Redes de petri

Algoritmo murata

Nadia Nohely Gonzalez

Definición

P={p 1,p2 ,...,pm} es un conjunto finito de Plazas/Lugares.

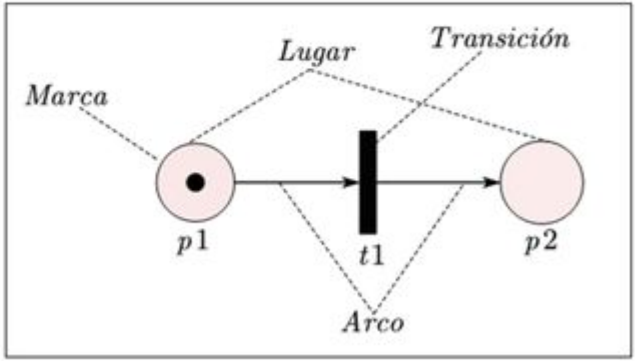
T={ t1,t2,...,tn } es un conjunto finito de transiciones.

F⊆ (P ×T) ∪ (T ×P) es un conjunto de arcos dirigidos.

W:F→{1, 2, 3, ...} es una función de pesos de los arcos.
*por convencion si no se nota explicitamente el peso es 1

M 0:P→{1, 2, 3, ...} es el marcado inicial de la red.

P∩T=∅ y P ∪ T≠∅



Definiciones:

Marcas de las Redes de Petri: Una marca μ es una asignación de tokens (puntos negros) a los sitios de una Red de Petri. El número y posición de los tokens puede cambiar durante la ejecución de una Red de Petri.

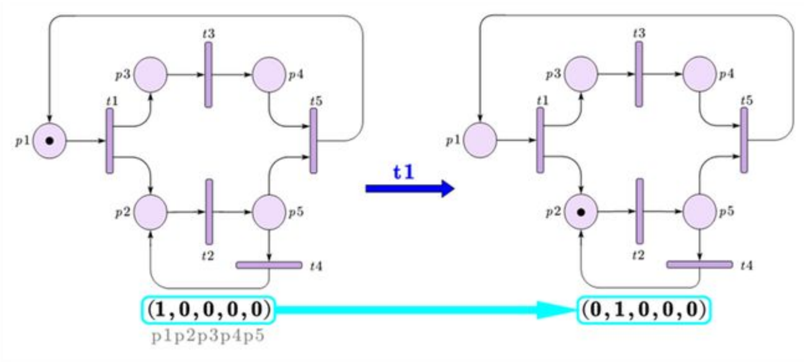
DEFINICIÓN: Una marca μ de una Red de Petri C = (P,T,I,O) es una función que va desde el conjunto de sitios P a los enteros no negativos N.

Métodos de análisis de propiedades dinámicas:

1. El árbol de cobertura.

2. Matriz de incidencia y ecuación de estado.

3. Reglas de reducción



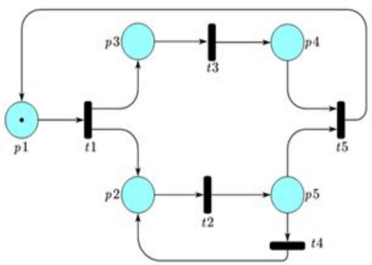
Análisis

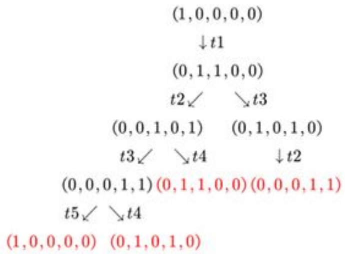
En las redes de petri el estado del sistema (marcado) va variando según las reglas del disparo

1- transición habilitada si cada lugar de entrada p de t es marcada con al menos w(p,t) marcas,donde w(p,t) es el peso del arco de p a t.

2- transición puede o no ser disparada

3- El disparo de una transición t habilitada remueve w(p,t) marcas de cada lugarde entrada p de t y agrega w(t,p) marcas a cada lugar de salida p de t,donde w(t,p) es el peso de los arcos de t a p





El árbol de cobertura: Dada la Red de Petri (N,M 0) con marcado inicial M 0, se puede obtener tantos nuevos marcados como transiciones habilitadas disparadas. Este proceso resulta en un árbol de marcados infinito para una PN no acotada. Para redes acotadas, el árbol de cobertura es llamado árbol de alcanzabilidad.

Herramientas

Matriz de Incidencia: Para una Red de Petri N con n transiciones y m lugares, la matriz de incidencia es una matriz de enteros de n×m, llamada A = [aij], y aij está dada por:

$$C^{-}=\begin{array}{c}\begin{pmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&0&1\\0&0&0&1&1\end{pmatrix}\begin{array}{l}t1\\t2\\t3\\t4\\t5\end{array}\\ \begin{array}{ccccc}p1&p2&p3&p4&p5\end{array}\end{array}$$

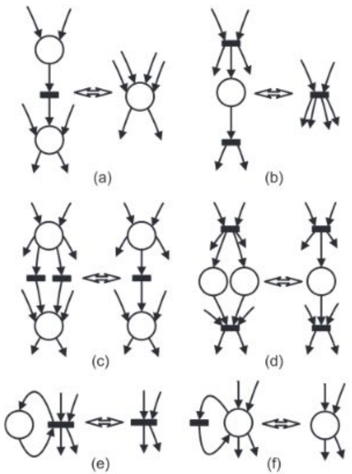
Matriz de incidencia previa

$$C^{+}=\begin{array}{c}\begin{pmatrix}0&1&1&0&0\\0&0&0&0&1\\0&0&0&1&0\\0&1&0&0&0\\1&0&0&0&0\end{pmatrix}\begin{array}{l}t1\\t2\\t3\\t4\\t5\end{array}\\ \begin{array}{ccccc}p1&p2&p3&p4&p5\end{array}\end{array}$$

Matriz de incidencia posterior

$$C=C^{+}-C^{-}=\begin{array}{c}\begin{pmatrix}-1&1&1&0&0\\0&-1&0&0&1\\0&0&-1&1&0\\0&1&0&0&-1\\1&0&0&-1&-1\end{pmatrix}\begin{array}{l}t1\\t2\\t3\\t4\\t5\end{array}\\ \begin{array}{ccccc}p1&p2&p3&p4&p5\end{array}\end{array}$$

Matriz de incidencia



Reglas de reducción: Las reglas de reducción permiten convertir sistemas complejos en sistemas más simples de menos lugares y transsiciones, preservando sus propiedades originales. La figura 5 presenta las seis reglas de reducción, las cuales son:

a) Fusión de lugares en serie
b) Fusión de transiciones en serie
c) Fusión de transiciones paralelas
d) Fusión de lugares parelelos
e) Eliminación de auto bucles en lugares
f) Eliminación de auto bucles en transiciones

Utilizamos las herramientas para crear un programa que crea las diferentes marcas de una red de petri

utilizando el algoritmo $$M_d = M_0 + A^T \sum_{k=1}^{d} U_k$$

ecuación de la sumatoria de todos los vectores de disparo $$A^T \sum_{k=1}^{d} U_k = A^T x$$

*podemos usar por ejemplo el algoritmos de caminos de dijkstra

Habilitación de una rama $M_0 > e * A_{ij}^-$
donde e es el vector habilitado para disparar,
condición para la habilitación de rama

e: matriz de 1 X t
c-:matriz de t x p
donde t son las transiciones y los lugares

Desarrollo para una red de petri (2 transiciones y 2 nodos)

$$M_d = M_0 + A_{ij} * \mu_k$$

1xp 1xp 1xt txp

M_d = Estado final

M_0 = Estado actual de un sistema o

A_{ij} = Matriz de Incidencia

μ_k = Vector indica las transiciones que permiten el desplazamiento de un token

$$A_{ij} = A_{ij}^{+} - A_{ij}^{-} \Rightarrow A_{ij} = \begin{array}{c c} & \begin{array}{cc} T_1 & T_2 \end{array} \\ \begin{array}{c} P_1 \\ P_2 \end{array} & \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \end{array}$$

(matriz de entrada - matriz de salida)

Matriz de incidencia $A_{ij}^{+} - A_{ij}^{-}$

$$A_{ij} = \begin{array}{c c} & \begin{array}{cc} T_1 & T_2 \end{array} \\ \begin{array}{c} P_1 \\ P_2 \end{array} & \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \end{array} - \begin{array}{c c} & \begin{array}{cc} T_1 & T_2 \end{array} \\ \begin{array}{c} P_1 \\ P_2 \end{array} & \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \end{array}$$

μ_k

$$\mu = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$$

Ecuación de estado final

$$M_d = (m_1, m_2) + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} * \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

Condición para habilitar una rama

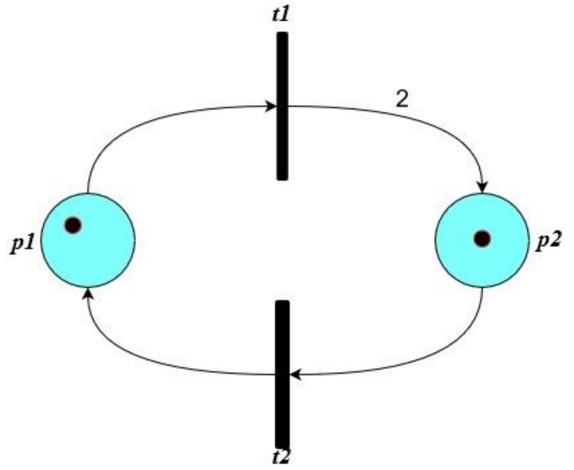
e = vector de transición

$$M_0 > e * A_{ij}^{-}$$

$$M_0 > (e_1, e_2) * \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

Ejemplo 2 x 2

Gráfico



Salida de consola

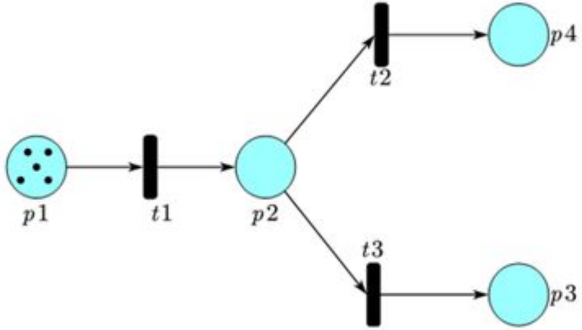
```
matriz Previa
matriz:
1 0
0 1
______
matriz posterior
matriz:
0 2
1 0
______
matriz:
-1 2
1 -1
```

```
______Termina Crear Marcas______
[[1, 1], [0, 3], [2, 0]] all marcas
desconectado
```

```
Todas las marcas: [[1, 1], [0, 3], [2, 0]]  /n
Marcas Infinitas: [array([1, 2]), array([1, 2])]
```

Ejemplo 3 x 4

Gráfico



$$M0 - Pre(P,T) + Pos(T,P) = MF$$

$$(5\ \ 0\ \ 0\ \ 0) - \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = (4\ \ 1\ \ 0\ \ 0)$$

```
matrices
matriz: Previa
1 0 0 0
0 1 0 0
0 1 0 0
______
matriz: Posterior
0 1 0 0
0 0 0 1
0 0 1 0
______
matriz: incidencia
-1 1 0 0
0 -1 0 1
0 -1 1 0
```

Salida de consola

```
------
[5, 0, 0, 0]
[4, 1, 0, 0]
[3, 2, 0, 0]
[2, 3, 0, 0]
[1, 4, 0, 0]
[0, 5, 0, 0]
[0, 4, 0, 1]
[0, 3, 0, 2]
[0, 2, 0, 3]
[0, 1, 0, 4]
[0, 0, 0, 5]
[0, 0, 1, 4]
[0, 1, 1, 3]
[0, 0, 2, 3]
[0, 2, 1, 2]
[0, 1, 2, 2]
[0, 0, 3, 2]
[0, 3, 1, 1]
[0, 2, 2, 1]
```

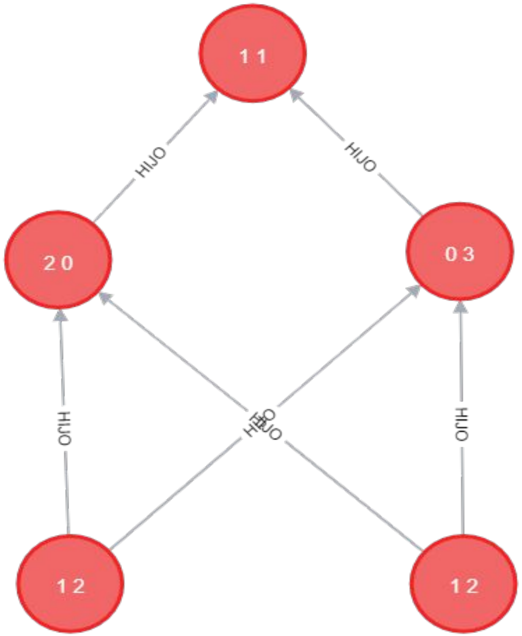
```
[0, 1, 3, 1]
[0, 0, 4, 1]
[0, 4, 1, 0]
[0, 3, 2, 0]
[0, 2, 3, 0]
[0, 1, 4, 0]
[0, 0, 5, 0]
[1, 3, 0, 1]
[1, 2, 0, 2]
[1, 1, 0, 3]
[1, 0, 0, 4]
[1, 0, 1, 3]
[1, 1, 1, 2]
[1, 0, 2, 2]
[1, 2, 1, 1]
[1, 1, 2, 1]
[1, 0, 3, 1]
[1, 3, 1, 0]
[1, 2, 2, 0]
[1, 1, 3, 0]
[1, 0, 4, 0]
[2, 2, 0, 1]
```

```
[2, 2, 0, 1]
[2, 1, 0, 2]
[2, 0, 0, 3]
[2, 0, 1, 2]
[2, 1, 1, 1]
[2, 0, 2, 1]
[2, 2, 1, 0]
[2, 1, 2, 0]
[2, 0, 3, 0]
[3, 1, 0, 1]
[3, 0, 0, 2]
[3, 0, 1, 1]
[3, 1, 1, 0]
[3, 0, 2, 0]
[4, 0, 0, 1]
[4, 0, 1, 0]
```

Salida Gráfica del Proyecto en Neo4j

Neo4j

Salida 2x2



Salida 3x4