AULA 6

Redes de Petri (Chapter 4 - Cassandras).

Uma alternativa para os Automatos para modelos atemporais aplicados em Sistemas de Eventos Discretos é a denominada Redes de Petri. (desenvolvida inicialmente em 1960 por C.A. Petri).

Nas Redes de Petri as transições são representadas explicitamente, e elas lidam com os eventos de acordo com certas regras.

A Rede de Petri é um grafo. Os autômatos podem sempre ser representados por Redes de Petri; por outro lado nem todas Redes de Petri podem ser representadas por Autômato de Estado Finito.

Conceitos Básicos

a) Notação e Definições

Eventos são associados à Transições

Informações relacionadas com as condições da ocorrência das transições são colocadas num "lugar". Estes lugares são vistos como entradas das transições. (nós)

Outros lugares são vistos como saídas das transições.

Os lugares e transições são conectados por intermédio de arcos.

Uma Rede de Petri pode ser formalizada como:

$$(P, T, A, \omega)$$

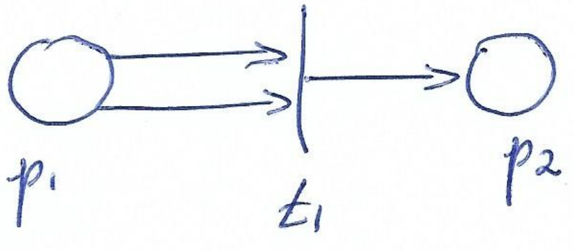
$P \rightarrow$ conjunto finito de lugares

$T \rightarrow$ conjunto finito de transições

$A \subseteq (P \times T) \cup (T \times P) \rightarrow$ conjunto de arcos dos lugares para as transições e das transições para os lugares.

$\omega : A \rightarrow \{1, 2, 3, \ldots\}$: função peso dos arcos (representa o nº de arcos).

Exemplo 1



$P = \{p_1, p_2\}$

$T = \{t_1\}$

$A = \{(p_1, t_1), (t_1, p_2)\}$

$\omega(p_1, t_1) = 2$

$\omega(t_1, p_2) = 1$

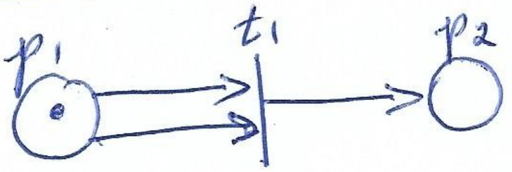
b) Marcação e Espaço de Estados

Necessita-se de mecanismos para verificar se as condições de disparo das transições são válidas. Este mecanismo é feito por meio das "marcas" ou "tokens". Quando existe o "token" a condição está satisfeita.

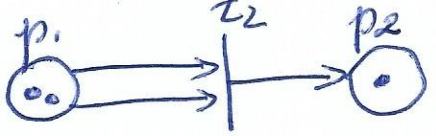
Definição: Uma Rede de Petri marcada é formalizada como (P, T, A, w, x) onde x representa as marcas nos lugares P.

$$x = [x(p_1), x(p_2), \ldots x(p_n)]$$

Exemplo 2:



$x_1 = [1, 0]$



$x_2 = [2, 1]$

Definição: Uma transição $t_j \in T$ é dita habilitada se $x(p_i) \geq w(p_i, t_j)$ para todos $p_i \in I(t_j)$

"input"

Dinâmica da Rede de Petri

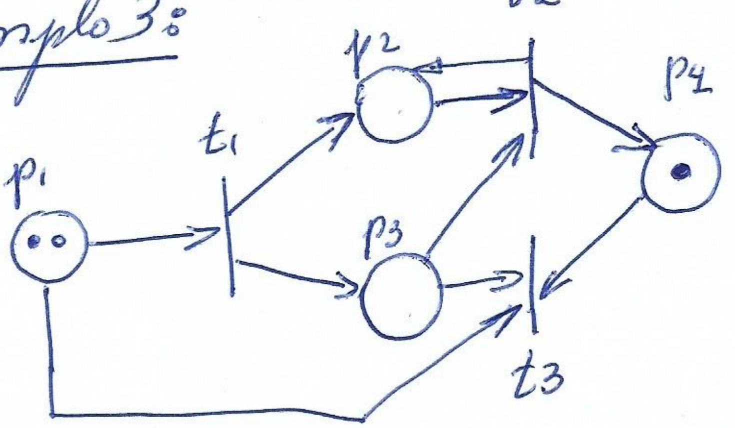
A dinâmica ocorre através da movimentação de tokens pela Rede de Petri.

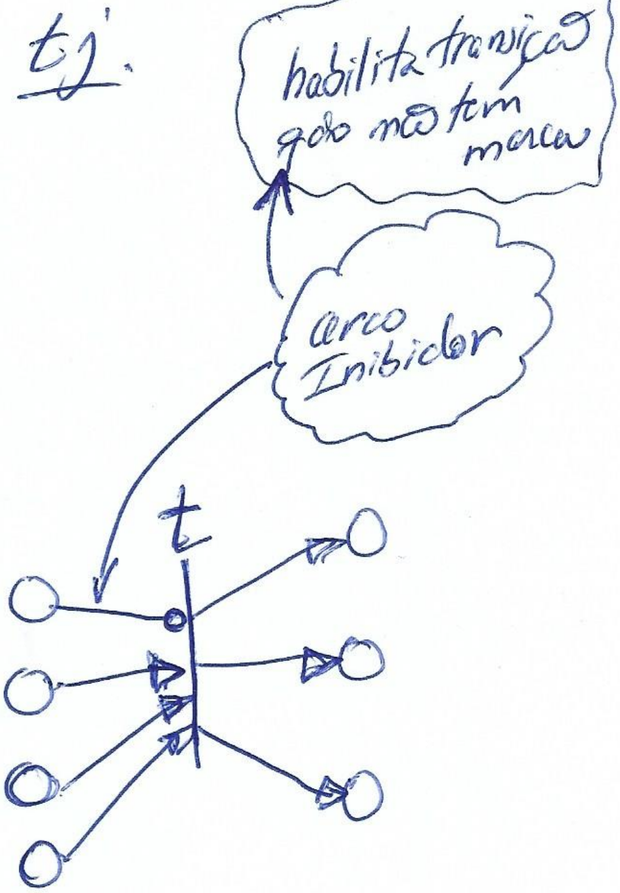
Para haver uma <u>transição disparada</u> tem-se:

(1) Ela precisa estar habilitada

(2) Após o disparo da transição t_j: se $\underline{p_i}$ é um lugar de <u>entrada de</u> t_j, esse lugar perde o número de marcas correspondente ao peso do arco (p_i, t_j); se $\underline{p_i}$ é um lugar de <u>saída</u> de t_j esse lugar ganha marcas conforme o peso do arco (t_j, p_i). O $\underline{p_i}$ pode ser um lugar de <u>entrada</u> e <u>saída</u> da transição $\underline{t_j}$.

<u>Exemplo 3:</u>



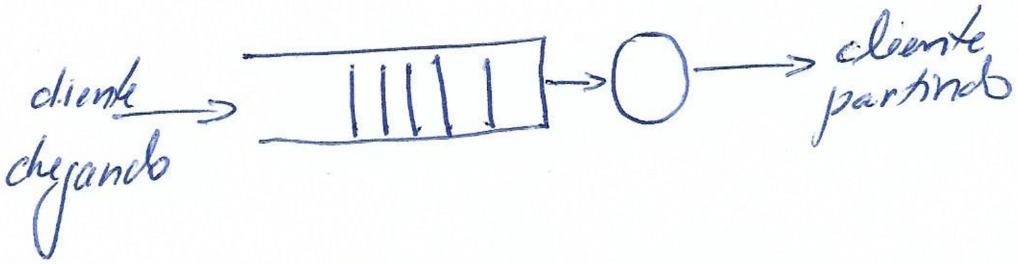


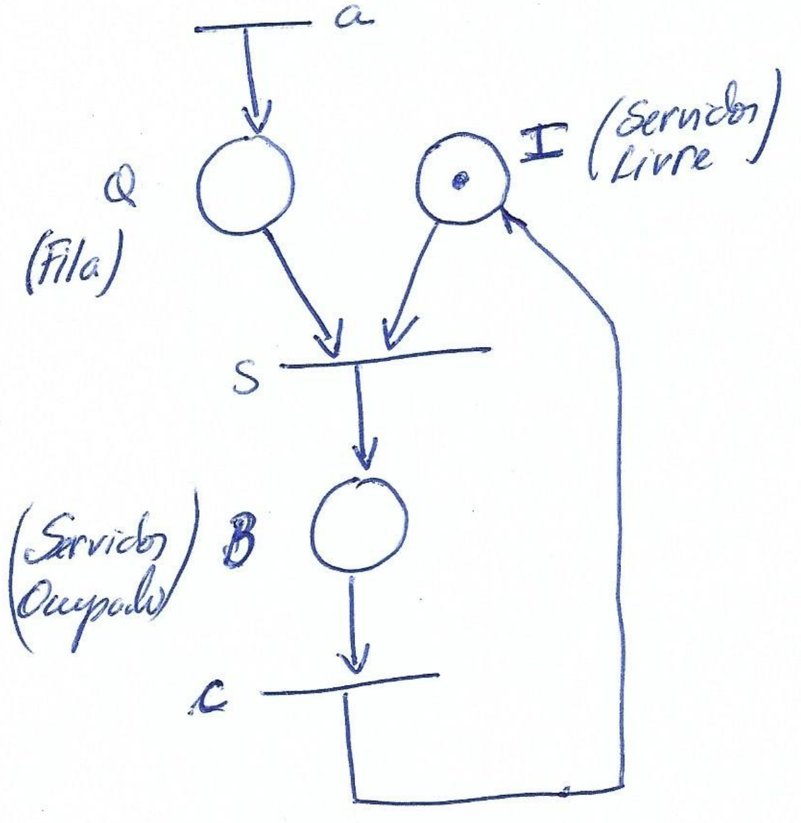
Definição: Estados Alcançáveis

Conjunto de todos os estados alcançáveis assumindo que cada transição habilitada é disparada uma de cada vez.

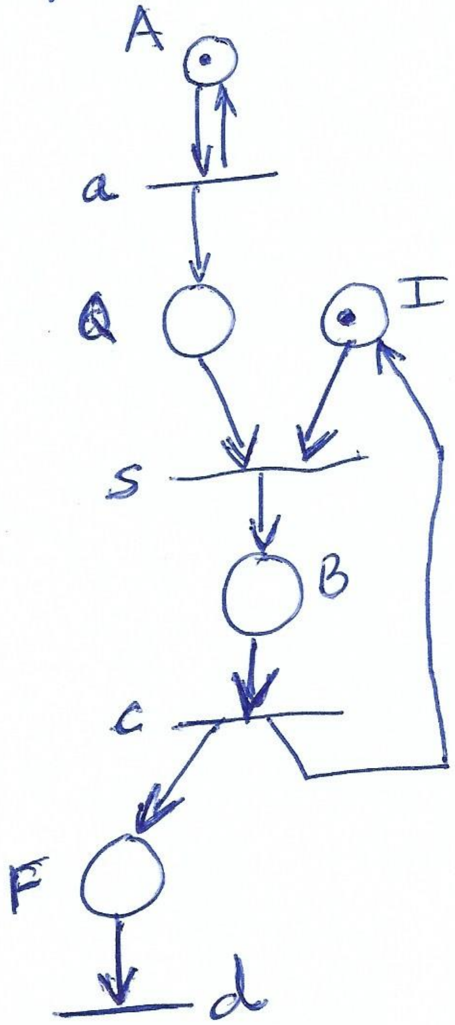
Redes de Petri para Representar Sistemas de Fila

Eventos:
- a: clientes chegando
- s: inicia-se serviço
- c: serviço completo ou cliente saindo

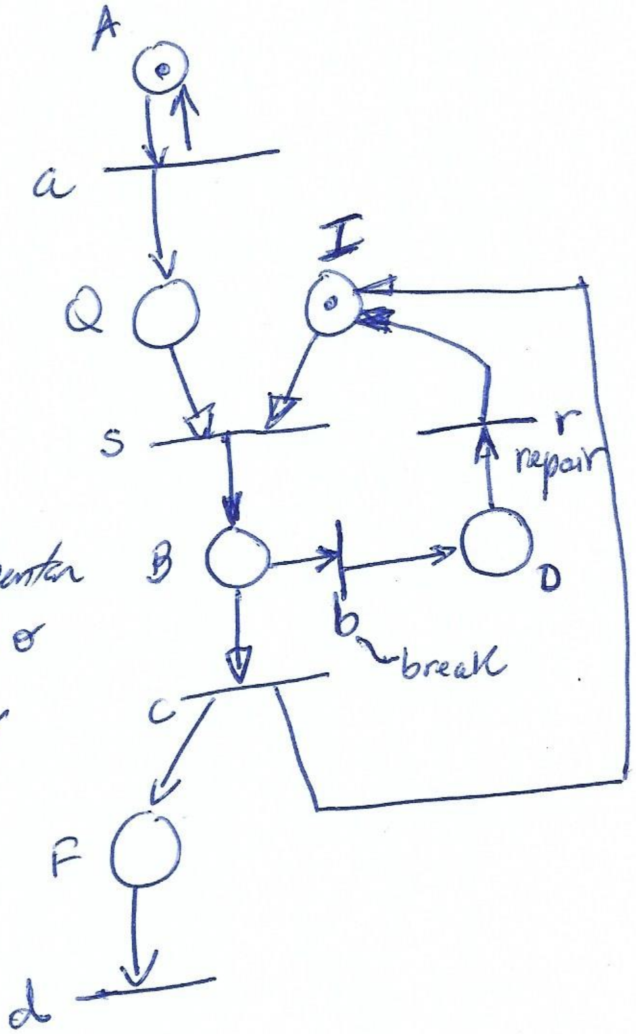




Se for necessário diferenciar entre "serviço completo" e "cliente partindo" a Rede de PETRI pode ser adaptada por:



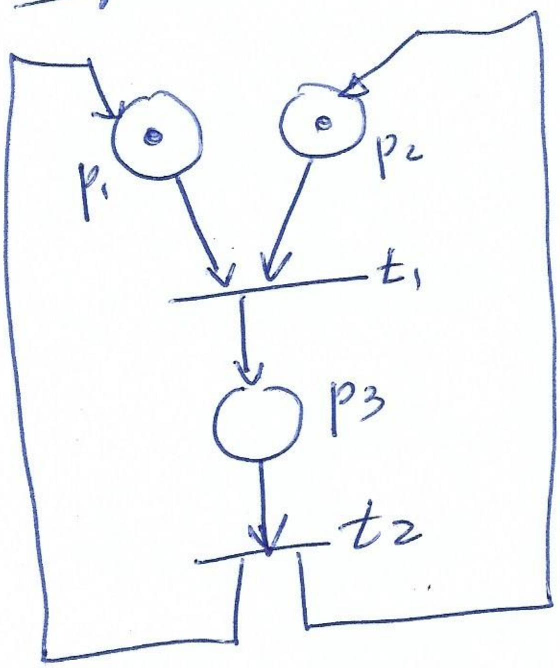
para representar a falha e o reparo do servidor



Árvore de Cobertura ou Grafo de Alcançabilidade

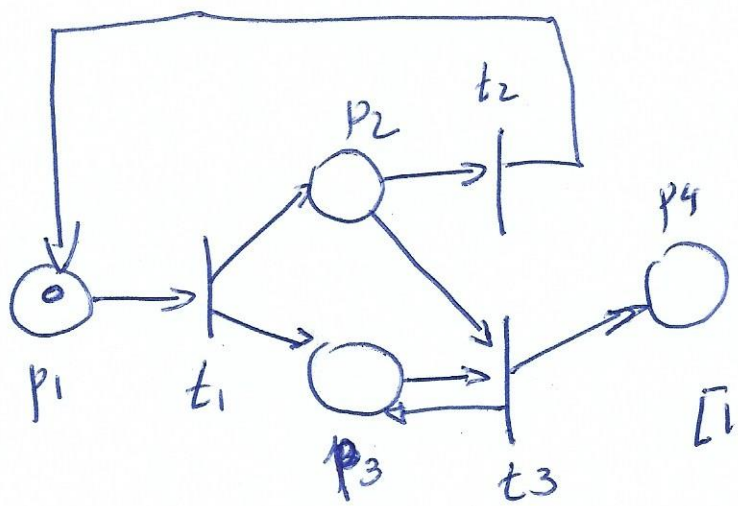
É baseado na construção de uma árvore onde os nós são estados da Rede de Petri e os arcos as transições.

Exemplo 3:



[p1, p2, p3]

$$[1\ 1\ 0] \xrightarrow{t_1} [0\ 0\ 1] \xrightarrow{t_2} [1\ 1\ 0]$$



$$[1\ 0\ 0\ 0] \xrightarrow{t_1} [0\ 1\ 1\ 0]$$

$$[0\ 1\ 1\ 0] \xrightarrow{t_2} [1\ 0\ 1\ 0] \qquad [0\ 1\ 1\ 0] \xrightarrow{t_3} [0\ 0\ 1\ 1]$$

$$[1\ 0\ 1\ 0] \xrightarrow{t_1} [0\ 1\ 2\ 0]$$

$$[0\ 1\ 2\ 0] \xrightarrow{t_2} [1\ 0\ 2\ 0] \qquad [0\ 1\ 2\ 0] \xrightarrow{t_3} [0\ 0\ 2\ 1]$$

Redes de Petri Temporais

(Chapter 5.3 - Cassandras)

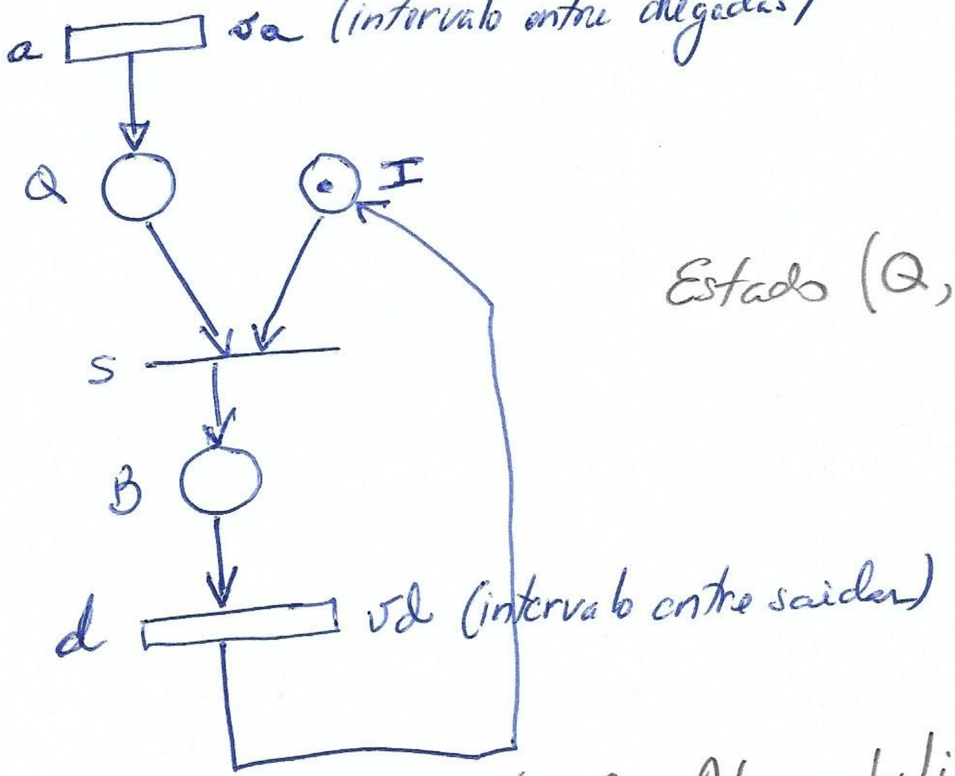
Esta formalização estende o conceito de Redes de Petri incluindo a estrutura temporal.

A única diferença é que uma sequência de clock (v_j) estará associada com uma transição (t_j). Quando a transição t_j estiver habilitada, ela não é disparada imediatamente, mas há um atraso correspondente ao valor (v_j). Durante este atraso, os tokens são conservados nos lugares de entrada da transição t_j.

$\Rightarrow$ Rede de Petri Temporizada: $(\underbrace{P, T, A, w, x}_{\text{Rede de Petri}}, V)$

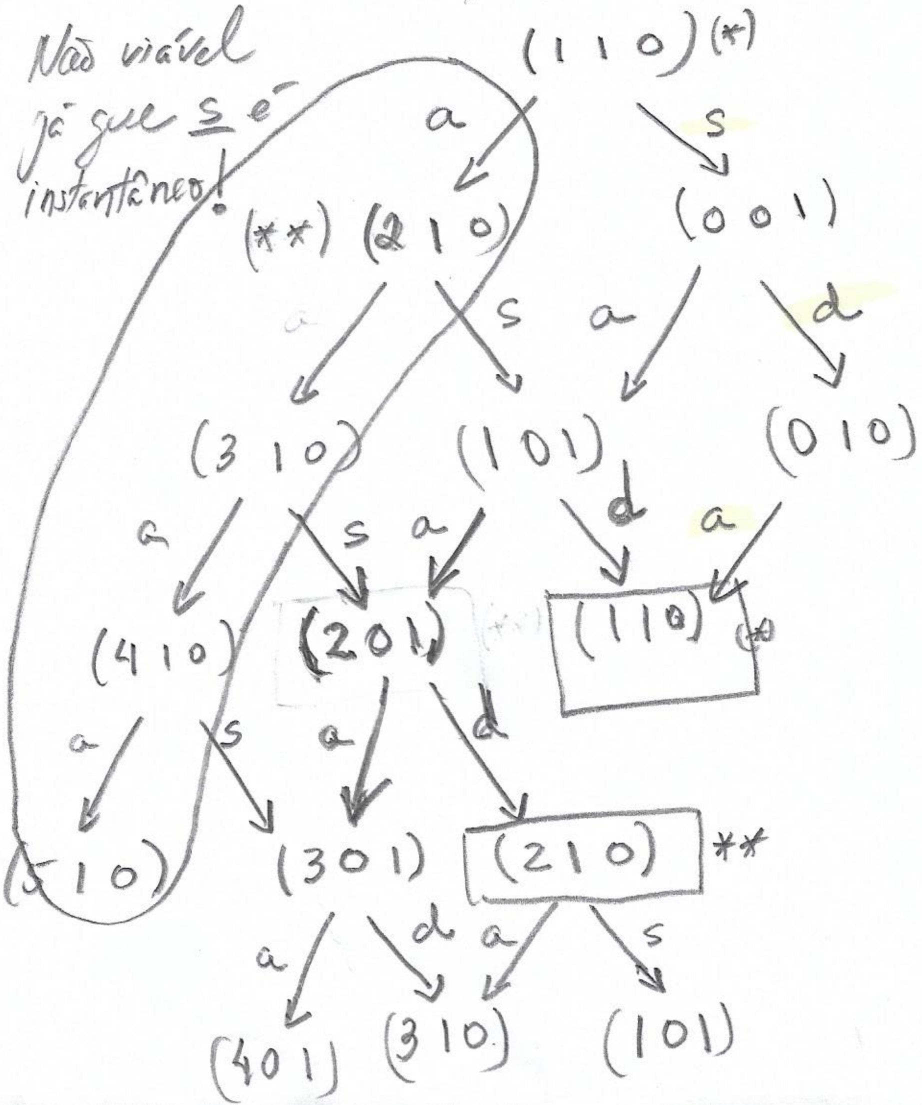
Estrutura de Tempo: $V = \{v_j : t_j \in \underbrace{T_D}_{\text{conjunto de transições com atraso.}}\}$

Sistema de Filas como Redes de Petri Temporizadas



Estado (Q, I, B)

Grafo de Alcançabilidade



$\vartheta_a > \vartheta_d$

equilíbrio!

Árvore de Alcançabilidade e Cadeias de Markov

Redes de Petri Temporizadas e Estocásticas → Árvore de Alcançabilidade
↓
Devido à natureza exponencial das taxas de disparo das transições, pode-se obter uma Cadeia de Markov com tempo Contínuo;

Passos para se obter a Cadeia de Markov Associada:

1) O espaço de estado da Cadeia de Markov corresponde à Árvore de Alcançabilidade da Rede de Petri com marcação inicial So

2/ A taxa de mudança do estado i para o estado j (q_{ij}) é a soma das taxas de disparo de todas as transições habilitadas no estado i e que provocam uma mudança para o estado j

Solução da Cadeia de Markov Associada

→ Probabilidade em Equilíbrio de cada uma das marcações da Rede de Petri Temporal:

$$\Pi . Q = 0 \quad e \quad \sum_i \Pi i = 1$$

$Q = [q_{ij}]$ é a matriz de <u>taxa instantânea de transições</u>.

$\underline{\Pi}$ é o vetor das probabilidades das marcações (estados) em equilíbrio do sistema.

→ Probabilidade de uma Condição Particular: corresponde à soma das probabilidades de todas as marcações onde essa condição é válida.

→ Valor médio do número de marcas num determinado lugar da rede: é igual à soma do número de marcas <u>no lugar i</u>, em cada uma das marcações da rede, multiplicado pela probabilidade de rede se encontrar naquela marcação: $E[m(P_i)]$

$$E[m(P_i)] = \sum_{i=1}^{K} \underbrace{\begin{pmatrix} \text{nº de marcas no} \\ \underline{\text{lugar } i} \text{ em cada uma} \\ \text{das marcações possíveis} \end{pmatrix}}_{m} \cdot \underbrace{\begin{pmatrix} \text{probabilidade} \\ \text{de rede se} \\ \text{encontrar} \\ \text{naquela} \\ \text{marcação} \end{pmatrix}}_{\Pi_i}$$

→ Probabilidade de Disparo de uma transição t_j numa Rede de Petri: $\underline{\underline{P[\text{disparo de } t_j]}}$

$$P[\text{disparo } t_j] = \underbrace{\sum}_{\substack{\text{todas as marcações} \\ \text{onde } t_j \text{ encontra-se} \\ \text{habilitada}}} \begin{pmatrix} \text{probabilidade} \\ \text{dessas} \\ \text{marcações} \end{pmatrix} \times \left[\frac{\text{taxa de disparo de transição } t_j}{\text{taxa total de disparo de todas as transições habilitadas nessa marcação}} \right]$$

→ Número médio de disparos de uma transição t_j na unidade de tempo:

$$\sum \underbrace{\pi_i}_{\substack{\text{probabilidade} \\ \text{da marcação } (i) \\ \text{onde a transição } t_j \\ \text{encontra-se habilitada}}} \cdot \underbrace{l_j}_{\substack{\text{taxa de disparo} \\ \text{de transição } t_j \text{ correspondente} \\ \text{à marcação } (i)}}$$

Resultado de Little

Considerando que marcas não são perdidas numa Rede de Petri, pode-se relacionar o número médio de marcas, numa dada região da Rede de Petri, com o tempo médio que essas marcas levam para percorrer essa região, desde a sua chegada/entrada até a sua saída dessa região.

O parâmetro que regula essa relação é a taxa média de entrada das marcas nessa região da Rede de Petri.

$\Downarrow$

Número Médio de Marcas na região r = Taxa Média de Entrada de Marcas na região r * Tempo Médio Gasto pelas Marcas para passar pela região r

$$\boxed{E[mr] = \lambda \cdot E[Tr]}$$

Resultado de Little.

Plataform Independent Petri Net Editor - PIPE

O programa PIPE é um Editor de Redes de Petri com capacidade de resolução analítica e de simulação dessas redes.

Ele possui uma interface gráfica que facilita a entrada dos modelos em Redes de Petri.

Este programa tem por finalidade desenhar de forma gráfica, Redes de Petri que depois podem ser simuladas ou analisadas por meio da resolução das cadeias de Markov associadas.

O PIPE opera no ambiente Java e é gratuito, fruto de um desenvolvimento colaborativo principalmente entre a Politécnico de Milão e o Imperial College da Inglaterra.