





**CINVESTAV-IPN**

Biblioteca de Ingeniería Eléctrica



FB000009760

CENTRO DE INVESTIGACION Y DE  
ESTUDIOS AVANZADOS DEL  
I. P. N.  
BIBLIOTECA  
INGENIERIA ELECTRICA

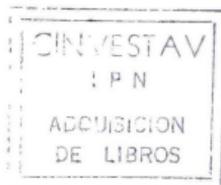
CENTRO DE INVESTIGACION Y DE ESTUDIOS  
AVANZADOS  
DEL  
INSTITUTO POLITECNICO NACIONAL

DEPARTAMENTO DE INGENIERIA ELECTRICA

SECCION COMPUTACION

CENTRO DE INVESTIGACION Y DE  
ESTUDIOS AVANZADOS DEL  
I. P. N.  
BIBLIOTECA  
INGENIERIA ELECTRICA

**SIMULACION CON REDES DE PETRI**



Tesis que presenta el Lic. **Valdemar González Avila** para obtener el grado de MAESTRO EN CIENCIAS en la especialidad de INGENIERIA ELECTRICA.

Trabajo dirigido por el M. en C. **José Oscar Olmedo Aguirre.**

CLASSIF.	93.11
ACQUIS.	AI-13814
PRCHA.	23/01/94
PROCED.	DDN
	\$

*Las ciencias aplicadas no existen,  
sólo las aplicaciones de la ciencia.*

*Louis Pasteur.*

CENTRO DE INVESTIGACION Y DE  
ESTUDIOS AVANZADOS DEL  
I. P. N.  
BIBLIOTECA  
INGENIERIA ELECTRICA

A mis padres

*Elda Elena y Timoteo*

por su cariño, comprensión y apoyo incondicional.

CENTRO DE INVESTIGACION Y DE  
ESTUDIOS FINANCIEROS DEL  
I. P. N.  
BIBLIOTECA  
INGENIERIA ELECTRICA

A mis hermanos

*Georgina, Ofelia,  
Raúl, Timoteo,  
Elena y Jesús*

con mucho cariño.

Al M. en C. *J. Oscar Olmedo Aguirre*  
por su acertada dirección de este trabajo.

A todos los maestros y amigos del CINVESTAV  
por su apoyo durante el programa de maestría.

CENTRO DE INVESTIGACION Y DE  
ESTUDIOS AVANZADOS DEL  
I. P. N.  
BIBLIOTECA  
INGENIERIA ELECTRICA

A *Naed*  
por los momentos compartidos.

## INDICE DE CONTENIDO

INTRODUCCION	1
CAPITULO 1. Teoría de las Redes de Petri	
1.1 Introducción	5
1.2 Estructura de Redes de Petri	5
1.3 Representación gráfica	6
1.4 Representación matricial	7
1.5 Reglas de ejecución de Redes de Petri	9
1.6 Espacio de estados de Redes de Petri	13
1.7 Modelado con Redes de Petri	16
1.7.1 Eventos y condiciones	16
1.8 Análisis de Redes de Petri	20
1.8.1 Propiedades de las Redes de Petri	20
1.8.2 Técnicas de análisis	23
1.8.2.1 El árbol de alcanzabilidad	23
1.8.2.2 Matriz de ecuaciones	24
CAPITULO 2. Análisis de la vivacidad de las Redes de Petri por resolución de un sistema lineal	
2.1 Introducción	27
2.2 Principio del método	27
CAPITULO 3. El sistema <i>SSRdP</i> para simular Redes de Petri	
3.1 Introducción	47
3.2 Ejecutando <i>SSRdP</i>	50
3.3 El menú principal	50

3.3.1 La opción Archivo	52
3.3.2 La opción Entrada	53
3.3.3 La opción Salida	57
3.3.4 La opción Pantalla	59
3.3.5 La opción Ejecutar	60
3.3.6 La opción Analizar	62

#### CAPITULO 4. Aplicaciones

4.1 Introducción	66
4.2 El problema del productor-consumidor	66
4.3 El problema de los filósofos comensales	69

CONCLUSIONES	75
--------------	----

REFERENCIAS	76
-------------	----

CENTRO DE INVESTIGACION Y DE  
ESTUDIOS AVANZADOS CITA  
I. P. N.  
BIBLIOTECA  
INGENIERIA ELECTRICA

## INDICE DE FIGURAS

<b>Fig.</b>	<b>Descripción</b>	<b>Capítulo</b>	<b>Pag.</b>
1.1	La gráfica de una RdP	1	7
1.2	Una gráfica de RdP marcada	1	9
1.3	Una RdP marcada para ilustrar las reglas de ejecución	1	11
1.4	El marcado que resulta al disparar la transición t3 en la fig. 1.3	1	12
1.5	El marcado que resulta al disparar la transición t2 en la fig. 1.4	1	12
1.6	Una celda de ensamble	1	18
1.7	Una RdP para modelar una celda de ensamble	1	19
1.8	Una RdP para ilustrar la reservación de recursos	1	22
1.9	Una RdP para ilustrar la alcanzabilidad con el enfoque matricial	1	25
2.1	Una RdP que no es viva	2	33
2.2	Una RdP que es viva	2	37
2.3	Una RdP viva	2	40
2.4	Una RdP con lugares que tienen multiplicidad de entrada mayor que 1	2	42
3.1	El menú principal de SSRdP	3	51
3.2	La ventana pull-down Archivo	3	52
3.3	La pantalla cuando se crea un lugar	3	53
3.4	La pantalla cuando SSRdP pide que seleccione un lugar	3	54
3.5	La pantalla cuando SSRdP pide el número de marcas	3	55
3.6	La pantalla después de que se marca un lugar	3	56
3.7	La pantalla después de crear una transición	3	57
3.8	La pantalla cuando SSRdP pide la multiplicidad	3	59

3.9	La pantalla después de establecer una relación de entrada	3	60
3.10	Una RdP para ilustrar la ejecución	3	61
3.11	La pantalla que muestra el menú con las transiciones habilitadas	3	62
3.12	La pantalla después de disparar la RdP de la fig. 3.10	3	64
3.13	La pantalla cuando SSRdP pregunta por un nuevo marcado	3	64
3.14	La pantalla cuando SSRdP proporciona el vector de disparo para alcanzar a $\mu'$	3	65
3.15	La pantalla cuando SSRdP proporciona el vector de disparo que bloquea a la red	3	65
4.1	El problema del productor-consumidor modelado con SSRdP	4	67
4.2	SSRdP después de ejecutarse en forma aleatoria	4	68
4.3	La alcanzabilidad en el problema del productor-consumidor	4	69
4.4	El análisis de la vivacidad del productor-consumidor	4	70
4.5	El problema de los filósofos comensales	4	71
4.6	La alcanzabilidad en el problema de los filósofos comensales	4	72
4.7	SSRdP cuando proporciona el vector de disparo en los filósofos comensales	4	73
4.8	El análisis de la vivacidad en los filósofos comensales	4	74

## INTRODUCCION

En diversas ramas de la ciencia y tecnología un fenómeno no se estudia directamente sino a través de un modelo. Un modelo es una representación, generalmente en términos matemáticos, de lo que se cree son las características más importantes del sistema bajo estudio. Al manipular dicha representación se espera obtener nuevos conocimientos acerca del fenómeno modelado, sin el peligro, costo, o inconveniencia de manejar el fenómeno real.

Las Redes de Petri (RdP) son una herramienta útil para la modelación y análisis de sistemas dinámicos con eventos discretos.

Carl A. Petri desarrolló la teoría de las RdP para modelar y analizar sistemas de comunicación. Su trabajo fue principalmente un desarrollo teórico de los conceptos básicos a partir de los cuales las RdP se han desarrollado [Petri62]. Más tarde las RdP mostraron su utilidad para la modelación, control y evaluación del comportamiento de sistemas. Específicamente, estas son útiles para modelar sistemas que involucren los siguientes conceptos [Desrochers90] :

- 1.- Concurrencia,
- 2.- Actividades asíncronas,
- 3.- Bloqueo (*Deadlock* ),
- 4.- Conflicto,
- 5.- Manejo de eventos,

como es el caso de los sistemas operativos, bases de datos, programas de computadoras, protocolos de comunicación, sistemas en tiempo real, etc.

Una RdP se compone de 4 partes : un conjunto de lugares  $P$ , un conjunto de transiciones  $T$ , una función de entrada  $I$  y una función de salida  $O$ . Las funciones de entrada y salida relacionan lugares y transiciones. La función de entrada es un mapeo de cada transición  $t_j$  a una colección de lugares  $I(t_j)$  conocida como los lugares de entrada de la transición. La función de salida  $O$  mapea a cada  $t_j$  a una colección de lugares  $O(t_j)$  conocida como los lugares de salida de la transición.

Las RdP pueden proveer modelos exactos debido a las siguientes cuatro razones :

1.- Capturan las relaciones de precedencia e interacción de eventos concurrentes y asíncronos.

2.- Son modelos lógicos derivados de conocer la forma como trabaja el sistema. Como resultado, son fáciles de entender debido a su naturaleza gráfica.

3.- Los *deadlocks*, conflictos y el tamaño de los *buffers* se pueden modelar en forma fácil y concisa.

4.- Debido a su fundamento matemático permiten el análisis cuantitativo y cualitativo del sistema.

Los objetivos básicos considerados para el desarrollo de este trabajo de tesis son :

1) Proponer un método matemático basado en la solución de un sistema de ecuaciones lineales que permita verificar la presencia o ausencia de bloqueo dentro de una RdP.

2) Implementar en un sistema de cómputo el método descrito anteriormente.

3) Diseñar y construir un ambiente visual para la modelación, simulación y análisis de sistemas usando las RdP.

El sistema de simulación de RdP desarrollado (SSRdP), es un sistema para computadoras personales IBM-PC compatibles implementado para crear, ejecutar y analizar RdP, esto permite que el uso de RdP resulte más atractivo y menos monótono. Dicho sistema tiene como características principales, las siguientes :

1.- Cuenta con un editor gráfico lo que permite crear y editar una RdP en una forma amigable.

2.- Permite ejecutar una RdP y ver gráficamente los diferentes estados que alcanza el sistema. Existen dos modos de ejecución : de todas las transiciones que tienen satisfechas sus condiciones para su ocurrencia,

i) El usuario elige la transición a ocurrir (lo que provoca que cambie el estado del sistema).

ii) El sistema elige de manera aleatoria la transición a ocurrir.

3.- Permite analizar una RdP, aquí podemos determinar si una RdP tiene las dos propiedades siguientes :

i) Determinar si un estado dado es alcanzable a partir del estado inicial (Alcanzabilidad).

ii) Determinar si existe algún estado alcanzable en el cual la RdP quede bloqueada (Vivacidad).

Este trabajo está basado en las siguientes áreas :

1.- Programación visual. El sistema cuenta con un editor gráfico que permite dibujar de una forma amigable la estructura de una RdP, ejecutar una RdP y ver gráficamente los diferentes estados que alcanza el sistema.

2.- Simulación. Al ejecutar una RdP se simula el comportamiento del sistema que se está modelando.

3.- Teoría de las Redes de Petri. El sistema se puede usar para la enseñanza de la teoría de las RdP.

En el primer capítulo de este trabajo, se proporcionan los conceptos básicos sobre las RdP.

El material que se presenta en el segundo capítulo es un método de análisis que permite determinar la vivacidad en una RdP. El método propuesto, contenido de este capítulo, es una de las aportaciones originales de la tesis no reportada en la literatura existente.

En el capítulo 3 se describe el sistema de cómputo que se implementó para manejar las RdP, representa la integración de los conocimientos considerados en los capítulos 1 y 2 en una herramienta automatizada.

Finalmente, en el capítulo 4 se presentan algunos ejemplos de los tipos de sistemas que pueden ser simulados con *SSRdP*, el sistema desarrollado.

# CAPITULO 1

## TEORIA DE LAS REDES DE PETRI

### 1.1 Introducción

En este capítulo se dan las definiciones básicas de las RdP, fundamentales para el buen entendimiento de éstas [Peterson81].

Este formalismo se basa en la teoría de *bags*. La teoría de *bags*, es una extensión natural de la teoría de conjuntos. Sin embargo, a diferencia de los conjuntos, los *bags* permiten múltiples ocurrencias de los elementos. En la teoría de conjuntos, un elemento es miembro de un conjunto o no lo es. En la teoría de *bags*, un elemento puede estar cero veces en un *bag* ó cualquier número positivo de veces.

Ejemplos de *bags* sobre el dominio {a, b,c,d} son los siguientes :

$$B_1 = \{a, b, c\}$$

$$B_2 = \{a, b, c, c\}$$

$$B_3 = \{c, c, a\}$$

### 1.2 Estructura de RdP

**Definición 1.1** Una RdP es una cuádrupla  $R = \langle P, T, I, O \rangle$  tal que

P es un conjunto finito y no vacío de lugares.

T es un conjunto finito y no vacío de transiciones

$P \cap T = \emptyset$  ; es decir, lugares y transiciones son conjuntos disjuntos

$I: T \rightarrow P^\infty$  es la función de entrada

$O: T \rightarrow P^\infty$  es la función de salida.

Las dos funciones mapean transiciones en *bags* de lugares. ( $P^\infty$  es el conjunto de *bags* sobre el dominio P, en las cuales no existe límite en el número de ocurrencias de un elemento).

Un lugar  $p_i$  es un lugar de entrada de una transición  $t_j$  si  $p_i \in I(t_j)$ ;  $p_i$  es un lugar de salida si  $p_i \in O(t_j)$ . Las entradas y salidas de una transición son *bags* de lugares, lo que permite a un lugar ser entrada o salida múltiple de una transición. La multiplicidad de un lugar de entrada  $p_i$  de una transición  $t_j$  es el número de ocurrencias del lugar en el *bag* de entrada de la transición, es decir,  $\#(p_i, I(t_j))$ . Análogamente, la multiplicidad de un lugar de salida  $p_i$  para una transición  $t_j$  es  $\#(p_i, O(t_j))$ .

A continuación se tiene un ejemplo de una estructura de RdP:

### Ejemplo 1.1

$$R=(P,T,I,O)$$

$$P=\{p_1, p_2, p_3, p_4\}, T=\{t_1, t_2, t_3\}$$

$$I(t_1) = \{ p_1, p_2, p_3 \} \quad O(t_1) = \{ p_1 \}$$

$$I(t_2) = \{ p_4 \} \quad O(t_2) = \{ p_2, p_2, p_3 \}$$

$$I(t_3) = \{ p_3 \} \quad O(t_3) = \{ p_4 \}$$

### 1.3 Representación gráfica

Una RdP se representa gráficamente por un grafo dirigido bipartido. Los lugares se representan por círculos y las transiciones por barras. Los lugares y transiciones se conectan por arcos dirigidos. Un arco dirigido de un lugar  $p_i$  a una transición  $t_j$  define el lugar como una entrada de la transición. Múltiples entradas a una transición son indicadas por múltiples arcos desde el lugar de entrada a la transición. Un lugar de salida es indicado por un arco desde la transición al lugar. Análogamente múltiples salidas son representadas por múltiples arcos.

**Definición 1.2** Una gráfica  $G$  de RdP es una gráfica múltiple dirigida bipartida,  $G=(V,A)$  donde  $V = \{ v_1, v_2, \dots, v_s \}$  es un conjunto de vértices,  $s \geq 0$ , y  $A=\{a_1, a_2, \dots, a_r\}$  es un *bag* de arcos dirigidos,  $r \geq 0$ ;

$a_i=(v_i, v_k)$ , con  $v_j, v_k \in V$ . El conjunto  $V$  puede partirse en dos conjuntos disjuntos,  $P$  y  $T$ , tales que  $V = P \cup T$ ,  $P \cap T = \emptyset$ , y para cada arco dirigido  $a_i \in A$ , si  $a_i = (v_j, v_k)$ , entonces  $v_j \in P$  y  $v_k \in T$  ó  $v_j \in T$  y  $v_k \in P$ .

En la Fig. 1.1 se tiene la gráfica que representa la estructura de la RdP del ejemplo 1.1.

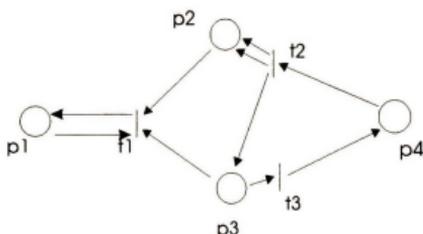


Fig. 1.1 La gráfica de una RdP

#### 1.4 Representación matricial

Una forma equivalente de definir una RdP es por medio de dos matrices  $D^-$  y  $D^+$  para representar las funciones de entrada y salida. Sea  $|P| = n$  (número de lugares de la red) y sea  $|T| = m$  (número de transiciones de la red). Cada matriz es un arreglo de  $n$  filas (una por cada lugar) por  $m$  columnas (una por cada transición).

Se denomina matriz de entrada a la matriz

$$D^-[i,j] = \#(p_i, I(t_j)). \quad (\text{Multiplicidad de entrada del lugar } i\text{-ésimo en la transición } j\text{-ésima.})$$

Se denomina matriz de salida a la matriz

$$D^+[i,j] = \#(p_i, O(t_j)). \quad (\text{Multiplicidad de salida del lugar } i\text{-ésimo en la transición } j\text{-ésima.})$$

Una RdP se puede definir como  $\langle P, T, D^-, D^+ \rangle$  lo que es equivalente a la definición anterior  $R = \langle P, T, I, O \rangle$ .

Sea  $e[j]$  el m-vector unidad el cual tiene 1 en la j-ésima entrada y cero en las demás. La transición  $t_j$  es representada por el m-vector unidad  $e[j]$ . Al multiplicar  $D^-$  por  $e[j]$  se obtiene la columna j-ésima de la matriz  $D^-$ . La columna j-ésima contiene la multiplicidad de entrada de cada lugar en la transición  $j$ . De manera análoga para  $D^+$ .

A continuación se dan las matrices de entrada y salida para la RdP de la fig. 1.1

$$D^- = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad D^+ = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

**Definición 1.3** El marcado  $M$  de una RdP es una aplicación  $M: P \rightarrow \mathbb{N}$ , o sea, la asignación de un número entero no negativo (número de marcas (*tokens*)) a cada lugar.

**Definición 1.4** Una RdP marcada  $M = (R, \mu)$  es una estructura de RdP  $R = (P, T, I, O)$  y un marcado  $\mu$ . Esto también se escribe como  $M = (P, T, I, O, \mu)$ .

En la gráfica asociada a una RdP, el marcado  $\mu$  se representa por una distribución, en los lugares, de objetos denominados marcas. Una marca se representa gráficamente por un punto en el interior del círculo que define el lugar que lo contiene. Aunque el número de marcas en una RdP difícilmente excede a 5 ó 6, es posible que un marcado tenga 10, 20 ó cientos de marcas asignadas a un lugar. En este caso, sería inconveniente dibujar cientos de marcas dentro de los círculos que representan los lugares. La convención en este caso es escribir el número de marcas en el lugar.

En la fig. 1.2 se tiene una gráfica de RdP marcada.

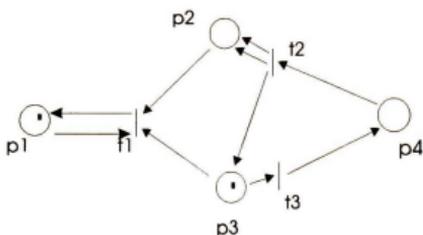


Fig. 1.2 Una gráfica de RdP marcada.

Si  $|P| = n$ , entonces un marcado se representa, en forma matricial, por un vector de  $n$  elementos. La entrada  $i$ -ésima de este vector contiene el número de marcas asignadas al  $i$ -ésimo lugar. El vector de marcado para la RdP marcada de la fig. 1.2 es  $\mu = [1, 0, 1, 0]^t$ .

### 1.5 Reglas de ejecución de RdP

La ejecución de una RdP se controla por el número y distribución de marcas en la RdP. Las marcas residen en los lugares y controlan la ejecución de las transiciones en la red.

Una RdP se ejecuta disparando transiciones. Una transición se dispara removiendo marcas de sus lugares de entrada y creando nuevas marcas las cuales son distribuidas en sus lugares de salida.

Una transición puede dispararse si está habilitada. Una transición está habilitada si cada uno de sus lugares de entrada tiene al menos tantas marcas como arcos a la transición. Se necesitan múltiples marcas para arcos múltiples de entrada. Las marcas en los lugares de entrada que habilitan una transición son sus marcas habilitantes.

**Definición 1.5 (a)** Una transición  $t_j$  en una RdP marcada  $M=(P,T,I,O, \mu)$  está habilitada si para toda  $p_i \in P$ ,  $\mu(p_i) \geq \#(p_i, I(t_j))$ .

**Definición 1.5 (b) (Enfoque matricial)** Una transición  $t_j$  en una RdP marcada  $M=(P,T,D^-,D^+,\mu)$ , está habilitada si cada entrada del vector de marcado es mayor o igual que la entrada correspondiente de la  $j$ -ésima columna de la matriz de entrada, esto lo vamos a denotar como  $\mu \geq D^- e[j]$ .

Una transición se dispara removiendo todas sus marcas activantes de sus lugares de entrada y depositando en cada uno de sus lugares de salida una marca por cada arco de la transición al lugar. Marcas múltiples se producen por arcos múltiples de salida.

En general, disparar una transición cambiará el marcado  $\mu$  de la RdP a un nuevo marcado  $\mu'$ . Ya que solo las transiciones que estén habilitadas pueden dispararse, el número de marcas en cada lugar es siempre no negativo cuando una transición se dispara. Al disparar una transición nunca se puede intentar remover una marca de donde no la hay. Si no hay suficientes marcas en cualquier lugar de entrada de una transición, entonces dicha transición no está habilitada y no puede dispararse.

**Definición 1.6 (a)** Una transición  $t_j$  en una RdP marcada  $M=(P,T,I,O,\mu)$  puede dispararse si está habilitada. Al disparar una transición habilitada  $t_j$  se obtiene un nuevo marcado  $\mu'$ , definido por  $\mu'(p_i) = \mu(p_i) - \#(p_i, I(t_j)) + \#(p_i, O(t_j))$ ,  $p_i \in P$ ,  $t_j \in T$ .

**Definición 1.6 (b) (Enfoque matricial)** Una transición  $t_j$  en RdP marcada  $M=(P,T,D^-,D^+,\mu)$  puede dispararse si está habilitada. El resultado de disparar la transición  $t_j$  en el marcado  $\mu$  es

$$\begin{aligned} \delta(\mu, t_j) &= \mu' = \mu - D^- e[j] + D^+ e[j] \\ &= \mu + (D^+ - D^-) e[j] \\ \mu' &= \mu + D e[j] \end{aligned}$$

A esta ecuación se le conoce como la ecuación de estado, donde  $D = D^+ - D^-$  es llamada la matriz de incidencia.

Como ejemplo, considere la RdP marcada de la fig. 1.3. Con este marcado  $t_3$  es la única transición habilitada. La transición  $t_1$  no está habilitada porque no existe ninguna marca en  $p_2$ , el cual es lugar de entrada de esta transición. Similarmente, la transición  $t_2$  no está habilitada porque no existe ninguna marca en  $p_4$ . Ya que la transición  $t_3$  está habilitada, esta se puede disparar. Si se dispara la transición  $t_3$ , remueve una marca de  $p_3$  y deposita una marca en  $p_4$ . Así, el nuevo marcado que resulta al disparar  $t_3$  se muestra en la fig. 1.4.

En la RdP marcada de la fig. 1.4 la única transición habilitada es  $t_2$ . Si se dispara esta transición remueve la marca de  $p_4$  y deposita marcas en  $p_2$  y  $p_3$  ( 2 marcas en  $p_2$  ya que es salida múltiple de  $t_2$ ). Esto produce el marcado de la fig. 1.5. En esta RdP marcada, las transiciones  $t_1$  y  $t_3$  están activas.

Los disparos de transiciones pueden continuar siempre que exista por lo menos una transición habilitada. Cuando no existen transiciones habilitadas la ejecución se detiene.

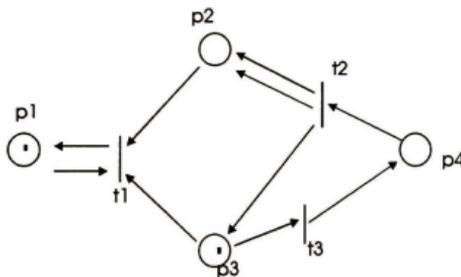


Fig. 1.3 Una RdP marcada para ilustrar las reglas de ejecución.

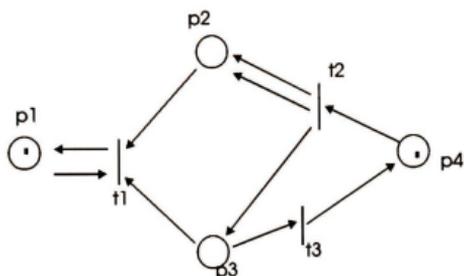


Fig. 1.4. El marcado que resulta al disparar la transición t3 en la fig. 1.3

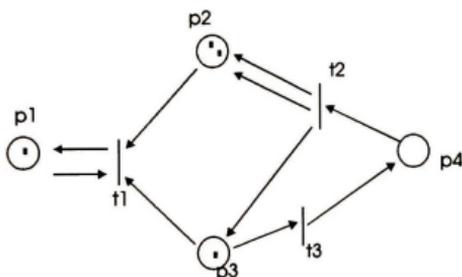


Fig 1.5. El marcado que resulta al disparar la transición t2 en la fig. 1.4

Ahora vamos a mostrar la ejecución de la RdP de la fig. 1.3 pero con el enfoque matricial.

La matriz de entrada y la matriz de salida son

$$D^- = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad D^+ = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

La matriz de incidencia y el vector de marcado son

$$D = D^+ - D^- = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & 0 \\ -1 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

$$\mu_0 = [1, 0, 1, 0]^t$$

La única transición habilitada es  $t_3$  ya que la desigualdad  $\mu_0 \geq D^- e[3]$  ( $[1, 0, 1, 0]^t \geq [0, 0, 1, 0]^t$ ) es verdadera.

La transición  $t_1$  no está habilitada ya que la desigualdad  $\mu_0 \geq D^- e[1]$  ( $[1, 0, 1, 0]^t \geq [1, 1, 1, 0]^t$ ) es falsa. Similarmente la transición  $t_2$  no está habilitada ya que la desigualdad  $\mu_0 \geq D^- e[2]$  ( $[1, 0, 1, 0]^t \geq [0, 0, 0, 1]^t$ ) es falsa.

El resultado de disparar la transición  $t_3$  en el marcado  $\mu_0$  al aplicar la ecuación de estado es:

$$\begin{aligned} \mu_1 &= \mu_0 + D e[3] \\ &= [1, 0, 1, 0]^t + [0, 0, -1, 1]^t \end{aligned}$$

$$\mu_1 = [1, 0, 0, 1]^t$$

Este es el marcado que se muestra en la fig. 1.4.

En la RdP marcada de la fig 1.4 la única transición habilitada es  $t_2$ . La ejecución de la red puede continuar mientras existan transiciones habilitadas.

### ***1.6 Espacio de estados de RdP***

El estado de una RdP se define por su marcado. El disparo de una transición representa un cambio en el estado de la RdP mediante

un cambio en el mercado de la red. El espacio de estados de una RdP con  $n$  lugares es el conjunto de todos los marcados, es decir,  $N^n$ . El cambio en el estado provocado por el disparo de una transición se define por la función  $\delta$ , llamada la función siguiente-estado. Cuando  $\delta$  se aplica a un marcado (estado)  $\mu$  y a una transición  $t_j$ , produce el nuevo marcado (estado) que resulta de disparar la transición  $t_j$  en el marcado  $\mu$ . Ya que  $t_j$  sólo puede dispararse si está habilitada,  $\delta$  no está definida si  $t_j$  no está habilitada en el marcado  $\mu$ . Si  $t_j$  está habilitada, entonces  $\delta(\mu, t_j) = \mu'$ , donde  $\mu'$  es el marcado que resulta de remover las marcas de las entradas de  $t_j$  y añadir marcas a las salidas de  $t_j$ .

**Definición 1.7** La función siguiente-estado  $\delta : N \times T \rightarrow N$ , para una RdP  $R=(P,T,I,O)$  con marcado  $\mu$  y transición  $t_j \in T$ , está definida si y solo si  $\mu(p_i) \geq \#(p_i, I(t_j))$ , para todo  $p_i \in P$ . Si así sucede entonces  $\delta(\mu, t_j) = \mu'$ , donde  $\mu'(p_i) = \mu(p_i) - \#(p_i, I(t_j)) + \#(p_i, O(t_j))$ , para toda  $p_i \in P$ .

Dada una RdP  $R=(P,T,I,O)$  y un marcado inicial  $\mu_0$ , podemos ejecutar la RdP por disparos sucesivos de transiciones. Al disparar una transición habilitada  $t_j$  en el marcado inicial, se produce un nuevo marcado  $\mu_1 = \delta(\mu_0, t_j)$ . En este nuevo marcado podemos disparar cualquier otra transición habilitada. Si alcanzamos un marcado en el que ninguna transición está habilitada, entonces ninguna transición puede dispararse, la función siguiente-estado no está definida y la ejecución debe detenerse.

Dos sucesiones resultan de la ejecución de una RdP: la sucesión de marcado ( $\mu_0, \mu_1, \mu_2, \dots$ ) y la sucesión de transiciones que se dispararon ( $t_{j_0}, t_{j_1}, t_{j_2}, \dots$ ). Estas dos sucesiones se relacionan por la función  $\delta$ , tal que  $\delta(\mu_k, t_{j_k}) = \mu_{k+1}$ , para  $k \in N \cup \{0\}$ . Dada una sucesión de transiciones y  $\mu_0$ , podemos fácilmente obtener la sucesión de marcados para la ejecución de la RdP y dada la sucesión de marcados, excepto por unos casos degenerados, podemos obtener la sucesión de transiciones. Ambas sucesiones nos proporcionan un registro de la ejecución de la RdP.

**Definición 1.8** Para una RdP  $R=(P,T,I,O)$  con marcado  $\mu$ , un marcado  $\mu'$  es inmediatamente alcanzable si existe una transición  $t_j \in T$  tal que  $\delta(\mu, t_j) = \mu'$ .

Se puede extender este concepto para definir el conjunto de marcados alcanzables para una RdP marcada. Si  $\mu'$  es inmediatamente alcanzable desde  $\mu$  y  $\mu''$  es inmediatamente alcanzable desde  $\mu'$ , se dice que  $\mu''$  es alcanzable desde  $\mu$ . Se define el conjunto de estados alcanzables  $A(R, \mu)$  de una RdP con marcado  $\mu$ , como aquel que está formado por todos los marcados que son alcanzables desde  $\mu$ . Un marcado  $\mu'$  está en  $A(R, \mu)$  si existe una sucesión de disparos de transiciones que pasen del marcado  $\mu$  al marcado  $\mu'$ .

**Definición 1.9** El conjunto de estados alcanzables  $A(R, \mu)$ , para una RdP  $R=(P,T,I,O)$  con marcado  $\mu$ , es el conjunto más pequeño de marcados definido por:

- i)  $\mu \in A(R, \mu)$ .
- ii) Si  $\mu' \in A(R, \mu)$  y  $\mu'' = \delta(\mu', t_j)$  para alguna  $t_j \in T$ , entonces  $\mu'' \in A(R, \mu)$ .

En el enfoque matricial para una secuencia de disparos de transiciones  $\sigma = t_{j_1} t_{j_2} \dots t_{j_k}$  a partir de un marcado inicial  $\mu_0$ , se tiene

$$\begin{aligned} \mu' &= \delta(\mu_0, \sigma) = \delta(\mu_0, t_{j_1} t_{j_2} \dots t_{j_k}) \\ &= \mu_0 + D e[j_1] + D e[j_2] + \dots + D e[j_k] \\ &= \mu_0 + D (e[j_1] + e[j_2] + \dots + e[j_k]) \\ &= \mu_0 + D f(\sigma) \end{aligned}$$

El vector  $f(\sigma) = e[j_1] + e[j_2] + \dots + e[j_k]$  se le conoce como el *vector de disparo o característico* de la secuencia  $t_{j_1} t_{j_2} \dots t_{j_k}$ . El  $i$ -ésimo elemento de  $f(\sigma)$ ,  $f(\sigma)_i$ , es el número de veces que la transición  $t_i$  se dispara en la secuencia  $t_{j_1} t_{j_2} \dots t_{j_k}$ . El *vector de disparo* es entonces un vector de enteros no negativos.

## ***1.7 Modelado con RdP***

Las RdP se diseñaron y se utilizan principalmente para modelar sistemas. Muchos sistemas, especialmente aquellos con componentes independientes, pueden modelarse con una RdP, considerando la ocurrencia de diversos eventos y actividades, así como el flujo de información u otros recursos dentro de los sistemas.

### ***1.7.1 Eventos y condiciones***

La simple inspección de un sistema en términos de RdP se concentra en dos conceptos primitivos: eventos y condiciones. Los eventos son acciones que tienen lugar en el sistema, cuyo estado controla la ocurrencia de dichos eventos. El estado de un sistema puede describirse como un conjunto de condiciones. Una condición es un predicado o descripción lógica del estado de un sistema. Como tal, una condición puede ser verdadera o falsa.

Ya que los eventos son acciones, estos pueden ocurrir. Para que un evento ocurra es necesario que se cumplan ciertas condiciones. Estas son las precondiciones del evento. La ocurrencia del evento puede causar que se dejen de cumplir las precondiciones y que se cumplan otras condiciones las que se le conoce como las postcondiciones del evento.

Como un ejemplo, considere el problema de modelar la celda de ensamble de la fig 1.6. Para llevar a cabo una tarea de ensamble se necesita cualquier transportador y dos robots vecinos. Cada transportador requiere primero el robot izquierdo y después de adquirirlo, requiere el robot derecho. Después de esto la operación de ensamble empieza. El transportador libera ambos rotos cuando termina la tarea de ensamble.

*Nota. Los robots izquierdo y derecho se determinan posicionandose en el centro de la fig. 1.6 mientras se observa a cada transportador.*

Las condiciones del sistema son:

- a.  $T_1$  requiere su robot izquierdo  $R_1$
- b.  $T_2$  requiere su robot izquierdo  $R_2$
- c.  $T_3$  requiere su robot izquierdo  $R_3$
  
- d.  $T_1$  requiere su robot derecho  $R_2$
- e.  $T_2$  requiere su robot derecho  $R_3$
- f.  $T_3$  requiere su robot derecho  $R_1$
  
- g.  $T_1$  y sus dos robots  $R_1$  y  $R_2$  están ocupados
- h.  $T_2$  y sus dos robots  $R_2$  y  $R_3$  están ocupados
- i.  $T_3$  y sus dos robots  $R_3$  y  $R_1$  están ocupados
  
- j.  $R_1$  está libre
- k.  $R_2$  está libre
- l.  $R_3$  está libre

Los eventos son:

- 1.  $T_1$  adquiere su robot izquierdo  $R_1$
- 2.  $T_2$  adquiere su robot izquierdo  $R_2$
- 3.  $T_3$  adquiere su robot izquierdo  $R_3$
- 4.  $T_1$  adquiere su robot derecho  $R_2$
- 5.  $T_2$  adquiere su robot derecho  $R_3$
- 6.  $T_3$  adquiere su robot derecho  $R_1$
- 7.  $T_1$  libera a  $R_1$  y  $R_2$
- 8.  $T_2$  libera a  $R_2$  y  $R_3$
- 9.  $T_3$  libera a  $R_3$  y  $R_1$

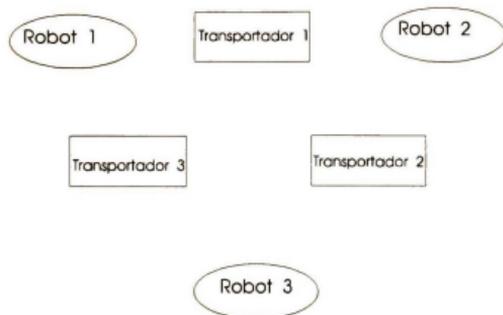


Fig. 1. 6. Una celda de ensamble

Las precondiciones del evento 1 ( $T_1$  adquiere su robot izquierdo  $R_1$ ) son (a)  $T_1$  requiere su robot izquierdo  $R_1$  y (j)  $R_1$  está libre. La postcondición del evento 1 es (d)  $T_1$  requiere su robot derecho  $R_2$ .

De manera análoga se pueden definir las precondiciones y postcondiciones de los otros eventos y construir la siguiente tabla de eventos y sus precondiciones y postcondiciones.

<b>Evento</b>	<b>Precondiciones</b>	<b>Postcondiciones</b>
1	a, j	d
2	b, k	e
3	c, l	f
4	d, k	g
5	e, l	h
6	f, j	i
7	g	a, j, k
8	h	b, k, l
9	i	c, j, l

Este sistema se puede modelar fácilmente con una RdP. Las condiciones se modelan por lugares y los eventos por transiciones. Las entradas de una transición son las precondiciones del evento correspondiente y las salidas son las postcondiciones. La ocurrencia de un evento corresponde al disparo de la transición correspondiente.

El que se cumpla una condición se representa por una marca en el lugar que corresponde a la condición. Cuando una transición se dispara remueve las marcas activantes lo que representa que las precondiciones se dejan de cumplir y crea nuevas marcas lo que representa que se cumplan las postcondiciones.

La RdP de la fig. 1.7 es un modelo de RdP de la celda de ensamble.

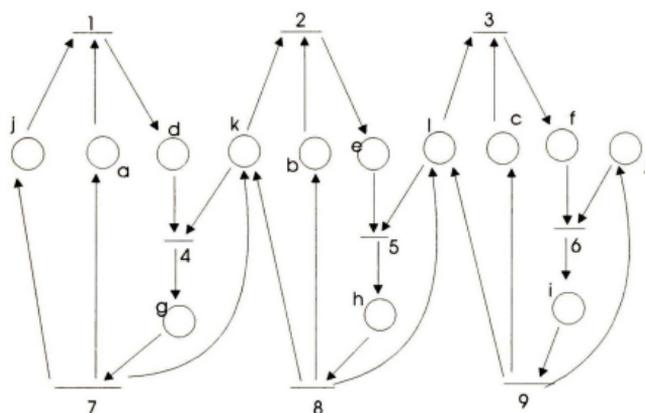


Fig. 1.7. Una RdP para modelar una celda de ensamble.

## 1.8 Análisis de RdP

Una vez que se modela un sistema con un RdP es de utilidad analizar la red para determinar que propiedades tiene. Es decir, en el análisis se determinará si el modelo del sistema que se ha diseñado cumple una serie de propiedades que caracterizarán su buen funcionamiento.

En esta sección se describen las propiedades de las RdP y algunas de las técnicas de análisis que existen.

### 1.8.1 Propiedades de las RdP

**Alcanzabilidad:** El concepto de alcanzabilidad es esencial para entender las definiciones de las propiedades de las RdP.

**Definición 1.10.** Un marcado  $\mu_r$  es alcanzable desde  $\mu_o$  si existe una secuencia de disparos de transiciones que produzcan a  $\mu_r$ .

**Seguridad:** Para una RdP que modela un dispositivo de hardware, una de las propiedades más importantes es la seguridad. Un lugar en una RdP es seguro si el número de marcas en ese lugar nunca excede a uno. Una RdP es segura si todos los lugares son seguros.

**Definición 1.11** Un lugar  $p_i \in P$  de una RdP  $R=(P,T,I,O,\mu_o)$  es seguro si para todo  $\mu' \in A(R,\mu_o)$ ,  $\mu'(p_i) \leq 1$ . Una RdP es segura si cada lugar es seguro.

**Acotamiento:** Seguridad es un caso especial de una propiedad más general llamada acotamiento. Un lugar es k-seguro ó k-acotado si el número de marcas en ese lugar no puede exceder a un entero k.

**Definición 1.12.** Un lugar  $p_i \in P$  de una RdP  $R=(P,T,I,O,\mu_o)$  es k-seguro si para todo  $\mu' \in A(R,\mu_o)$ ,  $\mu'(p_i) \leq k$ .

Un lugar que es 1-seguro es simplemente llamado seguro. Note que la cota  $k$  en el número de marcas que puede estar en un lugar puede ser una función del lugar (es decir, el lugar  $p_1$  puede ser 3-seguro mientras que el lugar  $p_2$  sea 8-seguro). Sin embargo, si un lugar  $p_i$  es  $k$ -seguro, entonces este también es  $k'$ -seguro para toda  $k' > k$ . Ya que hay un número finito de lugares podemos escoger  $k$  que sea el máximo de las cotas de cada lugar y definir una RdP  $k$ -segura si todos sus lugares son  $k$ -seguros.

**Vivacidad.** Un problema que puede aparecer en la reservación de recursos es el bloqueo (*deadlock*). Como ejemplo, considere un sistema con dos recursos diferentes  $q$  y  $r$  y dos procesos  $a$  y  $b$ . Si ambos procesos necesitan ambos recursos, va a ser necesario compartirlos. Para realizar esto se necesita que cada proceso solicite un recurso y después lo libere. Ahora suponga que el proceso  $a$  primero solicita al recurso  $q$  y después al recurso  $r$  y finalmente libera a ambos. El proceso  $b$  lo hace de manera análoga pero primero solicita el recurso  $r$  y después a  $q$ . La fig 1.8 ilustra los 2 procesos y la reservación de recursos con una RdP.

El marcado inicial indica que los recursos  $q(p4)$  y  $r(p5)$  están disponibles y los procesos  $a$  y  $b$  están listos. Una ejecución de esta red es  $t_1t_2t_3t_4t_5t_6$  y otra es  $t_4t_5t_6t_1t_2t_3$ . Ninguna de estas ejecuciones provoca un *deadlock*. Sin embargo considere la secuencia que empieza  $t_1t_4$ , el proceso  $a$  tiene a  $q$  y quiere a  $r$ , el proceso  $b$  tiene a  $r$  y quiere a  $q$ . El sistema está bloqueado y ningún proceso puede proceder.

**Definición 1.13.** Una RdP es viva (libre de bloqueo) con respecto a un marcado inicial  $\mu_o$  si para toda  $\mu' \in A(R, \mu_o)$ , es posible disparar al menos a una transición. El que una red sea viva garantiza la ausencia de bloqueo.

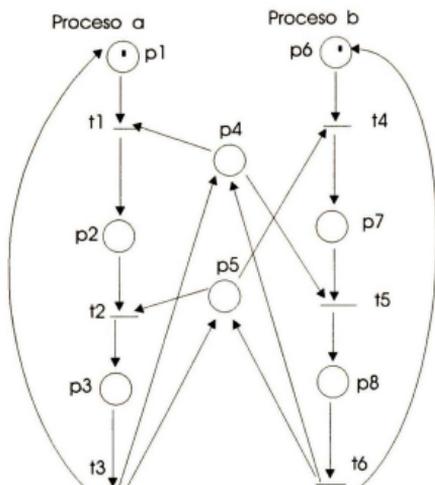


Fig. 1.8. Una RdP para ilustrar la reservación de recursos.

**Secuencias de disparo.** Otro enfoque del análisis se concentra en secuencias de disparos de transiciones más que en estados. Esto está relacionado con la vivacidad, ya que podemos preguntar, si se puede disparar la transición  $t_j$ . Pero de manera más general podemos querer determinar si es posible que ocurra una secuencia específica de disparos de transiciones o si cualquiera de un conjunto de secuencias de disparos es posible. Estas preguntas de análisis llevan al concepto de lenguajes de RdP.

**Problemas de equivalencia y subconjunto.** Una clase de problemas es la optimización. Si una RdP exhibe un cierto comportamiento, como es indicado por su conjunto de secuencias de disparos de transiciones y su conjunto de alcanzabilidad, un problema de interés es cambiar (optimizar) la RdP sin afectar su comportamiento.

Esto puede involucrar borrar transiciones que nunca pueden dispararse y lugares que nunca pueden ser marcados o tal vez la redefinición de algunas transiciones. Se puede mostrar que 2

diferentes RdP marcadas con el mismo número de transiciones (pero tal vez diferente número de lugares) generará la misma secuencia de disparos de transiciones o que 2 diferentes RdP marcas con el mismo número de lugares (pero tal vez diferente número de transiciones) generará el mismo conjunto de alcanzabilidad. Esto permitirá modificar RdP para incrementar paralelismo, decrementar el costo de implementación u otras optimizaciones.

En estos casos, el interés es determinar si dos RdP son equivalentes o si una es un subconjunto de la otra. Se debe ser cuidadoso con estos problemas para definir la noción de equivalencia o contención. Si se define equivalencia como conjuntos iguales de alcanzabilidad, entonces no se puede cambiar el número de lugares; mientras que, si se requiere igualdad de conjuntos de secuencias de disparos de transiciones no se puede cambiar las transiciones.

### ***1.8.2 Técnicas de análisis.***

Las propiedades presentadas anteriormente son las más comunes encontradas en la literatura. Las técnicas de análisis van a permitir encontrar algunas de las propiedades que tienen las RdP.

Existen dos técnicas de análisis que más se han usado con RdP, estas son:

- 1) El árbol de alcanzabilidad.
- 2) Matriz de ecuaciones.

#### ***1.8.2.1 El árbol de alcanzabilidad***

Este representa el conjunto de marcados alcanzables a partir de un marcado inicial de una RdP. El marcado inicial va a ser el nodo raíz del árbol. A partir de este marcado, se disparan todas las transiciones activas, esto lleva a un nuevo marcado en el cual pueden estar activas otras transiciones. Si se toma cada uno de estos nuevos marcados como una nueva raíz, se puede generar sucesivamente todos los marcados alcanzables. La estructura del

árbol preserva el orden de disparo de las transiciones. Todo marcado en el conjunto de alcanzabilidad va ser producido y así para cualquier RdP con un conjunto de alcanzabilidad infinito el correspondiente árbol también será infinito. Aún una RdP con un conjunto de alcanzabilidad finito puede tener un árbol infinito. El árbol representa todas las secuencias posibles de disparos de transiciones. Cada camino en el árbol, que inicia en la raíz corresponde a una secuencia válida de disparos de transiciones. Para que el árbol sea una herramienta útil se debe encontrar un medio para limitarlo a un tamaño finito. Se debe notar que si la representación de un conjunto infinito es finito, entonces un número infinito de marcados debe ser mapeado dentro de la misma representación. Esto en general, va a provocar una pérdida de información, lo que puede significar que algunas propiedades de las RdP no puedan ser determinadas.

### 1.8.2.2 Matriz de ecuaciones

El enfoque matricial de la teoría de las RdP proporciona una herramienta útil para atacar el problema de alcanzabilidad. Suponga que un marcado  $\mu'$  es alcanzable desde un marcado  $\mu$ . Entonces existe una secuencia  $\sigma$  (posiblemente nula) de disparos de transiciones que lleva de  $\mu$  a  $\mu'$ . Esto significa que el vector característico  $f(\sigma)$  de  $\sigma$  es una solución en los enteros no negativos, para  $\chi$  en la ecuación matricial

$$\mu' = \mu + D\chi \quad (1.1)$$

Entonces si  $\mu'$  es alcanzable desde  $\mu$ , la ecuación 1.1 tiene una solución en los enteros no negativos, si la ecuación 1.1 no tiene solución, entonces  $\mu'$  no es alcanzables desde  $\mu$ .

Como un ejemplo considere la RdP de la figura 1.9. Las matrices  $D^-$  y  $D^+$  son:

$$D^- = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad D^+ = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

La matriz D y el vector de marcado inicial son:

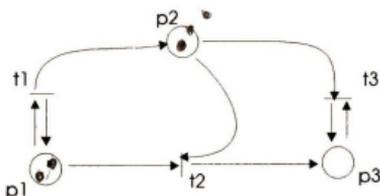


Fig. 1.9. Una RdP para ilustrar la alcanzabilidad con el enfoque matricial.

$$D = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 1 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad \mu_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Para determinar si el marcado  $[0,3,1]^T$  es alcanzable desde el marcado  $[1,0,0]^T$ , tenemos que resolver la ecuación:

$$\begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ 1 & 1 & -1 \\ 0 & 1 & 0 \end{bmatrix} x$$

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 1 & -1 \\ 0 & 1 & 0 \end{bmatrix} x = \begin{bmatrix} -1 \\ 3 \\ 1 \end{bmatrix}$$

El vector  $[2,1,0]^T$  es una solución la cual tiene asociada la secuencia  $\sigma = t_1 t_1 t_2$ . Se puede comprobar en la fig. 1.15 que el marcado  $[0,3,1]^T$  es alcanzable desde el marcado  $[1,0,0]^T$  con la secuencia  $\sigma$ .

Podemos afirmar que el marcado  $[2,0,1]^T$  no es alcanzable desde el marcado  $[1,0,0]^T$ , ya que la ecuación

$$\begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ 1 & 1 & -1 \\ 0 & 1 & 0 \end{bmatrix} x$$

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 1 & -1 \\ 0 & 1 & 0 \end{bmatrix} x = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

no tiene solución.

**Deficiencias del enfoque matricial.** La matriz  $D$  por sí sola no refleja apropiadamente la estructura de la RdP. Los lugares que son entradas y salidas de la misma transición se representan en la misma posición de las matrices  $D^-$  y  $D^+$ , esto va a cancelar la salida en la matriz  $D = D^+ - D^-$ .

Otro problema es la carencia de información en el vector de disparo, es decir aunque nosotros sabemos el número de veces que se va a disparar cada transición no podemos saber el orden de los disparos.

Otro problema es que la solución a la ecuación 1.1 es necesaria para la alcanzabilidad pero no es suficiente. Es decir, la ecuación 1.1 puede tener solución y sin embargo el marcado puede no ser alcanzable.

## CAPITULO 2

### ANALISIS DE LA VIVACIDAD DE LAS RdP POR RESOLUCION DE UN SISTEMA LINEAL

#### 2.1 Introducción

Tal vez la propiedad más importante de las RdP es la vivacidad. Algunas veces es esencial poder asegurar que una RdP no podrá alcanzar en algún momento un marcado en el cual la red no pueda seguir ejecutándose, en este momento se dice que la RdP está bloqueada (*deadlock*).

En este capítulo se presenta un método matemático basado en la solución de un sistema de ecuaciones lineales que permite verificar la presencia o ausencia de bloqueo dentro de una RdP.

El material que se presenta en este capítulo es una extensión al artículo "*Analyse de la pseudo-vivacité des Réseaux de Petri par résolution d'un système linéaire*" [Agaoua87], este artículo tiene la restricción en que la multiplicidad de los lugares de entrada tiene que ser a lo más 1. En este trabajo se salva esta restricción.

#### 2.2 Principio del método

La negación de la definición de vivacidad de una RdP constituye la definición de bloqueo de una RdP.

**Definición 2.1.** Una RdP marcada  $R=(P,T,D^-,D^+,\mu_0)$  es susceptible de bloqueo si existe un marcado  $\mu$  que pertenece al conjunto de marcados alcanzables a partir de un marcado  $\mu_0$  en el que ninguna transición está habilitada.

La definición anterior es equivalente a la formulación siguiente:

$\exists \mu \in A(R, \mu_0) \wedge \forall t_j \in T \exists p_i \in P$  tal que  $\mu[i] < D^{-}[i, j]$  donde  $i \in [1, \dots, n]$  y  $j \in [1, \dots, m]$ . (1)

Para agilizar la escritura los elementos de  $D^{-}$  se denotarán por  $e_{ij}$  y el marcado que cumple la proposición (1) se denotará por  $\mu$ .

La expresión (1) quiere decir lo siguiente :

$$\begin{aligned} & (\mu[1] < e_{11} \vee \mu[2] < e_{21} \vee \dots \vee \mu[n] < e_{n1} ) \\ & \wedge (\mu[1] < e_{12} \vee \mu[2] < e_{22} \vee \dots \vee \mu[n] < e_{n2} ) \\ & \quad \cdot \\ & \quad \cdot \\ & \quad \cdot \\ & \wedge (\mu[1] < e_{1m} \vee \mu[2] < e_{2m} \vee \dots \vee \mu[n] < e_{nm} ) \end{aligned} \tag{2}$$

La desigualdad  $\mu[i] < e_{ij}$  nunca se cumple si  $e_{ij} = 0$  ya que los valores de  $\mu[i]$  representa el número de marcas asignadas al lugar  $i$ -ésimo el cual es un entero no negativo.

Entonces si  $e_{ij} \neq 0$  la desigualdad  $\mu[i] < e_{ij}$  se cumple si y solo si  $\mu[i]/e_{ij} = 0$  donde el operador  $'/'$  representa la división entera.

Con esto la expresión (2) se simplifica a:

$$\begin{aligned} & (\mu[1]/e_{11} = 0 \text{ si } e_{11} \neq 0 \vee \mu[2]/e_{21} = 0 \text{ si } e_{21} \neq 0 \vee \dots \vee \mu[n]/e_{n1} \text{ si } e_{n1} \neq 0 ) \\ & \quad \cdot \\ & \quad \cdot \\ & \quad \cdot \\ & \wedge (\mu[1]/e_{1m} = 0 \text{ si } e_{1m} \neq 0 \vee \mu[2]/e_{2m} = 0 \text{ si } e_{2m} \neq 0 \vee \dots \vee \mu[n]/e_{nm} \text{ si } e_{nm} \neq 0 ) \end{aligned} \tag{3}$$

En adelante adoptaremos la siguiente convención:

$$\forall k \quad \begin{aligned} e_{ij} \cdot (\mu[k]/e_{ij}) &= \mu[k]/e_{ij} \text{ si } e_{ij} \neq 0 \\ e_{ij} \cdot (\mu[i]/e_{ij}) &= 1 \quad \text{si } e_{ij} = 0 \end{aligned} \quad (4)$$

Con esta notación la expresión (3) es equivalente al siguiente sistema:

$$\begin{aligned} e_{11} \cdot (\mu[1]/e_{11}) \cdot e_{21} \cdot (\mu[2]/e_{21}) \dots e_{n1} \cdot (\mu[n]/e_{n1}) &= 0 \\ \cdot \\ \cdot \\ \cdot \\ e_{1m} \cdot (\mu[1]/e_{1m}) \cdot e_{2m} \cdot (\mu[2]/e_{2m}) \dots e_{nm} \cdot (\mu[n]/e_{nm}) &= 0 \end{aligned}$$

Es decir,

$$\begin{aligned} \prod_{j=1}^n e_{j1} \cdot (\mu[j]/e_{j1}) &= 0 \\ \cdot \\ \cdot \\ \cdot \\ \prod_{j=1}^n e_{jm} \cdot (\mu[j]/e_{jm}) &= 0 \end{aligned} \quad (5)$$

El objetivo de la notación (4) es ignorar todos los factores donde  $e_{ij} = 0$  ya que en este caso nunca se cumple la desigualdad  $\mu[i] < e_{ij}$  por lo que los únicos factores que interesan son los que cumplen con  $e_{ij} \neq 0$ .

Finalmente se obtiene la expresión (5) la cual es muy similar al producto clásico de una matriz por un vector. En efecto, si se tiene una matriz  $A$   $n \times m$  y un vector  $X$   $m \times 1$  las componentes del vector  $Y = AX$  están definidas por:

$$Y[1] = \sum_{j=1}^m a_{1j} X_j$$

·  
·  
·

$$Y[n] = \sum_{j=1}^m a_{nj} X_j$$

Esta última expresión difiere del sistema (5) en la sustitución del signo  $\Sigma$  por el signo  $\Pi$ , el producto  $a_{ij} X_j$  se sustituye por un producto y una división entera  $a_{ij} \cdot (X_j/a_{ij})$ . Los índices en el sistema (5) corresponden a la matriz transpuesta de  $D^-$ .

Se define el producto  $*$  de la matriz  $(D^-)^t$  de dimensiones  $m \times n$  y el vector  $\mu$  con  $n$  componentes como:

$$(D^-)^t * \mu = \begin{bmatrix} \prod_{j=1}^n e_{1j} * \mu[j] / e_{1j} \\ \prod_{j=1}^n e_{mj} * \mu[j] / e_{mj} \end{bmatrix}$$

En el producto  $*$  se utilizará la convención (4).

### Ejemplo 2.1.

Normalmente el producto matricial clásico es:

$$(D^-)^t \mu = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} = \begin{bmatrix} \mu_1 \\ 2\mu_2 + \mu_3 \\ \mu_1 + \mu_2 \\ \mu_1 + \mu_3 \end{bmatrix}$$

Por otra parte, el producto \* es:

$$(D^-)^T * \mu = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 / 2 \cdot \mu_3 \\ \mu_1 \cdot \mu_2 \\ \mu_1 \cdot \mu_3 \end{bmatrix}$$

Entonces el sistema (5) se puede escribir como:

$$(D^-)^T * \mu = 0$$

El marcado  $\mu$  está ligado a la existencia de una sucesión de disparos  $\sigma$  de la siguiente manera:

$$\mu = \mu_0 + D f(\sigma)$$

donde  $f(\sigma)$  es el vector característico de  $\sigma$ .

Finalmente se obtiene la siguiente proposición:

Una RdP marcada  $R=(P,T,D^-,D^+,\mu_0)$  se bloquea para la secuencia de disparos  $\sigma$  si

$$(D^-)^T * (\mu_0 + D f(\sigma)) = 0$$

donde  $f(\sigma)$  es el vector característico de  $\sigma$ .

(6)

Para poder contestar si una RdP se bloquea primero se debe resolver la ecuación de la proposición (6) en  $f(\sigma)$ . Si  $f(\sigma)$  es una solución, es necesario encontrar una sucesión válida de disparos de transiciones  $\sigma$  que le corresponda a  $f(\sigma)$ , ya que  $f(\sigma)$  puede tener varias sucesiones válidas de disparos asociadas o puede ser que no exista ninguna sucesión válida asociada a  $f(\sigma)$ .

Al tratar de resolver la ecuación de la proposición (6) se pueden presentar tres casos: (1) el sistema tiene solución en  $f(\sigma)$  y tiene al

menos una sucesión válida de disparos de transiciones asociada. (2) El sistema tiene solución en  $f(\sigma)$  pero no existe ninguna sucesión válida de disparos de transiciones asociada. (3) El sistema no tiene solución en  $f(\sigma)$ .

Estos casos se ilustran en los siguientes tres ejemplos :

### Ejemplo 2.2.

Caso 1) El sistema tiene solución en  $f(\sigma)$  y tiene al menos una sucesión válida de disparos de transiciones asociada. Entonces la RdP se bloqueará con la sucesión de disparos de transiciones asociada a  $f(\sigma)$ . Es conveniente aclarar que no necesariamente la RdP se bloqueará con todas las sucesiones válidas de disparos de transiciones asociadas a  $f(\sigma)$ . Para ilustrar este caso ocuparemos la red de la fig. 2.1.

Las matrices de entrada y salida son:

$$D^- = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad D^+ = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

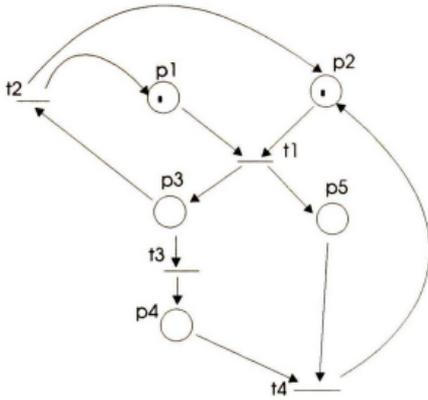


Fig. 2.1 Una RdP que no es viva.

El vector de marcado inicial y la matriz de incidencia son los siguientes:

$$\mu_o = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad D = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 1 \\ 1 & -1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & -1 \end{bmatrix}$$

Denotamos al vector característico  $f(\sigma)$  como:

$$f(\sigma) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Como se vió en el capítulo anterior, cada entrada del vector característico representa el número de veces que se dispara la transición asociada, es decir,  $x_1$  representa el número de veces que se dispara la transición  $t_1$ ,  $x_2$  representa el número de veces que se dispara la transición  $t_2$  y así para cada una de las entradas del vector.

La ecuación de la proposición 6 da lo siguiente:

$$(D^-)^T * (\mu_o + Df(\sigma)) = 0$$

Sustituyendo el vector de marcado inicial, la matriz de incidencia y el vector característico se obtiene la siguiente ecuación:

$$(D^-)^T * \left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 1 \\ 1 & -1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \right\} = 0$$

Realizando las operaciones matriciales que se encuentran entre llaves y sustituyendo la transpuesta de la matriz de entrada llegamos a la siguiente ecuación:

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 - x_1 + x_2 \\ 1 - x_1 + x_2 + x_4 \\ x_1 - x_2 - x_3 \\ x_3 - x_4 \\ x_1 - x_4 \end{bmatrix} = 0$$

Realizando el producto \* obtenemos el siguiente sistema de ecuaciones:

$$\left\{ \begin{array}{l} (1-x_1+x_2)(1-x_1-x_2+x_4)=0 \\ (x_1-x_2-x_3)=0 \\ (x_1-x_2-x_3)=0 \\ (x_3-x_4)(x_1-x_2)=0 \end{array} \right.$$

En este paso se deben resolver todos los sistemas que se forman al combinar cada uno de los factores hasta encontrar un sistema que tenga solución. El número de factores que aparecen en cada renglón está determinado por el número de entradas diferentes de cero en cada renglón de la matriz  $(D^-)^T$ . Entonces el número de factores está acotado por el número de lugares. Los sistemas que resultan al combinar todos los factores son los siguientes:

$$S1 \left\{ \begin{array}{l} (1-x_1+x_2)=0 \\ (x_1-x_2-x_3)=0 \\ (x_1-x_2-x_3)=0 \\ (x_3-x_4)=0 \end{array} \right. \quad S2 \left\{ \begin{array}{l} (1-x_1+x_2+x_4)=0 \\ (x_1-x_2-x_3)=0 \\ (x_1-x_2-x_3)=0 \\ (x_3-x_4)=0 \end{array} \right.$$

$$S3 \left\{ \begin{array}{l} (1-x_1-x_2)=0 \\ (x_1-x_2-x_3)=0 \\ (x_1-x_2-x_3)=0 \\ (x_1-x_4)=0 \end{array} \right. \quad S4 \left\{ \begin{array}{l} (1-x_1+x_2+x_4)=0 \\ (x_1-x_2-x_3)=0 \\ (x_1-x_2-x_3)=0 \\ (x_1-x_4)=0 \end{array} \right.$$

El sistema S1 resulta al tomar el primer factor del renglón 1, el primer factor del renglón 2, el primer factor del renglón 3 y el primer factor del renglón 4. Para el sistema S2 se toma el segundo factor del renglón 1 y los demás factores son los mismos que están en S1. De esta manera se obtienen S3 y S4.

Al resolver el sistema S1 encontramos que este tiene más de una solución, la solución general es la siguiente:

$$\begin{cases} x_1 = k \\ x_2 = k - 1 \\ x_3 = 1 \\ x_4 = 1 \end{cases}$$

donde k tiene que ser un entero mayor o igual a uno para que la solución sea entera no negativa esto es necesario ya que las entradas del vector característico indican el número de veces que se dispara cada transición.

Una solución particular se tiene para  $k=1$

$$f(\sigma) = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Este vector indica que la transición 1 se dispara una vez, la transición 2 no se dispara, la transición 3 se dispara una vez y la transición 4 también. Debe quedar claro que el vector no proporciona información acerca del orden en el que se deben disparar las transiciones ( esta es una desventaja del método).

Lo que resta por hacer es encontrar una secuencia válida de disparos de transiciones que cumpla con las condiciones que da el vector característico que encontramos y además que esta secuencia bloquee a la red. Encontramos que la secuencia  $\sigma = t_1 t_3 t_4$  cumple con las condiciones pedidas.

Se puede comprobar fácilmente que la RdP queda bloqueada con esta secuencia de disparos, por lo que se concluye que la red no es viva.

### Ejemplo 2.3.

Caso 2) El sistema tiene solución en  $f(\sigma)$  pero no existe ninguna sucesión válida de disparos de transiciones asociada. En este caso decimos que la RdP es viva. Para este ejemplo ocuparemos la red de la fig. 2.2.

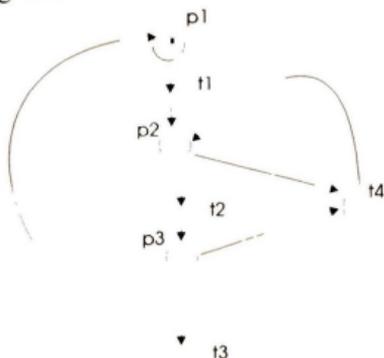


Fig. 2.2 Una RdP que es viva.

La matrices de entrada y salida son las siguientes:

$$D^- = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$D^+ = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

El vector de marcado inicial y la matriz de incidencia son los siguientes:

$$\mu_v = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$D = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & -1 \end{bmatrix}$$

Sustituyendo el vector de marcado inicial, la matriz de incidencia y el vector característico se obtiene la siguiente ecuación:

$$(D^-)^T * \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \right\} = 0$$

Realizando las operaciones matriciales que se encuentran entre llaves y sustituyendo la transpuesta de la matriz de entrada llegamos a la siguiente ecuación:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1-x_1+x_3 \\ x_1-x_2 \\ x_2-x_3-x_4 \end{bmatrix} = 0$$

Realizando el producto \* obtenemos el siguiente sistema de ecuaciones:

$$\begin{cases} 1-x_1+x_3=0 \\ x_1-x_2=0 \\ x_2-x_3-x_4=0 \\ (x_1-x_2)(x_2-x_3-x_4)=0 \end{cases}$$

Los sistemas que resultan al combinar todos los factores son los siguientes:

$$S_1 \begin{cases} 1-x_1+x_3=0 \\ x_1-x_2=0 \\ x_2-x_3-x_4=0 \\ x_1-x_2=0 \end{cases} \quad S_2 \begin{cases} 1-x_1+x_3=0 \\ x_1-x_2=0 \\ x_2-x_3-x_4=0 \\ x_2-x_3-x_4=0 \end{cases}$$

Se nota que los dos sistemas que se obtuvieron son el mismo.

Al resolver el sistema se encuentra que tiene más de una solución, la solución general es la siguiente:

$$\begin{cases} x_1 = k \\ x_2 = k \\ x_3 = k - 1 \\ x_4 = 1 \end{cases}$$

donde  $k$  es un entero mayor o igual a 1.

$$f(\sigma) = \begin{bmatrix} k \\ k \\ k - 1 \\ 1 \end{bmatrix}$$

Este vector indica que la transición 4 se debe disparar una vez, pero de la fig. 2.2 se nota que esta transición nunca se puede disparar, entonces no existe ninguna secuencia válida de disparos de transiciones que le corresponda a  $f(\sigma)$ .

Concluimos que la RdP de la fig. 2.2 es viva.

#### **Ejemplo 2.4.**

Caso 3) Al resolver la ecuación de la proposición (6) se puede dar el caso que el sistema no tiene solución en  $f(\sigma)$ . Aquí se puede concluir que la RdP es viva. Para este ejemplo vamos a utilizar la RdP de la fig. 2.3.

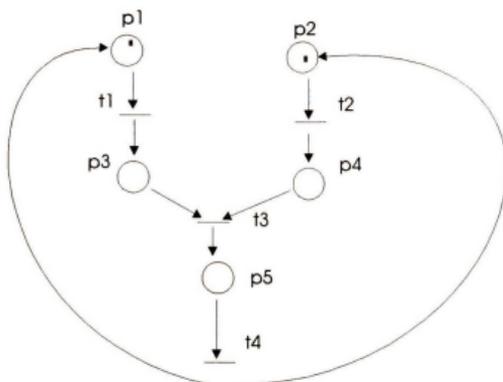


Fig. 2.3 Una RdP que es viva.

Las matrices de entrada y salida son las siguientes:

$$D^- = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad D^+ = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

El vector de marcado inicial y la matriz de incidencia son los siguientes:

$$\mu_0 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad D = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

Sustituyendo el vector de marcado inicial, la matriz de incidencia y el vector característico se obtiene la siguiente ecuación:

$$(D^-)^T * \left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \right\} = 0$$

Realizando las operaciones matriciales que se encuentran entre llaves y sustituyendo la transpuesta de la matriz de entrada llegamos a la siguiente ecuación:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1-x_1+x_4 \\ 1-x_2+x_4 \\ x_1-x_3 \\ x_2-x_3 \\ x_3-x_4 \end{bmatrix} = 0$$

Realizando el producto \* obtenemos el siguiente sistema de ecuaciones:

$$\begin{cases} 1-x_1+x_4=0 \\ 1-x_2+x_4=0 \\ (x_1-x_3)(x_2-x_3)=0 \\ x_3-x_4=0 \end{cases}$$

Al combinar los factores resultan dos sistemas que son los siguientes:

$$S_1 \begin{cases} 1 - x_1 + x_4 = 0 \\ 1 - x_2 + x_4 = 0 \\ x_1 - x_3 = 0 \\ x_3 - x_4 = 0 \end{cases} \quad S_2 \begin{cases} 1 - x_1 + x_4 = 0 \\ 1 - x_2 + x_4 = 0 \\ x_2 - x_3 = 0 \\ x_3 - x_4 = 0 \end{cases}$$

Ninguno de los sistemas anteriores tiene solución, entonces no existe un vector característico que bloquee a la red.

A continuación se muestra un ejemplo para ilustrar el método cuando la RdP tiene lugares con multiplicidad de entrada mayor que 1.

**Ejemplo 2.5.** Para este ejemplo se va a ocupar la RdP de la fig. 2.4.

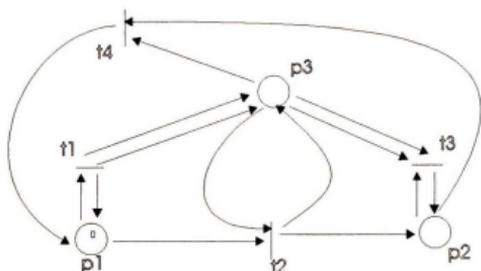


Fig. 2.4 Una RdP con lugares que tienen multiplicidad de entrada mayor que 1.

Las matrices entrada y salida son las siguientes:

$$D^- = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 2 & 1 \end{bmatrix} \quad D^+ = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 2 & 1 & 0 & 0 \end{bmatrix}$$

El vector de marcado inicial y la matriz de incidencia son los siguientes:

$$D = \begin{bmatrix} 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 2 & 0 & -2 & -1 \end{bmatrix} \quad \mu_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Sustituyendo el vector de marcado inicial, la matriz de incidencia y el vector característico en la ecuación de la proposición 6 se obtiene la siguiente ecuación:

$$(D^-)^T * \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 2 & 0 & -2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \right\} = 0$$

Realizando las operaciones matriciales que se encuentran entre llaves y sustituyendo la transpuesta de la matriz de entrada llegamos a la siguiente ecuación:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 - x_2 + x_4 \\ x_2 - x_4 \\ 2x_1 - 2x_3 - x_4 \end{bmatrix} = 0$$

Realizando el producto \* obtenemos el siguiente sistema de ecuaciones:

$$\begin{cases} 1 - x_2 + x_4 = 0 \\ (1 - x_2 + x_4)(2x_1 - 2x_3 - x_4) = 0 \\ (x_2 - x_4)(2x_1 - 2x_3 - x_4) / 2 = 0 \\ (x_2 - x_4)(2x_1 - 2x_3 - x_4) = 0 \end{cases}$$

Al combinar los factores resultan los siguientes sistemas:

$$S_1 \begin{cases} 1-x_2+x_4=0 \\ 1-x_2+x_4=0 \\ x_2-x_4=0 \\ x_2-x_4=0 \end{cases}$$

$$S_2 \begin{cases} 1-x_2+x_4=0 \\ 2x_1-2x_3-x_4=0 \\ x_2-x_4=0 \\ x_2-x_4=0 \end{cases}$$

$$S_3 \begin{cases} 1-x_2+x_4=0 \\ 1-x_2+x_4=0 \\ (2x_1-2x_3-x_4)/2=0 \\ x_2-x_4=0 \end{cases}$$

$$S_4 \begin{cases} 1-x_2+x_4=0 \\ 2x_1-2x_3-x_4=0 \\ (2x_1-2x_3-x_4)/2=0 \\ x_2-x_4=0 \end{cases}$$

$$S_5 \begin{cases} 1-x_2+x_4=0 \\ 1-x_2+x_4=0 \\ x_2-x_4=0 \\ 2x_1-2x_3-x_4=0 \end{cases}$$

$$S_6 \begin{cases} 1-x_2+x_4=0 \\ 2x_1-2x_3-x_4=0 \\ x_2-x_4=0 \\ 2x_1-2x_3-x_4=0 \end{cases}$$

$$S_7 \begin{cases} 1-x_2+x_4=0 \\ 1-x_2+x_4=0 \\ (2x_1-2x_3-x_4)/2=0 \\ 2x_1-2x_3-x_4=0 \end{cases}$$

$$S_8 \begin{cases} 1-x_2+x_4=0 \\ 2x_1-2x_3-x_4=0 \\ (2x_1-2x_3-x_4)/2=0 \\ 2x_1-2x_3-x_4=0 \end{cases}$$

La ecuación  $(2x_1-2x_3-x_4)/2=0$  se satisface si y sólo si  $2x_1-2x_3-x_4=0$  ó  $2x_1-2x_3-x_4=1$ . Entonces el sistema  $S_3$  es equivalente a los dos sistemas siguientes:

$$S_{31} \begin{cases} 1 - x_2 + x_4 = 0 \\ 1 - x_2 + x_4 = 0 \\ 2x_1 - 2x_3 - x_4 = 0 \\ x_2 - x_4 = 0 \end{cases}$$

$$S_{32} \begin{cases} 1 - x_2 + x_4 = 0 \\ 1 - x_2 + x_4 = 0 \\ 2x_1 - 2x_3 - x_4 = 1 \\ x_2 - x_4 = 0 \end{cases}$$

El sistema  $S_4$  es equivalente a:

$$S_{41} \begin{cases} 1 - x_2 + x_4 = 0 \\ 2x_1 - 2x_3 - x_4 = 0 \\ 2x_1 - 2x_3 - x_4 = 0 \\ x_2 - x_4 = 0 \end{cases}$$

$$S_{42} \begin{cases} 1 - x_2 + x_4 = 0 \\ 2x_1 - 2x_3 - x_4 = 0 \\ 2x_1 - 2x_3 - x_4 = 1 \\ x_2 - x_4 = 0 \end{cases}$$

El sistema  $S_7$  es equivalente a:

$$S_{71} \begin{cases} 1 - x_2 + x_4 = 0 \\ 1 - x_2 + x_4 = 0 \\ 2x_1 - 2x_3 - x_4 = 0 \\ 2x_1 - 2x_3 - x_4 = 0 \end{cases}$$

$$S_{72} \begin{cases} 1 - x_2 + x_4 = 0 \\ 1 - x_2 + x_4 = 0 \\ 2x_1 - 2x_3 - x_4 = 1 \\ 2x_1 - 2x_3 - x_4 = 0 \end{cases}$$

El sistema  $S_8$  es equivalente a:

$$S_{81} \begin{cases} 1 - x_2 + x_4 = 0 \\ 2x_1 - 2x_3 - x_4 = 0 \\ 2x_1 - 2x_3 - x_4 = 0 \\ 2x_1 - 2x_3 - x_4 = 0 \end{cases}$$

$$S_{82} \begin{cases} 1 - x_2 + x_4 = 0 \\ 2x_1 - 2x_3 - x_4 = 0 \\ 2x_1 - 2x_3 - x_4 = 1 \\ 2x_1 - 2x_3 - x_4 = 0 \end{cases}$$

Finalmente, el conjunto de sistemas que tenemos que resolver es  $\{S_1, S_2, S_{31}, S_{32}, S_{41}, S_{42}, S_5, S_6, S_{71}, S_{72}, S_{81}, S_{82}\}$ . Una solución particular al sistema  $S_{71}$  es:

$$f(\sigma) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Este vector tiene asociada la secuencia válida  $\sigma = t_1 t_2 t_3$ , se puede comprobar fácilmente que esta secuencia bloquea a la red, con lo que se concluye que la red no es viva.

## CAPITULO 3

### EL SISTEMA SSRdP PARA SIMULAR RdP

#### 3.1 Introducción

Como se mencionó en los capítulos anteriores, las RdP son una herramienta poderosa para la modelación y el análisis de sistemas dinámicos con eventos discretos. Aplicar esta herramienta en la modelación de sistemas implica los siguientes pasos :

1) Construir una tabla de eventos y condiciones (precondiciones y postcondiciones) que ocurren en el sistema que se está modelando.

2) Representar gráficamente la RdP En la edición de las RdP tenemos las siguientes relaciones :

-Una condición, modelada por un *lugar*, se representa por un *círculo*.

-Un evento, modelado por una *transición*, se representa por una *barra*.

-Las marcas se representan por *puntos* dentro de los *lugares*.

-Las relaciones de entrada y salida se representan por *arcos dirigidos* que conectan los *lugares* con las *transiciones*.

3) Una vez que se ha creado una RdP existen dos alternativas : la primera es ejecutar la RdP y la segunda es analizarla. Ejecutar una RdP se puede describir de la siguiente manera :

i) Buscar transiciones habilitadas, si no existen termina la ejecución. Para saber si una transición está habilitada o no, se

necesita comparar el número de marcas que tiene cada lugar de entrada con su multiplicidad.

ii) De todas las transiciones habilitadas, elegir una para dispararla.

iii) Disparar la transición que se eligió. Esto consiste en remover las marcas activantes y producir nuevas marcas las cuales se van a propagar hacia los lugares de salida. Lo que se tiene que hacer en la gráfica es dibujar el nuevo estado de la red. Recuerde que el estado de la red está determinado por la distribución de las marcas en los lugares.

iv) Volver al inciso i.

Como se puede ver, la ejecución de una RdP involucra actividades hasta cierto punto repetitivas. Por ejemplo el saber si una transición está habilitada o no, se reduce a comparar enteros no negativos, procedimiento que se puede convertir en algo complejo si el número de transiciones y de lugares de entrada de cada transición son muy grandes. Al disparar una transición se requiere restar y sumar enteros no negativos. Estas operaciones son muy sencillas, sin embargo pueden producir errores si el número de lugares de entrada de la transición que se dispara y el número de marcas asignadas a estos lugares son demasiado grandes. Esto es una gran desventaja ya que puede provocar que se obtenga una información errónea del modelo del sistema bajo estudio.

Otra desventaja que existe en el manejo de una RdP es el tiempo que se tiene que invertir para realizar las actividades descritas anteriormente.

Por otra parte, analizar una RdP significa determinar las propiedades que tiene una RdP, para realizar esto se pueden aplicar los métodos descritos en los capítulos anteriores.

Mediante un sistema de cómputo es posible implementar los métodos de análisis de las RdP, así como superar las desventajas mencionadas anteriormente. El sistema de simulación de RdP (SSRdP) es un sistema para computadoras personales IBM-PC compatibles. Dicho sistema crea, ejecuta y analiza RdP, esto ayuda a realizar las actividades mencionadas anteriormente de manera que el uso de RdP se vuelve más atractivo y menos monótono. Este sistema tiene las siguientes características :

1) Cuenta con un editor gráfico lo que permite crear y editar una RdP en forma amigable.

2) Permite ejecutar una RdP y ver gráficamente los diferentes estados que alcanza el sistema. Existen dos modos de ejecución : de todas las transiciones habilitadas,

- i) El sistema elige de manera aleatoria la transición habilitada que se va a disparar.
- ii) El usuario elige la transición habilitada que se va a disparar.

3) Permite analizar una RdP, aquí podemos determinar si una RdP tiene las dos propiedades siguientes :

- i) Alcanzabilidad.
- ii) Vivacidad.

Para determinar la alcanzabilidad se implementó el método matricial explicado en el capítulo 1. Una vez creada una RdP con su marcado inicial, el usuario puede preguntar si algún otro marcado es alcanzable a partir del marcado inicial, en caso afirmativo el sistema responde proporcionando el vector de disparo con el cual el marcado dado es alcanzable.

Para determinar la vivacidad, se implementó el método matricial explicado en el capítulo 2. Una vez creada una RdP con su marcado inicial, el usuario puede preguntar si la red se bloquea. Si este es el caso el sistema proporciona el vector de disparo causante del bloqueo.

En la programación del sistema se utilizó el estilo de la programación orientada a objetos debido a que nos permite la elaboración de programas claros, fáciles de entender y de mantener. Se utilizó el lenguaje orientado a objetos C++ de Borland versión 3.1.

### **3.2 Ejecutando SSRdP**

Antes de ejecutar SSRdP se debe tener instalado el manejador del ratón (Mouse) y declarar una variable del medio ambiente del sistema operativo (SO). Esto último se lleva a cabo tecleando en el indicador del SO *set bgipath=Trayectoria* seguido de un retorno de carro (tecla *Enter*), donde *Trayectoria* es el directorio donde se encuentra el archivo EGAVGA.BGI. Después de esto ya se puede ejecutar SSRdP simplemente tecleando SSRdP seguido de un retorno de carro en el indicador del SO. Cuando empieza la ejecución de SSRdP aparece la pantalla mostrada en la fig. 3.1 Esta es la pantalla del menú principal y consiste de tres partes, de arriba hacia abajo :

- 1) El menú principal.
- 2) La ventana de edición.
- 3) La ventana de mensajes del sistema.

El sistema empieza mostrando el indicador del mouse en la ventana de edición, lo que quiere decir que está listo para empezar a crear o editar una RdP. Para pasar al menú principal se presiona el botón izquierdo del ratón.

### **3.3 El menú principal**

El menú principal se utiliza para ordenar a SSRdP ejecutar un proceso, como editar, ejecutar o analizar una RdP. Hay dos maneras de hacer una selección en el menú principal :

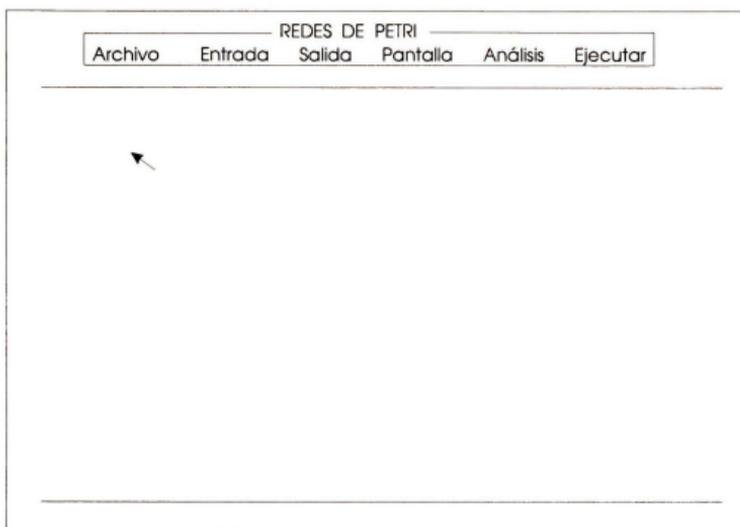


Fig. 3.1. El menú principal de SSRdP.

- 1) Usando las teclas de flechas para seleccionar la opción deseada y después un retorno de carro.
- 2) Pulsando la primer letra de la opción que desee. Por ejemplo para seleccionar *Ejecutar* tiene que pulsar *E*.

Todas las opciones del menú principal tienen su propio submenú, que se presenta como un menú debajo del menú principal (pull-down). Por ejemplo si selecciona la opción *Archivo* se activa el submenú *Archivo* como se muestra la fig. 3.2.

El número de submenús varía de acuerdo con las necesidades de cada opción en el menú principal. Para pasar del menú principal al editor gráfico es necesario pulsar la tecla *Escape*. En seguida se explican cada una de las opciones del menú principal y sus subopciones.

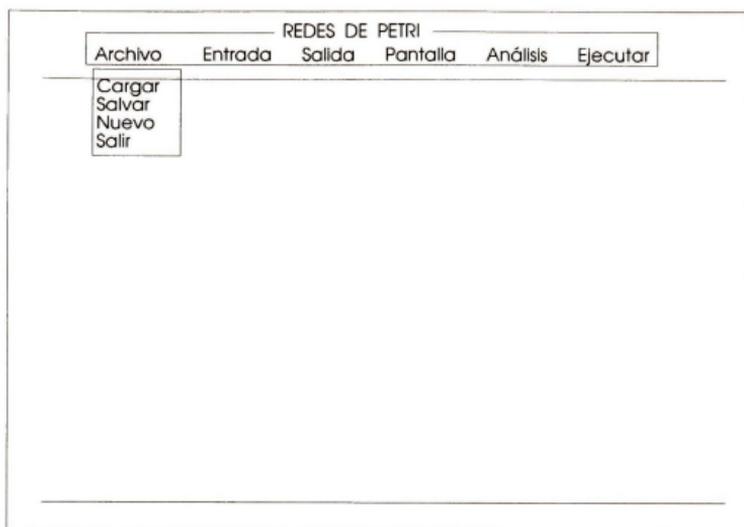


Fig. 3.2 La ventana pull-down Archivo.

### 3.3.1 La opción Archivo

La opción *Archivo* tiene cuatro subopciones (ver fig. 3.2). Estas son :

- i) Cargar
- ii) Salvar
- iii) Nuevo
- iv) Salir

La opción *Cargar* pide al usuario el nombre de un archivo y después carga el archivo en el editor gráfico. La opción *Salvar* registra la RdP cargada actualmente en el editor, a un archivo. La opción *Nuevo* borra la RdP que cargada en el editor. Finalmente, la opción *Salir* sirve para abandonar el sistema.

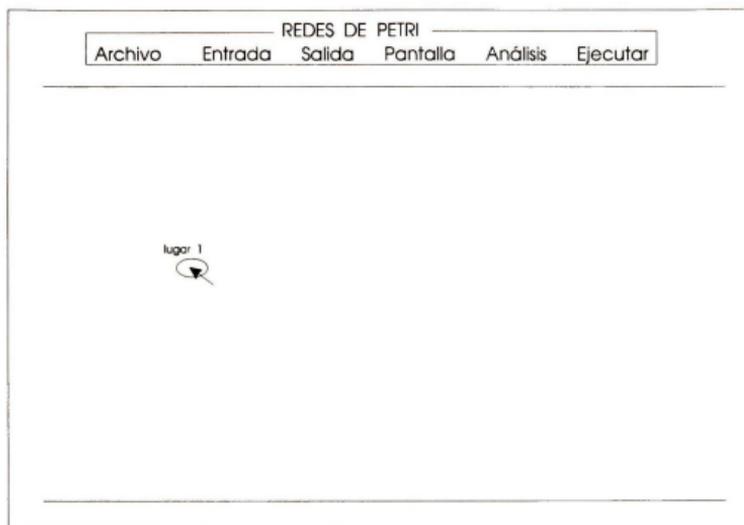


Fig. 3.3 La pantalla cuando se crea un lugar.

### 3.3.2 La opción *Entrada*

Esta opción tiene las siguientes subopciones

- i) Crea
- ii) Relaciona
- iii) Arco
- iv) Borra
- v) Marcar

La opción *Crea* permite definir un *lugar* (*círculo*) en la posición actual del cursor del ratón. La opción *Relaciona* permite indicar que un *lugar* (definido con la opción *Crea*) es un lugar de entrada de alguna *transición*, en esta opción el sistema inicia preguntando al usuario el *lugar* y la *transición* que va a relacionar, después pregunta la multiplicidad de entrada. Una vez que tiene esta información, el sistema traza una recta con flecha del *lugar* a la *transición*. No se dibujan arcos múltiples para indicar una multiplicidad mayor que 1, si

es este el caso, en la mitad del arco se escribe la multiplicidad. Cuando se relaciona un *lugar* con una *transición* como lugar de entrada, el arco que traza el sistema siempre es una recta con flecha del *lugar* a la *transición*. Por motivos de estética o visibilidad algunas veces es conveniente poder "estirar" esta recta para formar un arco, la opción *Arco* permite hacer esto, el sistema pide al usuario un punto por donde desee que pase el arco. Se requieren entonces tres puntos para formar un arco: el *lugar*, el punto dado por el usuario y la *transición*. La opción *Borra* permite eliminar una relación de entrada previamente creada. Finalmente la opción *Marcar* permite asignar *marcas* (tokens) a los *lugares*. Con ésto el usuario puede marcar la RdP que esté cargada en el editor. A diferencia de la forma habitual, no se escriben puntos dentro de los *lugares*, simplemente se escribe el número de *marcas* que tiene asignado cada *lugar* a un lado del *círculo*.

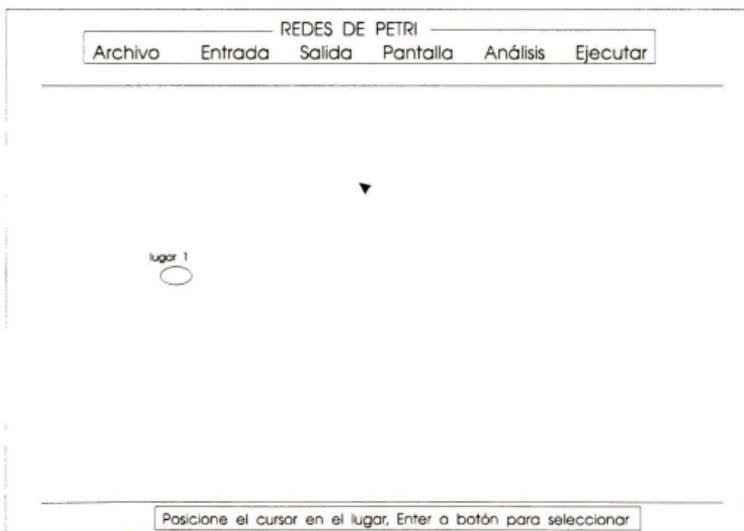


Fig. 3.4 La pantalla cuando *SSRdP* pide que seleccione un lugar.

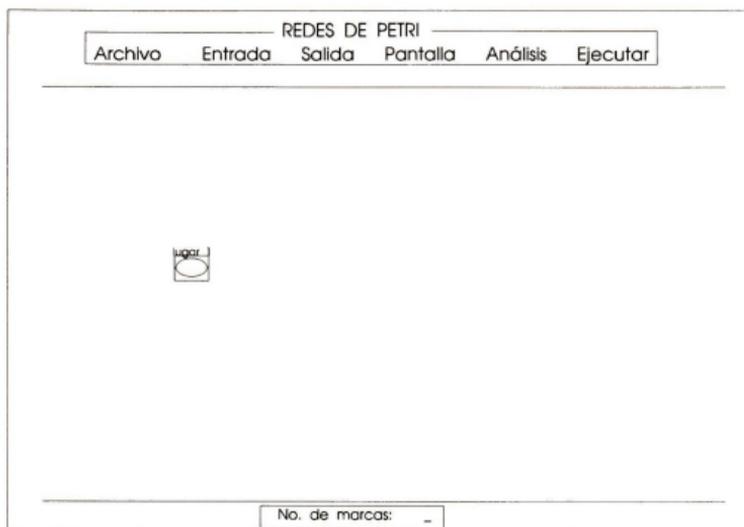


Fig. 3.5 La pantalla cuando *SSRdP* pide el número de marcas.

### ***Ejemplo 3.1***

En este ejemplo se ilustra como podemos crear un *lugar* y asignarle dos marcas. Los pasos a seguir son los siguientes :

#### **1) Crear un *lugar***

1.1) Llevar el cursor del ratón a la posición donde se desea aparezca el *lugar*.

1.2) Se presiona el botón izquierdo del ratón para ir al menú principal.

1.3) Seleccionar la opción *Entrada* y después la opción *Crea*. Al entrar a esta opción el sistema pregunta por el identificador para el *lugar* que se va a crear. Suponga que el usuario teclea "lugar 1" como identificador.

1.4) El sistema responde dibujando un *círculo* en la posición donde se encuentra el cursor del ratón y escribe el

identificador arriba del *círculo*. La pantalla aparece como se muestra en la fig. 3.3.

## 2) Asignar *marcas*.

2.1) Elegir la opción *Entrada* y después la subopción *Marcar*. El sistema pide al usuario que indique el *lugar* que se va a marcar como se muestra en la fig. 3.4.

2.2) Para seleccionar el *lugar* se lleva el cursor del ratón dentro del *círculo* que corresponda. Lo seleccionamos presionando la tecla retorno de carro o el botón izquierdo del ratón.

2.3) El sistema pide el número de *marcas* que se van a asignar, como se muestra en la fig. 3.5.

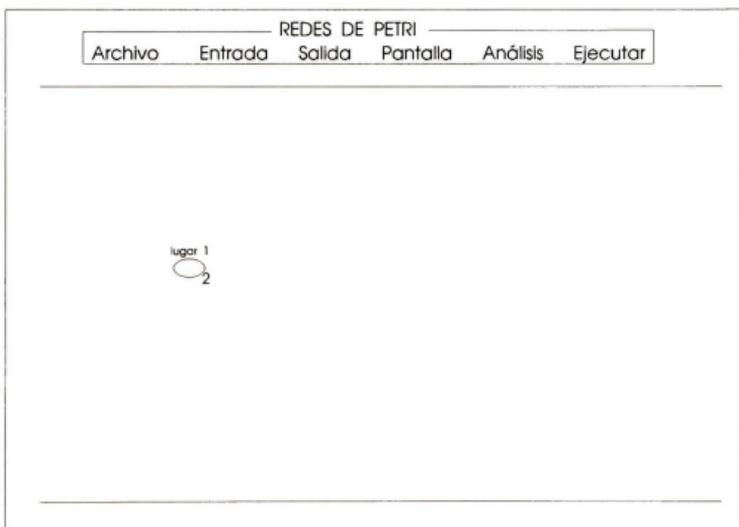


Fig. 3.6 La pantalla después de que se marca un lugar.

Después de que el usuario teclea el número de marcas, la pantalla aparece como se muestra en la fig. 3.6. Esta figura contiene el estado actual del editor después de que un lugar con 2 marcas ha sido creado.

### 3.3.3 La opción Salida

Esta opción tiene las siguientes subopciones :

- i) Crea
- ii) Relaciona
- iii) Arco
- iv) Borra

La opción *Crea* permite crear una *transición* (dibujar una *barra*) en la posición actual del cursor del ratón. Las opciones *Relaciona*, *Arco* y *Borra* son similares a las así tituladas en la opción *Entrada* con la diferencia de que éstas son para lugares de salida.

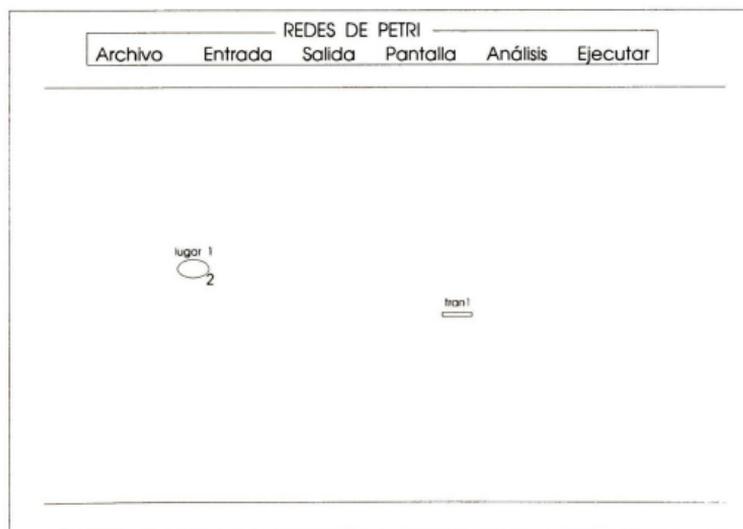


Fig. 3.7. La pantalla después de crear una transición.

### **Ejemplo 3.2.**

En este ejemplo se ilustra como crear una transición y relacionarla con un lugar de entrada. Se va a usar el lugar que se creó en el ejemplo 3.1. Los pasos a seguir son los siguientes :

#### **1) Crear un *transición*.**

1.1) Llevar el cursor del ratón a la posición donde se desea aparezca la *transición*..

1.2) Se presiona el botón izquierdo del ratón para ir al menú principal.

1.3) Seleccionar la opción *Salida* y después la opción *Crea*. Al entrar a esta opción el sistema pregunta por el identificador para la *transición* que se va a crear. Suponga que el usuario teclea "tran 1" como identificador.

1.4) El sistema responde dibujando un *barra* en la posición donde se encuentra el cursor del ratón y escribe el identificador arriba de la *barra*. La pantalla aparece como se muestra en la fig. 3.7.

#### **2) Relacionar la *transición* con el lugar de entrada.**

2.1) Elegir la opción *Entrada* y después la subopción *Relaciona*.

2.2) *SSRdP* pide al usuario indique el *lugar* y la *transición* que desea relacionar. Inmediatamente después pide la *multiplicidad* de entrada, como se muestra en la fig. 3.8. Suponga que el usuario teclea 1.

2.3) *SSRdP* responde dibujando un *arco dirigido* del *lugar* a la *transición*. La fig. 3.9 contiene el estado actual del editor después de que se ha establecido una relación de entrada entre la *transición tran1* y el *lugar, lugar1*.

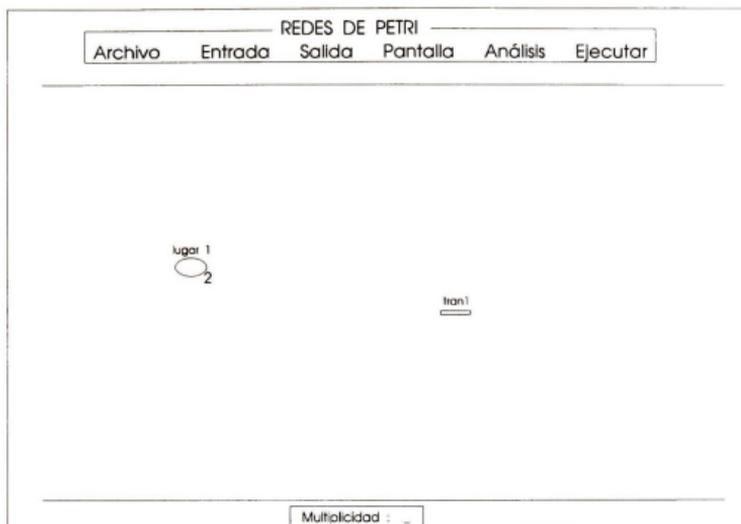


Fig. 3.8. La pantalla cuando *SSrDP* pide la multiplicidad.

### 3.3.4 La opción *Pantalla*

Esta opción tiene tres subopciones

- i) Mover Nodo
- ii) Borrar Nodo
- iii) Zoom

Un nodo en *SSrDP* es o bien un lugar, o una transición. Las primeras dos subopciones sirven para mover o borrar un nodo respectivamente. Con la opción *Zoom* podemos ver una pantalla completa del editor o sólo una parte de ésta.

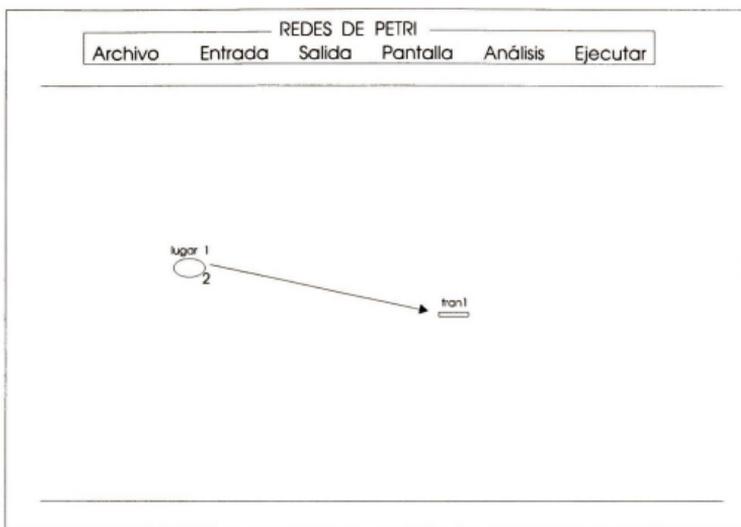


Fig. 3.9. La pantalla después de establecer una relación de entrada.

### 3.3.5 La opción Ejecutar

La opción *Ejecutar* sirve para simular la ejecución de la RdP cargada actualmente en el editor. Esta tiene dos subopciones :

- i) Manual
- ii) Aleatoria

La primera opción, muestra un menú que contiene todas las transiciones habilitadas y el usuario elige la transición que se dispara, mientras que en la segunda opción el sistema elige de manera aleatoria la transición que se va a disparar.

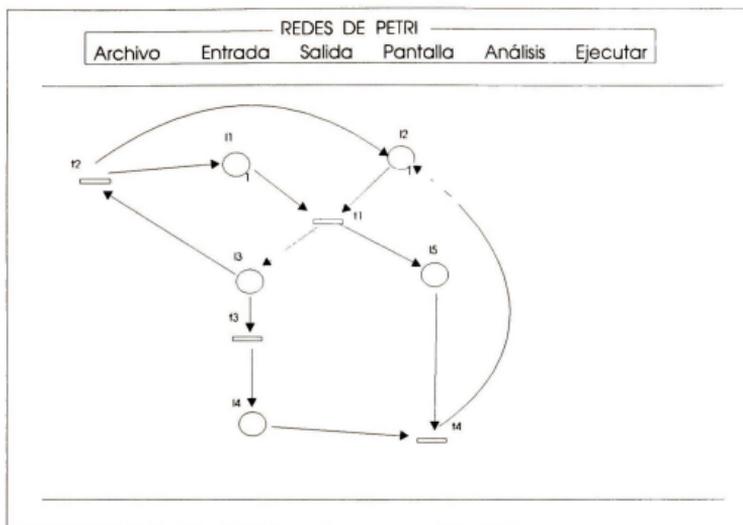


Fig. 3.10. Una RdP para ilustrar la ejecución.

### Ejemplo 3.3.

Este ejemplo muestra como ejecutar en forma manual una RdP que esté actualmente cargada en el editor. Suponga que *SSRdP* tiene cargada la RdP que se muestra en la fig. 3.10. Como se puede ver el marcado de la RdP es  $\mu = [1 \ 1 \ 0 \ 0 \ 0]^t$ . Para llevar a cabo la ejecución se siguen los siguiente pasos :

1) Seleccionar la opción *Ejecutar*.

2) Seleccionar la opción *Manual*. Enseguida *SSRdP* prepara la RdP para la ejecución y espera que el usuario presione el botón izquierdo del ratón para mostrar un menú que contiene todas las *transiciones* habilitadas, ver fig. 3.11

3) *SSRdP* dispara la *transición* seleccionada por el usuario. La fig. 3.12 muestra la RdP después de disparar la *transición t1*, donde el

nuevo marcado es  $\mu = [0 \ 0 \ 1 \ 0 \ 1]$ . Este proceso puede continuar hasta que ya no existan transiciones habilitadas o que el usuario presione la tecla *Escape* cuando se presenta el menú de transiciones habilitadas.

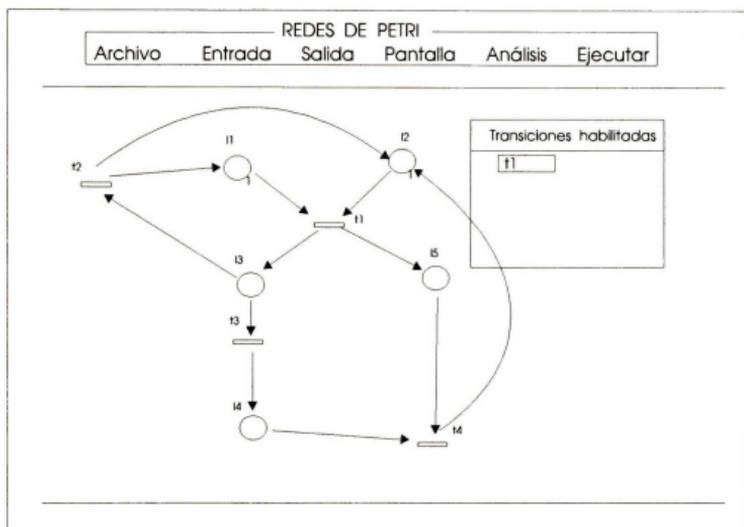


Fig. 3.11. La pantalla que muestra el menú con las transiciones habilitadas.

### 3.3.6 La opción Analizar

Esta opción permite analizar la RdP que este cargada en el editor. Esta tiene dos subopciones :

- i) Alcanzabilidad
- ii) Vivacidad

Con la primera opción se puede determinar si un marcado dado es *alcanzable* a partir del marcado que tenga la RdP. La segunda opción permite saber si la RdP cargada en el editor es *viva*.

### ***Ejemplo 3.4***

Este ejemplo muestra como determinar si un mercado dado es alcanzable a partir del mercado que tenga la RdP actualmente cargada en el editor. Suponga que *SSRdP* tiene cargada la RdP que se muestra en la fig. 3.10. Se siguen los siguientes pasos :

1) Seleccionar la opción *Análisis* .

2) Seleccionar la opción *Alcanzabilidad*. Posteriormente *SSRdP* pregunta por el nuevo mercado, ver fig. 3.13. Suponga que el usuario teclea el vector  $\mu' = [0 \ 0 \ 0 \ 1 \ 1]^t$ .

3) *SSRdP* responde con el vector de disparo para alcanzar a  $\mu'$ , como se muestra en la fig. 3.14.

### ***Ejemplo 3.5***

La fig. 3.15 muestra como responde *SSRdP* cuando se le ordena que analice la vivacidad en la RdP de la fig. 3.10.

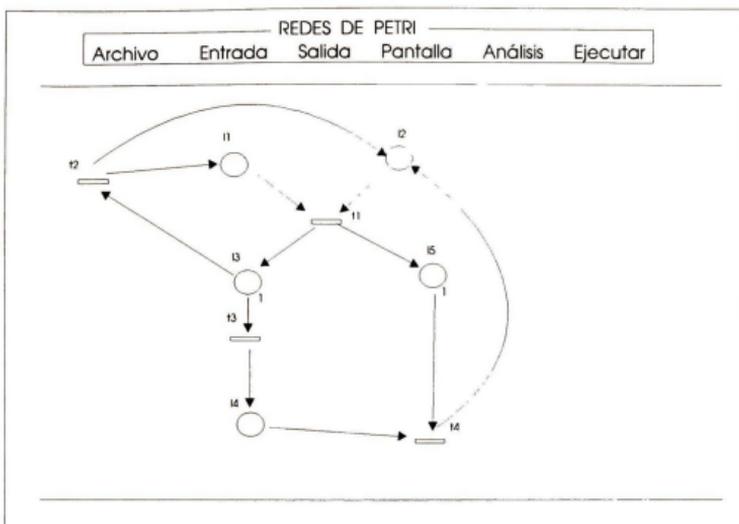


Fig. 3.12. La pantalla después de disparar la RdP de la fig. 3.10.

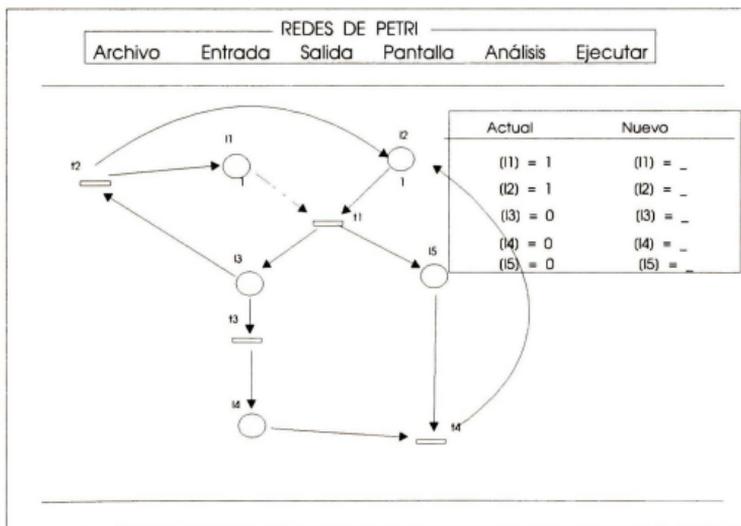


Fig. 3.13. La pantalla cuando SSRdP pregunta por un nuevo marcado.

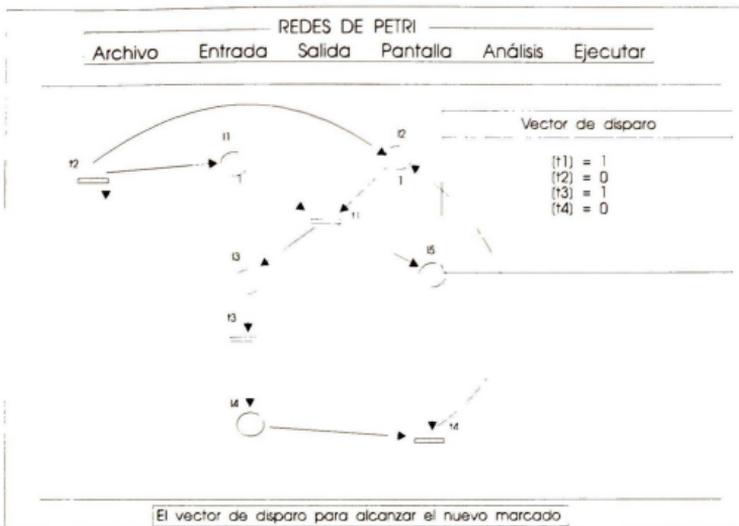


Fig. 3.14. La pantalla cuando SSRdP proporciona el vector de disparo para alcanzar a  $\mu'$

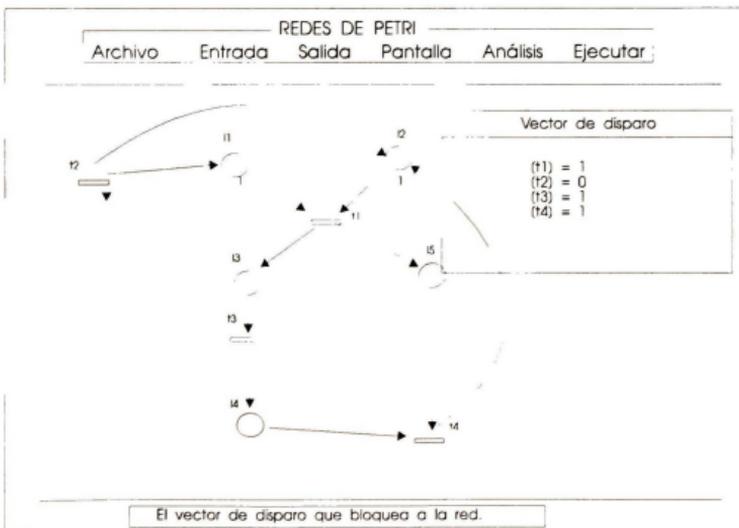


Fig. 3.15. La pantalla cuando SSRdP proporciona el vector de disparo que bloquea a la red.

## CAPITULO 4

### APLICACIONES

#### 4.1 Introducción

En este capítulo se presentan algunos ejemplos de los tipos de sistemas que pueden ser simulados con *SSRdP*. Si bien estos ejemplos han sido extraídos de la literatura existente, su presentación aquí es con el propósito de ilustrar la forma como *SSRdP* los resuelve.

#### 4.2 El problema del productor-consumidor

El problema del productor-consumidor involucra un *buffer* compartido de datos. Se tienen dos procesos : el proceso *productor* y el procesor *consumidor*. El proceso *productor* crea objetos los cuales son almacenados en un *buffer*, el *consumidor* espera hasta que se deposite un objeto en el *buffer*, para entonces removerlo y consumirlo. Esto se puede modelar por la RdP de la fig. 4.1. El lugar *p3* representa el *buffer*; cada marca representa un objeto que se ha producido pero todavía no ha sido consumido. Las actividades del *productor* son modeladas por los lugares *p1* y *p2*, y las transiciones *t1* y *t2* de la siguiente manera :

- 1) La presencia de una marca en *p1* representa al *productor* listo para producir.
- 2) La ocurrencia de la transición *t1* representa la producción de un objeto.
- 3) La presencia de una marca en *p2* indica que se ha producido un objeto.
- 4) La ocurrencia de la transición *t2* representa el almacenamiento en el *buffer* del objeto producido.

De manera análoga las actividades del *consumidor* se modelan por los lugares  $p4$  y  $p5$ , y las transiciones  $t3$  y  $t4$  :

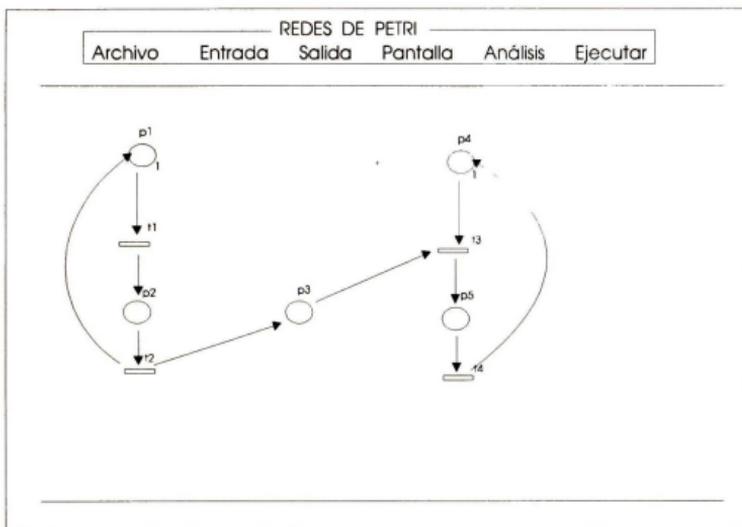


Fig. 4.1. El problema del productor-consumidor modelado con SSRdP.

- 1) La presencia de una marca en el lugar  $p4$  indica que el *consumidor* está listo para consumir.
- 2) La ocurrencia de la transición  $t3$  remueve un objeto del *buffer* (lugar  $p3$ ).
- 3) La presencia de una marca en el lugar  $p5$  representa que se ha removido un objeto del *buffer*.
- 4) Cuando ocurre la transición  $t4$  se consume el objeto removido.

La fig. 4.2 muestra el estado de *SSRdP* después de que se ha ejecutado en forma aleatoria. El marcado que se tiene es :  $[1,0, 10,1,0]^t$  . La interpretación de ésto es :

El *productor* y el *consumidor* están listos para realizar sus actividades respectivas, se han producido 10 objetos y se encuentran depositados en el *buffer*.

Una pregunta interesante es saber si un marcado determinado es alcanzable a partir del marcado inicial,  $[1, 0, 0, 1, 0]^t$ . Por ejemplo, se podría preguntar a *SSRdP* si se pueden tener 2 objetos depositados en el *buffer*. Esto se logra preguntando si el marcado  $[1, 0, 2, 1, 0]^t$  es alcanzable a partir del marcado inicial.

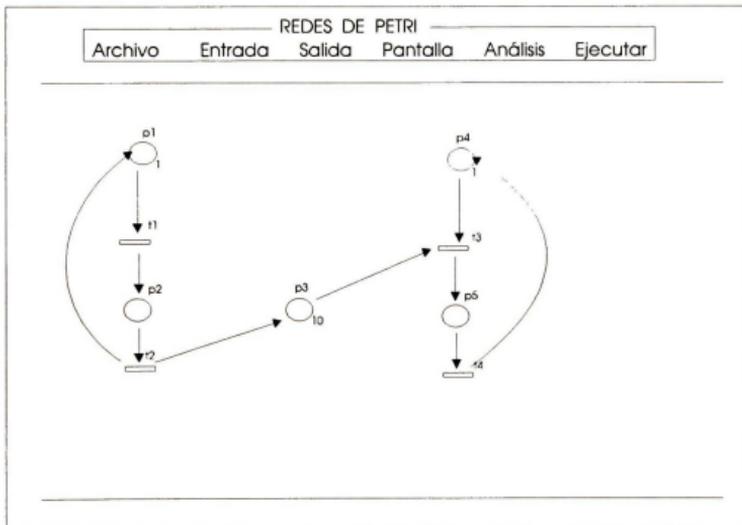


Fig. 4.2. *SSRdP* después de ejecutarse en forma aleatoria.

*SSRdP* responde a esta pregunta proporcionando el vector de disparo  $[2, 2, 0, 0]$ , como se muestra en la fig. 4.3. Este vector de disparo indica que para poder alcanzar el marcado  $[1, 0, 2, 1, 0]^t$ , las transiciones *t1* y *t2* se disparan 2 veces y las transiciones *t3* y *t4* no se disparan .

Para terminar con este ejemplo, la fig. 4.4 contiene la respuesta al análisis de la vivacidad de la RdP.

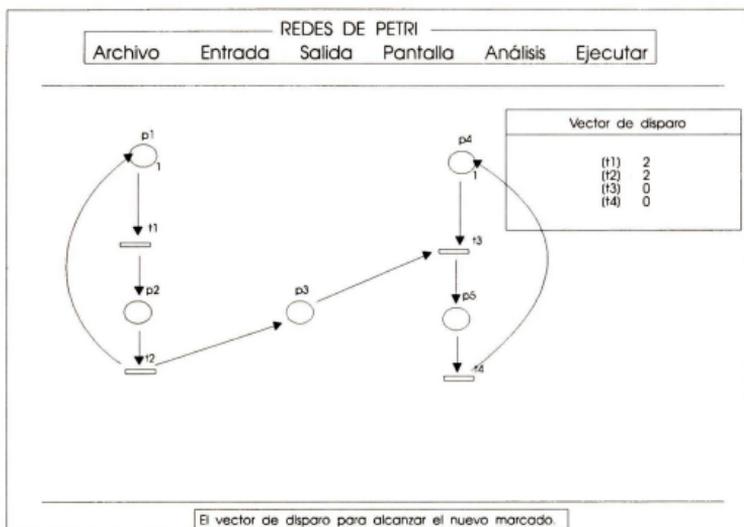


Fig. 4.3. La alcanzabilidad en el problema del productor-consumidor.

### 4.3 El problema de los filósofos comensales

Este problema es un ejemplo clásico de modelado de sistemas con RdP :

Cinco filósofos se encuentran sentados en una mesa circular, en donde hay cinco platos de *spaguetti* y tenedores colocados uno entre cada filósofo. Cada filósofo puede permanecer ya sea meditando o comiendo. Para que un filósofo pase del estado de *meditación* al estado de *alimentación* necesita tener disponibles los dos tenedores que comparte con sus vecinos.

El problema esencial consiste en que si todos los filósofos toman su tenedor izquierdo y después intentan tomar el tenedor derecho, tendrán que esperar para siempre y permanecer hambrientos (condición de bloqueo). Es necesario entonces obtener un modelo que coordine las "actividades" de los filósofos de tal forma que se evite la situación de bloqueo.

La fig. 4.5 ilustra la RdP en *SSRdP* que soluciona este problema. Los lugares *ten1*, ..., *ten5* representan los tenedores y como inicialmente están libres, contienen una marca. Cada filósofo se representa por 2 lugares  $m_i$  y  $c_i$  que representan los estados *meditando* y *comiendo* respectivamente. La presencia de una marca en uno de estos lugares representa la ocurrencia del estado correspondiente.

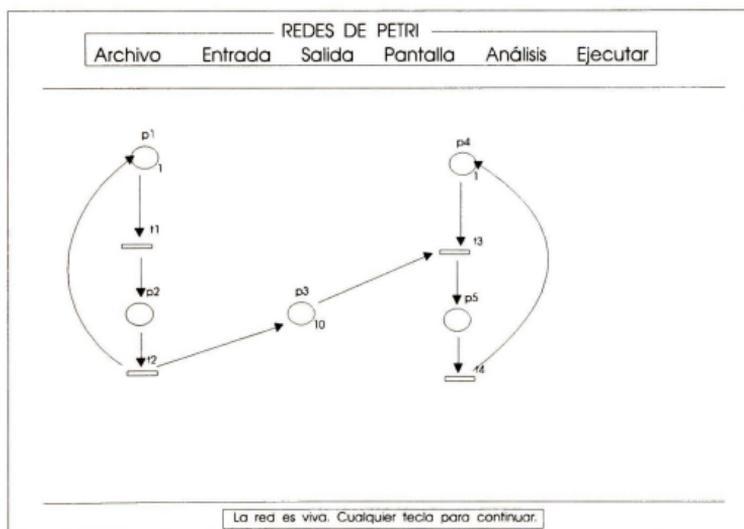


Fig. 4.4. El análisis de la vivacidad del productor-consumidor.

Una vez que se tiene el modelo del sistema bajo estudio (fig. 4.5), el siguiente paso es analizar el modelo, es decir, determinar si el

modelo es o no una representación apropiada del sistema; un enfoque para lograr esto es verificar que el modelo cumpla las restricciones que impone la descripción del problema. Para este ejemplo, una de las restricciones más importantes es que 2 filósofos vecinos no pueden estar comiendo a un mismo tiempo. Para probar que esta restricción es respetada por el modelo, se puede cuestionar si en algún momento el filósofo 1 y el filósofo 2, o el filósofo 2 y el filósofo 3, o el filósofo 3 y el filósofo 4, etc. pueden estar comiendo a un mismo tiempo, para el primer caso, esto se reduce a preguntar a *SSRdP* si el marcado :

$$\mu [m1,c1,ten1,m2,c2,ten2,m3,c3,ten3,m4,c4,ten4,m5,c5,ten5]=[0,1,0,0,1,0,1,0,1,1,0,1,1,0,0]$$

es alcanzable a partir del marcado inicial :

$$\mu [m1,c1,ten1,m2,c2,ten2,m3,c3,ten3,m4,c4,ten4,m5,c5,ten5]=[1,0,1,1,0,1,1,0,1,1,0,1,1,0,1]$$

*SSRdP* responde que el marcado no es alcanzable, ver fig. 4.6.

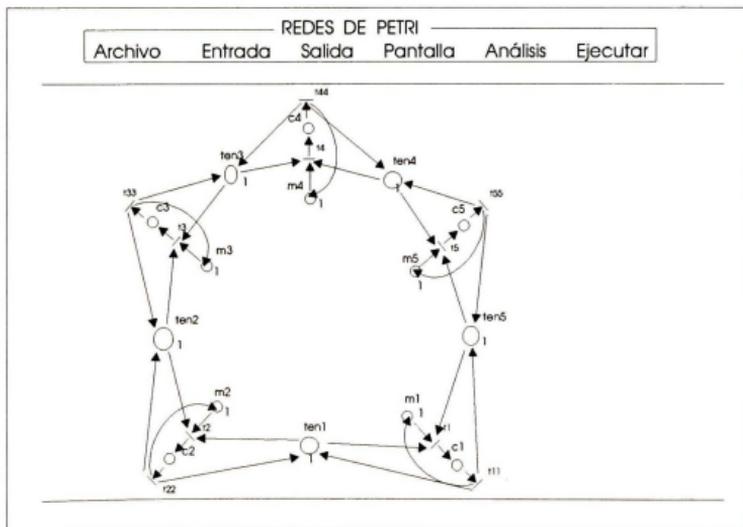


Fig. 4.5. El problema de los filósofos comensales.

La contraparte de lo anterior es preguntar a *SSRdP* si dos filósofos no vecinos pueden estar comiendo al mismo tiempo. El interés de esta pregunta es determinar si dos filósofos pueden realizar la misma actividad (*comiendo*) concurrentemente, con lo que se probaría que un filósofo puede empezar a comer sin necesidad de esperar a que no haya ninguno comiendo. Por ejemplo, saber si el *filósofo 1* y el *filósofo 3* pueden comer al mismo tiempo consiste en preguntar si el marcado :

$$\mu [m_1, c_1, ten_1, m_2, c_2, ten_2, m_3, c_3, ten_3, m_4, c_4, ten_4, m_5, c_5, ten_5] = [0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0]$$

es alcanzable a partir del marcado inicial.

Como se ve en la fig. 4.7, *SSRdP* responde que efectivamente el marcado se puede alcanzar y proporciona el vector de disparo.

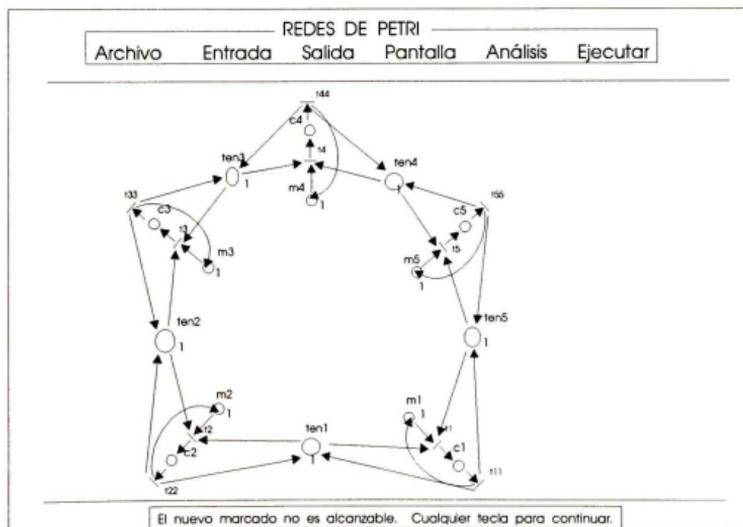


Fig. 4.6. La alcanzabilidad en el problema de los filósofos comensales.

Note en el modelo (fig. 4.5) que los lugares *ten1*, ..., *ten5* son los que coordinan el cambio de estado (*meditando* y *comiendo*) de los filósofos y como en los lugares mencionados no puede haber más de una marca no se puede dar que dos filósofos vecinos puedan comer al mismo tiempo. Con este razonamiento, podemos inferir que la RdP no se bloquea.

Finalmente, el análisis de la vivacidad dice que la RdP es viva (fig. 4.8.), esto demuestra el razonamiento contenido en el párrafo anterior.

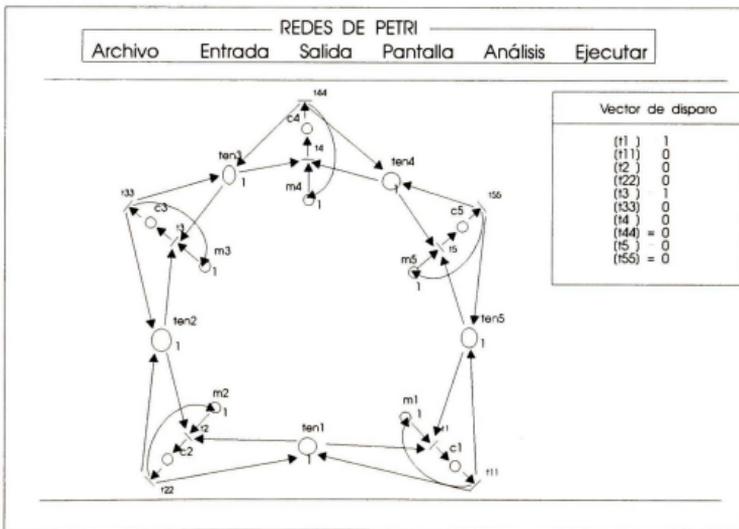


Fig. 4.7. SSRdP cuando proporciona el vector de disparo en los filósofos comensales.

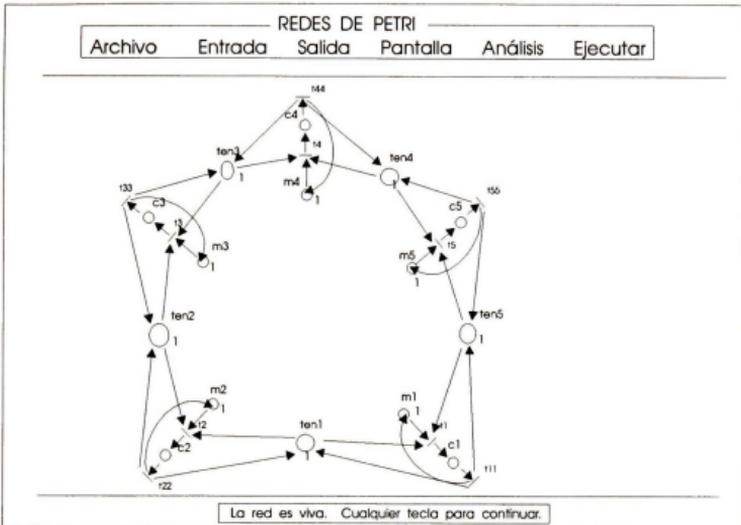


Fig. 4.8. El análisis de la vivacidad en los filósofos comensales.

## CONCLUSIONES

**I.** Las RdP son una herramienta poderosa para modelar sistemas. Una de las propiedades más importantes de estas es su flexibilidad para modelar una gran variedad de sistemas en una forma clara y sencilla. Ejemplos de sistemas que se pueden modelar con RdP son: sistemas operativos, bases de datos, programas de computadoras, protocolos de comunicación, sistemas en tiempo real, etc. En este trabajo se ha presentado un sistema que permite manejar, simular y analizar RdP. No obstante que, como toda propuesta, *SSRdP* podría ser mejorado, su estado actual lo hace ser un sistema bastante competitivo además de ofrecer una manera fácil y atractiva de explotar esta herramienta de modelación. Una de las aplicaciones inmediatas de *SSRdP* es utilizarlo en la enseñanza de la teoría RdP.

**II.** Una de las aportaciones de este trabajo es el método matemático propuesto que permite verificar la presencia o ausencia de bloqueo dentro de una RdP presentado en el capítulo 2. Esta propuesta representa una aportación original no registrada en la literatura existente, implementada en el sistema *SSRdP*.

**III.** El sistema implementado mantiene la limitante de resolver únicamente el primer sistema de ecuaciones de acuerdo con el método de análisis de vivacidad planteado en el capítulo 2 de este impreso, una implementación completa debe de resolver todos los sistemas de ecuaciones lineales resultantes. La implementación completa del método representa una dirección de trabajo a futuro.

## REFERENCIAS

[Agaoua87] Agaoua S., Ladet P.,  
*Analyse de la pseudo-vivacité des Réseaux de Petri par résolution  
d'un système linéaire*, R.A.I.R.O. APII, vol. 21, 1987, pp. 553-568.

[Desrochers90] Alan A. Desrochers,  
*Modeling and control of Automated Manufacturing Systems*, IEEE  
Computer Society Press Tutorial, 1990.

[Peterson81] James L. Peterson,  
*Petri Net Theory and the Modeling of Systems*  
Prentice Hall 1981

[López90] Ernesto López M., Antonio Ramírez T.,  
*Modelado de sistemas de manufactura con Redes de Petri coloreadas*,  
Informe Técnico, CINVESTAV, departamento de Ingeniería Eléctrica,  
1990.

[Petri62] Carl A. Petri,  
*Kommunikation mit Automaten*,  
Tesis Doctoral, Universidad de Bonn, 1962.

[Rozenberg91] G. Rozenberg,  
*Lecture Notes in Computer Science*,  
G. Goos and J. Hartmanis Eds.,  
Springer-Verlag 1991.

[Silva85] Manuel Silva,  
*Las Redes de Petri: en la Automática y la Informática*,  
Editorial Ac, Libros Científicos y Técnicos Madrid.

[Stroustrup87] Bjarne Stroustrup,  
*C++ Programming Language*,  
Addison-Wesley 1987.

EL JURADO DESIGNADO POR LA SECCION DE COMPUTACION DEL DEPARTAMENTO DE INGENIERIA ELECTRICA, APROBO EL DIA 22 DEL MES DE OCTUBRE DEL AÑO 1993, EL TRABAJO DE TESIS :

**SIMULACION CON REDES DE PETRI**

DESARROLLADO POR EL ALUMNO :

**VALDEMAR GONZALEZ AVILA**



---

M. EN C. JOSE OSCAR OLMEDO AGUIRRE  
Coordinador Académico



---

DR. SERGIO V. CHAPA VERGARA  
Jefe de la Sección



---

M. EN C. CARLOTA TAMAYO I.  
Prof. Auxiliar Investigador

BIBLIOTECA DE INGENIERIA ELECTRICA  
FECHA DE DEVOLUCION

El lector está obligado a devolver este libro  
antes del vencimiento de préstamo señalado  
por el último sello.

13 MAR. 1995  
14 AGO. 1996  
17 ABR. 1997  
-9 NOV. 1999  
09 DIC. 1999  
-6 JUL. 2000  
20 JUL. 2000  
12 SET. 2000  
15 JUN. 2002  
-6 NOV. 2002  
27 NOV. 2002

DEVOLUCION

AUTOR GONZALEZ AVILA, V.

TITULO SIMULACION CON REDES DE PETRI

CLASIF. XM  
93.11

RGTR. BI  
13814

NOMBRE DEL LECTOR	FECHA PREST.	FECHA DEVOL.
LEON SOLANO	4/8/97	2/9/97
IGNACIO UGUA P.	20/11/97	15/12/97
Dr. Guillermo Morales	13-12-98	13/12/98
Noe Sierra Romero	17/7/96	7/10/96
Sofia	6/3/97	7/3/97
Oliver González Martha	20/3/97	21/4/97
Amador Meneses Viveros	19/2/99	5/1/99
Amador Meneses Viveros	21/11/99	7/1/2000
Tco. Beltrán Carbajal	15/05/2000	9
Andrés Blanco O	22/0	
Gmo Jimenez D		
José Trujillo F		

