Redes de Petri:
Modelagem, Análise Qualitativa e de Desempenho

Por
Paulo Maciel
Centro de Informática
Universidade Federal de Pernambuco

Objetivo

- Compreender os conceitos fundamentais da teoria das redes de Petri.
- Modelar problemas de natureza concorrente, não determinísticas, assíncronas etc.
- Conhecer os métodos de análise de propriedades qualitativas.
- Estudar os modelos temporizados.
- Modelar problemas considerando aspectos temporais.
- Analisar e efetuar estimativa considerando os modelos temporais.
- Aprofundamento de conhecimento em área de interesse dos participantes (facilitar).

Apresentacão

- Definições
- Classificação
- Conceitos Básicos
- Modelos Semânticos
- Redes Elementares
- Modelagem de Problemas Clássicos
- Classes
- Linguagem PN
- Propriedades
  - Comportamentais
  - Estruturais
- Análise Qualitativa
  - Grafo de Marcações
  - Equação de Estados
    - Invariantes
  - Reduções
- Modelagem e Análise de Problemas

Apresentacão

- Classificação dos Modelos Temporizados
- Transition Time PN
  - Modelo Semântico
  - Grafo de Estado
- Transition Timed PN
  - Modelo Semântico
  - Grafo de Estado
- Modelagem
- Análise Quantitativa
- Revisao sobre Cadeia de Markov (MC)
- SPN
  - Grafo de estado (MC)
- GSPN
  - Grafo de Estado
  - Cadeia de Markov Embutida
- DSPN
- Modelagem e Análise de Desempenho

Apresentacão

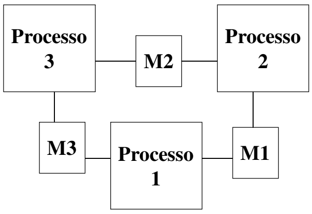
- Modelos de Alto Nível
  - Introdução à CPN
- Seminários em Áreas de Interesse
- Bibliografias:
  - **Lectures on Petri Nets (Advances in Petri Nets) - Spring Verlag, Editado por W. Reisig, G Rozenberg**
  - Aplications of Petri nets in Manufacturing Systems. IEEE press. A. Desrochers, R. Al-Jaar,
  - Pratice of Petri Nets in Manufacturing. Chapman & Hall. Dicesare et al.
  - Hardware Design and Petri Nets. Kluwer. Editado por A Yakovlev et al.
  - Performance Modelling with Deterministic and Stochastic Petri nets. Wiley. C. Lidemann.
  - Performance Modelling of Automated Manufacturing Systems. Prentice Hall. Viswanadham and Narahari
  - Petri net Theory and the Modelling of Systems. Prentice Hall. Peterson.
  - **Introdução às Redes de Petri e Aplicações. SBC. P. Maciel. R. D. Lins e P. Cunha.**
  - **Artigos diversos**

Motivação

- Considere uma situação onde se deseja representar de forma precisa o comportamento de um sistema de manufatura, responsável pela fabricação de três tipos de produtos diferentes.
- A realização das atividades de manufatura de cada produto é denominada um processo. Estes processos podem ser executado paralelamente.
- O ambiente de manufatura disponibiliza três máquinas (recursos) para realização das atividades dos processos.
- Cada par de processos compartilha entre si uma máquina.
- E cada processo precisa simultaneamente de duas máquinas para realização de uma determinada atividade.

Motivação

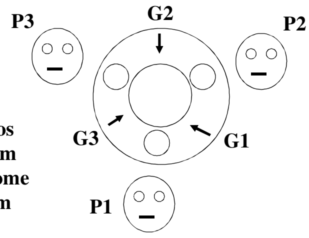
- Estrutura do Problema



Motivação

- O Problema Jantar dos Filósofos

Filósofos
• Pensam
•Tem fome
•Comem



Como especificar adequadamente este problema de forma que o modelo obtido não trave (*deadlock*) e que todos os filósofos tenham oportunidade de comer ?

Classificação dos Modelos

- Espaço de Estados - SS = (S, T, Δ, S_0)
  - S - Conjunto de Estados
  - T - Conjunto de Transições Estruturais
    - Uma transição estrutural pode ser uma ação atômica de uma linguagem de programação.
  - $\Delta \subseteq S \times T \times S$ - Relação de Próximo Estado.
    - Modela uma mudança de estado do sistema
  - S_0 - Conjunto de Estados Iniciais

Espaço de Estados

- SS = (S, T, ✎, S_0)
  - S - Conjunto de Estados
  - T - Conjunto de Transições Estruturais
    - Uma transição estrutural pode ser uma ação atômica de uma linguagem de programação.
  - ✎ - Conjunto de Transições Semânticas que satisfaz a relação ♋ S ✋ T ✋ S .
    - Uma transição semântica modela uma mudança de estado do sistema
  - S_0 - Conjunto de Estados Iniciais

Abstração de Estados e Transições

- ⌘ - Conjunto de proposições atômicas
  - Uma proposição ➘ é uma função de S no conjunto {Verdadeiro, Falso}. ➘: S → {V,F}
- ⊠ - Conjunto de ações observáveis (Alfabeto)
- ⇖ - Símbolo que representa uma ação não-observável

Abstração de Estados e Transições

- Os valores dos observáveis, num determinado espaço de estado

SS = (S, T, ✎, S_0), são definido por duas funções de avaliação:

- $🡅_{⌘}$: S → $2^{⌘}$ atribui a cada estado s ♑S um conjunto de proposições $🡅_{⌘}$(s) ♋ ⌘.
  - Por exemplo, $➘_1$(s)=V significa que a proposição $➘_1$ é verificada no estado s.
- $🡅_{⊠}$: T → ⊠ ☠ {⇖} dar nome as transições estruturais.

Classificação dos Modelos

- Modelos Baseados em Estado
  – Consideram apenas o conjunto S para modelar e se referir as propriedades do sistema.
  – Maioria das lógicas temporais: LTL
- Modelos Baseados em Ações
  – Consideram apenas o conjunto T para modelar e se referir as propriedades dos sistemas.
  – As álgebras de processos: CCS, CSP, COSY, FSP
- Modelos Mistos
  – Consideram ambos os conjuntos S e T.
  – Redes de Petri

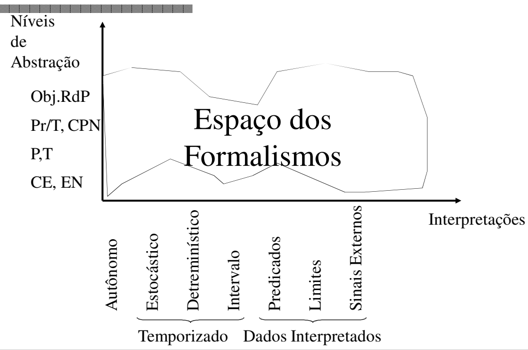
Redes de Petri

- Família de técnicas de descrição formal
- Inicialmente proposta por Carl A. Petri, na Universidade de Darmstadt, 1962 - Alemanha
  – *Kommunikation mit Automaten*

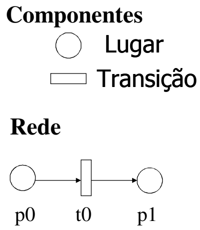
Redes de Petri

- Áreas de Aplicação:
  – Concorrência
  – Arquitetura de Computadores
  – Protocolo de Redes
  – Sistemas Operacionais
  – Sistemas de Produção
  – Sistemas Digitais
  – *Hardware/Software Co-design*
  – Engenharia de Software
  – Sistemas de Tempo Real
  – Modelagem e Avaliação de Desempenho
  – Diagnóstico de Falhas
  – Controle de Tráfego
  – *Workflow*
  – Administração
  – Química
  – etc

Redes de Petri

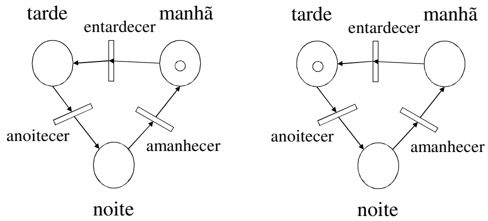


Redes de Petri

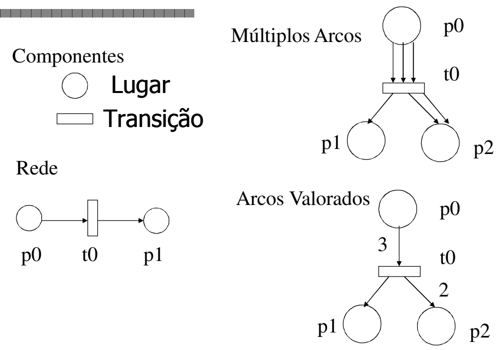


Redes de Petri

- Períodos do Dia

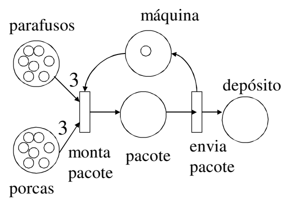


Redes de Petri



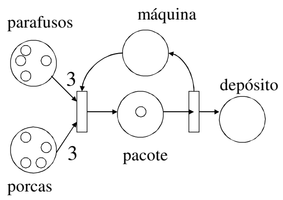
Redes de Petri

- Linha de Produção



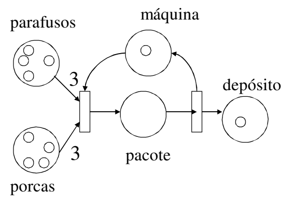
Redes de Petri

- Linha de Produção



Redes de Petri

- Linha de Produção



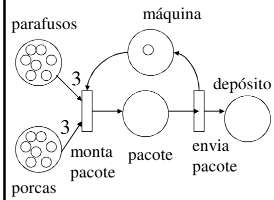
Redes de Petri

- Definição: *Place/Transition Nets* - Teoria *Bag* (multiconjuntos)

- $N=(P,T,I,O,M_o)$
  - P - Conjunto de Lugares - P={p0, ..., pn}
  - T - Conjunto de transições - T={t0, ..., tm}
  - I - Conjunto de *bags* de entrada - I: $T \rightarrow P^{\infty}$
  - O - Conjunto de *bags* de saída - O: $T \rightarrow P^{\infty}$
  - M_0- Vetor marcação inicial - M_0:$P \rightarrow N$

Redes de Petri

- Linha de Produção



$M_0=|7,7,0,1,0|$

- $R_{LP}=(P,T,I,O,M_0)$

P={parafusos, porcas, pacote, máquina depósito}

T={monta_pacote,envia_pacote}

I={I(monta_pacote),I(envia_pacote)}

O={O(monta_pacote),O(envia_pacote)}

I(monta_pacote)=[parafusos, parafusos, parafusos, porcas, porcas, porcas, máquina], I(envia_pacote)=[pacote]

O(monta_pacote)=[pacote]

O(envia_pacote)=[máquina,depósito]

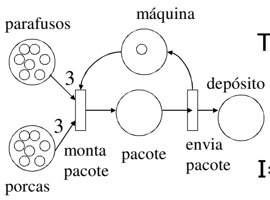
Redes de Petri

- Definição: Place/Transition Nets - Teoria Matricial

⌫ $N=(P,T,I,O,M_o)$
- P - Conjunto de Lugares - P={p0, ..., pn}
- T - Conjunto de transições - T={t0, ..., tm}
- I - Matriz de entrada - I: $T \times P \rightarrow N$
- O - Matriz de saída - O: $T \times P \rightarrow N$
- M_0- Marcação inicial - M_0:$P \rightarrow N$

Redes de Petri

- Linha de Produção



- $R_{LP}=(P,T,I,O,M_0)$

P={parafusos, porcas, pacote, máquina depósito}

T={monta_pacote,envia_pacote}

$$I=\begin{array}{cc} m\_p & e\_p \\ \left|\begin{array}{cc} 3 & 0 \\ 3 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{array}\right| \end{array} \qquad O=\begin{array}{cc} m\_p & e\_p \\ \left|\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{array}\right| \end{array} \begin{array}{l} \\ \text{parafusos} \\ \text{porcas} \\ \text{máquina} \\ \text{depósito} \\ \text{pacote} \end{array}$$

$M_0=|7,7,1,0,0|$

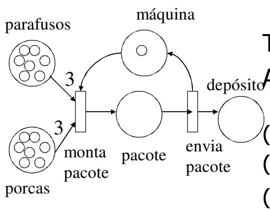
Redes de Petri

- Definição: Place/Transition Nets - Relação de Fluxo

⌫ $N=(P,T,A,V,M_o)$
- P - Conjunto de Lugares - Estados locais
- T - Conjunto de transições - Ações
- A - Arcos - $A \subseteq (P \times T) \cup (T \times P)$
- V - Valoração - V: $A \rightarrow N$
- M_0- Marcação inicial - M_0:$P \rightarrow N$
  - Seja $X = P \cup T$
  - ${}^{\bullet}x=\{y \in Y \mid (y,x) \in A\}$ - Conjunto de entrada
  - $x^{\bullet}=\{y \in Y \mid (x,y) \in A\}$ - Conjunto de saída

Redes de Petri

- Linha de Produção



- $R_{LP}=(P,T,A,V,M_0)$

P={parafusos, porcas, pacote, máquina depósito}

T={monta_pacote,envia_pacote}

A={(parafusos,monta_pacote),(porcas, monta_pacote),(monta_pacote,pacote), (pacote,envia_pacote), (envia_pacote,máquina), (envia_pacote,depósito)}

V=|3,3,1,1,1,1|

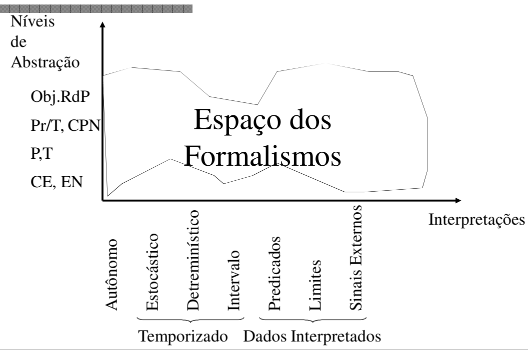
M_0=|7,7,0,1,0|

Redes de Petri

- **Classificação**

Níveis de Abstração	Modelo Representativo
Fundamental	Elementary Net System Condition/Event Net
Intermediário	Place/Transition Net
Alto Nível	CPN, Predicate/Transition Nets

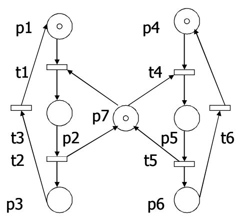
Redes de Petri



Redes de Petri

- P={p_1,p_2,p_3,p_4,p_5,p_6,p_7}
- T={t_1,t_2,t_3,t_4,t_5,t_6}

$$I=\begin{vmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\\1&0&0&1&0&0\end{vmatrix} \quad O=\begin{vmatrix}0&0&1&0&0&0\\1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&0&0&0&1\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&1&0&0&1&0\end{vmatrix}$$



Redes de Petri

Semântica de Disparo de Transição

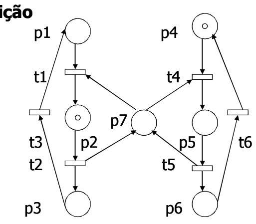
- Regras de habilitação

  M[tj> , M(pi) >=I(pi,tj)
  ∀ pi ∈ P

- Regras de disparo

  Se M[tj>M'

  M'(pi)=M0(pi) - I(pi,tj)+O(pi,tj)
  ∀ pi ∈ P



Redes de Petri

Semântica de Disparo de Transição

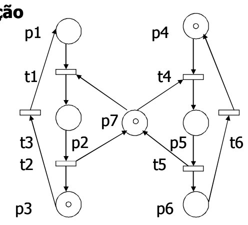
- Regras de habilitação

  M[tj> , M(pi) >=I(pi,tj)
  ∀ pi ∈ P

- Regras de disparo

  Se M[tj>M'

  M'(pi)=M0(pi) - I(pi,tj)+O(pi,tj)
  ∀ pi ∈ P



Redes de Petri

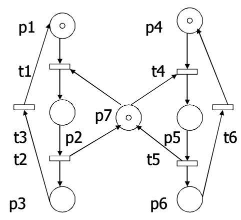
- Regras de habilitação

  M[tj> , M(pi) >=I(pi,tj)
  ∀ pi ∈ P

- Regras de disparo

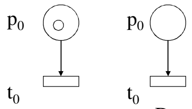
  Se M[tj>M'

  M'(pi)=M0(pi) - I(pi,tj)+O(pi,tj)
  ∀ pi ∈ P



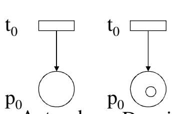
Redes de Petri

- Transição *Sink*
- Transição *Source*



Antes do disparo

Depois do disparo

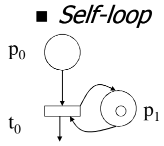


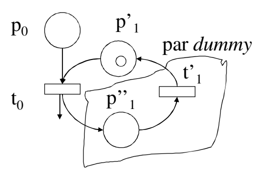
Antes do disparo

Depois do disparo

Redes de Petri

- *Self-loop*





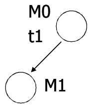
- Rede Pura

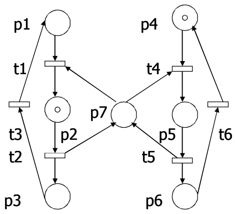
R=(P,T,I,O,) sse $I(p_j,t_i) \times O(p_j,t_i) = 0$, $\forall t_i \in T$, $\forall p_j \in P$

Ou seja, rede sem *self-loop*

Redes de Petri

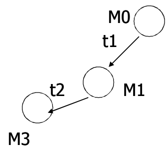
•**Grafo de Marcações Acessíveis (Alcançáveis)**

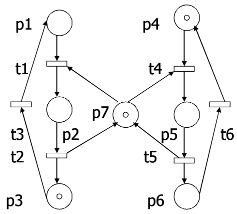




Redes de Petri

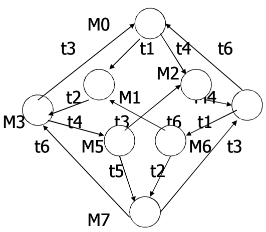
•**Grafo de Marcações Acessíveis (Alcançáveis)**

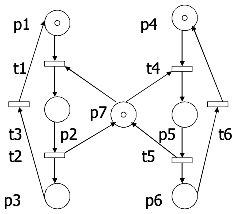




Redes de Petri

•**Grafo de Marcações Acessíveis (Alcançáveis)**





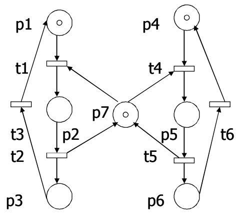
Redes de Petri

Se M[tj>M'

M'(pi)=M0(pi) - I(pi,tj)+O(pi,tj)
$\forall$ pi $\in$ P
M'(pi)=M0(pi) - I.s(tj)^T+O.s(tj) ^{T}
$\forall$ pi $\in$ P
M'(pi)=M0(pi) +(O - I).s(tj)^T
$\forall$ pi $\in$ P

Para uma seqüência sq=t0,t1,...tk

M'(pi)=M0(pi) +(O - I).[s(t0)^T+...+ s(tk)^T]
$\forall$ pi $\in$ P
M'(pi)=M0(pi) +C.$\overline{S}$, $\forall$ pi $\in$ P



Redes de Petri

- Equação de Estado
  (Eq. Fundamental)

Se M[tj>M'

M'(pi)=M0(pi) - I(pi,tj)+O(pi,tj)
$\forall$ pi $\in$ P
M'(pi)=M0(pi) - I.s(tj)^T+O.s(tj) ^{T}
$\forall$ pi $\in$ P
M'(pi)=M0(pi) +(O - I).s(tj)^T
$\forall$ pi $\in$ P

Para uma seqüência sq=t0,t1,...tk

M'(pi)=M0(pi) +(O - I). [s(t0)^T+ ...+ s(tk)^T], $\forall$ pi $\in$ P

Equação de Estado

$\boxed{\text{M'(pi)=M0(pi) +C.}\overline{\text{S}}\text{, } \forall \text{ pi} \in \text{P}}$

Vetor Característico (V. de Parikh)

$\overline{S}$ = [s(t0)^T+ ...+ s(tk)^T]

Redes de Petri

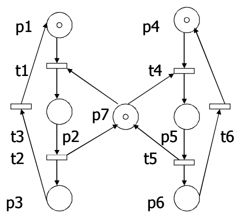
- Equação de Estados

M'(pi) = M_0(pi) + C $\times$ $\overline{s}$, $\forall$ pi $\in$ P

Matriz de Incidência

C = O - I

$$\begin{vmatrix} m1 \\ m2 \\ m3 \\ m4 \\ m5 \\ m6 \\ m7 \end{vmatrix} \begin{matrix} = \\ = \\ = \\ = \\ = \\ = \\ = \end{matrix} \begin{vmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{vmatrix} + \begin{vmatrix} -1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{vmatrix}$$



Redes de Petri

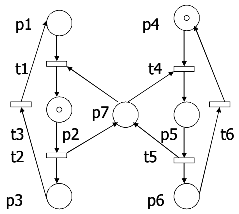
- Equação de Estados

$M'(pi) = M_0(pi) + C \times s, \forall\ pi \in P$

✍Matriz de Incidência

C = O - I

m1	=	0
m2	=	1
m3	=	0
m4	=	1
m5	=	0
m6	=	0
m7	=	0



Redes de Petri

Semântica de Passo Simples (*simple step*)

Step $s_j \subseteq T$

- Regras de habilitação

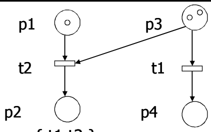
$M[s_j>$

$M(pi) \geq |O(pi) \cap s_j|, \forall\ pi \in P$

- Regras de disparo

$M[s_j>M'$

$M'(pi)=M0(pi) - |O(pi) \cap s_j| + |I(pi) \cap s_j|, \quad \forall\ pi \in P$



s = { t1,t2 }

O(p1) = {t2} ≥ |{t2} ∩ { t1,t2 }| = 1

M(p1) = 1

O(p3) = {t1,t2} ≥ |{t1,t2} ∩ { t1,t2 }| = 2

M(p3) = 2

Portanto s é disparável

Redes de Petri

Semântica de Passo Simples (*simple step*)

Step $s_j \subseteq T$

- Regras de habilitação

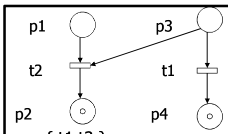
$M[s_j >$

$M(pi) \geq |O(pi) \cap s_j|, \forall\ pi \in P$

- Regras de disparo

M[sj>M'

$M'(pi)=M0(pi) - |O(pi) \cap s_j| + |I(pi) \cap s_j|, \forall\ pi \in P$



s = { t1,t2 }

O(p1) = {t2} ≥ |{t2} ∩ { t1,t2 }| = 1

M(p1) = 1

O(p3) = {t1,t2} ≥ |{t1,t2} ∩ { t1,t2 }| = 2

M(p3) = 2

Portanto, s é disparável

Redes de Petri

Semântica de Passo (*step*)

Step s: $T \rightarrow \aleph$

- Regras de habilitação **s**

$M[s_j>$

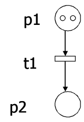
$M(pi) \geq \sum_{\forall t \in O(pi)} s(t), \forall\ pi \in P$

- Regras de disparo

M[sj>M'

M'(pi)=M0(pi)

- $\sum_{\forall t \in O(pi)} s(t)$
- $+ \sum_{\forall t \in I(pi)} s(t), \forall\ pi \in P$



O(p1)={t1} O(p2)=∅

s=s(t1) = 2 $\sum_{t1 \in O(p2)} s(t1) = 0$

$\sum_{t1 \in O(p1)} s(t1) = 2$

M(p1) = 2 M(p2) = 0

Portanto, s é disparável

Redes de Petri

Semântica de Passo (*step*)

Step s: $T \rightarrow \aleph$

- Regras de habilitação **s**

$M[s_j>$

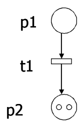
$M(pi) \geq \sum_{\forall t \in O(pi)} s(t), \forall\ pi \in P$

- Regras de disparo

M[sj>M'

M'(pi)=M0(pi)

- $\sum_{\forall t \in O(pi)} s(t)$
- $+ \sum_{\forall t \in I(pi)} s(t), \forall\ pi \in P$



O(p1)={t1} O(p2)=∅

s=s(t1) = 2 $\sum_{t1 \in O(p2)} s(t1) = 0$

$\sum_{t1 \in O(p1)} s(t1) = 2$

M(p1) = 2 M(p2) = 0

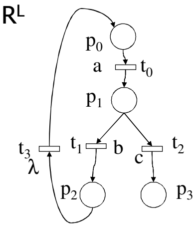
Portanto, s é disparável

Associando Rótulos as Transições

- Alfabeto. Um alfabeto é conjunto de símbolos Σ que representa um conjunto de eventos.
- Ação Muda (*silent action*). λ representa uma ação muda.
- *Kleene-Closure.* Seja Σ um alfabeto. Σ^* é o conjunto de todas as *strings* finitas formadas por elementos de Σ. Σ^* é definido por *Kleene-Closure.*
- Função de Nomeação. Seja uma rede R=(P,T,I,O) e um alfabeto Σ. A função σ: $T \rightarrow \Sigma$ é definida como função de nomeação.

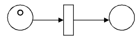
Associando Rótulos as Transições

- Rede Rotulada. Seja uma rede R=(P,T,I,O). A rede rotulada $R^L=(P,T,I,O,\Sigma,\sigma)$, onde Σ é um alfabeto e σ a função de nomeação.
- $\Sigma=\{\lambda,a,b,c\}$
- $\Sigma^*=\{\lambda,a,b,c,aa,ab,ac,...\}$
- R^L

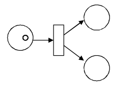


Redes Básicas

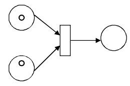
- Sequenciamento



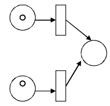
- Distribuição



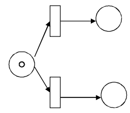
- Junção



- *Merging*

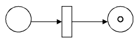


- Escolha

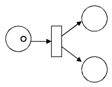


Redes Básicas

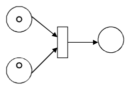
- Sequenciamento



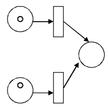
- Distribuição



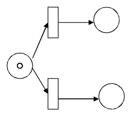
- Junção



- *Merging*

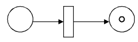


- Escolha

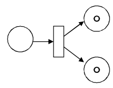


Redes Básicas

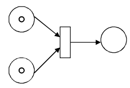
- Sequenciamento



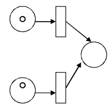
- Distribuição



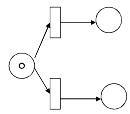
- Junção



- *Merging*

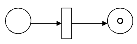


- Escolha

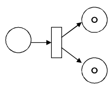


Redes Básicas

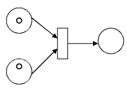
- Sequenciamento



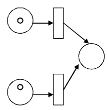
- Distribuição



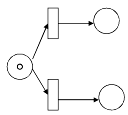
- Junção



- *Merging*

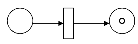


- Escolha

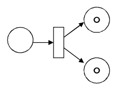


Redes Básicas

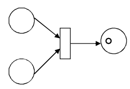
- Sequenciamento



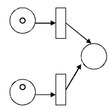
- Distribuição



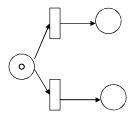
- Junção



- *Merging*

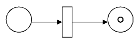


- Escolha

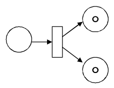


Redes Básicas

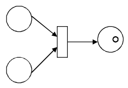
- Sequenciamento



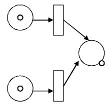
- Distribuição



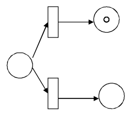
- Junção



- *Merging*

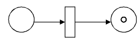


- Escolha

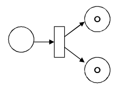


Redes Básicas

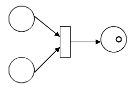
- Sequenciamento



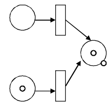
- Distribuição



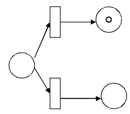
- Junção



- *Merging*

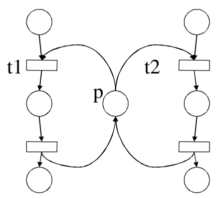


- Escolha



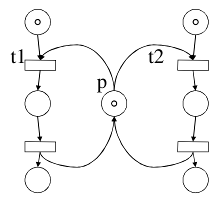
Conflito Estrutural

- N=(P,T,I,O), t1, t2 $\in$ T estão em conflito estrutural sse $\exists$p $\in$ P tal que I(p,t1) $\times$ I(p,t2) $\neq$ 0



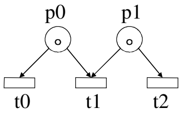
Conflito Efetivo

- N=(P,T,I,O,M_0), se t1, t2 $\in$ T estão em conflito efetivo para M se estão em conflito estrutual e M[t1>, M[t2> e M(p) < I(p,t1) + I(p,t2)



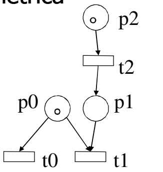
Confusão

- Simétrica



t0 e t2 são concorrentes, no entanto estão em conflito com t1. O disparo de t1 impossibilita o disparo de t0 e t2.

- Assimétrica



t0 e t2 são concorrentes, no entanto se t2 dispara primeiro t0 e t1 estarão em conflito efetivo.

Redes de Petri

Semântica de Disparo de Transição

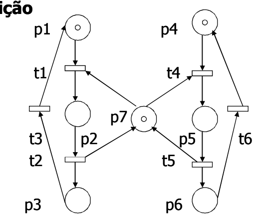
- Regras de habilitação

M[tj> ,   M(pi) >=I(pi,tj) $\forall$ pi $\in$ P

- Regras de disparo

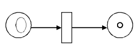
Se M[tj>M'

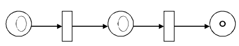
M'(pi)=M0(pi) - I(pi,tj)+O(pi,tj) $\forall$ pi $\in$ P



Fluxo de Controle

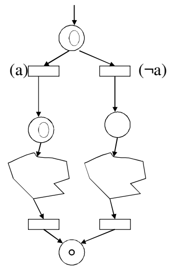
- Computação Simples
- Seqüenciamento





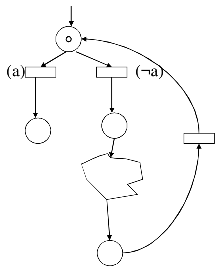
Fluxo de Controle

- If-then-else



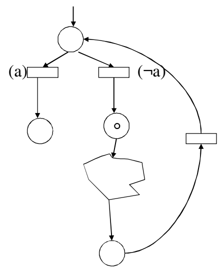
Fluxo de Controle

- While-do



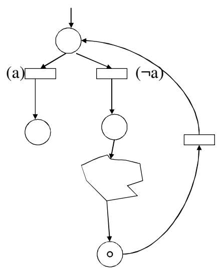
Fluxo de Controle

- While-do



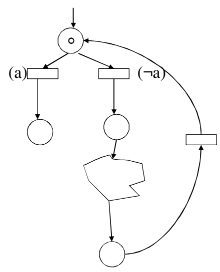
Fluxo de Controle

- While-do



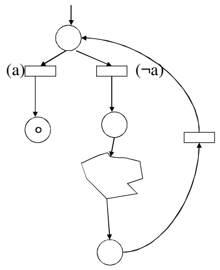
Fluxo de Controle

- While-do



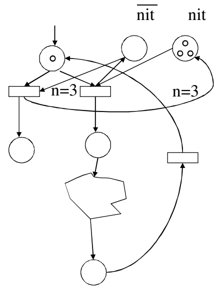
Fluxo de Controle

- While-do



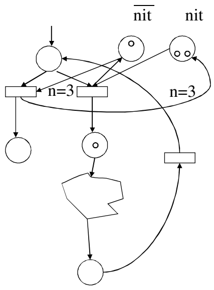
Fluxo de Controle

- For-do



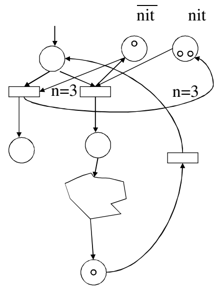
Fluxo de Controle

- For-do



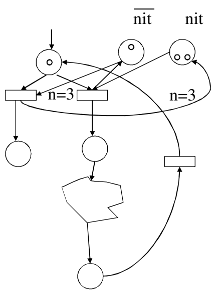
Fluxo de Controle

- For-do



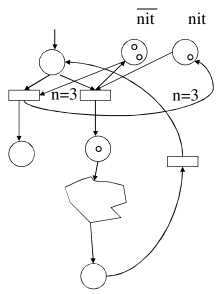
Fluxo de Controle

- For-do



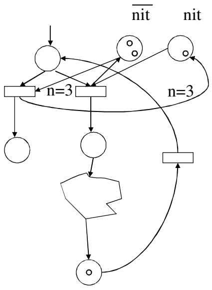
Fluxo de Controle

- For-do



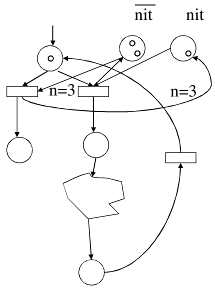
Fluxo de Controle

- For-do



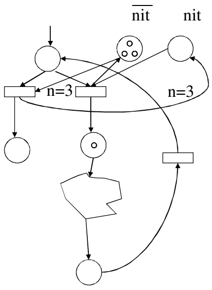
Fluxo de Controle

- For-do



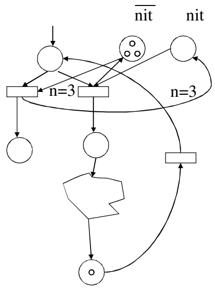
Fluxo de Controle

- For-do



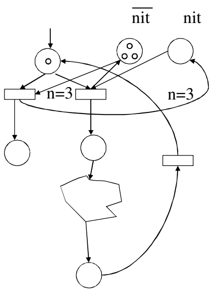
Fluxo de Controle

- For-do



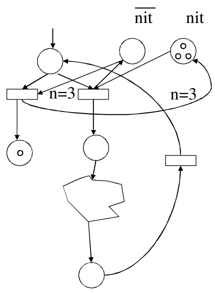
Fluxo de Controle

- For-do



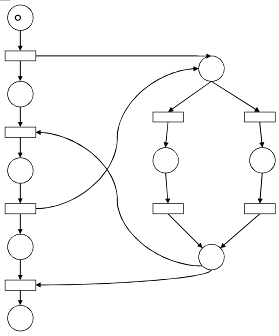
Fluxo de Controle

- For-do



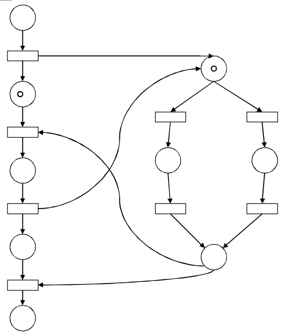
Fluxo de Controle

- Sub-rotina



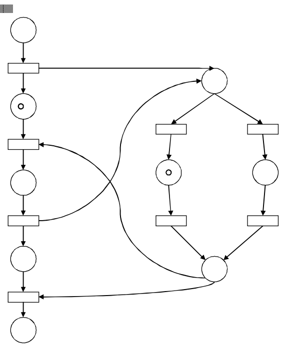
Fluxo de Controle

- Sub-rotina



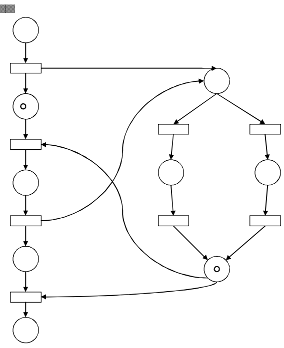
Fluxo de Controle

- Sub-rotina



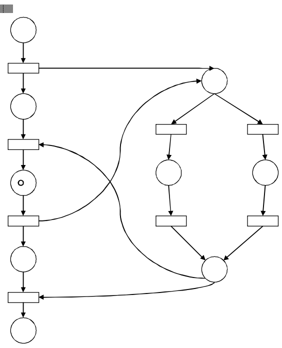
Fluxo de Controle

- Sub-rotina



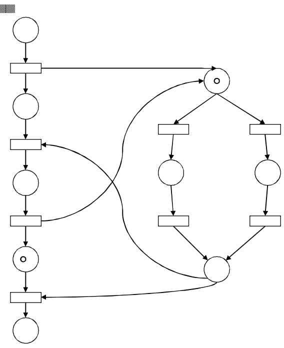
Fluxo de Controle

- Sub-rotina



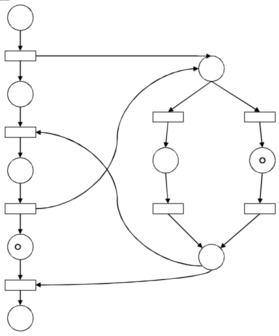
Fluxo de Controle

- Sub-rotina



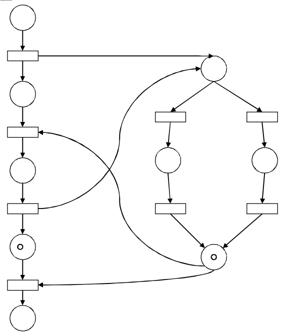
Fluxo de Controle

- Sub-rotina



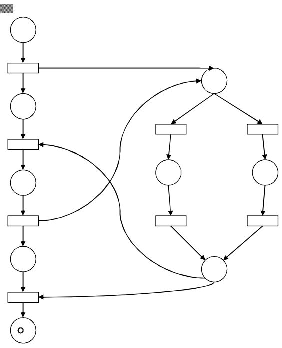
Fluxo de Controle

- Sub-rotina

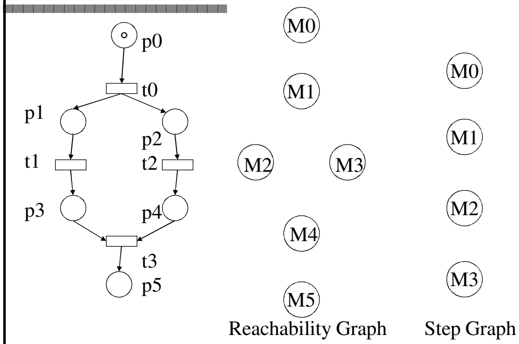


Fluxo de Controle

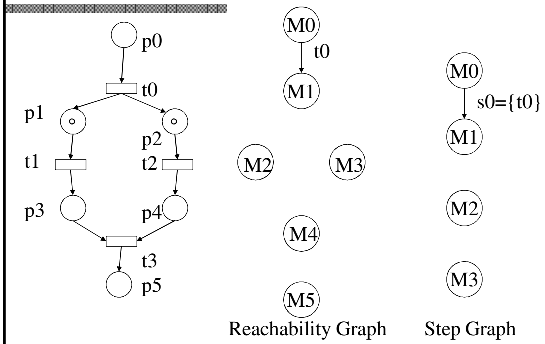
- Sub-rotina



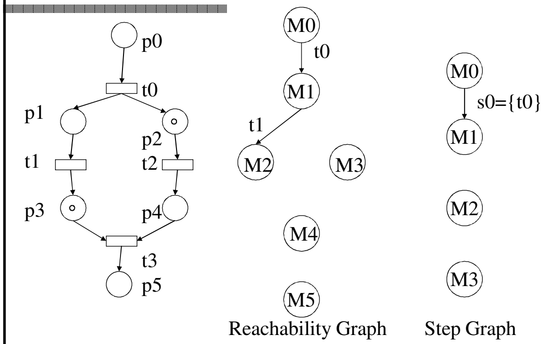
Processos Paralelos



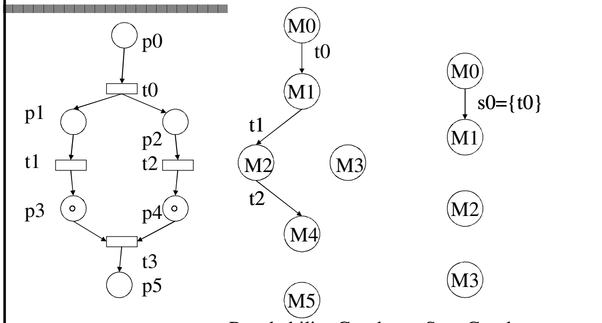
Processos Paralelos



Processos Paralelos

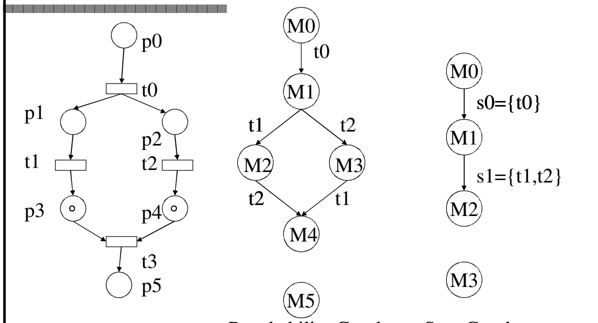


Processos Paralelos



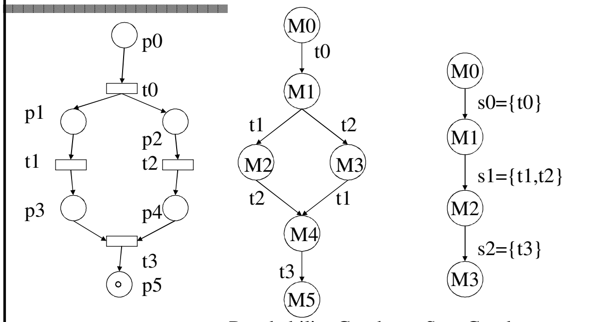
Reachability Graph   Step Graph

Processos Paralelos



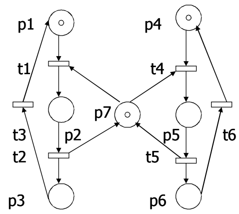
Reachability Graph   Step Graph

Processos Paralelos

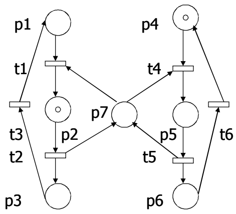


Reachability Graph   Step Graph

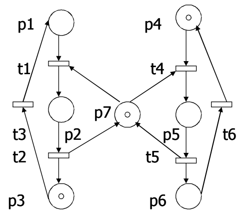
Exclusão Mútua



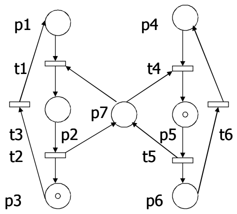
Exclusão Mútua



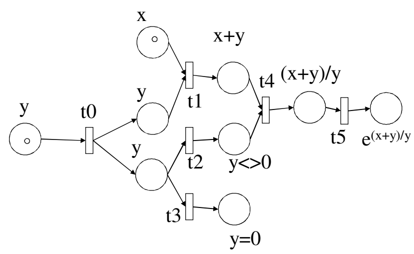
Exclusão Mútua



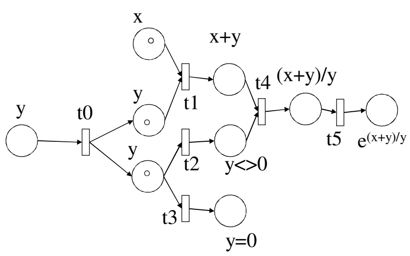
Exclusão Mútua



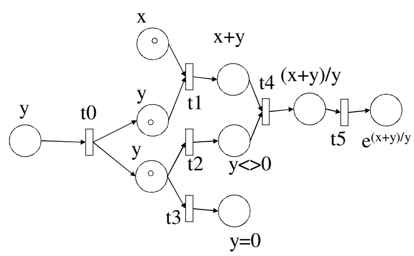
Computação Data-Flow



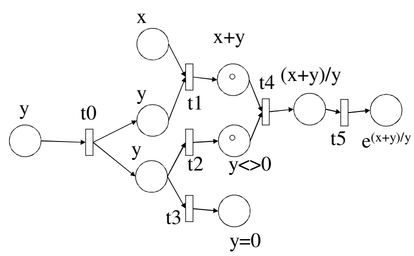
Computação Data-Flow



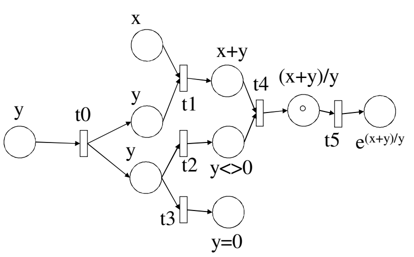
Computação Data-Flow



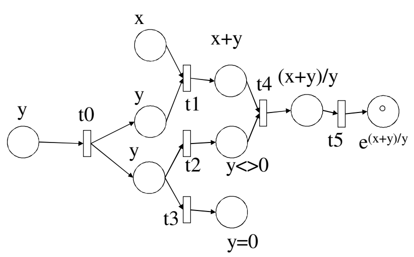
Computação Data-Flow



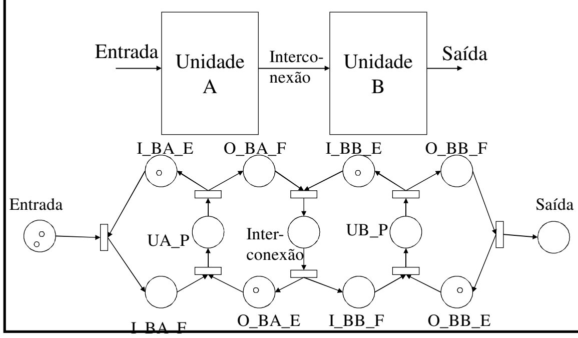
Computação Data-Flow



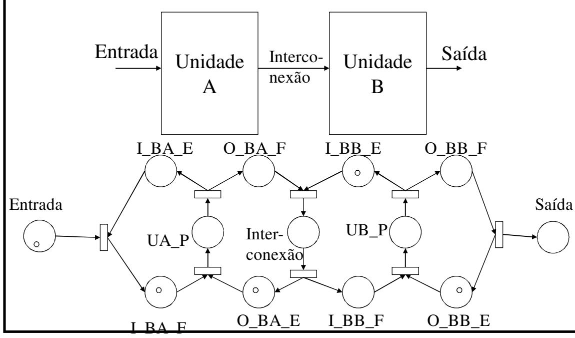
Computação Data-Flow



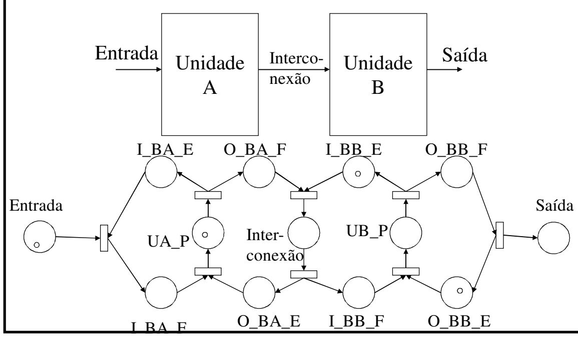
Pipeline



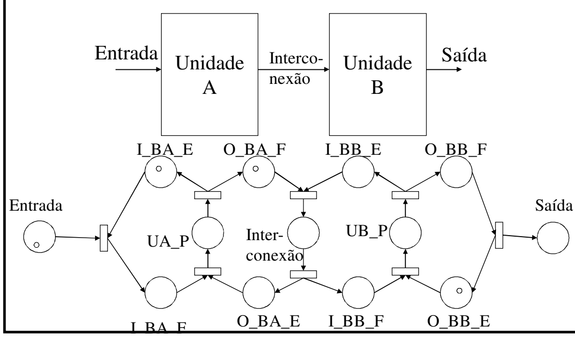
Pipeline



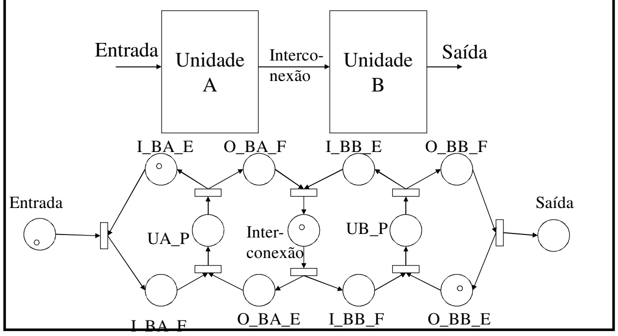
Pipeline



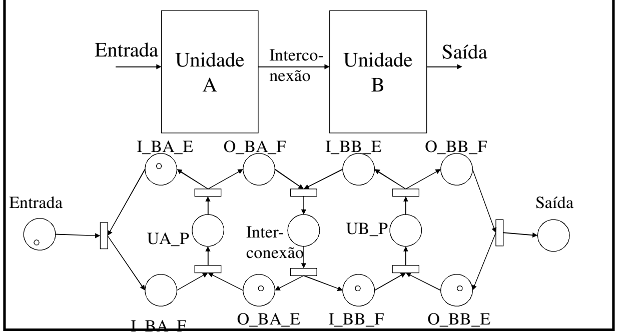
Pipeline



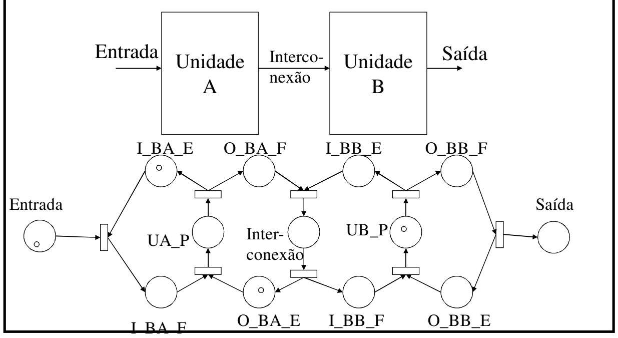
Pipeline



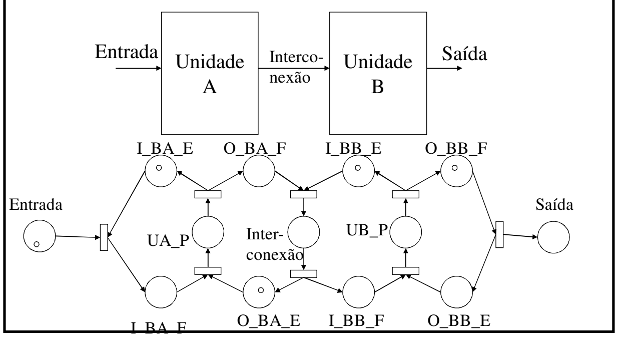
Pipeline



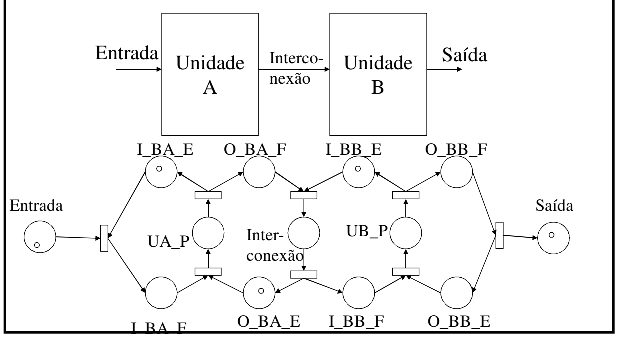
Pipeline



Pipeline

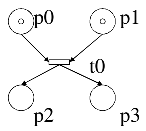


Pipeline



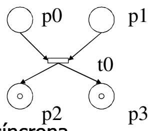
Comunicação

Síncrona

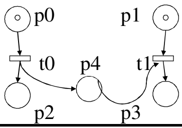


Comunicação

Síncrona

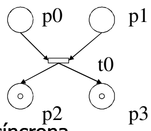


Assíncrona

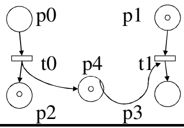


Comunicação

Síncrona

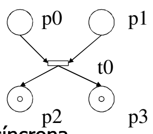


Assíncrona

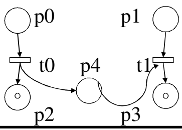


Comunicação

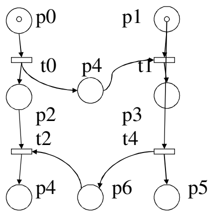
Síncrona



Assíncrona

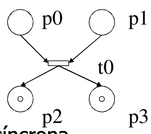


Síncrona (send/ack)

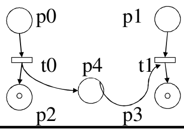


Comunicação

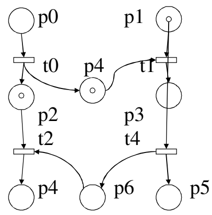
Síncrona



Assíncrona

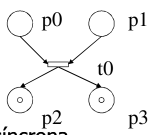


Síncrona (send/ack)

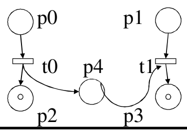


Comunicação

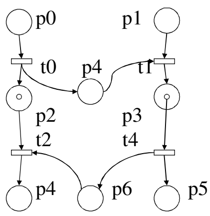
Síncrona



Assíncrona

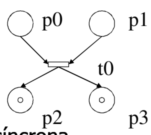


Síncrona (send/ack)

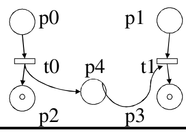


Comunicação

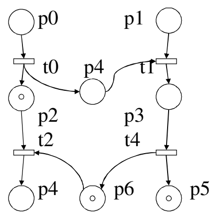
Síncrona



Assíncrona

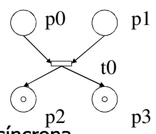


Síncrona (send/ack)

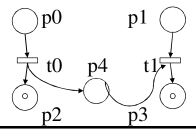


Comunicação

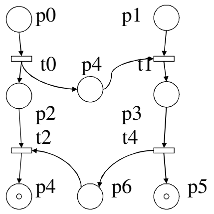
Síncrona



Assíncrona

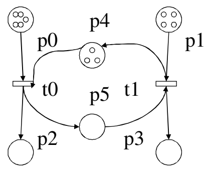


Síncrona (send/ack)



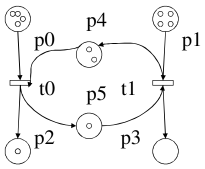
Comunicação

Buffer Limitado



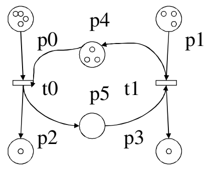
Comunicação

Buffer Limitado



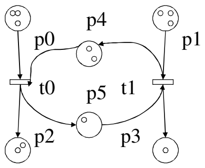
Comunicação

Buffer Limitado



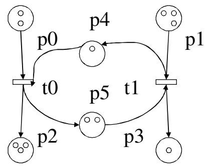
Comunicação

Buffer Limitado



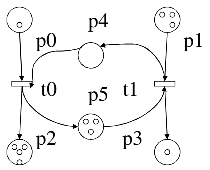
Comunicação

Buffer Limitado



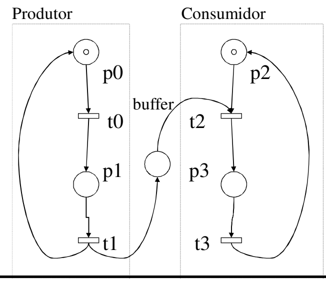
Comunicação

Buffer Limitado



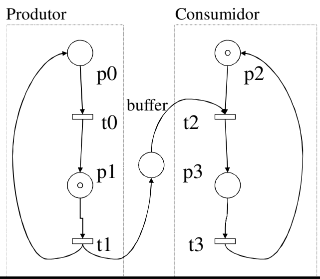
Produtor/Consumidor

Buffer Ilimitado



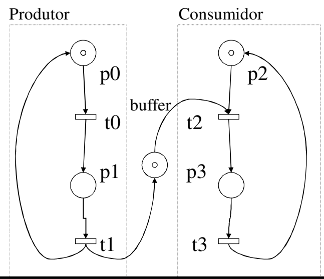
Produtor/Consumidor

Buffer Ilimitado



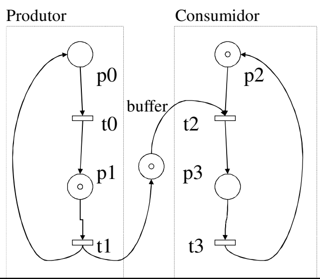
Produtor/Consumidor

Buffer Ilimitado



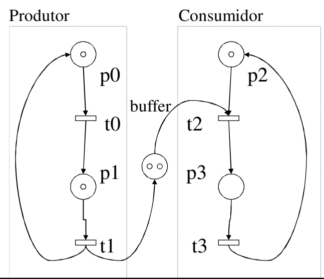
Produtor/Consumidor

Buffer Ilimitado



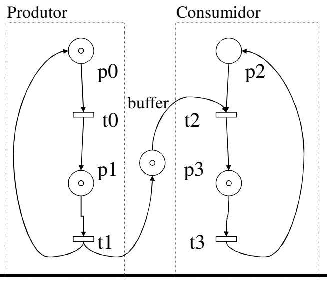
Produtor/Consumidor

Buffer Ilimitado



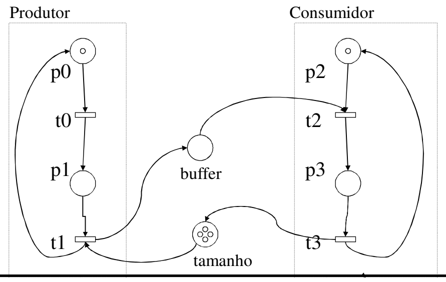
Produtor/Consumidor

Buffer Ilimitado



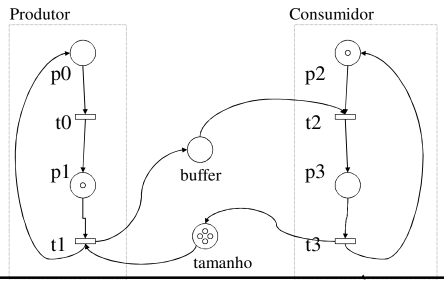
Produtor/Consumidor

Buffer Limitado



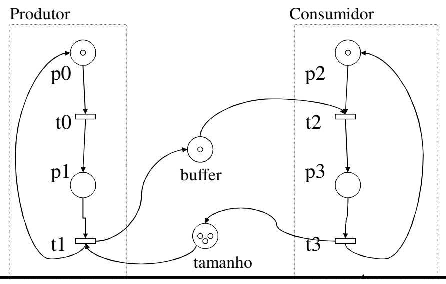
Produtor/Consumidor

Buffer Limitado



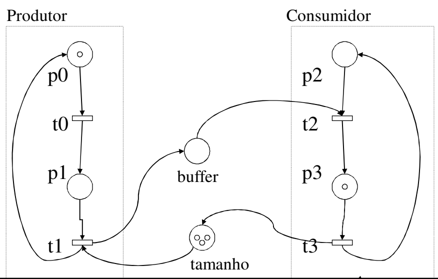
Produtor/Consumidor

Buffer Limitado



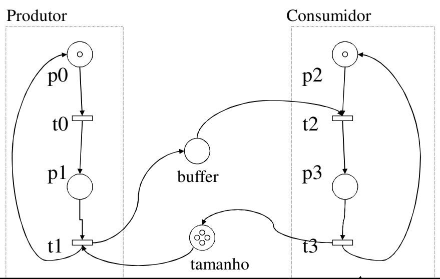
Produtor/Consumidor

Buffer Limitado



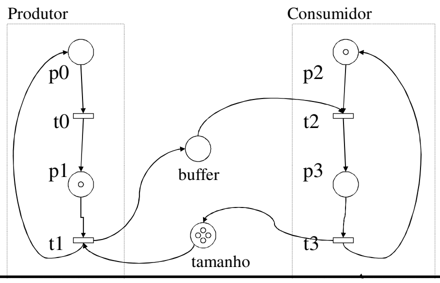
Produtor/Consumidor

Buffer Limitado



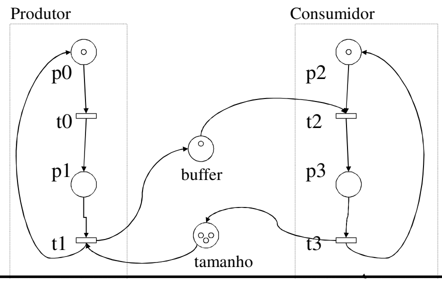
Produtor/Consumidor

Buffer Limitado



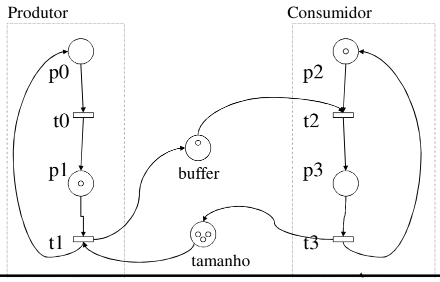
Produtor/Consumidor

Buffer Limitado



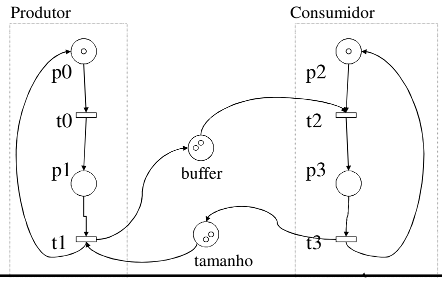
Produtor/Consumidor

Buffer Limitado



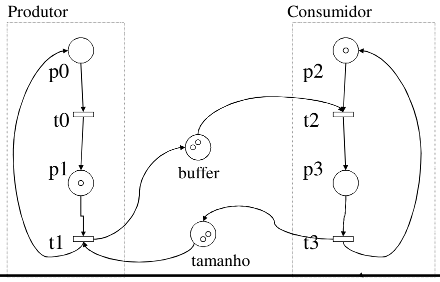
Produtor/Consumidor

Buffer Limitado



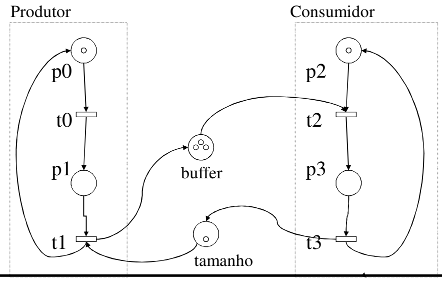
Produtor/Consumidor

Buffer Limitado



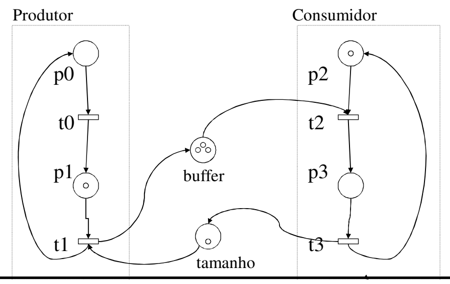
Produtor/Consumidor

Buffer Limitado



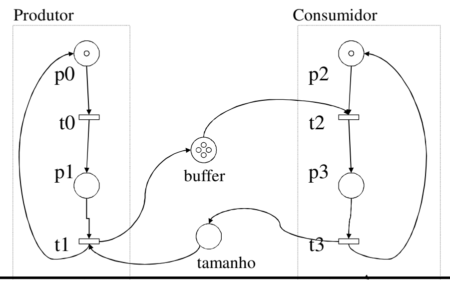
Produtor/Consumidor

Buffer Limitado



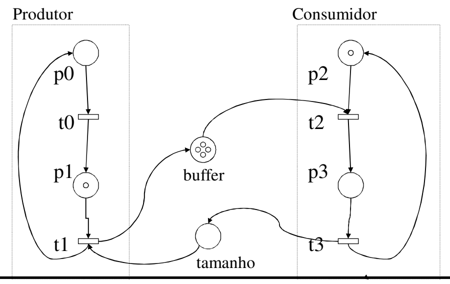
Produtor/Consumidor

Buffer Limitado

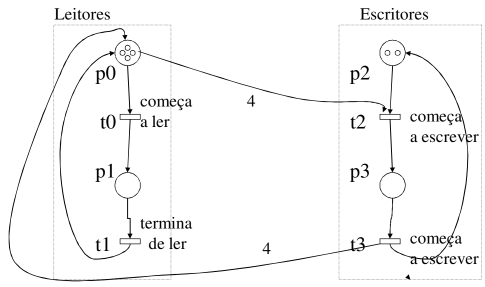


Produtor/Consumidor

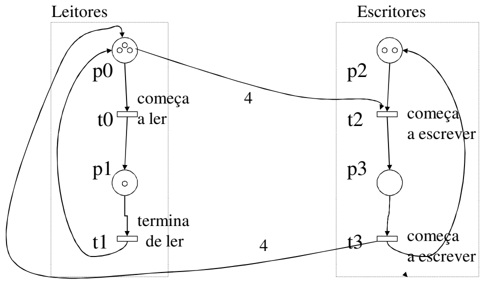
Buffer Limitado



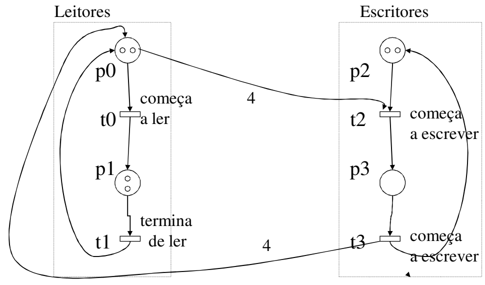
Leitores/Escritores



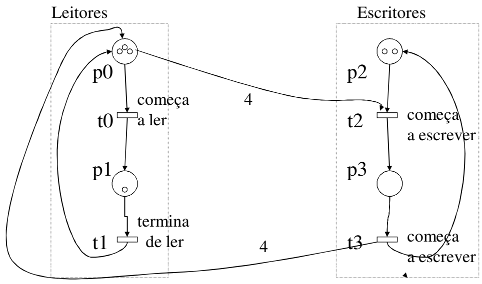
Leitores/Escritores



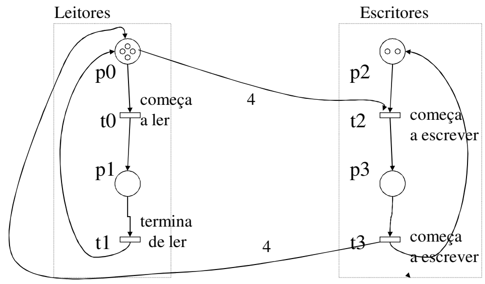
Leitores/Escritores



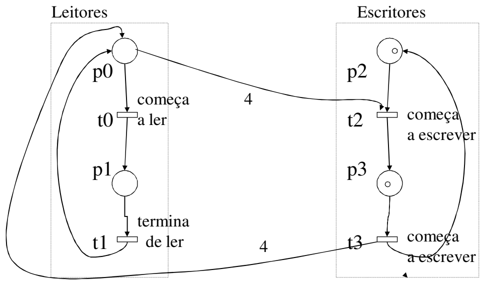
Leitores/Escritores



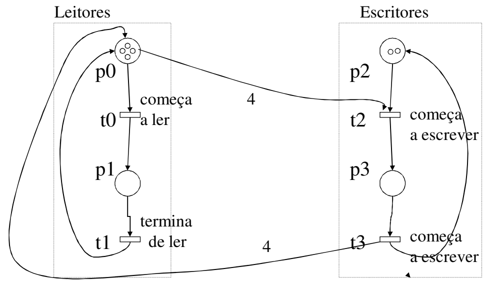
Leitores/Escritores



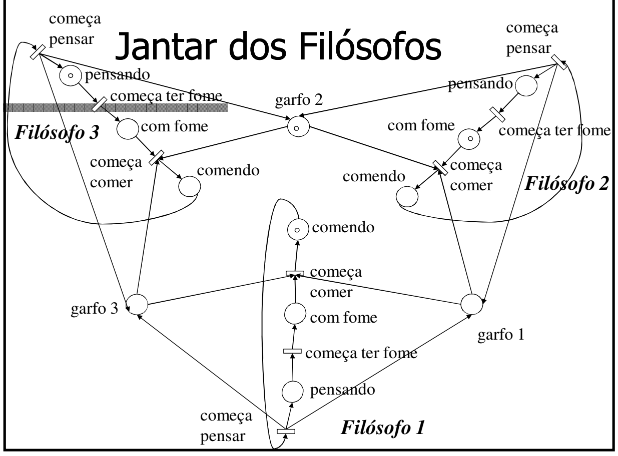
Leitores/Escritores



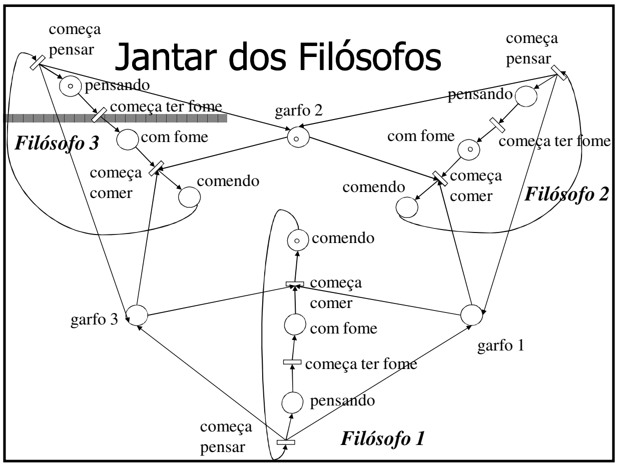
Leitores/Escritores



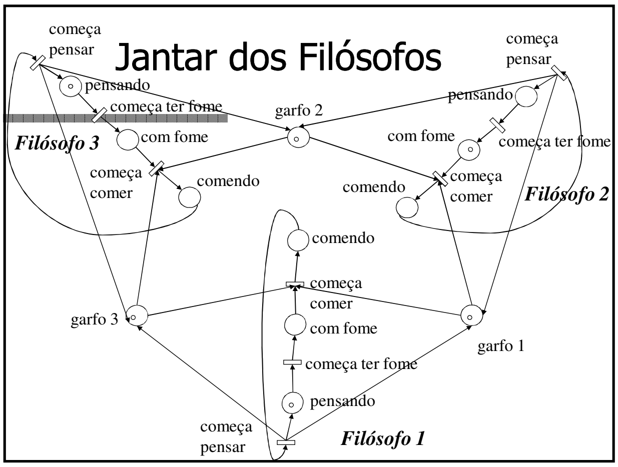
Jantar dos Filósofos



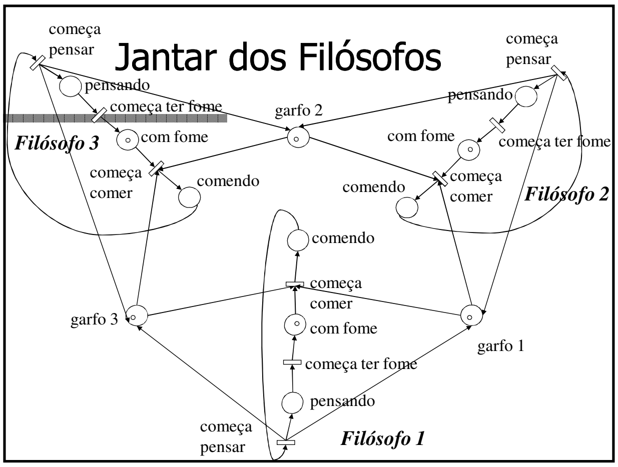
Jantar dos Filósofos



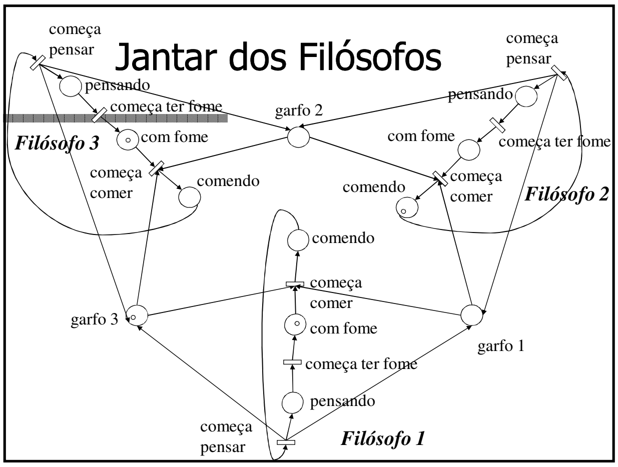
Jantar dos Filósofos



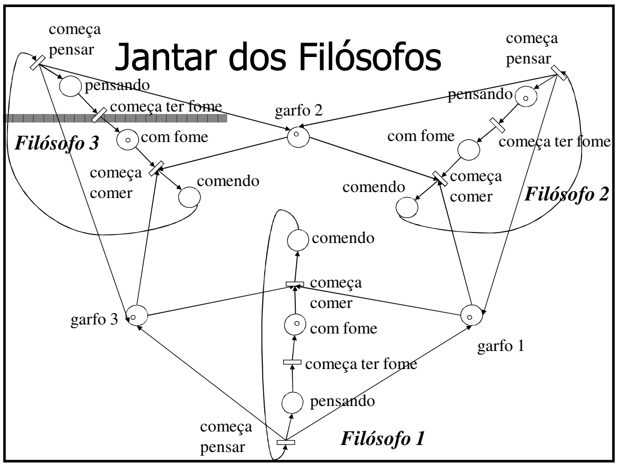
Jantar dos Filósofos



Jantar dos Filósofos



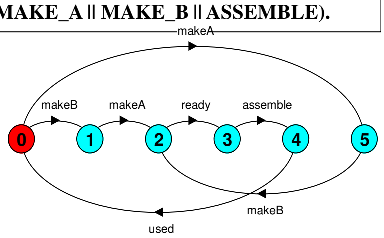
Jantar dos Filósofos



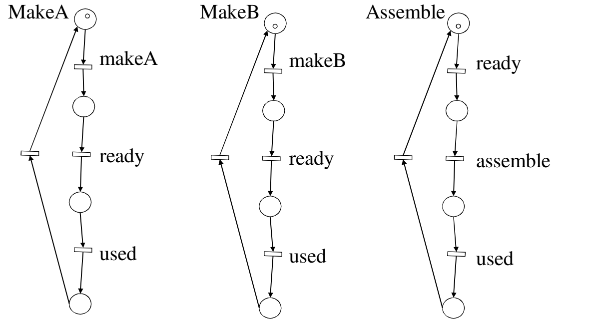
Modelando Interações Múltiplos Processos

MAKE_A = (makeA->ready->used->MAKE_A).
MAKE_B = (makeB->ready->used->MAKE_B).
ASSEMBLE = (ready->assemble->used->ASSEMBLE).

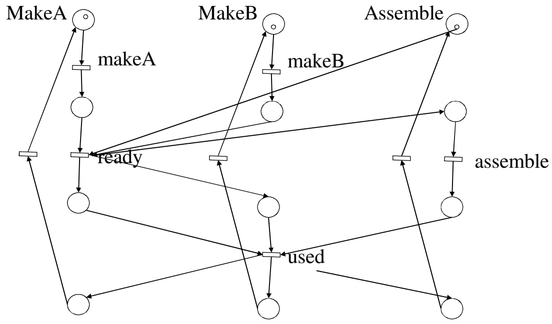
||FACTORY = (MAKE_A || MAKE_B || ASSEMBLE).



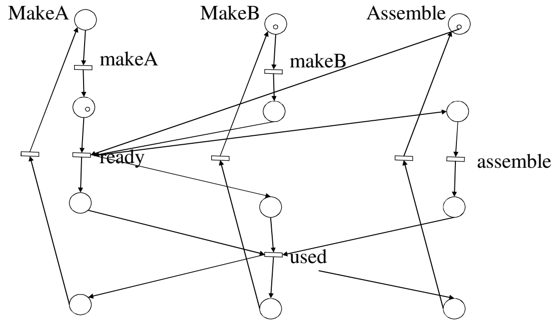
Modelando Interações Múltiplos Processos



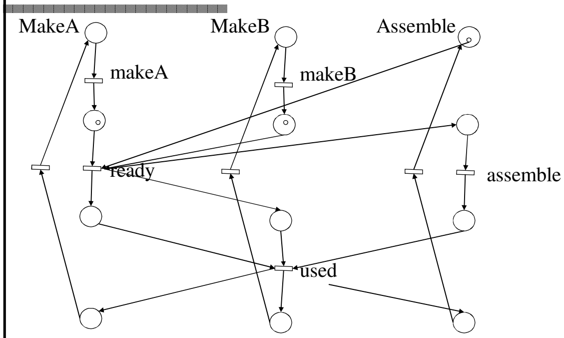
Modelando Interações Múltiplos Processos



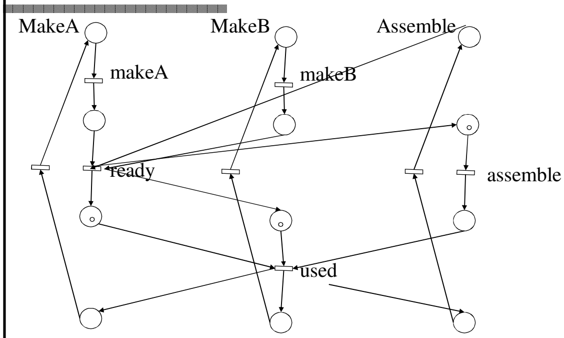
Modelando Interações Múltiplos Processos



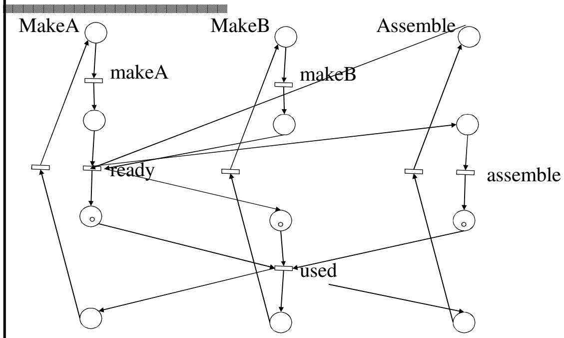
Modelando Interações Múltiplos Processos



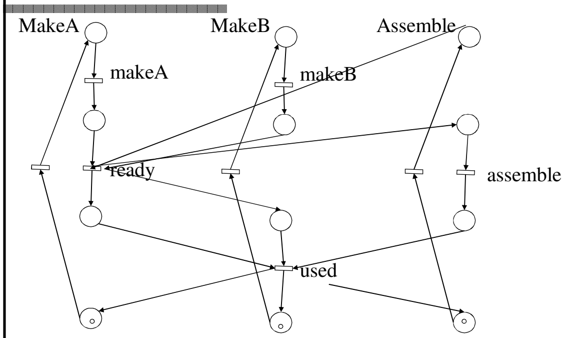
Modelando Interações Múltiplos Processos



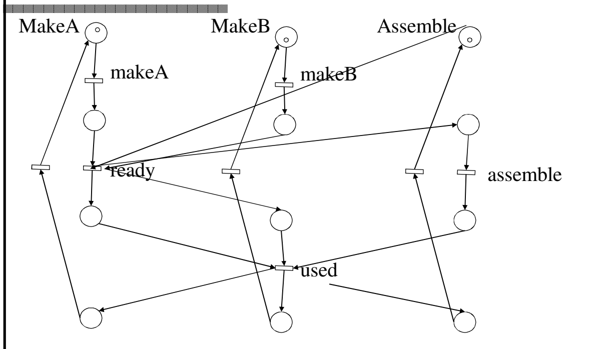
Modelando Interações Múltiplos Processos



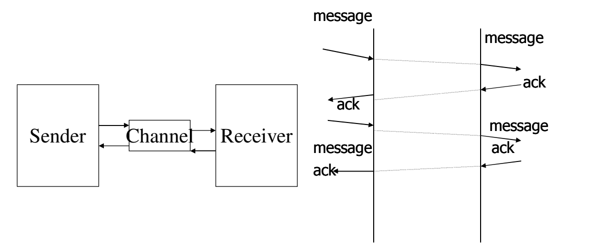
Modelando Interações Múltiplos Processos



Modelando Interações Múltiplos Processos



Fluxo de Controle de um Protocolo Simples



Protocolo 1 Simplex sem restrições

```
void sender1(void)
{
frame s;
packet buffer;
while (true) {
   from_network_layer(&buffer);
   s.info = buffer;
   to_physical_layer(&s);
}
}
```

```
void receiver1(void)
{
frame r;
event_type event;
while(true) {
   wait_for_event(&event);
   from_physical_layer(&r);
   to_network_layer(&r.info);
}
}
```

Protocolo 2 Stop and Wait

```
void sender1(void)
{
frame s;
packet buffer;
event_type event;
while (true) {
   from_network_layer(&buffer);
   s.info = buffer;
   to_physical_layer(&s);
   wait_for_event(&event);
}
}
```

```
void receiver1(void)
{
frame r;
event_type event;
while(true) {
   wait_for_event(&event);
   from_physical_layer(&r);
   to_network_layer(&r.info);
   to_physical_layer(&s);
}
}
```

Protocolo 3 (Retransmissão e Canal não confiável)

```
void sender1(void)
{seq_nr next;_frame_to_send;
frame s; packet buffer;
event_type event;
next_frame_to_send=0;
from_network_layer(&buffer);
while (true) {
   s.info=buffer;
   s.seq= next_frame_to_send;
   to_physical_layer(&s);
   start_timer(s.seq);
   wait_for_event(&event);
   if(event==frame_arrival) {
      from_network_layer(&buffer);
      if(s.ack==next_frame_to_send){
         from_network_layer(&buffer);
         inc(next_frame_to_send);
}}}}
```

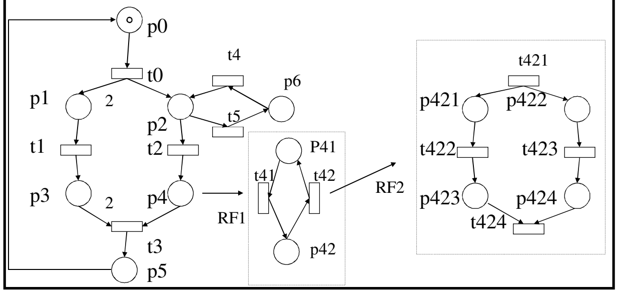
```
void receiver1(void)
{
frame r,s; seq_nr next;
frame_expected;
event_type event;
while(true) {
   wait_for_event(&event);
   if(event==frame_arrival) {
      from_physical_layer(&r);
      if(r.seq==frame_expected){
         to_network_layer(&r.info);
         inc(frame_expected);
      }
      s.ack=1-frame_expected;
      to_physical_layer(&s);
   }
 }
}
```

Abordagem de Modelagem - Princípios -

- Top-down
  - Refinamento
- Botton-up
  - Composição
- Híbrida
  - Refinamento e Composição

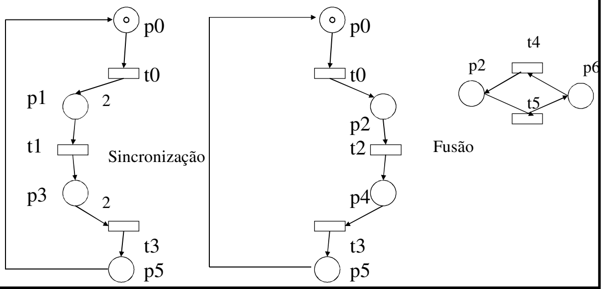
Abordagem de Modelagem
- Princípios -

- Top-down



Abordagem de Modelagem
- Princípios -

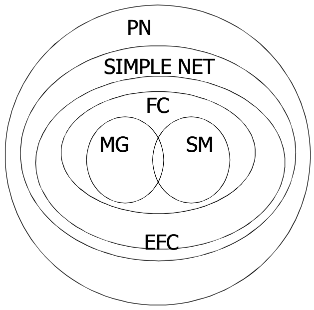
- Botton-up



Abordagem de Modelagem
- Princípios -

- Maiores detalhes serão apresentados após a descrição de propriedades!
  - Liveness, boundedness, safeness, reversibilidade

Sub-Classes



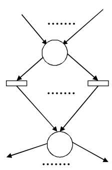
Sub-Classes

- Poder de Modelagem × Poder de Decisão

- Complexidade × Decidibilidade

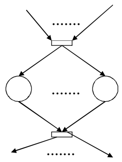
Sub-Classes

- Máquina de Estado (SM)
  - Definição: Seja uma rede R=(P,T,I,O). R é um SM sse $|I(t_i)|=|O(t_i)|=1, \forall\, t_i \in T$.



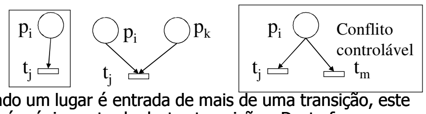
Sub-Classes

- Grafo-Marcado (MG)
  - Definição: Seja uma rede R=(P,T,I,O). R é um MG sse $|I(p_i)|=|O(p_i)|=1$, $\forall\ p_i \in P$.



Sub-Classes

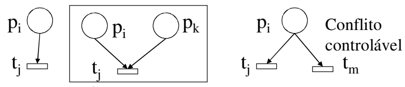
- Escolha-Livre (FC)
  - Definição: Seja uma rede R=(P,T,I,O). R é um FC sse $I(t_j)=\{p_i\}$, $\forall\ t_i \in T$ ou $O(p_i)=\{t_j\}$, $p_i \in P$.



  - Quando um lugar é entrada de mais de uma transição, este lugar é a única entrada destas transicões. Desta forma, todas estas transições estarão habilitada ou não estarão. Possibilita livremente a escolha do evento a ser realizado.

Sub-Classes

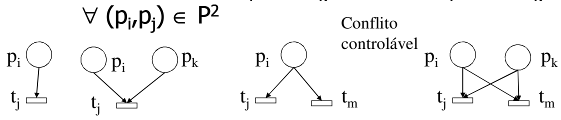
- Escolha-Livre (FC)
  - Definição: Seja uma rede R=(P,T,I,O). R é um FC sse $I(t_j)=\{p_i\}$, $\forall\ t_i \in T$ ou $O(p_i)=\{t_j\}$, $p_i \in P$.

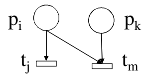


  - Quando um lugar é entrada de mais de uma transição, este lugar é a única entrada destas transicões. Desta forma, todas estas transições estarão habilitada ou não estarão. Possibilita livremente a escolha do evento a ser realizado.

Sub-Classes

- Escolha-Livre Estendida (EFC)
  - Definição: Seja uma rede R=(P,T,I,O). R é um EFC sse $O(p_i) \cap O(p_k) \neq \{\} \Rightarrow O(p_i) = O(p_k)$ $\forall\ (p_i, p_j) \in P^2$

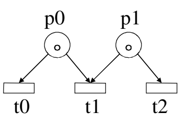




*Redes com esta estrutura não são EFC - Solução do conflito não é livre

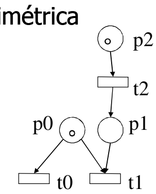
Confusão

- Simétrica



t0 e t2 são concorrentes, no entanto estão em conflito com t1. O disparo de t1 impossibilita o disparo de t0 e t2.

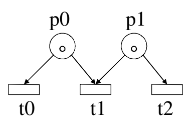
- Assimétrica

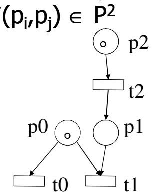


t0 e t2 são concorrentes, no entanto se t2 dispara primeiro t0 e t1 estarão em conflito efetivo.

Sub-Classes

- Redes Simples (*Simple Nets*) (SN)
  - Definição: Seja uma rede R=(P,T,I,O). R é um SN sse $O(p_i) \cap O(p_k) \neq \{\} \Rightarrow O(p_i) \subseteq O(p_k)$ ou $O(p_i) \supseteq O(p_k)$ $\forall (p_i, p_j) \in P^2$





SN

Máquinas de Estados

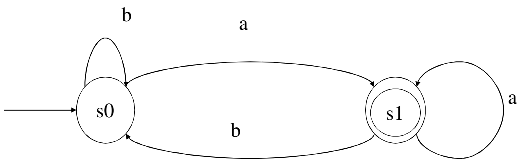
- Máquina de Estados Determinística
- Máquina de Estados Não-Determinística
- Máquina de Estados Finitos Não-Determinísitca
- Máquinas de Estados Finitos Determinística
  – Máquina de Estados Finitos Determinística com Entradas e Saídas

Máquina de Estados Determinística

- SM= $(S, E, f, \Gamma, s_0, S_m)$
  – S – Conjunto de estados
  – $s_0 \in S$– Estado inicial
  – E – Alfabeto (conjunto de eventos)
  – $f : S \times E \rightarrow S$ – Função de próximo estado
  – $\Gamma : S \times E$ – Relação dos eventos factíveis
  – $S_m \subseteq S$ - Conjunto de estados marcados

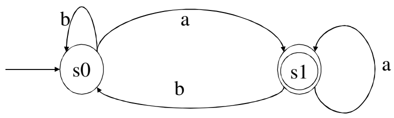
Máquina de Estados Determinística

S={s0,s1}, E={a,b}, f(s0,a) = s1, f(s0,b) = s0, f(s1,a)=s1, f(s1,b)=s0, Γ(s0) = {a,b}, Γ(s1) = {a,b}, S_m={s1}



Máquina de Estados Determinística

- Linguagem Gerada
  L(SM) = {st $\in$ E* | f(s_0, st) é definida}
  se f for uma função total então L(SM) = E*
- Linguagem Marcada
  Lm(SM) = {st $\in$ L(SM) | f(s_0, st) $\in$ S_m}



Lm(SM) = {a, aa, ba, aaa, aba, ...}
L(SM)= E*

Máquina de Estados Determinística

- Duas Máquinas de Estados SM1 e SM2 são ditas equivalentes se

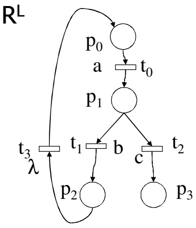
L(SM1) = L(SM2)
Lm(SM1) = Lm(SM2)

Associando Rótulos as Transições

- Alfabeto. Um alfabeto é conjunto de símbolos Σ que representa um conjunto de eventos.
- Ação Muda (*silent action*). λ representa uma ação muda.
- *Kleene-Closure.* Seja Σ um alfabeto. Σ^* é o conjunto de todas as *strings* finitas formadas por elementos de Σ. Σ^* é definido por *Kleene-Closure.*
- Função de Nomeação. Seja uma rede R=(P,T,I,O) e um alfabeto Σ. A função σ: T $\rightarrow$ Σ é definida como função de nomeação.

Associando Rótulos as Transições

- Rede Rotulada. Seja uma rede R=(P,T,I,O). A rede rotulada R^L=(P,T,I,O,Σ,σ), onde Σ é um alfabeto e σ a função de nomeação.
- Σ={λ,a,b,c}
- Σ*={λ,a,b,c,aa,ab,ac,...}
- R^L

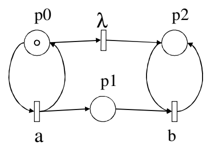


Linguagem Gerada pelas Redes de Petri

- Linguagem. Seja um alfabeto Σ. Uma linguagem $L \subseteq \Sigma^*$ é um conjunto de *strings* de comprimentos finito sobre Σ.
- Linguagem Gerada por RdP. Seja R^L=(P,T,I,O,Σ,σ,M_0) uma rede marcada rotulada. $L(R^L) = \{\sigma(s) \in \Sigma^* \mid s \in T^*$ e $f(M_0,s)$ é definida$\}$. Onde $f: M \times T \rightarrow M$ é a função de transição.

Linguagem Gerada pelas Redes de Petri

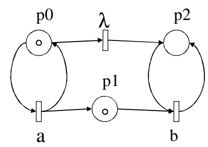
R^L



$L(R^L) = \{$

Linguagem Gerada pelas Redes de Petri

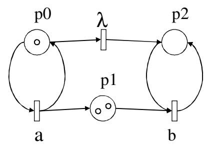
R^L



$L(R^L) = \{a$

Linguagem Gerada pelas Redes de Petri

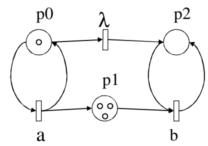
R^L



$L(R^L) = \{a,a,$

Linguagem Gerada pelas Redes de Petri

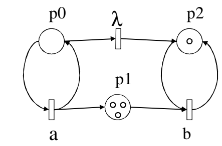
R^L



$L(R^L) = \{a,a,a, ...,$

Linguagem Gerada pelas Redes de Petri

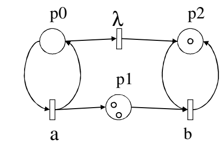
R^L



$L(R^L) = \{a,a,a, ...,$

Linguagem Gerada pelas Redes de Petri

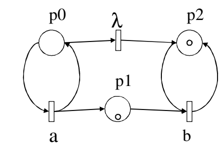
R^L



$L(R^L) = \{a,a,a, ...,b$

Linguagem Gerada pelas Redes de Petri

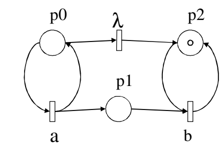
R^L



$L(R^L) = \{a,a,a, ...,b,b$

Linguagem Gerada pelas Redes de Petri

R^L



$L(R^L) = \{a,a,a, ...,b,b,b\}$

$L(R^L) = \{a^n b^n \mid n \geq 0\}$

Expressividades das Linguagens Geradas pelas RdP

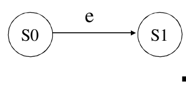
Redes de Petri × Automatos Finitos

- A expressividade das linguagens RdP é superior ao das geradas pelos automatos finitos.
- Prova
  - Mostrar que qualquer Automato Finito pode ser representado por uma RdP. Portanto, as linguagens geradas pelos Automatos Finitos podem ser geradas pelas RdP.
  - Mostrar que há RdP que não pode ser representada por um Automato Finito. Portanto, nem toda linguagem gerada por uma RdP pode ser representada pela linguagem geradas pelos Automatos Finitos.

Expressividades das Linguagens Geradas pelas RdP

- Automato Finito

AF =(S,s_0,E,f)

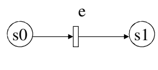


$S=\{s_0,s_1\}$
$E=\{e\}$
$f:s_0 \times e \rightarrow s_1$

→

- Rede de Petri

R=(P,T,I,O)

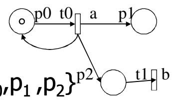


$P=\{s_0,s_1\}$
$T=\{e\}$
$I(e)=\{s_0\}$
$O(e)=\{s_1\}$

Expressividades das Linguagens Geradas pelas RdP

- Rede de Petri

R=(P,T,I,O) - é finita



P={p_0,p_1 ,p_2}

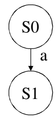
T={t_0,t_1}; Σ={a,b}

I(t_0)={p_0}, I(t_1)={p_2}

O(t_0)={p_0, p_1 ,p_2}, O(t_1)=$\varnothing$

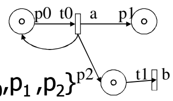
L(R)={$a^n b^n$ | n$\geq$0}

Expressividades das Linguagens Geradas pelas RdP



- Rede de Petri

R=(P,T,I,O) - é finita



P={p_0,p_1 ,p_2}

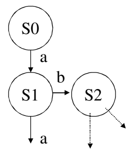
T={t_0,t_1}; Σ={a,b}

I(t_0)={p_0}, I(t_1)={p_2}

O(t_0)={p_0, p_1 ,p_2}, O(t_1)=$\varnothing$

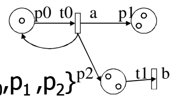
L(R)={$a^n b^n$ | n$\geq$0}

Expressividades das Linguagens Geradas pelas RdP



- Rede de Petri

R=(P,T,I,O) - é finita



P={p_0,p_1 ,p_2}

T={t_0,t_1}; Σ={a,b}

I(t_0)={p_0}, I(t_1)={p_2}

O(t_0)={p_0, p_1 ,p_2}, O(t_1)=$\varnothing$

L(R)={$a^n b^n$ | n$\geq$0}

•Número infinito de estados,portanto impossível de ser representada por um Automato Finito.Desta forma, a linguagem gerada pela rede não pode ser gerada por um Automato Finito

Propriedades

- Verificação
- Prova
- Análise -
- Validação

Propriedades

- **Verificação** - é baseada em algorítmos determinísticos que objetivam responder se um modelo possui ou não uma determiminada propriedade.
  - *one sided* - respostas: sim ou não sei
  - mais genérico - respostas: sim ou não (maior complexidade)
- **Prova** - é baseada em algorítmos não-determinísticos que procuram argumentos formais para a afirmação ou não da existência de uma propriedade em um modelo

Propriedades

- **Análise** - é baseada em algorítmos que não objetivam a verificação de uma determinada propriedade. Ao contrário, são algorítmos mais genéricos que fornecem informações sobre diversas propriedades e seus resultados podem ser utlizados como base para algorítmos de verificação.
  - Mesmo que a resposta seja sim ou não, não há prioridade de um sim sobre o não
  - Também são usadas para responder questões do tipo: Quais são os conjuntos de lugares cujo somatório de marcas permanece constante?

Propriedades

- **Validação** - é o processo de obtenção da confiabilidade de que sistema funciona como desejado.
  - O comportamento do sistema é comparado com a expectativa da pessoa a qual está validando o sistema.
  - É um processo inerentemente informal.
  - A princípio é um processo realizado sobre o sistema já implementado. No entano, pode também ser realizado sobre um modelo (simulação).
  - Verificação ou análise não garantem que o sistema funciona como desejado. De fato, obtêm as propriedades dos modelos.

Propriedades Comportamentais

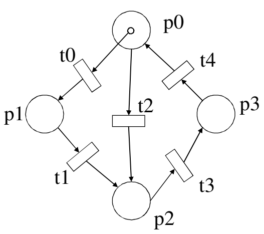
- Alcançabilidade (*Reachability*)

Indica a possibilidade de atingirmos uma determinada marcação pelo disparo de um número finito de transições, a partir de uma marcação inicial.

- Marcação Alcançável: seja $M_i[t_j>M_k$ e $M_k[t_h>M_l$ então $M_i[t_j t_h>M_l$. Por recorrência designamos o disparo de uma seqüência $s \in T^*$ por $M[s>M'$. Dizemos que M' é alcançável de M. O conjunto de todas as possíveis marcações alcançáveis de a partir de M_0 na rede N=(R, M_0) é denotado por $A(R; M_0)=\{M' \in \aleph^m | M_0[s>M'\}$ (RS). $m = |P|$

Propriedades Comportamentais

- Alcançabilidade (*Reachability*)

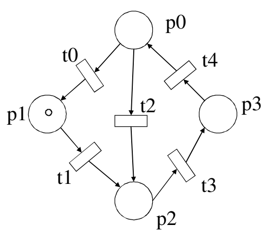


•M'=(0,0,0,1) é acessível a partir de M_0?

•É. Pelo disparo de $s'=t_0t_1t_3$ e de $s''=t_2t_3$

Propriedades Comportamentais

- Alcançabilidade (*Reachability*)

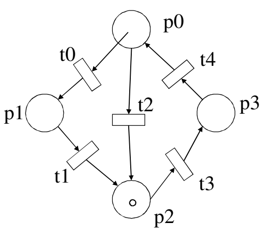


•M'=(0,0,0,1) é acessível a partir de M_0?

•É. Pelo disparo de $s'=t_0t_1t_3$ e de $s''=t_2t_3$

Propriedades Comportamentais

- Alcançabilidade (*Reachability*)

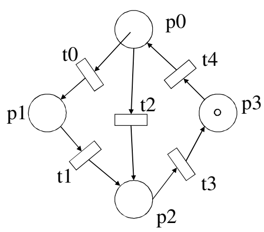


•M'=(0,0,0,1) é acessível a partir de M_0?

•É. Pelo disparo de $s'=t_0t_1t_3$ e de $s''=t_2t_3$

Propriedades Comportamentais

- Alcançabilidade (*Reachability*)

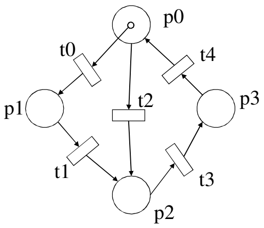


•M'=(0,0,0,1) é acessível a partir de M_0?

•É. Pelo disparo de $s'=t_0t_1t_3$ e de $s''=t_2t_3$

Propriedades Comportamentais

- Alcançabilidade (*Reachability*)

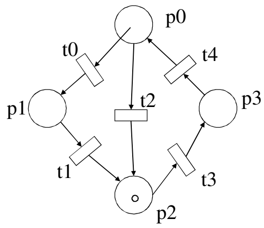


•M'=(0,0,0,1) é acessível a partir de M_0?

•É. Pelo disparo de s'=$t_0t_1t_3$ e de s''=t_2t_3

Propriedades Comportamentais

- Alcançabilidade (*Reachability*)

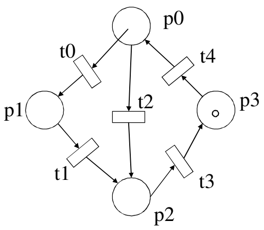


•M'=(0,0,0,1) é acessível a partir de M_0?

•É. Pelo disparo de s'=$t_0t_1t_3$ e de s''=t_2t_3

Propriedades Comportamentais

- Alcançabilidade (*Reachability*)



•M'=(0,0,0,1) é acessível a partir de M_0?

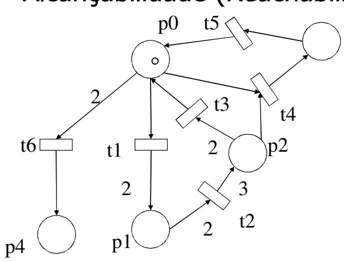
•É. Pelo disparo de s'=$t_0t_1t_3$ e de s''=t_2t_3

Propriedades Comportamentais

- Alcançabilidade (*Reachability*)
  - Problemas associados: alcançbilidade de sub-marcação e verificação de *deadlock*

Propriedades Comportamentais

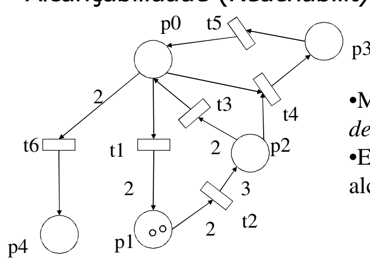
- Alcançabilidade (*Reachability*)



•M'=(0,0,0,0,1) é um *deadlock*.
•Esta marcação é alcançável?

Propriedades Comportamentais

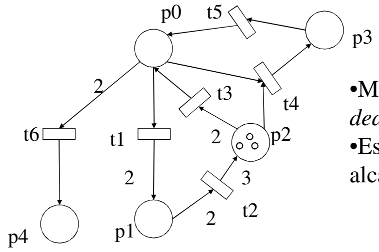
- Alcançabilidade (*Reachability*)



•M'=(0,0,0,0,1) é um *deadlock*.
•Esta marcação é alcançável?

Propriedades Comportamentais

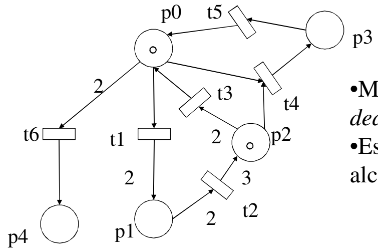
- Alcançabilidade (*Reachability*)



•M'=(0,0,0,0,1) é um *deadlock*.
•Esta marcação é alcançável?

Propriedades Comportamentais

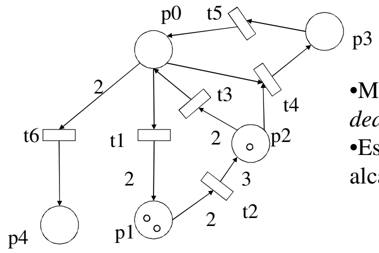
- Alcançabilidade (*Reachability*)



•M'=(0,0,0,0,1) é um *deadlock*.
•Esta marcação é alcançável?

Propriedades Comportamentais

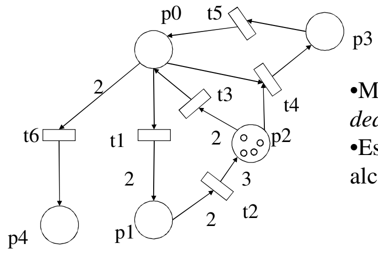
- Alcançabilidade (*Reachability*)



•M'=(0,0,0,0,1) é um *deadlock*.
•Esta marcação é alcançável?

Propriedades Comportamentais

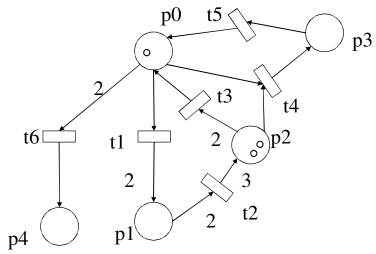
- Alcançabilidade (*Reachability*)



•M'=(0,0,0,0,1) é um *deadlock*.
•Esta marcação é alcançável?

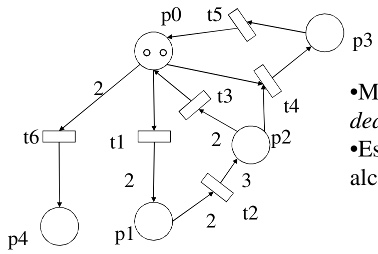
Propriedades Comportamentais

- Alcançabilidade (*Reachability*)



Propriedades Comportamentais

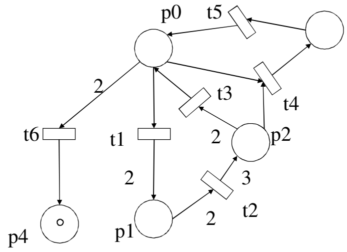
- Alcançabilidade (*Reachability*)



•M'=(0,0,0,0,1) é um *deadlock*.
•Esta marcação é alcançável?

Propriedades Comportamentais

- Alcançabilidade (*Reachability*)



•M'=(0,0,0,0,1) é um *deadlock*.
•Esta marcação é alcançável?

Resposta: É

Propriedades Comportamentais

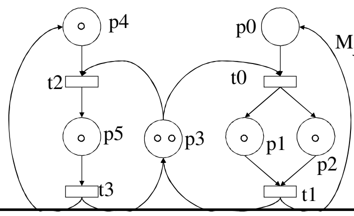
- Limitação (*Boundedness*)

Um lugar é dito limitado se o número de marcas que este lugar pode armazenar é finito.

- Limitação : seja um lugar $p_i \in P$ de uma rede marcada N1=(R, M_0). Este lugar é dito k-limitado (*k-bounded*) ou simplesmente limitado se para toda marcação acessível $M \in A(R; M_0)$, $M(p_i) \leq k$, onde $k \in \aleph$.

Propriedades Comportamentais

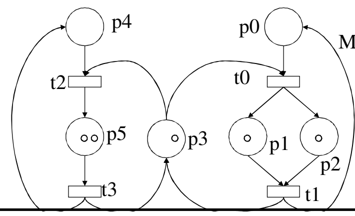
- Rede Limitada : Dizemos que uma rede N1=(R, M_0) é limitada (*bounded*) se $k(p_i) \leq \infty$, $\forall\, p_i \in P$.



$$\begin{array}{cc} & p_0\; p_1\; p_2\; p_3\; p_4\; p_5 \\ M_j = & (\,0,\; 1,\; 1,\; 2,\; 1,\; 1) \end{array}$$

Propriedades Comportamentais

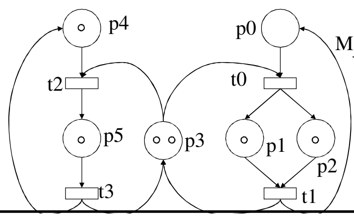
- Rede Limitada : Dizemos que uma rede N1=(R, M_0) é limitada (*bounded*) se $k(p_i) \leq \infty$, $\forall\, p_i \in P$.



$$\begin{array}{cc} & p_0\; p_1\; p_2\; p_3\; p_4\; p_5 \\ M_j = & (\,0,\; 1,\; 1,\; 1,\; 0,\; 2) \end{array}$$

Propriedades Comportamentais

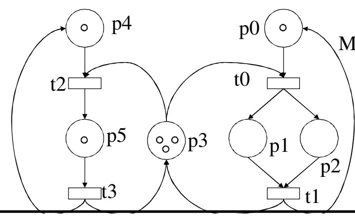
- Rede Limitada : Dizemos que uma rede N1=(R, M_0) é limitada (*bounded*) se $k(p_i) \leq \infty$, $\forall\, p_i \in P$.



$$\begin{array}{cc} & p_0\; p_1\; p_2\; p_3\; p_4\; p_5 \\ M_j = & (\,0,\; 1,\; 1,\; 2,\; 1,\; 1) \end{array}$$

Propriedades Comportamentais

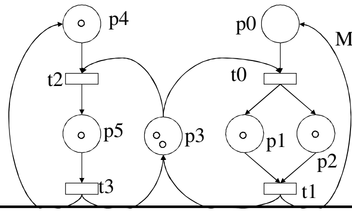
- Rede Limitada : Dizemos que uma rede N1=(R, M_0) é limitada (*bounded*) se $k(p_i) \leq \infty$, $\forall\, p_i \in P$.



$$\begin{array}{cc} & p_0\; p_1\; p_2\; p_3\; p_4\; p_5 \\ M_j = & (\,1,\; 0,\; 0,\; 3,\; 1,\; 1) \end{array}$$

Propriedades Comportamentais

- **Rede Limitada** : Dizemos que uma rede N1=(R, M_0) é limitada (*bounded*) se $k(p_i) \leq \infty$, $\forall\ p_i \in P$.



$\quad\quad\ \ p_0\ p_1\ p_2\ p_3\ p_4\ p_5$
$M_j = (\ 1, 0, 0,\ 3, 1, 1)$

- p_4 é 2-limitado
- p_3 é 4-limitado
- Rede limitada

Propriedades Comportamentais

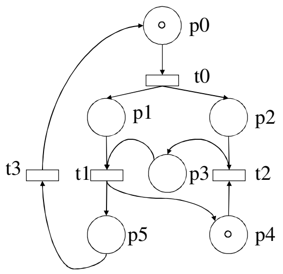
- Segurança (*Safeness*)

É uma particularização do conceito de *Boundedness*.

- Lugar Seguro: seja um lugar $p_i \in P$ de uma rede marcada N1=(R, M_0). p_i é seguro (*safe*) se $M(p_i) \leq 1$, $\forall\ M \in A(R; M_0)$.
- Rede Segura: uma rede marcada N1=(R, M_0) é definida como segura se $M(p_i) \leq 1$, $\forall\ M \in A(R; M_0)$, $\forall\ p_i \in P$.

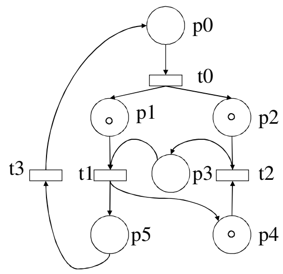
Propriedades Comportamentais

- Segurança (*Safeness*)



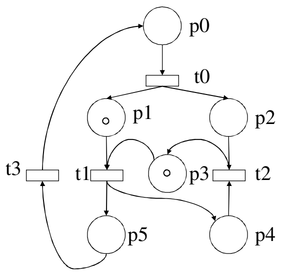
Propriedades Comportamentais

- Segurança (*Safeness*)



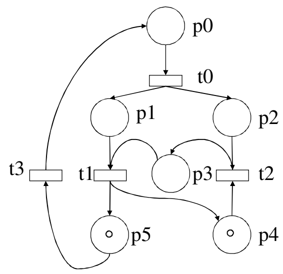
Propriedades Comportamentais

- Segurança (*Safeness*)



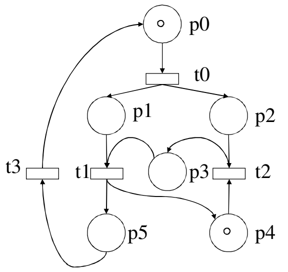
Propriedades Comportamentais

- Segurança (*Safeness*)



Propriedades Comportamentais

- Segurança (*Safeness*)



•Rede safe

Propriedades Comportamentais

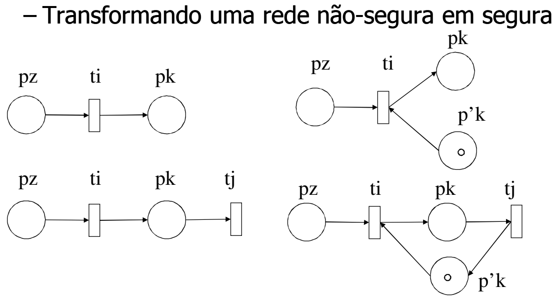
- Segurança (*Safeness*)

  Transformando uma rede não-segura em segura

  – Se $p_k \in O(t_i)$ e $p_k \notin I(t_i)$, então cria-se o lugar $p'_k \in I(t_i)$ e $M(p'_k)=1$
  – Se $p_k \in I(t_j)$, então $p'_k \in O(t_j)$

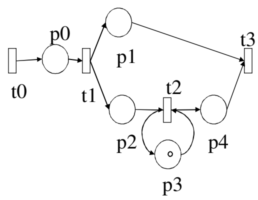
Propriedades Comportamentais

- Segurança (*Safeness*)
  – Transformando uma rede não-segura em segura

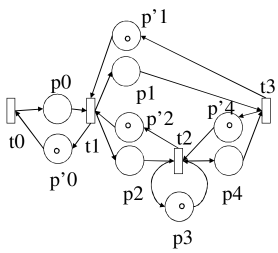


Propriedades Comportamentais

- Segurança (*Safeness*)
  – Transformando uma rede não-segura em uma segura



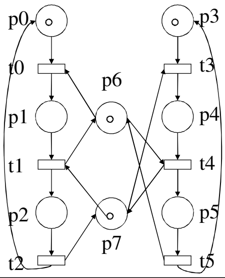
Rede não-segura



Rede segura

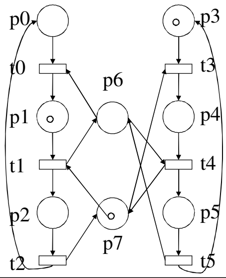
Propriedades Comportamentais

- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



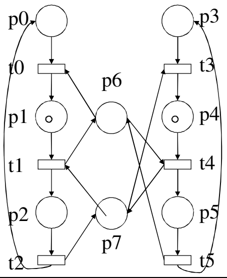
Propriedades Comportamentais

- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



Propriedades Comportamentais

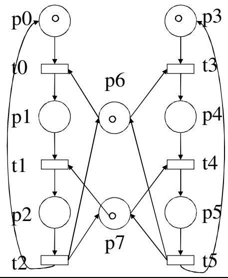
- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



Deadlock

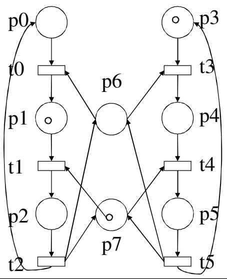
Propriedades Comportamentais

- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



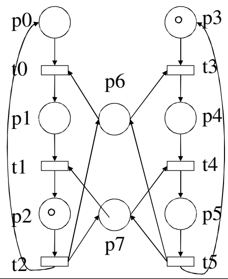
Propriedades Comportamentais

- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



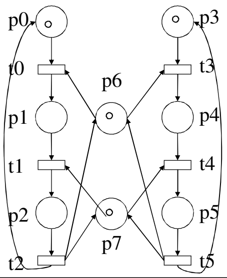
Propriedades Comportamentais

- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



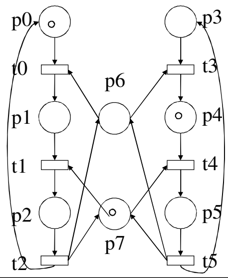
Propriedades Comportamentais

- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



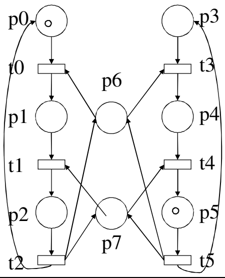
Propriedades Comportamentais

- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



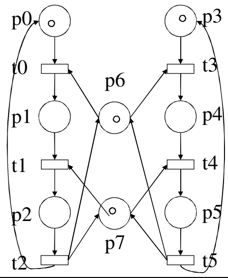
Propriedades Comportamentais

- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



Propriedades Comportamentais

- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



Sem *deadlock*

Propriedades Comportamentais

Para que um sistema esteja em *deadlock* é necessário que estas três (3) condições sejam satisfeitas:

- **Exclusão Mútua** - um processo alocou um determinado recurso e ele tem acesso exclusivo aquele recurso.
- **Espera Circular** - esta situação ocorre quando um processo **P1** possui um recurso **R1** e solicita um recurso **R2** e um processo **P2** possui um recurso **R2** e solicita o recurso **R1.**
- **Ausência de Preempção** - os recurso só podem ser liberados pela ação explicita do processo que tem o recursos alocado. Não há outra entidade com autoridade para liberação do recurso.

Propriedades Comportamentais

- Transição Potencialmente Disparável:

chamamos uma trnasição t_j potencialmente disparável em uma marcação M_0 se $\exists$ M' $\in$ A(R; M_0) tal que M'[t_j>.

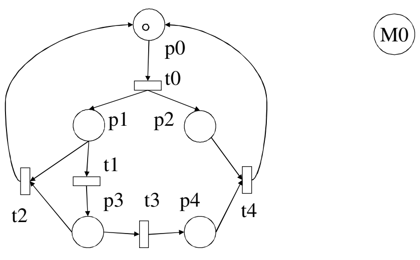
- Rede *Live*: uma rede N=(R,M_0) é dita *live* se $\exists$ s $\in$ L(N) tal que $t_i \in$ s, $\forall t_i \in$ T.

Outra definição

- Rede *Live*: uma rede N=(R,M_0) é dita *live* se $\exists$ s $\in$ T* tal que $t_i \in$ s, M'[s>, $\forall$M' $\in$ A(R, M_0), $\forall t_i \in$ T.

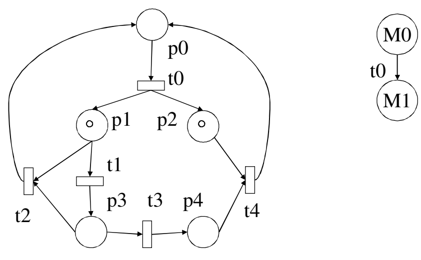
Propriedades Comportamentais

- *Liveness* é mais forte do que *deadlock freedom*



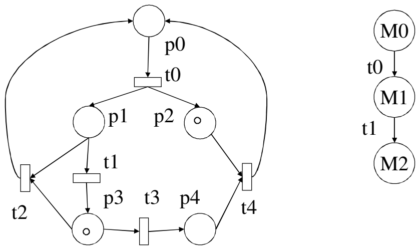
Propriedades Comportamentais

- *Liveness* é mais forte do que *deadlock freedom*



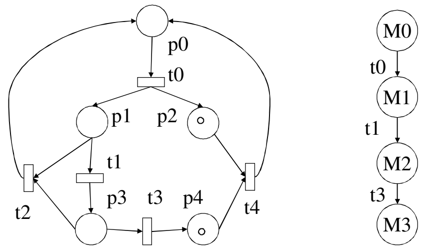
Propriedades Comportamentais

- *Liveness* é mais forte do que *deadlock freedom*



Propriedades Comportamentais

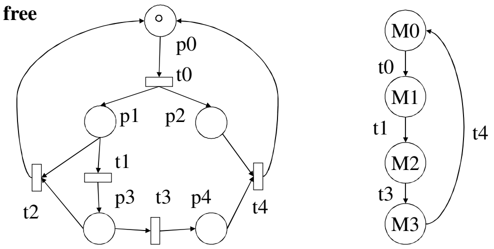
- *Liveness* é mais forte do que *deadlock freedom*



Propriedades Comportamentais

- *Liveness* é mais forte do que *deadlock freedom*

•Deadlock free
•Non-live



Propriedades Comportamentais

- *Liveness* é uma propriedade "cara" de se analisar.
- Uma transição t_i tem sido classificada em níveis.
- Níveis de *liveness*:
  0 . Morta (*dead*) ou nível N0, se $\nexists$ s $\in$ L(R,M_0), tal que $t_i \in$ s, ou seja, $\nexists$ M' $\in$ A(R,M_0), tal que M'[t_i> .
  1 . N1-*live*, se t_i pode ser disparada pelo menos uma vez em alguma seqüência s $\in$ L(R,M_0).
  2 . N2-*live*, se t_i pode ser disparada pelo menos *k* vez em alguma seqüência s $\in$ L(R,M_0).
  3 . N3-*live*, se t_i aparece um número infinito de vezes em alguma seqüência s $\in$ L(R,M_0).
  4 . N4-*live*, ou simplesmente *live*, se t_i é N1-*live* $\forall$ M' $\in$ A(R,M_0),

Propriedades Comportamentais

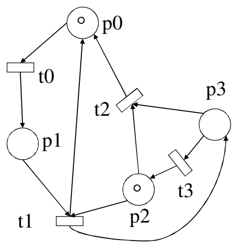
- Cobertura (*coverability*) de uma marcação: seja a marcação M' em uma rede N=(R,M_0). M' é dita coberta se existe M'' $\in$ A(R, M_0) tal que M''(p_i)≥M'(p_i) $\forall p_i \in$ P.

Propriedades Comportamentais

- Reversibilidade: inicialmente uma rede é dita reversível se para cada marcação M_i no conjunto das marcações acessíveis a marcação inicial pode ser novamente alcançada.
- *Home-State*: seja uma marcação $M_k \in$ A(R,M_0). M_k é denominada *home-state* se M_i [>M_k, $\forall M_i \in$ A(R,M_0).
- Reversibilidade: uma rede N=(R,M_0) é reversível se $\exists$ M_k , tal que M_i[>M_k , $\forall M_i \in$ A(R,M_0). (relaxamento do que foi descrito acima)

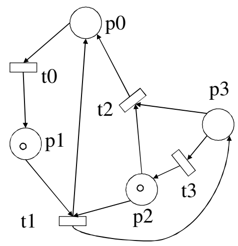
Propriedades Comportamentais

- Reversibilidade



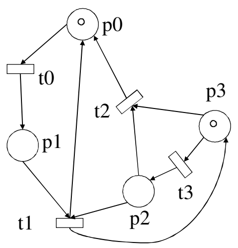
Propriedades Comportamentais

- Reversibilidade



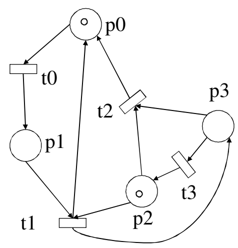
Propriedades Comportamentais

- Reversibilidade



Propriedades Comportamentais

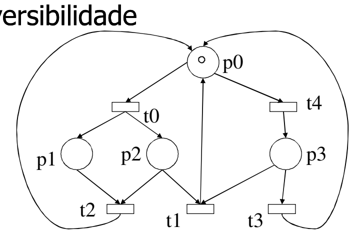
- Reversibilidade



Reversível

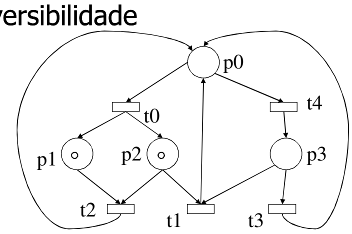
Propriedades Comportamentais

- Reversibilidade



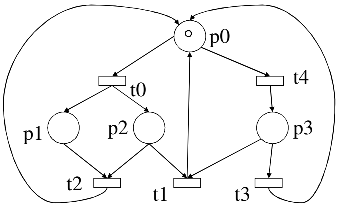
Propriedades Comportamentais

- Reversibilidade



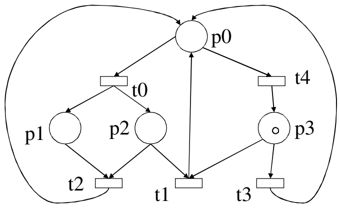
Propriedades Comportamentais

- Reversibilidade



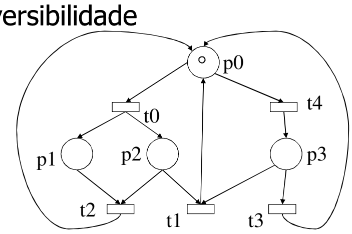
Propriedades Comportamentais

- Reversibilidade



Propriedades Comportamentais

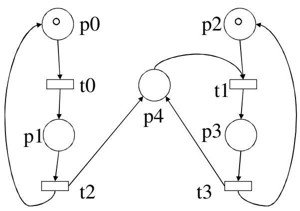
- Reversibilidade



Reversível

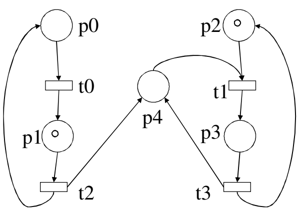
Propriedades Comportamentais

- Reversibilidade



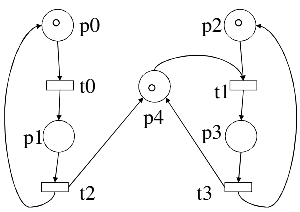
Propriedades Comportamentais

- Reversibilidade



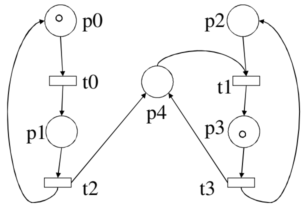
Propriedades Comportamentais

- Reversibilidade



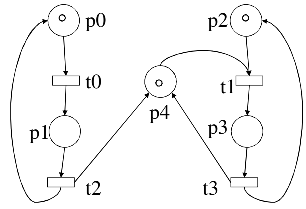
Propriedades Comportamentais

- Reversibilidade



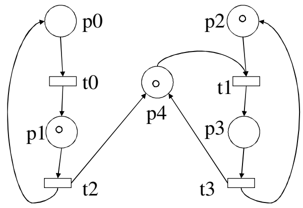
Propriedades Comportamentais

- Reversibilidade



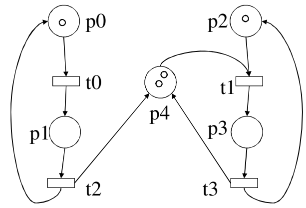
Propriedades Comportamentais

- Reversibilidade



Propriedades Comportamentais

- Reversibilidade



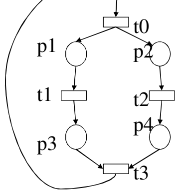
Irreversível

Propriedades Comportamentais

- Persistência: uma rede é dita persistente se para qualquer par de transições o disparo de uma não desabilita a outra.
- Persistência: seja uma rede $N=(R,M_0)$. N é dita persistente se para todo par $(t_i, t_j) \in T^2$ tal que $\exists$ $M_k[t_i>$ e $M_k[t_j> \Rightarrow M_k[t_i>M''$, $M''[t_j>$ e vice-versa. Onde $M_k, M'' \in A(R,M_0)$.

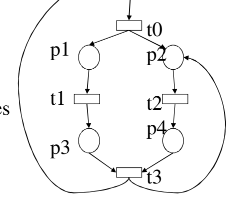
Propriedades Comportamentais

- Persistência: todo grafo-marcado é persistente. Nem toda rede presistente é uma grafo-marcado.



Grafo-marcado

Persistentes



Não é grafo-marcado

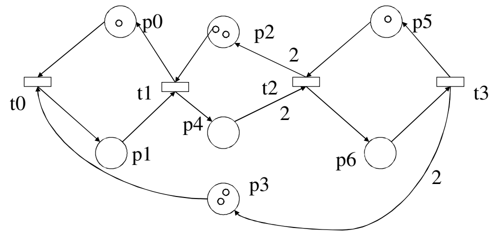
Propriedades Comportamentais

- Justiça Limitada (*Bounded Fairness*): uma rede é dita *bounded fair* se para qualquer par de transições o número de disparo de uma enquanto a outra não dispara é limitado.

- Justiça Limitada: seja uma rede $N=(R,M_0)$, $s \in L(R,M_0)$ uma seqüência e um par $(t_i, t_j) \in T^2$ e $\overline{s(t_i)}$ o número de disparos de t_i em **s**. Se $|\overline{s(t_i)} - \overline{s(t_j)}| \neq \infty$, $\forall (t_i, t_j) \in T^2$ N é dita *bounded fair* se todo par $(t_i, t_j) \in T^2$ é *B-fair*

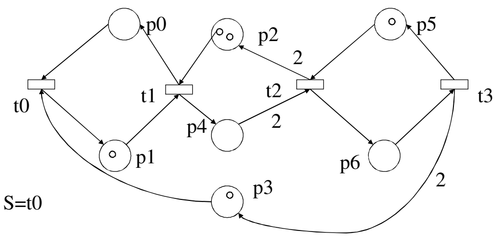
Propriedades Comportamentais

- Justiça Limitada



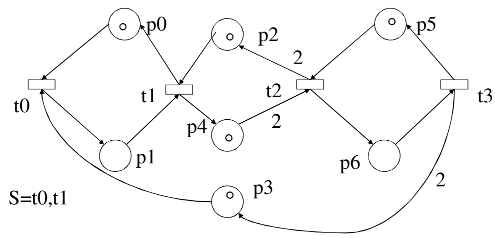
Propriedades Comportamentais

- Justiça Limitada



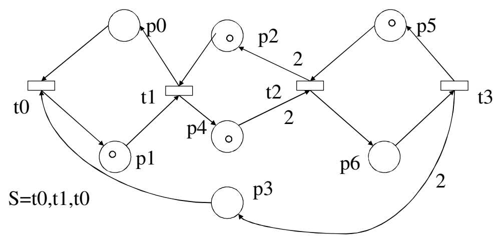
Propriedades Comportamentais

- Justiça Limitada



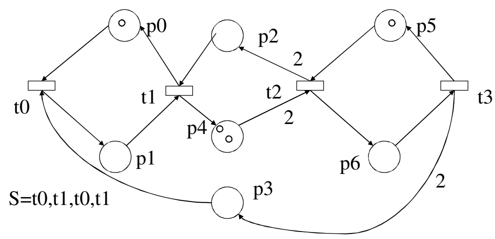
Propriedades Comportamentais

- Justiça Limitada



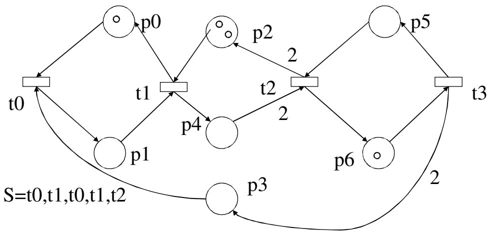
Propriedades Comportamentais

- Justiça Limitada



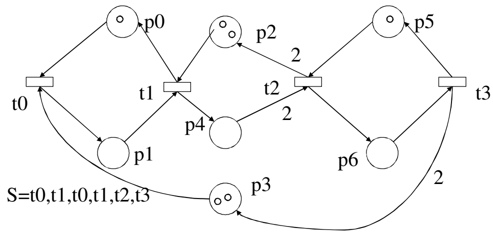
Propriedades Comportamentais

- Justiça Limitada



Propriedades Comportamentais

- Justiça Limitada



Propriedades Comportamentais

- Distância Sincrônica: relaciona o grau de dependência mútua entre eventos.
- Distância Sincrônica: seja uma rede $N=(R,M_0)$, t_i, $t_j \in T$ é, $L(N)$ a linguagem de N. $d(t_i, t_j)=\max_{s\in L(N)} \{|\overline{s(t_i)} - \overline{s(t_j)}|\}$ é definida como a distância sincrônica entre as transições.
  - Se $d(t_i, t_j) \to \infty \Rightarrow$ que o modelo é *unfair.*
  - Se $d(t_i, t_j) = 1 \Rightarrow$ que a cada disparo de t_i, t_j dispara.

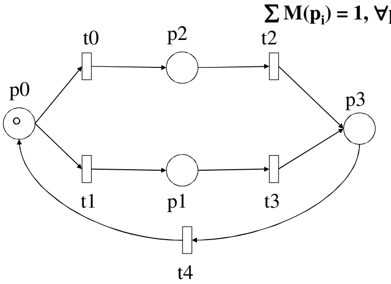
Propriedades Comportamentais

- Conservação: está relacionada ao somatório de marcas a medida que as transições são disparadas.
- Rede Estritamente Conservativa: seja uma rede $N=(R,M_0)$ e $M\in A(R,M_0)$ uma marcação alcançável. N é dita estritamente conservativa $\sum M(p_i) = \sum M_0(p_i)$, $\forall p_i \in P$ e $\forall M\in A(R,M_0)$.

Propriedades Comportamentais

- Rede Estritamente Conservativa

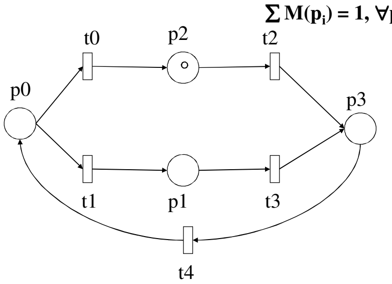
$$\sum M(p_i) = 1, \forall p_i \in P$$



Propriedades Comportamentais

- Rede Estritamente Conservativa

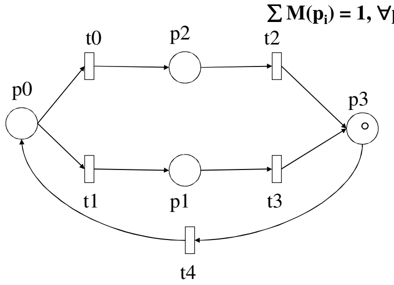
$$\sum M(p_i) = 1, \forall p_i \in P$$



Propriedades Comportamentais

- Rede Estritamente Conservativa

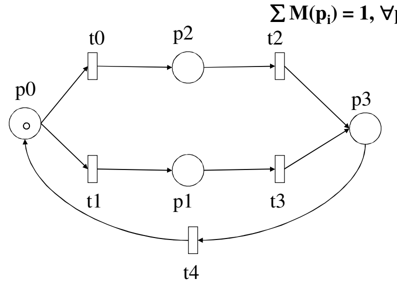
$\sum M(p_i) = 1, \forall p_i \in P$



Propriedades Comportamentais

- Rede Estritamente Conservativa

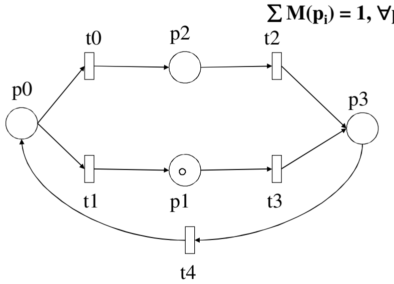
$\sum M(p_i) = 1, \forall p_i \in P$



Propriedades Comportamentais

- Rede Estritamente Conservativa

$\sum M(p_i) = 1, \forall p_i \in P$



Propriedades Comportamentais

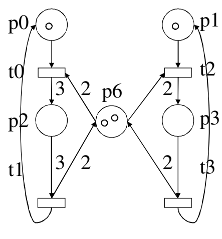
- Conservação: está relacionada ao somatório de marcas a medida que as transições são disparadas.

- Rede Conservativa: seja uma rede $N=(R,M_0)$, $M \in A(R,M_0)$ uma marcação alcançável e $W=(w_1,\ldots,w_n)$, onde n=#P. N é dita conservativa $\sum_{pi \in P} w_i \times M(p_i) = \sum_{pi \in P} w_i \times M_0(p_i$, $\forall M \in A(R,M_0)$.

Propriedades Comportamentais

- Conservação

$\sum_{pi \in P} w_4^i \times \mathbf{m_0(p_i) = 17, \forall p_i \in P}$

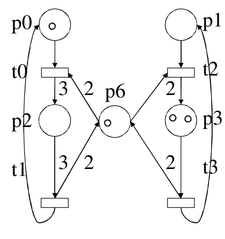


$W_1 = (3,0,1,0,0)$
$W_2 = (0,2,0,1,0)$
$W_3 = (0,0,4,3,6)$
$W_4 = (3,2,5,4,6)$

Propriedades Comportamentais

- Conservação

$\sum_{pi \in P} w_4^i \times \mathbf{m_0(p_i) = 17, \forall p_i \in P}$

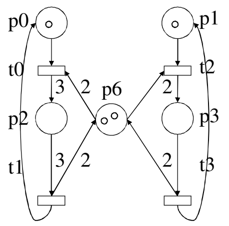


$W_1 = (3,0,1,0,0)$
$W_2 = (0,2,0,1,0)$
$W_3 = (0,0,4,3,6)$
$W_4 = (3,2,5,4,6)$

Propriedades Comportamentais

- Conservação

$\sum_{p_i \in P} w_4^i \times \mathbf{m_0(p_i) = 17}, \forall \mathbf{p_i \in P}$

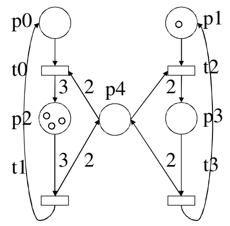


$W_1 = (3,0,1,0,0)$
$W_2 = (0,2,0,1,0)$
$W_3 = (0,0,4,3,6)$
$W_4 = (3,2,5,4,6)$

Propriedades Comportamentais

- Conservação

$\sum_{p_i \in P} w_4^i \times \mathbf{m_0(p_i) = 17}, \forall \mathbf{p_i \in P}$

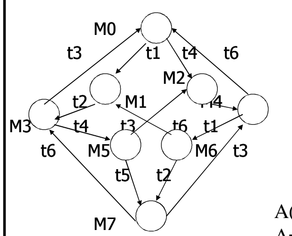


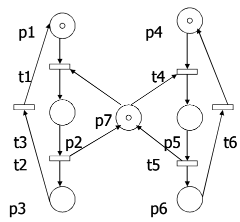
$W_1 = (3,0,1,0,0)$
$W_2 = (0,2,0,1,0)$
$W_3 = (0,0,4,3,6)$
$W_4 = (3,2,5,4,6)$

Análise de Proprieadades Comportamentais

- Grafo de Alcançabilidade: seja uma rede marcada $N=(R,M_0)$. $RG(R, M_0)=(RS,ARCS)$ define o grafo de alcançabilidade (*Reachability Graph*), onde $RS:P\rightarrow\aleph$ é o conjunto de vértice e representa o conjunto de alcançabilidade. $ARCS \subseteq RS \times T \times RS$ é uma relação representando os arcos.

Grafo de Marcações Acessíveis (Alcançáveis)

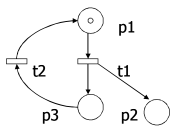




$A(N,M_0)=\{M_0,M_1,M_2,M_3,M_4,M_5,M_6,M_7\}$
A=RS - *reachability set*

Grafo de Marcações Acessíveis (Alcançáveis)

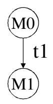
- RG Infinito

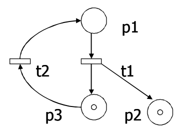


$A(N,M_0)=\{M_0,M_1,M_2,...\}$
A=RS - *reachability set*

Grafo de Marcações Acessíveis (Alcançáveis)

- RG Infinito

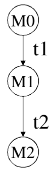


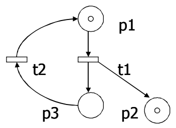


$A(N,M_0)=\{M_0,M_1,M_2,...\}$
A=RS - *reachability set*

Grafo de Marcações Acessíveis (Alcançáveis)

- RG Infinito

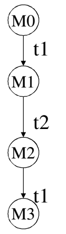


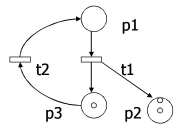


$A(N,M_0)=\{M_0,M_1,M_2,...\}$
A=RS - *reachability set*

Grafo de Marcações Acessíveis (Alcançáveis)

- RG Infinito





$A(N,M_0)=\{M_0,M_1,M_2,...\}$
A=RS - *reachability set*

Árvore de Cobertura

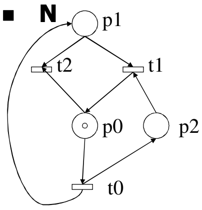
- Grafo de Cobertura: seja uma rede marcada $N=(R,M_0)$. $CG(R, M_0)=(CS,ARCS)$ define o grafo de cobertura (*Coverability Graph*), onde $CS:P\rightarrow \aleph \cup \{\omega\}$ é o conjunto de vértice e representa o conjunto de cobertura. $ARCS \subseteq RS \times T \times RS$ é uma relação representando os arcos.

Árvore de Cobertura

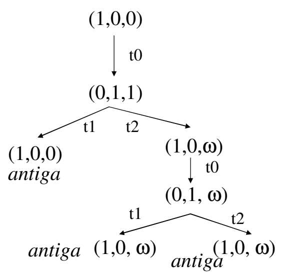
1. Rotule a marcação inicial M_0 como a raiz e discrimene-a como *nova*
2. Enquanto houver marcações *novas*, faça:
   1. Selecione uma marcação M.
      1. Se M é identica a uma marcação no caminho desde a raiz até M, então rotule-a como *antiga* e selecione uma *nova* marcação.
   2. Se nenhuma transição está habilitada para M, rotule esta marcação como *final*
   3. Enquanto houver transição habilitada para M, faça o seguinte para cada transição habilitada:
      1. Obtenha a nova marcação M'que resulta do disparo de uma transição **t** habilitada para M.
      2. Se no caminho da raiz para a marcação M existe uma marcação M'' tal que para todo lugar **p** $M'(p)\geq M''(p)$ e $M'\neq M''$ então substitui-se M'(p) por ω para cada **p** onde $M'(p)>M''(p)$.
      3. Introduza **M'** como nó e crie um arco rotulado com **t** de **M** para **M'**e rotule M'como *nova.*

Árvore de Cobertura

- **N**

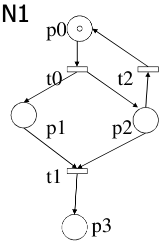


- CG



Árvore de Cobertura

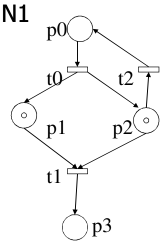
- N1



(p_0, p_1, p_2, p_3)
M_0 = (0, 0, 1, 0)
t2
M_1 =(1, 0, 0, 0)

Árvore de Cobertura

- N1

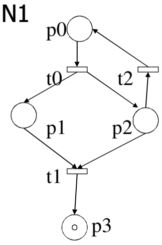


(p_0, p_1, p_2, p_3)

$M_0 = (0,\ 0,\ 1,\ 0)$ $\xrightarrow{t2}$ $M_1 = (1,\ 0,\ 0,\ 0)$ $\xrightarrow{t0}$ $M_2 = (0,\ \omega,\ 1,\ 0)$

Árvore de Cobertura

- N1



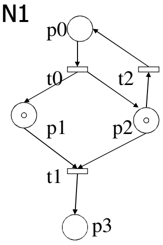
(p_0, p_1, p_2, p_3)

$M_0 = (0,\ 0,\ 1,\ 0)$ $\xrightarrow{t2}$ $M_1 = (1,\ 0,\ 0,\ 0)$ $\xrightarrow{t0}$ $M_2 = (0,\ \omega,\ 1,\ 0)$ $\xrightarrow{t1}$ $M_3 = (0,\ \omega,\ 0,\ 1)$

morta

Árvore de Cobertura

- N1



(p_0, p_1, p_2, p_3)

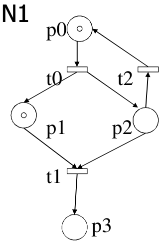
$M_0 = (0,\ 0,\ 1,\ 0)$ $\xrightarrow{t2}$ $M_1 = (1,\ 0,\ 0,\ 0)$ $\xrightarrow{t0}$ $M_2 = (0,\ \omega,\ 1,\ 0)$

$M_2 \xrightarrow{t1} M_3 = (0,\ \omega,\ 0,\ 1)$ *morta*

$M_2 \xrightarrow{t2}$

Árvore de Cobertura

- N1



(p_0, p_1, p_2, p_3)

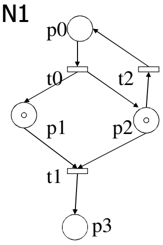
$M_0 = (0,\ 0,\ 1,\ 0)$ $\xrightarrow{t2}$ $M_1 = (1,\ 0,\ 0,\ 0)$ $\xrightarrow{t0}$ $M_2 = (0,\ \omega,\ 1,\ 0)$

$M_2 \xrightarrow{t1} M_3 = (0,\ \omega,\ 0,\ 1)$ *morta*

$M_2 \xrightarrow{t2} M_4 = (1,\ \omega,\ 0,\ 0)$

Árvore de Cobertura

- N1



(p_0, p_1, p_2, p_3)

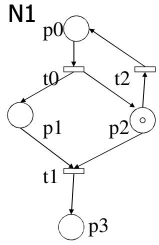
$M_0 = (0,\ 0,\ 1,\ 0)$ $\xrightarrow{t2}$ $M_1 = (1,\ 0,\ 0,\ 0)$ $\xrightarrow{t0}$ $M_2 = (0,\ \omega,\ 1,\ 0)$

$M_2 \xrightarrow{t1} M_3 = (0,\ \omega,\ 0,\ 1)$ *morta*

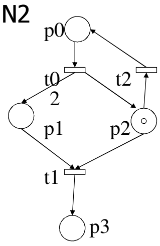
$M_2 \xrightarrow{t2} M_4 = (1,\ \omega,\ 0,\ 0)$ $\xrightarrow{t0}$ $M_2 = (0,\ \omega,\ 1,\ 0)$ *final*

Árvore de Cobertura

- N1



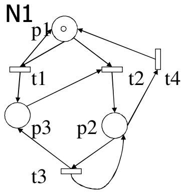
- N2



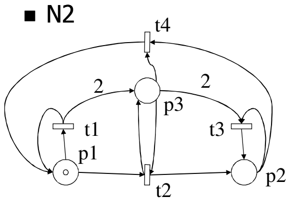
* ambas as redes têem a mesma árvore de cobertura

Árvore de Cobertura

- N1



- N2



S= $t_1t_2t_3$ leva a deadlock

* ambas as redes têm a mesma árvore de cobertura, mas N1 é *live* e N2 não

Verificação (*Model Checking*)

- Consiste em um processo de verificação de predicados (condições).
- Comumente se utilizam dois formalismos:
  - Um para representar o espaço de estados (total ou parcial)
    - *reachability graph, State graph, LTS*; e
  - Um segundo para se expressar o predicado
    - LTL (*Linear temporal Logic*) , CTL (*Computational Tree Logic*).

Verificação (*Model Checking*)

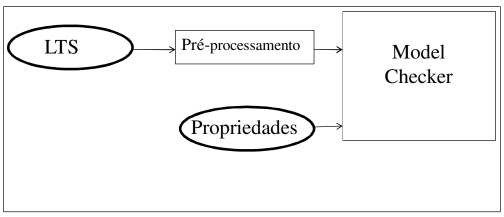
– Verificação Formal = verificar formalmente propriedades.

– Necessidade de modelo formal.

Model Checking

- Tem por objetivo verificar propriedades em um *LTS*.
- Analisa o espaço de estados de um modelo.
- A verificação é realizada em três etapas:
  - Definição das propriedades a serem verificadas,
  - Geração do espaço de estados,
  - *Model checking* (a verificação propriamente).

Verificação (*Model Checking*)



Ferramentas de *model checking* verificam propriedades.

Uma classificação das lógicas

–Lógica proposicional
- Consiste de fórmulas que consistem de variáveis booleanas e operadores lógicos:
  $\wedge$,$\vee$ e $\neg$.

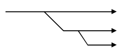
–Lógica de primeira ordem
- Inclui os quantificadores: $\exists$ e $\forall$.

–Lógica de alta ordem
- Permite o estudo de funções de outros "objetos".

Lógica temporal

- Lógica estendida com a noção de tempo.
- *Branching* × tempo linear:
  - Tempo linear:
    modelo temporal em que, a cada instante, apenas uma atividade é possível.

  - *Branching time:*
    modelo temporal em que, a cada instante, todas as possíveis atividades são representadas.



Espaço de estados

- Estrutura temporal $M := (S,R,L)$ - *Kripke structure*
  1. Conjunto finito de estados S
  2. Relação de transição $R \subseteq S \times S$ com $\forall s \in S, \exists s' \in S: (s,s') \in R$
  3. Função de nomeação $L: S \rightarrow \wp(V)$, em que V é o conjunto de variáveis proposicionais.

Um predicado é uma disjunção de conjunções de expressões atômica (que podem ser negadas).

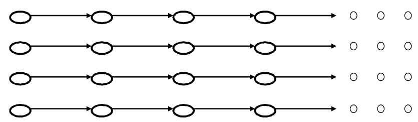
Operadores CTL

- Operadores temporais
  - Xp - p é satisfeito no próximo instante
  - Fp - p é satisfeito em algum momento no futuro
  - Gp - p é satisfeito de forma global a partir deste momento
  - p U q - p é satisfeito até que q seja é satisfeito

Operadores CTL

Operadores temporais**:**

Gp
Fp
Xp
pUq



328

Operadores CTL

- Quantificador de caminho
  - A ($\forall$)– para todos caminhos
  - E ($\exists$)– existe um caminho
- Operadores:
  - AX, AF, AG, AU, EX, EF, EG, EU

Verificação
(*Model Checking*)

- Sintaxe INA

```
<formula> ::= 'T' | 'P'<number> | '-'<formula> |
              '('<formula>'&'<formula>')' | '('<formula>'V'<formula>')' |
              'E''X'<formula> | 'E''F'<formula> | 'E''G'<formula> |
              'A''X'<formula> | 'A''F'<formula> | 'A''G'<formula> |
              'E''['<formula>'U'<formula>']' | 'A''['<formula>'U'<formula>']'
              'E''['<formula>'B'<formula>']' | 'A''['<formula>'B'<formula>']'
 <number> ::= <nonempty sequence of digits followed by a blank>
```

  - Os predicados são especificados por 'P´<number>

Verificação
(*Model Checking*)

- Sintaxe INA

`T` stands for the truth value "true", predicates are referred to with `P<number>`

`-` stands for the logical "not", `&` and `V` stand for the logical connectors "and" and "or"

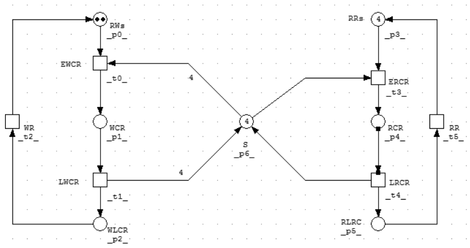
`EX`, `EF`, `EG`, `AX`, `AF`, `AG`, `E U`, `E B`, `A U`, and `A B` stand for the corresponding temporal-logical quantors

After entering `<Y>`, you can write the formula into a file with the extension `.ctl`. Thereafter, the predicates whose numbers were stated in the formula are requested. You can either read previously defined predicates from a file, or, by pressing `<esc>`, enter predicates directly.

You can have a proof constructed by INA which contains either only `witnesses and counter examples`, or yet additional information. You can write this proof into a file with the extension `.prf`.

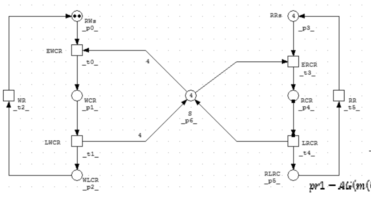
Verificação
(*Model Checking*)

- Exemplo (Leitores/Escritores)



Verificação
(*Model Checking*)

- Exemplo (Leitores/Escritores)



$pr1 - AG(m(WCR) > 1) \vee ((m(WCR) > 0) \wedge (m(RCR) > 0))$

$pr1 = AG\,(P1 \vee P2)$

$P1 = m(WCR) > 1$

$P2 = (m(WCR) > 0) \wedge (m(RCR) > 0)$

Verificação
(*Model Checking*)

```
Syntax:
<formula> ::= 'T' | 'P'<number> | '-'<formula> |
              '('<formula>'&'<formula>')' | '('<formula>'V'<formula>')' |
              'E''X'<formula> | 'E''F'<formula> | 'E''G'<formula> |
              'A''X'<formula> | 'A''F'<formula> | 'A''G'<formula> |
              'E''['<formula>'U'<formula>']' | 'A''['<formula>'U'<formula>']'
              'E''['<formula>'B'<formula>']' | 'A''['<formula>'B'<formula>']'
 <number> ::= <nonempty sequence of digits followed by a blank>
In the atomic mode, 'P'<number> holds iff the place <number> is marked.
Please, type the formula to check:
AG(P1 V P2 )
Formula to check:
AG(P1 VP2 )
.................................................ok? Y/N Y
.......................Write the formula to a file? Y/N Y
output file > RW.ctl
              overwrite? Y/N y
```

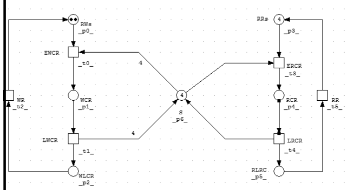
Comandos no INA

(Não observe os valores)

```
Predicate nr. 2:
Predicate files:
HTPN     mc4      n         q''      RW       rw1      UnliveTPN
Predicate input file > RW.pdc
Predicate nr. 1:
Predicate files:
HTPN     mc4      n         q''      RW       rw1      UnliveTPN
Predicate input file > terminal
Please, give the conjunction nr.  1
Token atoms:
place nr. 1.WCR: lower bound = 1    upper bound = 1
...................................Negate the atom? Y/N N
place nr. Q
..................................Next conjunction? Y/N Y
Please, give the conjunction nr.  2
Token atoms:
place nr. 1.WCR: lower bound = 1    upper bound = 2
...................................Negate the atom? Y/N N
place nr. 4.RCR: lower bound = 1    upper bound = 4
...................................Negate the atom? Y/N N
place nr. Q
..................................Next conjunction? Y/N N
......................................Write a file? Y/N N
.....................................Write a proof? Y/N Y
............but only witnesses and counterexamples? Y/N N
..........................Write to a separate file? Y/N Y
Proof output file > RW.prf
              overwrite? Y/N y
```

Verificação
(*Model Checking*)

- Exemplo (Leitores/Escritores)

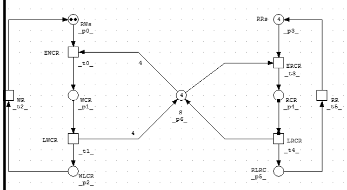


Arquivo .ctl (Fórmula a verificar)
AG(P1 VP2)*

Arquivo .pdc (Predicados)
predicate for net: 0 BNT
Conjunction nr: 2
m + p4.RCR   low = 1 upp = 4
m + p1.WCR   low = 1 upp = 2
Conjunction nr: 1
m + p1.WCR   low = 2 upp = 2
@

Verificação
(*Model Checking*)

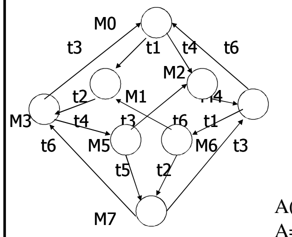
- Exemplo (Leitores/Escritores)

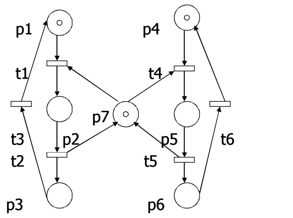


Arquivo .prf (descrição da verificação)
s1 sat? AG(P1 VP2)
..s1 sat? (P1 VP2)
...s1 sat? P1
...s1 sat P1 : FALSE
.......s1 sat? P2
.......s1 sat P2 : FALSE
..s1 sat (P1 VP2): FALSE
s1 sat AG(P1 VP2): **FALSE**

Descrição Linear e Objetos Estruturais

•Grafo de Marcações Acessíveis (Alcançáveis)





$A(N,M_0)=\{M_0,M_1,M_2,M_3,M_4,M_5,M_6,M_7\}$

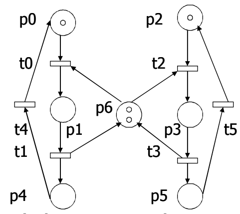
A=RS - *reachability set*

Redes de Petri

- Solução da Equação de Estado

$\boxed{M'(pi)=M0(pi) + C.\overline{S},\ \forall\ pi \in P}$

- $AL^{\aleph}(N,M_0)=LRS^{\aleph} = \{M \in \aleph^{|P|} \mid$ $\exists\, \overline{s} \in \aleph^{|T|}$ onde $M'(pi)=M0(pi) + C.\overline{S}\}$
- $RS(N) \subseteq LRS^{\aleph}(N)$
- Condição suficiente para não-alcaçabilidade (polinomial)
- Condição necessária para alcaçabilidade
- **NP-Completo para redes genéricas**



$M'=(0,0,0,1,1,0,1)$

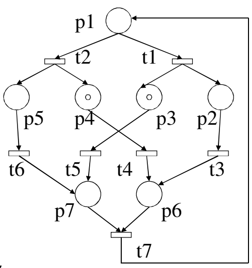
$\overline{s} = (s_0,s_0,1+s_3,s_3,s_0-1,s_3)$

Uma solução:

$\overline{s} = (1,1,1,0,0,0)$

Redes de Petri

- Solução da Equação de Estado
- $RS(N) \subseteq LRS^{\aleph}(N)$
- Condição suficiente para não-alcaçabilidade
- Condição necessária para alcaçabilidade

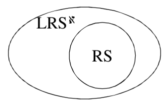


- para $M'=(0,1,0,0,1,0,0)$,

$\overline{s}=(1,1,0,2,2,0,2)$ é solução da ES, mas $M' \notin RS(N, M_0)$

Redes de Petri

- Solução da Equação de Estado
- $RS(N) \subseteq LRS^{\aleph}(N)$
- Condição suficiente para não-alcaçabilidade
- Condição necessária para alcaçabilidade

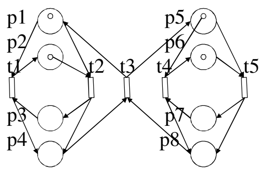


Redes de Petri

- Solução da Equação de Estado

$\boxed{M'(pi)=M0(pi) + C.\overline{S},\ \forall\ pi \in P}$

- $AL^{\Re}(N,M_0)=LRS^{\Re} = \{M \in \Re^{|P|} \mid$ $\exists\, \overline{s} \in \Re^{|T|}$ onde $M'(pi)=M0(pi) + C.\overline{S}\}$
- $RS(N) \subseteq LRS^{\aleph}(N) \subseteq LRS^{\Re}(N)$
- Condição suficiente para não-alcaçabilidade
- Condição necessária para alcaçabilidade
- **Polinomial para redes genéricas (método de eliminação de Gauss)**



$M'=(1,0,1,0,1,1,0,0)$

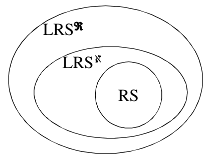
$M' \notin RS$, $M' \notin LRS^{\aleph}$, mas

$M' \in LRS^{\Re}$

$\overline{s} = (0,1,1,1/2,1/2)$

Redes de Petri

- Solução da Equação de Estado
- $RS(N) \subseteq LRS^{\aleph}(N) \subseteq LRS^{\Re}(N)$
- Condição suficiente para não-alcaçabilidade
- Condição necessária para alcaçabilidade



Redes de Petri

- Propriedades que se podem analisar com a Equação de Estados
  - Condição necessária para alcançabilidade
  - Condição suficiente para não-alcançabiliade
  - O limite de p é igual a k? Não existência de M tal que M(p) > k (condição suficiente)
  - Qual é o limite de p? (condição suficiente)
  - É um par de lugares $P'=(p_i,p_j)\subseteq P$ mutuamente exclusivo? Não existência de M tal que $M(p_i)+M(p_i)>1$
  - Deadlock freedom? Não existência de M onde nenhuma transição esteja habilitada (condição suficiente)

Redes de Petri

- É o limite de p igual a k? Não existência de M tal que M(p) > k (condição suficiente)

$M'(pi)=M0(pi) + C.\overline{S},\ \forall\ pi \in P$

Se $\nexists$ $M'(p)>k$, então $M'(p)=k$

- Exemplo:

É o limite de $p_3 = 4$?

Se não existir $M(p_3) = 4$ como solução de $M'(pi)=M0(pi) + C.\overline{S},\ \forall\ pi \in P$, então se sabe que marcação de p_3 não chega a 4

Redes de Petri

- Qual é o limite de p? (condição suficiente)

O limite de p é $b[p]=\max\{M(p) \mid M \in RS(N)\}$

$$\text{Seja } 1_{pi}=[\overset{p_0}{0}, \ldots, 0, \overset{p_i}{1}, 0, \ldots, \overset{p_m}{0}]$$

$sb[p]=\max\{1_{pi}.M \mid M-C.\overline{S}=M_0 \wedge M, S\geq 0\}$

Como a solução de $M(pi)=M_0(pi) + C.\overline{S},\ \forall\ pi \in P$ pode gerar soluções espúrias M pode ser uma delas, mas como $RS(N) \subseteq LRS^{x}(N)$, portanto $sb[p] \geq b[p]$

Redes de Petri

- É um par de lugares $P'=(p_l,p_j)\subseteq P$ mutuamente exclusivo? Não existência de M tal que $M(p_i)+M(p_i)>1$

Se $\nexists$ M tal que

$M(p_l)+M(p_j)>1$

$\forall\ M \in LRS^{x}=\{M \mid M(pi) = M_0(pi) + C.\overline{S},\ \forall\ p_i \in P,\ M \geq 1_{pi,pj}\ S\geq 0\}$

Então p_l,p_j são mutamente exclusivos

Redes de Petri

- É um sub-conjunto de lugares $P' \subseteq P$ mutuamente exclusivo?

Se $\nexists$ $M(pi)>1,\ \forall\ M \in LRS^{x}=\{M \mid M(pi)=M_0(pi) + C.\overline{S},\ M \geq 1_{pi}\ S\geq 0,\ \forall\ p_i \in P\}$

Ou seja, sendo a rede é *safe.*

Se $\nexists$ M tal que

$\sum M(p_l)>1,\ \forall\ p_l \in P'$

$\forall\ M \in LRS^{x}=\{M \mid M(pi) = M_0(pi) + C.\overline{S},\ \forall\ p_i \in P\}$

Então os lugares P' são mutamente exclusivos

Redes de Petri

- Deadlock freedom? Não existência de M onde nenhuma transição esteja habilitada (condição suficiente)

Se $\nexists$ M tal que $M=M_0 + C.\overline{S(t_j)},\ \forall\ t_j \in T,\ M, S\geq 0$

Não há uma marcação obtida pelo disparo de qualquer transição t_j. Ou seja, não há *deadlock*.

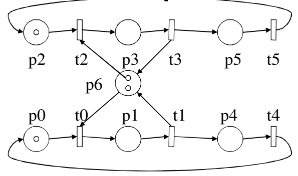
Invariantes

- Invariantes de Transição

M_0 [s> M e M= M_0

Então, de $M=M_0 + C.\overline{S} \Rightarrow M - M_0 = C.\overline{S}$, temos

$C.\overline{S} = 0 \Rightarrow C.\ I_t = 0$



I. s0 = s1 = s4
II. s2 = s3 = s5

Para s0 = 1 e s2= 0
- It1 = (1,1,0,0,1,0)

Para s0 = 0 e s2 =1
- It2=(0,0,1,1,0,1)
- It3 = It1 + It2 = (1,1,1,1,1)

Invariantes

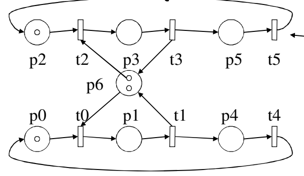
- Invariantes de Transição

M_0 [s> M e M= M_0

Então, de $M=M_0 + C.\overline{S} \Rightarrow M - M_0 = C.\overline{S}$, temos

$C.\overline{S} = 0 \Rightarrow C.\ I_t = 0$

I. T-semiflow (vetor)
II. Lei de repetição (equação)
III. Componente repetitivo (uma rede)



•It1 = (1,1,0,0,1,0)

Invariantes de Lugar

- N é dita conservativa se $\sum_{pi \in P} w_i \times M(p_i) = cte$
- Dado que toda marcação deve satisfazer a equação de estado, podemos fazer: $M' = M_0 + C.S$
- Portanto, para qualquer marcação inicial M_0

  $W^T.M' = W^T.M_0 = W^T.(M_0 + C.\overline{S})$

  $W^T.M_0 = W^T.M_0 + W^T.\ C.\overline{S}$

  $W^T.\ C.\overline{S} = 0$
- Como $\overline{S} \neq 0 \Rightarrow W^T.\ C = 0$
- Invariante de Lugar - Ip . C = 0

Invariantes de Lugar

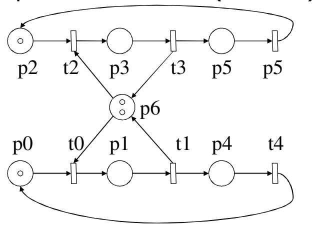
- T-*semiflow* (um vetor)
- Lei de repetição de transições (uma equação)
- Componente repetitivo (uma rede)
- T-*semiflow*: Seja um vetor It ≥ 0, tal que s_i ≥ . It é chamado de t-*semiflow* sse It . C = 0
- Suporte de um t-*semiflow*: Seja It um p-*semiflow*, ST é definido como suporte de t-*semiflow* ST={t | It(t)>0}
- Minimal T-semi*flow*: Um t-*semiflow* é minimo sse seu suporte nenhum outro suporte de qualquer outro t-*semiflow*.

Invariantes de Lugar

- P-*semiflow* (um vetor)
- Marcação invariante ou lei de conservação de marcas (uma equação)
- Componente conservativo (uma rede)
- P-*semiflow*: Seja um vetor Ip ≥ 0, tal que ω_i ≥ . Ip é chamado de p-*semiflow* sse Ip . C = 0
- Suporte de um p-*semiflow*: Seja Ip um p-*semiflow*, SP é definido como suporte de p-*semiflow* SP={p | IP(p)>0}
- Minimal P-semi*flow*: Um p-*semiflow* é minimo sse seu suporte nenhum outro suporte de qualquer outro p-*semiflow*.

Invariantes de Lugar

- P-*semiflow* (um vetor)
- Marcação invariante ou lei de conservação de marcas (uma equação)
- Componente conservativo (uma rede)

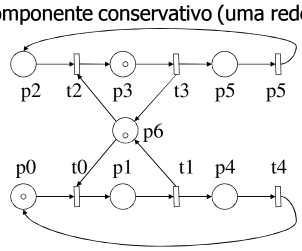


Ip . C = 0

	p0	p1	p2	p3	p4	p5	p6
Ip1=	1	1	0	0	1	0	0
Ip2=	0	0	1	1	0	1	0
Ip3=	0	1	0	1	0	0	1

Invariantes de Lugar

- P-*semiflow* (um vetor)
- Marcação invariante ou lei de conservação de marcas (uma equação)
- Componente conservativo (uma rede)

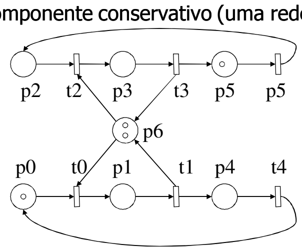


Ip . C = 0

p0,p1,p2,p3,p4,p5,p6
Ip1= |1, 1, 0, 0, 1, 0, 0|
Ip2= |0, 0, 1, 1, 0, 1, 0|
Ip3= |0, 1, 0, 1, 0, 0, 1|

Invariantes de Lugar

- P-*semiflow* (um vetor)
- Marcação invariante ou lei de conservação de marcas (uma equação)
- Componente conservativo (uma rede)

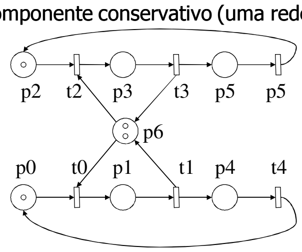


Ip . C = 0

p0,p1,p2,p3,p4,p5,p6
Ip1= |1, 1, 0, 0, 1, 0, 0|
Ip2= |0, 0, 1, 1, 0, 1, 0|
Ip3= |0, 1, 0, 1, 0, 0, 1|

Invariantes de Lugar

- P-*semiflow* (um vetor)
- Marcação invariante ou lei de conservação de marcas (uma equação)
- Componente conservativo (uma rede)

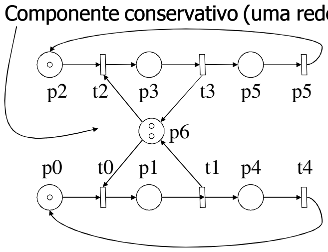


Ip . C = 0

p0,p1,p2,p3,p4,p5,p6
Ip1= |1, 1, 0, 0, 1, 0, 0|
Ip2= |0, 0, 1, 1, 0, 1, 0|
Ip3= |0, 1, 0, 1, 0, 0, 1|

Invariantes de Lugar

- P-*semiflow* (um vetor)
- Marcação invariante ou lei de conservação de marcas (uma equação)
- Componente conservativo (uma rede)



Ip . C = 0

p0,p1,p2,p3,p4,p5,p6
Ip1= |1, 1, 0, 0, 1, 0, 0|
Ip2= |0, 0, 1, 1, 0, 1, 0|
Ip3= |0, 1, 0, 1, 0, 0, 1|

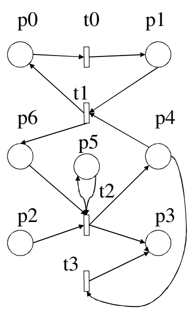
Invariantes de Lugar

Algoritmo para o cálculo dos mínimos p-*semiflows*

- Input:matriz de incidência C, onde m =|P | e n=|T |
- Ouput: conjunto de p-*semiflows*
  - A:=C , W:=I_n - I_n indentidade de dimensão n
  - for i:=1 to m, faça:
    - Some a matriz [A|W] todas as linhas que são combinação linear de pares de linha de [A|W] e anulam a i-ésima coluna de A
    - Elimine de [A|W] as linhas em que a coluna i-ésima não é nula.
  - Remova as linhas de W que os suportes não são mínimos e divida cada uma pelo MDC dos elementos não nulos. (todos os mínimos estão aqui)

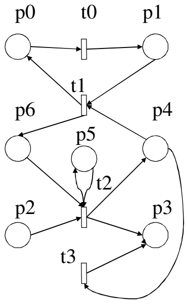
Invariantes de Lugar

0	1	2	3	0	1	2	3	4	5	6	
-1	1	0	0	1	0	0	0	0	0	0	0
1	-1	0	0	0	1	0	0	0	0	0	1
0	0	-1	1	0	0	1	0	0	0	0	2
0	0	1	-1	0	0	0	1	0	0	0	3
0	-1	1	0	0	0	0	0	1	0	0	4
0	0	0	0	0	0	0	0	0	1	0	5
0	1	-1	0	0	0	0	0	0	0	1	6



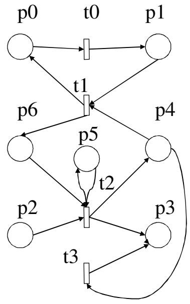
Invariantes de Lugar

0 1 2 3	0 1 2 3 4 5 6	
-1 1 0 0	1 0 0 0 0 0 0	0
1 -1 0 0	0 1 0 0 0 0 0	1
0 0 -1 1	0 0 1 0 0 0 0	2
0 0 1 -1	0 0 0 1 0 0 0	3
0 -1 1 0	0 0 0 0 1 0 0	4
0 0 0 0	0 0 0 0 0 1 0	5
0 1 -1 0	0 0 0 0 0 0 1	6



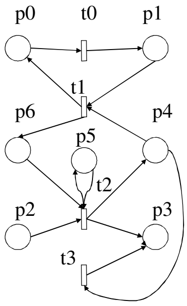
Invariantes de Lugar

0 1 2 3	0 1 2 3 4 5 6	
-1 1 0 0	1 0 0 0 0 0 0	0
1 -1 0 0	0 1 0 0 0 0 0	1
0 0 -1 1	0 0 1 0 0 0 0	2
0 0 1 -1	0 0 0 1 0 0 0	3
0 -1 1 0	0 0 0 0 1 0 0	4
0 0 0 0	0 0 0 0 0 1 0	5
0 1 -1 0	0 0 0 0 0 0 1	6
0 0 0 0	1 1 0 0 0 0 0	



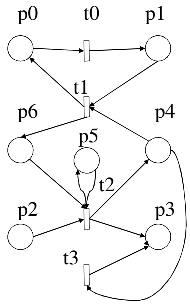
Invariantes de Lugar

0 1 2 3	0 1 2 3 4 5 6	
0 0 -1 1	0 0 1 0 0 0 0	2
0 0 1 -1	0 0 0 1 0 0 0	3
0 -1 1 0	0 0 0 0 1 0 0	4
0 0 0 0	0 0 0 0 0 1 0	5
0 1 -1 0	0 0 0 0 0 0 1	6
0 0 0 0	1 1 0 0 0 0 0	
0 0 0 0	0 0 0 0 1 0 1	



Invariantes de Lugar

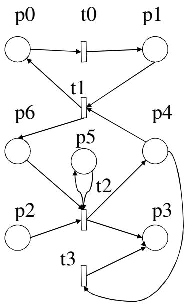
0 1 2 3	0 1 2 3 4 5 6	
0 0 -1 1	0 0 1 0 0 0 0	2
0 0 1 -1	0 0 0 1 0 0 0	3
0 0 0 0	0 0 0 0 0 1 0	5
0 0 0 0	1 1 0 0 0 0 0	
0 0 0 0	0 0 0 0 1 0 1	
0 0 0 0	0 0 1 1 0 0 0	



Invariantes de Lugar

0 1 2 3	0 1 2 3 4 5 6	
0 0 0 0	0 0 0 0 0 1 0	5
0 0 0 0	1 1 0 0 0 0 0	
0 0 0 0	0 0 0 0 1 0 1	
0 0 0 0	0 0 1 1 0 0 0	

Ip0={p5}
Ip1={p0,p1}
Ip2={p4,p6}
Ip3={p2,p3}



Redes de Petri:
Modelagem, Análise Qualitativa e de Desempenho

Por
Paulo Maciel
Centro de Informática
Universidade Federal de Pernambuco

Propriedades

- Verificação
- Prova
- Análise -
- Validação

Propriedades

- **Verificação** - é baseada em algorítmos determinísticos que objetivam responder se um modelo possui ou não uma determiminada propriedade.
  - *one sided* - respostas: sim ou não sei
  - mais genérico - respostas: sim ou não (maior complexidade)
- **Prova** - é baseada em algorítmos não-determinísticos que procuram argumentos formais para a afirmação ou não da existência de uma propriedade em um modelo

Propriedades

- **Análise** - é baseada em algorítmos que não objetivam a verificação de uma determinada propriedade. Ao contrário, são algorítmos mais genéricos que fornecem informações sobre diversas propriedades e seus resultados podem ser utlizados como base para algorítmos de verificação.
  - Mesmo que a resposta seja sim ou não, não há prioridade de um sim sobre o não
  - Também são usadas para responder questões do tipo: Quais são os conjuntos de lugares cujo somatório de marcas permanece constante?

Propriedades

- **Validação** - é o processo de obtenção da confiabilidade de que sistema funciona como desejado.
  - O comportamento do sistema é comparado com a expectativa da pessoa a qual está validando o sistema.
  - É um processo inerentemente informal.
  - A princípio é um processo realizado sobre o sistema já implementado. No entano, pode também ser realizado sobre um modelo (simulação).
  - Verificação ou análise não garantem que o sistema funciona como desejado. De fato, obtêm as propriedades dos modelos.

Propriedades Comportamentais

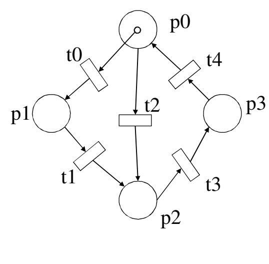
- Alcançabilidade (*Reachability*)

Indica a possibilidade de atingirmos uma determinada marcação pelo disparo de um número finito de transições, a partir de uma marcação inicial.

- Marcação Alcançável: seja $M_i[t_j>M_k$ e $M_k[t_h>M_l$ então $M_i[t_j\, t_h>M_l$. Por recorrência designamos o disparo de uma seqüência $s \in T^*$ por $M[s>M'$. Dizemos que M' é alcançável de M. O conjunto de todas as possíveis marcações alcançáveis de a partir de M_0 na rede $N=(R, M_0)$ é denotado por $A(R; M_0)=\{M' \in \aleph^m \mid M_0[s>M'\}$ (RS). $m = |P|$

Propriedades Comportamentais

- Alcançabilidade (*Reachability*)

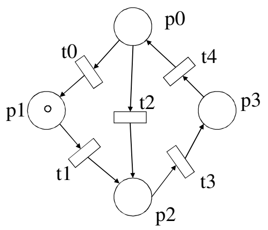


•M'=(0,0,0,1) é acessível a partir de M_0?

•É. Pelo disparo de $s'=t_0t_1t_3$ e de $s''=t_2t_3$

Propriedades Comportamentais

- Alcançabilidade (*Reachability*)

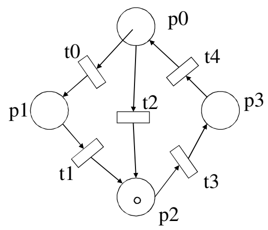


•M'=(0,0,0,1) é acessível a partir de M_0?

•É. Pelo disparo de $s'=t_0t_1t_3$ e de $s''=t_2t_3$

Propriedades Comportamentais

- Alcançabilidade (*Reachability*)

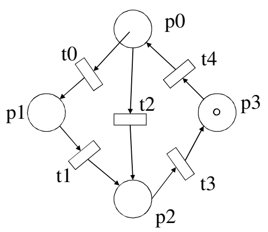


•M'=(0,0,0,1) é acessível a partir de M_0?

•É. Pelo disparo de $s'=t_0t_1t_3$ e de $s''=t_2t_3$

Propriedades Comportamentais

- Alcançabilidade (*Reachability*)

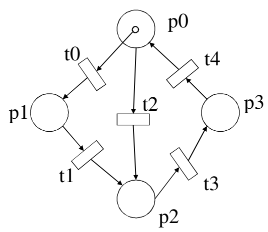


•M'=(0,0,0,1) é acessível a partir de M_0?

•É. Pelo disparo de $s'=t_0t_1t_3$ e de $s''=t_2t_3$

Propriedades Comportamentais

- Alcançabilidade (*Reachability*)

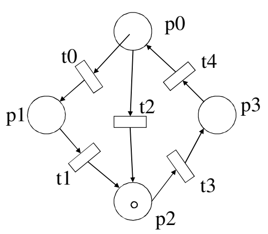


•M'=(0,0,0,1) é acessível a partir de M_0?

•É. Pelo disparo de $s'=t_0t_1t_3$ e de $s''=t_2t_3$

Propriedades Comportamentais

- Alcançabilidade (*Reachability*)

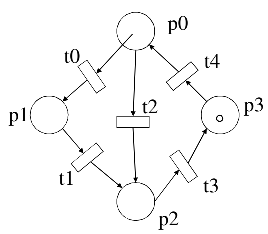


•M'=(0,0,0,1) é acessível a partir de M_0?

•É. Pelo disparo de $s'=t_0t_1t_3$ e de $s''=t_2t_3$

Propriedades Comportamentais

- Alcançabilidade (*Reachability*)



•M'=(0,0,0,1) é acessível a partir de M_0?

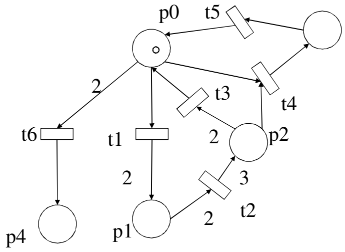
•É. Pelo disparo de $s'=t_0t_1t_3$ e de $s''=t_2t_3$

Propriedades Comportamentais

- Alcançabilidade (*Reachability*)
  - Problemas associados: alcançbilidade de sub-marcação e verificação de *deadlock*

Propriedades Comportamentais

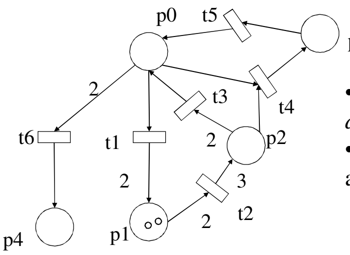
- Alcançabilidade (*Reachability*)



•M'=(0,0,0,0,1) é um *deadlock*.
•Esta marcação é alcançável?

Propriedades Comportamentais

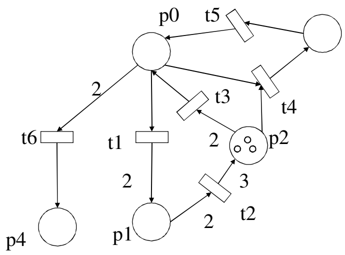
- Alcançabilidade (*Reachability*)



•M'=(0,0,0,0,1) é um *deadlock*.
•Esta marcação é alcançável?

Propriedades Comportamentais

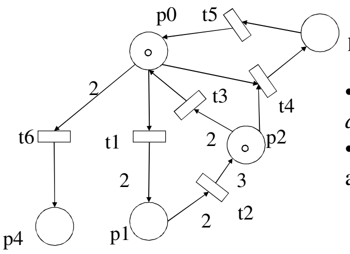
- Alcançabilidade (*Reachability*)



•M'=(0,0,0,0,1) é um *deadlock*.
•Esta marcação é alcançável?

Propriedades Comportamentais

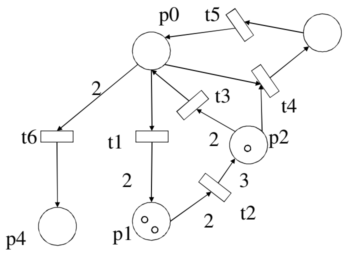
- Alcançabilidade (*Reachability*)



•M'=(0,0,0,0,1) é um *deadlock*.
•Esta marcação é alcançável?

Propriedades Comportamentais

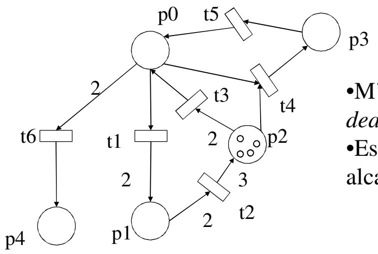
- Alcançabilidade (*Reachability*)



•M'=(0,0,0,0,1) é um *deadlock*.
•Esta marcação é alcançável?

Propriedades Comportamentais

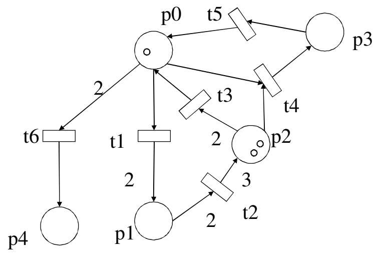
- Alcançabilidade (*Reachability*)



•M'=(0,0,0,0,1) é um *deadlock*.
•Esta marcação é alcançável?

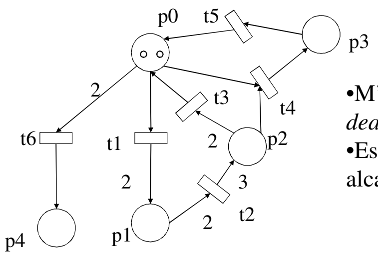
Propriedades Comportamentais

- Alcançabilidade (*Reachability*)



Propriedades Comportamentais

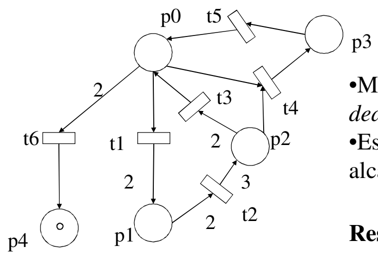
- Alcançabilidade (*Reachability*)



•M'=(0,0,0,0,1) é um *deadlock*.
•Esta marcação é alcançável?

Propriedades Comportamentais

- Alcançabilidade (*Reachability*)



•M'=(0,0,0,0,1) é um *deadlock*.
•Esta marcação é alcançável?

Resposta: É

Propriedades Comportamentais

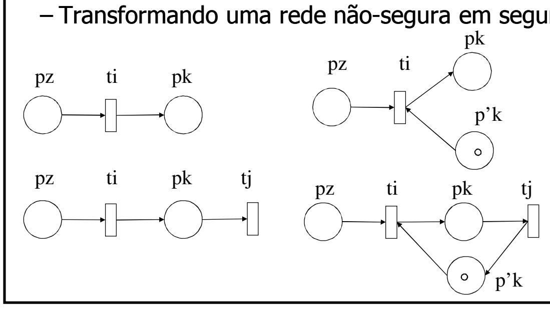
- Segurança (*Safeness*)

Transformando uma rede não-segura em segura

– Se $p_k \in O(t_i)$ e $p_k \notin I(t_i)$, então cria-se o lugar $p'_k \in I(t_i)$ e $M(p'_k)=1$
– Se $p_k \in I(t_j)$, então $p'_k \in O(t_j)$

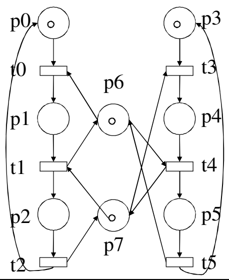
Propriedades Comportamentais

- Segurança (*Safeness*)
  – Transformando uma rede não-segura em segura



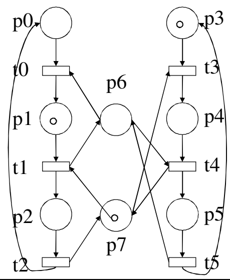
Propriedades Comportamentais

- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



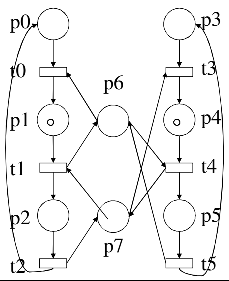
Propriedades Comportamentais

- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



Propriedades Comportamentais

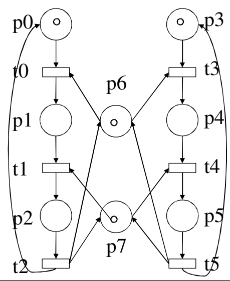
- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



Deadlock

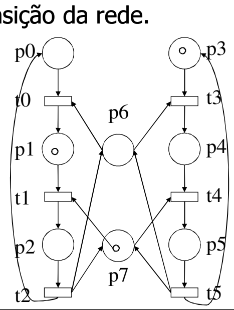
Propriedades Comportamentais

- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



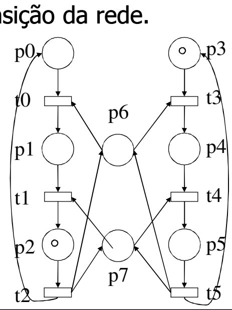
Propriedades Comportamentais

- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



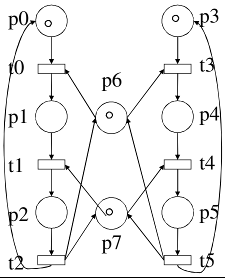
Propriedades Comportamentais

- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



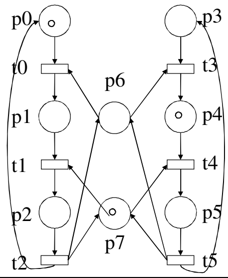
Propriedades Comportamentais

- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



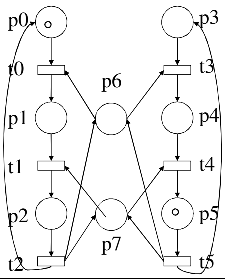
Propriedades Comportamentais

- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



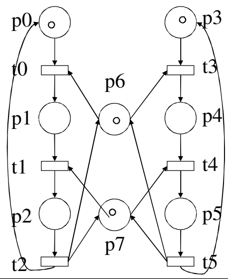
Propriedades Comportamentais

- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



Propriedades Comportamentais

- *Deadlock* - Indica a impossibilidade do disparo de qualquer transição da rede.



Sem *deadlock*

Propriedades Comportamentais

- Transição Potencialmente Disparável:
  chamamos uma trnasição t_j potencialmente disparável em uma marcação M_0 se $\exists$ M' $\in$ A(R; M_0) tal que M'[t_j>.

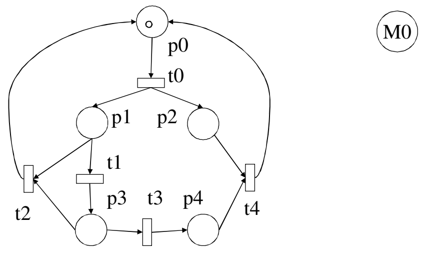
- Rede *Live*: uma rede N=(R,M_0) é dita *live* se $\exists$ s $\in$ L(N) tal que $t_i \in$ s, $\forall t_i \in$ T.

<u>Outra definição</u>

- Rede *Live*: uma rede N=(R,M_0) é dita *live* se $\exists$ s $\in$ T* tal que $t_i \in$ s, M'[s>, $\forall$M' $\in$ A(R, M_0), $\forall t_i \in$ T.

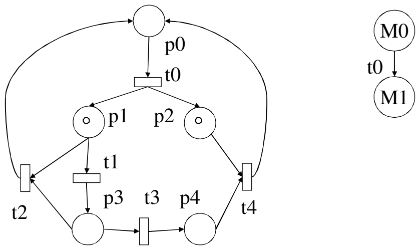
Propriedades Comportamentais

- *Liveness* é mais forte do que *deadlock freedom*



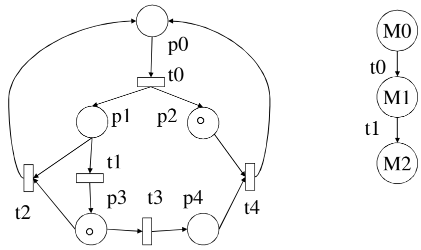
Propriedades Comportamentais

- *Liveness* é mais forte do que *deadlock freedom*



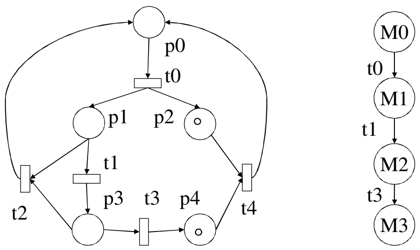
Propriedades Comportamentais

- *Liveness* é mais forte do que *deadlock freedom*



Propriedades Comportamentais

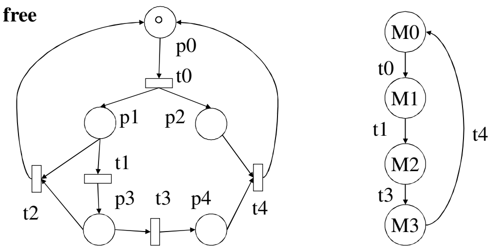
- *Liveness* é mais forte do que *deadlock freedom*



Propriedades Comportamentais

- *Liveness* é mais forte do que *deadlock freedom*

•Deadlock free
•Non-live



Propriedades Comportamentais

- *Liveness* é uma propriedade "cara" de se analisar.
- Uma transição t_i tem sido classificada em níveis.
- Níveis de *liveness*:
  0 . Morta (*dead*) ou nível N0, se $\nexists$ s $\in$ L(R,M_0), tal que $t_i \in$ s, ou seja, $\nexists$ M' $\in$ A(R,M_0), tal que M'[t_i> .
  1 . N1-*live*, se t_i pode ser disparada pelo menos uma vez em alguma seqüência s $\in$ L(R,M_0).
  2 . N2-*live*, se t_i pode ser disparada pelo menos *k* vez em alguma seqüência s $\in$ L(R,M_0).
  3 . N3-*live*, se t_i aparece um número infinito de vezes em alguma seqüência s $\in$ L(R,M_0).
  4 . N4-*live*, ou simplesmente *live*, se t_i é N1-*live* $\forall$ M' $\in$ A(R,M_0),

Propriedades Comportamentais

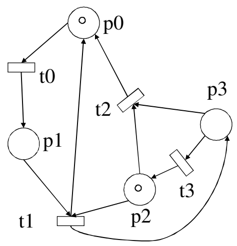
- Cobertura (*coverability*) de uma marcação: seja a marcação M' em uma rede N=(R,M_0). M' é dita coberta se existe M'' $\in$ A(R, M_0) tal que M''(p_i)$\geq$M'(p_i) $\forall p_i \in$ P.

Propriedades Comportamentais

- Reversibilidade: inicialmente uma rede é dita reversível se para cada marcação M_i no conjunto das marcações acessíveis a marcação inicial pode ser novamente alcançada.
- *Home-State*: seja uma marcação $M_k \in A(R,M_0)$. M_k é denominada *home-state* se M_i $[>M_k$, $\forall M_i \in A(R,M_0)$.
- Reversibilidade: uma rede $N=(R,M_0)$ é reversível se $\exists$ M_k, tal que $M_i[>M_k$, $\forall M_i \in A(R,M_0)$. (relaxamento do que foi descrito acima)

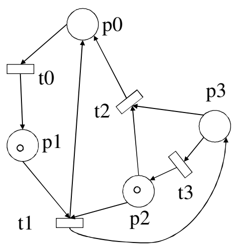
Propriedades Comportamentais

- Reversibilidade



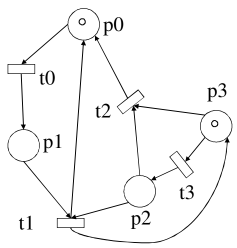
Propriedades Comportamentais

- Reversibilidade



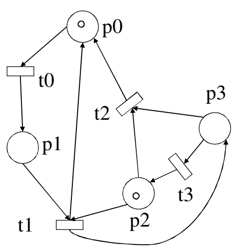
Propriedades Comportamentais

- Reversibilidade



Propriedades Comportamentais

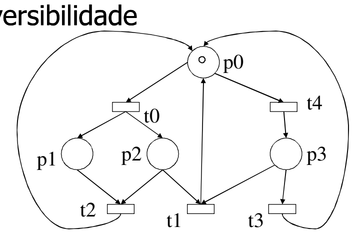
- Reversibilidade



Reversível

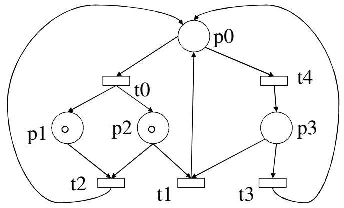
Propriedades Comportamentais

- Reversibilidade



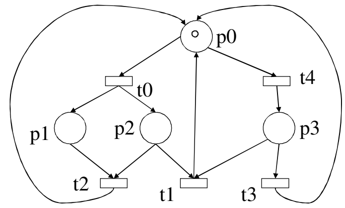
Propriedades Comportamentais

- Reversibilidade



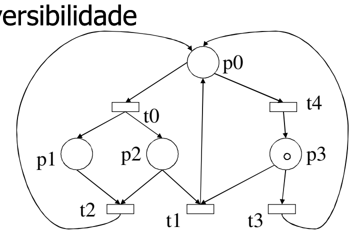
Propriedades Comportamentais

- Reversibilidade



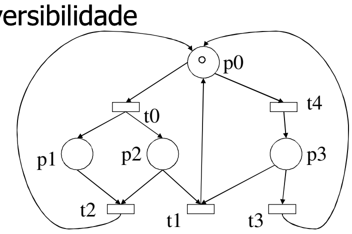
Propriedades Comportamentais

- Reversibilidade



Propriedades Comportamentais

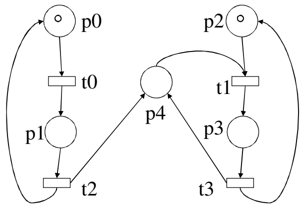
- Reversibilidade



Reversível

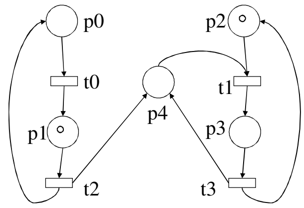
Propriedades Comportamentais

- Reversibilidade



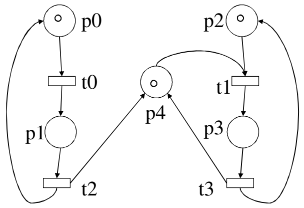
Propriedades Comportamentais

- Reversibilidade



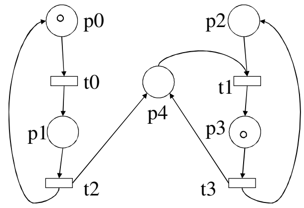
Propriedades Comportamentais

- Reversibilidade



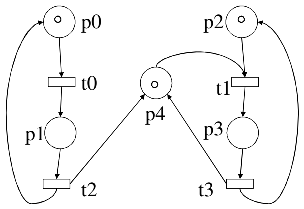
Propriedades Comportamentais

- Reversibilidade



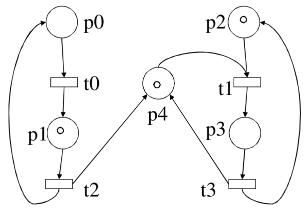
Propriedades Comportamentais

- Reversibilidade



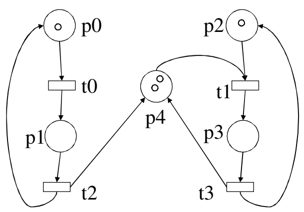
Propriedades Comportamentais

- Reversibilidade



Propriedades Comportamentais

- Reversibilidade



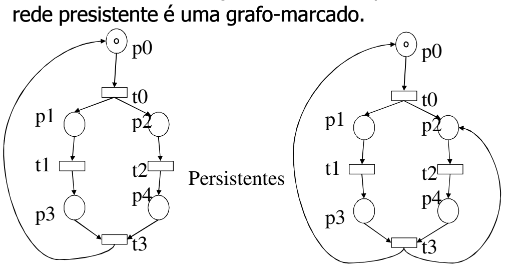
Irreversível

Propriedades Comportamentais

- Persistência: uma rede é dita persistente se para qualquer par de transições o disparo de uma não desabilita a outra.
- Persistência: seja uma rede $N=(R,M_0)$. N é dita persistente se para todo par $(t_i, t_j) \in T^2$ tal que $\exists$ $M_k[t_i>$ e $M_k[t_j> \Rightarrow M_k[t_i>M''$, $M''[t_j>$ e vice-versa. Onde $M_k, M'' \in A(R,M_0)$.

Propriedades Comportamentais

- Persistência: todo grafo-marcado é persistente. Nem toda rede presistente é uma grafo-marcado.



Grafo-marcado

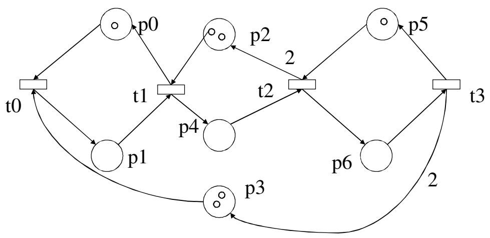
Não é grafo-marcado

Propriedades Comportamentais

- Justiça Limitada (*Bounded Fairness*): uma rede é dita *bounded fair* se para qualquer par de transições o número de disparo de uma enquanto a outra não dispara é limitado.
- Justiça Limitada: seja uma rede $N=(R,M_0)$. N é dita *bounded fair* se todo par $(t_i, t_j) \in T^2$ é *B-fair*

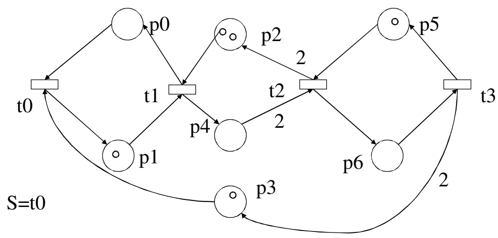
Propriedades Comportamentais

- Justiça Limitada



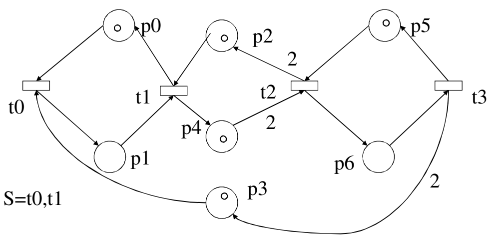
Propriedades Comportamentais

- Justiça Limitada



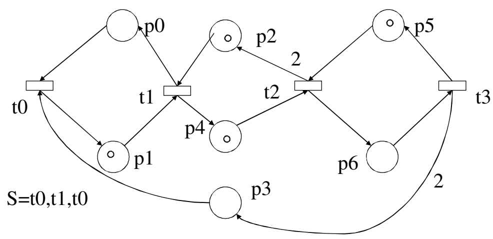
Propriedades Comportamentais

- Justiça Limitada



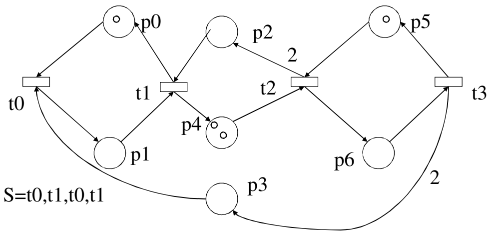
Propriedades Comportamentais

- Justiça Limitada



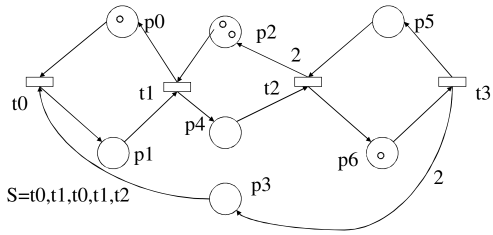
Propriedades Comportamentais

- Justiça Limitada



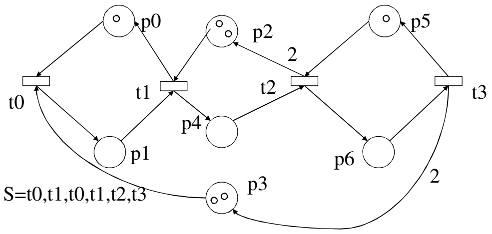
Propriedades Comportamentais

- Justiça Limitada



Propriedades Comportamentais

- Justiça Limitada



Propriedades Comportamentais

- Distância Sincrônica: relaciona o grau de dependência mútua entre eventos.
- Distância Sincrônica: seja uma rede $N=(R,M_0)$, t_i, $t_j \in T$ é, $L(N)$ a linguagem de N. $d(t_i, t_j)=\max_{s \in L(N)} \{|s(t_i) - s(t_j)|\}$ é definida como a distância sincrônica entre as transições.
  - Se $d(t_i, t_j) \rightarrow \infty \Rightarrow$ que o modelo é *unfair.*
  - Se $d(t_i, t_j) = 1 \Rightarrow$ que a cada disparo de t_i, t_j dispara.

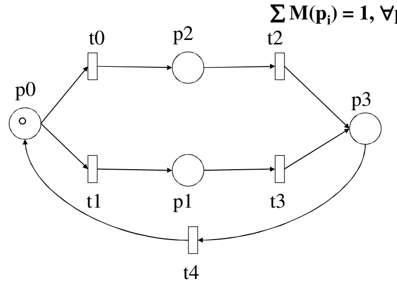
Propriedades Comportamentais

- Conservação: está relacionada ao somatório de marcas a medida que as transições são disparadas.
- Rede Estritamente Conservativa: seja uma rede $N=(R,M_0)$ e $M \in A(R,M_0)$ uma marcação alcançável. N é dita estritamente conservativa $\sum M(p_i) = \sum M_0(p_i)$, $\forall p_i \in P$ e $\forall M \in A(R,M_0)$.

Propriedades Comportamentais

- Rede Estritamente Conservativa

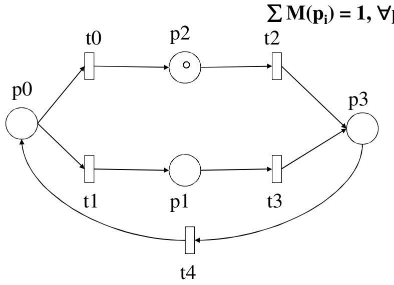
$\sum \mathbf{M(p_i) = 1, \forall p_i \in P}$



Propriedades Comportamentais

- Rede Estritamente Conservativa

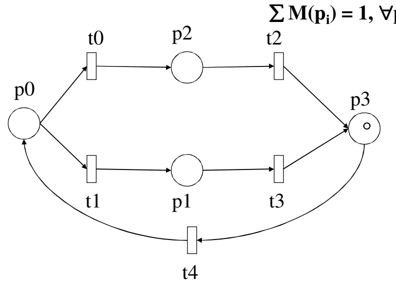
$\sum M(p_i) = 1, \forall p_i \in P$



Propriedades Comportamentais

- Rede Estritamente Conservativa

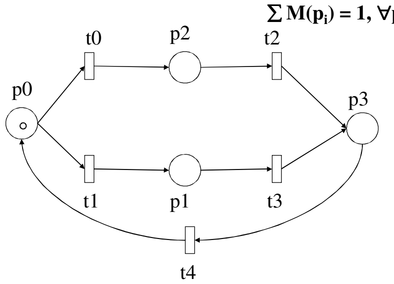
$\sum M(p_i) = 1, \forall p_i \in P$



Propriedades Comportamentais

- Rede Estritamente Conservativa

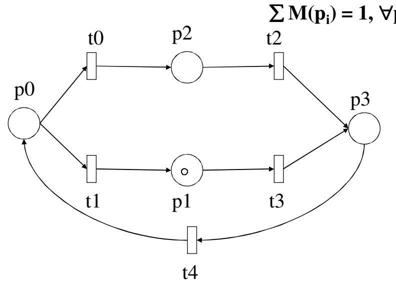
$\sum M(p_i) = 1, \forall p_i \in P$



Propriedades Comportamentais

- Rede Estritamente Conservativa

$\sum M(p_i) = 1, \forall p_i \in P$



Propriedades Comportamentais

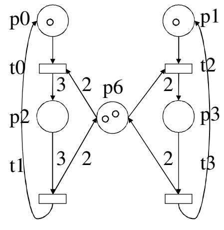
- Conservação: está relacionada ao somatório de marcas a medida que as transições são disparadas.

- Rede Conservativa: seja uma rede $N=(R,M_0)$, $M \in A(R,M_0)$ uma marcação alcançável e $W=(w_1,\ldots,w_n)$, onde $n=\#P$. N é dita conservativa $\sum_{pi \in P} w_i \times M(p_i) = \sum_{pi \in P} w_i \times M_0(p_i, \forall M \in A(R,M_0)$.

Propriedades Comportamentais

- Conservação

$\sum_{pi \in P} w_4^i \times \mathbf{m_0(p_i) = 17, \forall p_i \in P}$

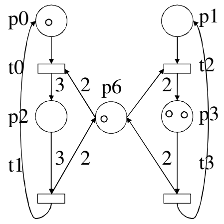


$W_1 = (3,0,1,0,0)$
$W_2 = (0,2,0,1,0)$
$W_3 = (0,0,4,3,6)$
$W_4 = (3,2,5,4,6)$

Propriedades Comportamentais

- Conservação

$\Sigma_{p_i \in P} w_4^i \times \mathbf{m_0(p_i) = 17, \forall p_i \in P}$

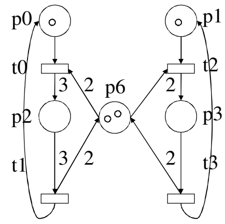


$W_1 = (3,0,1,0,0)$
$W_2 = (0,2,0,1,0)$
$W_3 = (0,0,4,3,6)$
$W_4 = (3,2,5,4,6)$

Propriedades Comportamentais

- Conservação

$\Sigma_{p_i \in P} w_4^i \times \mathbf{m_0(p_i) = 17, \forall p_i \in P}$

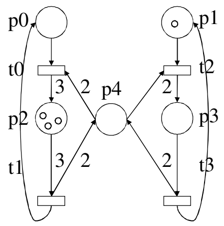


$W_1 = (3,0,1,0,0)$
$W_2 = (0,2,0,1,0)$
$W_3 = (0,0,4,3,6)$
$W_4 = (3,2,5,4,6)$

Propriedades Comportamentais

- Conservação

$\Sigma_{p_i \in P} w_4^i \times \mathbf{m_0(p_i) = 17, \forall p_i \in P}$

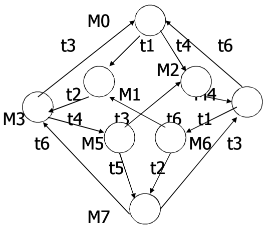


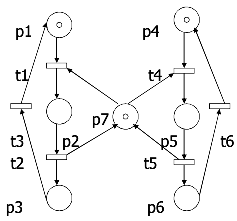
$W_1 = (3,0,1,0,0)$
$W_2 = (0,2,0,1,0)$
$W_3 = (0,0,4,3,6)$
$W_4 = (3,2,5,4,6)$

Análise de Proprieadades Comportamentais

- Grafo de Alcançabilidade: seja uma rede marcada $N=(R,M_0)$. $RG(R, M_0)=(RS,ARCS)$ define o grafo de alcançabilidade (*Reachability Graph*), onde RS é o conjunto de vértice e representa o conjunto de alcançabilidade. $ARCS \subseteq RS \times T \times RS$ é uma relação representando os arcos.

Grafo de Marcações Acessíveis (Alcançáveis)

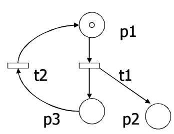




$A(N,M_0)=\{M_0,M_1,M_2,M_3,M_4,M_5,M_6,M_7\}$
A=RS - *reachability set*

Grafo de Marcações Acessíveis (Alcançáveis)

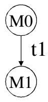
- RG Infinito

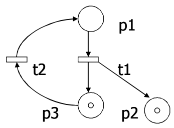


$A(N,M_0)=\{M_0,M_1,M_2,...\}$
A=RS - *reachability set*

Grafo de Marcações Acessíveis (Alcançáveis)

- RG Infinito

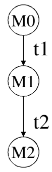


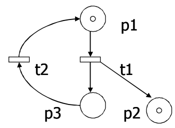


A(N,M_0)={M_0,M_1,M_2,...}
A=RS - *reachability set*

Grafo de Marcações Acessíveis (Alcançáveis)

- RG Infinito

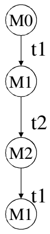


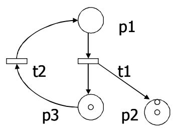


A(N,M_0)={M_0,M_1,M_2,...}
A=RS - *reachability set*

Grafo de Marcações Acessíveis (Alcançáveis)

- RG Infinito

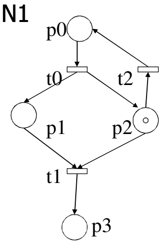




A(N,M_0)={M_0,M_1,M_2,...}
A=RS - *reachability set*

Árvore de Cobertura

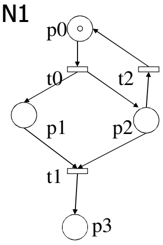
- N1



(p_0, p_1, p_2, p_3)
M_0 = (0, 0, 1, 0)

Árvore de Cobertura

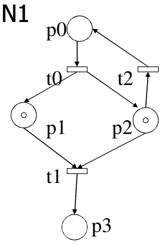
- N1



(p_0, p_1, p_2, p_3)
M_0 = (0, 0, 1, 0)
t2
M_1 =(1, 0, 0, 0)

Árvore de Cobertura

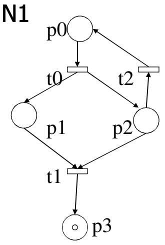
- N1



(p_0, p_1, p_2, p_3)
M_0 = (0, 0, 1, 0)
t2
M_1 =(1, 0, 0, 0)
t0
M_2 =(0, ω, 1, 0)

Árvore de Cobertura

- N1



(p_0, p_1, p_2, p_3)

$M_0 = (0, 0, 1, 0)$

$\downarrow t2$

$M_1 = (1, 0, 0, 0)$

$\downarrow t0$

$M_2 = (0, \omega, 1, 0)$

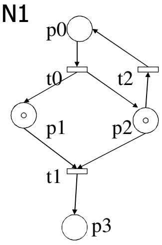
$\downarrow t1$

$M_3 = (0, \omega, 0, 1)$

morta

Árvore de Cobertura

- N1



(p_0, p_1, p_2, p_3)

$M_0 = (0, 0, 1, 0)$

$\downarrow t2$

$M_1 = (1, 0, 0, 0)$

$\downarrow t0$

$M_2 = (0, \omega, 1, 0)$

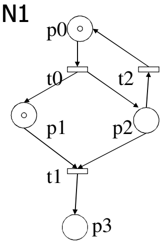
$\downarrow t1$ $\quad \searrow t2$

$M_3 = (0, \omega, 0, 1)$

morta

Árvore de Cobertura

- N1



(p_0, p_1, p_2, p_3)

$M_0 = (0, 0, 1, 0)$

$\downarrow t2$

$M_1 = (1, 0, 0, 0)$

$\downarrow t0$

$M_2 = (0, \omega, 1, 0)$

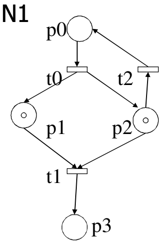
$\downarrow t1$ $\quad \searrow t2$

$M_3 = (0, \omega, 0, 1)$ $\quad M_4 = (1, \omega, 0, 0)$

morta

Árvore de Cobertura

- N1



(p_0, p_1, p_2, p_3)

$M_0 = (0, 0, 1, 0)$

$\downarrow t2$

$M_1 = (1, 0, 0, 0)$

$\downarrow t0$

$M_2 = (0, \omega, 1, 0)$

$\downarrow t1$ $\quad \searrow t2$

$M_3 = (0, \omega, 0, 1)$ $\quad M_4 = (1, \omega, 0, 0)$

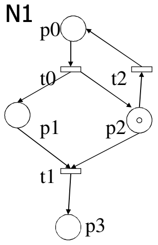
morta $\quad \downarrow t0$

$M_2 = (0, \omega, 1, 0)$

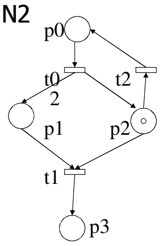
final

Árvore de Cobertura

- N1



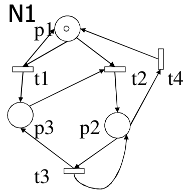
- N2



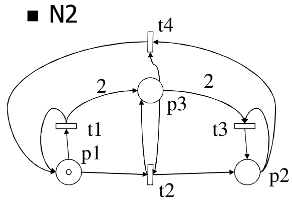
* ambas as redes têem a mesma árvore de cobertura

Árvore de Cobertura

- N1



- N2

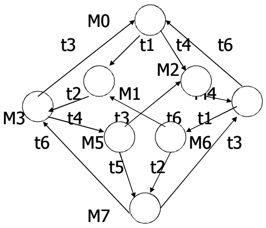


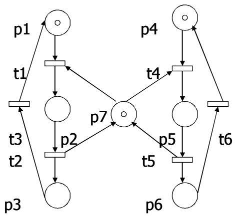
$S = t_1 t_2 t_3$ leva a deadlock

* ambas as redes têem a mesma árvore de cobertura, mas N1 é *live* e N2 não

Descrição Linear e Objetos Estruturais

•Grafo de Marcações Acessíveis (Alcançáveis)





$A(N,M_0)=\{M_0,M_1,M_2,M_3,M_4,M_5,M_6,M_7\}$

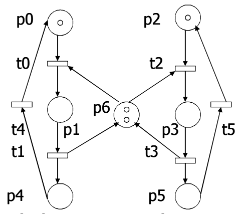
A=RS - *reachability set*

Redes de Petri

- Solução da Equação de Estado

$\boxed{M'(pi)=M0(pi) + C.\overline{S},\ \forall\ pi \in P}$

- $AL^{\bullet}(N,M_0)=LRS^{\bullet}=\{M \in \bullet^{|P|} \mid \exists\ \overline{s} \in \bullet^{|T|}$ onde $M'(pi)=M0(pi) + C.\overline{S}\}$
- $RS(N) \subseteq LRS^{\bullet}(N)$
- Condição suficiente para não-alcaçabilidade (polinomial)
- Condição necessária para alcaçabilidade
- **NP-Completo para redes genéricas**



$M'=(0,0,0,1,1,0,1)$

$\overline{s} = (s_0,s_0,1+s_3,s_3,s_0-1,s_3)$

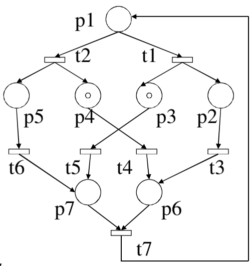
Uma solução:

$\overline{s} = (1,1,1,0,0,0)$

Redes de Petri

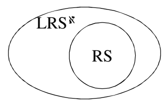
- Solução da Equação de Estado
- $RS(N) \subseteq LRS^{\bullet}(N)$
- Condição suficiente para não-alcaçabilidade
- Condição necessária para alcaçabilidade
- para $M'=(0,1,0,0,1,0,0)$,

$\overline{s}=(1,1,0,2,2,0,2)$ é solução da ES, mas $M' \notin RS(N, M_0)$



Redes de Petri

- Solução da Equação de Estado
- $RS(N) \subseteq LRS^{\bullet}(N)$
- Condição suficiente para não-alcaçabilidade
- Condição necessária para alcaçabilidade

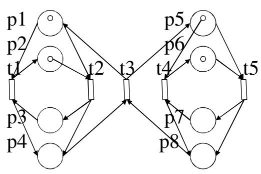


Redes de Petri

- Solução da Equação de Estado

$\boxed{M'(pi)=M0(pi) + C.\overline{S},\ \forall\ pi \in P}$

- $AL^{\Re}(N,M_0)=LRS^{\Re}=\{M \in \Re^{|P|} \mid \exists\ \overline{s} \in \Re^{|T|}$ onde $M'(pi)=M0(pi) + C.\overline{S}\}$
- $RS(N) \subseteq LRS^{\bullet}(N) \subseteq LRS^{\Re}(N)$
- Condição suficiente para não-alcaçabilidade
- Condição necessária para alcaçabilidade
- **Polinomial para redes genéricas (método de eliminação de Gauss)**



$M'=(1,0,1,0,1,1,0,0)$

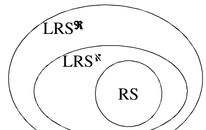
$M' \notin RS$, $M' \notin LRS^{\bullet}$, mas

$M' \in LRS^{\Re}$

$\overline{s} = (0,1,1,1/2,1/2)$

Redes de Petri

- Solução da Equação de Estado
- $RS(N) \subseteq LRS^{\bullet}(N) \subseteq LRS^{\Re}(N)$
- Condição suficiente para não-alcaçabilidade
- Condição necessária para alcaçabilidade



Redes de Petri

- Propriedades que se podem analisar com a Equação de Estados
  - Condição necessária para alcançabilidade
  - Condição suficiente para não-alcançabiliade
  - É o limite de p igual a k? Não existência de M tal que M(p) > k (condição suficiente)
  - Qual é o limite de p? (condição suficiente)
  - É um par de lugares $P'=(p_i,p_j) \subseteq P$ mutuamente exclusivo? Não existência de M tal que $M(p_i)+M(p_i)>1$
  - Deadlock freedom? Não existência de M onde nenhuma transição esteja habilitada (condição suficiente)

Redes de Petri

- É o limite de p igual a k? Não existência de M tal que M(p) > k (condição suficiente)

$M'(pi)=M0(pi) + C.\overline{S}, \ \forall \ pi \in P$

Se $\nexists$ M'(p)>k, então M'(p)=k

- Exemplo:

É o limite de $p_3 = 4$?

Se não existir $M(p_3) = 5$ como solução de

$M'(pi)=M0(pi) + C.\overline{S}, \ \forall \ pi \in P$, então o limite de p_3 é 4

Redes de Petri

- Qual é o limite de p? (condição suficiente)

O limite de p é $b[p]=\max\{M(p) \mid M \in RS(N)\}$

Seja $1_{pi}=\overset{p_0}{[0}, ..., 0, \overset{p_i}{1}, 0, ..., \overset{p_m}{0]}$

$sb[p]=\max\{1_{pi}.M \mid M - C.\overline{S} = M_0 \wedge M, \overline{S} \geq 0\}$

Como a solução de $M(pi)=M_0(pi) + C.\overline{S}, \ \forall \ pi \in P$ pode gerar soluções espúrias M pode ser uma delas, mas como $RS(N) \subseteq LRS^*(N)$, portanto $sb[p] \geq b[p]$

Redes de Petri

- É um par de lugares $P'=(p_l,p_j) \subseteq P$ mutuamente exclusivo? Não existência de M tal que $M(p_i)+M(p_i)>1$

Se $\nexists$ M tal que

$M(p_l)+M(p_j)>1$

$\forall \ M \in LRS^*=\{M \mid M(pi) = M_0(pi) + C.\overline{S}, \ \forall \ p_i \in P\}$

Então p_l,p_j são mutamente exclusivos

Redes de Petri

- É um sub-conjunto de lugares $P' \subseteq P$ mutuamente exclusivo?

Se $\nexists$ $M(pi)>1, \ \forall \ M \in LRS^*=\{M \mid M(pi)=M_0(pi) + C.\overline{S}, \ \forall \ p_i \in P$

Ou seja, sendo a rede é *safe.*

Se $\nexists$ M tal que

$\sum M(p_l)>1, \ \forall \ p_l \in P'$

$\forall \ M \in LRS^*=\{M \mid M(pi) = M_0(pi) + C.\overline{S}, \ \forall \ p_i \in P\}$

Então os lugares P' são mutamente exclusivos

Redes de Petri

- Deadlock freedom? Não existência de M onde nenhuma transição esteja habilitada (condição suficiente)

Se $\nexists$ M tal que $M=M_0 + C.\overline{S(t_j)}, \ \forall \ t_j \in T$

Não há uma marcação obtida pelo disparo de qualquer transição t_j. Ou seja, não há *deadlock*.

Invariantes de Transição

$M_0 [s> M$ e $M = M_0$

Então, de $M = M_0 + C.\overline{S} \Rightarrow M - M_0 = C.\overline{S}$, temos

$C.\overline{S} = 0 \Rightarrow C.\ I_t = 0$

- T-*flow* (um vetor): $I_t : T \rightarrow \Im$
  - T-semi*flow* (um vetor): $I_t : T \rightarrow \aleph$
- Lei de repetição de transições (uma equação)
- Componente repetitivo (uma rede)

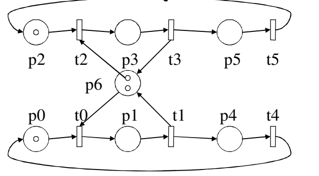
Invariantes de Transição

- T-*semiflow* (um vetor)
- Lei de repetição de transições (uma equação)
- Componente repetitivo (uma rede)

$M_0 [s> M$ e $M = M_0$

Então, de $M = M_0 + C.\overline{S} \Rightarrow M - M_0 = C.\overline{S}$, temos

$C.\overline{S} = 0 \Rightarrow C.\ I_t = 0$



I. s0 = s1 = s4
II. s2 = s3 = s5

Para s0 = 1 e s2= 0
- It1 = (1,1,0,0,1,0)

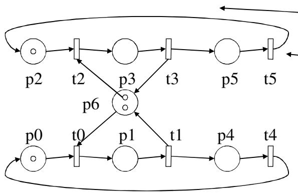
Para s0 = 0 e s2 =1
- It2=(0,0,1,1,0,1)
- It3 = It1 + It2 = (1,1,1,1,1,1)

Invariantes de Transição

$M_0 [s> M$ e $M = M_0$

Então, de $M = M_0 + C.\overline{S} \Rightarrow M - M_0 = C.\overline{S}$, temos

$C.\overline{S} = 0 \Rightarrow C.\ I_t = 0$



I. T-semiflow (vetor)
II. Lei de repetição (equação)
III. Componente repetitivo (uma rede)

- It1 = (1,1,0,0,1,0)

Invariantes de Transição

- T-*semiflow* (um vetor)
- Lei de repetição de transições (uma equação)
- Componente repetitivo (uma rede)
- T-*semiflow*: Seja um vetor It ≥ 0, tal que $s_i \geq 0$. It é chamado de t-*semiflow* sse C.It = 0
- Suporte de um t-*semiflow*: Seja It um p-*semiflow*, ST é definido como suporte de t-*semiflow* ST={t | It(t)>0}
- Minimal T-semi*flow*: Um t-*semiflow* é minimo sse seu suporte não contém nenhum suporte de qualquer outro t-*semiflow*.

Invariantes de Lugar

- N é dita conservativa se $\sum_{pi \in P} w_i \times M(p_i) = cte$
- Dado que toda marcação deve satisfazer a equação de estado, podemos fazer: $M' = M_0 + C.\overline{S}$
- Portanto, para qualquer marcação inicial M_0

  $W^T.M' = W^T.M_0 = W^T.(M_0 + C.\overline{S})$

  $W^T.M_0 = W^T.M_0 + W^T.\ C.\overline{S}$

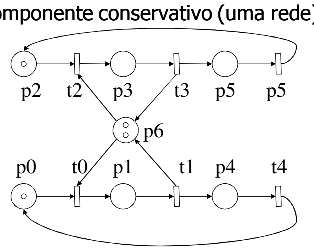
  $W^T.\ C.\overline{S} = 0$
- Como $\overline{S} \neq 0 \Rightarrow W^T.\ C = 0$
- Invariante de Lugar - Ip . C = 0

Invariantes de Lugar

- P-*semiflow* (um vetor)
- Marcação invariante ou lei de conservação de marcas (uma equação)
- Componente conservativo (uma rede)
- P-*semiflow*: Seja um vetor Ip ≥ 0, tal que $\omega_i \geq$. Ip é chamado de p-*semiflow* sse Ip . C = 0
- Suporte de um p-*semiflow*: Seja Ip um p-*semiflow*, SP é definido como suporte de p-*semiflow* SP={p | IP(p)>0}
- Minimal P-semi*flow*: Um p-*semiflow* é minimo sse seu suporte não contém nenhum suporte de qualquer outro p-*semiflow*.

Invariantes de Lugar

- P-*semiflow* (um vetor)
- Marcação invariante ou lei de conservação de marcas (uma equação)
- Componente conservativo (uma rede)

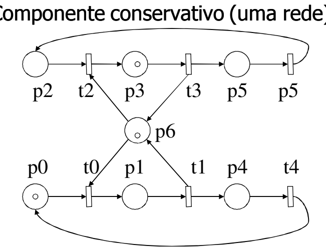


Ip . C = 0

p0,p1,p2,p3,p4,p5,p6
Ip1= |1, 1, 0, 0, 1, 0, 0|
Ip2= |0, 0, 1, 1, 0, 1, 0|
Ip3= |0, 1, 0, 1, 0, 0, 1|

Invariantes de Lugar

- P-*semiflow* (um vetor)
- Marcação invariante ou lei de conservação de marcas (uma equação)
- Componente conservativo (uma rede)

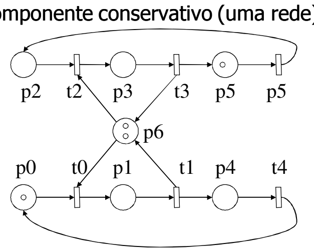


Ip . C = 0

p0,p1,p2,p3,p4,p5,p6
Ip1= |1, 1, 0, 0, 1, 0, 0|
Ip2= |0, 0, 1, 1, 0, 1, 0|
Ip3= |0, 1, 0, 1, 0, 0, 1|

Invariantes de Lugar

- P-*semiflow* (um vetor)
- Marcação invariante ou lei de conservação de marcas (uma equação)
- Componente conservativo (uma rede)

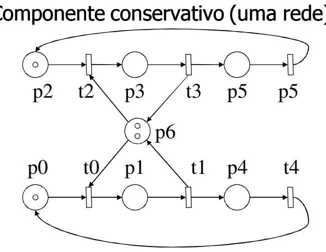


Ip . C = 0

p0,p1,p2,p3,p4,p5,p6
Ip1= |1, 1, 0, 0, 1, 0, 0|
Ip2= |0, 0, 1, 1, 0, 1, 0|
Ip3= |0, 1, 0, 1, 0, 0, 1|

Invariantes de Lugar

- P-*semiflow* (um vetor)
- Marcação invariante ou lei de conservação de marcas (uma equação)
- Componente conservativo (uma rede)

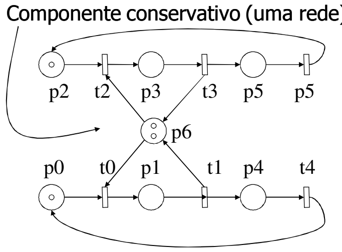


Ip . C = 0

p0,p1,p2,p3,p4,p5,p6
Ip1= |1, 1, 0, 0, 1, 0, 0|
Ip2= |0, 0, 1, 1, 0, 1, 0|
Ip3= |0, 1, 0, 1, 0, 0, 1|

Invariantes de Lugar

- P-*semiflow* (um vetor)
- Marcação invariante ou lei de conservação de marcas (uma equação)
- Componente conservativo (uma rede)



Ip . C = 0

p0,p1,p2,p3,p4,p5,p6
Ip1= |1, 1, 0, 0, 1, 0, 0|
Ip2= |0, 0, 1, 1, 0, 1, 0|
Ip3= |0, 1, 0, 1, 0, 0, 1|

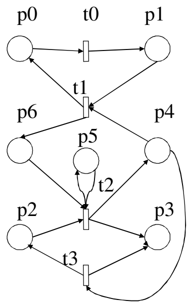
Invariantes de Lugar

Algoritmo para o cálculo dos mínimos p-*semiflows*

- Input:matriz de incidência C, onde m =|P | e n=|T |
- Ouput: conjunto de p-*semiflows*
  - A:=C , W:=I_n - I_n indentidade de dimensão n
  - for i:=1 to m, faça:
    - Some a matriz [A|W] todas as linhas que são combinação linear de pares de linha de [A|W] e anulam a i-ésima coluna de A
    - Elimine de [A|W] as linhas em que a coluna i-ésima não é nula.
  - Remova as linhas de W que os suportes não são mínimos e divida cada uma pelo MDC dos elementos não nulos. (todos os mínimos estão aqui)

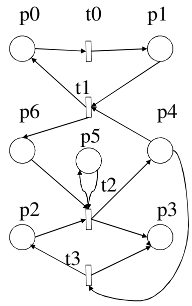
Invariantes de Lugar

0	1	2	3	0	1	2	3	4	5	6	
-1	1	0	0	1	0	0	0	0	0	0	0
1	-1	0	0	0	1	0	0	0	0	0	1
0	0	-1	1	0	0	1	0	0	0	0	2
0	0	1	-1	0	0	0	1	0	0	0	3
0	-1	1	0	0	0	0	0	1	0	0	4
0	0	0	0	0	0	0	0	0	1	0	5
0	1	-1	0	0	0	0	0	0	0	1	6



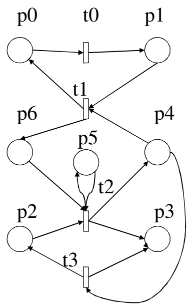
Invariantes de Lugar

0	1	2	3	0	1	2	3	4	5	6	
-1	1	0	0	1	0	0	0	0	0	0	0
1	-1	0	0	0	1	0	0	0	0	0	1
0	0	-1	1	0	0	1	0	0	0	0	2
0	0	1	-1	0	0	0	1	0	0	0	3
0	-1	1	0	0	0	0	0	1	0	0	4
0	0	0	0	0	0	0	0	0	1	0	5
0	1	-1	0	0	0	0	0	0	0	1	6



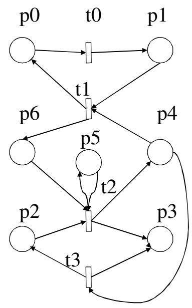
Invariantes de Lugar

0	1	2	3	0	1	2	3	4	5	6	
-1	1	0	0	1	0	0	0	0	0	0	0
1	-1	0	0	0	1	0	0	0	0	0	1
0	0	-1	1	0	0	1	0	0	0	0	2
0	0	1	-1	0	0	0	1	0	0	0	3
0	-1	1	0	0	0	0	0	1	0	0	4
0	0	0	0	0	0	0	0	0	1	0	5
0	1	-1	0	0	0	0	0	0	0	1	6
0	0	0	0	1	1	0	0	0	0	0	



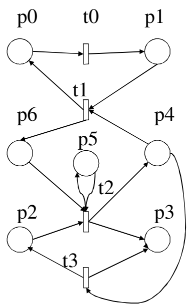
Invariantes de Lugar

0	1	2	3	0	1	2	3	4	5	6	
0	0	-1	1	0	0	1	0	0	0	0	2
0	0	1	-1	0	0	0	1	0	0	0	3
0	-1	1	0	0	0	0	0	1	0	0	4
0	0	0	0	0	0	0	0	0	1	0	5
0	1	-1	0	0	0	0	0	0	0	1	6
0	0	0	0	1	1	0	0	0	0	0	
0	0	0	0	0	0	0	0	1	0	1	



Invariantes de Lugar

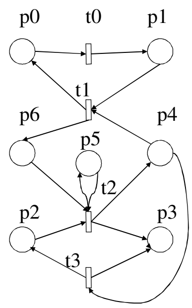
0	1	2	3	0	1	2	3	4	5	6	
0	0	-1	1	0	0	1	0	0	0	0	2
0	0	1	-1	0	0	0	1	0	0	0	3
0	0	0	0	0	0	0	0	0	1	0	5
0	0	0	0	1	1	0	0	0	0	0	
0	0	0	0	0	0	0	0	1	0	1	
0	0	0	0	0	0	1	1	0	0	0	



Invariantes de Lugar

0	1	2	3	0	1	2	3	4	5	6	
0	0	0	0	0	0	0	0	0	1	0	5
0	0	0	0	1	1	0	0	0	0	0	
0	0	0	0	0	0	0	0	1	0	1	
0	0	0	0	0	0	1	1	0	0	0	

Ip0={p5}
Ip1={p0,p1}
Ip2={p4,p6}
Ip3={p2,p3}



Propriedades Estruturais

São propriedades inerentes a estrutura da rede. Não dependem da marcação do modelo.

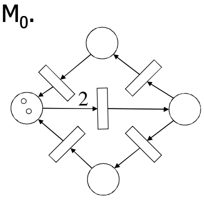
- Limitação Estrutural
- Conservação Estrutural
- Repetitividade
- Consistência

Propriedades Estruturais

- Limitação Estrutural - Seja uma rede R=(P,T,I,O) e M_0 uma marcação inicial. R é definida como estruturalmente limitada se R é limitada para qualquer M_0.
- Teorema- Uma rede R=(P,T,I,O) é estruralmente limitada sse $\exists$ W tal que $W.C \le 0$, onde $|W| = P$ e $\omega_i > 0$.

Propriedades Estruturais

- Limitação Estrutural - Seja uma rede R=(P,T,I,O) e M_0 uma marcação inicial. R é definida como estruturalmente limitada se R é limitada para qualquer M_0.



Propriedades Estruturais

- Limite Estrutural de um Lugar

Dado que para uma rede estruturalmente limitada qualquer M′ obedece $M' \le M_0 + C.S$, então

$W.M \le W.M_0$, onde $\omega_k > 0$, Portanto para um lugar p_i qualquer, temos:

$$M(p_i) \le W.M_0 / W(p_i)$$

$$M(p_i) \le \sum \omega_k . m_0(p_k) / \omega_i, \forall p_k \in P$$

Propriedades Estruturais

- Limite Estrutural de um Lugar

Seja W mínimo, temos:

$M(p_i) = \min\{ W.M_0 / W(p_i)\}, \forall W \in BIS$

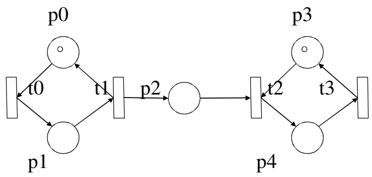
BIS – base invariant support

Propriedades Estruturais

- Conservação Estrutural - Seja uma rede R=(P,T,I,O) e M_0 uma marcação inicial. R é definida como estruturalmente conservativa se R é conservativa para qualquer M_0.
- Teorema- Uma rede R=(P,T,I,O) é estruralmente conservativa sse $\exists$ W tal que $W.C = 0$, onde $|W| = P$ e $\omega_i > 0$.

Propriedades Estruturais

- Conservação Estrutural



SP1={p0, p1}
SP2={p3, p4}

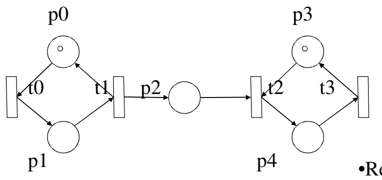
•Rede não é estruturalmente conservativa

Propriedades Estruturais

- Conservação Estrutural Parcial - Seja uma rede R=(P,T,I,O) e M_0 uma marcação inicial. R é definida como estruturalmente parcialmente conservativa se R tem algum componente conservativo para qualquer M_0.
- Teorema- Uma rede R=(P,T,I,O) é estruralmente parcialmente conservativa sse $\exists$ $W \neq 0$ tal que $W.C = 0$, onde $|W| = P$ e $\omega_i \geq 0$.

Propriedades Estruturais

- Conservação Estrutural Parcial



SP1={p0, p1}
SP2={p3, p4}

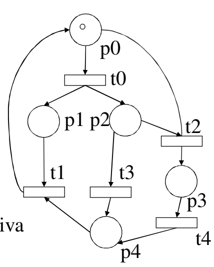
•Rede é parcialmente estruturalmente conservativa

Propriedades Estruturais

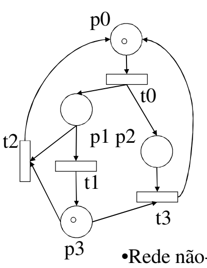
- Repetitividade - Seja uma rede R=(P,T,I,O). R é definida como repetitiva se $\exists$ M_0 tal que $M_0[s>M'$, onde $M' \geq M_0$ e $\overline{S}>0$ onde $\overline{s_i} > 0$
- Teorema- Uma rede R=(P,T,I,O) é repetitiva sse $\exists$ $\overline{S}$ tal que $C.\overline{S} \geq 0$, onde $|\overline{S}| = T$ e $\overline{s_i}>0$.

Propriedades Estruturais

- Repetitividade



•Rede repetitiva



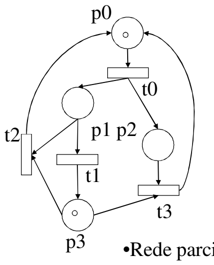
•Rede não-repetitiva

Propriedades Estruturais

- Repetitividade Parcial- Seja uma rede R=(P,T,I,O). R é definida como parcialmente repetitiva se $\exists$ M_0 tal que $M_0[s>M'$, onde $M' \geq M_0$ e $\overline{S} \neq 0$, onde $\overline{s_i} \geq 0$.
- Teorema- Uma rede R=(P,T,I,O) é repetitiva sse $\exists$ $\overline{S} \neq 0$ tal que $C.\overline{S} \geq 0$, onde $|\overline{S}| = T$ e $\overline{s_i} \geq 0$.

Propriedades Estruturais

- Repetitividade Parcial



•Rede parcialmente repetitiva

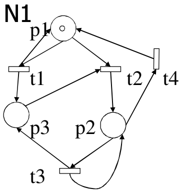
Propriedades Estruturais

- Consistência - Seja uma rede R=(P,T,I,O). R é definida como consistente se $\exists$ M_0 tal que M_0[s> M_0, onde M'= M_0 e $\overline{S}$>0 onde $\overline{s}_i > 0$.

- Teorema- Uma rede R=(P,T,I,O) é consistente sse $\exists$ $\overline{S}$ tal que C.$\overline{S}$ = 0, onde $|\overline{S}|$ = T e $\overline{s}_i$>0.

Propriedades Estruturais

- N1



•Consistente

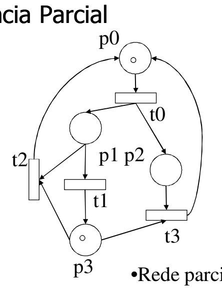
Propriedades Estruturais

- Consistência Parcial- Seja uma rede R=(P,T,I,O). R é definida como parcialmente consistente se $\exists$ M_0 tal que M_0[s> M_0, onde M'= M_0 e $\overline{S} \neq 0$, onde $\overline{s}_i \geq 0$.

- Teorema- Uma rede R=(P,T,I,O) é parcialmente consistente sse $\exists$ $\overline{S} \neq 0$ tal que C.$\overline{S}$ = 0, onde $|\overline{S}|$ = T e $\overline{s}_i \geq 0$.

Propriedades Estruturais

- Consistência Parcial



•Rede parcialmente consistente

Propriedades Estruturais

- *Siphon*- Seja uma rede R=(P,T,I,O). Um *siphon* S $\subseteq$ P é um conjunto de lugares que $^\bullet S \subseteq S^\bullet$.
- *Trap*- Seja uma rede R=(P,T,I,O). Um *trap* S $\subseteq$ P é um conjunto de lugares que $S^\bullet \subseteq {}^\bullet S$.
- Teorema: Assuma uma rede R=(P,T,I,O,M_0) com um *siphon* S. Se $\forall$ p$\in$S M_0(p) =0 $\Rightarrow \nexists$ M $\in$RS(R) tal que M(p) $\neq$0, $\forall$ p$\in$S
- Teorema: Assuma uma rede R=(P,T,I,O,M_0) com um *trap* S. Se $\exists$ p$\in$S M_0(p) $\neq$ 0 $\Rightarrow \nexists$ M $\in$RS(R) M(p)=0, $\forall$ p$\in$S

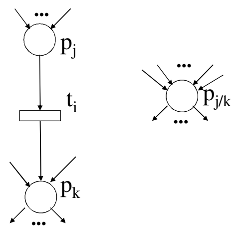
Reduções

- Análise por transformações
  - Análise de redes grandes dimensões não é um problema trivial
  - Reduções são utilizadas para análise
  - Refinamento são utlizados na síntese

Reduções

- Seja $N=(P,T,I,O,M_0)$ é uma rede e $t_i \in T$ uma transição, $I(t_i)=[p_j]$ e $O(t_i)=[p_k]$. N pode ser tranformada em $N'=(P',T',I',O',M'_0)$ pela fusão dos lugares p_j e p_k e eliminação de t_i. O lugar $p_{j/k} \in P'$ representa os lugares fundidos, onde $I(p_{j/k})= I(p_j) \cup I(p_k)$ e $O(p_{j/k})=O(p_k)$

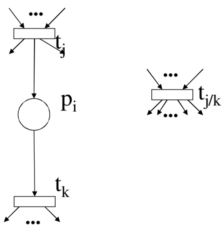
- Fusão de Lugares em Série



Reduções

- Seja $N=(P,T,I,O,M_0)$ é uma rede e $p_i \in P$ uma transição, $I(p_i)=[t_j]$ e $O(p_i) = [t_k]$. N pode ser tranformada em $N'=(P',T',I',O',M'_0)$ pela fusão das transições t_j e t_k e eliminação de p_i. A transição $t_{j/k} \in T'$ representa as transições fundidas, onde $I(t_{j/k})= I(t_j)$ e $O(t_{j/k})=O(t_j) \cup O(t_k)$

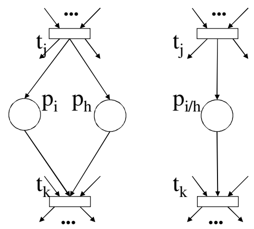
- Fusão de Transições em Série



Reduções

- Seja $N=(P,T,I,O,M_0)$ é uma rede e $p_i,p_h \in P$ lugares, $I(p_i)= I(p_h)=[t_j]$ e $O(p_i)= O(p_h)=[t_k]$. N pode ser tranformada em $N'=(P',T',I',O',M'_0)$ pela fusão dos lugares p_i e p_h. O lugar $p_{i/h} \in P'$ representa os lugares fundidos, onde $I(p_{i/h})= I(p_i)= I(p_h)=$ e $O(p_i)=O(p_h)$

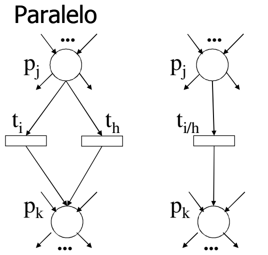
- Fusão de Lugares em Paralelo



Reduções

- Seja $N=(P,T,I,O,M_0)$ é uma rede e $t_i,t_h \in T$ transições, $I(t_i)= I(t_h)=[p_j]$ e $O(t_i)= O(t_h)=[p_k]$. N pode ser tranformada em $N'=(P',T',I',O',M'_0)$ pela fusão das transições t_i e t_h. A transição $t_{i/h} \in T'$ representa as transições fundidas, onde $I(t_{i/h})= I(t_i)= I(t_h)$ e $O(t_{i/h})=O(t_i) = O(t_h)$

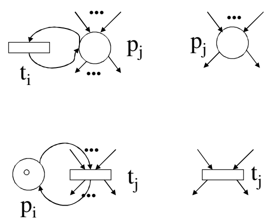
- Fusão de Transições em Paralelo



Reduções

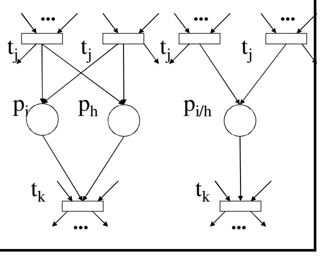
- Seja $N=(P,T,I,O,M_0)$ é uma rede e $e_i \in P \cup T$ um elemento. Se $I(e_i)=O(e_i)$, e se $e_i \in P$, $M(e_i) \geq \#O(e_i)$, então N pode ser transformada em $N'=(P',T',I',O',M'_0)$ pela eliminação de e_i.

- Eliminação de Auto-laços



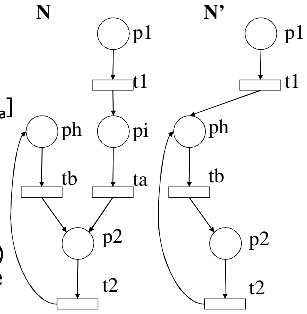
Reduções
Algumas Outras Reduções Implementadas em INA

- Fusão de Nós Congruentes
- Seja $N=(P,T,I,O,M_0)$ é uma rede e $e_i,e_h \in P \cup T$ elementos, $I(e_i)= I(e_h)$ e $O(e_i)= O(e_h)$. N pode ser transformada em $N'=(P',T',I',O',M'_0)$ pela fusão dos elementos e_i e e_h. O elemento $e_{i/h} \in P' \cup T'$ representa os elementos fundidos, onde $I(e_{i/h})= I(e_i)= I(e_h)=$ e O $e_{i/h})= O(e_i)=O(e_h)$
- Generalização das reduções paralelas apresentadas



Reduções
Algumas Outras Reduções Implementadas em INA

- *Merging* de lugares equivalentes Seja $N=(P,T,I,O,M_0)$ é uma rede e $p_i,p_h \in P$ lugares. Se $O(p_i)=[t_a]$ $O(p_h)=[t_b]$, $I[p_i, t_a] = I[p_h, t_b] =1$ e $O(t_a)= O(t_b)=[\ p_x \in P| p_x \neq p_i,p_h]$, então N pode ser transformada em $N'=(P',T',I',O',M'_0)$ onde $P'=P/\{p_i\}$, $T'=T/\{t_a \mid t_a \in O(p_i)$ onde p_i é o lugar eliminado$\}$ e $I'(p_h)=I(p_h)\cup I(p_i)$



Redes de Petri

- Lema: Uma rede $N=(P,T,I,O)$ é livre de *deadlock* sse nenhum vértice do RG(N) não tem vértice sem arcos de saída.
- Lema: Uma rede $N=(P,T,I,O)$ é reversível sse RG(N) é fortemente conectado.
- Teorema: O grafo de cobertura CG(N) e o grafo de alcançabilidade RG(N) de uma rede $N=(P,T,I,O)$ limitada são idênticos.

Redes de Petri

- Teorema da Conectividade Forte: Toda rede $N=(P,T,I,O)$ *fracamente conectada viva* e *limitada* é *fortemente conectada*.
- Teorema: Toda rede $N=(P,T,I,O)$ conectada com pelo menos um invariante de lugar e um invariante de transição que mapeiem todos os lugares e transições, respectivamente, em números positivos é fortemente conectada.

Redes de Petri

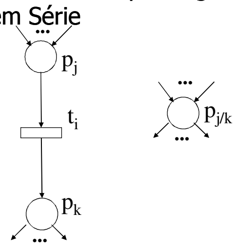
- Proposição: Seja $N=(P,T,I,O,M_0)$ uma rede viva, então $N=(P,T,I,O)$ é estruturalmente repetitiva. Caso $N=(P,T,I,O,M_0)$ seja também limitada, então $N=(P,T,I,O)$ é consistente.

Redes de Petri

- Corolário: Uma rede $N=(P,T,I,O,M_0)$ marcada finita é limitada se existe um invariante de lugar que mapeie todos os lugares da rede em números positivos.
- Teorema: O conjunto de invariantes mínimos de uma rede N é finito.

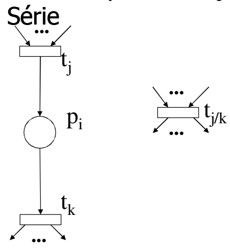
Refinamento

- Seja $N'=(P',T',I',O',M'_0)$ e $p_{j/k} \in P'$ um lugar. $N=(P,T,I,O,M_0)$ é uma rede obtida pelo refinamento de $p_{j/k} \in P'$ pelos lugares p_j e p_k e a transição t_i, onde $I(p_{j/k})= I(p_j)$, $O(p_{j/k})=O(p_k)$, $O(p_j)= I(p_k)= [t_i]$.
- Refinamento por Lugares em Série



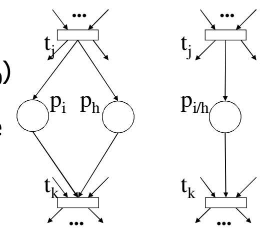
Refinamento

- Seja $N'=(P',T',I',O',M'_0)$ e $t_{j/k} \in T'$ uma transição. $N=(P,T,I,O,M_0)$ é uma rede obtida pelo refinamento de $t_{j/k} \in T'$ pelas transições t_j e t_k e o lugar p_i, onde $I(t_{j/k})= I(t_j)$, $O(t_{j/k})=O(t_k)$, $O(t_j)= I(t_k)= p_i$.
- Refinamento por Transições em Série



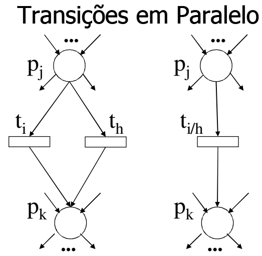
Refinamento

- Seja $N'=(P',T',I',O',M'_0)$. O lugar $p_{i/h} \in P'$ pode ser refinado por p_i, p_h, onde $I(p_i)= I(p_h)=[t_j]$ e $O(p_i)= O(p_h)=[t_k]$. $N=(P,T,I,O,M_0)$ é a rede obtida pelo refinamento de $p_{i/h}$, onde $p_i, p_h \in P'$.
- Refinamento por Lugares Paralelo



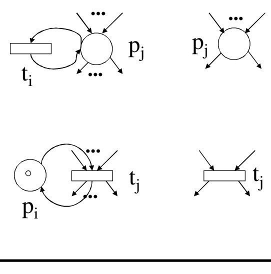
Refinamento

- Seja $N'=(P',T',I',O',M'_0)$. O lugar $t_{i/h} \in T'$ pode ser refinado por t_i, t_h, onde $I(t_i)= I(t_h)=[p_j]$ e $O(t_i)= O(t_h)=[p_k]$. $N=(P,T,I,O,M_0)$ é a rede obtida pelo refinamento de $t_{i/h}$, onde $t_i, t_h \in T'$.
- Refinamento por Transições em Paralelo



Refinamento

- Seja $N'= (P',T',I',O',M'_0)$ e $N=(P,T,I,O,M_0)$ uma rede obtida pelo refinamento de N'. Seja $e_i \in P \cup T$ um elemento obtido pelo refinamento de $e_j \in P' \cup T'$, $I(e_i)=O(e_i)$. Se $e_i \in P$ então $M(e_i) \geq \#O(e_i)$.
- Refinamento por Auto-laços



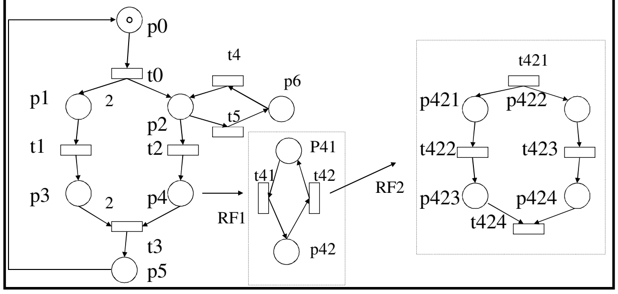
Abordagem de Modelagem

- Princípios -

- Top-down
  - Refinamento
- Botton-up
  - Composição
- Híbrida
  - Refinamento e Composição

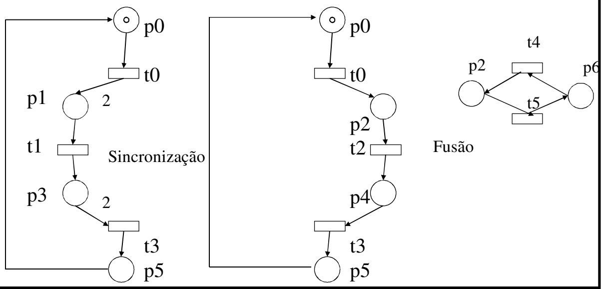
Abordagem de Modelagem
- Princípios -

- Top-down



Abordagem de Modelagem
- Princípios -

- Botton-up



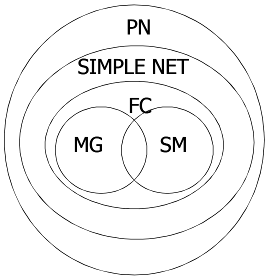
Propriedades Qualitativas

- Comportamentais
  - Reachability
  - Coverability
  - Deadlock freedom
  - Liveness
  - Reversibility
  - Boundedness
  - Safeness
  - Persistence
  - Fairness
- Estruturais
  - Repetitiveness
  - Consistence
  - Structural Boundedness
  - Conservation

Métodos de Análise

- Árvore de Alcançabilidade
- Análise dos Invariantes
- Reduções

Sub-Classes



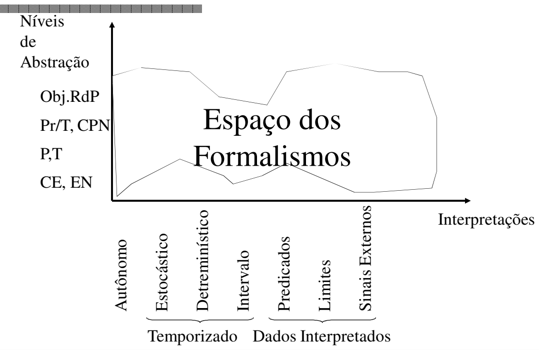
Redes de Alto Nível

- Pr/Tr net
- CPN

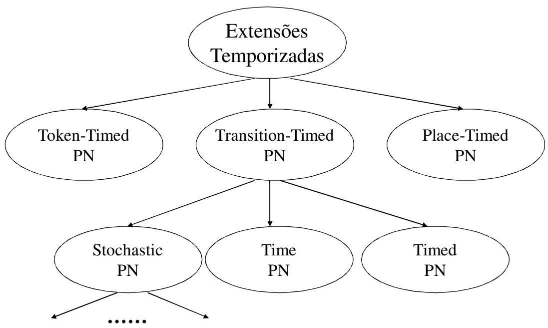
Análise de Desempenho

- Analíticos
  - Determinísticos
    - Melhor e pior casos
  - Probabilisticos
    - Valores prováveis
- Simulação
  - Análise exaustiva
- Implementação real
  - Medidas obtidas do sistema real
  - *Benchmarks*
  - Protótipos

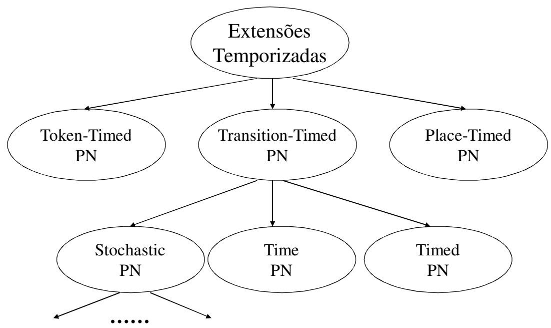
Redes de Petri



Redes Temporizadas



Redes Temporizadas



Redes Temporizadas

- Ramchandani, 1973 - Transition Timed Net
- Merlin, 1976 - Transition Time Net
- Sifakis, 1977 - Place Timed Net

Redes Temporizadas Estocásticas

- Natkin -1980
- Molloy - 1981
- Marsan et al. - 1984

É uma rede temporizada onde o *delay* associado à transição é uma variável aleatória de distribuição exponencial

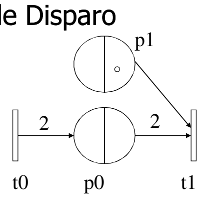
Redes Temporizadas

- Redes de Petri com Lugares Temporizados (PTPN) (Sifakis77)
  - Definição: PTPN=(P,T,F,K,W,M_0,Γ,υ), onde

P é o conjunto de lugares,
T o conjunto de transições,
$F \subseteq (P \times T) \cup (T \times P)$ uma relação que representa os arcos
W – Valoração (peso dos arcos) - W: $F \rightarrow N$
M_0- Marcação inicial - M_0:$P \rightarrow N$
$\Gamma=\{\gamma_1, \gamma_2,\ldots, \gamma_i,\ldots\}$ números reais denominada base de tempo.
υ:$P \rightarrow \Gamma$ um mapeamato que $\upsilon(p) = \gamma_j$

Redes Temporizadas - PTPN -

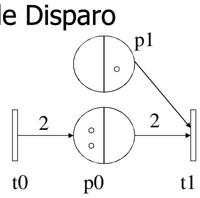
- Regra de Disparo



υ(p0) = 3

Redes Temporizadas - PTPN -

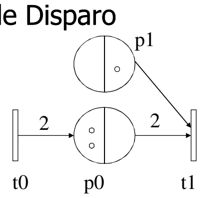
- Regra de Disparo



υ(p0) = 3 Instante=0

Redes Temporizadas - PTPN -

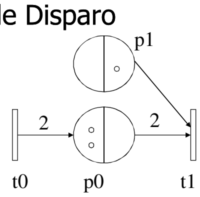
- Regra de Disparo



υ(p0) = 3 Instante=1

Redes Temporizadas - PTPN -

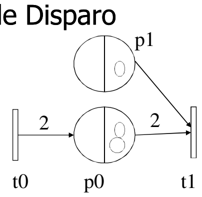
- Regra de Disparo



υ(p0) = 3 Instante=2

Redes Temporizadas - PTPN -

- Regra de Disparo



υ(p0) = 3 Instante=3

Redes Temporizadas Estocásticas

- Modelagem para Análise de Desempenho
  - Modelos para Simulação
  - Modelos Analíticos
    - Cadeias de Markov
    - Teoria das Filas
    - Redes de Petri

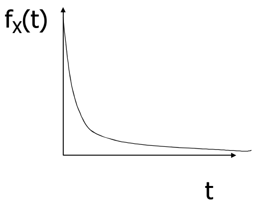
Redes Temporizadas Estocásticas

- Proriedade Markoviana
  - Ausência de Memória
- Variáveis Aleatórias com Propriedade Markoviana
  - Variável Aleatória Geométrica
  - Variável Aleatória Exponencial

Redes Estocásticas

Variável Aleatória Exponencial

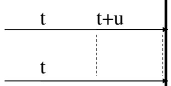
- fdp exponencial



FD - Função de Distribuição

- $f_X(t) = \mu e^{-\mu t}$
- $FD(t) = 1 - e^{-\mu t}$
- Valor Esperado
  $E(X) = \frac{1}{\mu}$
- Propriedade:
  Não possui memória

Redes Estocásticas



Variável Aleatória Exponencial

$\boxed{P\{X>t\} = e^{-\mu t}}$

- $P\{X>t+u \mid X > t\} = \dfrac{P\{X>t+u \wedge X>t\}}{P\{X>t\}}$ Probabilidade Condicional
- $P\{X>t+u \mid X > t\} = \dfrac{P\{X>t+u\}}{P\{X>t\}}$

$P\{X>t+u \mid X > t\} = \dfrac{e^{-\mu(t+u)}}{e^{-\mu t}} = \boxed{e^{-\mu u} = P\{X>u\}}$

Redes Estocásticas

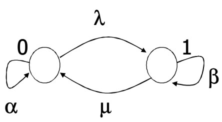
- Processo Estocástico é definido por um conjunto de variáveis aleatórias,

$\{X(t) : t \in T\}$, onde $X(t)$ é uma variável aleatória para cada $t \in T$.
t é denominado parametro e cada valor de $X(t)$ são estados.

- Tipos de Processos Estocásticos
  - Processos de espaço de estados e tempo discretos (DTMC)
  - Processos de espaço de estados contínuo e tempo discreto
  - Processos de espaço de estados discreto e tempo contínuo (CTMC)
  - Processos de espaço de estados e tempo contínuos

Redes Estocásticas

- O comportamento de uma rede estocástica é representado por CTMC



Matriz de Taxas

$$Q = \begin{array}{cc} 0 & 1 \\ \left|\begin{array}{cc} -(\alpha+\lambda) & \lambda \\ \mu & -(\beta+\mu) \end{array}\right| & \begin{array}{c} 0 \\ 1 \end{array} \end{array}$$

Matriz de Propabilidades de Próximo Estados

$$P = \left|\begin{array}{cc} \dfrac{\alpha}{\alpha+\lambda} & \dfrac{\lambda}{\alpha+\lambda} \\ \dfrac{\mu}{\beta+\mu} & \dfrac{\beta}{\beta+\mu} \end{array}\right| \begin{array}{c} 0 \\ 1 \end{array}$$

Redes Estocásticas

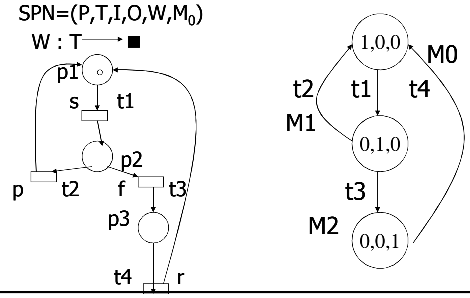
- Matriz Estocástica

$$P = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \qquad \begin{array}{l} U_1=(a_{11}\ a_{12}\ a_{13}) \\ U_2=(a_{21}\ a_{22}\ a_{23}) \\ U_3=(a_{31}\ a_{32}\ a_{33}) \end{array}$$

a_{ij} - probabilidade $\qquad \sum_{j}^{s} a_{ij}=1$

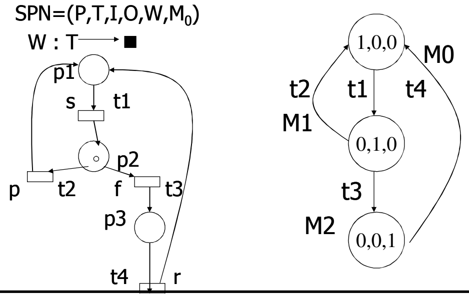
Redes Estocásticas

- Definição:
  SPN=(P,T,I,O,W,M_0)
  W : T ⟶ ■
- Grafo de Marcações



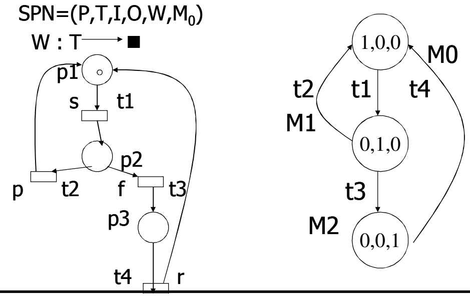
Redes Estocásticas

- Definição:
  SPN=(P,T,I,O,W,M_0)
  W : T ⟶ ■
- Grafo de Marcações



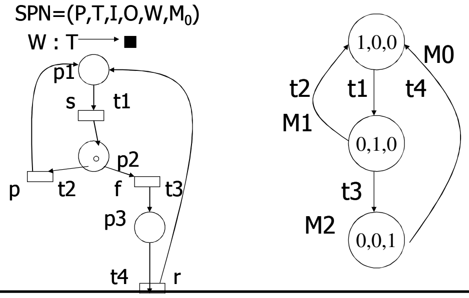
Redes Estocásticas

- Definição:
  SPN=(P,T,I,O,W,M_0)
  W : T ⟶ ■
- Grafo de Marcações



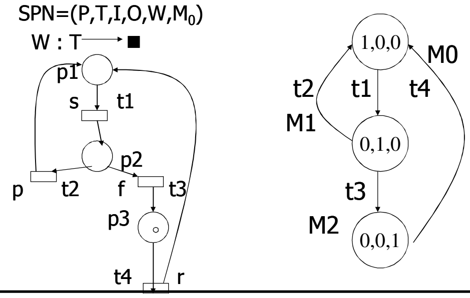
Redes Estocásticas

- Definição:
  SPN=(P,T,I,O,W,M_0)
  W : T ⟶ ■
- Grafo de Marcações



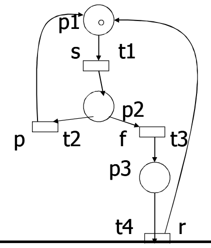
Redes Estocásticas

- Definição:
  SPN=(P,T,I,O,W,M_0)
  W : T ⟶ ■
- Grafo de Marcações

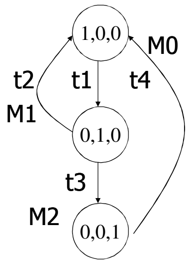


Redes Estocásticas

- Definição:
  SPN=(P,T,I,O,W,M_0)
  W : T ⟶ ■



- Grafo de Marcações



Redes Estocásticas

- Lema: seja N=(P,T,I,O,W,M_0) uma SPN. Dadas M_i,M_j $\in$ A(N), existe uma probabilidade de se atingir M_j imediatamente de M_i .
- Probabilidade de se atingir M_j de M_i

$p_{ij}= \lambda_{ij}/\lambda_{i}$, $\lambda_{ij}= \Sigma_{\tau\in Tij}\ \lambda(t,W)$, $\lambda_i= \Sigma_{\tau\in Ti}\ \lambda(t,W)$

$\lambda(t,W)$ é a taxa associada a transição através da W

$T_{ij} = \{t \in T : M_i[t > M_j\}$, $T_i = \{t \in T : M_i[t >\}$

Redes Estocásticas

- Tempo de permanência numa marcação *(mean sojourn time)*

$$m_i = \min_{tj\in Ti}(1/\lambda_j)$$

$T_i = \{t_j \in T : M_i[t_j >\}$

Redes Estocásticas

- Tempo de permanência numa marcação *(mean sojourn time)*

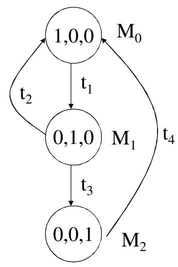
$$m_i = \min_{tj\in Ti}(1/\lambda_j)$$

$\lambda_j= \Sigma_{tj\in Ti}\ \lambda(t_j,W)$, $T_i = \{t_j \in T : M_i[t_j >\}$

Redes Estocásticas

$T_0 = \{t_1\}$, $T_{01} = \{t_1\}$
$T_1= \{t_2,t_3\}$, $T_{10}=\{t_2\}$
$T_{12}=\{t_3\}$
$T_2=\{t_4\}$, $T_{20}=\{t_4\}$

$$P = \begin{array}{c} \begin{array}{ccc} M0 & M1 & M2 \end{array} \\ \left|\begin{array}{ccc} 0 & 1 & 0 \\ \frac{p}{p+f} & 0 & \frac{f}{p+f} \\ 1 & 0 & 0 \end{array}\right| \end{array} \begin{array}{l} \\ M0 \\ M1 \\ M2 \end{array}$$



Redes Estocásticas

- Para garantir a existência de uma distribuição de probabilidade estacionária, a rede deve ser:
  - limitada *(bounded)*
  - reversível e
  - livre de bloqueio *(deadlock-free)*

Y . P = Y

$$\sum_{i=1}^{s} y_i=1 \qquad Y : S \longrightarrow ■^+$$

y_i é o numero relativo de visitas à marcação M_i

Redes Estocásticas

- Probabilidade estacionária de uma marcação M_i

$$p_i = y_i \,.\, m_i \,/\, \sum_{j=1}^{s} y_j \,.\, m_j$$

- Probabilidade que um lugar p_j tenha k marcas

p(pj,k) = ❶$_{i \in S1}$ p_i

$S_1 = \{ i \in \{1,2,..,S\} : M_i(p_j)=k\}$

- Número esperado de marcas no lugar p_j

Em(p_j) = ❶ k .p(pj,k) (k=1 … K)

K é o número máximo de marcas que o lugar pj pode conter

Redes Estocásticas

- Throughput rate de uma transição

TR(t_j) = ❶ p_i . $\lambda(t_j,W)$. q_{ij} ($i \in S_2$)

$S_2 = \{ i \in \{1,2,..,S\} : M_i[t_j>\}$

- p_i é a probabilidade estacionária de uma marcação Mi que habilita t_j
- $\lambda(t_j,W)$ é a taxa associada a transição
- qij= 1, se #O(I(t_j))=1; x, caso contrário

Redes Estocásticas

- Conclusões
  - Redes de Petri estocásticas são uma representação compacta de alto nível das CTMC
  - Isomorfismo com CTMC
  - Análise quantitativa
  - Análise qualitativa
  - Modelagem de sistemas concorrentes, não-determinísticos e assíncronos. Modelagem de sincronismo, escolha, mútua exclusão etc

Redes Estocásticas

- Extensões às SPN
  - GSPN (Marsan et al.)
  - DSPN (Lindermann, Ciardo)

Redes Estocásticas

- Bibliografia

- Modelling with Generalized Stochastic Petri Nets, A. Marsan et al, John Wiley & Sons, 1995.
- Performance Modelling with Deterministic and Stochastic Petri Nets, C. Lindermann, John Wiley & Sons, 1998.

- http://www.daimi.au.dk/PetriNets