PPNAL: Evaluación de prestaciones en un modelo algebraico para procesos probabilísticos y no deterministas

Fernando L. Pelayo   Fernando Cuartero   Valentín Valero   Diego Cazorla

Departamento de Informática *
Escuela Politécnica Superior de Albacete
Universidad de Castilla-La Mancha
Campus Univ. s/n. 02071, Albacete
{fpelayo, fernando, valentin}@info-ab.uclm.es, dcazorla@pol-ab.uclm.es

Resumen

Presentamos un lenguaje algebraico que permite describir procesos probabilísticos y no deterministas a la vez que evaluar prestaciones temporales: **PPNAL** (**P**erformance **P**robabilistic and **N**ondeterministic **A**lgebraic **L**anguage). El lenguaje está basado en un álgebra de procesos clásica, a la que se han añadido un operador de elección probabilística y en el que las acciones tienen información sobre su duración. Presentamos una primera versión de la semántica operacional del lenguaje y un algoritmo de evaluación de prestaciones, aplicando todos estos resultados al estudio de una serie de variaciones del protocolo del bit alternante, evaluándolo en diferentes situaciones.

1 Introducción

En la década de los 80 quedaron firmemente sentadas las bases fundamentales en el marco de las álgebras de procesos. Estos trabajos han provisto herramientas poderosas para el diseño y el análisis de una amplia variedad de sistemas concurrentes. Una de las principales ventajas de estas herramientas ha sido el diseño de los sistemas de una manera composicional, junto con una sencillez notable en los modelos obtenidos.

La proliferación de sistemas distribuidos, sistemas en tiempo real y sistemas tolerantes a fallos ha originado que factores como el tiempo y las probabilidades, abstraidos en los modelos iniciales basados en álgebras de procesos, pasen a ser considerados como un elemento especialmente importante de los mismos.

Es por ello que en estos momentos no es suficiente tener en cuenta únicamente factores cualitativos de los sistemas concurrentes. Debido a ello, diversos investigadores han estudiado posibles extensiones a estos modelos que permitan modelar factores cuantitativos. Así, se han propuesto extensiones a los modelos básicos de álgebras de procesos que incluyen tiempo, prioridades y probabilidades.

*Este trabajo ha sido financiado por el proyecto CICYT "Desarrollo formal de sistemas distribuidos", TIC97-0669-c03-02.

En consecuencia, han aparecido diversas extensiones de los modelos clásicos, de los cuales algunos incorporan el tiempo como un factor más de diseño. Tenemos extensiones temporizadas de CSP [Ort90, RR88], de CCS [Yi91], y de LOTOS [LOT88, QAdF89].

Otras extensiones incluyen la posibilidad de incluir en el lenguaje herramientas de descripción de comportamientos probabilísticos. Una primera aproximación a los procesos probabilísticos la tenemos en [BM89] donde se introduce el concepto de experimento probabilístico. Posteriormente, en [GJS90] se define PCCS, un lenguaje de naturaleza algebraica, basado en la versión síncrona de CCS [Mil89], en el cual se sustituye el operador no determinista suma por una versión probabilística, $\sum [p_i] E_i$, donde p_i es la probabilidad de que el proceso se comporte como E_i. En [vGSST90] se amplía este modelo presentando las interpretaciones reactiva, generativa y estratificada para procesos concurrentes probabilísticos.

Posteriormente, en [HJ90] se presenta un álgebra de procesos, *Timed Probabilistic Calculus of Communicating Systems* (TPCCS) que también extiende CCS con aspectos probabilísticos (además de temporizados). No obstante, en el aspecto probabilístico aparece una novedad importante respecto a PCCS, puesto que considera una versión asíncrona del lenguaje, lo cual dificulta sensiblemente la semántica, aunque permite modelos más naturales de los sistemas concurrentes, en general de carácter asíncrono. La probabilidad se introduce mediante un operador de elección probabilística. En cada estado se realiza una elección no determinista o probabilística y éstas aparecen con una alternancia estricta, por lo que se le da al modelo el nombre de *modelo alternante*. Además de los ya comentados, se han propuesto otros lenguajes basados en CCS, [Chr90, JS90, LS89], analizando diversos aspectos semánticos. Así, en [LS89] se define una noción de bisimulación sobre sistemas de transición probabilísticos, y en [Chr90] vuelve a tratarse este mismo tema. En [JS90] se estudian distintas nociones de equivalencia de procesos.

Sobre la base de CSP, tenemos [CdFV96, CdFV97], en los cuales se considera una interpretación *reactiva* de las probabilidades, mientras que en [NdFL95] se estudia una interpretación *generativa*. En ambos modelos se presentan semánticas de pruebas que son completamente abstractas respecto de semánticas denotacionales basadas en árboles de aceptación.

No obstante, la idea subyacente en todos estos modelos probabilísticos es sustituir un comportamiento no-determinista, por un comportamiento probabilístico, en el cual, los diversos posibles comportamientos del sistema aparecen cuantificados con una probabilidad. Ello exige un conocimiento cuantitativo de los posibles comportamientos de un sistema para poder proceder a su modelado. Sin embargo, en la práctica, esta distribución de probabilidades puede ser, en ocasiones, o bien desconocida o bien conocida sólo en parte [JL91, Yi94].

Por tanto se hace necesario tener operadores que nos permitan tratar el no-determinismo. Ello no quiere decir que debamos olvidar los aspectos probabilísticos, puesto que en el modelado de sistemas concurrentes nos podemos encontrar en ocasiones con comportamientos no-deterministas susceptibles de ser cuantificados (p.e. la probabilidad de fallo de un canal de comunicación), mientras que, en otras ocasiones no lo serán (p.e. la velocidad relativa de dos procesos). Es por ello que parece conveniente disponer de lenguajes que permitan modelar sistemas no-deterministas con la posibilidad de optar por una cuantificación o no dependiendo de la naturaleza de los mismos.

Siguiendo estas ideas, en [YL92] se presenta una extensión probabilística de CCS con un operador binario de elección $\oplus_p$, subindicado con una probabilidad $p \in (0, 1)$, que modela la elección probabilística interna con una interpretación generativa, mientras que el operador de elección original de CCS ($\oplus$) se mantiene para representar el no-determinismo clásico. En [Low93] se presenta una extensión probabilística de CSP (PCSP) manteniendo el no-determinismo. Para este lenguaje se define una semántica operacional y se proporcionan semánticas denotacionales en base a diferentes tipos de observaciones (semánticas de trazas, de escoba, de barbas) sobre árboles con tres tipos de nodos: probabilísticos, no deterministas y nodos acción.En [BVC97] se presenta TPAL, un lenguaje algebraico con probabilidades y tiempo, basado en LOTOS, en el que se sustituye la elección interna por una elección probabilística. Se exige una alternancia entre los dos operadores de elección.

Finalmente, parece razonable tener en cuenta aspectos de rendimiento ya en las primeras etapas de diseño de los sistemas. Es cierto que, a menudo, las consideraciones de eficiencia se dejan hasta que la funcionalidad del sistema ha sido completamente comprobada, sin embargo se puede objetar que, en ese caso, las posibles correcciones en la etapa final para mejorar rendimientos desemboquen en un nuevo rediseño global, con el consiguiente aumento de costes y alargamiento de plazos. Además, si los tests pasados al sistema para probar los requerimientos se hacen sobre un modelo diferente al que se usa para medir las prestaciones del mismo, esto puede producir una inconsistencia del sistema. Esto hace pues doblemente oportuno que un modelo apropiado incluya consideraciones sobre la eficiencia.

En [BG98] se presenta un lenguaje basado en un álgebra de procesos estocástica y probabilística (**EMPA**), las consideraciones realizadas son bastante similares a las del modelo que propugnamos. En particular, la sintaxis difiere en que mientras en **EMPA**los diversos tipos de información cuantitativa estan siempre asociados a las acciones, en nuestro lenguaje adoptamos la política clásica de definir una adecuada sintaxis a cada tipo de operación, con el objetivo de hacer más inteligibles los sistemas diseñados. Por otra parte, también aparecen interpretaciones diferentes en el propio concepto de algunas operaciones, que más adelante detallaremos.

El artículo se estructura de la siguiente manera: en la sección 2 se define la sintaxis del lenguaje como una extensión de **EPL** en que las acciones tienen una duración temporal asociada ([Hen88]). En la sección 3 se da su semántica operacional. En la sección 4 se da un algoritmo que permite calcular el coste temporal. Finalmente en la sección 5 se lleva a cabo un estudio extenso del protocolo del bit alternante.

2 Sintaxis de PPNAL

El lenguaje que presentamos toma como base un lenguaje algebraico clásico (**EPL** definido en [Hen88]) al cual añadimos una variación probabilística siguiendo ideas desarrolladas en [CCV98] junto con una variación markoviana similar a la desarrollada en [BG98]. Así pues, consideramos los dos operadores de elección clásicos: la elección interna, $\oplus$, y la elección externa, $+$, y añadimos un nuevo operador, la elección probabilística: $\oplus_p$. En cuanto a las acciones, podrán aparecer en los procesos con las formas a ó $\langle a, \lambda\rangle$. En esta última presentación, una acción tendrá una duración modelada por variables aleatorias de tipo exponencial, así, $\langle a, \lambda\rangle$ representa la ejecución de a tardando un tiempo que se distribuye como una variable aleatoria $Exp[\lambda]$. Las acciones cuya duración temporal sea insignificante las representaremos por su tipo exclusivamente y consideraremos que son acciones instantáneas.

Así pues, llamaremos $\Delta = \{a, b, c, ..\}$ al conjunto de tipos de acciones, las probabilidades serán denotadas por las letras $p, q, r, ..$ y los parámetros temporales por letras griegas.

El lenguaje será el conjunto de expresiones definidas por la siguiente notación BNF:

$$P ::= \mathbf{0} \mid a.P \mid \langle a, \lambda \rangle.P \mid P \oplus P \mid P + P \mid P \oplus_p P \mid P|P \mid P \backslash A$$

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

La interpretación de los operadores es la habitual. Cabe describir el significado del nuevo operador de elección probabilística $\oplus_p$ en sus formas binaria y general:

$P \oplus_p Q$, tiene un significado generativo clásico, es decir el proceso se comporta como P con probabilidad p , y con probabilidad $1 - p$ como Q.

3 Semántica operacional

La semántica operacional, que está todavía en un estadio preliminar, se definirá de la forma habitual por medio de un sistema de transiciones etiquetadas con tuplas de tres elementos siendo el primero una probabilidad, el segundo un tipo de acción y el tercero el parámetro de la variable aleatoria Exponencial que modela la duración de la acción (∞ para acciones inmediatas). Presentamos seguidamente una primera aproximación a la semántica, donde se tratan el prefijo y las elecciones.

Las reglas S1 hacen referencia al prefijo, y son inmediatas. Las reglas S2 definen la elección probabilística con un sentido generativo, mientras que las reglas S3 son las clásicas para un operador de elección interna.

En cuanto al significado de las reglas S4, seguimos la idea presentada en [BG98] como la política de la velocidad (*race policy*), es decir en caso de que una misma acción pueda ser ejecutada por los dos argumentos, el más rápido será el seleccionado para la ejecución, así si una acción es inmediata tendrá preferencia (si lo son las dos la elección será no determinista). En caso de que las dos sean temporizadas será elegida la que antes esté disponible.

Aún cuando no presentamos la semántica de la composición paralela, la idea será similar a [BG98], pues consideramos el paralelo como una elección externa generalizada a lo largo del tiempo. La diferencia fundamental será que permitiremos la sincronización entre acciones temporizadas, en cuyo caso la acción se ejecutará conjuntamente por los dos procesos, tomando como tiempo el más lento de los dos.

Consideramos **Tipos** la función definida sobre el conjunto de los multiconjuntos de acciones y que asocia a cada multiconjunto el conjunto formado por los tipos de sus acciones.

$$\mathbf{Tipos} : \{\mathit{multiconjuntos\ de\ acciones}\} \longrightarrow \mathcal{P}[\Delta]$$

D será la función definida sobre el conjunto de los procesos de **PPNAL** y que asocia a cada proceso su multiconjunto de acciones disponibles, **D** se define por inducción estructural sobre el conjunto de procesos de **PPNAL** así:

$$
\begin{array}{llll}
\mathbf{D}: & \{\textit{procesos de } \mathbf{PPNAL}\} & \rightarrow & \{\textit{multiconjuntos de acciones}\} \\
& \mathbf{0} & \rightsquigarrow & \varnothing \\
& a.P & \rightsquigarrow & \{a\} \\
& \langle a,\lambda\rangle.P & \rightsquigarrow & \{\langle a,\lambda\rangle\} \\
& P \oplus Q & \rightsquigarrow & \mathbf{D}[P] \cup \mathbf{D}[Q] \\
& P + Q & \rightsquigarrow & \mathbf{D}[P] \cup \mathbf{D}[Q] \\
& P \oplus_p Q & \rightsquigarrow & \mathbf{D}[P] \cup \mathbf{D}[Q] \\
& P|Q & \rightsquigarrow & \mathbf{D}[P] \cup \mathbf{D}[Q] \\
& P\backslash A & \rightsquigarrow & \mathbf{D}[P]\backslash A
\end{array}
$$

Dadas las definiciones, consideremos que el operador de elección probabilística distribuye sobre todos los demás y el de elección interna distribuye al de elección externa (véase [CCV98]. En la tabla 1 está representado el sistema de transiciones que define la semántica.

Tabla 1: Semántica operacional

(S1a)	$\dfrac{}{a.P \overset{1,a,\infty}{\longrightarrow} P}$	**(S1b)**	$\dfrac{}{\langle a,\lambda\rangle.P \overset{1,a,\lambda}{\longrightarrow} P}$
(S2a)	$\dfrac{P \overset{q,a,\lambda}{\longrightarrow} P'}{P \oplus_p Q \overset{pq,a,\lambda}{\longrightarrow} P}$	**(S2b)**	$\dfrac{Q \overset{t,a,\lambda}{\longrightarrow} Q'}{P \oplus_p Q \overset{(1-p)t,a,\lambda}{\longrightarrow} Q'}$
(S3a)	$\dfrac{P \overset{q,a,\lambda}{\longrightarrow} P'}{P \oplus Q \overset{q,a,\lambda}{\longrightarrow} P'}$	**(S3b)**	$\dfrac{Q \overset{q,a,\lambda}{\longrightarrow} Q'}{P \oplus Q \overset{q,a,\lambda}{\longrightarrow} Q'}$
(S4a)	$\dfrac{P \overset{q,a,\lambda}{\longrightarrow} P' \wedge a \notin \mathbf{Tipos}[\mathbf{D}[Q]]}{P + Q \overset{q,a,\lambda}{\longrightarrow} P'}$	**(S4b)**	$\dfrac{Q \overset{q,a,\lambda}{\longrightarrow} Q' \wedge a \notin \mathbf{Tipos}[\mathbf{D}[P]]}{P + Q \overset{q,a,\lambda}{\longrightarrow} Q'}$
(S4c)	$\dfrac{P \overset{q,a,\infty}{\longrightarrow} P'}{P + Q \overset{q,a,\infty}{\longrightarrow} P'}$	**(S4d)**	$\dfrac{Q \overset{q,a,\infty}{\longrightarrow} Q'}{P + Q \overset{q,a,\infty}{\longrightarrow} Q'}$
(S4e)	$\dfrac{P \overset{q,a,\lambda_1}{\longrightarrow} P' \wedge Q \overset{r,a,\lambda_2}{\longrightarrow} Q'}{P + Q \overset{q\cdot r,a,\lambda_1+\lambda_2}{\longrightarrow} P'}$	**(S4f)**	$\dfrac{P \overset{q,a,\lambda_1}{\longrightarrow} P' \wedge Q \overset{r,a,\lambda_2}{\longrightarrow} Q'}{P + Q \overset{q\cdot r,a,\lambda_1+\lambda_2}{\longrightarrow} Q'}$
(S4g)	$\dfrac{P \overset{q,a,\lambda_1}{\longrightarrow} P' \wedge Q \overset{r,a,\lambda_2}{\longrightarrow} Q'}{P + Q \overset{q\cdot(1-r),a,\lambda_1}{\longrightarrow} P'}$	**(S4h)**	$\dfrac{P \overset{q,a,\lambda_1}{\longrightarrow} P' \wedge Q \overset{r,a,\lambda_2}{\longrightarrow} Q'}{P + Q \overset{(1-q)\cdot r,a,\lambda_2}{\longrightarrow} Q'}$

4 Evaluación de prestaciones

Presentadas las reglas semánticas, pasamos a describir en dos fases cómo evaluar las prestaciones a partir del *grafo de transiciones* generado por tales reglas. Previamente recordaremos un par de resultados sobre cálculo de probabilidades que más adelante usaremos:

Una variable aleatoria discreta $\mathbf{X}$ tiene una distribución de probabilidad Binomial Negativa de parámetros k y p, lo denotaremos por $\mathbf{X} \backsim \mathbf{BN}[k,p]$, si su función de probabilidad es $\mathbf{P}(X=x) = \frac{(x-1)!}{(k-1)!(x-k)!} p^k (1-p)^{x-k}$. Su valor medio es $\mathbf{E}[\mathbf{X}] = \frac{k(1-p)}{p}$.

Una variable aleatoria contínua $\mathbf{Y}$ tiene una distribución de probabilidad Exponencial de parámetro λ ($\lambda > 0$), lo denotaremos por $\mathbf{Y} \backsim \mathbf{Exp}[\lambda]$, si su función de densidad de probabilidad es

$\mathbf{f_Y}[y] = \begin{cases} \lambda e^{-\lambda y} & y > 0 \\ 0 & e.o.c. \end{cases}$ Su valor medio es $\mathbf{E}[\mathbf{Y}] = 1/\lambda$.

4.1 Fase gráfica

En esta fase pasamos del *grafo de transiciones* al *grafo temporal reducido*.

A partir del *grafo de transiciones* (gráfico 1) obtenemos el *grafo temporal* (gráfico 2) reetiquetando cada arco con un par formado por la misma probabilidad y el inverso del parámetro temporal de la acción correspondiente (este segundo componente del par es la esperanza matemática de la variable aleatoria Exponencial que modela la duración de la acción, esto es, su tiempo medio de ejecución).

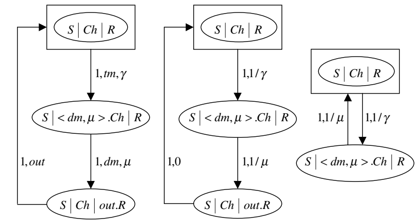
Del *grafo temporal* se eliminarán los arcos cuyo segundo parámetro sea nulo así como los nodos origen de los mismos, lo que solapará estados entre los que no hay gasto temporal (en tal caso, la probabilidad del arco eliminado en el solapamiento, multiplicará la probabilidad de cada arco que parta del nodo destino del arco eliminado), obteniendo finalmente el *grafo temporal reducido* (gráfico 3) en el que todos los arcos llevan asociado un gasto temporal medio no nulo.

4.2 Fase numérica

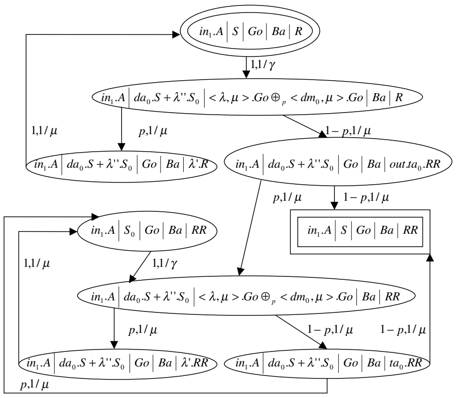
Partimos del *grafo temporal reducido* que tendrá un nodo marcado como inicial (doble óvalo) y otro como final (doble rectángulo), entre los cuales queremos calcular el tiempo medio para evolucionar.

Apoyándonos en el teorema de la probabilidad total, en la independencia de sucesos y en la linealidad del operador Esperanza Matemática, y denotando al tiempo esperado para evolucionar desde el nodo i al nodo final como T_i, se tiene que: $T_i = \sum_{j=1}^{n_i} p_j(1/\lambda_j + T_{e_{ij}})$ donde del nodo i parten n_i arcos, y el arco j está etiquetado con $p_j, 1/\lambda_j$ conectando los nodos i y e_{ij}, podemos afirmar que: "El tiempo medio para evolucionar desde el estado inicial hasta el estado final es $T_{estado\ inicial}$".

Es interesante destacar que $T_{estado\ final} = 0$ y que por cada bucle del *grafo temporal reducido* se genera un sistema compatible determinado de ecuaciones lineales. Todo lo cual nos permite afirmar que es evaluable el tiempo entre los estados inicial y final en función de los parámetros temporales de las acciones y de las probabilidades de las elecciones correspondientes.



Caso 1: Gráficos 1, 2 y 3



Caso 2: Gráfico 3

5 Implementación y prestaciones del Protocolo del Bit Alternante

En esta sección ilustraremos la potencia y versatilidad de **PPNAL** por medio de unos sencillos, aunque no triviales, ejemplos de aplicación al diseño de sistemas concurrentes.

El protocolo del bit alternante [BSW69] es un protocolo de comunicación que permite a dos estaciones, emisor y receptor, establecer una comunicación segura (todo mensaje que el emisor envíe llegará al receptor una vez exactamente) a través de un medio que puede fallar. Su nombre proviene del hecho de aumentar cada mensaje con un bit adicional.

Caso 1: Sistema de comunicaciones compuesto por un emisor, un receptor y un canal seguro (no es necesario el bit). Emisor y canal sincronizan en tm (transmisión del mensaje). Canal y receptor sincronizan en dm (entrega del mensaje).

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

Alternating Bit Protocol 1:

$$ABP \equiv (S \mid Ch \mid R) \backslash \{tm, dm\}$$

En los gráficos 1, 2 y 3 del caso 1 podemos ver los $grafos\ de\ transiciones$, $temporal$ y $temporal\ reducido$ respectivamente del modelo de estudio. A partir de este último grafo evaluamos el tiempo medio para completarse la transmisión de un mensaje: $T_{estado\ inicial} = 1(1/\gamma + T_1) = 1/\gamma + T_1$; $T_1 = 1(1/\mu + T_{estado\ final}) = 1/\mu$; obteniendo finalmente:

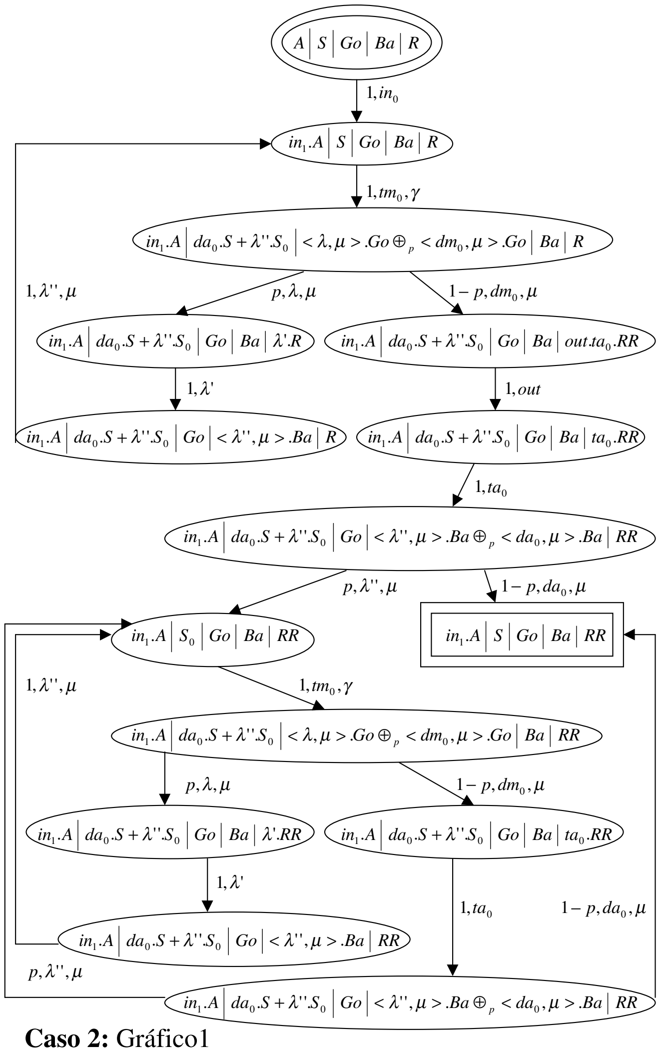
$$T = \frac{1}{\gamma} + \frac{1}{\mu}$$

Caso 2: Sistema de comunicaciones compuesto por un emisor, un receptor y un canal no seguro con probabilidad de fallo p. Activador y emisor sincronizan en in_j (determinando el bit j). Emisor y canal sincronizan en tm_j (transmisión del mensaje) a la ida y en da_j (entregando el reconocimiento) y en λ'' (entregando mensaje de error) a la vuelta. Canal y receptor sincronizan en dm_j (entrega del mensaje) y en λ (entregando mensaje de error) a la ida y en ta_j (transmisión del reconocimiento) y en λ' (transmitiendo mensaje de error) a la vuelta. Cabe aclarar que se ha llamado al canal de ida Go (Emisor$\longrightarrow$Receptor) y al de vuelta $Ba(ck)$ (Receptor$\longrightarrow$Emisor) y que el receptor tiene tantos estados como bits distintos pueden ser añadidos a los mensajes que recibe.

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

Alternating Bit Protocol 2:

$$ABP \equiv (A \mid S \mid Go \mid Ba \mid R)\backslash\{in_j, tm_j, dm_j, ta_j, da_j, \lambda, \lambda', \lambda''\}$$



Caso 2: Gráfico1

En primer lugar veamos que el mensaje llega:
$P[llegue] = \sum_{n=0}^{\infty} P[llegue\ por\ primera\ vez\ en\ el\ intento(n+1)] =$
$= \sum_{n=0}^{\infty} P[falle\ en\ los\ primeros(n)intentos]P[llegue\ en\ el\ intento(n+1)] =$
$= \sum_{n=0}^{\infty}(1-(1-p)^2)^n(1-p)^2 = \sum_{n=0}^{\infty}(2p-p^2)^n(1-p)^2 = \frac{(1-p)^2}{1-(2p-p^2)} = 1.$

Calculemos el número de intentos para completar la comunicación de un mensaje: La variable aleatoria que modeliza el número de intentos hasta el n-simo éxito es la Binomial Negativa, en nuestro caso necesitamos estudiar el número de intentos hasta el primer éxito siendo $\mathbf{P}[exito] = (1-p)^2$ (éxito es equivalente a que no falle el canal en ningún sentido). Así pues el número medio de intentos será la Esperanza Matemática de la correspondiente $\mathbf{X} \backsim \mathbf{BN}[1, (1-p)^2]$ que es $\mathbf{E}[\mathbf{X}] = \frac{1-(1-p)^2}{(1-p)^2} = \frac{2p-p^2}{(1-p)^2}$, esto es, el cociente entre la probabilidad de fallo del canal en algún sentido y la probabilidad de funcionamiento correcto del canal.

Estudiemos el coste temporal medio de completar la comunicación de un mensaje: En los gráficos 1 y 3 del caso 2 podemos ver los *grafos de transiciones* y *temporal* *reducido* respectivamente del modelo de estudio. A partir de este último, evaluamos el tiempo medio para completarse la transmisión de un mensaje (para mayor claridad se enumeran los estados de arriba a abajo y de izquierda a derecha en la misma altura) : $T_{estado\ inicial} = 1/\gamma + T_1$; $T_1 = p(1/\mu + T_2) + (1-p)(1/\mu + T_3)$; $T_2 = 1/\mu + T_{estado\ inicial} = 1/\mu + 1/\gamma + T_1$; $T_1 = \frac{p}{1-p}(2/\mu + 1/\gamma) + 1/\mu + T_3$; $T_3 = (1-p)(1/\mu + T_{estado\ final}) + p(1/\mu + T_4) = \frac{1-p}{\mu} + p(1/\mu + T_4)$; $T_4 = 1/\gamma + T_5$; $T_5 = p(1/\mu + T_6) + (1-p)(1/\mu + T_7)$; $T_6 = 1/\mu + T_4$; $T_7 = (1-p)(1/\mu + T_{estado\ final}) + p(1/\mu + T_4) = \frac{1-p}{\mu} + p(1/\mu + T_4)$; $T_4 = (1/\gamma + 2/\mu)/(1-p)^2$; obteniendo finalmente:

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

Caso 3: Sistema de comunicaciones compuesto por n emisores, m receptores y un canal no seguro con probabilidad de fallo p. Un problema se puede presentar cuando nos encontramos no uno, sino n emisores que comparten el canal de comunicación con m receptores. Obviamente sólo uno de los emisores debe estar habilitado para comunicarse, en un momento determinado. En el marco de lenguajes puramente probabilísticos sería necesario decidir probabilísticamente cuál sería el emisor seleccionado para establecer la comunicación, lo cual no es una política sensata, sino que parece preferible dejar para una selección desconocida (el primero que llegue) la decisión de qué emisor debe comenzar la transmisión.

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

Alternating Bit Protocol 3:

$$ABP \equiv (A \mid S \mid Go \mid Ba \mid R)\backslash\{in_j, tm_{ijk}, dm_{jk}, ta_j, da_j, \lambda, \lambda', \lambda''\}$$

Los gráficos, de este modelo, y del modelo del caso 4, son iguales al del caso 2 a excepción de que ahora han de competir por el recurso canal, n emisores. Una vez se dispone del recurso, sabemos que el emisor i tardará $T = (1/\gamma_i + 2/\mu)/(1-p)^2$ en completar la comunicación, luego nuestro problema pasa de lleno al campo de teoría de colas. La cola para acceder al canal se ordena según el parámetro temporal de cada acción que compite en la elección interna que decide la adjudicación del recurso canal. Supongamos que la posición esperada por un emisor en la cola para acceder al canal es la $(j+1) - sima$, así, tardará en completar su comunicación el tiempo necesario para que la completen los

j emisores que le preceden en la cola, es decir, $\sum_{n=1}^{j}(1/\gamma_n + 2/\mu)/(1-p)^2$ más el tiempo para completarla él mismo que es $(1/\gamma_i + 2/\mu)/(1-p)^2$ totalizando:

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

Caso 4: Sistema de comunicaciones compuesto por n emisores, m receptores y 2 canales no seguros con probabilidad de fallo p. En esencia es igual al caso 3 a excepción de que necesitamos tantos estados para cada receptor como $2^{numero\ de\ canales}$, en nuestro caso suponemos que un emisor puede mantener simultáneamente la comunicación de tantos mensajes como canales disponga.

$$\begin{array}{ll}
\text{Activators:} & A_0 = in_{00}.in_0.in_{10}.in_0.A_0 \\
 & A_1 = in_{01}.in_1.in_{11}.in_1.A_1 \\
\text{Senders:} & S = \bigoplus_{i=1}^{n} S_{ik} \\
 & S_{ik} = in_{jg}.S_{ijgk} \\
 & S_{ijgk} = \langle tm_{ijgk}, \gamma_i\rangle.(da_{jg}.in_g.S + \lambda''_g.S_{ijgk}) \\
\text{Channels:} & Go_g = tm_{ijgk}.(\langle\lambda_g, \mu\rangle.Go_g \oplus_p \langle dm_{jgk}, \mu\rangle.Go_g) \\
 & Ba_g = ta_{jg}.(\langle\lambda''_g, \mu\rangle.Ba_g \oplus_p \langle da_{jg}, \mu\rangle.Ba_g) + \lambda'_g.\langle\lambda''_g, \mu\rangle.Ba_g \\
\text{Receivers:} & R = \lambda_0.\lambda'_0.R + \lambda_1.\lambda'_1.R \sum_{k=1}^{m} \sum_{j=0,g=0}^{1} dm_{jgk}.out_k.ta_{jg}.R^{jg} \\
 & R^{jg} = \lambda_0.\lambda'_0.R^{jg} + \lambda_1.\lambda'_1.R^{jg} \sum_{k=1}^{m} \sum_{j=0,g=0}^{1} \left\{ \begin{array}{ll} dm_{rsk}.ta_{rs}.R^{rs} & r = j \wedge s = g \\ dm_{rsk}.out_k.ta_{rs}.R^{rs} & e.o.c. \end{array} \right.
\end{array}$$

Alternating Bit Protocol 4:

$$\begin{array}{c} ABP \equiv (A_0 \mid A_1 \mid S \mid S \mid Go_0 \mid Ba_0 \mid Go_1 \mid Ba_1 \mid R \mid R)\backslash \\ \backslash\{in_{jg}, in_g, tm_{ijgk}, dm_{jgk}, ta_{jg}, da_{jg}, \lambda_g, \lambda'_g, \lambda''_g\} \end{array}$$

Su estudio es idéntico al del caso 3 excepto que el tiempo necesario para completar la comunicación los j emisores que le preceden es la mitad del tiempo necesario si hubiese un solo canal, así, tardará en completar su comunicación el tiempo necesario para que la completen los j emisores que le preceden en la cola, es decir, $(\sum_{n=1}^{j}(1/\gamma_n + 2/\mu)/(1-p)^2)/2$ más el tiempo para completarla él mismo que es $(1/\gamma_i + 2/\mu)/(1-p)^2$ totalizando:

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

Así pues, si dispusieramos de k canales obtendríamos un tiempo:

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

y en el límite cada emisor tardaría lo mismo que si dispusiera del canal exclusivamente.

6 Conclusiones y trabajos futuros

Hemos presentado un lenguaje algebraico, que combina aspectos clásicos de álgebras de procesos, ampliado con aspectos probabilísticos, que además está concebido para modelar y estudiar evaluación de prestaciones. Hemos iniciado el estudio de la semántica operacional así como un principio de algoritmo para la evaluación de prestaciones. Finalmente hemos aplicado estos resultados al estudio de 4 casos básicos del Protocolo de comunicaciones del Bit Alternante.

Nuestros trabajos próximos serán, en principio definir formalmente la totalidad del lenguaje y su semántica, incluyendo una definición de equivalencia por bisimulación. Además, definiremos formalmente los algoritmos que nos permitan la evaluación de prestaciones, y los aplicaremos a casos prácticos no triviales. Por otra parte, estos algoritmos serán susceptibles de ser implementados para obtener una herramienta de análisis de sistemas concurrentes.

Referencias

[BG98] Marco Bernardo and R. Gorrieri. A tutorial on empa: A theory of concurrent processes with nondeterminism, priorities, probabilities and time. *Theoretical Computer Science*, 202(1):1–54, 1998.

[BM89] B. Bloom and A.R. Meyer. A remark on bisimulation between probabilistic processes. In Meyers and Tsailin, editors, *Logik at Botik*. Springer-Verlag, 1989.

[BSW69] K.A. Bartlett, R.A. Scantlebury, and P.T. Wilkinson. A note on reliable full-duplex transmission over half-duplex links. *CACM*, 12(5):260–261, 1969.

[BVC97] 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.

[CCV98] D. Cazorla, F. Cuartero, and V. Valero. PNAL: Un modelo algebraico para procesos probabilísticos y no deterministas. In *Jornadas de Concurrencia*, Pamplona, 1998.

[CdFV96] 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.

[CdFV97] 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.

[Chr90] I. Christoff. *Testing Equivalence for Probabilistic Processes*. PhD thesis, Uppsala University, Uppsala, Sweden, 1990.

[GJS90] A. Giacalone, C.-C. Jou, and S.A. Smolka. Algebraic reasoning for probabilistic concurrent systems. In *Proceedings of Working Conference on Programming Concepts and Methods, IFIP TC 2*, Sea of Galilee, Israel, 1990.

[Hen88] M. Hennessy. *Algebraic Theory of Processes.* MIT Press, 1988.

[HJ90] H. Hansson and B. Jonsson. A calculus for communicating systems with time and probabilities. In *Proceedings Real-Time Systems Symposium*, Orlando, Florida, 1990.

[JL91] B. Jonson and K.G. Larsen. Specification and refinement of probabilistic processes. In *LICS'91*, Amsterdam, 1991.

[JS90] C. Jou and S.A. Smolka. Equivalences, congruences and complete axiomatizations for probabilistic processes. In *Proceedings of CONCUR'90, LNCS 458*, 1990.

[LOT88] *LOTOS. A formal description technique based on the temporal ordering of observational behaviour, IS 8807, TC97/SC21*, 1988.

[Low93] G. Lowe. Representing nondeterministic and probabilistic behaviour in reactive processes. Technical Report PRG-TR-11-93, Oxford university computing laboratory, 1993.

[LS89] K.G. Larsen and A. Skou. Bisimulation through probabilistic testing. In *Proceedings of 16th ACM Symp. on Principles of Programming Languages*, Austin, TX, 1989.

[Mil89] R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.

[NdFL95] M. Núñez, D. de Frutos, and L. Llana. Acceptance trees for probabilistic processes. In *CONCUR'95, LNCS 962*, pages 249–263, 1995.

[Ort90] Y. Ortega. *En busca del tiempo perdido.* PhD thesis, Universidad Complutense de Madrid, 1990.

[QAdF89] J. Quemada, A. Azcorra, and D. de Frutos. A timed calculus for LOTOS. In *FORTE'89*, Vancouver, Canada, 1989.

[RR88] G.M. Reed and A.W. Roscoe. A timed model for communicating sequential processes. *Theoretical Computer Science*, 58:249–261, 1988.

[vGSST90] R. van Glabbeek, S.A. Smolka, B.U. Steffen, and C.M.N. Tofts. Reactive, generative ans stratified models of probabilistic processes. In *Proceedings of 5th Annual IEEE Symposium on Logic in Computer Science*, pages 130–141, Philadelphia, PA, 1990.

[Yi91] W. Yi. Towards a theory of testing for CCS with probability. In *Proc. Chalmers Workshop on Concurrency*, 1991.

[Yi94] W. Yi. Algebraic reasoning for real-time probabilistic processes with uncertainty information. In *Formal Techniques in Real-Time and Fault-Tolerant Systems, LNCS 863*, pages 680–693, Kiel, Germany, 1994.

[YL92] W. Yi and K.G. Larsen. Testing probabilistic and nondeterministic processes. In *Protocol Specification, Testing and Verification XII*, pages 47–61, 1992.