similar el significado que se da al operador de elección interna que representan. Respecto al operador de elección externa, también es similar la notación que se emplea, pero desconocemos el significado que se pretende que la misma tenga, puesto que en la versión incompleta del trabajo de que disponemos (gentilmente enviada por su autora) no llega a discutirse dicho operador. Lo mismo podemos decir respecto al operador de ocultación, aunque en esta ocasión la sintaxis difiere, pues no se cuantifica probablísticamente la operación, por lo que desconocemos cómo se tratará la ejecución de las acciones internas que genera.

Con respecto al operador de composición paralela, al ser totalmente sincronizado, es elemental su significado y coincide con el nuestro en el que caso de que el conjunto de sincronización es el total de acciones del alfabeto. No ocurre lo mismo en el caso de interleaving, en el que las probabilidades son *internas*. Discrepamos con esta interpretación, puesto que sería tanto como afirmar que en dos terminales conectadas a un computador central, sería el sistema el que debería decidir qué terminal será atendida. Ello parece razonable en el caso de que ambos realicen una petición, pero si uno está inactivo, por ejemplo porque su operador está tomando un café, entendemos que el otro no debería esperar también por el hecho de que el sistema haya decidido atender al primero en primer lugar.

Por otra parte, la forma de estudiar los procesos difiere notablemente de la

utilizada en nuestro trabajo, pues se dedica fundamentalmente a estudiar las distribuciones de probabilidades que se generan en el marco de la Teoría de Probabilidades, obteniendo a partir de dicho estudio una serie de leyes que obedecen los procesos.

1.1.3 Otros modelos

La primera referencia de la que tenemos constancia, en la que se discute un modelo algebraico de procesos concurrentes en el que se incorporan aspectos cuantitativos probabilísticos, es [LS89], donde la aproximación a los procesos concurrentes se hace en base a las transiciones operacionales que generan, al margen de sintaxis concreta alguna para los mismos.

En dicho trabajo, una vez definido el sistema de transiciones probabilístico como modelo semántico de los procesos probabilísticos, se define un sistema de testeo de los mismos, mediante el cual se pueden obtener propiedades de los procesos concurrentes analizando su respuesta frente a los tests. La semántica obtenida se compara con la inducida por una noción de bisimulación probabilística, basada en la bisimulación fuerte. Se muestra que, por medio de dichos tests, es posible distinguir con una probabilidad tan cercana a uno como se desee, entre procesos que no sean equivalentes por bisimulación, lo que prueba que, en este sentido probabilístico, el mecanismo de testeo sí captura la bisimulación, a diferencia de lo que sucede en el marco del no-determinismo no cuantificado probabilísticamente (véase [HM85]).

Además, las propiedades obtenibles por testeo pueden ser descritas por medio de una *Lógica Modal Probabilística*, similar a la de Hennessy y Milner [HM85], de forma que se obtiene un entorno que permite validar un proceso con respecto a su especificación, siempre que ésta venga dada por una serie de propiedades descritas con dicho lenguaje.

Bloom y Meyer ([BM89]) estudian la relación existente entre la bisimulación original ([Mil89]) y la probabilística, mostrando que dos procesos (no probabilísticos) P y Q son bisimilares si y sólo si existe una asignación de probabilidades

a las transiciones de ambos, tales que los procesos resultantes P' y Q' son probabilísticamente bisimilares.

Por otra parte, Jones y Plotkin en [JP90] proporcionan un entorno general para definir la semántica de lenguajes de programación probabilísticos, en el que se estudian constructores tales como la composición secuencial, las instrucciones condicionales, y los bucles, junto a una elección probabilística interna y a una composición paralela con una decisión probabilística de qué proceso debe ser el que ejecute la primera acción, no existiendo en cambio constructores que permitan una elección de tipo externo como el de CSP.

1.2 Resumen y Plan del trabajo

En este trabajo nos hemos marcado como objetivo el estudio de un lenguaje apropiado para la especificación de procesos concurrentes que, junto con las facetas que quedan capturadas por los modelos clásicos desarrollados al efecto, nos permitiese además el modelado de propiedades cuantitativas probabilísticas de los procesos.

En concreto hemos tomado como base el modelo CSP definido por C.A.R. Hoare en [Hoa85], al cual hemos añadido una serie de mecanismos probabilísticos que permiten indicar la distribución de probabilidades que un sistema utiliza para tomar sus decisiones. Para dicho lenguaje hemos buscado una semántica apropiada, que al mismo tiempo que nos permitiese capturar la información probabilística añadida, mantuviese en la mayor medida posible la corrección de las leyes que se tienen en el lenguaje CSP original.

Como hemos visto en la sección anterior, los modelos hasta ahora estudiados, se basan bien en las ideas *reactivas* en las cuales las elecciones se entienden como externas, de manera que si en un momento dado existe la posibilidad de ejecutar una acción y el entorno desea hacerlo, ello sucederá con seguridad, o en las *generativas*, en las que se considera que el sistema es completamente autónomo, y por tanto toma todas sus decisiones de forma interna. Pero en CSP coexisten ambos tipos de elección, y para ambos deseamos conservar sus características propias. Ello nos lleva a interpretar la elección externa de forma *reactiva*, y sin embargo la

interna de modo *generativo*. Curiosamente, si bien las definiciones de ambos tipos de modelos han probado ser relativamente sencillas, la mezcla adecuada de ambos resulta ser mucho más complicada, fundamentalmente por la necesidad de tener que renunciar a los modelos clásicos de espacios de probabilidad a la hora de definir la semántica del lenguaje, pues ni es cierto que la suma total de las probabilidades de todos los cómputos ha de ser 1 como sucede en el caso generativo, ni la suma de las probabilidades de ejecutar cada traza posible ha de ser igualmente igual a la unidad, como sucede en el caso reactivo.

Con estas ideas en mente, intentamos una primera aproximación, basada en las técnicas operacionales descritas en los primeros trabajos aparecidos sobre procesos probabilísticos, en concreto PCCS. Sin embargo, con estas técnicas llegamos a la conclusión de que no es posible mantener el comportamiento del operador de ocultación de CSP, que cuando actúa en un contexto de elección externa, introduce por un lado una componente de elección interna, pero mantiene al mismo tiempo un cierto grado de elección externa, como expresa la ley clásica de CSP

$$(a \rightarrow P \,\Box\, b \rightarrow Q)\backslash a \;\equiv\; P\backslash a \;\sqcap\; (P\backslash a \,\Box\, b \rightarrow Q\backslash a)$$

Puesto que pretendíamos el mantenimiento de una versión adecuada de esta ley en nuestro lenguaje PCSP, o lo que es lo mismo, deseábamos mantener diferenciadas las características de los dos operadores de elección de CSP, llegamos a la conclusión de que era necesario distinguir en cada momento la capacidad de elección de que dispone el entorno exterior para seleccionar la acción a ejecutar en cada instante del desarrollo del proceso.

Pensamos que el marco más natural para poner de manifiesto la capacidad de influencia que en cada momento tiene el medio exterior en el funcionamiento de un proceso, era el de la semántica de pruebas [Hen85]. Como quiera que era preciso decidir la acción a ejecutar en los casos en que el proceso está dispuesto a ejecutar una serie de acciones, y el entorno a su vez permite que se opte entre varias de ellas, introdujimos probabilidades en los tests, cuyo único objetivo es resolver dichas elecciones. De esta forma, las decisiones internas del sistema, cuantificadas en la sintaxis del mismo, junto con las decisiones del medio exterior, cuantificadas en la distribución de probabilidad del test, dan lugar a una única distribución que

nos define la probabilidad de aceptación de un test por parte de un proceso, que denotaremos por $P \mid t$.

La idea que permite visualizar la manera en que se define dicha probabilidad, consiste en imaginar que en cada paso el sistema resuelve en primer lugar todas las elecciones internas que encuentra antes de la ejecución de acción visible alguna, entre las que se encuentran tanto aquellas indicadas explícitamente por el correspondiente operador, como las generadas implícitamente por los operadores de elección externa, paralelismo y ocultación. Tras la resolución de dichas elecciones internas, el proceso alcanza un determinado *estado*, que viene definido por el conjunto de acciones que el sistema queda en disposición de aceptar. Alcanzado el estado, corresponde al entorno representado por el test, decidir la acción a ejecutar. Ello da lugar a una nueva elección que se resolverá en base a la distribución de probabilidad indicada en el test, si bien una vez normalizada la misma, al excluirse del test aquellas acciones que no estén permitidas en el estado alcanzado.

Une vez que el estudio por medio de los tests nos conduce a desentrañar los secretos de la evolución de un proceso probabilístico, surge de manera natural la idea en que basar la definición de un dominio de procesos que nos sirva para definir una semántica denotacional de los mismos, equivalente a la de pruebas previamente definida. Obtenemos en concreto, que un proceso probabilístico vendrá dado por un árbol de decisiones, que alternativamente serán de dos tipos: en primer lugar decisiones internas que corresponderán a nodos cuyas ramas hijas están etiquetadas por subconjuntos disjuntos del alfabeto de acciones de los procesos. Cada una de dichas ramas tendrá también asociada una probabilidad, de forma que la suma de todas las probabilidades de esas ramas sea a lo sumo 1, representando la diferencia hasta 1 la probabilidad con que el proceso diverge en ese punto, al quedar sumergido en una serie de elecciones internas indefinidas. Por otra parte, cada uno de dichos arcos finaliza en un nodo externo, del que a su vez pende una rama distinta para cada acción del conjunto que etiqueta a la rama con la que se accede al nodo externo en cuestión, la cual irá etiquetada por la correspondiente acción, siendo su destino un nuevo nodo interno.

Una vez definida en base a dicho dominio la semántica denotacional, comprobamos que la misma es equivalente a la semántica de pruebas definida previamente.

Es decir, si dos procesos pasan con la misma probabilidad cada test que les es ofrecido a ambos, entonces serán equivalentes denotacionalmente, y viceversa.

Seguidamente nos planteamos la construcción de un sistema de axiomas y reglas que nos permita razonar en un marco algebraico sobre la equivalencia semántica entre procesos sintácticos. De cara a probar la completitud de dicho sistema, utilizamos el hecho de que todo proceso PCSP puede reescribirse utilizando dicho sistema lógico, en una *forma normal* de la forma

$$P = \prod_{A \subseteq \Sigma} [p_A] \Box_{a \in A} \left[\frac{1}{|A|}\right] a \rightarrow P_{a,A}$$

donde $\sum_{A \subseteq \Sigma} p_A \leq 1$, y $P_{a,A}$ está también en forma normal.

Podemos observar que, curiosamente, las formas normales son extraordinariamente simples comparadas con las correspondientes al caso ordinario, pues las condiciones de cierre bajo convexidad y unión que en aquel caso aparecían, dejan de ser necesarias, y tampoco las distintas continuaciones tras la ejecución de una misma acción han de ser necesariamente idénticas. Ello se debe al hecho de que en el presente caso probabilístico, los tests son capaces de detectar todos los posibles estados de un proceso, y de aislar la continuación correspondiente a cada uno de ellos.

Por último definimos una semántica operacional equivalente a las anteriores, para lo cual la misma ha de tener una conexión muy fuerte con la semántica denotacional previamente definida, en el sentido de que las transiciones de la misma han de ir etiquetadas además de por la acción ejecutada a la que corresponden, por un conjunto de acciones que representa la *oferta* del proceso en el estado en el que se encuentra. Ambos datos irán acompañados por la probabilidad correspondiente, que representa la probabilidad de alcanzar dicho estado, y por tanto es independiente de la acción ejecutada a la postre.

Para terminar, se prueba que la semántica operacional definida es equivalente a las anteriores, lo cual no ofrece excesiva dificultad, dado el parecido ya reseñado entre la misma y la denotacional previamente definida.

En conclusión, hemos obtenido cuatro modelos semánticos de PCSP, todos

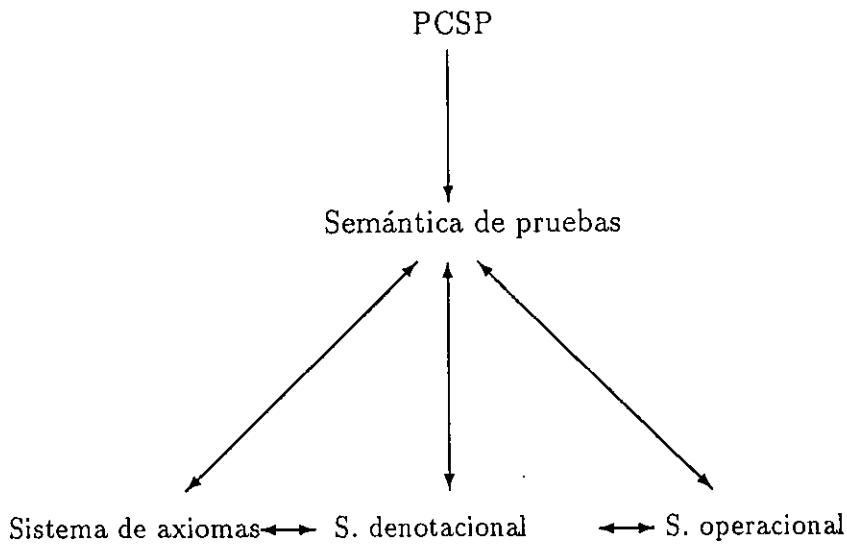


Figura 1: Resultados obtenidos

ellos equivalentes, que nos sirven para razonar a distintos niveles sobre el comportamiento de los procesos probabilísticos. Ello queda expresado de forma gráfica en la figura 1.

En lo que respecta al contenido de los siguientes capítulos del trabajo, indicaremos que comenzaremos en el capítulo 2, describiendo de una manera rápida las técnicas utilizadas a lo largo del trabajo, en particular la semántica de los lenguajes, y la aplicación de las mismas al lenguaje CSP clásico.

En el siguiente capítulo introduciremos el lenguaje PCSP, presentando la que fue nuestra primera aproximación a la definición de la semántica del mismo. Como ya hemos indicado, se trata de una semántica operacional, la cual se corresponde de forma natural con una semántica denotacional de trazas. El capítulo concluye con un ejemplo no trivial que ilustra la aplicabilidad del lenguaje. Una versión previa y resumida del contenido de este capítulo, puede encontrarse en [CFV92].

Desgraciadamente, un modelo tan simple como el presentado, nunca podrá capturar el significado que deseamos para el operador de ocultación, que no es otro que el que entendemos más coherente con la interpretación clásica en CSP del mismo operador. Dicha imposibilidad se mostrará en particular para la semántica

en principio propuesta, por medio de un contraejemplo adecuado.

En el capítulo 4 presentamos la semántica de pruebas que nos permite solventar las dificultades anteriores. Siguiendo el proceso habitual en estos casos, comenzamos definiendo el concepto de test (probabilístico en este caso), para a continuación definir la probabilidad con la que cada proceso pasa un test, por inducción estructural respecto de su sintaxis. Ello queda lejos de ser trivial en lo que se refiere al estudio de varios operadores, especialmente los de elección externa y ocultación, pues como ya hemos indicado, el punto de partida para la definición es la noción de estado, la cual no queda de manifiesto de forma explícita en absoluto, en la sintaxis de los procesos.

En el capítulo 5, estudiamos una semántica denotacional, comenzando por definir el dominio de procesos semánticos, para luego definir las correspondientes versiones semánticos de cada operador del lenguaje. Por último probamos la equivalencia entre esta semántica denotacional y la de pruebas, la cual se basa en la posibilidad de computar en base a los tests la probabilidad de alcanzar cada estado, junto con la de aislar la continuación de cada proceso tras la ejecución de una acción en un determinado estado.

En el capítulo 6 introducimos el sistema de prueba que nos servirá para razonar algebraicamente sobre la equivalencia entre procesos. Probaremos que el mismo es correcto y completo respecto a la semántica de testeo, en base a la noción adecuada de forma normal previamente comentada.

En el siguiente capítulo construimos una nueva semántica operacional inspirada en las ideas que han ido surgiendo a lo largo de los anteriores capítulos, gracias a lo cual nos es posible probar la equivalencia entre la misma y las presentadas con anterioridad.

Finalmente, en el capítulo 8 exponemos las conclusiones obtenidas con esta Tesis Doctoral, e indicamos las líneas a seguir para extender el mismo en un futuro.

Capítulo 2

Generalidades

Dedicaremos este capítulo a la presentación de los conceptos fundamentales en relación con el tema de las semánticas de lenguajes formales, concretando en especial la exposición en lo referente a su aplicación al modelo CSP. Ello nos servirá de punto de partida de cara a abordar el estudio de la versión probabilística de dicho modelo que constituye el tema central del presente trabajo.

2.1 Sintaxis de lenguajes

Definición 2.1 (Signatura)
Dada una colección de nombres de tipos $D_1, \ldots, D_n$, una signatura $\Sigma = (Op, \rho)$, sobre la misma viene dada por un conjunto Op de nombres de funciones, a cada uno de los cuales, f se le asocia un dominio, que es una tupla (eventualmente vacía) de nombres de tipos, y un rango, que es también un nombre de tipo. Ello lo representaremos en la forma

$$f : D_{i_1} \times \ldots \times D_{i_m} \longrightarrow D_j$$

Por su parte ρ es una función

$$\rho : Op \longrightarrow \mathbb{N}$$

que define la aridad de cada operación, con la restricción de que el dominio de cada operación tiene tantos argumentos como indica su aridad.

A las funciones de Op se les llama *operadores* de la signatura, y en el caso particular de que su aridad sea nula, diremos que se trata de *constantes.* □

Definición 2.2 (Σ-Algebra)
Dada una signatura Σ, una Σ-Algebra es un par (A, Σ_A) donde

- A es una colección de conjuntos, uno por cada nombre de tipo. Al conjunto A_i asociado al dominio D_i le denominaremos *soporte* del mismo.
- Σ_A es un conjunto de funciones definidas sobre los conjuntos soportes de la Σ-álgebra,

  $$\{ f_A / f \in Op \}$$

  cada una de las cuales va asociada a una operación de la signatura Σ, siendo su dominio de aplicación, rango y aridad compatibles con el perfil definido para la misma en la signatura. Es decir, si

  $$f : D_{i_1} \times \ldots \times D_{i_m} \longrightarrow D_j$$

  entonces

  $$f_A : A_{i_1} \times \ldots \times A_{i_m} \longrightarrow A_j$$

□

Así pues, una Σ-Algebra es simplemente una interpretación de los símbolos que aparecen en la signatura Σ, consistente en un conjunto por cada nombre y una interpretación de los símbolos de operaciones como funciones entre dichos conjuntos. Naturalmente, una misma signatura puede tener multitud de interpretaciones distintas, incluso aun cuando los conjuntos soportes sean los mismos. Una de ellas, de particular importancia es la conocida como *Algebra Libre* o *Algebra de Términos*, cuyos conjuntos soportes están constituídos por objetos puramente formales, que son los denominados *términos* de la signatura, los cuales son secuencias de símbolos de la signatura Σ, bien construídos en atención a los dominios y rangos de sus operaciones. La interpretación de las funciones de la signatura se define simplemente anteponiendo a los argumentos de la aplicación de una función, el símbolo de función correspondiente a la misma.

En lo sucesivo nos centraremos en el estudio de las signaturas *homogéneas*, que son aquéllas que tienen un sólo tipo. Ello nos es posible, pues si bien cuando se estudian lenguajes en general se precisan álgebras heterogéneas, en el caso concreto de las álgebras de procesos es suficiente contar con álgebras homogéneas.

Definición 2.3 (Términos)
El conjunto de términos bien formados correspondiente a la signatura Σ, que denotaremos por T_Σ, es el menor conjunto de cadenas de símbolos de operación de la misma que satisfacen

1. Si $f \in \Sigma$ tiene aridad 0, entonces la cadena formada por el símbolo f está en T_Σ.
2. Si $f \in \Sigma$ tiene aridad $k > 0$, entonces la cadena formada por los símbolos $f(t_1, \ldots, t_k)$ está en T_Σ, siempre que $t_1, \ldots, t_k$ sean términos de T_Σ.

□

Definición 2.4 (Algebra Libre)
El álgebra libre sobre la signatura homogénea Σ, es aquélla que tiene como soporte el conjunto de términos de la misma, interpretándose las funciones por medio de la simple aplicación sintáctica de las mismas, esto es, si $f \in \Sigma$ tiene aridad $k > 0$, entonces la función

$$f_{T_\Sigma} : T_\Sigma^k \longrightarrow T_\Sigma$$

aplica cada tupla de términos $\langle t_1, \ldots, t_k \rangle$ en el término $f(t_1, \ldots, t_k)$. En el caso de las constantes omitiremos los paréntesis correspondientes a los, en tal caso, inexistentes argumentos, de manera que la constante f vendrá representada por el mismo término f. □

De cara a definir los procesos a nivel sintáctico, utilizaremos una signatura homogénea adecuada, sobre el tipo de procesos *Proc*, representando los términos del álgebra libre correspondiente, los procesos finitos sintácticos.

De cara a la incorporación de los procesos infinitos, introducimos una serie de elementos auxiliares. Contaremos en primer lugar, con un conjunto infinito

contable de identificadores Idf, que se suelen denominar *variables* de proceso, si bien a nivel formal trataremos los mismos como nuevas constantes distinguidas. Por otra parte, introducimos una familia de operadores unarios, cada uno de los cuales viene asociado a una variable $\xi \in Idf$, y se representa con la notación $\mu\ \xi.P$.

Los términos correspondientes a la signatura así ampliada, serán en los sucesivo los denominados *procesos sintácticos.*

De esta definición se desprende que los identificadores son en sí mismos términos de la nueva álgebra, y por tanto procesos, que se verán como variables. Dicha interpretación se mantiene hasta que los mismos ocurren en el contexto de un término recursivo asociado a ese mismo identificador.

Para formalizar esta situación, introducimos las nociones de ocurrencias de identificadores *libres* y *ligadas*, inducidas por la construcción $\mu\xi.P$.

Definición 2.5 Sea P un término y ξ una ocurrencia de un identificador en P. Decimos entonces que:

- ξ ocurre **libre** en P sii nos encontramos en alguno de los siguientes casos:

$$\begin{array}{lll} P & = \xi & \\ P & = op(P_1, \ldots, P_n) & \text{donde } \xi \text{ ocurre libre en algún } P_i \\ P & = \mu\delta.P' & \text{con } \delta \neq \xi \text{ y } \xi \text{ ocurriendo libre en } P' \end{array}$$

  Por tanto, una aparición de ξ es libre si no está en el marco de una recursión correspondiente a la variable en cuestión.

- Una aparición de ξ es **ligada** si no es libre.

- El término P es **cerrado** si no tiene apariciones libres de identificadores.

□

Por ejemplo, en el término $\mu\ \xi.(\xi \ \square\ a \longrightarrow STOP)$, el identificador ξ ocurre en el contexto de μ, por lo que su ocurrencia no es libre; se trata por lo tanto de un proceso cerrado.

Como quiera que al definir la semántica, cada operador de recursión acabará con la *variabilidad* de los identificadores que liga, tendremos que los procesos cerrados tendrán un significado concreto, aún a pesar de que contengan variables. Es en tales procesos en quienes estamos en último término interesados, si bien la técnica de análisis estructural que utilizamos a la hora de analizar los procesos, nos obliga también a estudiar los procesos que contengan variables libres. Al respecto introducimos la siguiente

Definición 2.6 (Sustituciones sintácticas)
Sean P y Q dos términos, el término $P[Q/\xi]$ es el que se obtiene sustituyendo toda aparición libre de ξ en el término P por el término Q. En el caso de que la variable sustituída se sobreentienda por el contexto, escribiremos simplemente $P[Q]$ para expresar el resultado de la correspondiente sustitución. Por último, generalizamos dicha notación para cubrir el caso en el que realizamos reiteradamente una misma sustitución, definiedo al respecto

$$P^0[Q] = Q,\ \ldots,\ P^{n+1}[Q] = P[P^n[Q]]$$

□

2.2 Semántica de lenguajes

Como ya hemos indicado anteriormente, una misma sintaxis admite multitud de interpretaciones, definidas por medio de las correspondientes álgebras asociadas a la misma. Ya hemos destacado entre ellas el álgebra de términos, que de manera informal podríamos describir como aquella interpretación que en realidad no interpreta nada. Por esa razón, el álgebra de términos tiene por lo general más elementos de los que sería intutivamente deseable, en el sentido de que habitualmente tendremos distintos modos de describir sintácticamente un mismo proceso semántico. En consecuencia se precisan mecanismos más complejos para definir formalmente lo que conocemos como semántica de un lenguaje, que en último término no será otra cosa que un mecanismo para distinguir exactamente aquellos procesos que deseemos considerar como no equivalentes.

A continuación haremos un rápido repaso a las diversas técnicas para describir semánticamente un lenguaje.

2.2.1 Semántica denotacional

El objetivo de una semántica denotacional es asignar a cada objeto del lenguaje (en nuestro caso a cada proceso sintáctico) un *objeto matemático* definido de un modo preciso, captando el significado pretendido del mismo. La semántica será entonces una *función matemática* que asignará a cada proceso un objeto.

Dichos objetos son elementos tomados de un conjunto $\mathcal{D}$, que llamaremos *dominio semántico*, dotado de cierta estructura, con el fin de lograr que las ecuaciones recursivas $\xi = \mu\xi.P$ que definen el significado pretendido de la recursión, tengan solución. Para ello utilizamos la técnica de los *puntos fijos*. En concreto nosotros recurriremos a un *orden parcial* entre los elementos del dominio semántico.

Definición 2.7 Una relación $\leq$ sobre un conjunto $\mathcal{D}$ es un orden parcial cuando verifica las propiedades antisimétrica, reflexiva y transitiva.

- $\forall x \in \mathcal{D}\ x \leq x$
- $\forall x, y \in \mathcal{D}\ x \leq y\ \wedge\ y \leq x\ \Rightarrow\ x = y$
- $\forall x, y, z \in \mathcal{D}\ x \leq y\ \wedge\ y \leq z\ \Rightarrow\ x \leq z$

□

Definición 2.8 Dado un dominio $\mathcal{D}$ dotado de un orden parcial $\leq$, decimos que un conjunto $X \subseteq \mathcal{D}$ es *dirigido* si todo subconjunto finito de X tiene una cota superior en X.

$$\forall Y \in \mathcal{PF}(X)\ \exists z \in X : \forall v \in Y,\ v \leq z$$

□

Definición 2.9 Un dominio $\mathcal{D}$ dotado de un orden parcial $\leq$ es un *orden parcial completo (CPO)* si verifica las dos condiciones siguientes:

1. Existe un elemento mínimo, que denotaremos por $\perp$.

$$\forall x \in \mathcal{D} : \perp \leq x$$

2. Todo subconjunto dirigido X tiene límite (cota superior mínima), que denotaremos por $\sqcup X$.

$$\begin{aligned} &\forall x \in X : \quad x \leq \sqcup X \\ &\forall d \in \mathcal{D} : \quad (\forall x \in X : x \leq d) \Rightarrow \sqcup X \leq d \end{aligned}$$

□

Proposición 2.1 Dado un CPO $\mathcal{D}$, si consideramos el conjunto $\mathcal{D}^n = \mathcal{D} \times \ldots \times \mathcal{D}$ (n veces), y lo dotamos del denominado *orden puntual*, o también llamado de *componente a componente*, definido por

$$(d_1, \ldots, d_n) \sqsubseteq (e_1, \ldots, e_n) \Leftrightarrow \forall i \in \{1, \ldots, n\} d_i \sqsubseteq e_i$$

obtenemos un nuevo CPO. □

Definición 2.10 Sean $\mathcal{D}, \mathcal{D}'$ dos CPO's y $\phi : \mathcal{D} \longrightarrow \mathcal{D}'$; decimos que

1. ϕ es **estricta** si conserva el elemento mínimo:

$$\phi(\perp_{\mathcal{D}}) = \perp_{\mathcal{D}'}$$

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

$$d_1 \sqsubseteq_{\mathcal{D}} d_2 \Rightarrow \phi(d_1) \sqsubseteq_{\mathcal{D}'} \phi(d_2)$$

3. ϕ es **continua** si conserva los límites de los conjuntos dirigidos:

$$\forall X \subseteq \mathcal{D}, X \text{ dirigido } : \phi(\sqcup_{\mathcal{D}} X) = \sqcup_{\mathcal{D}'} \phi(X)$$

□

Proposición 2.2 Sean $\mathcal{D}$ y $\mathcal{D}'$ dos CPO's, y $\phi : \mathcal{D}^n \longrightarrow \mathcal{D}'$ una función entre ellos. Tenemos que ella es continua sii lo es componente a componente. □

Proposición 2.3 Sean $\mathcal{D}$ y $\mathcal{D}'$ dos CPO's, y $\phi : \mathcal{D}^n \longrightarrow \mathcal{D}'$ una función entre ellos. Tenemos que ϕ es continua sii

1. ϕ es monótona y
2. $\forall X \subseteq \mathcal{D},\ X$ dirigido $: \phi(\sqcup X) \leq \sqcup \phi(X)$.

□

Teorema 2.1 (Punto fijo, Knaster-Tarski)
Dado $\mathcal{D}$ un CPO, toda función $\phi : \mathcal{D} \longrightarrow \mathcal{D}$ monótona tiene un menor punto fijo en $\mathcal{D}$, que denotaremos $fix(\phi)$. Si además ϕ es continua se verifica

$$fix(\phi) = \sqcup\{\phi^n(\bot) \mid n \geq 0\}$$

siendo $\phi^0(d) = d, \ldots, \phi^{n+1}(d) = \phi(\phi^n(d))$. □

Tras estas consideraciones estamos en condiciones de definir qué es un *modelo denotacional* y qué una semántica denotacional basada en el mismo.

Definición 2.11 Un *modelo denotacional* $\mathcal{M}$ corrrespondiente a una signatura homogénea, viene dada por un CPO $\mathcal{D}_M$ y un conjunto $\{op_M \mid op \in Op\}$ de operadores continuos $op_M : \mathcal{D}_M \times \ldots \times \mathcal{D}_M \longrightarrow \mathcal{D}_M$. □

Definición 2.12 Sea $\mathcal{M}$ un modelo denotacional, entonces un *entorno* es una función $\mathcal{E} : Idf \longrightarrow \mathcal{D}_M$ que asigna a cada identificador un elemento del dominio semántico. □

Definición 2.13 Sea $\mathcal{M}$ un modelo denotacional y sea Ent el conjunto de entornos, entonces la *semántica denotacional* inducida por $\mathcal{M}$ viene dada por la función

$$\mathcal{M}[\![\cdot]\!] : T_\Sigma \longrightarrow (Ent \longrightarrow \mathcal{D}_M)$$

definida por

1. $\mathcal{M}[\![f(P_1,\ldots,P_n)]\!](\mathcal{E}) = f_M(\mathcal{M}[\![P_1]\!](\mathcal{E}),\ldots,\mathcal{M}[\![P_n]\!](\mathcal{E}))$,
2. $\mathcal{M}[\![\xi]\!](\mathcal{E}) = \mathcal{E}(\xi)$,
3. $\mathcal{M}[\![\mu\xi.P]\!](\mathcal{E}) = \mathrm{fix}(\lambda d.\mathcal{M}[\![P]\!](\mathcal{E}[d/\xi]))$, donde $\mathcal{E}[d/\xi]$ es el entorno idéntico a $\mathcal{E}$ salvo en ξ, donde su valor es el $d \in \mathcal{D}_M$ indicado.

□

Evidentemente, la necesidad de los entornos sólo aparece en el contexto de términos abiertos, para los cuales es preciso interpretar de alguna manera las variables libres contenidas en los mismos. Por el contrario el valor semántico de los términos cerrados será independiente del entorno, por lo que tiene sentido hablar de los mismos sin referirnos a ningún entorno concreto.

2.2.2 Semántica operacional

En las semánticas operacionales, el significado de un término (proceso) ya no se obtiene directamente por medio de una función, sino a partir de la descripción de la forma en que dicho proceso va *evolucionando*, transformándose en otros términos. Ello se formaliza por medio de un sistema de transiciones, definido al efecto. Agrupando todas las posibles evoluciones de todos los términos, tendremos descrito el comportamiento de cada proceso.

La forma más usual de representar la evolución de los procesos es mediante un *sistema de transiciones etiquetado.*

Definición 2.14 Un *sistema de transiciones etiquetado* viene dado por una terna $\langle S, A, \{\xrightarrow{a}, a \in A\}\rangle$ donde

- S es un conjunto a cuyos elementos llamamos los *estados* del sistema,
- A es un conjunto, cuyos elementos nos servirán para etiquetar las transiciones, y
- $\xrightarrow{a} \subset S \times S$ es un conjunto de transiciones etiquetadas por a, cada una de las cuales está permitida en su primer estado, conduciéndonos al segundo.

□

Si $s_1, s_2 \in S$, el significado de la transición $s_1 \xrightarrow{a} s_2$ es que un proceso en estado s_1 puede "evolucionar en la forma a" pasando al estado s_2.

2.2.3 Semántica ecuacional o algebraica

En la semántica ecuacional o algebraica, la técnica utilizada para caracterizar un lenguaje es puramente (valga la redundancia) algebraica. Introducimos para ello un conjunto de ecuaciones (axiomas) y unas reglas que definen un sistema de derivación lógico al que denominamos *sistema de prueba ecuacional*, o simplemente sistema de prueba.

Una ecuación es un par de términos, que en general contendrán variables, escrito en la forma $t = t'$. Su significado es que para cualquier asignación de términos a las variables que aparecen en la ecuación, los dos términos resultantes son equivalentes.

Junto a las ecuaciones que demos, tendremos una serie de reglas, cada una de las cuales estára formada por un conjunto finito de ecuaciones que llamamos *premisas*, y una ecuación, que denominamos *conclusión*. El significado es que si bajo una asignación de variables, se satisfacen las premisas de una regla, también se satisfará la conclusión de la misma.

El sistema lógico resultante nos permite realizar *pruebas*, que son secuencias de sentencias de la forma

$$t_1 = t_1', \quad t_2 = t_2', \quad \ldots, t_i = t_i', \quad \ldots$$

donde cada sentencia es o bien un axioma, o bien puede ser derivada aplicando alguna regla a algunas de las sentencias anteriores. Indicaremos mediante la notación $\vdash t = t'$ el que $t = t'$ es una sentencia derivable en el sistema de prueba.

Sea R una relación de equivalencia, decimos que el sistema de prueba es *correcto* respecto a R si para todo par de términos cerrados t y t', se tiene

$$\vdash t = t' \Rightarrow \langle t, t' \rangle \in R$$

Por otra parte, decimos que es *completo* respecto a R si

$$\langle t, t' \rangle \in R \Rightarrow \vdash t = t'$$

Un sistema de prueba es un mecanismo muy abstracto para caracterizar la semántica de un lenguaje, que tiene como ventaja a su favor el de su sencillez y concisión; pero a cambio, se trata de un mecanismo indirecto, y por otra parte las ecuaciones y reglas que los constituyen pueden ser poco intuitivas y en ocasiones demasiado artificiales, por lo cual los sistemas de prueba suelen desarrollarse y presentarse tras haber definido otra semántica más explícita (denotacional, operacional, o similar), con respecto a la cual hemos de demostrar que el sistema de prueba propuesto es correcto y completo, con lo cual tendremos una presentación equivalente más abstracta de la semántica original.

2.3 Sintaxis de CSP

CSP es un lenguaje algebraico utilizado para describir sistemas compuestos por procesos que evolucionan de forma concurrente. La idea básica en que se apoya, consiste en considerar que en principio dichos procesos evolucionan de una forma independiente, hasta que deban ejecutar acciones comunes a varios de ellos, para lo cual habrán de sincronizar. Este mecanismo de sincronización permite además modelar la comunicación entre procesos. En el modelo, como sucede en general en todos aquellos campos en los que se aplican técnicas de abstracción, se ignoran todas aquellas características que resultan irrelevantes para el estudio de la cuestión en la que hemos fijado nuestro interés, que en este caso es la *concurrencia*. En consecuencia se abstrae el significado concreto de las acciones ejecutadas por los procesos, que en consecuencia quedan exclusivamente representadas por el identificador que las denota.

En línea con lo visto en la sección anterior, la sintaxis del lenguaje CSP viene dada por una signatura sobre un único género *de procesos*, que incluye operadores para representar todos los elementos fundamentales de la concurrencia, unos explícitamente, y otros por medio de combinaciones de dichas operaciones. Como ya hemos indicado anteriormente, el punto de partida sobre el que se definen los

procesos es el conjunto de acciones que los mismos pueden ejecutar. Dichas acciones se describen en la siguiente subsección.

2.3.1 Acciones y Alfabeto de un Proceso

Cada proceso de CSP está definido sobre un conjunto de acciones, que representan la realización de eventos por el mismo. Como ya hemos precisado, en CSP no nos importa el significado de las acciones; éstas pueden representar intutivamente cuestiones tan diversas como asignaciones en un lenguaje de programación, la introducción de una moneda en la ranura de una máquina, el funcionamiento de un reloj, el envío de una señal telefónica, etc.

El conjunto de acciones que un proceso puede realizar se denomina su *alfabeto*. En la versión del lenguaje que nosotros utilizaremos consideraremos que todos los procesos tienen un mismo alfabeto, que supondremos finito, pero arbitrariamente grande, de manera que contenga todas las acciones que precisemos utilizar.

Por otra parte, y dentro del mecanismo de abstracción de las características de las acciones que se consideran irrelevantes para nuestro estudio, entendemos que las acciones no llevan asociada una duración concreta, si bien asumimos que la misma es siempre finita. Dicha abstracción nos lleva a considerar la ejecución de las acciones como un proceso instantáneo. En consecuencia el factor tiempo, en lo que de cuantitativo tiene, no será tenido en cuenta al realizar la descripción de procesos mediante este lenguaje, ni por tanto podrá ser estudiado en el marco del modelo. Ello por supuesto no implica que el tiempo no se refleje en términos cualitativos, pues, como veremos es precisamente la ordenación temporal (relativa) de las acciones ejecutadas por un proceso, la que nos proporciona la semántica del mismo.

Otra consecuencia de la visión de un proceso como un agente capaz de ejecutar una serie de acciones (totalmente) ordenadas temporalmente, es que en CSP, al menos en tanto nos quedemos con su semántica de entrelazamiento (*interleaving*) clásica, no es posible describir la ejecución simultánea de acciones, o más exactamente la necesidad o imposibilidad de que tal cosa suceda. Así, cuando no exista relación de causalidad entre dos acciones, tendremos la posibilidad de ejecutar

las mismas en un orden cualquiera. Podríamos en cambio considerar que cuando aparece la necesidad de sincronizar para ejecutar una cierta acción, justamente se está imponiendo la ejecución simultánea de las distintas acciones que sincronizan. Ello sería admisible desde el punto de vista intuitivo, pero no así a nivel formal, pues cuando se produce la ejecución de una acción por medio de una cierta sincronización, se entiende que dicha acción es una sóla, si bien de cara a su ejecución deben colaborar los distintos procesos que se sincronizan.

En general, denotaremos a las acciones con letras minúsculas $a, b, \ldots$, etc, mientras que a los conjuntos de acciones los denotaremos usualmente con las primeras letras mayúsculas, $A, B, \ldots$, etc. Por último, indicaremos que en general los procesos serán denotados mediante las letras mayúsculas, $P, Q, \ldots$, etc.

2.3.2 Operadores del lenguaje

Seguidamente describiremos brevemente las operaciones que forman la signatura del álgebra de procesos CSP. No entraremos sin embargo, en un análisis de las propiedades de las mismas, cuyo desarrollo detallado puede encontrarse en [Hoa85].

La sintaxis de los procesos finitos del lenguaje CSP viene definida por la siguiente expresión BNF:

$$P ::= STOP \mid CHAOS \mid a \rightarrow P \mid P \sqcap P \mid P \,\Box\, P$$

$$\mid P \parallel P \mid P \mid\mid\mid Q \mid P \backslash a$$

STOP: El proceso más sencillo imaginable es el proceso que nunca hace nada. Lo denotamos por STOP, y simboliza una máquina rota, incapaz de realizar acción alguna.

CHAOS: Un proceso que puede comportarse en todo momento de cualquier forma, siendo su comportamiento absolutamente imprevisible. Se trata en consecuencia del proceso más no determinista de todos.

PREFIJO: Nos permite indicar que un proceso comenzará ejecutando una determinada acción. En concreto, $a \rightarrow P$ realizará la acción a, y después pasará a

comportarse de la forma indicada por el proceso P.

ELECCION INTERNA: Nos permite definir procesos cuyo comportamiento es decidido por el sistema, escogiendo para ello entre sus dos argumentos a su libre arbitrio, sin intervención posible del medio exterior en dicha decisión, ni conocimiento (en principio) por parte del mismo de la decision tomada.

ELECCION EXTERNA: El proceso $P \Box Q$ se comporta como P (resp. Q) si el entorno pide la realización de una acción que sólo puede hacer P (resp. Q). Si el entorno pide una acción que pueden hacer ambos, entonces es de nuevo el sistema quien toma libremente la decisión de cuál de los dos procesos será el que la ejecute, siendo el proceso escogido el que continuará ejecutándose.

CONCURRENCIA: El operador $||$ permite agrupar procesos que se ejecutan concurrentemente, sincronizando para la ejecución de cada acción. Así, dada una acción a, cuando uno de los argumentos desea ejecutar dicha acción, debe esperar a que el otro se encuentre también en disposición de ejecutar esa misma acción. Por ejemplo, en una máquina de expedición de chicles, la acción moneda puede corresponder a la introducción de una moneda por parte de un usuario. Por tanto, cuando la máquina llega a ese punto, debe quedar esperando a que el usuario introduzca una moneda (es decir, ejecute también esa acción).

INTERLEAVING: El operador $|||$ modela otro tipo de ejecución en paralelo en el cual no existe sincronización. En esta composición cada uno de los procesos puede evolucionar en respuesta a los requerimientos del medio exterior con independencia del otro.

Estos dos últimos son los operadores paralelos de la versión original [Hoa85] de CSP, que es la que discutiremos en este breve capítulo introductorio. Sin embargo, se ha desarrollado posteriormente [Bro83] una nueva versión del lenguaje, en la que dichos operadores son sustituidos por un único operador de concurrrencia más general, el cual lleva un argumento adicional, que es un conjunto de acciones, que define el *conjunto de acciones de sincronización.* Así, el proceso $P \|_A Q$ corresponde a la ejecución concurrente de los procesos P y Q, los cuales deben sincronizar para ejecutar las acciones del conjunto A, mientras pueden ejecutar

independientemente las acciones que no pertenezcan a dicho conjunto. Dado que esta nueva versión del modelo es más compacta, al contar con un operador menos, pero al mismo tiempo más flexible, dada la mayor generalidad del nuevo operador paralelo, se ha hecho práctica común la utilización de la misma, tendencia a la que nosotros nos uniremos, siendo la misma la que tomaremos como punto de partida para definir nuestro lenguaje PCSP, objeto de esta tesis.

OCULTACION (HIDDING): En un proceso puede interesarnos abstraer los efectos de alguna acción en particular, para lo cual dispondremos del operador de *ocultación*, cuyo efecto consiste en hacer invisibles, y por tanto incontrolables por el medio exterior, las ejecuciones de la acción ocultada.

De cara a la definición de procesos infinitos, que resultan de capital importancia cuando se desea modelar sistemas *reactivos*, el lenguaje cuenta con un operador recursivo adicional.

RECURSION: En CSP se utiliza la técnica descrita en la sección anterior, de introducción de procesos variables junto con los operadores (sintácticos) de punto fijo, para definir los procesos infinitos (recursivos). Así, $\mu X.(a \rightarrow X)$ representa un proceso que realiza indefinidamente la acción a, lo que ilustra la interpretación habitual de un proceso $\mu X.P$ definido recursivamente, consistente en verlo como una solución (distinguida) de la ecuación asociada $X = P$.

2.4 Semántica denotacional de CSP

El comportamiento de un proceso lo describimos de manera denotacional por medio de su conjunto de *rechazos*. Un rechazo de un proceso es un par formado por una secuencia de acciones ejecutadas por el mismo (una *traza*) y un conjunto finito de acciones, cuyo significado es que el proceso puede ejecutar dicha traza, tras lo cual puede rechazar simultaneamente la ejecución de las acciones indicadas en el conjunto. Por tanto el conjunto de rechazos F que define un proceso es un subconjunto de $\mathcal{P}(\Sigma^* \times \mathcal{P}(\Sigma))$, que ha de satisfacer una serie de condiciones:

1. $(\langle\rangle, \emptyset) \in F$, donde $\langle\rangle$ representa la traza vacía.

2. $(st, \emptyset) \in F \Rightarrow (s, \emptyset) \in F$.

3. $V \subseteq W$ y $(s, W) \in F \Rightarrow (s, V) \in F$.

4. Sea $U = \{a \mid (sa, \emptyset) \in F\}$ y sea W un subconjunto finito de $\Sigma - U$, entonces $(s, V) \in F \Rightarrow (s, V \cup W) \in F$.

La función semántica $[\![\ ]\!]$ que define la semántica denotacional de CSP se define, como es usual, por inducción estructural, en la forma siguiente

$$[\![STOP]\!] = \{(\langle\rangle, X) \ / \ X \subseteq \Sigma, \quad X \text{ finito}\}$$

$$[\![CHAOS]\!] = \{(s, X)/s \in \Sigma^* \wedge X \subseteq \Sigma, \ X \text{ finito}\}$$

$$[\![a \rightarrow P]\!] = \{(\langle\rangle, X)/X \subseteq (\Sigma - \{a\}) \wedge X \text{ finito}\} \cup \{(\langle a\rangle s, X)/(s, X) \in [\![P]\!]\}$$

$$[\![P \sqcap Q]\!] = [\![P]\!] \cup [\![Q]\!]$$

$$[\![P \ \Box \ Q]\!] = \{(\langle\rangle, X)/(\langle\rangle, X) \in [\![P]\!] \cap [\![Q]\!]\} \cup \{(s, X)/s \neq \langle\rangle \wedge ((s, X) \in [\![P]\!] \cup [\![Q]\!])\}$$

$$[\![P \parallel Q]\!] = \{(s, X \cup Y)/(s, X) \in [\![P]\!] \wedge (s, Y) \in [\![Q]\!]\}$$

$$[\![P \ ||| \ Q]\!] = \{(u, X)/\exists s, t.(s, X) \in [\![P]\!] \wedge (t, X) \in [\![Q]\!] \wedge u \in merge(s, t)\}$$

$merge(s, t)$ es el conjunto de mezclas de las trazas s y t

$$[\![P \backslash a]\!] = \{(s \backslash a, X)/(s, X \cup \{a\}) \in [\![P]\!]\} \cup \{((s \backslash a)t, X)/(s, X \cup \{a\}) \in [\![P]\!] \ \wedge \ (t, X) \in [\![CHAOS]\!] \ \wedge \ \forall n \ (sa^n, \emptyset) \in [\![P]\!]\}$$

$s \backslash a$ se obtiene eliminando las ocurrencias de acciones a en s

2.5 Semántica operacional de CSP

Como hemos indicado, el significado operacional de un proceso vendrá dado por las posibles evoluciones del mismo. En este caso introducimos al efecto un par de sistemas de transiciones: uno etiquetado que representa las evoluciones con acciones

visibles, y otro no etiquetado para indicar las evoluciones (internas) autónomas de un proceso. Así, la transición

$$P \xrightarrow{a} Q$$

representa que el proceso P ejecuta la acción a, y pasa a comportarse como el proceso Q, mientras que

$$P \longrightarrow Q$$

nos indica que el proceso P puede pasar a comportarse, por decisión interna, como el proceso Q, sin que sea visible la ejecución de acción alguna.

La tabla 1 recoge la definición estructurada del conjunto de transiciones operacionales de los procesos, tanto visibles como internas.

2.6 Sistema de prueba para CSP

Es posible describir de una forma más compacta la semántica de CSP, por medio de un Sistema de Prueba. El mismo consta de un conjunto de axiomas y reglas, que reflejan las propiedades de cada operador del lenguaje, junto con las interrelaciones entre los mismos. Dichos axiomas definen la noción de *equivalencia semántica* entre procesos, junto con el orden entre los mismos inducido por la relación de orden en que se basa la definición de la semántica denotacional. El uso de esta relación de orden es preceptivo, aún en el caso de que en principio sólo estemes interesados en la equivalencia entre procesos, de cara a incluir en el marco axiomático el tratamiento de los procesos definidos recursivamente.

El núcleo del sistema lo constituye el conjunto de axiomas correspondiente a los procesos finitos definibles a partir de la constante STOP, por medio del operador prefijo y los dos operadores de elección, que presentamos en la tabla 2. Por su parte, en la tabla 3 se muestran las reglas generales que nos indican la relación entre la equivalencia y la relación de orden entre procesos, y el hecho de que todos los susodichos operadores son monótonos respecto de dicho orden.

Posteriormente, en la tabla 4 incluímos los axiomas que nos permiten razonar sobre procesos infinitos. Al respecto tenemos por una parte los axiomas correspondientes al proceso $CHAOS$ que denota sintácticamente al elemento mínimo del

1) $\dfrac{}{a \to P \xrightarrow{a} P}$

2) $\dfrac{}{P \sqcap Q \longrightarrow P}$

3) $\dfrac{}{P \sqcap Q \longrightarrow Q}$

4) $\dfrac{P \xrightarrow{a} Q}{P \sqcap R \xrightarrow{a} Q}$

5) $\dfrac{P \xrightarrow{a} Q}{R \sqcap P \xrightarrow{a} Q}$

6) $\dfrac{P \xrightarrow{a} P'}{P \,\Box\, Q \xrightarrow{a} P'}$

7) $\dfrac{Q \xrightarrow{a} Q'}{P \,\Box\, Q \xrightarrow{a} Q'}$

8) $\dfrac{P \longrightarrow P',\ Q \longrightarrow Q'}{P \,\Box\, Q \longrightarrow P' \,\Box\, Q'}$

9) $\dfrac{P \longrightarrow P'}{P \parallel Q \longrightarrow P' \parallel Q}$

10) $\dfrac{Q \longrightarrow Q'}{P \parallel Q \longrightarrow P \parallel Q'}$

11) $\dfrac{P \xrightarrow{a} P',\ Q \xrightarrow{a} Q'}{P \parallel Q \xrightarrow{a} P' \parallel Q'}$

12) $\dfrac{P \longrightarrow P'}{P \,|||\, Q \longrightarrow P' \,|||\, Q}$

13) $\dfrac{Q \longrightarrow Q'}{P \,|||\, Q \longrightarrow P \,|||\, Q'}$

14) $\dfrac{P \xrightarrow{a} P'}{P \,|||\, Q \xrightarrow{a} P' \,|||\, Q}$

15) $\dfrac{Q \xrightarrow{a} Q'}{P \,|||\, Q \xrightarrow{a} P \,|||\, Q'}$

16) $\dfrac{P \longrightarrow Q}{P \setminus a \longrightarrow Q \setminus a}$

17) $\dfrac{P \xrightarrow{a} Q}{P \setminus a \longrightarrow Q \setminus a}$

18) $\dfrac{P \xrightarrow{b} Q,\ b \neq a}{P \setminus a \xrightarrow{b} Q \setminus a}$

19) $\dfrac{P[P/X] \longrightarrow P'}{\mu X.P \longrightarrow P'}$

20) $\dfrac{P[P/X] \xrightarrow{a} P'}{\mu X.P \xrightarrow{a} P'}$

Tabla 1: Semántica operacional de CSP

$(A1)$	$P \sqcap P \equiv P$	$(A7)$	$P \sqcap (Q \square R) \equiv (P \sqcap Q) \square (P \sqcap R)$
$(A2)$	$P \square P \equiv P$	$(A8)$	$P \square (Q \sqcap R) \equiv (P \square Q) \sqcap (P \square R)$
$(A3)$	$P \sqcap Q \equiv Q \sqcap P$	$(A9)$	$P \square STOP \equiv P$
$(A4)$	$P \square Q \equiv Q \square P$	$(A10)$	$P \sqcap Q \sqsubseteq P$
$(A5)$	$P \sqcap (Q \sqcap R) \equiv (P \sqcap Q) \sqcap R$	$(A11)$	$(a \rightarrow P) \sqcap (a \rightarrow Q) \equiv a \rightarrow P \sqcap Q$
$(A6)$	$P \square (Q \square R) \equiv (P \square Q) \square R$	$(A12)$	$(a \rightarrow P) \square (a \rightarrow Q) \equiv a \rightarrow P \sqcap Q$

Tabla 2: Axiomas para procesos finitos

(O1)	$\dfrac{P \sqsubseteq Q \sqsubseteq P}{P \equiv Q}$	(M1)	$\dfrac{P \sqsubseteq Q}{a \rightarrow P \sqsubseteq a \rightarrow Q}$
(O2)	$\dfrac{P \equiv Q}{P \sqsubseteq Q \sqsubseteq P}$	(M2)	$\dfrac{P \sqsubseteq Q \ \wedge \ P' \sqsubseteq Q'}{(P \sqcap P') \sqsubseteq (Q \sqcap Q')}$
(O3)	$\dfrac{P \sqsubseteq Q \sqsubseteq R}{P \sqsubseteq R}$	(M3)	$\dfrac{P \sqsubseteq Q \ \wedge \ P' \sqsubseteq Q'}{(P \square P') \sqsubseteq (Q \square Q')}$

Tabla 3: Reglas para procesos finitos

$(B1)$	$P \sqcap CHAOS \equiv CHAOS$	$(B3)$	$CHAOS \sqsubseteq P$
$(B2)$	$P \Box CHAOS \equiv CHAOS$	$(B4)$	$P[(\mu\xi.P)/\xi] \sqsubseteq \mu\xi.P$

$$(\mathrm{R}) \quad \frac{\forall Q \in FIN(P): Q \sqsubseteq R}{P \sqsubseteq R}$$

Tabla 4: Axiomas y reglas para procesos recursivos

dominio de procesos, y los axiomas correspondientes al operador de recursión.

Por último, en la tabla 5 se reflejan los axiomas correspondientes a los operadores de composición paralela, entrelazamiento y ocultación. Dichos axiomas nos muestran que en todos los casos se trata de operaciones derivadas, por poderse reescribir toda aplicación de las mismas sobre términos construíbles con las operaciones básicas antes consideradas, a un nuevo término en el que sólo aparecen dichas operaciones.

De cara a probar la completitud de dicho sistema de prueba introducimos unas formas normales o canónicas de los términos definibles con el lenguaje, las cuales nos permiten describir la totalidad de los procesos semánticos así definibles, quedando además definidos de forma muy explícita los correspondientes valores semánticos, lo que facilita enormemente la manipulación y comprensión de los mismos.

Definición 2.15 Sea $\mathcal{A}$ una familia de conjuntos de acciones. Llamaremos *alfabeto* del mismo, y lo denotaremos por $\Sigma(\mathcal{A})$, al conjunto $\Sigma(\mathcal{A}) = \{a \in \Sigma \mid \exists A \in \mathcal{A} : a \in A\}$. Diremos que dicha familia es *saturada* sii para cada $K \subseteq \Sigma(\mathcal{A})$ se tiene que si $A \subseteq K$ para un cierto $A \in \mathcal{A}$, entonces $K \in \mathcal{A}$. □

Definición 2.16 Diremos que un término P está en *Forma Normal* cuando tiene la forma

$$P = \bigsqcap_{A \in \mathcal{A}} \{ \Box_{a \in A}\, a \rightarrow P_a \}$$

donde $\mathcal{A}$ es una familia saturada, y cada P_a está a su vez en forma normal.

En el caso particular de que $\mathcal{A}$ sea vacía, convenimos en que se trata del proceso $CHAOS$, mientras que la posible aparición de un conjunto vacío en la familia, daría lugar a un proceso $STOP$. □

$$
\begin{array}{ll}
(PAR0) & P\|CHAOS \equiv CHAOS \\
(PAR1) & P\|(Q \sqcap R) \equiv (P\|Q) \sqcap (P\|R) \\
(PAR2) & P\|Q \equiv \underset{a \in B \cap C}{\Box} a \rightarrow (P_a\|Q_a) \\
(PAR3) & P\|Q \equiv Q\|P \\
 & \\
(INT0) & P \;|||\; CHAOS \equiv CHAOS \\
(INT1) & P \;|||\; STOP \equiv P \\
(INT2) & P \;|||\; (Q \sqcap R) \equiv (P \;|||\; Q) \sqcap (P \;|||\; R) \\
(INT3) & P \;|||\; Q \equiv (\underset{i \in I}{\Box} (a_i \rightarrow (P_i \;|||\; Q)) \Box (\underset{j \in J}{\Box} (b_j \rightarrow (P \;|||\; Q_j)) \\
 & \\
(HID0) & CHAOS\backslash b \equiv CHAOS \\
(HID1) & (P \sqcap Q)\backslash b \equiv (P\backslash b) \sqcap (Q\backslash b) \\
(HID2) & (b \rightarrow P)\backslash c \equiv \\
 & \left\{ \begin{array}{ll} (b \rightarrow P\backslash c) & \text{si } b \neq c \\ P\backslash c & \text{si } b = c \end{array} \right. \\
(HID3) & \underset{a \in A}{\overset{a}{\Box}} \rightarrow P_a\backslash c \equiv \\
 & \left\{ \begin{array}{ll} \underset{a \in A}{\Box} a \rightarrow (P_a\backslash c) & \text{si } c \notin A \\ (P_c\backslash c) \sqcap \underset{a \in A}{\Box} a \rightarrow (P_a\backslash c) & \text{si } c \in A \end{array} \right.
\end{array}
$$

Tabla 5: Axiomas para procesos derivados

Puesto que dos formas normales nunca son equivalentes, probar la equivalencia entre dos procesos se reduce a comprobar que su forma normal es la misma, mientras que la prueba de completitud del sistema (en lo que se refiere a procesos finitos) se reduce a probar que, en efecto, el mismo nos permite reescribir todo término en su forma normal.

Capítulo 3

Primera aproximación al lenguaje PCSP

El lenguaje CSP ([Hoa85] y [Bro83]) ofrece un formalismo adecuado para la especificación de sistemas distribuidos no deterministas. En el presente capítulo presentaremos la extensión probabilística de dicho lenguaje, a cuyo estudio se ha dedicado el presente trabajo. El lenguaje resultante, al que hemos denominado CSP probabilístico (PCSP), hemos procurado que mantenga en lo posible las propiedades fundamentales del lenguaje madre, incorporando de manera coherente las nociones probabilísticas básicas.

3.1 Sintaxis de PCSP

La sintaxis de PCSP es una sencilla variante de la de CSP, con la única diferencia de añadir una distribución de probabilidades a todos aquellos operadores que conllevan (explícita o implícitamente) una elección. La expresamos mediante la siguiente expresión BNF

$$P ::= STOP \mid X \mid a \rightarrow P \mid [p_1]P \sqcap [p_2]P \mid [p_1]P \,\Box\, [p_2]P$$

$$\mid [p_1]P \parallel_A [p_2]P \mid P\backslash(a,p) \mid \mu X.P$$

donde $p \in [0,1]$, p_1, $p_2 \in [0,1]$, con $p_1 + p_2 = 1$ y $X \in Idf$.

El significado intuitivo de cada uno de los operadores es el siguiente:

STOP : STOP sigue representando a una máquina rota, incapaz de realizar acción alguna.

PREFIJO Dados un proceso P y una acción a, $a \rightarrow P$ es un proceso que en primer lugar ejecuta con seguridad, es decir con probabilidad 1, la acción a para después pasar a comportarse como el proceso P.

ELECCIÓN INTERNA: Dados dos procesos P y Q, y $p_1, p_2 \in [0,1]$ tal que $p_1 + p_2 = 1$, entonces $[p_1]P \sqcap [p_2]Q$ es un proceso que se comporta como P con probabilidad p_1 y como Q con probabilidad p_2.

ELECCIÓN EXTERNA: Dados dos procesos P y Q, y $p_1, p_2 \in [0,1]$ tal que $p_1 + p_2 = 1$, entonces $[p_1]P \,\Box\, [p_2]Q$ es un proceso cuyo comportamiento es en principio determinista. Así, si el entorno del proceso decide ejecutar una acción y uno de los dos procesos P y Q, pero no ambos, están dispuestos a ello, la acción será ejecutada por el proceso dispuesto a ello, lo que sucederá con la misma probabilidad con la que ambos argumentos hayan alcanzado los estados correspondientes, continuando después la ejecución del proceso que ejecutó la acción. Sólo en el caso de que ambos estén en disposición de ejecutar dicha acción la elección será aleatoria, de acuerdo con las probabilidades p_1 y p_2. Y como quiera que las distribuciones de probabilidad que rigen ambos procesos son independientes, la probabilidad de ejecutar una acción es siempre mayor en el proceso resultante de elegir entre dos, que en cualquiera de ellos, pues ahora una acción sólo será rechazada cuando ninguno de los procesos esté en disposición de ejecutarla.

Obsérvese, que aún en el caso de que todas las posibles acciones a ejecutar por ambos procesos fuesen comunes, la elección externa no sería equivalente a la interna, ya que para calcular la probabilidad con la que se ejecutará una acción, además de las probabilidades asociadas a la elección misma, hay que tener en cuenta, y ello se hace de manera diferente según se trate de la elección interna o externa, las probabilidades que cada proceso tiene de aceptar la ejecución de dicha acción. En consecuencia, aún bajo las hipótesis restrictivas mencionadas, la equivalencia entre los resultados de aplicar los dos operadores de elección, sólo

se tendrá en el caso de que ambos argumentos sean deterministas, es decir estén en disposición de ejecutar todas las acciones que ofrezcan con probabilidad 1. Encontramos así una primera diferencia, por otra parte bien razonable, con el caso no-probabilístico, en el que ambos operadores de elección coinciden cuando se aplican sobre argumentos deterministas que ofrecen un mismo conjunto de acciones.

COMPOSICIÓN PARALELA: Dados dos procesos P y Q, $p_1, p_2 \in [0,1]$ tales que $p_1 + p_2 = 1$ y un conjunto de acciones A, entonces $[p_1]P \parallel_A [p_2]Q$ es un proceso que corresponde a la ejecución concurrente de P y Q, sincronizando sobre las acciones de A. Dada una acción $a \in A$, para que pueda ejecutarse, ambos procesos deben estar en condiciones de ejecutarla, obteniéndose como resultado de esa ejecución conjunta, una única acción a. Para las acciones no contenidas en el conjunto de sincronización A, se actúa exactamente igual que en el caso de la elección externa, de manera que si uno sólo de los procesos puede ejecutar la acción lo hará, mientras el otro permanecerá en el mismo estado; mientras que si ambos están en disposición de hacerlo, se decidirá de manera aleatoria, en base a las probabilidades p_1 y p_2 quien lo hará, continuando también el proceso no escogido en su estado de partida.

OCULTACION: Dado un proceso P y un par (a, q), con $a \in \Sigma$ y $q \in [0,1]$, $P \backslash (a, q)$ es un proceso en el que las ejecuciones de la acción oculta a no pueden ser observadas, decidiéndose internamente cuando se harán, cuando ello sea posible, en base a la probabilidad q indicada. En realidad ello sólo debe reflejarse a nivel formal en el marco de aquellas elecciones externas en las que aparezca la acción ocultada como una de las opciones. En este caso, el entorno no mantiene siempre su posibilidad de elegir, pues el sistema puede decidir autónomamente la ejecución de la acción ocultada.

RECURSIÓN: La recursión se incorporará exactamente como en el lenguaje madre CSP, tanto en lo que a sintaxis como en lo que a semántica hace referencia.

Todas estas ideas informales sobre los comportamientos de los procesos probabilísticos serán formalizadas, en una primera aproximación, que a la postre comprobaremos que es en exceso simplista, en la siguiente sección, en la que presentamos una primera semántica operacional para nuestro lenguaje. En concreto,

una primera consecuencia de dicha simplicidad es, como comprobaremos, la imposibilidad de definir la semántica del operador de ocultación, que por lo tanto consideraremos excluido temporalmente.

3.2 Semántica operacional de PCSP

La semántica que presentaremos a continuación, y que posteriormente utilizaremos vía un ejemplo sencillo pero no trivial, para razonar en lo posible, sobre los comportamientos de procesos probabilísticos, es una semántica operacional. Así, vamos a definir la evolución o ejecución de un proceso mediante un sistema probabilístico de transiciones (veáse [LS89]), cuyas transiciones serán aquellas generables por medio de un conjunto de reglas, cada una de las cuales contribuye a definir el significado de uno de los operadores del lenguaje. La definición de la semántica seguirá el estilo estructurado de Plotkin [Plo81] y Milner [Mil89].

Definición 3.1 Una *transición probabilística observable* es una tupla $\langle P, Q, a, p\rangle$ donde P y Q son procesos, $a \in \Sigma$, y $p \in (0,1]$. Usualmente representaremos una transición en la forma:

$$P \xrightarrow{a}_p Q$$

□

En la tabla 6 presentamos las reglas que permiten generar las transiciones probabilísticas que definen esta primera semántica operacional para PCSP.

En algunas de dichas reglas aparece la noción derivada $P \xrightarrow{a}_p$ que representa la probabilidad con la que el proceso P aceptará la ejecución de la acción, y cuya definición formal dejamos para más adelante. Antes introduciremos una serie de conceptos y terminología, auxiliares.

Definición 3.2 Llamamos semántica operacional del lenguaje PCSP (sin operador de ocultación) al multiconjunto de transiciones probabilísticas que pueden ser inferidas aplicando el sistema de reglas anterior, donde cada transición aparece tantas veces como formas diferentes haya de derivarla. □

1) $$\frac{}{a \rightarrow P \xrightarrow{a}_1 P}$$

2) $$\frac{P \xrightarrow{a}_p Q}{[q]P \sqcap [1-q]R \xrightarrow{a}_{pq} Q}$$

3) $$\frac{P \xrightarrow{a}_p Q}{[q]R \sqcap [1-q]P \xrightarrow{a}_{p(1-q)} Q}$$

4) $$\frac{P \xrightarrow{a}_p P',\ Q \xrightarrow{a}_q}{[r]P \,\Box\, [1-r]Q \xrightarrow{a}_s P'}$$
donde $s = pqr + p(1-q)$

5) $$\frac{P \xrightarrow{a}_p ,\ Q \xrightarrow{a}_q Q'}{[r]P \,\Box\, [1-r]Q \xrightarrow{a}_s Q'}$$
donde $s = pq(1-r) + (1-p)q$

6) $$\frac{P \xrightarrow{a}_p P' ,\ Q \xrightarrow{a}_{p'} ,\ a \notin A}{[q]P \parallel_A [1-q]Q \xrightarrow{a}_s [q]P' \parallel_A [1-q]Q}$$
donde $s = pp'q + p(1-p')$

7) $$\frac{P \xrightarrow{a}_{p'} ,\ Q \xrightarrow{a}_p Q' ,\ a \notin A}{[1-q]P \parallel_A [q]Q \xrightarrow{a}_s [1-q]P \parallel_A [q]Q'}$$

8) $$\frac{P \xrightarrow{a}_p P' ,\ Q \xrightarrow{a}_{p'} Q' ,\ a \in A}{[q]P \parallel_A [1-q]Q \xrightarrow{a}_{pp'} [q]P' \parallel_A [1-q]Q'}$$
donde $s = pp'q + p(1-p')$

9) $$\frac{P[\mu X.P/X] \xrightarrow{a}_p P'}{\mu X.P \xrightarrow{a}_p P'}$$
donde $P[Q/X]$ representa la sustitución sintáctica en P del identificador X por el proceso Q

Tabla 6: Primera semántica operacional

Notación: Sea $\mathcal{T} = P \xrightarrow{a}_p Q$, una transición inferida usando las reglas semánticas. Llamaremos una *derivada inmediata* de P a la tupla $\langle a, Q, p \rangle$. De la misma forma, diremos que a es una *acción inmediata* de P con probabilidad p para ser ejecutada por medio de $\mathcal{T}$, y que Q es un *proceso a-derivado* de P con probabilidad p correspondiente a $\mathcal{T}$.

Comentaremos brevemente a continuación las distintas reglas del sistema. Las tres primeras son suficientemente claras, por lo que no precisan de mayor comentario. En las reglas cuarta y quinta, la probabilidad de que el proceso $[r]P \,\Box\, [1-r]Q$ ejecute la acción a y pase a comportarse como P' o como Q' dependiendo de la regla aplicada, se calcula considerando la posibilidad de que o bien P o bien Q ejecuten dicha acción. Sin embargo, si ambos procesos pueden ejecutar dicha acción, sólo uno de ellos podrá ser elegido para hacerlo, y esta elección se hará teniendo en cuenta la probabilidad r indicada en la elección externa, que en concreto define la probabilidad con la que seleccionaremos el primer proceso cuando esta situación se presente.

Las reglas sexta y séptima definen la parte de la semántica del operador paralelo correspondiente al caso de *entrelazamiento*. Con ellas se indica que si la acción a ejecutar no pertenece al conjunto de sincronización, será realizada por el proceso componente que esté capacitado para ello. Si ambos procesos lo estuvieran, se elegirá aleatoriamente, de manera análoga a como se ha descrito para el operador de elección externa, atendiendo en este caso a la probabilidad asociada al operador de composición paralela, que indica también en este caso, la probabilidad con la que será escogida en tales situaciones la primera componente.

La octava regla describe el caso en el que la acción a ejecutar pertenece al conjunto de sincronización. Precisamos entonces que la acción sea ejecutada por ambos procesos simultáneamente, y esto obviamente ocurrirá con una probabilidad igual al producto de las correspondientes a las transiciones de los procesos componentes que hayamos combinado para obtener la transición en cuestión del proceso paralelo.

Finalmente, la última regla trata al operador recursivo en la forma tradicional, no necesitando por ello de ningún comentario adicional.

Ahora podemos definir formalmente el concepto de probabilidad con la que un proceso P ejecuta una acción a, lo que se hace en la forma siguiente:

Definición 3.3 Sea a una acción y P un proceso. Diremos que p es la probabilidad con la que P ejecuta la acción a, inducida por la presente semántica operacional, lo que representaremos en la forma

$$P \xrightarrow{a}_{p}$$

sii $p = \sum_{i \in I} p_i$ donde I es el multiconjunto de transiciones

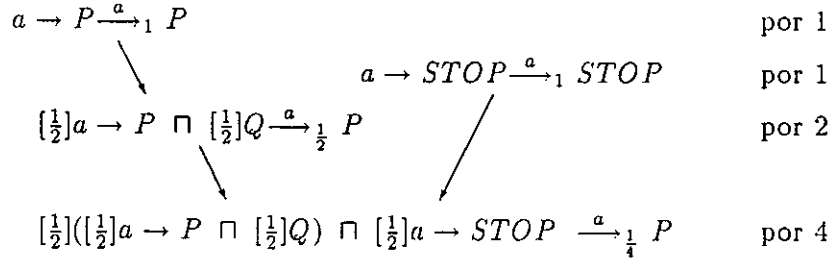
$$P \xrightarrow{a}_{p_i} P_i$$

que ejecutan la acción a, que pueden ser derivadas, con derivaciones diferentes, utilizando las reglas del sistema. □

Ejemplo 3.1 La transición

$$[\tfrac{1}{2}]([\tfrac{1}{2}]a \rightarrow P \sqcap [\tfrac{1}{2}]Q) \sqcap [\tfrac{1}{2}]a \rightarrow STOP \xrightarrow{a}_{\frac{1}{4}} P$$

puede inferirse utilizando el sistema de reglas.



Con lo cual se ha derivado la transición tras cuatro pasos de inferencia, correspondientes a la aplicación de las reglas 1, 1, 2 y 4, respectivamente. □

Como ya hemos indicado, de la forma en que se define el multiconjunto de transiciones derivables, se sigue que es importante distinguir la forma en la que se ha obtenido cada transición concreta. Y que en efecto es necesario trabajar con multiconjuntos para definir la semántica operacional del lenguaje, queda ilustrado

por ejemplo, considerando el proceso $P = [\frac{1}{2}]a \rightarrow STOP \sqcap [\frac{1}{2}]a \rightarrow STOP$, para el cual tenemos que

$\mathcal{T}_1 = P \xrightarrow{a}_{\frac{1}{2}} STOP$ es derivable aplicando las reglas 1 y 2, mientras

$\mathcal{T}_2 = P \xrightarrow{a}_{\frac{1}{2}} STOP$ lo es aplicando las reglas 1 y 3.

Con ambas derivaciones se infiere que P puede ejecutar la acción a con probabilidad $\frac{1}{2}$. Sin embargo, parece vital que quede constancia de que se trata de dos computaciones diferentes, a fin de que pueda inferirse a la postre que efectivamente la acción a será ejecutada con probabilidad 1.

Una alternativa más formal a la hora de distinguir las transiciones, según la forma en la que las mismas hayan sido generadas, es la utilizada en [GJS90], donde las transiciones están indexadas con índices que indican la vía en que han sido generadas. Pero como quiera que dicho tecnicismo es en sí mismo trivial, nosotros hemos preferido no utilizarlo de forma explícita, para no perder al lector con un tecnicismo adicional, que sin embargo no añade ninguna idea nueva, sin por tanto afectar en absoluto a los resultados que pueden ser obtenidos, ni suponer cambio alguno en lo que se refiere a las ideas y técnicas fundamentales que han de ser utilizadas para obtenerlos.

Comentamos por último el tema del operador de ocultación. Para el mismo, como ya advertimos, no hemos presentado regla alguna, lo que naturalmente implica que no ha quedado definida su semántica. Ello se debe simplemente a que resulta absolutamente imposible capturar el comportamiento *razonable* (al menos el que nosotros hemos entendido como tal) de dicho operador, en un marco tan sencillo como el de una semántica como la presente, que pretenda razonar en base exclusivamente al estudio de acciones individuales.

El problema radica en el hecho de que al aplicar el operador de ocultación sobre una elección externa, ésta se convierte en parcialmente interna. En cambio el significado de las elecciones internas no varía al aplicarles una ocultación. Consideremos al respecto el siguiente

Ejemplo 3.2 Sean P y Q los siguientes procesos:

$$P = [\tfrac{1}{2}]([\tfrac{1}{2}]a \rightarrow STOP \;\Box\; [\tfrac{1}{2}]b \rightarrow STOP) \sqcap [\tfrac{1}{2}]STOP$$
$$Q = [\tfrac{1}{2}]a \rightarrow STOP \sqcap [\tfrac{1}{2}]b \rightarrow STOP$$

Ambos procesos son equivalentes respecto de cualquier noción basada en la presente semántica operacional, pues la misma es idéntica para ambos. Sin embargo, si ocultamos la acción a el comportamiento deseado pasa a ser diferente, pues en $P\backslash(a,q)$ la probabilidad de ejecutar la acción b debe disminuir, mientras que en $Q\backslash(a,q)$ debe seguir siendo $\frac{1}{2}$, lo que es imposible de lograr en un marco composicional, si, como es el caso, las semánticas de P y Q son idénticas. □

La única vía que nos permitirá discernir a nivel semántico, si en un momento dado un proceso puede realizar una elección externa, para, entre otras cosas, poder capturar debidamente el significado del operador de ocultación, consiste en poder ofrecer al mismo no sólo acciones aisladas, sino también más de una acción simultáneamente, con lo que podremos descubrir si el proceso está en ese momento dispuesto a ejecutar distintas acciones, lo que implicaría que nos encontramos ante una elección externa entre todas las acciones que podamos ejecutar.

Estas ideas serán desarrolladas extensamente en los capítulos venideros. De momento nos limitaremos a estudiar los procesos PCSP en base a la semántica que hemos definido. Ello implica en particular que el operador de ocultación queda excluido por el momento.

Comenzaremos definiendo las computaciones de los procesos probabilísticos, pues las transiciones por sí mismas sólo definen pasos aislados en la evolución de un proceso. En consecuencia encadenaremos dichas transiciones para lograr una imagen más global de las ejecuciones de los procesos.

Definición 3.4 Llamamos computaciones de un proceso probabilístico P a las secuencias de transiciones:

$$P = P_0 \xrightarrow{a_1}_{p_1} P_1, \ldots, P_{n-1} \xrightarrow{a_n}_{p_n} P_n$$

permitidas por la semántica operacional de PCSP. Usualmente las representaremos en la forma

$$C = P \xrightarrow{a_1}_{p_1} P_1 \xrightarrow{a_2}_{p_2} \cdots \xrightarrow{a_n}_{p_n} P_n$$

Llamaremos una *derivada* de P generada por C al triple (s, P_n, p), donde $s = a_1 a_2 \ldots a_n$ y $p = p_1 \cdot p_2 \cdots p_n$. Usualmente representaremos estas derivadas con la

notación

$$P \stackrel{s}{\Longrightarrow}_p P_n$$

De nuevo cada proceso tendrá un multiconjunto de computaciones, en el que cada computación aparece tantas veces como indique el producto de las veces que aparece cada una de sus componentes en la semántica del lenguaje. □

La noción de derivada puede ser definida recursivamente en la forma siguiente:

Definición 3.5 Sea s una secuencia (posiblemente vacía) de acciones, $p \in (0,1]$ y P, Q dos procesos. Entonces, $P \stackrel{s}{\Longrightarrow}_p Q$ es una computación si y sólo si es posible inferirla por medio del uso de las reglas

1. $P \stackrel{\langle\rangle}{\Longrightarrow}_1 P$
2. Si $P \stackrel{a}{\longrightarrow}_p P'$ y $P' \stackrel{s}{\Longrightarrow}_q Q$, entonces $P \stackrel{as}{\Longrightarrow}_{pq} Q$

□

En la sección siguiente ilustraremos el estudio de esta semántica por medio de un ejemplo no trivial, que mostrará una aplicación concreta de nuestro lenguaje, el cual veremos que incluso dotado de una semántica limitada como la que hemos desarrollado por el momento, puede abordar de forma elegante y satisfactoria, un ejemplo que en el caso de otros modelos anteriores de procesos probabilísticos desarrollados por otros autores, sólo podía abordarse de forma indirecta, y con consideraciones muy ligadas a las características propias del ejemplo en cuestión, que hacen muy improbable la generalización de los razonamientos a marcos más complejos.

3.3 Ejemplo. Protocolo AUY

Presentaremos un ejemplo en el cual modelamos el protocolo AUY (ver [AUY79]) utilizado en la transmisión de datos, que asegura una transmisión segura sobre un sistema en el que los canales de comunicación pueden fallar. Este ejemplo también

se ha estudiado en [GJS90], en el marco del álgebra de procesos (PCCS) allí desarrollada. Sin embargo, en nuestra opinión, este tipo de álgebra síncrona plantea algunos problemas que dificultan el modelado de sistemas. Unos de ellos derivados de la sincronía, pues es necesario sincronizar *a mano* todas las componentes del sistema que no están sincronizadas de una manera natural, mientras que otros vienen derivados de la inexistencia de un operador explícito de elección externa, que sólo puede ser parcialmente simulado por medio de una adecuada combinación de las probabilidades en el único operador de elección que incluye, junto con la restricción de los procesos a un adecuado conjunto de acciones observables. Este mecanismo no es muy intuitivo, puesto que un cambio en las probabilidades, o una restricción a un conjunto inadecuado de acciones, implicaría la desaparición de la posibilidad de esta interpretación de la elección como externa. Por otra parte, no resulta en absoluto evidente generalizar esta simulación a una ocurrencia arbitraria de la elección externa.

En nuestro lenguaje estos problemas desaparecen, puesto que disponemos de un operador paralelo asíncrono, y de un operador explícito de elección externa, ambos con una semántica adecuada.

Por simplicidad, omitiremos a lo largo del ejemplo las probabilidades en las apariciones de los operadores de elección externa y composición paralela, puesto que de hecho no son necesarias, ya que nunca aparecerán elecciones entre procesos que puedan ejecutar acciones comunes, como primera acción de los mismos. En el capítulo 6 presentaremos una justificación formal de que ello puede hacerse.

El sistema consiste de dos agentes, un transmisor y un receptor, que se comunican mediante mensajes. El transmisor envía un mensaje al receptor usando un canal no seguro, que puede perderlo con una probabilidad conocida p. Suponemos que si el mensaje original se pierde durante la transmisión, el canal lo reemplazará con un mensaje nulo λ, que enviará al receptor. Este tomará el mensaje de su canal de entrada, y si no es λ, lo enviará al exterior, devolviendo un mensaje de reconocimiento al transmisor, que enviará por un segundo canal. De nuevo, el canal podrá perder el mensaje (suponemos que con la misma probabilidad p, aunque esta suposición no es importante, pues los resultados que obtendremos podrían generalizarse fácilmente), en cuyo caso el transmisor recibirá un mensaje nulo λ'.

Si el transmisor recibe esta señal λ', repetirá la emisión, enviando de nuevo el mensaje original. Finalmente, cuando el transmisor reciba la señal de reconocimiento comenzará un nuevo ciclo del sistema.

El proceso PCSP que modela este protocolo es el proceso P definido como sigue:

Transmisor:

$$T = in \rightarrow T_1$$

$$T_1 = msg \rightarrow (ack \rightarrow T \Box \lambda' \rightarrow T_1)$$

Canal Primero:

$$C_1 = msg \rightarrow ([p]\lambda \rightarrow C_1 \sqcap [1-p]msg' \rightarrow C_1)$$

Canal Segundo:

$$C_2 = (ack' \rightarrow ([p]\lambda' \rightarrow C_2 \sqcap [1-p]ack \rightarrow C_2)) \Box (\lambda'' \rightarrow \lambda' \rightarrow C_2)$$

Receptor:

$$R = (msg' \rightarrow out \rightarrow ack' \rightarrow R) \Box (\lambda \rightarrow \lambda'' \rightarrow R)$$

Protocolo AUY:

$$P = T \|_{\{msg,ack,\lambda'\}} ((C_1 \|_{\emptyset} C_2) \|_{\{msg',ack',\lambda,\lambda''\}} R)$$

El comportamiento de este proceso se muestra en el sistema probabilístico de transiciones representado en la figura 2.

La definición de los procesos intermedios P_1 a P_7 viene dada por

$$\begin{aligned}
P_1 &= T_1 \|_{\{msg,ack,\lambda'\}} ((C_1 \|_{\emptyset} C_2) \|_{\{msg',ack',\lambda,\lambda''\}} R) \\
P_2 &= (ack \rightarrow T \Box \lambda' \rightarrow T_1) \|_{\{msg,ack,\lambda'\}} ((([p]\lambda \rightarrow C_1 \sqcap [1-p]msg' \rightarrow C_1) \|_{\emptyset} C_2) \\
&\quad \|_{\{msg',ack',\lambda,\lambda''\}} R) \\
P_3 &= (ack \rightarrow T \Box \lambda' \rightarrow T_1) \|_{\{msg,ack,\lambda'\}} ((C_1 \|_{\emptyset} C_2) \|_{\{msg',ack',\lambda,\lambda''\}} (out \rightarrow ack' \rightarrow R)) \\
P_4 &= (ack \rightarrow T \Box \lambda' \rightarrow T_1) \|_{\{msg,ack,\lambda'\}} ((C_1 \|_{\emptyset} C_2) \|_{\{msg',ack',\lambda,\lambda'\}} (\lambda'' \rightarrow R)) \\
P_5 &= (ack \rightarrow T \Box \lambda' \rightarrow T_1) \|_{\{msg,ack,\lambda'\}} ((C_1 \|_{\emptyset} C_2) \|_{\{msg',ack',\lambda,\lambda''\}} (ack' \rightarrow R)) \\
P_6 &= (ack \rightarrow T \Box \lambda' \rightarrow T_1) \|_{\{msg,ack,\lambda'\}} ((C_1 \|_{\emptyset} ([p]\lambda' \rightarrow C_2 \sqcap [1-p]ack \rightarrow C_2)) \\
&\quad \|_{\{msg',ack',\lambda,\lambda''\}} R) \\
P_7 &= (ack \rightarrow T \Box \lambda' \rightarrow T_1) \|_{\{msg,ack,\lambda'\}} ((C_1 \|_{\emptyset} (\lambda'' \rightarrow C_2)) \|_{\{msg',ack',\lambda,\lambda''\}} R)
\end{aligned}$$

Nuestro lenguaje nos permite razonar sobre diferentes propiedades del protocolo, tales como la fiabilidad (*reliability*) y el tiempo esperado de transmisión. Por

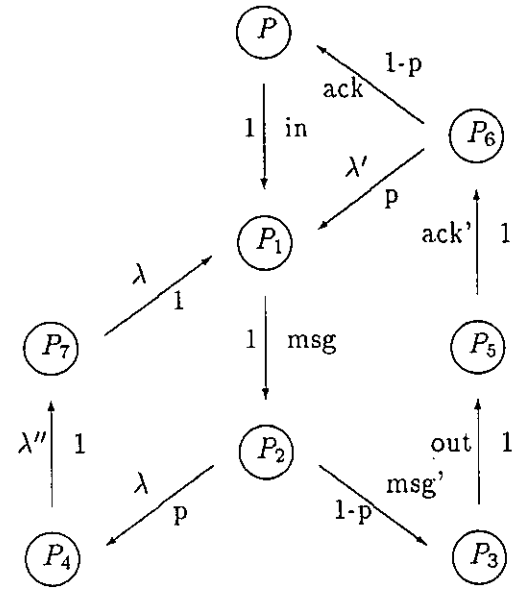


Figura 2: Sistema Probabilístico de Transiciones para el Protocolo AUY

ejemplo, es sencillo comprobar que la probabilidad con la que un ciclo termina tras un bucle simple es $(1-p)^2$. Sin embargo, el hecho más importante es que todos los elementos de la modelización del sistema son muy naturales, y no hemos necesitado ningún truco como el que se utiliza (¡y necesariamente!) en el modelo presentado en [GJS90].

Para finalizar, formalizaremos seguidamente la prueba de la seguridad del protocolo (cualquier mensaje será transmitido, y el reconocimiento se recibirá con probabilidad 1), y calcularemos la longitud media de un ciclo de transmisiones.

Cualquier computación de nuestro sistema comenzará con la transición

$$P \xrightarrow{in}_1 P_1$$

y un ciclo terminará cuando el estado P se recupere.

Desde P_1 tenemos las siguientes computaciones correspondientes a un bucle básico del protocolo:

$$P_1 \overset{s_1}{\Longrightarrow}_{p_1} P_1 \quad \text{Con } s_1 = \langle msg\, \lambda\, \lambda''\, \lambda' \rangle \text{ y } p_1 = p$$
$$P_1 \overset{s_2}{\Longrightarrow}_{p_2} P_1 \quad \text{Con } s_1 = \langle msg\, msg'\, out\, ack'\, \lambda' \rangle \text{ y } p_2 = p(1-p)$$
$$P_1 \overset{s_3}{\Longrightarrow}_{p_3} P \quad \text{Con } s_1 = \langle msg\, msg'\, out\, ack'\, ack \rangle \text{ y } p_3 = (1-p)^2$$

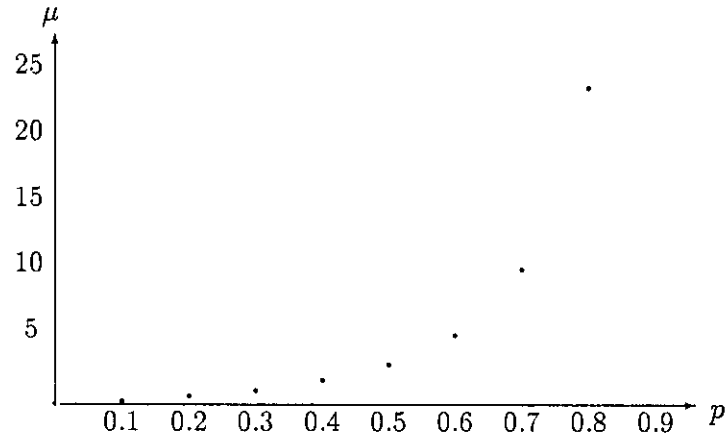


Figura 3: Tiempo medio de transmisión en protocolo AUY

Entonces, denotaremos con $f(n)$ la probabilidad de completar el ciclo tras n bucles fallidos, o lo que es lo mismo tras $n+1$ bucles básicos. A partir de las probabilidades de las computaciones previas, este valor puede ser calculado mediante

$$f(n) = (p + p(1-p))^n (1-p)^2 = (1-p)^2 p^n (2-p)^n$$

Entonces la probabilidad de finalizar el ciclo en tiempo finito viene dada por la expresión

$$\sum_{i=0}^{\infty} f(i) = (1-p)^2 \sum_{i=0}^{\infty} (2p-p^2)^i = \frac{(1-p)^2}{1-(2p-p^2)} = \frac{(1-p)^2}{(1-p)^2} = 1$$

Esto prueba que el protocolo es seguro, puesto que la probabilidad de tener una transmisión correcta es 1.

Definiremos el tiempo medio de emisión como el número medio de emisiones realizadas hasta que un reconocimiento se reciba por el transmisor. Entonces, este tiempo medio de emisión μ, viene dado por

$$\mu = \sum_{i=0}^{\infty} i\, f(i) = (1-p)^2 \sum_{i=0}^{\infty} i(2p-p^2)^i = (1-p)^2 \frac{2p-p^2}{(1-(2p-p^2))^2} =$$

$$(1-p)^2 \frac{p(2-p)}{(1-p)^4} = \frac{p(2-p)}{(1-p)^2}$$

Por otra parte, dado un canal de comunicación, decimos que es *seguro* si no puede aceptar indefinidamente inputs sin producir ninguna señal en el output. En el caso probabilístico, dicha propiedad se tendrá (con probabilidad 1) si la probabilidad con la que se ejecuta una computación de una secuencia de acciones de input que no contenga ningún output, tiende a cero al crecer la longitud de la computación. En concreto, dado el proceso

$$P = \mu X.in \rightarrow ([p]X \sqcap [1-p]out \rightarrow X)$$

podemos probar que es seguro para cualquier $p < 1$.

3.4 Semántica de trazas

En la presente sección presentaremos una semántica de trazas probabilísticas para PCSP, cuya presentación final será independiente de la anterior semántica operacional, realizándose en un marco denotacional, pero cuya justificación sí basaremos en dicha semántica operacional. Para ello comenzaremos viendo cómo definir de una forma razonable una semántica de trazas probabilísticas a partir de dicha semántica operacional.

3.4.1 Trazas Probabilísticas

Las trazas (ordinarias) de un proceso, son las posibles secuencias de acciones que el mismo puede realizar. En el caso probabilístico, una definición razonable de las mismas, a partir de nuestra semántica operacional, sería la siguiente:

$$Tr(P) = \{s \ / \ \exists Q \in Proc, p \in (0,1] : P \stackrel{s}{\Longrightarrow}_p Q\}$$

Pero ello supondría ignorar la información probabilística que las transiciones nos suministran. En consecuencia parece oportuno asociar a cada traza la probabilidad con la que la misma podrá ser realizada, dando lugar al concepto de traza probabilística.

Una primera aproximación a la definición formal de dicho concepto, nos llevaría a definir el conjunto de trazas probabilísticas de un proceso P como el conjunto

de pares $\langle s, p\rangle$ tales que existe un proceso P', s-derivado de P con probabilidad p. Sin embargo, ya hemos observado que una misma secuencia de acciones s puede aparecer varias veces en derivaciones distintas de un proceso, con las misma o distintas probabilidades asociadas. Además un mismo par $\langle s, p\rangle$ puede aparecer varias veces, ligado a sendos procesos s-derivados de P, iguales o no, por lo que no resultaría razonable hablar simplemente en este caso de *conjunto* de trazas, siendo lo oportuno hablar en cambio de un *multiconjunto* de trazas. Dicho multiconjunto vendrá dado por

$$MTr(P) = \{\langle s, p\rangle, s \in \Sigma^*,\ p \in (0,1]\ /\ \exists Q \in Proc,\ \text{tal que}\ P \stackrel{s}{\Longrightarrow}_p Q\}$$

donde cada par $\langle s, p\rangle$ aparecerá tantas veces como derivaciones tengamos de computaciones $P \stackrel{s}{\Longrightarrow}_p Q$.

Ejemplo 3.3 Algunos ejemplos de multiconjuntos de trazas de procesos son entonces los siguientes:

$$\begin{aligned}
MTr(STOP) &= \{(\langle\rangle, 1)\} \\
MTr(a \rightarrow STOP) &= \{(\langle\rangle, 1), (\langle a\rangle, 1)\} \\
MTr([\tfrac{1}{2}]a \rightarrow STOP \sqcap [\tfrac{1}{2}]b \rightarrow STOP) &= \{(\langle\rangle, 1), (\langle a\rangle, \tfrac{1}{2}), (\langle b\rangle, \tfrac{1}{2})\} \\
MTr([\tfrac{1}{2}]a \rightarrow STOP \sqcap [\tfrac{1}{2}]a \rightarrow STOP) &= \{(\langle\rangle, 1), (\langle a\rangle, \tfrac{1}{2}), (\langle a\rangle, \tfrac{1}{2})\} \\
MTr([\tfrac{1}{2}]a \rightarrow STOP \Box [\tfrac{1}{2}]b \rightarrow STOP) &= \{(\langle\rangle, 1), (\langle a\rangle, 1), (\langle b\rangle, 1)\} \\
MTr([1/3]a \rightarrow STOP \Box [2/3]b \rightarrow STOP) &= \{(\langle\rangle, 1), (\langle a\rangle, 1), (\langle b\rangle, 1)\}
\end{aligned}$$

□

En general, tenemos que:

1. $(\langle\rangle, 1) \in MTr(P)$ para todo proceso P.
2. $(st, p) \in MTr(P)$ implica que $\exists q,\ q \geq p,\ q \leq 1$ tal que $(s, q) \in MTr(P)$.

Los multiconjuntos de trazas probabilísticas, nos indican las posibles trazas, junto con las probabilidades de que aparezcan. Ahora bien, la aparición de la traza $(\langle a\rangle, \frac{1}{2})$ dos veces, nos indica que existen dos maneras distintas de ejecutar la acción a. Sin embargo, para un observador exterior no existirá la posibilidad de detectar la existencia de dichas transiciones diferentes, e incluso el funcionamiento

de cara al exterior de un proceso tal sería indistinguible de otro con la traza $(\langle a \rangle, 1)$. Por ello parece razonable que agrupemos en una sola las distintas trazas correspondientes a una misma secuencia de acciones. En concreto, a partir del multiconjunto de trazas de un proceso, definimos el siguiente *conjunto de trazas probabilísticas* del mismo:

$$Tr(P) = \{(s, \sum_{x_i \in X} x_i) \text{ Donde } X = \{x_i \ / \ (s, x_i) \in MTr(P)\}\}$$

Ejemplo 3.4 Los conjuntos de trazas correspondientes a los procesos del ejemplo anterior son entonces los siguientes:

$$\begin{aligned}
Tr(STOP) &= \{(\langle\rangle, 1)\} \\
Tr(a \rightarrow STOP) &= \{(\langle\rangle, 1), (\langle a \rangle, 1)\} \\
Tr([\tfrac{1}{2}]a \rightarrow STOP \sqcap [\tfrac{1}{2}]b \rightarrow STOP) &= \{(\langle\rangle, 1), (\langle a \rangle, \tfrac{1}{2}), (\langle b \rangle, \tfrac{1}{2})\} \\
Tr([\tfrac{1}{2}]a \rightarrow STOP \sqcap [\tfrac{1}{2}]a \rightarrow STOP) &= \{(\langle\rangle, 1), (\langle a \rangle, 1)\} \\
Tr([\tfrac{1}{2}]a \rightarrow STOP \Box [\tfrac{1}{2}]b \rightarrow STOP) &= \{(\langle\rangle, 1), (\langle a \rangle, 1), (\langle b \rangle, 1)\} \\
Tr([1/3]a \rightarrow STOP \Box [2/3]b \rightarrow STOP) &= \{(\langle\rangle, 1), (\langle a \rangle, 1), (\langle b \rangle, 1)\}
\end{aligned}$$

□

Proposición 3.1 El concepto de traza probabilística está bien definido, es decir, las sumas de probabilidades $\sum_{x \in X} x_i$ que pueden aparecer al definir las trazas de un proceso no exceden nunca de 1.

Demostración: Viene dada por el hecho de que las posibles transiciones distintas correspondientes a una misma traza, provendrán de elecciones, tanto internas, como externas, como provenientes de operadores paralelos, que el proceso tendrá que resolver. Ahora bien, en todos los casos las probabilidades de las dos distintas computaciones a las que cada elección dará lugar están ponderadas por factores cuya suma es uno, por lo que un sencillo razonamiento por inducción nos conduciría a la prueba del resultado deseado. □

3.4.2 Dominio de procesos

La definición de la semántica de los procesos por medio de sus conjuntos de trazas, nos conduciría a la siguiente relación de equivalencia:

Definición 3.6 Sean A, B multiconjuntos de trazas. Decimos que $A \equiv B$ si y sólo si $Tr(A) = Tr(B)$. □

De esta manera, podemos definir un proceso (semántico) como cada clase de equivalencia correspondiente a la relación anterior.

Definición 3.7 Definimos un proceso como la clase de equivalencia formada por los multiconjuntos de trazas equivalentes al conjunto de pares $\{(s,p) : s \in \Sigma^*, p \in (0,1]\}$, de manera que se cumplan las siguientes condiciones:

1. $\forall P\ (\langle\rangle, 1) \in P$
2. $\forall s,t \in \Sigma^*, \forall p \in (0,1]\ \exists p' \geq p$ tal que $(st,p) \in P \Rightarrow (s,p') \in P$ □

Llamaremos *Proc* al conjunto de todos los procesos sobre el alfabeto Σ. En general, cuando hablemos de un conjunto de trazas, estaremos referenciando a la clase de los multiconjuntos equivalentes a él.

Por otra parte, para que el conjunto de procesos pueda ser considerado un dominio semántico, debemos definir un preorden sobre el mismo.

Definición 3.8 (Orden entre procesos)
Sean P y Q procesos, según la definición anterior. Decimos que $P \sqsubseteq Q$ sii para todo $(s,p) \in P$ existe $p' \geq p$, tal que $(s,p') \in Q$. □

Proposición 3.2 La relación así definida es de orden.
Demostración:

- $P \sqsubseteq P$. Trivial.
- $P \sqsubseteq Q$ y $Q \sqsubseteq P \Rightarrow P = Q$.

$$\begin{aligned}
(s,p) \in P &\Rightarrow \exists p' \geq p (s,p') \in Q && \text{pues } P \sqsubseteq Q \\
&\Rightarrow \exists q \geq p' (s,q) \in P && \text{pues } Q \sqsubseteq P \\
&\Rightarrow q = p && \text{pues } (s,p),(s,q) \in P \\
&\Rightarrow p = p' \\
&\Rightarrow (s,p) \in Q
\end{aligned}$$

$$(s,p) \in Q \Rightarrow (s,p) \in P \text{ análogamente}$$

- $P \sqsubseteq Q \sqsubseteq R \Rightarrow P \sqsubseteq R$

$$\begin{aligned}
(s,p) \in P &\Rightarrow \exists p' \geq p \ t.q. \ (s,p') \in Q \ \text{ pues } P \sqsubseteq Q \\
&\Rightarrow \exists q \geq p' \ \ t.q. \ (s,q) \in R \ \text{ pues } Q \sqsubseteq R \\
&\Rightarrow \exists q \geq p \ \ t.q. \ (s,q) \in R \\
&\Rightarrow P \sqsubseteq R
\end{aligned}$$

□

Definición 3.9 Sea $f : Proc \longrightarrow Proc$ una función entre procesos. Tenemos las siguientes definiciones:

1. f es **monótona** si cumple que:

$$P \sqsubseteq Q \Rightarrow f(P) \leq f(Q)$$

2. Una **cadena** es una secuencia de procesos

$$\{P_i, i \geq 0\} \ t.q. \ \forall i \ P_i \sqsubseteq P_{i+1}$$

3. El **límite** de una cadena denotado con $\sqcup P_i$ es un proceso con:

$$P_i = \{(s_{ij}, p_{ij})\} \forall i \geq 0 \Rightarrow \sqcup P_i = \{(s_j, p_j)\} \text{ con } (s_j, p_{ij}) \in P_i \ \ \forall i \geq i0$$

$$\text{donde } p_j = \lim_{i \geq i_0} p_{ij}$$

Este límite existe, pues $0 \leq p_{0j} \leq p_{1j} \leq \ldots \leq p_{ij} \leq \ldots \leq 1$.

Es decir

$$P_i = \{(s,p) / \exists j_0 \ t.q. \ \forall j \geq j_0 \Rightarrow (s,p_j) \in P_j, \text{ y } \lim_{j \to \infty} p_j = p\}$$

4. f es **continua** si

$$f(\sqcup P_i) = \sqcup f(P_i)$$

□

Teorema 3.1 $(Proc, \sqsubseteq)$ es un orden parcial completo (cpo).
Demostración:

- Tiene elemento mínimo.

  $\forall P \in Proc \quad STOP \sqsubseteq P.$

- Toda cadena tiene cota superior mínima.

  Sea $\{P_i\}_{i \in I}$ una cadena. Entonces $\sqcup P_i$ es la cota superior mínima:

  1. $\sqcup P_i$ es cota superior

     $(s_i, p_i) \in P_i \Rightarrow (s_i, p) \in \sqcup P_i$ puesto que $p \geq p_i$

     $\Rightarrow P_i \sqsubseteq \sqcup P_i$

  2. $\sqcup P_i$ es mínima

     Sea un proceso Q tal que $P_i \sqsubseteq Q$ para todo i.

     Entonces:

     $$(s,p) \in \sqcup P_i \Rightarrow \exists j_0 \ / \ (s, p_{j0}) \in P_{j0}, \text{ con } p_{j0} \leq p$$
     $$\Rightarrow \exists q \geq p_{j0} \ / \ (s,q) \in Q$$

     Además, para todo $j \geq j_0 p_j \leq p$ pues $\lim_j p_j = p$

     Como $q \geq p_j$ para todo $j \Rightarrow q \geq p$

     Luego $Q \sqsupseteq \sqcup P_i$

□

3.4.3 Semántica denotacional

Veamos seguidamente la definición de la semántica de los procesos correspondientes a la sintaxis definida, entendiendo un proceso a nivel sem;antico como un conjunto de trazas probabilísticas.

STOP y PREFIJO Definimos las trazas probabilísticas de los procesos STOP y prefijo para una determinada acción a en la manera siguiente:

$$\begin{aligned} STOP &= \{\langle\rangle, 1\} \\ a \rightarrow P &= \{(\langle\rangle, 1)\} \cup \{(as, p)/(s, p) \in P\} \end{aligned}$$

Ejemplo 3.5 Sea el proceso $P = a \rightarrow P$, un proceso que siempre hace la acción a, con probabilidad 1. Es decir el proceso $\mu X.(a \rightarrow X)$. Su definición semántica será:

$$P = \{(\langle\rangle, 1), (\langle a\rangle, 1), (\langle aa\rangle, 1), \ldots\}$$

□

ELECCIÓN INTERNA

$$\begin{aligned} [p]P \sqcap [1-p]Q &= \{(s, q) \ / \ (s, q') \in P \quad q = pq'\} \\ &\cup \{(s, q) \ / \ (s, q') \in Q \quad q = (1-p)q'\} \end{aligned}$$

Ejemplo 3.6 Sea el proceso $P = [\frac{1}{2}](a \rightarrow P) \sqcap [\frac{1}{2}]STOP$. Este proceso ejecuta un número finito de acciones a y después se para. Su definición semántica es:

$$P = \{(\langle\rangle, 1), (\langle a\rangle, \frac{1}{2}), (\langle aa\rangle, 1/4), (\langle aaa\rangle, 1/8), \ldots\}$$

Aún cuando en el modelo de trazas están excluidas las trazas infinitas, en este caso observamos que ello estaría doblemente justificado, pues la probabilidad de que el proceso ejecute una cantidad arbitrariamente grande de acciones a es 0. □

ELECCIÓN EXTERNA

$$\begin{aligned} [p]P \Box [1-p]Q &= \{(s, q) \ / \ (s, q) \in P \text{ y } (s, q) \notin Q\} \\ &\cup \{(s, q) \ / \ (s, q) \in Q \text{ y } (s, q) \notin P\} \\ &\cup \{(s, q') \ / \ (s, q') \in P \text{ y } (s, q') \in Q \text{ y } q = pq'\} \\ &\cup \{(s, q') \ / \ (s, q') \in P \text{ y } (s, q') \in Q \text{ y } q = (1-p)q'\} \end{aligned}$$

Ejemplo 3.7 Sea el proceso $P = [\frac{1}{2}](a \rightarrow P)\Box[\frac{1}{2}]STOP$. Este proceso podrá ejecutar cualquier secuencia de acciones a con probabilidad 1. Su definición semántica es:

$$P = \{(\langle\rangle, 1), (\langle a\rangle, 1), (\langle aa\rangle, 1), \ldots\}$$

Por lo tanto, este proceso será equivalente al dado por $P = a \rightarrow P$.

Si en cambio consideramos el proceso dado por $P = [\frac{1}{2}](a \rightarrow P)\Box[\frac{1}{2}]a \rightarrow STOP$, tenemos que su semántica viene dada por:

$$P = \{(\langle\rangle, 1), (\langle a\rangle, \frac{1}{2}), (\langle aa\rangle, 1/4), \ldots\}$$

por lo que es equivalente al proceso dado por $P = a \rightarrow ([\frac{1}{2}](a \rightarrow P) \sqcap [\frac{1}{2}]STOP)$. □

COMPOSICIÓN PARALELA

$$[p]P\|_A[1-p]Q = \{(s,q)/s \in mezcla(s_1, s_2)$$

$$\text{y } q = pm(P, Q, A, s, p) \text{ y } (s_1, r) \in P \text{ y } (s_2, r') \in Q\}$$

Donde tenemos las siguientes definiciones de funciones, por un lado el conjunto de las posibles mezclas entre dos trazas viene dado por la función

$$mezcla : \Sigma^* \times \Sigma^* \longrightarrow p\Sigma^*$$

$$\begin{aligned} mezcla(\langle\rangle, s) &= mezcla(s, \langle\rangle) = \{s\} \\ mezcla(\langle a\rangle s, \langle b\rangle t) &= \{\langle a\rangle u \ / \ u \in mezcla(s, \langle b\rangle t)\} \\ &\quad \cup \{\langle b\rangle u \ / \ u \in mezcla(\langle a\rangle s, t)\} \end{aligned}$$

Por otro lado, dada una posible mezcla entre sendas trazas de los procesos P y Q, la probabilidad de que una traza mezcla de ambas sea ejecutada por el proceso $[p]P\|_A[1-p]Q$ viene dada por la función

$$pm : Proc \times Proc \times p\Sigma \times \Sigma^* \times (0,1] \longrightarrow [0,1]$$

$$
\begin{aligned}
&pm(P,Q,A,\langle\rangle,p) \;=\; 1\\
&pm(P,Q,A,\langle a\rangle s,p) \;=\\
&\quad\begin{cases}
qq'\cdot pm(P/a,Q/a,A,s,p) & \text{Si } a\in A\\
(pp'q+p(1-p))\cdot pm(P/a,Q,A,s,p) \;+ & \\
(pp'(1-q)+(1-p)q)\cdot pm(P,Q/a,A,s,p) & \text{Si } a\notin A
\end{cases}
\end{aligned}
$$

Donde q y q' son los valores que verifican que $(\langle a\rangle,q)\in P$, $(\langle a\rangle,q')\in Q$, tomándose como 0 si no existen q o q' con tales características.

Por último, la función P/a, que define el comportamiento de un proceso tras la ejecución de una acción, se define mediante:

$$
P/a=\begin{cases}
\{(s,q)\ /\ (\langle a\rangle s,q')\in P \text{ y } q=q'/p\} & \text{si } (\langle a\rangle,p)\in P\\
STOP & \text{si } \nexists p>0\ :\ (\langle a\rangle,p)\in P
\end{cases}
$$

Ejemplo 3.8 Vemos a continuación algunos ejemplos de composición de procesos en paralelo, junto con lo que usualmente se denomina su expansión no-determinista.

$$
\begin{aligned}
[\tfrac{1}{2}]a\to STOP\|_{\{a\}}[\tfrac{1}{2}]a\to STOP &\;=\; a\to STOP\\
[\tfrac{1}{2}]a\to P\|_{\{a,b\}}[\tfrac{1}{2}]b\to Q &\;=\; STOP
\end{aligned}
$$

En este caso tenemos que todas las trazas no vacías de $a\to P$ comienzan con a, y las de $b\to Q$ con b, por lo que no pueden sincronizar.

Dados los procesos $P=[\frac{1}{2}]a\to P\sqcap[\frac{1}{2}]b\to P$ y $Q=a\to Q$, tenemos que

$$
[\frac{1}{2}]P\|_{\{a\}}[\frac{1}{2}]Q=[\frac{1}{2}](a\to([\frac{1}{2}]P\|_{\{a\}}[\frac{1}{2}]Q))\sqcap[\frac{1}{2}]STOP
$$

Dados los procesos $P=[\frac{1}{2}](a\to P)\Box[\frac{1}{2}]([\frac{1}{2}]a\to P\sqcap[\frac{1}{2}]b\to P)$ y $Q=a\to Q$, tenemos que

$$
[\frac{1}{2}]P\|_{\{a,b\}}[\frac{1}{2}]Q=a\to([\frac{1}{2}]P\|_{\{a\}}\frac{1}{2}]Q)=Q
$$

□

El hecho de que *Proc* sea un CPO, nos ofrece otra posibilidad a la hora de definir procesos, que nos permitirá describir comportamientos infinitos, aumentando la potencia de nuestro lenguaje, que hasta ahora sólo sirve para describir procesos de comportamiento finito.

Si $f : Proc \longrightarrow Proc$, es una función entre procesos, podemos definir un proceso X como aquel que satisface la ecuación $X = f(X)$. Podemos garantizar que una ecuación tal tendrá al menos una solución, siempre que la función en cuestión sea continua. Aún así, en ocasiones, estas ecuaciones recursivas no tendrán una única solución, por lo que de cara a definir unívocamente el proceso, deberemos quedarnos con $\mu X.f(X)$ que es, por definición, el menor proceso (respecto del orden entre funciones inducido por el orden sobre procesos) que satisface la ecuación $X = f(X)$.

Siendo f una función continua entre procesos, dicho proceso está bien definido como asegura la siguiente

Proposición 3.3 . Si $f : Proc \longrightarrow Proc$ es continua, $\mu X.f(X)$ está bien definido.

Demostración: Consideramos $\mu X.f(X)$ como el límite de la cadena:

$$STOP, f(STOP), f(f(STOP)), \ldots, f^i(STOP), \ldots$$

Donde, en general, f^i denota la aplicación reiterada de la función f un número i de veces. Entonces tenemos

$$f(\sqcup_{i\geq 0} f^i(STOP)) \;=\; \sqcup_{i\geq 1} f^i(STOP) \;=\; \sqcup_{i\geq 0} f^i(STOP)$$

y si $X = f(X)$ tenemos en particular, puesto que $STOP \sqsubseteq X$, que $f(STOP) \sqsubseteq f(X) = X$, y por tanto, reiterando el razonamiento, obtenemos que

$$\forall i \geq 0 \;\; f^i(STOP) \sqsubseteq X$$

por lo que pasando al límite

$$\sqcup_{i\geq 0} f^i(STOP) \sqsubseteq X$$

□

OCULTACION

Como cuando definíamos las transiciones de cada proceso en el marco de la semántica operacional, hemos dejado para el final el operador de ocultación, pues nos vuelve a plantear un problema análogo al que en aquel caso nos encontrábamos.

Tampoco la semántica de trazas es suficientemente potente para distinguir una serie de procesos que deben ser distintos, pero que sin embargo resultan tener las mismas trazas. Este problema aparece también en la versión clásica de CSP sin probabilidades (ver [Hoa85]) en la que precisamente se necesita distinguir entre los dos operadores de elección si se desean distinguir las situaciones de bloqueo (*deadlock*). En nuestra versión probabilística, la semántica de trazas tampoco nos permite definir coherentemente el operador de ocultación, pues este operador debe tener (intuitivamente) comportamientos diferentes sobre procesos que tienen las mismas trazas, pero distintos formatos de elección.

Ejemplo 3.9 Consideremos los siguientes procesos

$$\begin{aligned} P = &\ [\tfrac{1}{2}]([\tfrac{1}{2}]a \to STOP \,\Box\, [\tfrac{1}{2}]b \to STOP) \sqcap [\tfrac{1}{2}]STOP \\ Q = &\ [\tfrac{1}{2}]a \to STOP \sqcap [\tfrac{1}{2}]b \to STOP \end{aligned}$$

Ambos procesos tienen las mismas trazas, que en concreto vienen dadas por

$$P = Q = \{(\langle\rangle, 1), (\langle a\rangle, \frac{1}{2}), (\langle b\rangle, \frac{1}{2})\}$$

Por tanto, ambos procesos serán iguales bajo la semántica de trazas. Sin embargo, nosotros pretendemos que sobre cada uno de ellos, al ocultar la acción a, los comportamientos se vuelvan distintos, pues en el caso de P, aún cuando se haya optado en principio por el primer argumento de la elección interna, todavía el medio puede decidir internamente seleccionar la ejecución de la acción a, descartando por ello la ejecución de la acción b, con lo cual disminuye su probabilidad; mientras que en el caso de Q mantenemos la probabilidad $\frac{1}{2}$ de ejecutar la acción b. Más exactamente, buscamos entonces un modelo en el que satisfaga que:

$$\begin{aligned} P\backslash(a, \tfrac{1}{2}) &= [\tfrac{1}{2}]([\tfrac{1}{2}]STOP \sqcap [\tfrac{1}{2}]b \to STOP) \sqcap [\tfrac{1}{2}]STOP \\ &= [\tfrac{3}{4}]([\tfrac{1}{2}]STOP \sqcap [\tfrac{1}{4}]b \to STOP) \\ Q\backslash(a, \tfrac{1}{2}) &= [\tfrac{1}{2}]STOP \sqcap [\tfrac{1}{2}]b \to STOP \end{aligned}$$

O lo que es lo mismo, que las trazas probabilísiticas de estos procesos sean

$$\begin{aligned} P\backslash(a,\tfrac{1}{2}) &= \{(\langle\rangle,1),(\langle b\rangle,1/4)\} \\ Q\backslash(a,\tfrac{1}{2}) &= \{(\langle\rangle,1),(\langle b\rangle,\tfrac{1}{2})\} \end{aligned}$$

Ahora bien, evidentemente no podemos pretender tener un modelo composicional en el que ocurra $P = Q$ y sin embargo $P\backslash(a,\frac{1}{2}) \neq Q\backslash(a,\frac{1}{2})$. □

La semántica de trazas está presentada en [JS90] de una forma similar, aunque en dicho trabajo se define la misma a partir de la semántica operacional, definiendo la función de trazas en la forma

$$Tr : Proc \longrightarrow (\Sigma^* \rightarrow [0,1])$$

$$Tr(P)(s) = \sum_{P' \in Proc} \mu_s(P,P')$$

donde $\mu_s(P,P')$ es el valor tal que existe una computación con:

$$C = P \xrightarrow{s}_{\mu_s(P,P')} P'$$

Capítulo 4

Semántica de pruebas

La información que es posible obtener de un proceso cuando al mismo le es ofrecida para su ejecución una sóla acción, tal y como hemos hecho al estudiar la semántica de trazas, no nos basta para caracterizar por completo a dicho proceso.

Efectivamente, como ha sido indicado previamente, la mera probabilidad de que una acción sea o no sea aceptada, no es indicativa del grado de elección que en ese instante el medio exterior puede ejercer en el control del sistema. Esto es particularmente grave a la hora de definir el comportamiento de la operación de ocultación. Si queremos (como parece natural) que nuestro modelo sea consistente (tanto como ello sea posible) con el modelo clásico de CSP sin probabilidades, este operador debe comportarse de una forma diferente, según sea el grado de elección que se tenga al ocultar una acción. En concreto, al producirse la ocultación de una acción, la facultad de decidir la ejecución de la misma cuando aparezca como opción de una elección externa, debe pasar al sistema, con lo que dicha elección se convertirá en una elección entre interna y externa.

En consecuencia necesitamos una semántica más potente, que nos permita conocer, en la medida en que ello sea necesario, el estado concreto en que en cada momento se encuentra un proceso. Nos ha parecido que el marco más apropiado para presentar una semántica tal es el de las semánticas de pruebas (*testing semantics*), basadas en la idea de testear el funcionamiento de un proceso, ofreciéndole en cada momento durante su ejecución, un conjunto de acciones visibles entre las que ha

de seleccionar aquélla que será ejecutada a continuación. El proceso se repetirá un determinado número de veces, hasta que o bien el observador se dé por satisfecho, en cuyo caso se dice que el proceso *ha pasado la prueba*, o bien se llegue a un estado en el que el proceso no puede ejecutar ninguna de las acciones que en ese momento se le ofrecen, en cuyo caso decimos que la prueba *ha fallado.*

La semántica de pruebas, basada en la denominada *equivalencia de pruebas*, fué desarrollada por De Nicola y Hennessy en 1984 (ver [NH84]) como una alternativa a la equivalencia observacional clásica de CCS, basada en el concepto de *bisimulación*, por entender que esta última era demasiado restrictiva. La prueba de que una semántica tal se puede describir por medio de unos ciertos árboles, a los que se denomina *árboles de aceptación*, y de que por tanto se trata a la postre de una nueva presentación de la *semántica de fallos* de CSP [Bro83], aparece primeramente en [Hen85], para adquirir un tono más depurado y detallado en [Hen88].

La idea de ampliar los árboles de aceptación mediante la posibilidad de chequear también la posibilidad del rechazo de ciertas acciones aparece en [Phi85] lo que permite distinguir una mayor cantidad de procesos.

En nuestro caso parece razonable que, puesto que nos movemos en un marco probabilístico, dotemos al conjunto de acciones ofrecidas al proceso en cada momento, de una distribución de probabilidad, de forma que la acción que definitivamente será la elegida se seleccione de entre el subconjunto de acciones que resulten en dicho momento viables, atendiendo a dicha distribución.

La idea de ampliar los tests mediante el uso de una distribución de probabilidad sobre las diferentes ramas de los árboles que los describen, ya aparece, aunque en el marco de la definición de una semántica de bisimulación, en [LS89].

4.1 Arboles probabilísticos de observación

En primer lugar procederemos a definir el tipo de pruebas a las cuales someteremos los procesos con el objetivo de detectar su comportamiento. Este tipo de pruebas tendrán forma de *árboles probabilísticos de observación*, los cuales son aquéllos que quedan descritos mediante la siguiente sintaxis:

Definición 4.1 (Arboles Probabilísticos de Observación)
Definimos el conjunto de árboles probabilísticos de observación, que denominaremos *PTEST* como sigue:

$$t ::= \omega \mid 0 \mid a.t \mid \sum_{i=1}^{n}[p_i]a_i.t_i$$

donde $a_i \in \Sigma$, $a_i \neq a_j$ para $i \neq j$, $p_i \in [0,1]$ y $\sum_{i=1}^{n} p_i = 1$. □

Seguidamente, procederemos a describir el significado de los árboles que se pueden generar con la sintaxis anterior. En primer lugar, mediante el árbol ω, denotamos el test *éxito*, es decir un test que siempre será aceptado. Por el contrario, el árbol 0 denota el test *fracaso* que representa un test imposible de ser pasado con éxito. El árbol $a.t$ representa un test secuencial, que ofrece en principio en exclusivo la acción a, y en caso de que ésta se acepte, pasa a comportarse como el test t. Por último con $\sum_{i=1}^{n}[p_i]a_i.t_i$ representamos un test que ofrece las acciones a_i para su aceptación. Si todas ellas son aceptadas, una de ellas será seleccionada atendiendo a las probabilidades p_i. Por el contrario, si sólo un subconjunto de las mismas es aceptado, las probabilidades de las restantes se distribuyen ponderadamente entre las aceptadas, es decir, proporcionalmente a sus propias probabilidades; tras ello se escoge una de las acciones posibles, y se continúa con el test t_i correspondiente a la misma.

Obsérvese que hemos permitido que la probabilidad de alguna rama del árbol sea nula, con el objetivo de cubrir los casos en que las mismas tengan prioridad mínima de ser seleccionadas, lo que hemos modelado, siguiendo a [SS90], mediante una probabilidad ínfima. Más exactamente, tendremos que tales ramas serán seleccionadas sólo en el caso de que ninguna otra con probabilidad positiva pueda serlo. Cuando éste sea el caso, y por tanto todas las ramas aceptadas tengan probabilidad nula, la elección entre ellas se hará atendiendo a una distribución uniforme.

Por otra parte, es claro que el test secuencial $a.t$ es idéntico al test ramificado $\sum_{i=1}^{n}[p_i]a_i.t_i$ correspondiente al caso en el que $n = 1, p_1 = 1, a_1 = a$ y $t_1 = t$, pero hemos preferido mantener las dos presentaciones sintácticas del mismo, para evitar el acarreo de probabilidades cuando las mismas resulten irrelevantes.

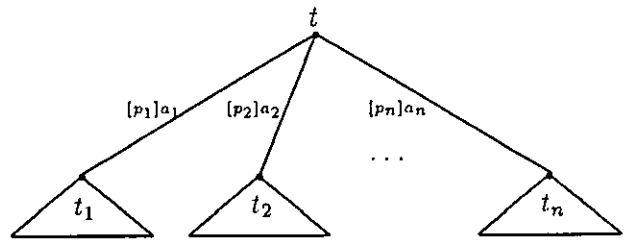


Figura 4: Arbol de observación $t = \sum_{i=1}^{n}[p_i]a_i.t_i$

A fin de facilitar el manejo de los mismos, definiremos las siguientes operaciones sobre los árboles probabilísticos de observación:

Primer nivel. Si $t \in PTEST$, definimos su *primer nivel*, que será un test de a lo sumo un sólo piso, y denotaremos por t_0, en la forma:

$$\begin{aligned} \omega_0 &= \omega \\ 0_0 &= 0 \\ (a.t)_0 &= a.\omega \\ (\sum_{i=1}^{n}[p_i]a_i.t_i)_0 &= \sum_{i=1}^{n}[p_i]a_i.\omega \end{aligned}$$

Arbol tras una acción. Sea $t \in PTEST, a \in \Sigma$, definimos el árbol t tras a, que denotaremos por t/a, mediante:

$$\begin{aligned} \omega/a &= 0 \\ 0/a &= 0 \\ (b.t)/a &= \begin{cases} t & si\ b = a \\ 0 & si\ b \neq a \end{cases} \\ (\sum_{i=1}^{n}[p_i]a_i.t_i)/a &= \begin{cases} t_j & si\ a_j = a \\ 0 & si\ \nexists j \text{ con } a_j = a \end{cases} \end{aligned}$$

Conjunto inicial. LLamaremos *conjunto inicial* de acciones ofrecido por un test t, y lo denotaremos por $Ft(t)$, al conjunto de acciones del primer nivel del

mismo, definido por

$$\begin{aligned} Ft(\omega) &= \emptyset \\ Ft(0) &= \emptyset \\ Ft(a.t) &= \{a\} \\ Ft(\sum_{i=1}^{n}[p_i]a_i.t_i) &= \{a_i \mid i = 1,\ldots,n\} \end{aligned}$$

Restricción a una acción. Sea $t \in PTEST, a \in \Sigma$; definimos la *restricción* de t sobre la acción a, que denotaremos por t_a, en la forma

$$t_a = \begin{cases} a.(t/a) & \text{si } a \in Ft(t) \\ 0 & \text{si } a \notin Ft(t) \end{cases}$$

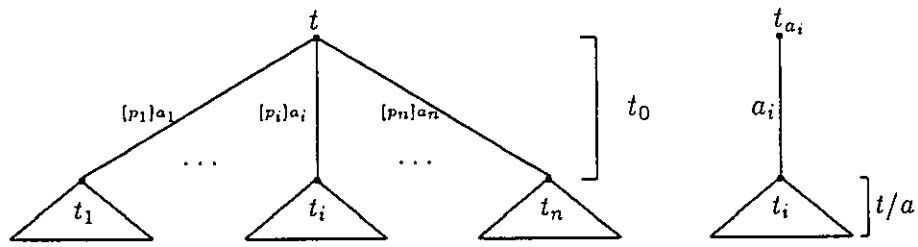
Profundidad. Sea $t \in PTEST$, definimos la *profundidad* del test t, que denotamos por $Prof(t)$, en la forma

$$\begin{aligned} Prof\ (\omega) &= 0 \\ Prof\ (0) &= 1 \\ Prof\ (a.t) &= 1 + Prof(t) \\ Prof\ (\sum_{i=1}^{n}[p_i]a_i.t_i) &= 1 + Max(Prof(t_1),\ldots,Prof(t_n)) \end{aligned}$$

Por conveniencia, consideramos que la profundidad de 0 es uno, lo que queda justificado por el hecho de que dicho test puede ser presentado de manera equivalente, en la forma $a.\omega$ siendo a una acción cualquiera que no pueda ser ejecutada por ningún proceso.

4.2 Paso de un test

En el caso no-determinista se definía el paso de un test por parte de un proceso, indicando cuales son los tests que serían pasados con seguridad, y cuales los que podrían ser pasados en alguna ocasión. En nuestro marco probabilístico, parece oportuno hacerlo, indicando para cada proceso P, y cada árbol de observación t, cual es la probabilidad con la que el proceso P aceptará (o pasará) el test definido



$$Ft(t) = \{a_1, a_2, \ldots, a_n\}$$

Figura 5: Operaciones sobre árboles

por el árbol de observación t. Dicha probabilidad la denotaremos en la forma $P \mid t$. Como es usual, de cara a la presentación de la correspondiente definición formal, procederemos por inducción estructural, atendiendo al constructor de procesos que aparece en la cabeza del proceso a investigar.

4.2.1 Stop

El proceso $STOP$ es incapaz de realizar ninguna acción, por tanto sólamente puede aceptar el test de éxito, debiendo rechazar cualquier otro.

$$STOP \mid t = \begin{cases} 1 & \text{si } t = \omega \\ 0 & \text{si } t \neq \omega \end{cases}$$

4.2.2 Prefijo

Puesto que el proceso $a \rightarrow P$ acepta la ejecución de la acción a para después comportarse como el proceso P, su semántica vendrá dada por una función que aceptará todo árbol que contenga a la acción a en su primer nivel, con una probabilidad igual a aquélla con la que el proceso P acepte el árbol t/a. Si la acción a no está en el primer nivel del test, el mismo será rechazado.

$$(a \rightarrow P) \mid t = \begin{cases} 0 & \text{si } a \notin Ft(t) \text{ y } t \neq \omega \\ P \mid t/a & \text{si } a \in Ft(t) \\ 1 & \text{si } t = \omega \end{cases}$$

4.2.3 Elección interna

Sean P y Q dos procesos, el proceso $[p]P \sqcap [1-p]Q$ realizará una elección interna entre ellos, de acuerdo con las probabilidades indicadas. Esta elección no estará influída por el medio en que se encuentra inmerso el proceso.

Así pues, este proceso aceptará las observaciones que P o Q acepten, pero ponderadas por los respectivos factores p y $1-p$.

$$([p]P \sqcap [1-p]Q) \mid t = p \cdot (P \mid t) + (1-p) \cdot (Q \mid t)$$

4.2.4 Elección externa

Asímismo, el proceso $[p]P \Box [1-p]Q$ se comportará bien como P o como Q, pero en este caso será el observador exterior quien jugará el principal papel en la elección del comportamiento. Así, una acción podrá ser rechazada sólo si es rechazada simultáneamente por los dos procesos componentes.

Cuando ofrecemos al proceso no una sóla acción, sino un conjunto de ellas, desearíamos en principio que el comportamiento fuese similar. Es decir, el rechazo del conjunto de acciones se producirá si y sólo si todas las acciones del conjunto son simultáneamente rechazadas por los dos procesos argumentos de la elección.

Una primera dificultad en este caso consiste en que en el conjunto ofrecido podrían aparecer distintas acciones que son aceptadas, unas por un proceso y otras por el otro. Consideremos por ejemplo el siguiente proceso:

$$[\frac{1}{2}]a \rightarrow STOP \Box [\frac{1}{2}]b \rightarrow STOP$$

Si ofrecemos al mismo el test $[\frac{1}{2}]a.\omega + [\frac{1}{2}]b.\omega$, tendríamos que cada una de las acciones ofrecidas sería aceptada por un proceso y no por el otro. En consecuencia deberíamos comenzar decidiendo cual es la acción que será ejecutada, lo que conllevará una elección sobre cual será el proceso que continuará ejecutándose.

Es cierto, sin embargo, que en este ejemplo concreto, la situación planteada no resulta problemática a la postre, puesto que lo que en último término nos importa es si el test es o no aceptado, y en este caso lo sería en cualquier caso, sea a o b

la acción que finalmente se ejecutase. En general, si el árbol tiene un sólo nivel, el comportamiento de la elección externa es fácilmente definible, indicando que se aceptará un test si y sólo si alguno de sus argumentos lo acepta, lo que queda formalizado en la forma siguiente:

Definición 4.2 (Paso de un test de profundidad 1)
Sean P, Q dos procesos, y t_0 un test de profundidad uno. La probabilidad de que el proceso $[p]P \Box [1-p]Q$ acepte t_0 viene dada por

$$[p]P \Box [1-p]Q \mid t_0 = (P \mid t_0)(1-(Q \mid t_0))+(Q \mid t_0)(1-(P \mid t_0))+(P \mid t_0)(Q \mid t_0)$$

□

Ahora bien, en el caso de que el test pueda continuar o fallar tras la ejecución de alguna de las acciones que ofrece, sí que necesitamos precisar qué acción será seleccionada, puesto que de ello podrá depender la aceptación global del test. Consideremos al efecto la siguiente variante sencilla del proceso del ejemplo anterior:

$$[\frac{1}{2}]a \rightarrow c \rightarrow STOP \ \Box \ [\frac{1}{2}]b \rightarrow STOP$$

Si aplicamos al mismo el test $[\frac{1}{2}]a.c.\omega + [\frac{1}{2}]b.c.\omega$, no tiene el mismo efecto en absoluto el que seleccionemos la acción a o la acción b como primera acción a ejecutar, puesto que en un caso el proceso tras dicha ejecución estará en disposición de aceptar el resto del test, mientras en el otro, ello no será posible.

Ahora bien, como ya hemos comentado, las probabilidades asociadas a este operador de elección externa no tienen sentido si los conjuntos de acciones que puede aceptar cada componente son disjuntos, tal y como sucede en este caso. Así pues, para resolver la elección será necesario recurrir a las probabilidades asociadas al test, que será quien decida (probabilísticamente) desde el exterior la acción a ejecutar.

Este mismo procedimiento de selección de la acción a ejecutar se aplica también en el caso de que los conjuntos de acciones aceptados por ambos procesos no sean disjuntos. Pero en este caso, si la acción seleccionada puede ser ejecutada por ambos procesos, todavía tenemos que decidir cual de ellos lo hará, determinando

así la continuación posterior. Para ello, debemos hacer uso en este caso de las probabilidades asociadas a los argumentos del operador de elección externa.

Así, nuestra idea para definir la probabilidad de aceptación de un test t por parte del proceso $[p]P\Box[1-p]Q$ consiste en distinguir las contribuciones de todas las acciones ofrecidas en el primer nivel del test. Ello nos conduce a la siguiente definición esquemática:

$$[p]P\Box[1-p]Q \mid t = \sum_{a \in Ft(t)} \text{prob. de aceptar } t \text{ a partir de la acción } a$$

Sin embargo, la probabilidad de aceptar el resto del test t, una vez ejecutada la acción a podrá depender de la forma en que dicha acción haya sido ejecutada, o más exactamente de los *estados* en los que se encontraran los procesos P y Q.

En suma, para definir la semántica del operador de elección externa, debemos ser capaces de hablar sobre los diferentes estados en los que puede quedar cada proceso después de resolver las elecciones internas que tenga en cabeza. Por *estado* entenderemos en este contexto el conjunto de acciones visibles que están en disposición de ser aceptadas si le son ofrecidas por el medio externo.

Pero previamente deberemos definir la probabilidad con la que cada estado posible de un proceso es alcanzado. Dichas probabilidades pueden calcularse a partir de las probabilidades con las que son aceptados una serie de tests simples de profundidad uno. Con cada uno de ellos calcularemos la suma de las probabilidades de estar en todos los estados que contienen alguna de las acciones ofrecidas en dichos tests. Una vez conocidas dichas sumas, basta combinarlas adecuadamente, en base a las reglas básicas de la probabilidad, para obtener la probabilidad de alcanzar cada uno de los estados posibles.

Como quiera que dichos tests tienen profundidad uno, ya conocemos la probabilidad con la que los pasan cada proceso. A partir de ellas se calcula la probabilidad de encontrarnos en cada estado, como sigue:

Definición 4.3 (Probabilidad de alcanzar un estado)
Sea P un proceso y A un estado posible del mismo. Denotaremos por $p(P, A)$ la

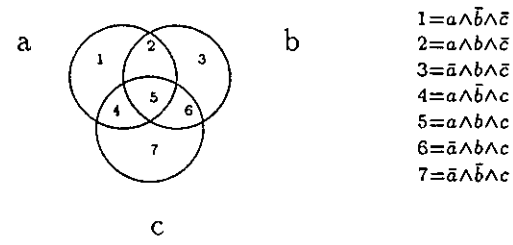


Figura 6: Estados posibles para 3 acciones

probabilidad de que el proceso P alcance el estado A, que definimos en la forma:

$$\begin{aligned} p(P,\emptyset) &= 1 - P \mid t_\Sigma \\ p(P,A) &= 1 - (P \mid t_{\Sigma-A}) - \sum_{B \subset A} p(P,B) \end{aligned}$$

donde para cada conjunto de acciones $A = \{a_1, \ldots, a_n\}$, el test t_A viene dado por:

$$t_A = \sum_{i=1}^{n} [\frac{1}{n}] a_i.\omega$$

□

Ejemplo 4.1 Consideremos el alfabeto $\Sigma = \{a, b, c\}$. En la figura 6 tenemos todos los estados posibles en los que puede estar un proceso arbitrario con dicho alfabeto. Podemos calcular la probabilidad de alcanzar cada uno de ellos por medio del siguiente sistema de ecuaciones:

$$\begin{aligned}
p(P,\emptyset) &= 1 - [P \mid ([\tfrac{1}{3}]a.\omega + [\tfrac{1}{3}]b.\omega + [\tfrac{1}{3}]c.\omega)] \\
p(P,\{a\}) &= [P \mid ([\tfrac{1}{3}]a.\omega + [\tfrac{1}{3}]b.\omega + [\tfrac{1}{3}]c.\omega)] - [P \mid ([\tfrac{1}{2}]b.\omega + [\tfrac{1}{2}]c.\omega)] \\
p(P,\{b\}) &= [P \mid ([\tfrac{1}{3}]a.\omega + [\tfrac{1}{3}]b.\omega + [\tfrac{1}{3}]c.\omega)] - [P \mid ([\tfrac{1}{2}]a.\omega + [\tfrac{1}{2}]c.\omega)] \\
p(P,\{c\}) &= [P \mid ([\tfrac{1}{3}]a.\omega + [\tfrac{1}{3}]b.\omega + [\tfrac{1}{3}]c.\omega)] - [P \mid ([\tfrac{1}{2}]a.\omega + [\tfrac{1}{2}]b.\omega)] \\
p(P,\{a,b\}) &= [P \mid ([\tfrac{1}{3}]a.\omega + [\tfrac{1}{3}]b.\omega + [\tfrac{1}{3}]c.\omega)] - [P \mid c.\omega] - p(P,\{a\}) - p(P,\{b\}) \\
p(P,\{a,c\}) &= [P \mid ([\tfrac{1}{3}]a.\omega + [\tfrac{1}{3}]b.\omega + [\tfrac{1}{3}]c.\omega)] - [P \mid b.\omega] - p(P,\{a\}) - p(P,\{c\}) \\
p(P,\{b,c\}) &= [P \mid ([\tfrac{1}{3}]a.\omega + [\tfrac{1}{3}]b.\omega + [\tfrac{1}{3}]c.\omega)] - [P \mid a.\omega] - p(P,\{b\}) - p(P,\{c\}) \\
p(P,\Sigma) &= [P \mid ([\tfrac{1}{3}]a.\omega + [\tfrac{1}{3}]b.\omega + [\tfrac{1}{3}]c.\omega)] - p(P,\{a\}) - p(P,\{b\}) - \\
&\quad p(P,\{c\}) - p(P,\{a,b\}) - p(P,\{a,c\}) - p(P,\{b,c\})
\end{aligned}$$

□

Obsérvese que $P \mid ([1/3]a.\omega + [1/3]b.\omega + [1/3]c.\omega)$, que en este caso particular no es otra cosa que $P \mid t_\Sigma$, representa la probabilidad de estar en algún estado en el que es posible la ejecución de alguna de las acciones del alfabeto.

Una vez que el sistema alcance un determinado estado A, será el medio exterior el que decida la acción que se va a ejecutar, eligiendo para ello entre las acciones del estado. Puesto que este medio exterior está representado en esta ocasión por el test t que se ofrece al proceso para su aceptación, la selección se hará atendiendo al valor de las probabilidades indicadas en el mismo.

La acción que será ejecutada debe estar forzosamente en el conjunto $A \cap Ft(t)$, por lo que las acciones de $Ft(t) - A$ no serán tomadas en consideración. La probabilidad asociada en el test a estas acciones, que no pueden ser ejecutadas, la repartiremos proporcionalmente entre el resto de las acciones de $Ft(t)$. La definición del correspondiente concepto de *probabilidad normalizada,* es la siguiente:

Definición 4.4 (Probabilidades Normalizadas)
Sea $t = \sum_{a \in Ft(t)} [p_a]a.t_a$ un test, $A \subsetneq \Sigma$ y $a \in Ft(t) \cap A$. La probabilidad $s(a, A, t)$ de seleccionar la acción a como la acción a ser ejecutada por parte de un proceso arbitrario P que está en el estado A, al que le es ofrecido el test t, viene dada por:

$$s(a,A,t) = \begin{cases} \dfrac{p_a}{1 - \sum_{b \in Ft(t)-A} p_b} & \text{Si } p_a \neq 0 \\ \dfrac{1}{|Ft(t) \cap A|} & \text{Si } \forall b \in Ft(t) \cap A,\ p_b = 0 \\ 0 & \text{Si } p_a = 0 \wedge \exists b \in Ft(t) \cap A,\ p_b \neq 0 \end{cases}$$

□

Sólo nos queda, para completar la definición de probabilidad de aceptar un test por parte de una elección externa, definir la probabilidad de aceptar el resto de un test, una vez que la acción a ha sido seleccionada como la primera acción visible a ejecutar, encontrándose cada uno de los argumentos de la elección en un determinado estado.

Definición 4.5 (Continuación de una elección externa)
Sean P, Q dos procesos, $A, B \subsetneq \Sigma$, $a \in A \cup B$, t un test y $p \in [0,1]$. La probabilidad de aceptar el resto del test t por parte del proceso $[p]P \,\square\, [1-p]Q$, una vez que la acción a ha sido ejecutada, encontrándose en el momento de dicha ejecución el proceso P en el estado A, y el proceso Q en el estado B, viene dada por

$$Cont([p]P\Box[1-p]Q, t, A, B, a) =$$

$$\begin{cases} P/(a,A) \mid t/a & \text{Si } a \in A,\ a \notin B \\ Q/(a,B) \mid t/a & \text{Si } a \in B,\ a \notin A \\ p \cdot (P/(a,A) \mid t/a) + (1-p) \cdot (Q/(a,B) \mid t/a) & \text{Si } a \in A \cap B \end{cases}$$

donde $P/(a,A)$ denota el resto del proceso P tras la ejecucion de la acción a, cuando dicho proceso se encuentra en el estado A. La definición formal de este nuevo operador auxiliar será introducida en la siguiente subsección.

Cuando los argumentos de la elección puedan ser sobreentendidos por el contexto, denotaremos abreviadamente mediante $Cont(\Box, t, A, B, a)$ a la probabilidad $Cont([p]P\Box[1-p]Q, t, A, B, a)$. □

Definición 4.6 (Probabilidad de aceptar un test)
Sean P, Q dos procesos, t un test y $p \in [0,1]$. La probabilidad de aceptar el test t por parte del proceso $[p]P\Box[1-p]Q$ queda definida por

$$[p]P\Box[1-p]Q \mid t =$$

$$\begin{cases} 1 & \text{Si } t = \omega \\ 0 & \text{Si } t = 0 \\ \sum\limits_{a \in Fi(t)} \sum\limits_{\substack{A,B \subseteq \Sigma \\ a \in A \cup B}} s(a, A \cup B, t) \cdot p(P,A) \cdot p(Q,B) \cdot Cont(\Box, t, A, B, a) & \text{otro caso} \end{cases}$$

□

Ejemplo 4.2 Consideremos el siguiente proceso

$$R = [\tfrac{1}{2}](a \to c \to STOP)\Box[\tfrac{1}{2}]([\tfrac{1}{2}]a \to STOP \sqcap [\tfrac{1}{2}]b \to STOP)$$

Çalculemos la probabilidad de aceptar el test $t = \frac{1}{2}a.c.\omega + \frac{1}{2}b.\omega$ por parte del mismo. Para ello procederemos a calcular los estados posibles de los argumentos de la elección externa.

Dichos argumentos, P y Q son

$$P = a \to c \to STOP$$
$$Q = [\tfrac{1}{2}]a \to STOP \sqcap [\tfrac{1}{2}]b \to STOP$$

Tenemos pues un alfabeto de dos acciones, a y b. El proceso P tiene un único estado, pues como primera acción sólo puede ejecutar la acción a, mientras que Q tiene dos estados, uno donde sólo puede ejecutar a y otro donde sólo puede ejecutar b. Más exactamente, tenemos

$$\begin{aligned} p(P,\{a\}) &= 1 \\ p(Q,\{a\}) &= 1/2 \\ p(Q,\{b\}) &= 1/2 \end{aligned}$$

Los factores de normalización para la acción a son

$$\begin{aligned} s(a,\{a\},t) &= 1 \\ s(a,\{a,b\},t) &= 1/2 \end{aligned}$$

y las continuaciones para la misma acción

$$\begin{aligned} Cont(\Box,t,\{a\},\{a\},a) &= 1/2 \\ Cont(\Box,t,\{a\},\{b\},a) &= 1 \end{aligned}$$

Respecto a la acción b, su ejecución por parte del proceso Q sólo es posible en un estado, y por parte de P en ninguno, lo que nos da en total un único estado de R en el cual puede ejecutarse dicha acción.

El factor de normalización para la acción b, en dicho estado es

$$s(b,\{a,b\},t) = 1/2$$

y la única continuación para la misma acción

$$Cont(\Box,t,\{a\},\{b\},a) = 1$$

Finalmente tenemos

$$\begin{aligned} [1/2]P\Box[1/2]Q \mid t = {} & s(a,\{a\},t)p(P,\{a\})p(Q,\{a\})Cont(\Box,t,\{a\},\{a\},a) \\ & + s(a,\{a\},t)p(P,\{a\})p(Q,\{b\})Cont(\Box,t,\{a\},\{b\},a) \\ & + s(b,\{a\},t)p(P,\{a\})p(Q,\{b\})Cont(\Box,t,\{a\},\{b\},b) \\ = {} & 1\cdot 1\cdot \tfrac{1}{2}\cdot\tfrac{1}{2} + \tfrac{1}{2}\cdot 1\cdot\tfrac{1}{2}\cdot 1 + \tfrac{1}{2}\cdot 1\cdot\tfrac{1}{2}\cdot 1 \\ = {} & \tfrac{3}{4} \end{aligned}$$

□

4.2.5 Resto de un proceso tras una acción en un estado

En esta subsección definiremos la semántica del proceso denotado con la sintaxis $P/(a, A)$, que anteriormenete introdujimos. El hecho de que hablemos de $P/(a, A)$ y no simplemente de P/a como se hace en CSP ordinario, se debe a que en la definición 4.4 hemos distinguido el estado en que se encuentra el proceso en cuestión cuando ejecuta la acción a seleccionada. Pero es que además, en el caso probabilístico la semántica de pruebas nos va a permitir aislar las continuaciones correspondientes a cada estado, lo que curiosamente no era posible en el caso no-determinista. De cara a justificar dicha afirmación, consideremos el siguiente proceso

$$Q_1 = ([\frac{1}{2}]([\frac{1}{2}]a \rightarrow b \rightarrow STOP \Box [\frac{1}{2}]b \rightarrow STOP) \sqcap [\frac{1}{2}]a \rightarrow STOP)$$

Este proceso puede encontrarse, antes de ejecutar acción alguna, en dos estados distintos: uno en el que pueden ser ejecutadas las acciones del conjunto $\{a, b\}$, y un segundo en el que sólo puede ser ejecutada la acción a. Es claro que el comportamiento de P tras la ejecución de la acción a sería distinto en función de que la acción ejecutada corresponda a uno u otro estado. En concreto, dichos comportamientos son los siguientes:

$$\begin{aligned} Q_1/(a, \{a, b\}) &= b \rightarrow STOP \\ Q_1/(a, \{a\}) &= STOP \end{aligned}$$

lo que podremos comprobar una vez hayamos introducido las correspondientes definiciones formales.

En general, para describir el comportamiento de $P/(a, A)$ en presencia de un test arbitrario $t \in PTEST$, nos valdremos de una serie de tests derivados de t a cada uno de los cuales, que vendrá asociado a la acción a y a un estado B, con $B \subseteq A - \{a\}$, lo denotaremos por $t^{(a,B)}$,.

En concreto, para definir $P/(a, A) \mid t$, procederemos por recurrencia, comenzando por el estado $A = \{a\}$, y obteniendo sucesivamente los valores $P/(a, A) \mid t$ correspondientes a estados cada vez mayores, a partir de los valores de estas probabilidades para todos los estados $B \subset A$.

Observamos en primer lugar que para todo conjunto de acciones A, si el mismo no es un estado alcanzable por el proceso P, o lo que es lo mismo, si $p(P, A) = 0$, tendremos $P/(a, A) \mid t = 0$.

Volvamos de nuevo al estado $A = \{a\}$ y supongamos que $p(P, \{a\}) > 0$. En tal caso, podemos calcular la probabilidad con la que P alcanza el estado $\{a\}$ y acepta el test $a.t$ en dicho estado. Ello se logra aplicando al proceso P el test

$$t^{(a,\emptyset)} = [0]a.t + \sum_{b \in \Sigma - \{a\}} \left\lceil \frac{1}{|\Sigma| - 1} \right\rceil b.0$$

Utilizando este valor, definimos la probabilidad $P/(a, \{a\}) \mid t$ como sigue:

$$P/(a, \{a\}) \mid t = \frac{P \mid t^{(a,\emptyset)}}{p(P, \{a\})}$$

Consideremos ahora los estados con dos acciones $A = \{a, b\}$. Introducimos ahora el test

$$t^{(a,\{b\})} = [0]a.t + [0]b.0 + \sum_{c \in \Sigma - \{a,b\}} \left\lceil \frac{1}{|\Sigma| - 2} \right\rceil c.0$$

Entonces $P \mid t^{(a,\{b\})}$ nos ofrece la suma de las probabilidades con las que P alcanza el estado $\{a\}$, y acepta el test $a.t$ desde él, más la mitad de la probabilidad de alcanzar el estado $\{a, b\}$, y tras ejecutar la acción a, aceptar el test t por parte del proceso continuación al que así se accede. En consecuencia, $P/(a, \{a, b\}) \mid t$ puede ser definido en la forma:

$$P/(a, \{a, b\}) \mid t = \frac{2 \cdot (P \mid t^{(a,\{b\})} - P \mid t^{(a,\emptyset)})}{p(P, \{a, b\})}$$

Finalmente, para un estado general A, el test a considerar depende de si este conjunto es el alfabeto completo Σ, o no lo es. Si $A \neq \Sigma$, el test es el siguiente:

$$t^{(a,A-\{a\})} = [0]a.t + \sum_{b \in A - \{a\}} [0]b.0 + \sum_{c \in \Sigma - A} \left\lceil \frac{1}{|\Sigma - A|} \right\rceil c.0$$

mientras que para $A = \Sigma$, el test está definido por

$$t^{(a,\Sigma-\{a\})} = \left\lceil \frac{1}{|\Sigma|} \right\rceil a.t + \sum_{c \in \Sigma - \{a\}} \left\lceil \frac{1}{|\Sigma|} \right\rceil c.0$$

En cualquiera de ambos casos tenemos

$$P \mid t^{(a,A-\{a\})} = \sum_{B \subseteq A - \{a\}} \frac{p(P, B \cup \{a\}) \cdot [P/(a, B \cup \{a\}) \mid t]}{|B \cup \{a\}|}$$

De aquí, cuando $p(P,A) \neq 0$, obtenemos la definición

$$P/(a,A) \mid t = \frac{|A| \cdot (P \mid t^{(a,A-\{a\})} - \sum_{C \subset A-\{a\}} \frac{(P/(a,C \cup \{a\}) \mid t) \cdot p(P, C \cup \{a\})}{|C|+1})}{p(P,A)}$$

Desarrollando dicha definición recursiva obtenemos la siguiente definición explícita equivalente:

$$P/(a,A) \mid t = \frac{|A| \cdot (P \mid t^{(a,A-\{a\})} - \sum_{C \subset A-\{a\}} (-1)^{|A-C|} \cdot P \mid t^{(a,C)})}{p(P,A)}$$

Ejemplo 4.3 Sea Q_1 el proceso introducido al comienzo de la subsección. Aplicándole al mismo el test $t^{(a,\emptyset)} = [0]a.b.\omega + [1]b.0$, obtenemos

$$Q_1 \mid t^{(a,\emptyset)} = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 0 = \frac{1}{2}$$

puesto que si Q_1 está en el estado $\{a,b\}$, la acción b será seleccionada y el test fallará; mientras que si el proceso está en el estado $\{a\}$, la acción a será la ejecutada y el test será aceptado. Entonces tenemos

$$Q_1/(a,\{a\}) \mid t = \frac{Q_1 \mid t^{(a,\emptyset)}}{p(Q_1,\{a\})}$$

De cara a ilustrar la aplicación de la definición en el caso de que el estado a estudiar no sea el total, consideremos un segundo proceso, Q_2, en este caso sobre el alfabeto de tres acciones $\Sigma = \{a,b,c\}$, definido por medio de

$$Q_2 = ([\frac{1}{2}]([\frac{1}{2}]a \to b \to STOP \Box [\frac{1}{2}]b \to STOP) \sqcap [\frac{1}{2}]a \to b \to STOP)$$

Aquí, si pretendemos calcular $Q_2/(a,\{a,b\}) \mid b.\omega$, podemos razonar de la misma manera, considerando los tests

$$\begin{aligned} b.\omega^{(a,\emptyset)} &= [0]a.b.\omega + [\tfrac{1}{2}]b.0 + [\tfrac{1}{2}]c.0 \\ b.\omega^{(a,\{b\})} &= [0]a.b.\omega + [0]b.0 + [1]c.0 \end{aligned}$$

Obtenemos

$$Q_2/(a,\{a,b\}) \mid b.\omega = \frac{2 \cdot (Q_2 \mid b.\omega^{(a,\{b\})} - Q_2 \mid b.\omega^{(a,\emptyset)})}{p(Q_2,\{a,b\})} = \frac{2 \cdot (\frac{3}{4} - \frac{1}{2})}{\frac{1}{2}} = 1$$

□

4.2.6 Composición paralela

El mecanismo para definir la semántica de la composición paralela es muy similar al aplicado en la subsección anterior, en la que estudiamos la semántica de la elección externa. Esto es debido a que la composición paralela puede ser descrita como una elección externa entre las primeras acciones ejecutables por cualquiera de sus componentes, seguida de la composición paralela de la continuación de aquel proceso de donde la acción ejecutada ha sido tomada, y el otro argumento. Definiremos en consecuencia la función auxiliar $Cont([p]P\|_A[1-p]Q, t, B, C, a)$, que abreviaremos $Cont(\|_A, t, B, C, a)$ cuando el proceso en cuestión se pueda sobreentender por el contexto, que indicará la probabilidad de aceptar el resto del test t por parte del resto del proceso $[p]P\|_A[1-p]Q$, una vez que la acción a ha sido ejecutada, suponiendo que antes de la ejecución de a, los procesos P y Q estaban en los estados A y B, respectivamente.

Definición 4.7 Sean P, Q dos procesos, t un test, $p \in [0,1]$, $A, B, C \subseteq \Sigma$, y $a \in \Sigma$. Definimos

$$Cont(\|_A, t, B, C, a) =$$

$$\begin{cases} (P/(a,B)\|_A Q) \mid t/a & \text{Si } a \notin A,\ a \in B,\ a \notin C \\ (P\|_A Q/(a,B)) \mid t/a & \text{Si } a \notin A,\ a \notin B,\ a \in C \\ p \cdot [(P/(a,B)\|_A Q) \mid t/a] + & \\ (1-p) \cdot [(P\|_A Q/(a,B)) \mid t/a] & \text{Si } a \notin A, a \in B \cap C \\ (P/(a,B)\|_A Q/(a,C)) \mid t/a & \text{Si } a \in A \cap B \cap C \\ 0 & \text{Si } a \in A, a \notin B \cap C \end{cases}$$

□

Entonces, la semántica del operador paralelo será la siguiente:

Definición 4.8 (Probabilidad de aceptar un test por una composición paralela) Sean P, Q dos procesos, t un test, $p \in [0,1]$ y $A \subseteq \Sigma$. La probabilidad de aceptación del test t por parte del proceso $[p]P\|_A[1-p]Q$ viene dada por

$([p]P\|_A[1-p]Q) \mid t =$

$$\begin{cases} 1 & \text{Si } t = \omega \\ 0 & \text{Si } t = 0 \\ \sum\limits_{a \in Fi(t) \cap A} \sum\limits_{\substack{B,C \subseteq \Sigma \\ a \in B \cap C}} s(a,D,t) \cdot p(P,B) \cdot p(Q,C) \cdot Cont(\|_A, t, B, C, a) + & \\ \sum\limits_{a \in Fi(t) - A} \sum\limits_{\substack{B,C \subseteq \Sigma \\ a \in B \cup C}} s(a,D,t) \cdot p(P,B) \cdot p(Q,C) \cdot Cont(\|_A, t, B, C, a) & \text{otro caso} \end{cases}$$

siendo $D = (B \cup C - A) \cup (A \cap B \cap C)$

□

4.2.7 Propiedades de la semántica de pruebas

Nos queda únicamente el operador de ocultación para concluir la definición de la semántica de pruebas del lenguaje PCSP. Sin embargo en este caso la idea intuitiva en que se basa la misma es más compleja, lo que a la postre da lugar a una definición formal notablemente complicada. Por ello nos ha parecido oportuno estudiar previamente una serie de propiedades de la parte de la semántica ya definida, los cuales entendemos que contribuirán a clarificar la definición de la semántica de la operación pendiente, amén de resultar de hecho interesantes de por sí.

En consecuencia, y en tanto no pasemos a definir la semántica del operador de ocultación, consideraremos en lo sucesivo que en los procesos PCSP que aparezcan, no interviene dicho operador.

Proposición 4.1 Para todo test $t \in PTEST$, y todo estado $A \subseteq \Sigma$ se satisfacen las siguientes propiedades de la operación $/(a, A)$.

1. $STOP/(a,A) \mid t = STOP \mid t$
2. $(a \rightarrow P)/(a,\{a\}) \mid t = P \mid t$
3. $p([p]P \sqcap [1-p]Q, A) \cdot ([p]P \sqcap [1-p]Q/(a,A) \mid t) =$
   $p \cdot p(P,A) \cdot (P/(a,A) \mid t) + (1-p) \cdot p(Q,A) \cdot (Q/(a,A) \mid t)$

4. $p([p]P \Box [1-p]Q, A) \cdot [p]P \Box [1-p]Q/(a,A) \mid t =$

$$\sum_{A=B \cup C} p(P,B) \cdot p(Q,C) \cdot Cont([p]P \Box [1-p]Q, t, B, C, a)$$

5. $p([p]P \|_D [1-p]Q, A) \cdot [p]P \|_D [1-p]Q/(a,A) \mid t =$

$$\sum_{A=B \cup C} p(P,B) \cdot p(Q,C) \cdot Cont([p]P \|_D [1-p]Q, t, B, C, a) -$$
$$\sum_{\substack{B \cap C = \emptyset \\ B,C \subseteq D}} p(P,B) \cdot p(Q,C)) \cdot Cont([p]P \|_D [1-p]Q, t, B, C, a)$$

Demostración: Probaremos cada una de las propiedades anteriores

1. $$\begin{aligned} STOP/(a,A) \mid t &= |A| \cdot \frac{STOP \mid t^{(a,A-\{a\})} - \sum_{C \subset A-\{a\}} STOP \mid t^{(a,C)}}{p(STOP, A)} \\ &= 0 \\ &= STOP \mid t \end{aligned}$$

2. $a \to P/(a,\{a\}) \mid t = |\{a\}| \cdot \frac{a \to P \mid t^{a,\emptyset}}{p(a \to P, \{a\})} = a \to P \mid a.t = P \mid t$

3. En primer lugar, de la definición de $/(a,A)$ se sigue

$$([p]P \sqcap [1-p]Q)/(a,A) \mid t =$$

$$\frac{|A|([p]P \sqcap [1-p]Q \mid t^{(a,A-\{a\})} - \sum_{C \subset A-\{a\}} (-1)^{|A-C|}[p]P \sqcap [1-p]Q \mid t^{(a,C)})}{p([p]P \sqcap [1-p]Q, A)}$$

Aplicando la definición de elección interna y separando los numeradores obtenemos

$$([p]P \sqcap [1-p]Q)/(a,A) \mid t =$$

$$p \cdot \frac{|A|(P \mid t^{(a,A-\{a\})} - \sum_{C \subset A-\{a\}} (-1)^{|A-C|} P \mid t^{(a,C)})}{p([p]P \sqcap [1-p]Q, A)} +$$

$$(1-p) \cdot \frac{|A|(Q \mid t^{(a,A-\{a\})} - \sum_{C \subset A-\{a\}} (-1)^{|A-C|} Q \mid t^{(a,C)})}{p([p]P \sqcap [1-p]Q, A)}$$

De donde, pasando el denominador al otro lado de la igualdad, y multiplicando y dividiendo los términos del lado derecho respectivamente por $p(P, A)$ y $p(Q, A)$, obtenemos la propiedad deseada

$$p([p]P \sqcap [1-p]Q, A) \cdot ([p]P \sqcap [1-p]Q/(a, A) \mid t) =$$

$$p \cdot p(P, A) \cdot (P/(a, A) \mid t) + (1-p) \cdot p(Q, A) \cdot (Q/(a, A) \mid t)$$

4. Al igual que en el caso anterior, tenemos en primer lugar, a partir de la definición de $/(a, A)$, que

$$([p]P \Box [1-p]Q)/(a, A) \mid t =$$

$$\frac{|A|([p]P \Box [1-p]Q \mid t^{(a,A-\{a\})} - \sum_{C \subset A-\{a\}} (-1)^{|A-C|}[p]P \Box [1-p]Q \mid t^{(a,C)})}{p([p]P \Box [1-p]Q, A)}$$

$P \mid t^{(a,B)}$ nos da la suma de las probabilidades de aceptar el test t supuesto que el proceso ha alcanzado algún estado correspondiente a un subconjunto cualquiera de B junto con la acción a (que es la definitivamente seleccionada), dividido en cada caso por el número de acciones del estado, que será la probabilidad (uniforme) de seleccionar la acción a en ese estado.

Entonces,

$$|A|([p]P \Box [1-p]Q \mid t^{(a,A-\{a\})} - \sum_{C \subset A-\{a\}} (-1)^{|A-C|}[p]P \Box [1-p]Q \mid t^{(a,C)})$$

nos ofrece la probabilidad de aceptar el test t, después de ser ejecutada la acción a tras haber alcanzado el proceso $[p]P \Box [1-p]Q$ el estado A.

Ahora bien, si el proceso $[p]P \Box [1-p]Q$ alcanza el estado A, es porque el proceso P habrá alcanzado un estado B, y Q estado C, de forma que $A = B \cup C$. Por lo que tenemos, para todos los pares de estados (B, C) en esas condiciones

$$([p]P \Box [1-p]Q \mid t^{(a,A-\{a\})} - \sum_{C \subset A-\{a\}} (-1)^{|A-C|}[p]P \Box [1-p]Q \mid t^{(a,C)}) =$$

$$\sum_{a \in B-C} p(P, B) \cdot p(Q, C) \cdot P/(a, B) \mid t +$$
$$\sum_{a \in C-B} p(P, B) \cdot p(Q, C) \cdot Q/(a, C) \mid t +$$
$$\sum_{a \in B \cap C} p(P, B) \cdot p(Q, C) \cdot p \cdot (P/(a, B) \mid t) + (1-p) \cdot (Q/(a, C) \mid t)$$

Por otra parte, aplicando la definición de $Cont$ se tiene

$$\sum_{A=B\cup C} p(P,B)\cdot p(Q,C)\cdot Cont([p]P\Box[1-p]Q,t,B,C,a) \;=$$
$$\sum_{a\in B-C} p(P,B)\cdot p(Q,C)\cdot P/(a,B) \mid t +$$
$$\sum_{a\in C-B} p(P,B)\cdot p(Q,C)\cdot Q/(a,C) \mid t +$$
$$\sum_{a\in B\cap C} p(P,B)\cdot p(Q,C)\cdot p\cdot (P/(a,B) \mid t) + (1-p)\cdot (Q/(a,C) \mid t)$$

por lo que concluimos la propiedad deseada

$$\sum_{A=B\cup C} p(P,B)\cdot p(Q,C)\cdot Cont([p]P\Box[1-p]Q,t,B,C,a) \;=$$
$$[p]P\Box[1-p]Q/(a,A) \mid t\cdot p([p]P\Box[1-p]Q,A)$$

5. El razonamiento es similar al empleado en el caso anterior

□

Proposición 4.2 Sea $P \in$ PCSP y $t \in PTEST$ un test que le es ofrecido para su aceptación. Tenemos entonces que

$$P \mid t = \sum_{a\in Ft(t)} \sum_{a\in A\subseteq\Sigma} s(a,A,t)\cdot p(P,A)\cdot P/(a,A) \mid t/a$$

Demostración: Consideraremos cada uno de los operadores cuya semántica hemos estudiado hasta el momento.

STOP: La propiedad es trivial.

PREFIJO: Por definición del operador prefijo, tenemos que $s(a,\{a\},t) = 1$ si $a \in Ft(t)$, y puesto que $p(a \to P,\{a\}) = 1$, se cumple la propiedad.

ELECCIÓN INTERNA: Partimos de la igualdad 3 demostrada en la prop. 4.1

$$p([p]P \sqcap [1-p]Q,A)\cdot ([p]P \sqcap [1-p]Q/(a,A) \mid t/a) \;=$$
$$p\cdot p(P,A)\cdot (P/(a,A) \mid t/a) + (1-p)\cdot p(Q,A)\cdot (Q/(a,A) \mid t/a)$$

Aplicando la misma se tiene

$$
\begin{aligned}
&[p]P \sqcap [1-p]Q \mid t \\
&\quad = p \cdot (P \mid t) + (1-p) \cdot (Q \mid t) \\
&\quad = p \sum_{a \in Ft(t)} \sum_{a \in A} s(a,A,t) \cdot p(P,A) \cdot (P/(a,A) \mid t/a) + \\
&\qquad (1-p) \sum_{a \in Ft(t)} \sum_{a \in A} s(a,A,t) \cdot p(Q,A) \cdot (Q/(a,A) \mid t/a) \\
&\quad = \sum_{a \in Ft(t)} \sum_{a \in A} s(a,A,t) \cdot \big[p \cdot p(P,A) \cdot (P/(a,A) \mid t/a) + \\
&\qquad (1-p) \cdot p(Q,A) \cdot (Q/(a,A) \mid t/a)\big] \\
&\quad = \sum_{a \in Ft(t)} \sum_{a \in A} s(a,A,t) \cdot p([p]P \sqcap [1-p]Q, A) \cdot ([p]P \sqcap [1-p]Q/(a,A) \mid t/a)
\end{aligned}
$$

ELECCIÓN EXTERNA: De nuevo aplicamos la prop. 4.1; en este caso su apartado 4, que nos dice

$$\sum_{A=B \cup B} p(P,B) \cdot p(Q,C) \cdot Cont([p]P \Box [1-p]Q, t, B, C, a) =$$

$$[p]P \Box [1-p]Q/(a,A) \cdot p([p]P \Box [1-p]Q, A)$$

Utilizando el mismo se tiene

$$
\begin{aligned}
&[p]P \Box [1-p]Q \mid t \\
&\quad = \sum_{a \in Ft(t)} \sum_{a \in B \cup C} s(a, B \cup C, t) \cdot p(P,B) \cdot p(Q,C) \cdot Cont(\Box, t, B, C, a) \\
&\quad = \sum_{a \in Ft(t)} \sum_{a \in A} s(a,A,t) \cdot p([p]P \Box [1-p]Q, A) \cdot ([p]P \Box [1-p]Q/(a,A) \mid t/a)
\end{aligned}
$$

COMPOSICIÓN PARALELA: Podemos razonar de una manera análoga al caso anterior, lo que nos conduce al resultado

$$
\begin{aligned}
&[p]P \|_A [1-p]Q \mid t \\
&\quad = \sum_{a \in Ft(t)} \sum_{a \in B \cup C} s(a, B \cup C, t) \cdot p(P,B) \cdot p(Q,C) \cdot Cont(\|_A, t, A, B, a) \\
&\quad = \sum_{a \in Ft(t)} \sum_{a \in D} s(a,D,t) \cdot p([p]P \|_A [1-p]Q, D) \cdot ([p]P \|_A [1-p]Q/(a,D) \mid t/a)
\end{aligned}
$$

□

Proposición 4.3 Para todo $P \in \mathrm{PCSP}$ se tiene $\sum_{A \subseteq \Sigma} p(P,A) = 1$.

Demostración: Por inducción estructural. Para $STOP$ y el operador prefijo, el resultado es trivial.

ELECCION INTERNA:

$$\begin{aligned}\sum_{A\subseteq\Sigma} p([p]P \sqcap [1-p]Q, A) &= \sum_{A\subseteq\Sigma} p\cdot p(P,A) + (1-p)\cdot p(Q,A)\\ &= p\cdot\sum_{A\subseteq\Sigma} p(P,A) + (1-p)\cdot\sum_{A\subseteq\Sigma} p(Q,A)\\ &= p + 1 - p \;=\; 1\end{aligned}$$

ELECCION EXTERNA:

$$\begin{aligned}\sum_{A\subseteq\Sigma} p([p]P \Box [1-p]Q, A) &= \sum_{A\subseteq\Sigma}\sum_{A=B\cup C} p(P,B)\cdot p(Q,C)\\ &= \sum_{B\subseteq\Sigma} p(P,B)\cdot\sum_{C\subseteq\Sigma} p(Q,C)\\ &= \sum_{B\subseteq\Sigma} p(P,B)\cdot 1 \;=\; 1\end{aligned}$$

COMPOSICION PARALELA:

$$\begin{aligned}&\sum_{\emptyset\neq A\subseteq\Sigma} p([p]P\|_D[1-p]Q, A) \;=\\ &\sum_{A=B\cup C} p(P,B)\cdot p(Q,C) - \sum_{\substack{B\cap C=\emptyset\\ B,C\subseteq D}} p(P,B)\cdot p(Q,C)\end{aligned}$$

Por otra parte

$$p([p]P\|_D[1-p]Q,\emptyset) \;=\; \sum_{\substack{B\cap C=\emptyset\\ B,C\subseteq D}} p(P,B)\cdot p(Q,C)$$

entonces

$$\begin{aligned}\sum_{A\subseteq\Sigma} p([p]P\|_D[1-p]Q, A) &= \sum_{A\subseteq\Sigma}\sum_{A=B\cup C} p(P,B)\cdot p(Q,C)\\ &= \sum_{B\subseteq\Sigma} p(P,B)\cdot\sum_{C\subseteq\Sigma} p(Q,C)\\ &= \sum_{B\subseteq\Sigma} p(P,B)\cdot 1 \;=\; 1\end{aligned}$$

□

4.2.8 Ocultación

Para definir el comportamiento del operador de ocultación, nos basaremos en el resultado enunciado en la prop. 4.2, en el sentido de que diseñaremos la correspondiente definición, de manera que dicho resultado siga siendo válido para procesos que contengan el operador de ocultación. En consecuencia, nos apoyaremos en la descripción expandida del paso de tests por parte de los procesos, que dicho resultado nos suministra, describiéndose la semántica del proceso resultante de la ocultación en la misma manera.

Por tanto, comenzaremos calculando la probabilidad con la que el proceso resultado de la ocultación alcanza cada posible estado, para luego pasar al estudio de las correspondientes continuaciones.

Para ello veremos los procesos $P\backslash(a,q)$ como límite de una sucesión de aproximaciones, que notaremos $P\backslash_n(a,q)$, que nos ofrecen la información sobre el comportamiento del proceso resultado de la ocultación, que se puede obtener a partir del conocimiento del proceso dado sobre tests de profundidad menor o igual a n.

- Para la primera aproximación, obviamente $P\backslash_0 = STOP$, por lo que

$$p(P\backslash_0(a,q),A) = \begin{cases} 1 & \text{Si } A = \emptyset \\ 0 & \text{En otro caso} \end{cases}$$

- Para calcular la segunda aproximación, utilizaremos el comportamiento de P sobre los tests de un sólo nivel. Tenemos entonces que el proceso en el que se ha ocultado una acción alcanzará un estado (en el que obviamente no estará la acción oculta) si el mismo podía ser alcanzado antes de la ocultación, es decir

$$p(P\backslash_1(a,q),A) = p(P,A)$$

- Si avanzamos otro paso, tendremos ya tres formas diferentes de alcanzar un estado A por parte del proceso $P\backslash_2(a,q)$. En primer lugar, el estado puede alcanzarse directamente por P como en el caso anterior. Podemos también alcanzar en principio cualquier otro estado B que contenga la acción ocultada a, y tras la ejecución de la misma (lo que ocurrirá con probabilidad

q) alcanzar el estado A. Pero hay una tercera posibilidad cuya captación resulta más difícil a nivel intuitivo, que consiste en que se alcance por parte de P un cierto estado B, conteniendo la acción a, y tal que $B - \{a\} \subseteq A$, de forma que tras la ejecución de la acción a se pueda alcanzar un estado C tal que $B \cup C = A$. Tal caso refleja la situación en la que el sistema ha decidido en principio no ejecutar la acción oculta a (lo que ocurre con probabilidad $1 - q$) aunque queda en disposición de ejecutarla con seguridad en el caso de que la oferta del entorno contenga acciones de $C - B$, para eventualmente hacer posible su ejecución, pues desde luego ello no es posible en tanto el proceso no cambie de estado.

En definitiva obtenemos

$$\begin{aligned} & p(P\backslash_2(a,q),A) \;=\; \\ & \quad p(P,A) \;+\; q\cdot(\sum_{a\in B\subseteq\Sigma} p(P,B)\cdot p(P/(a,B),A))+ \\ & \quad (1-q)\cdot(\sum_{a\in B\subseteq A\cup\{a\}} p(P,B)\cdot \sum_{B\cup C=A} p(P/(a,B),C)\;) \end{aligned}$$

o lo que es lo mismo

$$\begin{aligned} & p(P\backslash_2(a,q),A) \;=\; \\ & \quad p(P,A) \;+\; q\cdot(\sum_{a\in B\subseteq\Sigma} p(P,B)\cdot p(P\backslash_1(a,q),A))+ \\ & \quad (1-q)\cdot(\sum_{a\in B\subseteq A\cup\{a\}} p(P,B)\cdot \sum_{B\cup C=A} p(P/(a,B)\backslash_1(a,q),C)\;) \end{aligned}$$

- En definitiva el caso general correspondiente a la aproximación enésima, inspirado por lo obtenido en los casos particulares anteriores, queda como sigue

$$\begin{aligned} & p(P\backslash_n(a,q),A) \;=\; \\ & \quad p(P,A) \;+\; q\cdot(\sum_{a\in B\subseteq\Sigma} p(P,B)\cdot p(P/(a,B)\backslash_{n-1}(a,q),A))+ \\ & \quad (1-q)\cdot(\sum_{a\in B\subseteq A\cup\{a\}} p(P,B)\cdot \sum_{B\cup C=A} p(P/(a,B)\backslash_{n-1}(a,q),C)\;) \end{aligned}$$

Definimos entonces

$$p(P\backslash(a,q),A) \;=\; \lim_n p(P\backslash_n(a,q),A)$$

lo que es correcto, ya que la sucesión en cuestión es no decreciente, puesto que los términos que aparecen en cada aproximación se mantienen en la siguiente, siendo

todos ellos no negativos, y por otro lado los valores $p(P\backslash_n(a,q),A)$ se pueden efectivamente entender como probabilidades al ser menores o iguales a 1, puesto que el valor de los mismos está acotado por 1. Ello es en efecto así, pues los términos sumados en la parte derecha de la igualdad corresponden a distintos estados que contienen la acción a, a los que se les agrega el término correspondioente al propio estado A, por lo que en definitiva obtenemos un valor menor o igual al de la probabilidad de alcanzar un estado cualquiera, que en virtud de la prop. 4.3 sabemos que es 1.

Finalmente, tomando límites en ambos lados de la igualdad anterior obtenemos la siguiente caracterización recurrente de los valores $p(P\backslash_n(a,q),A)$.

$$\begin{aligned} & p(P\backslash(a,q),A) \; = \\ & \quad p(P,A) \; + \; q \cdot (\sum_{a \in B \subseteq \Sigma} p(P,B) \cdot p(P/(a,B)\backslash(a,q),A)) + \\ & \quad (1-q) \cdot (\sum_{a \in B \subseteq A \cup \{a\}} p(P,B) \cdot \sum_{B \cup C = A} p(P/(a,B)\backslash(a,q),C) \;) \end{aligned}$$

Observación: En tanto y cuando nos limitemos a trabajar con procesos finitos, el sistema de ecuaciones así resultante podría considerarse como una definición equivalente de la probabilidad con la que el proceso $P\backslash(a,q)$ alcanza cada estado A. Sin embargo ello no seguirá siendo cierto en general, cuando pasemos a admitir los procesos infinitos, pues en tal caso puede suceder que dicho sistema no tenga solución única. Ello en concreto sucederá cuando tenemos una situación de divergencia, lo que en particular será tenido en cuenta al estudiar los procesos infinitos en la siguiente sección.

Si ahora recordamos el resultado de la prop. 4.2, en virtud del cual obtendríamos en este caso

$$P\backslash(a,q) \mid t = \sum_{a \notin A \subset \Sigma} \sum_{b \in A} s(b,A,t) \cdot p(P\backslash(a,q),A) \cdot P\backslash(a,q)/(b,A) \mid t/b$$

observamos que, puesto que tenemos ya calculados los valores $p(P\backslash(a,q),A)$, nos quedan por calcular las continuaciones $P\backslash(a,q)/(b,A) \mid t/b$, para tener completada la definición del proceso. Sin embargo no es posible calcular estos últimos valores al margen de aquellos, puesto que la probabilidad de continuar aceptando un test tras una acción en un estado, va indisolublemente asociada a la probabilidad de alcanzar dicho estado.

En concreto, si nos fijamos en cada estado A y cada acción del mismo b, y utilizamos el hecho de que la continuación tras alcanzar dicho estado, dependerá en cada caso de la forma concreta en que el mismo haya sido alcanzado, utilizando la definición implícita obtenida para la probabilidad de alcanzar el estado en cuestión, obtenemos

$$
\begin{aligned}
& p(P\backslash(a,q),A)\cdot P\backslash(a,q)/(b,A) \mid t = \\
& \quad p(P,A)\cdot P/(b,A)\backslash(a,q) \mid t + \\
& \quad q\cdot(\sum_{a\in B\subseteq\Sigma} p(P,B)\cdot p(P/(a,B)\backslash(a,q),A)\cdot P/(a,B)\backslash(a,q)/(b,A) \mid t) + \\
& \quad (1-q)\cdot(\sum_{a\in B\subseteq A\cup\{a\}} p(P,B)\cdot \sum_{B\cup C=A} p(P/(a,B)\backslash(a,q),C)\cdot \\
& \qquad \begin{cases} P/(b,B)\backslash(a,q) \mid t & \text{Si } b\in B \\ P/(a,B)\backslash(a,q)/(b,C) \mid t & \text{Si } b\notin B \end{cases})
\end{aligned}
$$

lo que sustituido en la expresión anterior nos conduce a la definición definitiva de la semántica del operador de ocultación.

$$
\begin{aligned}
& P\backslash(a,q) \mid t = \\
& \sum_{a\notin A\subset\Sigma}\sum_{b\in A} s(b,A,t)\cdot \\
& \qquad p(P,A)\cdot P/(b,A)\backslash(a,q) \mid t/b + \\
& \qquad q\cdot(\sum_{a\in B\subseteq\Sigma} p(P,B)\cdot p(P/(a,B)\backslash(a,q),A)\cdot P/(a,B)\backslash(a,q)/(b,A) \mid t/b) + \\
& \qquad (1-q)\cdot(\sum_{a\in B\subseteq A\cup\{a\}} p(P,B)\cdot \sum_{B\cup C=A} p(P/(a,B)\backslash(a,q),C)\cdot \\
& \qquad\quad \begin{cases} P/(b,B)\backslash(a,q) \mid t/b & \text{Si } b\in B \\ P/(a,B)\backslash(a,q)/(b,C) \mid t/b & \text{Si } b\notin B \end{cases})
\end{aligned}
$$

Observación: Al igual que ya indicamos en nuestra observación anterior, esta definición recursiva no sería válida en aquellos casos en los que su aplicación no terminase, por culpa de la existencia de una secuencia infinita de ocurrencias de la acción ocultada, es decir por culpa de una *divergencia* causada por dicha ocultación. Sin embargo, utilizando las mismas ideas podríamos obtener una definición correcta, incluso en el caso en el que apareciera una divergencia, sin más que repetir el proceso utilizando las aproximaciones ya utilizadas para formalizar con precisión la definición de las probabilidades con las que el proceso resultado de la ocultación alcanza cada dicho estado. Si no hemos hecho tal cosa explícitamente en esta ocasión, ha sido exclusivamente por no alargar en exceso la exposición, pues de

hecho la preceptiva formalización no introduciría ni nuevas ideas, ni tampoco adicionales dificultades técnicas, sobre la ya realizada.

Una presentación equivalente de la definición anterior, puede ser obtenida reordenando los sumatorios que en ella aparecen, en base a los estados alcanzables por el proceso P, en lugar de hacerlo en base a los alcanzables por $P\backslash(a,q)$.

$$
\begin{aligned}
&P\backslash(a,q) \mid t = \\
&\sum_{a\notin A\subset\Sigma}\sum_{b\in A} s(b,A,t)\cdot p(P,A)\cdot (P/(b,A))\backslash(a,q) \mid t/b + \\
&\sum_{a\in A\subset\Sigma} p(P,A)\cdot [q\cdot P/(a,A)\backslash(a,q)) \mid t) + \\
&\quad (1-q)\cdot \sum_{C\subseteq\Sigma} s(b,A\cup C,t)\cdot p(P/(a,A)\backslash(a,q),C)\cdot \\
&\quad (\sum_{b\in A} P/(b,A)\backslash(a,q) \mid t/b + \sum_{b\in C-A} P/(a,A)\backslash(a,q)/(b,C) \mid t/b\ )\ ]
\end{aligned}
$$

Los siguientes ejemplos muestran que la semántica de observaciones que hemos propuesto, formaliza adecuadamente el comportamiento (razonable) deseado del operador de ocultación.

Ejemplo 4.4 Sea el proceso $P = [\frac{1}{2}]a \rightarrow c \rightarrow STOP \Box [\frac{1}{2}]b \rightarrow STOP$. Tenemos entonces un único estado para él que es el $\{a,b\}$. Cualquier otro estado tiene probabilidad 0. Así pues tenemos que

$$P \mid t = p_a\cdot (P/(a,\{a,b\}) \mid t/a) + p_b\cdot (P/(b,\{a,b\}) \mid t/b)$$

Calculemos $P\backslash(a,1/3)$ para cada uno de los distintos tests de profundidad uno sobre el alfabeto $\{b,c\}$. Con vistas a no alargar interminablemente los cálculos precisoa, admitiremos, lo que de hecho a nivel intuitivo es obvio, que los procesos $b \rightarrow STOP\backslash(a,1/3)$ y $c \rightarrow STOP\backslash(a,1/3)$ son equivalentes respectivamente a los procesos $b \rightarrow STOP$ y $c \rightarrow STOP$. Tales equivalencias podrían comprobarse sin más que desarrollar los cálculos de las probabilidades con las que dichos procesos pasan un test cualquiera. Obtenemos entonces

$$
\begin{aligned}
&P\backslash(a,1/3) \mid c.\omega \\
&\quad = \frac{1}{3}P/(a,\{a,b\})\backslash(a,1/3) \mid c.\omega + \frac{2}{3}P/(a,\{a,b\})\backslash(a,1/3) \mid c.\omega
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{3}(c \rightarrow STOP \mid c.\omega) + \frac{2}{3}(c \rightarrow STOP \mid c.\omega) \\
&= \frac{1}{3} + \frac{2}{3} = 1
\end{aligned}
$$

$$
\begin{aligned}
&P\backslash(a,1/3) \mid b.\omega \\
&= \frac{1}{3}P/(a,\{a,b\})\backslash(a,1/3) \mid b.\omega + \frac{2}{3}\cdot(p(P/(a,\{a,b\}),\{b\})\cdot \\
&\quad P/(b,\{a,b,c\})\backslash(a,1/3) \mid t/b) \\
&= \frac{1}{3}(c \rightarrow STOP \mid b.\omega) + \frac{2}{3}\cdot 1\cdot(STOP \mid \omega) \\
&= \frac{1}{3}\cdot 0 + \frac{2}{3} = \frac{2}{3}
\end{aligned}
$$

$$
\begin{aligned}
P\backslash(a,1/3) \mid [\tfrac{1}{2}]b.\omega + [\tfrac{1}{2}]c.\omega = {} & \tfrac{1}{3}P/(a,\{a,b\})\backslash(a,1/3) \mid [\tfrac{1}{2}]b.\omega + [\tfrac{1}{2}]c.\omega + \\
& \tfrac{2}{3}\cdot(s(b,\{b\},t)\cdot P/(b,\{a,b\})\backslash(a,1/3) \mid t/b \\
= {} & \tfrac{1}{3}(c \rightarrow STOP \mid [\tfrac{1}{2}]b.\omega + [\tfrac{1}{2}]c.\omega) + \\
& \tfrac{2}{3}\cdot 1\cdot(STOP \mid \omega) \\
= {} & \tfrac{1}{3} + \tfrac{2}{3} = 1
\end{aligned}
$$

Finalmente generalizaremos los resultados anteriores, calculando el comportamiento de este proceso P sobre un test cualquiera t con el alfabeto $\Sigma = \{a,b,c\}$. En primer lugar, si $b \in Ft(t)$ tenemos

$$
\begin{aligned}
P\backslash(a,1/3) \mid t = {} & \tfrac{1}{3}P/(a,\{a,b\})\backslash(a,1/3) \mid t + \\
& \tfrac{2}{3}\cdot p(P/(a,\{a,b\}),\{b\})\cdot(s(b,\{b\},t)\cdot \\
& P/(b,\{a,b\})\backslash(a,1/3) \mid t/b \\
= {} & \tfrac{1}{3}(c \rightarrow STOP \mid t) + \tfrac{2}{3}\cdot(STOP \mid t/b)
\end{aligned}
$$

En segundo lugar, si $b \notin Ft(t)$

$$
\begin{aligned}
P\backslash(a,1/3) \mid t = {} & \tfrac{1}{3}P/(a,\{a,b\})\backslash(a,1/3) \mid t + \\
& \tfrac{2}{3}P/(a,\{a,b\})\backslash(a,1/3) \mid t \\
= {} & P/(a,\{a,b\})\backslash(a,1/3) \mid t \\
= {} & c \rightarrow STOP \mid t
\end{aligned}
$$

En definitiva obtenemos que el proceso en cuestión se comporta igual que el proceso

$$[\tfrac{1}{3}]c \rightarrow STOP \sqcap [\tfrac{2}{3}]([\tfrac{1}{2}]b \rightarrow STOP \Box [\tfrac{1}{2}]c \rightarrow STOP$$

□

Ejemplo 4.5 Consideremos ahora los procesos

$$\begin{aligned} P = &\ [\tfrac{1}{2}]([\tfrac{1}{2}]a \to STOP \Box [\tfrac{1}{2}]b \to STOP) \sqcap [\tfrac{1}{2}]STOP \\ Q = &\ [\tfrac{1}{2}]a \to STOP \sqcap [\tfrac{1}{2}]b \to STOP \end{aligned}$$

que en su momento no pudimos distinguir en el marco de nuestra primera semántica de trazas. Ahora tenemos que ambos tienen distintas observaciones, pues en concreto para el test $t = [\frac{1}{2}]a.\omega + [\frac{1}{2}]b.\omega$ se tiene

$$\begin{aligned} P \mid t &= \tfrac{1}{2} \\ Q \mid t &= 1 \end{aligned}$$

Pretendemos que para estos procesos se cumpla que $P\backslash(a, 1/2) \neq Q\backslash(a, 1/2)$. Y de hecho ello es efectivamente así, pues para el árbol $b.\omega$ tenemos:

$$\begin{aligned} P\backslash(a,1/2) \mid (b.\omega) = &\ s(b,\emptyset,b.\omega)\cdot p(P,\emptyset)\cdot(STOP \mid \omega) + \\ &\ p(P,\{a,b\})\cdot(1/2\cdot((STOP \mid b.\omega) + 1/2\cdot(STOP \mid \omega)) \\ = &\ 0 + 0 + \tfrac{1}{4} = \tfrac{1}{4} \end{aligned}$$

$$\begin{aligned} Q\backslash(a,1/2) \mid (b.\omega) = &\ s(b,\{b\},b.\omega)\cdot p(P,\{b\})\cdot(STOP \mid \omega) + \\ &\ p(P,\{a\})\cdot 1/2\cdot((STOP \mid b.\omega) \\ = &\ \tfrac{1}{2} + 0 = \tfrac{1}{2} \end{aligned}$$

Por otra parte, en consistencia con lo que se tiene en el modelo clásico, tenemos que

$$\begin{aligned} P\backslash(a,1/2) &= [\tfrac{1}{2}]([\tfrac{1}{2}]STOP \sqcap [\tfrac{1}{2}]b \to STOP) \sqcap [\tfrac{1}{2}]STOP \\ &= [\tfrac{3}{4}]([\tfrac{1}{2}]STOP \sqcap [\tfrac{1}{4}]b \to STOP) \\ Q\backslash(a,1/2) &= [\tfrac{1}{2}]STOP \sqcap [\tfrac{1}{2}]b \to STOP \end{aligned}$$

si bien los cálculos concretos que nos demostrarían dichas equivalencias resultarían tremendamente tediosos. No obstante, en el capítulo 6 introduciremos nuevas herramientas formales, con ayuda de las cuales la complejidad de dichas pruebas se reducirá notablemente. □

4.3 Recursión y divergencias

Finalmente nos queda definir la semántica de pruebas del operador de recursión, lo que nos conducirá a una definición completa de la semántica de nuestro lenguaje

PCSP.

Puesto que el proceso $\mu X.P(X)$ podemos entenderlo como la solución (distinguida) de la ecuación $X = P(X)$, entonces debe cumplirse

$$\mu X.P(X) \mid t = P(\mu X.P(X)) \mid t$$

De donde, si poseemos algún conocimiento del proceso P, podremos eventualmente utilizarlo para obtener nueva información sobre el mismo, siendo posible repetir el proceso, de una forma en principio ilimitada. Ello nos conduce a la secuencia de aproximaciones finitas del proceso, en el marco de la técnica de *mínimo punto fijo*. Dicha secuencia viene dada por:

$$\begin{aligned} P_0 &= STOP \\ P_1 &= P(STOP) \\ P_2 &= P(P_1) \\ \cdots &\quad \cdots \\ P_n &= P(P_{n-1}) \end{aligned}$$

Por medio de la misma llegamos a la siguiente primera definición de la semántica del operador recursivo:

$$\mu X.P(X) \mid t = \lim_n P_n \mid t$$

Lamentablemente la simple aplicación de la misma nos conduce a resultados indeseados, e incluso formalmente incorrectos en el marco en que de momento nos desenvolvemos, cuando aprece una ocultación en el marco de la recursión. Por ejemplo, dado el proceso $P = a \rightarrow P$, el mismo produce una ejecución infinita de acciones a, de donde tendremos problemas a la hora de definir el significado del proceso $P\backslash(a, q)$.

Para calcular la probabilidad de que este proceso alcance un estado A en el que no aparezca la acción a, aplicamos la definición correspondiente, asumiendo que, como podríamos de hecho probar, el proceso P tras ejecutar una serie de acciones a en el estado $\{a\}$, continúa comportandose como el propio proceso P. Entonces

tenemos

$$\begin{aligned} p(P\backslash_1(a,q),A) &= p(P,A) = 0 \\ p(P\backslash_2(a,q),A) &= p(P,A) + q\cdot(\sum_{a\in B} p(P,B)\cdot p(P/(a,B),A)) + \\ &\quad (1-q)\cdot(\sum_{a\in B} p(P,B)\cdot \sum_{C\cup(B-\{a\})=A} p(P/(a,B),C)) \end{aligned}$$

Como en este caso $a \notin A$, y $a \notin C$, tenemos que el valor de dicha probabilidad es 0. En general, partiendo de que $p(P\backslash_{n-1}(a,q),A) = 0$, obtenemos

$$\begin{aligned} p(P\backslash_n(a,q),A) &= p(P,A) + q\cdot(\sum_B p(P,B)\cdot p(P/(a,B)\backslash_{n-1}(a,q),A)) + \\ &\quad (1-q)(\sum_B p(P,B)\cdot \sum_C p(P/(a,B)\backslash_{n-1}(a,q),C)) = 0 \end{aligned}$$

y en definitiva $p(P\backslash(a,q),A) = 0$.

Este resultado no es en absoluto sorprendente en lo que se refiere a los estados no vacíos, puesto que es razonable que dicho proceso no pueda alcanzar ningún estado estable en el que pueda ejecutar acción visible alguna.

En consecuencia podríamos llegar a la conclusión de que el proceso está parado, siendo por tanto incapaz de ejecutar ninguna acción, en cuyo caso podríamos identificar al proceso $P\backslash(a,q)$ con el proceso $STOP$, que tiene dicho comportamiento.

Sin embargo, el resultado también es cierto para el estado $\emptyset$, que por tanto también tiene probabilidad 0 de ser alcanzado. Ello nos lleva a que el proceso que estamos considerando no puede ser equivalente a $STOP$, ya que el mismo tiene probabilidad nula de encontrarse en todos los estados, excepto para el estado vacío que alcanza con probabilidad 1. Por otra parte, para todos los procesos que hemos encontrado hasta el momento, la suma de las probabilidades de alcanzar los estados posibles era 1, lo que evidentemente ha dejado ahora de cumplirse.

Si queremos que dicho resultado siga siendo válido, debemos admitir la posibilidad de que un proceso se encuentre en un nuevo estado *indeterminado*, en el que no se puede aceptar ninguna acción, pero que sin embargo no es igual al estado vacío.

Un proceso en esas condiciones, lo consideraremos en una situación de *divergencia.*

En particular, denotaremos por DIV al proceso que sólo puede diverger. Se trata de un nuevo proceso sintáctico, que por tanto será necesario tratar adecuadamente. Como quedará formalizado a continuación, la semántica de dicho proceso nos conducirá al elemento mínimo del (nuevo) dominio semántico de procesos probabilísticos. Por ello utilizaremos al mismo como punto de partida a la hora de redefinir la semántica de los procesos recursivos.

Definición 4.9 Sea $\mu X.P(X)$ un proceso definido recursivamente, entonces el paso de un test $t \in PTEST$ por parte del mismo se define como el límite

$$\mu X.P(X) \mid t = \lim_n P^n(DIV) \mid t$$

□

Donde hemos utilizado la nueva cadena de aproximaciones al proceso, con base en DIV, en lugar de la antigua con base en $STOP$.

Tenemos entonces que

Proposición 4.4 La definición del paso de test para procesos recursivos es correcta.

Demostración: Debemos probar que el límite en cuestión existe. Ello es así, pues dada la sucesión de procesos

$$\{P_n(DIV)\}_{n\in\mathbb{N}}$$

tenemos en primer lugar, que para todo test t la sucesión numérica que forman las probabilidades de paso de los mismos es creciente. Aunque podríamos probar dicho resultado en este momento, nos permitiremos retrasar dicha prueba hasta el capítulo siguiente, en cuyo corolario 5.5 se retomará la cuestión, a fin de evitar el desarrollo en este momento de toda la artillería formal necesaria para obtener el resultado, pues consideramos a dicho capítulo el marco natural para desplegarla.

Admitamos entonces que dichas sucesiones son crecientes, de donde, puesto que se trata de valores de probabilidades, y por tanto acotados por 1, obtenemos que el límite de la sucesión existe, y además es también menor o igual que 1. Con lo cual, queda probada la corrección de la definición. □

Si nos centramos ahora en el problema de calcular la probabilidad de que un proceso diverja, tenemos que la misma se obtendría complementando la probabilidad con la que el proceso alcanza cualquiera de los antiguos estados *determinados*. La suma de todos ellos, con excepción del estado $\emptyset$, se obtiene mediante $P \mid t_\Sigma$, que nos indica la probabilidad con la que el proceso acepta ejecutar una acción cualquiera. Ahora bien, con los tests definidos hasta el momento no podemos discriminar entre la divergencia y el estado vacío, sino que únicamente podemos calcular la suma de ambos valores. Es preciso pues, introducir un nuevo test, que denotaremos por $\surd$, el cual reconecerá la parada de un proceso. Naturalmente, que hemos entonces de definir, por inducción estructural como de ordinario, la probabilidad con la que los distintos procesos pasan el nuevo test. Ello lo haremos a continuación, pudiendose concluir tras ello, como deseamos, que para todo proceso P se tiene

$$p(P, \emptyset) \;=\; P \mid \surd$$

Con este objetivo en mente, comenzamos definiendo

$$\begin{aligned} STOP \mid \surd \;&=\; 1 \\ DIV \mid \surd \;&=\; 0 \end{aligned}$$

En lo que se refiere al prefijo, tenemos que el proceso $a \rightarrow P$, claramente no está en un estado vacío, por lo que tomamos

$$a \rightarrow P \mid \surd \;=\; 0$$

La elección interna alcanzará un estado vacío si alguna de sus componentes lo alcanza, de modo que, ponderando con las respectivas probabilidades, obtenemos

$$[p]P \sqcap [1-p]Q \mid \surd = p \cdot (P \mid \surd) + (1-p) \cdot (Q \mid \surd)$$

Por el contrario, la elección externa sólo parará si ambos procesos están incapacitados para proseguir, luego

$$[p]P \,\Box\, [1-p]Q \mid \surd \;=\; P \mid \surd \cdot Q \mid \surd$$

En cuanto a la composición paralela, en lo que se refiere a las acciones de no sincronización, ocurre algo análogo a lo que sucede para la elección externa.

Sin embargo el proceso resultante también parará si ambas componentes alcanzan estados disjuntos que sean subconjuntos del conjunto de sincronización. Ambos casos quedan correctamente capturados, definiendo

$$[p]P \|_A [1-p]Q \mid \surd = \sum_{B,C \subset A, B \cap C = \emptyset} p(P,B) \cdot p(Q,C)$$

Finalmente, el proceso resultante de la ocultación de una acción, parará (alcanzará el estado vacío) de manera similar a como alcanza cualquier otro estado. Es decir, o bien el proceso previo a la ocultación para, o bien el proceso alcanza un estado B, y tras ejecutar el sistema la acción ocultada a (con probabilidad q) el proceso resultante para; o bien el proceso original alcanza el estado $\{a\}$, y para después de ejecutar la misma. Se tiene, en suma

$$\begin{aligned} P\backslash(a,q) \mid \surd = \; & P \mid \surd + \\ & q \cdot (\sum_B p(P,B) \cdot P/(a,B)\backslash(a,q) \mid \surd + \\ & (1-q) \cdot p(P,\{a\}) \cdot P/(a,\{a\})\backslash(a,q) \mid \surd \end{aligned}$$

en donde de nuevo debemos interpretar de una manera operacional, es decir como límite de las correspondientes aproximaciones, el uso recurrente del operador de ocultación, a fin de que dicha definición recurrente tenga sentido.

Naturalmente, en lo que al operador de recursión hace referencia, al mismo es aplicable la definición general que ya habíamos adelantado

$$\mu X.P(X) \mid \surd = \lim_n P_n(DIV) \mid \surd$$

Concluímos este capítulo con una breve referencia a algunas propiedades interesantes de la semántica de pruebas que acabamos de definir.

En primer lugar observamos que en este modelo se tienen (automáticamente) propiedades de equidad (*fairness*) en lo que se refiere a una sucesión infinita de elecciones internas, lo que no sucede en el caso del modelo CSP ordinario, ni en general en ninguno de los modelos habituales de descripción de procesos. Así, en nuestro modelo probabilístico se cumple

$$\mu X.([p]a \to P \sqcap [1-p]X) \equiv a \to P$$

lo que nos indica que si el medio en se encuentra un proceso está empeñado en ejecutar una a, y el proceso en cuestión está en disposición de aceptarlo en principio con probabilidad p (no nula), y en caso de no hacerlo queda en disposición de que el medio vuelva a intentarlo, permaneciendo constante la probabilidad de aceptación, podemos garantizar que en tiempo finito, la acción será ejecutada con seguridad.

Sin embargo, dicho resultado no se mantiene para el caso del operador de elección externa, para el que se tiene el mismo resultado que se tenía en el modelo sin probabilidades. Así:

$$\mu X.([p]a \rightarrow P \Box [1-p]X) \equiv DIV$$

Ello es (técnicamente) debido a que para calcular la probabilidad de que el proceso resultante de una elección externa alcance un estado, debemos comenzar por calcular la probabilidad con la que cada uno de sus argumentos lo alcanza, lo que en el caso recursivo que nos ocupa nos sume en un círculo vicioso, que nos conduce a que todo estado *determinado* se alcanza con probabilidad nula.

Más en general se tiene que el operador de elección externa sigue siendo estricto respecto de la divergencia, como sucedía en el modelo no probabilístico. Ello sin embargo no es así en lo que se refiere a la elección interna. Esto hace que en nuestro modelo probabilístico deje de ser cierto que la elección interna sea siempre *peor* (pospondremos la formalización del término hasta el capítulo siguiente, en el que estudiaremos las relaciones de orden entre procesos) que la elección externa.

Ello contradice en alguna forma el objetivo prioritario que nos habíamos impuesto, de procurar mantener al máximo las propiedades del modelo ordinario. Al respecto señalaremos que detectada la *irregularidad* tratamos de resolverla, pero encontramos que todo cambio en ese sentido de la definición de la semántica del operador de elección externa, nos acababa conduciendo a un operador no contínuo. En consecuencia, el único camino posible para mantener dicha propiedad, consistiría en *forzar* al operador de elección interna a ser *estricto.* Tal modificación forzada no nos pareció natural; por otra parte, reexaminada la cuestión encontramos que no cabía calificar a la propiedad perdida como de *naturalmente deseable* en un marco general, sino como de propiedad obtenida implícitamente en el marco

no-determinista no-probabilístico del modelo ordinario, a partir de otras consideraciones propias de dicho modelo. Por lo tanto la pérdida de dicha propiedad no nos parece de lamentar, si bien la constatación del hecho nos parece oportuno, pues ilustra algunas de las bases en las que se fundamenta la formalización de nuestro modelo probabilístico.

Capítulo 5

Semántica denotacional

La semántica de pruebas estudiada en el capítulo anterior define el significado de cada proceso de una forma implícita. Puede resultar interesante disponer de un mecanismo alternativo explícito, para definir dicho significado. Ello se logra por medio de una semántica denotacional, cuyo estudio constituirá la temática del presente capítulo.

Un primer paso en todo modelo denotacional consiste en establecer el dominio semántico (conjunto de objetos matemáticos con una estructura adecuada) donde la función semántica tomará sus valores. Posteriormente se definen dichos valores para cada constructor de procesos como una función entre los objetos del dominio.

Como vimos en el capítulo 3, utilizar como dominio semántico las trazas de los procesos no es suficiente para capturar el comportamiento de todos ellos. En consecuecia, nos basaremos en las ideas desarrolladas en el captítulo anterior, a la hora de construir el dominio semántico de procesos adecuado.

Una vez que tengamos definido el dominio de procesos, el mecanismo será el usual, es decir, estableceremos funciones entre los procesos (semánticos) que representen a cada una de las operaciones sintácticas, siendo el significado del operador recursión el menor punto fijo de la cadena de aproximaciones que empiezan en el menor proceso semántico, el cual, en línea con lo visto en el capítulo anterior, se corresponderá con la semántica del proceso divergente DIV.

Finalmente, una vez que tengamos definido el modelo denotacional de PCSP, comprobaremos que su potencia descriptiva es la misma que la del modelo de pruebas, estableciendo la equivalencia entre ambas semánticas.

5.1 Dominio de Procesos

Para establecer cuales serán los objetos que representarán a los procesos semánticos, observamos que cada paso de los cómputos de un proceso se puede descomponer en dos etapas diferenciadas: en primer lugar el proceso, tras una serie de decisiones internas, alcanza un *estado* estable, en el cual es posible ejecutar una serie de acciones; posteriormente, a partir de dicho estado, el medio exterior selecciona entre las acciones que constituyen el mismo, cual es la acción a ejecutar.

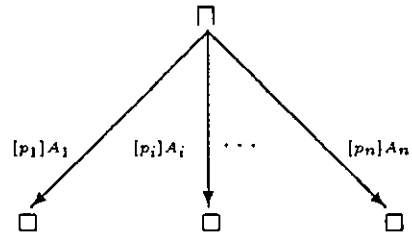
Así pues, la idea fundamental para definir el comportamiento de un proceso será la descripción de los estados que el mismo alcanza a lo largo de su evolución a lo largo del tiempo, de manera que si podemos caracterizar todos estos estados alcanzables tendremos definitivamente descrito el proceso. Naturalmente, en un medio probabilístico como en el que nos movemos, será preciso incorporar en el punto adecuado la información probabilística que rige el comportamiento de cada proceso.

Así, podemos describir un proceso probabilístico por medio de un árbol de decisiones sobre las acciones que pueden ser ejecutadas. Estas decisiones pueden ser de dos tipos, **internas y externas**, refiriéndose el calificativo en cuestión a quién será el sujeto que realizará dicha decisión. En el primer caso, será el sistema el encargado de tomar la decisión, consistiendo ésta en seleccionar un *estado* estable, en el que se ofrecen al medio una serie de acciones, según una determinada distribución de probabilidad. En cuanto a las decisiones externas consistirán en la elección por parte del medio exterior de la acción a ejecutar, una vez alcanzado un estado, de entre el conjunto de acciones que componen el mismo.

Dichas decisiones se van tomando en forma alternativa, de manera que los dos tipos de nodos, *internos* y *externos*, se irán sucediendo alternativamente a lo largo del árbol, comenzando siempre con un nodo interno como raíz del árbol.

De cada nodo interno surgen una serie de ramas que tienen como destinos nodos externos, y van etiquetadas con conjuntos de acciones diferentes y con la probabilidad de que sea seleccionado dicho conjunto como el siguiente estado estable.

La representación gráfica de un nodo interno se muestra en la siguiente figura:



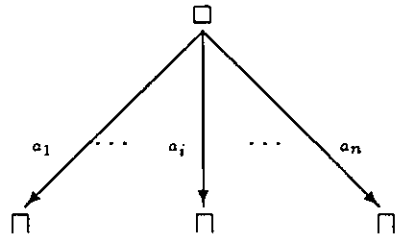
en donde

$$\forall i \; p_i > 0$$
$$\sum_{i=1}^{n} p_i \leq 1$$
$$i \neq j \Rightarrow A_i \neq A_j$$

Nótese que no exigimos que la suma de las probabilidades sea 1, permitiéndose que la misma pueda ser inferior. La diferencia entre dicho valor y la unidad, captura la probabilidad de divergencia en la situación representada por la raíz del árbol. Así, en el caso límite en el que $n = 0$, y por tanto la suma en cuestión es nula, obtendremos la representación (semántica) del proceso DIV. Por ello, en lo sucesivo nos permitiremos usar la misma notación DIV, para referirnos a dicho árbol trivial.

De cada nodo externo al que se llega por un arco etiquetado con el conjunto $A = \{a_1, \ldots, a_n\}$ y una probabilidad p_A cualquiera, surgirán a su vez ramificaciones etiquetadas con las distintas acciones que forman parte del estado correspondiente A. Evidentemente no existirán probabilidades en las etiquetas asociadas a estas ramas, ya que las elecciones que representan son deterministas, correspondiendo la decisión de cuál de sus acciones será la ejecutada, al medio exterior.

La representación gráfica de un nodo externo se muestra en la siguiente figura:



Ahora cuando un nodo externo es una hoja tenemos una oferta vacía que se corresponde con la semántica del proceso $STOP$.

Distinguimos por tanto el caso en el que un proceso alcanza el estado vacío ($\emptyset$), un estado estable donde no es posible ejecutar ninguna acción, como ocurre con el proceso $STOP$, del caso correspondiente al proceso DIV, en el que hay una incapacidad para alcanzar estado estable alguno.

Definición 5.1 (*Proceso Semántico*)
Definimos los procesos probabilísticos semánticos como los árboles no ordenados, finitos o infinitos, con dos tipos de nodos, que podemos generar (finita o infinitamente) con la siguiente gramática:

$$P ::= \prod_{A \in \mathcal{A}} [p_A] \, \Box_{a \in A} \, a.P$$

donde $\mathcal{A} \in \mathcal{P}(\Sigma)$, $\mathcal{A} \neq \emptyset$ y $\forall A \in \mathcal{A} \, p_A \neq 0$, $\sum_{A \subseteq \mathcal{A}} p_A \leq 1$.
Denotaremos por $\mathcal{P}$ al conjunto de todos los procesos semánticos. □

Observación: Remarcaremos que bajo un mismo nodo, tanto interno como externo, no pueden existir ramas repetidas, es decir etiquetadas de una misma forma.

Por otra parte recordamos el hecho de que si bien en la representación gráfica de los árboles existe un orden entre las distintas ramas, ello no es así en en su definición formal, pues en la misma se habla de conjuntos de hijos y no de listas de ellos.

En la figura 7 se muestran algunos ejemplos de (representaciones de) procesos semánticos.

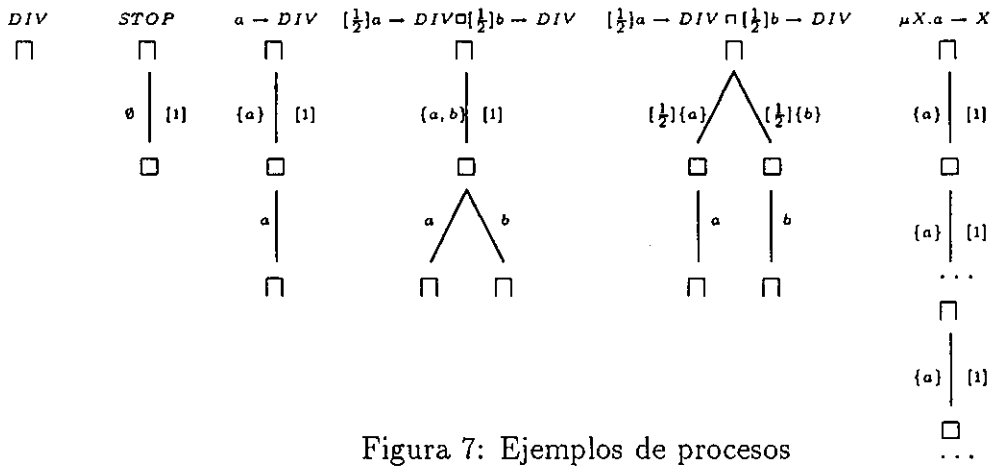


Figura 7: Ejemplos de procesos

Definición 5.2 (Probabilidad de un estado)
Sea P un proceso y $A \subseteq \Sigma$, definimos la probabilidad con la que P alcanza (inmediatamente) el estado A, y lo denotaremos por $p(P, A)$, como el valor numérico p_A con el que se alcanza un nodo externo etiquetado con A bajo la raíz de P, tomando como dicho valor 0 si A no pertenece al conjunto de etiquetas sobre hijos de dicha raíz. □

Definición 5.3 (Proceso tras una acción en un estado)
Sea P un proceso y $A \subseteq \Sigma$, definimos el proceso restante después de ejecutar la acción $a \in A$, y lo denotaremos por $P/(a, A)$, como el árbol que tiene como raíz el nodo interno que se alcanza tras el arco etiquetado con la acción a que parte del estado A bajo la raíz de P, de ser efectivamente A un estado de P. En caso contrario, tomamos $P/(a, A) = DIV$. □

Definición 5.4 (Probabilidad de un nodo externo)
Los nodos externos de cada proceso P están unívocamente determinados por la secuencia $\langle A_1 a_1 A_2 a_2 \ldots A_{n-1} a_{n-1} \rangle$, con $a_i \in A_i$ para todo $i \in \{1, \ldots, n-1\}$ que nos conduce al nodo interno padre correspondiente, junto con el estado A_n que etiqueta cada arco que sale de dicho nodo interno.

Entonces, dado un proceso P, y una secuencia y un estado que caracterizan a un posible nodo externo del mismo, definimos la probabilidad de que dicho proceso alcance el nodo externo representado por la secuencia s y el estado A, que

denotamos por $p(P, s, A)$, en la forma siguiente

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

□

De cara a poder comparar los procesos, y así poder definir la semántica de la recursión por medio de la técnica del menor punto fijo, introducimos una relación de orden entre los procesos.

Definición 5.5 (Orden entre procesos)
Sean P y Q procesos probabilísticos semánticos. Decimos que $P \sqsubseteq Q$ sii todo nodo externo del proceso P es también nodo externo de Q y se alcanza en Q con igual o mayor probabilidad que en P. Es decir si para toda secuencia s, la probabilidad de alcanzar el estado A tras la secuencia s es menor en el proceso P que en el proceso Q. Es decir

$$p(P, s, A) \leq p(Q, s, A)$$

□

Obsérvese que en la anterior definición podemos omitir, por quedar de hecho implícitas, la restricción a estados alcanzables de P y la condición de que dichos estados sean alcanzables en Q.

Proposición 5.1 La relación $\sqsubseteq$ entre procesos es una relación de orden, que tiene a DIV como menor elemento.

Demostración: Trivial. □

Teorema 5.1 $(\mathcal{P}, \sqsubseteq)$ es un orden parcial completo (cpo).

Demostración:

- Tiene elemento mínimo $\perp$. Se trata del árbol que verifica

$$p(\perp, A) = 0 \quad \forall A \subseteq \Sigma$$

que es obvio que no tiene más nodo que su raíz, correspondiendose por tanto con el árbol DIV. Evidentemente tenemos que $\forall P \in \mathcal{P} \quad \perp \sqsubseteq P$.

- Toda cadena tiene cota superior mínima.

  Sea $\{P_i\}_{i \in I}$ una cadena. Entonces el árbol $\sqcup P_i$, es aquél cuyos estados alcanzables quedan definidos en la forma

$$\begin{aligned} p(\sqcup P_i, A) &= \lim p(P_i, A) \\ p(\sqcup P_i, s, A) &= \lim p(P_i, s, A) \end{aligned}$$

  Se trata en efecto de la cota superior mínima de la cadena, pues:

  1. $\sqcup P_i$ está bien definido, pues al formar los procesos P_i una cadena, los valores $p(P_i, A)$ y $p(P_i, s, A)$ forman una sucesión creciente y acotada por 1. Luego existirá el límite, que será menor o igual a 1.

  2. $\sqcup P_i$ es cota superior. Es decir $\forall i \quad P_i \sqsubseteq \sqcup P_i$

     Sea P_i un proceso de la cadena y A un estado alcanzable tras la secuencia s, entonces A deberá ser alcanzable por P_j para todo $j \geq i$. Por lo tanto, al formar los valores $p(P_j, s, A)$ una sucesión creciente, se obtiene que $p(P_j, s, A) \leq \lim_i p(P_i, s, A)$, por lo que es cota superior.

  3. $\sqcup P_i$ es mínima.

     Sea Q un proceso tal que $\forall i \quad P_i \sqsubseteq Q$, entonces si A es un estado alcanzable tras la secuencia s por todos los procesos de la cadena a partir de un determinado proceso i, tenemos que A será alcanzable tras s por el proceso Q, y además $\forall j \geq i \quad p(P_j, s, A) \leq p(Q, s, A)$, por tanto

$$\begin{array}{lll} p(P_j, s, A) & \leq p(Q, s, A) & \forall j \geq i \\ \lim_{j \geq i} p(P_j, s, A) & \leq p(Q, s, A) & \\ p(\sqcup P_i, s, A) & \leq p(Q, s, A) & \\ \sqcup P_i & \sqsubseteq Q & \end{array}$$

□

5.2 Semántica Denotacional de PCSP

En la sección anterior hemos definido el dominio semántico para los procesos de PCSP, continuaremos en esta sección, de la manera usual, definiendo una colección de operadores semánticos entre elementos de este dominio que representarán cada una de las operaciones de la sintaxis definida para el lenguaje. Como es usual, denotaremos con $[\![P]\!]$ la semántica denotacional de cada proceso P.

5.2.1 STOP y DIV

El proceso $[\![STOP]\!]$ viene definido por un árbol con un nodo interno y otro externo bajo él; la rama que los une está etiquetada con el conjunto $\emptyset$ y la probabilidad 1. Por tanto

$$\begin{aligned} p([\![STOP]\!], \emptyset) &= 1 \\ p([\![STOP]\!], A) &= 0 \quad \text{si} \quad P \neq \emptyset \\ p([\![STOP]\!], \langle\rangle, \emptyset) &= 1 \\ p([\![STOP]\!], s, A) &= 0 \quad \text{si} \quad s \neq \langle\rangle \text{ ó } A \neq \emptyset \end{aligned} \tag{1}$$

Por otra parte, el proceso $[\![DIV]\!]$ viene definido por el árbol que sólo tiene un nodo interno del que no parte ninguna rama. Por tanto

$$\begin{aligned} p([\![DIV]\!], A) &= 0 \quad \forall A \subseteq \Sigma \\ p([\![DIV]\!], s, A) &= 0 \quad \forall s \ \forall P \end{aligned}$$

Las funciones constantes que definen la semántica de los procesos constantes son, de forma trivial, monótonas y continuas. Para el resto de las funciones que definen la semántica de los restantes operadores del lenguaje, deberemos probar en lo sucesivo su monotonía y continuidad.

5.2.2 Prefijo

Definición 5.6 Siendo $a \in \Sigma$ y $\mathcal{P}$ el conjunto de procesos semánticos, definimos la función $a \rightarrow: \mathcal{P} \longrightarrow \mathcal{P}$, cuya aplicación notaremos en la forma $a \rightarrow P$, en la

forma siguiente: Para cada $P \in \mathcal{P}$ el proceso $a \rightarrow P$ viene dado por el árbol de cuya raíz parte una única rama etiquetada con el conjunto $\{a\}$ y la probabilidad 1, que la une a un nodo externo bajo el cual existe un único nodo interno, siendo el subárbol que tiene a este nodo por raíz, idéntico al árbol P.

En cosecuencia, los nodos externos del árbol $a \rightarrow P$ serán accesibles tras una secuencia que comenzará con el par $(\{a\}, a)$ y continuará con una secuencia s que defina un nodo externo del árbol P. Tendremos entonces

$$\begin{aligned} p(a \rightarrow P, \{a\}) &= 1 \\ p(a \rightarrow P, A) &= 0 \quad \text{Si} \quad A \neq \{a\} \\ a \rightarrow P/(a, \{a\}) &= P \\ a \rightarrow P/(b, A) &= DIV \quad \text{Si} \quad A \neq \{a\} \ \text{o} \ b \neq a \end{aligned} \tag{2}$$

□

Proposición 5.2 La función prefijo es monótona y continua.

Demostración:

MONOTONÍA: Supongamos $P \sqsubseteq Q$, entonces para todo nodo externo correspondiente al estado A alcanzado tras s por el proceso P, existe un nodo externo correspondiente a A alcanzable tras s por Q, tal que $p(P, s, A) \leq p(Q, s, A)$. Tenemos entonces que para cada nodo externo correspondiente a un estado A alcanzable tras $\langle(\{a\}, a)\rangle s$ por el proceso $a \rightarrow P$ se tiene

$$\begin{aligned} p(a \rightarrow P, \langle(\{a\}, a)\rangle s, A) &= 1 \cdot p(P, s, A) \\ &\leq 1 \cdot p(Q, s, A) \\ &= p(a \rightarrow Q, \langle(\{a\}, a)\rangle s, A) \end{aligned}$$

CONTINUIDAD: Dada $\{P_i\}_{i \in I}$ una cadena ascendente de procesos, como el prefijo es monótono, tenemos entonces que $\{a \rightarrow P_i\}_{i \in I}$ es una cadena ascendente de procesos. Sea entonces s una secuencia tras la que se puede alcanzar el estado A por parte del proceso $\sqcup(a \rightarrow P_i)$. Tenemos en primer lugar que la secuencia s es de la forma $\langle(\{a\}, a)\rangle s'$, para alguna secuencia s'. Esto es así, pues todos los procesos de la cadena comienzan por ese par. Entonces, si en el límite no lo hiciera debería haber algún proceso de la cadena que no comenzase por la misma. Ahora bien, si

el nodo externo de estado A es alcanzable tras la secuencia s' por el proceso $\sqcup P_i$ con una probabilidad $p(\sqcup P_i, s', A)$, tenemos entonces, para todo nodo externo de $\sqcup(a \rightarrow P_i)$

$$\begin{aligned} p(\sqcup(a \rightarrow P_i), s, A) &= p(\sqcup P_i, (\{\{a\}, a\})s', A) \\ &= p(a \rightarrow \sqcup P_i, s, A) \end{aligned}$$

Por lo que se concluye que $\sqcup(a \rightarrow P_i) = a \rightarrow \sqcup P_i$ □

Definición 5.7 Sea P un proceso de la forma $P = a \rightarrow P'$, definimos su semántica denotacional en la forma $[\![P]\!] = a \rightarrow [\![P']\!]$. □

5.2.3 Elección interna

Siendo P y Q dos procesos, el proceso $[p]P \sqcap [1-p]Q$ vendrá dado por un árbol, cuyos estados resultan de la *unión ponderada* entre los de los dos componentes.

Definición 5.8 Definimos la operación de elección interna entre procesos probabilísticos $\sqcap : [0,1] \times \mathcal{P} \times \mathcal{P} \longrightarrow \mathcal{P}$, cuya aplicación notaremos en la forma $[p]P \sqcap [1-p]Q$, en la forma siguiente: Siendo P y Q dos procesos, los estados alcanzables por el proceso $[p]P \sqcap [1-p]Q$ serán aquéllos alcanzables por uno u otro de ellos, si bien con la probabilidad que resulte de la ponderación de las probabilidades correspondientes con los factores p y $1-p$. Más exactamente, si un estado A es alcanzable por P con probabilidad p_A, pero no lo es por Q, entonces este estado A será alcanzable por $[p]P \sqcap [1-p]Q$ con una probabilidad $p \cdot p_A$. Análogamente sucede en el caso contrario, siendo en este caso la probabilidad resultante $(1-p) \cdot q_A$. Por último, si es alcanzable por ambos componentes, con probabilidades respectivas p_A y q_A, el estado será alcanzable por $[p]P \sqcap [1-p]Q$ con probabilidad $p \cdot p_A + (1-p)q_A$.

En consecuencia se tiene siempre

$$p([p]P \sqcap [1-p]Q, A) = p \cdot p(P, A) + (1-p) \cdot p(Q, A)$$

Dicho resultado nos da pie para generalizar la definición de modo que queden cubiertos la totalidad de los nodos. Al respecto, definimos la probabilidad de

alcanzar cada estado A tras una secuencia s por parte del proceso $[p]P \sqcap [1-p]Q$, mediante

$$p([p]P \sqcap [1-p]Q, s, A) \;=\; p \cdot p(P, s, A) + (1-p) \cdot p(Q, s, A) \qquad (3)$$

□

Proposición 5.3 Fijado un cierto $p \in [0,1]$ tenemos que la función elección interna es monótona y continua para cada uno de sus argumentos en $\mathcal{P}$.

Demostración: Lo veremos sólo para el primer argumento, siendo la comprobación análoga para el segundo.

MONOTONIA: Sean P, P' tales que $P \sqsubseteq P'$. Por definición tenemos que, si tras la secuencia s el proceso P alcanza un estado A con una probabilidad $p(P, s, A)$, entonces

$$p(P, s, A) \leq p(P', s, A)$$

Por lo que

$$p([p]P \sqcap [1-p]Q, s, A) = p \cdot p(P, s, A) + (1-p) \cdot p(Q, s, A)$$

$$\leq \quad p \cdot p(P', s, A) + (1-p) \cdot p(Q, s, A) = p([p]P' \sqcap [1-p]Q, s, A)$$

Puesto que ello es cierto para todo estado, tenemos que

$$[p]P \sqcap [1-p]Q \;\sqsubseteq\; [p]P' \sqcap [1-p]Q$$

CONTINUIDAD: Sea $\{P_i\}_{i \in I}$ una cadena de procesos, entonces por la monotonía tenemos que $\{[p]P_i \sqcap [1-p]Q\}_{i \in I}$ es también una cadena, y en el límite, la probabilidad de alcanzar un nodo externo correspondiente al estado A tras la secuencia s es

$$\begin{aligned} p(\sqcup [p]P_i \sqcap [1-p]Q, s, A) &= \lim_i p([p]P_i \sqcap [1-p]Q, s, A) \\ &= \lim_i p \cdot p(P_i, s, A) + (1-p) \cdot p(Q, s, A) \\ &= p \cdot \lim_i p(P_i, s, A) + (1-p) \cdot p(Q, s, A) \\ &= p \cdot p(\sqcup P_i, s, A) + (1-p) \cdot p(Q, s, A) \\ &= p([p] \sqcup P_i \sqcap [1-p]Q, s, A) \end{aligned}$$

De donde concluímos

$$\sqcup([p]P_i \sqcap [1-p]Q) = [1-p]\sqcup P_i \sqcap [1-p]Q$$

□

Definición 5.9 Sean P, Q procesos y $p \in [0,1]$, definimos la semántica denotacional de $[p]P \sqcap [1-p]Q$ como el árbol dado por $[p][\![P]\!] \sqcap [1-p][\![Q]\!]$. □

5.2.4 Elección externa

Siendo P y Q dos procesos, definiremos el proceso $[p]P\Box[1-p]Q$, al igual que en el caso anterior, describiendo su conjunto de nodos externos, junto con la probabilidad de llegar a cada uno de ellos.

Comenzaremos estudiando los estados a los que se llega directamente desde la raíz. Como tales parece razonable tomar las uniones de los respectivos estados alcanzables por los procesos componentes, pues si el proceso P está en disposición de ejecutar las acciones del conjunto A y el proceso Q lo está de ejecutar las de B, tenemos entonces que el proceso $[p]P\Box[1-p]Q$ podrá ejecutar las acciones de $A \cup B$.

La probabilidad con la que se alcanza un estado tal del proceso composición será la suma de los productos de las probabilidades con las que los componentes alcanzan respectivamente estados A y B tales que $A \cup B = C$:

$$p([p]P\Box[1-p]Q, C) = \sum_{A \cup B = C} p(P, A) \cdot p(Q, B)$$

Consecuencia inmediata de esta definición es que la función elección externa es estricta, pues si uno de los dos procesos componentes no puede alcanzar ningún estado, es decir es DIV, la composición tampoco podrá alcanzar ningún estado. Es decir

$$[p]P\Box[1-p]DIV \;=\; DIV$$

Nos queda entonces, para definir completamente el proceso, el estudio de los nodos por debajo del primer piso de nodos externos. Al respecto, observamos que si X es un estado alcanzable tras la secuencia no vacía s por el proceso $[p]P\Box[1-p]Q$, este

nodo podrá haber sido alcanzado bien por una rama correspondiente al proceso P, o bien por una correspondiente al proceso Q.

Procede entonces una distinción de casos según sea la primera componente de la secuencia s. Tomemos $s = \langle (A, a) \rangle s'$ con lo que el conjunto A nos indica el estado estable alcanzado por el proceso $[p]P \Box [1-p]Q$ desde el que se ha producido la ejecución de la primera acción a. Sea p_A la probabilidad con la que se ha alcanzado dicho estado. Según hemos definido, el estado A alcanzado lo habrá sido por haber alcanzado P un estado B, y Q otro C tales que $A = B \cup C$. Entonces, tras s el proceso composición alcanzará un estado X bien porque P o porque Q habrán alcanzado dicho estado X tras las respectivas secuencias s_B ó s_C que son idénticas a s, salvo que su primer par ha de ser susituído respectivamente por (B, a) o por (C, a). Es decir $s_B = \langle (B, a) \rangle s'$, y $s_C = \langle (C, a) \rangle s'$.

En definitiva se obtiene la siguiente definición

$$p([p]P \Box [1-p]Q, s, X) = \begin{cases} \sum\limits_{\substack{A=B\cup C \\ a \in B - C}} p(P, s_B, X) \cdot p(Q, C) + \sum\limits_{\substack{A \cup B = C \\ a \in C - B}} p(Q, s_C, X) \cdot p(P, B) + \\ \sum\limits_{\substack{A \cup B = C \\ a \in A \cap B}} p \cdot p(P, s_B, X) \cdot p(Q, C) + (1-p) \cdot p(Q, s_C, X) \cdot p(P, B) \end{cases} \tag{4}$$

Es decir, si la primera acción ejecutada sólo pertenece a uno de los dos estados compuestos, se toma como continuación la del correspondiente proceso; mientras que si pertenece a ambos se hace una elección no-determinista probabilística, en base a las probabilidades indicadas en la composición.

Proposición 5.4 Fijado un cierto $p \in [0, 1]$ tenemos que la función elección externa es monótona y continua para cada uno de sus argumentos en $\mathcal{P}$.

Demostración: Al igual que en el caso anterior, lo veremos sólo para el primer argumento

MONOTONIA: Sean P, P' tales que $P \sqsubseteq P'$. Por definición tenemos que si A es alcanzable tras s por el proceso P entonces $p(P, s, A) \leq p(P', s, A)$.

Además, para todos los estados A y B tenemos que $p(P, A) \leq p(P', A)$ y que $p(Q, B) \leq p(Q, B)$.

Entonces

$$
\begin{aligned}
& p([p]P\Box[1-p]Q, s, X) \\
&= \sum_{\substack{A=B\cup C \\ a\in B-C}} p(P, s_B, X)\cdot p(Q, B) \;+\; \sum_{\substack{A\cup B=C \\ a\in C-B}} p(Q, s_C, X)\cdot p(P, C) \;+ \\
&\quad \sum_{\substack{A\cup B=C \\ a\in A\cap B}} p\cdot p(P, s_B, X)\cdot p(Q, B) \;+(1-p)\cdot p(Q, s_C, X)\cdot p(P, C) \\
&\leq \sum_{\substack{A=B\cup C \\ a\in B-C}} p(P', s_B, X)\cdot p(Q, B) \;+\; \sum_{\substack{A\cup B=C \\ a\in C-B}} p(Q, s_C, X)\cdot p(P', C) \;+ \\
&\quad \sum_{\substack{A\cup B=C \\ a\in A\cap B}} p\cdot p(P', s_B, X)\cdot p(Q, B) \;+\; (1-p)\cdot p(Q, s_C, X)\cdot p(P', C) \\
&= p([p]P'\Box[1-p]Q, s, X)
\end{aligned}
$$

CONTINUIDAD: Sea $\{P_i\}_{i\in I}$ una cadena de procesos, entonces por monotonía $\{[p]P_i\Box[1-p]Q\}_{i\in I}$ es una cadena, y en el límite, la probabilidad de alcanzar un estado X tras una secuencia s es

$$
\begin{aligned}
& p(\sqcup[p]P_i\Box[1-p]Q, s, X) \;=\; \lim_i p([p]P_i\Box[1-p]Q, s, X) \\
&= \lim_i \sum_{\substack{A=B\cup C \\ a\in B-C}} p(P_i, s_B, X)\cdot p(Q, B) \;+\; \sum_{\substack{A\cup B=C \\ a\in C-B}} p(Q, s_C, X)\cdot p(P_i, C) \;+ \\
&\quad \sum_{\substack{A\cup B=C \\ a\in A\cap B}} p\cdot p(P_i, s_B, X)\cdot p(Q, B) \;+\; (1-p)\cdot p(Q, s_C, X)\cdot p(P_i, C) \\
&= \sum_{\substack{A=B\cup C \\ a\in B-C}} \lim_i p(P_i, s_B, X)\cdot p(Q, B) \;+\; \sum_{\substack{A\cup B=C \\ a\in C-B}} p(Q, s_C, X)\cdot \lim_i p(P_i, C) \;+ \\
&\quad \sum_{\substack{A\cup B=C \\ a\in A\cap B}} p\cdot \lim_i p(P_i, s_B, X)\cdot p(Q, B) \;+\; (1-p)\cdot p(Q, s_C, X)\cdot \lim_i p(P_i, C) \\
&= \sum_{\substack{A=B\cup C \\ a\in B-C}} p(\sqcup P_i, s_B, X)\cdot p(Q, B) \;+\; \sum_{\substack{A\cup B=C \\ a\in C-B}} p(Q, s_C, X)\cdot p(\sqcup P_i, C) \;+ \\
&\quad \sum_{\substack{A\cup B=C \\ a\in A\cap B}} p\cdot p(\sqcup P_i, s_B, X)\cdot p(Q, B) \;+\; (1-p)\cdot p(Q, s_C, X)\cdot p(\sqcup P_i, C) \\
&= p([p]\sqcup P_i\Box[1-p]Q, s, X)
\end{aligned}
$$

□

Definición 5.10 Siendo P, Q procesos y $p\in[0,1]$, definimos la semántica denotacional de $[p]P\Box[1-p]Q$ como el árbol dado por $[p][\![P]\!]\Box[1-p][\![Q]\!]$. □

5.2.5 Composición paralela

Comenzaremos de nuevo obteniendo los estados inmediatamente bajo la raíz del proceso $[p]P\|_A[1-p]Q$, y las probabilidades con las que se alcanzan. Como quiera que en este caso la casuística es más compleja, comenzaremos estudiando los casos particulares más representativos, para luego, desde ellos, enunciar la que será la definición general del operador. Supongamos en primer lugar, que el estado alcanzable B es disjunto con el conjunto de sincronización: es decir $A \cap B = \emptyset$. Entonces un razonamiento análogo al realizado en el caso de la elección externa nos conduciría como primera aproximación para definir la probabilidad con la que se alcanza un estado tal, a

$$p([p]P\|_A[1-p]Q, B) = \sum_{C \cup D = B} p(P, C) \cdot p(Q, D)$$

Ahora bien, dicha definición no es en general correcto, pues es preciso tener en cuenta el hecho de que cuando P alcanza un estado que incluye algunas acciones del alfabeto de sincronización A, y el estado alcanzado por Q no las incluye, estas acciones no serán tomadas en consideración ante la imposibilidad de sincronizar para ejecutarlas, con lo que tenemos más formas de alcanzar un estado como el correspondiente a B. Naturalmente, lo mismo puede decirse respecto al proceso Q; y más en general, cuando los dos procesos alcanzan estados conteniendo sendos conjuntos disjuntos del conjunto de sincronización. En definitiva obtenemos

$$p([p]P\|_A[1-p]Q, B) = \sum_{C \cup D = B} \sum_{\substack{X_1, X_2 \subseteq A \\ X_1 \cap X_2 = \emptyset}} p(P, C \cup X_1) \cdot p(Q, D \cup X_2)$$

Por otra parte, si tenemos un estado B, alcanzable por $[p]P\|_A[1-p]Q$, que esté contenido en el conjunto de sincronización, es decir $B \subseteq A$, será necesario que dicho estado sea alcanzado simultáneamente tanto por P como por Q, con lo que obtendríamos, de nuevo en una primera aproximación,

$$p([p]P\|_A[1-p]Q, B) = p(P, B) \cdot p(Q, B)$$

Ahora bien, al igual que en el caso anterior, como no hemos tenido en cuenta aquellas acciones que sincronizan y que sólo pueden ser ejecutadas por uno de los

procesos, a la probabilidad de alcanzar un estado habrá que sumar la probabilidad de alcanzar aquellos estados mayores, cuyas acciones extras no puedan ser ejecutadas por el otro proceso. En definitiva obtendríamos

$$p([p]P\|_A[1-p]Q, B) = \sum_{\substack{X_1, X_2 \subseteq A-B \\ X_1 \cap X_2 = \emptyset}} p(P, B \cup X_1) \cdot p(Q, B \cup X_2)$$

Finalmente, para un estado arbitrario B obtendríamos la siguiente definición general:

$$p([p]P\|_A[1-p]Q, B) = \sum_{\substack{C \cup D = B \\ C \cap D \cap A = B \cap A}} \sum_{\substack{X_1, X_2 \subseteq A-B \\ X_1 \cap X_2 = \emptyset}} p(P, C \cup X_1) \cdot p(Q, D \cup X_2)$$

De nuevo para completar la descripción de $[p]P\|_A[1-p]Q$ hemos de definir la probabilidad con la que el proceso ejecuta una secuencia $s = \langle (a, B) \rangle s'$, en la que el conjunto B representa el primer estado alcanzado bajo la raíz, y tras dicha secuencia alcanza un cierto estado Z. En este caso, tras razonamientos análogos a los hechos para calcular la probabilidad de cada estado bajo la raíz, obtenemos la siguiente fórmula general:

$$p([p]P\|_A[1-p]Q, \langle (a, B) \rangle s', C) =$$

$$\sum_{\substack{C \cup D = B \\ C \cap D \cap A = B \cap A}} \sum_{\substack{X_1, X_2 \subseteq A-B \\ X_1 \cap X_2 = \emptyset}} p(P, C \cup X_1) \cdot p(Q, D \cup X_2) \cdot$$

$$\begin{cases} p([p]P/(a, C \cup X_1)\|_A[1-p]Q, s', Z) & \text{Si } a \in C - D \\ p([p]P\|_A[1-p]Q/(a, D \cup X_2), s', Z) & \text{Si } a \in D - C \\ p \cdot p([p]P/(a, C \cup X_1)\|_A[1-p]Q, s', Z) + & \\ (1-p) \cdot p([p]P\|_A[1-p]Q/(a, D \cup X_2), s', Z) & \text{Si } a \in D \cap C - A \\ p([p]P/(a, C \cup X_1)\|_A[1-p]Q/(a, D \cup X_2), s', Z) & \text{Si } a \in C \cap D \cap A \end{cases} \quad (5)$$

Proposición 5.5 La función composición paralela es monótona y continua.

Demostración: Nos centraremos, de nuevo, en el primer argumento de la composición.

MONOTONIA: Sean P, Q, P' tales que $P \sqsubseteq P'$. En consecuencia tenemos para toda secuencia s y todo conjunto A

$$\begin{aligned} p(P, A) &\leq p(P', A) \\ p(P, s, A) &\leq p(P', s, A) \end{aligned}$$

por lo que en primer lugar obtenemos

$$\begin{aligned} p([p]P\|_A[1-p]Q, B) &= \sum_{C,D}\sum_{X_1,X_2} p(P, C\cup X_1))\ \cdot\ p(Q, D\cup X_2) \\ &\leqq \sum_{C,D}\sum_{X_1,X_2} p(P', C\cup X_1))\ \cdot\ p(Q, D\cup X_2) \\ &= p([p]P'\|_A[1-p]Q, B) \end{aligned}$$

En lo que hace referencia al resto de estados externos, debemos probar que para todo estado X alcanzable por el proceso $[p]P\|_A[1-p]Q$ tras la ejecución de la secuencia $s = \langle(a, C)\rangle s'$ se cumple

$$p([p]P\|_A[1-p]Q, s, X) \leq p([p]P'\|_A[1-p]Q, s, X)$$

Ello se prueba por inducción respecto de la longitud de la secuencia. En realidad lo que hemos probado más arriba no es otra cosa que el caso correspondiente a la secuencia vacía, es decir el caso base de la demostración por inducción. Consideremos ahora una secuencia $s = \langle(a, C)\rangle s'$. Para la misma se tiene:

$$\begin{aligned} & p([p]P\|_A[1-p]Q, \langle(a, B)\rangle s', X) \\ &= \sum_{C,D}\sum_{X_1,X_2} p(P, C\cup X_1))\ \cdot p(Q, D\cup X_2)\ \cdot \\ &\quad \begin{cases} p([p]P/(a, C\cup X_1)\|_A[1-p]Q, s', X) & \text{Si } a\in C-D \\ p([p]P\|_A[1-p]Q/(a, D\cup X_2), s', X) & \text{Si } a\in D-C \\ p\cdot p([p]P/(a, C\cup X_1)\|_A[1-p]Q, s', X) + & \\ \quad (1-p)\cdot p([p]P\|_A[1-p]Q/(a, D\cup X_2), s', X) & \text{Si } a\in D\cap C-A \\ p([p]P/(a, C\cup X_1)\|_A[1-p]Q/(a, D\cup X_2), s', X) & \text{Si } a\in C\cap D\cap A \end{cases} \\ &\leq \sum_{C,D}\sum_{X_1,X_2} p(P', C\cup X_1))\ \cdot p(Q, D\cup X_2)\ \cdot \\ &\quad \begin{cases} p([p]P'/(a, C\cup X_1)\|_A[1-p]Q, s', X) & \text{Si } a\in C-D \\ p([p]P'\|_A[1-p]Q/(a, D\cup X_2), s', X) & \text{Si } a\in D-C \\ p\cdot p([p]P'/(a, C\cup X_1)\|_A[1-p]Q, s', X)+ & \\ \quad (1-p)\cdot p([p]P'\|_A[1-p]Q/(a, D\cup X_2), s', X) & \text{Si } a\in D\cap C-A \\ p([p]P'/(a, C\cup X_1)\|_A[1-p]Q/(a, D\cup X_2), s', X) & \text{Si } a\in C\cap D\cap A \end{cases} \\ &= p([p]P'\|_A[1-p]Q, \langle(a, B)\rangle s', X) \end{aligned}$$

en cuyo paso crucial, que es el que corresponde a la desigualdad, hemos utilizado la hipótesis de inducción, junto con la monotonía (trivial) del operador *tras*.

CONTINUIDAD: El razonamiento es similar al anterior. Sea $\{P_i\}_{i\in I}$ una cadena de procesos, entonces $\{[p]P_i\|_A[1-p]Q\}_{i\in I}$ es una cadena, y en el límite, la probabilidad de alcanzar un estado B viene dada por

$$\begin{aligned} p(\sqcup([p]P_i\|_A[1-p]Q), B) &= \lim_i p([p]P_i\|_A[1-p]Q, B) \\ &= \lim \sum_{C,D} \sum_{X_1,X_2} p(P_i, C\cup X_1)) \cdot p(Q, D\cup X_2) \\ &= \sum_{C,D} \sum_{X_1,X_2} \lim p(P_i, C\cup X_1)) \cdot \lim p(Q, D\cup X_2) \\ &= \sum_{C,D} \sum_{X_1,X_2} p(\sqcup P_i, C\cup X_1)) \cdot p(\sqcup Q, D\cup X_2) \\ &= p([p]\sqcup P_i\|_A[1-p]\sqcup Q, B) \end{aligned}$$

Por otra parte, y de nuevo por inducción respecto de la longitud de la secuencia s, podemos probar

$$p(\sqcup([p]P_i\|_A[1-p]Q), s, X) \;=\; p([p]\sqcup P_i\|_A[1-p]\sqcup Q, s, X)$$

□

Definición 5.11 Siendo P, Q procesos, $p\in[0,1]$, y $A\subseteq\Sigma$, definimos la semántica denotacional del proceso $[p]P\|_A[1-p]Q$ por medio del árbol dado por $[p][\![P]\!]\|_A[1-p][\![Q]\!]$. □

5.2.6 Ocultación

Como ya hemos visto, $P\backslash(a,q)$ representa un proceso que se comporta en principio como P pero en el que no son visibles, ni por tanto influenciables desde el exterior, las apariciones de la acción a.

Para definir este proceso $P\backslash(a,q)$, calcularemos en primer lugar la probabilidad de que el mismo alcance cualquier estado $A\subseteq\Sigma$. Para ello, como ya hicimos en el capítulo anterior, procederemos a explorar por niveles el árbol que representa al proceso P, definiendo una secuencia de aproximaciones al proceso $P\backslash(a,q)$, cuyos elementos representaremos por $P\backslash_n(a,q)$.

Cada una de dichas aproximaciones se define explorando el proceso P hasta la profundidad n. En concreto tomamos

- En la primera aproximación no analizamos en absoluto al proceso dado, por lo que obtenemos $P\backslash_0(a,q) = DIV$ o lo que es lo mismo

$$p(P\backslash_0(a,q), A) = 0$$

- Cuando analizamos sólo el primer nivel del árbol P tenemos que el proceso en el que se ha ocultado una acción alcanzará un estado A (en el que obviamente no estará la acción oculta), si el mismo podía ser alcanzado antes de la ocultación, con lo que tenemos

$$p(P\backslash_1(a,q), A) = p(P, A)$$

- Si exploramos un segundo nivel de P, tendremos que el proceso $P\backslash_2(a,q)$ seguirá alcanzando un estado A cuando el mismo se alcance directamente por P, como en el caso anterior. Ahora bien, en esta ocasión podrá ocurrir que se alcance un cierto estado B que contenga la acción a oculta y que tras la ejecución de la misma (lo que ocurrirá con probabilidad q) se alcance el estado A. Finalmente otra posibilidad es que se alcance un estado B, conteniendo la acción a, y tal que $B - \{a\} \subseteq A$, de forma que tras la ejecución de la acción a sea alcanzado un estado C tal que $B \cup C = A$.

  Entendemos en este caso que se ha desestimado en principio la ejecución de la acción interna a, lo que sucederá con probabilidad $1-q$, por lo quedamos entonces en disposición de ejecutar el resto de las acciones de B. Ahora bien, si el medio exterior está interesado en ejecutar una acción de $C - B$, donde C es un estado alcanzable tras la ejecución de la acción interna a, podrá conseguirlo imponiendo al sistema la ejecución de la acción a, aún cuando inicialmente había sido desestimada. En consecuencia tenemos

$$\begin{aligned} p(P\backslash_2(a,q), A) \;=\; & \\ & p(P,A) \;+\; q \cdot \Big(\sum_{a \in B \subseteq \Sigma} p(P,B) \cdot p(P/(a,B), A) \Big) \;+ \\ & (1-q) \cdot \Big(\sum_{a \in B \subseteq A \cup \{a\}} p(P,B) \cdot \sum_{B \cup C = A} p(P/(a,B), C) \end{aligned}$$

- Razonando de forma similar para una cantidad aritraria n de niveles, obtenemos

$$p(P\backslash_n(a,q), A) \;=$$

$$p(P,A) \;+\; q\cdot(\sum_{a\in B\subseteq\Sigma} p(P,B)\cdot p(P/(a,B)\backslash_{n-1}(a,q),A)) \;+$$
$$(1-q)\cdot(\sum_{a\in B\subseteq A\cup\{a\}} p(P,B)\cdot\sum_{B\cup C=A} p(P/(a,B)\backslash_{n-1}(a,q),C)$$

Finalmente definimos

$$p(P\backslash(a,q),A) \;=\; \lim_n p(P\backslash_n(a,q),A)$$

La definición es correcta ya que este límite existe, puesto que la sucesión formada por los valores $\{p(P\backslash_0(a,q),A),\ldots,p(P\backslash_n(a,q),A),\ldots\}$ es creciente, ya que cada uno de los mismos es una suma que particularmente incluye los sumandos del término anterior; y además cada valor está acotado por 1, como ya comprobamos en el capítulo anterior. Luego existe el límite, y al ser menor o igual a uno, podemos entenderlo como una probabilidad.

Si tomamos límites en los dos lados de la expresión, obtenemos la siguiente expresión recurrente:

$$\begin{aligned} p(P\backslash(a,q),A) \;=\; & p(P,A)+q\cdot(\sum_{a\in B\subseteq\Sigma} p(P,B)\cdot p(P/(a,B)\backslash(a,q),A)) \;+ \\ & (1-q)\cdot(\sum_{a\in B\subseteq A\cup\{a\}} p(P,B)\cdot\sum_{B\cup C=A} p(P/(a,B)\backslash(a,q),C) \end{aligned}$$

que sin embargo no puede tomarse como definición implícita del operador de ocultación (en lo que se refiere a los estados bajo la raíz), al no estar garantizada en general la unicidad de la solución. No obstante, dicha ecuación sí que admite una lectura "constructiva (equivalente de hecho a la formalizada por las aproximaciones), que nos conduce a la definición previamente dada, la cual consiste en partir de cero e ir sumando cada uno de los sumandos que vayamos conociendo para las expresiones de ocultación que aparecen en el lado derecho de la igualdad. Es fácil concluir que obtenemos así la menor solución (respecto del orden usual sobre los reales) de las que pueda tener la ecuación en cuestión.

Para completar la descripción del proceso $P\backslash(a,q)$, debemos definir la probabilidad de que, tras ejecutar una secuencia s que comience por un par (b,A), se

alcance un cierto estado D. Razonando de forma análoga a como lo hemos hecho en el anterior caso particular, obtenemos la definición implícita

$$\begin{aligned} p(P\backslash(a,q),\langle b,A\rangle s',D) \;=\; & p(P/(b,A)\backslash(a,q),s',D) \;+ \\ & q\cdot(\sum_{a\in B\subseteq\Sigma} p(P,B)\cdot p(P/(a,B)\backslash(a,q),\langle b,A\rangle s',D)) \;+ \\ & (1-q)\cdot(\sum_{a\in B\subseteq A\cup\{a\}} p(P,B)\cdot \\ & \begin{cases} \sum_{B\cup C=A} p(P/(a,B)\backslash(a,q),C)\cdot p(P/(b,B),s',D) & \text{Si } b\in B \\ \sum_{B\cup C=A} p(P/(a,C),\langle(b,C)\rangle s',D) & \text{Si } b\notin B \end{cases} \end{aligned} \tag{6}$$

cuya solución explícita podríamos formalizar por medio de la secuencia de aproximaciones utilizada en el caso particular previo correspondiente a los estados directamente accesibles.

Proposición 5.6 La función de ocultación es monótona y continua.

Demostración:

MONOTONIA: Sean P, P' tales que $P \sqsubseteq P'$. Entonces, para todo $A, B \subseteq \Sigma$, para todo $a \in A$ se tiene

$$\begin{aligned} p(P,A) &\leq p(P',A) \\ p(P/(a,A),B) &\leq p(P'/(a,A),B) \\ p(P,s) &\leq p(P',s) \\ p(P/(a,A),s) &\leq p(P'/(a,A),s) \end{aligned}$$

Tenemos entonces, que para cada n, la aproximación n-sima a la ocultación es monótona, lo cual probamos por inducción sobre n. En efecto:

Para $n = 0$ el resultado es trivial.

Supuesto que para valores menores de n, las aproximaciones a la ocultación son monótonas, tenemos que para $i < n$

$$\begin{aligned} p(P\backslash_i(a,q),A) &\leq p(P'\backslash_i(a,q),A) \\ p(P/(a,A)\backslash_i(a,q),B) &\leq p(P'/(a,A)\backslash_i(a,q),B) \\ p(P\backslash_i(a,q),s) &\leq p(P'\backslash_i(a,q),s) \\ p(P/(a,A)\backslash_i(a,q),s) &\leq p(P'/(a,A)\backslash_i(a,q),s) \end{aligned}$$

por lo que

$$
\begin{aligned}
& p(P\backslash_n(a,q), \langle b, A\rangle s') \\
&= p(P/(b,A)\backslash_{n-1}, s') \; + \\
&\quad q \cdot \Big(\sum_{a \in B \subseteq \Sigma} p(P\backslash_{n-1}, B) \cdot p(P/(a,B)\backslash_{n-1}, \langle b, A\rangle s') \Big) \; + \\
&\quad (1-q) \cdot \Big(\sum_{a \in B \subseteq A \cup \{a\}} p(P\backslash_{n-1}(a,q)) \cdot \\
&\qquad \begin{cases} \sum_{B \cup C = A} p(P/(a,B)\backslash_{n-1}, C) \cdot p(P/(b,B)\backslash_{n-1}, s') & \text{Si } b \in B \\ \sum_{B \cup C = A} p(P/(a,C)\backslash_{n-1}, \langle (b,C)\rangle s') & \text{Si } b \notin B \end{cases} \\
&\leq p(P'/(b,A)\backslash_{n-1}, s') \; + \\
&\quad q \cdot \Big(\sum_{a \in B \subseteq \Sigma} p(P'\backslash_{n-1}, B) \cdot p(P'/(a,B)\backslash_{n-1}, \langle b, A\rangle s') \Big) \; + \\
&\quad (1-q) \cdot \Big(\sum_{a \in B \subseteq A \cup \{a\}} p(P\backslash_{n-1}, B) \cdot \\
&\qquad \begin{cases} \sum_{B \cup C = A} p(P'/(a,B)\backslash_{n-1}, C) \cdot p(P'/(b,B)\backslash_{n-1}, s') & \text{Si } b \in B \\ \sum_{B \cup C = A} p(P'/(a,C)\backslash_{n-1}, \langle (b,C)\rangle s') & \text{Si } b \notin B \end{cases} \\
&= p(P'\backslash_n(a,q), \langle b, A\rangle s')
\end{aligned}
$$

En consecuencia, puesto que para todo n se cumple

$$p(P\backslash_n(a,q), s) \leq p(P'\backslash_n(a,q), s)$$

tenemos que

$$\lim_n p(P\backslash_n(a,q), s) \leq \lim_n p(P'\backslash_n(a,q), s)$$

por lo que

$$p(P\backslash(a,q), s) \leq p(P'\backslash(a,q), s)$$

y por lo tanto

$$P\backslash(a,q) \sqsubseteq P'\backslash(a,q)$$

CONTINUIDAD: En cuanto a la continuidad, sea $\{P_i\}_{i \in I}$ una cadena de procesos. Entonces por monotonía $\{P_i\backslash(a,q)\}_{i \in I}$ es una cadena, y en el límite, para un n

cualquiera, razonando de modo análogo a como lo hemos hecho en el punto anterior, tenemos

$$
\begin{aligned}
& p((\sqcup P_i)\backslash_n(a,q), \langle b, A\rangle s') \\
&= \lim_i p(P_i\backslash_n(a,q), \langle b, A\rangle s') \\
&= \lim_i [p(P_i/(b,A)\backslash_{n-1}, s') \; + \\
&\quad q \cdot \Big(\sum_{a\in B\subseteq\Sigma} p(P_i\backslash_{n-1}, B)\cdot p(P/(a,B)\backslash_{n-1}, \langle b, A\rangle s')\Big) \; + \\
&\quad (1-q)\cdot\Big(\sum_{a\in B\subseteq A\cup\{a\}} p(P_i\backslash_{n-1}(a,q), B)\cdot \\
&\qquad \begin{cases} \sum_{B\cup C=A} p(P/(a,B)\backslash_{n-1}, C)\cdot p(P/(b,B)\backslash_{n-1}, s') & \text{Si } b\in B \\ \sum_{B\cup C=A} p(P/(a,C)\backslash_{n-1}, (\langle b, C\rangle) s') & \text{Si } b\notin B \end{cases} \\
&= \lim_i p(P_i/(b,A)\backslash_{n-1}, s') \; + \\
&\quad q \cdot \Big(\sum_{a\in B\subseteq\Sigma} \lim_i p(P_i\backslash_{n-1}, B)\cdot \lim_i p(P/(a,B)\backslash_{n-1}, \langle b, A\rangle s')\Big) \; + \\
&\quad (1-q)\cdot\Big(\sum_{a\in B\subseteq A\cup\{a\}} \lim_i p(P_i\backslash_{n-1}(a,q), B)\cdot \\
&\qquad \begin{cases} \sum_{B\cup C=A} \lim_i p(P/(a,B)\backslash_{n-1}, C)\cdot \lim_i p(P/(b,B)\backslash_{n-1}, s') & \text{Si } b\in B \\ \sum_{B\cup C=A} \lim_i p(P/(a,C)\backslash_{n-1}, \langle (b,C)\rangle s') & \text{Si } b\notin B \end{cases} \\
&= p(\sqcup(P_i\backslash_n(a,q)), \langle b, A\rangle s')
\end{aligned}
$$

Con lo cual hemos probado que para todo n tenemos

$$(\sqcup P_i)\backslash_n(a,q) \;=\; \sqcup(P_i\backslash_n(a,q))$$

Para efectuar el paso al límite, observamos el siguiente diagrama

$$
\begin{array}{ccccccccc}
P_1\backslash_1(a,q) & \longrightarrow & \ldots & \longrightarrow & P_1\backslash_n(a,q) & \longrightarrow & \ldots & \longrightarrow & P_1\backslash(a,q) \\
\downarrow & & & & \downarrow & & & & \downarrow \\
P_i\backslash_1(a,q) & \longrightarrow & \ldots & \longrightarrow & P_i\backslash_n(a,q) & \longrightarrow & \ldots & \longrightarrow & P_i\backslash(a,q) \\
\downarrow & & & & \downarrow & & & & \downarrow \\
\sqcup P_i\backslash_1(a,q) & \longrightarrow & \ldots & \longrightarrow & \sqcup P_i\backslash_n(a,q) & \longrightarrow & \ldots & \longrightarrow & \sqcup P_i\backslash(a,q)
\end{array}
$$

Hemos probado que los elementos de la fila inferior son los límites de sus respectivas columnas, y por otra parte sabemos, por definición del operador ocultación como límite de las aproximaciones, que los elementos de la columna derecha son los límites de sus respectivas filas. Por tanto, resulta entonces inmediato, utilizando el hecho de que ambos límites iterados son iguales, por serlo de hecho al correspondiente límite doble, que los dos valores en la esquina inferior derecha del diagrama son iguales, por lo que podemos concluir

$$(\sqcup P_i)\backslash(a,q) \;=\; \sqcup(P_i\backslash(a,q))$$

□

5.2.7 Recursión

Como es usual en el marco de las definiciones denotacionales de la semántica, el significado de los procesos definidos recursivamente mediante ecuaciones como $\mu X.P(X)$, se obtiene por medio del límite de la secuencia

$$\begin{aligned} P_0 &= DIV \\ P_1 &= P(DIV) \\ \ldots &\quad \ldots \\ P_i &= P^n(DIV) \end{aligned}$$

que por ser todos los operadores incluídos en el término $P(X)$ contínuos es, como bien es sabido, el menor punto fijo de la ecuación definidora $X = P(X)$. En definitiva tomamos

$$[\![\mu X.P(X)]\!] = \sqcup [\![P_n]\!]$$

5.3 Equivalencia entre las Semánticas

Hemos definido hasta el momento, la semántica de los procesos probabilísticos, utilizando dos metodologías diferentes. Tenemos, por un lado el marco denotacional, en el que describimos los procesos por medio de árboles de decisiones; y por otro, la semántica de pruebas, en la que un proceso queda caracterizado por

sus respuestas (o más exactamente las probabilidades con las que los pasan) a los distintos tests que le son ofrecidos.

Sin embargo, y como sucede en el caso ordinario sin probabilidades, estas dos formas de definir la semántica no son tan distintas como parece a simple vista. De hecho, la información implícita que proporcionan sobre los procesos es la misma, de forma que dos procesos equivalentes en sus respuestas a los tests serán representados por el mismo árbol, y viceversa: si dos procesos son representados por el mismo árbol, ofrecerán las mismas respuestas a todo test que les sea ofrecido para su aceptación. En consecuencia, si denotamos por $\equiv_T$ a la equivalencia inducida por la semántica de pruebas, y por $\equiv_D$ a la equivalencia denotacional, se tiene

$$\forall P, P' \in PCSP : P \equiv_T P' \Leftrightarrow P \equiv_D P'$$

Para demostrar esta equivalencia, comenzaremos por probar que el uso que hemos hecho de una misma notación, a saber $p(P, A)$ y $P/(a, A)$, para referirnos en el seno de cada uno de los dos marcos semánticos, a dos pares de conceptos en principio diferentes, está justificado, por resultar a la postre equivalentes, lo que justifica la "*confusión*" notacional.

En primer lugar, en el marco de la semántica de pruebas la notación $p(P, A)$ representa el valor de una cierta combinación de los resultados de aplicar una serie de tests al proceso P, mientras que en la semántica denotacional, con la misma notación nos referimos a la etiqueta de una determinada rama del árbol, que se obtiene al evaluar en el álgebra de árboles probabilísticos $\mathcal{P}$, el términos sintáctico P. Evidentemente, un primer paso de cara a comparar dichos conceptos es reconducir ambos a lugares comunes. A tal efecto, comenzamos definiendo en primer lugar el paso de tests por parte de los árboles probabilísticos.

Definición 5.12 Siendo $T = \prod_{A \subseteq \Sigma} [p_A] \Box_{a \in A} a.T_{a,A}$ un árbol probabilístico, definimos $T \mid t$ en la forma siguiente:

1. $T \mid \omega = 1$

2. $T \mid \surd = p_\emptyset$

3. Para cualquier otro test se tiene

$$T \mid t = \sum_{A \cap Fi(t) \neq \emptyset} \sum_{a \in A \cap Fi(t)} s(a, A, t) \cdot p_A \cdot (T_{a,A} \mid t/a)$$

□

Aplicando recursivamente dicha definición, obtenemos que el paso de un test por parte de un proceso semántico se puede obtener sumando las aportaciones de las diversas ramas del test que terminan en ω o en $\surd$. Cada rama, formada por las acciones $a_1, \ldots, a_n$ contribuirá en cada momento con un valor, que depende de los conjuntos entre los que el proceso semántico va seleccionando las acciones a ejecutar a lo largo de sus cómputos. Además, cada una de dichas aportaciones es resultado a su vez de la suma de lo que aportan al efecto las ramas correspondientes a cada una de las acciones del conjunto en cuestión. Así, si a cada subárbol $T_{a,A}$ lo denotamos por $T/(a, A)$, y a los valores p_A que etiquetan las ramas que parten de la raíz del árbol T por $p(T, A)$, tenemos que para cada secuencia de estados $A_1, \ldots, A_n$, con $a_i \in A_i$, aparece un sumando cuyo valor es igual a

$$s(a_1, A_1, t) \cdot s(a_2, A_2, t/a_1) \cdots s(a_n, A_n, t/a_1/\ldots/a_{n-1}) \cdot p(T, A_1) \cdot$$

$$p(T/(a_1, A_1), A_2) \cdots p(T/(a_1, A_1)/\cdots/(a_{n-1}, A_{n-1}), A_n)$$

o lo que es igual, aplicando la def. 5.4,

$$s(a_1, A_1, t) \cdot s(a_2, A_2, t/a_1) \cdots s(a_n, A_n, t/a_1/\ldots/a_{n-1}) \cdot$$

$$p(T, (a_1, A_1) \ldots (a_{n-1}, A_{n-1}), A_n)$$

En definitiva, podemos obtener la probabilidad de paso de un test t por parte de un proceso semántico, sumando los resultados de sendos sumatorios, que corresponden a los conjuntos S_1 de las secuencias $r = a_1 A_1 \ldots a_{n-1} A_{n-1} a_n A_n$ con $a_i \in A_i$ tales que la secuencia de acciones asociada $a_1, \ldots, a_{n-1}, a_n$ nos conduce en el test t a ω; y S_2 formado por las secuencias análogas, pero que finalizan en el conjunto $A_n = \emptyset$ para las que la secuencia asociada de acciones nos conduce en el test t a $\surd$. Es decir

$$T \mid t = \sum_{r \in S_1} s(a_1^r, A_1^r, t) \cdots s(a_n^r, A_n^r, t/a_1^r/\ldots/a_{n-1}^r) \cdot p(T, r', A_n) + \sum_{r \in S_2} s(a_1^r, A_1^r, t) \cdots s(a_n^r, A_n^r, t/a_1^r/\ldots/a_{n-1}^r) \cdot p(T, r', \emptyset) \quad (7)$$

donde r' denota la secuencia $a_1 A_1 \ldots a_{n-1} A_{n-1}$

A partir de este resultado podemos probar para la semántica denotacional, un resultado análogo al enunciado en la prop. 4.1 para la semántica de pruebas, que es el siguiente:

Proposición 5.7 Para todo estado $A \subseteq \Sigma$ y toda acción $a \in A$, la operación $/(a, A)$, definida sobre los árboles probabilísticos satisface las propiedades:

1. $[\![STOP]\!]/(a, A) \mid t = [\![STOP]\!] \mid t$
2. $[\![(a \to P)]\!]/(a, \{a\}) \mid t = [\![P]\!] \mid t$
3. $p([\![[p]P \sqcap [1-p]Q]\!], A) \cdot ([\![[p]P \sqcap [1-p]Q]\!]/(a, A) \mid t) =$
   $p \cdot p([\![P]\!], A) \cdot ([\![P]\!]/(a, A) \mid t) + (1-p) \cdot p([\![Q]\!], A) \cdot ([\![Q]\!]/(a, A) \mid t)$
4. $p([\![[p]P \Box [1-p]Q]\!], A) \cdot [\![[p]P \Box [1-p]Q]\!]/(a, A) \mid t =$
   $\sum_{A=B \cup C} p([\![P]\!], B) \cdot p([\![Q]\!], C) \cdot Cont([\![[p]P \Box [1-p]Q]\!], t, B, C, a)$
5. $p([\![[p]P \|_D [1-p]Q]\!], A) \cdot [\![[p]P \|_D [1-p]Q]\!]/(a, A) \mid t =$
   $\sum_{A=B \cup C} p([\![P]\!], B) \cdot p([\![Q]\!], C) \cdot Cont([\![[p]P \|_D [1-p]Q]\!], t, B, C, a) -$
   $\sum_{\substack{B \cap C = \emptyset \\ B, C \subseteq D}} p([\![P]\!], B) \cdot p([\![Q]\!], C)) \cdot Cont([\![[p]P \|_D [1-p]Q]\!], t, B, C, a)$
6. En cuanto al operador de ocultación se tiene

$$p([\![P \backslash (a,q)]\!], A) \cdot [\![P \backslash (a,q)]\!]/(b, A) \mid t =$$
$$p([\![P]\!], A) \cdot [\![P/(b, A) \backslash (a,q)]\!] \mid t +$$
$$q \cdot (\sum_{a \in B \subseteq \Sigma} p([\![P]\!], B) \cdot p([\![P/(a, B) \backslash (a,q)]\!], A) \cdot [\![P/(a, B) \backslash (a,q)/(b, A)]\!] \mid t +$$
$$(1-q) \cdot (\sum_{a \in B \subseteq A \cup \{a\}} p([\![P]\!], B) \cdot \sum_{B \cup C = A} p([\![P/(a, B) \backslash (a,q)]\!], C) \cdot$$
$$\begin{cases} [\![P/(b, B) \backslash (a,q)]\!] \mid t & \text{Si } b \in B \\ [\![P/(a, B) \backslash (a,q)/(b, C)]\!] \mid t & \text{Si } b \notin B \end{cases})$$

Demostración: La demostración se basa simplemente en el uso de las distintas propiedades de "*descomposición estructural* de los valores $p(T, s, A)$, que se obtienen a partir de la definición de los mismos, distinguiendo casos según los distintos operadores en cabeza de T (ecuaciones 1, 2, 3, 4, 5, 6). Dichas ecuaciones expresan propiedades análogas a las que ahora queremos demostrar, pero referidas a estados alcanzables en lugar de al paso de tests por los árboles. A partir de dichas propiedades, y de la ecuación 7, que expresa el paso de tests en función de las probabilidades de alcanzar estados por parte del árbol en cuestión, se siguen inmediatamente las propiedades deseadas. □

Ahora pues, dado un árbol probabilístico R, podemos considerar los valores de paso de tests para el mismo, los cuales a su vez definen un proceso, que denotaremos por $\Upsilon(R)$, en el marco de la semántica de pruebas. Entonces, podemos computar los valores $p(\Upsilon(R), A)$ correspondientes a dicho proceso y los restos del proceso $\Upsilon(R)/(a, A)$. Pero a su vez, partiendo de estos valores, podemos obtener un nuevo árbol semántico, $\tau^{\Upsilon(R)}$, aplicándo la siguiente construcción general que nos permite pasar de un proceso semántico correspondiente a la semántica de pruebas, al correspondiente árbol probabilístico.

Definición 5.13 Sea R un proceso semántico correspondiente a la semántica de pruebas, entonces definimos el árbol τ^R en la forma siguiente:

$$\tau^R = \prod_{A \subseteq \Sigma} [p(R, A)] \, \Box_{a \in A} \, \tau^{R/(a,A)}$$

□

A partir de esta definición obtenemos la siguiente

Proposición 5.8 Para todo árbol probabilístico $R \in \mathcal{P}$ se tiene que el árbol probabilístico $\tau^{\Upsilon(R)}$ obtenido por medio de la composición de las definiciones 5.13 y 5.12, coincide con el árbol original R.

Demostración: Para el primer nivel del árbol, tenemos

$$\begin{aligned} p(\Upsilon(R), \emptyset) &= 1 - R \mid t_\Sigma \\ p(\Upsilon(R), A) &= 1 - (R \mid t_{\Sigma - A}) - \sum_{B \subseteq A} p(\Upsilon(R), B) \end{aligned}$$

con

$$\begin{aligned} R \mid t_{\Sigma} &= \sum_{A} \sum_{a \in A} \frac{1}{|A|} \cdot p(R, A) &&= \sum_{A} p(R, A) \\ R \mid t_{\Sigma - A} &= \sum_{B - A \neq \emptyset} \sum_{a \in B - A} \frac{1}{|B - A|} \cdot p(R, B) &&= \sum_{B - A \neq \emptyset} p(R, B) \end{aligned}$$

por lo que concluimos que $p(\Upsilon(R), \emptyset) = p(R, \emptyset)$ y que $p(\Upsilon(R), A) = P(R, A)$.

En lo que respecta a las continuaciones, tenemos que las mismas están definidas en la forma

$$\Upsilon(R/(a, A)) \mid t = \frac{|A| \cdot (R \mid t^{(a, A - \{a\})} - \sum_{C \subset A - \{a\}} (-1)^{|A - C|} \cdot R \mid t^{(a, C)})}{p(R, A)}$$

Aplicando la def. 5.12, y teniendo en cuenta la forma de los tests en litigio, tenemos

$$\begin{aligned} R \mid t^{(a, A - \{a\})} &= \sum_{B \subseteq A} \frac{1}{|B|} \cdot p(R, B) \cdot (R/(a, B) \mid t) \\ R \mid t^{(a, C)} &= \sum_{B \subseteq C \cup \{a\}} \frac{1}{|B|} \cdot p(R, B) \cdot (R/(a, B) \mid t) \end{aligned}$$

Operando en la expresión anterior para $\Upsilon(R)/(a, A)$, obtenemos inmediatamente que su valor coincide con el paso de tests por parte del proceso $R/(a, A)$. □

Lema 5.1 Sea P un proceso sintáctico finito, entonces tenemos que para todo test t, para todo estado $A \subseteq \Sigma$, y para toda acción $a \in A$ se tiene que

$$\begin{aligned} p(P, A) &= p([\![P]\!], A) \\ p(P, A) \cdot P/(a, A) \mid t &= p([\![P]\!], A) \cdot [\![P]\!]/(a, A) \mid t \end{aligned}$$

Demostración: Razonaremos por inducción estructural, respecto de la sintaxis de los procesos, utilizando las proposiciones 4.1, 5.7 y 5.8.

Para los procesos $STOP$, DIV y prefijo, el resultado es trivial, pues el primero sólo alcanza el estado vacio, mientras que el segundo no tiene estados en ninguna de ambas semánticas. Igualmente el único estado alcanzado por un proceso de la forma $a \rightarrow P$, es el mismo en las dos semánticas. En lo que hace referencia a las continuaciones, las mismas sólo aparecen en el caso del operador prefijo, y al ser la obtenida un proceso sintáctico, podemos limitarnos a aplicar la hipótesis de inducción estructural.

Para la elección interna, tenemos en primer lugar que la probabilidad de alcanzar un estado verifica

$$\begin{aligned} p([p]P \sqcap [1-p]Q, A) &= p \cdot p(P, A) + (1-p) \cdot p(Q, A) \\ &= p \cdot p([\![P]\!], A) + (1-p) \cdot p([\![Q]\!], A) \\ &= p([\![[p]P \sqcap [1-p]Q]\!], A) \end{aligned}$$

mientras que para sus continuaciones, aplicando las proposiciones 4.1 y 5.7, tenemos

$$\begin{aligned} & p([p]P \sqcap [1-p]Q, A) \cdot ([p]P \sqcap [1-p]Q/(a, A) \mid t \\ &= p \cdot p(P, A) \cdot (P/(a, A) \mid t) + (1-p) \cdot p(Q, A) \cdot (Q/(a, A) \mid t) \\ &= p \cdot p([\![P]\!], A) \cdot ([\![P]\!]/(a, A) \mid t) + (1-p) \cdot p([\![Q]\!], A) \cdot ([\![Q]\!]/(a, A) \mid t) \\ &= p([\![[p]P \sqcap [1-p]Q]\!], A) \cdot ([\![[p]P \sqcap [1-p]Q]\!]/(a, A) \mid t \end{aligned}$$

Respecto a la elección externa, para las probabilidades de alcanzar un estado tenemos

$$\begin{aligned} p([p]P \Box [1-p]Q, A) &= \sum_{A=B \cup C} p(P, B) \cdot p(Q, C) \\ &= \sum_{A=B \cup C} p([\![P]\!], B) \cdot p([\![Q]\!], C) \\ &= p([\![[p]P \Box [1-p]Q]\!], A) \end{aligned}$$

y para las continuaciones, también en virtud de las proposiciones 4.1 y 5.7

$$\begin{aligned} & p([p]P \Box [1-p]Q, A) \cdot [p]P \Box [1-p]Q/(a, A) \mid t \\ &= \sum_{A=B \cup C} p(P, B) \cdot p(Q, C) \cdot Cont([p]P \Box [1-p]Q, t, B, C, a) \\ &= \sum_{A=B \cup C} p([\![P]\!], B) \cdot p([\![Q]\!], C) \cdot Cont([\![[p]P \Box [1-p]Q]\!], t, B, C, a) \\ &= p([\![[p]P \Box [1-p]Q]\!], A) \cdot [\![[p]P \Box [1-p]Q]\!]/(a, A) \mid t \end{aligned}$$

donde la igualdad entre ambas aplicaciones de la función $Cont$ viene justificada por el hecho de que sus valores son, en los mismos casos, bien los correspondientes a $P/(a,A)$, a $Q/(a,A)$, o a la elección interna entre ambos.

En cuanto al operador paralelo, tenemos en primer lugar

$$\begin{aligned} p([p]P\|_A[1-p]Q,D) &= \sum_{B,C} p(P,B)\cdot p(Q,C) \\ &= \sum_{B,C} p([\![P]\!],B)\cdot p([\![Q]\!],C) \\ &= p([\![[p]P\|_A[1-p]Q]\!],D) \end{aligned}$$

siendo B y C tales que $D = (B \cup C) - A \cup (A \cap B \cap C)$.

Mientras que, por la misma razón anterior, las continuaciones verifican

$$\begin{aligned} & p([p]P\|_A[1-p]Q,D)\cdot [p]P\|_A[1-p]Q/(a,D) \mid t \\ &= \sum_{B,C} p(P,B)\cdot p(Q,C)\cdot Cont([p]P\|_D[1-p]Q,t,B,C,a) \\ &= \sum_{B,C} p([\![P]\!],B)\cdot p([\![Q]\!],C)\cdot Cont([\![[p]P\|_D[1-p]Q]\!],t,B,C,a) \\ &= p([\![[p]P\|_D[1-p]Q]\!],A)\cdot [\![[p]P\|_D[1-p]Q]\!]/(a,A) \mid t \end{aligned}$$

siendo B y C tales que $D = (B \cup C) - A \cup (A \cap B \cap C)$.

Finalmente, en el caso de la ocultación, además de la inducción estructural, nos vemos obligados, como ya viene siendo usual, a razonar además por inducción respecto de las aproximaciones por medio de las cuales definimos la semántica de la ocultación. Así, comenzando por las probabilidades de alcanzar un estado, en lo que hace referencia a la aproximación 1, tenemos

$$p(P\backslash_1(a,q),A) = p(P,A) = p([\![P]\!],A) = p([\![P\backslash_1(a,q)]\!],A)$$

En lo que se refiere a la aproximación de nivel n, como podemos reducir la aplicación de la ocultación de nivel n a aplicaciones de ocultaciones de nivel $n-1$, aplicando la hipótesis de inducción respecto del nivel de la aproximación, y la

hipótesis de inducción estructural, concluímos

$$
\begin{aligned}
& p(P\backslash_n(a,q),A) \\
&= p(P,A) + q\cdot(\sum_{a\in B\subseteq\Sigma} p(P,B)\cdot p(P/(a,B)\backslash_{n-1}(a,q),A)) + \\
&\quad (1-q)\cdot(\sum_{a\in B\subseteq A\cup\{a\}} p(P,B)\cdot \sum_{B\cup C=A} p(P/(a,B)\backslash_{n-1}(a,q),C) \\
&= p([\![P]\!],A) + q\cdot(\sum_{a\in B\subseteq\Sigma} p([\![P]\!],B)\cdot p([\![P/(a,B)\backslash_{n-1}(a,q)]\!],A)) + \\
&\quad (1-q)\cdot(\sum_{a\in B\subseteq A\cup\{a\}} p([\![P]\!],B)\cdot \sum_{B\cup C=A} p([\![P/(a,B)\backslash_{n-1}(a,q)]\!],C) \\
&= p([\![P\backslash_n(a,q)]\!],A)
\end{aligned}
$$

En cuanto a las continuaciones, el razonamiento es análogo, de manera que por medio de una doble inducción, estructural y respecto del nivel de las aproximaciones, y de la aplicación de las proposiciones 4.1 y 5.7, obtenemos

$$
\begin{aligned}
& p(P\backslash(a,q),A)\cdot P\backslash(a,q)/(b,A) \mid t \\
&= p(P,A)\cdot P/(b,A)\backslash(a,q) \mid t + \\
&\quad q\cdot(\sum_{a\in B\subseteq\Sigma} p(P,B)\cdot p(P/(a,B)\backslash(a,q),A)\cdot P/(a,B)\backslash(a,q)/(b,A) \mid t) + \\
&\quad (1-q)\cdot(\sum_{a\in B\subseteq A\cup\{a\}} p(P,B)\cdot \sum_{B\cup C=A} p(P/(a,B)\backslash(a,q),C)\cdot \\
&\qquad \begin{cases} P/(b,B)\backslash(a,q) \mid t & \text{Si } b\in B \\ P/(a,B)\backslash(a,q)/(b,C) \mid t & \text{Si } b\notin B \end{cases}) \\
&= p([\![P]\!],A)\cdot [\![P/(b,A)\backslash(a,q)]\!] \mid t + \\
&\quad q\cdot(\sum_{a\in B\subseteq\Sigma} p([\![P]\!],B)\cdot p([\![P/(a,B)\backslash(a,q)]\!],A)\cdot [\![P/(a,B)\backslash(a,q)/(b,A)]\!] \mid t) + \\
&\quad (1-q)\cdot(\sum_{a\in B\subseteq A\cup\{a\}} p([\![P]\!],B)\cdot \sum_{B\cup C=A} p([\![P/(a,B)\backslash(a,q)]\!],C)\cdot \\
&\qquad \begin{cases} [\![P/(b,B)\backslash(a,q)]\!] \mid t & \text{Si } b\in B \\ [\![P/(a,B)\backslash(a,q)/(b,C)]\!] \mid t & \text{Si } b\notin B \end{cases}) \\
&= p([\![P\backslash(a,q)]\!],A)\cdot [\![P\backslash(a,q)/(b,A)]\!] \mid t
\end{aligned}
$$

donde para abreviar la exposición nos hemos permitido la utilización de la caracterización recursiva de la definición de la semántica del operador de ocultación.

□

Como corolario inmediato de este lema tenemos el

Corolario 5.1 Para todo proceso sintáctico P y para todo test t se tiene

$$P \mid t = [\![P]\!] \mid t$$

Demostración: Si P es finito el resultado es inmediato a partir del lema 5.1, para el caso de que sea recursivo, tenemos

$$\begin{aligned} P \mid t &= \lim_n P_n \mid t \\ [\![P]\!] \mid t &= \lim_n [\![P'_n]\!] \mid t \end{aligned}$$

Donde los procesos P_n son las aproximaciones sintácticas finitas a P. Entonces, por ser éstas finitas, sabemos que

$$P_n \mid t = [\![P_n]\!] \mid t$$

de donde concluímos

$$P \mid t = \lim_n P_n \mid t = \lim_n [\![P_n]\!] \mid t = [\![P]\!] \mid t$$

obteniendose por tanto que el resultado es en efecto extensible a procesos definidos recursivamente. □

Lema 5.2 Sea P un proceso sintáctico finito, entonces para todo test t se satisface

$$[\![P]\!]/(a,A) \mid t = \frac{|A| \cdot ([\![P]\!] \mid t^{(a,A-\{a\})} - \sum_{C \subset A-\{a\}} (-1)^{|A-C|} \cdot [\![P]\!] \mid t^{(a,C)})}{p([\![P]\!], A)}$$

Demostración: Por un lado para $a \notin C$ se tiene

$$[\![P]\!] \mid t^{(a,C)} = \sum_{B \subset C} \frac{1}{|B|} \cdot p([\![P]\!], B \cup \{a\}) \cdot [\![P]\!]/(a, B \cup \{a\}) \mid t/a$$

Además

$$[\![P]\!] \mid t^{(a,A-\{a\})} = \sum_{a \in B \subset A} \frac{1}{|B|} \cdot p([\![P]\!], B \cup \{a\}) \cdot [\![P]\!]/(a, B \cup \{a\}) \mid t/a$$

por lo que

$$[\![P]\!] \mid t^{(a,A-\{a\})} - \sum_{C \subset A-\{a\}} (-1)^{|A-C|} [\![P]\!] \mid t^{(a,C)} =$$

$$\frac{1}{|A|} \cdot p([\![P]\!], A) \cdot [\![P]\!]/(a, A) \mid t/a$$

de donde se obtiene el resultado deseado. □

Corolario 5.2 Los resultados obtenidos en el lema 5.1 y el corolario 5.1 son también válidos para las continuaciones arbitrariamente profundas de la forma $P/(a_1, A_1)/\cdots/(a_n, A_n)$.

Demostración: A partir del lema 5.2 podemos expresar el paso de un test por parte de cada continuación de un proceso semántico, en función de los pasos de tests por parte del proceso dado. Por otra parte, sabemos que para procesos sintácticos, ello es trivialmente cierto, por definición. Finalmente, como el paso de tests por parte de los procesos sintácticos es igual al paso de test por parte de sus semánticas denotacionales, como indica el corolario 5.1, concluimos que para los procesos resultantes de la aplicación de una operación Resto, se satisface en efecto el resultado deseado. Por último, podemos generalizar el mismo a aplicaciones reiteradas del operador, por medio de una simple inducción sobre el número de veces que aplicamos el mismo. □

Corolario 5.3 El resultado enunciado en la proposición 4.2 para procesos sintácticos, es cierto también para procesos de la forma $P/(a_1, A_1)/\cdots/(a_n, A_n)$.

Demostración:

$$\begin{aligned}
P/(a_1, A_1)/\cdots/(a_n, A_n) \mid t &= [\![P]\!]/(a_1, A_1)/\cdots/(a_n, A_n) \mid t \\
&= \sum_{A \subseteq \Sigma} \sum_{a \in A} s(a, A, t) \cdot p([\![P]\!]/(a_1, A_1)/\cdots/(a_n, A_n), A) \cdot \\
&\qquad ([\![P]\!]/(a_1, A_1)/\cdots/(a_n, A_n)/(a, A) \mid t/a \\
&= \sum_{A \subseteq \Sigma} \sum_{a \in A} s(a, A, t) \cdot p(P/(a_1, A_1)/\cdots/(a_n, A_n), A) \cdot \\
&\qquad (P/(a_1, A_1)/\cdots/(a_n, A_n)/(a, A) \mid t/a
\end{aligned}$$

□

Lema 5.3 Siendo P un proceso sintáctico finito, consideremos su semántica de pruebas, y denotemos por R_P^{Υ} el árbol probabilístico obtenido a partir de la misma por medio de la definición 5.13. Entonces para todo test t se tiene

$$P \mid t = R_P^{\Upsilon} \mid t$$

y la misma propiedad es cierta para las continuaciones del proceso P de la forma $P/(a_1, A_1)/\cdots/(a_n, A_n)$.

Demostración: Partimos de las expansiones de ambos lados de la igualdad que nos suministran la proposición 4.2 y la definición 5.12. Por otra parte, tenemos que, en virtud de la definición del árbol R_P^{Υ} se tiene

$$\begin{aligned} p(R_P^{\Upsilon}, A) &= p(P, A) \\ (R_P^{\Upsilon})/(a, A) &= R_{P/(a,A)}^{\Upsilon} \end{aligned}$$

Bastaría entonces para concluir el resultado deseado, que $P/(a, A) \mid t/a$ fuese igual a $R_{P/(a,A)}^{\Upsilon} \mid t/a$. Pero vemos que se trata del mismo resultado que deseamos, si bien aplicado al proceso $P/(a, A)$ y al test t/a, en lugar de al proceso P y al test t de partida. Ahora bien, el corolario 5.3 nos indica que el resultado de la prop. 4.2, que hemos aplicado como punto de partida de nuestro razonamiento, es también válido para procesos de la forma $P/(a, A)$, mientras que por su parte la def. 5.12 no precisa de generalización alguna, por ser la misma válida para árboles cualesquiera.

Podemos, por tanto, reordenar nuestro razonamiento, planteandonos la demostración del resultado deseado pero para procesos sintácticos extendidos, que tengan, además de las operaciones de PCSP, en cabeza de la expresión una secuencia de operaciones tras, es decir procesos de la forma $P/(a_1, A_n)/\cdots/(a_n, A_n)$, con $n \geq 0$. Ello se haría por inducción respecto de la profundidad del test que se aplique, no presentando la demostración en cuestión dificultad especial alguna. □

Y llegamos por fin a los siguientes resultados, de los que se sigue la equivalencia entre la semántica denotacional y la de pruebas.

Teorema 5.2 Para todo par de procesos sintácticos, la equivalencia respecto de la semántica de pruebas implica la equivalencia respecto de la semántica denotacional,

es decir

$$\forall P, P' \in PCSP : P \equiv_T P' \Rightarrow P \equiv_D P'$$

Demostración: Partimos de que para todo test t se tiene $P \mid t = P' \mid t$. Entonces, por el corolario 5.1 tenemos que

$$P \mid t = [\![P]\!] \mid t$$

$$P' \mid t = [\![P']\!] \mid t$$

por lo que

$$\Upsilon([\![P]\!]) = \Upsilon([\![P']\!])$$

de donde obtenemos que los árboles inducidos $\tau^{\Upsilon([\![P]\!])}$ y $\tau^{\Upsilon([\![P']\!])}$ son iguales, y por tanto, en virtud de la prop. 5.8 se concluye $[\![P]\!] = [\![P']\!]$. □

Teorema 5.3 La equivalencia respecto de la semántica denotacional implica la equivalencia respecto de la semántica de pruebas, es decir

$$\forall P, P' \in Proc : P \equiv_D P' \Rightarrow P \equiv_T P'$$

Demostración: Partimos de la igualdad de los árboles semánticos correspondientes a P y Q. Al ser dichos árboles iguales, tenemos

$$[\![P]\!] \mid t = [\![P']\!] \mid t$$

de donde, aplicando el corolario 5.1, obtenemos

$$P \mid t = [\![P]\!] \mid t = [\![P']\!] \mid t = P' \mid t$$

por lo que $P \equiv_T P'$. □

Un resultado importante, que se sigue inmediatamente de los teoremas y las proposiciones anteriores, es que la relación de orden en la que se basa la definición de la semántica denotacional, implica la relación de orden inducida por la semántica de pruebas.

Corolario 5.4 La relación de orden denotacional implica la relación de orden inducida por la semántica de pruebas. Es decir, Para todo $P, P' \in PCSP$ se tiene

$$[\![P]\!] \sqsubseteq [\![Q]\!] \Rightarrow \forall t \in PTEST\ P \mid t \leq Q \mid t$$

Demostración: Por la prop. 4.2 tenemos

$$P \mid t = \sum_{a \in Ft(t)} \sum_{a \in A \subseteq \Sigma} s(a, A, t) \cdot p(P, A) \cdot P/(a, A) \mid t/a$$

Entonces, puesto que $[\![P]\!] \sqsubseteq [\![Q]\!]$ tenemos que para todo $A \subseteq \Sigma$ y para todo $a \in A$ se cumple

$$\begin{aligned} p([\![P]\!], A) &\leq p([\![Q]\!], A) \\ [\![P]\!]/(a, A) &\sqsubseteq [\![Q]\!]/(a, A) \end{aligned}$$

y en tal caso, por inducción respecto a la profundidad de los tests se concluye inmediatamente

$$\begin{aligned} P \mid t &= \sum_{a \in Ft(t)} \sum_{a \in A \subseteq \Sigma} s(a, A, t) \cdot p(P, A) \cdot P/(a, A) \mid t/a \\ &\leq \sum_{a \in Ft(t)} \sum_{a \in A \subseteq \Sigma} s(a, A, t) \cdot p(Q, A) \cdot Q/(a, A) \mid t/a \\ &= Q \mid t \end{aligned}$$

□

Este resultado nos sirve en particular para concluir la demostración de la corrección de la definición del paso de tests para procesos recursivos, que en el capítulo precedente dejábamos incompleta, retrasando su conclusión hasta que se contase con los resultados auxiliares precisos, que son justamente los que ahora acabamos de probar. Más exactamente, el resultado cuya prueba habíamos dejado pendiente, es el que se presenta en el siguiente

Corolario 5.5 Para todo proceso sintáctico recursivo $\mu X.P(X)$, y para todo test t la sucesión

$$P_0 \mid t, P_1 \mid t, \ldots, P_n \mid t, \ldots$$

es creciente.

Demostración: Se obtiene inmediatamente a partir del corolario anterior, y de que la siguiente cadena es en efecto ascendente:

$$[\![P_0]\!] \sqsubseteq [\![P_1]\!] \sqsubseteq \ldots \sqsubseteq [\![P_n]\!] \sqsubseteq \ldots$$

□

Sin embargo, el recíproco del corolario 5.4 no es cierto, pudiendose encontrar procesos relacionados respecto al paso de tests, que no obstante, no son relacionables denotacionalmente. Ello puede comprobarse por medio del siguiente

Ejemplo 5.1 Consideremos los siguientes procesos

$$\begin{aligned} P &= [\tfrac{1}{2}]a \rightarrow STOP \sqcap [\tfrac{1}{2}]b \rightarrow STOP \\ Q &= [\tfrac{1}{2}]([\tfrac{1}{2}]a \rightarrow STOP \,\Box\, [\tfrac{1}{2}]b \rightarrow STOP) \sqcap [\tfrac{1}{2}]STOP \end{aligned}$$

definidos sobre el alfabeto $\{a, b\}$. Tenemos que para todo test t se cumple

$$Q \mid t \leq P \mid t$$

Ahora bien, en lo que se refiere a la semántica denotacional, si consideramos los estados $\{a\}$ y $\{a, b\}$, tenemos

$$\begin{aligned} p(P, \{a\}) &= \tfrac{1}{2} \quad p(Q, \{a\}) = 0 \\ p(P, \{a, b\}) &= 0 \quad p(Q, \{a, b\}) = \tfrac{1}{2} \end{aligned}$$

de donde se sigue que dichos procesos no están relacionados respecto de la relación de orden correspondiente a dicha semántica. □

Hemos pues definido un modelo denotacional, para el cual hemos comprobado la equivalencia entre la correspondiente semántica denotacional y la semántica de pruebas previamente definida. Ahora bien, como acabamos de comprobar, dicha equivalencia entre las semánticas se tiene sólo a nivel de las equivalencias entre procesos que inducen, no manteniéndose la misma al nivel de las relaciones de orden correspondientes.

Más exactamente, tenemos que aunque la relación de orden denotacional implica el orden respecto a la semántica de pruebas, las dos relaciones de orden no

coinciden, puesto que el recíproco no es cierto en general. Podría entonces interesarnos definir una nueva semántica denotacional, basada en una nueva relación de orden, que resulte ser absolutamente equivalente a la semántica de pruebas. Obsérvese que de cara a definir el nuevo modelo denotacional es razonable mantener el mismo conjunto soporte, pues la equivalencia entre procesos debe seguir siendo la misma. En consecuencia nos limitaremos a buscar una nueva relación de orden sobre el dominio de los árboles probabilísticos, que coincida a la postre con la inducida por los tests.

5.4 Una nueva relación de orden

Como veíamos en el capítulo anterior, los tests de la forma $t^{(a,A)}$, mediante los cuales hemos definido la operación $P/(a,A)$, capturan cada uno de ellos la probabilidad de que un proceso alcance un estado incluído en el conjunto A, y continúe tras ello aceptando el resto del test t.

Como ya hemos indicado, la relación de orden en que hemos basado la definición de la semántica denotacional no coincide con la inducida por los tests, que recordemos viene dada por

$$P \sqsubseteq_T Q \;\Leftrightarrow\; \forall t \in PTEST \;\; P \mid t \leq Q \mid t$$

Ello es debido a que la relación de orden que hemos utilizado está basada en la operación $P/(a,A)$, operación que en el marco de la semántica de pruebas solo es posible definir por medio de una combinación de sumas y restas de resultados de paso de ciertos tests. Entonces, si tenemos que $P \sqsubseteq_T Q$, tendríamos que los sumandos que aparecen con signo positivo en la expresión que define una cierta aplicación de la operación *tras*, sí que son mayores en el caso en que aplicamos dicha operación a P, que cuando lo hacemos a Q; sin embargo, en el caso de los sumandos que aparecen con signo menos, la relación de orden que tendríamos sería justo la contraria de la deseada, por lo que en definitiva nada podemos concluir en general, en lo que se refiere a la ordenación entre P y Q respecto de la relación de orden utilizada para definir la semántica denotacional. Ello quedo probado en particular por medio del contraejemplo presentado en el Ejemplo 5.1.

La detección de la razón última por la cual no se tiene la equivalencia entre las dos relaciones de orden, nos indica el camino para definir en el marco denotacional la deseada relación de orden equivalente a la inducida por los tests. En efecto, de cara a definir la misma deberíamos basarnos en una nueva operación *resto de un proceso* tras la ejecución de una acción, que sea capturable por un único test, o en el peor de los casos, por medio de sumas de pasos de tests, sin que intervenga resta alguna. En concreto, y como quiera que nuestros problemas provienen de que la operación *tras* hasta ahora considerada quedaba definida por medio de una combinación de tests de la forma $t^{(a,A)}$, lo que haremos será definir la nueva operación *resto* por medio de uno sólo de dichos tests.

Definición 5.14 (Nuevo resto de un proceso tras la ejecución de una acción)
Sea P un proceso, $A \subseteq \Sigma$, y $a \in A$; definimos el proceso $P//(a,A)$ como el árbol que resulta de eliminar en P todas aquellas ramas que partan de la raíz que vayan etiquetadas con conjuntos B tales que $B \not\subseteq A$ o bien $a \notin B$, normalizando tras ello el resultado obtenido. □

Definición 5.15 (Nuevo orden entre procesos)
La nueva relación de orden entre procesos, que denotaremos mediante $\sqsubseteq'_D$ se define recursivamente, diciendo que se tiene $P \sqsubseteq'_D Q$ si y sólo si se cumplen

1. $\forall\, A \subseteq \Sigma, a \in A, \quad \sum_{a \in A' \subseteq A} p(P,A') \leq \sum_{a \in A' \subseteq A} p(Q,A')$
2. $\forall\, A \subseteq'_D \Sigma, a \in A, \quad P//(a,A) \sqsubseteq'_D Q//(a,A)$

□

Teorema 5.4 La nueva relación de orden definida en la semántica denotacional es equivalente a la relación de orden inducida por los tests, es decir que se tiene $P \sqsubseteq_T Q \;\Leftrightarrow\; P \sqsubseteq'_D Q$.

Demostración: Para la implicación $\Rightarrow$ tenemos que probar

$$\forall t \in Tree \quad P \mid t \leq Q \mid t \;\Rightarrow\; P \sqsubseteq'_D Q$$

Respecto a la primera afirmación tenemos

$$P \sqsubseteq_T Q \;\Rightarrow\; \forall t \in Tree \quad P \mid t \leq Q \mid t$$

Por otra parte, dado un test cualquiera t, de la definición de la operación $t^{(a,A)}$ y de la caracterización de $P \mid t$ se obtiene que

$$P \mid t^{(a,A)} = \sum_{A' \subseteq A} p(P, A') \cdot \frac{1}{|A'|} \cdot (P/(a, A') \mid t)$$

Pero considerando la nueva operación definida en esta sección, llegamos a

$$P \mid t^{(a,A)} = P//(a, A) \mid t$$

y puesto que

$$P \mid t_A = \sum_{a \in A' \subseteq A} p(P, A')$$

tenemos que, si para cualquier $t \in PTEST$, se cumple que $P \mid t \leq Q \mid t$, podemos afirmar

1. $\forall\, A \subseteq \Sigma,\ a \in A$

$$P \mid t_A \leq Q \mid t_A$$

luego

$$\sum_{a \in A' \subseteq A} p(P, A') \leq \sum_{a \in A' \subseteq A} p(Q, A')$$

2. $\forall\, A \subseteq \Sigma,\ a \in A$

$$P \mid t^{(a,A)} \leq Q \mid t^{(a,A)}$$

luego

$$P//(a, A) \sqsubseteq'_D Q//(a, A)$$

por lo que concluimos que $P \sqsubseteq'_D Q$ como pretendíamos.

La otra implicación ($\Leftarrow$) es más complicada.

Para comprobarla, supondremos en primer lugar, para ir fijando ideas, que nuestro alfabeto está compuesto tan sólo de dos acciones: $\Sigma = \{a, b\}$. Entonces, la expresión que define la probabilidad de aceptación de un test t puede escribirse en la forma

$$P \mid t =$$

$$\begin{array}{l}
\frac{p_a}{p_a}\, p(P,\{a\}) \cdot P/(a,\{a\}) \mid t/a + \frac{p_a}{p_a+p_b} p(P,\{a,b\}) \cdot P/(a,\{a,b\}) \mid t/a + \\
\frac{p_b}{p_b}\, p(P,\{b\}) \cdot P/(b,\{b\}) \mid t/b + \frac{p_b}{p_a+p_b} p(P,\{a,b\}) \cdot P/(b,\{a,b\}) \mid t/b =
\end{array}$$

$$\begin{array}{l}
\frac{p_a}{p_a+p_b} \cdot [(p_a+p_b) \cdot p(P,\{a\}) \cdot P/(a,\{a\}) \mid t/a + \\
\qquad p_a \cdot p(P,\{a,b\}) \cdot P/(a,\{a,b\}) \mid t/a] + \\
\frac{p_b}{p_a+p_b} \cdot [(p_a+p_b) \cdot p(P,\{b\}) \cdot P/(b,\{b\}) \mid t/b + \\
\qquad p_b \cdot p(P,\{a,b\}) \cdot P/(b,\{a,b\}) \mid t/b]
\end{array}$$

Si consideramos que $P/(a,A) \mid t/a \leq Q/(a,A) \mid t/a$, hipótesis asumible en un razonamiento por inducción sobre la profundidad del test, tenemos que para la acción a, por tenerse $P \sqsubseteq'_D Q$, se cumple

$$\begin{array}{l}
p(P,\{a\}) \cdot P/(a,\{a\}) \mid t/a \;\leq\; p(Q,\{a\}) \cdot Q/(a,\{a\}) \mid t/a \\
p(P,\{a\}) \cdot P/(a,\{a\}) \mid t/a + p(P,\{a,b\}) \cdot P/(a,\{a,b\}) \mid t/a \;\leq \\
p(Q,\{a\}) \cdot Q/(a,\{a\}) \mid t/a + p(Q,\{a,b\}) \cdot Q/(a,\{a,b\}) \mid t/a
\end{array}$$

por lo que obtenemos el resultado

$$\begin{array}{rl}
& (p_a+p_b) \cdot p(P,\{a\}) \cdot P/(a,\{a\}) \mid t/a + p_a \cdot p(P,\{a,b\}) \cdot P/(a,\{a,b\}) \mid t/a \\
= & p_a \cdot (p(P,\{a\}) \cdot P/(a,\{a\}) \mid t/a + p(P,\{a,b\}) \cdot P/(a,\{a,b\}) \mid t/a) + \\
& (p_a+p_b) \cdot (p(P,\{a\}) \cdot P/(a,\{a\}) \mid t/a) \\
\leq & p_a \cdot (p(Q,\{a\}) \cdot Q/(a,\{a\}) \mid t/a + p(Q,\{a,b\}) \cdot Q/(a,\{a,b\}) \mid t/a) + \\
& (p_a+p_b) \cdot (p(Q,\{a\}) \cdot Q/(a,\{a\}) \mid t/a)
\end{array}$$

Al ser esto también cierto para la acción b, obtenemos

$$P \mid t \leq Q \mid t$$

Si pasamos ahora a considerar el caso de un alfabeto de tres acciones $\Sigma = \{a,b,c\}$, podemos razonar de una manera parecida. Para simplificar, utilizaremos la siguiente notación

a	representa a	p_a	Para toda acción a
P_A	representa a	$p(P,A) \cdot P/(a,A) \mid t/a$	Para toda acción a, $a \in A$

Entonces tenemos que

$$\begin{array}{rl}
P \mid t = & P_a \frac{a}{a} + P_{ab} \frac{a}{a+b} + P_{ac} \frac{a}{a+c} + P_{abc} \frac{a}{a+b+c} + \\
& P_b \frac{b}{b} + P_{ba} \frac{b}{a+b} + P_{bc} \frac{b}{b+c} + P_{abc} \frac{b}{a+b+c} + \\
& P_c \frac{c}{c} + P_{ca} \frac{c}{a+c} + P_{bc} \frac{c}{b+c} + P_{abc} \frac{c}{a+b+c} +
\end{array}$$

Si nos restringimos a la acción a tenemos

$$\begin{aligned} & P_a\tfrac{a}{a} + P_{ab}\tfrac{a}{a+b} + P_{ac}\tfrac{a}{a+c} + P_{abc}\tfrac{a}{a+b+c} = \\ & \tfrac{a}{a(a+b)(a+c)(a+b+c)} \cdot (\quad P_a(a+b)(a+c)(a+b+c) + \\ & \qquad P_{ab}a(a+c)(a+b+c) + \\ & \qquad P_{ac}(a+b)a(a+b+c) + \\ & \qquad P_{abc}(a+b)(a+c)a) \end{aligned}$$

Puesto que $\frac{a}{a(a+b)(a+c)(a+b+c)}$ aparece tanto en el desarrollo de P como en el de Q, podemos simplificarlo, y con vistas a extraer como factor común en el desarrollo anterior aquél que aparece multiplicando a P_{abc}, reescribimos

$$\begin{aligned} & P_a(a+b)(a+c)(a+b+c) + P_{ab}a(a+c)(a+b+c) + P_{ac}(a+b)a(a+b+c) + \\ & P_{abc}(a+b)(a+c)a \quad = \\ & (P_a + P_{ab} + P_{ac} + P_{abc}) \cdot (a+b)(a+c)a + \\ & (P_a(a+b)(a+c)(b+c) + P_{ab}a(a+c)c + P_{ac}a(a+b)b) \end{aligned}$$

Como $P \sqsubseteq'_D Q$ tenemos entonces que

$$(P_a + P_{ab} + P_{ac} + P_{abc}) \cdot (a+b)(a+c)a \leq (Q_a + Q_{ab} + Q_{ac} + Q_{abc}) \cdot (a+b)(a+c)a$$

por lo que nos queda

$$\begin{aligned} & P_a(a+b)(a+c)(b+c) + P_{ab}a(a+c)c + P_{ac}a(a+b)b \quad = \\ & P_a(a+b)(a+c)c + P_{ab}a(a+c)c + P_a(a+b)(a+c)b + P_{ac}a(a+b)b \end{aligned}$$

Pero tenemos que

$$\begin{aligned} & P_a(a+b)(a+c)c + P_{ab}a(a+c)c \quad = \\ & (P_a + P_{ab})a(a+c)c + P_a(b(a+c)c) \quad \leq \\ & (Q_a + Q_{ab})a(a+c)c + Q_a(b(a+c)c) \end{aligned}$$

y que

$$\begin{aligned} & P_a(a+b)(a+c)b + P_{ac}a(a+c)b \quad = \\ & (P_a + P_{ac})a(a+b)b + P_a(c(a+b)b) \quad \leq \\ & (Q_a + Q_{ac})a(a+b)b + Q_a(c(a+b)b) \end{aligned}$$

por lo que finalmente podemos concluir, sin más que repetir el razonamiento, exactamente como lo hicimos para el caso del alfabeto con dos acciones, y repitiendo el razonamiento para el resto de las acciones del alfabeto,

$$P \mid t \leq Q \mid t$$

Vemos pues, que para el caso del alfabeto con tres acciones, hemos reducido el problema a un par de instancias del mismo para un alfabeto con dos acciones, por cada acción en el alfabeto. Ello nos da la idea de como enfocar el problema en el caso general. Pero a fin de facilitar al lector la comprensión del proceso, antes de exponer la solución general expondremos la misma para el caso de un alfabeto con cuatro acciones. Entonces la comprobación de que la aportación de cada acción es menor en P que en Q puede reducirse a un conjunto de tres comprobaciones análogas sobre un alfabeto de tres acciones. Ello puede ilustrarse esquemáticamente por medio de la siguiente tabla, correspondiente en particular a las aportaciones de la acción a.

P_a	$(a+b)$	$(a+c)$	$(a+d)$	$(a+b+c)$	$(a+b+d)$	$(a+c+d)$	$(a+b+c+d)$
P_{ab}	a	$(a+c)$	$(a+d)$	$(a+b+c)$	$(a+b+d)$	$(a+c+d)$	$(a+b+c+d)$
P_{ac}	$(a+b)$	a	$(a+d)$	$(a+b+c)$	$(a+b+d)$	$(a+c+d)$	$(a+b+c+d)$
P_{ad}	$(a+b)$	$(a+c)$	a	$(a+b+c)$	$(a+b+d)$	$(a+c+d)$	$(a+b+c+d)$
P_{abc}	$(a+b)$	$(a+c)$	$(a+d)$	a	$(a+b+d)$	$(a+c+d)$	$(a+b+c+d)$
P_{abd}	$(a+b)$	$(a+c)$	$(a+d)$	$(a+b+c)$	a	$(a+c+d)$	$(a+b+c+d)$
P_{acd}	$(a+b)$	$(a+c)$	$(a+d)$	$(a+b+c)$	$(a+b+d)$	a	$(a+b+c+d)$
P_{abcd}	$(a+b)$	$(a+c)$	$(a+d)$	$(a+b+c)$	$(a+b+d)$	$(a+c+d)$	a

Al restar la última fila de todas las demás nos queda

P_a	$(a+b)$	$(a+c)$	$(a+d)$	$(a+b+c)$	$(a+b+d)$	$(a+c+d)$	$(b+c+d)$
P_{ab}	a	$(a+c)$	$(a+d)$	$(a+b+c)$	$(a+b+d)$	$(a+c+d)$	$(c+d)$
P_{ac}	$(a+b)$	a	$(a+d)$	$(a+b+c)$	$(a+b+d)$	$(a+c+d)$	$(b+d)$
P_{ad}	$(a+b)$	$(a+c)$	a	$(a+b+c)$	$(a+b+d)$	$(a+c+d)$	$(b+c)$
P_{abc}	$(a+b)$	$(a+c)$	$(a+d)$	a	$(a+b+d)$	$(a+c+d)$	d
P_{abd}	$(a+b)$	$(a+c)$	$(a+d)$	$(a+b+c)$	a	$(a+c+d)$	c
P_{acd}	$(a+b)$	$(a+c)$	$(a+d)$	$(a+b+c)$	$(a+b+d)$	a	b

que efectivamente, podemos transformar en las tres tablas siguientes:

P_a	$(a+b)$	$(a+c)$	$(a+b+c)$	$(a+d)$	$(a+b+d)$	$(a+c+d)$	d
P_{ab}	a	$(a+c)$	$(a+b+c)$	$(a+d)$	$(a+b+d)$	$(a+c+d)$	d
P_{ac}	$(a+b)$	a	$(a+b+c)$	$(a+d)$	$(a+b+d)$	$(a+c+d)$	d
P_{abc}	$(a+b)$	$(a+c)$	a	$(a+d)$	$(a+b+d)$	$(a+c+d)$	d

P_a	$(a+b)$	$(a+d)$	$(a+b+d)$	$(a+c)$	$(a+b+c)$	$(a+c+d)$	c
P_{ab}	a	$(a+d)$	$(a+b+d)$	$(a+c)$	$(a+b+c)$	$(a+c+d)$	c
P_{ad}	$(a+b)$	a	$(a+b+d)$	$(a+c)$	$(a+b+c)$	$(a+c+d)$	c
P_{abd}	$(a+b)$	$(a+d)$	a	$(a+c)$	$(a+b+c)$	$(a+c+d)$	c

P_a	$(a+c)$	$(a+d)$	$(a+c+d)$	$(a+b)$	$(a+b+c)$	$(a+b+d)$	b
P_{ac}	a	$(a+d)$	$(a+c+d)$	$(a+b)$	$(a+b+c)$	$(a+b+d)$	b
P_{ad}	$(a+c)$	a	$(a+c+d)$	$(a+b)$	$(a+b+c)$	$(a+b+d)$	b
P_{acd}	$(a+c)$	$(a+d)$	a	$(a+b)$	$(a+b+c)$	$(a+b+d)$	b

Pero observamos que por ejemplo en la primera todas las filas contienen el factor común $(a+d)(a+b+d)(a+c+d)d$, que por tanto podemos simplificar, al aparecer por igual en los dos lados de la desigualdad que estamos estudiando. Análogamente ocurre con las otras dos tablas, en las que también aparecen sendos factores comunes. Pero la simplificación de los mismos reduce las tablas a las correspondientes a un alfabeto de tres acciones, para los que ya sabemos que se cumple la desigualdad perseguida.

Finalmente, generalizaremos el razonamiento para un alfabeto de n acciones, para lo cual procederemos por inducción, siguiendo el mismo proceso que hemos presentado para el caso de cuatro acciones. En consecuencia, procederemos a estudiar la tabla que define las aportaciones de cada acción, restando a cada fila de la misma la última fila de dicha tabla, comprobándose que la tabla así obtenida puede descomponerse, tras las oportunas simplificaciones de los correspondientes factores comunes, en una familia de $n-1$ tablas con $n-1$ acciones cada una de ellas, con lo que quedará demostrada la desigualdad deseada.

La tabla original en cuestión sería la siguiente:

$$
\begin{array}{lccccccc}
P_{a_1} & (a_1+a_2) & (a_1+a_3) & \ldots & (a_1+a_2+a_3) & \ldots & (a_1+a_2+\ldots+a_n) & + \\
P_{a_1a_2} & a_1 & (a_1+a_3) & \ldots & (a_1+a_2+a_3) & \ldots & (a_1+a_2+\ldots+a_n) & + \\
P_{a_1a_3} & (a_1+a_2) & a_1 & \ldots & (a_1+a_2+a_3) & \ldots & (a_1+a_2+\ldots+a_n) & + \\
\ldots & & & & & & & \\
P_{a_1a_2a_3} & (a_1+a_2) & (a_1+a_3) & \ldots & (a_1+a_2+a_3) & \ldots & (a_1+a_2+\ldots+a_n) & + \\
\ldots & & & & & & & \\
P_{a_1\ldots a_n} & (a_1+a_2) & (a_1+a_3) & \ldots & (a_1+a_2+a_3) & \ldots & a_1 &
\end{array}
$$

La suma global T_n por ella definida, la podemos reescribir en la forma

$$T_n = \sum_{a \in A \subseteq \Sigma} P_A \cdot \prod_{a \in A' \neq \{a\}} c_{A,A'}$$

donde, para cada par A, A' el valor $c_{A,A'}$ viene definido por

$$c_{A,A'} = \begin{cases} \sum_{b \in A'} b & \text{Si } A \neq A' \\ a & \text{Si } A = A' \neq \{a\} \end{cases}$$

Entonces, tras restar los sumandos correspondientes a la última fila obtenemos una nueva tabla, cuyo valor definido por la misma T'_n viene dado por

$$T'_n = T_n - \sum_{A \subseteq \Sigma} P_A \cdot \prod_{a \in A' \neq \{a\}} c_{\Sigma,A'}$$

y puesto que sabemos

$$\sum_{A \subseteq \Sigma} P_A \leq \sum_{A \subseteq \Sigma} Q_A$$

obtenemos

$$\sum_{A \subseteq \Sigma} P_A \cdot \prod_{a \in A' \neq \{a\}} c_{\Sigma,A'} \leq \sum_{A \subseteq \Sigma} Q_A \cdot \prod_{a \in A' \neq \{a\}} c_{\Sigma,A'}$$

Queda entonces por probar que el valor T'_n correspondiente al desarrollo del proceso P es menor que el correspondiente al desarrollo del proceso Q. Pero ello es cierto, pues

$$\begin{aligned}
T'_n &= \sum_{A \neq \Sigma} \prod_{A' \neq \Sigma} c_{A,A'} \cdot (\Sigma - A) \\
&= \sum_{b \in \Sigma - \{a\}} \sum_{A \neq \Sigma} \prod_{A' \neq \Sigma} c_{A,A'} \\
&= \sum_{b \in \Sigma - \{a\}} \sum_{A \neq \Sigma} \Big(\prod_{b \notin A' \neq \Sigma} c_{A,A'} \cdot \prod_{b \in A' \neq \Sigma} c_{A,A'} \Big) \\
&= \sum_{b \in \Sigma - \{a\}} \prod_{b \notin A' \neq \Sigma} c_{A,A'} \cdot \sum_{A \neq \Sigma} \prod_{b \in A' \neq \Sigma} c_{A,A'}
\end{aligned}$$

cuyo último paso es correcto, gracias a que los coeficientes $c_{A,A'}$ correspondientes a los conjuntos A' con $b \notin A'$ aparecen en todos los sumandos, por lo que pueden extraerse como factor común. Como dichos factores comunes son los mismos para P que para Q, pueden ser simplificados, quedándonos entonces una familia de términos cada uno de los cuales corresponde a una tabla de un alfabeto con $n-1$ acciones, por lo que podemos aplicar la hipótesis de inducción para concluir la desigualdad deseada. □

En base a esta nueva relación de orden podríamos definir un modelo denotacional alternativo (sobre el mismo conjunto de árboles probabilísticos), en base a las mismas definiciones de los operadores semánticos. Naturalmente, deberíamos comprobar que los mismos siguen siendo monótonos y contínuos, en relacion al nuevo orden. Ello exigiría pruebas similares a las realizadas en la Sección 5.2, y como quiera que las mismas no añadirían ninguna idea novedosa, y por otra parte el resultado a probar no resulta especialmente importante, hemos decidido omitir.

Ahora bien, la semántica de PCSP definida por dicho modelo coincidiría con la previamente definida, de cara a probar dicha afirmación bastaría probar que los supremos respecto de ambos ordenes de las cadenas de la forma $\langle F_{\perp}^{n} \rangle_{n \in \mathbb{N}}$ coinciden. Pero ello es así dado que $\sqsubseteq'_D = \sqsubseteq_T$, y por la forma en que están definidas las semánticas inducidas por estas relaciones de orden, se tiene

$$[\![P_n]\!]' \mid t = P \mid t$$

Por otra parte, el corolario 5.1 nos indica que

$$[\![P_n]\!] \mid t = P \mid t$$

de donde se concluye inmediatamente la igualdad entre ambas semánticas denotacionales.

Capítulo 6

Sistema de axiomas y reglas

En este capítulo introduciremos un Sistema de Axiomas y Reglas correcto y completo respecto de la semántica de pruebas, que nos permitirá probar equivalencias y relaciones de orden entre procesos. Comenzaremos nuestro estudio restringiéndonos a un subconjunto básico de PCSP, que denominaremos FPCSP, que incluirá sólo a los términos finitos que pueden construirse a partir del proceso *STOP*, utilizando el operador prefijo y las dos clases de elecciones. Para este lenguaje, ofreceremos un conjunto de axiomas y reglas de inferencia, las cuales probaremos que forman un sistema correcto y completo respecto de la semántica de pruebas.

Posteriormente incluiremos en el lenguaje los operadores de composición paralela y ocultación. Veremos que los mismos son operadores derivados, por lo que podemos dar una serie de axiomas que permiten eliminarlos de la sintaxis de los procesos (módulo la equivalencia semántica).

Finalmente, generalizaremos el lenguaje incorporando los términos infinitos, definidos de manera recursiva. Además de la preceptiva regla infinitaria, que es necesario introducir usualmente para cubrir la recursión, será necesario en esta ocasión introducir una serie de modificaciones en el sistema, a fin de conseguir la deseada completitud del sistema.

6.1 PCSP finito

Llamaremos FPCSP (PCSP finito) al lenguaje generado por la siguiente sintaxis:

$$P \;=\; STOP \mid a \rightarrow P \mid [p]P \Box [1-p]P \mid [p]P \sqcap [1-p]P$$

Construiremos un sistema lógico cuyas fórmulas atómicas expresarán la equivalencia, por medio de de un símbolo de relación binaria $\equiv$, entre dos términos FPCSP . En el sistema de prueba incluiremos axiomas que expresen propiedades como la idempotencia, simetría, asociatividad, distributividad de ciertos operadores del lenguaje, junto a otros que definen el comportamiento de otros operadores respecto del operador prefijo, que en alguna forma es el más básico, por ser el único que en sí mismo expresa un comportamiento dinámico de cambio de estado. Por otra parte, las reglas de inferencia asertarán sobre la monotonía de los operadores respecto de la equivalencia etre procesos.

6.1.1 Sistema de axiomas y corrección de los mismos

A continuación se presentan los axiomas del sistema que se propone, junto con la prueba de corrección de cada uno de ellos, adjuntando una breve explicación cuando ello se considere necesario.

- El operador $\sqcap$ es idempotente.

  A1] $[p]P \sqcap [1-p]P \;\equiv\; P$

  La demostración es trivial.

- El operador $\sqcap$ es conmutativo.

  A2] $[p]P \sqcap [1-p]Q \;\equiv\; [1-p]Q \sqcap [p]P$

  Demostración: Sea $t \in PTEST$, entonces

$$\begin{aligned} [p]P \sqcap [1-p]Q \mid t &= p \cdot P \mid t + (1-p) \cdot Q \mid t \\ &= (1-p) \cdot Q \mid t + p \cdot P \mid t \\ &= [1-p]Q \sqcap [p]P \mid t \end{aligned}$$

- El operador $\sqcap$ es asociativo, salvo la necesidad de reequilibrar las probabilidades involucradas.

  A3] $[p]P \sqcap [q+r]([\frac{q}{q+r}]Q \sqcap [\frac{r}{q+r}]R) \equiv [p+q]([\frac{p}{p+q}]P \sqcap [\frac{q}{p+q}]Q) \sqcap [r]R$

  Demostración: Sea $t \in PTEST$, entonces

$$\begin{aligned}
&[p]P \sqcap [q+r]([\frac{q}{q+r}]Q \sqcap [\frac{r}{q+r}]R) \mid t \\
&\quad = p \cdot (P \mid t) + q \cdot (Q \mid t) + r \cdot (R \mid t) \\
&\quad = [p+q]([\frac{p}{p+q}]P \sqcap [\frac{q}{p+q}]Q) \sqcap [r]R \mid t
\end{aligned}$$

La asociatividad y conmutatividad del operador $\sqcap$, nos permite extender de manera natural el mismo a un número arbitrario de operandos, y no sólo utilizarlo como operador binario, tal y como ha sido definido en la sintaxis. Así, si $\sum_i p_i = 1$ y $\{P_i | i = 1, \ldots, n\}$ es una familia de procesos, definimos el proceso

$$\bigsqcap_{i=1}^{n} [p_i]P_i$$

que denota al proceso resultado de una serie cualquiera de operaciones binarias de elección interna, que tengan como argumentos a los procesos P_i que se indican, y probabilidades adecuadamente ponderadas en función de los p_i dados. Más exactamente, definimos

$$\bigsqcap_{i=1}^{n} [p_i]P_i \;=\; [p_1]P_1 \sqcap [\sum_{i=2}^{n} p_i]([\frac{p_2}{1-p_1}]P_2 \sqcap \;\ldots [\frac{p_n}{1-\sum_{i=1}^{n-1} p_i}]P_n \ldots)$$

donde el orden de los operandos es significativo en principio, si bien a la postre ello no sería cierto, pues las propiedades de asociatividad y conmutatividad de la elección interna binaria nos permiten reescribir la expresión en el lado derecho de la definición en la correspondiente a cualquier reordenación y reagrupación de sus operandos.

- El operador $\square$ es conmutativo.

  A4] $[p]P \square [1-p]Q \equiv [1-p]Q \square [p]P$

Demostración: En efecto, si $t \in PTEST$, tenemos

$$\begin{aligned}
&[p]P\Box[1-p]Q \mid t \\
&= \sum_{a\in Ft(t)} \sum_{\substack{A,B\subset\Sigma \\ a\in A\cup B}} s(a,A\cup B,t)\cdot p(P,A)\cdot p(Q,B)\cdot Cont(\Box,t,A,B,a) \\
&= \sum_{a\in Ft(t)} \sum_{\substack{A,B\subset\Sigma \\ a\in B\cup A}} s(a,A\cup B,t)\cdot p(P,B)\cdot p(Q,A)\cdot Cont(\Box,t,B,A,a) \\
&= [1-p]Q\Box[p]P \mid t
\end{aligned}$$

- El proceso $STOP$ es elemento neutro para el operador $\Box$.

  A5] $[p]P\Box[1-p]STOP \equiv P$

  Demostración:Para todo $t \in PTEST$ se tiene

$$\begin{aligned}
&[p]P\Box[1-p]STOP \mid t \\
&= \sum_{a\in Ft(t)} \sum_{\substack{A\subset\Sigma \\ a\in A}} s(a,A,t)\cdot p(P,A)\cdot p(STOP,\emptyset)\cdot Cont(\Box,t,A,\emptyset,a) \\
&= \sum_{a\in Ft(t)} \sum_{\substack{A\subset\Sigma \\ a\in A}} s(a,A,t)\cdot p(P,A)\cdot (P/(a,A) \mid (t/a) \\
&= P \mid t
\end{aligned}$$

- El operador prefijo distribuye sobre el operador $\sqcap$.

  A6] $[p](a\to P)\sqcap[1-p](a\to Q) \equiv a\to([p]P\sqcap[1-p]Q)$

  Demostración: Sea $t \in PTEST$, con $a \in Ft(t)$. Entonces

$$\begin{aligned}
&[p](a\to P)\sqcap[1-p](a\to Q) \mid t \\
&= p\cdot(a\to P \mid t)+(1-p)\cdot(a\to Q \mid t) \\
&= p\cdot(P \mid t/a)+(1-p)\cdot(Q \mid t/a) \\
&= (a\to([p]P\sqcap[1-p]Q)) \mid t
\end{aligned}$$

  Por otra parte, si $a \notin Ft(t)$ y $t \neq \omega$ tenemos

$$\begin{aligned}
&[p](a\to P)\sqcap[1-p](a\to Q) \mid t &= 0 \\
&(a\to([p]P\sqcap[1-p]Q)) \mid t &= 0
\end{aligned}$$

Finalmente, para $t = \omega$

$$\begin{aligned} [p](a \to P) \sqcap [1-p](a \to Q) \mid t &= 1 \\ (a \to ([p]P \sqcap [1-p]Q)) \mid t &= 1 \end{aligned}$$

Esta propiedad del operador prefijo es interesante con vistas a retrasar las elecciones internas hasta después de seleccionar cual es la acción que vamos a ejecutar, en los casos en que ello tiene sentido.

- La elección externa entre dos procesos que comienzan con una misma acción se convierte en interna.

  A7] $[p](a \to P) \Box [1-p](a \to Q) \equiv a \to ([p]P \sqcap [1-p]Q)$

  Demostración: Sea $t \in PTEST$, con $a \in Ft(t)$

$$\begin{aligned} & [p](a \to P) \Box [1-p](a \to Q) \mid t \\ &= s(a, \{a\}, t) \cdot p(a \to P, \{a\}) \cdot p(a \to Q, \{a\}) \cdot Cont(\Box, t, \{a\}, \{a\}, a) \\ &= 1 \cdot 1 \cdot 1 \cdot (p \cdot (P \mid t/a) + (1-p) \cdot (Q \mid t/a)) \\ &= (a \to ([p]P \sqcap [1-p]Q)) \mid t \end{aligned}$$

  Por otra parte, si $a \notin Ft(t)$

$$\begin{aligned} [p](a \to P) \Box [1-p](a \to Q) \mid t &= 0 \\ (a \to ([p]P \sqcap [1-p]Q)) \mid t &= 0 \end{aligned}$$

  Finalmente, para $t = \omega$

$$\begin{aligned} [p](a \to P) \Box [1-p](a \to Q) \mid t &= 1 \\ (a \to ([p]P \Box [1-p]Q)) \mid t &= 1 \end{aligned}$$

  Mediante este axioma, podremos eliminar las elecciones externas que de hecho introducen un factor de no-determinismo interno, debido a que la elección externa que proporcionan no es tal, pues entre acciones iguales, quien en definitiva resuelve es el sistema.

- Las probabilidades asociadas a una elección externa entre procesos que comienzan por acciones distintas no tienen ningún significado. Es decir, si $a \neq b$, $\forall p, q \in [0,1]$ se tiene

A8] $[p](a \to P)\Box[1-p](b \to Q) \equiv [q](a \to P)\Box[1-q](b \to Q)$

Demostración: En efecto, si $t \in PTEST$

$$\begin{aligned}
&[p](a \to P)\Box[1-p](b \to Q) \mid t \\
&\quad = s(a,\{a,b\},t)\cdot p(a \to P,\{a\})\cdot p(b \to Q,\{b\})\cdot Cont(\Box,t,\{a\},\{b\},a) \\
&\qquad + s(b,\{a,b\},t)\cdot p(a \to P,\{a\})\cdot p(a \to Q,\{b\})\cdot Cont(\Box,t,\{a\},\{b\},b) \\
&\quad = s(a,\{a,b\},t)\cdot P \mid t/a \;+\; s(b,\{a,b\},t)\cdot Q \mid t/a \\
&\quad = [q](a \to P)\Box[1-q](b \to Q) \mid t
\end{aligned}$$

Como resultado de este axioma, cuando la elección externa es entre acciones distintas, no tiene sentido hablar de probabilidades, pues el sistema nunca tendrá que decidir qyé proceso ejecutará la acción elegida, puesto que cada una de ellas sólo puede ser ejecutada por un operando. Esto nos permitiría llegar a prescindir en tales casos de esta información en la sintaxis del operador. Sin embargo, a fin de mantenernos fieles a la misma, lo que haremos cuando se nos presente esta situación a nivel sintáctico, será utilizar como distribución de probabilidad (irrelevante) una concreta, que será aquélla que viene dada por un reparto equiprobable entre los procesos componentes.

Mientras que en CSP clásico se cumplían las dos leyes de distributividad que relacionan las elecciones interna y externa, en nuestro PCSP sólo será válida una de ellas, concretamente la distributividad de $\Box$ sobre $\sqcap$, no siendo cierta en general en nuestro modelo probabilístico, la distributividad contraria.

- El operador $\Box$ distribuye sobre $\sqcap$.

  A9] $[p]P\Box[1-p]([q]Q \sqcap [1-q]R) \equiv$
  $[q]([p]P\Box[1-p]Q) \sqcap [1-q]([p]P\Box[1-p]R)$

Demostración: Para probar la correción de este axioma tomemos $t \in PTEST$.

Por un lado sabemos que

$$[q]Q \sqcap [1-q]R \mid t = q(Q \mid t) + (1-q)(R \mid t)$$

Entonces, llamando $S = [q]Q \sqcap [1-q]R$ tenemos

$$[p]P\Box[1-p]S \mid t = \sum_{a \in Fi(t)} \sum_{\substack{A,B \subset \Sigma \\ a \in A \cup B}} s(a, A \cup B, t)\cdot p(P,A)\cdot p(S,B)\cdot Cont(\Box,t,A,B,a)$$

Puesto que S es una elección interna, la probabilidad de que S alcance el estado B es

$$p(S,B) \;=\; q \cdot p(Q,B) + (1-q) \cdot p(R,B)$$

entonces, tenemos que este estado B puede ser alcanzado de tres formas diferentes: bien por haber sido alcanzado por el proceso Q (lo que ocurre con probabiblidad q), siendo este estado imposible de ser alcanzado por el proceso R; bien por darse la situación inversa, esto es ser alcanzado por R siendo también imposible de ser alcanzado por Q; o bien por ser alcanzado por uno u otro siendo también posible que lo hubiese alcanzado el contrario. Así pues, denotando por B_1 a los estados en los que ocurre la primera situación, por B_2 a los correspondientes a la segunda, y por B_3 a los de la última, se tiene

$$\begin{array}{lll} p(Q,B_1) > 0 & \& & p(R,B_1) = 0 \\ p(Q,B_2) = 0 & \& & p(R,B_2) > 0 \\ p(Q,B_3) > 0 & \& & p(R,B_3) > 0 \end{array}$$

Podemos descomponer la probabilidad de que la expresión en el lado izquierdo del axioma A9 pase un test t, en la forma siguiente:

$$\begin{aligned} [p]P\Box[1-p]S \mid t \;=\; & \sum_{a \in Ft(t)} \Big[\\ & \sum_{A,B_1} s(a, A \cup B_1, t) \cdot p(P,A) \cdot p(Q,B_1) \cdot q \cdot Cont(\Box, t, A, B_1, a) + \\ & \sum_{A,B_2} s(a, A \cup B_2, t) \cdot p(P,A) \cdot p(R,B_2) \cdot (1-q) \cdot Cont(\Box, t, A, B_2, a) + \\ & \sum_{A,B_3} s(a, A \cup B_3, t) \cdot p(P,A) \cdot p(Q,B_3) \cdot q \cdot Cont(\Box, t, A, B_1, a) + \\ & \sum_{A,B_3} s(a, A \cup B_3, t) \cdot p(P,A) \cdot p(R,B_3) \cdot (1-q) \cdot Cont(\Box, t, A, B_1, a) \Big] \end{aligned}$$

Ahora bien, la continuación del proceso $[p]P\Box[1-p]S$ tras ejecutar la acción a está condicionada por el estado que ha sido alcanzado por los procesos componentes P y S. Así, cuando sea S quien continúe la ejecución, si el estado alcanzado es del tipo B_1 deberá ser Q el que lo haga (puesto que R no puede); si es del tipo B_2 será R el que continúe; mientras que si es del tipo B_3, entonces con probabilidad q continuará Q y con probabilidad $1-q$ lo hará R. Por tanto la suma anterior

puede escribirse en la forma

$$q \sum_{a \in Fi(t)} \sum_{A,B} s(a, A \cup B, t) p(P, A) p(Q, B) Cont([p]P \Box [1-p]Q, t, A, B, a) +$$
$$(1-q) \sum_{a \in Fi(t)} \sum_{A,B} s(a, A \cup B, t) p(P, A) p(R, B) Cont([p]P \Box [1-p]R, t, A, B, a)$$

Por lo cual obtenemos

$$[p]P \Box [1-p]([q]Q \sqcap [1-q]R) \mid t =$$
$$q \cdot ([p]P \Box [1-p]Q \mid t) + (1-q) \cdot ([p]P \Box [1-p]R \mid t) =$$
$$[q]([p]P \Box [1-p]Q) \sqcap [1-q]([p]P \Box [1-p]R) \mid t$$

lo que prueba la corrección del axioma.

La distributividad contraria, tal y como ya se ha indicado no se satisface en general en nuestro modelo probabilístico. Esto puede comprobarse mediante el siguiente ejemplo.

Ejemplo 6.1 Consideremos el siguiente proceso

$$[\frac{1}{2}](a \rightarrow STOP) \sqcap [\frac{1}{2}]([p]P \Box [1-p]Q)$$

Supongamos que ni P ni Q pueden ejecutar la acción a inicialmente. Entonces este proceso no es equivalente al que se obtendría aplicando la posible ley distributiva, el cual vendría dado por

$$[p]([\frac{1}{2}](a \rightarrow STOP) \sqcap [\frac{1}{2}]P) \Box [1-p]([\frac{1}{2}](a \rightarrow STOP) \sqcap [\frac{1}{2}]Q)$$

Ello es en efecto así, puesto que por ejemplo, en el primer caso la probabilidad de aceptar el test $a.\omega$ es $\frac{1}{2}$, mientras que en el segundo, la probabilidad es $\frac{3}{4}$. □

Desgraciadamente, en PCSP tampoco podemos dar un axioma general equivalente al presentado en A3 para la elección interna, pues en este lenguaje la propiedad asociativa para el operador $\Box$ no es cierta, ni aún con un adecuado reequilibrio de las probabilidades. Ello es así, pues en general la equivalencia siguiente

$$[p]P \Box [q+r]([\frac{q}{q+r}]Q \Box [\frac{r}{q+r}]R) \equiv [p+q]([\frac{p}{p+q}]P \Box [\frac{q}{p+q}]Q) \Box [r]R$$

no se tiene.

Comprobaremos que ello es así mediante el siguiente contraejemplo:

Ejemplo 6.2 Dados los dos procesos siguientes

$$\begin{aligned} P_1 &= \tfrac{1}{3}(a \to b \to STOP) \,\Box\, \tfrac{2}{3}(\tfrac{1}{2}a \to STOP \,\Box\, \tfrac{1}{2}b \to STOP) \\ P_2 &= \tfrac{2}{3}(\tfrac{1}{2}a \to b \to STOP) \,\Box\, \tfrac{1}{2}a \to STOP) \,\Box\, \tfrac{1}{3}b \to STOP) \end{aligned}$$

que se corresponden con las instancias de las expresiones que aparecen a ambos lados de la anterior equivalencia correspondientes a una misma valoración de las variables que en la misma aparecen, tenemos que al serles ofrecido el test $a.b.\omega$, para su aceptación, el primero de ellos lo pasa con éxito con probabilidad $\frac{1}{3}$, mientras que el proceso P_2 aceptará el test con probabilidad $\frac{1}{2}$. □

La no asociatividad del operador $\Box$ podría llegar a tener unos efectos catastróficos en nuestro proceso de búsqueda de formas normales, al complicarse el mismo tremendamente, pues en principio no quedamos capacitados para generalizar el operador a una familia arbitraria de argumentos.

Afortunadamente, aunque el operador no sea asociativo, dicha generalización sigue siendo posible, aún cuando no podremos hablar exactamente del propio operador $\Box$ como un operador n-ario, pues la generalización en cuestión no es absoluta sino sólo parcial, si bien, afortunadamente, la misma resulta suficiente para los objetivos que perseguimos.

Como punto de partida de cara a su definición, nos fijaremos en los axiomas A7 y A8, que nos suministran las propiedades precisas para abordar la generalización deseada.

El axioma A7 nos indica que una elección externa entre dos acciones iguales convierte a aquella en interna. Tenemos entonces que restringido a este caso el operador sí es asociativo, pues la elección interna lo es, de manera que podemos generalizar el operador elección externa a n operandos siempre que éstos comiencen ofreciendo exclusivamente una misma acción.

Por su parte, el axioma A8 nos dice que al elegir externamente entre dos acciones distintas, las probabilidades del operador carecen de utilidad, por lo que de

nuevo en este caso la generalización es posible, pues si los n procesos a combinar comienzan con acciones diferentes, podemos repartir la probabilidad asociada a los mismos de manera uniforme, tomando $\frac{1}{n}$ para cada uno, pues dicha distribución de propiedad resulta a la postre irrelevante.

Es preciso sin embargo una generalización más amplia, la cual precisará unos axiomas más generales, los cuales no pueden ser definidos en base al operador binario de elección externa, sino que precisan de la correspondiente variante n-aria, que pasamos a definir a continuación.

Definición 6.1 Sea un conjunto de acciones $\{a_i\}_{i=1,\ldots,n} \subseteq \Sigma$ no vacío, y un conjunto de procesos $\{P_i\}_{i=1,\ldots,n}$; entonces definimos el proceso $\Box_{i=1}^{n} [\frac{1}{n}]a_i \rightarrow P_i$, resultado de aplicar a dichos argumentos el nuevo operador de elección externa prefijada, en la forma:

1. $\Box_{i=1}^{1} [1]a_i \rightarrow P_i \;=\; a_1 \rightarrow P_1$
2. $\Box_{i=1}^{n} [\frac{1}{n}]a_i \rightarrow P_i \;=\; [\frac{1}{n}]a_1 \rightarrow P_1 \;\Box\; [\frac{n-1}{n}](\Box_{i=1}^{n-1} a_{i+1} \rightarrow P_{i+1}) \quad \text{con } n > 1.$

□

A partir de esta definición, podemos obtener la probabilidad de que un proceso construido con este operador pase un test. Como siempre, la misma se calculará a partir de las probabilidades de que alcance cada posible estado, junto con las probabilidades de que a partir de cada uno de dichos estados pase el resto del test.

1. $p(\Box_{i=1}^{n} [\frac{1}{n}]a_i \rightarrow P_i, A) = \begin{cases} 1 & \text{Si } A = \{a_1, \ldots, a_n\} \\ 0 & \text{Si } A \neq \{a_1, \ldots, a_n\} \end{cases}$
2. $(\Box_{i=1}^{n} [\frac{1}{n}]a_i \rightarrow P_i)/(a, A) \mid t = \begin{cases} P_i \mid t & \text{Si } A = \{a_1, \ldots, a_n\} \text{ y } a = a_i \\ 0 & \text{Si } A \neq \{a_1, \ldots, a_n\} \text{ ó } a \notin A \end{cases}$

Utilizando este nuevo operador, podemos escribir los mismos procesos PCSP (módulo equivalencia semántica), pero dejando de utilizar el operador binario □,

usando en su lugar el nuevo operador. Para capturar axiomáticamente tal hecho, sustituimos los axiomas anteriores $A7$ y $A8$ por uno nuevo, más general

$$\text{A78]} \quad [p]\overset{n}{\underset{i=1}{\Box}}[\tfrac{1}{n}]a_i \to P_i \Box [1-p]\overset{m}{\underset{j=1}{\Box}}[\tfrac{1}{m}]b_j \to P_j \;\equiv\; \overset{l}{\underset{k=1}{\Box}}[\tfrac{1}{l}]c_k \to R_k$$

donde

$$l = |\{a_1, \ldots a_n\} \cup \{b_1, \ldots b_m\}|$$

$$R_k = \begin{cases} P_i & \text{Si } c_k = a_i \wedge c_k \notin \{b_j\} \\ Q_j & \text{Si } c_k = b_j \wedge c_k \notin \{a_i\} \\ [p]P_i \sqcap [1-p]Q_j & \text{Si } c_k = a_i \wedge c_k \notin \{b_j\} \end{cases}$$

Proposición 6.1 El axioma A78 es correcto.

Demostración: Llamando

$$P_1 = \overset{n}{\underset{i=1}{\Box}}[\tfrac{1}{n}]a_i \to P_i$$
$$P_2 = \overset{m}{\underset{j=1}{\Box}}[\tfrac{1}{m}]a_j \to P_j$$
$$P = \overset{l}{\underset{k=1}{\Box}}[\tfrac{1}{l}]c_k \to R_k$$

tenemos que para todo test $t \in PTEST$, se satisface

$$[p]P_1 \Box [1-p]P_2 \mid t = \sum_{a \in Ft(t)} \sum_{\substack{A,B \subset \Sigma \\ a \in A \cup B}} s(a, A \cup B, t)\, p(P_1, A)\, p(P_2, B)\, Cont(\Box, t, A, B, a)$$

Ahora bien, por la definición de los procesos P_1 y P_2, tenemos que los únicos estados que dichos procesos pueden alcanzar son, respectivamente, $\{a_i \mid i = 1, \ldots, n\}$ y $\{b_j \mid j = 1, \ldots, m\}$. Por lo que

$$[p]P_1 \Box [1-p]P_2 \mid t = \sum_{a \in Ft(t)} s(a, \{c_k\}, t) \cdot Cont(\Box, t, \{a_i\}, \{b_j\}, a) = P \mid t$$

□

Además, también debemos modificar el axioma $A6$ para poder así generalizar la propiedad que el mismo captura, a una situación más general que corresponde al uso del nuevo operador de elección externa prefijada, la cual no sería posible tratar limitandonos al uso de la anterior formulación del axioma.

Efectivamente, en dicho axioma sólo se indicaba la distributividad entre el prefijo y la elección interna, que ahora se extiende a la elección externa prefijada, dando lugar al siguiente enunciado:

A6'] $[p](\underset{a\in A}{\Box}[\frac{1}{|A|}]a \to P_a) \sqcap [1-p](\underset{a\in A}{\Box}[\frac{1}{|A|}]a \to Q_a) \equiv$

$$\underset{a\in A}{\Box}[\tfrac{1}{|A|}]a \to ([p]P_a \sqcap [1-p]Q_a)$$

Proposición 6.2 El axioma $A6'$ es correcto.

Demostración: Sea $t \in PTEST$, entonces

$$\begin{aligned}
&[p](\underset{a\in A}{\Box}[\frac{1}{|A|}]a \to P_a) \sqcap [1-p](\underset{a\in A}{\Box}[\frac{1}{|A|}]a \to Q_a) \mid t \\
&= p \cdot P_a \mid t/a + (1-p) \cdot Q_a \mid t/a \\
&= ([p]P_a \sqcap [1-p]Q_a) \mid t/a \\
&= \underset{a\in A}{\Box}[\frac{1}{|A|}]a \to ([p]P_a \sqcap [1-p]Q_a) \mid t
\end{aligned}$$

□

Como se sigue inmediatamente de su definición, el nuevo operador n-ario $\overset{n}{\underset{i=1}{\Box}}$ generaliza al operador prefijo, pues $a_1 \to P_1$ puede escribirse como $\overset{1}{\underset{i=1}{\Box}}[1]a_i \to P_i$. El operador binario $\Box$ también queda generalizado, si bien ello sólo es cierto de forma inmediata, de una manera parcial. En particular, si $a_1 \neq a_2$ tenemos que el proceso resultado de la operación binaria $[p]a_1 \to P_1 \Box [1-p]a_2 \to P_2$ se reescribiría como $\overset{2}{\underset{i=1}{\Box}}[\frac{1}{2}]a_i \to P_i$.

Sin embargo, contamos con el siguiente resultado general, que nos muestra que tras las manipulaciones adecuadas, el operador binario $\Box$ puede ser totalmente eliminado, apareciendo elecciones externas prefijadas en su lugar.

Proposición 6.3 Si $P = [p]P_1 \Box [1-p]P_2$, entonces podemos eliminar la aparición del operador binario $\Box$, recurriendo al operador de elección prefijada.

Demostración:

- Si uno de los procesos es STOP entonces tenemos

$$[p]P\Box[1-p]STOP \equiv P$$

- Si uno de los procesos es una elección interna $\prod_{i=1}^{n}[p_i]Q_i$ tenemos que, aplicando el axioma A9

$$[p]P\Box[1-p](\prod_{i=1}^{n}[p_i]Q_i)\prod_{i=1}^{n}[p_i]([p]P\Box[1-p]Q_i)$$

por lo que por medio de una simple inducción, el resultado quedaría probado.

- Por último, si los dos procesos son elecciones externas prefijadas

$$\begin{aligned} P_1 &= \Box_{i=1}^{n}[\tfrac{1}{n}]a_i \rightarrow P_i \\ P_2 &= \Box_{j=1}^{m}[\tfrac{1}{m}]a_j \rightarrow P_j \end{aligned}$$

basta aplicar el axioma A78.

□

En realidad, la introducción del operador general $\Box_{i=1}^{n}$, puede ser considerada como un mero *truco* sintáctico (*azúcar* sintáctica) para facilitar las pruebas de equivalencia entre procesos. Como quiera que este operador no es necesario para la descripción y especificación de procesos, no se ha considerado razonable introducirlo en la sintaxis original.

6.1.2 Reglas de Inferencia

Las reglas de inferencia de nuestro sistema lógico, son las siguientes:

$$\text{(E1)}\quad \frac{}{P \equiv P}$$

$$\text{(E2)}\quad \frac{P \equiv Q}{Q \equiv P}$$

$$\text{(E3)}\quad \frac{P \equiv Q \ \wedge \ Q \equiv R}{P \equiv R}$$

$$\text{(C1)}\quad \frac{P \equiv Q}{a \rightarrow P \ \equiv \ a \rightarrow Q}$$

$$\text{(C2)}\quad \frac{P \equiv Q \ \wedge \ P' \equiv Q'}{([p]P \sqcap [1-p]P') \ \equiv \ ([p]Q \sqcap [1-p]Q')}$$

$$\text{(C3)}\quad \frac{P \equiv Q \ \wedge \ P' \equiv Q'}{([p]P \,\square\, [1-p]P') \ \equiv \ ([p]Q \,\square\, [1-p]Q')}$$

Las tres primeras nos indican que la igualdad es una relación de equivalencia, cumpliendo en consecuencia las propiedades reflexiva, simétrica y transitiva. Por su parte, las tres últimas nos dicen que dicha relación de equivalencia es una congruencia respecto de los operadores básicos del lenguaje. Puesto que la semántica de pruebas a caracterizar es composicional, la prueba de corrección de las mismas es trivial.

Teorema 6.1 (Corrección)
Para todo par de términos $P,\ Q\ \in FPCSP$ tenemos que $\vdash P \equiv Q$ implica que

$$\forall t \in PTEST \quad P \mid t = Q \mid t$$

Demostración: Corolario inmediato del hecho de que todos los axiomas y reglas del sistema son correctos. □

6.1.3 Completitud

Hemos mostrado que si es posible probar en nuestro sistema la equivalencia entre procesos, entonces necesariamente las observaciones de dichos procesos coinciden. Ahora, nuestro objetivo será el recíproco, es decir probar que si las observaciones de dos procesos son las mismas, entonces el sistema de prueba debe servirnos para probar su equivalencia.

Como es habitual en este tipo de pruebas, para llegar a dicha conclusión nos serviremos de un mecanismo de *Formas Normales*. Definiremos una clase de Formas Normales, y demostraremos que todo término es equivalente, vía el sistema de axiomas, a una Forma Normal. Posteriormente probaremos que si dos Formas Normales son diferentes, entonces existe algún test que las distingue.

Esencialmente, las Formas Normales son las representaciones sintácticas de los procesos semánticos del capítulo 5, si bien como de momento nos centramos en procesos finitos, sólo aparecen términos finitos y sin divergencias. Por tanto se trata de elecciones internas generalizadas, de acuerdo con una cierta distribución, cada uno de cuyos componentes será una elección externa prefijada entre un conjunto de procesos guardados, cada uno de ellos por una acción diferente. Imponemos además la condición de que los conjuntos de guardas de cada uno de los procesos componentes de la elección interna generalizada deben ser diferentes.

Definición 6.2 Un término P decimos que está en *Forma Normal* si y sólo si tiene una estructura de la forma:

$$P = \bigsqcap_{i=1}^{n} [p_i] \Box_{j=1}^{m} [\frac{1}{m}] a_{ij} \rightarrow P_{ij}$$

donde

$n \geq 1$

$m \geq 0$ Si $m = 0$ abreviamos con $STOP$

$\forall i \;\; p_i > 0$

$\sum_{i=1}^{n} p_i \;=\; 1$

$j \neq k \Rightarrow a_{ij} \neq a_{ik}$

$i \neq k \Rightarrow \{a_{ij}\}_{j\in 1,m_i} \neq \{a_{kj}\}_{j\in 1,m_k}$
P_{ij} está en forma normal

□

Como indicamos más arriba, los conjuntos $\{a_{ij}\}_{j\in J}$ se corresponden con los *estados* que aparecían al estudiar las semánticas precedentes

Comentario: Hemos de remarcar dos importantes diferencias con respecto a las formas normales correspondientes al lenguaje CSP clásico. En primer lugar, tenemos que los conjuntos $\{a_{ij}\}_{j\in J}$ no necesitan ser convexos, puesto que existen tests que nos encuentran la probabilidad de alcanzar cada estado. De esta manera, aunque sean alcanzables dos estados A, B tales que $A \subset B$, no necesitamos necesariamente (más correcto sería decir no podemos) incluir todos los estados C con $A \subset C \subset B$, puesto que los mismos pueden perfectamente no ser alcanzables. Por otra parte, en caso de que alguno de ellos lo sea, la probabilidad con la que se alcance el mismo es completamente independiente de aquéllas con las que se alcanzan los estados A y B.

En segundo lugar, los términos P_{ij} que representan las continuaciones tras una misma acción, pero para diferentes valores de i, es decir, bajo distintas componentes de una misma elección interna, no necesitan ser iguales.

Veamos al respecto el siguiente

Ejemplo 6.3 Consideremos el siguiente proceso

$$P_1 = [\frac{1}{2}]a \rightarrow STOP \sqcap [\frac{1}{2}]([\frac{1}{2}]a \rightarrow b \rightarrow STOP \Box [\frac{1}{2}]b \rightarrow STOP)$$

Observamos que tiene dos estados, $\{a\}$ y $\{a, b\}$, cuya probabilidad de ser alcanzados $\frac{1}{2}$. Entonces, cualquier proceso que sea equivalente al mismo, debe tener asímismo estos dos estados, y si pretendemos que la continuación del mismo después de ejecutar la acción a sea la misma en cualquiera de los dos estados, la forma de ese proceso deberá ser

$$P_2 = [\frac{1}{2}]a \rightarrow P \sqcap [\frac{1}{2}]([\frac{1}{2}]a \rightarrow P \Box [\frac{1}{2}]b \rightarrow Q$$

Ahora bien, al aplicar a este proceso test $t = [0]a.b.\omega + [1]b.0$, obtenemos

$$\begin{aligned} P_1 \mid t &= 0 \\ P_2 \mid t &= \tfrac{1}{2} \cdot P \mid b.\omega \end{aligned}$$

Ello es así, puesto que en P_1, si se alcanza el estado donde puede ejecutarse la acción b, por la forma del test t será obligatorio ejecutar dicha acción y continuar, lo que llevará a un fracaso; y si se alcanza el estado $\{a\}$, tras aceptar la acción, no podrá ejecutarse la acción b, puesto que no existe. Por el contrario, en P_2 tenemos una probabilidad $\frac{1}{2}$ de alcanzar el estado $\{a\}$, tras lo cual el proceso se comporta como P. Por lo tanto, si queremos que P_1 sea equivalente a P_2, deberíamos tener $P \mid t = 0$.

Ahora bien, si consideramos el test $t' = a.b.\omega$, tenemos que

$$\begin{aligned} P_1 \mid t' &= \tfrac{1}{2} \\ P_2 \mid t' &= P \mid b.\omega = 0 \end{aligned}$$

Puesto que ello no es posible, queda probado que no puede existir ningún proceso con las características que habíamos impuesto. □

Por comodidad, escribiremos usualmente las formas normales con la siguiente notación alternativa:

$$P = \bigsqcap_{A \in \mathcal{A}} [p_A] \Box_{a \in A} [\frac{1}{|A|}]a \to P_{a,A}$$

donde los $\mathcal{A}$ son subconjuntos finitos de $\mathcal{P}(\Sigma)$. Cuando en particular aparezcan elecciones externas generalizadas entre un conjunto de acciones vacío, notaremos al correspondiente subtérmino por $STOP$.

Ejemplo 6.4 Los siguientes procesos están todos ellos en Forma Normal

$$[\tfrac{1}{2}]a \to STOP \sqcap [\tfrac{1}{2}]STOP$$
$$[\tfrac{1}{2}]a \to b \to STOP \sqcap [\tfrac{1}{2}]([\tfrac{1}{2}]a \to STOP \Box [\tfrac{1}{2}]b \to STOP)$$
$$a \to ([\tfrac{1}{2}]b \to STOP \sqcap [\tfrac{1}{2}]STOP)$$

Por el contrario, los siguientes no están en Forma Normal.

$$[\tfrac{1}{2}]a \to STOP \sqcap [\tfrac{1}{2}]a \to b \to STOP$$
$$[\tfrac{1}{2}]a \to STOP \Box [\tfrac{1}{2}]a \to b \to STOP$$
$$[\tfrac{1}{2}]a \to STOP \Box [\tfrac{1}{2}]STOP$$
$$[\tfrac{1}{2}]([\tfrac{1}{2}]a \to b \to STOP \sqcap [\tfrac{1}{2}]b \to STOP) \Box [\tfrac{1}{2}]([\tfrac{1}{2}]a \to STOP \sqcap [\tfrac{1}{2}]b \to STOP)$$

□

Veamos ahora la primera propiedad anunciada de las formas normales.

Lema 6.1 Cualquier término $P \in FPCSP$ puede ser transformado, utilizando el Sistema de Axiomas, en otro término equivalente en Forma Normal.

Demostración: Por inducción sobre la profundidad del término.

- Si $P = STOP$ entonces está en Forma Normal, sin más que considerar el término correspondiente a $\mathcal{A} = \{\emptyset\}$.
- Si $P = a \rightarrow P'$, por hipótesis de inducción P' puede ser transformado en su Forma Normal $\tilde{P}'$, por lo que tomando $\mathcal{A} = \{\{a\}\}$ y $P_{a,\{a\}} = \tilde{P}'$ obtenemos la Forma Normal de P.
- Si $P = [p]P_1 \sqcap [1-p]P_2$, tenemos que, por hipótesis de inducción se cumple

$$P_1 \equiv \bigsqcap_{A \in \mathcal{A}} [p_A] \Box_{a \in A} [\tfrac{1}{|A|}]a \rightarrow P_{a,A}$$

$$P_2 \equiv \bigsqcap_{B \in \mathcal{B}} [p_B] \Box_{b \in B} [\tfrac{1}{|B|}]b \rightarrow Q_{b,B}$$

donde $\mathcal{A}$ y $\mathcal{B}$ son subconjuntos de $\mathcal{P}(\Sigma)$.

Entonces tenemos que

$$P \equiv \bigsqcap_{C \in \mathcal{C}} [p_C] \Box_{c \in C} [\frac{1}{|C|}]c \rightarrow R_{c,C}$$

donde $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$ y

$$\begin{array}{ll} C = A \in \mathcal{A} \wedge C \notin \mathcal{B} & \Rightarrow\ p_C = p \cdot p_A \ \wedge\ \forall c \in C:\ R_{c,C} = P_{a,A} \\ C = B \in \mathcal{B} \wedge C \notin \mathcal{A} & \Rightarrow\ p_C = (1-p) \cdot p_B \ \wedge\ \forall c \in C:\ R_{c,C} = Q_{b,B} \\ C = A \in \mathcal{A} \wedge C = B \in \mathcal{B} & \Rightarrow\ p_C = p \cdot p_A + (1-p) \cdot p_b \\ & \wedge\ \forall c \in C:\ R_{c,C} = [p]P_{c,A} \sqcap [1-p]P_{c,B} \end{array}$$

con lo que en los dos primeros casos obtenemos una expresión en forma normal, mientras que en el último podemos aplicar la hipótesis de inducción a los términos $P_{c,A}$ y $P_{c,B}$ para obtener la forma normal buscada.

- Si $P = [p]P_1 \square [1-p]P_2$, tenemos de nuevo por hipótesis de inducción

$$P_1 \equiv \bigsqcap_{A \in \mathcal{A}} [p_A] \bigsqcup_{a \in A} [\tfrac{1}{|A|}]a \to P_{a,A}$$

$$P_2 \equiv \bigsqcap_{B \in \mathcal{B}} [p_B] \bigsqcup_{b \in B} [\tfrac{1}{|B|}]b \to Q_{b,B}$$

donde $\mathcal{A}$ y $\mathcal{B}$ son subconjuntos de $\mathcal{P}(\Sigma)$.

Entonces, aplicando el axioma A78 obtenemos

$$P \equiv \bigsqcap_{\substack{A \in \mathcal{A} \\ B \in \mathcal{B}}} [p_A \cdot p_B] \bigsqcup_{c \in A \cup B} [\frac{1}{|A \cup B|}]c \to R_{c,A,B}$$

donde, como sucedía en el caso anterior, los subtérminos $R_{c,A,B}$ son los términos en forma normal adecuados en cada caso, para la obtención de los cuales será necesario en ocasiones volver a utilizar la hipótesis de inducción.

Ahora bien, en general todavía no habremos llegado a la forma normal deseada, por poder aparecer en el término anterior distintos pares de conjuntos (A, B) con $A \in \mathcal{A}$, $B \in \mathcal{B}$ cuya unión sea igual a un mismo conjunto C: $A \cup B = C$. En tal caso hemos de aplicar el axioma $A6'$, para obtener

$$P = \bigsqcap_{C \in \mathcal{C}} [p_C] \bigsqcup_{c \in C} [\frac{1}{|C|}]c \to \underline{(\bigsqcap_{A \cup B = C} [\frac{p_A \cdot p_B}{p_C}] R_{c,A,B})}$$

siendo

$$\mathcal{C} = \{A \cup B \;/\; A \in \mathcal{A} \wedge B \in \mathcal{B}\}$$

$$p_C = \sum_{\substack{A \in \mathcal{A}, B \in \mathcal{B} \\ A \cup B = C}} p_A \cdot p_B$$

$$R_{c,A,B} = \begin{cases} P_{c,A} & \text{Si } c \in A \wedge c \notin B \\ Q_{c,B} & \text{Si } c \in B \wedge c \notin A \\ [p]P_{c,A} \sqcap [1-p]P_{c,B} & \text{Si } c \in A \cap B \end{cases}$$

□

Lema 6.2 Si tenemos dos formas normales distintas N_1 y N_2 entonces existirá un $t \in PTEST$ tal que $N_1 \mid t \neq N_2 \mid t$.

Demostración: Tenemos que

$$N_1 = \bigsqcap_{A\in\mathcal{A}} [p_A] \bigsqcup_{a\in A} [\tfrac{1}{m}]a \to P_{a,A}$$

$$N_2 = \bigsqcap_{B\in\mathcal{B}} [p_B] \bigsqcup_{b\in B} [\tfrac{1}{m}]b \to Q_{b,B}$$

donde $\mathcal{A}$ y $\mathcal{B}$ son subconjuntos de $\mathcal{P}(\Sigma)$. Entonces nos encontramos en alguno de los siguientes casos:

- Caso 1: Los conjuntos $\mathcal{A}$ y $\mathcal{B}$ son distintos. Supondremos entonces, sin pérdida de generalidad, que existe un conjunto A tal que $A \in \mathcal{A}$ y $A \notin \mathcal{B}$.

  Dado el anterior conjunto A, sabemos que mediante una combinación de diversos tests podemos calcular $p(N_1, A)$ que tendrá un valor positivo, digamos que p_A . Pero con la misma combinación de tests se calcula $p(N_2, A)$ cuyo valor es 0. Por lo cual se concluye que alguno de los tests que aparecen en la combinación anterior debe darnos valores distintos para N_1 y para N_2.

- Caso 2. Los procesos N_1 y N_2 tienen los mismos estados, pero con distinta probabilidad de ser alcanzados.

  En este caso se sigue el mismo razonamiento que en el caso anterior, pues existe un A, tal que $p(N_1, A) \neq p(N_2, A)$.

- Caso 3. Los procesos N_1 y N_2 tienen los mismos estados, con la misma probabilidad de alcanzarlos.

  En este caso, si $N_1 \neq N_2$, ello debe ser así por existir alguna diferencia entre las continuaciones de ambos procesos: es decir, para algún (a, A) se debe tener $P_{a,A} \neq Q_{a,A}$. Por hipótesis de inducción, encontraremos algún t tal que

  $$P_{a,A} \mid t \neq Q_{a,A} \mid t$$

  Entonces como

  $$N_1/(a, A) = P_{a,A}$$
  $$N_2/(a, A) = Q_{a,A}$$

  y la operación $/(a, A)$ viene definida por

  $$P/(a, A) \mid t = \frac{|A| \cdot (P \mid t^{(a,A-\{a\})} - \sum_{C\subset A-\{a\}} -1^{|C|}(P \mid t^{(a,C)}))}{p(P, A)}$$

como tenemos que $p(N_1, A) = p(N_2, A)$, se obtiene

$$(N_1 \mid t^{(a,A-\{a\})} - \sum_{C \subset A-\{a\}} -1^{|C|} \cdot (N_1 \mid t^{(a,C)}) \neq$$

$$(N_2 \mid t^{(a,A-\{a\})} - \sum_{C \subset A-\{a\}} -1|C| \cdot (N_2 \mid t^{(a,C)})$$

De donde finalmente deducimos que para algún B

$$N_1 \mid t^{(a,B)} \neq N_2 \mid t^{(a,B)}$$

□

Teorema 6.2 (Completitud)
Para todo par de términos P, $Q \in PROC$ tenemos que si $\forall t \in PTEST\ P \mid t = Q \mid t$ entonces $\vdash P \equiv Q$.

Demostración: Consecuencia inmediata de los dos lemas anteriores. □

6.2 Otros operadores

En esta sección generalizaremos el sistema de prueba al resto de operaciones de nuestro lenguaje PCSP, excluída de momento la recursión. Comprobaremos que se trata de operadores derivados, que por lo tanto pueden ser reducidos a los operadores básicos anteriores. Para ello deberemos añadir una serie de axiomas para cada nueva operación. Como quiera que los axiomas nos muestran cómo eliminar los nuevos operadores, bastará probar la corrección de los mismos para concluir la corrección y completitud del sistema global.

6.2.1 Composición paralela

Puesto que ya sabemos que todo proceso finito construído con los nuevos operadores básicos se puede poner en forma normal, bastará que veamos axiomas que nos permitan eliminar el operador paralelo al aplicarlo sobre dos formas normales.

- En primer lugar, el axioma $P1$ nos indica que la composición paralela distribuye sobre la elección interna.

  P1] $[p]P\|_A[1-p]([q]Q \sqcap [1-q]R) \equiv$
  $[q]([p]P\|_A[1-p]Q) \sqcap [1-q]([p]P\|_A[1-p]R)$

- Por otra parte, si el proceso $STOP$ está en paralelo con una elección externa generalizada, sólo se podrán ejecutar de esta elección externa aquellas acciones que no necesiten sincronizarse.

  P2] $[p] \underset{b\in B}{\Box} b \to P_b\|_A[1-p]STOP \equiv \underset{b\in B-A}{\Box} b \to P_b$

- La composición paralela es conmutativa, lo que nos permite no duplicar los axiomas anteriores para cubrir el caso del otro argumento.

  P3] $[p]P\|_A[1-p]Q \equiv [1-p]Q\|_A[p]P$

- La composición paralela entre dos procesos que están en elección externa generalizada, se convierte en una elección externa entre todas las acciones que pueden ser ejecutadas en primer lugar por los componentes, con la salvedad de que las acciones que pertenecen al conjunto de sincronización deben ser ejecutada simultáneamente por ambos procesos. Posteriormente, el proceso continúa por medio de la composición paralela de las continuaciones. Como de costumbre, este axioma será referenciado normalmente como la ley de expansión.

  P4] $[p](\underset{b\in B}{\Box} b \to P_b\|_A[1-p]\underset{c\in C}{\Box} c \to Q_c) \equiv$

  $[\frac{1}{2}] \quad \underset{a\in A\cap B\cap C}{\Box} a \to ([p]P_a\|_A[1-p]Q_a)$

  $\Box \quad [\frac{1}{2}]([p] \quad \underset{b\in B-A}{\Box} b \to ([p]P_b\|_A[1-p]\underset{c\in C}{\Box} c \to Q_c)$

  $\Box \quad [1-p] \quad \underset{c\in C-A}{\Box} c \to ([p]\underset{b\in B}{\Box} b \to P_b\|_A[1-p]Q_c))$

Proposición 6.4 Los axiomas P1 a P4 son correctos.

Demostración: La corrección de los axiomas P1 y P2 se sigue de forma similar a la de los axiomas análogos correspondientes al caso de la elección externa (A5 y A9), por lo que no consideramos necesario ahondar en más detalles. Por otra

parte la corrección de P3 es trivial, ya que los pasos de tests por parte del operador composición paralela no tienen en cuenta el orden de los operandos. Finalmente, para P4 tenemos

$$
\begin{aligned}
&[p](\underset{b\in B}{\Box} b \to P_b)\|_A[1-p](\underset{c\in C}{\Box} c \to Q_c) \mid t \\
&= \sum_{a\in Ft(t)} s(a,D,t)\cdot Cont(\|_A,t,B,C,a) \\
&= \sum_{a\in Ft(t)} s(a,D,t)\cdot \\
&\quad \begin{cases}
([p]P_a\|_A[1-p]\underset{c\in C}{\Box} c \to P_c) \mid t/a & \text{Si } a\in B-C-A \\
([p]\underset{b\in B}{\Box} b \to P_b\|_A[1-p]Q_a) \mid t/a & \text{Si } a\in C-B-A \\
p\cdot[([p]P_a\|_A[1-p]\underset{c\in C}{\Box} c \to P_c) \mid t/a]+ & \\
(1-p)\cdot[([p]\underset{b\in B}{\Box} b \to P_b\|_A[1-p]Q_a) \mid t/a] & \text{Si } a\in (B\cap C)-A \\
([p]P_a\|_A[1-p]Q_a) \mid t/a & \text{Si } a\in A\cap B\cap C \\
0 & \text{En otro caso}
\end{cases}
\end{aligned}
$$

donde $D = (B \cup C - A) \cup (A \cap B \cap C)$.

Y, por la definición del paso de tests, éste es el mismo resultado que el obtenido al aplicar un test t al lado derecho de la equivalencia enunciada por el axioma. □

Por tanto, aplicando los axiomas anteriores, podemos 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 siempre aplicados a pares de términos más simples que los originales. Por lo que, reiterando la aplicación de dichos axiomas, podremos conseguir finalmente que los operadores paralelos desaparezcan por completo.

6.2.2 Ocultación

Nos limitaremos de nuevo a enunciar las leyes precisas para eliminar la aplicación del operador de ocultación a una forma normal.

En primer lugar, los siguientes axiomas cubren los casos en los que el operador de ocultación sobre el proceso STOP o sobre un proceso en el que sólo aparecen elecciones externas generalizadas.

- **H1]** $STOP\backslash(a,q) \equiv STOP$
- **H2]** $(\underset{b\in B}{\Box} b \rightarrow P_b\backslash(a,q)) \equiv$

$$\begin{cases} \underset{b\in B}{\Box} b \rightarrow (P_b\backslash(a,q)) & \text{Si } a \notin B \\ [q]P_a\backslash(a,q) \sqcap [1-q]([0]P_a\backslash(a,q) \,\Box\, [1] \underset{b\in B-\{a\}}{\Box} b \rightarrow (P_b\backslash(a,q)) & \text{Si } a \in B \end{cases}$$

Estos dos axiomas son correctos, como queda reflejado en la siguiente

Proposición 6.5 Los axiomas H1 y H2 son correctos.

Demostración: El axioma H1 es trivialmente correcto. Para probar la corrección de H2, tomemos $t \in PTEST$, y supongamos en primer lugar que $a \notin B$; entonces

$$\begin{aligned} (\underset{b\in B}{\Box} b \rightarrow P_b\backslash(a,q)) \mid t &= \sum_{b\in B} s(b,B,t)\cdot(P/(b,A)\backslash(a,q)) \mid t/b \\ &= \underset{b\in B}{\Box} b \rightarrow (P_b\backslash(a,q)) \mid t \end{aligned}$$

Mientras que si $a \in B$, se tiene

$$\begin{aligned} &\underset{b\in B}{\Box} b \rightarrow P_b\backslash(a,q)) \mid t = \\ &\sum_{a\notin A\subset\Sigma} \sum_{b\in A} s(b,A,t) \cdot (\, q\cdot p(P_a\backslash(a,q),A)\cdot P_a\backslash(a,q)/(b,A) \mid t/b) + \\ &\quad (1-q)\cdot \sum_{B\cup C=A\cup\{a\}} p(P_a\backslash(a,q),C)\cdot \begin{cases} P_b\backslash(a,q) \mid t/b & \text{Si } b\in B \\ P_a\backslash(a,q)/(b,C) \mid t/b & \text{Si } b\notin B \end{cases} \,) \end{aligned}$$

Ahora bien, como

$$\begin{aligned} &P_a\backslash(a,q) \mid t = \\ &\quad \sum_{a\notin A\subset\Sigma} \sum_{b\in A} s(b,A,t)\cdot p(P_a\backslash(a,q),A)\cdot P_b/(a,B)\backslash(a,q)/(b,A) \mid t/b \end{aligned}$$

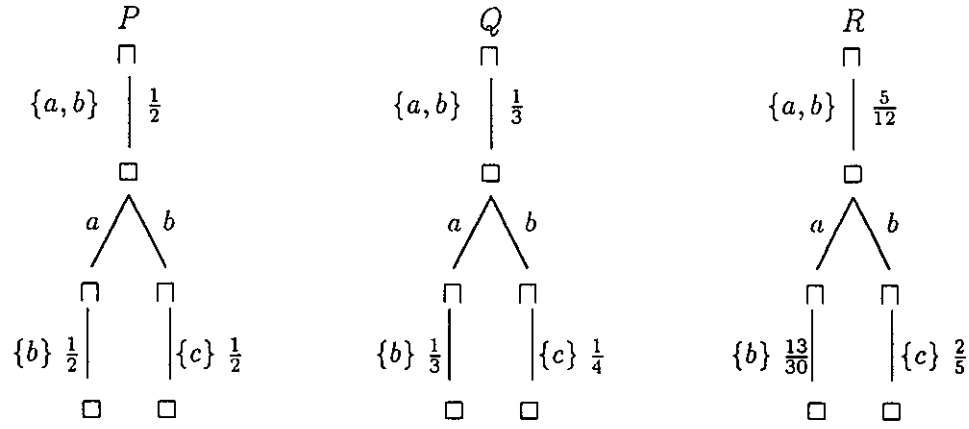


Figura 8: Contraejemplo de distributividad

tenemos que

$$[0]P_a\backslash(a,q)\Box[1]\Box_{b\in B-\{a\}} b \to (P_b\backslash(a,q)) \mid t =$$

$$\sum_{a\notin A}\sum_{B\cup C=A\cup\{a\}}\sum_{b\in A} s(b,A,t)\cdot p(P_a\backslash(a,q),C)\cdot \begin{cases} P_b\backslash(a,q) \mid t/b & \text{Si } b\in B \\ P_a\backslash(a,q)/(b,C) \mid t/b & \text{Si } b\notin B \end{cases}$$

De donde se obtiene el resultado deseado. □

De cara a eliminar el operador de ocultación cuando aparece aplicado sobre una elección interna, parece claro que necesitamos alguna ley de distributividad. La primera candidata considerada fue, por razones obvias, la correspondiente ley general, que de hecho es válida en el lenguaje CSP sin probabilidades. La misma podría ser formulada como sigue

$$([r]P \sqcap [1-r]Q)\backslash(a,q) \equiv [r]P\backslash(a,q) \sqcap [1-r]Q\backslash(a,q)$$

Lamentablemente, dicho axioma no es válido en PCSP, como se puede comprobar por medio del siguiente

Ejemplo 6.5 Sean P, Q y $R = [\frac{1}{2}]P \sqcap [\frac{1}{2}]Q$ los procesos representados en la figura 8, en la que hemos prescindido de aquellas partes de los procesos en las que no estamos interesados. Tenemos que los mismos verifican

$$\begin{array}{llll} p(P\backslash(a,q),\{b\}) = 1/4\cdot q & & p(R\backslash(a,q)/(b,\{b\}),\{c\}) = 1/8\cdot(1-q) \\ p(Q\backslash(a,q),\{b\}) = 1/9\cdot q & & p(Q\backslash(a,q)/(b,\{b\}),\{c\}) = 1/36\cdot(1-q) \\ p(R\backslash(a,q),\{b\}) = 13/72\cdot q & & p(R\backslash(a,q)/(b,\{b\}),\{c\}) = 13/180\cdot(1-q) \end{array}$$

Tenemos entonces que

$$p(R\backslash(a,q),\{b\})) = \frac{1}{2}\cdot p(P\backslash(a,q),\{b\}) + \frac{1}{2}\cdot p(Q\backslash(a,q),\{b\})$$

y sin embargo

$$p(R\backslash(a,q)/(b,\{b\}),\{c\}) \neq$$
$$\frac{1}{2}\cdot p(P\backslash(a,q)/(b,\{b\}),\{c\}) + \frac{1}{2}\cdot p(Q\backslash(a,q)/(b,\{b\}),\{c\})$$

□

La razón por la que ello es así, consiste en que de cara a que cada uno de los procesos P, Q y R alcancen el estado $\{c\}$ tras ejecutar la acción b en el estado $\{b\}$, es necesario que la acción oculta a no sea ejecutada. Pero a la hora de calcular el estado alcanzado cuando tal cosa sucede (que la acción ocultada no se ejecute), y en particular la probabilidad de que el estado finalmente alcanzado sea $\{b\}$, hemos de calcular la probabilidad de que en dichos procesos se alcancen, tras la ejecución de la citada acción a ocultada, estados contenidos en dicho conjunto $\{b\}$. Una vez calculadas dichas probabilidades, se multiplican por la probabilidad de alcanzar el estado $\{c\}$ tras la ejecución de la acción b. Pero cuando multiplicamos las dos probabilidades en cuestión, correspondientes al proceso R, resulta que estamos multiplicando dos factores que dependen simultáneamente de P y de Q, por lo que en el producto nos aparecen (como es deseable) sumandos con "dependencias cruzadas de P y de Q. Dichos sumandos no aparecen cuando en el lado derecho del axioma sumamos ponderadamente los valores obtenidos de P y de Q por separado, y son por tanto los culpables de que la igualdad exigida por la deseable regla de distributividad no se tenga en general.

Ahora bien, el citado problema desaparece cuando P y Q no pueden alcanzar inicialmente estados iguales, pues aún cuando al ocultar una acción podamos generar entradas iguales (por ejemplo, al ocultar la acción a, tanto el estado $\{b\}$ como el estado $\{a,b\}$ generan el estado $\{b\}$), tenemos que, al computar las probabilidades con las que el proceso resultado de la elección interna R alcanza un cierto estado, los sumandos que aparecen se refieren a estados originales de los procesos P y Q dados, y como estamos suponiendo que tales estados son siempre distintos, resulta que el fenómeno indeseado de cruce de dependencias al que antes hacíamos referencia, desaparece.

Pero, afortunadamente, las ocurrencias del operador de elección interna que aparecen en una forma normal siempre cumplen, por definición, dicha condición; por lo que en caso de ser correcta la ley de distributividad restringida a tales situaciones, no tendríamos entonces problema alguno de cara a eliminar la aplicación del operador de ocultación sobre una forma normal.

Y en efecto ello es así, teniéndose en concreto el siguiente axioma:

- **H3]** $(\bigsqcap_{B\subseteq\Sigma} [p_B] \Box_{b\in B} b \rightarrow P_{b,B})\backslash(a,q) \equiv \bigsqcap_{B\subseteq\Sigma} [p_B] \Box_{b\in B} (b \rightarrow P_{b,B}\backslash(a,q))$

que resulta ser correcto, como indica la siguiente

Proposición 6.6 El axioma $H3$ es correcto.

Demostración: Sea $t \in PTEST$; si tomamos $R = \bigsqcap_{B\subseteq\Sigma} [p_B] \Box_{b\in B} b \rightarrow P_{b,B}$, aplicando la definición de paso de tests, para el proceso en el lado izquierdo del axioma obtenemos

$$
\begin{aligned}
& R\backslash(a,q) \mid t \\
&= \sum_{a\notin B\subset\Sigma} \sum_{b\in B} s(b,B,t)\cdot p(R\backslash(a,q),B)\cdot R\backslash(a,q)/(b,B) \mid t/b \\
&= \sum_{a\notin B\subset\Sigma} \sum_{b\in B} s(b,B,t)\cdot p(R,B)\cdot R/(b,B)\backslash(a,q) \mid t/b \; + \\
&\quad \sum_{a\in A\subseteq\Sigma} p(R,A)\cdot [q\cdot p(R/(a,A)\backslash(a,q),B)\cdot (R/(a,A)\backslash(a,q) \mid t/b) \; + \\
&\quad (1-q)\cdot \sum_{A-\{a\}\cup C=B} p(R/(a,A)\backslash(a,q),C)\cdot \\
&\quad \begin{cases} R/(b,B)\backslash(a,q) \mid t/b & \text{Si } b\in B \\ R/(a,A)\backslash(a,q)/(b,C) \mid t/b & \text{Si } b\notin B \end{cases}] \\
&= \sum_{a\notin B\subset\Sigma} \sum_{b\in B} s(b,B,t)\cdot p_B\cdot P_{b,B}\backslash(a,q) \mid t/b \; + \\
&\quad \sum_{a\in A\subseteq\Sigma} p_A\cdot [q\cdot p(P_{a,A}\backslash(a,q),B)\cdot (P_{a,A}\backslash(a,q) \mid t/b) \; + \\
&\quad (1-q)\cdot \sum_{A-\{a\}\cup C=B} p(P_{a,A}\backslash(a,q),C)\cdot \begin{cases} P_{b,B}\backslash(a,q) \mid t/b & \text{Si } b\in B \\ P_{a,A}\backslash(a,q)/(b,C) \mid t/b & \text{Si } b\notin B \end{cases}]
\end{aligned}
$$

De la misma forma, aplicando la definición de elección interna, obtenemos para el proceso que aparece en la parte derecha

$$(\bigsqcap_{B\subseteq\Sigma}[p_B]\,\Box_{b\in B}(b\rightarrow P_b\backslash(a,q))\mid t = \sum_{B\subseteq\Sigma}p_B\cdot(\Box_{b\in B}(b\rightarrow P_b\backslash(a,q))\mid t)$$

lo que, aplicando el axioma $H2$, nos conduce a

$$\sum_{a\notin B\subseteq\Sigma}p_B\cdot\Box_{b\in B}b\rightarrow(P_{b,B}\backslash(a,q))\mid t +$$

$$\sum_{a\in B\subseteq\Sigma}p_B\cdot[q]P_{a,A}\backslash(a,q)\sqcap[1-q]([0]P_{a,A}\backslash(a,q)\,\square\,[1]\Box_{b\in B-\{a\}}b\rightarrow(P_{b,B}\backslash(a,q))\mid t$$

$$=\sum_{a\notin B\subseteq\Sigma}\sum_{b\in B}s(b,B,t)\cdot p_B\cdot(P_{b,B}\backslash(a,q))\mid t/b +$$

$$\sum_{a\in A\subseteq\Sigma}p_A\cdot$$

$$(q\cdot\sum_{a\notin B\subseteq\Sigma}\sum_{b\in B}s(b,B,t)\cdot p(P_{a,A}\backslash(a,q),B)\cdot(P_{a,A}\backslash(a,q)/(b,B)\mid t/b) +$$

$$(1-q)\cdot\sum_{C\subseteq\Sigma-\{a\}}\sum_{b\in A-\{a\}\cup C}s(b,A-\{a\}\cup C,t)\cdot p(P_{a,A}\backslash(a,q),C)\cdot$$

$$\begin{cases}P_{b,A}\backslash(a,q)\mid t/b & \text{Si } b\in A\\ P_{a,A}\backslash(a,q)/(b,C)\mid t/b & \text{Si } b\notin A\end{cases})$$

En donde, si en el último sumando, denotamos por B al estado alcanzado $A-\{a\}\cup C$, y procedemos a una redistribución adecuada de los sumandos, llegamos al mismo resultado que el obtenido para el lado izquierdo del axioma, con lo que queda probada la corrección del mismo. □

6.3 PCSP recursivo

Los términos recursivos del lenguaje se tratarán, como es usual en el marco de las álgebras de procesos, mediante su aproximación por medio de *términos finitos*, teniendo en cuenta el hecho de que la semántica de aquellos viene dada por el límite de la cadena que forman los valores semánticos de sus aproximaciones finitas.

Las citadas aproximaciones finitas han de tener como punto de partida un proceso sintáctico cuya semántica sea el menor elemento del dominio semántico. En nuestro caso se tratará del proceso que hemos denominado DIV. Entonces, si

P es un proceso definido mediante la expresión recursiva $\mu X.P[X]$ construiremos la cadena de aproximaciones $(P^n)_{n\in\mathbb{N}}$, siendo cada P^n el proceso definido por

$$P^0 = DIV$$
$$P^{n+1} = P[P^n/X]$$

Es obvio entonces que (a nivel sintáctico), cada P^n es un proceso finito de PCSP, de forma que el estudio que hemos realizado en las secciones previas es válido para ellos.

Los elementos de la cadena de aproximaciones de un proceso definido recursivamente, forman lo que llamaremos su conjunto de aproximaciones finitas.

Definición 6.3 (Aproximaciones finitas)
Sea P un proceso, el conjunto de *aproximaciones finitas* de P, que denotaremos por $APX(P)$, se define mediante

$$APX(P) = \{P^n : n \in \mathbb{N}\}$$

□

De la necesidad de caracterizar mediante sus aproximaciones a los procesos recursivos, nos surge una nueva necesidad, que consiste en pasar a axiomatizar la relación de orden $\sqsubseteq$ entre procesos, en lugar de la equivalencia entre los mismos. Dichos axiomas pasan a ser la componente básica del sistema axiomático, quedando la equivalencia como noción derivada a través de la propiedad antisimétrica de la relación (de orden) con la que se trabaja.

6.3.1 Nuevos axiomas y reglas

En esta subsección enumeraremos los axiomas y reglas que pasan a configurar nuestro nuevo sistema axiomático. Para ello, partimos del sistema presentado en las secciones 6.1 y 6.2, sustituyendo las reglas $E1$, $E2$ y $E3$ que cubren las propiedades de la relación de equivalencia y las reglas $C1$, $C2$ y $C3$ que asertan sobre la congruencia de dicha relación respecto de los operadores del lenguaje, por

las análogas correspondientes pero referidas a la relación de orden entre procesos, que son en concreto los siguientes:

C1'] $\dfrac{P \sqsubseteq Q}{a \rightarrow P \sqsubseteq a \rightarrow Q}$ C2'] $\dfrac{P \sqsubseteq Q \ \wedge \ P' \sqsubseteq Q'}{(P \boxplus P') \sqsubseteq (Q \boxplus Q')}$

C3'] $\dfrac{P \sqsubseteq Q \ \wedge \ P' \sqsubseteq Q'}{([p]P \sqcap [1-p]P') \sqsubseteq ([p]Q \sqcap [1-p]Q')}$ X1] $\dfrac{P \sqsubseteq Q}{P \sqsubseteq [p]P \sqcap [1-p]Q \sqsubseteq Q}$

R1] $\dfrac{}{P[\mu\xi.P/\xi] \sqsubseteq \mu\xi.P}$ R2] $\dfrac{\forall Q \in APX(P) : Q \sqsubseteq R}{P \sqsubseteq R}$

R3] $\dfrac{\forall n \in \mathbb{N} : [\frac{n-1}{n}]P \sqcap [\frac{1}{n}]DIV \sqsubseteq R}{P \sqsubseteq R}$ O1'] $\dfrac{P \sqsubseteq Q \sqsubseteq P}{P \equiv Q}$

O2'] $\dfrac{P \equiv Q}{P \sqsubseteq Q \sqsubseteq P}$ O3'] $\dfrac{P \sqsubseteq Q \sqsubseteq R}{P \sqsubseteq R}$

Análogamente a lo que sucedía en el caso finito, la prueba de que la relación $\sqsubseteq$ es de orden, y de que es preservada por todos los operadores del lenguaje, es trivial, dada la definición composicional de los operadores, por lo que para este caso tampoco nos detendremos en la prueba de corrección de dichas reglas. En lo que respecta a la regla $X1$, que indica que si dos procesos están relacionados, entonces la elección interna entre ellos ocupa una posición intermedia, surge de manera inmediata de la definición de la operación de elección interna. Y respecto a las reglas que hacen referencia a la recursión, puesto que tenemos definido

$$P[\mu\xi.P/\xi] \mid t \ = \ \mu\xi.P \mid t$$

resulta evidente $R1$. Análogamente, de la definición de paso de un test por parte del proceso $\mu\xi.P$ se deriva la corrección de la regla $R2$, mientras que de la definición de la elección interna se sigue la corrección de $R3$. Posponemos la justificación de la presencia de este último axioma, que representa la novedad más notable frente a los que aparecen en la versión no probabilística del lenguaje, hasta el momento concreto en que debamos utilizarlo en la prueba de la completitud del sistema.

D1]	$DIV \sqsubseteq P$
D2]	$[p]P \sqcap [1-p]DIV \sqsubseteq P$
D3]	$[p]P \Box [1-p]DIV \equiv DIV$
D4]	$[p]P \|\|_A [1-p]DIV \equiv DIV$
D5]	$DIV \backslash (a,q) \equiv DIV$

Tabla 7: Axiomas para divergencia

Por otra parte, tenemos en la tabla 7 los axiomas que nos indican como se comporta el proceso DIV frente a cada uno de los operadores básicos del lenguaje. De hecho, estrictamente hablando, sólo serían necesarios los axiomas $D1$ y $D2$ pues los otros son derivables a partir de los restantes axiomas que nos definen el comportamiento de cada una de las operaciones involucradas en ellos. No obstante, los hemos incluido para que quede explícito el caracter estricto de dichas operaciones respecto del proceso DIV.

La corrección de todos los axiomas anteriores es igualmente obvia, teniendo en cuenta la definición del proceso DIV.

Una modificación adicional, para cubrir el caso del proceso $STOP$, que en el sistema para el caso finito resultaba innecesaria debido a la presencia de la regla $E1$, consiste en la adición del siguiente axioma:

A0] $STOP \equiv STOP$

Del mismo, junto con $DIV \equiv DIV$ que es derivable de $D1$ y $O1'$, se puede inferir $P \equiv P$, primero para procesos finitos, y después vía $R1$ y $R2$ para los infinitos.

En suma, en las tablas 8 y 9 presentamos completo el sistema de axiomas y reglas para PCSP.

Corolario 6.1 (Corrección)
El sistema axiomático es correcto respecto a la equivalencia y al orden inducido por la semántica de pruebas. Es decir:

$$\vdash P \sqsubseteq Q \Rightarrow \forall t \in PTEST \;\; P \mid t \leq Q \mid t$$

A0]	$STOP \equiv STOP$
A1]	$[p]P \sqcap [1-p]P \equiv P$
A2]	$[p]P \sqcap [1-p]Q \equiv [1-p]Q \sqcap [p]P$
A3]	$[p]P \sqcap [q+r]([\frac{q}{q+r}]Q \sqcap [\frac{r}{q+r}]R) \equiv [p+q]([\frac{p}{p+q}]P \sqcap [\frac{q}{p+q}]Q) \sqcap [r]R$
A4]	$[p]P \Box [1-p]Q \equiv [1-p]Q \Box [p]P$
A5]	$[p]P \Box [1-p]STOP \equiv P$
A6′]	$[p](\Box_{a\in A}[\frac{1}{\lvert A\rvert}]a \to P_a) \sqcap [1-p](\Box_{a\in A}[\frac{1}{\lvert A\rvert}]a \to Q_a) \equiv$ $\Box_{a\in A}[\frac{1}{\lvert A\rvert}]a \to ([p]P_a \sqcap [1-p]Q_a)$
A78]	$[p]\Box_{i=1}^{n}[\frac{1}{n}]a_i \to P_i \Box [1-p]\Box_{j=1}^{m}[\frac{1}{m}]b_j \to P_j \equiv \Box_{k=1}^{l}[\frac{1}{l}]c_k \to R_k$
A9]	$[p]P \Box [1-p]([q]Q \sqcap [1-q]R) \equiv$ $[q]([p]P \Box [1-p]Q) \sqcap [1-q]([p]P \Box [1-p]R)$

Tabla 8: Axiomas para operadores básicos

□

Lema 6.3 Sea P el proceso infinito definido recursivamente mediante el término $\mu X.P(X)$, donde en P no aparecen recursiones; entonces para toda aproximación finita $P^n \in APX(\mu X.P(X))$ se tiene $\vdash P^n \sqsubseteq \mu X.P(X)$.

Demostración: Procederemos por inducción respecto de n. En primer lugar tenemos

$$P^0 = DIV \sqsubseteq \mu X.P(X) \quad \text{aplicando el axioma } D1$$

Entonces, para la aproximación $n+1$, puesto que en el proceso P la variable X aparece en un contexto de operadores finitos, es decir distintos de la recursión, podemos probar utilizando el sistema de axiomas que

$$\begin{aligned} P^{n+1} &= P(P^n|X) \\ &\sqsubseteq P(\mu X.P(X)|X) && \text{Por las reglas } C1', C2', C3' \\ &\sqsubseteq \mu X.P(X) && \text{Por la regla } R1 \end{aligned}$$

□

El lema se puede generalizar a los procesos que contengan un número indeterminado de definiciones recursivas, tento independientes como anidadas. Para ello

P1] $[p]P\|_A[1-p]([q]Q \sqcap [1-q]R) \equiv$
$[q]([p]P\|_A[1-p]Q) \sqcap [1-q]([p]P\|_A[1-p]R)$

P2] $[p]\Box_{b\in B} b \to P_b\|_A[1-p]STOP \equiv \Box_{b\in B-A} b \to P_b$

P3] $[p]P\|_A[1-p]Q \equiv [1-p]Q\|_A[p]P$

P4] $[p](\Box_{b\in B} b \to P_b\|_A[1-p]\Box_{c\in C} c \to Q_c) \equiv$

$$[\tfrac{1}{2}] \quad \Box_{a\in A\cap B\cap C} a \to ([p]P_a\|_A[1-p]Q_a)$$

$$\Box \quad [\tfrac{1}{2}]([p] \quad \Box_{b\in B-A} b \to ([p]P_b\|_A[1-p]\Box_{c\in C} c \to Q_c)$$

$$\Box \quad [1-p] \quad \Box_{c\in C-A} c \to ([p]\Box_{b\in B} b \to P_b\|_A[1-p]Q_c))$$

H1] $STOP\backslash(a,q) \equiv STOP$

H2] $(\Box_{b\in B} b \to P_b\backslash(a,q)) \equiv$

$$\begin{cases} \Box_{b\in B} b \to (P_b\backslash(a,q)) & \text{Si } a \notin B \\ [q]P_a\backslash(a,q) \sqcap [1-q]([0]P_a\backslash(a,q)\Box & \\ \quad [1]\Box_{b\in B-\{a\}} b \to (P_b\backslash(a,q)) & \text{Si } a \in B \end{cases}$$

H3] $(\sqcap_{B\subseteq\Sigma}[p_B]\Box_{b\in B} b \to P_b)\backslash(a,q) \equiv \sqcap_{B\subseteq\Sigma}[p_B]\Box_{b\in B}(b \to P_b\backslash(a,q))$

Tabla 9: Axiomas para procesos derivados

se podría proceder de dos formas diferentes: bien *aplanando* las recursiones hasta obtener un sistema recursivo que defina un vector de variables de proceso, por medio de un vector de expresiones conteniendo como variables libres a las mismas; o bien por inducción respecto del número de recursiones anidadas, pues podemos partir de que el razonamiento indicado en el lema es válido cuando tenemos única recursión, y también podemos generalizar dicho razonamiento, utilizando la hipótesis de inducción, al caso en el que tenemos un número n de recursiones anidadas.

6.3.2 Completitud del nuevo Sistema de Axiomas

Nuestro objetivo en esta sección ha sido encontrar un sistema de axiomas que caracterice la equivalencia de pruebas entre los procesos. Ahora bien, el tratamiento de la recursión nos ha obligado a trabajar con una relación de orden, en lugar de con la relación de equivalencia directamente. Cabría esperar, en consecuencia, que de cara a cubrir nuestro objetivo, la axiomatización presentada caracterizara el orden entre procesos inducido por los tests. Esto no ha resultado en realidad así, aunque sin embargo sí que es cierto que la equivalencia que hemos definido axiomáticamente coincide con la equivalencia respecto al paso de tests. Esto es posible, pues recordemos que de hecho, la relación de orden axiomatizada coincide con la relación de orden entre procesos inducida por la semántica denotacional, con lo que la relación de equivalencia axiomatizada coincide a su vez con la denotacional, y, como quedó probado en la sección 5.4, esta relación de equivalencia coincide con la de pruebas, aun cuando las relaciones de orden de partida no sean iguales. De modo que, si probamos la completitud del sistema de axiomas presentado, respecto del orden inducido por la semántica denotacional, queda de inmediato probada la completitud del mismo respecto de la equivalencia inducida por el paso de tests.

Naturalmente, del hecho probado en la sección 5.4 de que las dos relaciones de orden no coinciden, se sigue que tan sólo se tiene que $P \sqsubseteq_D P' \Rightarrow P \sqsubseteq_T P'$ (véase el ejemplo 5.1), de donde tenemos que serían precisos axiomas adicionales de cara a caracterizar el orden de pruebas entre procesos. No obstante, dado que ello no nos ha parecido en principio especialmente interesante, y sin embargo

hemos podido comprobar mediante ejemplos que la tarea no parece en principio en absoluto trivial, hemos decidido posponer el estudio de la cuestión para otra ocasión.

Para probar la completitud del sistema de axiomas presentado respecto de la semántica denotacional original, procederemos de nuevo de la forma habitual, introduciendo las formas normales adecuadas, que son en esta ocasión las siguientes:

Definición 6.4 (Nuevas formas normales)
Diremos que un proceso PCSP está en forma normal cuando tiene la forma

$$\prod_{i=1}^{n} [p_i] \Box_{j=1}^{m_i} [\frac{1}{m_i}] \, a_{ij} \rightarrow P_{ij}$$

donde

$n \geq 0$ (Si $n = 0$ notaremos al término por DIV

$\forall i \in \{1, \ldots, n\} m_i \geq 0$ Si $m_i = 0$ notaremos al subtérmino por $STOP$

$\forall i \;\; p_i > 0$

$\sum_{i=1}^{n} p_i \;\leq\; 1$

$j \neq k \Rightarrow a_{ij} \neq a_{ik}$

$i \neq k \Rightarrow \{a_{ij}\}_{j \in 1,m_i} \neq \{a_{kj}\}_{j \in 1,m_k}$

$\forall i \forall j \, P_{ij}$ está en forma normal

□

En concreto, la única diferencia con las formas normales correspondientes al caso finito radica en el hecho de que las sumas de probabilidades de cada elección interna generalizada, no tienen por qué ser iguales a 1, pudiendo en general ser inferiores. En tal caso, la diferencia hasta 1 corresponderá, como se vió al estudiar la semántica denotacional, a la probabilidad de divergencia en el punto correspondiente.

Al igual que en el caso finito abreviaremos el proceso correspondiente a la elección externa entre cero acciones con $STOP$. Pero ahora, cabe además la posibilidad de que una elección interna sea entre 0 procesos, lo que corresponderá al proceso DIV, que por lo tanto también se encuentra en forma normal. Usaremos

también en ocasiones, la notación conjuntista

$$\prod_{A\in\mathcal{A}} [p_A] \Box_{a\in A} [\frac{1}{|A|}] \, a \rightarrow P_{a,A}$$

e incluso, y aunque sintácticamente ello no sea correcto, omitiremos en ocasiones, por simplificar, las probabilidades de las elecciones externas, pues como ya sabemos las mismas carecen en estos casos de utilidad alguna, escribiendo

$$\prod_{A\in\mathcal{A}} [p_A] \Box_{a\in A} a \rightarrow P_{a,A}$$

Lema 6.4 Cualquier término finito $P \in PCSP$ puede ser transformado utilizando el Sistema de Axiomas, en otro término equivalente en Forma Normal.

Demostración: La demostración es similar a la correspondiente al lenguaje de procesos finitos FPCSP, por lo que omitimos su desarrollo. □

Teorema 6.3 (Completitud)
El sistema de axiomas presentado es completo respecto a la relación de orden $\sqsubseteq$ en que se basa la semántica denotacional original del lenguaje PCSP.

Demostración: Tenemos que comprobar que

$$\forall\, P, Q \in PCSP \;:\; [\![P]\!] \sqsubseteq [\![Q]\!] \Rightarrow P \sqsubseteq Q$$

En el caso de que tanto P como Q sean finitos, seguimos un razonamiento parecido al caso finito. Tenemos entonces que, utilizando el sistema de prueba, P y Q pueden ser transformados en términos en forma normal, por lo que podemos restringirnos al estudio de las formas normales equivalentes. Consideramos, por tanto, P y Q de la forma

$$P = \prod_{A\in\mathcal{A}} [p_A] \Box_{a\in A} a \rightarrow P_{a,A}$$

$$Q = \prod_{B\in\mathcal{B}} [q_B] \Box_{b\in B} b \rightarrow Q_{b,B}$$

donde $\mathcal{A}$ y $\mathcal{B}$ son subconjuntos de $\mathcal{P}(\Sigma)$.

La prueba entonces será por inducción respecto a la complejidad del proceso Q, entendiendo por tal la medida que prima la profundidad de los procesos, y entre

dos procesos con la misma profundidad considera más complejo aquel que tiene más estados.

Por otra parte, podemos suponer $P \neq DIV$, pues para $P = DIV$ el resultado es inmediato aplicando el axioma $D1$. Entonces, de forma similar a lo que sucedía en el caso finito, nos podemos encontrar en alguno de los siguientes casos:

- Caso 1: Los conjuntos $\mathcal{A}$ y $\mathcal{B}$ son distintos. Entonces, como $[\![P]\!] \sqsubseteq [\![Q]\!]$, tenemos que existe un conjunto B tal que $B \in \mathcal{B}$ y $B \notin \mathcal{A}$, teniendo el resto de los estados de Q probabilidad mayor o igual de ser alcanzados que en el proceso P. Pero entonces, en virtud del axioma $A3$, podemos escribir

$$\begin{aligned} P &= [1-q_B]P' \sqcap [q_B]DIV \\ Q &= [1-q_B]Q' \sqcap [q_B] \Box_{b\in B}\, b \rightarrow Q_{b,B} \end{aligned}$$

  donde P' es idéntico a P, pero con las probabilidades iniciales p_A para cada estado A divididas por el factor $1-q_B$; y lo mismo sucede con Q'. Es decir, se tiene

$$\begin{aligned} P' &= \bigsqcap_{A\in\mathcal{A}} [\tfrac{p_A}{1-q_b}] \Box_{a\in A}\, a \rightarrow P_{a,A} \\ Q' &= \bigsqcap_{C\in\mathcal{B}-\{B\}} [\tfrac{q_C}{1-q_B}] \Box_{b\in C}\, b \rightarrow Q_{b,C} \end{aligned}$$

  Entonces, por el axioma D1, se tiene

$$DIV \sqsubseteq \Box_{b\in B}\, b \rightarrow Q_{b,B}$$

  Por otra parte, como $[\![P']\!] \sqsubseteq [\![Q']\!]$, y además Q' tiene un número menor de estados iniciales, por hipótesis de inducción tenemos

$$P' \sqsubseteq Q'$$

  De donde, aplicando la regla $C3'$, concluimos

$$P \sqsubseteq Q$$

- Caso 2. Los procesos P y Q tienen los mismos estados, pero con distinta probabilidad de ser alcanzados.

En este caso se sigue el mismo razonamiento que en el caso anterior, pues existe un A, tal que $p_A \leq q_A$. Como para el resto de estados, la probabilidad en P es más pequeña que en Q, la suma seguirá siendo menor, por lo que, para los procesos P' y Q', dados por

$$\begin{aligned} P &= [1-q_A]P' \sqcap [q_A]([\tfrac{p_A}{q_A}] \underset{a\in A}{\square} a \to P_{a,A} \sqcap [\tfrac{q_A-p_A}{q_A}]DIV) \\ Q &= [1-q_A]Q' \sqcap [q_A]([\tfrac{p_A}{q_A}] \underset{a\in A}{\square} a \to Q_{a,A} \sqcap [\tfrac{q_A-p_A}{q_A}] \underset{a\in A}{\square} a \to Q_{a,A}) \end{aligned}$$

se tiene, por hipótesis de inducción

$$P' \sqsubseteq Q'$$

de donde, aplicando $C2'$ y $C3'$ se sigue la conclusión deseada.

- Caso 3. Los procesos P y Q tienen los mismos estados iniciales, con la misma probabilidad de alcanzar cada uno de ellos.

  En este caso, como $[\![P]\!] \sqsubseteq [\![Q]\!]$, se tendrá que $\forall A \in \mathcal{A}$ y $\forall a \in A$, $[\![P_{a,A}]\!] \sqsubseteq [\![Q_{a,A}]\!]$, por lo que, por hipótesis de inducción, concluimos

  $$P_{a,A} \sqsubseteq Q_{a,A}$$

  Entonces, aplicando reiteradamente el axioma $C2'$, obtenemos que para todo estado A se satisface

  $$\underset{a\in A}{\square} a \to P_{a,A} \sqsubseteq \underset{a\in A}{\square} a \to Q_{a,A}$$

  y aplicando de nuevo reiteradamente el axioma $C3'$ concluimos

  $$\underset{A\in\mathcal{A}}{\sqcap} [p_A] \underset{a\in A}{\square} a \to P_{a,A} \sqsubseteq \underset{A\in\mathcal{A}}{\sqcap} [p_A] \underset{a\in A}{\square} a \to Q_{a,A}$$

En lo que respecta al caso caso en el que alguno de los procesos es infinito, comenzaremos considerando el caso en que P es infinito y Q es finito. Entonces las aproximaciones finitas de P verifican

$$[\![P^0]\!] \sqsubseteq [\![P^1]\!] \sqsubseteq \cdots \sqsubseteq [\![P^n]\!] \cdots \sqsubseteq [\![P]\!] \sqsubseteq [\![Q]\!]$$

Puesto que cada P^n es finito, tenemos entonces, en virtud de lo ya probado anteriormente que, para todo n se verifica

$$P^n \sqsubseteq Q$$

por lo que, aplicando la regla $R2$ podemos concluir

$$P \sqsubseteq Q$$

Por el contrario, en el caso de que P sea finito y Q infinito definido recursivamente, tenemos que como el único medio de llegar a los procesos infinitos es por via de sus aproximaciones finitas, el camino más razonable para llegar a demostrar $[\![P]\!] \sqsubseteq [\![Q]\!]$ consistiría en garantizar la existencia de un cierto n tal que la aproximación finita Q^n del proceso Q cumpla

$$[\![P]\!] \sqsubseteq [\![Q^n]\!]$$

Si el dominio semántico sobre el que trabajamos fuese ω-algebraico la existencia de un n tal estaría garantizada; sin embargo, y desgraciadamente, ello no es así, como prueba el siguiente contraejemplo:

Consideremos el proceso recursivo definido por la ecuación

$$P \;=\; [\tfrac{1}{2}]a \rightarrow STOP \sqcap [\tfrac{1}{2}]P$$

Sus aproximaciones finitas vienen dadas por

$$P^n \;=\; [1 - \frac{1}{2^n}]a \rightarrow STOP \;\sqcap\; [\frac{1}{2^n}]DIV$$

Entonces, como tenemos la igualdad $[\![P]\!] = \sqcup[\![P^n]\!]$, se tiene en particular $[\![P]\!] \sqsubseteq \sqcup[\![P^n]\!]$, y sin embargo es obvio que no existe ningún n tal que $[\![P]\!] \sqsubseteq [\![P^n]\!]$, puesto que ello implicaría en particular que para dicho n se tendría $1 \leq 1 - \frac{1}{2^n}$, lo que es absurdo.

Es por tanto preciso, utilizar un camino alternativo de cara a probar la relación $P \sqsubseteq Q$. Encontramos así la causa por la que incluímos en nuestro sistema lógico la regla $R3$, que sin duda habrá resultado en su momento un tanto chocante, por lo inusual de la misma en este tipo de situaciones, pero que ahora quedará completamente justificada.

Partimos de que

$$[\![P]\!] \sqsubseteq [\![Q]\!] = \sqcup [\![Q^n]\!]$$

Entonces, si existiese un n con $[\![P]\!] \sqsubseteq [\![Q^n]\!]$, la simple aplicación de $R2$ nos permitiría concluir la prueba. Pero ya sabemos que en general la existencia de un n tal no está garantizada. Centrémonos en dicho caso: entonces, como la sucesión Q^n es una cadena, tendremos que para toda secuencia s y todo estado A se cumplirá

$$p(P, s, A) \leq p(Q, s, A) = \lim_n p(Q^n, s, A)$$

En particular, para los pares (s, A) para los cuales $p(Q, s, A) > 0$ se tendrá que, para todo $k > 0$

$$(1 - \frac{1}{k}) \cdot p(P, s, A) < \lim_n p(Q^n, s, A)$$

y como quiera que el conjunto de dichos pares es finito, existirá un n suficientemente grande, tal que para todo $k \in \mathbb{N}$ se satisfaga

$$p([\frac{k-1}{k}]P \sqcap [\frac{1}{k}]DIV, s, A) \leq p(Q^n, s, A)$$

De donde

$$[\![[\frac{k-1}{k}]P \sqcap [\frac{1}{k}]DIV]\!] \sqsubseteq [\![Q^n]\!]$$

y puesto que tanto P como Q^n son finitos, obtenemos

$$[\frac{k-1}{k}]P \sqcap [\frac{1}{k}]DIV \sqsubseteq Q^n$$

Por lo que estamos en condiciones de aplicar la regla $R3$, lo que nos conduce a

$$P \sqsubseteq Q^n$$

y por último, aplicando $R1$ y $R2$ concluimos

$$P \sqsubseteq Q$$

Finalmente, si ambos procesos P y Q son infinitos, y se satisface $[\![P]\!] \sqsubseteq [\![Q]\!]$, entonces, en virtud de lo probado en los casos anteriores, tenemos que para toda aproximación finita P^n de P se verifica $P^n \sqsubseteq Q$ y por tanto, aplicando la regla $R2$ podemos concluir que $P \sqsubseteq Q$. □

Nótese, sin embargo, el paralelismo existente entre las reglas $R2$ y $R3$. Este parecido hace que exista otra alternativa de cara a resolver el problema que se nos presentó en la demostración anterior. La misma nos permitiría mantener en el sistema, en lo que se refiere al tratamiento de la recursión, únicamente las reglas $R1$ y $R2$, que son las usuales en un marco como en el que nos encontramos. La alternativa consiste en introducir una leve diferencia en la definición de las aproximaciones finitas de un proceso recursivo, que pasarían a venir definidas en la forma

$$\begin{aligned} P'_0 &= DIV \\ P'_{n+1} &= [\tfrac{n}{n+1}]P^n \sqcap [\tfrac{1}{n+1}]DIV \end{aligned}$$

donde los P^n son las anteriores aproximaciones finitas.

De esta manera, el significado conjunto de las reglas $R2$ y $R3$ quedaría englobado en la propia regla $R2$.

Finalmente, y puesto que nuestro trabajo se ha basado en principio en la semantica de pruebas, terminamos con el siguiente

Corolario 6.2 El sistema axiomático es completo respecto a la equivalencia semántica inducida por la semántica de pruebas.

Demostración: Se sigue de la equivalencia existente entre la semántica denotacional y la de pruebas, a nivel de las equivalencias entre procesos inducidas por las mismas. □

Capítulo 7

Semántica operacional

En la introducción al lenguaje PCSP, realizada en el capítulo 3, concluíamos que la semántica operacional allí presentada, basada en las transiciones de la forma

$$P \xrightarrow{a}_p Q$$

no era adecuada para caracterizar el significado semántico que pretendíamos, debido a que la información proporcionada por dichas transiciones no es suficiente para describir la capacidad de elección de que dispone el entorno en un momento dado. Bajo la luz arrojada por los capítulos anteriores, parece evidente que la información necesaria para cubrir dicho objetivo, precisa de la incorporación a las transiciones, de una u otra forma, de la noción de *estado*, por lo que en este capítulo trabajaremos con transiciones más complejas, que reflejarán de una forma explícita dicha información.

7.1 Transiciones probabilísticas

En primer lugar, modificamos la noción de transición, por medio de la siguiente

Definición 7.1 Definimos una **transición probabilística observable basada en estados** (en lo sucesivo simplemente transiciones probabilísticas) como una tupla $\langle P, Q, a, A, p\rangle$ donde P y Q son procesos probabilísticos, $a \in A \subseteq \underline{\Sigma}$, y

$p \in (0,1]$. Usualmente representaremos una transición tal en la forma

$$T = P \xrightarrow{a,A}_p Q$$

□

Ahora bien, al igual que ocurría en el capítulo 3, una misma transición podrá ser derivada de múltiples formas, correspondiendo cada derivación a un cómputo diferente. Por tanto, por las mismas razones ya aducidas en dicho capítulo, de nuevo hablaremos de multiconjuntos de transiciones derivables.

Asímismo, de manera similar a cómo en dicho capítulo necesitábamos la noción de probabilidad de ejecución de una acción, que denotábamos por

$$P \xrightarrow{a}_p$$

en este caso, lo que necesitamos es la probabilidad de alcanzar un estado, puesto que una vez alcanzado un estado, el medio exterior seleccionará de entre ellas que acción ejecuta, sin que intervenga al respecto ninguna nueva elección probabilística.

En consecuencia, junto con las reglas que definen las nuevas transiciones probabilísticas, tendremos otras que para cada estado $A \subseteq \Sigma$, definirán la probabilidad p con la que cada proceso P alcanza el estado en cuestión, lo que representamos en la forma

$$P \xrightarrow{A}_p$$

Presentamos a continuación las reglas que permiten derivar las transiciones correspondientes a cada uno de los operadores del lenguaje.

PR) $$\frac{}{a \rightarrow P \xrightarrow{a,\{a\}}_1 P}$$

I1) $$\frac{P \xrightarrow{a,A}_p Q}{[q]P \sqcap [1-q]R \xrightarrow{a,A}_{pq} Q}$$

I2) $$\frac{P \xrightarrow{a,A}_p Q}{[q]R \sqcap [1-q]P \xrightarrow{a,A}_{p(1-q)} Q}$$

Estas tres primeras reglas son simples adaptaciones de las correspondientes de la primera versión de la semántica operacional, y no precisan por ello de mayor comentario.

$$\textbf{E1)}\ \frac{P \xrightarrow{a,A}_p P',\ Q \xrightarrow{B}_q ,\ a \in B}{[r]P \,\square\, [1-r]Q \xrightarrow{a,A\cup B}_{pqr} P'} \qquad \textbf{E2)}\ \frac{P \xrightarrow{A}_p ,\ Q \xrightarrow{a,B}_q Q' ,\ a \in A}{[r]P \,\square\, [1-r]Q \xrightarrow{a,A\cup B}_{pq(1-r)} Q'}$$

$$\textbf{E3)}\ \frac{P \xrightarrow{a,A}_p P',\ Q \xrightarrow{B}_q ,\ a \notin B}{[r]P \,\square\, [1-r]Q \xrightarrow{a,A\cup B}_{pq} P'} \qquad \textbf{E4)}\ \frac{P \xrightarrow{A}_p ,\ Q \xrightarrow{a,B}_q Q' ,\ a \notin A}{[r]P \,\square\, [1-r]Q \xrightarrow{a,A\cup B}_{pq} Q'}$$

Las reglas E1 a E4 son también adaptaciones de las anteriores reglas aparecidas en la primera versión, si bien ya no resultan inmediatas, en tanto y cuando el concepto de estado, y en consecuencia la probabilidad de alcanzarlo juegan un papel fundamental en las mismas. Así, si A es el estado que alcanza P y B el estado alcanzado por Q, entonces el estado alcanzado por la elección externa es $A \cup B$. Por otra parte, las reglas E1 y E3 (y análogamente E2 y E4) indican que la acción a será ejecutada por la componente P en dos situaciones: cuando dicha acción esté en A y no en B, pero también cuando esté en ambos conjuntos, si bien en tal caso deberemos ponderar con la probabilidad indicada en la operación de elección para obtener la probabilidad de la transición en cuestión.

$$\textbf{P1)}\ \frac{P \xrightarrow{a,B}_p P' ,\ Q \xrightarrow{C}_{p'} ,\ a \notin A \cup C}{[q]P \parallel_A [1-q]Q \xrightarrow{a,D}_{pp'} [q]P' \parallel_A [1-q]Q}$$

$$\textbf{P2)}\ \frac{P \xrightarrow{B}_{p'} ,\ Q \xrightarrow{a,C}_p Q' ,\ a \notin A \cup B}{[1-q]P \parallel_A [q]Q \xrightarrow{a,D}_{pp'} [1-q]P \parallel_A [q]Q'}$$

$$\textbf{P3)}\ \frac{P \xrightarrow{a,B}_p P' ,\ Q \xrightarrow{C}_{p'} ,\ a \in C - A}{[q]P \parallel_A [1-q]Q \xrightarrow{a,D}_{pp'q} [q]P' \parallel_A [1-q]Q}$$

$$\textbf{P4)}\ \frac{P \xrightarrow{B}_{p'} ,\ Q \xrightarrow{a,C}_p Q' ,\ a \in B - A}{[1-q]P \parallel_A [q]Q \xrightarrow{a,D}_{pp'q} [1-q]P \parallel_A [q]Q'}$$

P5) $$\frac{P \xrightarrow{a,B}_p P' \ , \ Q \xrightarrow{a,C}_{p'} Q' \ , \ a \in A}{[q]P \parallel_A [1-q]Q \xrightarrow{a,D}_{pp'} [q]P' \parallel_A [1-q]Q'}$$
donde $D = (B \cup C - A) \cup (B \cap C)$

Una vez más, el tratamiento de la composición paralela es similar al de la elección externa, con la salvedad de que el estado correspondiente a cada transición será el dado por $(B \cup C - A) \cup (B \cap C)$ siendo B el estado correspondiente a la transición de P que interviene en su generación, C el correspondiente de Q, y A el conjunto de sincronización.

O1) $$\frac{P \xrightarrow{b,A}_p P', a \notin A}{P \setminus (a,q) \xrightarrow{b,A}_p P' \setminus (a,q)}$$

O2) $$\frac{P \xrightarrow{a,A}_p P', \ P'\setminus(a,q) \xrightarrow{b,B}_{p'} P''}{P \setminus (a,q) \xrightarrow{b,B}_{pp'q} P''}$$

O3) $$\frac{P \xrightarrow{a,A}_p P', \ P \xrightarrow{b,A}_p Q, \ P'\setminus(a,q) \xrightarrow{B}_{p'}}{P \setminus (a,q) \overset{b,A-\{a\}\cup B}{\longrightarrow}_{pp'(1-q)} Q \setminus (a,q)}$$

O4) $$\frac{P \xrightarrow{a,A}_p P', \ P'\setminus(a,q) \xrightarrow{b,B}_{p'} P'', \ b \notin A}{P \setminus (a,q) \overset{b,A-\{a\}\cup B}{\longrightarrow}_{pp'(1-q)} P''}$$

La descripción de las reglas correspondientes al operador de ocultación es sensiblemente más complicada a la de las anteriores. Básicamente repetimos el razonamiento seguido en el capítulo 4 al definir la probabilidad de alcanzar un estado por parte del proceso $P\setminus(a,q)$.

Así, la regla O1 indica que si un estado que no contiene la acción a ocultar es alcanzado por el proceso P con probabilidad p, dicho estado sigue siendo alcanzado con la misma probabilidad, después de ocultar la acción en cuestión.

La regla O2 indica que si P alcanza un estado que contiene la acción a ocultar, ésta podrá ejecutarse con probabilidad q, con lo que se alcanzarían los estados accesibles en dicha situación.

La regla O3 se refiere al caso en el que la acción ocultada no se ejecuta. En este caso, las acciones que pueden ejecutarse en principio son las de $A - \{a\}$, siendo A el estado del proceso antes de la ocultación, que contiene a la acción a ocultada. La acción b ejecutada por el proceso resultante de la ocultación es en efecto una de las acciones de dicho conjunto. Sin embargo, el estado en el que se ha producido la elección de dicha acción no es el definido por dicho conjunto, sino el obtenido uniendo al mismo uno de los estados del proceso continuación del proceso original, tras la ejecución por parte del mismo de la acción ocultada. Con ello reflejamos el hecho de que si bien el proceso había *decidido* en principio no ejecutar la acción ocultada, el entorno podría *obligar* al proceso a ejecutar la acción oculta, si desease ejecutar una acción que sólo es posible tras la ejecución de la acción ocultada.

La regla O4 cubre precisamente la posibilidad de ejecutar acciones que no están en A, aún cuando en un principio el sistema no haya decidido ejecutar la acción oculta, caso ya cubierto por la regla O2. Ahora la probabilidad $1 - q$ restante vuelve a revertir en la acción oculta, con lo que en definitiva se obtiene que la misma será ejecutada forzosamente, si el medio exterior lo necesita.

Obsérvese que las acciones que pueden ser ejecutadas en virtud de esta regla son exclusivamente aquellas que no están en A, pues la ejecución de las acciones de A en el caso de que el proceso decida en principio no ejecutar la acción ocultada, correspondería siempre a los correspondientes cómputos que eluden en efecto la ejecución de la acción ocultada, caso cubierto por la regla anterior O3.

$$\text{R)} \quad \frac{P[\mu X.P|X] \xrightarrow{a,A}_p P'}{\mu X.P \xrightarrow{a,A}_p P'}$$

Finalmente, la regla de recursión tampoco sufre ningún cambio notable respecto a la ya presentada en el capítulo 3. Estudiaremos su funcionamiento mediante el siguiente

Ejemplo 7.1 Consideremos el proceso recursivo $\mu X.[\frac{1}{2}]a \rightarrow STOP \sqcap [\frac{1}{2}]X$. Las transiciones que podemos obtener para el mismo serán las siguientes:

En primer lugar, la expresión P con una variable libre X a considerar será

$$P = [\frac{1}{2}]a \rightarrow STOP \sqcap [\frac{1}{2}]X$$

la misma expresión, en la que sustituimos la variable X por el propio proceso recursivo será entonces

$$P[\mu X.P|X] = [\frac{1}{2}]a \rightarrow STOP \sqcap [\frac{1}{2}]\mu X.P$$

A partir de esta expresión, y de I1 obtenemos

$$P[\mu X.P|X] \xrightarrow{a,A}_{\frac{1}{2}} STOP$$

de donde, utilizando la regla R, concluimos

$$\mu X.P \xrightarrow{a,A}_{\frac{1}{2}} STOP$$

Ahora bien, a partir de esta transición, y utilizando I2 obtenemos

$$P[\mu X.P|X] \xrightarrow{a,A}_{\frac{1}{4}} STOP$$

por lo que, nuevamente por medio de R, llegamos a

$$\mu X.P \xrightarrow{a,A}_{\frac{1}{4}} STOP$$

Y así sucesivamente, obtenemos las distintas transiciones operacionales que describen el operador de recursión. □

Ha quedado pendiente la definición operacional de las probabilidades con las que cada proceso P alcanza cada uno de sus estados A, para lo cual introducimos las siguientes reglas:

ES) $\dfrac{}{STOP \xrightarrow{\emptyset}_1}$ **ER)** $\dfrac{}{a \rightarrow P \xrightarrow{\{a\}}_1}$

EI1) $\dfrac{P \xrightarrow{A}_q}{[p]P \sqcap [1-p]Q \xrightarrow{A}_{pq}}$ **EI2)** $\dfrac{Q \xrightarrow{A}_r}{[p]P \sqcap [1-p]Q \xrightarrow{A}_{(1-p)r}}$

EE) $\dfrac{P \xrightarrow{A}_q,\ Q \xrightarrow{B}_r}{[p]P \Box [1-p]Q \xrightarrow{A \cup B}_{qr}}$ **EP)** $\dfrac{P \xrightarrow{B}_q,\ Q \xrightarrow{C}_r}{[p]P \|_A [1-p]Q \xrightarrow{D}_{qr}}$

con $D = (B \cup C) - A \cup (B \cap C)$

EO1) $\dfrac{P \xrightarrow{A}_p, a \notin A}{P \backslash (a,q) \xrightarrow{A}_p}$ **EO2)** $\dfrac{P \xrightarrow{a,A}_p P',\ P' \backslash (a,q) \xrightarrow{B}_r}{P \backslash (a,q) \xrightarrow{B}_{pqr}}$

EO3) $\dfrac{P \xrightarrow{a,A}_p P',\ P' \backslash (a,q) \xrightarrow{B}_r}{P \backslash (a,q) \xrightarrow{A \cup B - \{a\}}_{pr(1-q)}}$

ER) $\dfrac{P[\mu X.P|X] \xrightarrow{A}_p}{\mu X.P \xrightarrow{A}_p}$

A excepción de las correspondientes al operador de ocultación, todas estas reglas son bastante naturales, en función de la interpretación que pretendemos para los distintos operadores. En lo que se refiere a la ocultación, tenemos tres diferentes tipos de estados. En primer lugar, los estados alcanzables por P que no incluyan la acción a, pueden ser alcanzados después de ocultar, con la misma probabilidad que antes de hacerlo.

En segundo lugar, si P alcanza un estado que contiene la acción a, ésta se ejecutará internamente con probabilidad q, con lo que los estados alcanzables tras ello son también alcanzables por el proceso resultante de la ocultación, pero naturalmente con la probabilidad resultante de multiplicar por q la que se tenga tras la ejecución de la acción oculta.

Finalmente, en el mismo caso la acción a no se ejecutará (en principio) autónomamente con probabilidad $1-q$, por lo que el estado sería $A-\{a\}$, el cual, en línea con lo explicado al comentar las reglas generadoras de las transiciones correspndientes a este operador, hay que extender con las acciones de cada estado accesible tras la (posible) ejecución impuesta de la acción oculta.

Pero como quiera que en general tenemos múltiples formas de llegar a un mismo estado, para obtener la probabilidad total de alcanzar cada uno de ellos, deberemos sumar las probabilidades correspondientes a las distintas derivaciones que nos conducen a cada estado, lo que formalizamos a continuación:

Definición 7.2 (Probabilidad global de alcanzar un estado)
Sea $A \subseteq \Sigma$, y P un proceso. Definimos la **probabilidad global** de que el proceso P alcance el estado A, como la suma $\sum_{i\in I} p_i$, donde $\{P \xrightarrow{A}_{p_i} \mid i \in I\}$ es el multiconjunto de informaciones parciales sobre la probabilidad de alcanzar ese estado, que pueden inferirse mediante el sistema de reglas presentado, utilizando distintas derivaciones. □

En aras de una notación precisusa, deberíamos representar la probabilidad global con la que un proceso P alcanza el estado A, utilizando una notación diferente a la utilizada para denotar las informaciones parciales sobre la misma. Sin embargo, hemos preferido abusar de la notación, reutilizando para denotar las probabilidades globales esa misma notación, pues una vez utilizadas las informaciones parciales para calcular las globales, las mismas dejan de interesarnos. Por lo tanto, en lo sucesivo, y tambien en las reglas del sistema que generaban las transiciones probabilísticas (pero no, por supuesto, en las que definen las probabilidades parciales de alcanzar un estado), cuando indiquemos $P \xrightarrow{A}$ nos referiremos a la correspondiente probabilidad global.

Definición 7.3 (Semántica Operacional)
Llamamos semántica operacional del lenguaje PCSP al multiconjunto de transiciones probabilísticas que pueden ser inferidas con el sistema de reglas anterior, donde cada transición aparece tantas veces como formas diferentes haya de derivarla. □

Esta definición de semántica operacional va a ser la adecuada para capturar la semántica de pruebas definida en el capítulo 4, pero como usualmente ocurre, la presencia de estados intermedios no es deseable de cara a caracterizar la noción de equivalencia inducida, como se observa en el siguiente

Ejemplo 7.2 Consideremos los procesos

$$\begin{aligned} P &= [\tfrac{1}{2}]a \to b \to STOP \sqcap [\tfrac{1}{2}]a \to STOP \\ Q &= a \to ([\tfrac{1}{2}]b \to STOP \sqcap [\tfrac{1}{2}]STOP) \end{aligned}$$

Para los mismos tenemos como únicas transiciones las siguientes:

$$\begin{aligned} &P \xrightarrow{a,\{a\}}_{\frac{1}{2}} b \to STOP\ , \quad P \xrightarrow{a,\{a\}}_{\frac{1}{2}} STOP \\ &Q \xrightarrow{a,\{a\}}_{1} [\tfrac{1}{2}]b \to STOP \sqcap [\tfrac{1}{2}]STOP \end{aligned}$$

Por tanto, ambos deben ser equivalentes, si bien ello no queda plasmado directamente por la semántica operacional, ya que la forma en que se realizan los cómputos de ambos es sintácticamente diferente. □

Ahora bien, en el capítulo anterior hemos comprobado que la elección interna es, módulo la normalización precisa, asociativa, gracias a lo cual podíamos generalizarla a n operandos. De hecho, dicha extensión puede llevarse más lejos, hasta capturar familias numerables de operandos. Basándonos en dicha idea introduciremos un nuevo tipo de transición que nos servirá para razonar directamente sobre las equivalencias entre procesos.

Definición 7.4 Sea A un conjunto de acciones, con $a \in A$, y P un proceso. Definimos una **transición global** como una tupla $\langle P, Q, a, A, p\rangle$, donde $p = \sum_{i \in I} p_i$, siendo $\{P \xrightarrow{a,A}_{p_i} P_i \mid i \in I\}$ el multiconjunto de transiciones que constituyen la semántica operacional del proceso P, y donde

$$Q = \prod_{i \in I} [\frac{p_i}{p}]P_i$$

Representaremos las transiciones globales en la forma

$$P \overset{a,A}{\longmapsto}_p Q$$

Por extensión, si se tiene $P \xrightarrow{A}_0$ admitiremos que $\forall a \in A$, $\forall Q$ se tiene la transición global $P \overset{a,A}{\longmapsto}_0 Q$. □

Definición 7.5 (Semántica Operacional Abstracta)
Llamamos semántica operacional abstracta del lenguaje PCSP al conjunto de transiciones globales que pueden ser inferidas por medio de la definición anterior. □

7.2 Computaciones de procesos probabilísticos

También la definición de computaciones puede ser adaptada de una forma natural de la introducida en el capítulo 3. Una computación de un proceso probabilístico P es una secuencia de transiciones de la forma

$$P = P_0 \overset{a_1,A_1}{\longmapsto}_{p_1} P_1, \ldots, P_{n-1} \overset{a_n,A_n}{\longmapsto}_{p_n} P_n$$

que usualmente representaremos en la forma

$$C = P \overset{a_1,A_1}{\longmapsto}_{p_1} P_1 \overset{a_2,A_2}{\longmapsto}_{p_2} \cdots \overset{a_n,A_n}{\longmapsto}_{p_n} P_n$$

Llamaremos una *derivada* de P generada por C al triple $\langle s, P_n, p\rangle$, donde $s = (a_1, A_1)\ldots(a_n, A_n)$ y $p = p_1 \cdots p_n$. Usualmente representaremos estas derivadas con la notación:

$$P \overset{s}{\Longrightarrow}_p P_n$$

y diremos que P ejecuta con probabilidad p las acciones $a_1, \ldots, a_n$ pasando por los estados $A_1, \ldots, A_n$ para pasar a comportarse como el proceso P_n, que diremos que es un proceso *s-derivado* de P con probabilidad p.

Esta noción puede ser formalizada en la forma siguiente:

Definición 7.6 Sea s una secuencia (posiblemente vacía) de pares (a_i, A_i), tales que $a_i \in A_i$, $p \in (0,1]$, y P y Q dos procesos. Entonces, $P \overset{s}{\Longrightarrow}_p Q$ es una computación probabilística sii puede ser inferida por medio de las reglas

1. $P \overset{()}{\Longrightarrow}_1 P$
2. Si $P \overset{a,A}{\longmapsto}_p P'$ y $P' \overset{s}{\Longrightarrow}_q Q$, entonces $P \overset{(a,A)s}{\Longrightarrow}_{pq} Q$

□

Definición 7.7 (Equivalencia por observación de estados)
Sean P y Q dos procesos. Decimos que son *equivalentes por observación de estados* y lo denotamos por $P \equiv_O Q$ sii se cumple que

$$P \overset{a_1,A_1}{\longmapsto}_{p_1} P_1 \overset{a_2,A_2}{\longmapsto}_{p_2} \cdots \overset{a_n,A_n}{\longmapsto}_{p_n} P_n$$

es una computación de P sii existen $\{Q_1, \ldots, Q_n\}$ tales que

$$Q \overset{a_1,A_1}{\longmapsto}_{p_1} Q_1 \overset{a_2,A_2}{\longmapsto}_{p_2} \cdots \overset{a_n,A_n}{\longmapsto}_{p_n} Q_n$$

es una computación de Q. □

7.3 Equivalencia entre las semánticas

Al igual que hemos comprobado para el modelo denotacional de PCSP, este nuevo modelo operacional también es equivalente al modelo de pruebas (y por tanto al denotacional). Esto es, si denotamos por $\equiv_T$ a la equivalencia inducida por los tests, y $\equiv_O$ a la nueva equivalencia operacional por observación de estados, se tiene

$$P \equiv_T P' \Leftrightarrow P \equiv_O P'$$

De cara a probar dicha equivalencia, tenemos que, puesto que la semántica de pruebas de un proceso P viene caracterizada por los valores $p(P, A)$ y los procesos $P/(a, A)$, como se comprobó en la prop. 4.2, bastará comprobar que para cada proceso sintáctico, se tiene en primer lugar que $p(P, A) = p$ equivale a la existencia de una transición $P \xrightarrow{A}_p$. En segundo lugar, hemos de probar que el proceso semántico $P/(a, A)$, definido en el marco de la semántica de pruebas, equivale a un cierto proceso sintáctico Q, tal que existe una transición global $P \overset{a,A}{\longmapsto}_p Q$. En este último caso, tenemos que comenzar precisando la noción de equivalencia adecuada, puesto que se trata de comparar un proceso $P/(a, A)$ definido en el marco de la semántica de paso de tests, con otro Q que lo está con técnicas operacionales.

Para ello, procedemos a generalizar la noción de equivalencia, de modo que podamos, aunque sea por medio de un abuso de la notación, manejar procesos definidos a nivel semántico como si de procesos sintácticos se tratara. En particular,

ello nos permitirá escribir para cualquier proceso P, la transición

$$P \overset{a,A}{\longmapsto}_{p(P,A)} P/(a,A)$$

Para realizar la prueba de esta equivalencia no podemos proceder, como hemos hecho hasta ahora, por medio de la inducción estructural respecto de la construcción de los procesos sintácticos en juego. Ello es debido a un problema que aparece a la hora de considerar el operador de ocultación, puesto que las transiciones definidas para $P\backslash(a,q)$ se obtienen a partir de las transiciones de procesos de la forma $P/(a,A)\backslash(a,q)$, pero la inducción estructural no nos permite suponer que la propiedad que deseamos probar se satisface para el proceso $P/(a,A)$, sino sólo para el propio proceso P.

Por ello, nos vemos obligados a plantear otro tipo de inducción más compleja, basando la misma en una medida más compleja de la complejidad de los procesos (finitos), incluyendo como nuevo operador el operador *tras*, bajo la cual se tenga que todo proceso $P/(a,A)$ es más simple que el correspondiente proceso P, de manera que podamos aplicar a los mismos la hipótesis de inducción.

Para ello definiremos un *nuevo orden estructural*, basado en el número máximo de acciones en secuencia que podría ejecutar el proceso, haciendo intervenir también en la definición, para que entre en juego en caso de igualdad respecto de dicho primer criterio, el número de operaciones anidadas que aparecen en el mismo. Ello se refleja en el hecho de que la función *nivel* que define la complejidad de cada proceso, no toma simples valores enteros, sino secuencias no crecientes de los mismos. Para conseguir un orden bien fundado sobre el que podamos plantear una inducción, tendremos en cuenta el hecho de que dichas secuencias están ordenadas respecto del orden lexicográfico, que es en efecto bien fundado cuando se restringe a dicho tipo de secuencias. Todo ello se formaliza como sigue:

Definición 7.8 (Nivel de un operador)
Dado un proceso sintáctico finito $P \in PCSP$ definimos el *nivel* de P como la función

$$nivel : PCSP \longrightarrow \mathbb{N}^*$$

dada por

$$nivel(STOP) = nivel(DIV) = \langle 1 \rangle$$
$$nivel(a \rightarrow P) = nivel(P) + \langle 1 \rangle$$
$$nivel([p]P \sqcap [1-p]Q) = ord(nivel(P) \cdot nivel(Q))$$
$$nivel([p]P \Box [1-p]Q) = ord(nivel(P) \cdot nivel(Q))$$
$$nivel([p]P \|_A [1-p]Q) = ord(nivel(P) + nivel(Q))$$
$$nivel(P \backslash (a,q)) = nivel(P) + \langle 1 \rangle$$

donde la suma de secuencias que aparece es la operación que asocia a cada par de secuencias la obtenida sumando a cada elemento de la primera cada elemento de la segunda; y *ord* es la función que reordena en orden no creciente los elementos de una secuencia.

Por otra parte, definimos el *primer nivel* de un proceso como el primero (y por tanto máximo) de los valores de la secuencia que define su nivel. Es decir

$$nivel1 : PCSP \longrightarrow \mathbb{N}$$

definido en la forma $nivel1(P) = cabeza(nivel(P))$. □

Para poder razonar por inducción en base a la medida de complejidad inducida por la función *nivel*, comenzaremos probando el siguiente resultado técnico:

Lema 7.1 El orden lexicográfico sobre el subconjunto de $\mathbb{N}^*$ formado por las cadenas de naturales no crecientes, es un orden bien fundado.

Demostración: Comprobemos que no existen cadenas decrecientes de longitud infinita. Ello lo haremos mediante una inducción doble: en primer lugar respecto del primer elemento (el mayor) de la secuencia, y después, de forma anidada, sobre el número de veces que dicho elemento aparece.

Si el primer elemento es 1, es evidente que cualquier sucesión decreciente de cadenas formadas por sólo unos tiene un número finito de elementos, puesto que para ir decreciendo debemos eliminar en cada paso de la sucesión algún 1 de la cadena de partida.

Supuesto que no existen cadenas decrecientes infinitas cuyo primer elemento sea una secuencia que comienza por un valor inferior a n, consideremos una secuencia decreciente de valores cuyo primer elemento sea igual a n.

Razonemos ahora por inducción sobre el número de elementos iguales a dicho valor que contiene la secuencia. La base de esta segunda inducción consistirá en considerar que sólo hay uno de estos términos. Observamos entonces que si dicho valor n no lo tocamos a lo largo de la sucesión de secuencias, la sucesión que se obtiene al quitar dicho valor en todas ellas, conformaría una nueva sucesión decreciente formada por secuencias todas ellas de elementos inferiores a n, por lo que debería de ser una cadena finita. En consecuencia, si la cadena de partida fuese infinita debería contener secuencias cuyo primer elemento fuese inferior a n, pero en tal caso, la hipótesis de la primera inducción nos llevaría a un absurdo.

Por otra parte, supongamos que el primer elemento de la cadena comienza por una secuencia de exactamente k (con $k > 1$) valores iguales a n. En tal caso la hipótesis de la segunda inducción, nos garantiza la finitud de las cadenas que tienen como primer elemento una secuencia que tiene en cabeza una subsecuencia con menos de k valores iguales a n. Entonces, razonando como en el caso anterior, concluimos que en los elementos de la cadena de partida, deberemos eliminar en algún momento alguna n, puesto que en caso contrario tendríamos una cadena finita. Pero al eliminar una n, obtenemos una secuencia que comienza con menos de k valores n, por lo que podemos aplicar la hipótesis de la segunda inducción para concluir que la cadena que resta a partir de dicho punto es necesariamente finita. □

Definición 7.9 (Nuevo orden estructural)
Definimos el **nuevo orden estructural** entre procesos sintácticos finitos de PCSP, que denotaremos $P < Q$, como la relación que se obtiene por medio de las siguientes reglas:

$$\frac{}{P < op(P,S)} \qquad \frac{P < Q,\ Q < R}{P < R}$$

$$\frac{P < Q}{op(P,S) < op(Q,S)} \qquad \frac{nivel1(P_1) < Q,\ nivel1(P_2) < Q}{[p]P_1 \sqcap [1-p]P_2 < Q}$$

donde *op* representa tanto a cada uno de los constructores sintácticos de procesos PCSP finitos, como a sus operadores simétricos, y S denota a un proceso PCSP finito arbitrario. □

A partir de las reglas que definen esta relación, se sigue trivialmente que se trata de una relación de orden estricto que refina al orden estructural entre procesos. Por otra parte tenemos que dicha relación es un orden bien fundado, lo que nos permitirá razonar por inducción sobre dicho orden.

Lema 7.2 La relación $<$ entre procesos PCSP sintácticos finitos es un orden bien fundado.

Demostración: De las reglas que definen la relación, se desprende inmediatamente que $P < Q \Rightarrow nivel(P) < nivel(Q)$. De aquí, y del lema técnico anterior, concluimos que la relación es, en efecto, un orden bien fundado. □

Además de refinar el orden estructural tenemos que el proceso $P/(a, A)$ será menor que el correspondiente proceso P, propiedad que es la que perseguíamos a la hora de definir este nuevo orden. Ello se concluye a partir del siguiente:

Lema 7.3 Sean P y Q dos procesos PCSP finitos tales

$$P \xrightarrow{a,A}_p Q$$

Se tiene entonces que $Q < P$ y $nivel1(P) < nivel1(Q)$.

Demostración: Por inducción respecto del número de reglas utilizadas para probar la transición. La última regla en cuestión será una de las siguientes:

- PR: Entonces $P < a \rightarrow P$ por definición de $<$. Además $nivel1(a \rightarrow P) = nivel1(P) + 1$.
- I1: Tenemos entonces $P = [p]P_1 \sqcap [1-p]P_2$. Entonces, por hipótesis de inducción $Q < P_1$, y como quiera que $P_1 < P$ concluimos que $Q < P$.

  Por la misma razón, $nivel1(Q) < nivel1(P_1) \leq nivel1(P)$.

- I2, E1, E2, E3, E4: Análogos.

- P1: $P = [p]P_1 \|_A [1-p]P_2$ y $Q = [p]P_1' \|_A [1-p]P_2$. Por hipótesis de inducción tenemos $P_1' < P_1$, por lo que concluimos

$$Q = [p]P_1' \|_A [1-p]P_2 < [p]P_1 \|_A [1-p]P_2 = P$$

  Y también, $nivel1(Q) = nivel1(P_1') + nivel1(P_2) < nivel1(P_1) + nivel1(P_2) = nivel1(P)$.

- P2, P3, P4: Similares.

- P5: Partimos de $P_1' < P_1$ y de $P_2' < P_2$. Entonces para el proceso $Q = [p]P_1' \|_A [1-p]P_2'$ se tiene:

$$Q = [p]P_1' \|_A [1-p]P_2' < [p]P_1 \|_A [1-p]P_2' < [p]P_1 \|_A [1-p]P_2 = P$$

  Además, $nivel1(Q) = nivel1(P_1') + nivel1(P_2') < nivel1(P_1) + nivel1(P_2) = nivel1(P)$.

- O1: Por h.i. tenemos $P' < P$, por lo que $P'\backslash(a,q) < P\backslash(a,q)$.

  Y $nivel1(P'\backslash(a,q)) = nivel1(P') + 1 < nivel1(P) + 1 = nivel1(P\backslash(a,q))$.

- O2: Por h.i. tenemos $P'' < P'$ y además $Q < P''\backslash(a,q)$. Entonces

$$P''\backslash(a,q) < P'\backslash(a,q) \;=\; P$$

  por lo que

$$Q < P'\backslash(a,q) \;=\; P$$

  Cumpliéndose también $nivel1(Q) = nivel1(P'') + 1 < nivel1(P') + 1 = nivel1(P'\backslash(a,q)) = nivel1(P)$.

- O3: Similar a O1.

- O4: Similar a O2.

□

Corolario 7.1 Dada una transición $P \stackrel{a,A}{\longmapsto}_p Q$, correspondiente a la semántica operacional de un proceso finito P, se verifica $Q < P$.

Demostración: Si consideramos el multiconjunto de transiciones de la forma

$$P \stackrel{a,A}{\longrightarrow}_{p_i} P_i$$

tenemos, en virtud del lema anterior, que para todo i se verifica $P_i < P$, y $nivel1(P_i) < nivel1(P)$. Entonces, de la definición del proceso Q como

$$Q = \bigsqcap_{i \in I} [\frac{p_i}{p}] P_i \text{ con } p = \sum_{i \in I} p_i$$

y teniendo en cuenta que I es finito, por serlo el proceso P, concluimos que estamos en condiciones de aplicar reiteradamente la última regla de la definición de la relación de orden, con lo que concluimos $Q < P$. □

Finalmente estamos en condiciones de probar el siguiente

Teorema 7.1 Dado un proceso finito P correspondiente a la sintaxis de PCSP, para el que, con el sistema de transiciones etiquetado definido, se tienen las transiciones

$$P \stackrel{a,A}{\longmapsto}_{p_A} P'$$
$$P \stackrel{A}{\longrightarrow}_{p_A}$$

se verifica que $p_A = p(P, A)$, y además el proceso P' es equivalente, respecto al paso de tests, al proceso semántico $P/(a, A)$.

Demostración: Procederemos por inducción respecto al nuevo orden estructural entre procesos, lo que se reducirá a aplicar el orden estructural ordinario para todos los operadores a excepción del de la ocultación.

DIV y STOP: Trivial en virtud de su definición.

Prefijo: Por un lado, tenemos que $\forall A \subseteq \Sigma$

$$a \rightarrow P \stackrel{A}{\longrightarrow}_p \Leftrightarrow p(a \rightarrow P, A) = p$$

Lo cual es cierto, pues en ambos casos $p = 1$ si $A = \{a\}$, y $p = 0$ en caso contrario. Además, como $(a \rightarrow P)/(a, \{a\})$ es equivalente a P, se sigue inmediatamente el resultado deseado.

Elección interna: Llamando $R = [r]P \sqcap [1-r]Q$, tenemos en primer lugar, por hipótesis de inducción

$$\begin{aligned} P \xrightarrow{A}_{p_A} &\Leftrightarrow p(P, A) = p_A \\ Q \xrightarrow{A}_{p'_A} &\Leftrightarrow p(Q, A) = p'_A \end{aligned}$$

de donde obtenemos, por medio de las reglas EI1 y EI2, y de la definición 4.3

$$\begin{aligned} &R \xrightarrow{A}_{p \cdot p_A + (1-p) \cdot p'_A} \\ &p(R, A) = p \cdot p_A + (1-p) \cdot p'_A \end{aligned}$$

lo que nos permite concluir

$$R \xrightarrow{A}_{r_A} \Leftrightarrow p(R, A) = r_A$$

Además, de nuevo por hipótesis de inducción, podemos asumir

$$\begin{aligned} P &\overset{a,A}{\longmapsto}_{p_A} P/(a, A) \\ Q &\overset{a,A}{\longmapsto}_{q_A} Q/(a, A) \end{aligned}$$

de donde, mediante las reglas I1 e I2, y la def. 7.4, obtenemos

$$R \overset{a,A}{\longmapsto}_{r \cdot p_A + (1-r) \cdot q_A} \left[\frac{r \cdot p_A}{r \cdot p_A + (1-r) \cdot q_A}\right] P/(a, A) \sqcap \left[\frac{(1-r) \cdot q_A}{r \cdot p_A + (1-r) \cdot q_A}\right] Q/(a, A)$$

Por otro lado, aplicando la propiedad 3 enunciada en la prop. 4.1, tenemos

$$p(R, A) \cdot (R/(a, A) \mid t) = r \cdot p(P, A) \cdot (P/(a, A) \mid t) + (1-r) \cdot p(Q, A) \cdot (Q/(a, A) \mid t)$$

$$R/(a, A) \mid t = \frac{r \cdot p(P, A) \cdot (P/(a, A) \mid t) + (1-r) \cdot p(Q, A) \cdot (Q/(a, A) \mid t)}{r \cdot p(P, A) + (1-r) p(Q, A)}$$

Y por lo tanto, podemos concluir

$$R \overset{a,A}{\longmapsto}_{r \cdot p_A + (1-r) q_A} R/(a, A)$$

Elección externa: Nuevamente, llamamos $R = [r]P \square [1-r]Q$, y suponemos que, por hipótesis de inducción, se verifica

$$\begin{aligned} P \xrightarrow{A}_{p_A} &\Leftrightarrow p(P, A) = p_A \\ Q \xrightarrow{A}_{p'_A} &\Leftrightarrow p(Q, A) = p'_A \end{aligned}$$

Entonces, en virtud de la regla EE tenemos

$$R \xrightarrow{A \cup B}_{p_A p'_A} \Leftrightarrow p(P, A) = p_A \ \wedge \ p(Q, B) = p'_B$$

por lo que, en definitiva

$$R \xrightarrow{C}_{\sum_{A \cup B = C} p_A p'_B} \Leftrightarrow p(P, A) = p_A \ \wedge \ p(Q, B) = p'_B$$

Como ello es válido para todo par A, B, tenemos

$$R \xrightarrow{C}_{r_C} \Leftrightarrow p(R, C) = r_C$$

Por otra parte, también por hipótesis de inducción, tenemos

$$P \xmapsto{a,A}_{p_A} P/(a, A)$$
$$Q \xmapsto{a,B}_{q_B} Q/(a, B)$$

de donde, aplicando las reglas E1 a E4, obtenemos

$$\begin{array}{ll}
R \xrightarrow{a, A \cup B}_{p_A q_B} P/(a, A) & \text{si } a \in A - B \\
R \xrightarrow{a, A \cup B}_{p_A q_B} Q/(a, B) & \text{si } a \in B - A \\
R \xrightarrow{a, A \cup B}_{r p_A q_B} P/(a, A) & \text{si } a \in A \cap B \\
R \xrightarrow{a, A \cup B}_{(1-r) p_A q_B} Q/(a, B) & \text{si } a \in A \cap B
\end{array}$$

En consecuencia, obtenemos la transición global

$$R \xmapsto{a,C}_{s} R'$$

donde $s = \sum_{A \cup B = C} p_A p'_B$, y el proceso R' es el que viene dado por

$$\begin{aligned}
R' \ = \ & \left[\frac{\sum_{a \in A - B} p_A q_B}{\sum_{A \cup B = C} p_A q_B}\right] \prod_{a \in A - B} \left[\frac{p_A q_B}{\sum_{a \in A - B} p_A q_B}\right] P/(a, A) \\
\sqcap \ & \left[\frac{\sum_{a \in B - A} p_A q_B}{\sum_{A \cup B = C} p_A q_B}\right] \prod_{a \in B - A} \left[\frac{p_A q_B}{\sum_{a \in B - A} p_A q_B}\right] Q/(a, B) \\
\sqcap \ & \left[\frac{\sum_{a \in B \cap A} r p_A q_B}{\sum_{A \cup B = C} p_A q_B}\right] \prod_{a \in A \cap B} \left[\frac{r p_A q_A}{\sum_{a \in A \cap B} r p_A q_B}\right] P/(a, A) \\
\sqcap \ & \left[\frac{\sum_{a \in B \cap A} (1 - r) p_A q_B}{\sum_{A \cup B = C} p_A q_B}\right] \prod_{a \in A \cap B} \left[\frac{(1 - r) p_A q_A}{\sum_{a \in A \cap B} (1 - r) p_A q_B}\right] Q/(a, A)
\end{aligned}$$

Entonces, al igual que en el caso anterior, a partir de la propiedad 4 enunciada en la prop. 4.1, obtenemos que el proceso R' es equivalente al obtenido en la definición de $R/(a,C)$, por lo que podemos escribir

$$R \stackrel{a,C}{\longmapsto}_{p(R,C)} R/(a,C)$$

Composición paralela: Consideremos $R = [r]P\|_A[1-r]Q$. Por hipótesis de inducción tenemos

$$\begin{aligned} P \stackrel{B}{\longrightarrow}_{p_B} &\Leftrightarrow p(P,B) = p_B \\ Q \stackrel{C}{\longrightarrow}_{p'_C} &\Leftrightarrow p(Q,C) = p'_C \end{aligned}$$

Entonces, en virtud de la regla EP, llamando $D = (B \cup C) - A \cup (B \cap C)$, tenemos para cualquier par de estados B y C se verifica

$$R \stackrel{D}{\longrightarrow}_{p_B p'_C} \Leftrightarrow p(P,B) = p_B \ \wedge \ p(Q,C) = p'_C$$

por lo que, en definitiva, al igual que en el caso anterior

$$R \stackrel{D}{\longrightarrow}_{r_D} \Leftrightarrow p(R,D) = r_D$$

Por otra parte, tenemos también

$$\begin{aligned} P &\stackrel{a,B}{\longmapsto}_{p_B} P/(a,B) \\ Q &\stackrel{a,C}{\longmapsto}_{q_C} Q/(a,C) \end{aligned}$$

de donde

$$\begin{array}{ll} R \stackrel{a,D}{\longrightarrow}_{p_B q_C} [r]P/(a,B)\|_A[1-r]Q & \text{si } a \in B - (A \cup C) \\ R \stackrel{a,D}{\longrightarrow}_{p_B q_C} [r]P\|_A[1-r]Q/(a,C) & \text{si } a \in C - (A \cup B) \\ R \stackrel{a,D}{\longrightarrow}_{r p_B q_C} [r]P/(a,B)\|_A[1-r]Q & \text{si } a \in (B \cap C) - A \\ R \stackrel{a,D}{\longrightarrow}_{(1-r) p_A q_B} [r]P\|_A[1-r]Q/(a,C) & \text{si } a \in (B \cap C) - A \\ R \stackrel{a,D}{\longrightarrow}_{p_A q_B} [r]P/(a,B)\|_A[1-r]Q/(a,C) & \text{si } a \in A \cap B \cap C \end{array}$$

Por lo que, nuevamente aplicando la propiedad 5 enunciada en la prop. 4.1, y la definición 7.4 obtenemos

$$R \stackrel{a,D}{\longmapsto}_{p(R,D)} R/(a,D)$$

Ocultación: Este es el caso que justifica la utilización del nuevo orden estructural para basar la inducción, puesto que en los casos anteriores podríamos

habernos limitado a utilizar la correspondiente hipótesis de inducción respecto al orden estructural ordinario.

Consideraremos un proceso de la forma $P\backslash(a,q)$. Entonces, si $P \stackrel{a,A}{\longmapsto}_{p_A} P_{a,A}$ tenemos que $P < P\backslash(a,q)$ y $P_{a,A}\backslash(a,q) < P\backslash(a,q)$. En consecuencia, podemos aplicar la hipótesis de inducción al proceso P, obteniendo que $P/(a,A) = P_{a,A}$, lo que nos permite utilizar en lo sucesivo esta notación. Se tiene además

$$\begin{array}{lll} P \xrightarrow{A}_{p_A} & \Leftrightarrow & p(P,A) = p_A \\ P/(a,A)\backslash(a,q) \xrightarrow{B}_{p_B} & \Leftrightarrow & p(P/(a,A)\backslash(a,q),B) = p_B \end{array}$$

Consideremos por otra parte las distintas transiciones del proceso $P\backslash(a,q)$, que nos definen las probabilidades parciales con las que el mismo alcanza cada estado. Tenemos en primer lugar aquéllas que se obtienen aplicando la regla $EO1$, las cuales para cada $B \subseteq \Sigma - \{a\}$ para el que se tenga una transición del proceso P, nos conducen a

$$P\backslash(a,q) \xrightarrow{B}_{p(P,B)}$$

Por su parte, la regla $EO2$ nos permite generar las distintas transiciones de la forma

$$P\backslash(a,q) \xrightarrow{B}_{p(P,A)\cdot q\cdot p(P/(a,A)\backslash(a,q),B)}$$

cada una de las cuales corresponde a la ejecución de la acción ocultada a, en cada estado A con $a \in A$. En esta ocasión utilizamos la hipótesis de inducción para el proceso $P/(a,A)$.

Finalmente la regla $EO3$ corresponde a las situaciones en las que en principio no se ejecuta la acción ocultada, pero se deja abierta la posibilidad de hacerlo si ello se hace necesario. Su aplicación nos permite alcanzar el estado B por medio de una transición diferente para cada par de estados A y C tales que $(A - \{a\}) \cup C = B$, de la forma

$$P\backslash(a,q) \xrightarrow{B}_{p(P,A)\cdot(1-q)\cdot p(P/(a,A)\backslash(a,q),C)}$$

Globalizando todas las transiciones del proceso resultado de la ocultación, obtenemos

$$P\backslash(a,q) \xrightarrow{B}_{p}$$

donde p es el valor dado por

$$p = p(P,B) + q \cdot \sum_{A \subseteq \Sigma} p(P,A) \cdot p(P/(a,A)\backslash(a,q), B) +$$

$$(1-q) \cdot \sum_{A \subseteq \Sigma} p(P,A) \cdot \sum_{A-\{a\} \cup C = B} p(P/(a,A)\backslash(a,q), C)$$

Así, en definitiva podemos concluir

$$P\backslash(a,q) \xrightarrow{A}_{p_A} \Leftrightarrow p(P\backslash(a,q), A) = p_A$$

Análogamente obramos en lo referente al estudio de las continuaciones. Al efecto, tenemos que por hipótesis de inducción se satisface

$$P \overset{b,A}{\longmapsto}_{p(P,A)} P/(b,A)$$
$$P/(a,A)\backslash(a,q) \overset{b,B}{\longmapsto}_{p(P/(a,A)\backslash(a,q))} P/(a,A)\backslash(a,q)/(b,B)$$

junto con las hipótesis correspondientes a la probabilidad de alcanzar cada estado, ya enunciadas anteriormente.

Consideremos entonces las transiciones del proceso $P\backslash(a,q)$. En primer lugar tenemos aquellas que son obtenibles aplicando la regla $O1$. En concreto, para cada $a \notin B$ tenemos

$$P\backslash(a,q) \xrightarrow{b,B}_{p(P,B)} P/(b,B)\ \backslash(a,q)$$

Aplicando la regla $O2$, se obtienen las transiciones de la forma

$$P\backslash(a,q) \xrightarrow{b,B}_{q \cdot p(P,A) \cdot p(P/(a,A)\backslash(a,q),B)} P/(a,A)\backslash(a,q)/(b,B)$$

En cuanto a la regla $O3$, su aplicación produce las transiciones

$$P\backslash(a,q) \overset{b,A-\{a\}\cup B}{\longrightarrow}_{(1-q) \cdot p(P,A) \cdot p(P/(a,A)\backslash(a,q),B)} P/(b,A)\ \backslash(a,q)$$

Y, finalmente, con la regla $O4$ obtenemos

$$P\backslash(a,q) \overset{b,A-\{a\}\cup B}{\longrightarrow}_{(1-q) \cdot p(P,A) \cdot p(P/(a,A)\backslash(a,q),B)} P/(a,A)\backslash(a,q)/(b,B)$$

Mediante la globabilización de todas ellas obtenemos el proceso derivado correspondiente, que sería el definido por

$$[\tfrac{p(P,B)}{p_B}]P/(b,B)\backslash(a,q)$$
$$\sqcap \prod_{a\in B\subseteq\Sigma} [\tfrac{q\cdot p(P,B)\cdot p(P/(a,B)\backslash(a,q),A)}{p_B}]P/(a,B)\backslash(a,q)/(b,A))$$
$$\sqcap \prod_{a\in B\subseteq A\cup\{a\}} [\frac{(1-q)\cdot p(P,B)\cdot \sum_{B\cup C=A} p(P/(a,B)\backslash(a,q),C)}{p_B}]$$
$$\begin{cases} P/(b,B)\backslash(a,q) & \text{Si } b\in B \\ P/(a,B)\backslash(a,q)/(b,C) & \text{Si } b\notin B \end{cases}$$

que es equivalente al proceso $P\backslash(a,q)/(b,B)$ definido en el capítulo 4, por lo que concluimos

$$P\backslash(a,q)\overset{b,B}{\longmapsto}_{p_B} P\backslash(a,q)/(b,B)$$

□

Seguidamente extenderemos este resultado a los procesos infinitos, para ello en primer lugar definimos la noción de *orden sintáctico* entre procesos.

Definición 7.10 (Orden sintáctico)
Definimos el orden sintáctico entre procesos, que denotaremos con $\sqsubseteq_S$, a la menor relación de orden, congruente con los operadores PCSP, y tal que verifica

$$\forall P\in PCSP \ : \ DIV \sqsubseteq_S P$$

□

Dicha relación cumple el resultado siguiente:

Lema 7.4 Sea P un proceso finito, y sean $P\xrightarrow{A}_p$ y $P\xrightarrow{a,A}_p Q$ transiciones del mismo correspondientes a su semántica operacional. Entonces, para todo proceso P' que verifique $P\sqsubseteq_S P'$, podemos repetir las derivaciones de las mismas, obteniendo $P'\xrightarrow{A}_p$ y $P'\xrightarrow{a,A}_p Q'$ con $Q\sqsubseteq_S Q'$.

Demostración: Por inducción respecto de la profundidad de la derivación, se sigue fácilmente, teniendo en cuenta que el orden sintáctico es congruente respecto de las operaciones sintácticas del lenguaje PCSP. □

Lema 7.5 Sea P un proceso PCSP para el que se tiene una transición cualquiera de una de las formas

$$P \xrightarrow{A}_p , \quad P \xrightarrow{a,A}_p Q$$

en cuya derivación se ha utilizado en total un número i de veces las reglas de recursión (R o ER). Entonces, para toda aproximación finita P^k con $k \geq i$ es posible repetir la derivación en cuestión, omitiendose, por innecesaria, la aplicación de las reglas de recursión, obteniéndose la correspondiente transición

$$P^k \xrightarrow{A}_p , \quad P^k \xrightarrow{a,A}_p Q^{k*}$$

donde $Q^{k-i} \sqsubseteq_S Q^{k*} \sqsubseteq_S Q^k$.

Demostración: Procederemos por inducción sobre la profundidad de las derivaciones. El caso base es aquel en el que se ha utilizado tan sólo una regla, y por tanto corresponde a los operadores $STOP$ o prefijo, para los cuales el resultado es trivial.

Centremonos en primer lugar, en las transiciones del primer tipo, y distingamos la última regla aplicada en su derivación.

Si la regla ha sido ER, entonces el proceso es $P = \mu X.E(X)$, y por tanto como premisa en la aplicación de la regla tenemos

$$E[\mu X.E(X)|X] \xrightarrow{A}_p$$

Entonces, aplicando la hipótesis de inducción, tenemos que $\forall k \geq i-1$ se puede derivar

$$E[\mu X.E(X)|X]^k \xrightarrow{A}_p$$

En consecuencia obtenemos

$$E^{k+1} \xrightarrow{A}_p$$

es decir

$$P^{k+1} \xrightarrow{A}_p$$

o lo que es lo mismo, $\forall k \geq i$

$$P^k \xrightarrow{A}_p$$

Si la última regla aplicada es distinta de ER, el razonamiento es similar en todos los casos, por lo que nos centraremos tan sólo en uno de ellos. Tomemos,

por ejemplo, un proceso de la forma $P\backslash(a,q)$, y supongamos que la última regla aplicada ha sido EO3. Entonces, para el proceso P habremos de haber obtenido las derivaciones

$$P \xrightarrow{a,A}_p P'$$
$$P'\backslash(a,q) \xrightarrow{B}_r$$

por lo que, por hipótesis de inducción, podemos concluir que si j es el número de veces que se han utilizado las reglas R o ER en la derivación de la primera de dichas transiciones, para cada $k \geq j$ podemos derivar

$$P^k \xrightarrow{a,A}_p P'^{k*}$$

para un cierto P'^{k*} con $P'^{k-j} \sqsubseteq_S P'^{k*}$.

Análogamente, para la segunda de dichas transiciones, en cuya derivación se habrán utilizado un total de $i-j$ veces las reglas de recursión, y teniendo en cuenta que $k \geq i$ implica $k-j \geq i-j$, la hipótesis de inducción nos conduce a

$$P'\backslash(a,q)^{k-j} \xrightarrow{B}_r$$

o lo que es lo mismo

$$P'^{k-j}\backslash(a,q) \xrightarrow{B}_r$$

Entonces, aplicando el lema 7.4 obtenemos

$$P'^{k*}\backslash(a,q) \xrightarrow{B}_r$$

por lo que podemos aplicar la regla EO3, concluyendo

$$P\backslash(a,q)^k \xrightarrow{B}_r$$

por medio de la misma derivación que nos condujo a la transición de partida, en la que omitimos, por innecesarias, las aplicaciones de las reglas de la recursión.

Pasando a las transiciones del segundo tipo, distinguiremos de nuevo la última regla aplicada en la derivación. Entonces, si se trata de la regla R, tenemos, al igual que en el caso anterior, $P = \mu X.E(X)$, por lo que la aplicación de dicha regla se producirá sobre la derivación

$$E[\mu X.E(X)|X] \xrightarrow{a,A}_p Q$$

en la que se han aplicado un total de $i-1$ veces las reglas de recursión. Entonces, por hipótesis de inducción $\forall k \geq i-1$

$$E[\mu X.E(X)|X]^k \xrightarrow{a,A}_p Q^{k*}$$

con $Q^{k-i} \sqsubseteq_S Q^{k*}$, o lo que es igual

$$E^{k+1} \xrightarrow{a,A}_p Q^{k*}$$

y por lo tanto

$$P^{k+1} \xrightarrow{a,A}_p Q^{k*}$$

Entonces, $\forall k \geq i$, aplicando i veces las reglas de recursión, obtenemos

$$P^k \xrightarrow{a,A}_p Q^{k*}$$

con $Q^{k-i} \sqsubseteq_S Q^{k*}$ como pretendíamos. De manera análoga obtendríamos $Q^{k*} \sqsubseteq_S Q^k$.

Por otra parte, si la última regla aplicada no es R, centrémonos de nuevo en un caso cualquiera por ser todos ellos similares. En concreto, supongamos que el punto de partida es de nuevo un proceso de la forma $P\backslash(a,q)$, para el cual se ha obtenido una transición utilizando en último lugar la regla O4. Entonces, para la primera de sus premisas será de la forma

$$P \xrightarrow{a,A}_p P'$$

y en su derivación se habrán aplicado un total de $j \leq i$ veces las reglas de recursión. Entonces, por hipótesis de inducción, para $k \geq j$

$$P^k \xrightarrow{a,A}_p P'^{(k)*}$$

con $P'^{k-j} \sqsubseteq_S P'^{k*}$

La segunda de las premisas, de la forma

$$P'\backslash(a,q) \xrightarrow{b,B}_r Q$$

habrá precisado $i-j$ aplicaciones de las reglas de recursión en su derivación, por lo que por hipótesis de inducción, para $k \geq i$

$$P'^{k-j}\backslash(a,q) \xrightarrow{b,B}_r Q^{(k-j)*}$$

con $Q^{k-i} \sqsubseteq_S Q^{(k-j)*}$.

Entonces, por el lema 7.4 tenemos

$$P'^{k*}\backslash(a,q) \xrightarrow{b,B}_{\tau} Q^{k**}$$

donde Q^{k**} verifica $Q^{(k-j)*} \sqsubseteq_S Q^{k**}$.

Por lo que estamos en disposición de aplicar la regla O4, lo que nos conduce a

$$P\backslash(a,q)^{k} \xrightarrow{b,A-\{a\}\cup B}_{pr(1-q)} Q^{k**}$$

con $Q^{k-i} \sqsubseteq_S Q^{k**}$. □

El resultado obtenido en el teorema 7.1 para los procesos finitos puede ser extendido a la totalidad de los procesos PCSP, como se muestra en el siguiente

Teorema 7.2 Sea $P \in PCSP$, un proceso para el cual se pueden derivar transiciones globales de la forma

$$P \xrightarrow{A}_{p}$$
$$P \xrightarrow{a,A}_{p} Q$$

a partir de las reglas de la semántica operacional (incluyendo las recursiones). Entonces $p = p(P,A)$, siendo éste el valor obtenido en la semántica de pruebas, y Q es equivalente a $P/(a,A)$ respecto al paso de tests.

Demostración: En lo que hace referencia a las transiciones del primer tipo, tenemos que la probabilidad de la transición global en cuestión

$$P \xrightarrow{A}_{p}$$

se habrá obtenido por medio del sumatorio $p = \sum_{i \in I} p_i$, en el que se suman las probabilidades de las transiciones parciales de la forma

$$P \xrightarrow{A}_{p_i}$$

derivables en el marco de la semántica operacional.

Por su parte, es evidente que toda transición de cada aproximación finita P^n de P se corresponde con una transición obtenible a partir de P, en cuya derivación

aparecerán aplicaciones de las reglas de recursión, cuando así se precise. En consecuencia, para las correspondientes transiciones globales

$$P^n \xrightarrow{A}_{p_n}$$

se verifica $p_n \leq p$.

Por otra parte, en virtud del lema 7.5, cada transición de P podrá ser derivada, por medio de la misma derivación (excluyendo las aplicaciones de las reglas de la recursión, que no serán precisas) por parte de todas y cada una de las aproximaciones de P a partir de un determinado índice.

En definitiva, si denotamos por I_n al subconjunto de I al que corresponden las transiciones de P^n, se tendrá que $\{I_n\}$ converge a I, y por tanto $\{\sum_{i \in I_n} p_i\}$ converge a p, por lo que $\{p_n\}$ converge a p.

Pero como quiera que los P^n son procesos finitos, en virtud del teorema 7.1 tenemos que cada p_n coincide con $p(P^n, A)$. Por lo tanto, al ser la semántica de pruebas continua, tenemos

$$p = \lim_n p_n = \lim_n p(P^n, A) = p(P, A)$$

En cuanto a las transiciones del segundo tipo, sabemos que el proceso Q en cuestión es el dado por

$$Q = \bigsqcap_{i \in I} [p_i]Q_i$$

donde $\{P \xrightarrow{a,A}_{p_i} Q_i \mid i \in I\}$ es el multiconjunto de transiciones del proceso P. Utilizando la notación y los razonamientos desarrollados en la parte anterior de la demostración, consideraremos la familia de subconjuntos finitos de I, $\{I_k\}_{k \in \mathbb{N}}$, cada uno de los cuales está formado por las transiciones de dicho conjunto en cuya derivación se hayan hecho un máximo de k aplicaciones de las reglas de recursión. En virtud del lema 7.5, tenemos entonces que para cada $k \in \mathbb{N}$ y cada $i \in I_k$ se verifica para todo $k' \geq k$ que

$$P^{k'} \xrightarrow{a,A}_{p_i} Q_i^{k'*}$$

para un cierto $Q^{k'*}$ tal que $Q^{k'-k} \sqsubseteq_S Q^{k'*} \sqsubseteq_S Q^{k'}$.

Se tiene entonces, que Q es el límite de la sucesión $\{ \bigsqcup_{k' \geq k} \prod_{i \in I_k} [p_i] Q_i^{k'*} \}_{k \in \mathbb{N}}$. Para comprobarlo, tenemos en primer lugar que de manera trivial, para cada $k \in \mathbb{N}$ se satisface

$$\bigsqcup_{k' \geq k} \prod_{i \in I_k} [p_i] Q_i^{k'*} \sqsubseteq_S \bigsqcup_{j \geq k} \prod_{i \in I_j} [p_i] Q_i^j = Q$$

En lo que a la relación recíproca se refiere, tenemos que para cada $j \geq k$ y para cada $i \in I_j$ en virtud del lema 7.5, para todo $k' \geq j$ se tienen también transiciones de la forma $P^{k'} \xrightarrow{a,A}_{p_i} Q_i^{k'*}$ con $Q_i^{k'-k} \sqsubseteq_S Q_i^{k'*}$. En particular ello se tiene para $k' = 2j$, de donde

$$\prod_{i \in I_j} [p_i] Q_i^j \sqsubseteq_S \prod_{i \in I_j} [p_i] Q_i^{(2j)*}$$

y tomando límites en el lado derecho de la desigualdad

$$\prod_{i \in I_j} [p_i] Q_i^j \sqsubseteq_S \bigsqcup_{j \geq k} \prod_{i \in I_j} [p_i] Q_i^{(2j)*}$$

por lo que tomando límites en el lado izquierdo, se obtiene

$$Q = \bigsqcup_{j \geq k} \prod_{i \in I_j} [p_i] Q_i^j \sqsubseteq_S \bigsqcup_{j \geq k} \prod_{i \in I_j} [p_i] Q_i^{(2j)*}$$

Por último, el resultado del enunciado del teorema, se obtiene como consecuencia del teorema 7.1.:

$$Q = \bigsqcup_{k' \geq k} \prod_{i \in I_k} [p_i] Q_i^{k'*} = \bigsqcup_{k \in \mathbb{N}} \prod_{i \in I_k} [p_i] P^k/(a, A) = P/(a, A)$$

□

Teorema 7.3 (Equivalencia entre el modelo operacional y el de pruebas)
Para todo par de procesos P, P' definidos con la sintaxis de PCSP, se tiene que los mismos son equivalentes respecto de la semántica operacional si y sólo si lo son respecto de la semántica de pruebas. Es decir

$$\forall P, P' \in PCSP: \quad P \equiv_T P' \Leftrightarrow P \equiv_O P'$$

Demostración: Razonaremos en base a la semántica denotacional, que ya sabemos que es equivalente a la de pruebas, y resulta más adecuada a la hora de

compararla con la semántica operacional, pues en cierta forma ambas se basan en las mismas ideas intuitivas.

Según el lema 7.5, si en el marco de la semántica operacional se tiene la transición $P \xrightarrow{a,A}_p Q$, entonces el proceso Q es equivalente respecto al paso de tests al proceso $P/(a,A)$, y p es igual a $p(P,A)$. Ahora bien, en el capítulo 5 hemos comprobado que los conceptos definidos por estas notaciones en el marco de la semántica de pruebas, coincidían con los conceptos correspondientes para la semántica denotacional, por lo que quedaba justificada la utilización de una misma notación en ambos casos. En consecuencia, y como quiera que ambas semánticas, denotacional y de pruebas, son equivalentes, concluímos que la semántica denotacional del proceso Q coincide con la del proceso $P/(a,A)$. Puesto que esto es cierto para todo par (a,A) con $a \in A$, tenemos que el primer nivel del árbol que define la semántica operacional de P se corresponde con el primer nivel (entendiendo por tales las agrupaciones de un nodo de elección externo, junto con todos los internos que cuelgan de él) del árbol que define la semántica denotacional.

Pero como quiera que lo afirmado es cierto para cualquier proceso, de igual forma podemos concluir que cada uno de los nodos de los niveles sucesivos de ambos árboles, denotacional y operacional, se corresponden de igual forma, de donde se sigue que ambos árboles son equivalentes módulo la distinta forma sintáctica en que están definidos, y por tanto las semánticas denotacional y operacional del proceso P en cuestión, son equivalentes.

□

Capítulo 8

Conclusiones

Nuestro principal objetivo en este trabajo ha sido la definición y estudio de un lenguaje apropiado para la especificación de procesos concurrentes que combine las características proporcionadas por los lenguajes clásicos, junto a otras de tipo cuantitativo probabilístico.

El punto de partida fue el lenguaje CSP, y la primera aproximación siguió las ideas expuestas en [GJS90], encontrándonos con dos dificultades que hubo que superar. En primer lugar, el lenguaje presentado en dicho trabajo era sensiblemente más sencillo de estudiar por medio de una semántica operacional, debido a que se trataba de un lenguaje síncrono y utilizaba un único tipo de elección que puede ser interpretada ya sea de manera reactiva o generativa, sin necesidad de combinar ambas aproximaciones en un modelo único.

Además, hemos pretendido mantener, en la medida de lo posible, la interpretación usual de los operadores de CSP, al definir la semántica de los correspondientes operadores de nuestro lenguaje.

En particular, la combinación de los modelos reactivo y generativo ha sido el principal obstáculo para modelar el operador de elección externa, mientras que en el caso de la ocultación el problema se ha presentado al intentar mantener su significado clásico.

La resolución de estas dificultades ha exigido una cierta complejidad en lo que

se refiere a las definiciones semánticas de algunos operadores, principalmente las de la elección externa y la ocultación. En lo que respecta al operador de composición paralela, su modelado no ha presentado dificultades adicionales, pues la idea de reducir el mismo a una elección externa a través de la correspondiente ley de expansión, sigue siendo aplicable en el caso probabilístico.

Sin embargo, en contraste con las dificultades encontradas a la hora de definir la semántica de los operadores, el sistema de axiomas correspondiente ha resultado ser de mayor sencillez que en el caso no probabilístico, lo que también queda de manifiesto en el dominio soporte de la semántica denotacional que hemos definido, o lo que viene a ser lo mismo a la postre, en la sencillez de las formas normales de los procesos. Al respecto de las mismas es reseñable también el hecho de que otros trabajos ([HJ90, Han91, Han92]) hayan utilizado una clase de procesos muy similar a la de aquéllas, en los que se exige la aparición en alternancia de elecciones internas de tipo probabilístico y elecciones externas de tipo no-determinista. Esta conexión entre ambos modelos ha aparecido de forma espontánea, sin haber ser sido perseguida ex-profeso. No obstante, entendemos que la misma no es en absoluto casual; de hecho, consideramos que el resultado que hemos obtenido, justifica en buena medida el tomar como punto de partida a la hora de definir un dominio de procesos probabilísticos, una clase de procesos como los considerados en los citados trabajos, en los que de hecho una justificación de la elección de dicha clase se echaba notablemente en falta.

Concluiremos indicando algunas posibles vías de continuación del presente trabajo, en algunas de las cuales tenemos previsto centrar nuestra propia atención en un futuro próximo.

En primer lugar se podrían considerar extensiones del modelo, en las que se debería procurar de mantener los resultados alcanzados en nuestro trabajo. En particular, nosotros estamos especialmente interesados en la introducción del factor tiempo, lo cual nos conduciría a un modelo temporal-probabilístico de CSP.

Otra línea de trabajo, siguiendo la misma línea del trabajo de Hansson [Han91], consistiría en el estudio de diversos casos de aplicación del modelo. De hecho, un primer ejemplo en esta línea lo hemos presentado ya en este trabajo, donde hemos

analizado un sencillo protocolo de comunicaciones (protocolo AUY). Lamentablemente, la mayor complejidad y extensión de nuestros estudios teóricos, ha imposibilitado en nuestro caso la mayor profundización en el marco de este trabajo en el terreno de las aplicaciones.

Si bien las definiciones de la semántica de los distintos operadores por las que hemos optado no fueron en absoluto caprichosas ni precipitadas, sino fruto de múltiples disquisiciones y de la constatación de los problemas técnicos de otras opciones que a priori podían resultar más sencillas o naturales, sería posible ahondar en el estudio de diferentes versiones del modelo. En particular, el modelo se simplificaría sin duda alguna con una interpretación distinta del operador de ocultación, en la que la decisión de ejecutar o no una acción oculta se tomase una sóla vez, siendo imposible forzar al proceso a ejecutarla cuando éste haya decidido lo contrario. Naturalmente, y en línea con lo ya indicado, ello conllevaría la pérdida absoluta de buena parte de las propiedades básicas de dicho operador en el modelo no probabilístico.

Por otra parte, resultaría interesante tratar de trasvasar nuestros resultados a un lenguaje basado en CCS, lo que ofrecería la ventaja de poder comparar más fácilmente los mismos con los obtenidos en otros trabajos relacionados de otros investigadores. La dificultad que encierra dicha tarea parece ser de orden similar a aquélla con la que nosotros hemos debido de luchar a lo largo del presente trabajo, pues si bien el operador de restricción de CCS es más simple que el de ocultación de CSP, y sólo existe un tipo de elección, tenemos, sin embargo, una mayor dificultad en el modelado del operador paralelo y en el tratamiento de la acción τ presente en el lenguaje CCS.

Por último, otras posibles continuaciones del trabajo se centrarían en el estudio desde un enfoque distinto de la semántica del lenguaje. Así, en lugar del enfoque fundamentalmente algebraico que nosotros hemos elegido, se podría hacer un mayor énfasis en las cuestiones puramente probabilísticas, en línea con lo hecho en [Sei92]. Un último punto relacionado con el anterior se centraría en el estudio de la semántica en el marco de los espacios métricos, siguiendo las ideas expuestas, si bien de forma muy somera, en [GJS90].

Bibliografía

[Abr88] D. Abramsky. *Observation Equivalence as a Testing Equivalence*, Theoretical Computer Science, 1988.

[AUY79] A.V.Aho, J.D.Ullman, M.Yannakakis. *Modeling Communication Protocols by Automata.* Proc. 20 th. IEEE Symp. on Foundations of Computer Science, 267-273,1979.

[BK84] J.A. Bergstra, J.W. Klop. *Algebra of Communicating Processes.* Technical Report CS-R8420. Centre for Mathematics and Computer Science. Departament of Computer Science, 1984.

[BM89] B. Bloom, A. R. Meyer. *A Remark on Bisimulation between Probabilistic Processes.* Logik at Botik, Meyer and Tsailin editors. Springer-Verlag, 1989.

[Bro83] Stephen D. Brookes *A Model For Communicating Sequential Processes.* Ph. D. Thesis, Oxford University, 1983.

[BR85] S. D. Brookes, A. W. Roscoe *An Improved Failures Model for Communicating Processes.* LNCS 197. Springer-Verlag. 1985.

[Broy92] M. Broy *Operational and denotational semantics with explicit concurrency.* Fundamenta Informaticae, 16, pp. 201-229. 1992.

[Chr89] Ivan Christoff. *Distinguishing Probabilistic Processes through Testing.* Proc. Nordic Workshop on Program Correctness, 1989.

[Chr90A] Ivan Christoff. *Testing Equivalences for Probabilistic Processes.*, Ph. D. Thesis, Dep. of Computer Systems, Uppsala University, 1990.

[Chr90B] Ivan Christoff. *Testing Equivalences and Fully Abstract Models for Probabilistic Processes.*, CONCUR-90, LNCS 458, 1990.

[Chr93] Linda Christoff. *Specification and Verification Methods for Probabilistic Processes.*, Ph. D. Thesis, Dep. of Computer Systems, Uppsala University, 1993.

[CS87] G.Costa, C. Stirling *Weak and strong fairness in CCS.* Information and Computation 73, pp. 207-244. 1987.

[CFV92] F. Cuartero, D. de Frutos, V. Valero. *CSP probabilístico (PCSP): Un Modelo Operacional.* Actas XVIII Conferencia Latinoamericana de Informática PANEL'92. Las Palmas de Gran Canaria, 1992.

[Cua86] Fernando Cuartero. *Una aproximación a CSP.* Trabajo de investigación. Dpto. Informática y Automática, Fac. de Matemáticas, UCM, 1986.

[CM88] K. M. Chandy, J. Misra. *Parallel Program Design: A Foundation.* Addison-Wesley, 1988.

[DNM90] P. Degano, R. De Nicola, U. Montanari. *A partial ordering semantics for CCS.* Theoretical Computer Science, 75. pp. 223-262. 1990.

[Dij65] E. W. Dijkstra. *Cooperating Sequential Processes.* Technical Report EWD-123, Eindhowen, The Netherlands, 1965.

[Fra86] N. Francez. *Fairness.* Springer-Verlag, 1986.

[GJS90] A. Giacalone, C.-C. Jou, S.A. Smolka. *Algebraic Reasoning for Probabilistic Concurrent Systems.* Proceedings of Working Conference on Programming Concepts and Methods. IFIP TC 2, Sea of Galilee, Israel, 1990.

[GSST90] R. van Glabbek, S.A. Smolka, B.U.Steffen, C.M.N.Tofts, *Reactive, Generative and Stratified Models of Probabilistic Processes.* Proceedings of 5th Annual IEEE Symposium on Logic in Computer Science, 130-141, Philadelphia, PA, 1990.

[Han91] Hans A. Hansson. *Time and Probability in Formal Design of Distributed Systems.* Ph. D. Thesis. Dep. of Computer Systems. Uppsala University, 1991.

[Han92] Hans A. Hansson. *Modeling Timeouts and Unreliable Media with a Timed Probabilistic Calculus.* Formal Description Techniques IV, North Holland, 1992

[HJ90] Hans Hansson, Bengt Jonsson. *A Calculus for Communicating Systems with Time and Probabilities.* Proc. Real-Time Systems Symposium. Florida, 1990.

[Hen85] M. Hennessy. *Aceptance Trees.* JACM, 32, n. 4, pp. 896-928. 1985.

[HM85] M. Hennessy, R. Milner. *Algebraic Laws for Nondeterminism and Concurrency.* JACM, 32, n. 1, pp. 137-161. 1985.

[Hen88] M. Hennessy. *Algebraic Theory of Processes.* The MIT Press, Cambridge, 1988.

[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, Englewood Cliffs, NJ, 1985.

[Her90] T. Hermann. *Probabilistic Self-Stabilization.* Infrmation Processing Letters, 1990.

[JP90] C. Jones, G. Plotkin. *A Probabilistic Powerdomain of Evaluations.* Proc. 16th Annual Symposium on Logic in Computer Science, 1989.

[JS90] C.-C. Jou, S.A. Smolka. *Equivalences, Congruences and Complete Axiomatizations for Probabilistic Processes.* CONCUR-90, LNCS 458, 1990.

[LS89] K.G.Larsen, A.Skou *Bisimulation through Probabilistic Testing.* Proceedings of 16th Annual ACM Symposium on Principles of Programming Languages, 1989.

[LQ91] D. Latella, P. Quaglia. *A probabilistic semantics for LOTOS.* Internal Report of LOTOSPHERE consortium. Sept 1991.

[Mig91a] Carlos Miguel. *State of the Art on Timed and Probabilistic Models.* Internal Report DIT - UPM, 1991.

[Mig91b] Carlos Miguel. *Técnicas de descripción formal aplicadas a la evaluación de prestaciones de sistemas de comunicación.* Tesis Doctoral, ETSI Telecomunicación, Universidad Politécnica de Madrid, 1991.

[Mil80] R. Milner. *A Calculus of Communicating Systems.* LNCS 92, Springer-Verlag, 1980.

[Mil83] R. Milner. *Calculi for Synchrony and Asynchrony.* Theoretical Computer Science, 25:267-310, 1983.

[Mil89] R. Milner. *Communication and Concurrency.* Prentice-Hall International, 1985.

[MT90] F. Moller, C. Tofts. *A Temporal Calculus of Communicating Systems.* Proc. CONCUR 90, LNCS 458, pp. 401-415. 1990.

[NH84] R. De Nicola, M. Hennessy. *Testing Equivalences for Processes.* Theoretical Computer Science 34, 1984.

[Nic85] R. De Nicola. *Two Complete axiom Systems for a Theory of Communicating Sequential Processes.* Information and Control 64 pp. 136-172. 1985.

[Old87] E. R. Olderog. *Theory of Communicating Processes.* Advances in Petri Nets. LNCS 255. Springer-Verlag, 1987.

[Ort90] Y. Ortega. *En busca del tiempo perdido.* Tesis Doctoral, Dpto. Informática y Automática. U. Complutense de madrid, 1990.

[Par81] D. M. R. Park. *Concurrency and Automata on Infinite Sequences.* Proc. 5th G.I. Conference on Theretical Computer Science, pp. 167-183. Springer-Verlag, 1981.

[Pet81] J.L. Peterson. *Petri Net Theory and the Modeleng of Systems.* Englewood Cliffs, New Jersey: Prentice Hall, Inc. 1981.

[Phi85] I. Phiplips. *Refusal Testing.* TCS, 50. North Holland. 1985

[Plo81] G.D. Plotkin. *A structural approach to operational semantics.* Technical Report DAIMI FN-19, Aarhus University. 1981.

[Pnu85] A. Pnueli. *Linear and Branching Structures in the Semantics and Logics of Reactive Systems.* 12th ICALP, LNCS 194. 1985.

[PZ86] A. Pnueli, L. Zuck. *Verification of Multiprocess Probabilistic Protocols.* Distributed Computingl. 1986.

[Pu90] W. Pugh. *Skip Lists: A Probabilistic Alternative to Balanced Trees.* Communications of the ACM. June, 1990.

[Rei85] W. Reisig. *Petri Nets.* EATCS Monographs on Theoretical Computer Science, 4. Springer-Verlag, 1985.

[RR87] G.M. Reed, A.W. Roscoe. *Metric Spaces as Models for Real-Time Concurrency.* LNCS 298. Springer-Verlag, 1987.

[RR88] G.M. Reed, A.W. Roscoe. *A Timed Model for Communicating Sequential Processes.* Theoretical Computer Science, 58, pp. 249-261. 1988.

[RZ92] J.J.M.M. Rutten, J.I. Zucker. *A semantics approach to fairness.* Fundamenta Informaticae XVI, pp. 1-38. 1992.

[Sei92] Karen Seidel. *Probabilistic CSP.* Ph. D. Thesis (Draft Version), Programming Research Group, Oxford, Jan. 1992.

[SS90] S.A.Smolka, B.U.Steffen, *Priority as Extremal Probability.* CONCUR-90, LNCS 458, 1990.

[Tau89] D. Taubner, W. Vogler. *Step failures semantics and a complete proof system.* Acta Informatica, 27, pp. 125-126. 1989.

[Tof88] C. Tofts. *Temporal Ordering for Concurrency.* LFCS Report Series. Univ. Edinburgh (UK), 1988.

[Win87] G. Winskel *Event Structures.* Advances in Petri Nets 86. LNCS 255. Springer-Verlag, 1987.

[Zic90] John J. Zic. *Some thoughts on Communication System Performance Specification.* Proc. Open Distributed Processing Whorkshop, Sydney 1990.