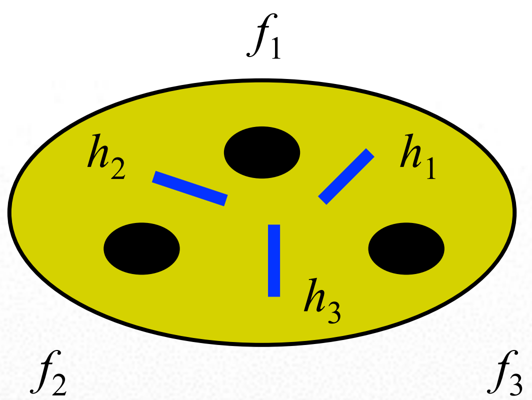
PMR 5237
Modelagem e Design de Sistemas Discretos em Redes de Petri

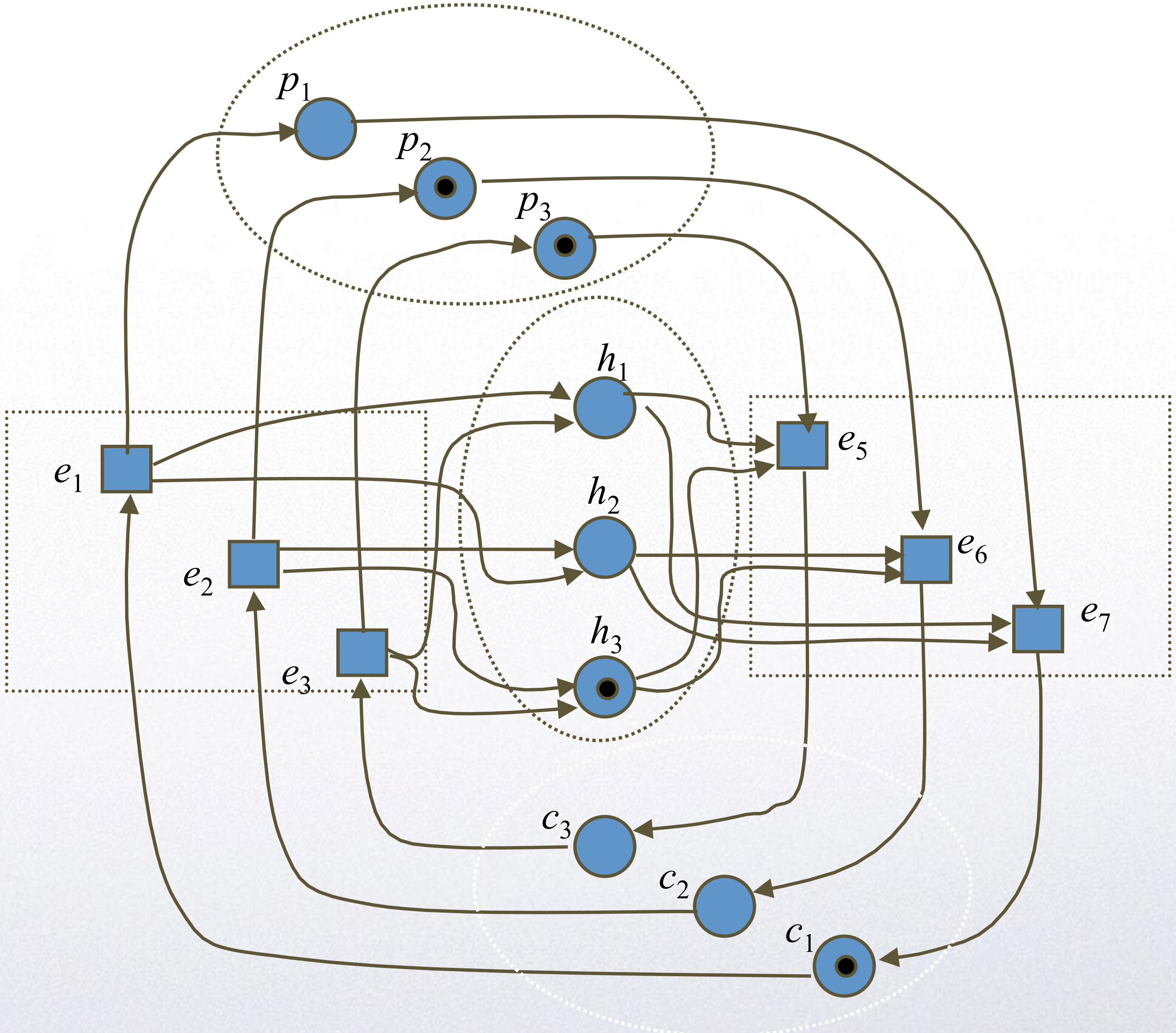
Aula 7: Formalização das Redes de Alto Nível

Prof. José Reinaldo Silva
reinaldo@usp.br

Dobramento em RdP

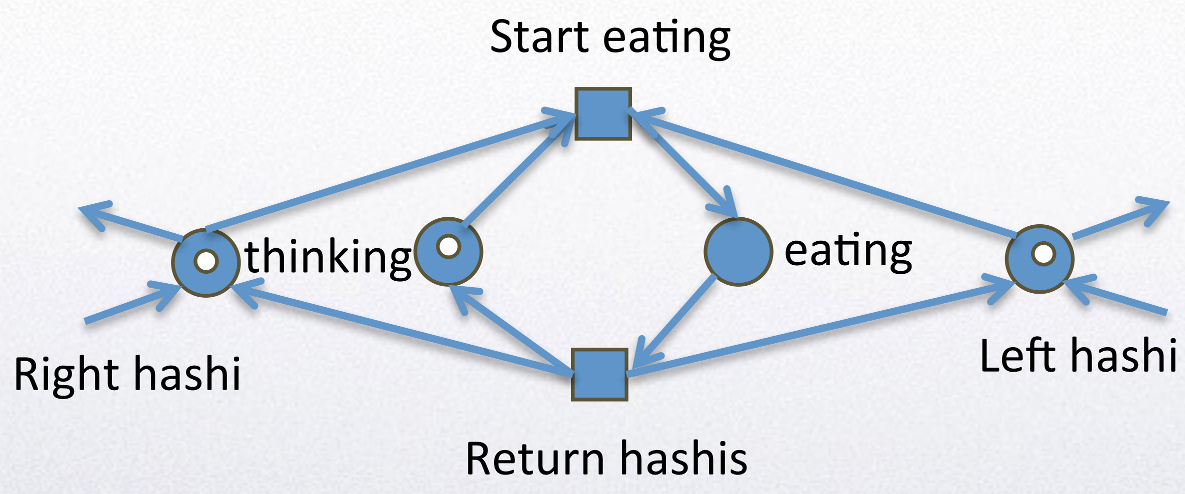
O exemplo dos filósofos





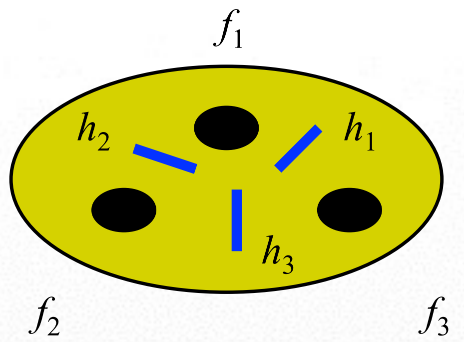
Modelagem e simetria

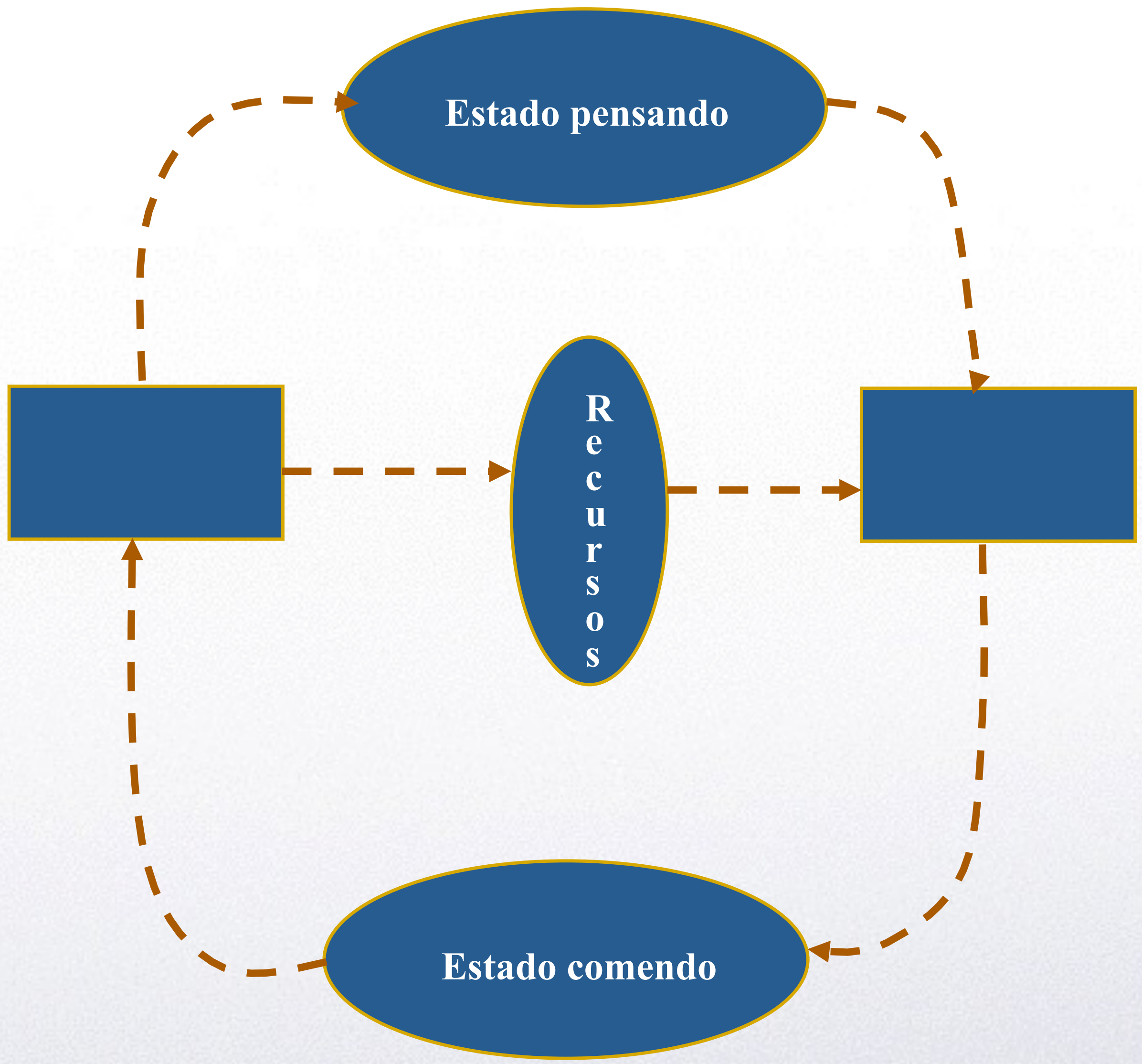
Certamente, uma forma de olhar o problema é procurar, logo de início, o relacionamento entre TODOS os seus elementos constituintes, como feito no slide anterior. Mas é possível também olhar cada um dos elementos composicionais, especialmente aqueles que apresentam propriedades reptetitivas. No exemplo dos filósofos, se olharmos cada um dos filósofos, temos:



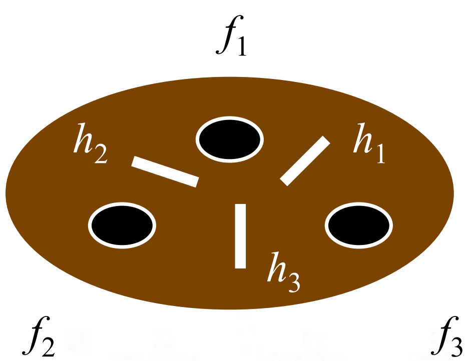
O dobramento

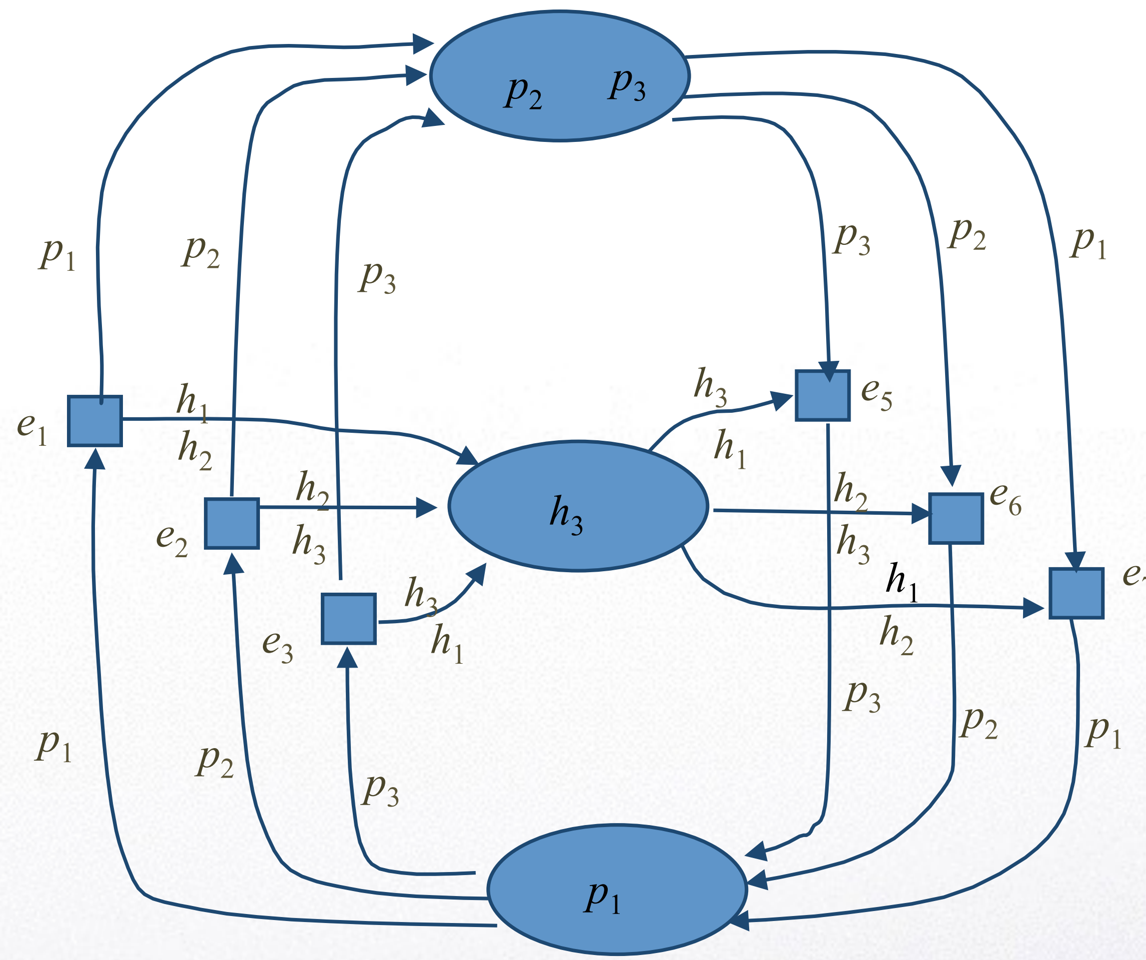
O exemplo dos filósofos





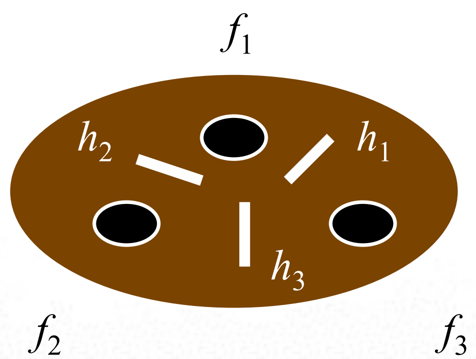
O exemplo dos filósofos





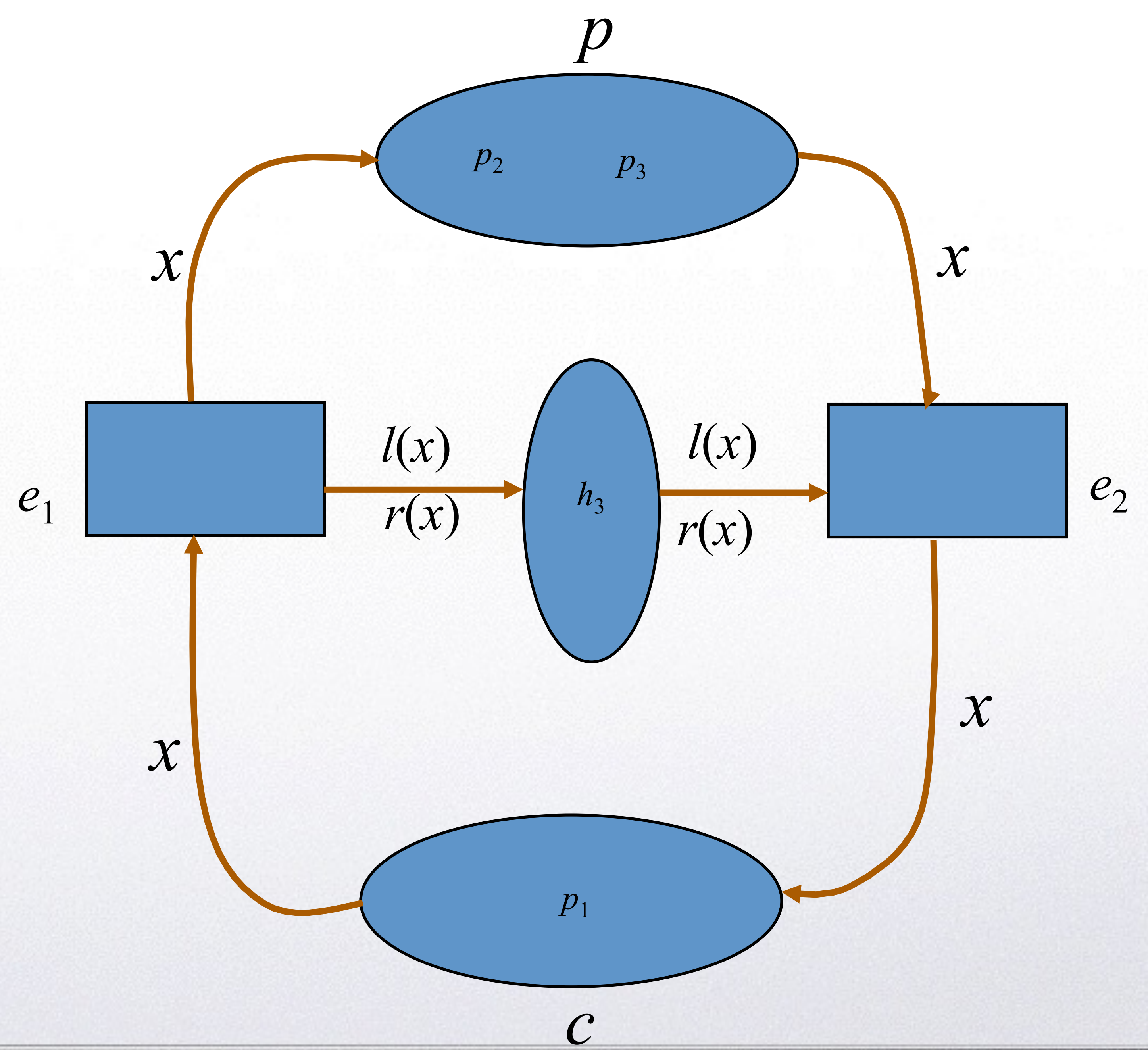
O colapso dos lugares leva à necessidade de se distinguir as marcas, tanto as que representam os filósofos quanto as que representam os recursos, isto é, os hashis.

O dobramento completo

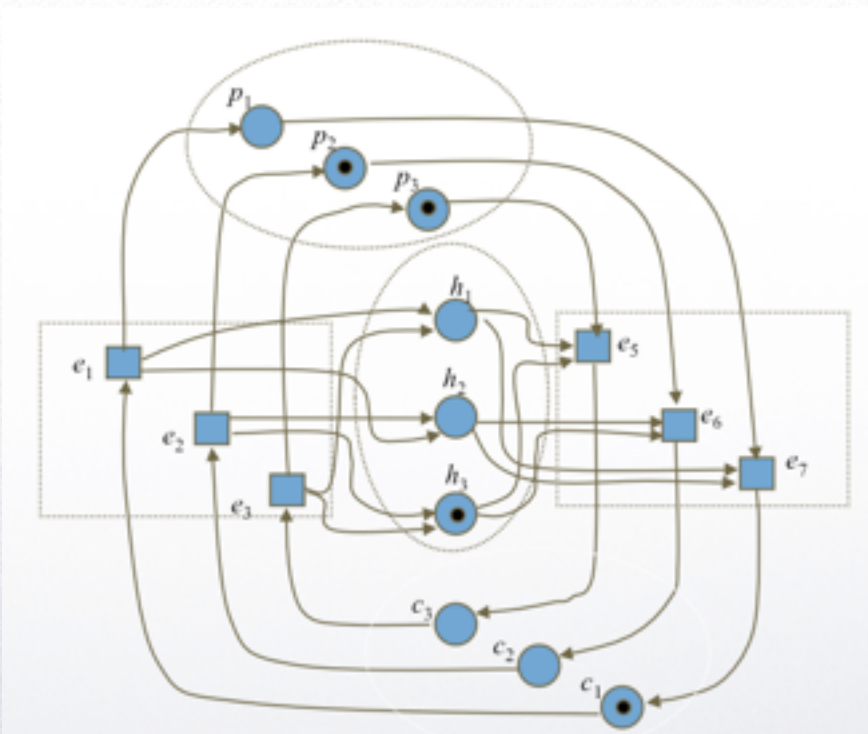


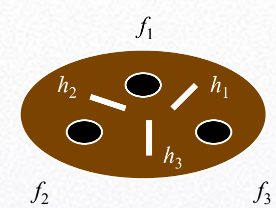
P={p_1, p_2, p_3}
H={h_1, h_2, h_3}
U=P ∪ H

$l : \mathrm{P} \rightarrow \mathrm{H}$
$p_i \rightarrow h_i$
$r : \mathrm{P} \rightarrow \mathrm{H}$
$p_1 \rightarrow h_2$
$p_2 \rightarrow h_3$
$p_3 \rightarrow h_1$



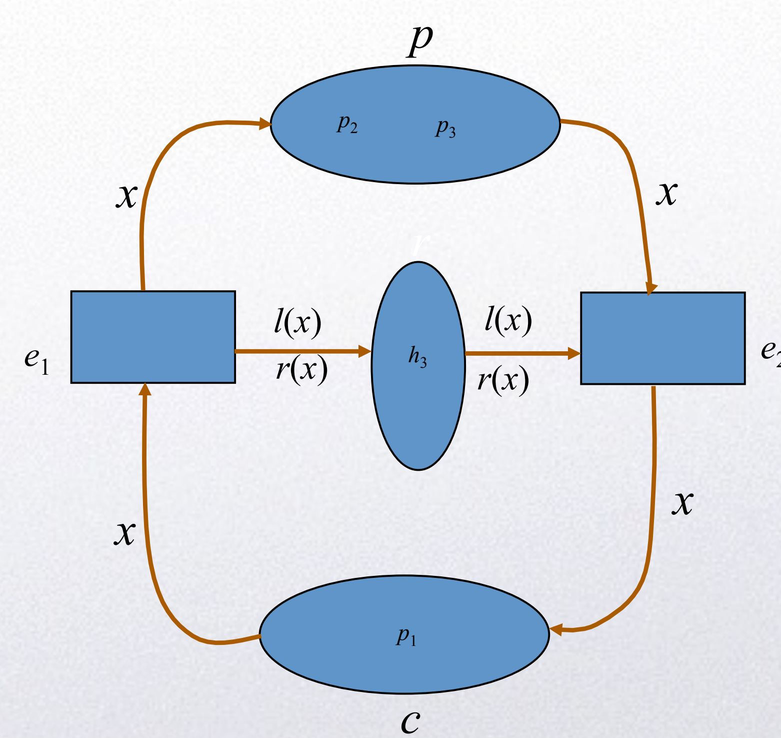
O dobramento leva de fato a uma rede mais compacta com um número menos de lugares e de transições. Entretanto, lembre que esta nova rede NÃO PODE ser separada das inscrições e tipos que a acompanham.





$P=\{p_1, p_2, p_3\}$
$H=\{h_1, h_2, h_3\}$
$U=P \cup H$

$l : P \rightarrow H$
$p_i \rightarrow h_i$
$r : P \rightarrow H$
$p_1 \rightarrow h_2$
$p_2 \rightarrow h_3$
$p_3 \rightarrow h_1$



Fatoração

A este processo de exploração das simetrias de uma rede para produzir uma nova rede dinamicamente equivalente à anterior mas de tamanho menor dá-se o nome de fatoração.

A rede resultante da fatoração é **chamada** de rede-quociente.

Multisets

Seja o conjunto (base set) {a,b,c,d,e,f,g,h}.

Em teoria de conjuntos, $\{a, c, f\} \cup \{c, f, g\} = \{a, c, f, g\}$.

Já em multisets (bags),

$$\{a, c, f\} \cup \{c, f, g\} = \{a, c, c, f, f, g\}$$

também descrito como 1\`a + 2\`c + 2\`f + 1\`g

Um multiset é um par (X, f), onde X é um conjunto (finito), e f é um mapeamento do conjunto X no conjunto dos números naturais positivos $\mathbb{N}^+$, onde X é chamado conjunto de suporte (ou conjunto base) do multiset e para todo $x \in X, f(x)$ é a multiplicidade de X.

Dado o seguinte conjunto:

f(d)=2

1 2 3 4 5 6 7
a b c d e f g

f(g)= 3

O multiset

a b c d d e f f g g g pode ser escrito como 1'a, 1'b, 1'c, 2'd, 1'e, 2'f, 3'g

Como vimos na aula passada podemos avançar bastante nas redes de alto nível mas com a aparente desvantagem de perder um pouco a intuição no processo de modelagem. Ademais, o problema do cálculo das propriedades não está claro até aqui, dado que não temos mais uma matriz de incidência composta de inteiros.

Portanto o nosso objetivo passa a ser, por enquanto, tentar compor uma rede de alto nível, baseada no dobramento das redes (seja durante a modelagem ou depois de sintetizada uma rede P/T), e nas simetrias, porém simples, e que, acima de tudo preserve as propriedades descritas na transparências anterior, e além disso, que garanta um algoritmo de transformação direto para as redes clássicas e de volta, das redes clássicas para uma rede de alto nível.

Referências

Seguiremos daqui em diante duas referências básicas: o artigo do Einar Smith, pertencente ao mesmo LNCS que contém o último curso de Redes de Petri dado pelos maiores pesquisadores da área, e a própria definição da rede de alto nível, segundo o padrão ISO/IEC 15.909-1 (ambos colocados no Moodle).

Smith, E.; Principles of High Level Nets, LNCS 1491, pp. 174-210.

ISO/IEC 15.909, Final Draft, v.4.7.1; High Level Nets: Concepts, Definitions and Graphical Notations, October, 2000.

Aplicações

- requirements analysis;
- development of specifications, designs and test suites;
- descriptions of existing systems prior to re-engineering;
- modelling business and software processes;
- providing the semantics for concurrent languages;
- simulation of systems to increase confidence;
- formal analysis of the behaviour of critical systems; and
- development of Petri net support tools.

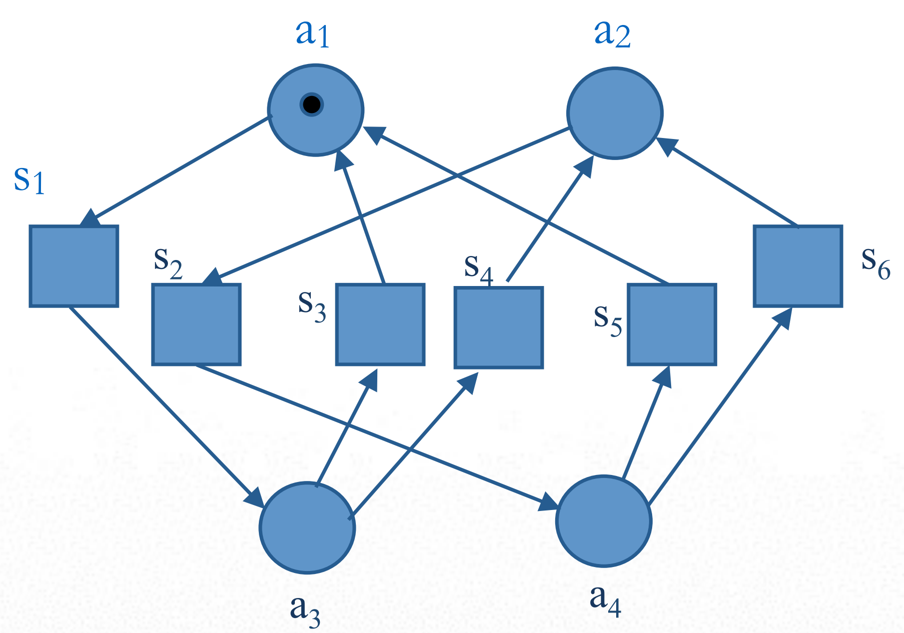
ISO/IEC 15.909, Final Draft, v.4.7.1; High Level Nets: Concepts, Definitions and Graphical Notations, October, 2000.

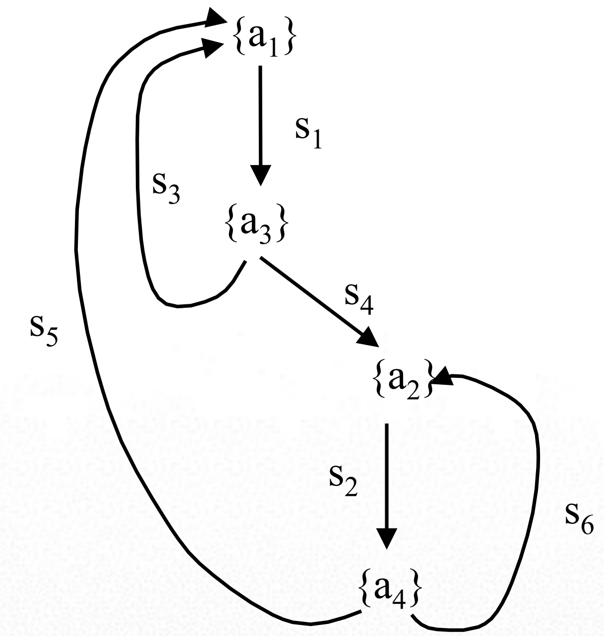
Fatoração: Novos desafios

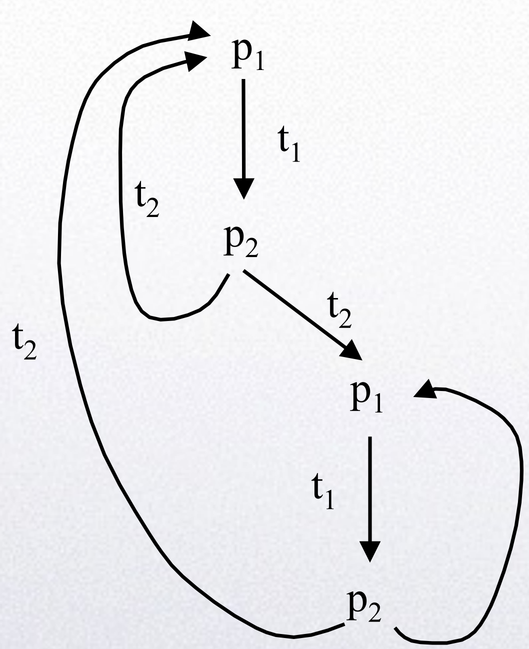
A este processo de exploração das simetrias de uma rede para produzir uma nova rede dinamicamente equivalente à anterior mas de tamanho menor dá-se o nome de fatoração.

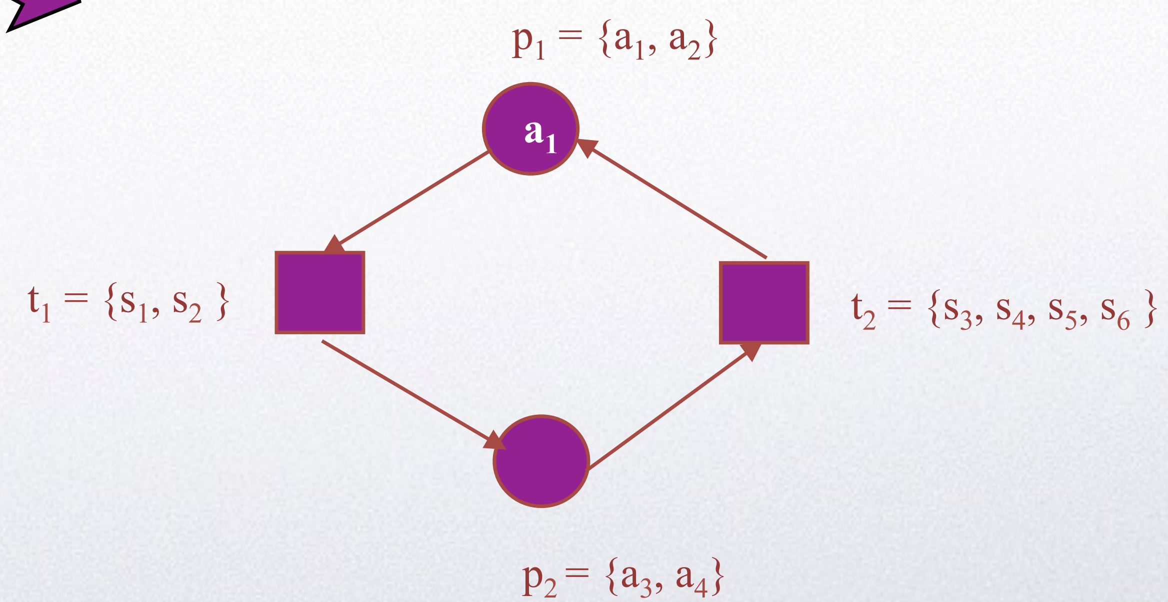
A rede resultante da fatoração é **chamada** de rede-quociente.

$$\text{Rede P/T} \underset{\text{Unfolding}}{\overset{\text{Fatoração}}{\rightleftarrows}} \text{Rede Quociente}$$


a_1
a_2
s_1
s_2
s_3
s_4
s_5
s_6
a_3
a_4


$\{a_1\}$
s_1
s_3
$\{a_3\}$
s_4
s_5
$\{a_2\}$
s_2
s_6
$\{a_4\}$


p_1
t_1
t_2
p_2
t_2
t_2
p_1
t_1
p_2


$p_1 = \{a_1, a_2\}$
a_1
$t_1 = \{s_1, s_2\}$
$t_2 = \{s_3, s_4, s_5, s_6\}$
$p_2 = \{a_3, a_4\}$

Formalmente,

Definition 34

Seja uma rede place/transition N e seja o seu domínio $X = P \cup T$. Existe uma equivalência ρ entre a rede N e sua rede quociente $\bar{N}$ tal que:

i) $\forall x \in X, \exists \bar{x}$ que denota uma classe de equivalênica $\{y \in X | x \rho y\}$.

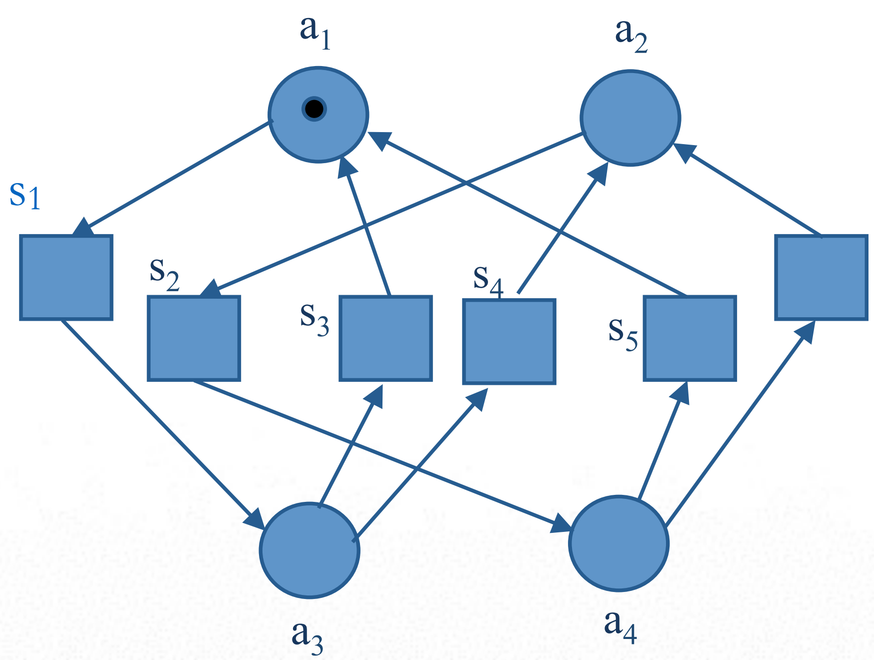
ii) Seja $Y \subseteq X$, então $\bar{Y} := \{\bar{x} | x \in Y\}$,

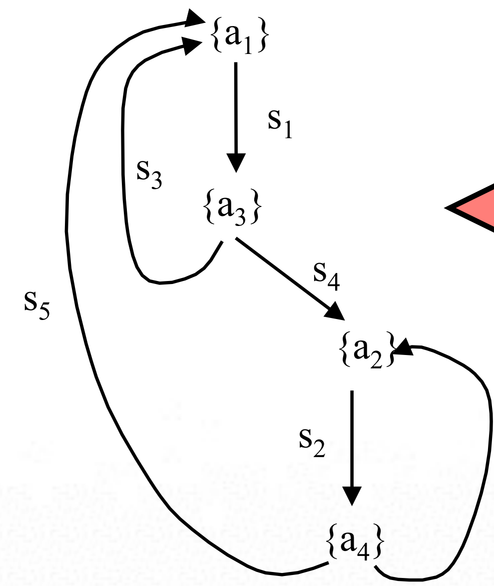
iii) ρ preserva o sort, isto é, $\rho \cap (P \times T) = \emptyset$,

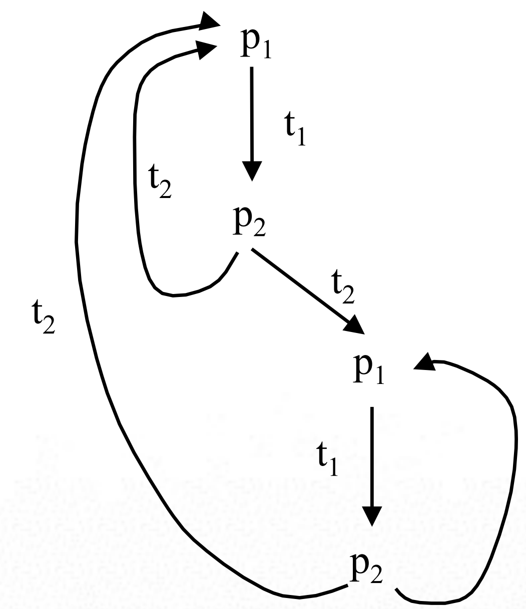
iv) A relação de fluxo $\bar{F}$ sobre o dominio $\bar{X}$ é definida por,

$$\bar{x} F \bar{y} \iff \exists x\prime \in \bar{x} \wedge \exists y\prime \in \bar{y} | x F y$$

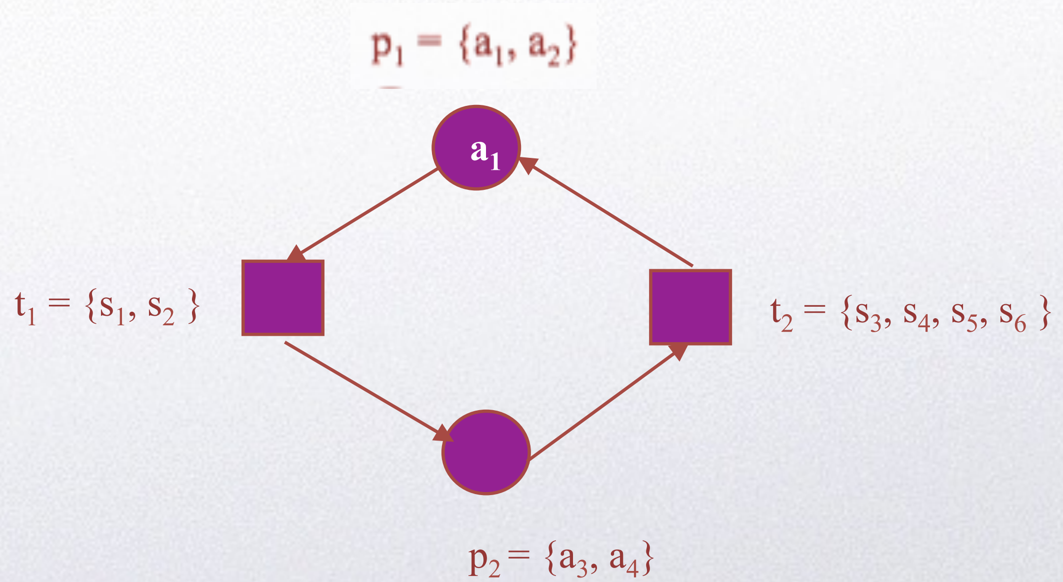
v) Denota-se a nova rede $(\bar{P}, \bar{T}; \bar{F})$ de $\bar{N}$.

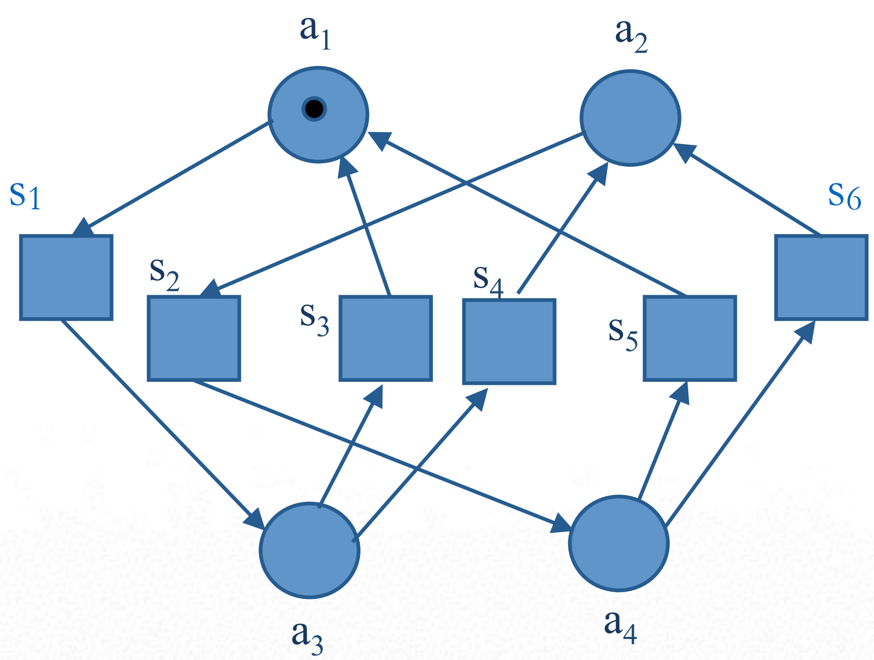


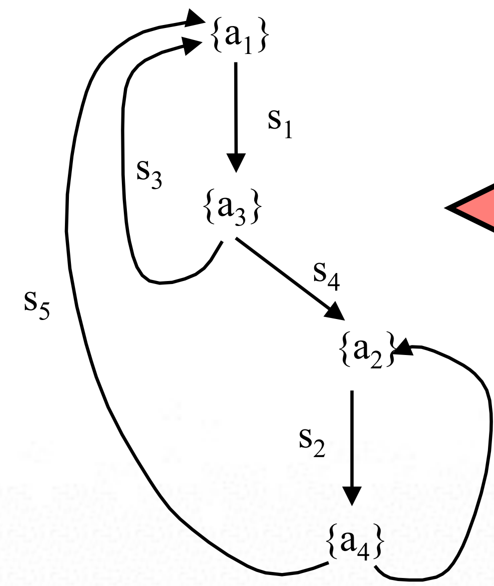


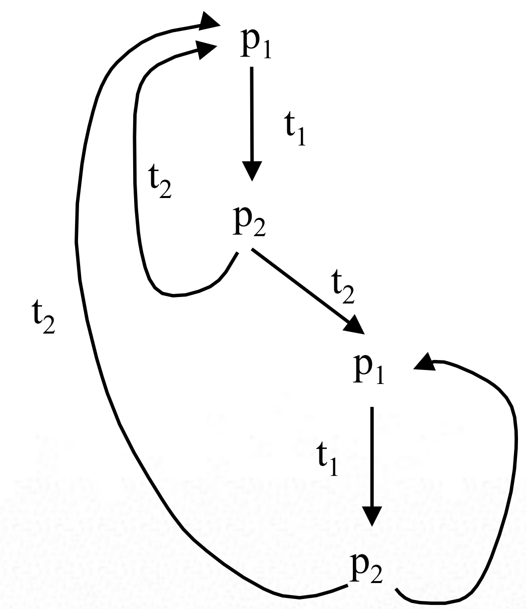


$$\rho:\ \bar{x}_1 = \{a_1, a_2\} \rightarrow p_1$$
$$\bar{x}_2 = \{a_3, a_4\} \rightarrow p_2$$
$$\bar{x}_3 = \{s_1, s_2\} \rightarrow t_1$$
$$\bar{x}_4 = \{s_3, s_4, s_5, s_6\} \rightarrow t_2$$









$$\rho: \ \bar{x}_1 = \{a_1, a_2\} \rightarrow p_1$$

$$\bar{x}_2 = \{a_3, a_4\} \rightarrow p_2$$

$$\bar{x}_3 = \{s_1, s_2\} \rightarrow t_1$$

$$\bar{x}_4 = \{s_3, s_4, s_5, s_6\} \rightarrow t_2$$

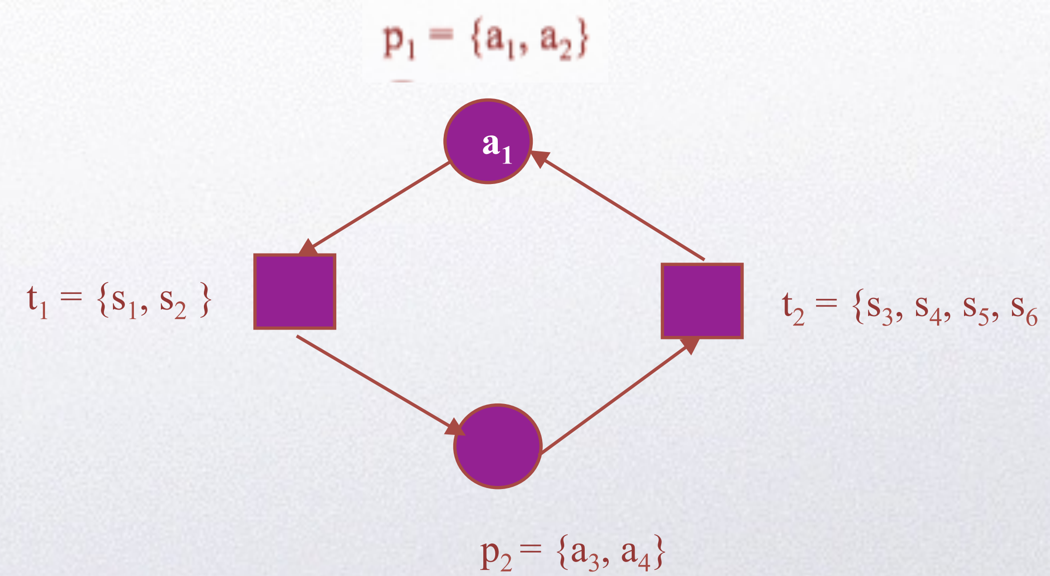
$$\bar{x} = \{x_i\}$$

$$\rho: \mathbf{M}(\bar{x}) = \mathbf{M}\{x_i\}$$

$$\mathbf{M}(p_1) = \mu\mathbf{M}\{a_1, a_2\} = 1a_1$$

$$\mathbf{M}(p_2) = \mu\mathbf{M}\{a_3, a_4\} = 0$$

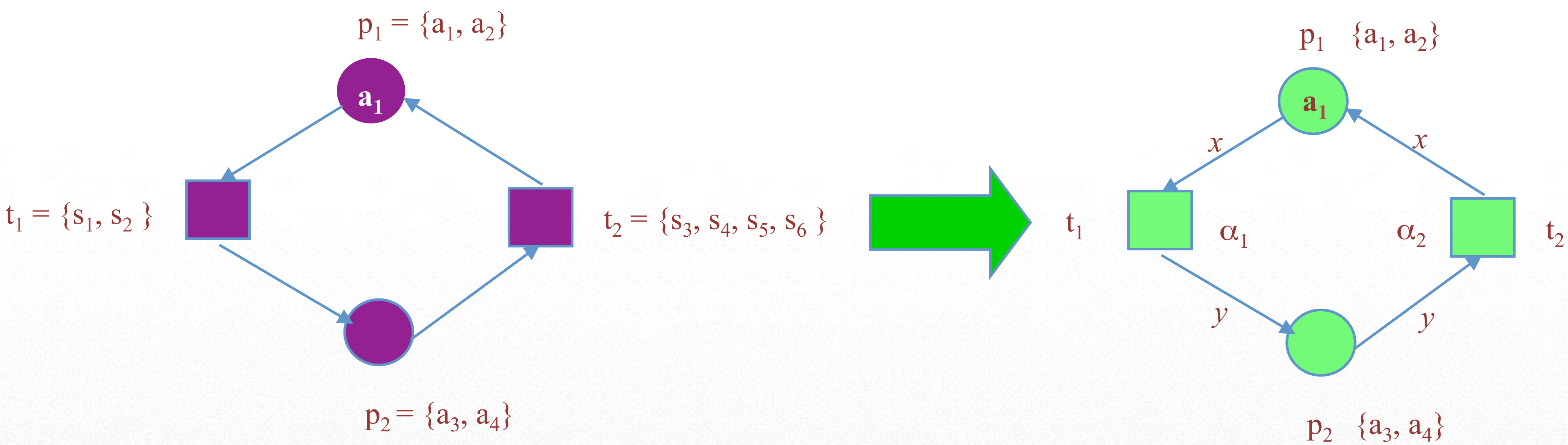
$$\overline{W}(p_1, t_1) = a_1 \vee a_2 = \overline{W}(t_1, p_2) = \overline{W}(p_2, t_2) = \overline{W}(t_2, p_1)$$



Definition 35

Seja uma rede P/T com estrutura N, $PT = (N, K, W, M_0)$. e uma bijeção (equivalência) ρ que preserva o sort. Chama-se rede quociente em relação a ρ ao sistema $\bar{PT} = (\bar{N}, \bar{K}, \bar{W}, \bar{M_0})$ que tem a mesma dinâmica que a rede original.

Basic High Level Net



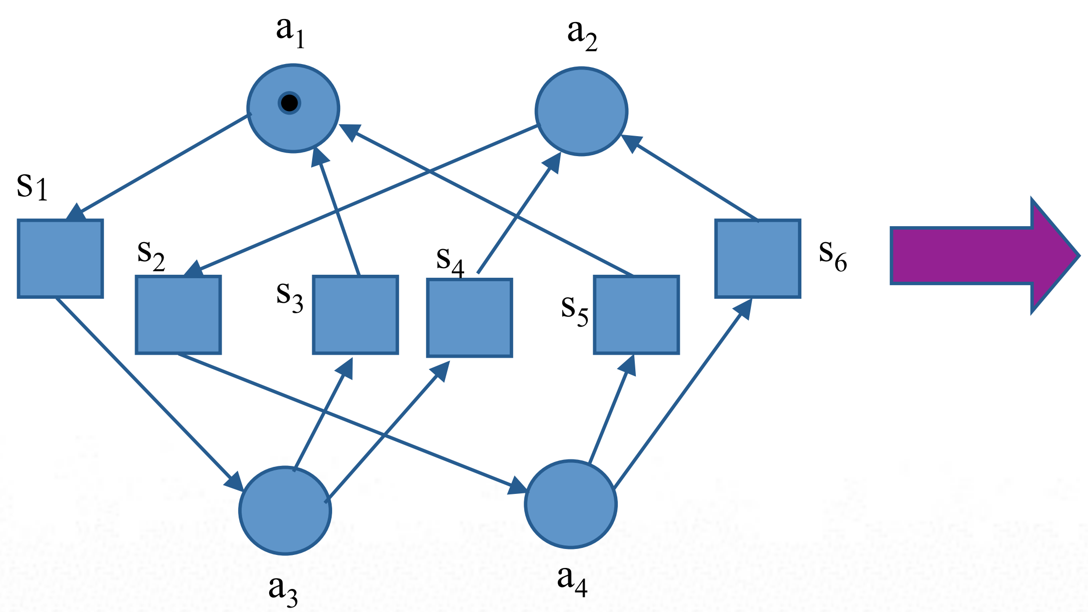
- lugares possuem marcas individualizadas
- transições ocorrem em diferentes modos restritos à regra de localidade (agir somente sobre o seu pre-set e pós-set)

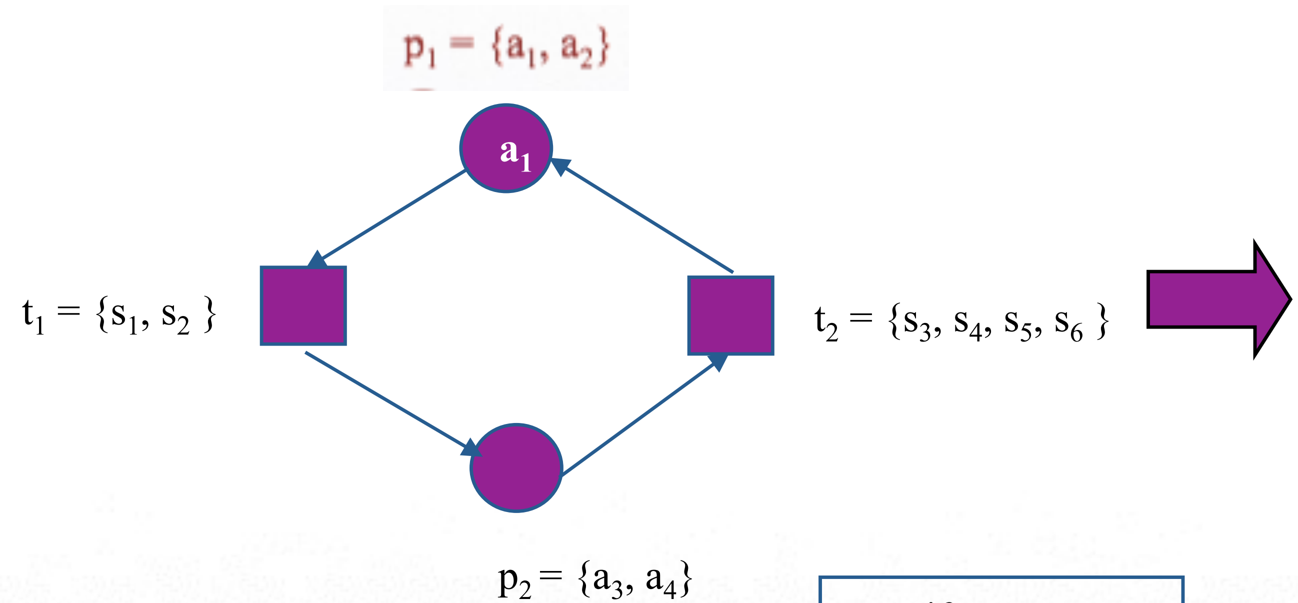
α_1 = if $x = a_1$
then $y = a_3$
else if $x = a_2$
then $y = a_4$

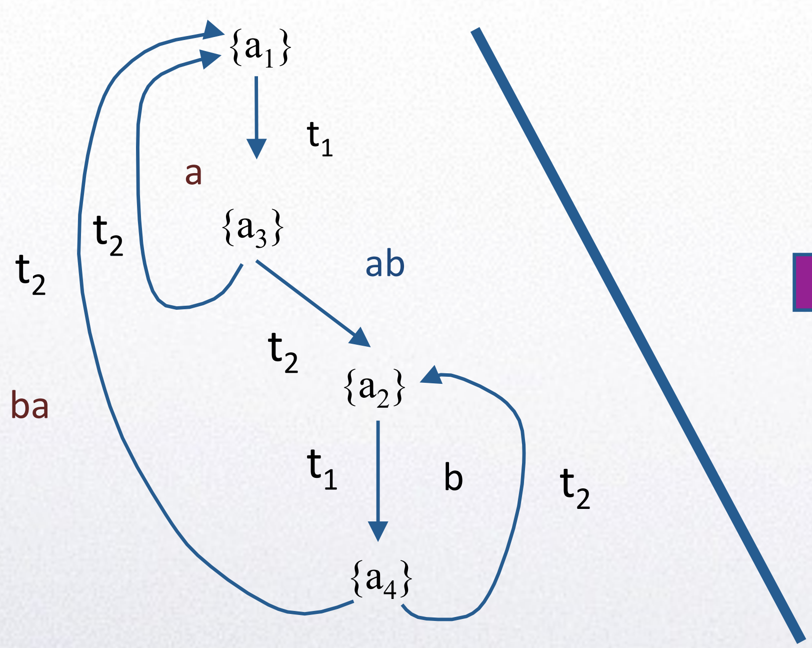
α_2 = if $y = a_3$
then $(x = a_1) \vee (x = a_2)$
else if $y = a_4$
then $(x = a_1) \vee (x = a_2)$

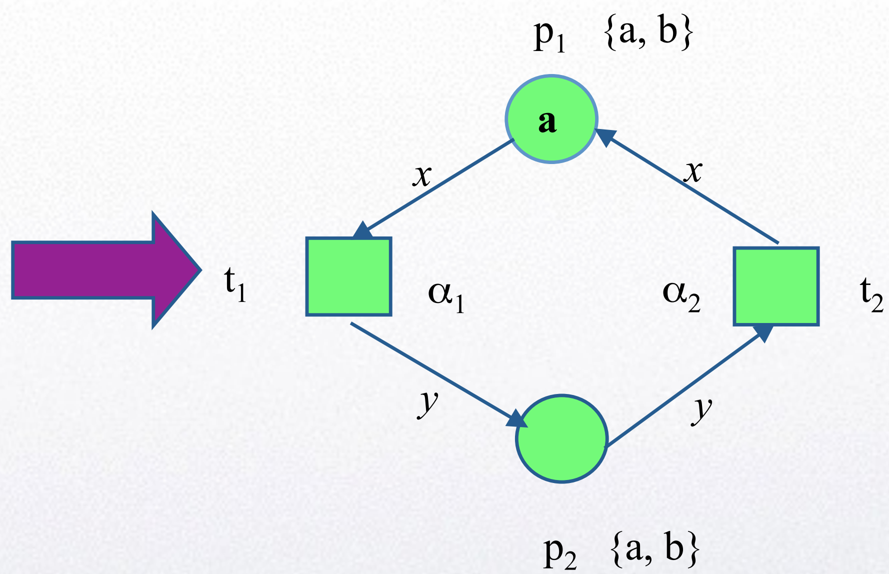
$P = \{a_1, a_2, a_3, a_4\}$

Smith, E.; Principles of High Level Nets, LNCS, 1491, Springer Verlag, 1998.









α_1 = if x= a
then y = a
else if x= b
then y =b

α_2 = if y= a
then $(x = a) \vee (x = b)$
else if y= b
then $(x = a) \vee (x = b)$

$P = \{a, b_2\}$

ISO/IEC 15.909

Def.36] Uma rede de Petri de alto nível, HLPN é uma estrutura dada pela n-upla HLPN=$(P,T,D;Time,Pre,Post,M_0)$ onde:

- P é um conjunto finito de elementos chamados lugares;
- T é um conjunto finito de elementos chamados transições;
- D é um conjunto finito, não-vazio, de domínios ou tipos;
- Pre, Post: TRANS $\rightarrow$ μPLACE, onde

$$\text{TRANS}=\{(t,m)\,|\,t \in T, m \in Type(t)\}$$

$$\text{PLACE}=\{(p,g)\,|\,p \in P, g \in Type(p)\}$$

- $M_0 \in \mu PLACE$ é o multiset que denota a marcação inicial da rede.

Uma transição $t \in T$ é dita habilitada em uma marcação M, se e somente se

$$\mathrm{Pre}(t) \leq M$$

(onde a desigualdade é entre multisets).

Relação de Ordem parcial : Dados dois multisets, representados pelos respectivos vetores de coeficientes sobre um conjunto de base (base set) S,

m= $(m_1, m_2, \ldots, m_i, \ldots, m_s)$,
n= $(n_1, n_2, \ldots, n_j, \ldots, n_s)$

Está definida a comparação entre estes dois multisets

m ≥ n se e somente se $m_i \geq n_i$ para todo i

Generalizando o conceito de transição temos que um passo T_μ é um multiset sobre os modos de transição. Um tal passo está habilitado em uma marcação M se e somente se

$$\mathrm{Pr}\,e(T_\mu) \leq M$$

onde,

$$\mathrm{Pre}(T_\mu) = \sum_{t \in T_\mu} T_\mu(t)\,\mathrm{Pr}\,e(t)$$

Um passo T_μ habilitado em uma marcação M ocorre, resultando em outra marcação M', obtida a partir da marcação original pela equação,

$$M' = M - Pre(T_\mu) + Post(T_\mu)$$

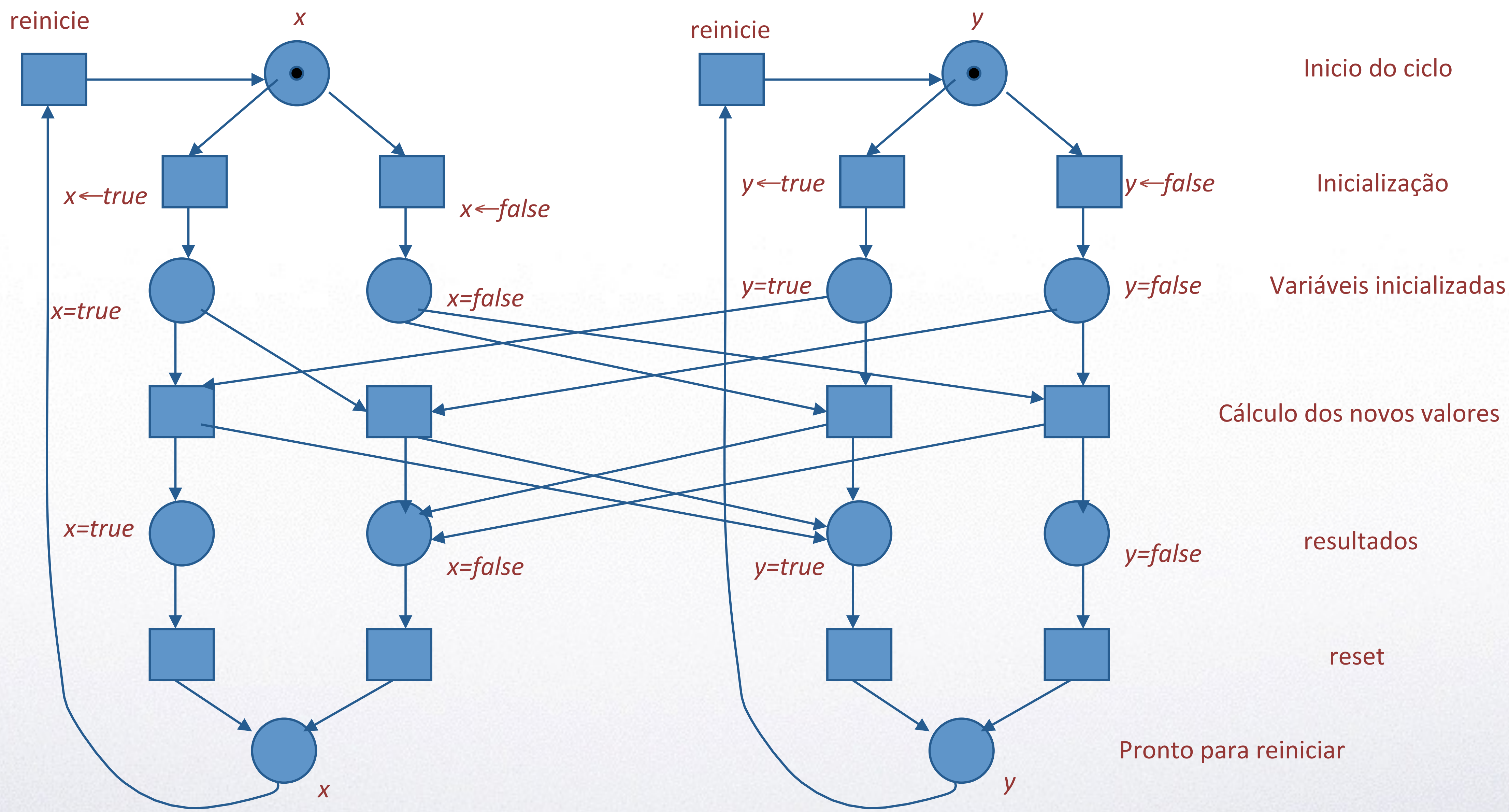
A High-level Petri Net Graph (HLPNG) comprises:

- *A Net Graph*, consisting of sets of nodes of two different kinds, called *places* and *transitions*, and *arcs* connecting places to transitions, and transitions to places.
- *Place Types*. These are non-empty sets. One type is associated with each place.
- *Place Marking*. A collection of elements (data items) chosen from the place's type and associated with the place. Repetition of items is allowed. The items associated with places are called *tokens*.
- *Arc Annotations*: Arcs are inscribed with expressions which may comprise constants, variables (e.g., x, y) and function images (e.g., $f(x)$). The variables are typed. The expressions are evaluated by assigning values to each of the variables. When an arc's expression is evaluated, it must result in a collection of items taken from the type of the arc's place. The collection may have repetitions.
- *Transition Condition*: A boolean expression (e.g., $x < y$) inscribing a transition.
- *Declarations*: comprising definitions of place types, typing of variables, and function definitions.

Um exemplo

Tomemos como exemplo uma máquina (de fato um problema de circuitos lógicos) onde duas entradas, representadas pelas variáveis, x e y, podem assumir valores lógicos em {*true*, *false*}. A máquina opera de tal modo que sobre a variável x se subscreve $x \wedge y$ enquanto que sobre o valor de y se se subscreve $x \vee y$. O conteúdo das variáveis é então eliminado e o sistema fica preparado para receber novos valores.

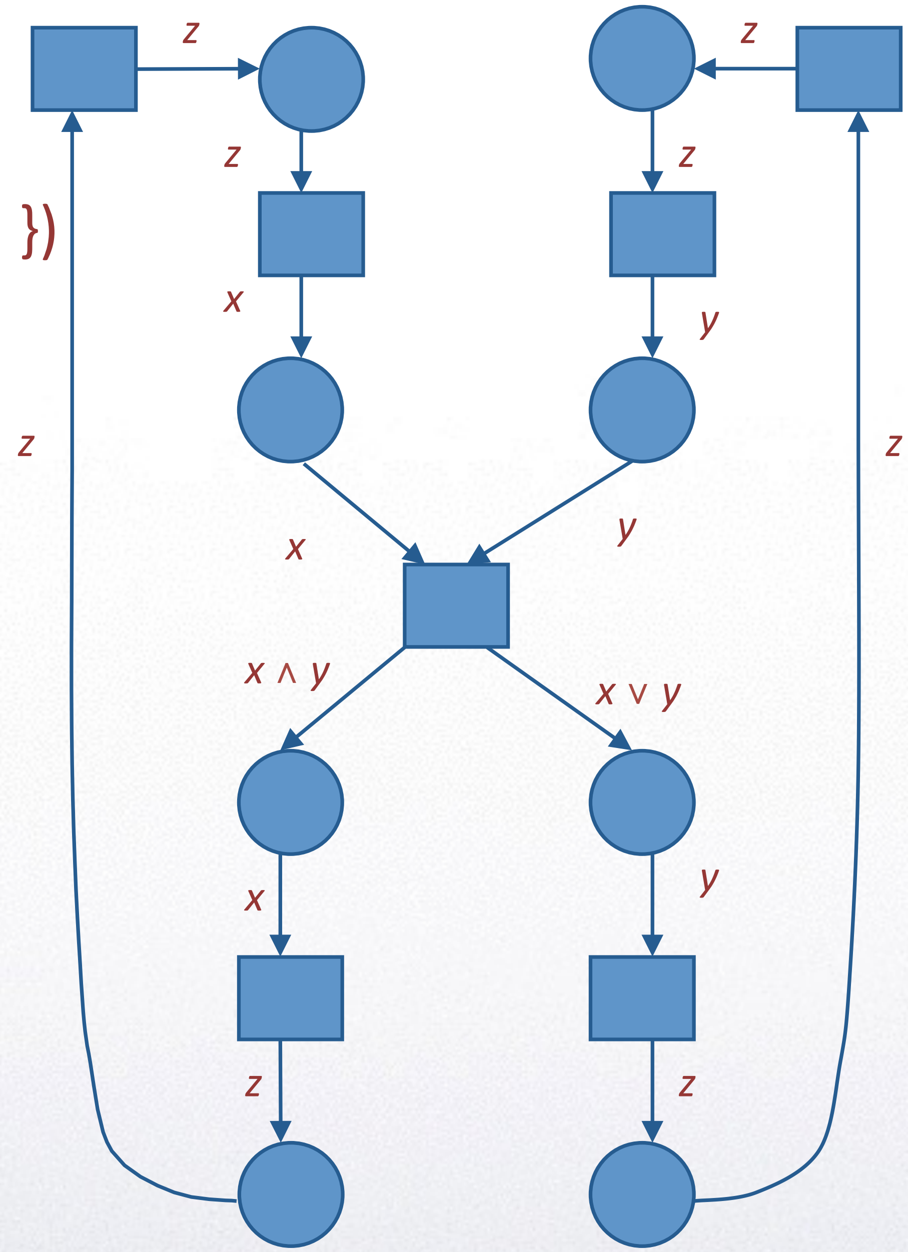
A rede de Petri representando a operação desta máquina é mostrada a seguir.


reinicie
x
reinicie
y
Inicio do ciclo
x←true
x←false
y←true
y←false
Inicialização
x=true
x=false
y=true
y=false
Variáveis inicializadas
Cálculo dos novos valores
x=true
x=false
y=true
y=false
resultados
reset
x
y
Pronto para reiniciar

$D=\{true, false\}$
$\phi=\{ \wedge, \vee \}$
$\delta=(D;\phi)=(\{true,false\}, \{ \wedge, \vee \})$

if x=true and y=true
then $x \wedge y = true$
else $x \wedge y =$ false

if x=false and y=false
then $x \vee y = false$
else $x \vee y =$ true



Inicio do ciclo

Inicialização

Variáveis inicializadas

Cálculo dos novos valores

resultados

reset

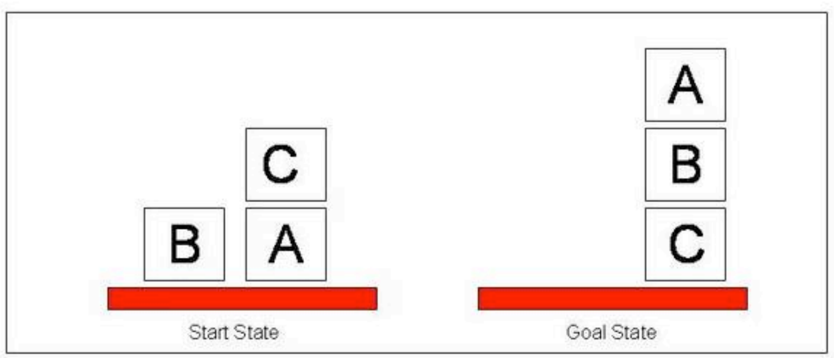
Pronto para reiniciar

Exercício

Até aqui as redes de alto nível HLPN foram apresentadas como o resultado do processo de fatoração (folding) de uma rede clássica P/T. Assim as redes HLPN têm sempre uma rede base P/T e fica sub-entendido que o processo clássico de modelagem seria sintetizar uma rede P/T e depois fatorá-la para achar a rede quociente e daí a rede HLPN.

Vamos exercitar agora um processo diferente: colocar um problema onde parece bastante atraente tentar sintetizar uma rede HLPN ao invés de uma rede clássica.

Voltando ao mundo dos blocos



The "Sussman anomaly" blocks-world planning problem.

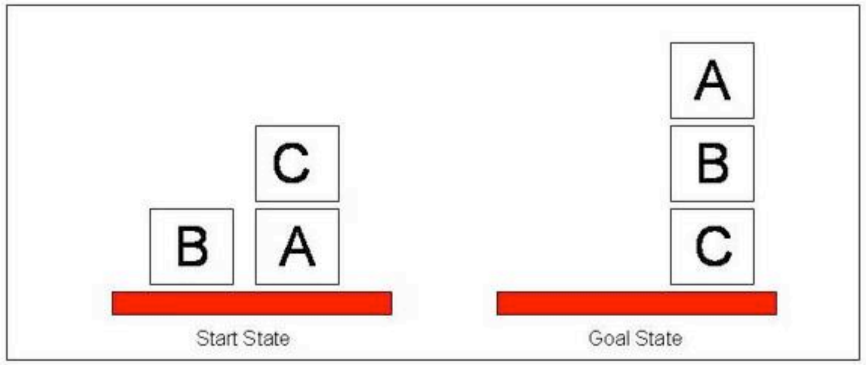
Imagine um mundo hipotético composto de blocos tridimensionais identificados por letras maiúsculas e um robô manipulador que só consegue pegar um bloco de cada vez. Outra regra importante é que este robô só pode pegar um bloco se este for o primeiro da pilha, isto é, não existe nenhum outro bloco sobre ele.
Com estas regras pretende-se fazer um plano de ações para que o robô transforme o estado inicial mostrado na figura no estado final.

Descrevendo os estados

Descritores dos estados: ONTABLE(x), ON(x,y), CLEAR(x), HANDEMPTY, HOLDING(x)

Estado inicial: ONTABLE(B), CLEAR(B), ONTABLE(A), ON(C,A), CLEAR(C), HANDEMPTY

Estado final: ONTABLE(C), ON(B,C), ON(A,B), CLEAR(A), HANDEMPTY



The "Sussman anomaly" blocks-world planning problem.

Descrevendo as ações

1. **Pickup(x)**
 (robô pega um bloco x da mesa)
Pré-condições: ONTABLE(x), CLEAR(x), HANDEMPTY
Pós-condição: HOLDING(x);
2. **Putdown(x)**
(robô deposita bloco x na mesa)
Pré-condição: HOLDING(x)
Pós-condição: ONTABLE(x), CLEAR(x), HANDEMPTY
3. **Stack(x, y)**
(robô empilha bloco x sobre o bloco y)
Pré-condição: HOLDING(x), CLEAR(y)
Pós-condição: ON(x,y), HANDEMPTY, CLEAR(x)
4. **Unstack(x, y)**
(robô tira bloco x de cima do bloco y)
Pré-condição: ON(x,y), CLEAR(x), HANDEMPTY
Pós-condição: CLEAR(y), HOLDING(x)

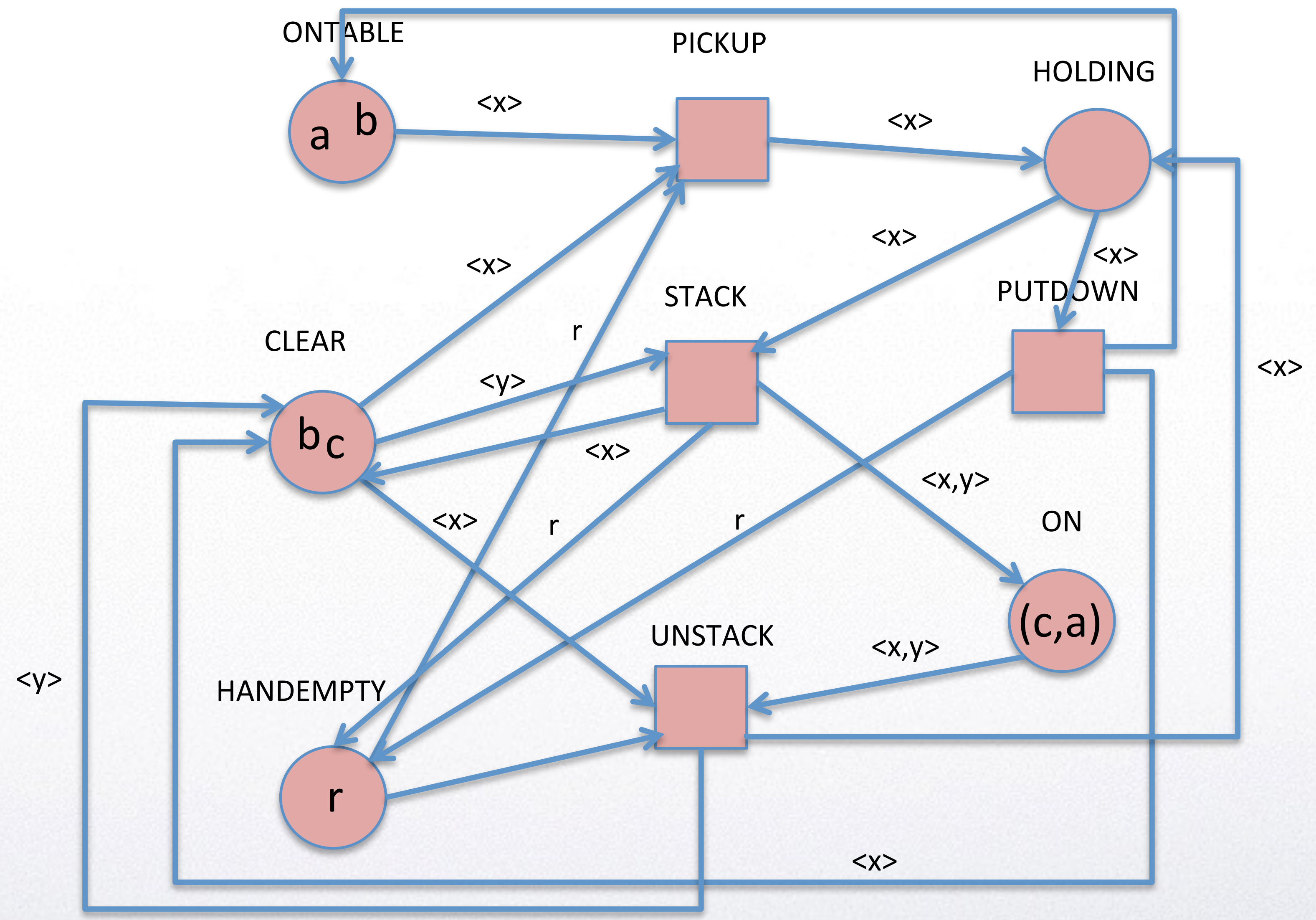
O problema (de planejamento)

Problema já está modelado e com uma representação já definida para especificação de estados, e uma série de ações que modificam estes estados. Estes estados atuam sobre um conjunto de blocos individualizados (distinguíveis). Portanto a modelagem deste tipo de problema é direta e deve ser feita com vantagem usando a HLPN (comparado a fazer a rede clássica primeiro).

A modelagem do problema fica assim bem simples.

Blk:={a,b,c}
R:= {r}
OnB: Blk X Blk

x,y: Blk



Você conseguria sintetizar a rede de alto nível (ou colorida) diretamente do enunciado do problema?

Tente fazer isso, mesmo já tendo visto a solução.

Fim