

# ***PMR 5237***

## Modelagem e Design de Sistemas

### Discretos em Redes de Petri

#### Aula 3: Redes Elementares e Redes P/T

Prof. José Reinaldo Silva  
[reinaldo@usp.br](mailto:reinaldo@usp.br)

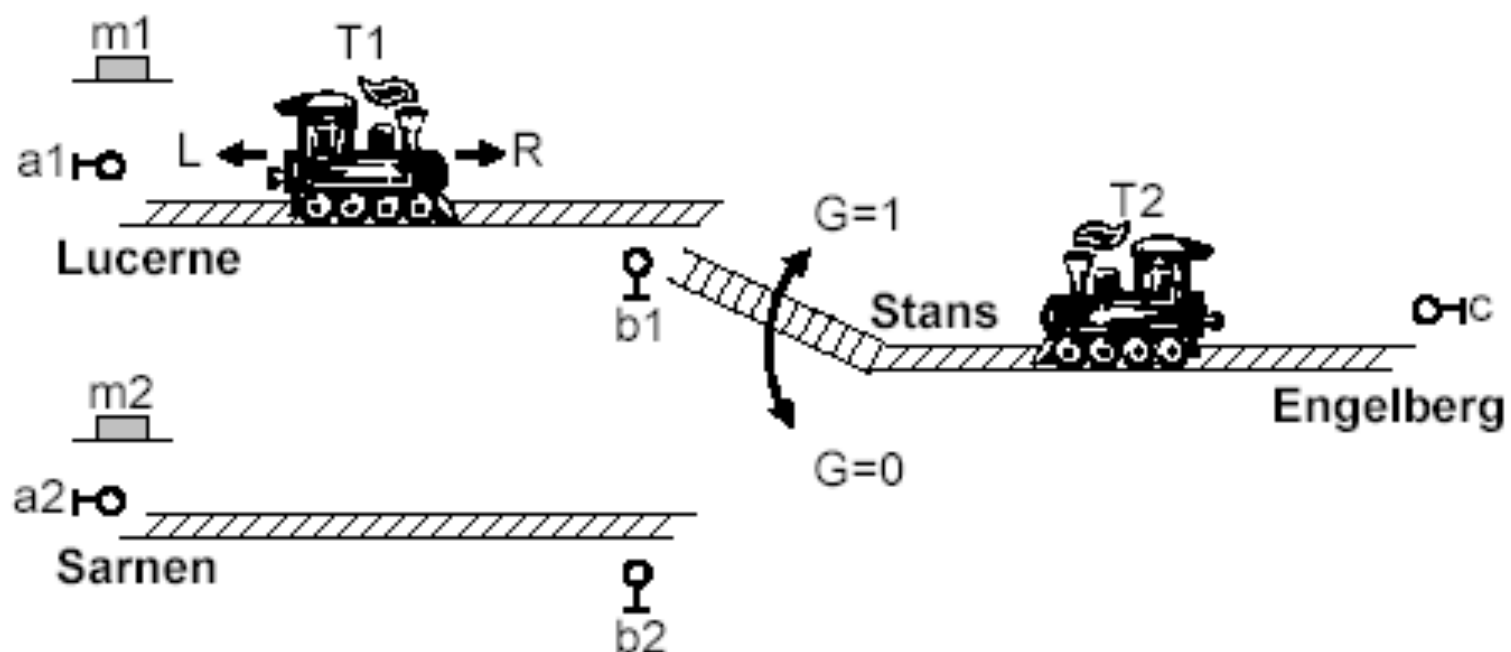


# Princípios para modelagem em Redes de Petri

As redes possuem propriedades típicas dos esquemas que as tornam  
Uma excelente representação formal para sistemas (dinâmicos) discretos,  
Entre os quais figuram :

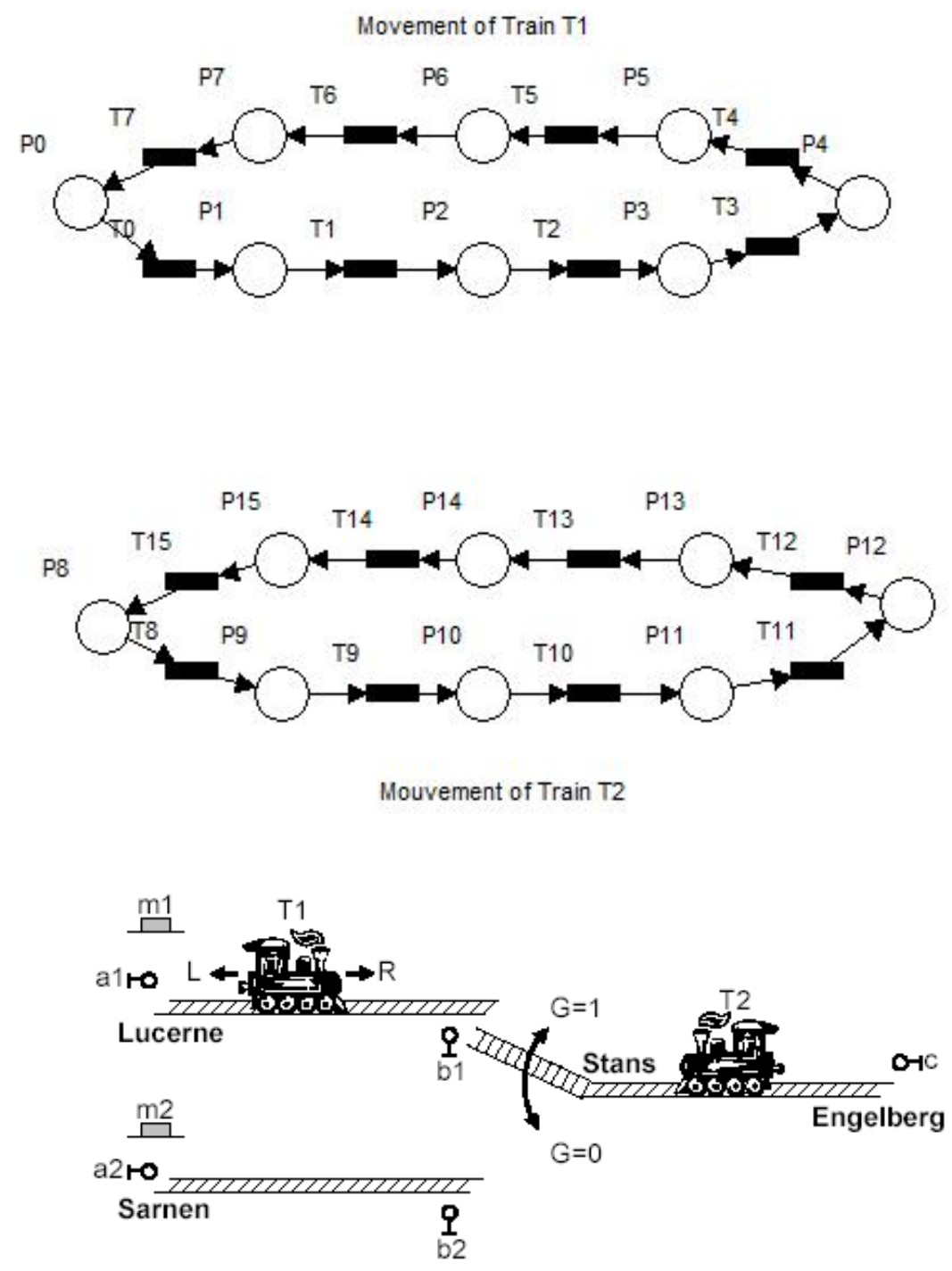
- o princípio da dualidade
- o princípio da localidade
- o princípio da concorrência
- o princípio da representação gráfica
- o princípio da representação algébrica

# Exemplo: manobrando linhas de trem



# O problema de automação e controle

Nos diagramas ao lado temos o modelo gráfico do movimento de cada trem (um esquema cuja interpretação do significado de lugares e transições se encontra nas transparências anteriores). O problema de automação aqui é do tipo semáforo, no sentido que somente um dos trens pode estar no trecho unificado de cada vez, e de sincronismo, dado que, se um dos trens (T1) faz o trajeto de Lucerne a Engelberg, ao voltar deve encontrar o gate G na posição I. Similarmente o outro trem (T2) deve encontrar este mesmo gate na posição G=0.

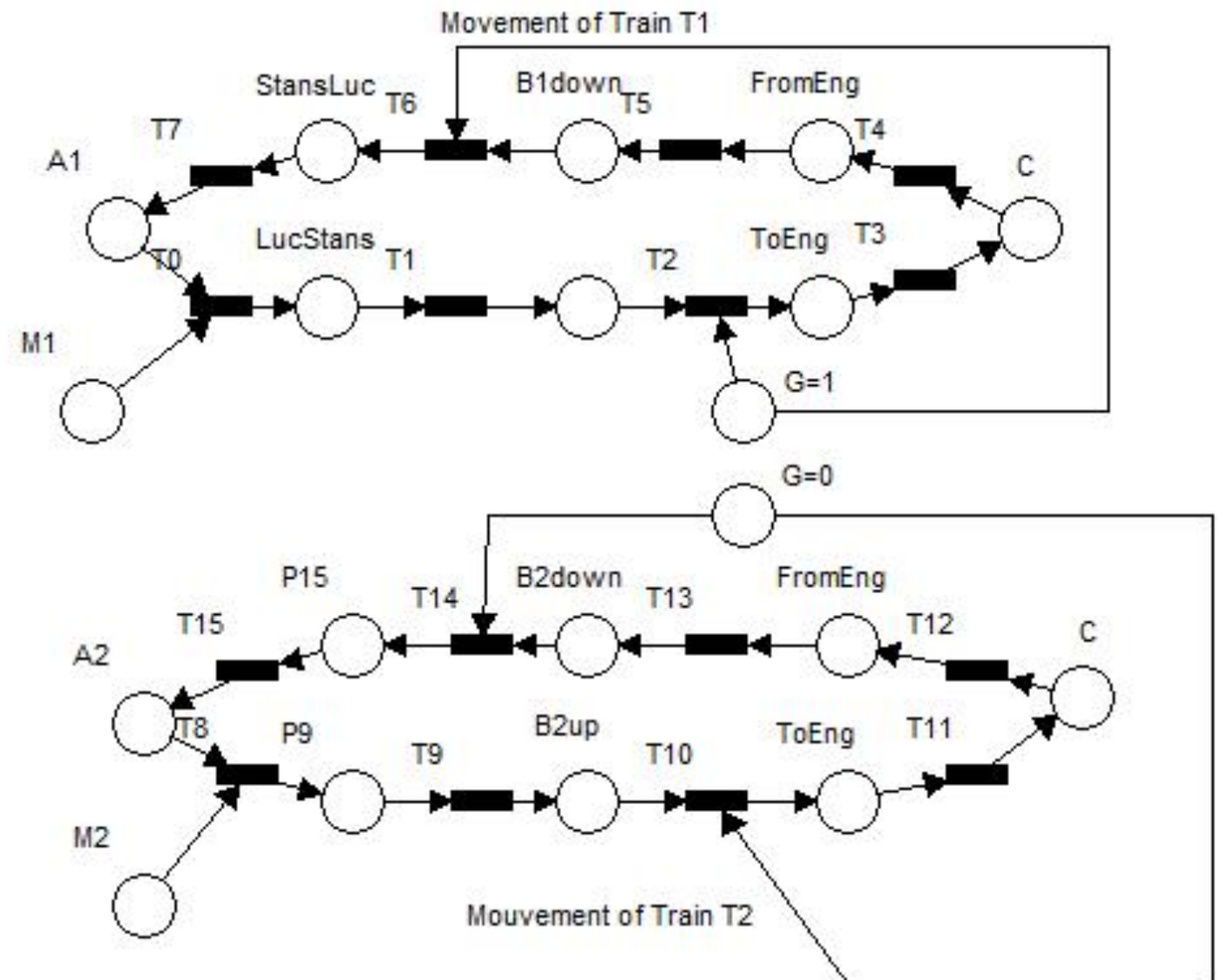
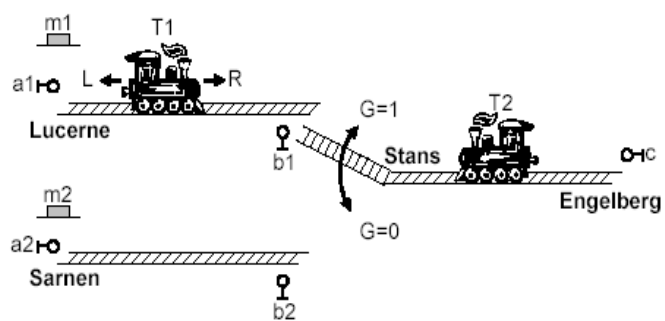


## Explorando simetrias



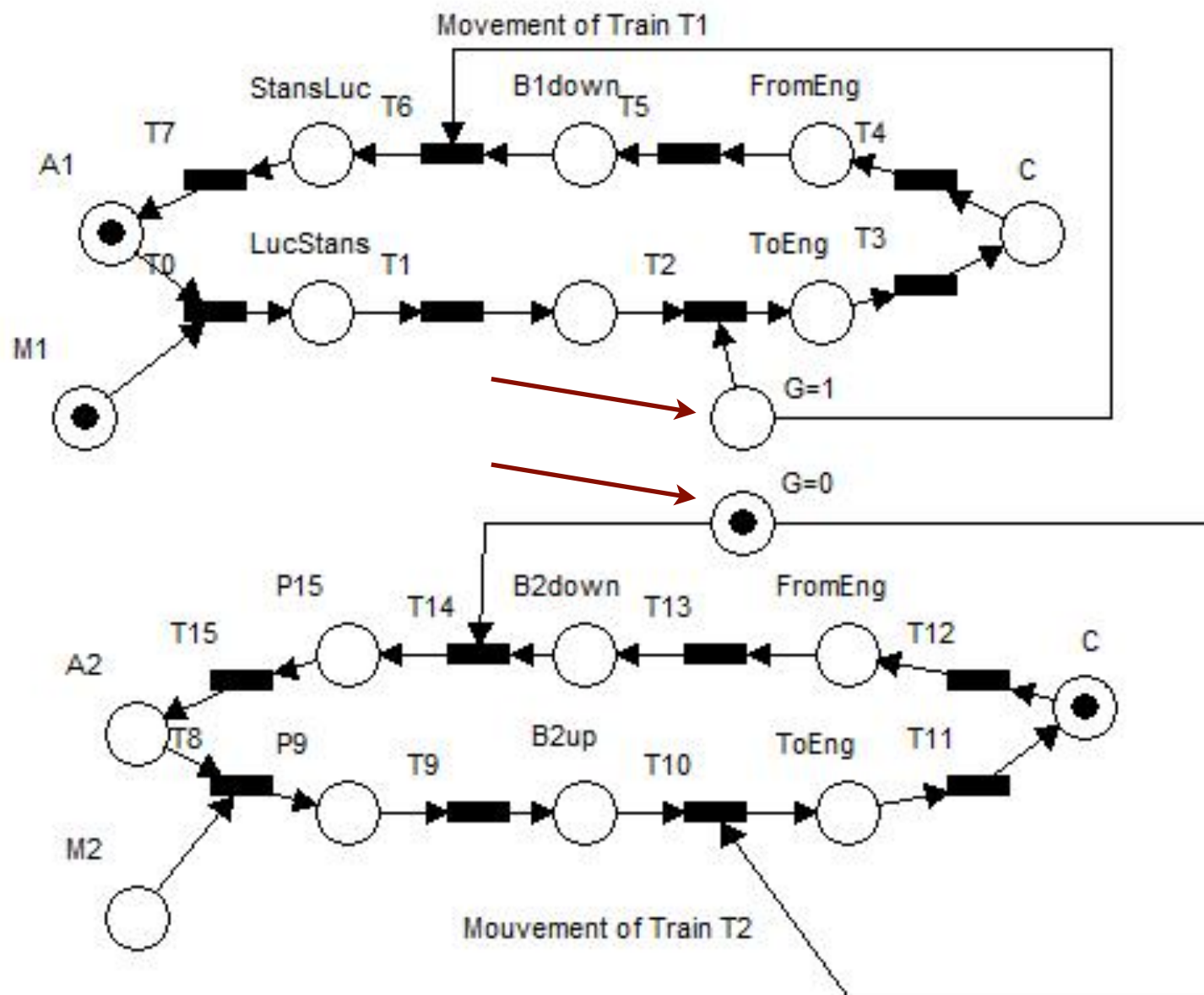
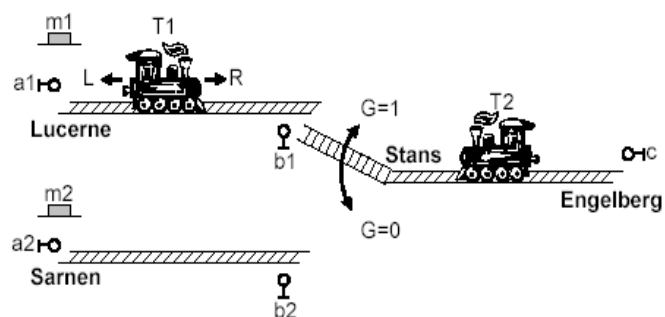
# Usando o gate para sincronizar o movimento

Modelamos então o estado do gate G e sua influência no movimento de cada trem



# Síntese do modelo obtido

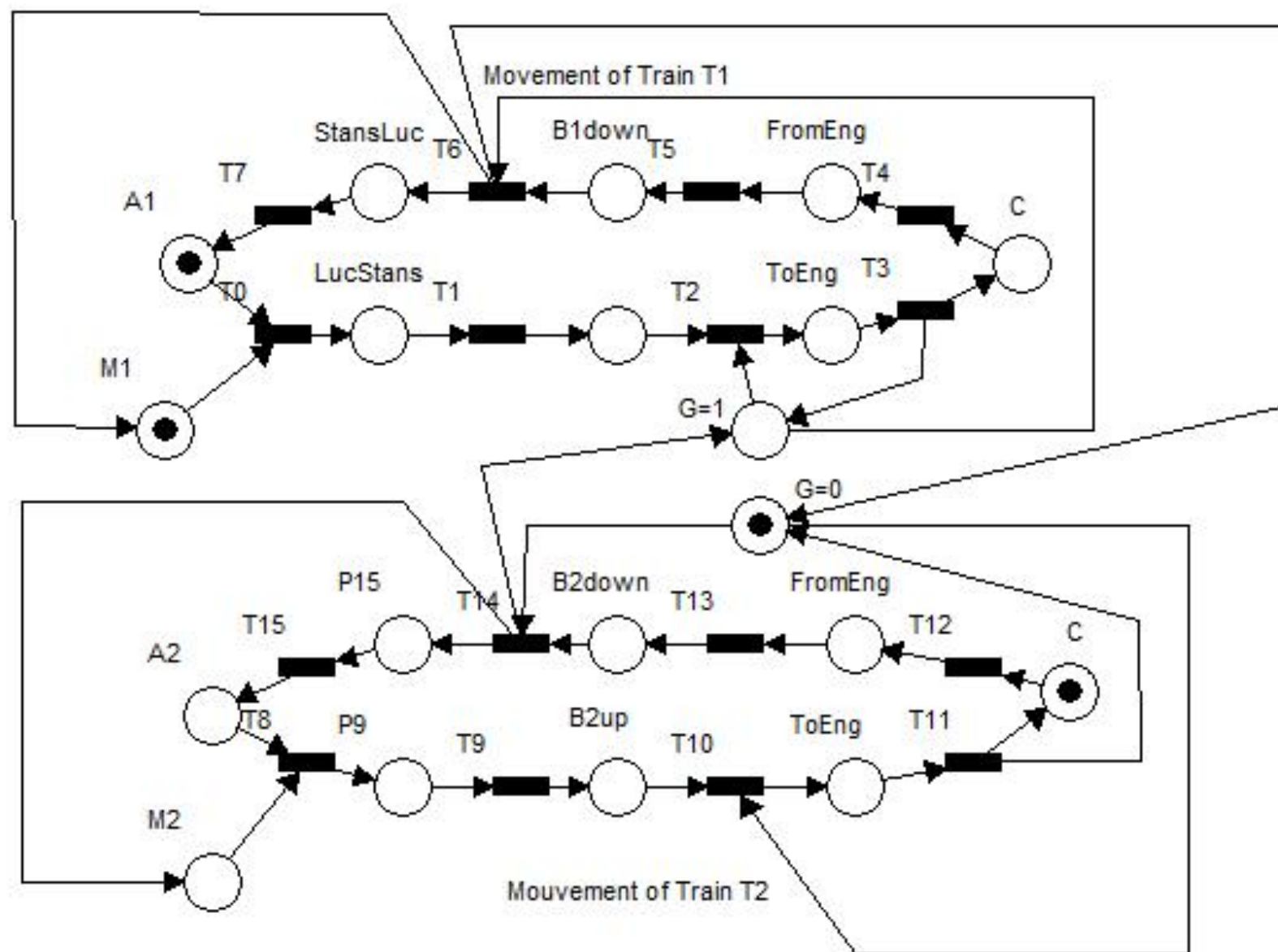
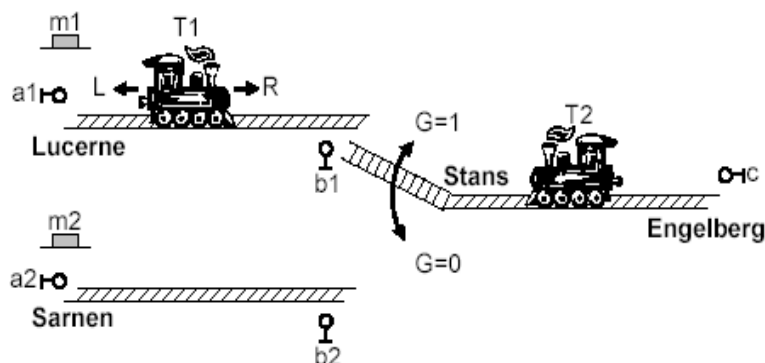
Inserindo o estado inicial temos o problema parcialmente modelado, isto é, apenas com a sincronização resolvida. Mas note que os lugares apontados pelas setas representam estados do mesmo gate  $G$ . Portanto se um deles é marcado automaticamente desmarca o outro, configurando um conflito





# O modelo completo

Garantindo a alternância de marcação do mutex, e também que o modelo seja cíclico, isto é, que retorna ao estado inicial depois de alguns disparos, temos o modelo completo.



# A equação de estado

Finalmente, podemos ter a equação que dá o fluxo de marcas (equação de estado) expressa na forma matricial como,

$$\dot{M}_i = M_0 + A^T \sum_0^{i-1} T_j = M_0 + A^T \sigma_{i-1}$$

Lista de exercícios: Exec. 4

Mostre que se o vetor de habilitação usado na equação de estado denotar uma situação de conflito o estado final é inconsistente, isto é, pode ter marcação negativa.



Como vimos na aula passada a equação de estado relaciona não apenas localidade (como definida até aqui) mas pode, se aplicada recursivamente, associar um dado estado com o estado inicial.

$$M_i = M_0 + A^T \sum_0^{i-1} T_j = M_0 + A^T \sigma_{i-1}$$

Isto em princípio nos habilitaria a “resolver” esta equação e assim determinar se um dado estado seria atingível a partir do estado inicial. No entanto, como já vimos, ter uma solução inteira positiva NÃO implica que o estado dado seja atingível. Você é capaz de reproduzir os argumentos?

# Processo de Modelagem

- Identificar todos os estados (determinar o espaço de estados)
- Identificar todas as transições (determinar as transições admissíveis)
- Identificar as possíveis trajetórias no espaço de estados privilegiando as simetrias
- Inserir os sincronismos, conflitos (mutex) e dependências entre trajetórias independentes
- verificar o modelo, o que de forma clássica significa usar um jogador de marcas

# Forward case class

Portanto é possível gerar estados a partir de um estado dado, que pode ser, por exemplo o estado inicial. O conjunto de estados gerados a partir deste gerador é chamado de forward case class e é denotado por  $[M_0]$

# Sistema Elementar

Portanto, para efeito de modelagem e análise de sistemas a escolha do estado inicial é sempre muito importante. Definiremos a seguir um tipo de redes de Petri, inserido na classe do que é chamado de redes clássicas.

## Definition (8)

Uma rede de Petri elementar é uma n-upla  $N = (S, T; F, M_0)$ , onde  $(S, T; F)$  é uma estrutura de rede como definido anteriormente.

O conjunto de estados que este sistema admite é determinado pela escolha do gerador  $M_0$  e é denotado por  $\mathcal{C}_N = \lceil M_0 \rceil$ , que é o seu "forward case class".

### Definition 9

Seja um sistema elementar  $N = (S, T; F, C_0)$ . Definimos como o case set de  $N$ , e denotamos por  $\mathbb{C}_N$ , o sub-conjunto minimal de  $\mathcal{P}(S)$  que satisfaz às seguintes condições:

- 1  $C_0 \in \mathbb{C}_N$ ;
- 2  $c_1 \in \mathbb{C}_N$  e  $\exists v \in T \mid c_1 \mid v \rangle c_2$ , então  $c_2 \in \mathbb{C}_N$ .

### Definition 10

Seja um sistema elementar  $N = (S, T; F, C_0)$ . O conjunto de todos os passos deste sistema, denotado por  $\mathcal{P}_N$ , é dado por:

$$\mathcal{P}_N = \{v \subseteq T^* \mid \exists c_1, c_2 \in \mathbb{C}_N. c_1 |v\rangle c_2\}$$



Seja a rede elementar  $N = (S, T; F, C_0)$  e o seu case set  $\mathbb{C}_N \subseteq \mathcal{P}(S)$

### Definition 11

Definimos a relação de atingibilidade  $R = (r \cup r^{-1})^*$ , onde  $r \subseteq \mathcal{P}(S) \times \mathcal{P}(S)$  de modo que  $c r c' \iff \exists v \in T^* | c | v \rangle c'$ .

### Proposition 1

O case  $\mathbb{C}$  é classe de equivalencia de  $R$ .

# Sistema Elementar

Seja um sistema elementar  $N = (P, T, F, C_{in})$ , podemos definir como a rede subjacente ao sistema  $N$ , ou simplesmente  $und(N)$ , à rede  $(P, T, F)$ , que também é chamada de estrutura de  $N$ .  
Note-se que a classe de estados definida por  $N$  e por  $und(N)$  é basicamente diferente.

# Sistema Sequencial

## Definition 12

Seja um sistema elementar  $N = (S, T; F, C_0)$ . Este sistema é dito sequencial se e somente se:

- $\|C_0\| = 1$ .
- $\forall C \in |C_0\rangle, \|C\| = 1$ .

# Máquinas de Estado

## Definition 13

Seja um sistema elementar  $N = (S, T; F, C_0)$ . Este sistema é dito uma máquina de estados se e somente se  $\forall t \in T, \|\bullet t\| = \|t\bullet\| = 1$ .

**Se um dado sistema é uma máquina de estado, isto implica que não há conflito ou sincronismo?**



# Grafos marcados e Redes Free Choice

## Definition 14

Seja um sistema elementar  $N = (S, T; F, C_0)$ . Este sistema é dito um grafo marcado se e somente se  $\forall s \in S, \|\bullet s\| = \|s\bullet\| = 1$ .

## Definition 15

Seja um sistema elementar  $N = (S, T; F, C_0)$ . Este sistema é dito uma rede *free choice* se e somente se  $\forall s \in S, \|s\bullet\| = 1$  ou  $\bullet(s\bullet) = \{s\}$ .

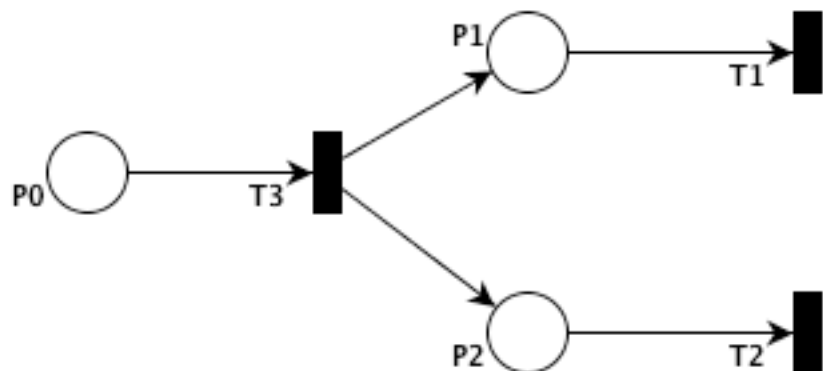
Desel, J. and Esparza, J.; Free Choice Nets, Cambridge University Press, 1995.

# Sincronismo

~~SM~~

MG

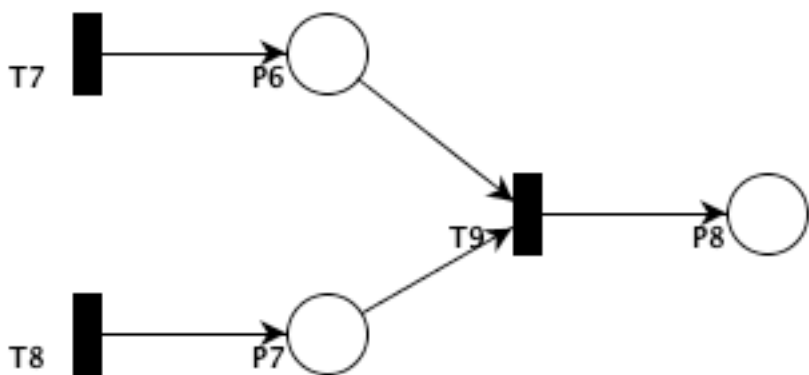
FC



~~SM~~

MG

FC



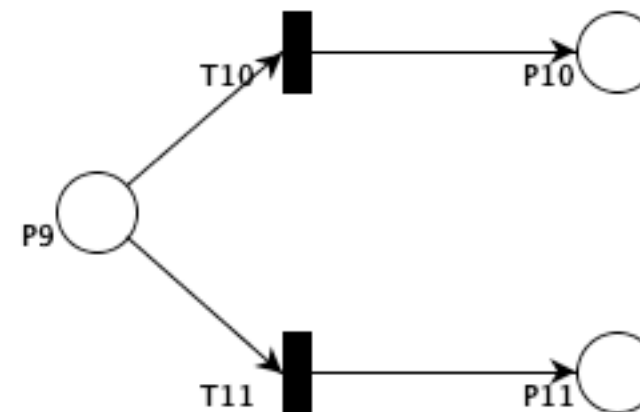
SM - State Machine  
 MG - Marked Graph  
 FC - Free Choice

# Conflito

SM

~~MG~~

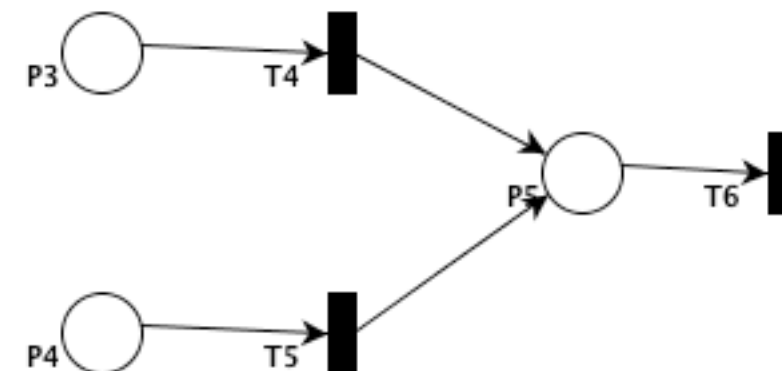
FC



SM

~~MG~~

FC





# Sistemas S e T

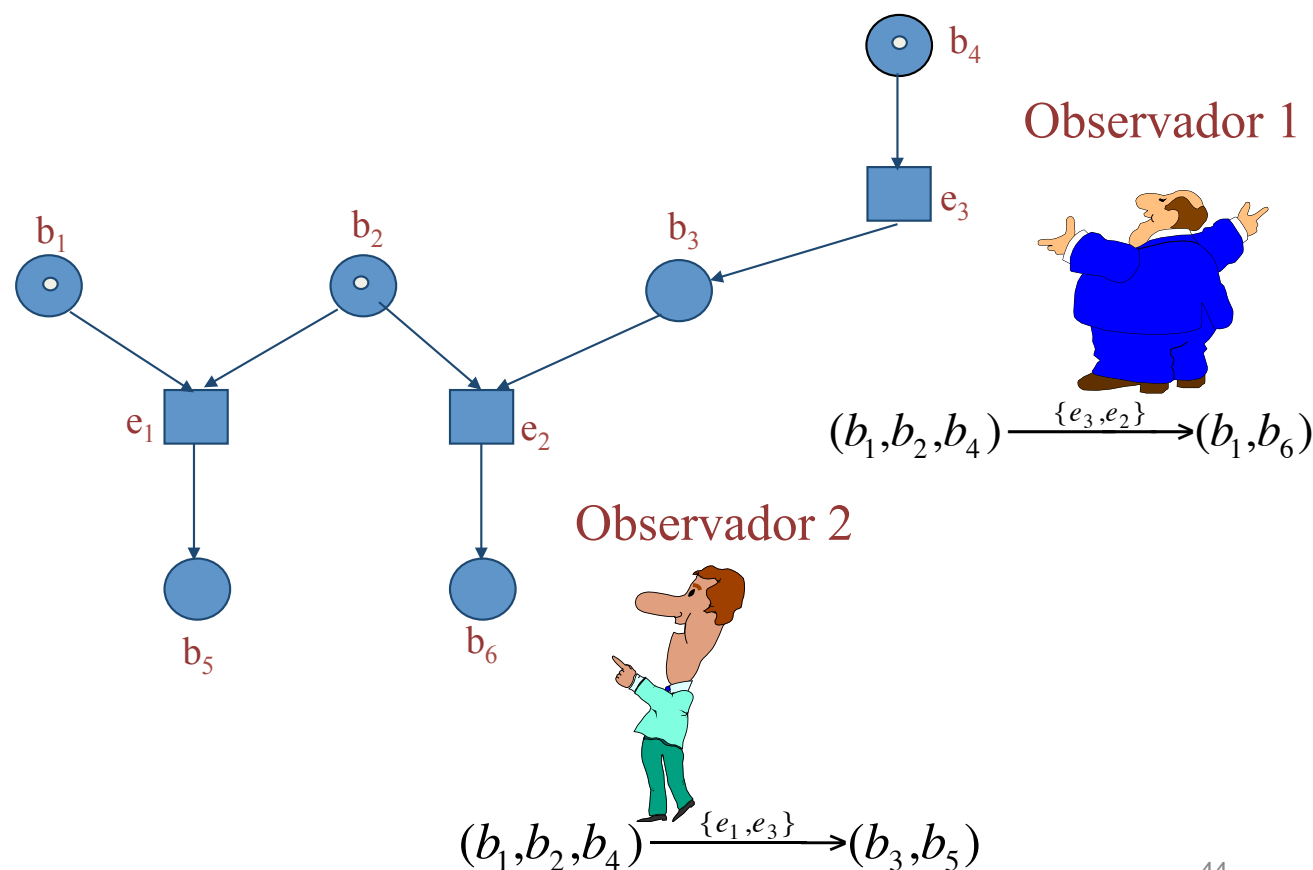
Os S-systems se caracterizam pelo fato de que cada elemento  $s$  tem somente uma pre-condição (um estado elementar) e uma pós-condição (também chamado de grafos marcados).

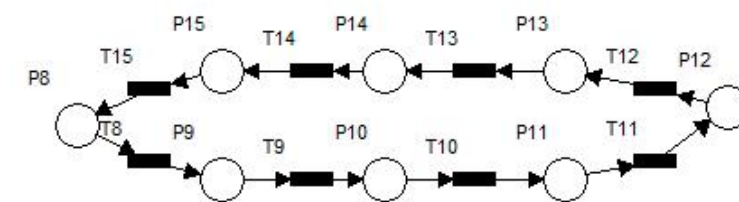
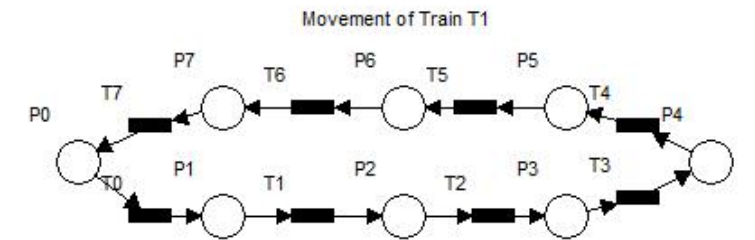
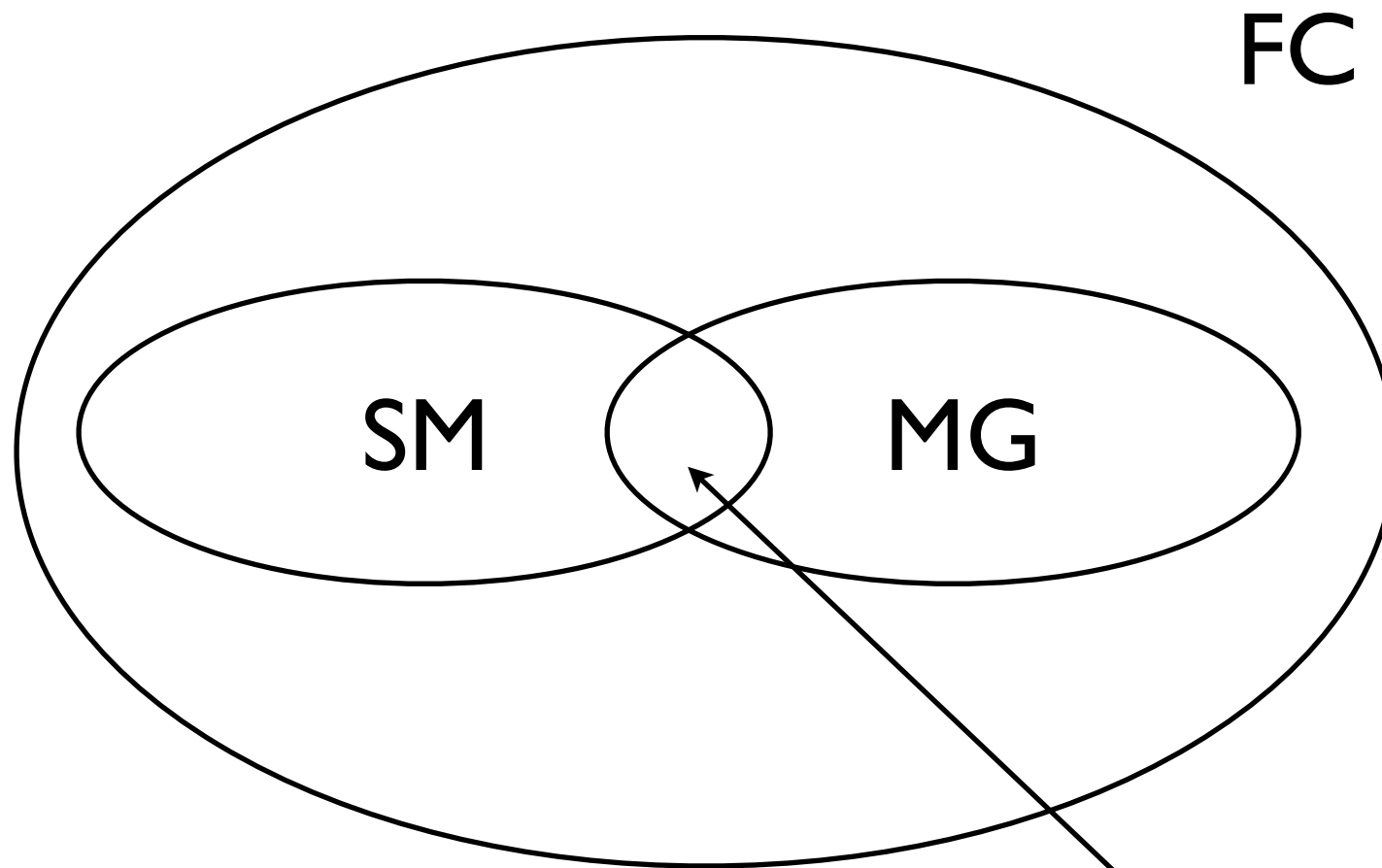
Os T-systems têm uma definição similar só que agora cada elemento  $t$  (transição) tem somente uma pre-condição e uma pós-condição. São as máquinas de estado.

Seja qual for o caso, o estudo destes sistemas é simples e até existem vários sistemas naturais e artificiais que podem ser modelados por um esquema como este. No primeiro caso se admite conflito mas não sincronismos e no segundo se admitem sincronismos mas não conflitos.

# Interpretação para as redes Free Choice

No caso das redes free choice o que acontece é que se admite tanto conflitos como sincronismos mas estes não podem ter nenhuma relação, isto é, devem ocorrer de forma independente. Isto elimina por exemplo os casos de confusão já mencionados.







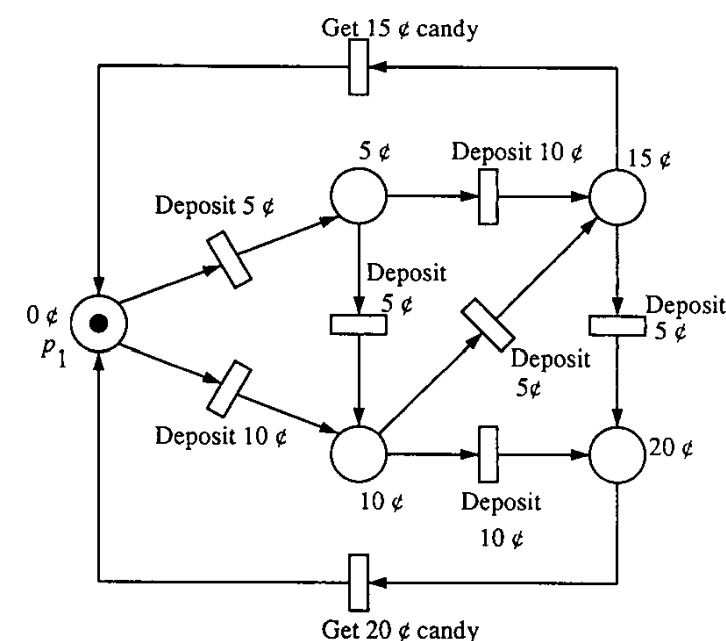
O estudo das redes Free Choice tem ocupado alguns dos grandes nomes entre os pesquisadores da área. Jorg Desel e Javier Esparza publicaram um livro só sobre este assunto, incluso na série Cambridge Tracts in Theoretical Computer Science.

# Exemplo

Quais as características desejáveis de um sistema automatizado?

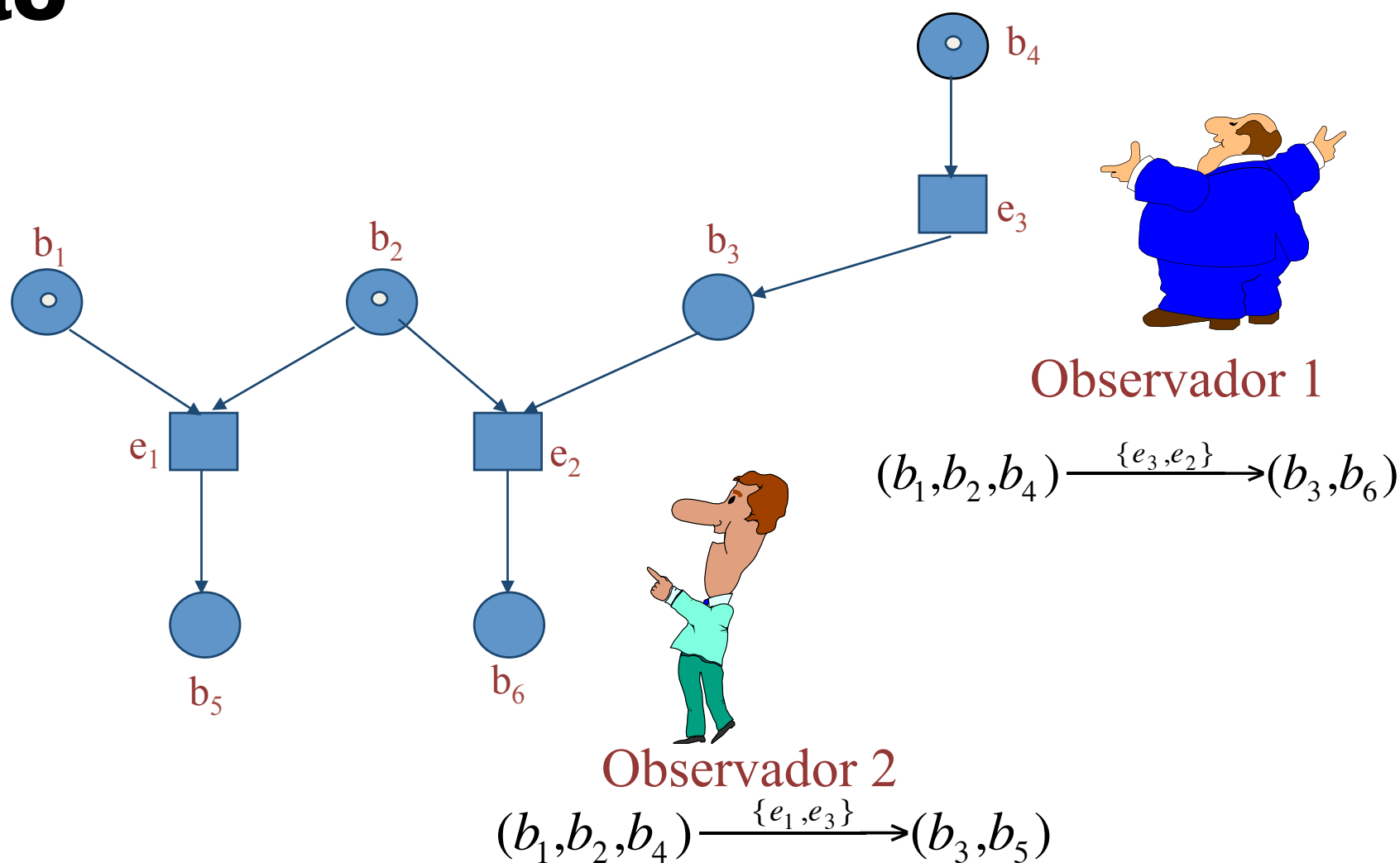
Um exemplo (retirado do artigo de Tadao Murata, Petri Nets: Properties Analysis and Applications), é o mostrado pela rede ao lado, cuja interpretação seria a de uma máquina de vender chocolates, usando moedas de 5c e 10c.

Imagine que os chocolates vendidos custam, respectivamente 15c e 20c. O controle da máquina é preparado para aceitar estes valores e habilitar a liberação somente do chocolate com o custo correspondente.



Extraído de Petri Nets: Properties Analysis and Applications, Tadao Murata, IEEE Proceedings, 1989..

# Desfazendo a confusão





# Contato ou confusão

Dois eventos  $t_1$  e  $t_2$  são ditos em contato (ou confusão) se e somente se ambos estão habilitados e se  $\cdot t_1 \cap t_2 \cdot \neq \phi$  ou se  $t_1 \cdot \cap \cdot t_2 \neq \phi$ .

# Sistema completo

Dado um sistema elementar  $N=(S,T;F, c_0)$ , definiremos como sendo o sistema  $N'$ , S-completo em relação a  $N$ , um outro sistema elementar  $N'=(S',T';F', c_0')$ , onde

i)  $S' = S \cup \bar{S}$ , onde  $\bar{S}$  é o dual de  $S$ , isto é,  
 $\bar{S} = \{\bar{s} \mid \forall s \in S. (\exists \bar{s}. (\bullet s = \bar{s} \bullet) \wedge (s \bullet = \bullet \bar{s})) \text{ e } m(\bar{s}) + m(s) = 1, \text{ onde } m(s) \text{ é a marcação de } s.$

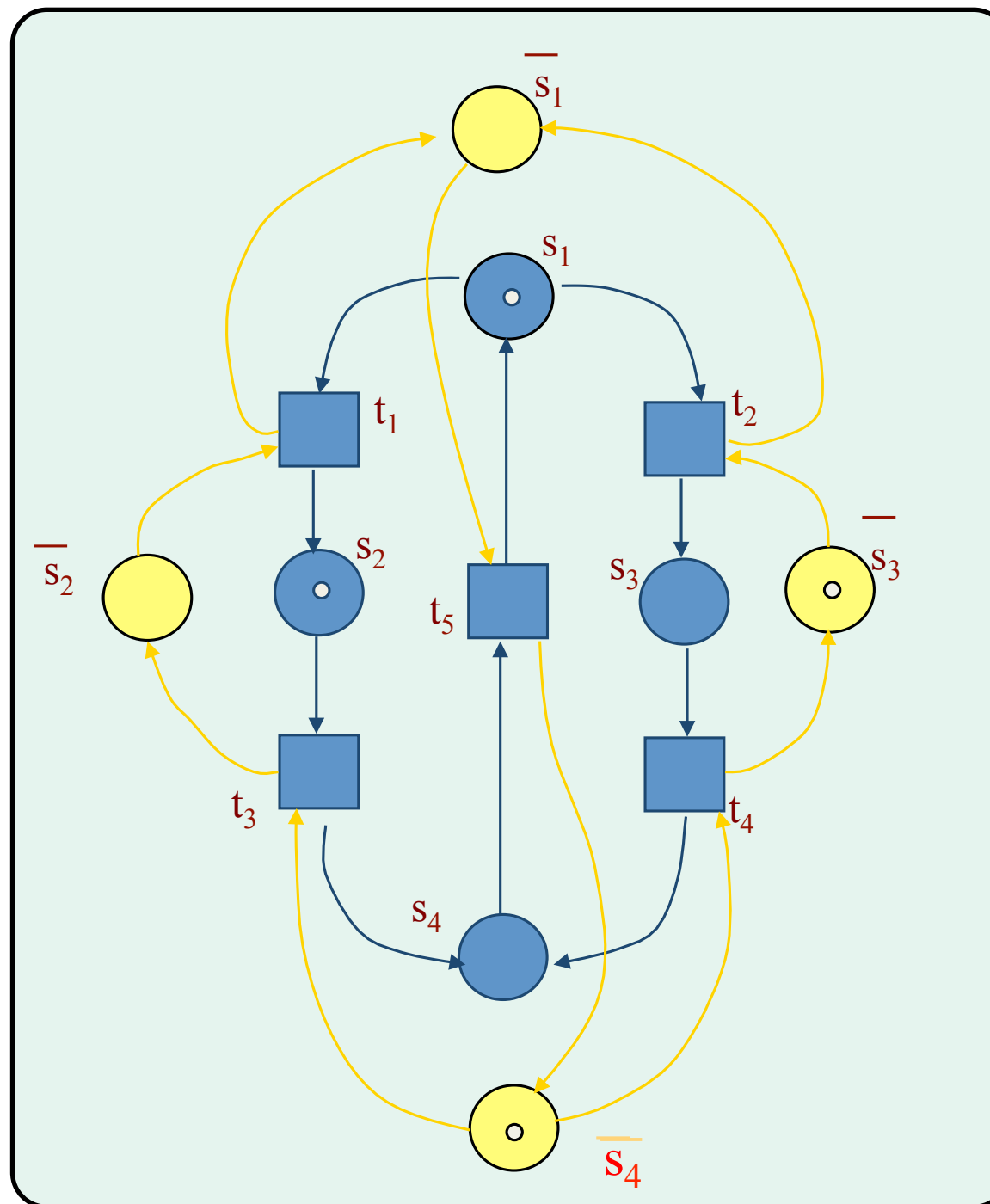
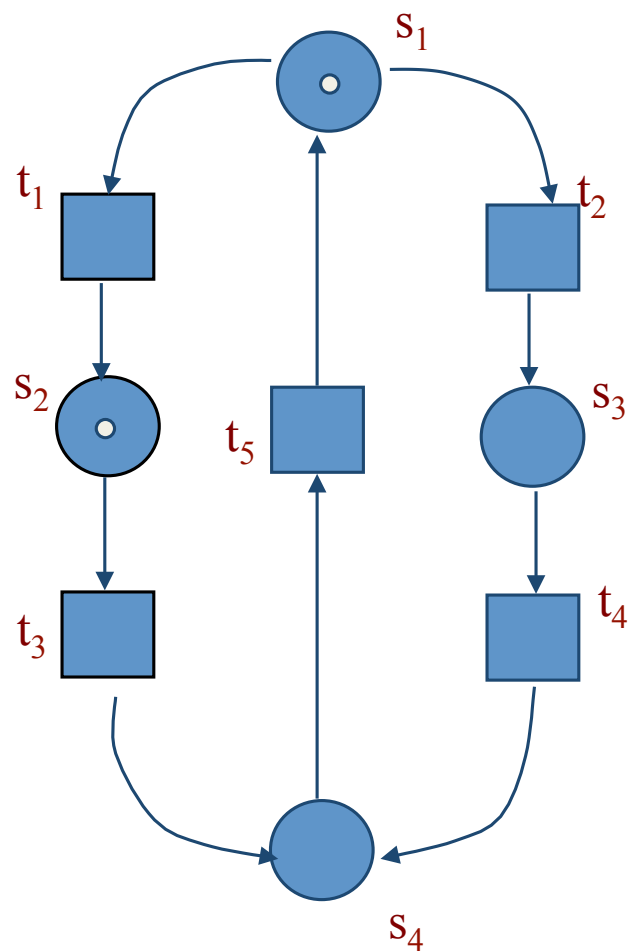
ii)  $T'=T$

*iii)*  $F' = F \cup \bar{F}$ , onde

$$\bar{F} = \{(t, \bar{s}) \mid t \in T \wedge (s, t) \in F\} \cup \{(\bar{s}, t) \mid t \in T \wedge (t, s) \in F\}$$

*iv)*  $c_0' = c_0 \cup \varphi(\bar{S})$  onde  $\varphi(\bar{S}) = \{\bar{s} \in \bar{S} \mid s \notin c_0\}$

# Exemplo



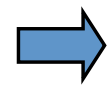
## Teorema 1

Seja um sistema elementar  $N = (S, T; F, C_0)$ . O sistema S-completo  $N' = (S', T'; F', C'_0)$  é livre de contato.

Dem] Lista de exercícios 2

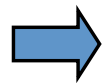
Hints

- ➔ Toda rede  $N$  possui um dual  $N'$
- ➔ Afora o contato, a seq. de eventos em  $N$  é igual a seq. de eventos em  $N'$



$$\forall c \in C_N. \exists c' \in C_{N'} \mid c \subset c'$$

portanto, existe uma bijeção  $\beta : C_{N'} \leftrightarrow C_N$



A seqüência de eventos N' é unívoca no que se refere a contato



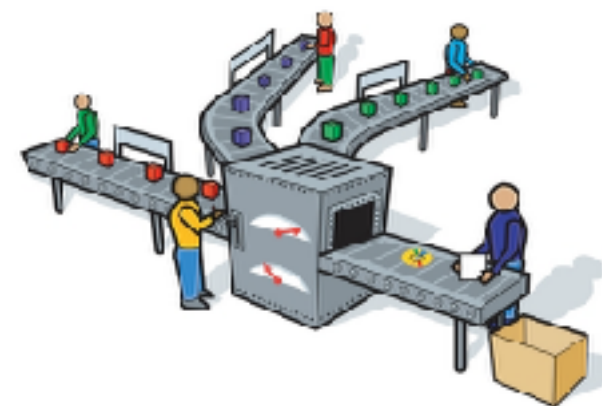
Até aqui o movimento das marcas representaram ações unitárias (aplica-se uma ação de cada vez, com um efeito bastante específico, como inserir uma moeda de 5c ou 10c em um repositório). O número de ações nos exemplos mostrados é sempre pequeno, embora possa ser repetido várias vezes.

Casos como estes são passíveis de serem representados com o que chamamos de Sistemas (ou redes) Elementares. Exemplos mais complexos podem ainda ser vistos desta forma como uma abstração ou análise qualitativa do sistema analisado.

Há casos onde este tipo de análise não é suficiente.

# Os sistemas produtivos

Os sistemas produtivos também se enquadram na categoria que acabamos de descrever, onde existe um estado inicial claramente definido e trata-se de sequenciar ações (não necessariamente um número pequeno) que leva a um estado final também bem definido (onde algum produto é fabricado ou montado). No final do processo o sistema é capaz de retornar ao estado inicial e repetir o mesmo processo novamente, seguindo exatamente os mesmos passos (e manufacturando um produto “igual”).



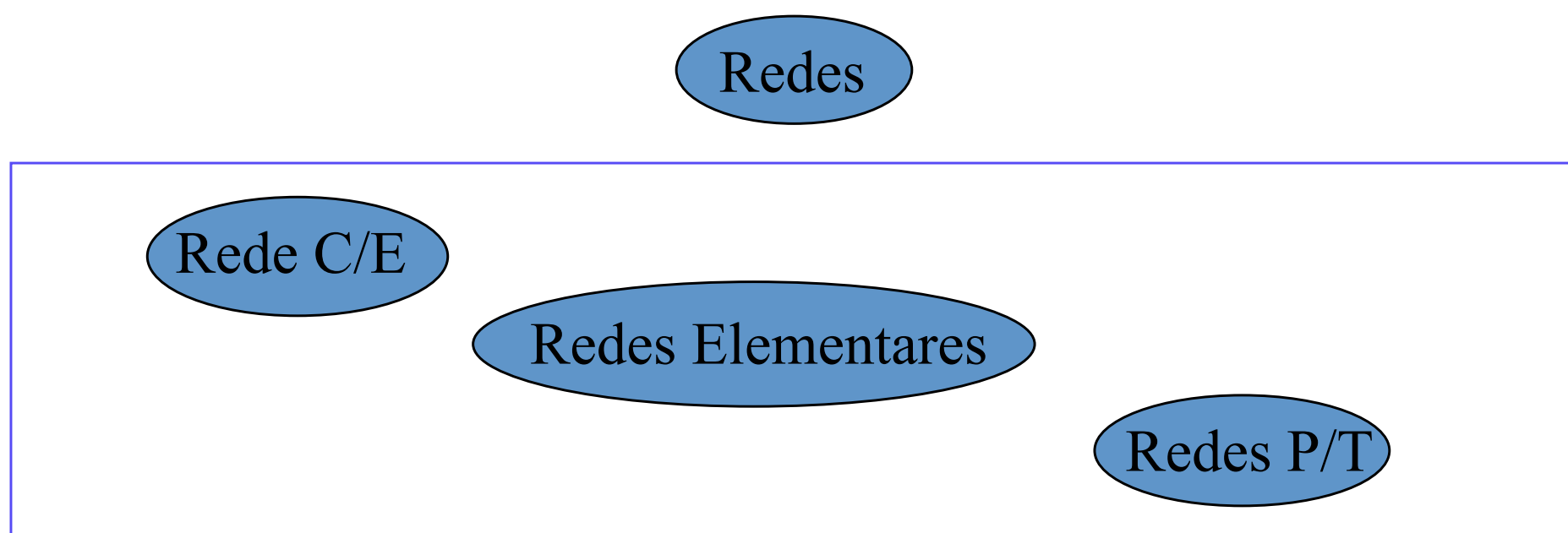
# Buffers

Buffers são usados para regular a velocidade de produção, especialmente quando se tem subprocessos que são mais rápidos que outros, pertencentes a um mesmo processo. Neste caso a modelagem deve ter em conta o número de peças no buffer





# As redes de Petri Clássicas



?

# Redes Place/Transition (P/T)

- número irrestrito ( $w$ ) de marcas em cada lugar
- relações de fluxo não unitárias (peso dos arcos)
- determinação de capacidade dos lugares ( $> 1$ )
- indistinguibilidade das marcas
- geração de estados à partir de um estado inicial

Redes elementares



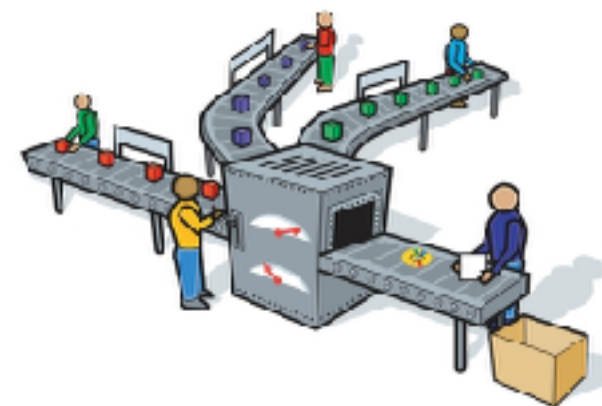
Redes P/T

# Fabricação Flexível

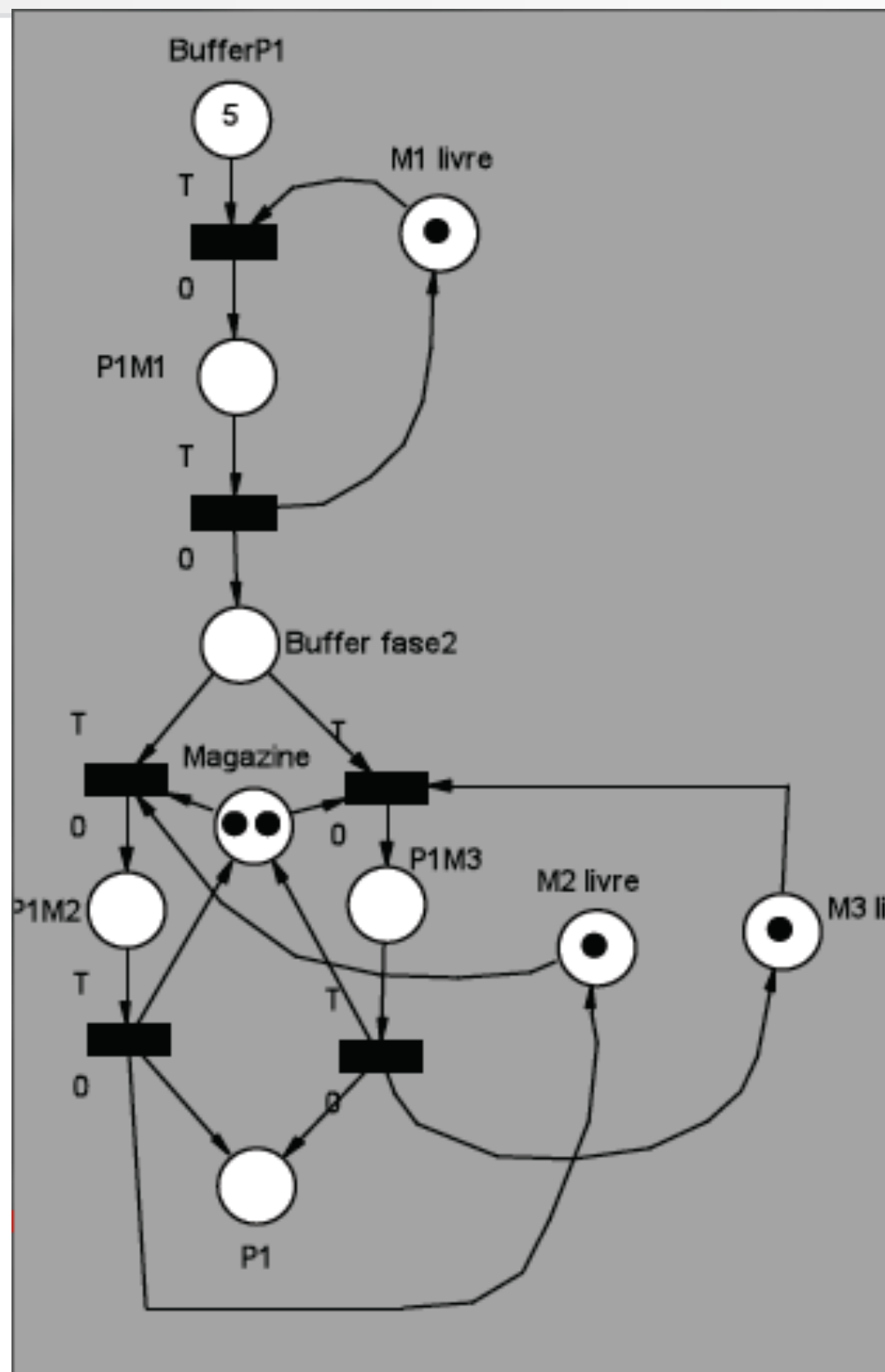
Sejam dois lotes de peças com seqüenciamento de processos distintos, e três máquinas, M1, M2 e M3 onde as duas últimas compartilham o mesmo magazine de ferramentas e executam os mesmos processos:

$P1 \equiv M1; (M2 \vee M3)$

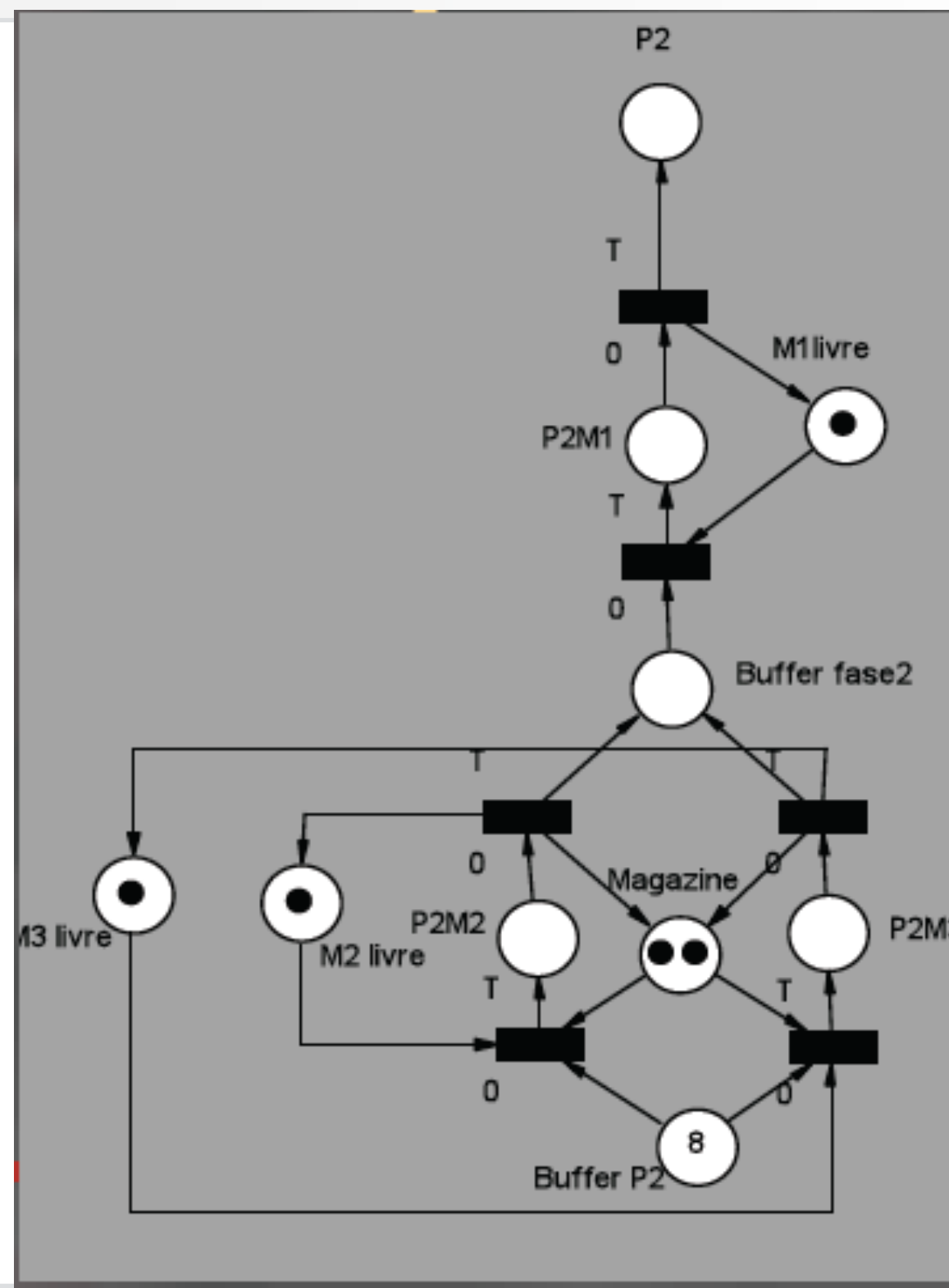
$P2 \equiv (M2 \vee M3); M1$



# Fabricação de P1

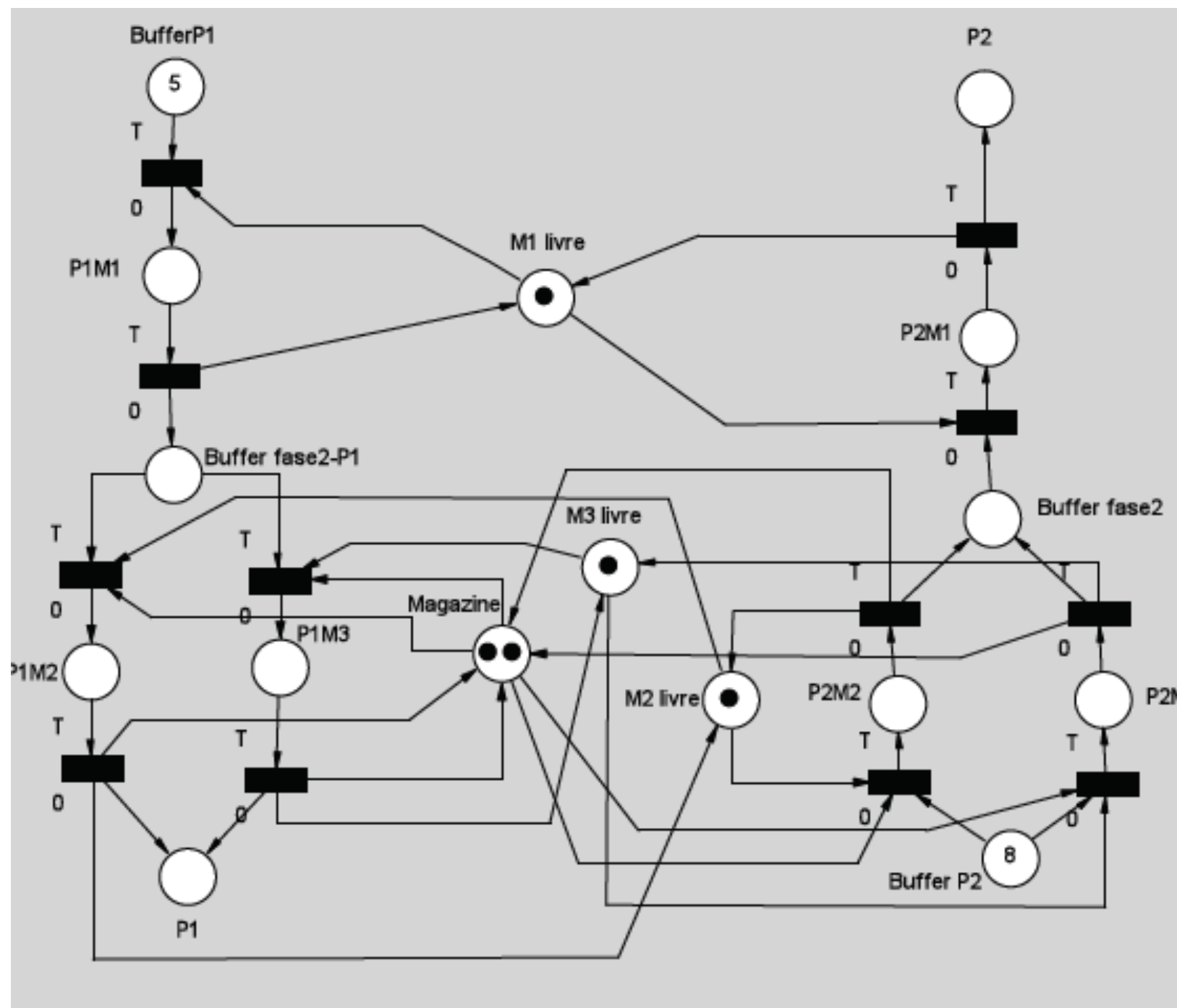


# Fabricação de P2





# Sincronizando P1 e P2



# 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.

## Milestone 2

Para a próxima aula vocês devem agora ter um documento preliminar para o projeto final (artigo) contendo:

1. Título (e não importa se o título for modificado no futuro)
2. Abstract em inglês
3. Introdução com uma explicação um pouco mais detalhada sobre o tema, a motivação e o resultado esperado;
4. Alguma bibliografia preliminar deve ser acrescentada (e lida).

# Leitura da semana

## Petri Nets: Properties, Analysis and Applications

---

TADAO MURATA, FELLOW, IEEE

*Invited Paper*

*This is an invited tutorial-review paper on Petri nets—a graphical and mathematical modeling tool. Petri nets are a promising tool for describing and studying information processing systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic.*

*The paper starts with a brief review of the history and the application areas considered in the literature. It then proceeds with introductory modeling examples, behavioral and structural properties, three methods of analysis, subclasses of Petri nets and their analysis. In particular, one section is devoted to marked graphs—the concurrent system model most amenable to analysis. In addition, the paper presents introductory discussions on stochastic nets with their application to performance modeling, and on high-level nets with their application to logic programming. Also included are recent results on reachability criteria. Suggestions are provided for further reading on many subject areas of these nets.*

to the faculty of Mathematics and Physics at the Technical University of Darmstadt, West Germany. The dissertation was prepared while C. A. Petri worked as a scientist at the University of Bonn. Petri's work [1], [2] came to the attention of A. W. Holt, who later led the Information System Theory Project of Applied Data Research, Inc., in the United States. The early developments and applications of Petri nets (or their predecessor) are found in the reports [3]–[8] associated with this project, and in the Record [9] of the 1970 Project MAC Conference on Concurrent Systems and Parallel Computation. From 1970 to 1975, the Computation Structure Group at MIT was most active in conducting Petri-net related research, and produced many reports and theories



*Fim*