

FVOCA

Módulo 1 (Aula Teórica)

Modelação e Análise Verificação e Validação de
Sistemas Computacionais - RdP

Eduardo Tovar <emt@isep.ipp.pt>

Índice

1. Introdução

- 2. Noções Básicas de Redes de Petri (RdP)
- 3. Regras de Evolução das RdP
- 4. RdP Generalizadas
- 5. Componentes de Modelação em RdP
- 6. Análise Computacional de Modelos RdP
- 7. Verificação de Propriedades dos Sistemas Modelados
- 8. Utilização da Ferramenta HP-SIM

Introdução (1)

- modernos sistemas computacionais têm:
 - crescente complexidade (e distribuição)
 - crescente exigência de confiança no funcionamento e de qualidade de serviço
- daí a necessidade de:
 - utilizar ferramentas e métodos rigorosos nas fases de especificação, teste e análise de desempenho do sistema
 - assegurar, logo na fase de concepção e especificação, elevada probabilidade de o sistema estar isento de erros de concepção

Introdução (2)

- ferramentas formais de modelação de sistemas de eventos discretos que permitam
 - especificação formal e não ambígua do sistema
 - grande capacidade de modelação
 - mecanismos de concorrência e sincronização
 - paralelismo
 - fluxo condicional e repetitivo
 - temporização
 - etc.
 - facilidade de validação do modelo e, mais importante, de validação, análise e teste do sistema modelado
 - ergonomia de utilização (também gráfica), e facilidade de exploração computacional do modelo

Introdução (3)

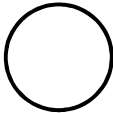




- estamos a falar de Redes de Petri (RdP)
 - introduzidas em 1962 por Carl Adam Petri (Alemanha)
 - Petri, C. A., “Fundamentals of a Theory of Asynchronous Information Flow”, Proc. of IFIP Congress 62. --- Amsterdam: North Holland Publ. Comp., 1963, Pages: 386-390
 - Petri, C. A., “Kommunikation mit Automaten”, Bonn: Institut für Instrumentelle Mathematik, Schriften des IIM Nr. 2, 1962, Second Edition:, New York: Griffiss Air Force Base, Technical Report RADC-TR-65--377, Vol.1, 1966, Pages: Suppl. 1, English translation
 - na web (página de repositório de RdPs):
 - <http://www.informatik.uni-hamburg.de/TGI/PetriNets/>

Índice

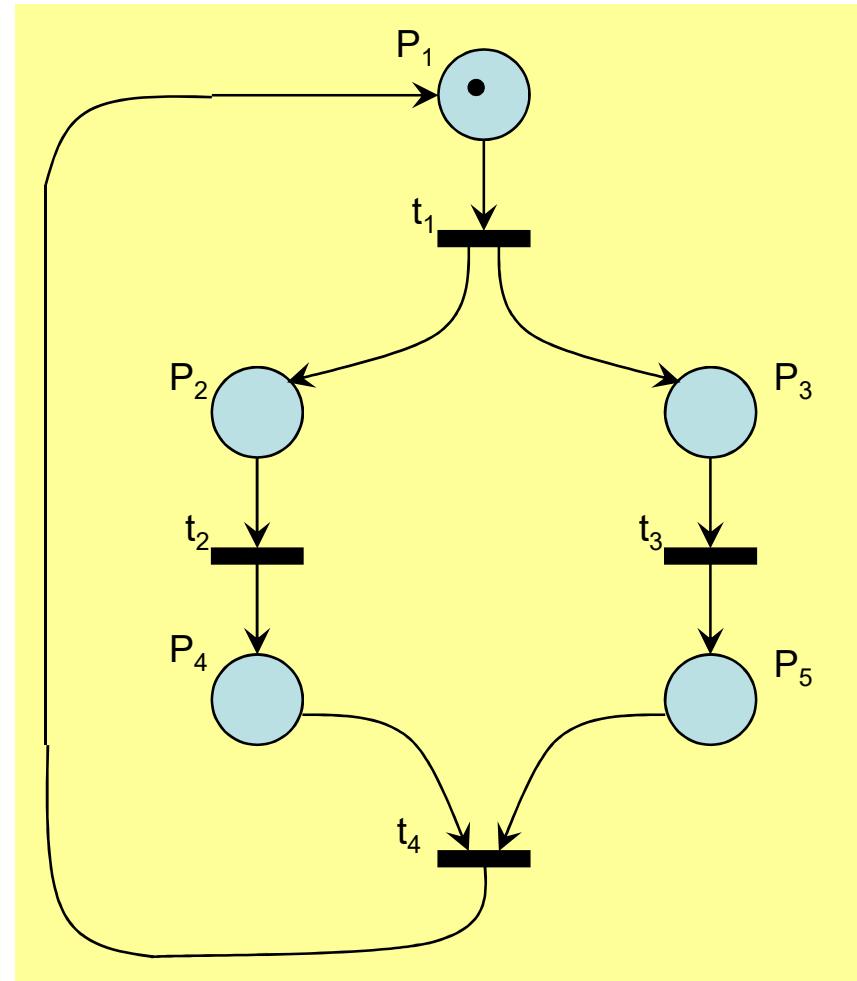
1. Introdução ✓
- 2. Noções Básicas de Redes de Petri (RdP)**
3. Regras de Evolução das RdP
4. RdP Generalizadas
5. Componentes de Modelação em RdP
6. Análise Computacional de Modelos RdP
7. Verificação de Propriedades dos Sistemas Modelados
8. Utilização da Ferramenta HP-SIM

Noções Básicas de RdP (1)

- elementos de definição de uma RdP
 - posição, representada pelo símbolo: 
 - transição, representada pelo símbolo: 
 - marca (ou testemunho), representada pelo símbolo: •
 - arco (orientado), representado por: 

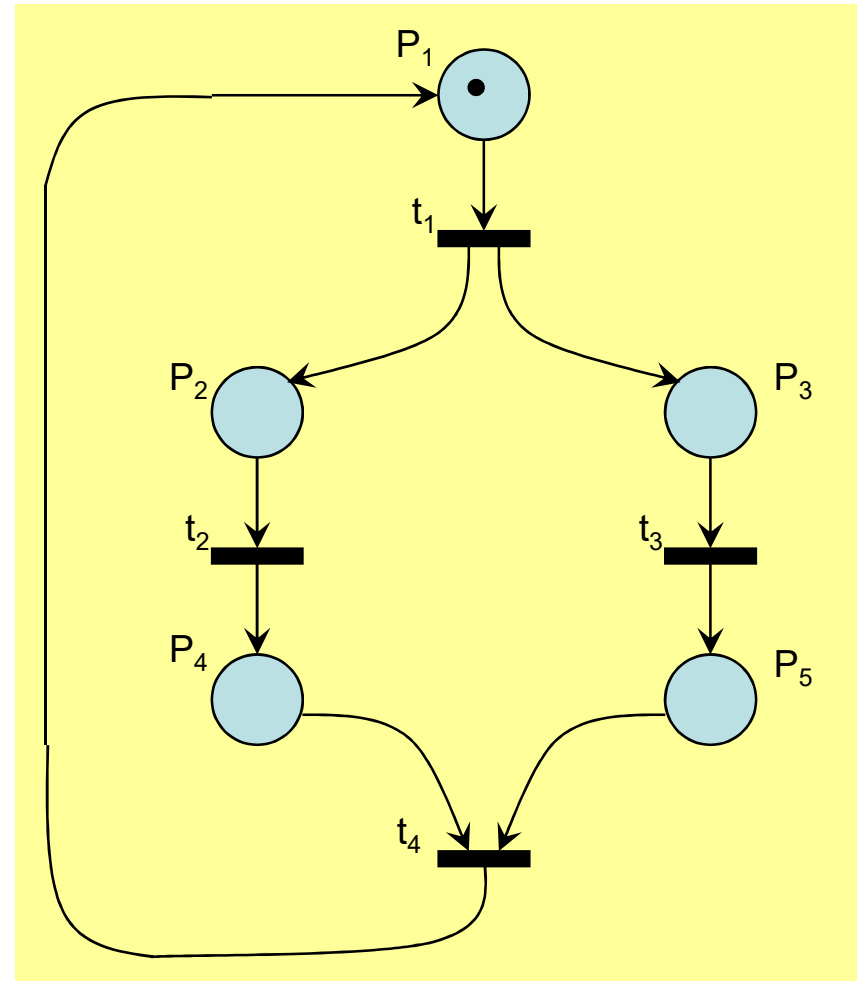
Noções Básicas de RdP (2)

- uma posição (P_x) pode ser interpretada como:
 - uma condição; um estado de espera; um estado provisório, um recurso; uma posição geográfica; etc.
- uma transição (t_x) corresponde a uma ocorrência ou acontecimento (evento)
- uma marca pode representar:
 - uma condição satisfeita; um objecto está presente; um recurso disponível; etc.



Noções Básicas de RdP (3)

- as posições e as transições são interligadas por arcos:
 - que são orientados
 - ligam uma posição a uma transição ou uma transição a uma posição
 - têm obrigatoriamente de ter uma posição ou uma transição nas extremidades



Noções Básicas de RdP (4)

- neste exemplo de RdP, que modela um sistema, existem:

- 5 posições:

$$P = \{P_1, P_2, P_3, P_4, P_5\}$$

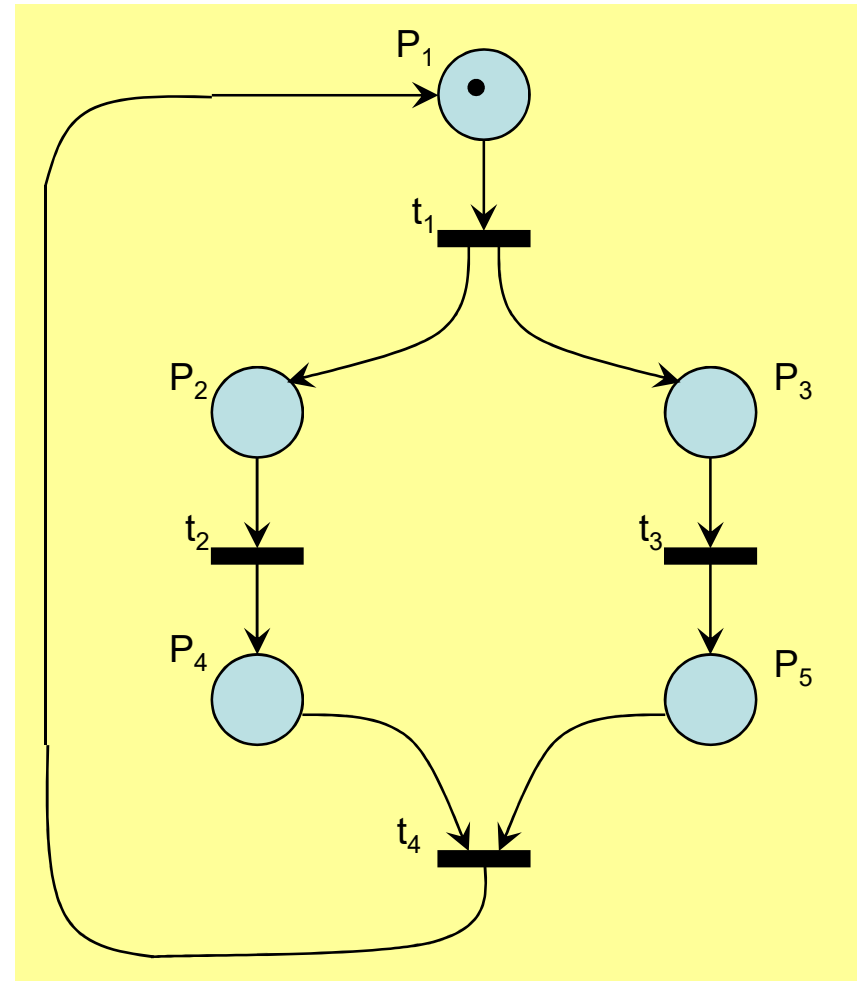
- 4 transições:

$$t = \{t_1, t_2, t_3, t_4\}$$

- e uma marcação inicial (o vector de marcação):

$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{matrix}$$

- a marcação (número de marcas em cada posição) define, para um determinado instante, o estado do sistema



Noções Básicas de RdP (5)

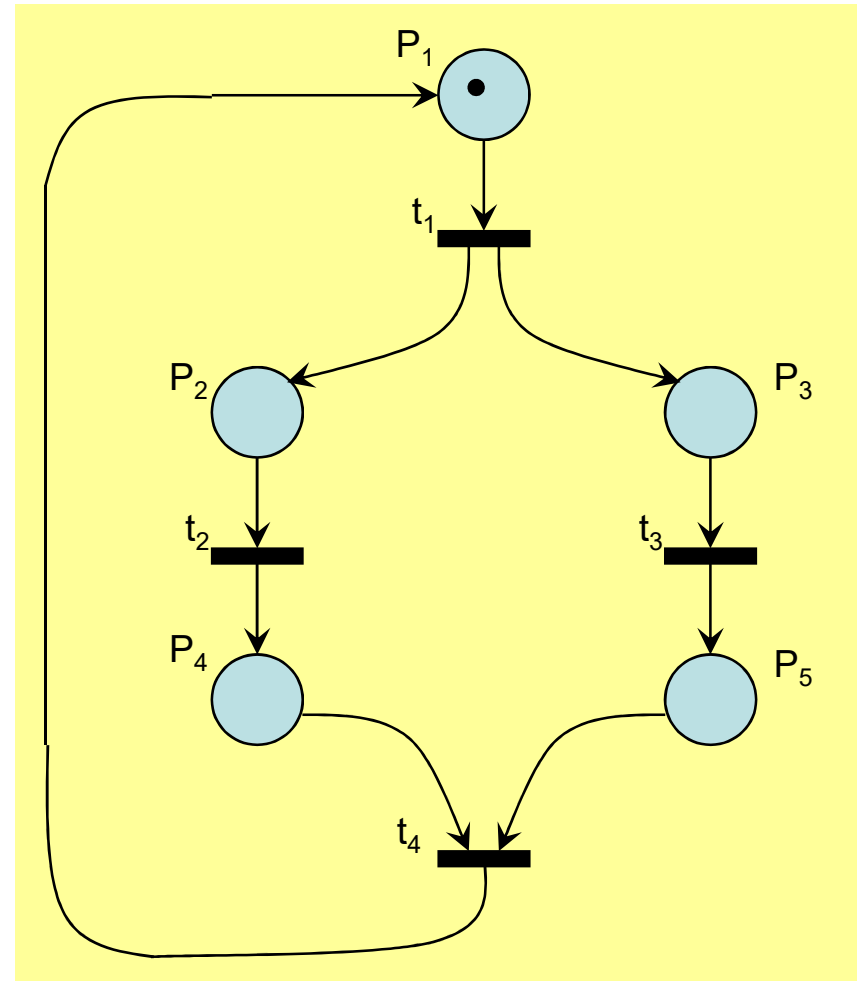
- relativamente às interligações entre as posições e transições:

- matriz de incidência anterior (W^-) às transições (arcos $P \rightarrow t$):

$$W^- = \begin{array}{cccc|l} t_1 & t_2 & t_3 & t_4 & \\ \hline 1 & 0 & 0 & 0 & P_1 \\ 0 & 1 & 0 & 0 & P_2 \\ 0 & 0 & 1 & 0 & P_3 \\ 0 & 0 & 0 & 1 & P_4 \\ 0 & 0 & 0 & 1 & P_5 \end{array}$$

- matriz de incidência posterior (W^+) às transições (arcos $t \rightarrow P$):

$$W^+ = \begin{array}{cccc|l} t_1 & t_2 & t_3 & t_4 & \\ \hline 0 & 0 & 0 & 1 & P_1 \\ 1 & 0 & 0 & 0 & P_2 \\ 1 & 0 & 0 & 0 & P_3 \\ 0 & 1 & 0 & 0 & P_4 \\ 0 & 0 & 1 & 0 & P_5 \end{array}$$



Noções Básicas de RdP (6)

- as matrizes de incidência W^- e W^+ e o vector de marcação inicial M_0 definem uma RdP, e constituem uma alternativa à representação gráfica
 - considere a seguinte RdP:

$$W^- = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad W^+ = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- qual o modelo gráfico?

Noções Básicas de RdP (7)

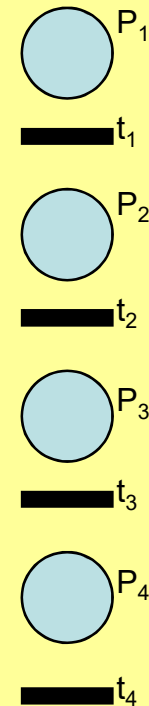
- as matrizes de incidência W^- e W^+ e o vector de marcação inicial M_0 definem uma RdP
 - passos para o modelo gráfico:

$$W^- = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad W^+ = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

solução:

existem 4 posições e 4 transições:



Noções Básicas de RdP (8)

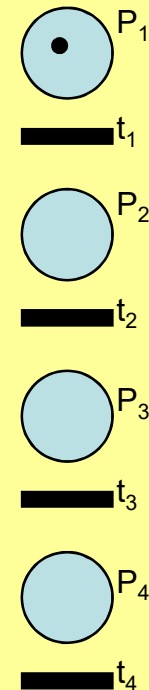
- as matrizes de incidência W^- e W^+ e o vector de marcação inicial M_0 definem uma RdP
 - passos para o modelo gráfico:

$$W^- = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad W^+ = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

solução:

a partir da marcação inicial M_0 :



Noções Básicas de RdP (9)

- as matrizes de incidência W^- e W^+ e o vector de marcação inicial M_0 definem uma RdP
 - passos para o modelo gráfico:

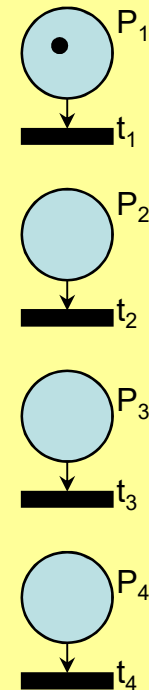
$$W^- = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$W^+ = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

solução:

utilizando a informação de W^- :



Noções Básicas de RdP (10)

- as matrizes de incidência W^- e W^+ e o vector de marcação inicial M_0 definem uma RdP
 - passos para o modelo gráfico:

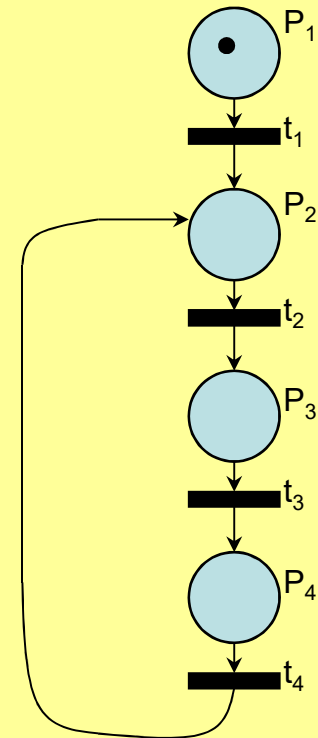
$$W^- = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$W^+ = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

solução:

mais a informação de W^- :



Índice

1. Introdução ✓
2. Noções Básicas de Redes de Petri (RdP) ✓
- 3. Regras de Evolução das RdP**
4. RdP Generalizadas
5. Componentes de Modelação em RdP
6. Análise Computacional de Modelos RdP
7. Verificação de Propriedades dos Sistemas Modelados
8. Utilização da Ferramenta HP-SIM

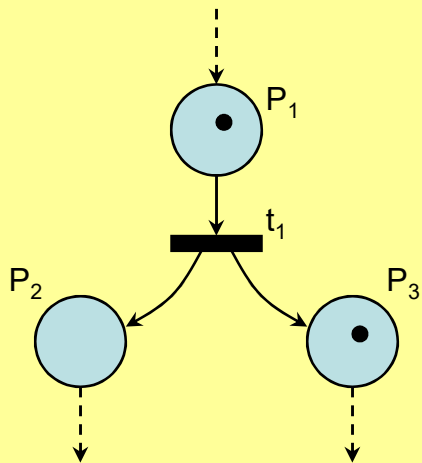
Regras de Evolução das RdP (1)

- numa RdP, a marcação (n.º de marcas em cada posição) define o estado do sistema modelado num determinado instante
- a evolução de estado do sistema corresponde, no modelo, à evolução da marcação da RdP
- a evolução dá-se por “disparo” de uma transição

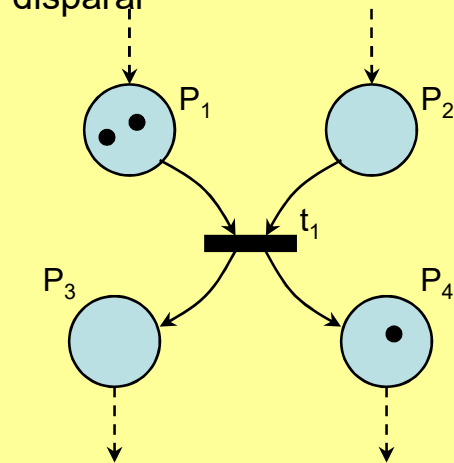
Regras de Evolução das RdP (2)

- **REGRA 1** (condição para o disparo de uma transição):
 - uma transição está habilitada (pode disparar) por uma determinada marcação se, e só se, todas as posições anteriores à transição têm pelo menos uma marca
 - exemplos:

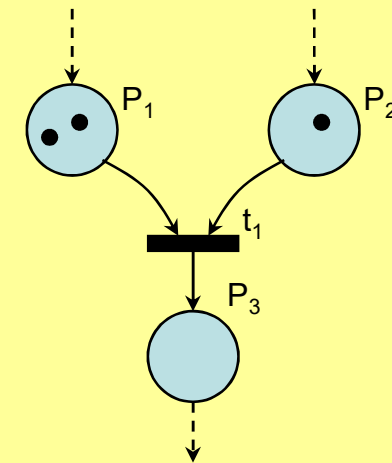
Ex. 1: t_1 pode disparar



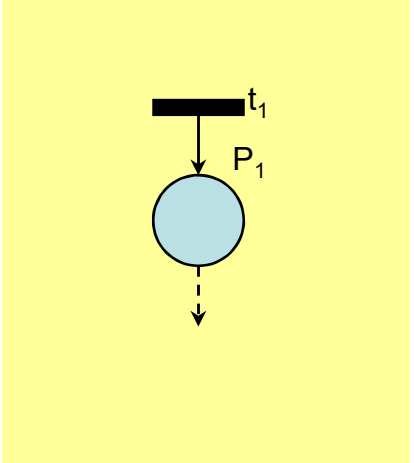
Ex. 2: t_1 **não** pode disparar



Ex. 3: t_1 pode disparar

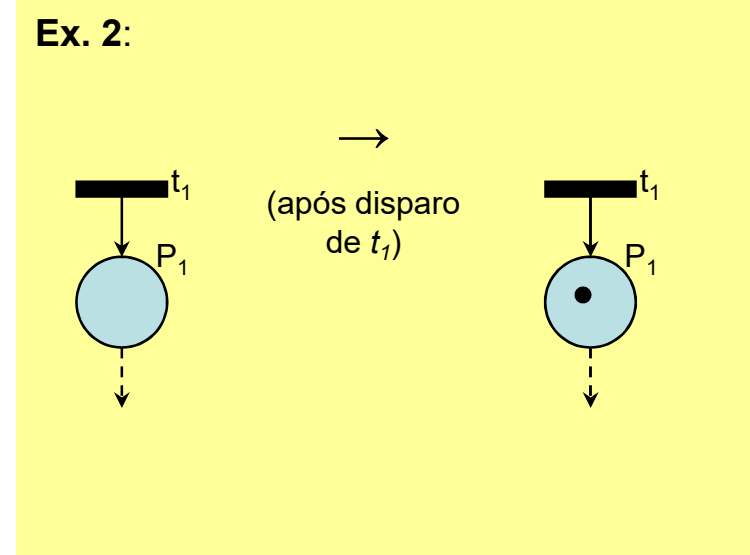
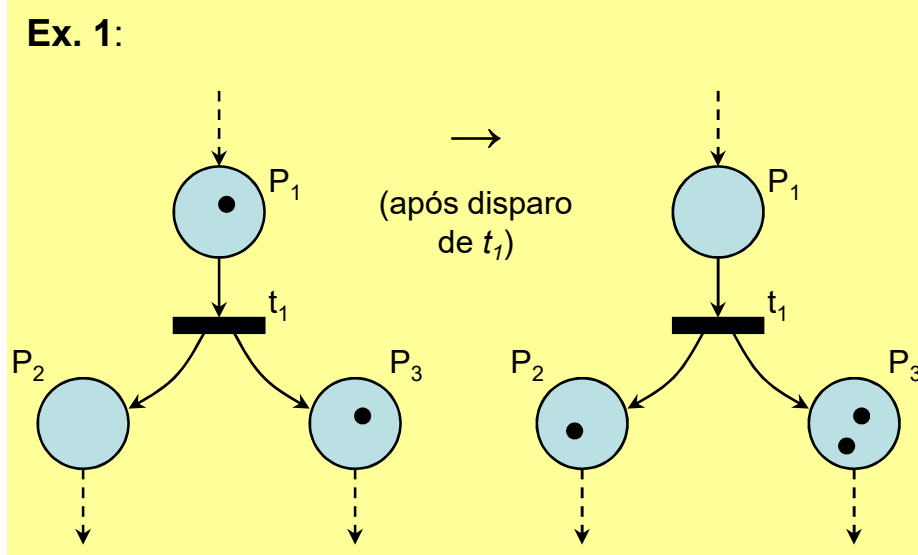


Ex. 4: t_1 pode disparar



Regras de Evolução das RdP (3)

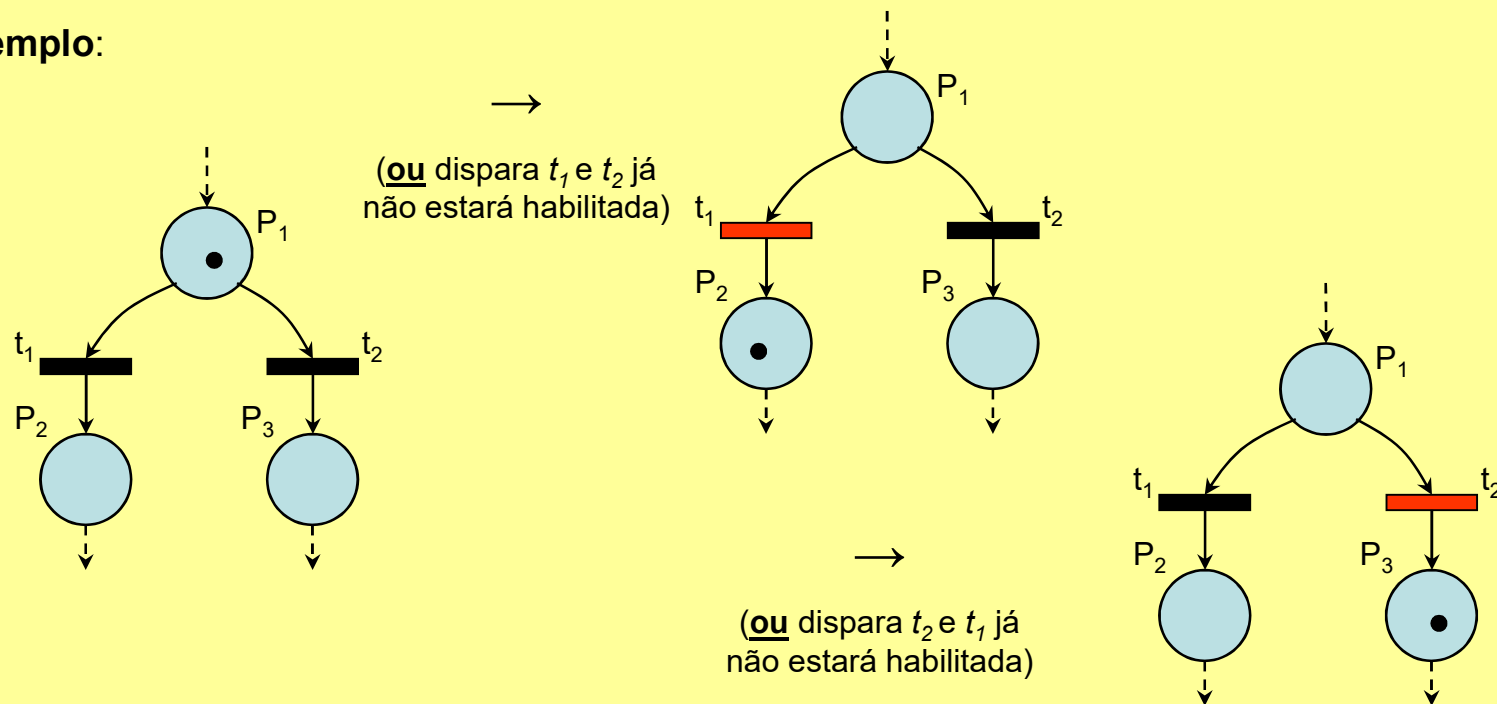
- **REGRA 2** (evolução da marcação após disparo):
 - após o disparo de uma transição:
 - o número de marcas de cada posição anterior à transição diminui em uma unidade
 - e o número de marcas de cada posição posterior à transição umenta em uma unidade
 - exemplos:



Regras de Evolução das RdP (4)

- **REGRA (3)** (eventos discretos):
 - num determinado instante só pode disparar uma transição (o disparo de uma transição – e a correspondente evolução da rede – corresponde a tempo nulo)

Exemplo:



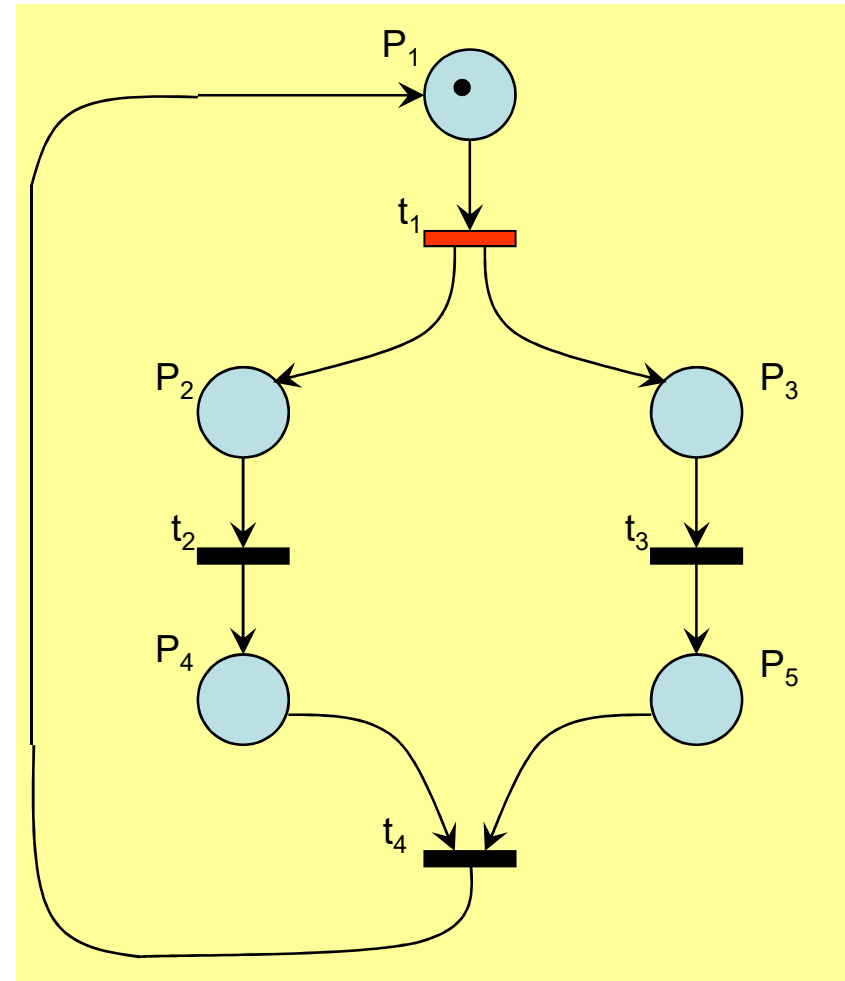
Regras de Evolução das RdP (5)

- voltando a este exemplo de RdP

– a marcação inicial é:

$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- t_1 é a única transição habilitada, pelo que só t_1 pode disparar

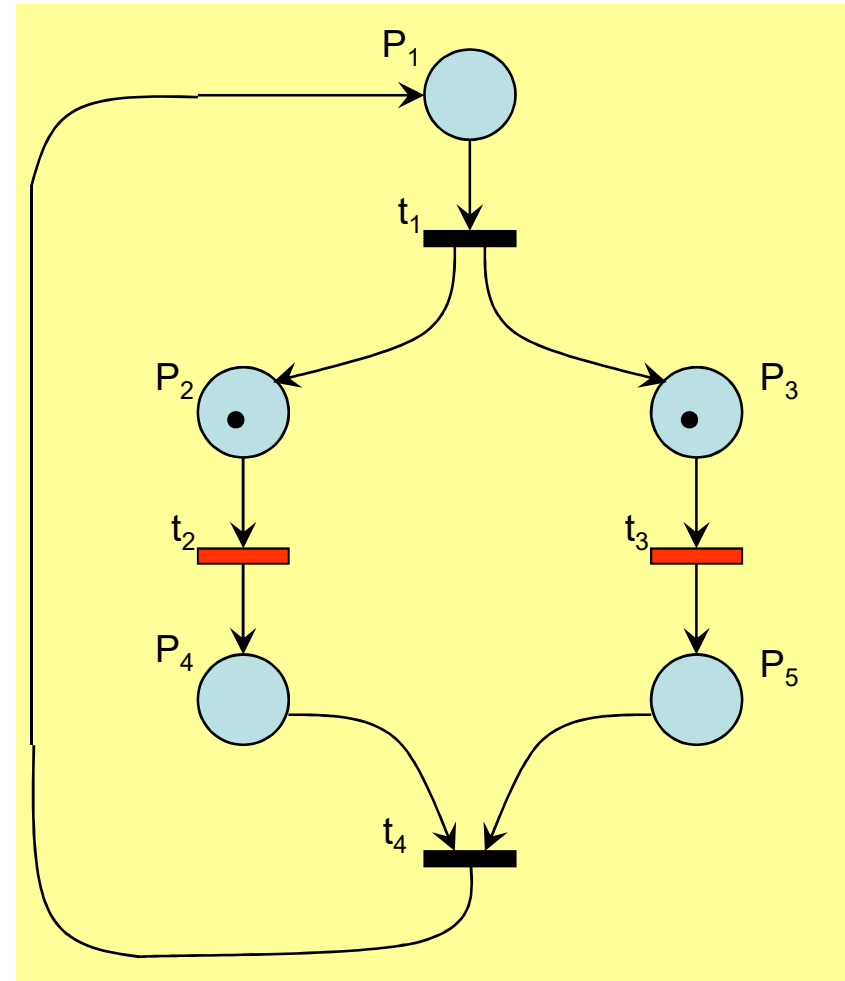


Regras de Evolução das RdP (6)

- após o disparo de t_1 a marcação resultante é a seguinte:

$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{t_1} M_1 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

- t_2 e t_3 passam a estar habilitadas

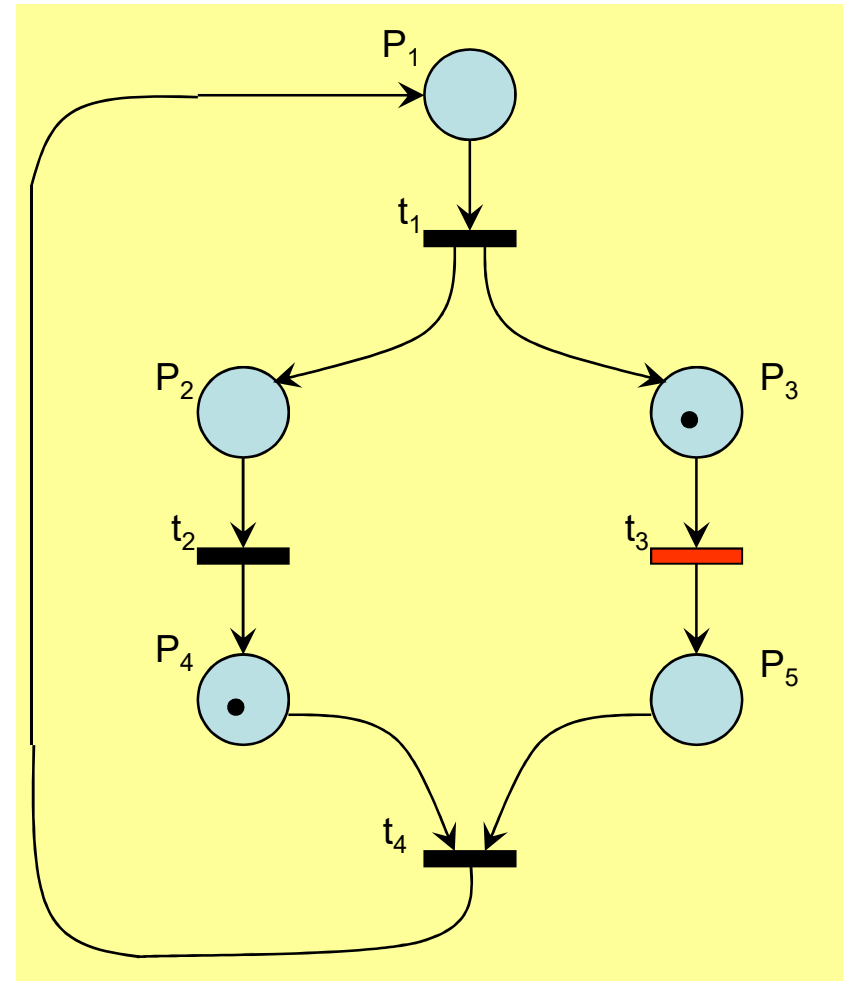


Regras de Evolução das RdP (7)

- se para M_1 for t_2 a disparar, a marcação resultante é a seguinte:

$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{t_1} M_1 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{t_2} M_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

- só t_3 passa a estar habilitada a partir de M_2

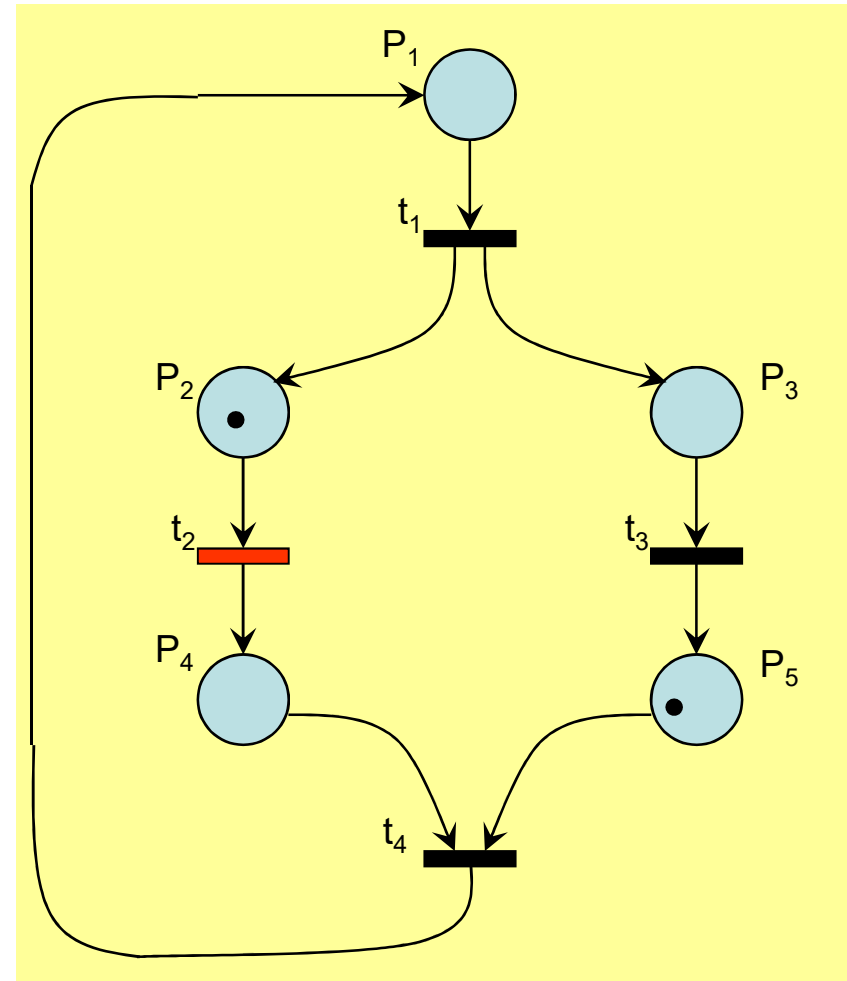


Regras de Evolução das RdP (8)

- se para M_1 for t_2 a disparar, a marcação resultante é a seguinte:

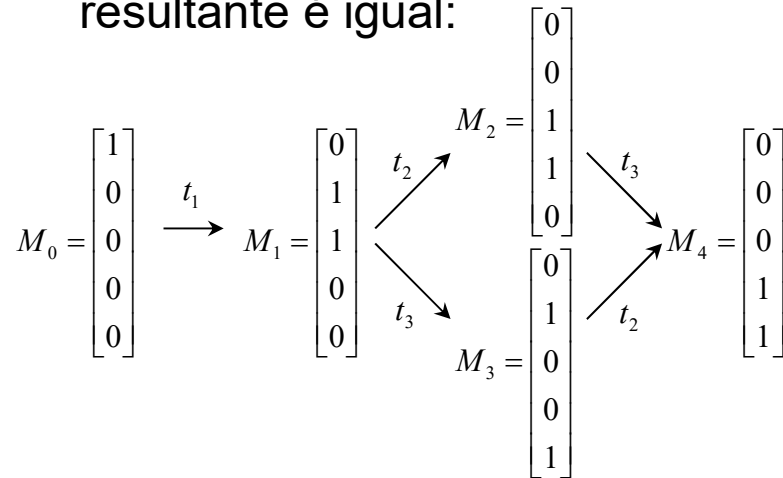
$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{t_1} M_1 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{t_3} M_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

- só t_2 passa a estar habilitada a partir de M_3

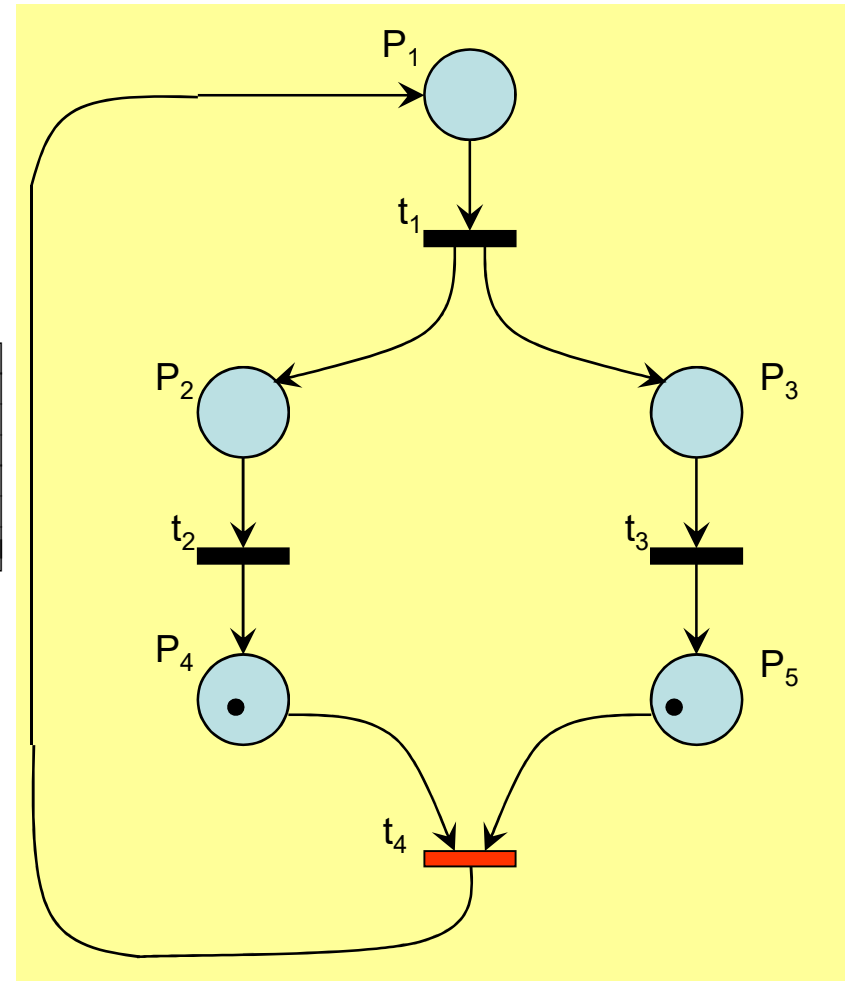


Regras de Evolução das RdP (9)

- a partir de M_3 só t_2 pode disparar, e a partir de M_2 só t_3 pode disparar, e a marcação resultante é igual:

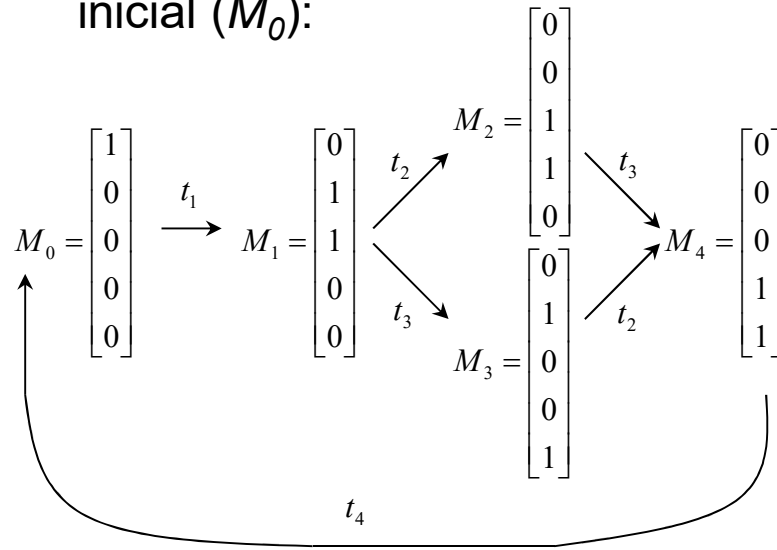


- só t_4 passa a estar habilitada a partir de M_4

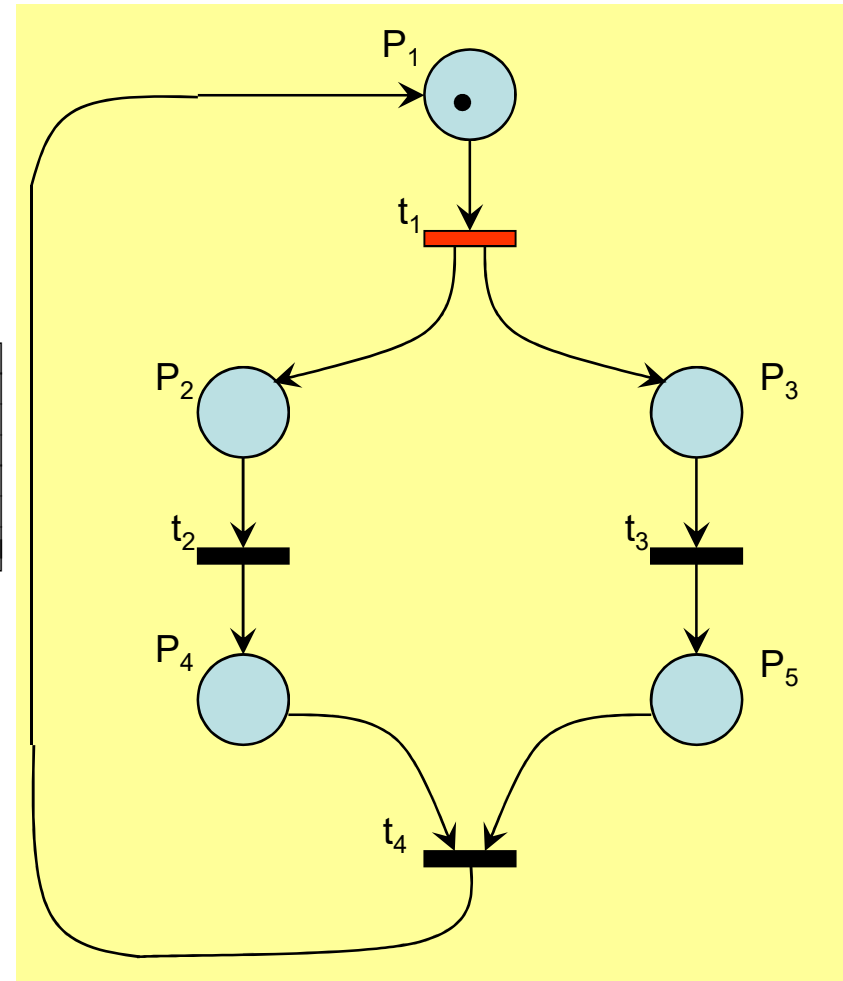


Regras de Evolução das RdP (10)

- a partir de M_4 só t_4 pode disparar, e obtém-se uma marcação igual à marcação inicial (M_0):



- o que obtivemos foi um gráfico de marcações acessíveis

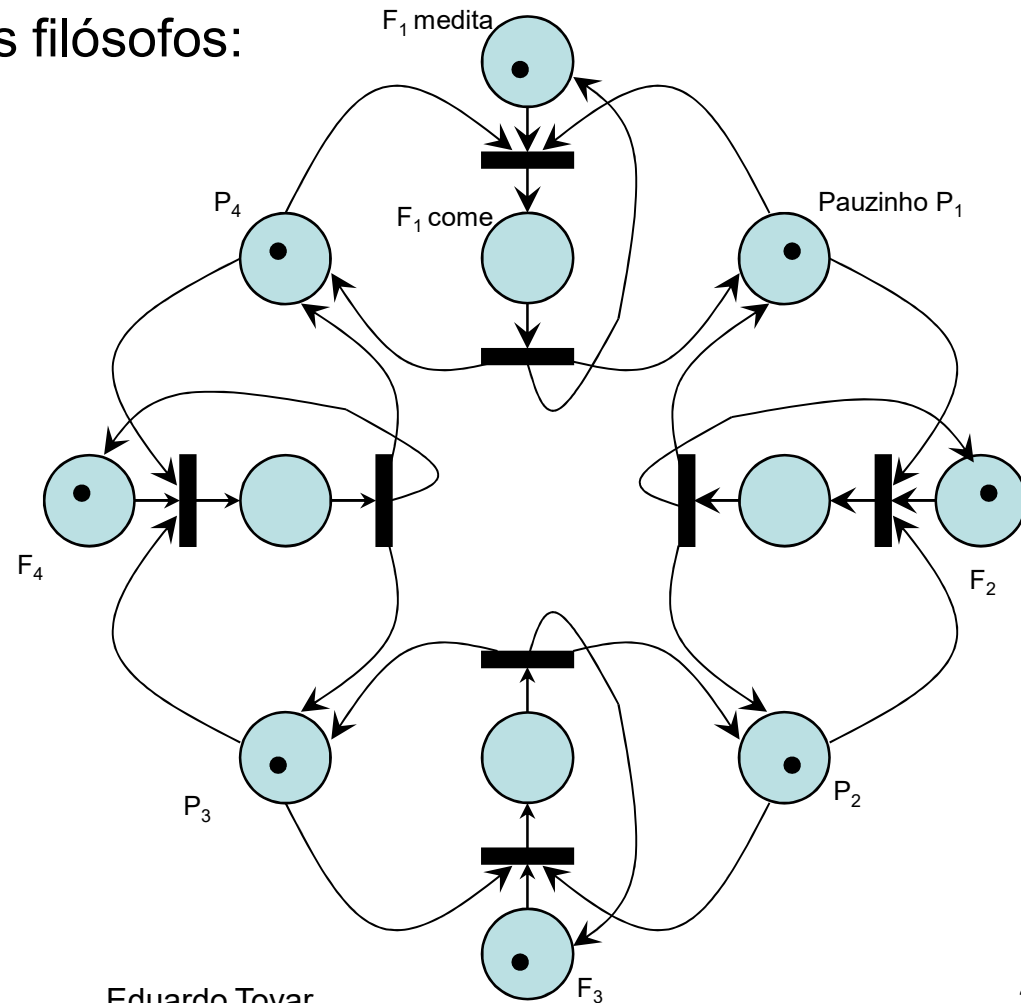


Regras de Evolução das RdP (11)

- Modelo RdP de um Sistema: EXEMPLO A
 - a refeição dos filósofos (Dijkstra's *Dining Philosophers*):
 - filósofos foram a um congresso
 - parte (mais) importante do congresso: jantar de gala
 - » comida chinesa
 - » mesas de quatro
 - » utilização de pauzinhos chineses (*hashis*) para comer
 - » só quatro pauzinhos por mesa, um entre cada filósofo
 - » os filósofos ou meditam, ou comem (se o pauzinho de cada lado – partilhado com o vizinho de mesa – estiver disponível)

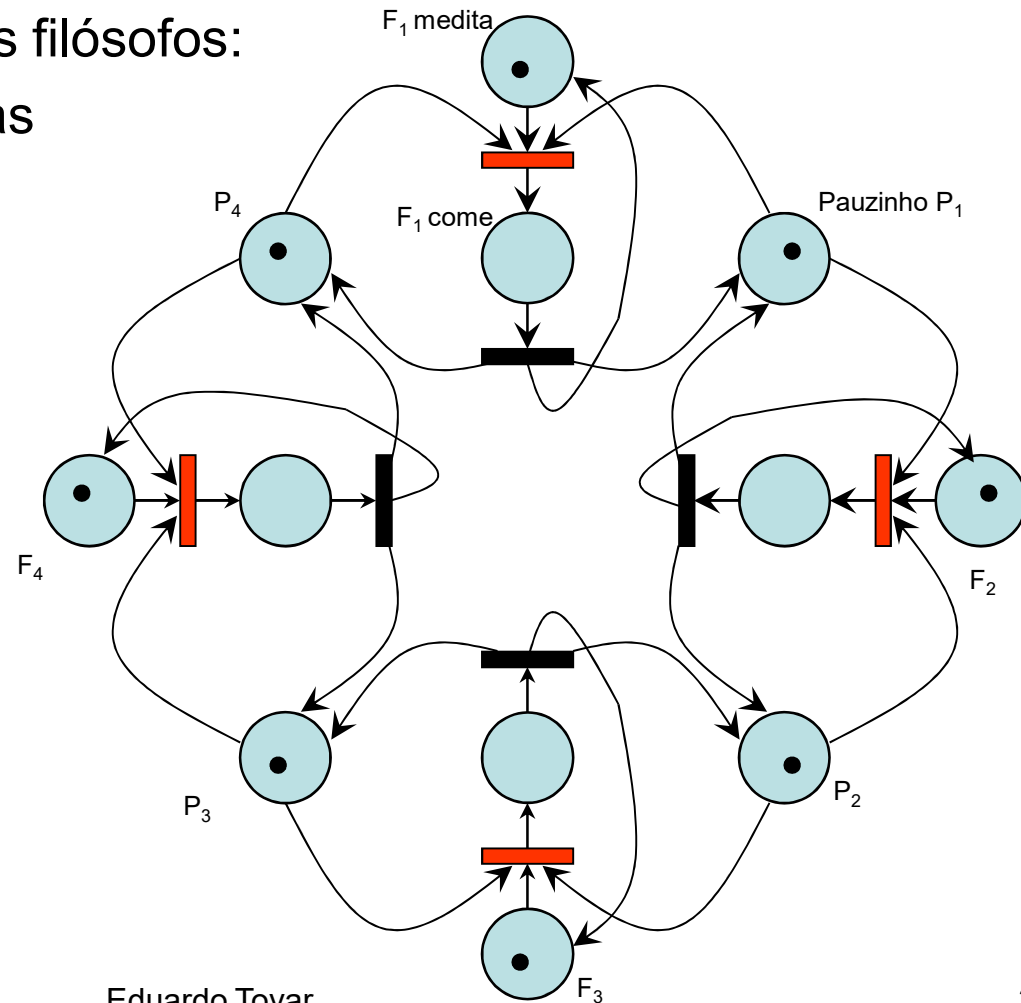
Regras de Evolução das RdP (12)

- exemplo da refeição dos filósofos:
 - modelo RdP



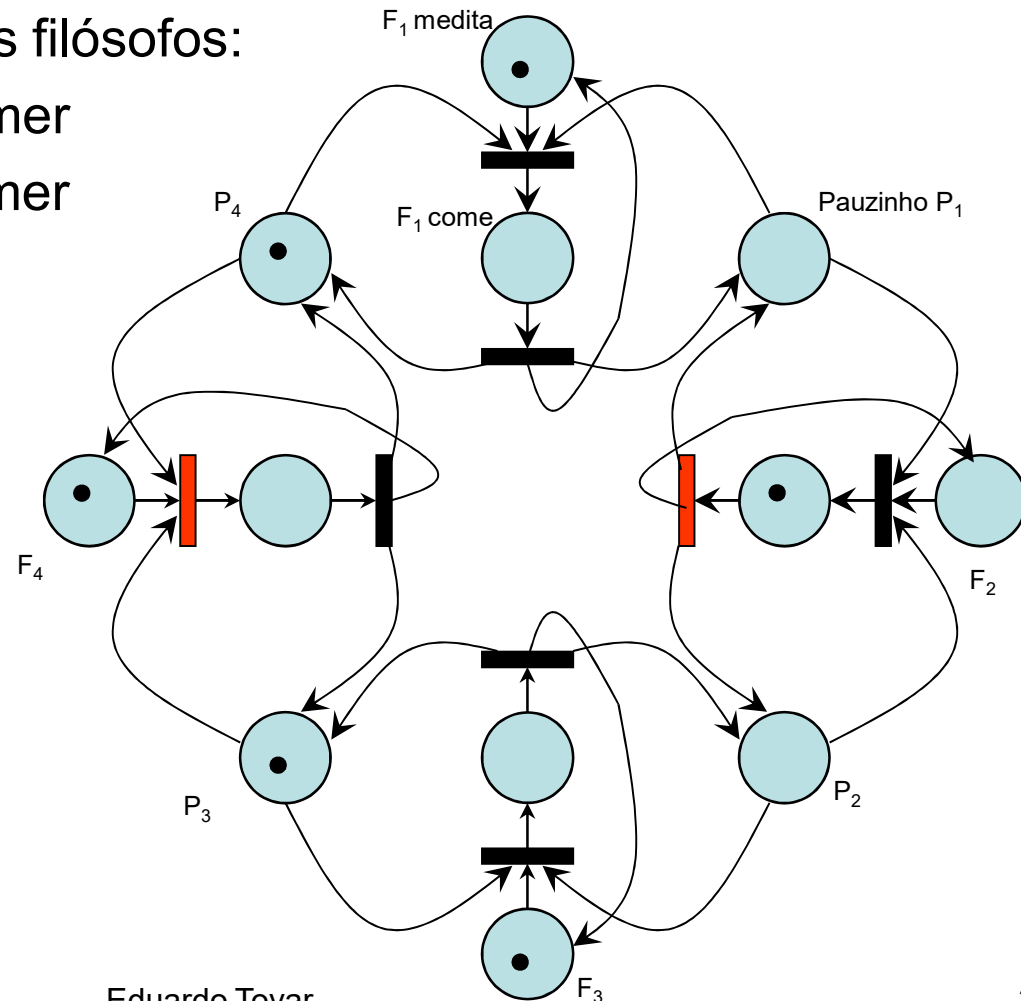
Regras de Evolução das RdP (13)

- exemplo da refeição dos filósofos:
 - transições habilitadas



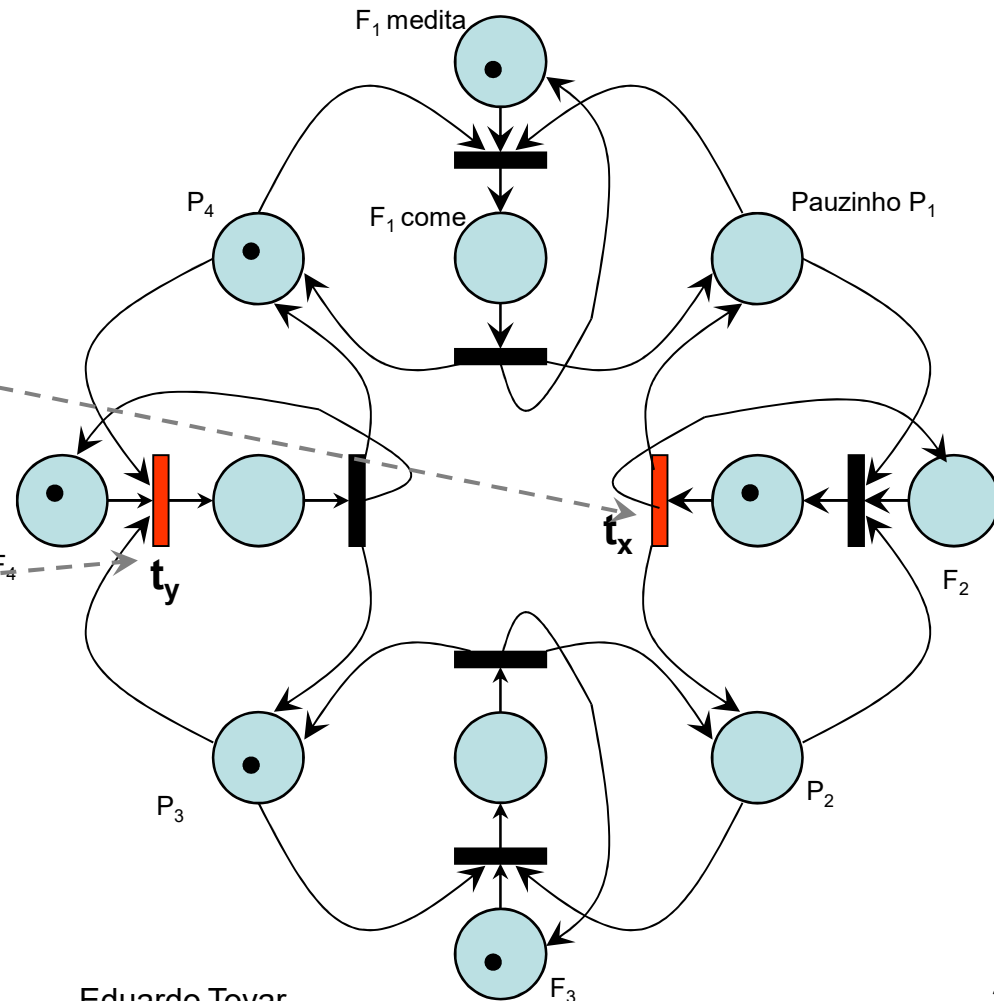
Regras de Evolução das RdP (14)

- exemplo da refeição dos filósofos:
 - filósofo 2 decide comer
 - só o F4 pode tb. comer



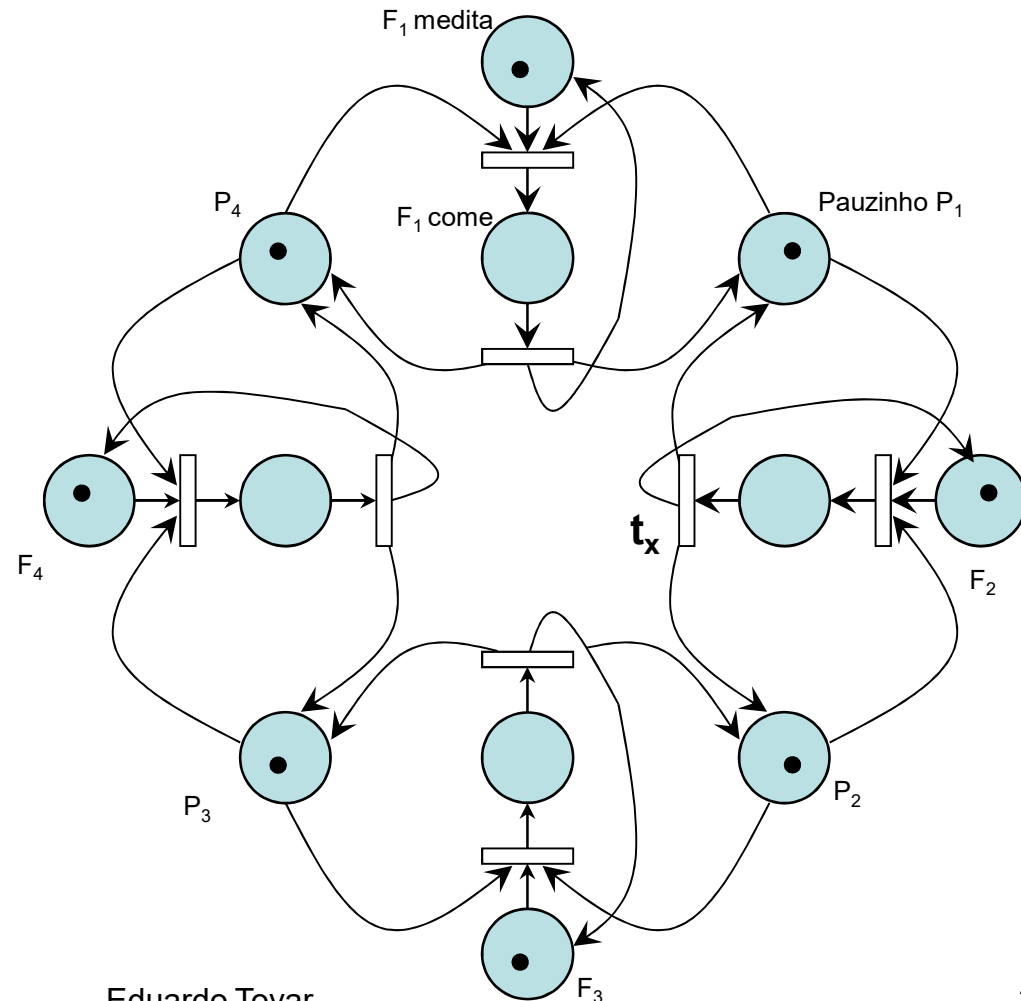
Regras de Evolução das RdP (15)

- **NOTA:** para além de uma transição ter de estar habilitada, existe uma condição lógica (verdadeira ou falsa) a ela associada:
 - a transição t_x está habilitada, mas o filósofo F2 continua a comer (condição booleana **farto** é falsa)
 - a transição t_y está habilitada, mas o filósofo F4 continua a meditar (condição booleana **fome** é falsa)



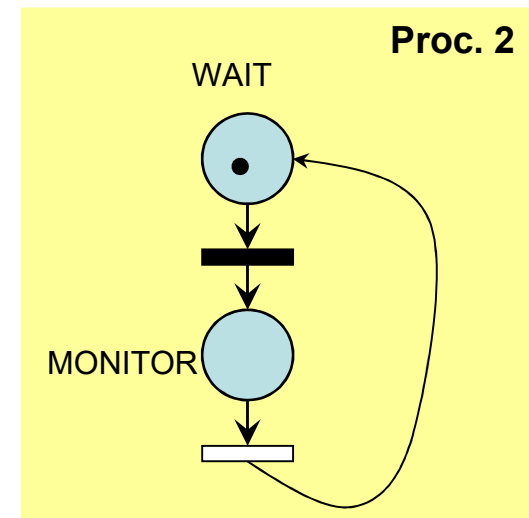
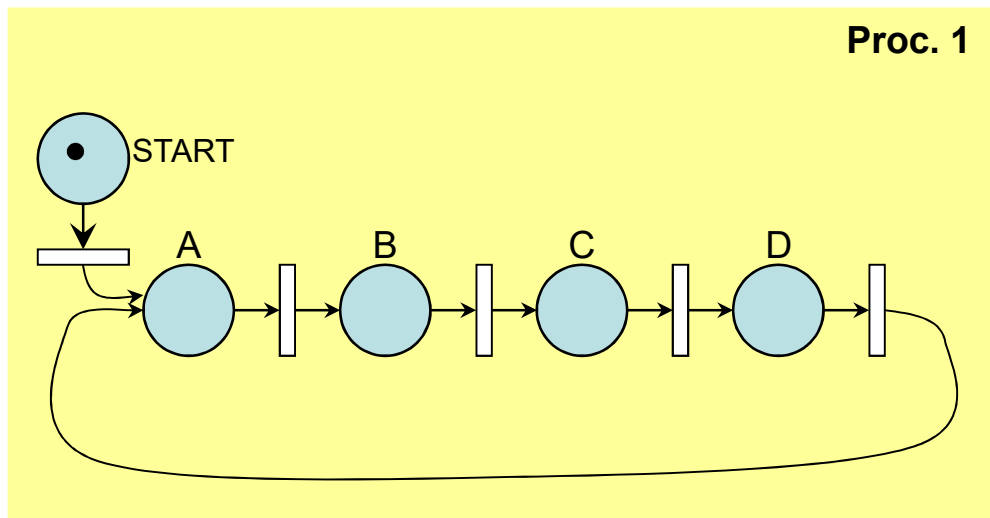
Regras de Evolução das RdP (16)

- por simplificação, essas condições lógicas às vezes não aparecem explicitamente nos modelos
- pode acontecer que a condição lógica associada a uma transição seja sempre verdadeira
- para explicitar isso vamos utilizar a seguinte notação
 - transição sempre verdadeira (disparo imediato):
■
 - transição condicionada por um valor lógico (por exemplo tempo):
□



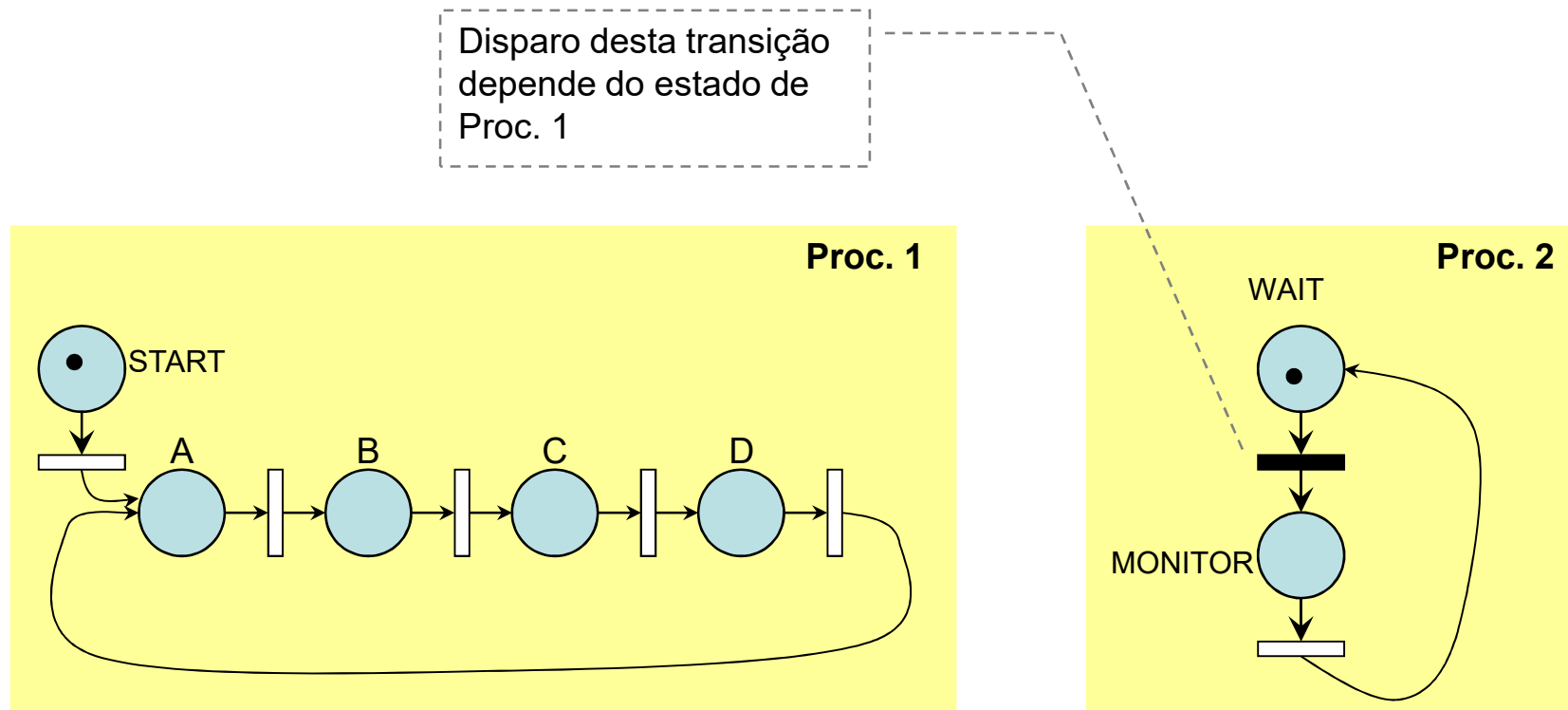
Regras de Evolução das RdP (17)

- modelo RdP de um Sistema: **EXEMPLO B**
 - Numa plataforma computacional existe um processo (**Proc. 1**) que executa continuamente. O Proc. 1 corresponde à execução sequencial de 4 procedimentos (A, B, C, D), cada um dos quais demora algum tempo a executar.
 - Existe na mesma plataforma um outro processo (**Proc. 2**). Proc. 2 tem dois estados (WAIT e MONITOR). Sempre que Proc. 1 começa a executar o procedimento B ou o D, e enquanto de mantiver a executar um desses procedimentos, Proc. 2 deve executar MONITOR (procedimento que demora algum tempo).



Regras de Evolução das RdP (18)

- modelo RdP de um Sistema: EXEMPLO B

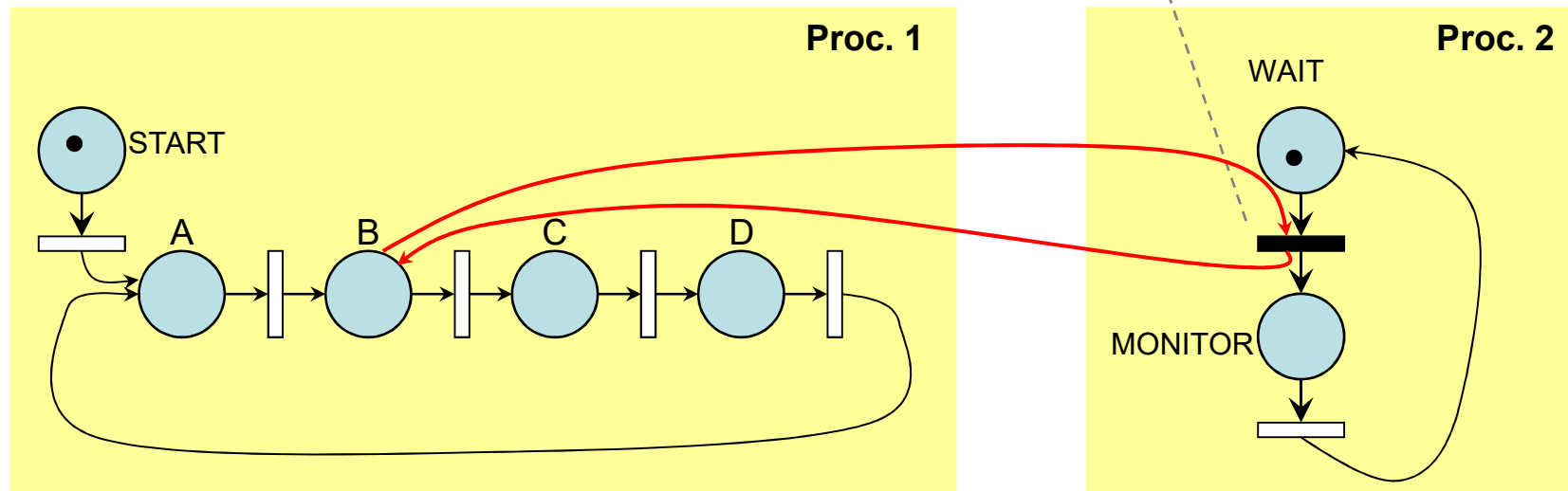


Regras de Evolução das RdP (19)

- modelo RdP de um Sistema: EXEMPLO B

O disparo desta transição passa a depender da LEITURA (não altera a marcação da posição “lida”) da posição B.

Falta acrescentar a sincronização com a posição D...

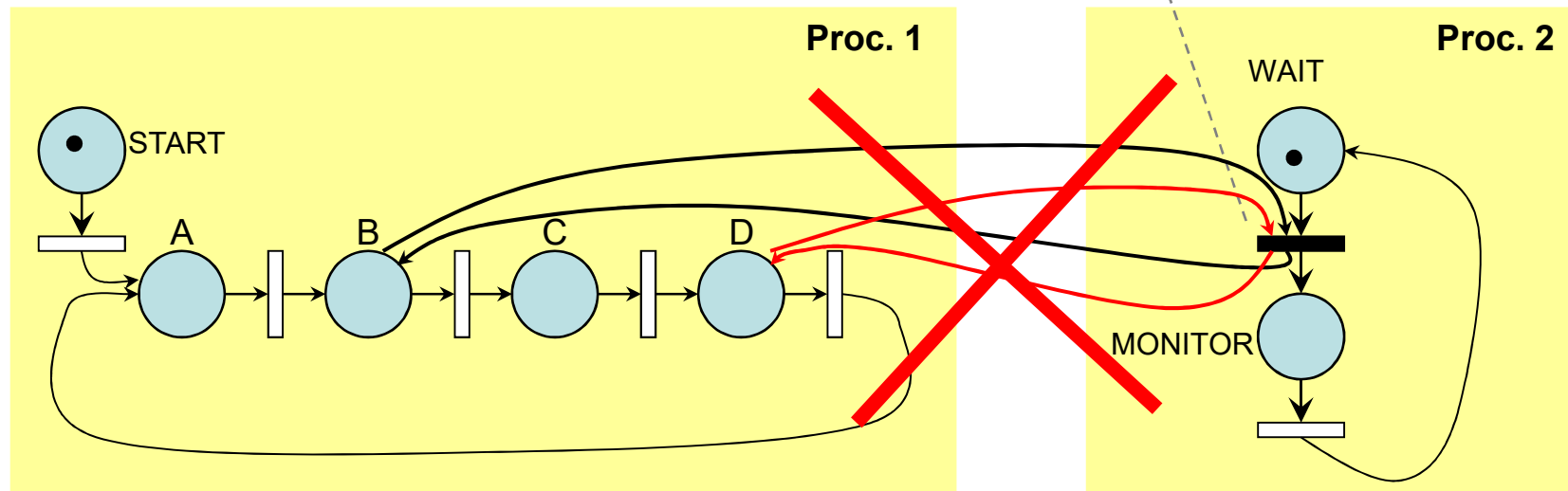


Regras de Evolução das RdP (20)

- modelo RdP de um Sistema: EXEMPLO B

Assim estamos a condicionar o disparo da transição à existência de uma marca em B e uma marca em D (coisa que nunca pode acontecer no Proc. 1). É um **AND** lógico.

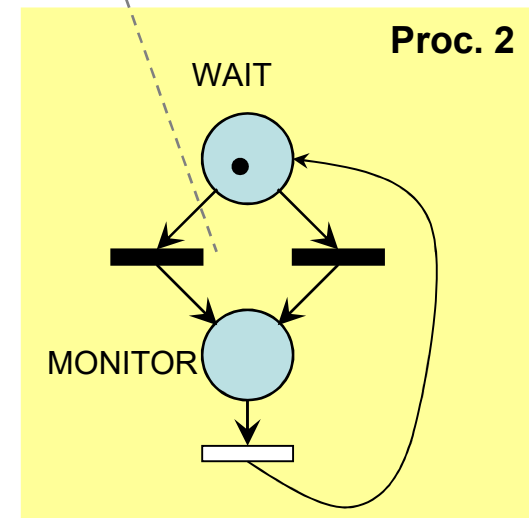
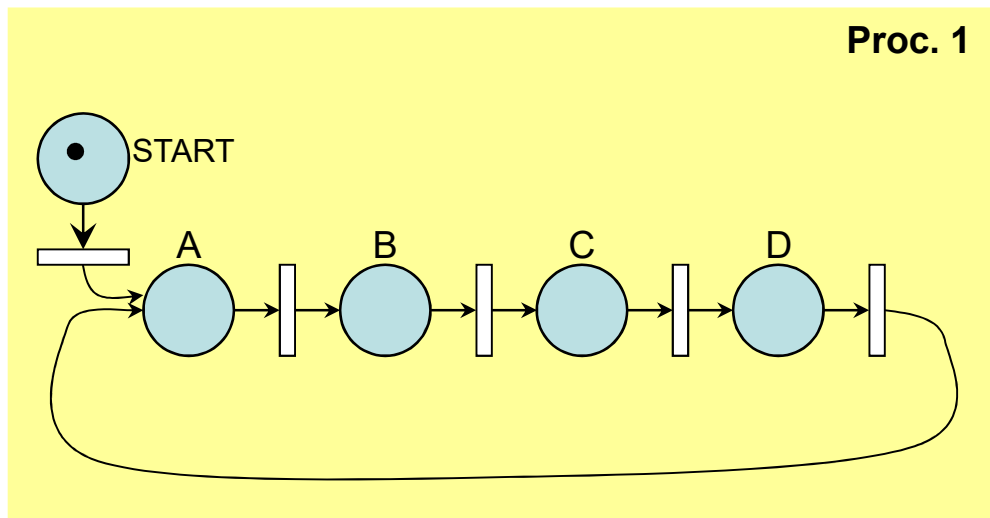
Queremos um **OR** lógico...



Regras de Evolução das RdP (21)

- modelo RdP de um Sistema: EXEMPLO B

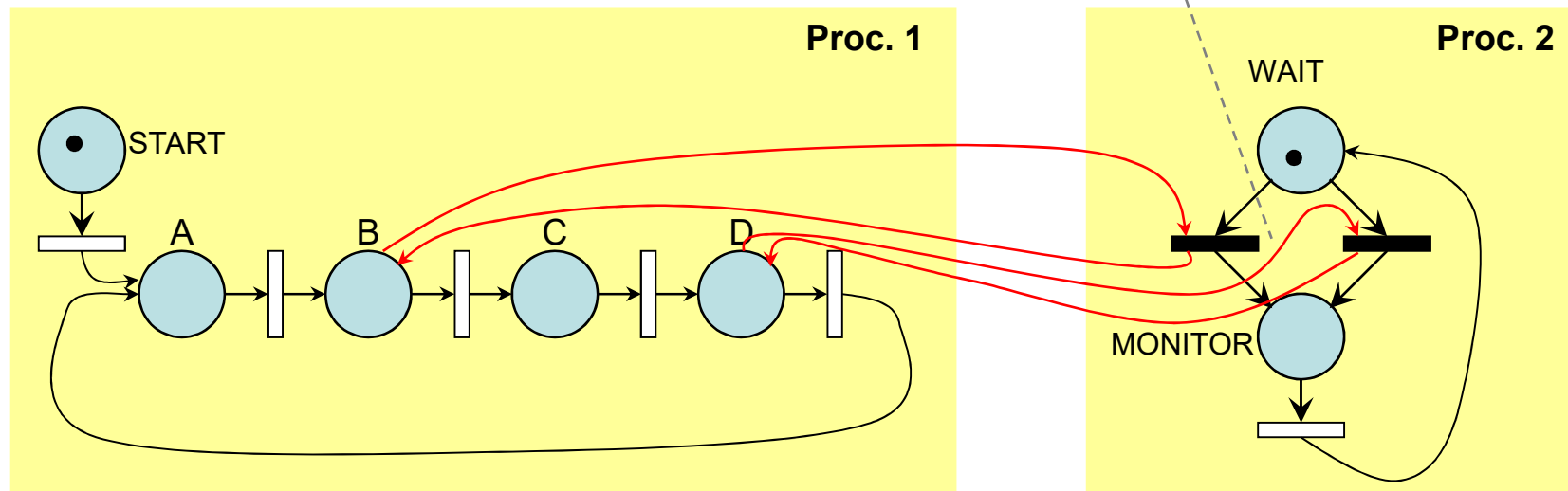
A evolução em Proc. 2 de WAIT para MONITOR pode fazer-se ou por ... ou por ... Ora aí está um **OR** lógico...



Regras de Evolução das RdP (22)

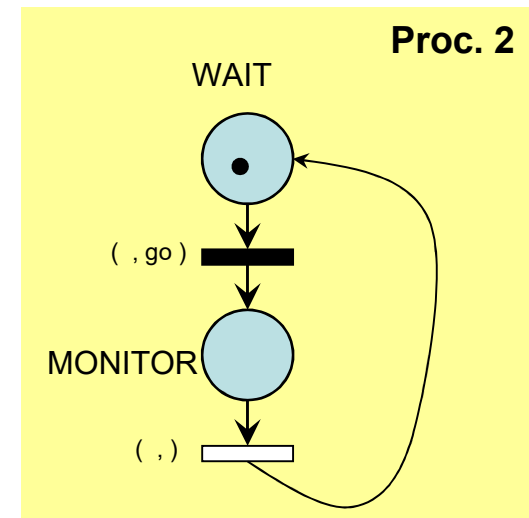
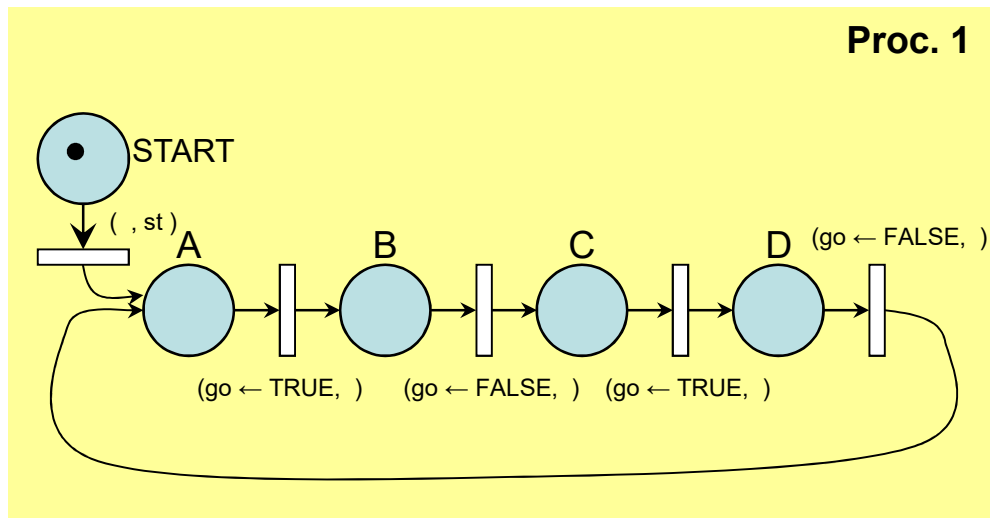
- modelo RdP de um Sistema: EXEMPLO B

A evolução em Proc. 2 de WAIT para MONITOR dá-se ou se houver uma marca em B, ou se houver uma marca em D



Regras de Evolução das RdP (23)

- modelo RdP de um Sistema: EXEMPLO B
 - (utilizando *Labeled Petri Nets* – LPN)
 - associado a uma transição existe uma etiqueta
 - par (operadores, condição lógica)
 - a ferramenta que vamos utilizar não permite etiquetas
 - por outro lado, perde-se (com operadores mais complexos) a objectividade e clareza do modelo gráfico e regras

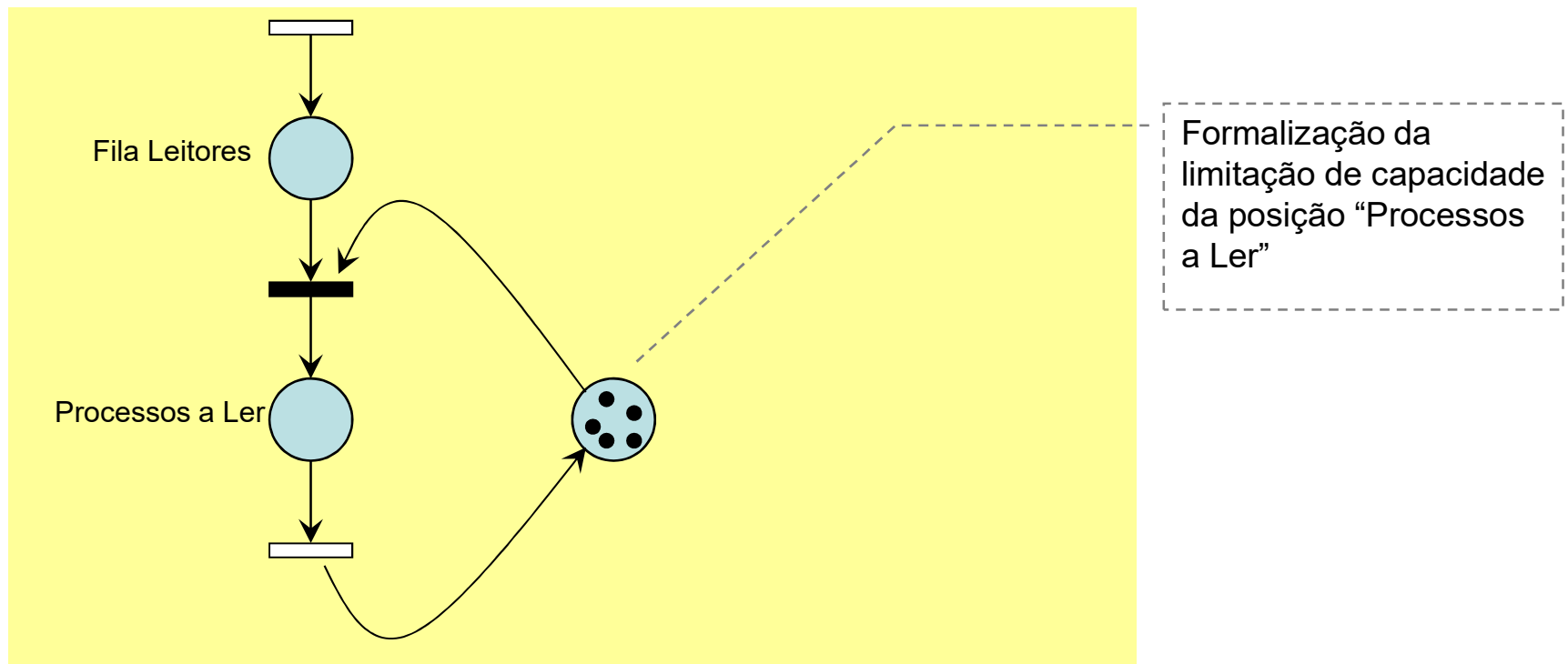


Índice

1. Introdução ✓
2. Noções Básicas de Redes de Petri (RdP) ✓
3. Regras de Evolução das RdP ✓
- 4. RdP Generalizadas**
5. Componentes de Modelação em RdP
6. Análise Computacional de Modelos RdP
7. Verificação de Propriedades dos Sistemas Modelados
8. Utilização da Ferramenta HP-SIM

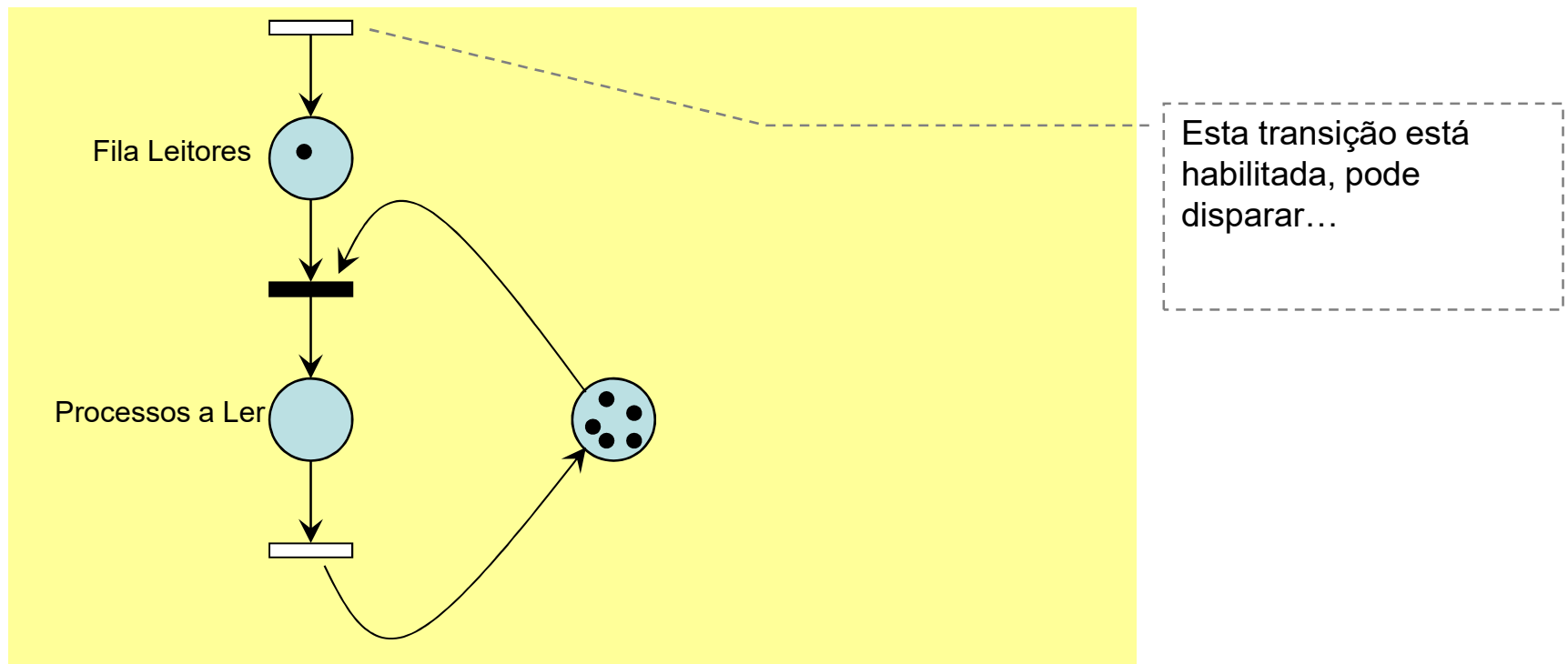
RdP Generalizadas (1)

- modelo RdP de um Sistema:
 - Numa plataforma computacional podem ser lançados processos que vão ler uma estrutura de dados. Por uma questão de performance do sistema, não são autorizados mais do que 5 leitores em simultâneo.



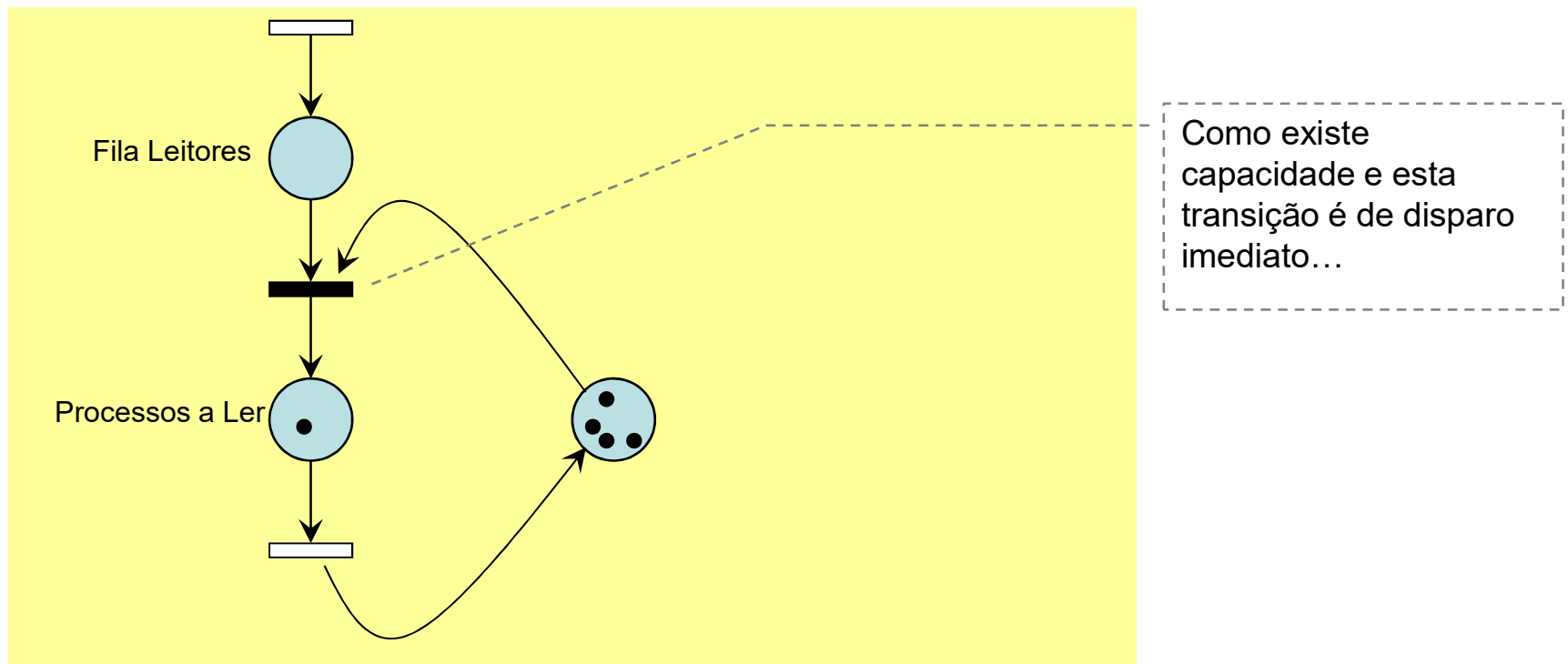
RdP Generalizadas (2)

- modelo RdP de um Sistema:
 - Numa plataforma computacional podem ser lançados processos que vão ler uma estrutura de dados. Por uma questão de performance do sistema, não são autorizados mais do que 5 leitores em simultâneo.



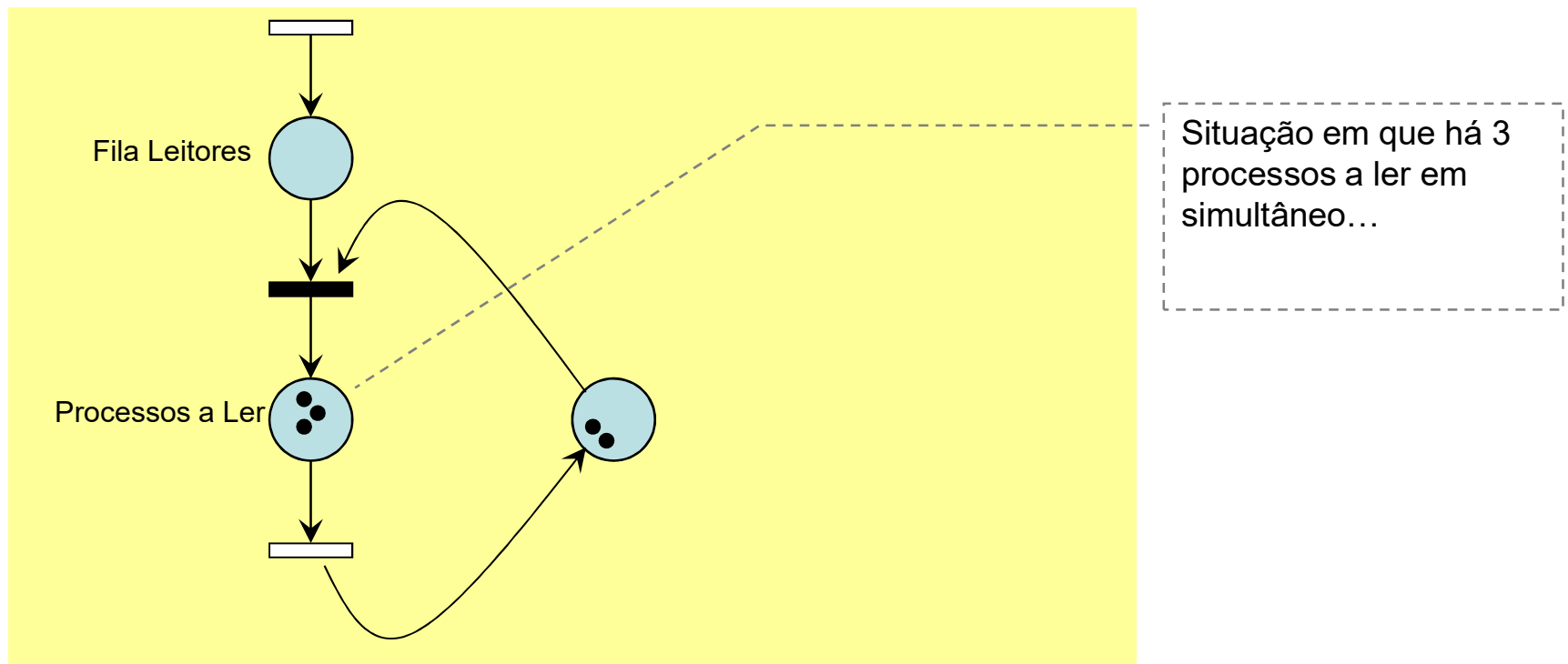
RdP Generalizadas (3)

- modelo RdP de um Sistema:
 - Numa plataforma computacional podem ser lançados processos que vão ler uma estrutura de dados. Por uma questão de performance do sistema, não são autorizados mais do que 5 leitores em simultâneo.



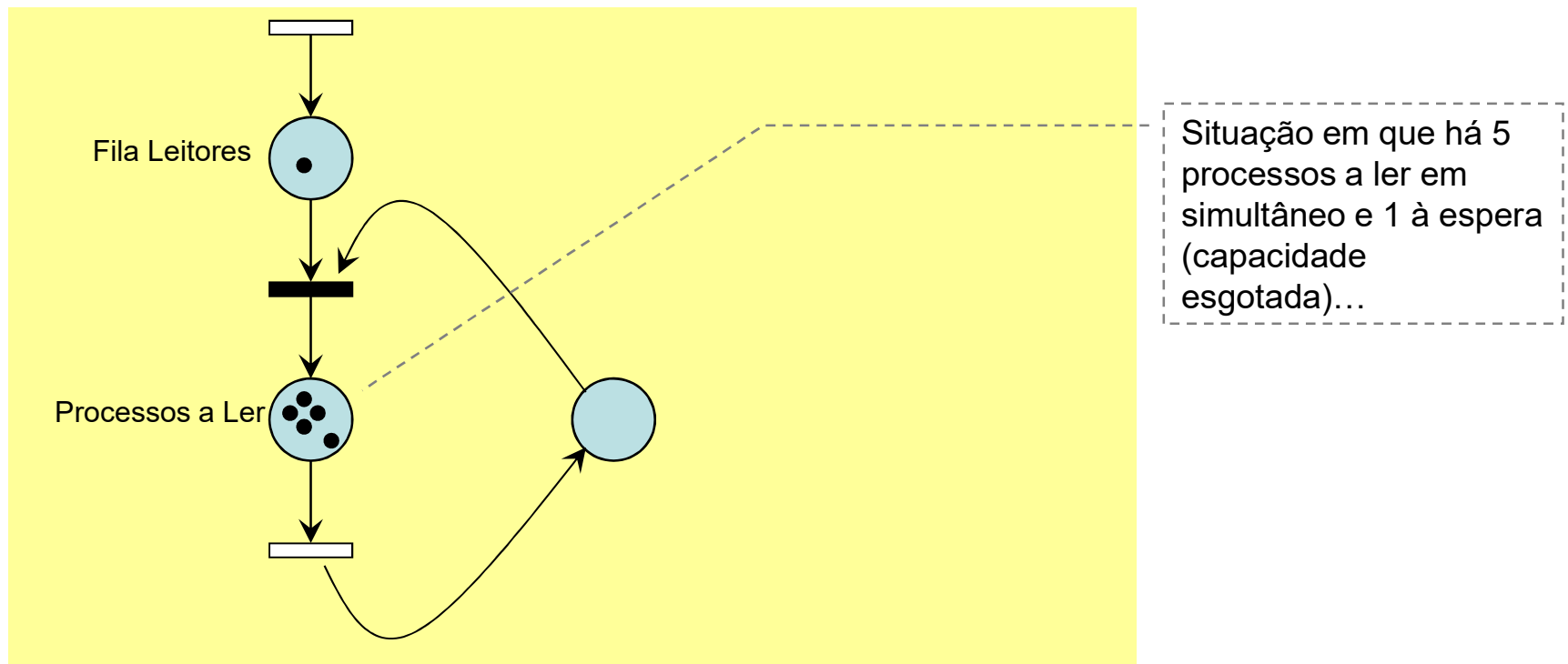
RdP Generalizadas (4)

- modelo RdP de um Sistema:
 - Numa plataforma computacional podem ser lançados processos que vão ler uma estrutura de dados. Por uma questão de performance do sistema, não são autorizados mais do que 5 leitores em simultâneo.



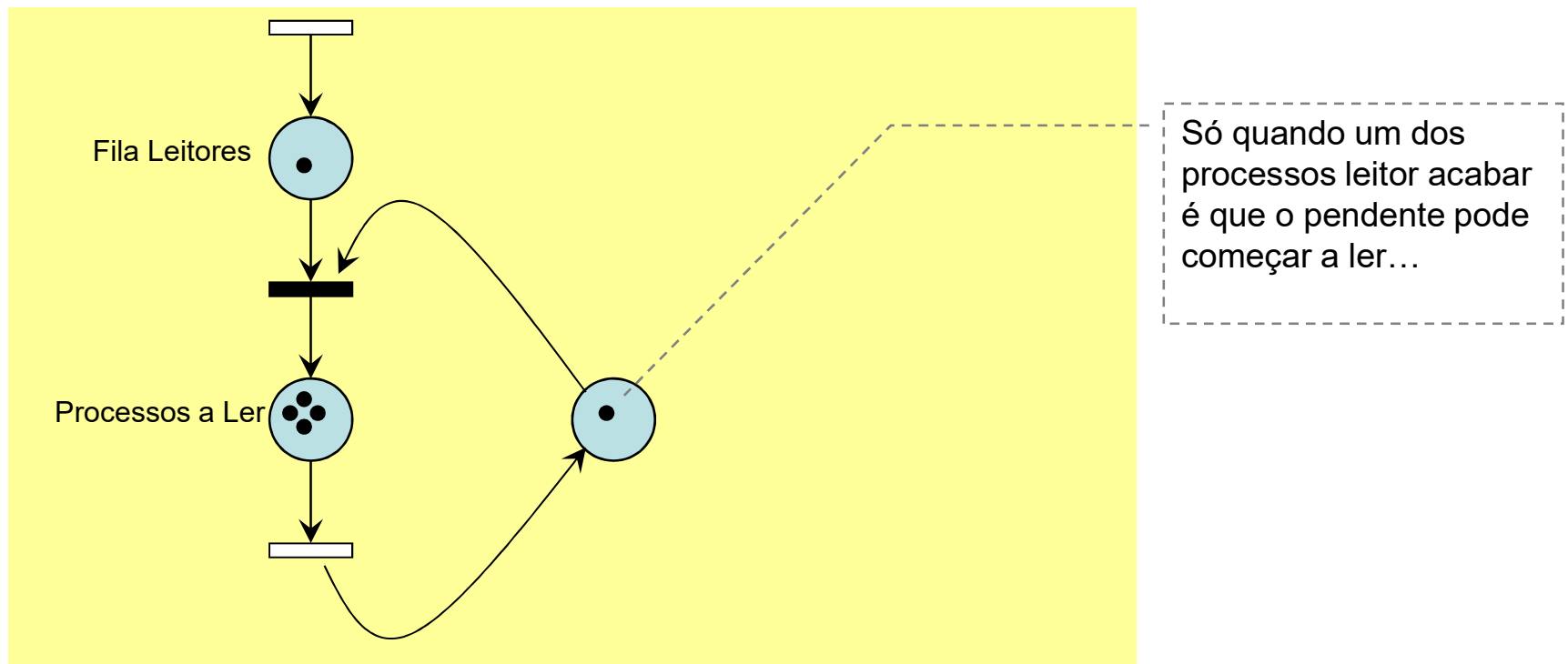
RdP Generalizadas (5)

- modelo RdP de um Sistema:
 - Numa plataforma computacional podem ser lançados processos que vão ler uma estrutura de dados. Por uma questão de performance do sistema, não são autorizados mais do que 5 leitores em simultâneo.



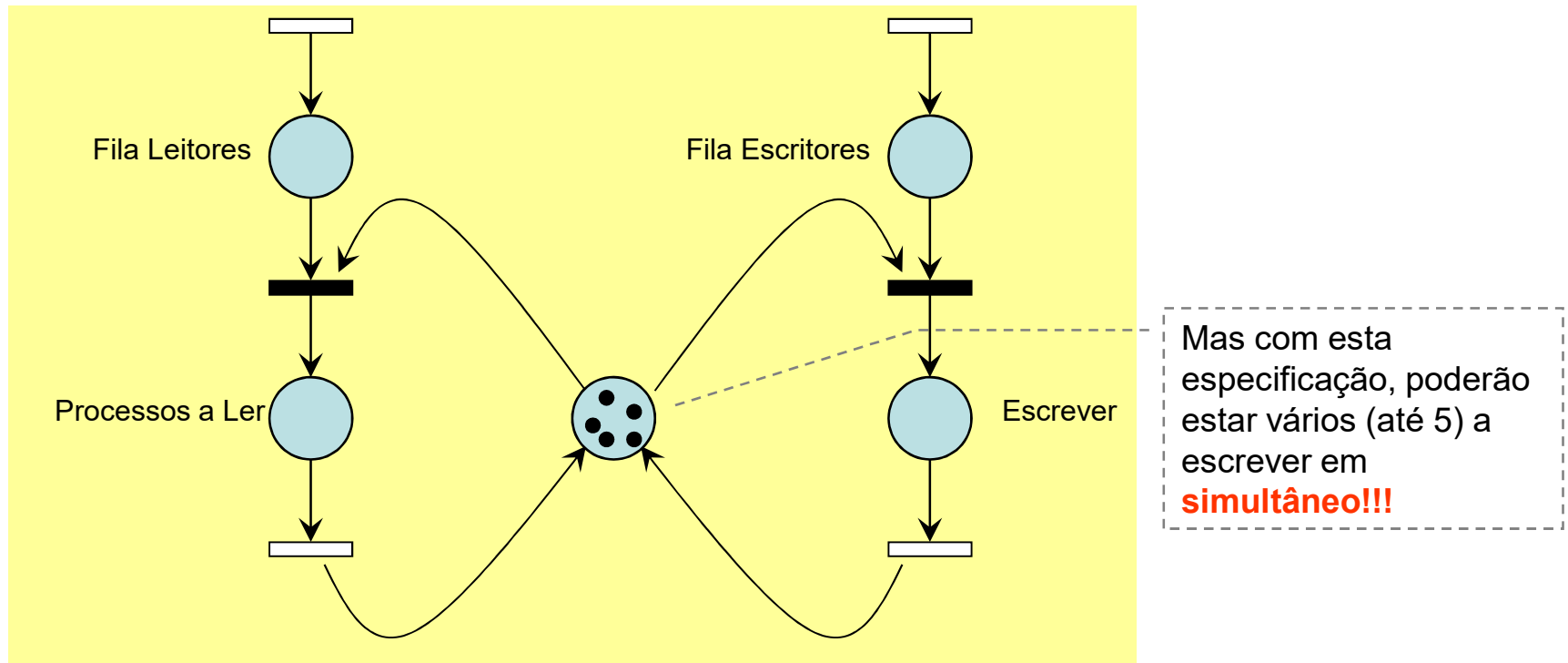
RdP Generalizadas (6)

- modelo RdP de um Sistema:
 - Numa plataforma computacional podem ser lançados processos que vão ler uma estrutura de dados. Por uma questão de performance do sistema, não são autorizados mais do que 5 leitores em simultâneo.



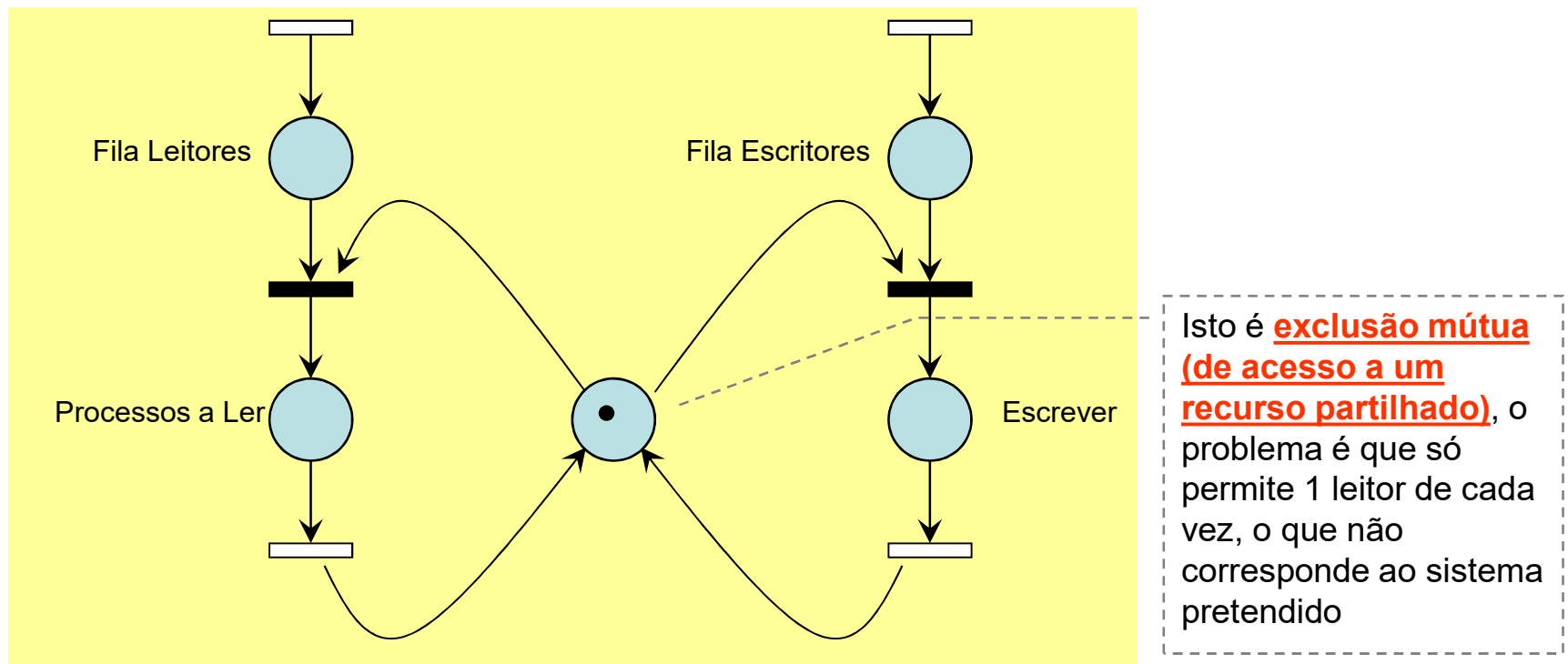
RdP Generalizadas (7)

- modelo RdP de um Sistema (com processos que escrevem):
 - ... Podem também ser lançados processos que vão escrever a estrutura de dados (nenhum outro poderá estar a ler)...



RdP Generalizadas (8)

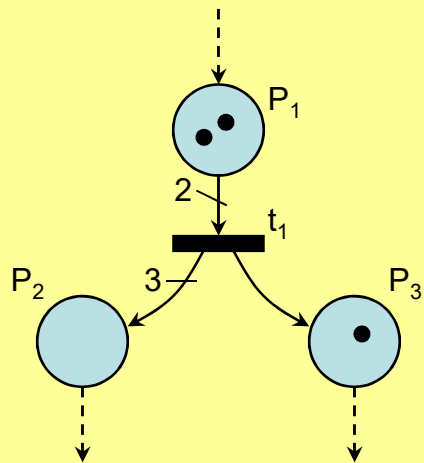
- modelo RdP de um Sistema (com processos que escrevem):
 - ... Podem também ser lançados processos que vão escrever a estrutura de dados (nenhum outro poderá estar a ler)...



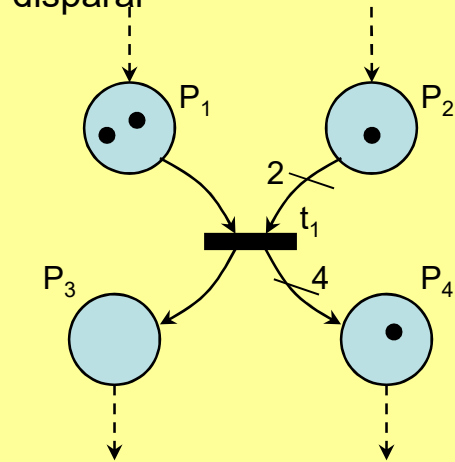
RdP Generalizadas (9)

- generalização das regras das RdP
 - os arcos têm pesos associados (até aqui o seu peso era sempre “1”)
 - **generalização da REGRA 1** (condição para o disparo de uma transição):
 - uma transição está habilitada (pode disparar) por uma determinada marcação se, e só se, todas as posições anteriores à transição têm um número de marcas igual ou superior ao peso do arco que as liga à transição

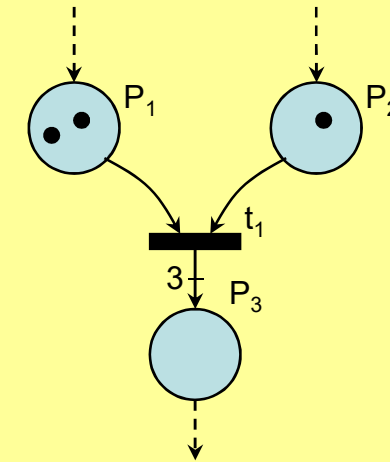
Ex. 1: t_1 pode disparar



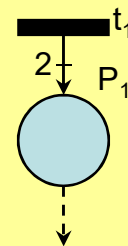
Ex. 2: t_1 **não** pode disparar



Ex. 3: t_1 pode disparar

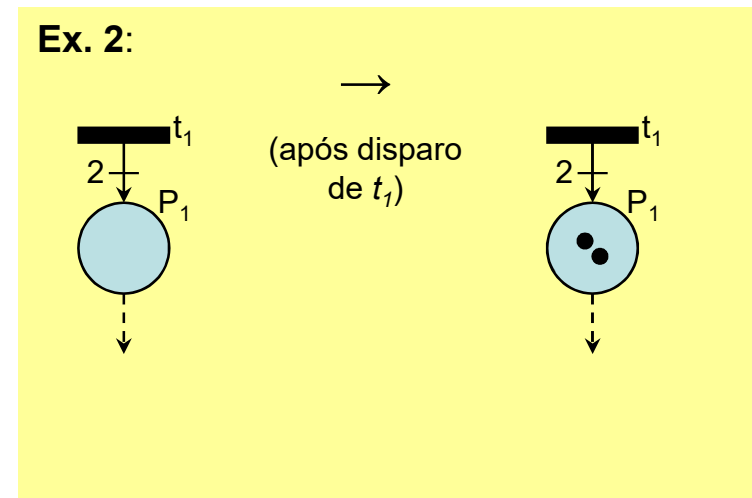
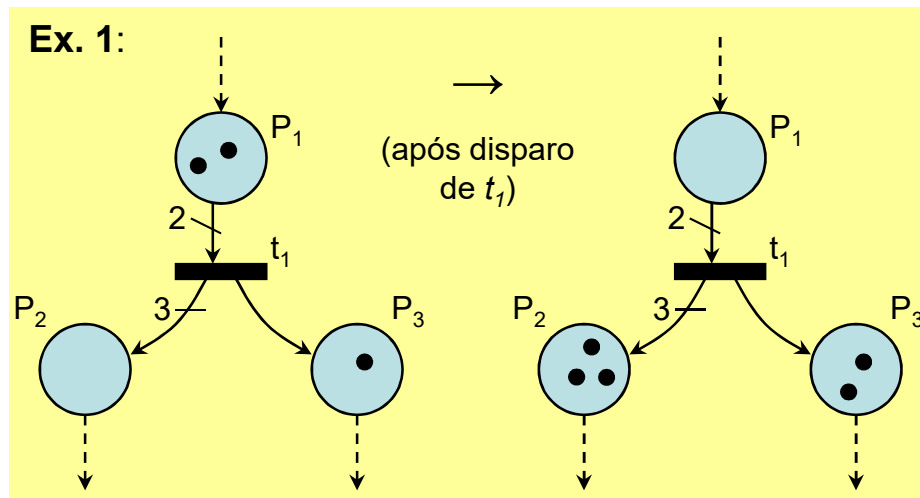


Ex. 4: t_1 pode disparar



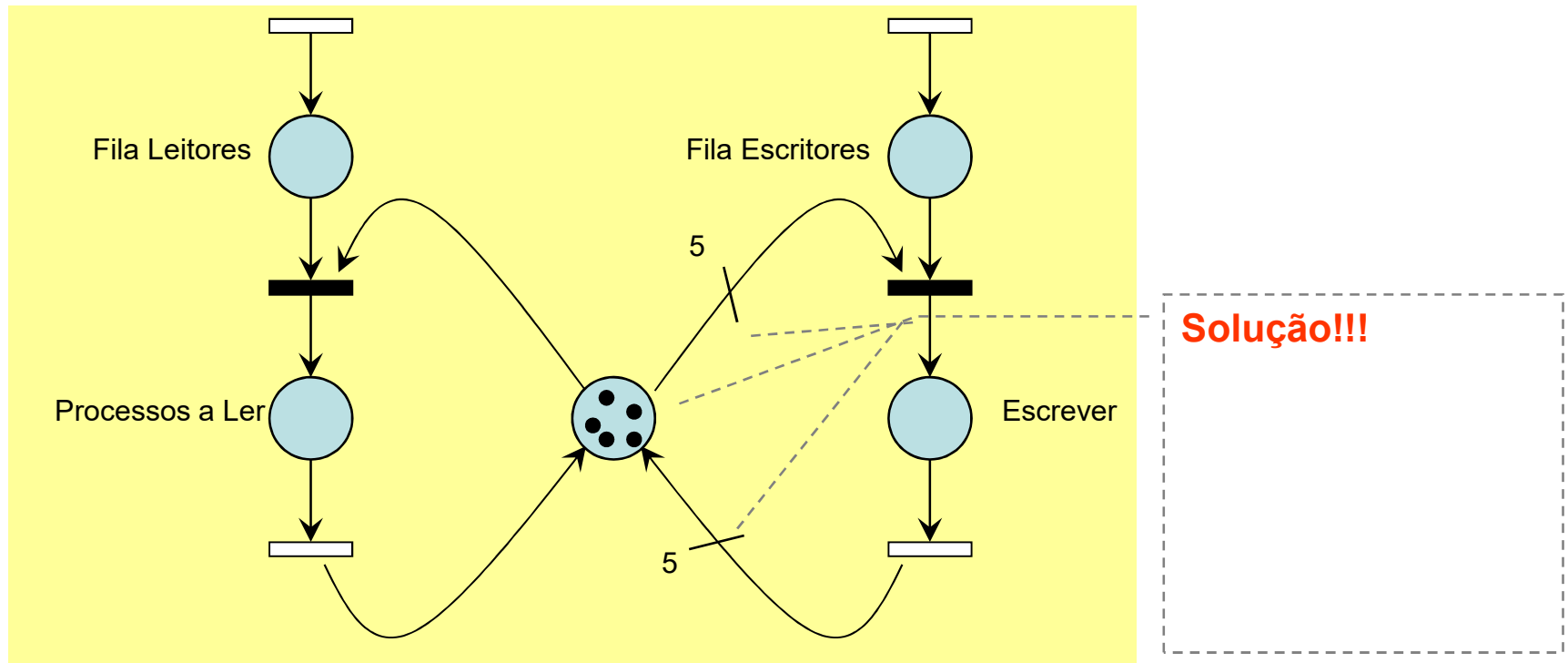
RdP Generalizadas (10)

- generalização das regras das RdP
 - os arcos têm pesos associados (até aqui o seu peso era sempre “1”)
 - **generalização da REGRA 2** (evolução da marcação após disparo):
 - o número de marcas de cada posição anterior à transição diminui em número igual ao peso do arco que liga a posição à transição
 - e o número de marcas de cada posição posterior à transição aumenta em número igual ao peso do arco que liga a posição à transição



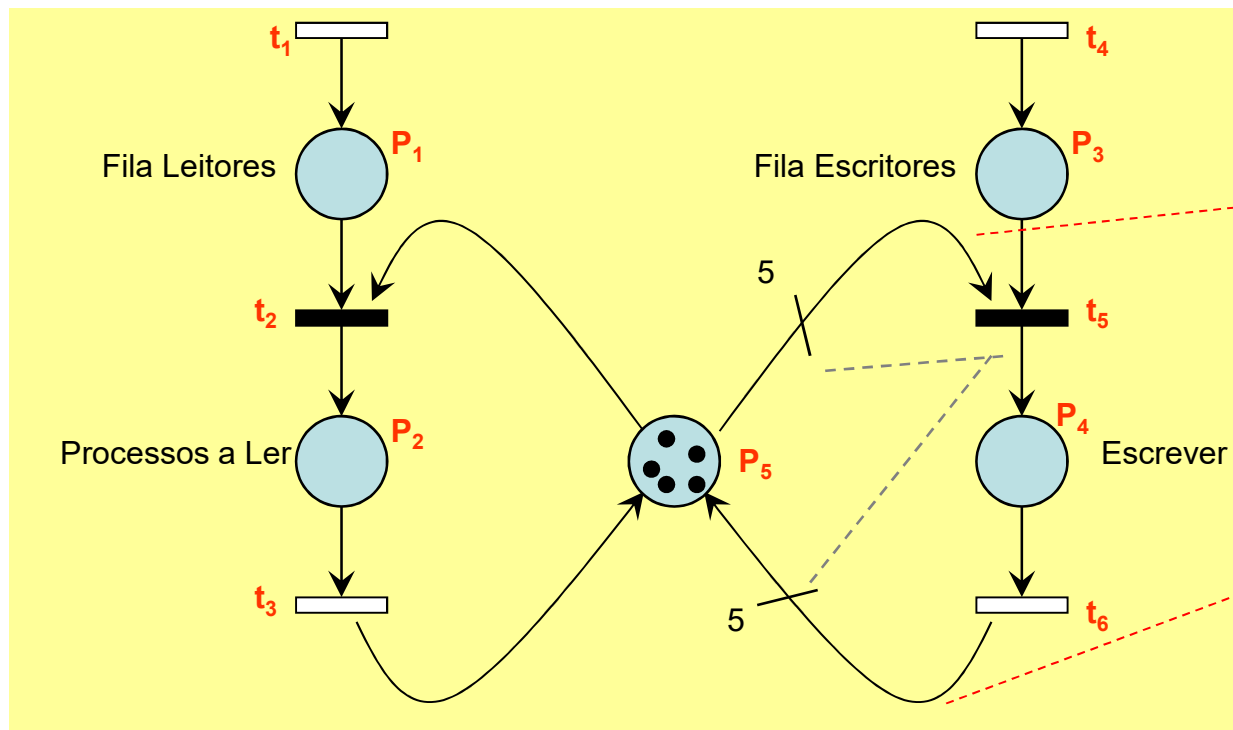
RdP Generalizadas (11)

- modelo RdP de um Sistema (com processos que escrevem):
 - ... Podem também ser lançados processos que vão escrever a estrutura de dados (nenhum outro poderá estar a ler)...



RdP Generalizadas (12)

- modelo RdP de um Sistema (com processos que escrevem):
 - ... Podem também ser lançados processos que vão escrever a estrutura de dados (nenhum outro poderá estar a ler)...



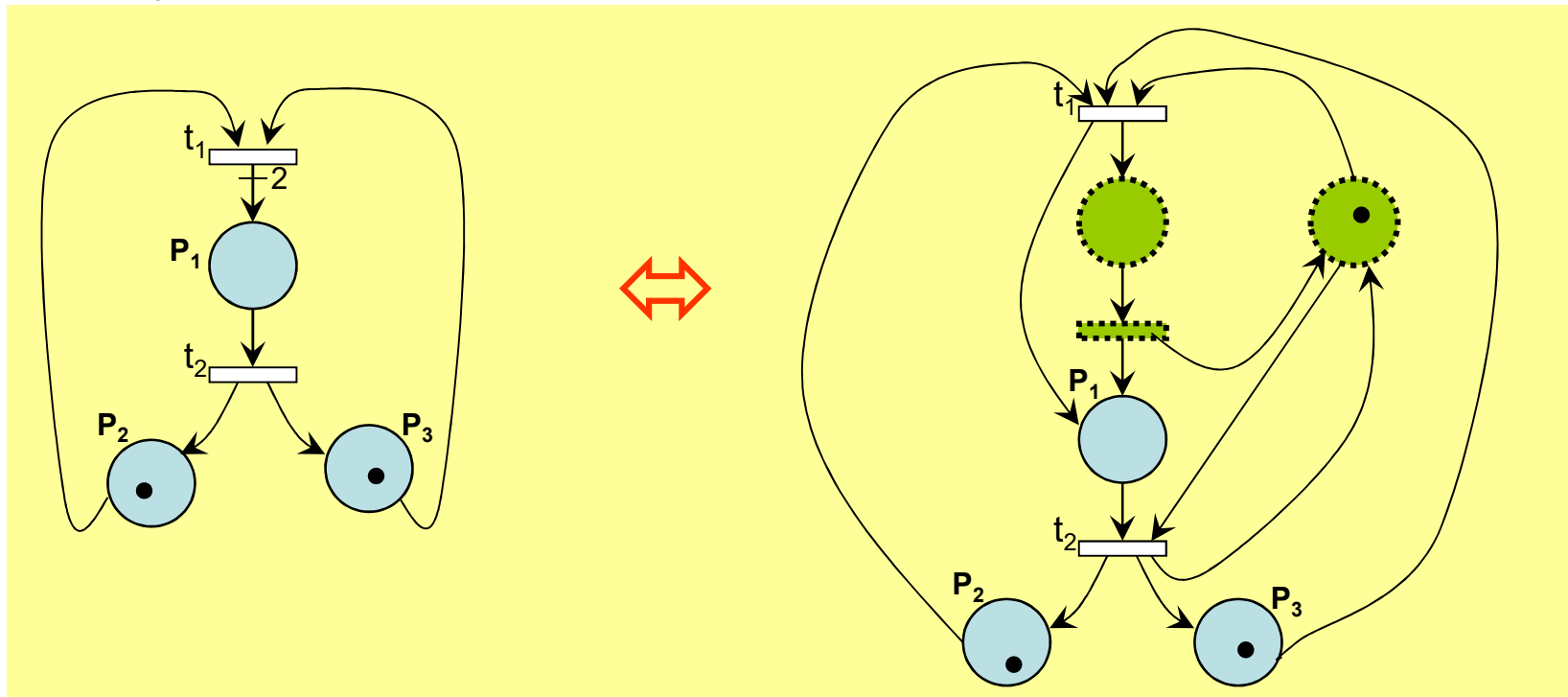
$$W^- = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 5 & 0 \end{bmatrix}$$

$$W^+ = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 5 \end{bmatrix}$$

$$M_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 5 \end{bmatrix}$$

RdP Generalizadas (13)

- as RdP generalizadas têm sempre um equivalente RdP ordinária (pesos dos arcos sempre “1”), mas é difícil, como se pode deduzir do exemplo do controlo de acessos a uma estrutura de dados, obter o equivalente
 - vejamos um exemplo mais simples:



Índice

1. Introdução ✓
2. Noções Básicas de Redes de Petri (RdP) ✓
3. Regras de Evolução das RdP ✓
4. RdP Generalizadas ✓
- 5. Componentes de Modelação em RdP**
6. Análise Computacional de Modelos RdP
7. Verificação de Propriedades dos Sistemas Modelados
8. Utilização da Ferramenta HP-SIM

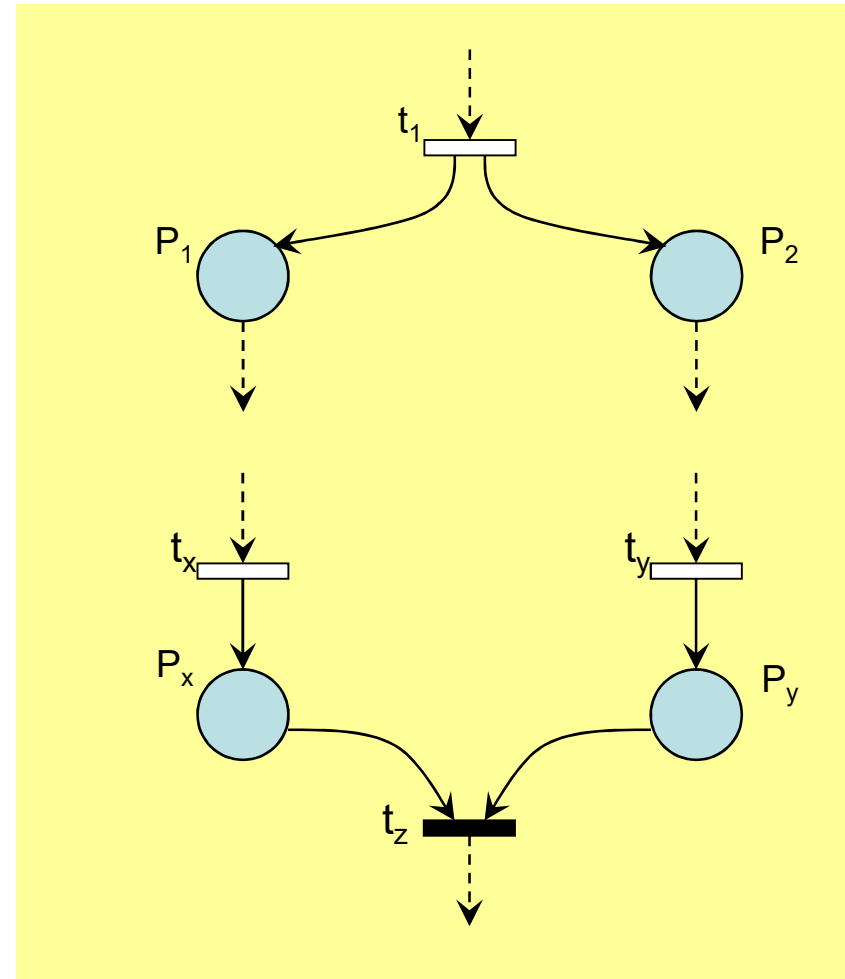
Componentes de Modelação (1)

- nos modelos RdP, existem algumas “figuras de modelação” (componentes de modelação) muito comuns
 - incluem-se (já vimos alguns):
 - paralelismo
 - sincronização
 - partilha de recursos
 - memorização
 - leitura
 - limitação de capacidade
 - alternância
 - alternância com exclusão
 - contador
 - leitura de zero marcas
 - etc.

Componentes de Modelação (2)

- paralelismo

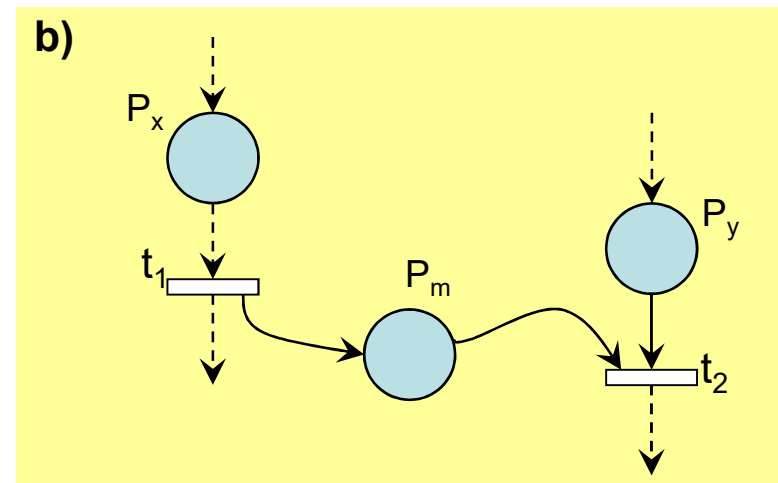
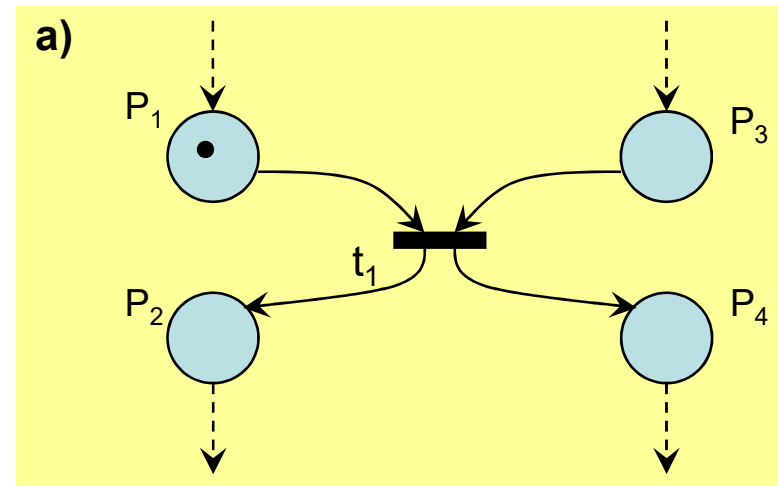
- após o disparo de t_1 e até ao disparo t_z , existem duas evoluções em paralelo:
 - de P_1 e a P_x , e de P_2 e a P_y
 - cada uma tem a sua dinâmica temporal própria
- no exemplo, as duas sequências independentes vão sincronizar em t_z



Componentes de Modelação (3)

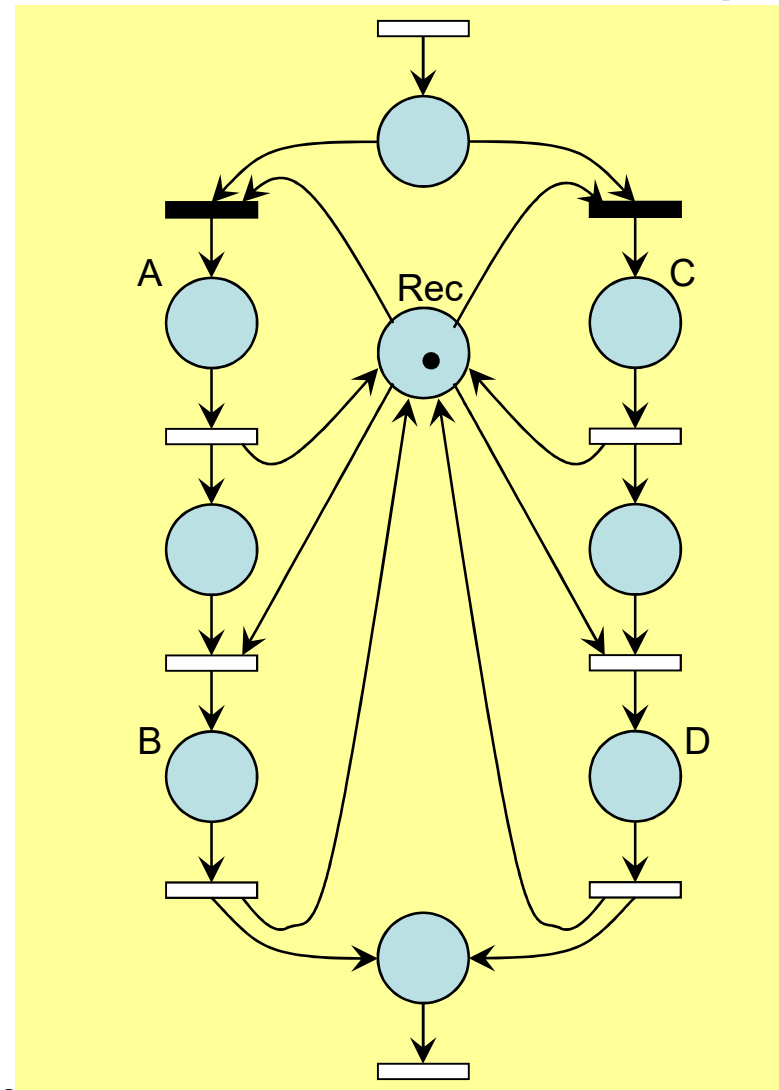
- sincronização

- à semelhança do exemplo anterior, em a) é representada um sincronização recíproca
- em b), a sequência do lado esquerdo é independente da sequência do lado direito (o inverso não é verdade)
 - t_2 só pode disparar após o disparo de t_1
 - este acontecimento é memorizado em P_m



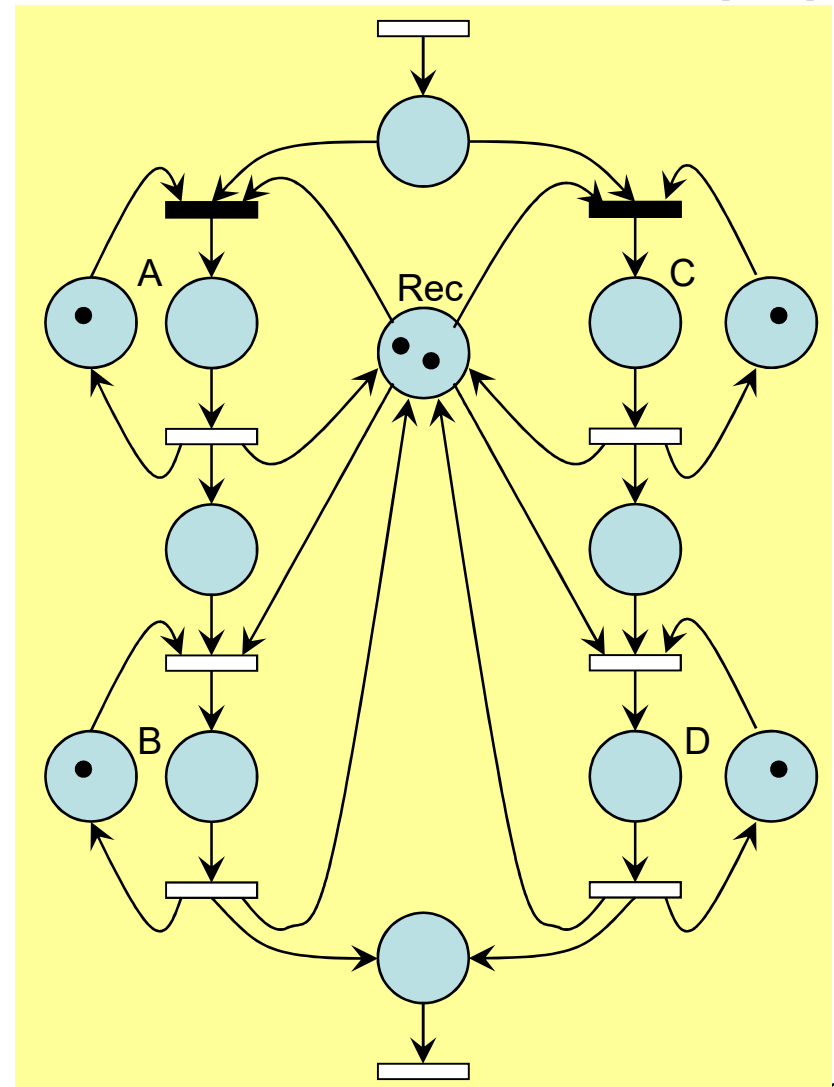
Componentes de Modelação (4)

- partilha de recursos
 - no exemplo ao lado, as operações A, B, C e D partilham o recurso *Rec*



Componentes de Modelação (5)

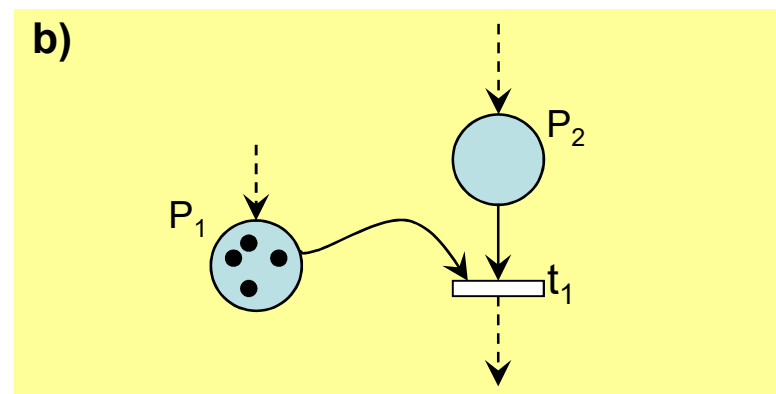
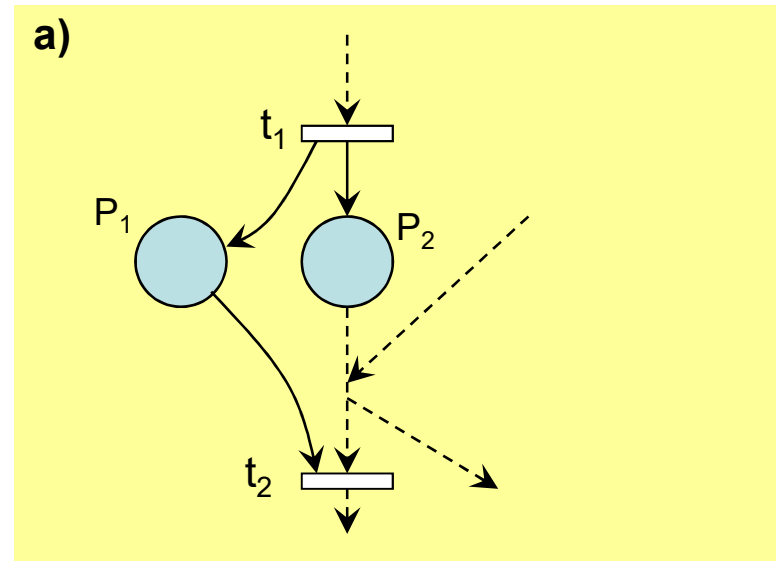
- partilha de recursos (2)
 - e se houvesse 2 recursos *Rec* disponíveis (mas A, B, C e D só podem utilizar 1 de cada vez)?
 - notar a utilização de limitadores de capacidade para as operações A, B, C e D



Componentes de Modelação (6)

- memorização

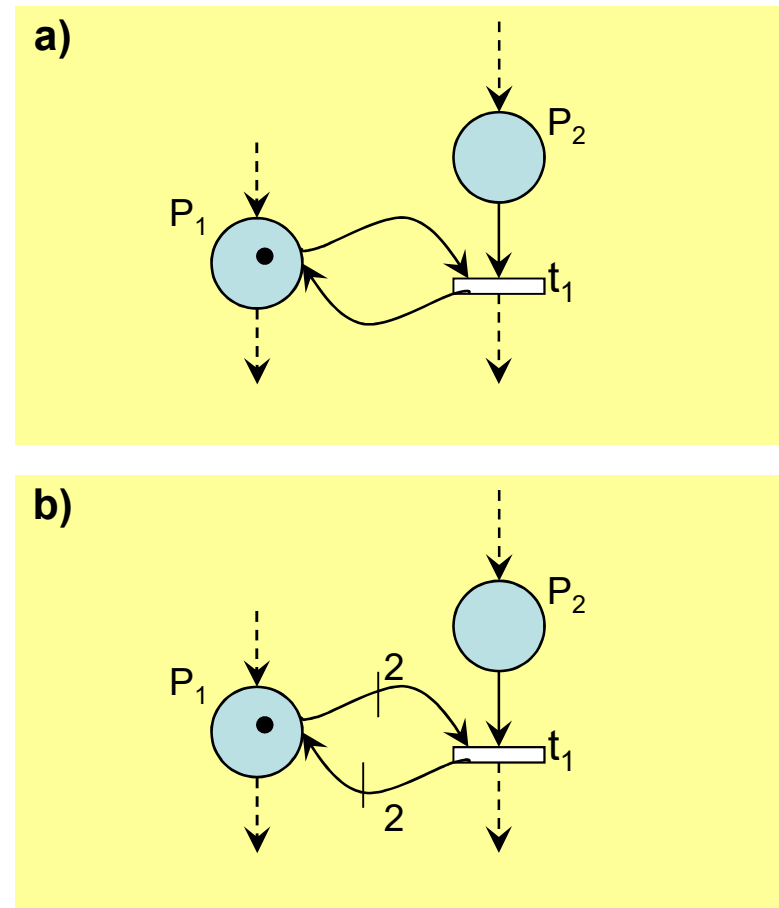
- em a), P_1 memoriza o facto de t_1 ter disparado, e autoriza o disparo posterior de t_2
 - de notar que t_2 poderia ficar habilitada por via de marcas provenientes de outro subsistema
- em b), P_1 vai memorizando um número, por exemplo de pedidos pendentes de serviço (no fundo trata-se de um contador)



Componentes de Modelação (7)

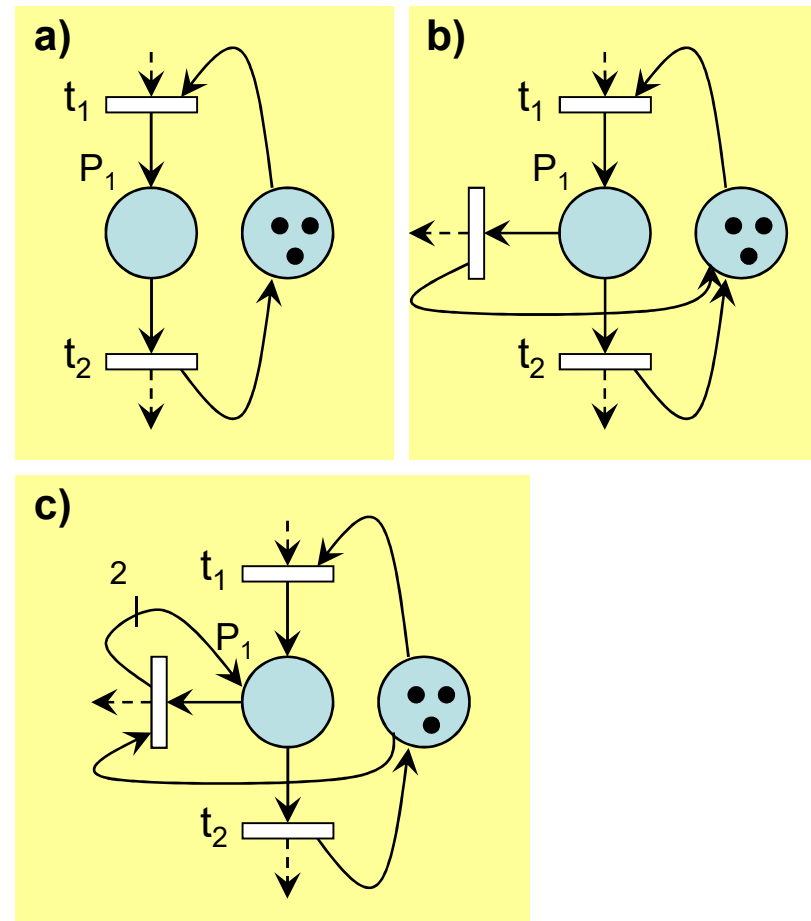
- leitura

- o disparo da transição t_1 é condicionado pela existência de pelo menos uma (ou pelo menos duas – b)) marca em P_1 , sem, no entanto, alterar, por via desse disparo, a marcação de P_1



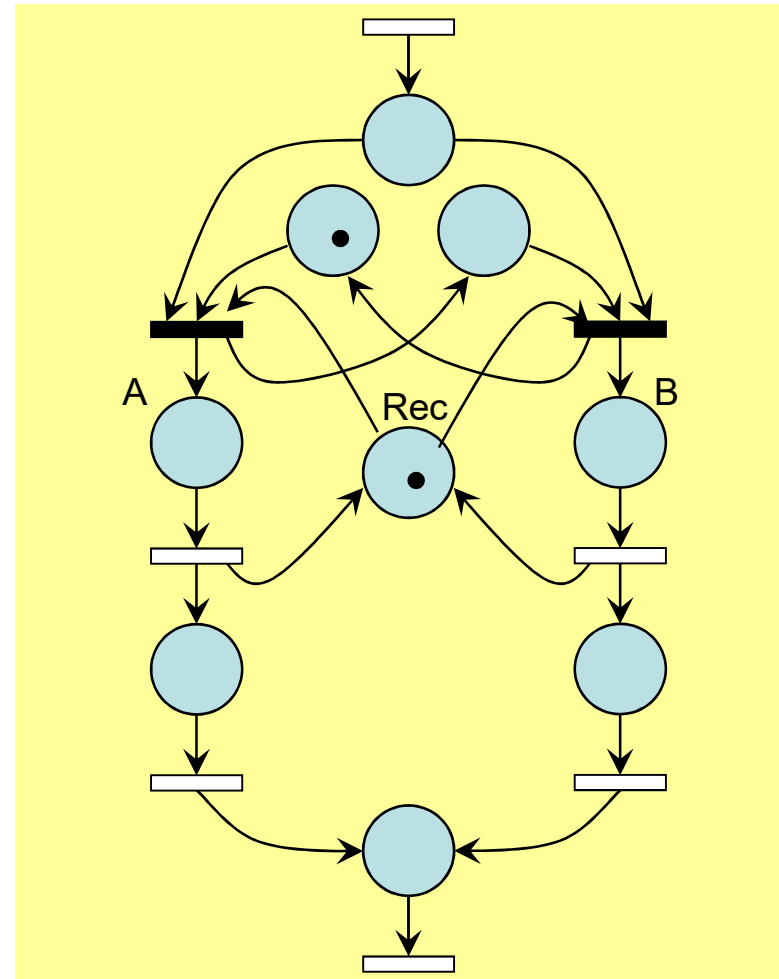
Componentes de Modelação (8)

- limitação de capacidade
 - vários exemplos de limitação de capacidade de P_1 a 3 marcas



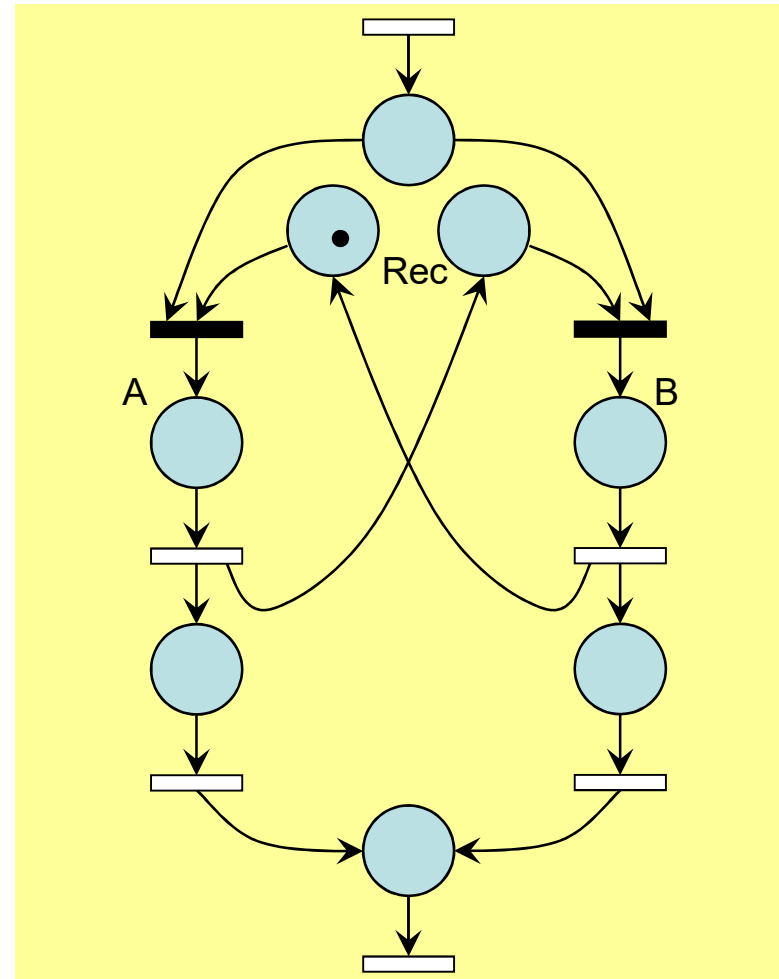
Componentes de Modelação (9)

- alternância
 - os processos A e B, utilizam, em exclusão mútua, o recurso *Rec*, e em alternância



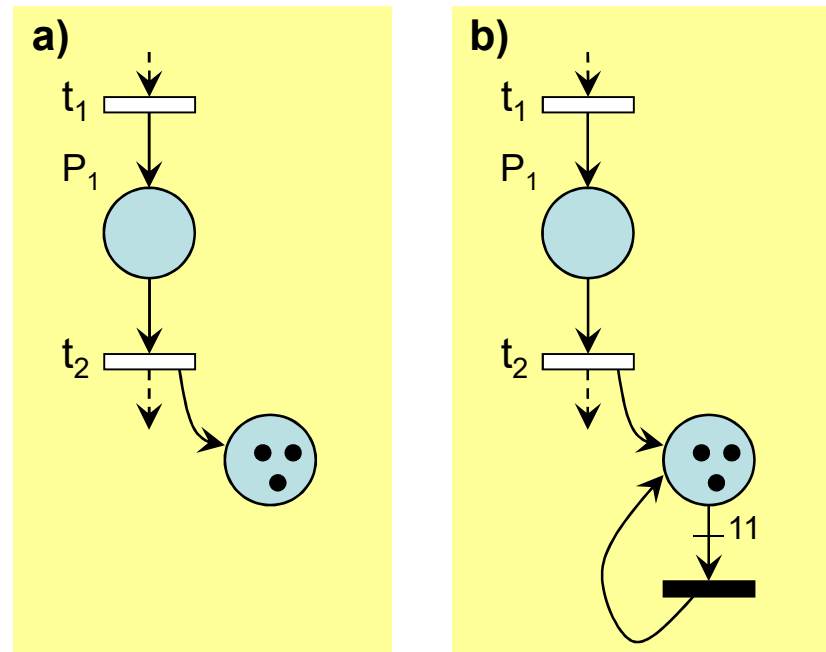
Componentes de Modelação (10)

- alternância (2)
 - o que está na página anterior, é alternância com exclusão mútua:
 - pode ser simplificado...



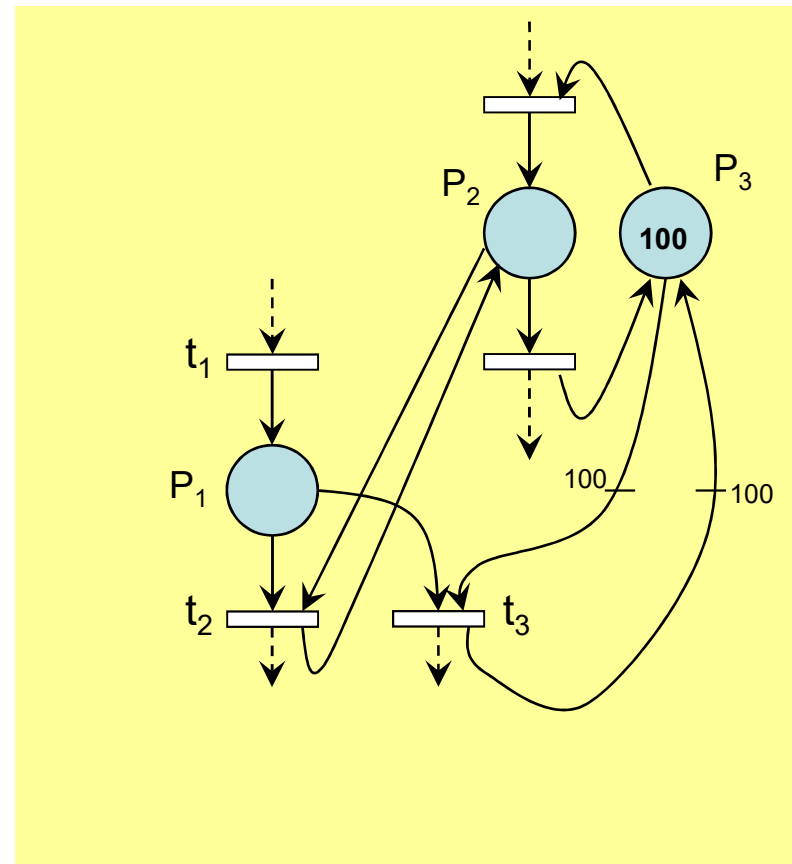
Componentes de Modelação (11)

- contador
 - em b) com *overflow* em 10



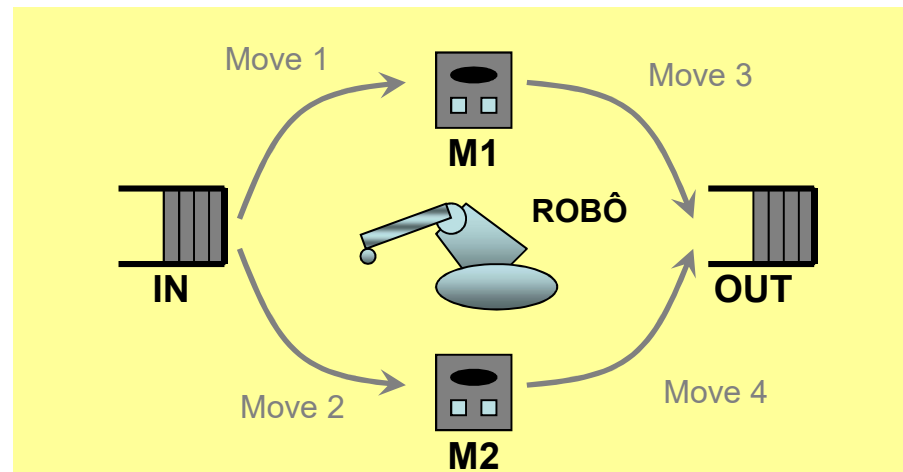
Componentes de Modelação (12)

- leitura de zero marcas
 - leitura de P_2 : se P_2 tiver uma marca, é disparada t_2 ; se tem zero é disparada t_3
 - P_3 funciona não como limitador mas como o “valor máximo” de P_2
 - se P_3 tiver 100 marcas, P_2 terá 0 marcas



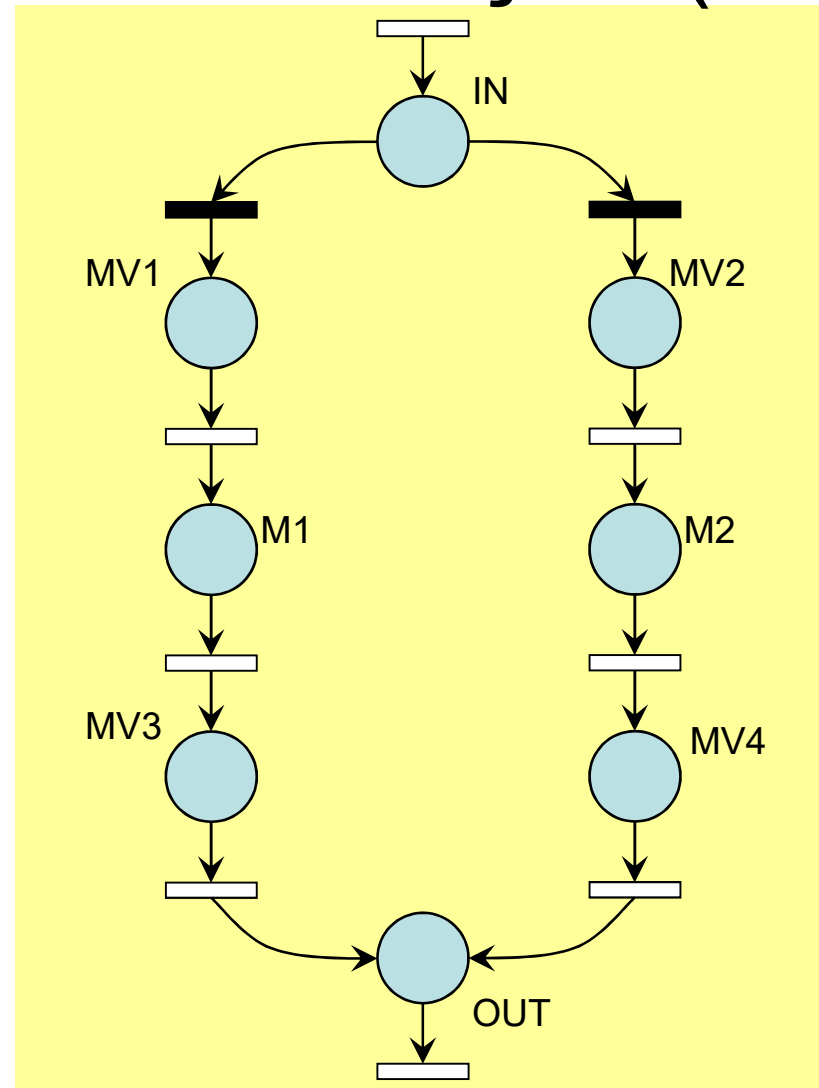
Componentes de Modelação (13)

- vamos agora ilustrar alguns destes conceitos com um exemplo
 - admita a análise e especificação de um software de controlo de um sistema industrial computadorizado
 - as peças a processar entram no sistema num *buffer* de entrada (IN) e podem ser processadas ou na máquina 1 (M1) ou na máquina 2 (M2), ambas com capacidade 1; as peças maquinadas são colocadas num *buffer* de saída (OUT); as 4 operações de transporte e manipulação são executadas por um robô



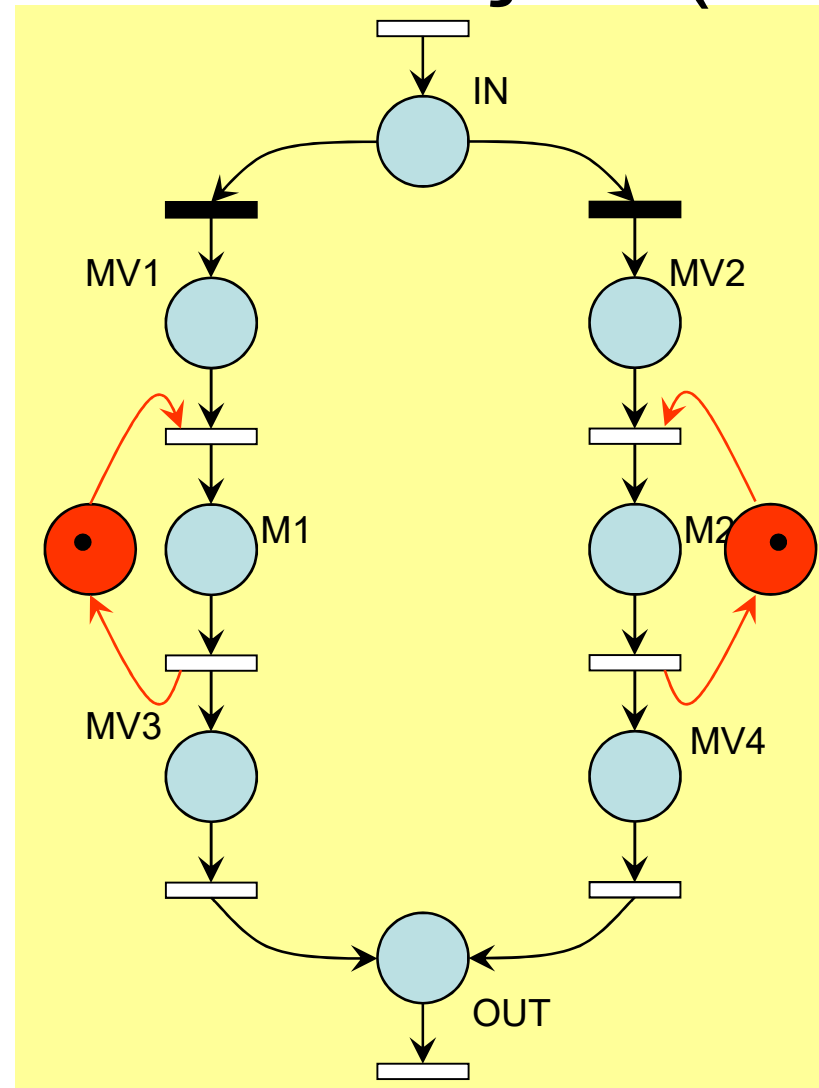
Componentes de Modelação (14)

- modelo RdP
 - o fluxo das operações, o “OU” máquina 1 “OU” 2, permitem facilmente esboçar o esqueleto da rede



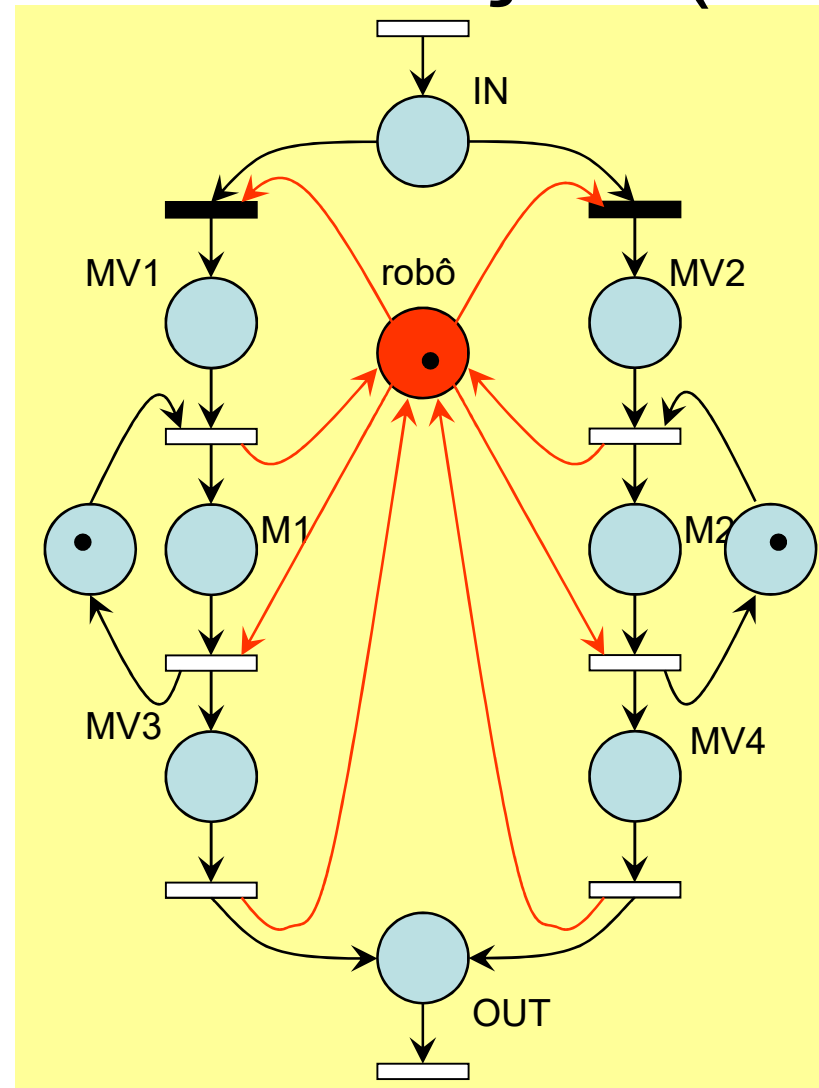
Componentes de Modelação (15)

- modelo RdP
 - o fluxo das operações, o “OU” máquina 1 “OU” 2, permitem facilmente esboçar o esqueleto da rede
 - acrescentar a noção de capacidade limitada das máquinas



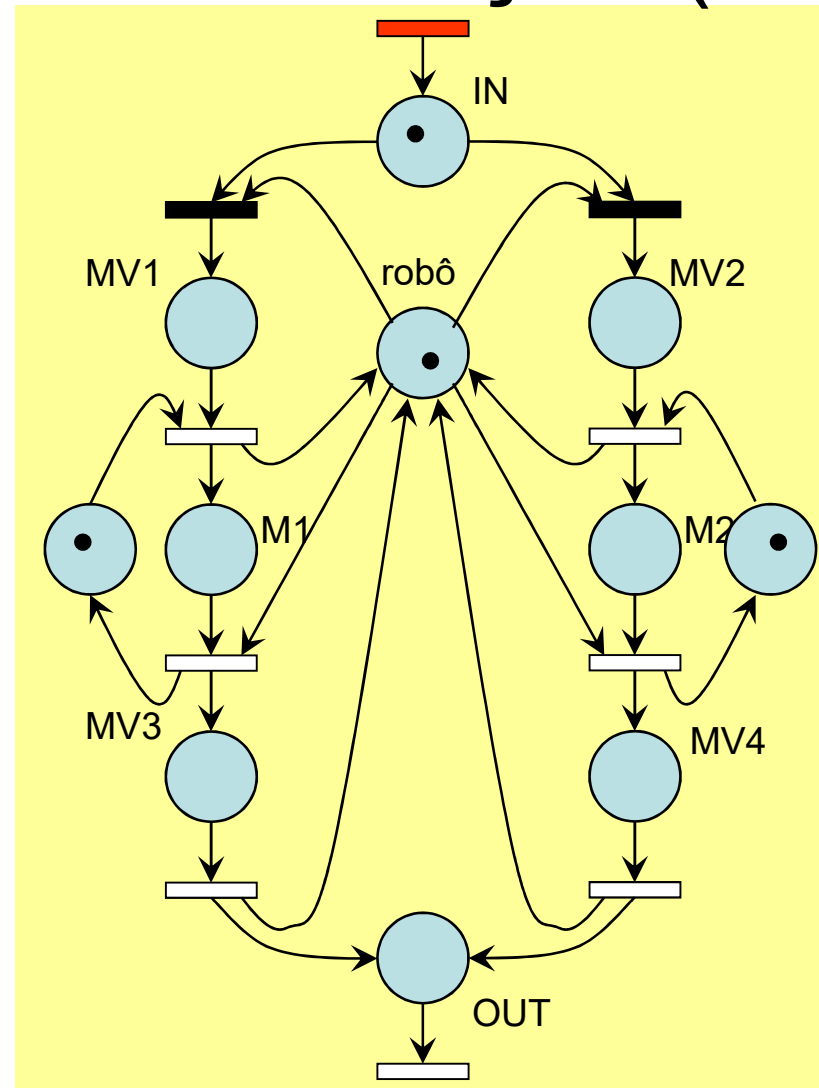
Componentes de Modelação (16)

- modelo RdP
 - o fluxo das operações, o “OU” máquina 1 “OU” 2, permitem facilmente esboçar o esqueleto da rede
 - acrescentar a noção de capacidade limitada das máquinas
 - acrescentar recurso partilhado (robô) pelas operações de movimento



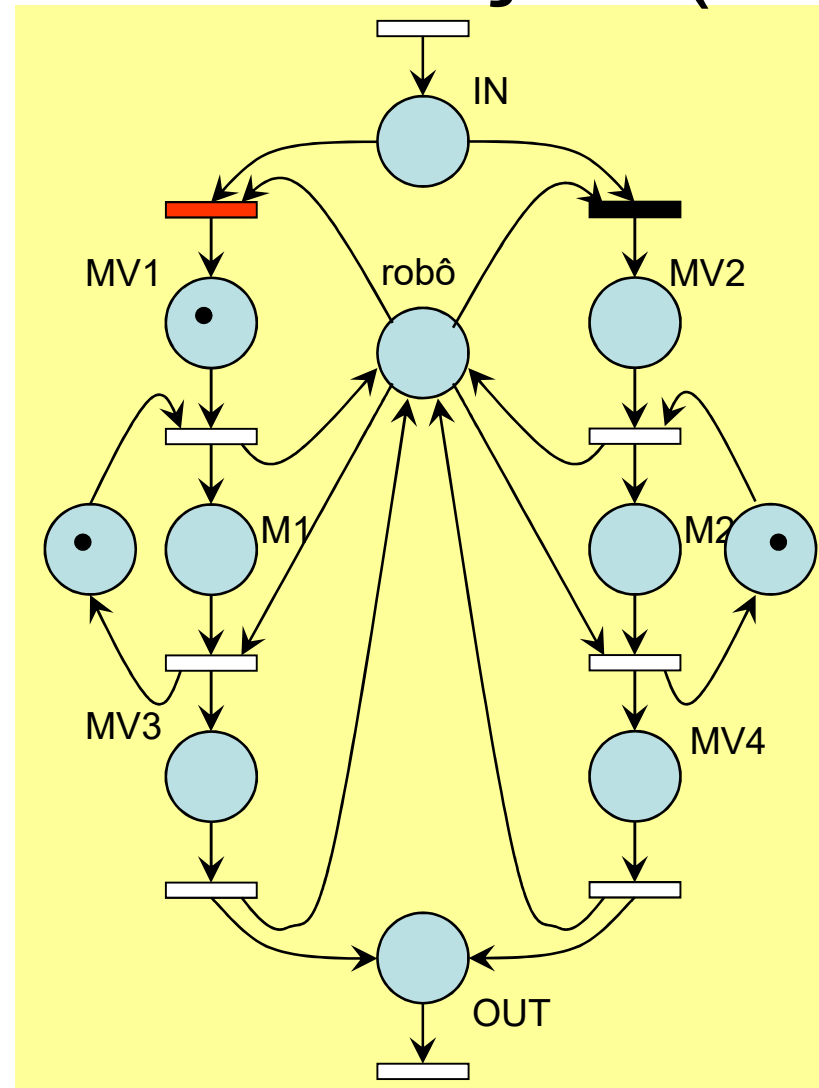
Componentes de Modelação (17)

- modelo RdP (análise)
 - o raciocínio parece correcto, mas a análise permite verificar que a especificação está errada:
 - admitindo que chega um componente ao *buffer* de entrada (IN)



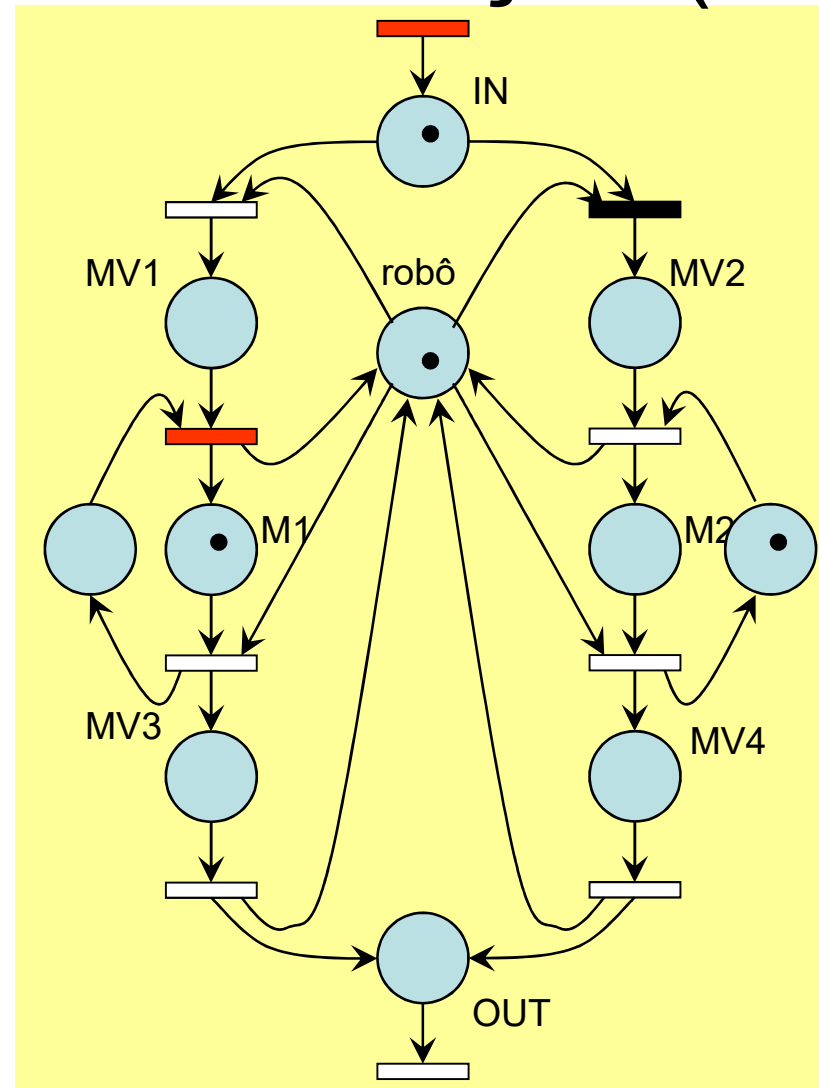
Componentes de Modelação (18)

- modelo RdP (análise)
 - esse componente pode ser processado na M1, pelo que é feita a operação de transporte MV1 (o recurso robô está disponível)



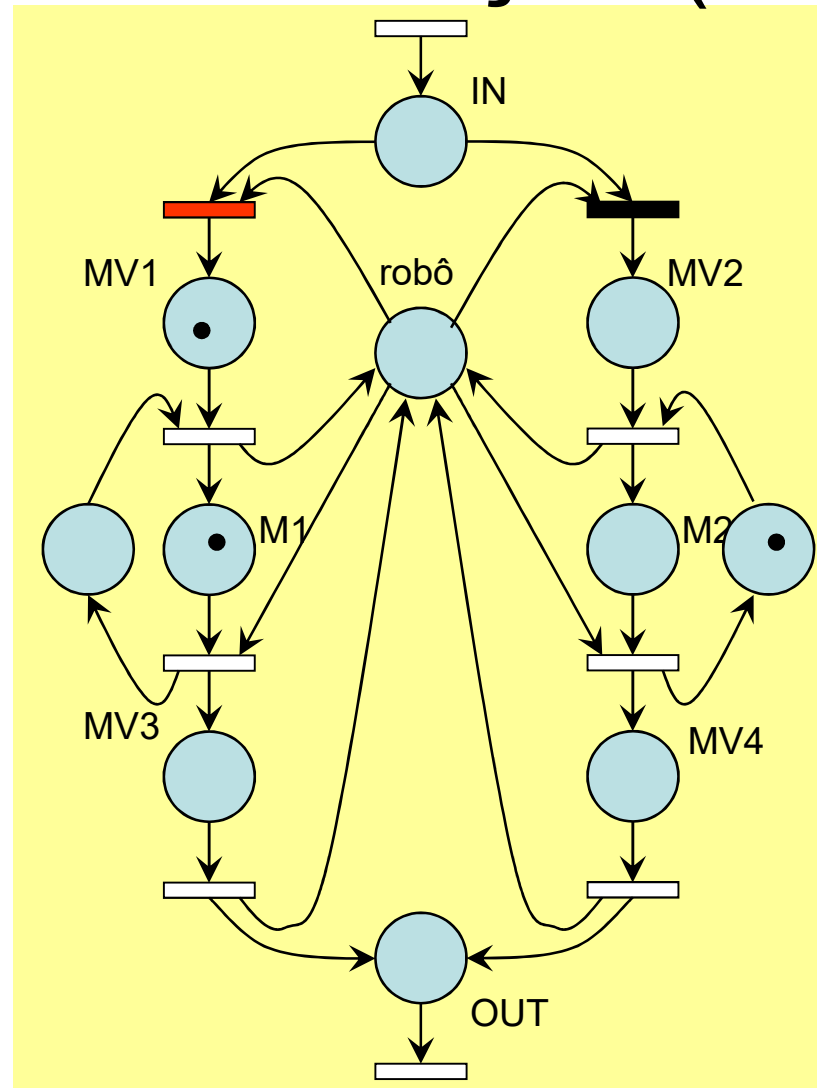
Componentes de Modelação (19)

- modelo RdP (análise)
 - finda a operação de transporte, o recurso robô é libertado e começa o processamento do componente em M1; e admita que entretanto chega um novo componente a IN



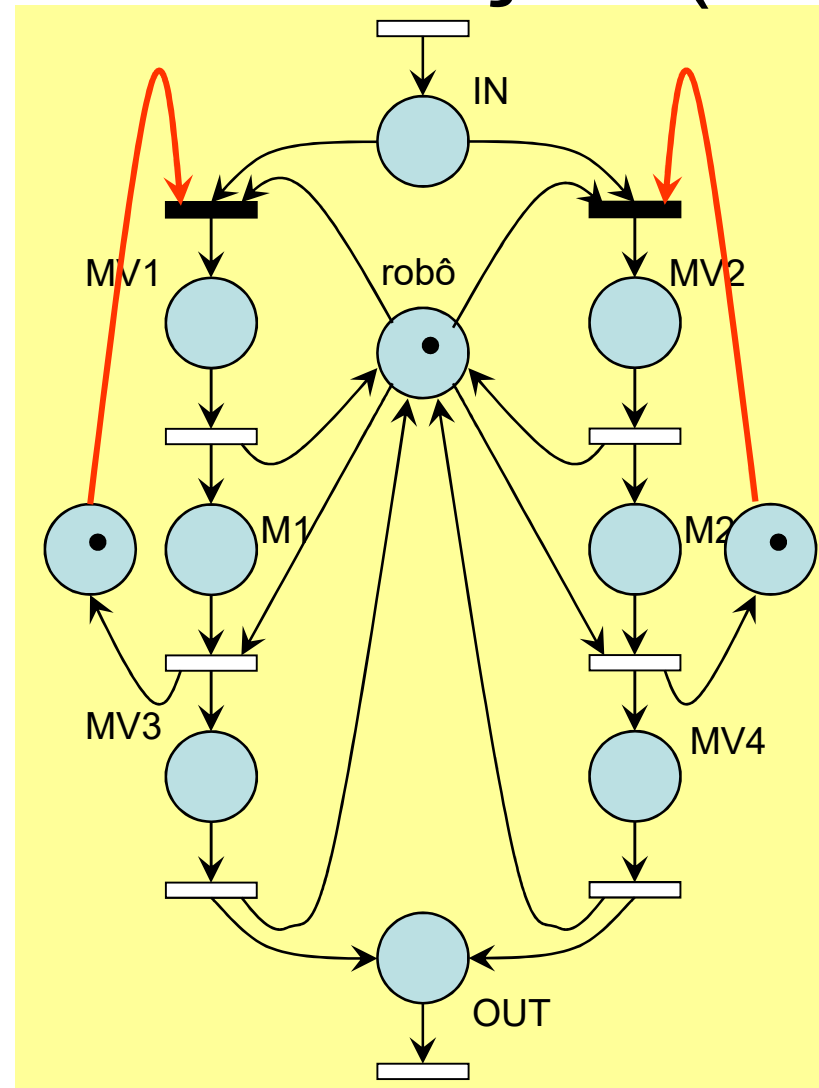
Componentes de Modelação (20)

- modelo RdP (análise)
 - esse novo componente poderá ser processado em M2, mas a especificação do modelo não impede que seja processado em M1, pelo que começa a ser feita a operação de transporte MV1
 - resulta uma situação de bloqueio!!!
 - a partir deste estado, só pode disparar a transição de entrada em IN



Componentes de Modelação (21)

- modelo RdP (correção do modelo)
 - a condição para iniciar o transporte para M1 (ou M2) deve ser ter robô disponível e máquinas vazias
 - esta RdP simples pôde ser facilmente analisada (e o problema identificado) por “trace” manual das evoluções possíveis
 - não exequível em modelos mais complexos, daí:
 - ferramentas computacionais que utilizam modelos de RdP para analisar um sistema



Índice

1. Introdução ✓
2. Noções Básicas de Redes de Petri (RdP) ✓
3. Regras de Evolução das RdP ✓
4. RdP Generalizadas ✓
5. Componentes de Modelação em RdP ✓
- 6. Análise Computacional de Modelos RdP**
7. Verificação de Propriedades dos Sistemas Modelados
8. Utilização da Ferramenta HP-SIM