

Redes Reconfigurables Controladas por Marcado: Redes de Petri con Cambios Dinámicos Estructurales*

Marisa Llorens¹ and Javier Oliver¹

DSIC, UPV, Camino de Vera s/n, E-46022 Valencia, Spain
{mllorens,fjoliver}@dsic.upv.es

Resumen El objetivo de este trabajo es la modelización y verificación de sistemas concurrentes sujetos a cambios dinámicos utilizando extensiones de redes de Petri. En estudios previos, hemos introducido los sistemas de reescritura de redes y una subclase de los mismos llamada redes reconfigurables. En un sistema de reescritura de redes, una configuración del sistema se describe como una red de Petri y un cambio de configuración como una regla de reescritura de grafos. Una red reconfigurable es un sistema de reescritura de redes en el que un cambio de configuración sólo implica la modificación de las relaciones de flujo de los lugares en el dominio de la regla en cuestión. En ambos modelos, el que una regla esté habilitada depende únicamente de la topología de la red. Aquí vamos a introducir los sistemas de reescritura de redes controladas por marcado y las redes reconfigurables controladas por marcado, en las que la habilitación de una regla también va a depender del marcado de la red. Vamos a presentar una implementación de redes reconfigurables controladas por marcado con redes de Petri.

1. Introducción

Una red de Petri [15,16] es un formalismo utilizado para modelizar, analizar, simular, controlar y evaluar el comportamiento de sistemas concurrentes y distribuidos. Este formalismo, sin embargo, no ofrece una manera directa de modelizar características tales como cambios dinámicos, modos de operación múltiples, etc. Se han diseñado extensiones de redes de Petri para permitir una formalización sencilla de tales características. En general, en dichas extensiones, lo que se gana en términos de poder de modelado se pierde en términos de decidibilidad de propiedades [7]. Hay que buscar, entonces, un equilibrio entre expresividad y computabilidad. El objetivo fundamental de este trabajo es la modelización, simulación y verificación de sistemas concurrentes sujetos a cambios dinámicos. El mecanismo que maneja los cambios dinámicos en tales sistemas debe estar explícitamente representado en el propio modelo de forma que en cada una de las

* Este trabajo ha sido parcialmente financiado por CICYT TIC 2001-2705-C03-01, por Acción Integrada Hispano-Alemana HA2001-0059 y por Proyecto de Investigación UPV 7176.

etapas de desarrollo del producto, los diseñadores pueden experimentar con los efectos de los cambios estructurales (utilizando prototipos). Esto significa que los cambios en la estructura se tienen en cuenta desde el principio del proceso de diseño en lugar de llevarse a cabo por un sistema externo, global (mediante algún mecanismo de manejo de excepciones), diseñado y añadido al modelo describiendo el comportamiento normal del sistema.

El modelo de los *sistemas de reescritura de redes* introducido en [3,11,12,13] surge a partir de dos líneas de investigación diferentes, ambas relacionadas con el formalismo de las redes de Petri y cuyo propósito es mejorar la expresividad del modelo básico de las redes de Petri para soportar la descripción de sistemas concurrentes sujetos a cambios dinámicos. La primera clase de modelos abarca varias propuestas que fusionan redes de Petri con *gramáticas de grafos* [4,6,17] mientras que la segunda clase, representada por las *redes automodificantes de Valk* [18,19], considera redes de Petri cuyas relaciones de flujo pueden variar en tiempo de ejecución. Ambas propuestas conducen a modelos expresivos que tienen indudables ventajas con respecto al resto de modelos existentes. Sin embargo, la mayoría de las propiedades básicas decidibles de las redes de Petri (acotabilidad de lugares, alcanzabilidad, interbloqueo y vivacidad) se pierden para estos modelos extendidos. Como consecuencia, no pueden construirse herramientas automáticas de verificación para estos modelos. Las *redes reconfigurables*, que fueron introducidas en [3,11,12,13] como una subclase particular de los sistemas de reescritura de redes, intentan combinar los aspectos más relevantes de ambas aproximaciones y constituyen una clase de modelos para la que todas las propiedades fundamentales anteriores son decidibles. La traducción de este modelo a redes de Petri es automática [12,13]. Esta equivalencia asegura que todas las propiedades fundamentales de las redes de Petri siguen siendo decidibles para las redes reconfigurables, con lo que en este modelo es factible la verificación automática. Por el contrario, la clase de los sistemas de reescritura de redes tiene el poder computacional de la máquina de Turing [12,13] y, por tanto, para esta clase no es posible encontrar herramientas de verificación automática.

En los *sistemas de reescritura de redes*, una configuración de un sistema se describe como una red de Petri y un cambio de configuración se describe como una regla de reescritura de grafos que consiste en reemplazar parte del sistema (que hace *matching* con la parte izquierda de la regla de reescritura) por otra (que viene dada por la parte derecha de la regla). Una *red reconfigurable* es un sistema de reescritura de redes en el que un cambio de configuración se limita a la modificación de las relaciones de flujo de los lugares implicados en la regla de reescritura. En otras palabras, para que una regla de reescritura esté habilitada y, entonces, pueda tener lugar un cambio de configuración, la topología de la red es lo único a tener en cuenta. Sin embargo, la mayoría de los cambios de configuración que ocurren en los sistemas reales dependen del estado del sistema (representado en una red por su marcado). Sería pues interesante que la habilitación de una regla de reescritura no sólo dependiera de la topología de la red sino también del marcado de la misma. Básicamente, la idea es añadir un mecanismo de control a los sistemas de reescritura de redes y a las redes

reconfigurables tal que en la red sólo puedan ocurrir cambios de configuración cuando se alcance cierto marcado mínimo.

En la Sección 2 se introducen los sistemas de reescritura de redes controladas por marcado. En la Sección 3, se presenta la definición de redes reconfigurables controladas por marcado y un ejemplo detallado. Una implementación de redes reconfigurables controladas por marcado con redes de Petri se muestra en la Sección 4. Finalmente, se describen algunos trabajos relacionados y se presentan las conclusiones en la Sección 5.

2. Sistemas de Reescritura de Redes Controladas por Marcado

En esta sección se introduce el modelo de los *sistemas de reescritura de redes controladas por marcado*. Para ello necesitamos las siguientes definiciones previas:

Si $R \subseteq X \times Y$ es una relación binaria, se dice que $X'R = \{y \in Y \mid \exists x \in X' (x, y) \in R\}$ denota la *imagen* de $X' \subseteq X$, y $RY' = \{x \in X \mid \exists y \in Y' (x, y) \in R\}$ denota la *imagen inversa* de $Y' \subseteq Y$. El *dominio* de R es, entonces, $Dom(R) = RY$ y el *codominio* de R es $Cod(R) = XR$.

Una *red de Petri* [15,16] es una tupla $\mathcal{N} = (P, T, F)$ donde: $P = \{p_1, p_2, \dots, p_m\}$ es un conjunto finito de lugares, $T = \{t_1, t_2, \dots, t_n\}$ es un conjunto finito de transiciones ($P \cap T = \emptyset$, $P \cup T \neq \emptyset$) y $F : (P \times T) \cup (T \times P) \rightarrow \{0, 1, 2, 3, \dots\}$ es un conjunto de arcos ponderados (relación de flujo). Una *red de Petri marcada* es un par (\mathcal{N}, M_0) donde \mathcal{N} es una red de Petri y $M_0 : P \rightarrow \{0, 1, 2, 3, \dots\}$ es el marcado inicial.

Sean $\mathcal{N} = (P, T, F)$ y $\mathcal{N}' = (P', T', F')$ dos redes de Petri. Decimos que \mathcal{N} y \mathcal{N}' son *isomorfas* si existe una biyección $\varphi : (P \cup T) \rightarrow (P' \cup T')$ tal que $F(x, y) \in \mathcal{N} = F'(\varphi(x), \varphi(y)) \in \mathcal{N}'$ para todo $x, y \in P \cup T$.

Un *embedding total* de una red de Petri $\mathcal{N} = (P, T, F)$ en una red de Petri $\mathcal{N}' = (P', T', F')$ es una *función inyectiva* $f : P \cup T \rightarrow P' \cup T'$ que hace corresponder lugares con lugares y transiciones con transiciones ($f(P) \subseteq P'$ y $f(T) \subseteq T'$) tal que para cualquier par de elementos $x, y \in P \cup T$, $F(x, y) = F'(f(x), f(y))$. La imagen de \mathcal{N} por f se denomina entonces *subred total* de \mathcal{N}' .

Definición 1. *Un sistema de reescritura de redes controladas por marcado es una estructura $N = (\mathcal{R}, (\Gamma_0, M_0))$ en la que $\mathcal{R} = \{r_1, \dots, r_h\}$ es un conjunto finito de reglas de reescritura y (Γ_0, M_0) es una red de Petri marcada.*

Una regla de reescritura $r \in \mathcal{R}$ es una estructura $r = (L, R, \tau, \bullet\tau, \tau\bullet, C, \mathbb{M})$ donde:

- $L = (P_L, T_L, F_L)$ y $R = (P_R, T_R, F_R)$ son redes de Petri llamadas parte izquierda y parte derecha de r , respectivamente;
- $\tau \subseteq (P_L \times P_R) \cup (T_L \times T_R)$, llamada relación de transferencia de r , es una relación binaria que relaciona lugares de L con lugares de R y transiciones de L con transiciones de R : $P_L\tau \subseteq P_R$, $\tau P_R \subseteq P_L$, $T_L\tau \subseteq T_R$ y $\tau T_R \subseteq T_L$;

- $\bullet\tau \subseteq \tau$, y $\tau^\bullet \subseteq \tau$ son subrelaciones de la relación de transferencia llamadas relación interfaz de entrada y relación interfaz de salida;
- C es un conjunto finito de lugares, llamados lugares de control, que es un subconjunto de los lugares de L , $C \subseteq P_L$; y
- \mathbb{M} es el marcado mínimo de los lugares de C requerido para que la regla esté habilitada.

En un sistema de reescritura de redes controladas por marcado N : una configuración es una red de Petri $\Gamma = (P, T, F)$; un estado es una red de Petri marcada (Γ, M) , siendo (Γ_0, M_0) el estado inicial; y un evento es o bien una transición o bien una regla de reescritura: $E = T \cup R$.

Definición 2. Un sistema de reescritura de redes [3,11,12,13] es un sistema de reescritura de redes controladas por marcado cuyo conjunto de lugares de control C es el conjunto vacío ($C = \emptyset$) para todas las reglas de reescritura, es decir, no existe ninguna restricción sobre el marcado de la red.

Para poder aplicar una regla de reescritura r en una configuración Γ , debemos en primer lugar identificar una subred total Γ' de Γ que sea isomorfa a la parte izquierda de la regla; esto es, que exista una biyección $\varphi : (P_L \cup T_L) \rightarrow (P' \cup T')$ tal que $F_L(x, y) \in L = F'(\varphi(x), \varphi(y)) \in \Gamma'$ para todo $x, y \in P_L \cup T_L$. Los elementos de Γ (lugares o transiciones) que no pertenezcan a Γ' constituyen el *contexto* de la regla. Se requiere también que $\forall p \in C, M(\varphi(p)) \geq \mathbb{M}(p)$. Para que la regla esté habilitada, se requiere además que un elemento x' de Γ' tenga un elemento x de su *preset* que pertenezca al contexto sólo si x' pertenece al *interfaz de entrada* de la regla y, simétricamente, que un elemento x' de Γ' tenga un elemento x de su *postset* que pertenezca al contexto sólo si x' pertenece al *interfaz de salida* de la regla. Cuando estas condiciones se dan, la reescritura puede tener lugar y se reemplaza la subred Γ' por la parte derecha R de la regla, fijándose las conexiones entre los elementos de R y los del contexto de acuerdo con la relación interfaz. La relación de transferencia no sólo se utiliza para reescribir la parte izquierda por la parte derecha de la regla sino también para transferir los tokens de Γ' a R . Nótese que, ya que la relación de transferencia puede ser cualquier tipo de relación, los tokens pueden duplicarse o pueden, incluso, desaparecer.

La evolución dinámica de un sistema de reescritura de redes controladas por marcado viene dada por su grafo de estados.

Definición 3. El grafo de estados de un sistema de reescritura de redes controladas por marcado $N = (\mathcal{R}, (\Gamma_0, M_0))$ es el grafo dirigido etiquetado cuyos nodos son los estados de N , es decir, redes de Petri marcadas y cuyos arcos (etiquetados con eventos de N) son de dos clases:

- disparo de una transición: arcos desde el estado (Γ, M) al estado (Γ', M') etiquetados con la transición t cuando la transición t puede dispararse en la red Γ con marcado M y lleva al marcado M' :

$$(\Gamma, M) \xrightarrow{t} (\Gamma', M') \iff (\Gamma = \Gamma' \text{ y } M[t]M' \text{ en } \Gamma).$$

- cambio de configuración: arcos desde el estado (Γ, M) al estado (Γ', M') etiquetados con la regla $r = (L, R, \tau, \bullet\tau, \tau^\bullet, C, \mathbb{M}) \in \mathcal{R}$, cuando:

- existe un embedding total $f : L \rightarrow \Gamma$ tal que $\forall x \notin f(L)$ e $y \in L$:

$$x \in \bullet f(y) \Rightarrow y \in \text{Dom}(\bullet\tau) \text{ y } x \in f(y)\bullet \Rightarrow y \in \text{Dom}(\tau\bullet)$$

- $\forall p \in C, M(f(p)) \geq \mathbb{M}(p)$ y
- siendo $\Gamma = (P, T, F)$ y $\Gamma' = (P', T', F')$, se cumple lo siguiente:

$$P' = P - f(P_L) + P_R \text{ tal que } P_L\tau = P_R$$

$$T' = T - f(T_L) + T_R \text{ tal que } T_L\tau = T_R,$$

donde el significado de $+$ ($-$) es añadir (eliminar) lugares/transiciones a (de) Γ . Los nombres de los lugares P_R (transiciones T_R) añadidos a Γ deben ser nombres nuevos para evitar conflictos.

La relación de flujo F' viene dada por

$$F'(x, y) = \begin{cases} F(x, y) & \text{si } x \notin R \wedge y \notin R \\ F_R(x, y) & \text{si } x \in R \wedge y \in R \\ \sum_{y_i \in \bullet\tau y} F(x, f(y_i)) & \text{si } x \notin R \wedge y \in R \\ \sum_{x_i \in \tau\bullet x} F(f(x_i), y) & \text{si } x \in R \wedge y \notin R \end{cases}$$

El marcado de un lugar $p \in P'$ es

$$M'(p) = \begin{cases} M(p) & \text{si } p \notin R \\ \sum_{p' \in \tau p} M(f(p')) & \text{si } p \in R \end{cases}$$

El siguiente ejemplo ilustra la definición de sistema de reescritura de redes controladas por marcado y el significado de cambio de configuración.

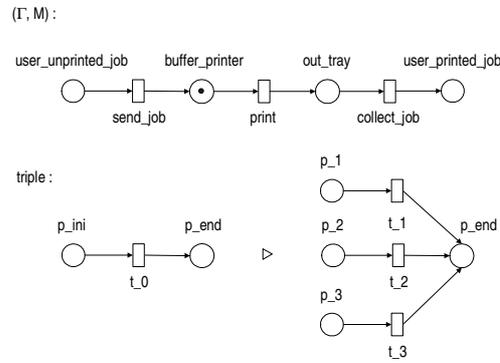


Figura 1. Sistema de Reescritura de Redes Controladas por Marcado modelizando un sistema de impresoras

Ejemplo 1. El sistema de reescritura de redes controladas por marcado de la Fig. 1 modela la impresión de varias copias de un mismo trabajo en distintas

impresoras. El token en el lugar $buffer_printer$ representa una copia del trabajo a imprimir. En este estado (Γ, M) , para obtener tres copias, el trabajo debe enviarse a imprimir tres veces y las tres copias sólo pueden imprimirse secuencialmente (una detrás de otra). La regla de reescritura $triple$ ofrece la posibilidad de imprimir cada una de ellas en una impresora distinta (asumiendo que hay algún trabajo para imprimir). El subconjunto de lugares de control es $C = \{p_ini\}$ y $\mathbb{M}(p_ini) = 1$. La relación de transferencia τ viene dada por $\tau = \{(\{p_ini\}, \{p_1, p_2, p_3\}), (\{t_0\}, \{t_1, t_2, t_3\}), (\{p_end\}, \{p_end\})\}$ y las relaciones interfaz de entrada e interfaz de salida son, en este caso, $\bullet\tau = \{(\{p_ini\}, \{p_1, p_2, p_3\})\}$ y $\tau\bullet = \{(\{p_end\}, \{p_end\})\}$, respectivamente. La Figura 2 muestra el nuevo estado (Γ', M') debido al cambio de configuración causado por la regla de reescritura.

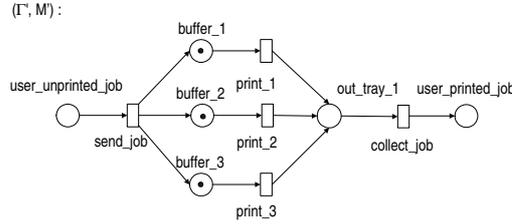


Figura 2. Cambio de configuración debido a la regla de reescritura $triple$

Proposición 1. *Los sistemas de reescritura de redes controladas por marcado tienen el mismo poder computacional que la máquina de Turing.*

Demostración. Se sigue de [12,13].

3. Redes Reconfigurables Controladas por Marcado

Las redes reconfigurables son una subclase de sistemas de reescritura de redes cuya relación de transferencia τ es una biyección [3,11,12,13]. En otras palabras, el conjunto de lugares y transiciones no sufre cambios por la aplicación de reglas de reescritura en las redes reconfigurables. Tales reglas están habilitadas si y sólo si la red tiene cierta topología. Además, dichas reglas únicamente cambian las relaciones de flujo de los lugares en sus dominios. Introducimos las redes reconfigurables controladas por marcado como una subclase de sistemas de reescritura de redes controladas por marcado cuya relación de transferencia τ es una biyección. Las redes reconfigurables controladas por marcado son una extensión de las redes reconfigurables [3,11,12,13] en la que la habilitación de una regla de reescritura también depende del marcado de la red.

Definición 4. *Una red reconfigurable controlada por marcado es una estructura $N = (P, T, \mathcal{R}, \gamma_0)$ siendo $P = \{p_1, \dots, p_n\}$ un conjunto finito y no vacío de*

lugares, $T = \{t_1, \dots, t_m\}$ un conjunto finito y no vacío de transiciones disjunto de P ($P \cap T = \emptyset$), $\mathcal{R} = \{r_1, \dots, r_h\}$ un conjunto finito de reglas de reescritura, y γ_0 el estado inicial.

Una regla de reescritura $r \in \mathcal{R}$ es una estructura $r = (D, \bullet r, r^\bullet, C, \mathbb{M})$ donde:

- $D \subseteq P$ es el dominio de r ,
- $\bullet r : (D \times T) \cup (T \times D) \rightarrow \mathbb{N}$ y $r^\bullet : (D \times T) \cup (T \times D) \rightarrow \mathbb{N}$ son las precondiciones y postcondiciones de r , (es decir, son las relaciones de flujo de los lugares del dominio antes y después del cambio de configuración debido a la regla r).
- C es un subconjunto de lugares de D , $C \subseteq D$, llamados lugares de control, y
- \mathbb{M} es el marcado mínimo requerido a los lugares de C para que la regla pueda habilitarse.

En una red reconfigurable controlada por marcado: una configuración es una red de Petri $\Gamma = (P, T, F)$; un estado γ es una red de Petri marcada $\gamma = (I, M)$ y su conjunto de eventos lo forman sus transiciones junto con sus reglas de reescritura: $E = T \cup \mathcal{R}$.

Definición 5. Una red reconfigurable [3,11,12,13] es una red reconfigurable controlada por marcado cuyo conjunto de lugares de control C es el conjunto vacío ($C = \emptyset$) para todas las reglas de reescritura; es decir, no existe ninguna restricción sobre el marcado de la red.

Representamos una regla de reescritura utilizando la notación de sumas formales como sigue:

$$r = \sum_{p \in D} p (\sum_{t \in T} \bullet r(p, t) \cdot t - \sum_{t \in T} \bullet r(t, p) \cdot t) \triangleright \sum_{p \in D} p (\sum_{t \in T} r^\bullet(p, t) \cdot t - \sum_{t \in T} r^\bullet(t, p) \cdot t)$$

Definición 6. El grafo de configuración $G(N)$ de una red reconfigurable controlada por marcado $N = (P, T, \mathcal{R}, \gamma_0)$ es el grafo dirigido etiquetado cuyos nodos son las configuraciones, y tal que existe un arco desde la configuración Γ a la configuración Γ' etiquetado con la regla $r = (D, \bullet r, r^\bullet, C, \mathbb{M}) \in \mathcal{R}$, que denotamos $\Gamma[r]\Gamma'$, si y solamente si se cumple lo siguiente:

- $\forall p \in C, M(p) \geq \mathbb{M}(p)$;
- $\forall p \in D, F(p, t) = \bullet r(p, t)$ y $F(t, p) = \bullet r(t, p)$, $F'(p, t) = r^\bullet(p, t)$ y $F'(t, p) = r^\bullet(t, p)$;
- $\forall p \notin D, F(p, t) = F'(p, t)$ y $F(t, p) = F'(t, p)$.

Nótese que se requiere que los lugares de control estén marcados con al menos el marcado \mathbb{M} . La relación de transición debe contener arcos de la misma multiplicidad que los arcos de la parte izquierda de la regla de reescritura, y no se permite la reescritura si existen arcos de una multiplicidad superior.

La evolución dinámica de una red reconfigurable controlada por marcado viene dada por su grafo de estados.

Definición 7. El grafo de estados de una red reconfigurable controlada por marcado $N = (P, T, \mathcal{R}, \gamma_0)$ es el grafo dirigido etiquetado cuyos nodos son los estados de N y cuyos arcos (etiquetados con eventos) son de dos clases:

- disparo de una transición: arcos desde el estado (Γ, M) al estado (Γ, M') etiquetados con la transición t cuando la transición t puede dispararse en la red Γ con marcado M y conduce al marcado M' ,
- cambio de configuración: arcos desde el estado (Γ, M) al estado (Γ', M) etiquetados con la regla $r \in \mathcal{R}$ si $\Gamma[r]\Gamma'$ es una transición del grafo de configuración de N .

Dicho de otro modo, el conjunto de arcos etiquetados del grafo de estados de N viene dado por

$$\{(\Gamma, M) \xrightarrow{t} (\Gamma, M') \mid M[t]M' \text{ en } \Gamma\} \cup \{(\Gamma, M) \xrightarrow{r} (\Gamma', M) \mid \Gamma[r]\Gamma' \text{ en } G(N)\}$$

El siguiente ejemplo muestra un sistema modelizado mediante una red reconfigurable controlada por marcado y un cambio de configuración que depende de la topología de la red y que también depende del marcado de la red.

Ejemplo 2 (Red de Transmisión). La Figura 3 es el estado inicial $\gamma_0 = (\Gamma_0, M_0)$ de una red reconfigurable controlada por marcado que representa una red de transmisión en la que desde dos máquinas se reciben bloques de información (representadas por las transiciones t_1 y t_8) para ser gestionados por un servidor principal (representado por el lugar p_7). El marcado inicial es $M_0 = (1, 2, 0, 1, 0, 0, 1)$.

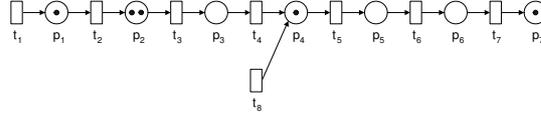


Figura 3. Estado inicial de la red de transmisión

La red tiene dos partes diferenciadas que están delimitadas por los lugares p_1 , p_3 y p_7 . En cada parte pueden ocurrir varios cambios de configuración, representados mediante reglas de reescritura. En la primera parte, los paquetes pueden enviarse de uno en uno, de dos en dos, de tres en tres o de cuatro en cuatro, dependiendo del peso del arco desde el lugar p_2 a la transición t_3 . En la segunda parte, la red ofrece tres posibilidades para la distribución de datos por la red: paquetes que siguen el camino normal $t_4 p_4 t_5 p_5 t_6 p_6 t_7 p_7$; paquetes que evitan la transición t_5 y siguen el camino $t_4 p_4 t_6 p_6 t_7 p_7$; y finalmente, paquetes que van directamente a través del camino $t_4 p_4 t_7 p_7$, evitando las transiciones t_5 y t_6 . Definimos ocho reglas de reescritura utilizando la notación de sumas formales introducida previamente. Mostramos la última (R_8) gráficamente en la Fig. 4.

$$\begin{array}{ll} R_1: p_2(t_2-t_3) \triangleright p_2(t_2-2t_3) & R_2: p_2(t_2-t_3) \triangleright p_2(t_2-3t_3) \\ R_3: p_2(t_2-t_3) \triangleright p_2(t_2-4t_3) & R_4: p_2(t_2-2t_3) \triangleright p_2(t_2-3t_3) \\ R_5: p_2(t_2-2t_3) \triangleright p_2(t_2-4t_3) & R_6: p_2(t_2-3t_3) \triangleright p_2(t_2-4t_3) \end{array}$$

$$R_7: p_4(t_4+t_8-t_5)+p_5(t_5-t_6) \triangleright p_4(t_4+t_8-t_6)+p_5(\emptyset)$$

$$R_8: p_4(t_4+t_8-t_5)+p_5(t_5-t_6)+p_6(t_6-t_7) \triangleright p_4(t_4+t_8-t_7)+p_5(\emptyset)+p_6(\emptyset)$$

El subconjunto de lugares de control C es el conjunto vacío para todas las reglas de reescritura excepto para las reglas R_7 y R_8 donde $C = \{p_4\}$ y $M(p_4) = 1$ (es decir, para cambiar de camino en la segunda parte de la red, debe haber al menos un paquete en el lugar p_4).

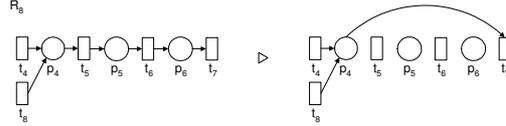


Figura 4. Regla de reescritura R_8

Así pues, la red reconfigurable controlada por marcado consta de 7 lugares, 8 transiciones y 8 reglas de reescritura. La Figura 5 muestra el estado alcanzado cuando la regla R_8 se aplica al estado inicial representado en la Fig. 3. En este nuevo estado, los paquetes se envían de uno en uno y siguen el camino $t_4p_4t_7p_7$, evitando las transiciones t_5 y t_6 .

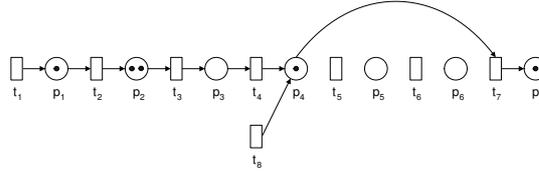


Figura 5. Nuevo estado alcanzado tras la aplicación de la regla R_8 al estado de la Figura 3

4. Implementación de Redes Reconfigurables Controladas por Marcado con Redes de Petri

Queremos probar que las redes reconfigurables controladas por marcado son equivalentes a las redes de Petri. A primera vista, puede parecer que no lo son debido al conjunto de reglas de reescritura de las redes reconfigurables controladas por marcado. Cuando se aplica una regla de reescritura en una de estas redes, tiene lugar un cambio de configuración (por $\Gamma_i[r]\Gamma_j$ denotamos el cambio de configuración debido a la regla de reescritura r desde la configuración Γ_i a la configuración Γ_j), es decir, un cambio en la estructura de la red. Obtener una red de Petri equivalente significa que esos cambios de configuración estén

presentes (es decir, todas las posibles configuraciones deben estar representadas en la red). Se deduce pues, que el número de configuraciones debe ser finito para que sea posible su representación.

Por $Conf(N) = \{\Gamma_0, \Gamma_1, \dots, \Gamma_k\}$ denotamos el conjunto de configuraciones de una red reconfigurable controlada por marcado N , donde Γ_0 es la configuración del estado inicial $\gamma_0 = (\Gamma_0, M_0)$ (la configuración inicial). Podemos entonces obtener fácilmente una red de Petri equivalente $\tilde{N} = (\tilde{P}, \tilde{T}, \tilde{F}, \tilde{M}_0)$ cuyo conjunto de lugares será $\tilde{P} = P \cup \{q_0, \dots, q_k\}$, es decir, los lugares de la red reconfigurable controlada por marcado original junto con un lugar específico asociado a cada posible configuración. Tendremos tantas copias del conjunto de transiciones como configuraciones haya, o sea $\{q_0, \dots, q_k\} \times T$. Podemos imaginar los lugares y las transiciones de una configuración $\Gamma_i = (P, T, F_i)$ como si estuvieran situados en dos planos paralelos distintos (es decir, un plano con el conjunto de lugares y un plano con el conjunto de transiciones conectados por las relaciones de flujo de la configuración representada). Entonces establecemos las relaciones de flujo de forma que la configuración Γ_i esté representada por el plano de transiciones $\{q_i\} \times T$ y el plano (compartido) de lugares P : $\tilde{F}(p, (q_i, t)) = F_i(p, t)$ y $\tilde{F}((q_i, t), p) = F_i(t, p)$ siendo $\Gamma_i = (P, T, F_i)$. El lugar q_i , asociado a la configuración Γ_i , contiene como mucho un token, y está marcado en los estados asociados a dicha configuración. Así pues, las relaciones de flujo serán $\tilde{F}(q_i, (q_j, t)) = 1$ si $i = j$ y $\tilde{F}(q_i, (q_j, t)) = 0$ en otro caso; $\tilde{F}((q_j, t), q_i) = 1$ si $i = j$ y $\tilde{F}((q_j, t), q_i) = 0$ en otro caso.

Sólo nos queda representar el cambio de configuración. Para este propósito, es suficiente añadir una transición extra r_{ij} para cada cambio de configuración $\Gamma_i[r] \Gamma_j$ cuya relación de flujo viene dada por $\tilde{F}(q_i, r_{ij}) = \tilde{F}(r_{ij}, q_j) = 1$, $\tilde{F}(p, r_{ij}) = \tilde{F}(r_{ij}, p) = \mathbb{M}(p)$ si $p \in C$ y $\tilde{F}(p, r_{ij}) = \tilde{F}(r_{ij}, p) = 0$, en otro caso. Así, el conjunto de transiciones es $\tilde{T} = (\{q_0, \dots, q_k\} \times T) \cup \tilde{R}$ donde \tilde{R} es el conjunto de transiciones tal que $r_{ij} \in \tilde{R}$ si $\exists r \in R$ tal que $\Gamma_i[r] \Gamma_j$ en $G(N)$. La red de Petri resultante está marcada inicialmente como: $\tilde{M}_0(p) = M_0(p)$ donde $p \in P$ y $\tilde{M}_0(q_i) = \begin{cases} 1 & \text{si } i = 0 \\ 0 & \text{en otro caso} \end{cases}$.

Para demostrar la correspondencia entre redes reconfigurables controladas por marcado y redes de Petri deben tenerse en cuenta algunas consideraciones:

- Si $\tilde{M} : \tilde{P} \rightarrow \mathbb{N}$ es un marcado alcanzable de \tilde{N} entonces $\sum_{i=0}^k \tilde{M}(q_i) = 1$.

Por tanto, en el conjunto de transiciones $\{q_0, \dots, q_k\} \times T \in \tilde{T}$ únicamente las transiciones de $\{q_i \times T\}$ están habilitadas.

- Denotamos $\gamma_{\tilde{M}} = (\Gamma_{\tilde{M}}, M)$ donde $M : P \rightarrow \mathbb{N}$ es un marcado de $\Gamma_{\tilde{M}}$ y $\Gamma_{\tilde{M}} = \Gamma_i$ es la configuración asociada a \tilde{M} tal que $\tilde{M}(q_i) = 1$. Inversamente, si

$\gamma = (G, M)$ es un estado alcanzable de N , asociamos la aplicación $\tilde{M}_\gamma : \tilde{P} \rightarrow \mathbb{N}$ con $\tilde{M}_\gamma(p) = M(p)$ si $p \in P$, $\tilde{M}_\gamma(q_i) = 1$ si $G = G_i$ y $\tilde{M}_\gamma(q_i) = 0$ en otro caso.
- $\tilde{M}_{(\gamma_{\tilde{M}})} = \tilde{M}$, $\gamma_{(\tilde{M}\gamma)} = \gamma$, $\gamma_{(\tilde{M}_0)} = (G_0, M_0)$ y $\tilde{M}_{\gamma_0} = \tilde{M}_0$.

Proposición 2. *Si $\gamma = (G, M)$ es un estado alcanzable de N entonces*

1. $\gamma[r]\gamma' \iff \tilde{M}_\gamma[r_{ij}]\tilde{M}_{\gamma'}$ siendo $\gamma = (G_i, M)$ y $\gamma' = (G_j, M)$.
2. $\gamma[t]\gamma' \iff \tilde{M}_\gamma[(q_i, t)]\tilde{M}_{\gamma'}$ siendo $\gamma = (G_i, M)$ y $\gamma' = (G_i, M')$.

La proposición previa muestra que:

1. Si cambiamos del estado $\gamma = (G_i, M)$ al estado $\gamma' = (G_j, M)$ en una red reconfigurable controlada por marcado debido al disparo de una regla de reescritura, lo que difiere en la red de Petri equivalente es el marcado de los lugares q_i y q_j :

$$\tilde{M}_\gamma(q_i) = 1 \implies \tilde{M}_{\gamma'}(q_i) = 0 \quad \text{y} \quad \tilde{M}_\gamma(q_j) = 0 \implies \tilde{M}_{\gamma'}(q_j) = 1.$$

De igual modo en el sentido contrario.

2. Si cambiamos del estado $\gamma = (G_i, M)$ al estado $\gamma' = (G_i, M')$ en una red reconfigurable controlada por marcado debido al disparo de una transición t , lo que cambia en la red de Petri equivalente es el marcado de los lugares $p \in P$ implicados en el disparo de la transición (q_i, t) , y viceversa.

Por tanto, se puede establecer que:

Proposición 3. *Los marcados \tilde{M}_γ en los que γ cubre los estados alcanzables de N son marcados alcanzables de \tilde{N} , y el grafo de marcado de \tilde{N} es isomorfo al grafo de marcado de N . Es este sentido, toda red reconfigurable controlada por marcado es equivalente a una red de Petri.*

De este modo, las redes reconfigurables controladas por marcado son equivalentes a las redes de Petri, pero proporcionan una forma más compacta de representar sistemas concurrentes cuya estructura evoluciona en tiempo de ejecución. Esto puede observarse en la Fig. 6, que ilustra un fragmento de la red de Petri equivalente a la red reconfigurable controlada por marcado del Ejemplo 2. Para facilitar la comprensión de la implementación de una red reconfigurable controlada por marcado con una red de Petri, sólo mostramos cómo representar un cambio de configuración (el cambio de camino en la segunda parte de la red evitando las transiciones t_5 y t_6 -regla de reescritura R_8 -) y sólo para los lugares y las transiciones implicadas. Con este pequeño fragmento, podemos imaginar el tamaño de la red de Petri completa y cómo podemos modelizar mucho mejor el sistema con una red reconfigurable controlada por marcado. En general, si la red reconfigurable controlada por marcado original $N = (P, T, \mathcal{R}, \gamma_0)$ tiene n lugares, m transiciones, r reglas de reescritura y el número de configuraciones accesibles es $k + 1$, la red de Petri equivalente obtenida $\tilde{N} = (\tilde{P}, \tilde{T}, \tilde{F}, \tilde{M}_0)$ consta de:

$n + (k + 1)$ lugares y $((k + 1) * m) + z$ transiciones, donde $z = \sum_{i=0}^k (|I_i^\bullet| + |\bullet I_i|)$ y $|I_i^\bullet|$ ($|\bullet I_i|$) es el número de arcos de salida (entrada) desde (hacia) la configuración I_i en el grafo de configuración de N , $G(N)$. Es decir, z es el número de transiciones de \tilde{R} .

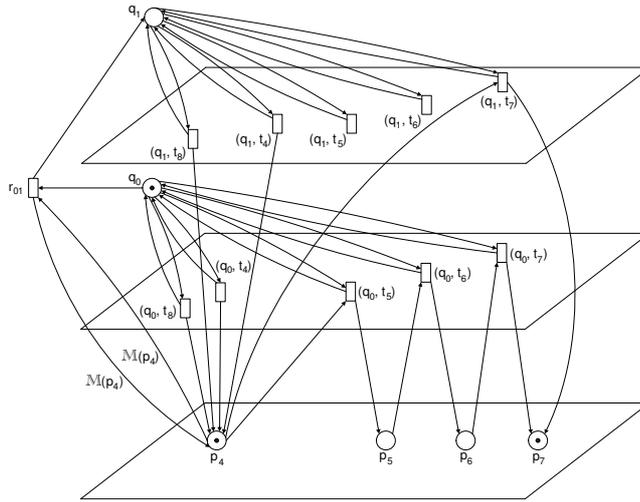


Figura 6. Parte de la red de Petri equivalente a la red reconfigurable controlada por marcado del Ejemplo 2

5. Trabajos Relacionados y Conclusiones

En estudios previos [3,11,12,13], hemos introducido los sistemas de reescritura de redes, que son una extensión de las redes de Petri apropiada para la modelización, simulación y verificación de sistemas concurrentes sujetos a cambios dinámicos. Los sistemas de reescritura de redes pueden modificar dinámicamente su propia estructura reescribiendo alguna de sus componentes. Esta reescritura depende solamente de la topología de la red. Aquí hemos introducido los sistemas de reescritura de redes controladas por marcado, una extensión de los sistemas de reescritura de redes donde la reescritura depende también del marcado de la red. Ambos modelos están basados en dos líneas de investigación distintas que extienden el modelo básico de las redes de Petri, haciendo posible la descripción de cambios dinámicos en sistemas concurrentes: *gramáticas de grafos* [6,17,4] y *redes automodificantes de Valk* [18,19]. Las reglas de reescritura de los sistemas de reescritura de redes controladas por marcado son muy similares a las producciones de las gramáticas de grafos (tienen parte izquierda, parte derecha e

interfaz), y la aplicación de una regla de reescritura es como una derivación directa en las gramáticas de grafos (bajo ciertas condiciones, cuando una ocurrencia (un *match*) de la parte izquierda se detecta en un grafo, puede reemplazarse por la parte derecha). Como en las redes automodificantes, modelizamos un sistema como una colección de redes de Petri, llamadas *configuraciones*, y un mecanismo que permite al sistema evolucionar de una configuración a otra bajo ciertas circunstancias.

Además de las redes automodificantes de Valk, en la literatura existen otros modelos que permiten la descripción de sistemas concurrentes dinámicos complejos. Las *redes móviles* de Asperti y Busi [1] originadas a partir de la fusión de redes de Petri con técnicas de manejo de nombres típicas del π -cálculo [14]; las *redes dinámicas* de Buscemi y Sassone [5] inspiradas en el join cálculo [9]. Ambas permiten la creación dinámica de componentes, como en nuestra propuesta. Otros modelos son las *redes Δ* de Gradiat y Vernadat [10], un formalismo de reescritura que integra las ventajas de las redes de Petri y las gramáticas de grafos, respectivamente, para la especificación del comportamiento y las transformaciones topológicas de un workflow, y el *formalismo POP* introducido por Engelfriet, Leih y Rozenberg [8], y el modelo relacionado de los *autómatas cooperativos* de Badouel, Darondeau y Tokmakoff [2], en el que los tokens son elementos activos con comportamiento dinámico. Para todos ellos, como en nuestro modelo, la descripción de los cambios es interna e incremental y su tratamiento es local. Además, la idea de reescritura subyace en todas estas propuestas; la configuración del sistema se describe como una red de Petri y un cambio de configuración se describe como una regla de reescritura de grafos que reemplaza la parte del sistema que hace *matching* con la parte izquierda de la regla de reescritura por la correspondiente parte derecha. Con respecto al poder expresivo, todas son equivalentes a la máquina de Turing, como nuestro modelo. El modelo de los sistemas de reescritura de redes controladas por marcado es más cercano a las redes de Petri.

El modelo de las redes reconfigurables controladas por marcado se introduce aquí como un sistema de reescritura de redes controladas por un marcado específico en el que la relación de transferencia τ es una biyección. Este modelo (aunque formalmente es equivalente a las redes de Petri) nos permite expresar de una forma más precisa sistemas en los que pueden ocurrir cambios dinámicos estructurales. Sin embargo, la traducción automática a redes de Petri asegura que todas las propiedades fundamentales de las redes de Petri siguen siendo decidibles para las redes reconfigurables controladas por marcado. Para este modelo es posible encontrar herramientas de verificación automáticas. Sin embargo, la transformación en redes de Petri equivalentes puede incrementar significativamente el tamaño de la red. Por tanto, puede ser más eficiente implementar directamente los métodos de verificación de propiedades de las redes de Petri sobre el modelo original, lo cual nos hace suponer que podemos definir las nociones de grafos de cobertura, invariantes lineales, *siphons* y *traps* directamente para una red reconfigurable controlada por marcado. Estos son temas de nuestra investigación actual. Por el contrario, la clase total de los sistemas de reescritura

de redes controladas por marcado tiene el poder computacional de la máquina de Turing y, por ello, no es posible encontrar herramientas automáticas de verificación en este caso. Sin embargo, este modelo sigue siendo interesante como herramienta de modelización y de simulación. Para alguno de los modelos descritos anteriormente, se han desarrollado herramientas software para edición y simulación de sistemas con un interfaz gráfico de usuario. Estas herramientas proveen soporte software para el diseño de prototipos de sistemas concurrentes dinámicos. Actualmente, estamos implementando un simulador para los sistemas de reescritura de redes controladas por marcado.

Referencias

1. A. Asperti and N. Busi. Mobile Petri Nets. Technical report UBLCS-96-10, University of Bologna, Italy, 1996.
2. E. Badouel, Ph. Darondeau, and A. Tokmakoff. Modelling Dynamic Agents Systems with Cooperating Automata. In *Proc. Int. Conf. on Parallel and Distributed Processing Techniques and Applications* (PDPTA'99), pp. 11–17, USA, 1999.
3. E. Badouel, M. Llorens, and J. Oliver. Modelling Concurrent Systems: Reconfigurable Nets. In *Proc. Int. Conf. on Parallel and Distributed Processing Techniques and Applications* (PDPTA'03), vol. IV, pp. 1568–1574, USA, 2003.
4. P. Baldan. Modelling Concurrent Computations: From Contextual Petri Nets to Graph Grammars. PhD Thesis, University of Pisa TD-1/00, 2000.
5. M. Buscemi and V. Sassone. High-level Petri nets as type theories in the Join calculus. In *Proc. 4th I.C. on Foundations of Software Science and Computation Structures* (FoSSaCS'01), Springer, LNCS, vol. 2030, pp. 104–120, Italy, 2001.
6. A. Corradini. Concurrent Computing: From Petri Nets to Graph Grammars. Invited talk at the *Joint COMPUGRAPH/SEMAGRAPH Workshop on Graph Rewriting and Computation*, Elsevier, ENTCS, vol. 2, 1995. <http://www.elsevier.nl/locate/entcs/volume2.html>.
7. C. Dufourd, A. Finkel, and Ph. Schnoebelen. Reset nets between decidability and undecidability. In *Proc. Int. Coll. on Automata, Languages and Programming* (ICALP'98), Springer-Verlag, LNCS, vol. 1443, pp. 103–115, Denmark, 1998.
8. J. Engelfriet, G. Leih, and G. Rozenberg. Net Based Description of Parallel Object-based Systems, or POTs and POPs. *Workshop on Foundations of Object-Oriented Languages* (FOOL'90), Springer-Verlag, LNCS, vol. 489, pp. 229–273, Noordwijkerhout, Netherlands, 1991.
9. C. Fournet, G. Gonthier, J. Lévy, L. Maranget, and D. Rémy. A Calculus of Mobile Agents. In *Proc. 7th Int. Conf. on Concurrency Theory* (CONCUR'96), Springer-Verlag, LNCS, vol. 1119, pp. 406–421, Pisa, Italy, 1996.
10. P. Gradiat, F. Vernadat, and P. Azéma. Layered Δ -Net Specification of a Workshop. In *Proc. Int. Conf. on Parallel and Distributed Processing Techniques and Applications* (PDPTA'99), vol. VI, pp. 2808–2814, USA, 1999.
11. M. Llorens, and J. Oliver. Modelización de Sistemas Concurrentes mediante Redes Reconfigurables. In *IX Jornadas de Concurrencia*, pp. 213–224, Sitges, Barcelona (Spain), 2001.
12. M. Llorens, and J. Oliver. Sistemas de Reescritura de Redes. In *XI Jornadas de Concurrencia*, pp. 237–250, Benicassim, Castellón (Spain), 2003.
13. M. Llorens. Redes Reconfigurables. Modelización y Verificación. Phd thesis, 2003.

14. R. Milner, J. Parrow, and D. Walker. A Calculus of Mobile Processes. In *Journal of Information and Computation*, Academic Press, vol. 100, no.1, pp. 1–77, 1992.
15. T. Murata. Petri Nets: Properties, Analysis and Applications. In *Proc. of the IEEE*, vol. 77, no.4, pp. 541–580, 1989.
16. J.L. Peterson. *Petri Net Theory and the Modeling of Systems*. Englewood Cliffs, N.J. Prentice-Hall, 1981.
17. H. Schneider. Graph Grammars as a Tool to Define the Behavior of Processes Systems: From Petri Nets to Linda. In *Proc. 4th Int. Conf. on Graph Grammars*, Williamsburg, USA, 1993.
18. R. Valk. Self-modifying Nets, a Natural Extension of Petri Nets. In *Proc. Int. Coll. on Automata, Languages and Programming (ICALP'78)*, Springer-Verlag, LNCS, vol. 62, pp. 464–476, Udine, Italy, 1978.
19. R. Valk. Generalizations of Petri Nets. In *Proc. 10th Symp. on Mathematical Foundations of Computer Science (MFCS'81)*, Springer-Verlag, LNCS, vol. 118, pp. 140–155, Strbske Pleso, Czechoslovakia, 1981.