MODELADO DE SISTEMAS CONCURRENTES CON REDES DE PETRI

Valentín Valero Ruiz   Antonio Bueno Aroca *

Escuela Politécnica Superior de Albacete
Departamento de Informática
Universidad de Castilla-La Mancha
valentin@info-ab.uclm.es   abueno@info-ab.uclm.es

Resumen Las redes de Petri permiten modelar y analizar sistemas concurrentes, siendo capaces de capturar las principales características de este tipo de sistemas: secuencialidad, concurrencia real, no-determinismo, sincronizaciones, etc. Las ventajas principales que tienen las redes de Petri frente a otros modelos son básicamente dos: por un lado pueden ser representadas gráficamente de una forma relativamente simple, lo que facilita notablemente la legibilidad del modelo, y a la vez, es un modelo con un buen soporte formal, lo que permite disponer de un conjunto de técnicas y herramientas para realizar de forma sistemática el análisis de propiedades sobre el modelo.

1 Introducción

En muchos campos de la Ciencia, el estudio de los fenómenos de interés no se realiza de forma directa, sino mediante *modelos* simplificados. Un *modelo* es una representación del sistema en la cual los aspectos irrelevantes han sido abstraidos (ver figura 1). Esta representación emplea por regla general algún tipo de formalismo matemático, que permite capturar de forma precisa las características o propiedades del sistema que pretender ser estudiadas.

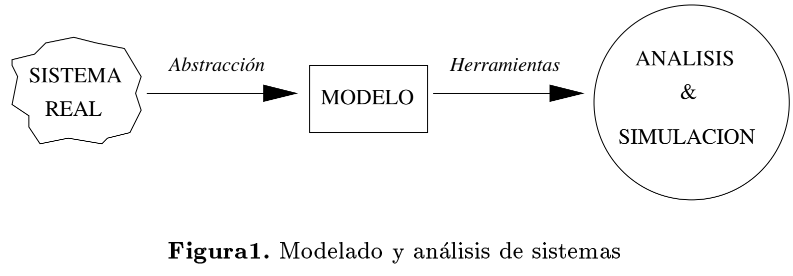

Figura1. Modelado y análisis de sistemas

* Este trabajo ha sido parcialmente financiado con el Subproyecto CICYT Evaluación de rendimientos de Sistemas Distribuidos, TIC2000-0701-C02-02.

El modelado de los sistemas informáticos no es una tárea simple, dada la complejidad y magnitud de este tipo de sistemas. Estos sistemas pueden constar de un cierto número de componentes interactuando. Surge así la primera idea básica del modelado, la identificación de una *estructura basada en componentes*, así como sus interacciones. Cada uno de estos componentes puede considerarse en sí mismo como un sistema, describiendo su conducta de forma separada del resto de componentes del sistema, excepto en lo que se refiere a sus relaciones con ellos. Por otra parte, cada componente en un momento dado se encontrará en un cierto *estado*, ésta es la segunda idea clave del modelado, en el estado de un componente capturamos la abstracción de la información relevante que nos permite describir su situación actual, y sus posibles acciones futuras.

Vemos que a lo largo del tiempo el estado de un componente va evolucionando, y esta transición entre estados se modeliza mediante la ejecución de eventos o acciones. Así disponemos de un *modelo funcional* del sistema, que nos permite describir los posibles estados de un sistema, así como las transiciones posibles entre estados.

Los sistemas que evolucionan de estado a estado por la ejecución de acciones simples se llaman *secuenciales*, mientras que aquellos sistemas que constan de varios subsistemas que evolucionan simultáneamente se denominan *concurrentes*. En muchas ocasiones, los sistemas concurrentes se modelan utilizando modelos secuenciales, como los diagramas de transición de estados, pues aunque se pierde cierta información, puede resultar más sencillo el análisis de ciertas propiedades del sistema.

Las redes de Petri no son un modelo reciente, fueron definidas por Carl Adam Petri en 1962 [5]. Es un modelo de naturaleza gráfica, aunque con un soporte formal preciso, que hace de las mismas un modelo muy útil en el diseño y análisis de sistemas concurrentes. Es precisamente esta primera característica, su simple representación gráfica, la que permite disponer de modelos de sistemas concurrentes fácilmente legibles, pues la concurrencia real de acciones queda identificada fácilmente, así como la sincronización de acciones. Por otra parte, la descripción formal de las mismas permite analizar ciertas propiedades del sistema de forma sistemática. Otra de las ventajas de este modelo es la disponibilidad de herramientas que dan soporte al diseño y análisis de sistemas con redes de Petri.

Asimismo, como muchos modelos empleados en el proceso de diseño de sistemas informáticos, las redes de Petri permiten el uso de técnicas de refinamientos progresivos, para ir obteniendo modelos cada vez más detallados del sistema o de sus componentes. Sin embargo, en muchos casos no resulta fácil crear un modelo de un sistema mediante redes de Petri, y es preciso acudir a otro tipo de técnicas, como las algebraicas, para posteriormente, transformar estas especificaciones en redes de Petri de forma automatizada.

Se han definido numerosas extensiones del modelo básico, con el fin de capturar ciertos aspectos adicionales en el modelado de sistemas, como por ejemplo restricciones temporales.

2 Definiciones básicas

Hemos identificado dos tipos de elementos en el proceso de modelado de sistemas, los elementos *pasivos* (condiciones o estados), y *activos* (acciones o eventos). Esta dualidad queda reflejada en la definición formal de las redes de Petri, en lo que se conoce como el *Principio de Dualidad*: los elementos pasivos se denominan *P-elementos* (lugares), mientras que los elementos activos se denominan *T-elementos* (transiciones).

Los P-elementos o lugares representan entidades del mundo real que se interpretan como elementos pasivos (pueden ser condiciones, recursos o instancias de recursos, variables, colas de espera, etc.). En cambio, las entidades que se interpretan como elementos activos se modelan por medio de los T-elementos, y pueden ser eventos, acciones, sentencias, transmisión de mensajes, etc.

Otra observación importante se refiere a la evolución de los sistemas por medio de los elementos activos. La posible ejecución de una acción en un sistema está condicionada a un entorno limitado, es decir, sólo un conjunto de elementos pasivos son relevantes a la hora de decidir si una acción puede ser o no realizada. A esto se le conoce como el *Principio de Localidad*, que viene a decir que la conducta de una transición depende únicamente de su entorno local.

Asimismo, y como consecuencia de este principio de localidad, puede ocurrir que dos transiciones puedan ser realizadas (*disparadas*) simultáneamente, pues sus entornos locales son disjuntos. A esto se le conoce como el *Principio de Concurrencia*. Es conveniente precisar que la posible simultaneidad de acciones no impone en absoluto la necesidad de hacerlas coincidir en el tiempo, este principio únicamente indica que varias transiciones pueden ejecutarse o dispararse a la vez, pues no hay una relación de causalidad entre ellas que lo impida, pero no se impone su ejecución solapada.

El comportamiento dinámico del sistema queda pues modelado por medio de la ejecución o *disparo* de las transiciones, según la llamada *Regla de Disparo*. Para conocer la situación o estado de la red se anotan los lugares con una colección de *tokens*, cuya interpretación puede ser diversa, según el sistema que se está modelando (un token en un lugar puede indicar que se cumple la condición asociada a dicho lugar, o un número n de tokens sobre un lugar puede representar por ejemplo el número de procesos que están sobre una cola de espera, etc.).

Gráficamente, los lugares se representan con círculos, mientras que las transiciones se representan por rectángulos o líneas. Los tokens sobre los lugares se representan mediante puntos o bien indicando el número de ellos sobre el lugar (ver figura 2).

Definición 1. (Redes de Petri Ordinarias)
Una terna $N = (P, T, F)$, formada por dos conjuntos P y T, y una relación F definida sobre $P \cup T$, se dice que es una *Red de Petri Ordinaria* si satisface:
1. $P \cap T = \emptyset$
2. $F \subseteq (P \times T) \cup (T \times P)$
3. $dom(F) \cup cod(F) = P \cup T$

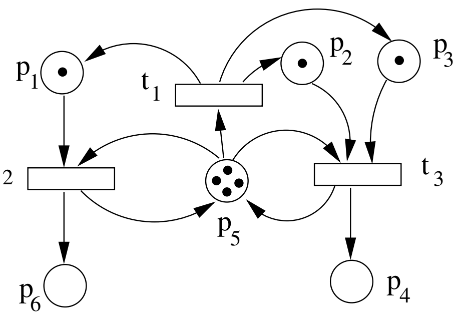

Figura2. Ejemplo de red de Petri marcada

Al conjunto P se le llama *conjunto de lugares*, al conjunto T *conjunto de transiciones* y a F *relación de flujo*. F relaciona lugares con transiciones en forma de arcos entre lugares y transiciones o entre transiciones y lugares. Aunque en principio P y T podrían ser infinitos, por regla general se trabaja únicamente con redes finitas.

Sea el conjunto $X = P \cup T$. Entonces, para todo $x \in X$ se definen los conjuntos siguientes: ${}^\bullet x = \{y \in X \mid (y,x) \in F\}$ (preconjunto de x), $x^\bullet = \{y \in X \mid (x,y) \in F\}$ (postconjunto de x).

Se dice que N es T-restringida si y sólo si ${}^\bullet t \neq \emptyset \neq t^\bullet \ \forall t \in T$. □

Definición 2. (Marcados de Redes de Petri Ordinarias)
Sea $N = (P,T,F)$ una Red de Petri Ordinaria. Una función $M : P \longrightarrow \mathbb{N}$ se llama un *marcado de N*. Entonces, (P,T,F,M) se llama una *Red de Petri Ordinaria Marcada.* □

Según hemos indicado anteriormente, los marcados de las redes de Petri se representan gráficamente incluyendo en cada lugar tantos puntos como tokens le corresponden, o bien anotando cada lugar con el número de tokens asociado al mismo. Dado que los marcados son funciones en $\mathbb{N}$, desde un punto de vista algebraico se representan como vectores en $\mathbb{N}^n$, siendo $n = |P|$.

La semántica de una Red de Petri está definida mediante la siguiente regla de disparo, que define el marcado alcanzado tras el disparo de una transición.

Definición 3. (Regla de Disparo)
Sea $N = (P,T,F,M)$ una Red de Petri Ordinaria Marcada. Una transición $t \in T$ se dice que *está permitida bajo el marcado M*, lo cual se denota por $M[t\rangle$, si para todo lugar $p \in P$ tal que $(p,t) \in F$ se verifica $M(p) > 0$.

El disparo de una transición permitida t bajo el marcado M resulta en un nuevo marcado de la red, M', definido de la forma siguiente:

$$M'(p) = M(p) - W_f(p,t) + W_f(t,p) \quad \forall p \in P$$

siendo $W_f(x) = 1$ si $x \in F$ y $W_f(x) = 0$, para $x \notin F$, para todo $x \in (T \times P) \cup (P \times T)$. Ello se denota de la forma $M[t\rangle M'$. □

Definición 4. (Activación Concurrente de Transiciones)
Sea $N = (P, T, F, M)$ una Red de Petri Ordinaria Marcada. Sea $R \in \mathcal{B}(T)$ un multiconjunto de transiciones de la red. Se dice que R está permitido bajo M, lo que se denota por $M[R\rangle$ si y sólo si $M(p) \geq \sum_{t \in T} W_f(p,t) \cdot R(t)$, $\forall p \in P$.

El disparo de un multiconjunto de transiciones R bajo el marcado M conduce la red a un nuevo marcado M', dado por:

$$M'(p) = M(p) - \sum_{t \in T}(W_f(p,t) - W_f(t,p)) \cdot R(t)$$

Esta evolución de la red en un solo paso la denotaremos por $M[R\rangle M'$. □

Definición 5. (Redes de Petri Generalizadas)
Una Red de Petri Generalizada es una tupla $N = (P, T, F, K, W)$, donde:

1. (P, T, F) es una Red de Petri Ordinaria.
2. $K : P \longrightarrow \mathbb{N} \cup \{\infty\}$ es una función que indica el número máximo de tokens que puede haber en cada lugar (*función de capacidad*).
3. $W : F \longrightarrow \mathbb{N}^+$ es una función que indica la multiplicidad de cada arco (*pesos de los arcos*).

Se suele omitir la función de capacidad cuando ésta es infinita para todos los lugares. Además, habitualmente extenderemos la definición de W a todo el universo de posibles arcos, haciéndola nula para pares (p,t) o (t,p) que no estén en F. □

Definición 6. (Regla de Disparo para Redes de Petri Generalizadas)
Sea $N = (P, T, F, K, W)$ una Red de Petri Generalizada.

1. Una función $M : P \longrightarrow \mathbb{N}$ se dice que es un *marcado* de N si y sólo si $M(p) \leq K(p)$, para todo $p \in P$.
2. Un multiconjunto de transiciones R está permitido bajo un marcado M, lo que se denota por $M[R\rangle$, si y sólo si se cumplen las dos condiciones siguientes para todo $p \in P$:
   (a) $M(p) \geq \sum_{t \in T} W(p,t) \cdot R(t)$
   (b) $M(p) - \sum_{t \in T}(W(p,t) - W(t,p)) \cdot R(t) \leq K(p)$
   El disparo de R resulta en un nuevo marcado M', dado por:
   $$M'(p) = M(p) - \sum_{t \in T}(W(p,t) - W(t,p)) \cdot R(t), \ \forall p \in P$$
   Lo que se denota por $M[R\rangle M'$.

□

Definición 7. Sea $N = (P, T, F, K, W, M_0)$ una Red de Petri Marcada.

1. Se dice que $\sigma = M_0 t_1 M_1 \ldots t_n M_n$ es una secuencia de ocurrencia finita de N si y sólo si $\forall i \in \{1, \ldots, n\}$, $M_{i-1}[t_i\rangle M_i$. Observemos que dado el marcado inicial M_0, y dada la secuencia de transiciones $t_1 \ldots t_n$, es posible reconstruir la secuencia de ocurrencia completa. Por ello, en ocasiones éstas se denotan simplemente de la forma $M_0[\sigma\rangle M_n$, siendo $\sigma = t_1 \ldots t_n$. Asimismo, se denota por $[M_0\rangle$ al conjunto de marcados que pueden ser alcanzados a partir de M_0.

2. Se dice que $\sigma = M_0 R_1 M_1 \ldots R_n M_n$ es una secuencia de pasos finita de N si y sólo si $\forall i \in \{1, \ldots, n\}$, $M_{i-1}[R_i\rangle M_i$. De nuevo se extiende la notación habitual a secuencias de pasos: $M_0[\sigma\rangle M_n$. □

Ejemplo 1. Para la red mostrada en la figura 2 tenemos que para el marcado inicial $M_0 = (1, 1, 1, 0, 4, 0)$ el multiconjunto de transiciones $R = \{2.t_1, 1.t_2, 1.t_3\}$ está permitido, y su disparo genera el marcado $M' = (2, 2, 2, 1, 2, 1)$. □

3 Modelos y ejemplos

En esta sección vamos a mostrar la capacidad de modelado de las redes de Petri. Primero veremos cómo podemos modelar algunas de las situaciones que se nos plantean en general en los sistemas concurrentes, y despues veremos algunos de los clásicos problemas de concurrencia.

En la figura 3 se resumen las situaciones principales que podemos encontrarnos en sistemas concurrentes, y la forma de modelarlas con redes de Petri.

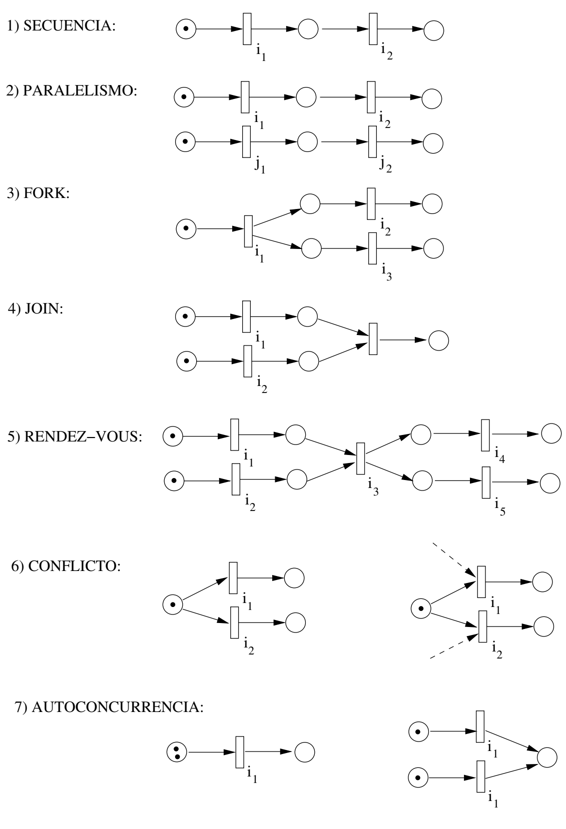

Figura3. Modelos simples

3.1 Exclusión mutua

En la figura 4.a podemos ver la forma de modelar la exclusión mutua sobre un recurso. Podemos observar que el recurso puede modelarse mediante un lugar con un token, que representa la única instancia disponible del mismo. Cuando uno de los procesos ha obtenido el recurso, dicho lugar queda sin marcas, de modo que el otro no puede obtenerlo.

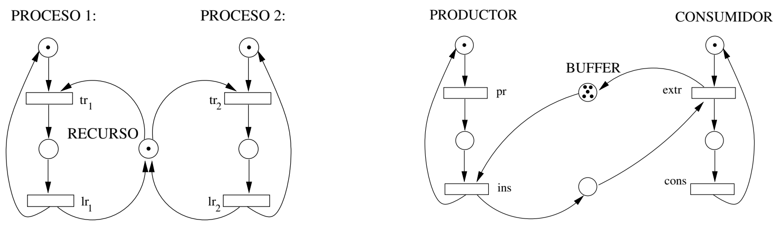

Figura4. (a) Exclusión mutua (b) Productor/Consumidor

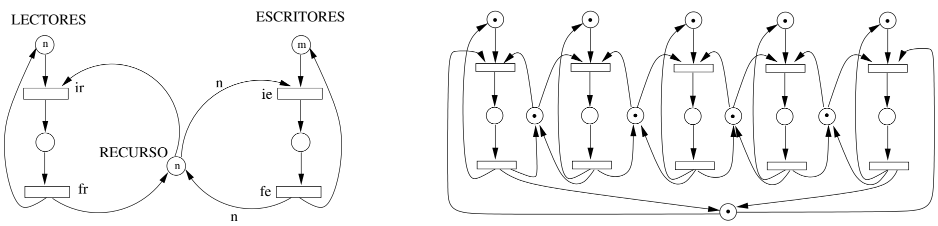

Figura5. (a) Lectores/Escritores (b) Problema de los 5 Filósofos

3.2 Productor/Consumidor

El clásico problema del productor/consumidor puede ser modelado con redes de Petri de la forma mostrada en la figura 4.b. Vemos que para modelar las sincronizaciones condicionales con el buffer hemos empleado dos lugares, uno para representar el número de celdas disponibles, y otro para modelar el número de celdas ocupadas. Observemos que si ponemos varios tokens sobre los lugares iniciales del productor y del consumidor obtenemos el modelo general para varios productores y varios consumidores.

3.3 Lectores/Escritores

En la figura 5.a podemos ver la red de Petri generalizada que modela el problema de los lectores/escritores para n procesos lectores y m procesos escritores. En este caso el acceso al recurso se controla con un lugar marcado inicialmente con n tokens, y los arcos que conectan este lugar con la subred que modela a los procesos escritores están etiquetados con peso n, de modo que un escritor sólo puede comenzar su operación de escritura sobre el recurso si los n tokens están disponibles. En cambio, cada lector sólo necesita un token del lugar asociado al recurso, de modo que varios de ellos pueden leer simultáneamente. Cuando un escritor finaliza su operación de escritura devuelve los n tokens a dicho lugar.

3.4 Problema de los filósofos

También podemos ver en la figura 5.b el problema de los 5 filósofos modelado con redes de Petri, en su versión de adquirir ambos tenedores simultáneamente para empezar a comer. Cada tenedor está representado por medio de un lugar marcado inicialmente con un token, lo que significa que está disponible, para ser utilizado por los filósofos que ocupan la parte izquierda y la parte derecha del mismo. Con esta solución no existe el problema del bloqueo mutuo, dado que un filósofo debe tener disponibles ambos tenedores para poder comer, y en ese caso, los cogería ambos de una sola vez.

4 Análisis de propiedades

Hemos visto que el proceso de diseño mediante redes de Petri no es una tarea simple, y requiere cierta experiencia, así como la utilización de ciertas técnicas que den un cierto soporte a esta tarea. Una consecuencia de ello es que el modelo resultante puede ser muy sofisticado, es decir, podemos generar una red de Petri con cientos o incluso miles de lugares y transiciones, de modo que no resulta fácil demostrar que dicha red cumple con las especificaciones del sistema.

La verificación de propiedades sobre redes de Petri tiene pues un doble objetivo, por un lado el poder demostrar que el diseño realizado es correcto, es decir, que se cumplen las especificaciones realizadas (por regla general utilizando algún tipo de lógica), y por otro, poder extraer propiedades sobre el comportamiento del sistema.

Las propiedades objeto de interés se relacionan con la conducta del sistema modelado, aunque básicamente pueden englobarse dentro de las siguientes:

1. *Alcanzabilidad*: Un marcado M es alcanzable si existe una secuencia de ocurrencia σ tal que $M_0[\sigma\rangle M$.
   Con esta propiedad pretendemos conocer si un cierto estado del sistema es alcanzable o no. En ocasiones este problema interesa plantearlo sólo parcialmente, pues no interesa fijarse en el marcado de todos los lugares, sino sólo de algunos de ellos, que son los relevantes para el tipo de problema planteado. En este caso se plantea la verificación de una propiedad de alcance de *submarcados*.
2. *Seguridad*: Una red de Petri N es n-segura si $\forall M \in [M_0\rangle$, se cumple que $M(p) \leq n$, $\forall p \in P$. Un caso especial son las redes 1-seguras, para las cuales

los lugares sólo pueden tener a lo sumo un token. Este tipo de redes permiten abordar con éxito la verificación de muchas de las propiedades sobre redes de Petri, y de hecho, son muy utilizadas en el modelado de sistemas, aunque por esta simplicidad del modelo se paga el coste de generar redes muy voluminosas, incluso para sistemas pequeños.

3. *Acotación*: La acotación o también denominada *limitación* de una red se plantea en términos locales y globales. Así, se dice que un lugar p es acotado si existe un $n_p \in \mathbb{N}$ tal que $M(p) \leq n_p$, $\forall M \in [M_0\rangle$. Cuando todos los lugares de una red son acotados se dice que ésta es acotada.
   La acotación es una propiedad que permite verificar propiedades del tipo de desbordamientos de un buffer, o para comprobar que el número de procesos en una cola de espera no supera nunca una cierta cantidad, etc.
4. *Vivacidad*: Se dice que una transición t es viva si para todo marcado alcanzable $M \in [M_0\rangle$ existe otro marcado $M' \in [M\rangle$ tal que t está permitida bajo M: $M[t\rangle$. Si todas las transiciones de N son vivas, se dice que N es viva.
   La vivacidad de una transición garantiza que con independencia del estado actual que el sistema haya alcanzado, éste siempre puede evolucionar hacia un estado que permite disparar la transición. Por ejemplo, mediante esta propiedad podríamos determinar si un proceso servidor podría estar eventualmente en condiciones de poder atender las peticiones de servicio que le lleguen de un cierto tipo, con independencia del estado actual en que se encuentre el sistema.
5. *Ausencia de bloqueos*: Diremos que una red de Petri es libre de bloqueos si ningún marcado alcanzable es un marcado *muerto*. Un marcado M se dice que es *muerto* si no permite disparar ninguna transición.
   Esta propiedad nos permitirá comprobar que el sistema nunca llega a una situación de bloqueo (*deadlock*), en la que éste no puede evolucionar de ninguna forma, ni siquiera de forma parcial por parte de alguno de sus componentes.
6. *Reversibilidad*: Una red de Petri se dice que es reversible si para cualquier marcado alcanzable $M \in [M_0\rangle$ existe una secuencia de ocurrencia σ tal que $M[\sigma\rangle M_0$.
   Esta propiedad nos permite comprobar que con independencia del estado actual alcanzado por el sistema, siempre podemos volver a la situación inicial.

En todos los casos anteriores, estas propiedades han sido planteadas partiendo de un marcado inicial concreto, y dependiendo del mismo se trata de verificar si la correspondiente propiedad se cumple o no. En algunos casos, el que una propiedad resulte cierta o no realmente no depende del marcado inicial de la red, sino de su estructura (*propiedades estructurales*). Por ejemplo, la red de la figura 6 es acotada estructuralmente, pues para cualquier marcado inicial dicha red será acotada.

5 Clases particulares de redes de Petri

En la sección anterior hemos mostrado las propiedades más relevantes dentro del estudio de la conducta de los sistemas modelados con redes de Petri. Con ellas

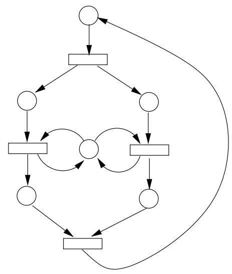

Figura6. Red acotada estructuralmente

se capturan muchos de los problemas prácticos que surgen al modelar sistemas concurrentes. Sin embargo, el estudio de estas propiedades no es una tarea simple, pues incluso aunque en muchos casos estas propiedades son decidibles, su análisis en el caso general es prohibitivo o imposible de realizar en la práctica. Tal es el caso de la propiedad de alcance, cuya decidibilidad fue resuelta en 1981 [2], si bien en la práctica no existe un algoritmo general que nos permita resolver si un marcado es o no alcanzable, dada una red de Petri cualquiera.

Sin embargo, si restringimos el campo de estudio a una clase de redes particulares, podemos obtener resultados concretos sobre la forma de dictaminar si las propiedades anteriores se cumplen o no. Así, si consideramos únicamente redes acotadas, tendremos que su conjunto de marcados alcanzables será finito, y con ello sería posible verificar todas las propiedades anteriores. El inconveniente de esta técnica es evidente, el conjunto de marcados alcanzables puede ser enorme, lo que se conoce como el *Problema de la Explosión de Estados*, y ello puede hacer intratable la propiedad a verificar. Para abordar este problema se han propuesto diversos algoritmos de reducción o simplificación, tanto de la estructura de la red como del conjunto de marcados alcanzables. Con ello se pretende obtener un modelo simplificado, a costa de perder cierta información, pero sobre el cual se preserva la propiedad objeto de interés.

En otros casos, se imponen restricciones sobre la estructura de la red, obteniendo unas clases de redes de Petri aptas para el modelado de determinado tipo de sistemas, y sobre las cuales resulta sencilla la verificación de algunas de las propiedades enunciadas. Seguidamente mostramos algunas clases de redes de Petri muy utilizadas en la práctica:

1. *Grafo de estados*: De cada transición entra y sale exactamente un arco:

   $$\forall t \in T \;:\; |{}^{\bullet}t| = |t^{\bullet}| = 1$$

   Esta clase de redes permite modelar conflictos, pero no permite modelar un *fork*, un *join*, o una *sincronización*. Por ello tienen esencialmente una capacidad descriptiva similar a la de los autómatas de estados finitos, si bien permiten modelar también la autoconcurrencia, pues es posible colocar varias marcas sobre los lugares.

2. *Grafo marcado*: En este caso para cada lugar de la red tenemos un arco de entrada y un arco de salida: $\forall p \in P : |^\bullet p| = |p^\bullet| = 1$.
   Esta clase de redes permiten modelar el *fork*, el *join*, la sincronización de actividades, pero no permiten modelar conflictos ni *sumideros* (lugares que reciben tokens de varias transiciones). Se emplean para modelar sistemas de ordenación de actividades, de forma similar a los grafos PERT.
3. *Red de libre elección*: Una red de Petri es de libre elección si
   $$\forall p \in P, \text{ si } |p^\bullet| > 1 \Rightarrow \forall t \in p^\bullet \ : \ |^\bullet t| = 1$$
   Ello quiere decir que los conflictos se permiten únicamente sin dependencias externas.
4. *Red simple*: Aunque su definición es un tanto aparatosa, se trata de una clase de redes que se puede encontrar frecuentemente al modelizar sistemas concurrentes. En ellas se impone la restricción de que para cada transición, a lo sumo uno de sus lugares de entrada puede ser compartido con otras transiciones:
   $$\forall t \in T, \ p \in {}^\bullet t \Rightarrow (p \in {}^\bullet t' \Rightarrow \nexists p' \neq p, \ p' \in {}^\bullet t', \, p' \in {}^\bullet t'')$$
5. *Red pura*: En ellas se impone la restricción de que no hay lugares que son simultáneamente precondición y postcondición de alguna transición:
   $$\forall t \in T, : \forall p \in P \ : \ W(t,p) \cdot W(p,t) = 0$$
   Hemos de observar que estos lugares aparentemente inócuos, en realidad afectan significativamente al comportamiento de la red, y por tanto, no pueden eliminarse alegremente. Por un lado, si están desmarcados pueden impedir el disparo de sus transiciones postcondición, y por otra, pueden serializar o limitar la autoconcurrencia de dichas transiciones. Ello no obstante, esta clase de redes es especialmente importante cuando se aplican técnicas algebraicas para analizar las propiedades de las redes.
6. *Redes T-restrictivas*: Son aquellas para las cuales todas las transiciones deben tener al menos un lugar precondición y un lugar postcondición:
   $$\forall t \in T \ : \ {}^\bullet t \neq \emptyset, \ \ t^\bullet \neq \emptyset$$
   La presencia de transiciones sin precondiciones suele ir asociada – en términos de modelado – a la ocurrencia de eventos externos asíncronos, de modo que estos pueden ocurrir en cualquier momento, como es el caso de las interrupciones. Sin embargo, esta forma de proceder en el modelado tiene un inconveniente desagradable, y es que la transición también puede ser disparada un número arbitrario de veces en autoconcurrencia, y si se considera una semántica de pasos ello conduce a ciertos problemas a la hora de definir y estudiar el comportamiento de la red.
   La restricción simétrica, que consiste en imponer que todas las transiciones tengan al menos una postcondición suele generar menos problemas, pero en algunos casos también resulta necesaria.

En la figura 7 mostramos gráficamente el tipo de situaciones que pueden ser modeladas con cada clase de redes, y el tipo de situaciones que no pueden ser modeladas.

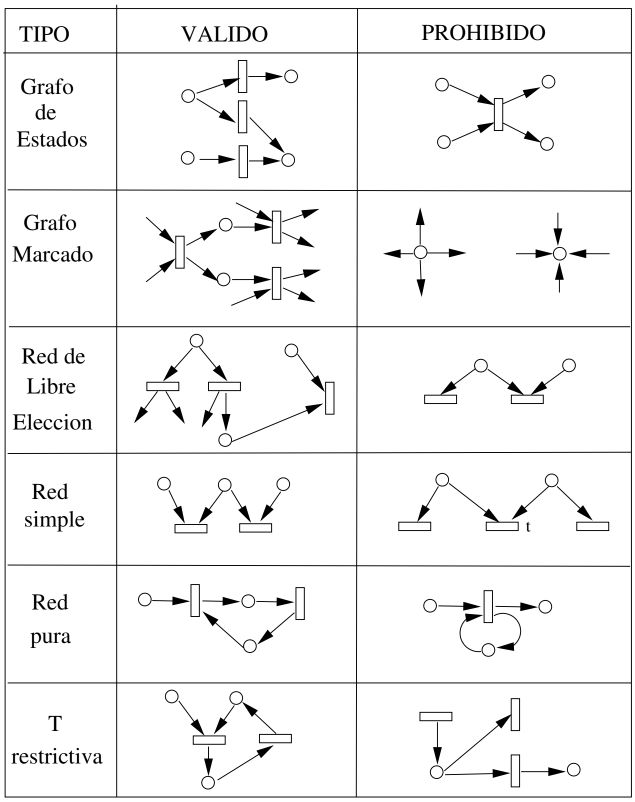

Figura7. Algunas clases de redes de Petri

6 Representación algebraica

Si restringimos nuestro campo de estudio al marco de las redes puras, entonces podemos capturar la estructura de una red de Petri mediante una *matriz de incidencia*, $C \in \mathcal{M}_{|P|\times|T|}$, definida de la forma siguiente:

$$C(i,j) = W(t_j,p_i) - W(p_i,t_j) \ , \quad i = \{1,\ldots,|P|\}, \, j \in \{1,\ldots,|T|\}$$

Por ejemplo, en la figura 8 mostramos una red de Petri pura, con su matriz de incidencia asociada.

Es evidente que esta definición puede ser aplicada también a redes no puras, pero en tal caso la representación matricial no identifica de forma unívoca a la

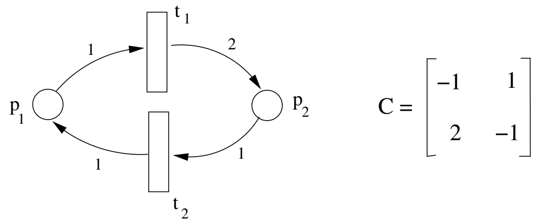

Figura8. Ejemplo de red de Petri y su matriz de incidencia

red. Por otra parte, si la red de Petri es ordinaria (el valor del peso sólo puede ser 1), entonces esta matriz únicamente constará de los valores $\{0, 1, -1\}$, y de hecho, para redes de cierta magnitud, esta matriz será por regla general dispersa, es decir, habrá un número de ceros considerable, dado que habitualmente los lugares y transiciones están conectados con un conjunto muy reducido de transiciones y lugares, respectivamente.

Proposición 1. (Ecuación de Estado)
Dada una red de Petri pura N con matriz de incidencia C, y dada una secuencia de ocurrencia $M_0[t_1 \dots t_n\rangle M$, entonces se tiene que $M = M_0 + C \cdot \sum_{i=1}^{n} U_i$, siendo $U_i \in \mathbb{N}^{|T|}$, definido como sigue:

$$\forall t \in T \ , \ \ U_i(t) = \begin{cases} 1 \text{ si } t_i = t \\ 0 \text{ en caso contrario} \end{cases}$$

Demostración. Basta aplicar un sencillo razonamiento por inducción sobre la longitud de la secuencia, obteniendo para el caso base según la regla de disparo que $M(p) = M_0(p) + W(t_1, p) - W(p, t_1)$, $\forall p \in P$, *lo cual coincide con el resultado de* $M_0 + C \cdot U_1$, *teniendo en cuenta la definición de* C.

Si suponemos ahora que para una secuencia de longitud $n-1$ *la propiedad es cierta, obtenemos que* $M_{n-1} = M_0 + C \cdot \sum_{i=1}^{n-1} U_i$, *y sabemos que* $M_{n-1}[t_n\rangle M$, *de modo que* $M = M_{n-1} + C \cdot U_n$. *Basta pues desplegar la ecuación para* $n-1$ *para obtener el resultado:*

$$M = M_{n-1} + C \cdot U_n = M_0 + (C \cdot \sum_{i=1}^{n-1} U_i) + C \cdot U_n = M_0 + C \cdot \sum_{i=1}^{n} U_i$$

□

Es importante precisar que la proposición anterior no puede ser aplicada en sentido inverso, pues de la existencia de un vector $U \in \mathbb{N}^{|T|}$ tal que $M = M_0 + C \cdot U$, no podemos concluir que exista una secuencia σ, tal que $M_0[\sigma\rangle M$. La justificación formal de este hecho se debe a que los marcados intermedios que van siendo obtenidos por la sucesiva ejecución de las transiciones no pueden tener componentes negativas.

No obstante, esta ecuación resulta especialmente útil a la hora de analizar propiedades sobre las redes de Petri, pues permite aplicar una colección de técnicas obtenidas a partir del álgebra lineal.

7 Extensiones del modelo

En esta sección mostramos algunas extensiones del modelo, que amplían su capacidad de modelado.

7.1 Redes con arcos inhibidores

Desde un punto de vista teórico, las redes de Petri no tienen la potencia descriptiva de una máquina de Turing, pues son incapaces de realizar un *test de cero* marcas (al disparar las transiciones sólo miramos que sus lugares precondición contengan un número de marcas mayor que cero). Es precisamente por esto que muchas de las propiedades que hemos mostrado son decidibles en este marco. Sin embargo, en ocasiones nos surge el problema de modelar algunos sistemas para los cuales esta capacidad de modelado resulta insuficiente. Tal es el caso de un recurso compartido por dos procesos, pero uno de ellos tiene prioridad sobre el otro, es decir, que si ambos procesos piden simultáneamente el recurso, el proceso prioritario debe ser aquél que obtenga el recurso.

Con la capacidad de modelado de una red de Petri lo único que podemos hacer es reflejar la existencia de un conflicto, pero no podríamos capturar el hecho de que uno de ellos tiene prioridad sobre el otro.

Podemos entonces extender el modelo añadiendo una nueva clase de arcos, los llamados *arcos inhibidores*, que únicamente conectan lugares con transiciones, y que como su nombre indica se emplean para *inhibir* el disparo de las transiciones sobre las que inciden, cuando el correspondiente lugar precondición se encuentre marcado. Se representan gráficamente mediante un arco cuya punta es un pequeño círculo, en lugar de la clásica punta de flecha. El ejemplo anterior puede ser modelado utilizando redes con arcos inhibidores (ver figura 9), donde podemos ver que el proceso $P1$ tiene prioridad sobre $P2$, pues si el lugar p está marcado, la transición t_2 no podrá ser ejecutada.

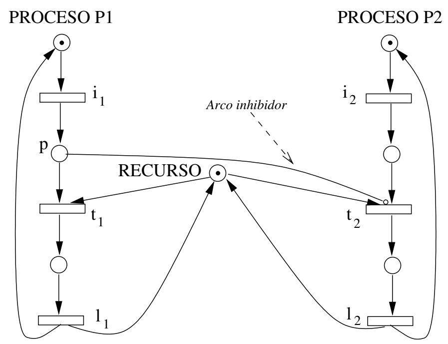

Figura9. Red de Petri con arcos inhibidores

En consecuencia, desde un punto de vista formal una red de Petri con arcos inhibidores es un par (N, A), siendo $N = (P, T, F, W)$ una red de Petri, y $A \subseteq P \times T$ un conjunto de arcos inhibidores (que puede ser vacío) tal que $A \cap F = \emptyset$. La condición de activación de las transiciones debe ser modificada, quedando ésta como sigue, $M[t\rangle$ si y sólo si:

$$\forall p \in P \,:\, ((p,t) \in F \;\Rightarrow\; M(p) \geq W(p,t)) \,\wedge\, ((p,t) \in A \;\Rightarrow\; M(p) = 0)$$

El marcado obtenido tras el disparo de t se define exactamente igual que para las redes sin arcos inhibidores: $M[t\rangle M'$.

7.2 Redes Coloreadas

En la descripción del problema de los lectores/escritores hemos visto que una misma subred puede ser utilizada para modelar la ejecución de varios procesos en paralelo. Sin embargo, en determinadas situaciones surge el problema de tener que diferenciar claramente estos procesos, o más bien, los tokens correspondientes a estos procesos, para evitar que algunas transiciones queden activadas con tokens provenientes de procesos distintos. Como ejemplo, supongamos que lanzamos dos procesos P_1, P_2, cuyo comportamiento es idéntico, ambos crean dos hilos, que realizan una primera acción, y luego sincronizan, para despues reiniciar su ejecución. La única forma de modelar correctamente este sistema con redes de Petri sería desplegando por separado ambos procesos, pues si intentamos capturar en una única subred la conducta de ambos procesos (figura 10.a) nos encontramos con que la transición c podría ser ejecutada con un token proveniente de P_1 y el otro de P_2, es decir, que estaríamos permitiendo la sincronización de hilos de diferentes procesos.

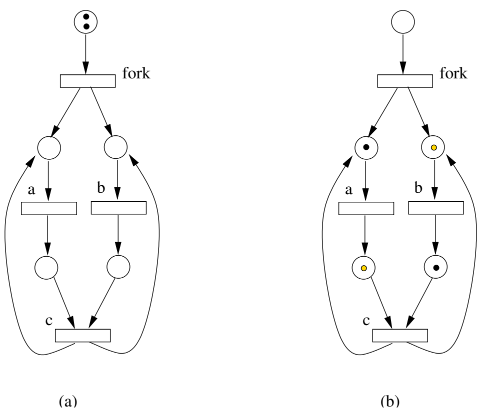

Figura10.

Ahora bien, si distinguimos los tokens que corresponden a cada proceso, dibujándolos por ejemplo con colores separados, e imponemos la restricción de que la transición c sólo puede ejecutarse si ambos tokens son del mismo color, entonces lograremos modelar correctamente este problema (figura 10.b).

La descripción anterior motiva perfectamente la generalización de las redes de Petri con la introducción de los colores. Con una visión algo más general que la mostrada en el ejemplo anterior podríamos definir una red coloreada como una red cuyos lugares y transiciones van anotados con un conjunto de colores. El conjunto de colores asociados a un lugar nos indica los colores posibles de los tokens sobre dicho lugar, mientras que el conjunto de colores de una transición nos indica las diferentes formas posibles de dispararla, cada una de las cuales requerirá un cierto número de tokens de cada color de cada lugar precondición, y generará a su vez un cierto número de tokens de cada color en cada lugar postcondición.

7.3 Redes Temporizadas

Hay numerosas variantes de modelos de redes de Petri que introducen el tiempo en la descripción de los sistemas. En unos casos se hace asociando duraciones a las transiciones, de modo que éstas *tardan* un cierto tiempo en completarse, en otros se asocia cierta información temporal a los lugares (representando un tiempo de *estancia* en ellos), o incluso a los mismos tokens, dotándolos así de una edad. Incluso hay modelos que asocian cierta información temporal a los arcos.

De todos estos modelos, uno de los más utilizados es el modelo propuesto por Merlin [3], que asocia a cada transición dos números enteros, que representan, respectivamente, el primer instante en el que la transición puede ser disparada, y el instante final en el que ésta *debe* ser disparada de hecho. Estos instantes son relativos al instante en el que la transición comenzó a estar permitida. El disparo de las transiciones en este modelo no toma tiempo, es decir, éstas son instantáneas, y en el mismo instante en que han sido disparadas producen los nuevos tokens sobre los lugares de salida.

Referencias

1. C. Girault, R. Valle. *Systems Engineering: A Petri Net Based Approach to Modelling, Verification, and Implementation.* MATCH, Human Capital and Mobility. Summer School, Jaca. 1998.
2. E.W. Mayr.*An Algorithm for the General Petri Reachability Problem.* Proc. 13th. ACM Symp. on Theory of Computing. 1981.
3. P. Merlin. *A Study of the Recoverability of Communication Protocols.* Thesis Doc. Computer Science Dep. Univ. California. 1974.
4. J.L. Peterson. *Petri Net Theory and The Modeling of Systems.* Prentice-Hall. 1981.
5. C. Petri. *Kommunikation mit Automaten.* Ph.D. dissertation, University of Bonn, Bonn, West Germany. 1962.
6. M. Silva. *Las Redes de Petri: en la Automática y la Informática.* Editorial AC. 1985.