

3.- REDES DE PETRI

3.1.-DEFINICIÓN Y CONCEPTOS.

3.2.-REGLAS DE EVOLUCIÓN DE MARCADO.

3.3.-ESTRUCTURAS BÁSICAS

3.4.-REDES DE PETRI FRENTE A GRAFOS DE ESTADO

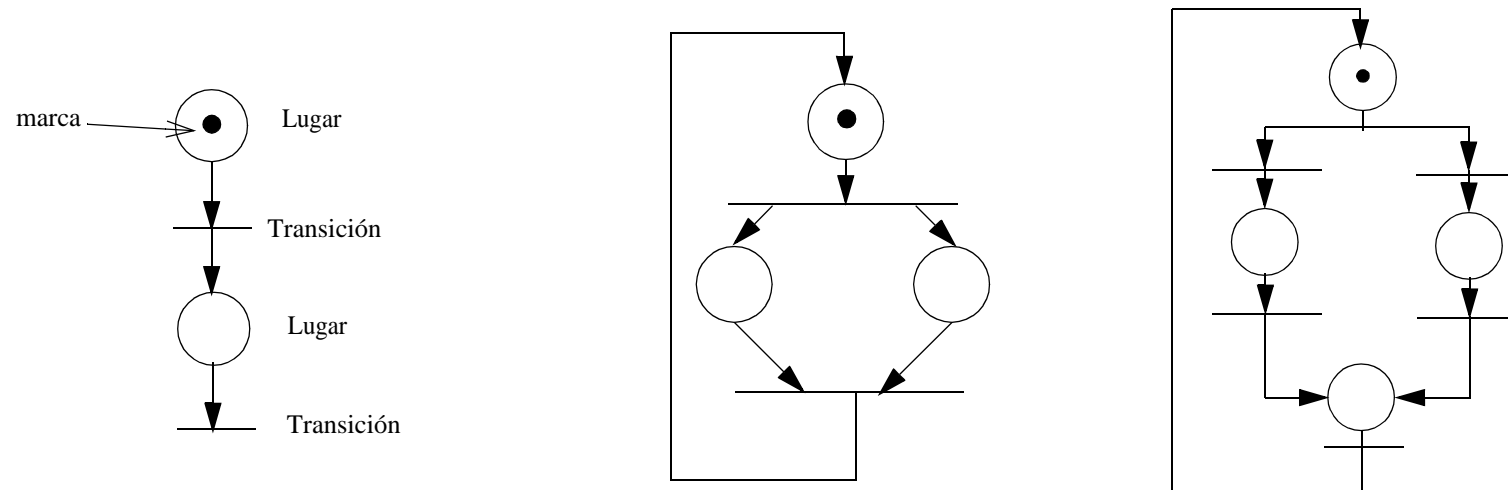
3.5.-PROPIEDADES Y VALIDACIÓN.

3.1.- DEFINICIÓN Y CONCEPTOS

Una red de Petri es un grafo orientado con dos tipos de nodos: **lugares** (representados mediante circunferencias) y **transiciones** (representadas por segmentos rectos verticales).

Los lugares y las transiciones se unen mediante arcos o flechas.





Un arco une siempre lugares con transiciones y nunca dos lugares o dos transiciones.

Una **transición** puede ser **destino** de **varios lugares** y un **lugar** puede ser el **destino** de **varias transiciones**.

Una **transición** puede ser **origen** de **varios lugares** y un **lugar** puede ser **origen** de **varias transiciones**.

Los lugares pueden presentar marcas (una marca se representa mediante un punto en el interior del círculo).

Cada lugar tiene asociada una acción o salida.

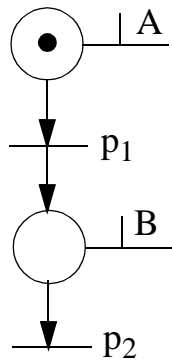
Los lugares que contiene marcas se consideran lugares activos.

Cuando un lugar está activo sus salidas están a uno.

A las transiciones se les asocia eventos (funciones lógicas de las variables de entrada).

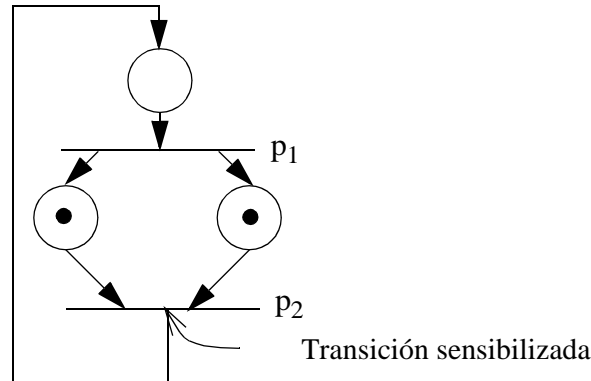
Una transición se dice que está sensibilizada cuando todos su lugares origen están marcados.

Cuando ocurre un evento asociado a una transición (la función lógica se hace uno), se dice que la transición está validada.



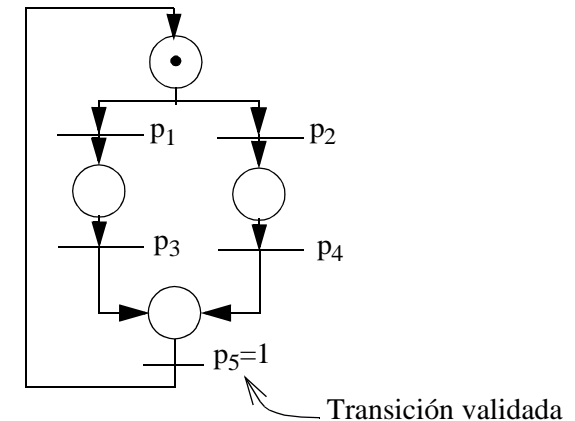
Salida A=1
Salida B=0

LUGAR ACTIVO



Transición sensibilizada

TRANSICIÓN SENSIBILIZADA



Transición validada

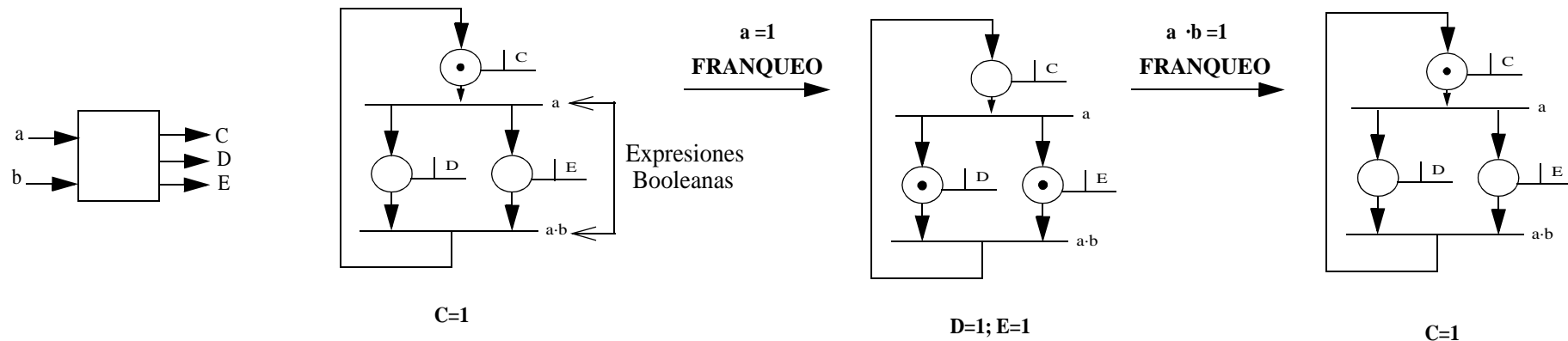
TRANSICIÓN VALIDADA

3.2.- REGLAS DE EVOLUCIÓN DE MARCADO.

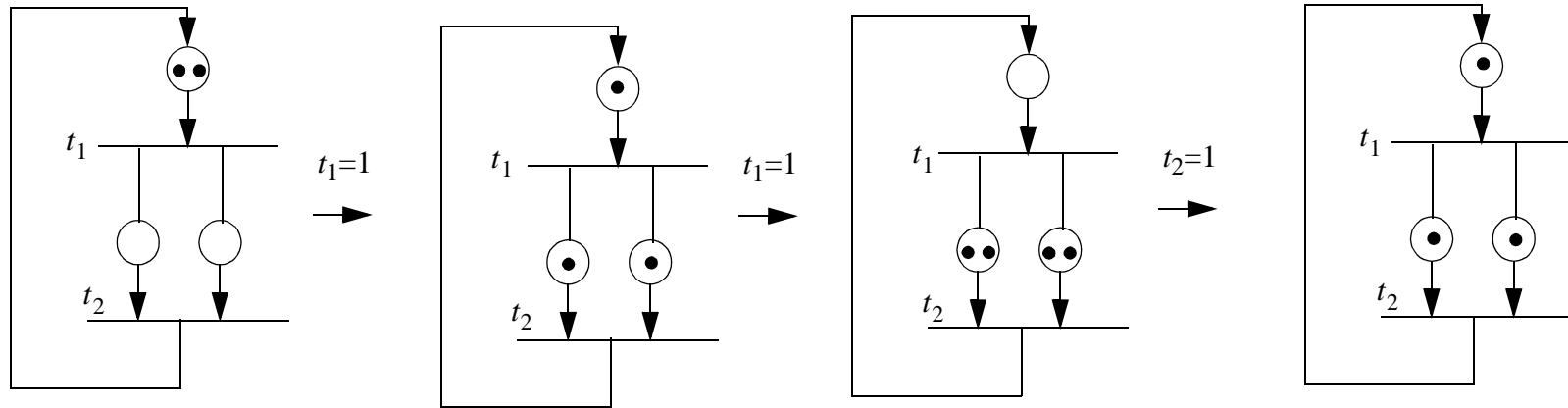
El marcado cambia al franquear las transiciones.

Para franquear una transición ha de estar validada y sensibilizada

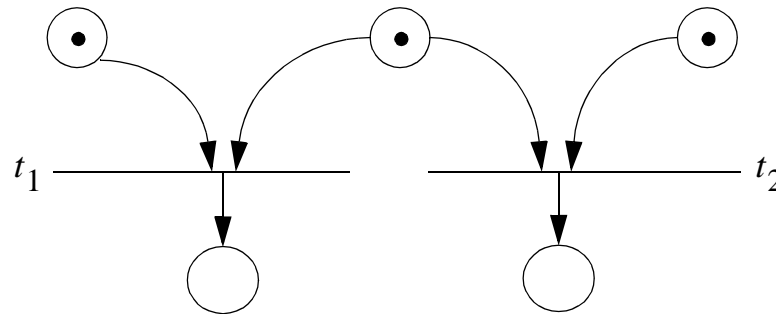
Cuando una transición se franquea desaparecen las marcas de los lugares origen y se añade una marca a cada uno de los lugares destino.



Un lugar puede tener más de una marca, **ejemplo:**



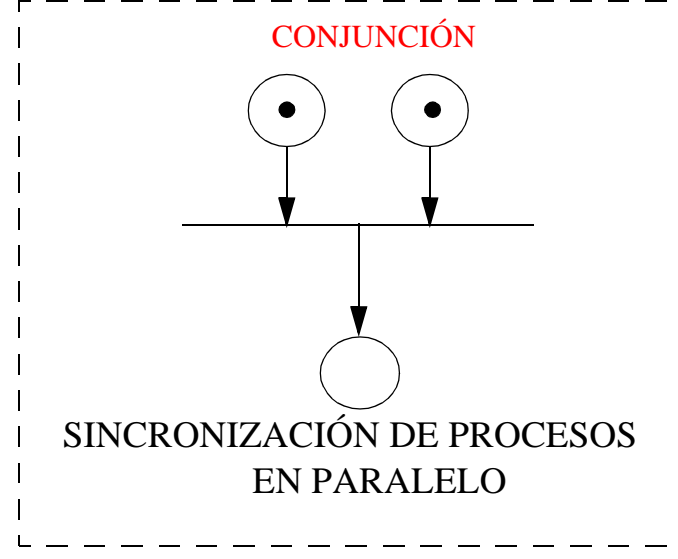
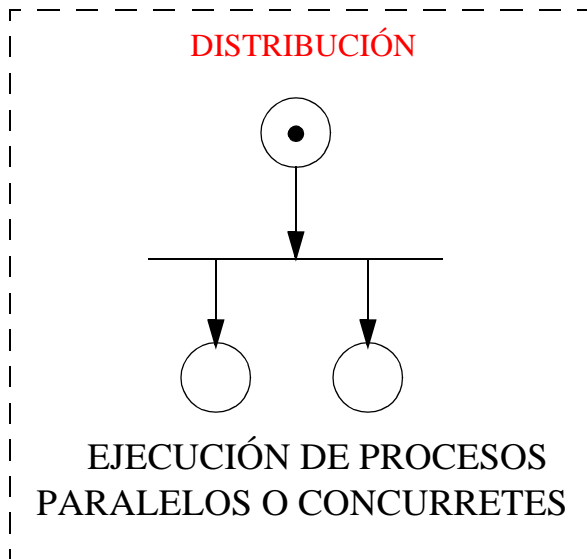
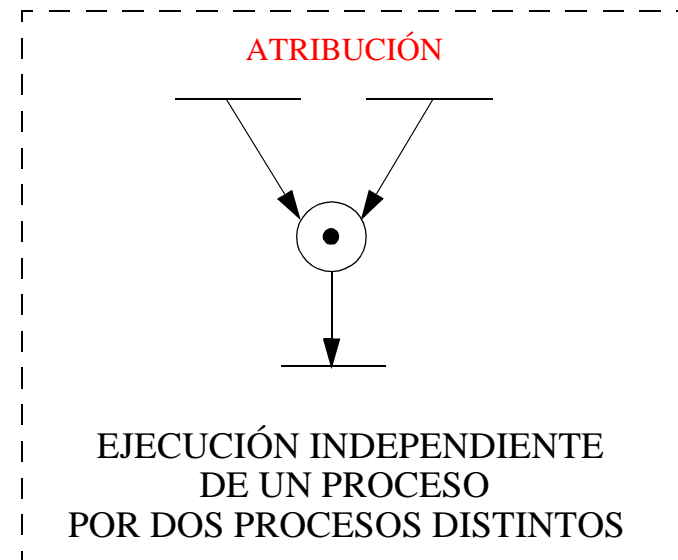
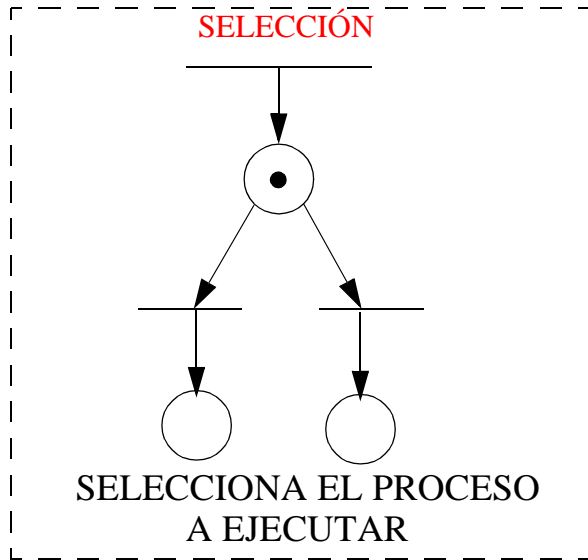
Cuando dos transiciones que están sensibilizadas a la vez, pueden entrar en conflicto, **ejemplo:**



Para que la red sea válida las condiciones de validación t_1 y t_2 no pueden darse a la vez.

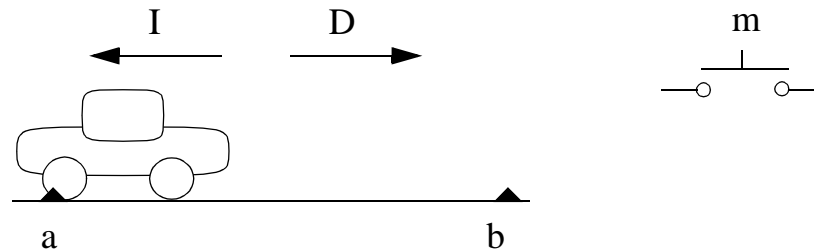


3.3.- ESTRUCTURAS BÁSICAS



3.4.- REDES DE PETRI FRENTE A GRAFOS DE ESTADO

Problema: Se pretende que el coche vaya de **a** hacia **b** y vuelva. Existe un botón de puerta en marcha. En el inicio el coche está en **a**



Entradas:

m: puesta en marcha.

a : sensor llegada a izquierda.

b : sensor llegada a derecha.

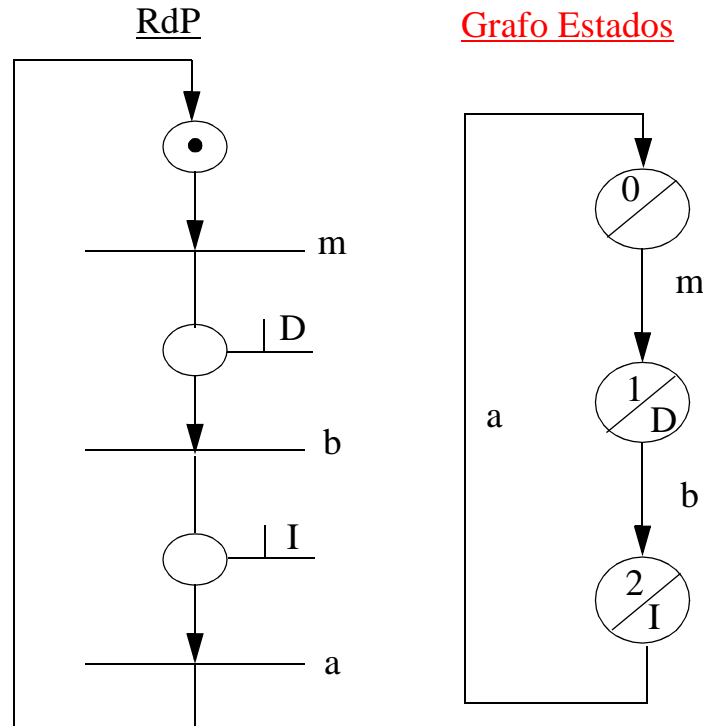
Salidas:

I : indica desplazamiento a izquierda.

D : indica desplazamiento a derecha.



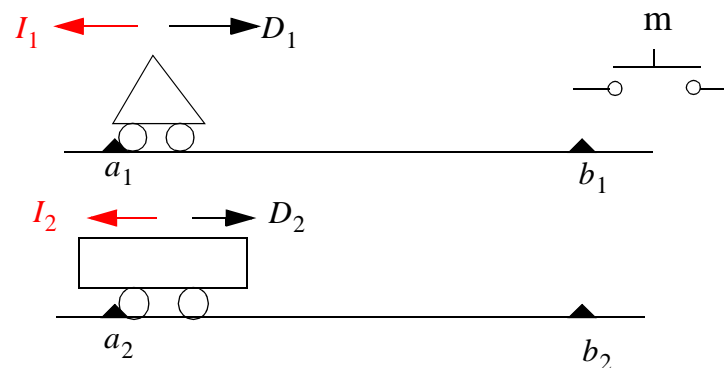
Comparación de la solución obtenida con la red de Petri y la obtenida con el grafo de transición de estados.



En este caso las representaciones son iguales pero en problemas más complejos la RdP se aproxima más al problema y trata mejor la ejecución de tareas concurrentes.

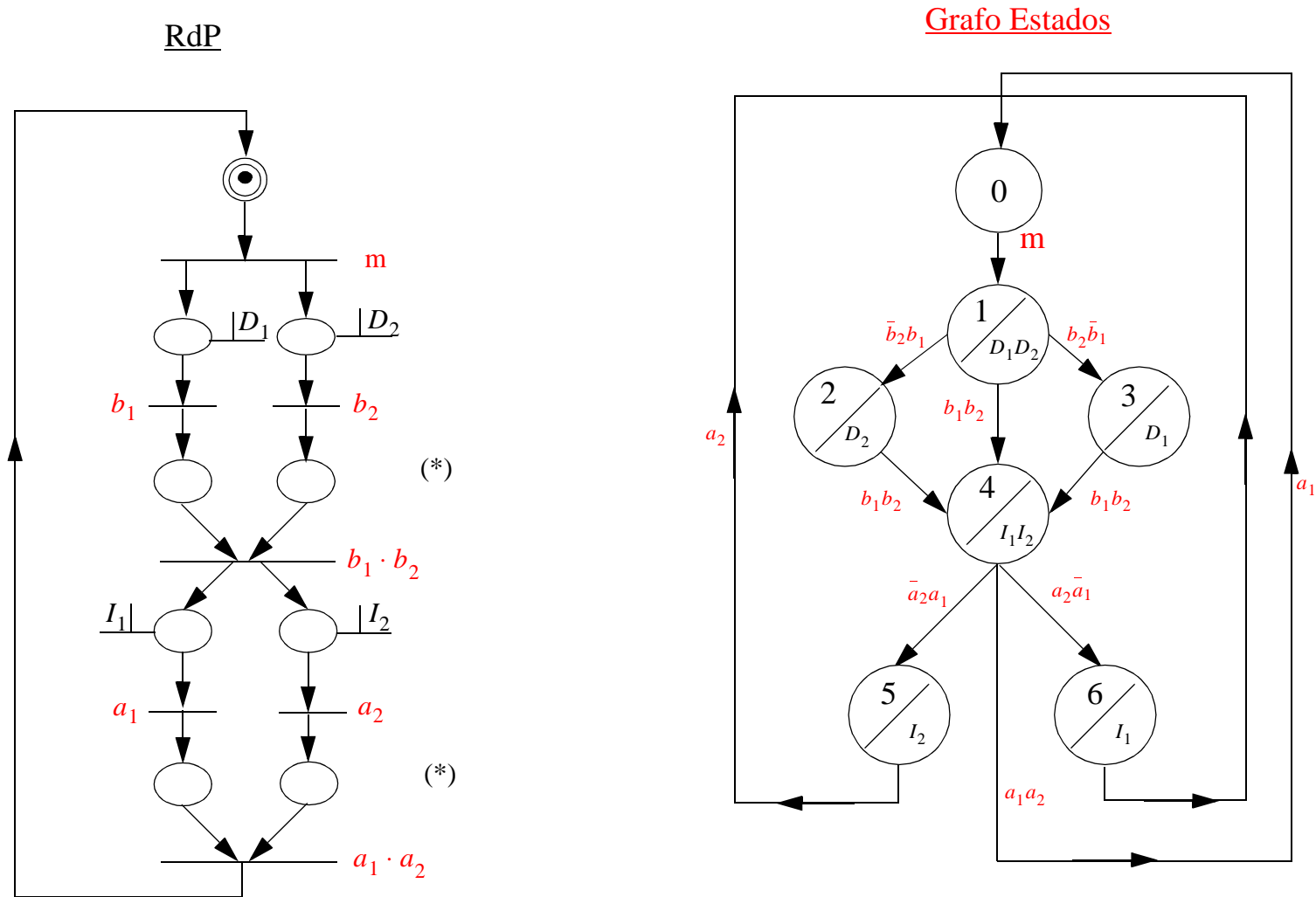


Problema: Se pretende que los dos carritos de la figura, que en principio están en el lado izquierdo, vayan al extremo derecho. El primero que llegue esperará al otro (sincronización). Después vuelven al lado izquierdo donde el sistema se quedará en reposo cuando lleguen los dos carritos (sincronización). Existen un botón de marcha m y dos sensores de llegada en cada lado a_1 y a_2 en el lado derecho y b_1 y b_2 en el lado izquierdo.



Las señales de entrada son m , a_1 , a_2 , b_1 y b_2 las salidas son I_1 , I_2 , B_1 , B_2





(*) Estados de Espera. Cuando el marcado se traslada a estos lugares a la vez, las dos carritos están listos para ponerse en marcha.

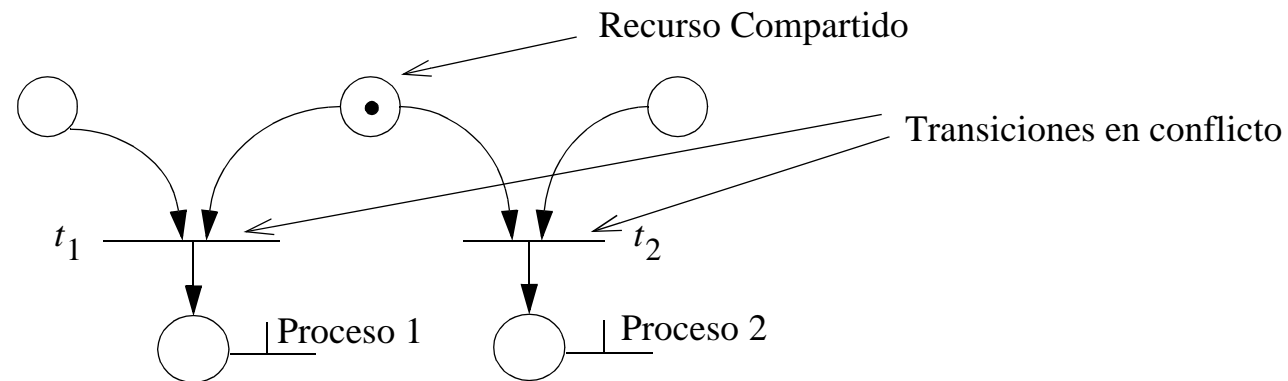


VENTAJAS REDES DE PETRI :

- 1) Tratamiento individual de procesos independientes.
- 2) Procesos paralelos o concurrentes.
- 3) Recursos compartidos.

RECURSOS COMPARTIDOS

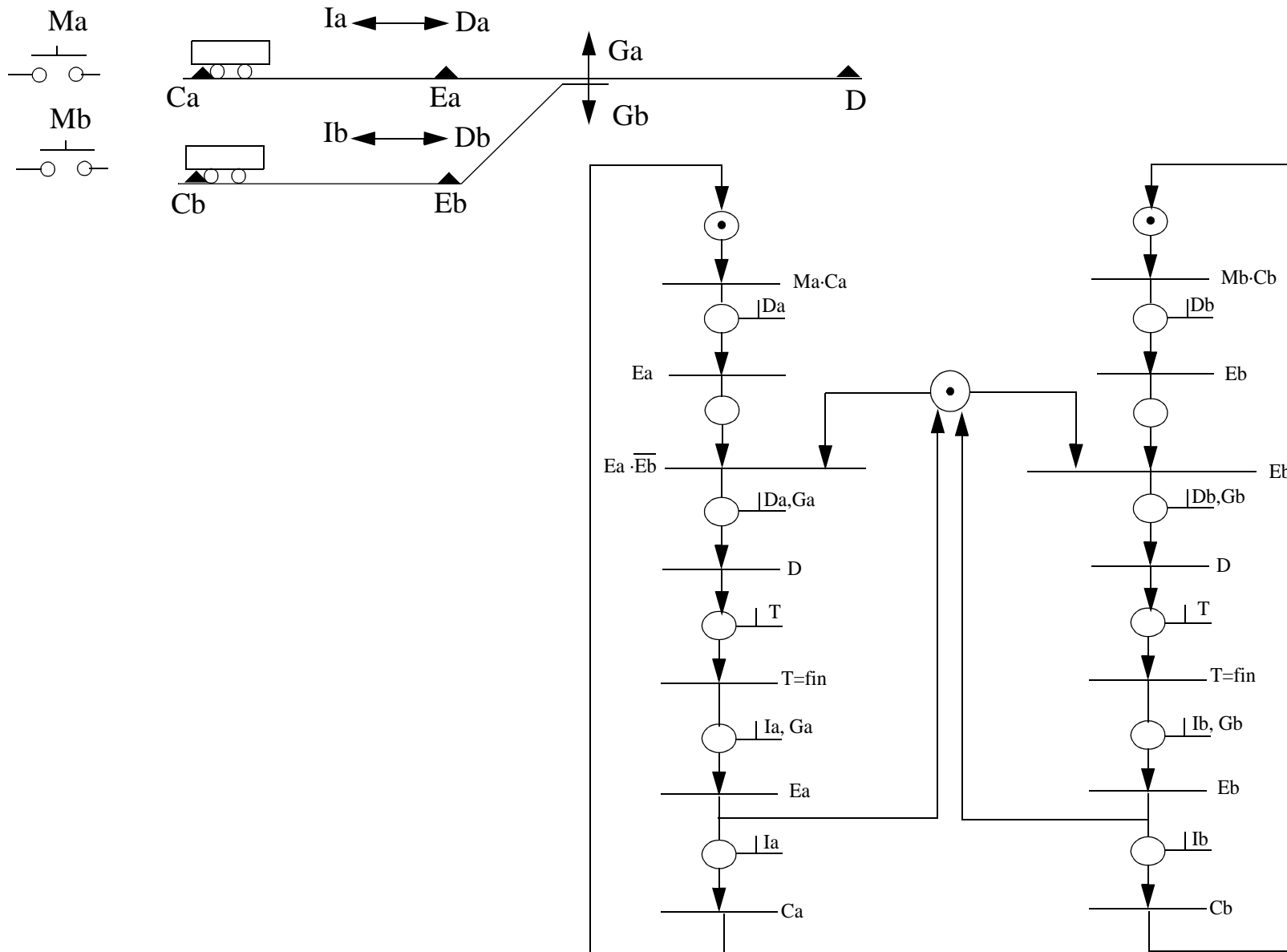
Las RdP **permiten modelar** sistemas donde **un recurso es compartido** por dos procesos de forma que el uso del recurso durante la ejecución de un proceso impide que dicho recurso sea utilizado por el otro proceso.



Un recurso compartido se modela mediante un lugar con una marca inicial y transiciones en conflicto



Ejemplo Recurso Compartido:



3.5.- PROPIEDADES Y VALIDACIÓN.

La validación consiste en comprobar que se cumplen las propiedades de:

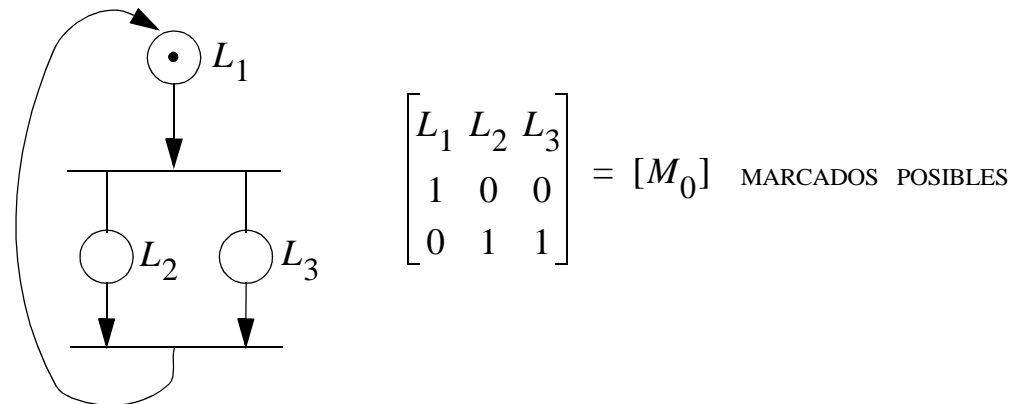
- VIVACIDAD; LIMITACIÓN; REVERSIBILIDAD

Hay que considerar :

M_0 : marcado inicial. De este se desprende el comportamiento del sistema.

$[M_0]$: vector de marcados posibles a partir de un marcado inicial.
(marcados alcanzables).

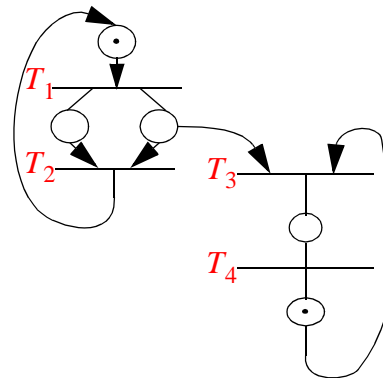
Ejemplo:



VIVACIDAD : Se trata de un concepto relacionado con la idea de “no bloqueo”.

Definición : Una transición se dice viva si para un marcado inicial existe una secuencia de franqueos para la cual se puede franquear esa transición. Si todas las transiciones de una red son vivas, la RdP se llama viva y así la red nunca se bloquea.

Ejemplo: Una red de Petri no viva.

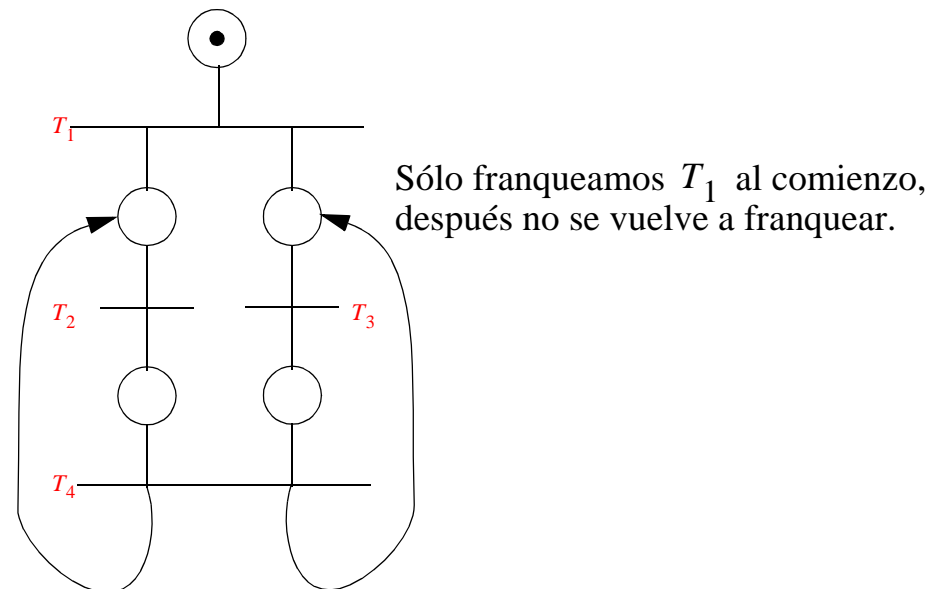


Para la secuencia de franqueos T_1, T_2, T_1, T_2 , no hay bloqueo. Si ahora se franquean las transiciones T_1, T_3, T_4 , ya no se puede franquear ninguna transición más, la red queda “*bloqueada*”.



Podemos tener redes pseudo-vivas en las que existen algunas transiciones vivas y no se bloquea totalmente.

Ejemplo: RdP Pseudoviva.

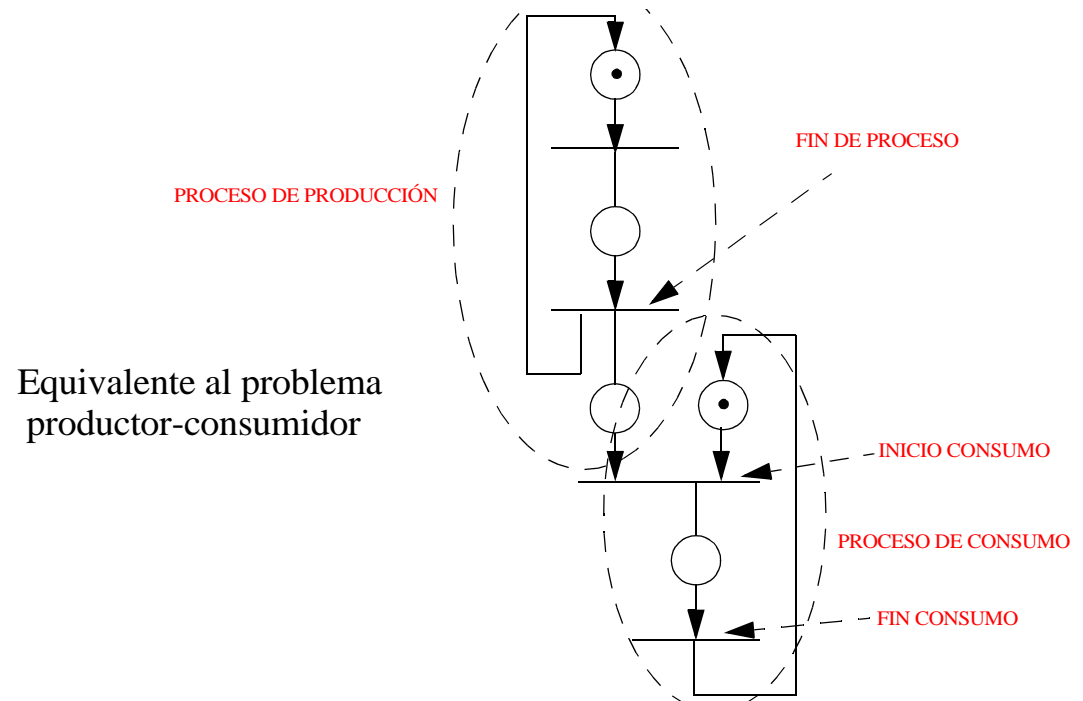


LIMITACIÓN:

Definición : Se dice que la red está k - limitada si para todo marcado alcanzable tenemos que ningún lugar tiene un número de marcas mayor que k . Las redes **1-limitadas** pueden implementarse mediante biestables, estas redes son conocidas como **binarias**.

Si la red diseñada genera más marcas que las que su limitación permite el modelado será erróneo

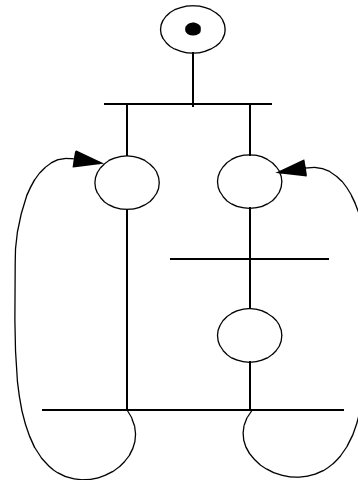
Ejemplo: Una red no limitada:



Reversibilidad :

Definición : Una RdP es reversible si para cualquier marcado alcanzable es posible volver al marcado inicial.

Ejemplo:



Esta RdP es pseudoviva, además no tiene la propiedad de reversibilidad ya que el marcado inicial no se puede obtener jamás .



