# Redes de Petri

Manuel Alcino Cunha

Departamento de Informática
Universidade do Minho

2005/06

# Redes de Petri

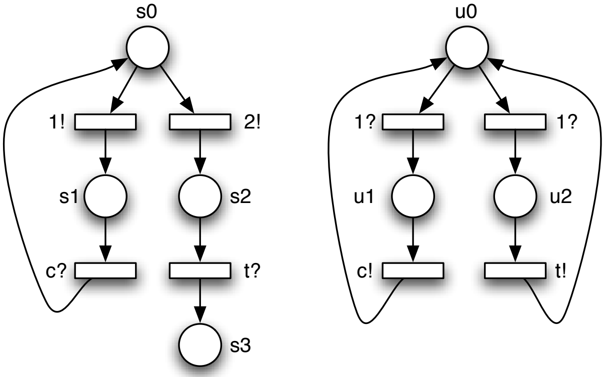
- Quando modelamos um sistema concorrente com um sistema de transição é sempre necessário pensar nos efeitos globais de uma acção.
- As redes de Petri preservam a noção de estado local e as causas e efeitos de uma acção são determinados localmente.
- Ao contrário dos sistemas de transição, onde a ênfase está nos estados, e dos métodos algébricos, onde a ênfase está nas acções, as redes de Petri dão igual ênfase aos dois componentes.

# Notação Gráfica

- Um círculo, denominado *lugar*, denota algo passivo:
  - um estado local de um componente;
  - uma condição;
  - um recurso partilhado;
  - um canal de comunicação; . . .
- Um rectângulo, denominado *acção*, denota algo activo:
  - Um evento, transição, acção;
  - A execução de uma instrução;
  - A transmissão de uma mensagem; . . .
- Os arcos determinam as causas e os efeitos:
  - Um arco entre um lugar e uma acção indica que essa condição é necessária para a executar (pré-condição);
  - Um arco entre uma acção e um lugar indica que a sua execução irá tornar essa condição verdadeira (pós-condição).

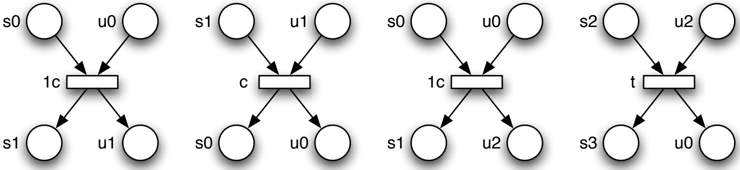
# Modelação

- No caso de sistemas não concorrentes a modelação com redes de Petri não traz grandes vantagens em relação aos sistemas de transição.

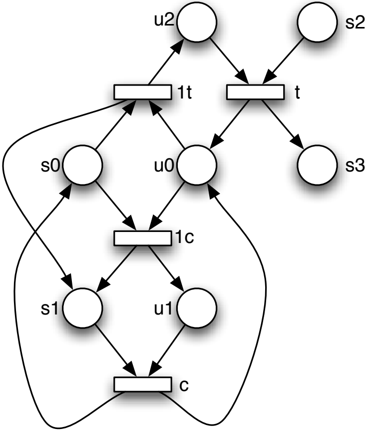


# Modelação

- Devido ao princípio da localidade é possível modelar cada uma das acções de um sistema concorrente independentemente.
- Para obter o modelo do sistema concorrente basta juntar todas as acções, fundindo lugares com o mesmo nome.
- Note que não podem existir duas acções com o mesmo nome, mas com vizinhanças diferentes.



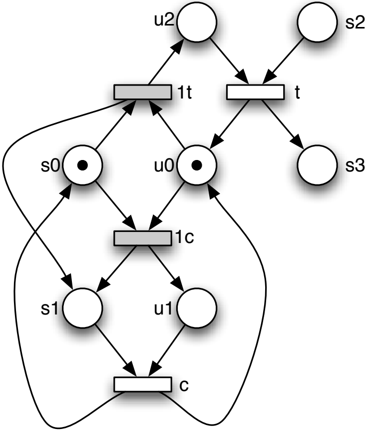
# Exemplo



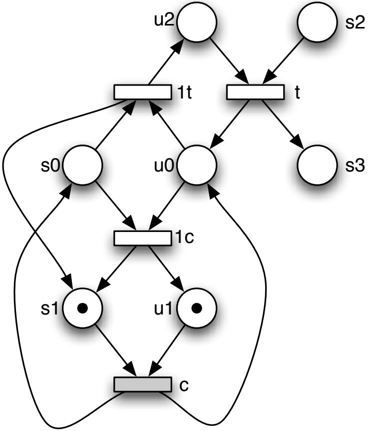
# Dinâmica

- O facto de um componente se encontrar num determinado estado é assinalado por uma marca no lugar respectivo.
- Vendo os lugares como condições, a presença de uma marca num lugar significa que essa condição se verifica.
- O estado global do sistema é a marcação actual da rede, ou seja o conjunto de estados que possuem marcas. O estado inicial é uma marcação.
- Uma acção está activa, ou pronta a executar, se todas as suas pré-condições estão marcadas.
- Quando ocorre as marcas são retiradas das pré-condições e são colocadas marcas em todas as pós-condições. O disparo é atómico.
- Que acontece se as pós-condições já tiverem marcas?

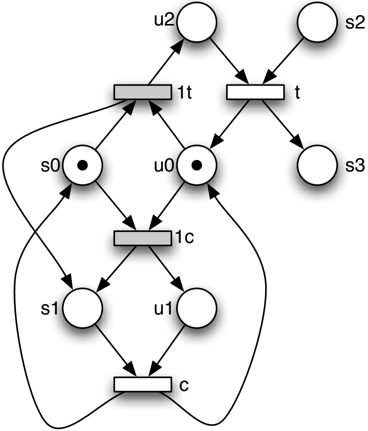
# Jogo das Marcas



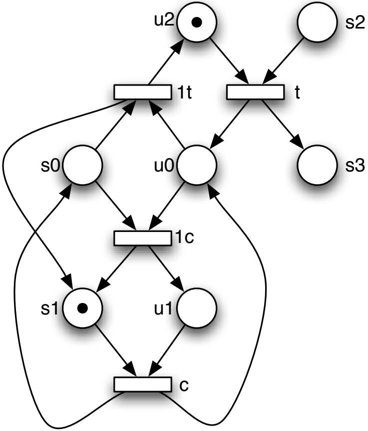
# Jogo das Marcas



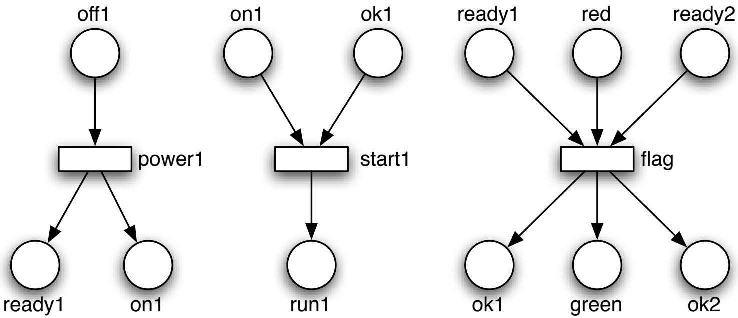
# Jogo das Marcas



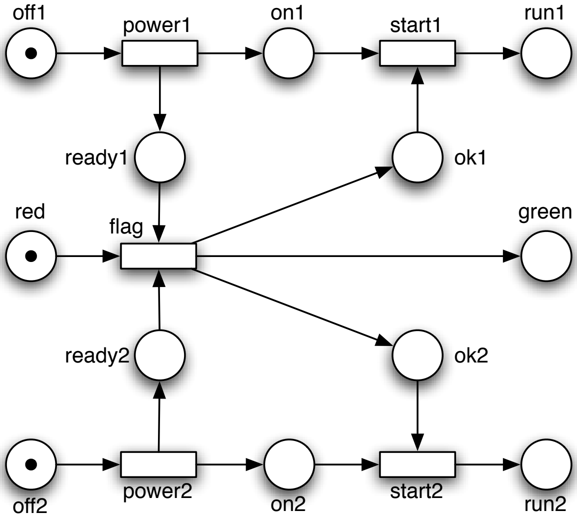
# Jogo das Marcas



# Exemplo


off1
power1
ready1
on1
on1
ok1
start1
run1
ready1
red
ready2
flag
ok1
green
ok2

# Exemplo


off1
power1
on1
start1
run1
ready1
ok1
red
flag
green
ready2
ok2
off2
power2
on2
start2
run2

# Tipos de Redes

**Elementares** Cada lugar contém no máximo uma marca. Uma acção só está activa se as pós-condições não estiverem marcadas.

**P/T** A capacidade dos lugares é ilimitada: é conveniente vê-los como recursos. Os arcos tem um peso: podem transportar mais do que uma marca. Podem ser extendidas com lugares de capacidades limitada e arcos inibidores.

**Coloridas** As marcas passam a ter informação associada. Os arcos podem indicar que tipo de marcas pretendem.

# Definição Formal

## Definição

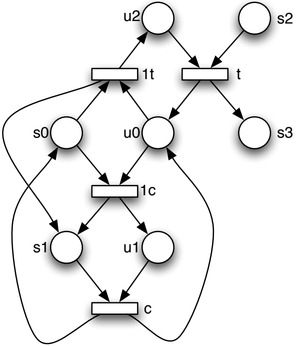
Uma *rede* é um tuplo

$$(L, A, F)$$

onde

- $L$ é um conjunto de lugares;
- $A$ é um conjunto de acções, tal que $A \cap L = \emptyset$;
- $F \subseteq (L \times A) \cup (A \times L)$ é uma relação de fluxo que representa os arcos.
- Se $L$ e $A$ são finitos então diz-se que a rede é finita.

# Exemplo



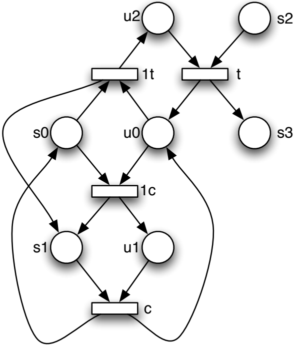

$$(L, A, F)$$

$$\begin{aligned}
L &= \{s_0, s_1, s_2, s_3, u_0, u_1, u_2\} \\
A &= \{1_c, 1_t, c, t\} \\
F &= \{(s_0, 1_c), (u_0, 1_c), (1_c, s_1), (1_c, u_1) \\
&\quad (s_1, c), (u_1, c), (c, s_0), (c, u_0) \\
&\quad (s_0, 1_t), (u_0, 1_t), (1_t, s_1), (1_t, u_2) \\
&\quad (s_2, t), (u_2, t), (t, s_3), (t, u_0)\}
\end{aligned}$$

# Pré e Pós-condições

- Dada uma rede $(L, A, F)$ e um elemento $e \in L \cup A$:
  - O conjunto dos seus elementos de entrada define-se como
$$^\bullet e \equiv \{x \in L \cup A \mid (x, e) \in F\}$$
  - O conjunto dos seus elementos de saída define-se como
$$e^\bullet \equiv \{x \in L \cup A \mid (e, x) \in F\}$$
- Se $a$ é uma acção, $^\bullet a$ denota as suas pré-condições e $a^\bullet$ as suas pós-condições.
- É conveniente extender estas noções a conjuntos. Dado $E \subseteq L \cup A$ temos
$$^\bullet E \equiv \bigcup_{e \in E} {}^\bullet e \equiv \{x \in L \cup A \mid \exists e \in E \cdot (x, e) \in F\}$$
$$E^\bullet \equiv \bigcup_{e \in E} e^\bullet \equiv \{x \in L \cup A \mid \exists e \in E \cdot (e, x) \in F\}$$

# Exemplo




$$
\begin{aligned}
{}^\bullet c &= \{s_1, u_1\} \\
c^\bullet &= \{s_0, u_0\} \\
{}^\bullet s_0 &= \{c\} \\
s_0^\bullet &= \{1_c, 1_t\} \\
{}^\bullet\{1_c, 1_1\} &= \{s_0, u_0\} \\
\{1_c, 1_1\}^\bullet &= \{s_1, u_1, u_2\}
\end{aligned}
$$

# Isomorfismos

- Duas redes $N_0 = (L_0, A_0, F_0)$ e $N_1 = (L_1, A_1, F_1)$ são isomórficas ($N_0 \cong N_1$) sse existirem duas funções bijectivas $\alpha : L_0 \rightarrow L_1$ e $\beta : A_0 \rightarrow A_1$ tal que

$$\forall l \in L_0, a \in A_0 \cdot (l, a) \in F_0 \;\Leftrightarrow\; (\alpha(l), \beta(a)) \in F_1$$

$$\forall l \in L_0, a \in A_0 \cdot (a, l) \in F_0 \;\Leftrightarrow\; (\beta(a), \alpha(l)) \in F_1$$

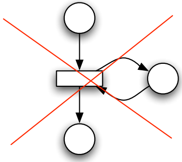
- Extendendo a aplicação de funções para conjuntos, podemos alternativamente ter

$$\forall a \in A_0 \cdot \alpha({}^\bullet a) = {}^\bullet\beta(a) \;\wedge\; \alpha(a^\bullet) = \beta(a)^\bullet$$

# Redes Puras e Simples

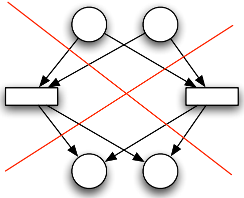
- Uma rede $(L, A, F)$ diz-e *pura* sse

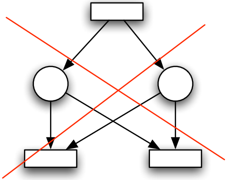
$$\forall x \in L \cup A \cdot {}^{\bullet}x \cap x^{\bullet} = \emptyset$$



- Uma rede $(L, A, F)$ diz-e *simples* sse

$$\forall x, y \in L \cup A \cdot ({}^{\bullet}x = {}^{\bullet}y \ \wedge \ x^{\bullet} = y^{\bullet}) \ \Rightarrow \ x = y$$





# Independência

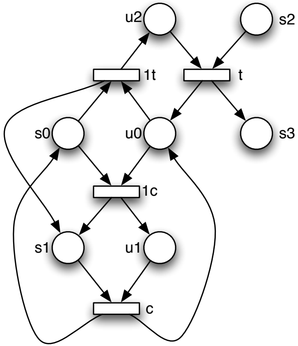
- Dada uma rede $(L, A, F)$, a *vizinhança* de uma acção $a \in A$ define-se como

$$loc(a) = {}^\bullet a \cup a^\bullet$$

- Duas acções $a_0 \neq a_1 \in A$ são *independentes* $(a_0 \# a_1)$ sse

$$loc(a_0) \cap loc(a_1) = \emptyset$$

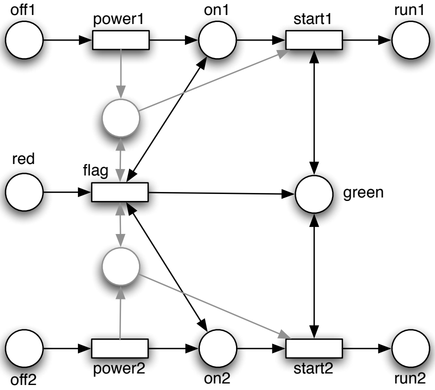
# Exemplo




- É simples e pura;
- Não tem transições independentes.

$$loc(1_t) = \{s_0, u_0, s_1, u_2\}$$
$$loc(1_c) = \{s_0, u_0, s_1, u_1\}$$
$$loc(c) = \{s_1, u_1, s_0, u_0\}$$
$$loc(t) = \{u_2, s_2, u_0, s_3\}$$

# Exemplo




- Não é simples;
- Não é pura.

$power_1 \# power_2$

$power_1 \# start_2$

$power_2 \# start_1$

$\neg(start_1 \# start_2)$

# Redes Elementares

## Definição

Uma *rede elementar* é um tuplo

$$(L, A, F, M_0)$$

onde

- $(L, A, F)$ é uma rede;
- $M_0 \subseteq L$ é a marcação inicial.

- Enquanto que $(L, A, F)$ nos dá uma caracterização estática do sistema, $M_0$ caracteriza o seu comportamento dinâmico.

# Activação e Disparo

- Dada uma rede elementar pura $(L, A, F, M_0)$ uma acção $a \in A$ está *activa* numa marcação $M \subseteq L$ ($M[a\rangle$) sse

$$^\bullet a \subseteq M \ \wedge \ a^\bullet \cap M = \emptyset$$

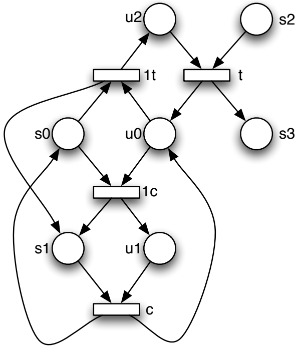
- Se a rede não for pura então $M[a\rangle$ sse

$$^\bullet a \subseteq M \ \wedge \ a^\bullet \cap (M - {}^\bullet a) = \emptyset$$

- Quando uma acção ocorre as pré-condições deixam de estar válidas e as pós-condições passam a ser válidas. Se $M[a\rangle$ então o *disparo* de $a$ leva a uma nova marcação $N$ ($M[a\rangle N$) sse

$$N = (M - {}^\bullet a) \cup a^\bullet$$

# Exemplo




$$\{s_0, u_0\}[1_t\rangle\{s_1, u_2\}$$

$$\neg\{s_0, s_2\}[1_t\rangle$$

$$\neg\{s_0, u_0, s_1\}[1_t\rangle$$

$$\{s_0, u_0, u_1\}[1_t\rangle\{s_1, u_2, u_1\}$$

# Concorrência

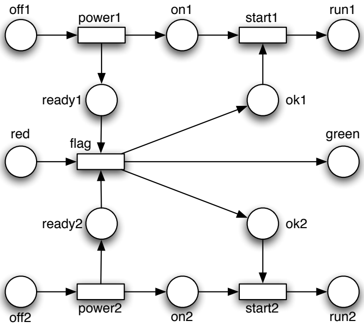
- É conveniente extender os conceitos de activação e disparo a conjuntos de acções.
- Dada uma rede elementar $(L, A, F, M_0)$ um conjunto de acções $B \subseteq A$ está *activo concorrentemente* numa marcação $M \subseteq L$ sse cada uma das acções está activa individualmente e é independente de todas as outras.
- Formalmente temos que $M[B\rangle$ sse

$$\forall a \in B \cdot M[a\rangle \ \wedge \ \forall a, b \in B \cdot a\#b$$

- O efeito do disparo de um conjunto de acções é a "soma" dos efeitos individuais de cada uma delas. Formalmente temos que $M[B\rangle N$ sse

$$N = (M - {}^{\bullet}B) \cup B^{\bullet}$$

# Exemplo



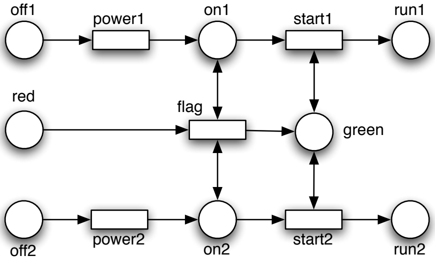

$$\{\mathrm{off}_1, \mathrm{red}, \mathrm{off}_2\}[\![\{\mathrm{power}_1, \mathrm{power}_2\}\rangle\{\mathrm{on}_1, \mathrm{ready}_1, \mathrm{red}, \mathrm{on}_2, \mathrm{ready}_2\}$$

$$\neg\,\{\mathrm{off}_1, \mathrm{red}, \mathrm{off}_2\}[\![\{\mathrm{power}_1, \mathrm{flag}\}\rangle$$

$$\{\mathrm{on}_1, \mathrm{ok}_1, \mathrm{green}, \mathrm{on}_2, \mathrm{ok}_2\}[\![\{\mathrm{start}_1, \mathrm{start}_2\}\rangle\{\mathrm{run}_1, \mathrm{green}, \mathrm{run}_2\}$$

$$\neg\,\{\mathrm{on}_1, \mathrm{ready}_1, \mathrm{ok}_1, \mathrm{red}, \mathrm{ready}_2\}[\![\{\mathrm{start}_1, \mathrm{flag}\}\rangle$$

# Exemplo




$$\{on_1, red, on_2\}[flag\rangle\{on_1, green, on_2\}$$

$$\{on_1, green, on_2\}[start_1\rangle\{run_1, green, on_2\}$$

$$\{on_1, green, on_2\}[start_2\rangle\{on_1, green, run_2\}$$

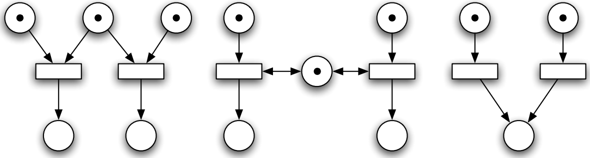
$$\neg\,\{on_1, green, on_2\}[\{start_1, start_2\}\rangle$$

# Conflito

- Dada uma rede elementar $(L, A, F, M_0)$ duas acções $a, b \in A$ estão em *conflito* numa marcação $M \subseteq L$ sse

$$M[a\rangle \wedge M[b\rangle \wedge \neg M[\{a, b\}\rangle$$

- A única razão para tal é $\neg\, a \# b$: as acções partilham pré-condições (*forwards conflict*) ou partilham pós-condições (*backwards conflict*).



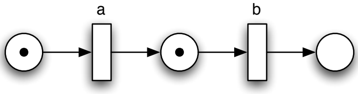
# Contacto

- Dada uma rede elementar $(L, A, F, M_0)$ existe um *contacto* entre duas acções $a, b \in A$ numa marcação $M \subseteq L$ sse
  - $b$ está activa em $M$ enquanto que $a$ não

$$M[b\rangle \wedge \neg M[a\rangle$$

  - mas depois de $b$ ocorrer $a$ já pode ocorrer

$$M[b\rangle N \wedge N[a\rangle$$

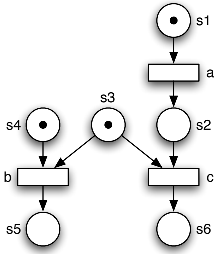



- Em geral, existe contacto numa marcação $M$ para um evento $a$ sse

$$^\bullet a \subseteq M \wedge a^\bullet \cap (M - {}^\bullet a) \neq \emptyset$$

# Confusão

- Uma mistura entre concorrência e conflito pode dar origem a uma situação denominada *confusão*. Em princípio duas acções concorrentes são independentes, mas em certos casos podem interferir uma com a outra.
- Na seguinte rede *a* e *b* são concorrentes, mas dependendo da ordem em que ocorrem podemos ter ou não conflitos.




$$\{s_1, s_3, s_4\}[\{a, b\}\rangle\{s_2, s_5\}$$

$$\{s_1, s_3, s_4\}[b\rangle\{s_1, s_5\}$$

$$\{s_1, s_5\}[a\rangle\{s_2, s_5\}$$

$$\{s_1, s_3, s_4\}[a\rangle\{s_2, s_3, s_4\}$$

$$\{s_2, s_3, s_4\}[b\rangle \wedge \{s_2, s_3, s_4\}[c\rangle$$

$$\neg \{s_2, s_3, s_4\}[\{b, c\}\rangle$$

# Grafo de Acessibilidade

- Dada uma rede elementar $R = (L, A, F, M_0)$ o conjunto de marcações acessíveis a partir de uma dada marcação $M \subseteq L$ é o menor conjunto $reach(M)$ tal que:

$$\begin{gathered} M \in reach(M) \\ \wedge \\ \exists a \in A \cdot M_1[a\rangle M_2 \ \wedge \ M_1 \in reach(M) \ \Rightarrow \ M_2 \in reach(M) \end{gathered}$$

- O *grafo de acessibilidade* de $R$ é um sistema de transição $(S, i, L, T)$ onde

$$\begin{aligned} S &= reach(M_0) \\ i &= M_0 \\ L &= A \\ T &= \{(M_1, a, M_2) \in S \times L \times S \mid M_1[a\rangle M_2\} \end{aligned}$$

# Cálculo do Grafo de Acessibilidade

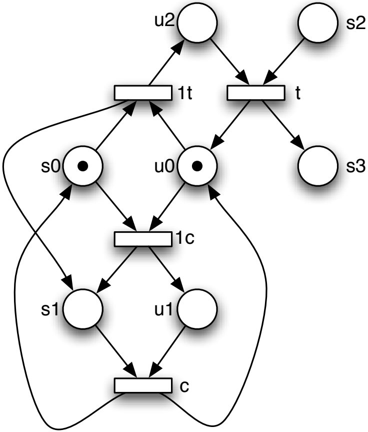
grafo $(L, A, F, M_0) \equiv$
    $S \leftarrow \{M_0\}$;
    $T \leftarrow \emptyset$;
    Stack $X \leftarrow$ **new** Stack();
    $X$.push($M_0$);
    **while** $\neg X$.empty()
        $M \leftarrow X$.pop();
        **for** $a \in$ enabled($M$);
            $N \leftarrow$ next($M, a$);
            **if** $N \notin S$
                $S \leftarrow S \cup \{N\}$;
                $X$.push($N$);
            $T \leftarrow T \cup \{(M, a, N)\}$;
    **return** $(S, M_0, A, T)$;

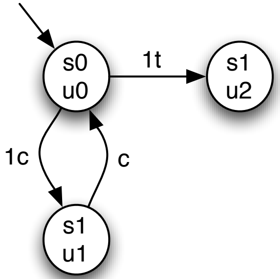
$$\text{enabled}(M) \equiv \{a \in A \mid M[a\rangle\}$$

$$\text{next}(M, a) \equiv N \Leftrightarrow M[a\rangle N$$

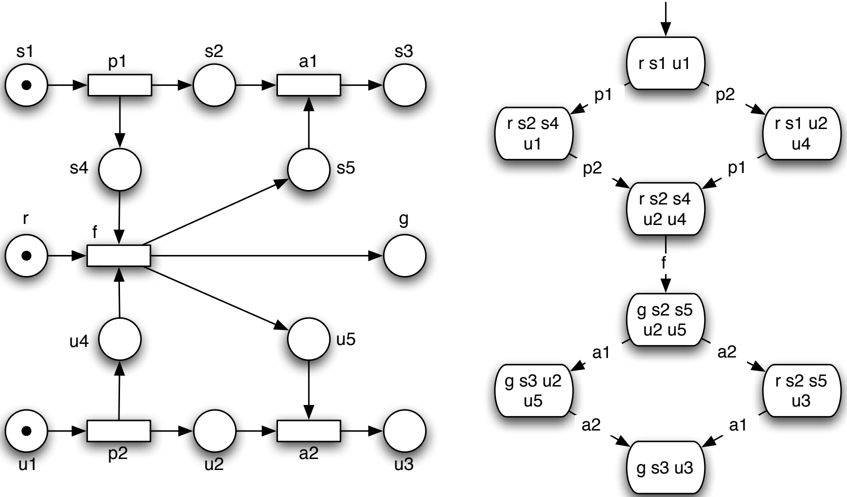
Trocando a *stack* por uma *queue* temos uma travessia do grafo *breath-first* em vez de *depth-first*, com a garantia de encontrar o menor caminho até um estado de erro.

# Exemplo

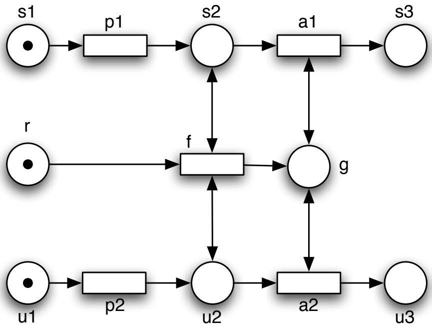


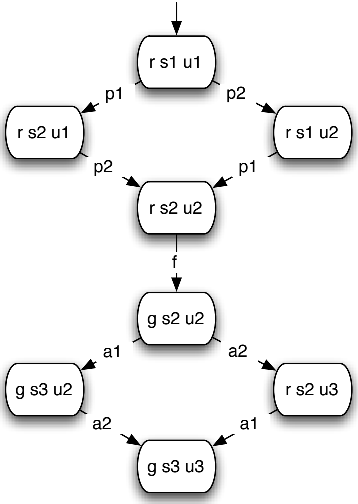



# Exemplo


s1
p1
s2
a1
s3
s4
s5
r
f
g
u4
u5
u1
p2
u2
a2
u3
r s1 u1
p1
p2
r s2 s4 u1
r s1 u2 u4
p2
p1
r s2 s4 u2 u4
f
g s2 s5 u2 u5
a1
a2
g s3 u2 u5
r s2 s5 u3
a2
a1
g s3 u3

# Exemplo


s1
p1
s2
a1
s3
r
f
g
u1
p2
u2
a2
u3

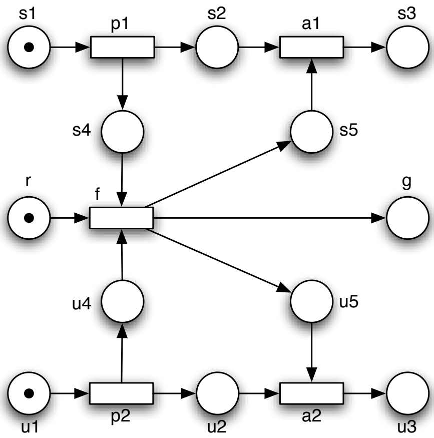

r s1 u1
p1
p2
r s2 u1
r s1 u2
p2
p1
r s2 u2
f
g s2 u2
a1
a2
g s3 u2
r s2 u3
a2
a1
g s3 u3

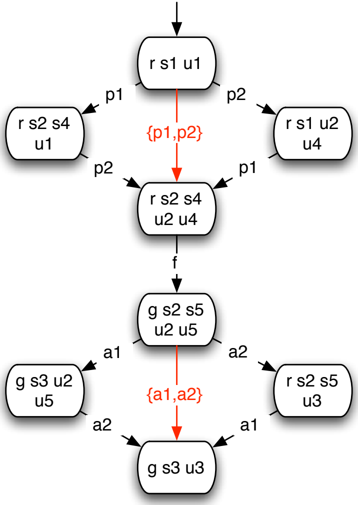
# Semântica Não-Entrelaçada

- Os sistemas de transição normais não capturam a concorrência verdadeira.
- Mas podemos usar a variante *step*, onde as transições são etiquetadas por conjuntos de acções.
- Usando semântica *step* o grafo de acessibilidade de $(L, A, F, M_0)$ é um sistema de transição $(S, i, L, T)$ onde

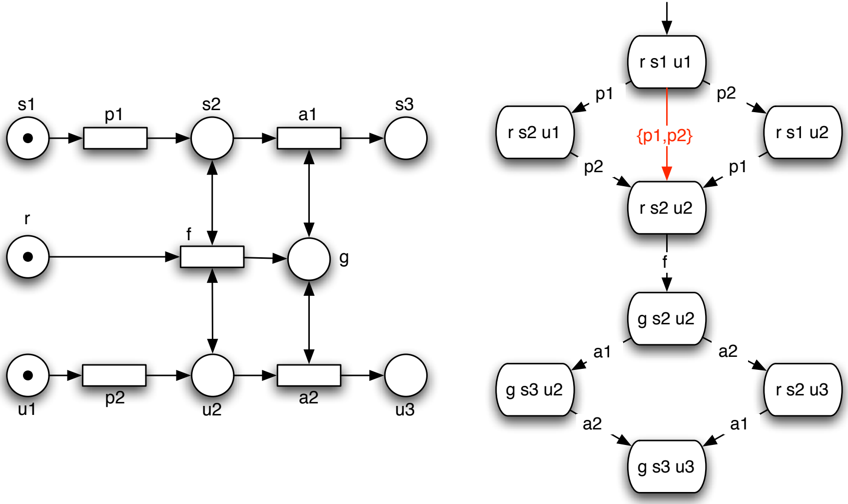
$$
\begin{aligned}
S &= reach(M_0) \\
i &= M_0 \\
L &= \mathcal{P}(A) \\
T &= \{(M_1, B, M_2) \in S \times L \times S \mid M_1[B\rangle M_2\}
\end{aligned}
$$

# Exemplo

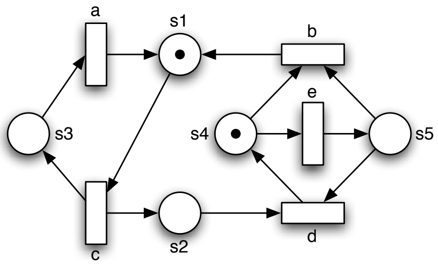

s1
p1
s2
a1
s3
s4
s5
r
f
g
u4
u5
u1
p2
u2
a2
u3

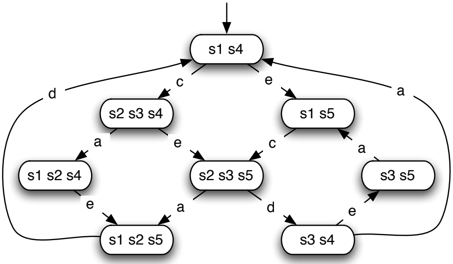

r s1 u1
p1
p2
r s2 s4
u1
{p1,p2}
r s1 u2
u4
p2
p1
r s2 s4
u2 u4
f
g s2 s5
u2 u5
a1
a2
g s3 u2
u5
{a1,a2}
r s2 s5
u3
a2
a1
g s3 u3

# Exemplo

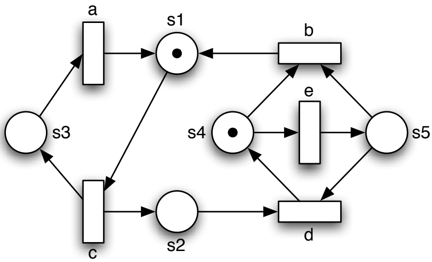

s1
p1
s2
a1
s3
r
f
g
u1
p2
u2
a2
u3
r s1 u1
p1
p2
r s2 u1
{p1,p2}
r s1 u2
p2
p1
r s2 u2
f
g s2 u2
a1
a2
g s3 u2
r s2 u3
a2
a1
g s3 u3

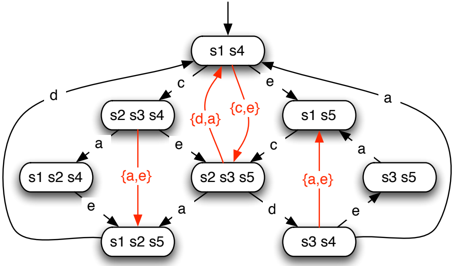
# Exemplo


a
s1
b
e
s3
s4
s5
c
s2
d


s1 s4
d
c
e
a
s2 s3 s4
s1 s5
a
e
c
a
s1 s2 s4
s2 s3 s5
s3 s5
e
a
d
e
s1 s2 s5
s3 s4

# Exemplo


a
s1
b
e
s3
s4
s5
c
s2
d


s1 s4
d
c
e
a
{d,a}
{c,e}
s2 s3 s4
s1 s5
a
e
c
a
s1 s2 s4
{a,e}
s2 s3 s5
{a,e}
s3 s5
e
a
d
e
s1 s2 s5
s3 s4

# Eliminação de Contactos

- Uma rede elementar $(L, A, F, M_0)$ diz-se *livre de contactos* sse
$$\forall a \in A, M \subseteq L \cdot {}^{\bullet}a \subseteq M \Rightarrow a^{\bullet} \cap (M - {}^{\bullet}a) = \emptyset$$
- Numa rede livre de contactos a condição de activação pode ser simplificada.
$$M[a\rangle \Leftrightarrow {}^{\bullet}a \subseteq M$$
- É possível transformar qualquer rede elemenar numa rede livre de contactos com o mesmo comportamento, ou seja, com um grafo de acessibilidade isomorfo.

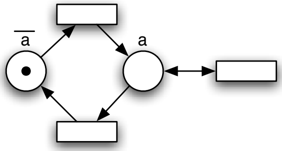
# Lugares Complementares

- Dada uma rede elementar $(L, A, F, M_0)$ dois lugares $s, \overline{s} \in L$ são *complementares* sse

$$\forall M \in reach(M_0) \cdot s \in M \dot{\vee} \overline{s} \in M$$

- Estaticamente caracterizam-se por inverterem o seu papel perante as acções

$$\forall a \in A \cdot (s \in {}^{\bullet}a \ \wedge \ s \notin a^{\bullet}) \ \Leftrightarrow \ (\overline{s} \in a^{\bullet} \ \wedge \ \overline{s} \notin {}^{\bullet}a)$$




- Os loops não são considerados pois conservam o número de marcas.

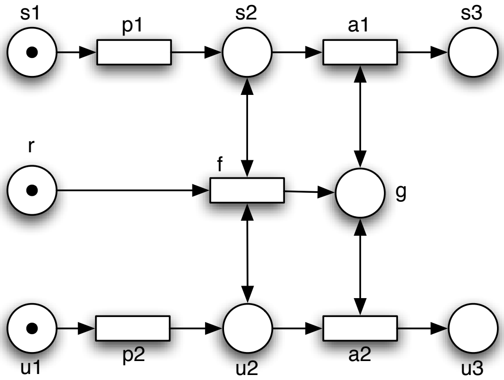
# Eliminação de Contactos

- Dada rede elementar $(L, A, F, M_0)$, um conjunto de novos lugares $\overline{L}$ disjunto de $L \cup A$, e uma bijecção $\gamma : L \rightarrow \overline{L}$, é possível calcular uma rede equivalente livre de contactos $(L', A', F', M_0')$ onde

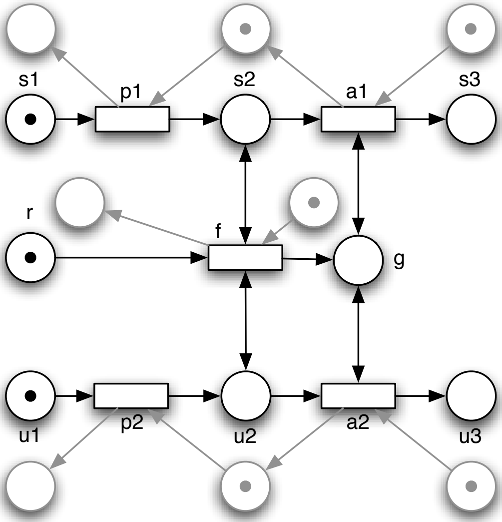
$$
\begin{aligned}
L' &= L \cup \overline{L} \\
A' &= A \\
F' &= F \cup \{(a, \gamma(s)) \mid (s, a) \in F \wedge (a, s) \notin F\} \\
&\quad \cup \{(\gamma(s), a) \mid (a, s) \in F \wedge (s, a) \notin F\} \\
M_0' &= M_0 \cup \gamma(L - M_0)
\end{aligned}
$$

- Caso se pretenda poupar nodos ou que a rede resultante seja simples, só devem ser criados lugares complementares para aqueles que não os tiverem já na rede original.

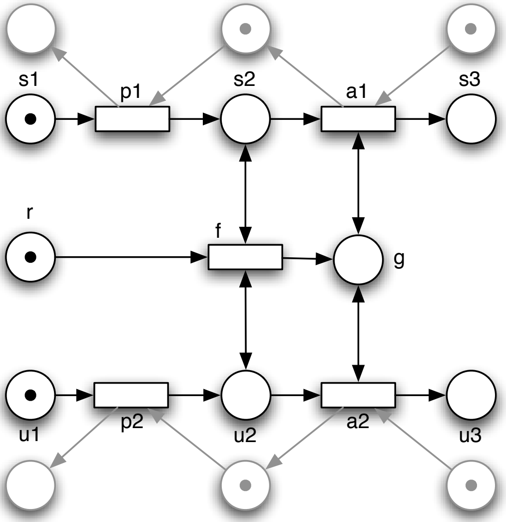
# Exemplo



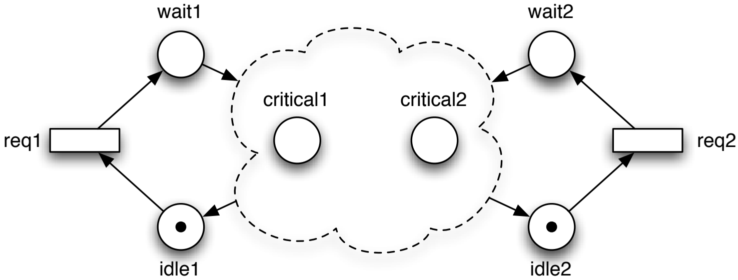
# Exemplo



# Exemplo



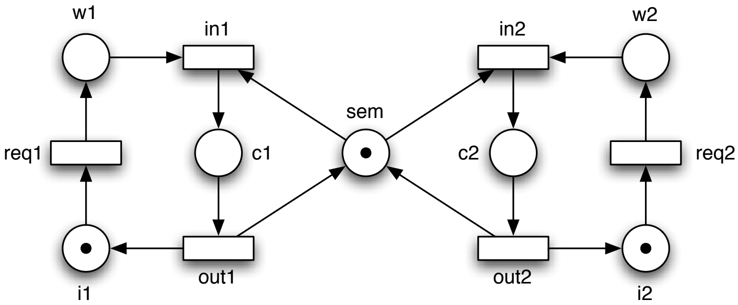
# Exclusão Mútua




**Exclusão Mútua** Os processos não podem estar simultaneamente nas regiões críticas.

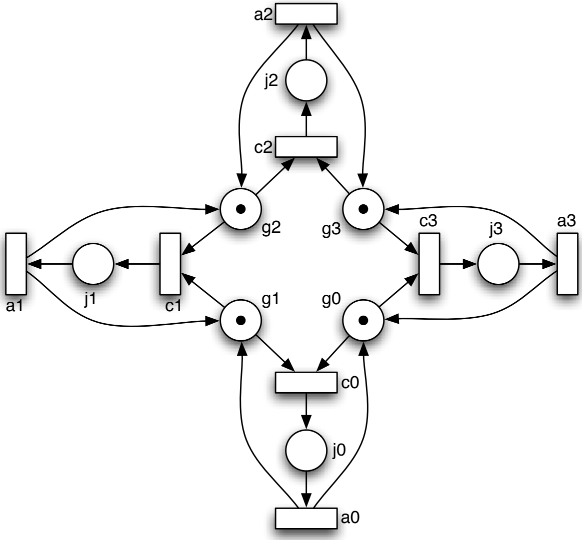
**Evolução** Sempre que um processo está à espera eventualmente estará na sua região crítica e depois na região não crítica.

# Exclusão Mútua com Semáforos

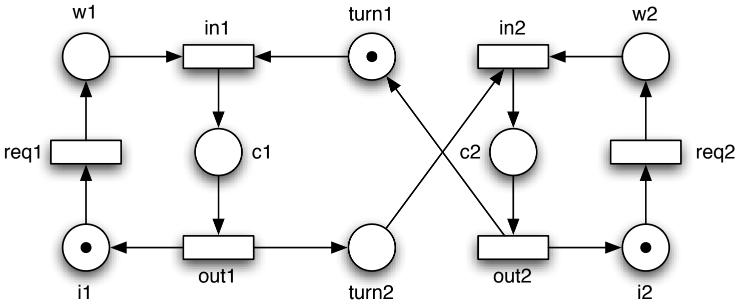



- A não ser que haja um mecanismo global que garanta justiça, não garante a propriedade de evolução.

# O Jantar dos Filósofos



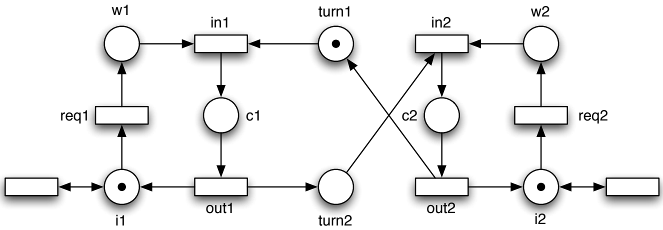
# Exclusão Mútua com Mutex




- Obriga a uma alternância estrita entre os processos.
- Se um dos processos não quiser aceder ao recurso crítico o outro pode ficar indefinidamente à espera. Em princípio não garante evolução.
- Mas será que o nosso modelo semântico permite alguma execução onde tal aconteça?

# Exclusão Mútua com Mutex

- De facto, o estado que representa a região não crítica é uma abstração de um sistema que não é modelado.
- Sendo assim, não podemos assumir que a acção de *request* é necessariamente executada sempre e logo depois da saída da região crítica.
- Para modelar correctamente este facto, alguns autores introduzem a nocção de acção quiescente, mas podemos obter o mesmo efeito com um *loop*.



# Algoritmo de Peterson

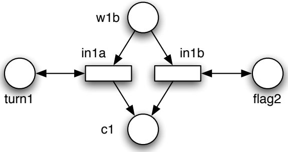
$flag_1 \leftarrow \text{false};\ flag_2 \leftarrow \text{false};$
$turn \leftarrow 1;$

$$
\begin{array}{l}
\textbf{while } \text{true} \\
\quad\quad \textit{Região não crítica} \\
\quad i_1 : flag_1 \leftarrow \text{true}; \\
\quad w_{1a} : turn \leftarrow 2; \\
\quad w_{1b} : \textbf{while } flag_2 \wedge turn = 2; \\
\quad\quad \textit{Região crítica} \\
\quad c_1 : flag_1 \leftarrow \text{false};
\end{array}
\quad \Big\| \quad
\begin{array}{l}
\textbf{while } \text{true} \\
\quad\quad \textit{Região não crítica} \\
\quad i_2 : flag_2 \leftarrow \text{true}; \\
\quad w_{2a} : turn \leftarrow 1; \\
\quad w_{2b} : \textbf{while } flag_1 \wedge turn = 1; \\
\quad\quad \textit{Região crítica} \\
\quad c_2 : flag_2 \leftarrow \text{false};
\end{array}
$$

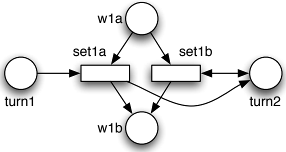
- Garante a exclusão mútua e a evolução.
- O primeiro a requisitar tem prioridade.
- Nenhum espera mais do que uma volta para entrar.

# Algoritmo de Peterson

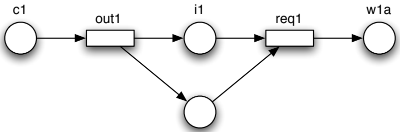
- $w_{1b}$ : **while** ($flag_2 \wedge turn = 2$);



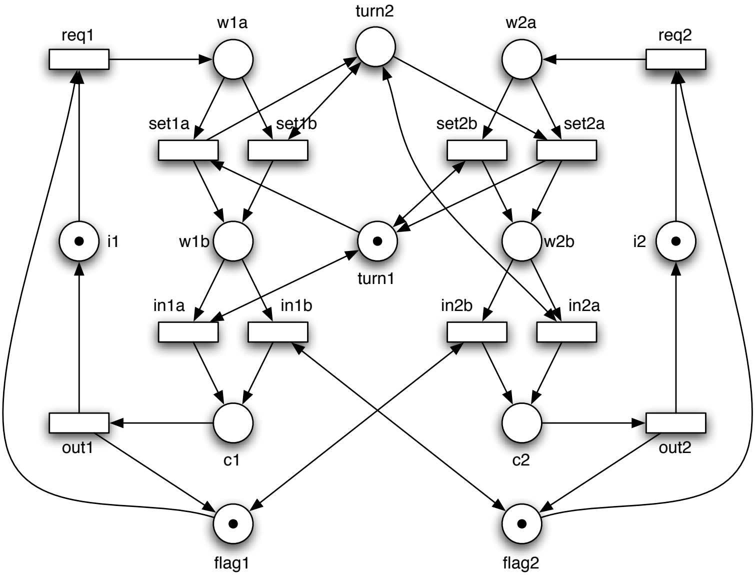

- $w_{1a}$ : $turn \leftarrow 2$;




- $c_1$ : $flag_1 \leftarrow$ false; $i_1$ : $flag_1 \leftarrow$ true;

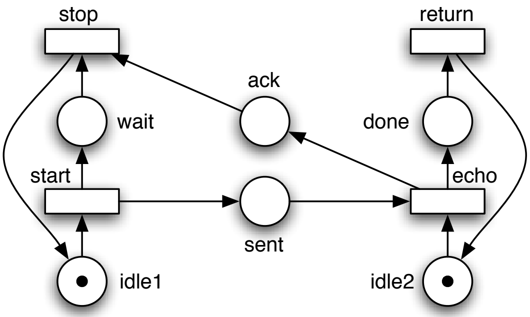


# Algoritmo de Peterson

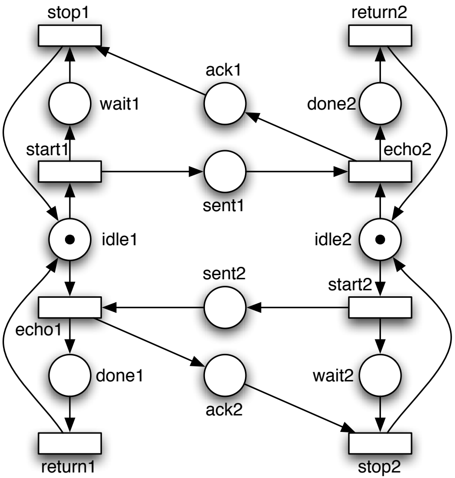


# Conversa Cruzada

- Dois agentes operam independentemente e apenas podem comunicar de forma assíncrona.
- De tempos a tempos pretendem interagir. Esta interação pode ser iniciada por qualquer dos dois.
- A interação começa com o envio de uma mensagem, devendo o outro agente acusar a sua recepção.



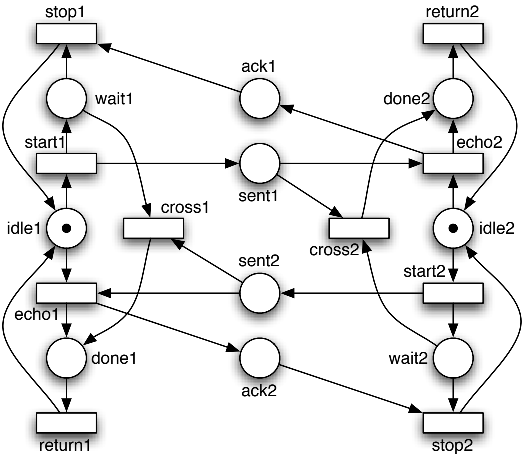
# Conversa Cruzada




- Em caso de conversa cruzada pode haver um *deadlock*.

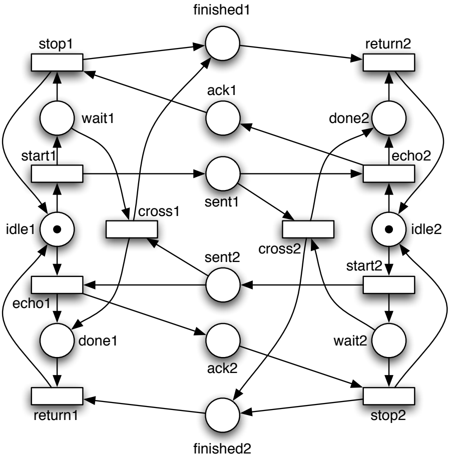
# Conversa Cruzada

- É possível adicionar uma acção para detectar a conversa cruzada, mas será suficiente?



# Conversa Cruzada

- Para evitar que os *rounds* se confundam é necessário acrescentar uma mensagem explícita de finalização.



# Redes P/T

- Cada lugar pode conter mais do que uma marca. Os lugares devem ser interpretados como recursos.
- Pode haver mais do que um arco entre um lugar e uma acção e vice-versa. O número de arcos repetidos será representado por um peso num único arco.
- Por omissão os lugares tem capacidade ilimitada:
  - Uma acção está activa se cada lugar de entrada tiver pelo menos tantas marcas quanto o peso indicado no arco respectivo.
  - Quando ocorre são retiradas as marcas dos lugares de entrada e adicionadas aos lugares de saída na quantidade indicada pelos pesos respectivos.
  - Não há contactos, mas o número de estados do grafo de acessibilidade pode ser ilimitado.

# Definição Formal

## Definição

Uma *rede P/T* é um tuplo

$$(L, A, F, W, M_0)$$

onde

- $(L, A, F)$ é uma rede;
- $W : F \to (\mathbb{N} - \{0\})$ determina o peso nos arcos;
- $M_0 : L \to \mathbb{N}$ é a marcação inicial.

- Normalmente as marcações serão representadas por multiconjuntos ou vectores de inteiros (assumindo uma ordem nos lugares).

# Activação e Disparo

- É conveniente extender a definição de $W$ a todo o conjunto $(L \times A) \cup (A \times L)$. Dada uma rede P/T $(L, A, F, W, M_0)$ temos

$$\underline{W}(x, y) = \begin{cases} 0 & \Leftarrow (x, y) \notin F \\ W(x, y) & \Leftarrow (x, y) \in F \end{cases}$$

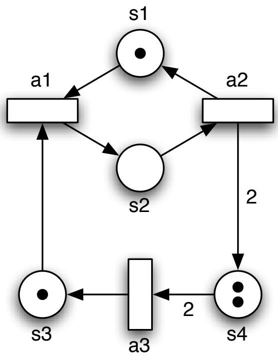
- Uma acção $a \in A$ está activa numa marcação $M : L \to \mathbb{N}$ sse

$$M[a\rangle \;\Leftrightarrow\; \forall s \in L \cdot M(s) \geq \underline{W}(s, a)$$

- Se $M[a\rangle$, a marcação resultante do disparo $M[a\rangle N$ define-se como

$$N(s) = M(s) - \underline{W}(s, a) + \underline{W}(a, s)$$

# Exemplo




$$\{s_1, s_3, s_4, s_4\}[a_1\rangle\{s_2, s_4, s_4\}$$

$$\{s_1, s_3, s_4, s_4\}[a_3\rangle\{s_1, s_3, s_3\}$$

$$\{s_2, s_4, s_4\}[a_2\rangle\{s_1, s_4, s_4, s_4, s_4\}$$

$$\neg\{s_1, s_3, s_4\}[a_3\rangle$$

# Concorrência

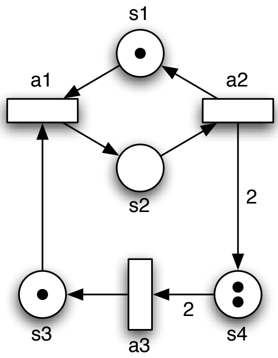
- A independência deixa de ser condição necessária para que duas acções possam ocorrer concorrentemente. Desde que as pré-condições tenham marcas suficientes tal pode acontecer.
- Dada uma rede P/T $(L, A, F, W, M_0)$ um conjunto de acções $B \subseteq A$ está activo concorrentemente numa marcação $M : L \rightarrow \mathbb{N}$ sse

$$M[B\rangle \Leftrightarrow \forall s \in L \cdot M(s) \geq \sum_{a \in B} \underline{W}(s, a)$$

- Se $M[B\rangle$, a marcação resultante do disparo $M[B\rangle N$ define-se como

$$N(s) = M(s) - \sum_{a \in B} (\underline{W}(s, a) + \underline{W}(a, s))$$

# Exemplo



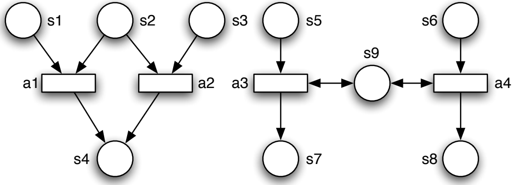

$$\{s_1, s_3, s_4, s_4\}[a_1, a_3\rangle\{s_2, s_3\}$$

$$\{s_2, s_4, s_4\}[a_2, a_3\rangle\{s_1, s_3, s_4, s_4\}$$

$$\{s_1, s_2, s_3\}[a_1, a_2\rangle\{s_1, s_2, s_4, s_4\}$$

# Conflito

- O conflito define-se da mesma forma que nas redes elementares.
- Não é possível existir um *backwards conflit*.




$$\neg\{s_1, s_2, s_3\}[a_1, a_2\rangle$$

$$\{s_1, s_2, s_2, s_3\}[a_1, a_2\rangle\{s_4, s_4\}$$

$$\neg\{s_5, s_6, s_9\}[a_3, a_4\rangle$$

# Grafo de Acessibilidade

- O grafo de acessibilidade define-se de forma idêntica às redes elementares.
- Mas pode não ser finito se a rede não for limitada.
- Uma rede $(L, A, F, W, M_0)$ diz-se *limitada* a $k$ marcas sse

$$\forall M \in reach(M_0), s \in L \cdot M(s) \leq k$$

- O grafo de acessibilidade de uma rede limitada a $k$ marcas tem no máximo $(k+1)^{|L|}$ estados.
- Na construção do grafo de acessibilidade podemos detectar que a rede é ilimitada quando se encontra uma marcação pontualmente maior que outra previamente visitada.

$$M \gneq N \Leftrightarrow \forall s \in L \cdot M(s) \geq N(s) \wedge \exists u \in L \cdot M(u) > N(u)$$

# Cálculo do Grafo de Acessibilidade

grafo $(L, A, F, W, M_0) \equiv$

$S \leftarrow \{M_0\}$;

$T \leftarrow \emptyset$;

Queue $X \leftarrow$ **new** Queue();

$X$.enqueue($M_0$);

**while** $\neg X$.empty()

$M \leftarrow X$.dequeue();

**for** $a \in$ enabled($M$);

$N \leftarrow$ next($M, a$);

**if** $\exists P \in S \cdot N \gneqq P \wedge N \in reach(P)$

**abort**;

**if** $N \notin S$

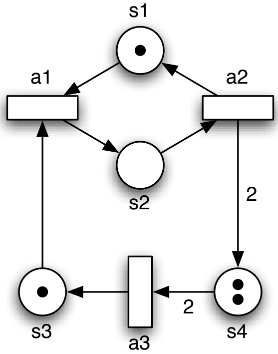
$S \leftarrow S \cup \{N\}$;

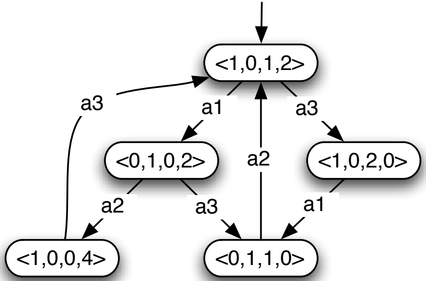
$X$.enqueue($N$);

$T \leftarrow T \cup \{(M, a, N)\}$;

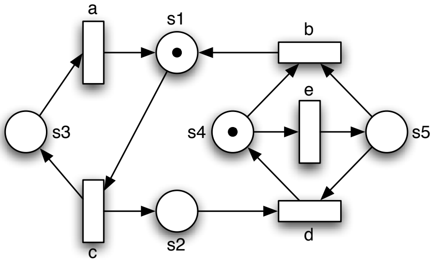
**return** $(S, M_0, A, T)$;

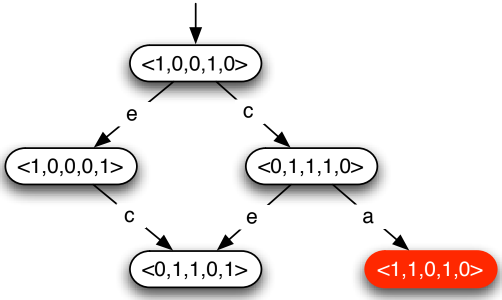
# Exemplo


s1
a1
a2
s2
2
s3
a3
2
s4


<1,0,1,2>
a3
a1
a3
<0,1,0,2>
a2
<1,0,2,0>
a2
a3
a1
<1,0,0,4>
<0,1,1,0>

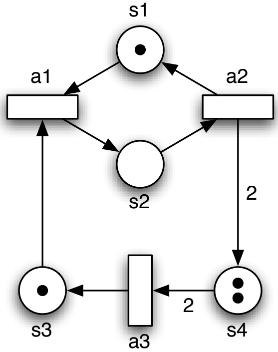
# Exemplo

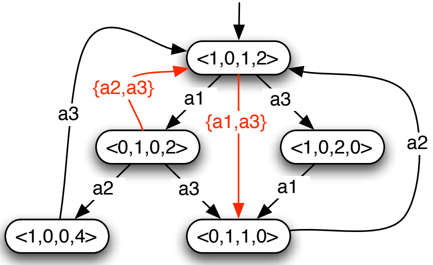


# Semântica Não-Entrelaçada

- Define-se de igual forma às redes elementares.



# Capacidades Explícitas

## Definição

Uma rede P/T com capacidades explícitas é um tuplo

$$(L, A, F, W, K, M_0)$$

onde

- $(L, A, F, W, M_0)$ é uma rede P/T normal;
- $K : L \to (\mathbb{N} \cup \{\infty\})$ determina a capacidade dos lugares.

- As capacidades, quando diferentes de $\infty$, serão anotadas junto ao identificador do lugar.
- Tal como nas redes elementares passamos a ter contactos.
- Mas podem ser convertidas numa rede P/T normal sem capacidades.

# Capacidades Explícitas

- Dada uma rede $(L, A, F, W, K, M_0)$ a condição de activação é diferente:

$$M[a\rangle \Leftrightarrow \begin{array}{l} \forall s \in {}^{\bullet}a \cdot M(s) \geq W(s,a) \wedge \\ \forall s \in a^{\bullet} \cdot M(s) - \underline{W}(s,a) + W(a,s) \leq K(s) \end{array}$$

- A eliminação de contactos baseia-se mais uma vez na introdução de lugares complementares numa rede normal. Dada uma rede $(L, A, F, W, M_0)$ dois lugares $s$ e $\overline{s}$ são complementares sse

$$\exists k \cdot \forall M \in reach(M_0) \cdot M(s) + M(\overline{s}) = k$$

- O lugar $\overline{s}$ conta o número de *vagas* disponíveis em $s$.

# Eliminação de Contactos

- Dada uma rede com capacidades $(L, A, F, W, K, M_0)$, uma bijecção $\gamma : \{s \in L \mid K(s) \neq \infty\} \to \overline{L}$, onde $\overline{L}$ é um conjunto de novos lugares disjunto de $L \cup A$, e $\Delta_{s,a} \equiv \underline{W}(a, s) - \underline{W}(s, a)$, então é possível definir uma rede P/T normal equivalente $(L', A', F', W', M'_0)$ onde

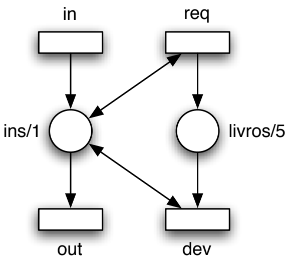
$$
\begin{aligned}
L' &= L \cup \overline{L} \ \wedge \ A' = A \\
F' &= F \ \cup \{(a, \gamma(s)) \mid K(s) \neq \infty \ \wedge \ \Delta_{s,a} < 0\} \\
&\quad\ \cup \{(\gamma(s), a) \mid K(s) \neq \infty \ \wedge \ \Delta_{s,a} > 0\}
\end{aligned}
$$

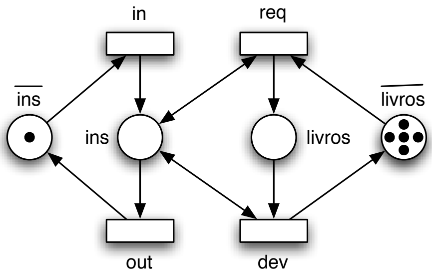
$$
W'(x, y) = \begin{cases} |\Delta_{\gamma^{-1}(x),y}| & \Leftarrow \ x \in \overline{L} \\ |\Delta_{\gamma^{-1}(y),x}| & \Leftarrow \ y \in \overline{L} \\ W(x, y) & \text{otherwise} \end{cases}
$$

$$
M'_0(s) = \begin{cases} M_0(s) & \Leftarrow \ s \in L \\ K(\gamma^{-1}(s)) - M_0(\gamma^{-1}(s)) & \Leftarrow \ s \in \overline{L} \end{cases}
$$

# Exemplo

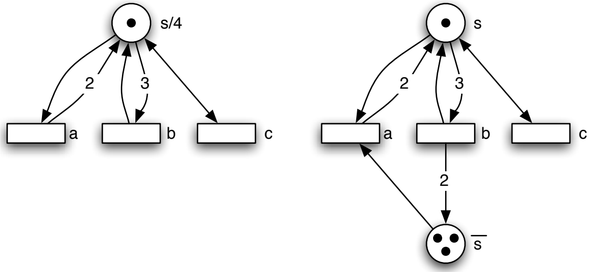
- Pretende-se modelar a interacção de um utente com uma biblioteca.
- Só pode requisitar e devolver livros (no máximo de 5) se estiver inscrito.



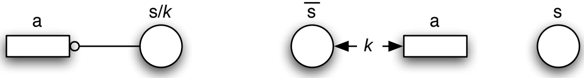
# Exemplo

- Atenção ao facto de nas redes P/T os loops não implicarem conservação de marcas.
- Nos nodos complementares nunca vão aparecer loops.



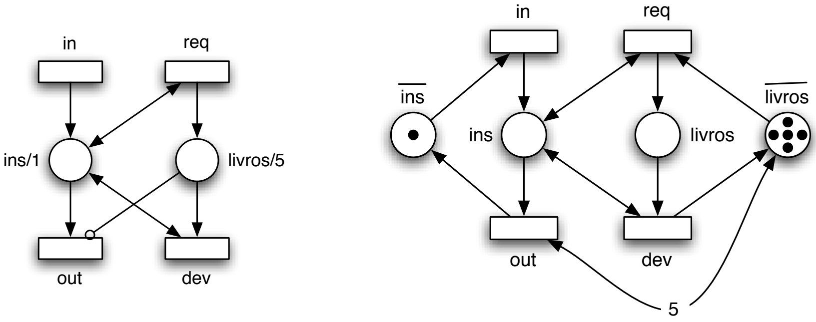
# Arcos Inibidores

- São arcos que ligam lugares a acções e que impedem uma acção de ocorrer quando o lugar tem marcas.
- Ao contrário das capacidades conferem um verdadeiro aumento na capacidade de modelação (ficam equivalentes às máquinas de Turing).
- Se o lugar ao qual está ligado tiver uma capacidade explícita ou, no caso geral, for limitado a $k$ marcas, é possível substituir o arco inibidor por um teste no lugar complementar (loop com peso $k$).

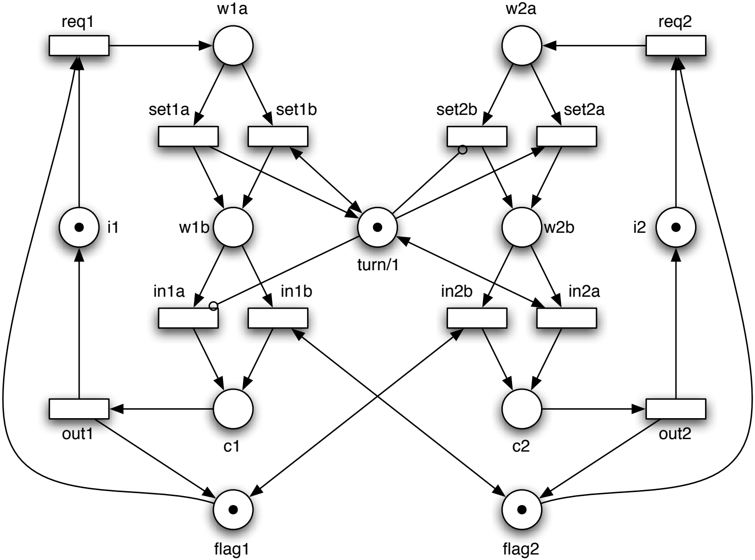


# Exemplo

- Um utente só pode deixar de pertencer à biblioteca depois de devolver todos os livros.



# Algoritmo de Peterson



# Grafo de Cobertura

- Quando uma rede é ilimitada, é possível calcular um *grafo de cobertura* que nos dá uma aproximação do seu comportamento.
- Os estados deste sistema de transição são *marcações*-$\omega$, onde $\omega$ irá denotar um número arbitrário de marcas (possivelmente infinito).
- É conveniente extender a aritmética dos naturais da seguinte forma, onde $n \in \mathbb{N}$.

$$n + \omega = \omega + n = \omega$$

$$\omega + \omega = \omega$$

$$\omega - n = \omega$$

$$0 \cdot \omega = 0 \qquad \omega \cdot \omega = \omega$$

$$n \cdot \omega = \omega \cdot n = \omega \Leftarrow n > 0$$

$$n \leq \omega \qquad \omega \leq \omega$$

# Grafo de Cobertura

- Com esta extensão da aritmética, a definição das condições de activação e disparo para marcações-$\omega$ é equivalente.
  - Se um lugar de entrada para uma acção tem $\omega$ marcas, considera-se que esse lugar tem marcas suficientes para o seu disparo, independentemente do peso no arco respectivo.
  - O disparo de uma acção associada a um lugar com $\omega$ marcas não afecta a sua marcação.
- Na construção do grafo de cobertura, sempre que se encontrar uma marcação pontualmente superior a outra previamente visitada serão colocadas $\omega$ marcas nos lugares com marcação superior.

# Cálculo do Grafo de Cobertura

grafo $(L, A, F, W, M_0) \equiv$

$S \leftarrow \{M_0\};$

$T \leftarrow \emptyset;$

Queue $X \leftarrow$ **new** Queue();

$X$.enqueue($M_0$);

**while** $\neg X$.empty()

  $M \leftarrow X$.dequeue();

  **for** $a \in$ enabled($M$);

    $N \leftarrow$ next($M, a$);

    $N \leftarrow$ omegas($M, a, N, S, T$);

    **if** $N \notin S$

      $S \leftarrow S \cup \{N\};$

      $X$.enqueue($N$);

    $T \leftarrow T \cup \{(M, a, N)\};$

**return** $(S, M_0, A, T);$

# Cálculo do Grafo de Cobertura

omegas $(M, a, N, S, T) \equiv$

$\quad$ **for** $P \in S$;

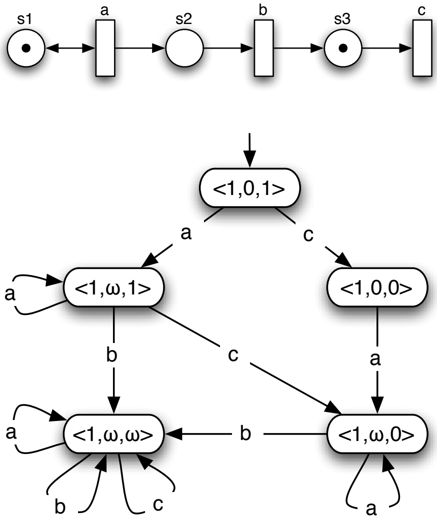
$\quad$ **if** $N \gneqq P \ \wedge\ N \in reach(P)$

$\quad\quad N \leftarrow N + (N - P) \cdot \omega$;

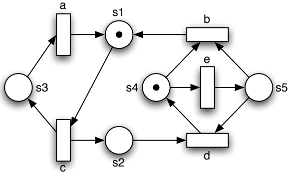
$\quad$ **return** $N$;

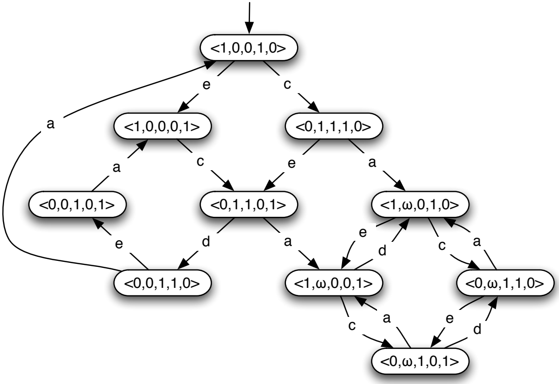
- A função reach define-se à custa do sistema de transição $T$ calculado até ao momento.
- A atribuição faz com que todos os lugares de $P$ cuja marcação é estritamente superior à de $N$ fiquem com $\omega$ marcas.

# Exemplo



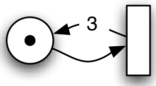
# Exemplo

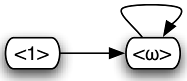

a
s1
b
e
s3
s4
s5
c
s2
d


<1,0,0,1,0>
<1,0,0,0,1>
<0,1,1,1,0>
<0,0,1,0,1>
<0,1,1,0,1>
<1,ω,0,1,0>
<0,0,1,1,0>
<1,ω,0,0,1>
<0,ω,1,1,0>
<0,ω,1,0,1>
a
e
c
d

# Grafo de Cobertura

- Se uma marcação é acessível numa determinada rede então existe alguma marcação-$\omega$ no seu grafo de cobertura que lhe é superior.
- A implicação contrária não se verifica: uma marcação menor que uma das marcações-$\omega$ do grafo de cobertura pode não ser acessível.
- Por exemplo, na seguinte rede apenas marcações com um número ímpar de marcas são acessíveis, mas o grafo de cobertura "esquece" esse facto.




- O grafo de acessibilidade determina com exactidão o comportamento de uma rede, enquanto que o grafo de cobertura apenas o (sobre) aproxima.

# Algumas Propriedades Básicas

- Uma rede pode ser:
  - **Limitada** Quando o número de marcações acessíveis é finito.
  - **Animada** Quando em todas as marcações acessíveis existe potencial para disparar qualquer acção. Uma propriedade relacionada, mas mais fraca, é a ausência de *deadlocks*.
  - **Invertível** Quando é sempre possível voltar à marcação inicial.
- Nas propriedades básicas também se inclui a exclusão mútua: não é possível que dois ou mais lugares estejam marcados simultaneamente.

# Técnicas de Verificação

**Simulação** Baseada no jogo das marcas. Não permite provar propriedades, mas ajuda a perceber o sistema e permite encontrar contra-exemplos.

**Enumeração** Baseada na construção do grafo de acessibilidade ou de cobertura. Devido ao problema da explosão de estados é dificil de aplicar em sistemas reais. Só permite provar propriedades para uma dada marcação inicial.

**Transformação** Baseada na transformação numa rede mais simples, mas que preserve as propriedades a ser verificadas.

**Estrutural** Tenta “deduzir” o comportamento da rede através da análise directa da sua estrutura. A marcação inicial deixa de ser relevante. Normalmente utilizam técnicas de programação linear.

# Verificação por Enumeração

- É conveniente divir as propriedades em duas classes. Dado um predicado $\Pi$ definido sobre marcações temos:
  - Propriedades de *segurança*

$$\forall M \in reach(M_0) \cdot \Pi(M)$$

  - Propriedade de *animação*

$$\forall M \in reach(M_0) \cdot \exists N \in reach(M) \cdot \Pi(M)$$

# Propriedades de Segurança

- Alguns exemplos de propriedades de segurança para uma rede $(L, A, F, W, M_0)$:
  - Um lugar $s$ está limitado a $k$ marcas.

$$\forall M \in reach(M_0) \cdot M(s) \leq k$$

  - Existe exclusão mútua entre os lugares $s$ e $u$.

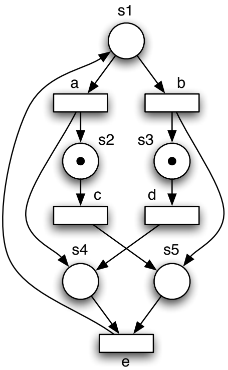
$$\forall M \in reach(M_0) \cdot M(s) = 0 \dot{\vee} M(u) = 0$$

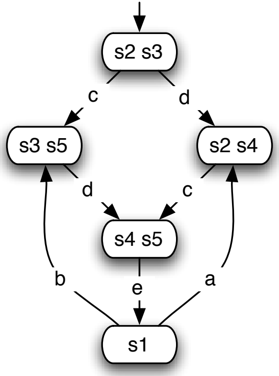
  - Não existem *deadlocks*.

$$\forall M \in reach(M_0) \cdot \exists a \in A \cdot M[a\rangle$$

- Estas propriedades podem ser verificadas percorrendo sistematicamente todos os nodos do grafo de acessibilidade.

# Propriedades de Segurança




- Todos os lugares são limitados a uma marca: a rede é limitada.
- Não tem *deadlocks*.
- Existe exclusão mútua entre $s_2$ e $s_5$, entre $s_3$ e $s_4$, e entre $s_1$ e todos os restantes lugares.

# Propriedades de Animação

- Alguns exemplos de propriedades de animação para uma rede $(L, A, F, W, M_0)$:
  - A rede é invertível.

$$\forall M \in reach(M_0) \cdot \exists N \in reach(M) \cdot N = M_0$$

  - Uma acção $a$ é animada.

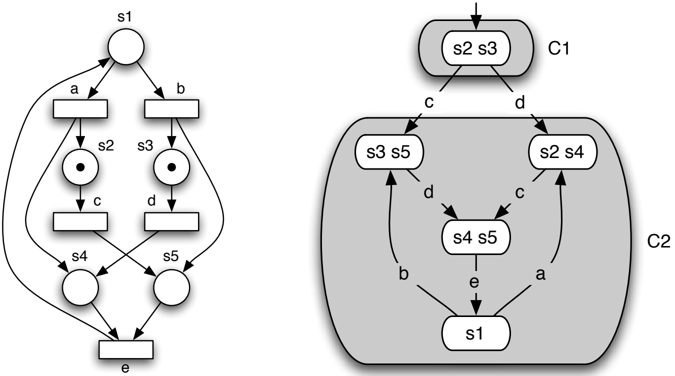
$$\forall M \in reach(M_0) \cdot \exists N \in reach(M) \cdot N[a\rangle$$

- Estas propriedades não podem ser verificadas usando uma simples pesquisa linear no grafo.
- É necessário determinar primeiro os componentes fortemente ligados do grafo.

# Propriedades de Animação

- Um conjunto de estados é fortemente ligado sse existe um caminho entre cada par de estados. Existem algoritmos que determinam estes componentes com complexidade linear no tamanho do grafo.
- A partir do grafo original é possível construir uma versão reduzida, onde os nodos são os seus componentes fortemente ligados, e existe um ramo entre dois componentes sse algum estado do segundo for acessível a partir de um estado do primeiro.
- Este grafo é necessariamente acíclico e os seus componentes terminais são aqueles dos quais não sai qualquer ramo.
- Para verificar uma propriedade de animação basta verificar se em todos os componentes terminais existe pelo menos uma marcação que verifica o respectivo predicado.

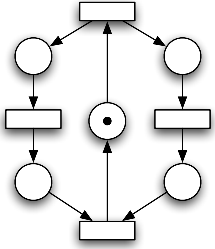
# Propriedades de Animação



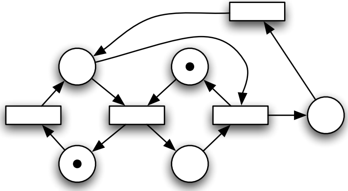

- Não é invertível porque o estado inicial não está em $C_2$.
- Todas as acções são animadas, pois todas estão presentes em $C_2$: a rede é animada.

# Independência das Propriedades

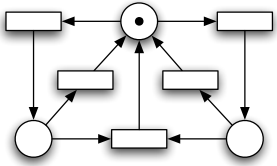
LAI



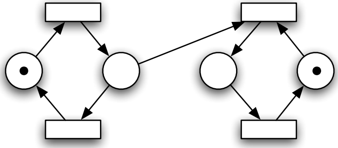
LA¬I



L¬AI

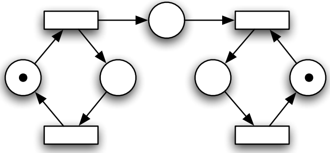


L¬A¬I

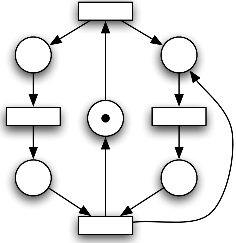


# Independência das Propriedades

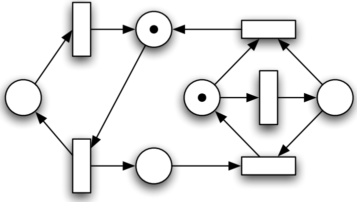
¬LAI



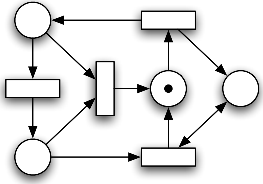
¬LA¬I



¬L¬AI

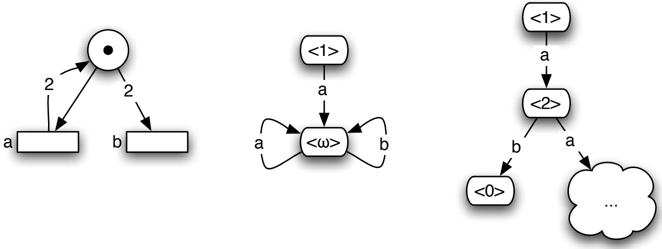


¬L¬A¬I



# Utilidade do Grafo de Cobertura

- Estas técnicas apenas podem ser aplicadas ao grafo de acessibilidade.
- O grafo de cobertura é sempre útil para determinar se um lugar é limitado (mesmo quando a rede não é limitada). Basta que a sua marcação seja sempre diferente de $\omega$.
- O seguinte exemplo mostra como o grafo de cobertura pode falhar a detecção de *deadlocks*.



# Verificação Estrutural

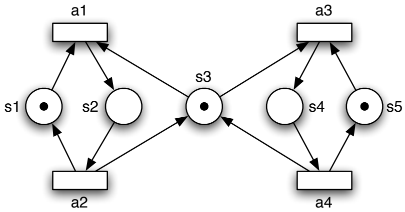
- Devido ao problema da explosão de estados, a verificação de propriedades por enumeração nem sempre é realizável em tempo útil.
- Se a rede for ilimitada, pode mesmo ser impossível verificar certas propriedades.
- A verificação estrutural torna possível verificar algumas propriedades sem ser necessário construir o grafo de acessibilidade.
- A técnica mais conhecida, e que vamos estudar, baseia-se no cálculo de *invariantes de lugar*.

# Matriz de Incidência

- Dada uma rede P/T $(L, A, F, W, M_0)$, a sua *matriz de incidência* $C : L \times A \to \mathbb{Z}$ é uma matriz onde a coluna correspondente ao índice $a \in A$ denota o efeito que o disparo de $a$ tem na marcação da rede.

$$C(s, a) = W(a, s) - W(s, a)$$

- É conveniente assumir que existe uma ordem em $L$ e em $A$ e, normalmente, os valores nulos não se representam.




| $C$ | $a_1$ | $a_2$ | $a_3$ | $a_4$ |
|---|---|---|---|---|
| $s_1$ | $-1$ | $1$ | | |
| $s_2$ | $1$ | $-1$ | | |
| $s_3$ | $-1$ | $1$ | $-1$ | $1$ |
| $s_4$ | | | $1$ | $-1$ |
| $s_5$ | | | $-1$ | $1$ |

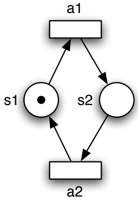
# Matriz de Incidência

- É possível calcular o efeito acumulado do disparo de uma sequência de acções usando simples multiplicação de matrizes.
- Assumindo que o número de vezes que cada acção disparou é contabilizado num vector $B : A \to \mathbb{N}$, o efeito aculumado do disparo é calculado como $C \times B$.
- No nosso exemplo, a marcação $M$ resultante do disparo da sequência $a_1, a_2, a_1$ pode ser determinado como $M = M_0 + C \times B$, onde $B = \langle 2, 1, 0, 0 \rangle$. Note que a orientação dos vectores depende do contexto.

$$\begin{array}{|c|}\hline 1 \\ \hline \\ \hline 1 \\ \hline \\ \hline 1 \\ \hline \end{array} + \begin{array}{|c|c|c|c|}\hline -1 & 1 & & \\ \hline 1 & -1 & & \\ \hline -1 & 1 & -1 & 1 \\ \hline & & 1 & -1 \\ \hline & & -1 & 1 \\ \hline \end{array} \times \begin{array}{|c|}\hline 2 \\ \hline 1 \\ \hline \\ \hline \\ \hline \end{array} = \begin{array}{|c|}\hline \\ \hline 1 \\ \hline \\ \hline \\ \hline 1 \\ \hline \end{array}$$

# Matriz de Incidência

- Devido à existência de loops, nem todas as marcações resultantes da equação anterior correspondem a marcações acessíveis.
- Tal como no grafo de cobertura, o conjunto de marcações obtidas via matriz de incidência é uma (sobre) aproximação do conjunto de marcações acessíveis.
- No entanto, por vezes é possível demonstrar que uma determinada marcação $M$ é inacessível, quando $M_0 + C \times B = M$ não possui qualquer solução natural (onde todos os componentes de $B$ são números naturais).




$$\begin{array}{|c|}\hline 1 \\ \hline \phantom{1} \\ \hline \end{array} + \begin{array}{|c|c|}\hline -1 & 1 \\ \hline 1 & -1 \\ \hline \end{array} \times C = \begin{array}{|c|}\hline 1 \\ \hline 1 \\ \hline \end{array}$$

# Invariantes de Lugar

**Definição**

Um *invariante de lugar* $I : L \rightarrow \mathbb{N}$ é uma solução natural e diferente de $\overline{0}$ da equação $I \times C = \overline{0}$.

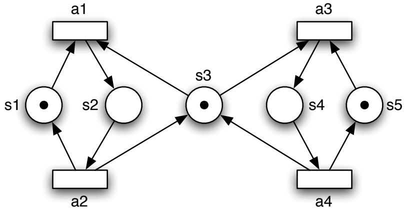
- Seja $M$ uma marcação acessível através de uma sequência de acções contabilizada em $B$, ou seja $M = M_0 + C \times B$. Se $I$ for um invariante de lugar verifica-se a seguinte equação:

$$I \times M = I \times (M_0 + C \times B) = I \times M_0 + I \times C \times B = I \times M_0$$

- Um invariante de lugar permite estabelecer uma relação precisa entre qualquer marcação acessível e a marcação inicial.

# Exemplo

- É trivial verificar que $I_1$,$I_2$ e $I_3$ são invariantes de lugar da seguinte rede.




$$I_1 = \langle 1, 1, 0, 0, 0 \rangle$$
$$I_2 = \langle 0, 1, 1, 1, 0 \rangle$$
$$I_3 = \langle 0, 0, 0, 1, 1 \rangle$$

- Se expandirmos a equação $I_2 \times M = I_2 \times M_0$ obtemos a seguinte equação, válida para qualquer marcação $M$.

$$M(s_2) + M(s_3) + M(s_4) = 1$$

- Esta equação diz-nos que o número de marcas nos lugares $s_2$, $s_4$ e $s_4$ é sempre constante e igual a 1. Com ela é possível demonstrar a exclusão mútua entre $s_2$ e $s_4$ sem contruir o grafo de acessibilidade.

# Invariantes Mínimos

- Qualquer vector obtido por combinação linear (soma e multiplicação escalar) de invariantes é também um invariante.
- Um invariante é *canónico* se o mdc de todos os seus componentes não nulos é igual a 1.
- O *suporte* de um invariante é o conjunto de todos os lugares cujo componente respectivo é não nulo.

## Proposição

Um invariante de lugar é *mínimo* sse é canónico e o seu suporte não contém estritamente o suporte de qualquer outro invariante. O conjunto de invariantes mínimos de uma rede é finito é único.

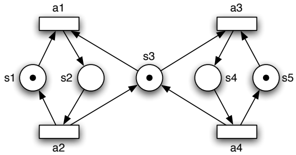
# Cálculo de Invariantes Mínimos

- Dada uma rede $(L, A, F, W, M_0)$, seu conjunto de invariantes mínimos pode ser calculado da seguinte forma.
  1. Seja $[\Phi|\Psi]$ a matriz resultante de justapor a matriz de incidência $\Phi = C$ com uma matriz identidade $\Psi$ de dimensão $|L|$.
  2. Para todo $1 \leq i \leq |A|$
     1. Adicionar à matrix $[\Phi|\Psi]$ todas as linhas que são combinações lineares de pares de linhas de $[\Phi|\Psi]$ e que anulam a $i$-ésima coluna de $\Phi$.
     2. Eliminar de $[\Phi|\Psi]$ todas as linhas nas quais a $i$-ésima coluna de $\Phi$ seja não nula.
  3. Transformar as linhas de $\Psi$ em invariantes canónicos e elinimar os invariantes não mínimos recorrendo à proposição do acetato anterior.

# Cálculo de Invariantes Mínimos

- Note que no final cada linha de $\Psi$ memoriza os coeficientes usados nas combinações lineares de linhas de $C$ que permitiram anular a respectiva linha em $\Phi$.
- Desta forma garante-se que cada linha $\Psi$ é necessariamente um invariante de lugar.
- Está demonstrado que numa rede podem existir um número de invariantes mínimo exponencial no número de de lugares $|L|$.
- Sendo assim, a complexidade deste algoritmo não pode ser polinomial.

# Exemplo




| | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ |
|---|---|---|---|---|---|---|---|---|---|
| $s_1$ | $-1$ | $1$ | | | $1$ | | | | |
| $s_2$ | $1$ | $-1$ | | | | $1$ | | | |
| $s_3$ | $-1$ | $1$ | $-1$ | $1$ | | | $1$ | | |
| $s_4$ | | | $1$ | $-1$ | | | | $1$ | |
| $s_5$ | | | $-1$ | $1$ | | | | | $1$ |
| $\mathbf{s_1 + s_2}$ | | | | | $1$ | $1$ | | | |
| $s_2 + s_3$ | | | $-1$ | $1$ | | $1$ | $1$ | | |
| $\mathbf{s_4 + s_5}$ | | | | | | | | $1$ | $1$ |
| $\mathbf{s_2 + s_3 + s_4}$ | | | | | | $1$ | $1$ | $1$ | |

# Limitação Estrutural

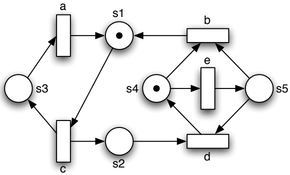
- Usando invariantes podemos provar que na rede anterior $s_3$ é limitado a uma marca.

$$M(s_2) + M(s_3) + M(s_4) = 1$$
$$M(s_3) = 1 - M(s_2) - M(s_4)$$

- De facto esta prova é válida para qualquer marcação inicial, pois se nestes lugares existirem $k$ marcas no início, cada um deles está limitado a $k$ marcas.
- Quando um lugar (ou uma rede) é limitado(a) para qualquer marcação inicial diz-se *estruturalmente limitado(a)*.
- Se exisitir um invariante (não necessariamente mínimo) cujo suporte mencione todos os lugares da rede então esta é estruturalmente limitada.

$$M(s_1) + 2 \cdot M(s_2) + M(s_3) + 2 \cdot M(s_4) + M(s_5) = k$$

# Exemplo




| $a$ | $b$ | $c$ | $d$ | $e$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | −1 | | | 1 | | | | |
| | | 1 | −1 | | | 1 | | | |
| −1 | | 1 | | | | | 1 | | |
| | −1 | | 1 | −1 | | | | 1 | |
| | −1 | | −1 | 1 | | | | | 1 |
| | 1 | | | | 1 | | 1 | | |
| | | | 1 | −1 | 1 | | 1 | 1 | |
| | | | −1 | 1 | 1 | | 1 | | 1 |
| | | | | | 2 | | 2 | 1 | 1 |

# Exemplo

- O seguinte invariante é válido nesta rede:

$$2 \cdot M(s_1) + 2 \cdot M(s_3) + M(s_4) + M(s_5) = 3$$

- Por exemplo, com este invariante conseguimos demonstrar que o lugar $s_1$ é limitado a uma marca.

$$\begin{aligned} 2 \cdot M(s_1) &= 3 - 2 \cdot M(s_3) - M(s_4) - M(s_5) \\ 2 \cdot M(s_1) &\leq 3 \\ M(s_1) &\leq 1 \end{aligned}$$

- Mas infelizmente não nos permite demonstrar a exclusão mútua entre $s_4$ e $s_5$.

$$M(s_4) + M(s_5) \leq 1$$

- Para conseguir demonstrar este facto temos que combinar esta técnica com a detecção de armadilhas...

# Armadilhas

## Definição

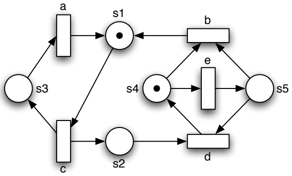
Dada uma rede P/T $(L, A, F, W, M_0)$, uma *armadilha* é um conjunto de lugares $S \subseteq L$ tal que $S^\bullet \subseteq {}^\bullet S$.

- Cada transição que retire marcas de uma armadilha tem que colocar de volta pelo menos uma marca.
- Uma armadilha $S$ diz-se *marcada* numa marcação $M$ sse

$$\exists s \in S \cdot M(s) \geq 1$$

- Se uma armadilha $S$ está marcada em $M_0$ então está marcada em todas as marcações acessíveis.

# Exemplo




- Nesta rede $S = \{s_1, s_3\}$ é uma armadilha pois

$$S^\bullet = \{a, c\} \subseteq \{a, b, c\} = {}^\bullet S$$

- Como $S$ está marcada em $M_0$ temos que $M(s_1) + M(s_3) \geq 1$.
- Este facto em conjunto com o invariante anteriormente calculado permite-nos provar a exclusão mútua.

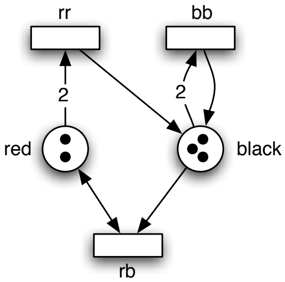
$$2 \cdot M(s_1) + 2 \cdot M(s_3) + M(s_4) + M(s_5) = 3$$

$$M(s_4) + M(s_5) = 3 - 2 \cdot (M(s_1) + M(s_3))$$

$$M(s_4) + M(s_5) \leq 1$$

# Motivação

- O facto de as marcas serem indistinguíveis obriga-nos a definir modelos pouco naturais para certos problemas.
- Por exemplo, no caso do jogo das bolas pretas e vermelhas somos obrigados a modelar o saco usando dois lugares.



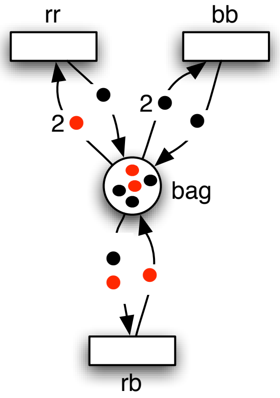

- Esta divisão não seria necessária se as marcas pudessem ser de cores diferentes.

# Marcas Coloridas

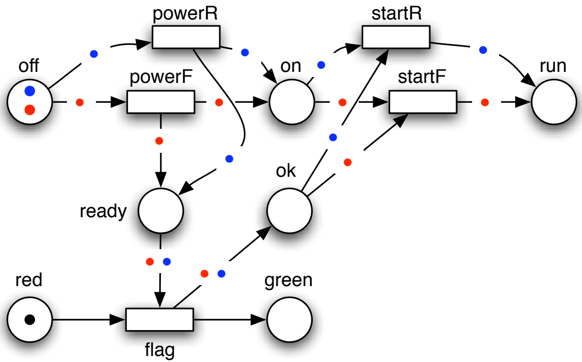
- Vamos então assumir que cada lugar pode ter marcas de cores diferentes. É conveniente ver as marcações como funções de lugares para multiconjuntos de cores.
- As cores permitidas em cada lugar podem ser diferentes.
- Com esta extensão não é suficiente indicar um peso em cada arco: é necessário indicar quantas marcas de cada cor uma acção necessita para ocorrer, e quantas marcas de cada cor coloca nos lugares de saída depois de ocorrer.
- Os pesos nos arcos vão passar também a devolver multiconjuntos de cores.

# Exemplo

- Podemos agora modelar o jogo usando um único lugar para representar todas as bolas no saco.



# Exemplo




- Se num lugar apenas é permitida uma cor, os arcos ligados a esse lugar não necessitam de discriminar a cor das marcas.
- Estas redes correspondem a uma variante simplificada das redes coloridas, designadas *redes de arcos constantes*.

# Redes de Arcos Constantes

## Definição

Uma *rede AC* é um tuplo

$$(L, A, F, C, T, W, M_0)$$

onde

- $(L, A, F)$ é uma rede;
- $C$ é um conjunto de cores;
- $T : L \rightarrow \mathcal{P}(C)$ determina o tipo de cada lugar;
- $W : F \rightarrow \mathcal{M}(C)$ determina o peso nos arcos;
- $M_0 : L \rightarrow \mathcal{M}(C)$ é a marcação inicial.

# Redes de Arcos Constantes

- Dada uma rede AC $(L, A, F, C, T, W, M_0)$ as marcações $M : L \to \mathcal{M}(C)$ estão sujeitas à seguinte restrição:

$$c \notin M(s) \Leftarrow c \notin T(s)$$

- Os pesos dos arcos estão sujeitos a uma restrição semelhante ($\underline{W}$ é equivalente a $W$, mas devolve $\emptyset$ quando o arco não existe):

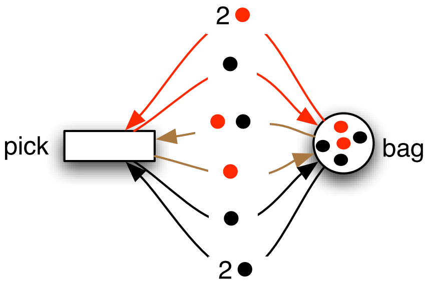
$$c \notin \underline{W}(s, a) \cup \underline{W}(a, s) \Leftarrow c \notin T(s)$$

- A dinâmica da rede define-se de forma usual.

$$M[a\rangle \Leftrightarrow \forall s \in L \cdot \underline{W}(s, a) \subseteq M(s)$$
$$M[a\rangle N \Leftrightarrow \forall s \in L \cdot N(s) = M(s) - \underline{W}(s, a) \cup \underline{W}(a, s)$$

# Motivação

- Nas redes AC continua por vezes a ser necessário desdobrar algumas acções que são únicas no sistema modelado.
- Por exemplo, no jogo a acção de retirar bolas do saco teve de ser desdobrada pois o seu efeito é não determinístico.
- Esta limitação pode ser evitada se as acções tiverem modos de funcionamento, que podem também ser vistos como cores.



# Redes Coloridas

## Definição

Uma *rede colorida* é um tuplo

$$(L, A, F, C, T, W, M_0)$$

onde

- $(L, A, F)$ é uma rede;
- $C$ é um conjunto de cores;
- $T : L \cup A \to \mathcal{P}(C)$ determina o tipo de cada elemento;
- $W : F \to C \hookrightarrow \mathcal{M}(C)$ determina o peso nos arcos;
- $M_0 : L \to \mathcal{M}(C)$ é a marcação inicial.

# Redes Coloridas

- Dada uma rede colorida $(L, A, F, C, T, W, M_0)$ os arcos estão sujeitos a uma restrição adicional:

$$dom(\underline{W}(s, a)) = dom(\underline{W}(a, s)) = T(a)$$

- A condição de activação pode definir-se da seguinte forma.

$$M[a\rangle \Leftrightarrow \exists m \in T(a) \cdot \forall s \in L \cdot \underline{W}(s, a)(m) \subseteq M(s)$$

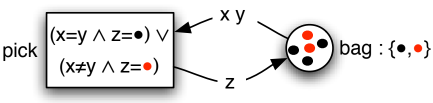
- Existe um disparo $M[a\rangle N$ sse

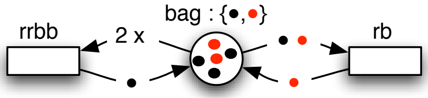
$$\exists m \in T(a) \cdot \forall s \in L \cdot N(s) = M(s) - \underline{W}(s, a)(m) \cup \underline{W}(a, s)(m)$$

- Fixado um modo de disparo, as condições de activação e de disparo de uma acção são idênticas às das redes de arcos constantes.

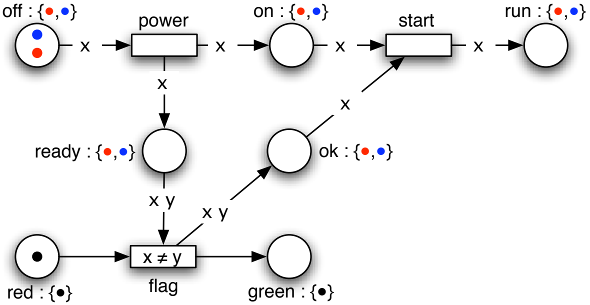
# Redes Coloridas

- Normalmente, o modo de disparo de uma acção é determinado por uma condição definida sobre um conjunto de variáveis.
- As variáveis são introduzidas nos arcos e o seu tipo é o mesmo dos lugares a que estão ligados. A sua visibilidade é restricta à vizinhança da acção.
- Existem tantos modos de disparo numa acção quantas as substituições que tornem a respectiva condição verdadeira.



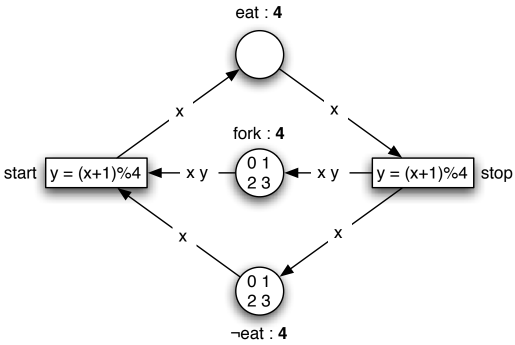



# Exemplo



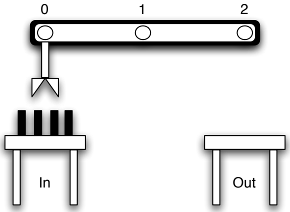

- Se não existe uma condição definida numa acção então qualquer substituição das variáveis é um possível modo de disparo.

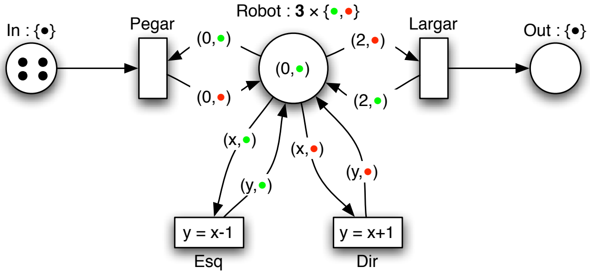
# Exemplo




- O conjunto $\{0 \ldots n - 1\}$ será denotado por **n**.
- Também podemos construir tipos de marcas usando o produto cartesiano. Neste caso é aceitável usar *pattern-matching* para ter acesso directo aos componentes.

# Exemplo

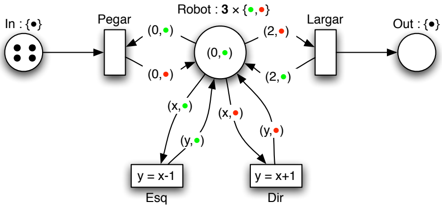

0
1
2
In
Out


In : {•}
Pegar
Robot : **3** × {●,●}
Largar
Out : {•}
(0,●)
(0,●)
(0,●)
(2,●)
(2,●)
(x,●)
(y,●)
(x,●)
(y,●)
y = x-1
Esq
y = x+1
Dir

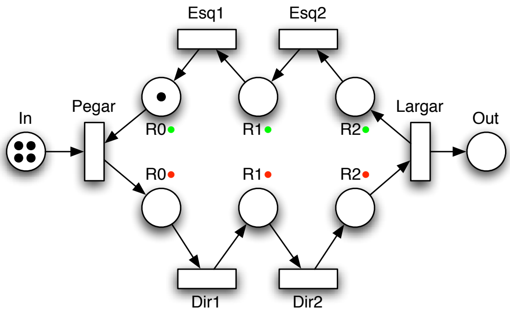
# Conversão de Redes Coloridas para Redes P/T

- Desde que o conjunto de cores $C$ seja finito é possível converter uma rede colorida $(L, A, F, C, T, W, M_0)$ numa rede P/T equivalente.
- Para cada lugar $s \in L$ é criado um lugar $s_c$ para cada $c \in T(s)$.
- Dado que $\mathcal{M}(C) \cong C \rightarrow \mathbb{N}$, a marcação inicial em $s_c$ é $M_0(s)(c)$.
- Para cada transição $a \in T$ é criada uma transição $a_\sigma$ para cada $\sigma \in T(a)$ (é criada uma transição por cada substituição $\sigma$ que torne a condição verdadeira).
- Se existir um arco entre $s$ e $a$ etiquetado com uma variável $x$, existirá um arco entre $s_c$ e $a_\sigma$ se $c$ for o valor atribuído a $x$ por $\sigma$.

# Exemplo



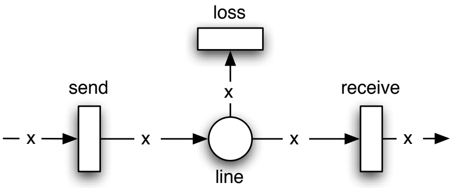
Robot : 3 × {●,●}
In : {●}
Pegar
Largar
Out : {●}
(0,●)
(0,●)
(0,●)
(2,●)
(2,●)
(x,●)
(y,●)
(x,●)
(y,●)
y = x-1
Esq
y = x+1
Dir

Esq1
Esq2
In
Pegar
R0●
R1●
R2●
Largar
Out
R0●
R1●
R2●
Dir1
Dir2

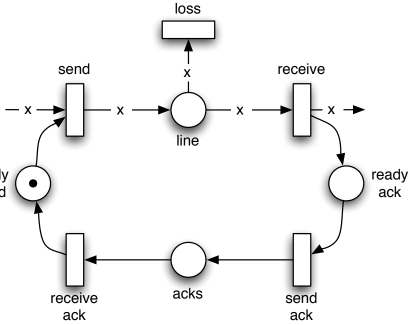
# Alternating Bit Protocol

- É um protocolo que garante a comunicação fiável sobre linhas de comunicação não fiáveis.
- Garante a entrega das mensagens, mesmo que para tal seja necessário repetir o envio.
- O canal pode ser modelado da seguinte forma.



# Introdução de Acks

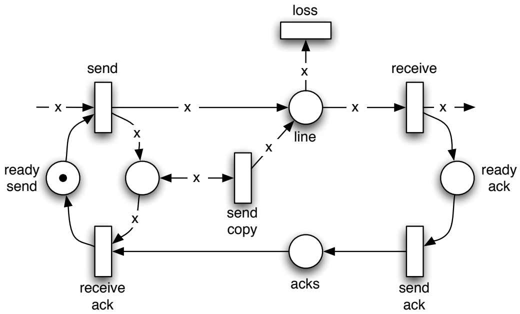
- Assumindo a existência de uma linha fiável entre o receptor e o emissor, é possível enviar mensagens de *acknowledgement*.




- Mas a perca de uma mensagem origina um *deadlock*. . .

# Repetição de Mensagens

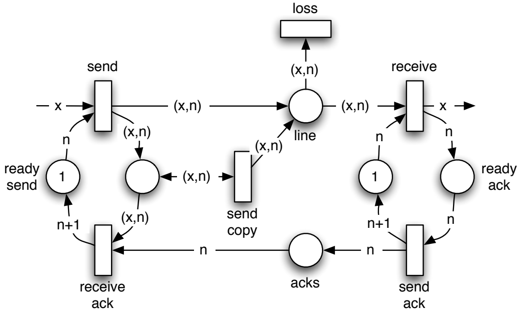
- Para evitar este *deadlock* o emissor pode repetir o envio de uma mensagem.




- Mas o receptor não consegue distinguir mensagens novas de repetições de mensagens antigas. . .

# Identificadores Únicos

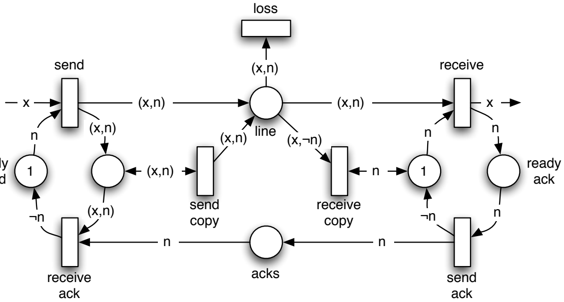
- É possível identificar cada mensagem univocamente.
- O receptor só aceita uma mensagem por identificador e envia um *ack* para cada mensagem recebida.




- Protocolo fiável, mas necessita de um número ilimitado de identificadores e não há *garbage collection* de mensagens redundantes. . .

# Alternating Bit Protocol

- De facto um bit chega para identificar a mensagem se a linha garantir a entrega por ordem.
- O *garbage collection* pode ser feito com uma acção que recebe e descarta uma cópia.



# Alternating Bit Protocol