

Un Ejemplo de Evaluación Formal de Prestaciones Mediante el Álgebra de Procesos Estocásticos: ROSA *

Fernando L. Pelayo Fernando Cuartero Valentín Valero Diego Cazorla
*Departamento de Informática
Escuela Politécnica de Albacete
Universidad de Castilla-La Mancha
Campus Universitario s/n. 02071. Albacete, España
Teléfono: +34 967 599200
Fax +34 967 599224
{fpelayo, fernando, valentin, dcazorla}@info-ab.uclm.es*

Resumen

En este artículo presentamos un lenguaje algebraico para la descripción de procesos estocásticos, junto con probabilidades y no-determinismo, el cual permite tanto evaluar algunos índices de rendimiento como verificar otro tipo de restricciones temporales: **ROSA** (**R**easoning **O**n **S**tochastic **A**lgebras). Como ejemplo de aplicación, se analiza el Protocolo del Bit Alternante, obteniendo el tiempo medio necesario para el envío de un mensaje, teniendo en cuenta que los canales pueden fallar con una probabilidad conocida.

Palabras Clave: Sistemas Distribuidos y Paralelismo, Teoría de Computación y Algoritmos.

1 Introducción

Las Álgebras de Procesos, usadas en el diseño y análisis de Sistemas Concurrentes (Sistemas Distribuidos, S. en Tiempo Real, S. Tolerantes a Fallos, etc.), han sido objeto de diversas formalizaciones. En la última década, además, aspectos cuantitativos del diseño de sistemas, como pueden ser tiempo, probabilidades y prioridades han adquirido bastante relevancia.

En [2], podemos encontrar una primera aproximación a incorporar probabilidades en Álgebras de Procesos, de forma que se introduce el concepto de experimento probabilístico. En [7] se define PCCS, un lenguaje algebraico basado en la versión síncrona de CCS, en el cual reemplazamos el operador de elección no-determinista por el operador de elección probabilística, $\sum[p_i]E_i$, donde p_i es la probabilidad con la cual el proceso se comporta como E_i . En [13] el modelo es extendido, mediante la presentación de las interpretaciones reactiva, generativa y estratificada para procesos concurrentes probabilísticos.

Podemos encontrar en [8] la definición de TPCCS *Timed Probabilistic Calculus of Communicating Systems*, el cual extiende CCS con probabilidades y tiempo. Dicho lenguaje presenta operadores de elección probabilística y de elección no-determinista de forma alternativa, es decir, en cada estado podemos encontrar o bien una elección probabilística o bien una elección no-determinista. Debido a esto el modelo se llama “modelo alternante”.

Asimismo, podemos encontrar extensiones de CSP en [4] y en [5], en las cuales se considera una interpretación reactiva de las probabilidades, mientras que en [11] se estudia una

*Este trabajo ha sido financiado por el proyecto CICYT “Desarrollo Formal de Sistemas Distribuidos”, TIC97-0669-c03-02.

interpretación generativa. En ambos modelos se presenta una Semántica de Pruebas que es completamente abstracta respecto a la Semántica Denotacional basada en Árboles de Aceptación.

Podemos observar que en todos estos modelos probabilísticos subyace la idea de modelar un comportamiento no-determinista por medio de un operador probabilístico, para lo cual es necesario cuantificar todos los comportamientos posibles del sistema mediante una probabilidad. Obviamente, esto requiere un completo conocimiento cuantitativo sobre el comportamiento del sistema, que desafortunadamente no siempre es posible, aunque a veces puede ser estimado [9].

Es sabido que los sistemas concurrentes pueden tener comportamientos probabilísticos y comportamientos no-deterministas, por lo tanto ambos deben ser capturados por los modelos que pretendan describirlos. Por ejemplo, un canal de comunicaciones puede fallar con una probabilidad conocida, asimismo, no es predecible el orden de ejecución de acciones involucradas en procesos paralelos (interleaving). Es lógico esperar de cualquier modelo que lo describa, que capture ambos comportamientos fielmente.

Siguiendo estos requerimientos, en [14] se presenta una elección probabilística de CCS. Esta extensión dispone de un operador de elección probabilística: $\oplus_p$, con una interpretación generativa de la misma, y el operador de elección de CCS: $\oplus$, se mantiene pasando a representar el no-determinismo clásico. PCSP es una extensión probabilística de CSP presentada en [10], esta extensión mantiene un operador no-determinista. También se presenta una Semántica Operacional, la cual distingue tres tipos de nodos: nodos probabilísticos, nodos no-deterministas y nodos acción.

TPAL es un lenguaje con probabilidades y tiempo, basado en LOTOS, en [3] se puede observar que la elección interna originaria de LOTOS ha sido sustituida por la elección probabilística. En este lenguaje se impone una alternancia entre los términos probabilísticos y los no-deterministas.

También es deseable tener en cuenta aspectos de prestaciones en el diseño de sistemas concurrentes. Es conocido que con frecuencia, los aspectos de eficiencia no se tienen en consideración hasta que el sistema funciona correctamente en cuanto a los demás requerimientos, i.e., una vez que el sistema ha sido completamente probado. Sin embargo, es obvio que existen algunos sistemas (Sistemas de Tiempo Real) en los cuales necesitamos conocer a priori si el sistema satisfará determinadas restricciones temporales. En consecuencia, cualquier modelo que pretenda describir sistemas concurrentes, deberá incluir capacidades para determinar consideraciones de eficiencia.

En [6] y [1] se presentan sendas Álgebras de Procesos Estocásticos que incluyen dichas capacidades, esta última, *Extended Markovian Process Algebra (EMPA)*, tiene algunas características similares a la que presentamos, como la concepción de tipos distintos de acciones en base a su duración (modelada por variables aleatorias de tipo Exponencial), o la *race policy* (política de aceleración de acciones). Sin embargo, en **ROSA**, el parámetro temporal de cada acción no restringe sus capacidades de sincronización, a diferencia de lo que ocurre en EMPA. Fundamentalmente motivados por la claridad en el diseño de sistemas (siguiendo algunas ideas presentadas en [12]), en **ROSA** seguimos la política clásica de definir una sintaxis apropiada (distinta) para cada operación. Mas aún, el modelo que presentamos tiene diferentes interpretaciones semánticas a las de EMPA con respecto a algunos operadores, fundamentalmente el de composición paralela.

El trabajo se estructura como sigue: en la Sección 2 se define la Sintaxis de **ROSA**, su Semántica Operacional se presenta en la Sección 3 y en la Sección 4 describimos el algoritmo de evaluación de prestaciones. Finalmente, en la Sección 5 presentamos el caso de estudio: el Protocolo del Bit Alternante.

2 Sintaxis del Lenguaje

Sea $\Delta = \{a, b, c, \dots\}$ un conjunto de tipos de acciones. Sean p, q, r probabilidades y sean las letras griegas parámetros temporales de las acciones.

Los términos de **ROSA** están definidos por:

$$P ::= \mathbf{0} \mid a.P \mid \langle a, \lambda \rangle.P \mid P \oplus P \mid P + P \mid P \oplus_r P \mid P \parallel_A P \mid \text{rec} X : P$$

donde $r \in (0, 1)$, $\lambda \in \mathbb{R}^+ - \{0\}$, $A \subset \Delta$ y P es un proceso de **ROSA**.

Los operadores son interpretados como en las Álgebras de Procesos clásicas, esto es, $\text{accion}.P$ representa el Operador Prefijo, $\oplus$ y $+$ son respectivamente las elecciones interna y externa, $\parallel_A$ es el operador de sincronización en acciones pertenecientes al conjunto A , y $\text{rec} X : P$ es el operador de recursión.

Los nuevos operadores incluidos en este trabajo son:

- $\langle a, \lambda \rangle.P$, que representa el operador de prefijo temporizado, en este caso la acción etiquetada con a tiene una duración que sigue una distribución Exponencial de parámetro λ . Observemos que ahora $a.P$ es un caso particular de este operador considerando $\lambda = \infty$, lo que significa que $\mathbf{E}[Exp[\infty]] = 0$, i.e., esta acción tiene duración 0.
- $P \oplus_r Q$ denota el operador de elección probabilística con la clásica interpretación generativa: Con probabilidad r el proceso se comporta como P y con probabilidad $1 - r$ se comporta como Q .

3 Semántica Operacional

La semántica operacional está definida al estilo Plotkin y Milner (ver Tabla 1), por medio de un sistema de transiciones etiquetadas que pueden ser de dos tipos:

- Transiciones probabilísticas, de la forma siguiente:

$$P \xrightarrow{r, a, \lambda} Q$$

donde r es una probabilidad, a es un tipo de acción y λ es un parámetro de una distribución Exponencial que modela la duración de la acción. Esta regla modela que el proceso P puede evolucionar ejecutando la acción etiquetada con a , con probabilidad r y con una duración descrita por una distribución Exponencial con parámetro λ .

- Transiciones no-deterministas:

$$P \longrightarrow Q$$

Por medio de estas reglas capturamos las decisiones internas del sistema, resolviendo así las elecciones no deterministas.

Las reglas S1, S2, S3 y la regla S6 son inmediatas por lo que no necesitan comentarios.

Las reglas S4 capturan la semántica de la elección externa; concretamente las reglas S4a y S4b representan la evolución mediante la ejecución de acciones que solo están disponibles en uno de los componentes de la elección. Cuando una acción puede ser ejecutada por ambos argumentos, las reglas de la S4c a la S4h establecen la evolución de $P + Q$, distinguiéndose varios casos dependiendo de la duración de las acciones involucradas. Si en ambos componentes la acción es inmediata (tiene duración 0), la elección será resuelta de forma no-determinista (reglas S4c y S4d), para lo cual se evoluciona a un estado intermedio resolviendo el no-determinismo, para después ejecutar la acción correspondiente. En caso de que la acción no sea inmediata en al menos uno de los dos componentes, la elección se realizará de forma probabilística, o, no-determinista primero y probabilística después, atendiendo a la siguiente distinción de casos (las acciones inmediatas tienen un cierto carácter de urgencia):

(S1a) $\frac{}{a.P \xrightarrow{1,a,\infty} P}$	(S1b) $\frac{}{\langle a, \lambda \rangle . P \xrightarrow{1,a,\lambda} P}$
(S2a) $\frac{P \xrightarrow{q,a,\lambda} P'}{P \oplus_p Q \xrightarrow{p,q,a,\lambda} P'}$	(S2b) $\frac{Q \xrightarrow{t,a,\lambda} Q'}{P \oplus_p Q \xrightarrow{(1-p) \cdot t, a, \lambda} Q'}$
(S3a) $\frac{}{P \oplus Q \rightarrow P}$	(S3b) $\frac{}{P \oplus Q \rightarrow Q}$
(S4a) $\frac{P \xrightarrow{q,a,\lambda} P' \wedge a \notin \mathbf{Type}[\mathbf{D}[Q]]}{P + Q \xrightarrow{q,a,\lambda} P'}$	(S4b) $\frac{Q \xrightarrow{q,a,\lambda} Q' \wedge a \notin \mathbf{Type}[\mathbf{D}[P]]}{P + Q \xrightarrow{q,a,\lambda} Q'}$
(S4c) $\frac{P \xrightarrow{q,a,\infty} P' \wedge Q \xrightarrow{r,a,\infty} Q'}{P + Q \rightarrow P \xrightarrow{q,a,\infty} P'}$	(S4d) $\frac{P \xrightarrow{q,a,\infty} P' \wedge Q \xrightarrow{r,a,\infty} Q'}{P + Q \rightarrow Q \xrightarrow{r,a,\infty} Q'}$
(S4e) $\frac{P \xrightarrow{q,a,\lambda_1} P' \wedge Q \xrightarrow{r,a,\lambda_2} Q' \wedge \lambda_2 \neq \infty}{P + Q \rightarrow P \xrightarrow{q \cdot r, a, \lambda_1 + \lambda_2} P'}$	(S4f) $\frac{P \xrightarrow{q,a,\lambda_1} P' \wedge Q \xrightarrow{r,a,\lambda_2} Q' \wedge \lambda_1 \neq \infty}{P + Q \rightarrow Q \xrightarrow{q \cdot r, a, \lambda_1 + \lambda_2} Q'}$
(S4g) $\frac{P \xrightarrow{q,a,\lambda_1} P' \wedge Q \xrightarrow{r,a,\lambda_2} Q' \wedge \exists \lambda_i \neq \infty}{P + Q \xrightarrow{q \cdot (1-r), a, \lambda_1} P'}$	(S4h) $\frac{P \xrightarrow{q,a,\lambda_1} P' \wedge Q \xrightarrow{r,a,\lambda_2} Q' \wedge \exists \lambda_i \neq \infty}{P + Q \xrightarrow{(1-q) \cdot r, a, \lambda_2} Q'}$
(S5a) $\frac{P \xrightarrow{q,a,\lambda} P' \wedge a \notin \mathbf{Type}[\mathbf{D}[Q]] \cup A}{P \parallel_A Q \xrightarrow{q,a,\lambda} P' \parallel_A Q}$	(S5b) $\frac{Q \xrightarrow{q,a,\lambda} Q' \wedge a \notin \mathbf{Type}[\mathbf{D}[P]] \cup A}{P \parallel_A Q \xrightarrow{q,a,\lambda} P \parallel_A Q'}$
(S5c) $\frac{P \xrightarrow{q,a,\infty} P' \wedge Q \xrightarrow{r,a,\infty} Q' \wedge a \notin A}{P \parallel_A Q \rightarrow P'' \parallel_A Q \xrightarrow{q,a,\infty} P' \parallel_A Q}$	(S5d) $\frac{P \xrightarrow{q,a,\infty} P' \wedge Q \xrightarrow{r,a,\infty} Q' \wedge a \notin A}{P \parallel_A Q \rightarrow P \parallel_A Q'' \xrightarrow{r,a,\infty} P \parallel_A Q'}$
(S5e) $\frac{P \xrightarrow{q,a,\lambda_1} P' \wedge Q \xrightarrow{r,a,\lambda_2} Q' \wedge a \notin A \wedge \lambda_2 \neq \infty}{P \parallel_A Q \rightarrow P'' \parallel_A Q \xrightarrow{q \cdot r, a, \lambda_1 + \lambda_2} P' \parallel_A Q}$	(S5f) $\frac{P \xrightarrow{q,a,\lambda_1} P' \wedge Q \xrightarrow{r,a,\lambda_2} Q' \wedge a \notin A \wedge \lambda_1 \neq \infty}{P \parallel_A Q \rightarrow P \parallel_A Q'' \xrightarrow{q \cdot r, a, \lambda_1 + \lambda_2} P \parallel_A Q'}$
(S5g) $\frac{P \xrightarrow{q,a,\lambda_1} P' \wedge Q \xrightarrow{r,a,\lambda_2} Q' \wedge a \notin A \wedge \exists \lambda_i \neq \infty}{P \parallel_A Q \xrightarrow{q \cdot (1-r), a, \lambda_1} P' \parallel_A Q}$	(S5h) $\frac{P \xrightarrow{q,a,\lambda_1} P' \wedge Q \xrightarrow{r,a,\lambda_2} Q' \wedge a \notin A \wedge \exists \lambda_i \neq \infty}{P \parallel_A Q \xrightarrow{(1-q) \cdot r, a, \lambda_2} P \parallel_A Q'}$
(S5i) $\frac{P \xrightarrow{q,a,\lambda_1} P' \wedge Q \xrightarrow{r,a,\lambda_2} Q' \wedge a \in A}{P \parallel_A Q \xrightarrow{q \cdot r, a, \min\{\lambda_1, \lambda_2\}} P' \parallel_A Q'}$	(S6a) $\frac{P[recX : P/X] \xrightarrow{q,a,\lambda} P'}{recX : P \xrightarrow{q,a,\lambda} P'}$

Tabla 1: Semántica Operacional de **ROSA**

- Si ambos componentes cooperan para ejecutar la acción (reglas S4e y S4f), se da una transición no-determinista primero y una probabilística después, cuya probabilidad es el producto de las probabilidades de ejecución de la acción en ambos términos. El nuevo parámetro temporal es la suma de los parámetros originales, es decir, “se acelera” la ejecución de la acción (*race policy*).
- En otro caso, capturamos con las reglas S4g y S4h el caso en que solo una de las acciones coopera en la ejecución de la acción. Ahora se mantiene la duración de la acción que coopera como duración de la acción ejecutada. Naturalmente la probabilidad es el producto de la probabilidad de ejecución de la acción que coopera y el complementario de la probabilidad de ejecución de la acción que no coopera.

Con las reglas S5 se describe el comportamiento del operador de composición paralela, el cual es similar al de elección externa cuando la acción a ejecutar no pertenece al conjunto de sincronización. La regla S5i captura la mencionada excepción, y representa que la acción más lenta impone su duración sobre la otra, por medio del parámetro temporal de la acción ejecutada que es $\min\{\lambda_1, \lambda_2\}$.

Hemos usado dos funciones auxiliares que nos permiten hacer algunas distinciones necesarias para dar la Semántica, tales como que una acción este disponible para su ejecución inminente en un proceso, o el tipo de la misma, **D** y **Type** respectivamente:

D es una función definida sobre el conjunto de los procesos de **ROSA**, la cual asocia a cada proceso el multiconjunto de acciones disponibles para su ejecución inminente, y está definida por medio de inducción estructural sobre el conjunto de los procesos de **ROSA**:

$$\begin{array}{ll}
\mathbf{D} : \{\text{procesos de ROSA}\} & \rightarrow \mathcal{M}[\Delta \times \mathbb{R}^+ - \{0\}] \\
\mathbf{0} & \rightsquigarrow \emptyset \\
a.P & \rightsquigarrow \{a\} \\
\langle a, \lambda \rangle.P & \rightsquigarrow \{\langle a, \lambda \rangle\} \\
P \oplus Q & \rightsquigarrow \mathbf{D}[P] \cup \mathbf{D}[Q] \\
P + Q & \rightsquigarrow \mathbf{D}[P] \cup \mathbf{D}[Q] \\
P \oplus_r Q & \rightsquigarrow \mathbf{D}[P] \cup \mathbf{D}[Q] \\
P ||_A Q & \rightsquigarrow \mathbf{D}[P] \cup \mathbf{D}[Q] \\
recX : P & \rightsquigarrow \mathbf{D}[P]
\end{array}$$

Type es una función definida sobre el conjunto de los multiconjuntos de acciones, que asocia a cada multiconjunto el conjunto formado por el tipo de sus acciones:

$$\mathbf{Type} : \mathcal{M}[\Delta \times \mathbb{R}^+ - \{0\}] \longrightarrow \mathcal{P}[\Delta]$$

4 Evaluación de Prestaciones

A partir de la Semántica Operacional podemos construir para cada proceso su *grafo de transiciones*, que nos va a permitir calcular algunos índices de rendimiento por medio del algoritmo que describen los siguientes 4 pasos:

1. Comenzamos eliminando del *grafo de transiciones* las transiciones no deterministas, considerando así el no-determinismo del sistema, de forma que cada eliminación produce a partir del grafo que la contiene dos nuevos grafos, como puede verse en la Fig.1. De esta forma se obtiene un conjunto de *grafos de transición deterministas* que solo tienen transiciones probabilísticas.
2. A continuación reetiquetaremos cada transición $\xrightarrow{q,a,\lambda}$ con $\xrightarrow{q,1/\lambda}$, de cada uno de los *grafos de transición deterministas* generados en el paso anterior, obteniendo de esta forma los llamados *grafos temporales deterministas*, que solo contienen información sobre la duración media asociada a cada transición y su probabilidad, y que carecen de la información del tipo de acción ejecutada (Fig.1).

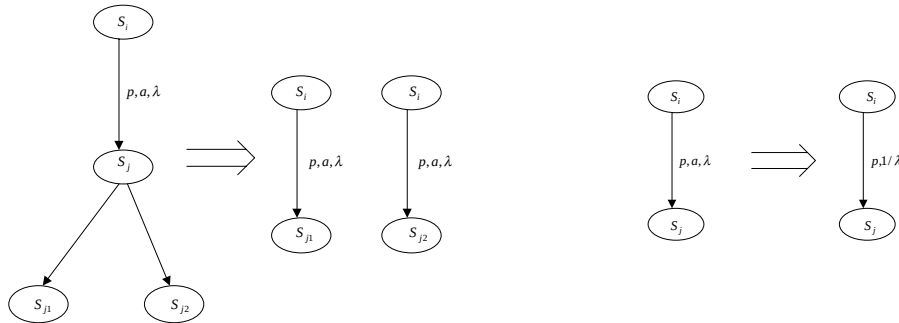


Figura 1: Algoritmo de Evaluación - Fases 1 y 2

3. Para cada uno de estos grafos se eliminan las transiciones con coste temporal nulo, por medio de la unión de los nodos que estas transiciones conectaban en uno solo etiquetado como cualquiera de los unidos (S_i o S_j) (Fig.2). La probabilidad del arco eliminado multiplicará a la de los arcos que parten del nuevo nodo $S_i \vee S_j$.
4. Una vez hemos completado las tres fases anteriores, hemos generado un conjunto de *grafos temporales reducidos deterministas* (correspondientes a las posibles evoluciones no-deterministas del sistema), para cada uno de los cuales podemos calcular el tiempo medio requerido para alcanzar un estado particular S_F desde el estado inicial S_0 (observemos que cada *grafo temporal reducido determinista* solo tiene un estado inicial, que marcamos con un doble óvalo y un único estado final, marcado con un doble rectángulo).

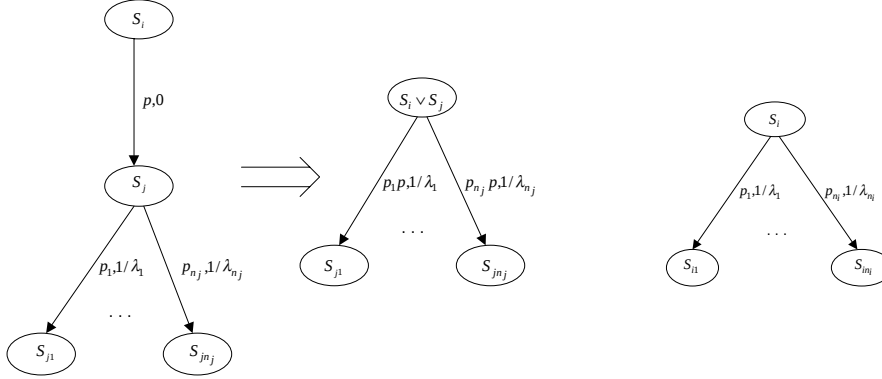


Figura 2: Algoritmo de Evaluación - Fases 3 y 4

Denotemos en cada *grafo temporal reducido determinista* como T_{S_i} el tiempo medio necesario para evolucionar desde el estado S_i al estado final (S_F).

Tenemos ($T_{S_i} = \sum_{j=1}^{n_i} p_j(1/\lambda_j + T_{S_{ij}})$) siendo n_i el número de arcos que parten del estado

S_i , y la etiqueta del arco que conecta el estado S_i con S_{ij} es $(p_j, 1/\lambda_j)$ como puede verse en la Fig.2.

Por lo tanto, el tiempo medio para alcanzar el estado final desde el estado inicial será T_{S_0} , y puede ser calculado desarrollando la ecuación recurrente descrita anteriormente.

Este algoritmo permite calcular tantos tiempos medios como diferentes formas “no-deterministas” existen de evolucionar desde el estado inicial hasta el final (de acuerdo a los *grafos temporales reducidos deterministas* obtenidos), este número será exponencial sobre el número de elecciones no-deterministas del *grafo de transiciones*.

El mayor y el menor de los valores así obtenidos proporcionan una valiosa información sobre el tiempo asociado a la ejecución del proceso objeto de nuestro estudio.

5 Especificación del Protocolo del Bit Alternante

El protocolo del bit alternante es un protocolo de comunicaciones que garantiza una comunicación segura (sin repetir mensajes) entre dos estaciones (Emisor y Receptor) a través de un canal que puede fallar con una probabilidad conocida, para lo cual se añade un bit a cada mensaje, que controla con su valor la pérdida y la no repetición del mismo. Estudiaremos cuatro casos distintos de este protocolo:

1. En un primer caso, consideramos 1 emisor, 1 receptor y 1 canal infalible.
2. En un segundo consideramos el mismo número de elementos, pero ahora el canal puede fallar con una probabilidad conocida p .
3. En el tercero extenderemos el modelo del caso previo considerando n emisores, m receptores y 1 canal que falla con probabilidad p .
4. En el último caso, consideramos n emisores, m receptores y 2 canales idénticos al del caso anterior.

Como el protocolo del bit alternante carece de comportamientos no-deterministas podremos obviar la primera fase del algoritmo de evaluación, pues el grafo generado por la Semántica Operacional es un *grafo de transición determinista*.

5.1 Caso 1

En este caso, el emisor (Sender) y el receptor (Receiver) sincronizarán en tm (transmisión del mensaje). El canal (Channel) y el receptor lo harán en dm (entrega del mensaje):

$$\begin{aligned} \text{Sender: } S &= \langle tm, \gamma \rangle . S \\ \text{Channel: } Ch &= tm . \langle dm, \mu \rangle . Ch \\ \text{Receiver: } R &= dm . out . R \end{aligned}$$

Alternating Bit Protocol 1:

$$\begin{aligned} ABP &\equiv S \parallel_{A_1} Ch \parallel_{A_1} R \\ A_1 &= \{tm, dm\} \end{aligned}$$

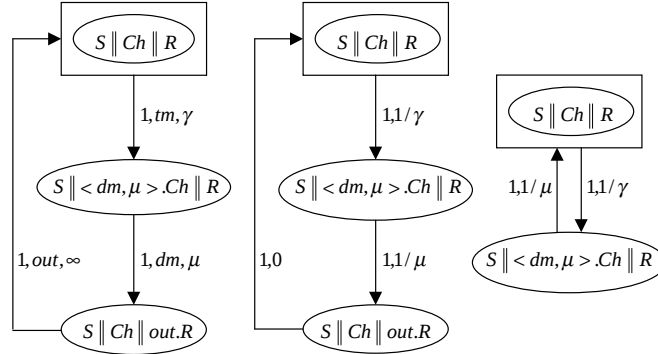


Figura 3: Caso 1 - *grafos de transición determinista, temporal determinista y temporal reducido determinista*

En la Fig.3 podemos observar (de izquierda a derecha): el *grafo de transición determinista*, el *grafo temporal determinista* y el *grafo temporal reducido determinista* correspondientes a este caso (en el cual los estados inicial y final coinciden).

A partir de este último grafo se puede calcular el tiempo medio para completar la transmisión de un mensaje:

$$T_{S_0} = 1(1/\gamma + T_{S_1}) = 1/\gamma + T_{S_1}$$

$$T_{S_1} = 1(1/\mu + T_{S_F}) = 1/\mu$$

En consecuencia:

$$T = \frac{1}{\gamma} + \frac{1}{\mu}$$

5.2 Caso 2

Consideremos ahora un nuevo proceso activador (Activator) que controla la alternancia del bit que se añade a cada mensaje. El activador y el emisor sincronizarán en in_j (correspondiente a añadir el bit j al mensaje). Emisor y receptor sincronizarán en tm_j (transmisión del mensaje con bit j), en da_j (entrega del acuse de recibo del mensaje con bit j) y en λ'' (entrega de mensaje de error), y canal y receptor sincronizarán en dm_j (entrega del mensaje con bit j), en λ (entrega de mensaje de error), en ta_j (transmisión del acuse de recibo del mensaje con bit j) y en λ' (transmisión de mensaje de error). Con la intención de aclarar el diseño en lo más posible se separa el canal en dos procesos, Go y Ba (uno para cada dirección), que nos permite capturar e identificar los diferentes mensajes de error que aparecen tras determinado tiempo, cuando el canal falla en la correspondiente dirección:

$$\begin{aligned}
 \text{Activator: } & A = in_0.in_1.A \\
 \text{Sender: } & S = in_j.S_j \\
 & S_j = \langle tm_j, \gamma \rangle. (da_j.S + \lambda''.S_j) \\
 \text{Channels: } & Go = tm_j.(\langle \lambda, \mu \rangle.Go \oplus_p \langle dm_j, \mu \rangle.Go) \\
 & Ba = ta_j.(\langle \lambda'', \mu \rangle.Ba \oplus_p \langle da_j, \mu \rangle.Ba) + \lambda'.\langle \lambda'', \mu \rangle.Ba \\
 \text{Receiver: } & R = \lambda.\lambda'.R + dm_0.out.ta_0.RR + dm_1.ta_1.R \\
 & RR = \lambda.\lambda'.RR + dm_0.ta_0.RR + dm_1.out.ta_1.R
 \end{aligned}$$

Alternating Bit Protocol 2:

$$\begin{aligned}
 ABP &\equiv A \parallel_{A_2} S \parallel_{A_2} Go \parallel_{A_2} Ba \parallel_{A_2} R \\
 A_2 &= \{in_j, tm_j, dm_j, ta_j, da_j, \lambda, \lambda', \lambda''\}
 \end{aligned}$$

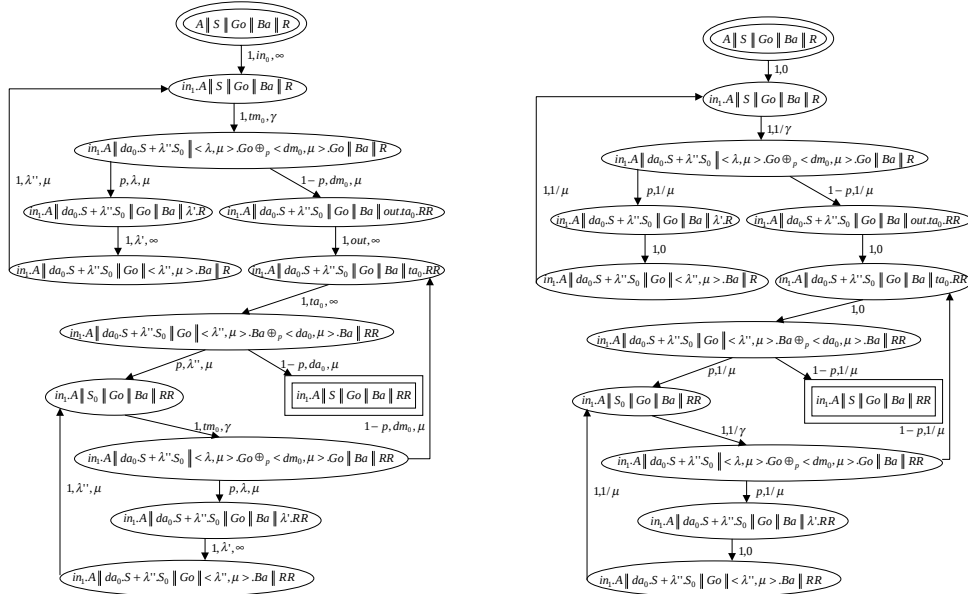


Figura 4: Caso 2 - grafo de transición determinista y grafo temporal determinista

Comprobemos, en primer lugar, que la especificación garantiza que el mensaje siempre llega al receptor:

$$\begin{aligned}
 P[llegue] &= \sum_{n=0}^{\infty} P[llegue \text{ por primera vez en el intento } (n+1)] = \\
 &= \sum_{n=0}^{\infty} P[falle en los primeros (n) intentos] \cdot P[llegue en el intento (n+1)] = \\
 &= \sum_{n=0}^{\infty} (1 - (1-p)^2)^n \cdot (1-p)^2 = (1-p)^2 / (1 - (2p - p^2)) = 1.
 \end{aligned}$$

Calculemos ahora el número medio de intentos necesarios para completar la transmisión de un mensaje:

La variable aleatoria que simula el número de intentos hasta los n primeros éxitos es la Binomial Negativa, luego como necesitamos estudiar el número de intentos hasta el primer éxito, donde $\mathbf{P}[exito] = (1 - p)^2$ (éxito es equivalente a que el canal funcione bien en ambas direcciones), entonces el valor buscado será la Esperanza matemática de $\mathbf{X} \sim \mathbf{NB}[1, (1 - p)^2]$, i.e., $\mathbf{E}[\mathbf{X}] = (1 - (1 - p)^2)/(1 - p)^2 = (2p - p^2)/(1 - p)^2$, que es el cociente entre las probabilidades de que el canal falle y de que el canal funcione bien.

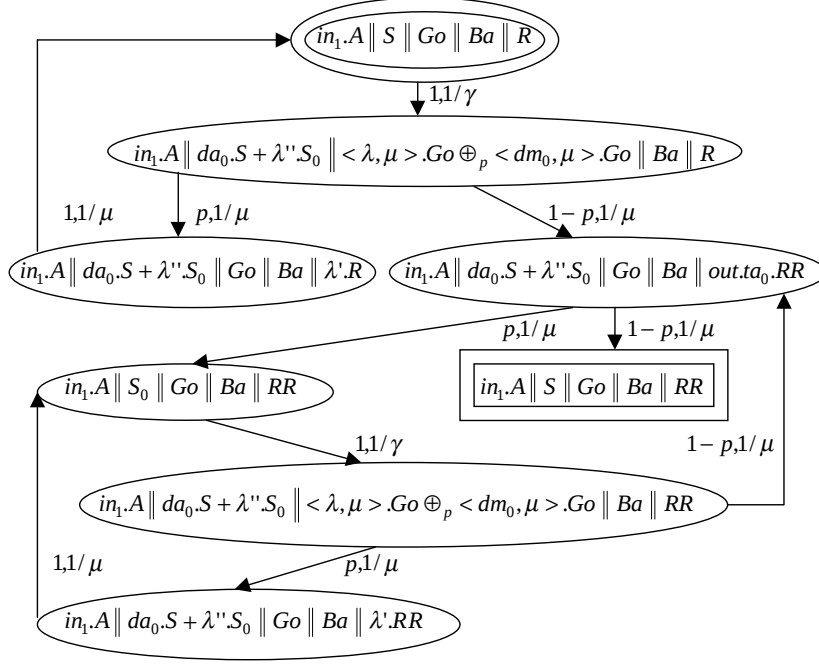


Figura 5: Caso 2 - grafo temporal reducido determinista

Finalmente, el tiempo medio T necesario para completar la comunicación de un mensaje puede ser obtenido a partir del *grafo temporal reducido determinista* (ver Fig.5, donde enumeramos los estados de arriba a abajo y en la misma altura de izquierda a derecha). A este grafo llegamos aplicando al *grafo de transición determinista* generado por la semántica operacional (ver Fig.4), la segunda fase del algoritmo de evaluación que nos conduce al *grafo temporal determinista* (Fig.4) y sobre este aplicamos la tercera fase del método descrito.

Por lo tanto, partiendo de la Fig.5 se obtiene fácilmente:

$$\begin{aligned}
 T_{estado\ inicial} &= T_0 = 1/\gamma + T_1 \\
 T_1 &= p(1/\mu + T_2) + (1 - p)(1/\mu + T_3) \\
 T_2 &= 1/\mu + T_0 = 1/\mu + 1/\gamma + T_1 \\
 T_3 &= (1 - p)(1/\mu + T_{S_F}) + p(1/\mu + T_4) = \frac{1-p}{\mu} + p(1/\mu + T_4) \\
 T_4 &= 1/\gamma + T_5 \\
 T_5 &= p(1/\mu + T_6) + (1 - p)(1/\mu + T_3) \\
 T_6 &= 1/\mu + T_4 \\
 &\vdots
 \end{aligned}$$

De ahí, podemos calcular el valor de T:

$$T = \frac{1/\gamma + 2/\mu}{(1-p)^2}$$

5.3 Caso 3

En este caso disponemos de n emisores compitiendo por el uso del canal, el cual será asignado de forma no-determinista:

$$\begin{aligned} \text{Activator: } A &= in_0.in_1.A \\ \text{Senders: } S &= \sum_{i=1}^n S_{ik} \\ S_{ik} &= in_j.S_{ijk} \\ S_{ijk} &= \langle tm_{ijk}, \gamma_i \rangle. (da_j.S + \lambda''.S_{ijk}) \\ \text{Channels: } Go &= tm_{ijk}.(\langle \lambda, \mu \rangle.Go \oplus_p \langle dm_{jk}, \mu \rangle.Go) \\ Ba &= ta_j.(\langle \lambda'', \mu \rangle.Ba \oplus_p \langle da_j, \mu \rangle.Ba) + \lambda'.\langle \lambda'', \mu \rangle.Ba \\ \text{Receivers: } R &= \lambda.\lambda'.R \sum_{k=1}^m (dm_{0k}.out_k.ta_0.RR + dm_{1k}.ta_1.R) \\ RR &= \lambda.\lambda'.RR \sum_{k=1}^m (dm_{0k}.ta_0.RR + dm_{1k}.out_k.ta_1.R) \end{aligned}$$

Alternating Bit Protocol 3:

$$\begin{aligned} ABP &\equiv A \parallel_{A_3} S \parallel_{A_3} Go \parallel_{A_3} Ba \parallel_{A_3} R \\ A_3 &= \{in_j, tm_{ijk}, dm_{jk}, ta_j, da_j, \lambda, \lambda', \lambda''\} \end{aligned}$$

Los grafos, tanto para este modelo como para el siguiente son el mismo que los del caso anterior, con la única excepción de la competencia por el recurso compartido, la cual introduce únicamente un nuevo estado por delante del estado inicial, y en la transición que los une se decide la asignación del recurso a uno u otro proceso, para lo cual se supone un coste temporal nulo.

Una vez que el emisor i ha obtenido el canal, necesitará un tiempo $T = (1/\gamma_i + 2/\mu)/(1-p)^2$ para completar la comunicación de su mensaje. Así pues, resuelto el caso 2, nuestro problema pasa al terreno de la Teoría de Colas clásica.

Supongamos que la posición esperada por el emisor i en la cola para acceder al canal es $j+1$, por lo tanto el emisor i debe esperar el tiempo requerido por los j emisores que le preceden, para que completen sus transmisiones, a saber: $\sum_{n=1}^j (1/\gamma_n + 2/\mu)/(1-p)^2$, a lo cual se le añade el tiempo para su propia transmisión: $(1/\gamma_i + 2/\mu)/(1-p)^2$, obteniendo de esta forma:

$$T = \frac{1/\gamma_i + 2/\mu + \sum_{n=1}^j (1/\gamma_n + 2/\mu)}{(1-p)^2}$$

5.4 Caso 4

Suponemos que ambos canales tienen la misma probabilidad de fallar, p . La especificación de este caso es esencialmente idéntica a la del caso anterior, excepto que se necesitan 2^2 estados para que cada receptor pueda identificar la posible repetición de los mensajes. También otorgamos a los emisores la posibilidad de poder mantener la comunicación paralela de tantos mensajes como canales tenga asignados:

Activators:

$$A_0 = in_{00}.in_0.in_{10}.in_0.A_0$$

$$A_1 = in_{01}.in_1.in_{11}.in_1.A_1$$

Senders:

$$S = \sum_{i=1}^n S_{ik}$$

$$S_{ik} = in_{jg}.S_{ijgk}$$

$$S_{ijgk} = \langle tm_{ijgk}, \gamma_i \rangle \cdot (da_{jg}.in_g.S + \lambda_g''.S_{ijgk})$$

Channels:

$$Go_g = tm_{ijgk} \cdot (\langle \lambda_g, \mu \rangle \cdot Go_g \oplus_p \langle dm_{jgk}, \mu \rangle \cdot Go_g)$$

$$Ba_g = ta_{jg} \cdot (\langle \lambda_g'', \mu \rangle \cdot Ba_g \oplus_p \langle da_{jg}, \mu \rangle \cdot Ba_g) + \lambda_g' \cdot \langle \lambda_g'', \mu \rangle \cdot Ba_g$$

Receivers:

$$R = \lambda_0 \cdot \lambda'_0 \cdot R + \lambda_1 \cdot \lambda'_1 \cdot R \sum_{k=1}^m \sum_{j=0}^1 \sum_{g=0}^1 dm_{jgk} \cdot out_k \cdot ta_{jg} \cdot R^{jg}$$

$$R^{jg} = \lambda_0 \cdot \lambda'_0 \cdot R^{jg} + \lambda_1 \cdot \lambda'_1 \cdot R^{jg} \sum_{k=1}^m \sum_{r=0}^1 \sum_{s=0}^1 \begin{cases} dm_{rsk} \cdot ta_{rs} \cdot R^{rs} & r = j \wedge s = g \\ dm_{rsk} \cdot out_k \cdot ta_{rs} \cdot R^{rs} & e.o.c. \end{cases}$$

Alternating Bit Protocol 4:

$$ABP \equiv A_0 \parallel_{A_4} A_1 \parallel_{A_4} S \parallel_{A_4} S \parallel_{A_4} Go_0 \parallel_{A_4} Ba_0 \parallel_{A_4} Go_1 \parallel_{A_4} Ba_1 \parallel_{A_4} R \parallel_{A_4} R$$

$$A_4 = \{in_{jg}, in_g, tm_{ijgk}, dm_{jgk}, ta_{jg}, da_{jg}, \lambda_g, \lambda'_g, \lambda_g''\}$$

Nos encontramos con el mismo análisis del tiempo necesario para que el emisor i complete la comunicación de su mensaje, que en el modelo con un solo canal, a excepción de que ahora el tiempo necesario para que los j emisores que le preceden completen sus respectivas comunicaciones es la mitad (en promedio) que en el caso anterior: $(\sum_{n=1}^j (1/\gamma_n + 2/\mu) / ((1-p)^2))/2$, de nuevo lo sumamos al tiempo necesario para su propia comunicación: $(1/\gamma_i + 2/\mu) / ((1-p)^2)$ obteniendo finalmente:

$$T = \frac{1/\gamma_i + 2/\mu + \sum_{n=1}^j (1/2\gamma_n + 1/\mu)}{(1-p)^2}$$

Generalizando al caso en que se disponga de k canales, se obtiene:

$$T = \frac{1/\gamma_i + 2/\mu + \sum_{n=1}^j (1/k\gamma_n + 2/k\mu)}{(1-p)^2}$$

Por consiguiente, es inmediato que en el límite $k \rightarrow \infty$, cada emisor necesitaría el mismo tiempo que si dispusiera de un canal para su uso exclusivo.

6 Conclusiones y Trabajos Futuros

Se ha presentado un lenguaje algebraico, **ROSA**, para la modelización y el análisis de sistemas con probabilidades y no-determinismo. Se ha definido su semántica operacional y un mecanismo de evaluación de prestaciones, que permite obtener para cada proceso un conjunto de grafos a partir de los cuales se puede computar el tiempo medio necesario para alcanzar el estado final a partir de cualquier estado. Finalmente se podrán calcular unas cotas mínimas y máximas para los tiempos medios de todo el sistema. En particular, se ha aplicado este método para la evaluación cualitativa y cuantitativa del protocolo del bit alternante en algunos casos interesantes, habiéndose obtenido como resultado los valores que cabía esperar.

Nuestro trabajo actual está orientado principalmente a definir una relación de equivalencia por bisimulación sobre **ROSA**, la que resultará valiosa para programar una herramienta para la aplicación del lenguaje. Finalmente, también estamos trabajando sobre la aplicación del lenguaje a otros ejemplos más complejos y de gran interés, como a algunos algoritmos para la compresión y transmisión de secuencias de audio y de vídeo.

Referencias

- [1] M. Bernardo and R. Gorrieri. A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time. *Theoretical Computer Science*, 202:1–54, 1998.
- [2] B. Bloom and A.R. Meyer. A remark on bisimulation between probabilistic processes. In Meyers and Tsailin, editors, *Logik at Botik*. Springer-Verlag, 1989.
- [3] A. Bueno, V. Valero, and F. Cuartero. TPAL: A timed-probabilistic model for concurrent processes. In *Proceedings of APSEC'97/ICSC'97*, Hong-Kong, 1997.
- [4] F. Cuartero, D. de Frutos, and V. Valero. PCSP: A denotational model for probabilistic processes. In *Proceedings of Third AMAST Workshop on Real-Time Systems*, 1996.
- [5] F. Cuartero, D. de Frutos, and V. Valero. A sound and complete proof system for probabilistic processes. In *Proceedings of Fourth AMAST Workshop on Real-Time Systems*, 1997.
- [6] P. R. D'Argenio, J. P. Katoen, and E. Brinksma. An algebraic approach to the specification of stochastic systems. *Programming Concepts and Methods*, pages 126–147, 1998.
- [7] A. Giacalone, C.-C. Jou, and S.A. Smolka. Algebraic reasoning for probabilistic concurrent systems. In *Proceedings of Working Conference on Programming Concepts and Methods, IFIP TC 2*, Sea of Galilee, Israel, 1990.
- [8] H. Hansson and B. Jonsson. A calculus for communicating systems with time and probabilities. In *Proceedings Real-Time Systems Symposium*, Orlando, Florida, 1990.
- [9] B. Jonsson and K.G. Larsen. Specification and refinement of probabilistic processes. In *LICS'91*, Amsterdam, 1991.
- [10] G. Lowe. Representing nondeterministic and probabilistic behaviour in reactive processes. Technical Report PRG-TR-11-93, Oxford university computing laboratory, 1993.
- [11] M. Núñez, D. de Frutos, and L. Llana. Acceptance trees for probabilistic processes. In *CONCUR'95, LNCS 962*, pages 249–263, 1995.
- [12] F. L. Pelayo, F. Cuartero, V. Valero, and D. Cazorla. PPNAL: Performance evaluation in an algebraic model for probabilistic and non-deterministic processes. In *Proceedings of 15th Annual UK Performance Engineering Workshop*, pages 1–12, Bristol, U.K., 1999.
- [13] R. van Glabbeek, S.A. Smolka, B.U. Steffen, and C.M.N. Tofts. Reactive, generative and stratified models of probabilistic processes. In *Proceedings of 5th Annual IEEE Symposium on Logic in Computer Science*, pages 130–141, Philadelphia, PA, 1990.
- [14] W. Yi and K.G. Larsen. Testing probabilistic and nondeterministic processes. In *Protocol Specification, Testing and Verification XII*, pages 47–61, 1992.