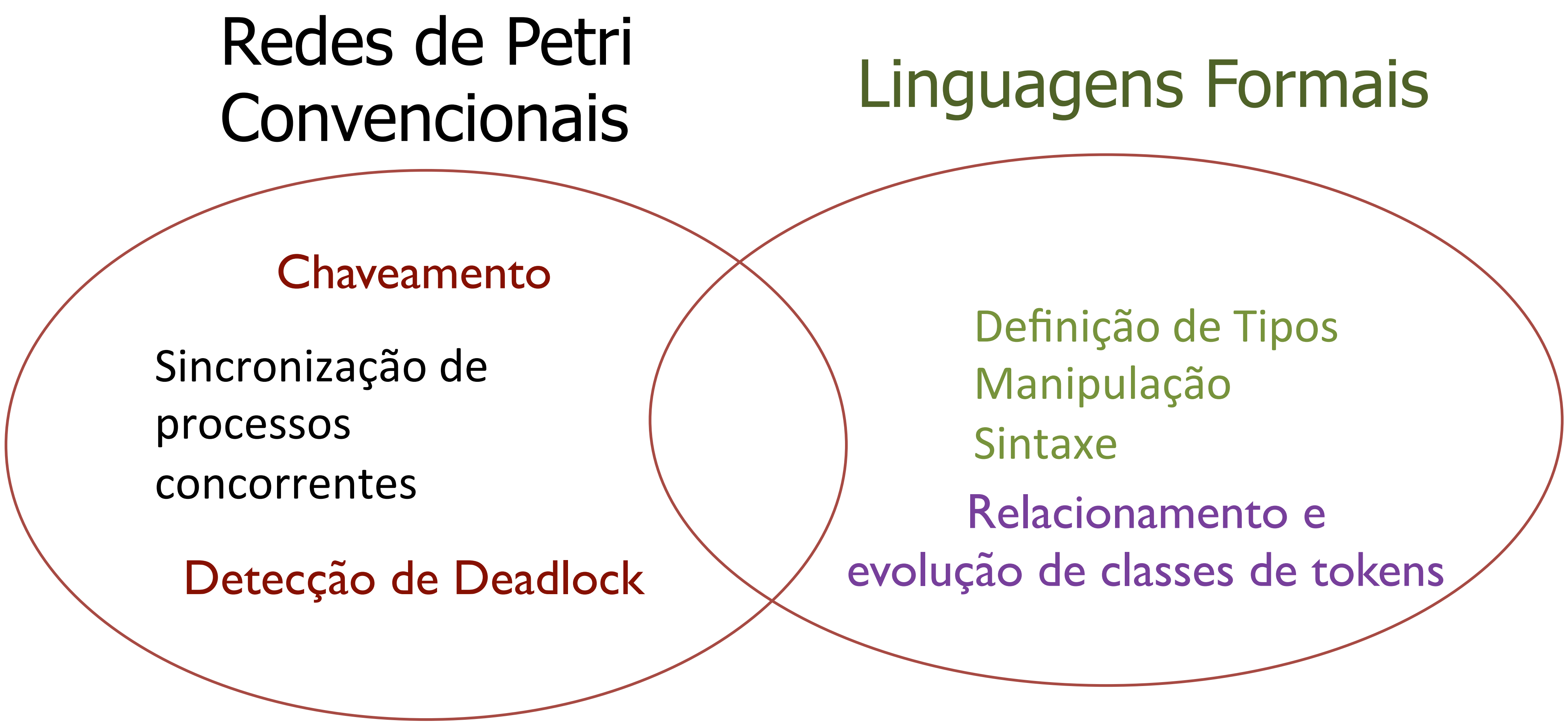
PMR 5237
Modelagem e Design de Sistemas Discretos em Redes de Petri

Aula 7: Formalização das Redes de Petri & Métodos de Análise

Prof. José Reinaldo Silva



Kurt Jensen, An Introduction to the Practical Use of Coloured Petri Nets, Lect. Notes in Comp. Science 1492, 1998.

Introdução informal: O problema da alocação de recursos

Na indústria automotiva moderna é comum se ter vários processos ou linhas de montagem e nestes um ou mais tipos de automóvel sendo montados em pipeline. Isso traz um problema, que é ter o tipo certo de insumo ou recurso no tempo correto, para a matriz correta.

Um problema similar e igualmente importante é ter processos homogêneos mas compartilhando o mesmo centro de recursos.

Processos especiais

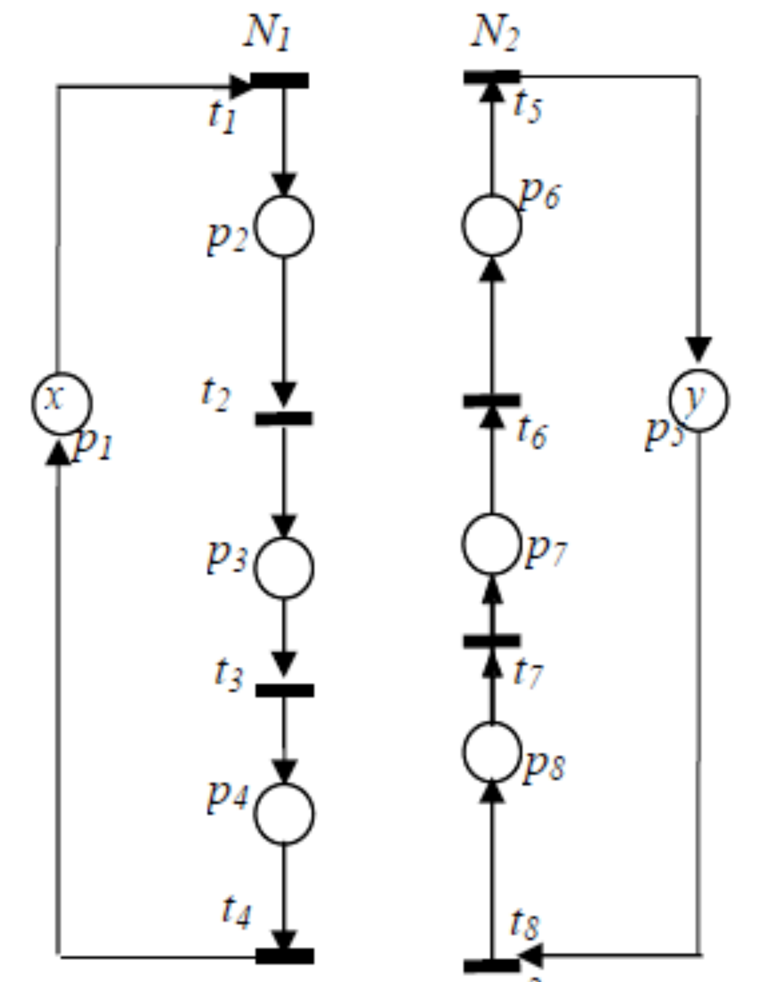


Fig. 1(b) Two S²P.

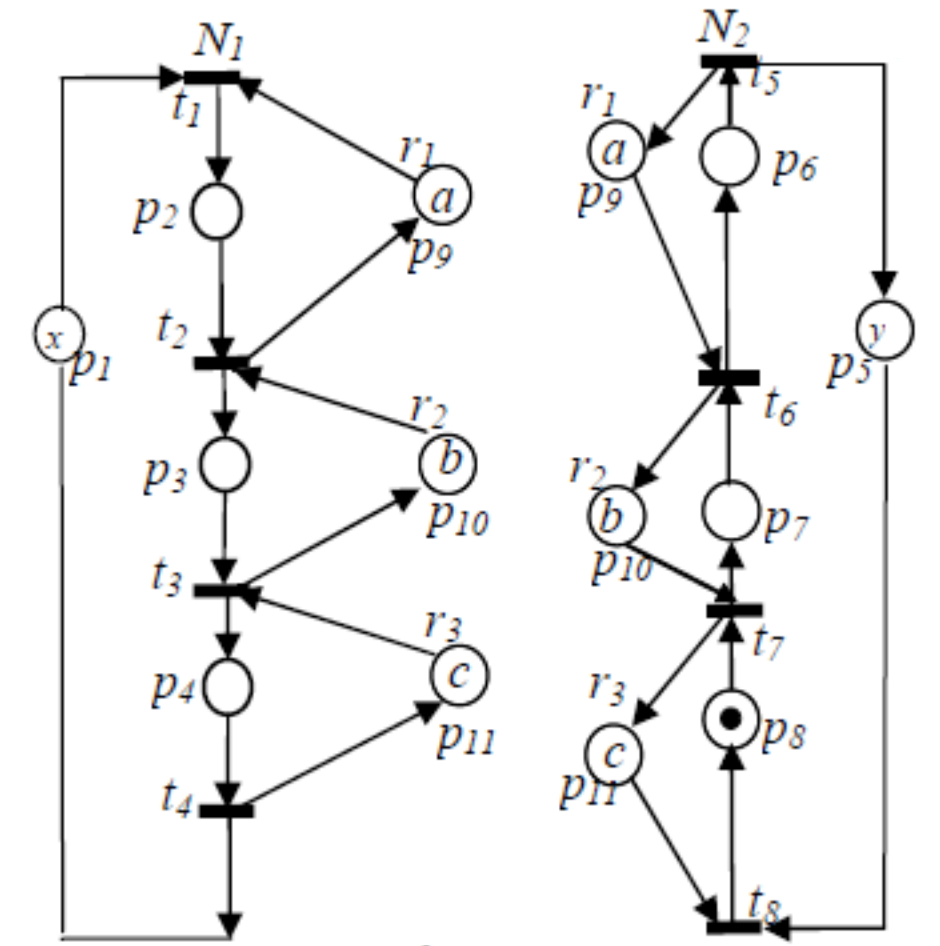


Fig. 1(c) S²PR of N_1 and N_2

O problema do acoplamento com recursos

O sistema S2PR pode ainda ser acoplado de modo que os processos sequenciais interferem um no outro podendo causar atrasos e até deadlocks devido à falta de recursos no momento devido.

Li, Z.W. and Zhou, M.C., "Deadlock resolution in automated manufacturing systems: A novel Petri net approach," Springer, London, 2009

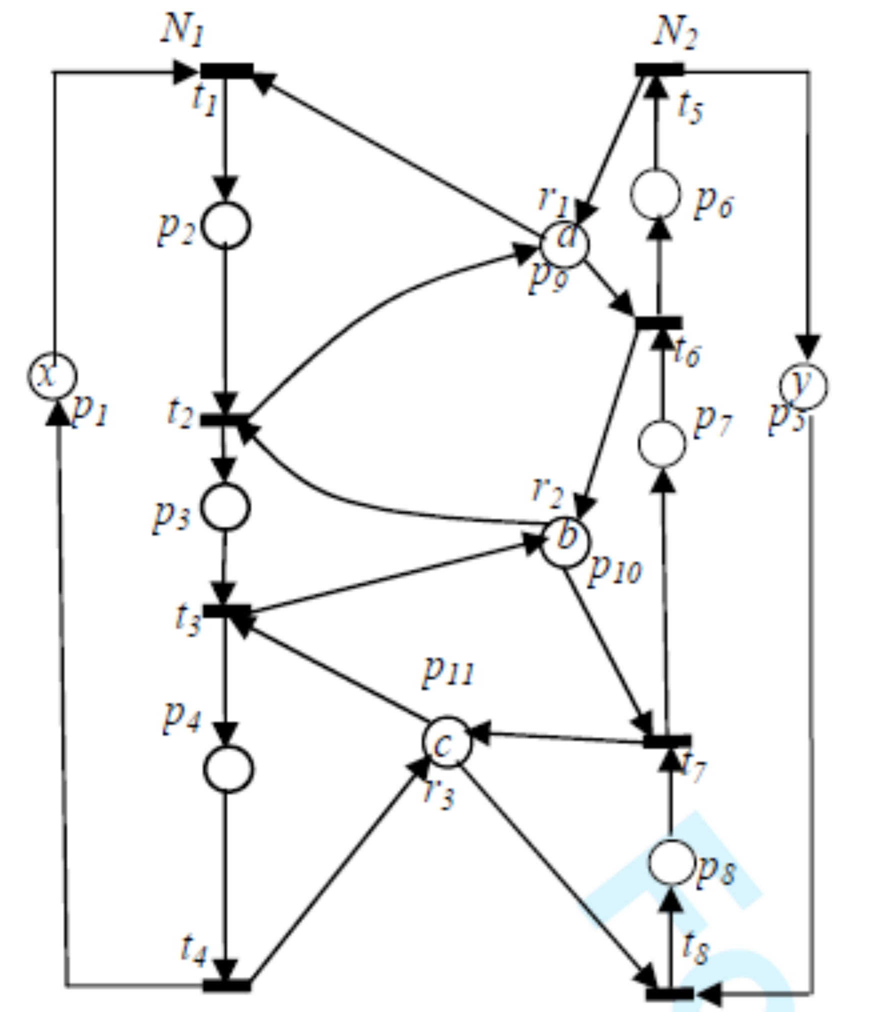
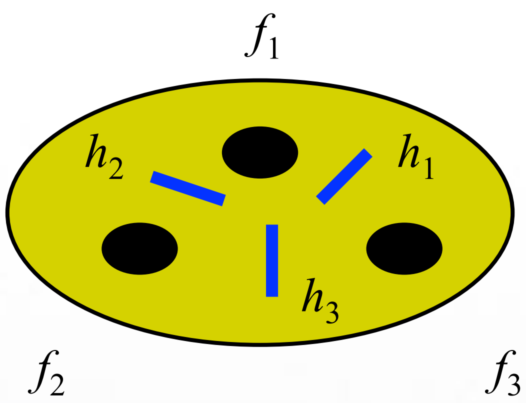
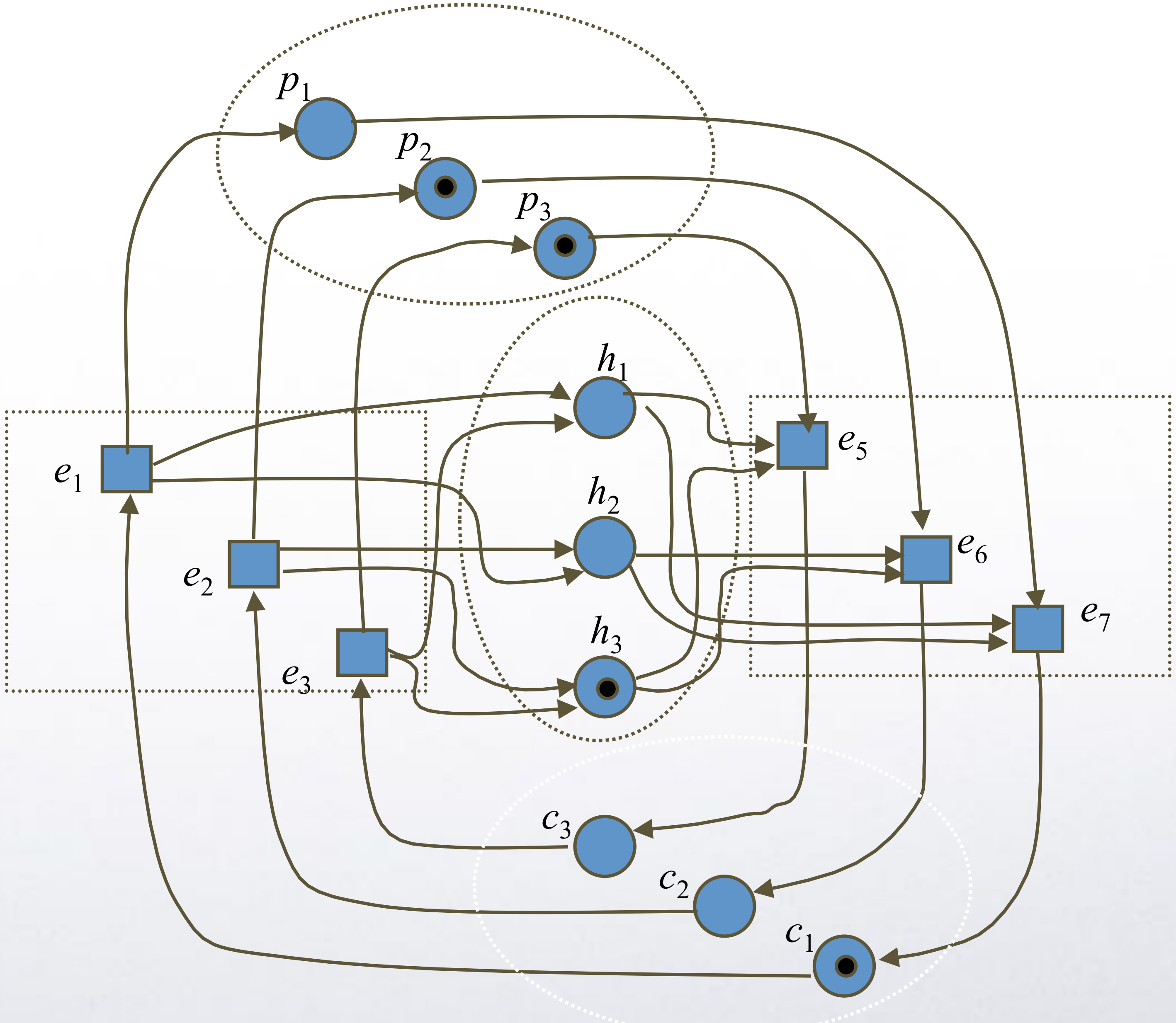


Fig. 1(a) S^3PR &3rd-order system ; $a=b=c=1$.

Dobramento em RdP

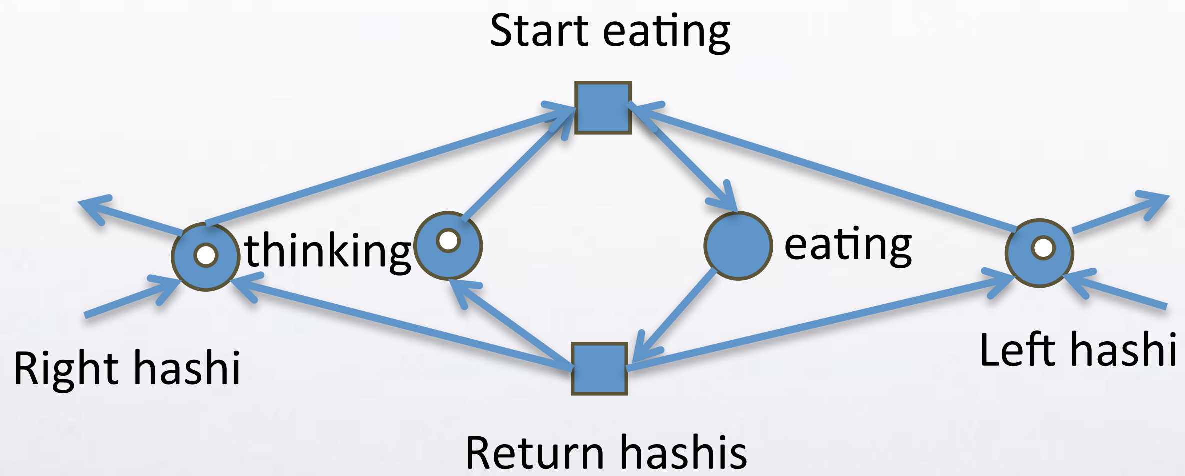
O exemplo dos filósofos





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:



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.

Tratamento Unificado das Redes de Petri

Estamos portanto em posição para tratar todas as redes como o mesmo ente matemático, com diferentes formalismos: o de álgebra linear & teoria dos grafos, ou o de teoria de tipos (standar ML) & grafos.

Já discutimos algumas vantagens e desvantagens de lidar com uma ou outra abordagem e as consequências nos métodos de análise e nas aplicações.
Uma delas é a existência da equação de estado e a possibilidade da análise de usar análise de invariantes.

Redes P/T: Definição

Definition 16

Uma rede Place/Transition P/T, é uma n-upla, $N = (S, T; F, W, K, M_0)$, onde,

- S é um conjunto finito de lugares;
- T é um conjunto finito de trasições;
- $F = (S \times T) \cup (T \times S)$ representa as relações de fluxo (arcos);
- $W : F \rightarrow \mathbb{N}^+$ representando o peso, isto é, a quantidade de marcas que flui em cada arco;
- $K : S \rightarrow \mathbb{N}$ é um mapeamento que atribui a cada lugar uma capacidade máxima para o armazenamento de marcas.
- M_0 é a marcação inicial.

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 $\bar{N}$.

o que é equivalente à definição 4.1.2 do livro texto (pag. 41).

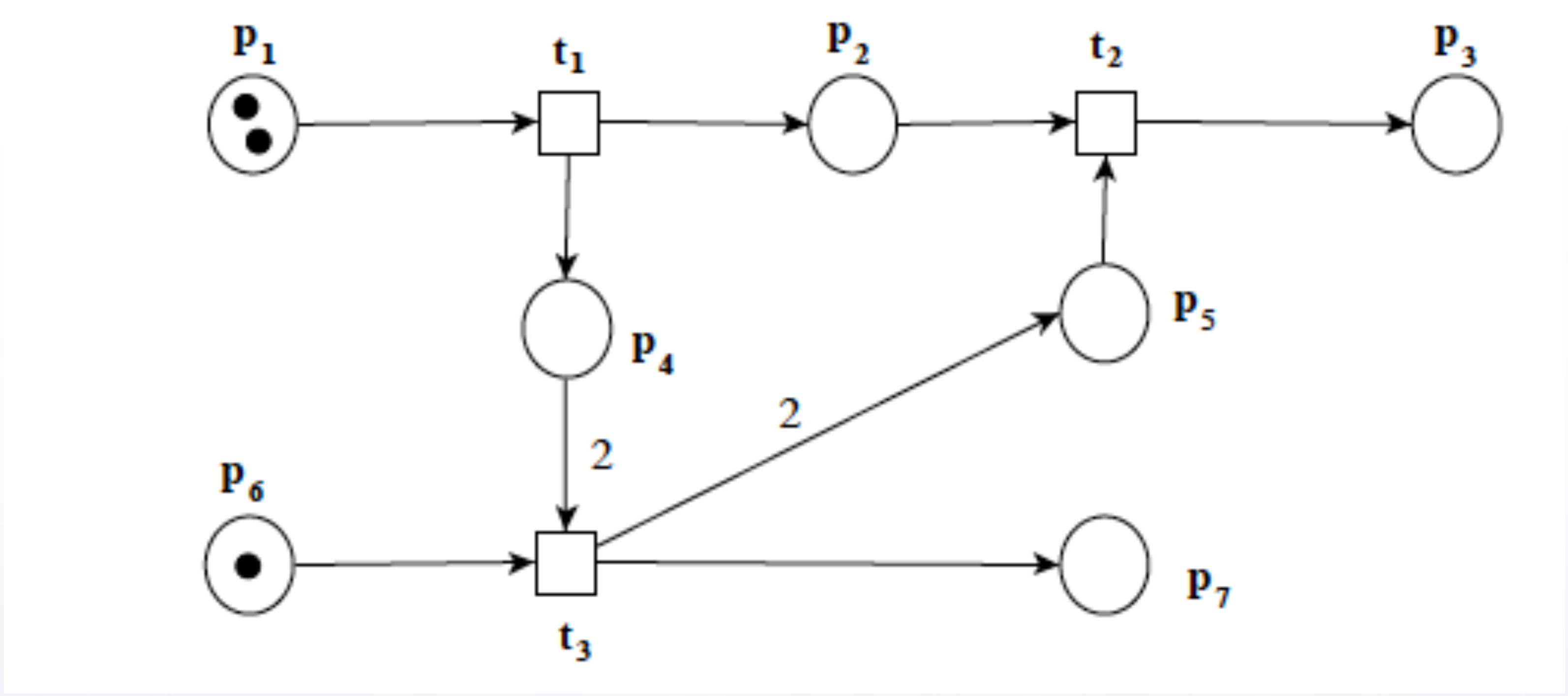
Outra maneira de definir a rede P/T

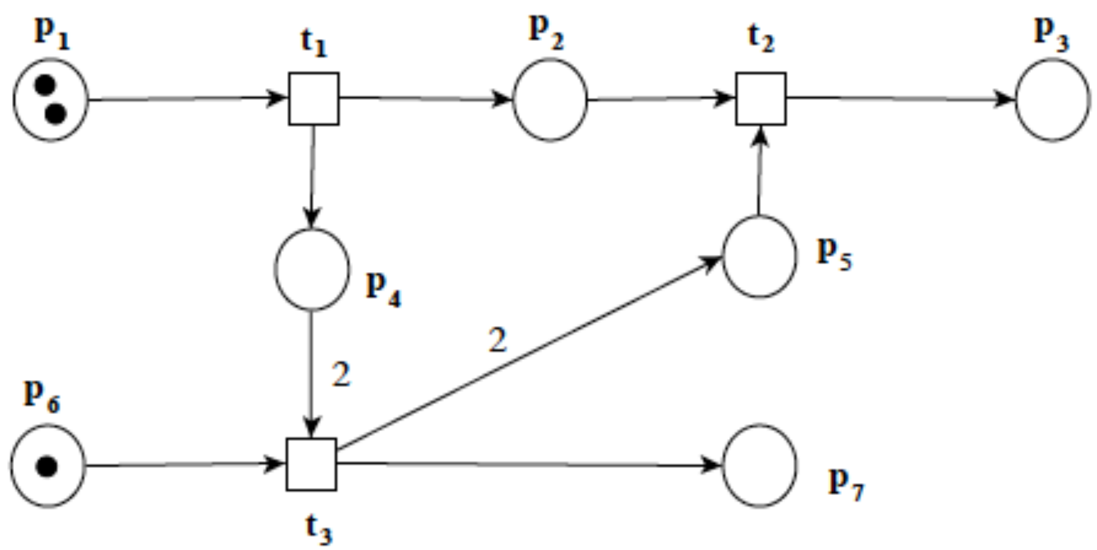
Definition 4.1.1. *A* place/transition net *(P/T net) is defined by a tuple* $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post} \rangle$, *where*

- P *is a finite set (the* set of places *of* $\mathcal{N}$*),*
- T *is a finite set (the* set of transitions *of* $\mathcal{N}$*), disjoint from* P*, and*
- $\mathbf{Pre}, \mathbf{Post} \in \mathbb{N}^{|P| \times |T|}$ *are matrices (the* backward *and* forward incidence matrices *of* $\mathcal{N}$*).* $\mathbf{C} = \mathbf{Post} - \mathbf{Pre}$ *is called the* incidence matrix *of* $\mathcal{N}$.

Girault, C. and Valk, R.; Petri Nets for Systems Engineering, Springer, 2003

vamos tomar como exemplo a rede N_3 do livro texto





A matriz de incidência C é dada por :

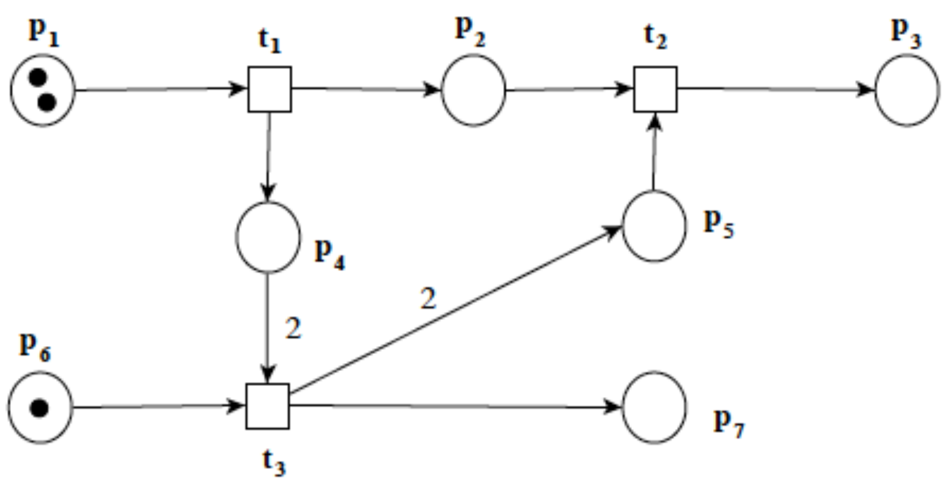
$$C = Post - Pre$$

Pre	t_1	t_2	t_3
p_1	1		
p_2		1	
p_3			
p_4			2
p_5		1	
p_6			1
p_7			

Post	t_1	t_2	t_3
p_1			
p_2	1		
p_3		1	
p_4	1		
p_5			2
p_6			
p_7			1

C	t_1	t_2	t_3
p_1	-1		
p_2	1	-1	
p_3		1	
p_4	1		-2
p_5		-1	2
p_6			-1
p_7			1

Definition 4.1.4. *A* marking *of a P/T net* $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post} \rangle$ *is a vector* $\mathbf{m} \in \mathbb{N}^{|P|}$. $\mathcal{N}$ *together with a marking* $\mathbf{m}_0$ *(initial marking) is called a* P/T net system $\mathcal{S} = \langle \mathcal{N}, \mathbf{m}_0 \rangle$ *or* $\mathcal{S} = \langle P, T, \mathbf{Pre}, \mathbf{Post}, \mathbf{m}_0 \rangle$. *A transition* $t \in T$ *is* enabled *in a marking* $\mathbf{m}$ *if* $\mathbf{m} \geq \mathbf{Pre}[\bullet, t]$. *In this case the* successor marking relation *is defined by* $\mathbf{m} \xrightarrow{t} \mathbf{m}' \Leftrightarrow \mathbf{m} \geq Pre[\bullet, t] \wedge \mathbf{m}' = \mathbf{m} + \mathbf{Post}[\bullet, t] - \mathbf{Pre}[\bullet, t] = \mathbf{m} + C[\bullet, t]$. $\mathbf{Pre}[\bullet, t]$ *denotes the t-column vector* $\mathbf{Pre}[\bullet, t] = (\mathbf{Pre}[p_1, t], \ldots, \mathbf{Pre}[p_{|P|}, t])$ *of the* $|P| \times |T|$ *matrix* $\mathbf{Pre}$. *The same holds for* $\mathbf{Post}[\bullet, t]$ *with respect to* $\mathbf{Post}$.[1]



Pre	t_1	t_2	t_3
p_1	1		
p_2		1	
p_3			
p_4			2
p_5		1	
p_6			1
p_7			

Post	t_1	t_2	t_3
p_1			
p_2	1		
p_3		1	
p_4	1		
p_5			2
p_6			
p_7			1

C	t_1	t_2	t_3
p_1	-1		
p_2	1	-1	
p_3		1	
p_4	1		-2
p_5		-1	2
p_6			-1
p_7			1

A marcação inicial é m_0=(2, 0, 0, 0, 0, 1, 0). A transição t_1 está habilitada e a marcação sucessora a m_0 é m′, onde

$$m' = m_0 + Pos(\bullet, t1) - Pre(\bullet, t1) = m_0 + C(\bullet, t1) =$$

$$(2, 0, 0, 0, 0, 1, 0) + (0, 1, 0, 1, 0, 0, 0) - (1, 0, 0, 0, 0, 0, 0) =$$

$$(1, 1, 0, 1, 0, 1, 0)$$

Qualquer que seja a rede P/T, é sempre possível obter o seu "dobramento" (folding) através de fatoração, resultando em uma rede quociente.

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.

Por sua vez a rede quociente é equivalente a uma rede de alto nível (HLPN),

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

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

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

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

Veremos mais adiante um processo chamado unfolding ("desdobramento"), que consiste justamente em desfazer os dobramentos resultantes do processo de fatoração.

Trata-se portanto em desfazer o colapso das simetrias que levaram aos dobramentos, e à identificação das marcas nas redes de alto nível/coloridas. Portanto, no unfolding cada tipo de marca deve ter sua própria sub-rede, simétrica e funcionalmente equivalentes entre si. A rede resultante deste processo é novamente uma rede P/T.

Dada uma rede P/T é sempre possível dobrá-la e obter com isso a correspondente rede quociente. Por sua vez a rede quociente pode ser trabalhada (baseada em sorts e teoria de tipos) para obter a rede de alto nível equivalente. É possível também traçar o caminho inverso e, à partir da rede de alto nível, obter a rede equivalente P/T, usando para isso as técnicas de unfolding já descritas.

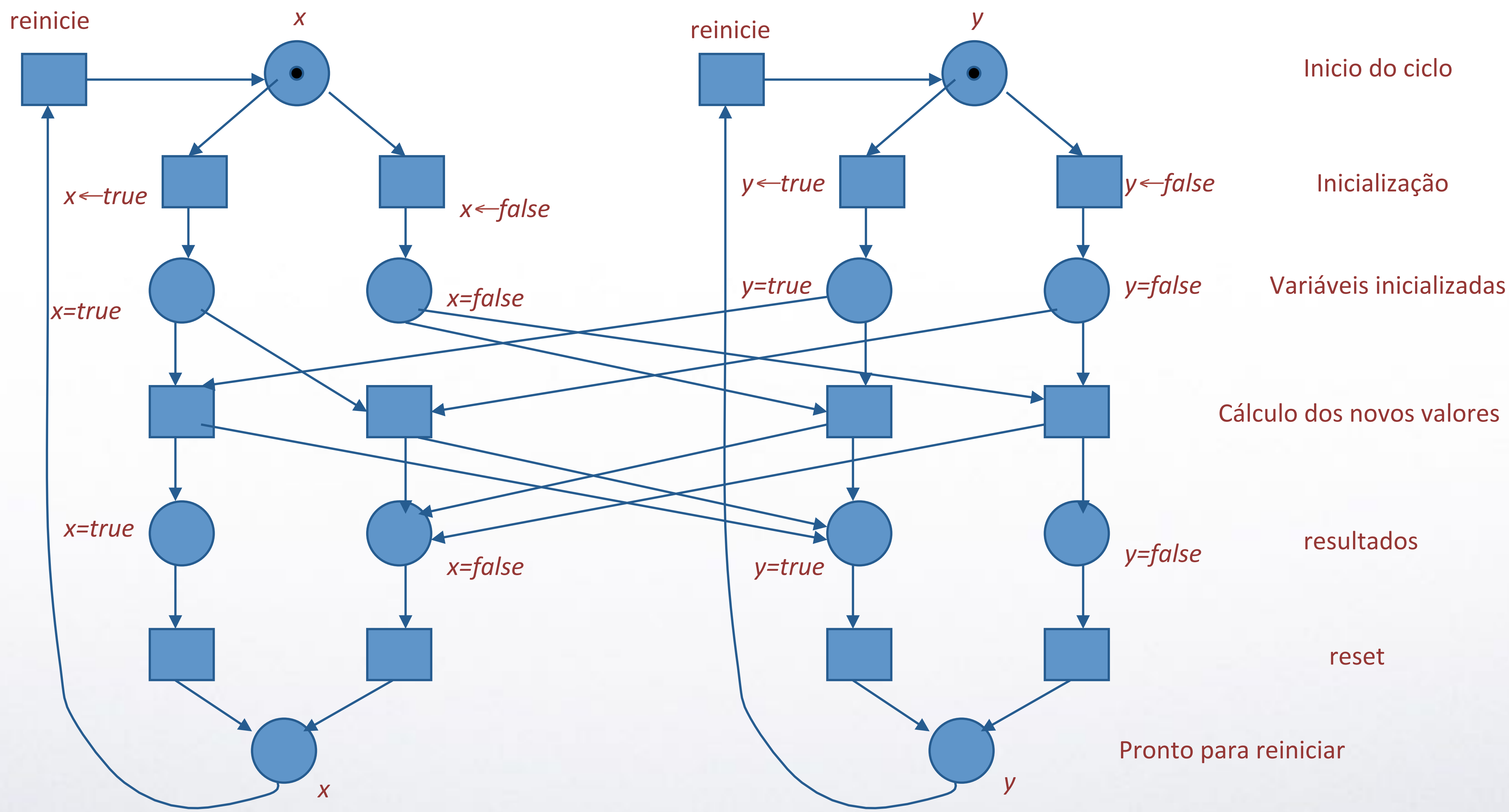
A High-level Petri Net Graph (HLPNG) comprises:

- *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.

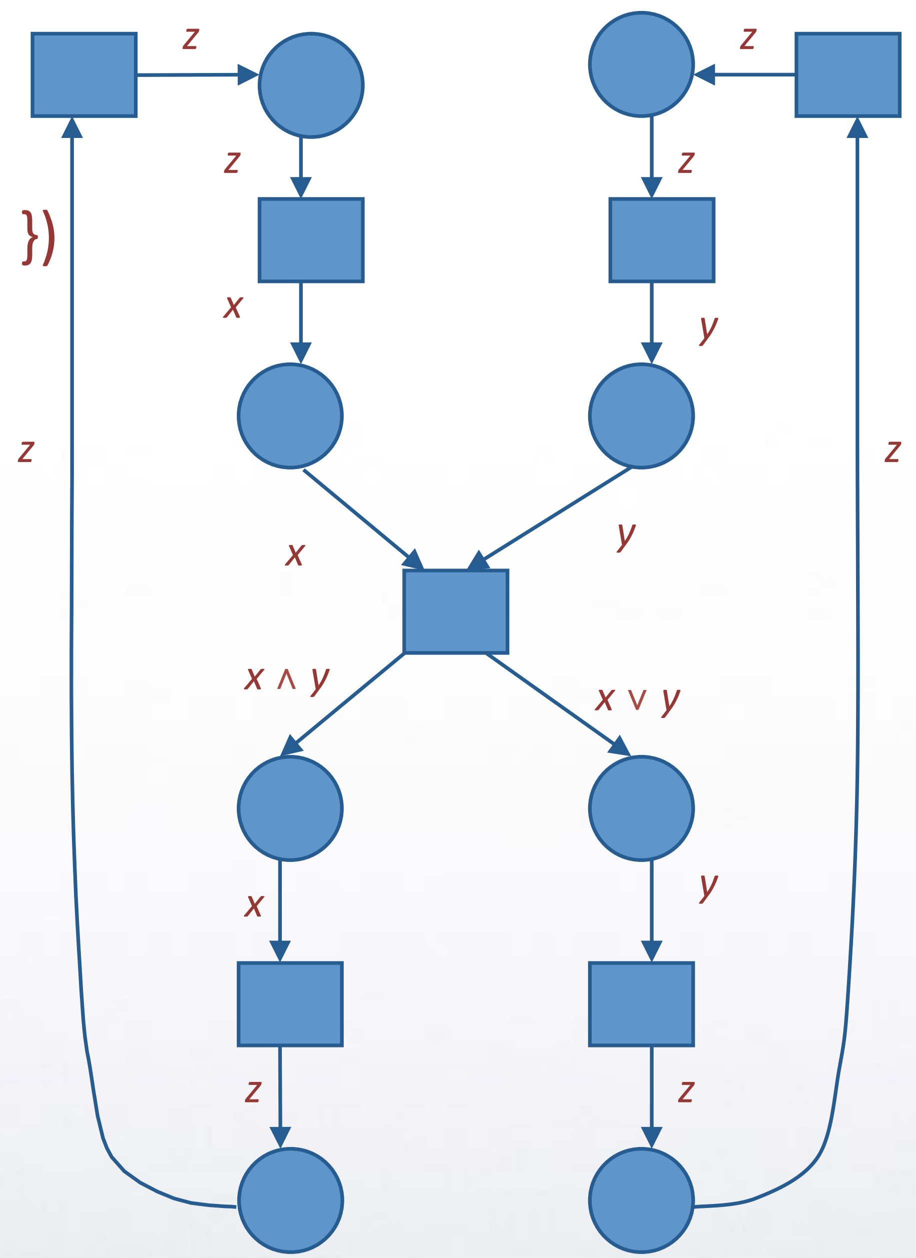
A rede de Petri representando a operação desta máquina é mostrada a seguir.


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
Pronto para reiniciar
x
y

D={true, false}
ϕ={ ∧, ∨ }
δ=(D;ϕ)=({true,false}, { ∧, ∨ })

if x=true and y=true
then x ∧ y = true
else x ∧ y = false

if x=false and y=false
then x ∨ y = false
else x ∨ y = true



Inicio do ciclo

Inicialização

Variáveis inicializadas

Cálculo dos novos valores

resultados

reset

Pronto para reiniciar

Exercício

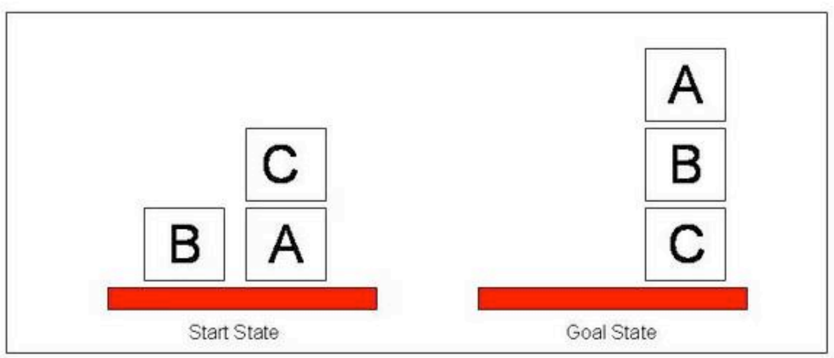
Até aqui as redes de alto nível HLPN foram apresentadas como o resultado do processo de fatoração (folding) de uma rede clássica P/T. Assim as redes HLPN têm sempre uma rede base P/T e fica sub-entendido que o processo clássico de modelagem seria sintetizar uma rede P/T e depois fatorá-la para achar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

Mas é notório que, exceto pela análise de algumas propriedades (como veremos adiante), as redes de alto nível inserem uma considerável síntese na modelagem e design do sistema.

Vale portanto a seguinte questão: seria possível sintetizar direto uma rede de alto nível dado um problema de modelagem e análise genérico? A rede P/T é uma fase intermediária realmente necessária?

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.

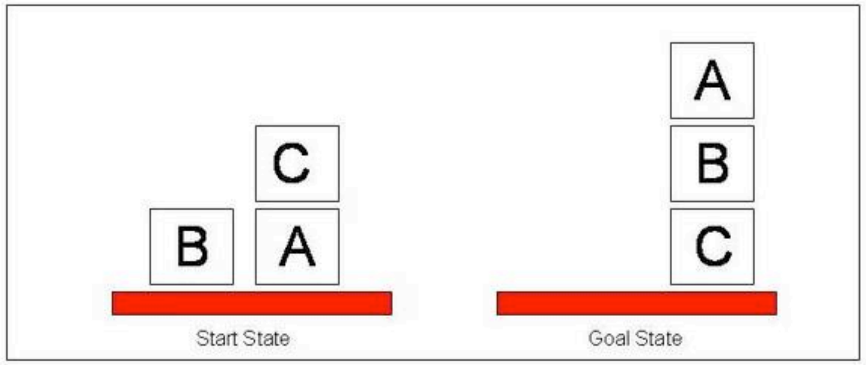
Com estas regras pretende-se fazer um plano de ações para que o robô transforme o estado inicial mostrado na figura no estado final.

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)

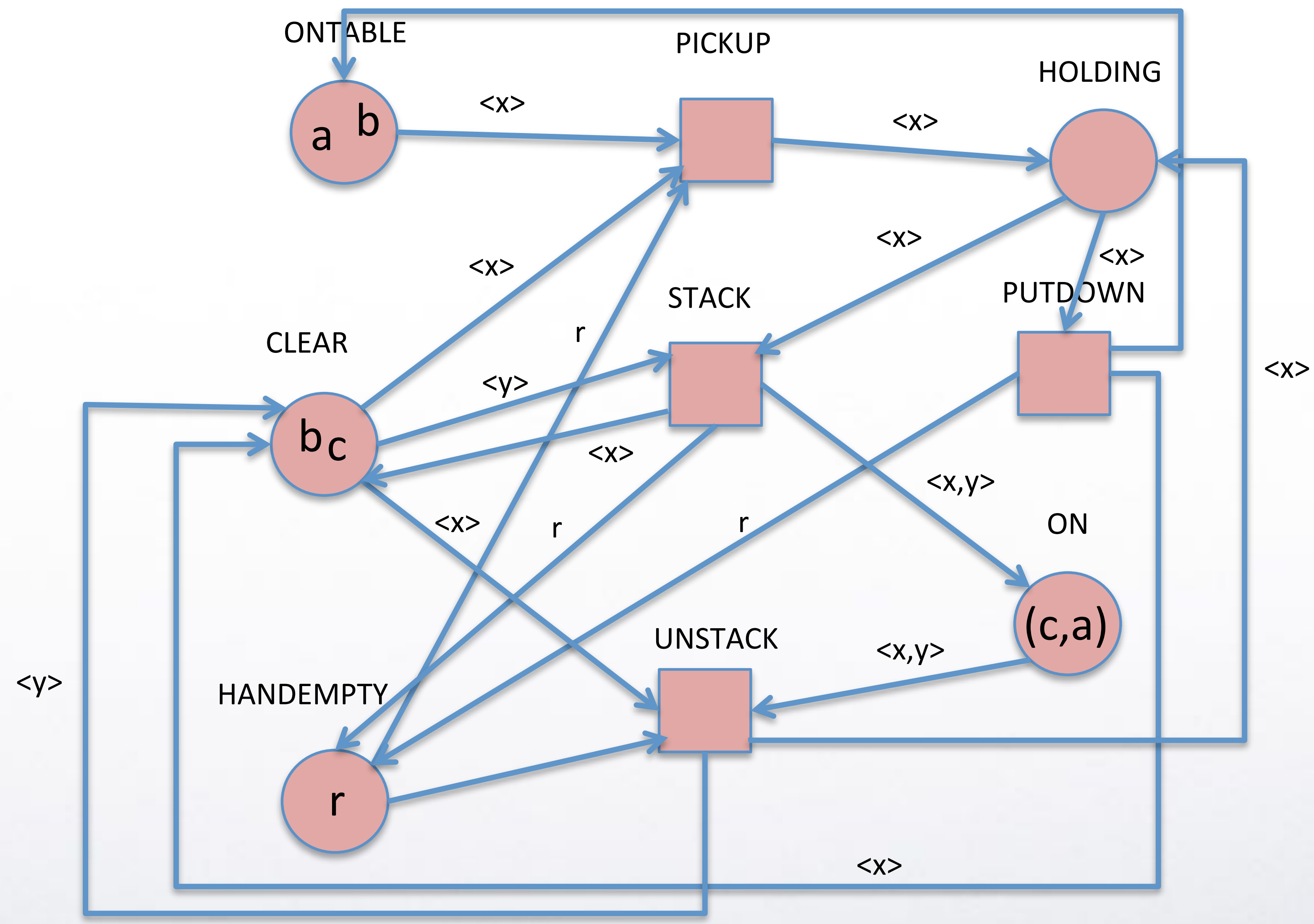
O problema (de planejamento)

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

Tente fazer isso, mesmo já tendo visto a solução.

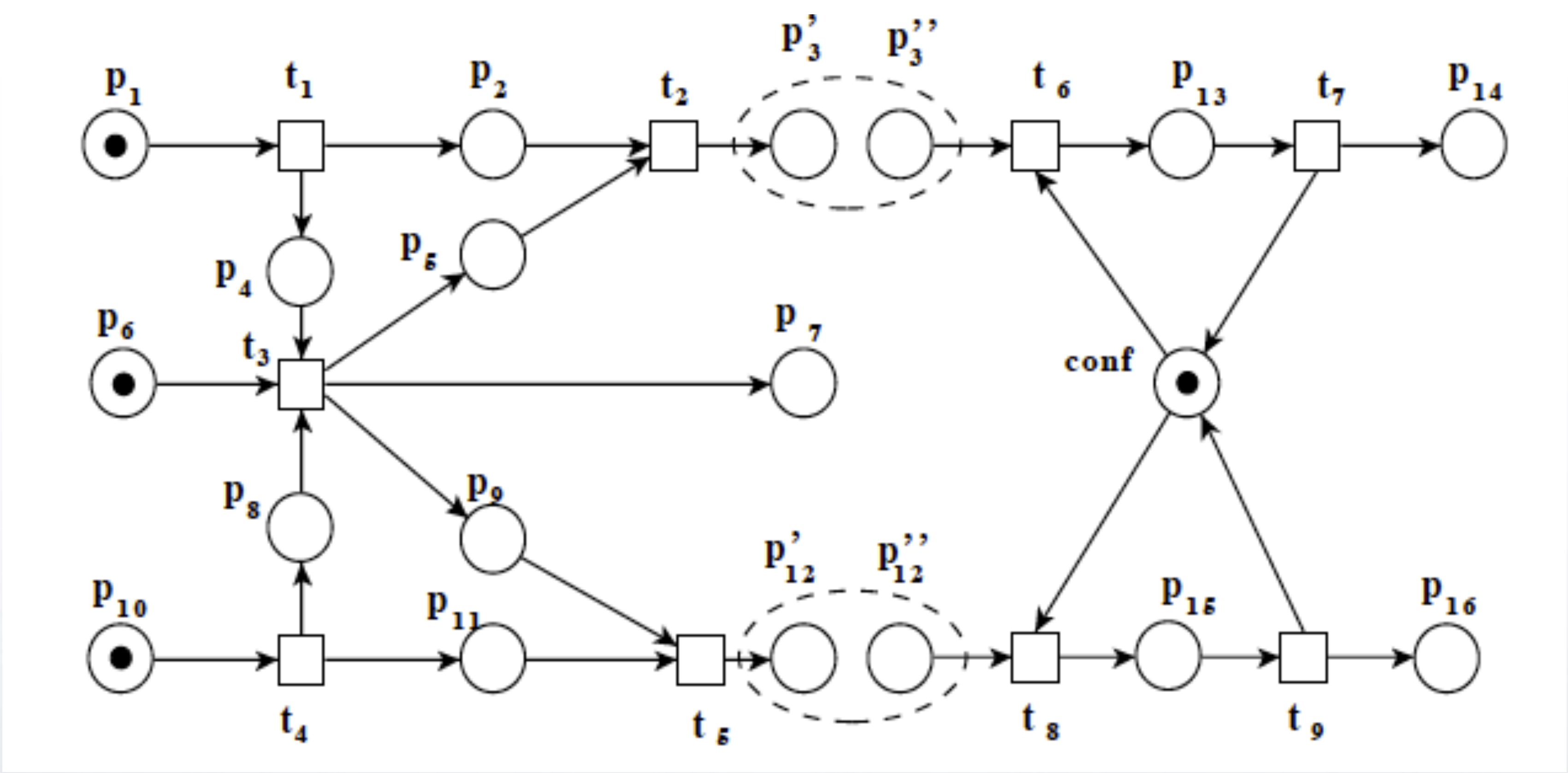
Coloured Petri Nets

Definition 4.3.1. *A* coloured Petri net *(CPN) is defined by a tuple* $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post}, \mathcal{C}, cd \rangle$, *where*

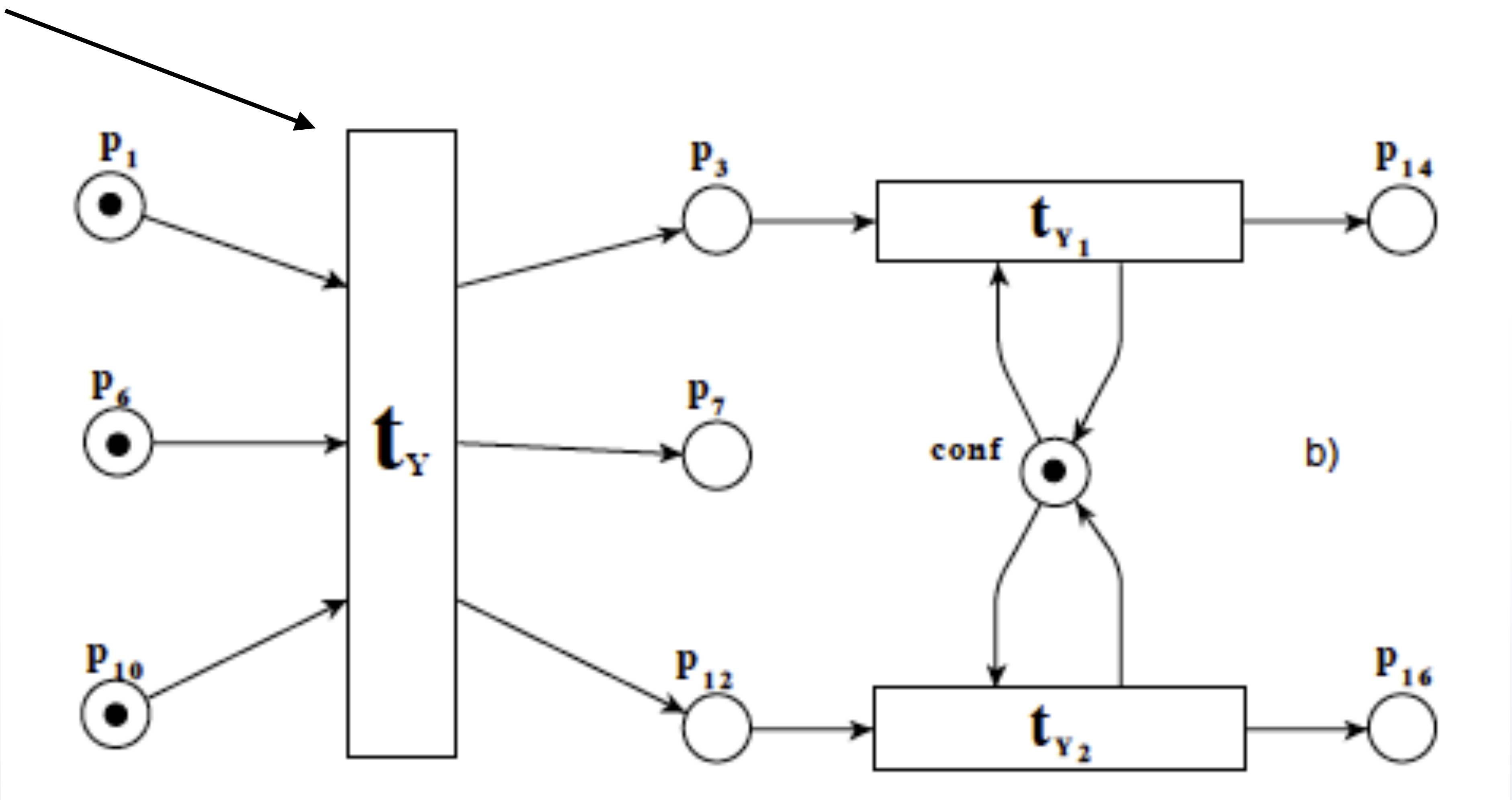
- P *is a finite set (the* set of places *of* $\mathcal{N}$*),*
- T *is a finite set (the* set of transitions *of* $\mathcal{N}$*), disjoint from* P*,*
- $\mathcal{C}$ *is the set of* colour classes*,*
- $cd : P \cup T \to \mathcal{C}$ *is the* colour domain mapping*, and*
- $\mathbf{Pre}, \mathbf{Post} \in \mathcal{B}^{|P| \times |T|}$ *are matrices (the* backward *and* forward incidence matrices *of* $\mathcal{N}$*) such that* $\mathbf{Pre}[p,t] : cd(t) \to \mathrm{Bag}(cd(p))$ *and* $\mathbf{Post}[p,t] : cd(t) \to \mathrm{Bag}(cd(p))$ *are mappings for each pair* $(p,t) \in P \times T$.

$\mathcal{B}$ *can be taken as the set of mappings of the form* $f : cd(t) \to \mathrm{Bag}(cd(p))$. *Again,* $\mathbf{C} = \mathbf{Post} - \mathbf{Pre}$ *is called the* incidence matrix.

Vamos considerar novamente o problema da largada de fórmula 1, agora seguido de um checkpoint onde se verifica a posição de cada carro.



macrotransição



É possível dobrar esta rede? O que fazer como o dobramento?

Histórico das redes CPN

As redes coloridas surgiram nos anos 80 conjugando a representação gráfica das redes de Petri com o Standard ML, que representa tipos e cores.

No final dos anos 80 e princípio dos anos 90 surgiu o ambiente Dsign CPN proposto pelo mesmo grupo de Ahus, Dinamarca (Kurt Jensen).

A idéia é simplesmente ter um formalismo mais abstrato

Definição informal das CPNs

Kurt Jensen

Coloured Petri Nets (CP-nets or CPNs) is a graphical language for constructing models of concurrent systems and analyzing their properties. CP-nets is a discrete-event modeling language combining Petri Nets and the functional programming language CPN ML which is based on Standard ML.

Aplicações

Aplicações Típicas	**Novas Aplicações**
Protocolo de Comunicação	Sistemas de workflow
Redes de Dados	Sitemas de manufatura
Algorítmos Distribuídos	Sistemas multi-agente
Sistemas Embarcados	Processos de negócio
	Análise de Requisitos

Uso prático das redes CPN

Como no caso das redes clássicas o uso prático das redes CPN está associado a **Simulação** do modelo, e portanto ao estudo de cenários específicos e ao processo de evolução das marcas.

Temos portanto o mesmo problema de desenvolver métodos alternativos de análise baseados nas propriedades da rede e fugir do problema da atingibilidade.

Novos métodos → Verificação e Model checking

Fim