

Engenharia de Automação Industrial

Bloco 4 - Modelagem de Sistemas a Eventos Discretos: Redes de Petri

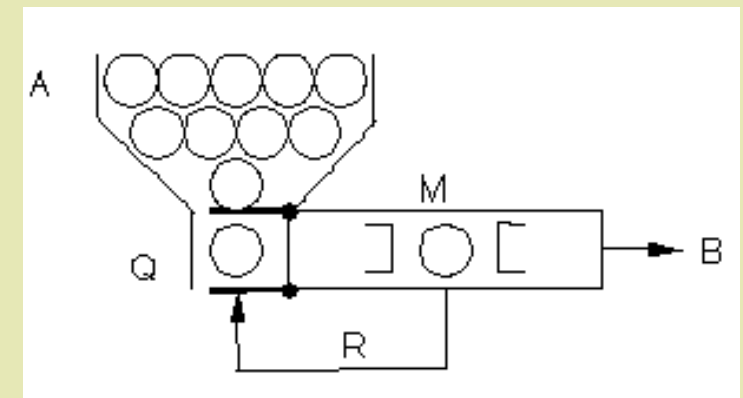
Sistemas a Eventos Discretos

Exemplo 1: Célula de manufatura

➤ Sistema automático de usinagem e inspeção, com retrabalho: quando o sistema Q está disponível, a primeira peça da fila de entrada é usinada por uma máquina M; em seguida a peça P é inspecionada e, se defeituosa, é reprocessada pela máquina, com prioridade em relação às outras peças da fila de entrada.

Os eventos de entrada do sistema são as alterações mostradas das variáveis Q, M, P, onde:

- a) Q = sistema livre ou ocupado;
- b) M = máquina livre ou ocupada;
- c) P = peça aprovada ou rejeitada pela inspeção.



Sistemas a Eventos Discretos

Exemplo 2: Terminal de caixa bancário

Os eventos de entrada no sistema, gerados pelo cliente, provocam mudanças de estado no terminal. Esses eventos são:

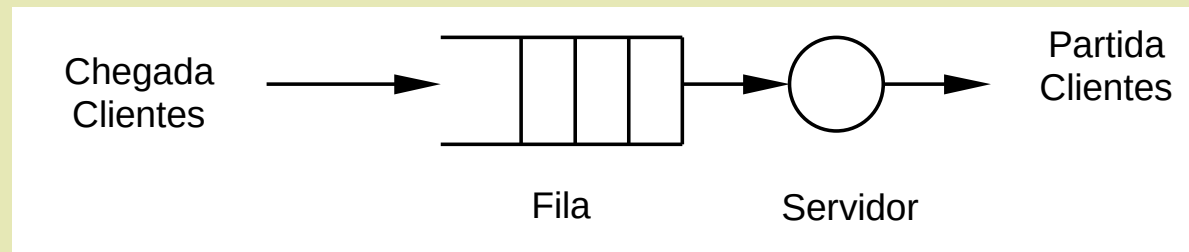
- a) Entrada do usuário no caixa;
- b) Passagem do cartão magnético;
- c) Digitação da senha;
- d) Digitação da operação de saldo;
- e) Digitação da operação de extrato;
- f) Digitação da data do extrato;
- g) Digitação da operação de saque;
- h) Digitação do valor do saque;
- i) Digitação de fim de operação.

Sistemas a Eventos Discretos

Exemplo 3: Unidade de atendimento de serviço ou de produção

Há sempre 3 elementos básicos:

- a) as entidades (pessoas ou peças) que chegam e esperam - os clientes
- b) os recursos de atendimento ou produção, limitados, que geram a espera - os servidores
- c) o espaço onde ocorre a espera - a fila
- d) os eventos são as chegadas e as saídas de clientes.

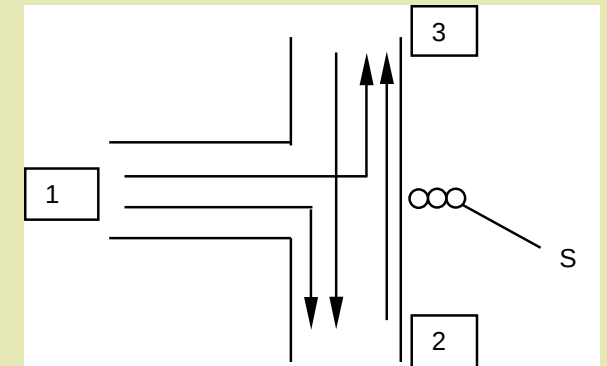


Sistemas a Eventos Discretos

Exemplo 4: Semáforo num cruzamento de ruas em T

As chegadas c_{ij} e as saídas s_{ij} de veículos são eventos; há 4 tipos de eventos ou trajetos:

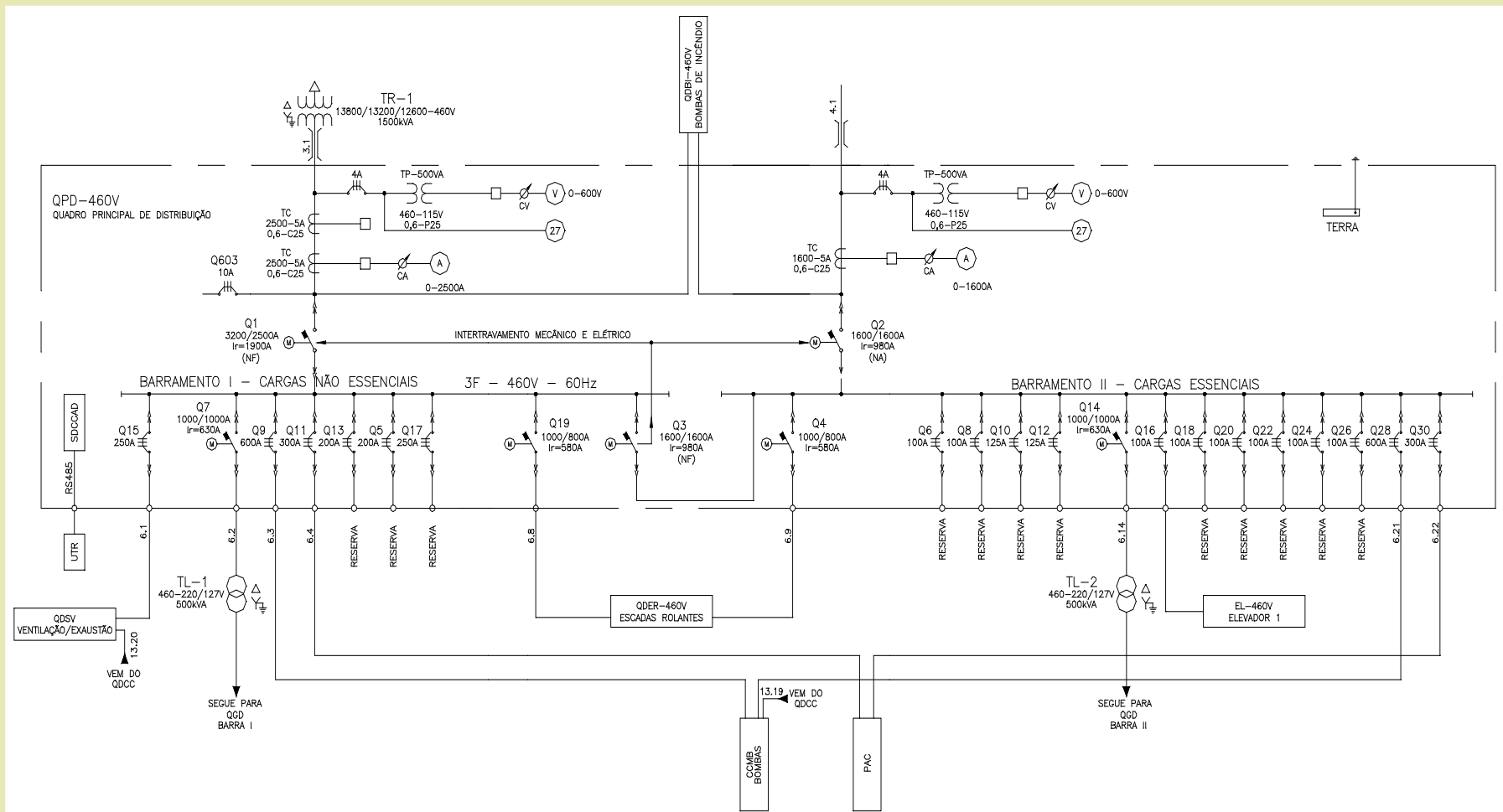
- a) $(1,2)$ = veículos vindo de 1 e virando para 2
- b) $(1,3)$ = veículos vindo de 1 e virando para 3
- c) $(2,3)$ = vindo de 2 e indo para 3
- d) $(3,2)$ = vindo de 3 e indo para 2



Há duas situações para o semáforo S:

- a) mostrando vermelho r para $(1,2)$ e $(1,3)$, verde g para $(2,3)$ e $(3,2)$, ou
- b) mostrando verde g para $(1,2)$ e $(1,3)$, vermelho r para $(2,3)$ e $(3,2)$.

Diagrama Unifilar do QPD



Redes de Petri

➤ Carl A. Petri criou este método de estudo dos sistemas dinâmicos a eventos discretos, em uma tese de doutoramento dedicada à Comunicação entre Autômatos (1962).

As Redes de Petri destacam-se na engenharia atual pelas seguintes qualidades:

- capturam as relações de precedência e os vínculos estruturais dos sistemas reais;
- são graficamente expressivas;
- modelam conflitos e filas;
- têm fundamento matemático e prático;
- admitem várias especializações (RPs temporizadas, coloridas, estocásticas, de confiabilidade etc.).

Redes de Petri

- Vamos definir Redes de Petri por meio de conjuntos, funções e também por grafos, de maneira que seu estudo pode decorrer da teoria dos conjuntos e/ou da teoria dos grafos.
- Uma Rede de Petri é um grafo orientado que tem dois tipos de nós: transições (transitions) e posições (places).
- Os arcos do grafo partem de algumas posições para algumas transições ou vice-versa; aos arcos associam-se números (inteiros) fixos, que são seus pesos.
- Cada posição pode conter um número (inteiro) de marcas (tokens), e estas, sob certas condições, podem mover-se ao longo dos arcos, respeitados os sentidos destes.

SISTEMAS DINÂMICOS A EVENTOS DISCRETOS

Redes de Petri : Por que modelar sistemas?

- Necessidade de conhecer as características dos processos industriais para estabelecer as ESPECIFICAÇÕES FUNCIONAIS dos mesmos
- Analisar o desempenho da planta quanto as características dos equipamentos disponíveis obtendo uma utilização otimizada dos mesmos, **EVITANDO CONFLITOS NO PROCESSO**
- Orientar de forma rápida programação dos CLPs, projetando controladores com programação otimizada de AUTOMAÇÃO
- Permitir a observação da planta quando das formas de operação:

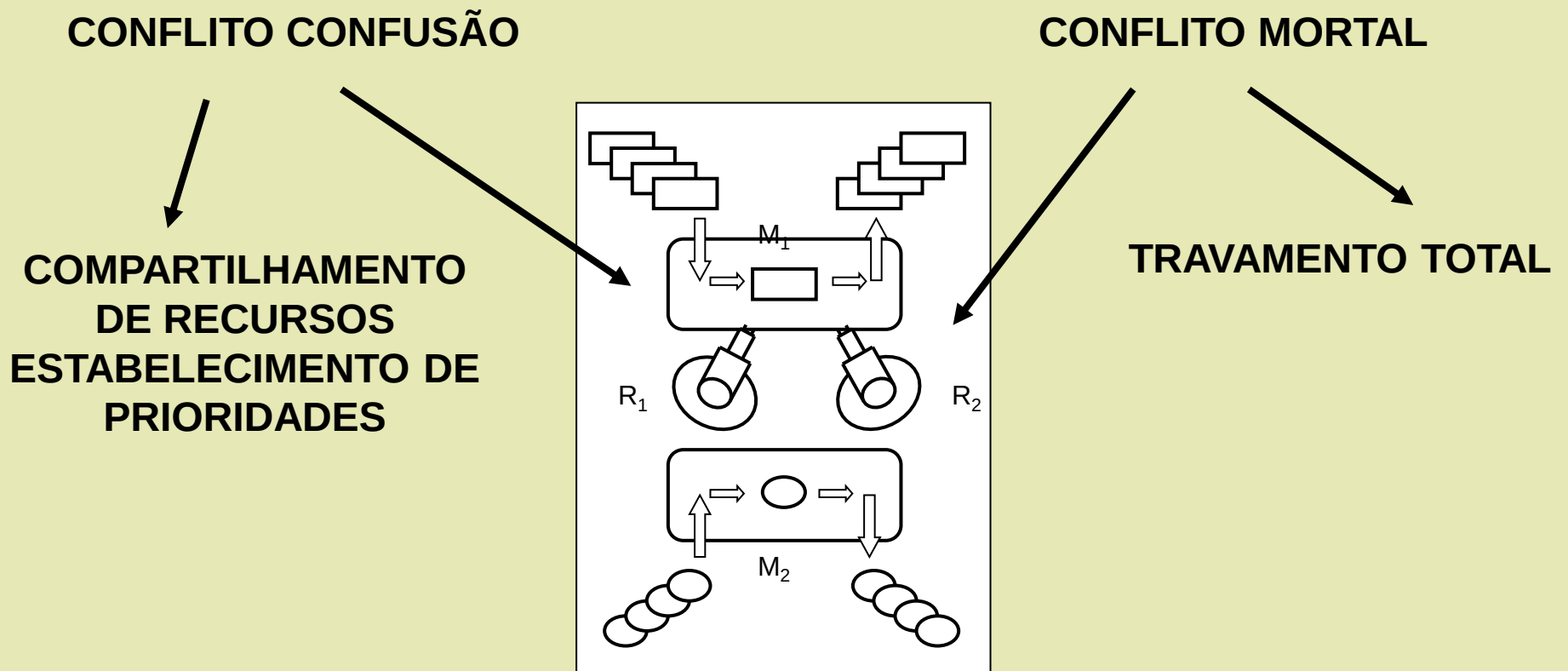
NORMAL – SEM APRESENTAÇÃO DE NENHUMA ANOMALIA INTERNA OU EXTERNA AO PROCESSO

ANORMAL - COM DEFEITOS OU FALHAS JUNTO AO PROCESSO

EMERGENCIAL – NECESSITANDO A INTERVENÇÃO IMEDIATA DO OPERADOR

Redes de Petri: Objetivos da Rede Petri na Automação Industrial

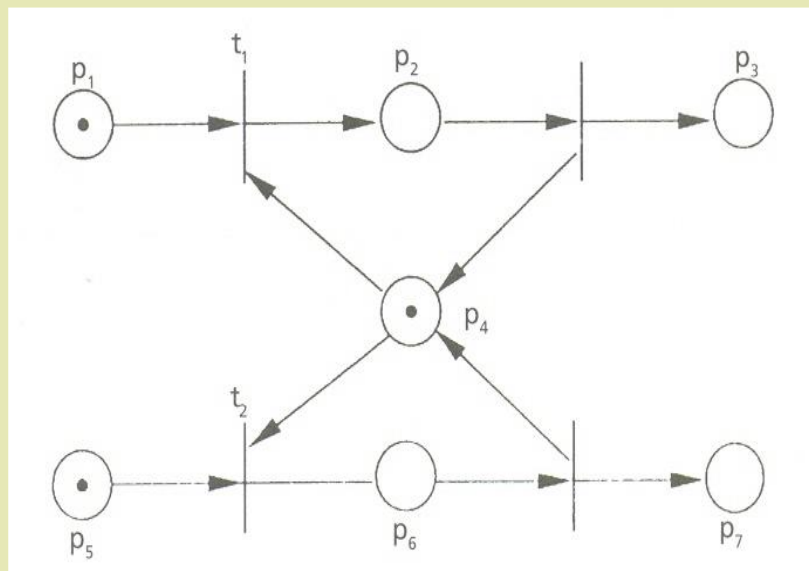
- 1) Permitir ao engenheiro a visão correta para a programação do controlador indicando todas as memórias físicas e virtuais necessárias ao programa aplicativo (ladder/sfc/fb)
- 2) Permitir que o programa opere normalmente diante dos conflitos confusão estabelecendo as prioridades e evite os conflitos mortais (deadlock).



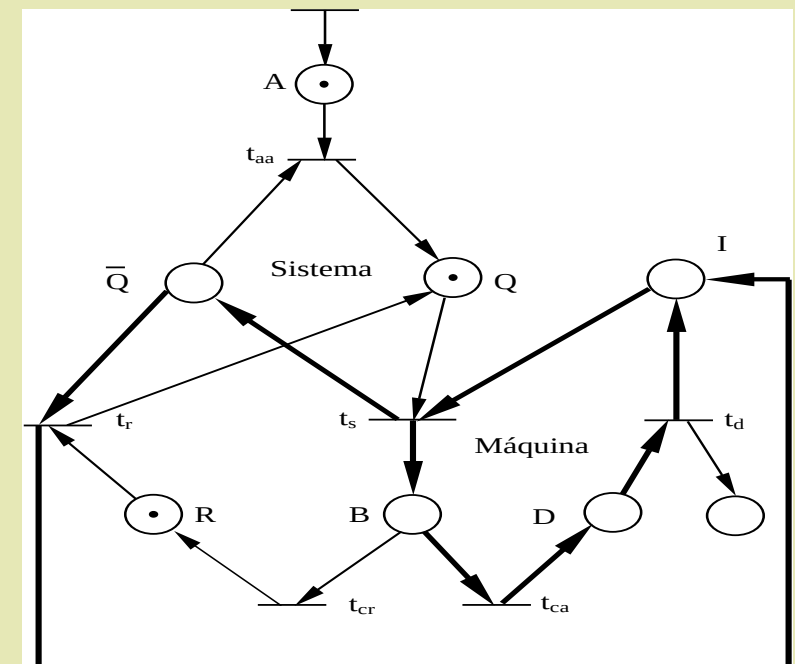
Redes de Petri : Objetivos da Rede Petri na Automação Industrial

- 1) Permitir ao engenheiro a visão correta para a programação do controlador indicando todas as memórias físicas e virtuais necessárias ao programa aplicativo (ladder/sfc/fb)
- 2) Permitir que o programa opere normalmente diante dos conflitos confusão e evite os conflitos mortais (deadlock).
- 3) Garanta transição de operações normal / anormal / emergencial

CONFLITO CONFUSÃO



CONFLITO MORTAL



Redes de Petri

➤ Definição formal de *Rede de Petri*:

- Uma Rede de Petri (RP) é uma quintupla (P, T, A, W, m_0) em que

· $P = \{p_1 \dots p_n\}$ é um conjunto finito de posições ou lugares

· $T = \{t_1 \dots t_m\}$ é um conjunto finito de transições

· A é um conjunto finito de arcos pertencente ao conjunto $(P \times T) \cup (T \times P)$, em que $(P \times T)$ representa o conjunto dos arcos orientados de p_i para t_j , também designados por (p_i, t_j) , e $(T \times P)$ representa o conjunto dos arcos orientados de t_i para p_j , ou (t_i, p_j)

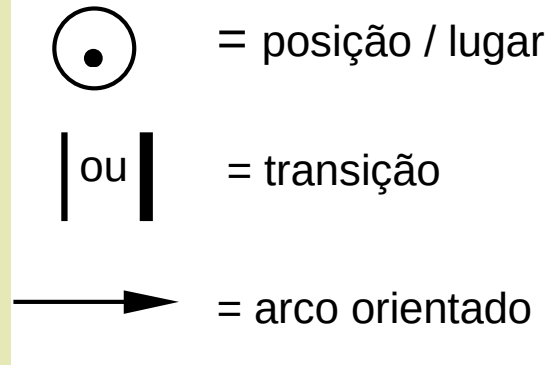
· W é a função que atribui um peso w (um número inteiro) a cada arco

· m_0 é um vetor cuja i -ésima coordenada define o número de marcas (tokens) na posição p_i , no início da evolução da rede.

Redes de Petri

- Os conjuntos T e P são disjuntos, i.é., $T \cap P = \emptyset$
- $n = |P|$ é a cardinalidade do conjunto P , o número de posições da RP.
- $m = |T|$ é o número de transições da RP.
- Em certos autores, o conjunto dos arcos A é substituído por dois conjuntos de funções: o das funções ou mapas de saída (O), que definem mapeamentos das transições a subconjuntos de posições; o das funções ou mapas de entrada (I), que definem mapeamentos das posições para subconjuntos de transições.
- Ao modelar processos industriais, as marcas (tokens) são utilizadas para representar quantidades de entidades, recursos ou peças, “residentes” nas posições ou “movidas” através das transições.

Redes de Petri

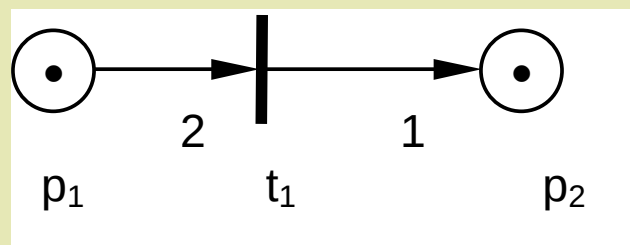


Exemplo 1: Seja uma RP definida conforme a quintupla $Q = (P, T, A, W, m_0)$ onde

$P = \{p_1, p_2\}$, $T = \{t_1\}$, $A = \{(p_1, t_1), (t_1, p_2)\}$

W : $w(p_1, t_1) = 2$, $w(t_1, p_2) = 1$

$m_0 = [1 \ 1]$ pois $m(p_1) = 1$, $m(p_2) = 1$

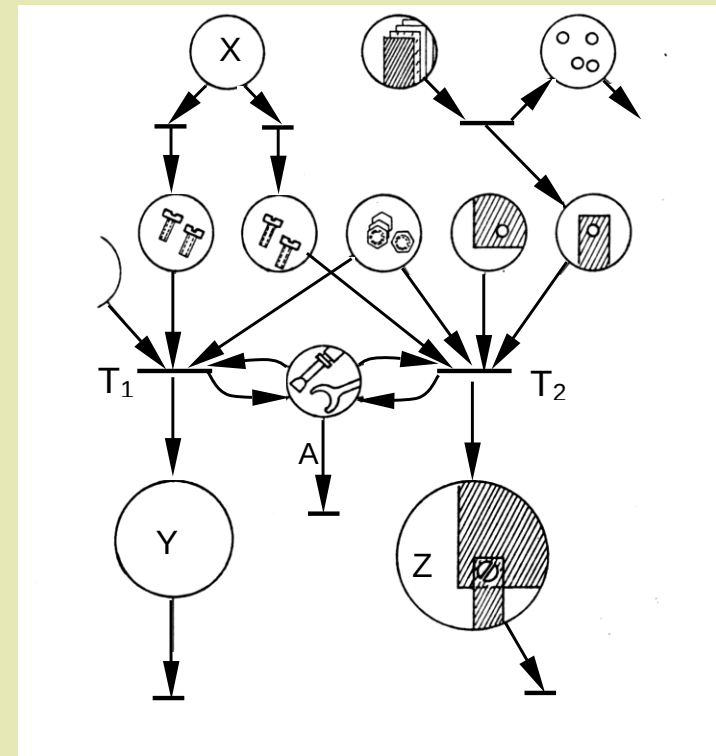


Redes de Petri

Exemplo 2:

A máquina X envia lotes de 100 parafusos a cada uma das duas operações de montagem T_1 e T_2 . Dos estoques, saem arruelas e peças previamente furadas; de um almoxarifado A, saem as ferramentas necessárias. Em Y e em Z, são armazenados subconjuntos montados.

Os discos são posições que representam estoques de peças ou de ferramentas, ou máquinas; as barras são transições e representam ações de montagem ou de transporte; os arcos direcionados indicam fluxos de marcas dentro do sistema.

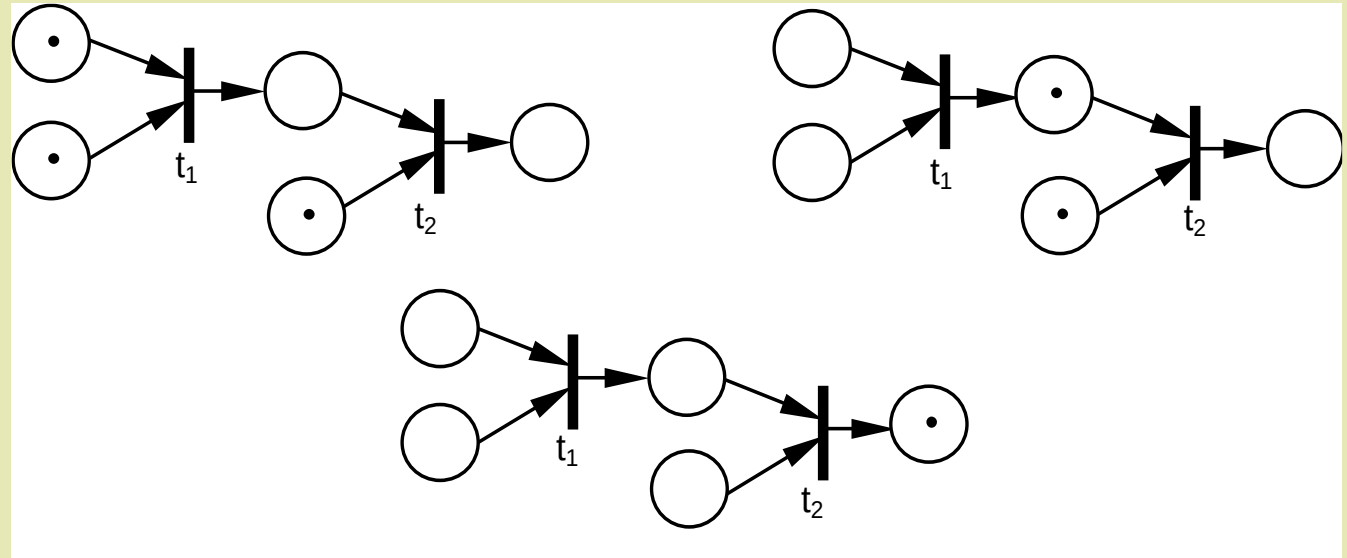


Redes de Petri

Exemplo 2: continuação

- ❖ As marcas na RP indicam quantidades de objetos nos estoques ou nas máquinas (ou quantidades de lotes);
- ❖ Interessante e potencialmente útil é o fato de que, à medida que as marcas caminham na RP, a natureza física dos objetos que elas representam pode modificar-se:
 - ✓ *ora são lotes de peças (parafusos, porcas, arruelas) que se transportam dos estoques para as máquinas; ora são subconjuntos (Y, Z) recém-montados, que se transportam ou armazenam; ora são máquinas ou operários ocupados (X).*

Redes de Petri



Pré-sets e Pós-sets

Pré-set de $t := \bullet t := \{\forall p_i \in P / (p_i, t) \in A\}$

ou seja, o pré-set de t é o conjunto das posições em P a partir das quais existe arco para a transição t

Pós-set de $t := t \bullet := \{\forall p_i \in P / (t, p_i) \in A\}$

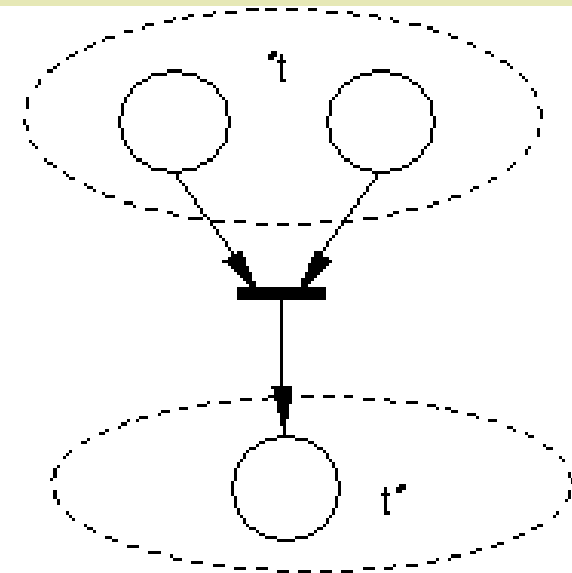
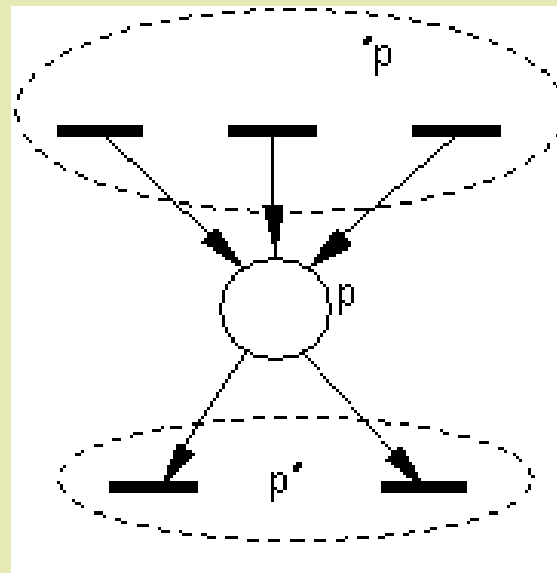
ou seja, o pós-set de t é o conjunto das posições em P para as quais existe arco oriundo da transição t .

Redes de Petri

Pré-sets e Pós-sets

Pré-set de $p := \bullet p := \{\forall t_j \in T / (t_j, p) \in A\}$

Pós-set de $p := p \bullet := \{\forall t_j \in T / (p, t_j) \in A\}$.



Pré-sets e pós-sets

Execução das Redes de Petri

➤ Chama-se de execução da RP a movimentação das marcas pela rede de acordo com certas regras; ocorre em duas fases: habilitação e disparo de transição.

Uma transição $t_j \in T$ numa RP é habilitada por uma marcação m se ocorre que, para todo $p_i \in \bullet t_j$, $m(p_i) \geq w(p_i, t_j)$, isto é, (marcação em p_i) $\geq$ (peso do arco de p_i a t_j).

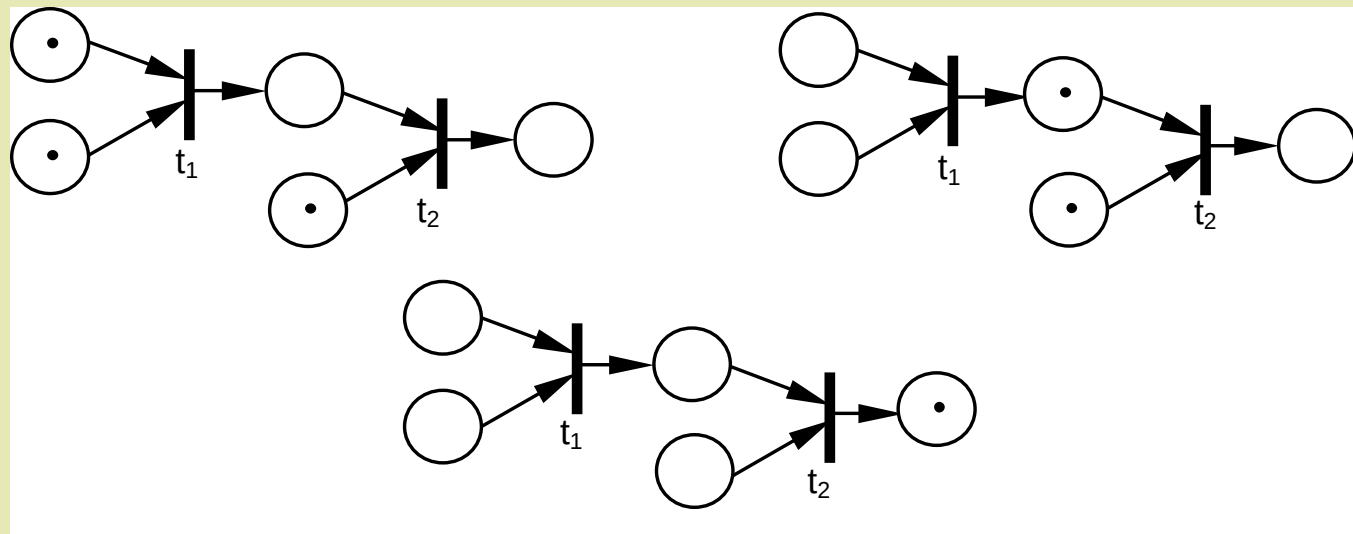
Uma transição é disparada por meio de duas operações:

- a) remoção das marcas das posições do pré-set (tantas marcas quanto for o peso do arco correspondente), e
- b) depósito, em cada uma das posições do pós-set, de tantas marcas quanto for o peso do arco correspondente.

Execução das Redes de Petri

➤ *Exemplo 1:*

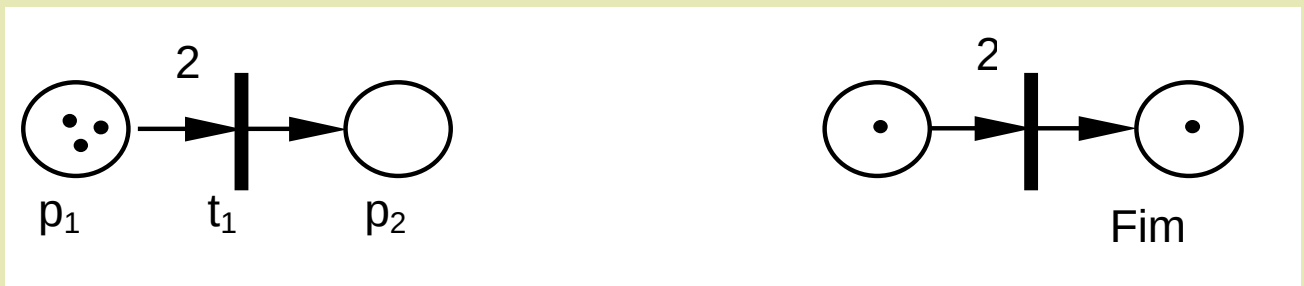
□ Na Figura , dada a marcação inicial, ocorrem duas execuções sucessivas, mediante a seqüência de disparo $t_1 t_2$.



Execução das Redes de Petri

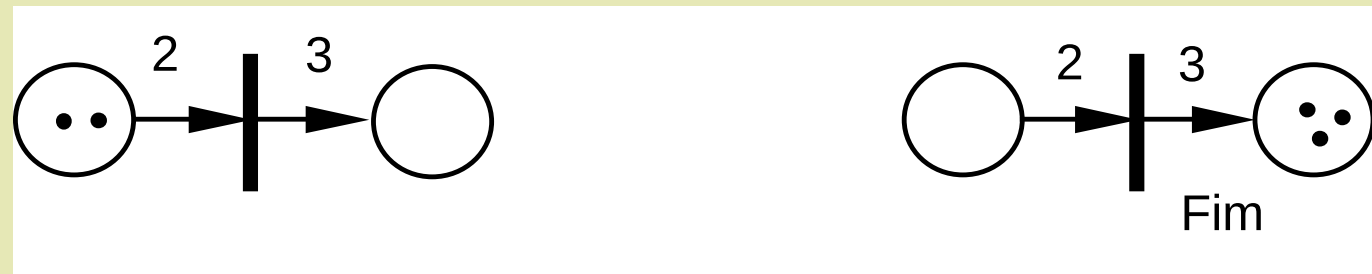
Exemplo 2:

Na Figura, um único disparo é possível; ele retira 2 marcas de p_1 e coloca 1 marca em p_2 .



Exemplo 3:

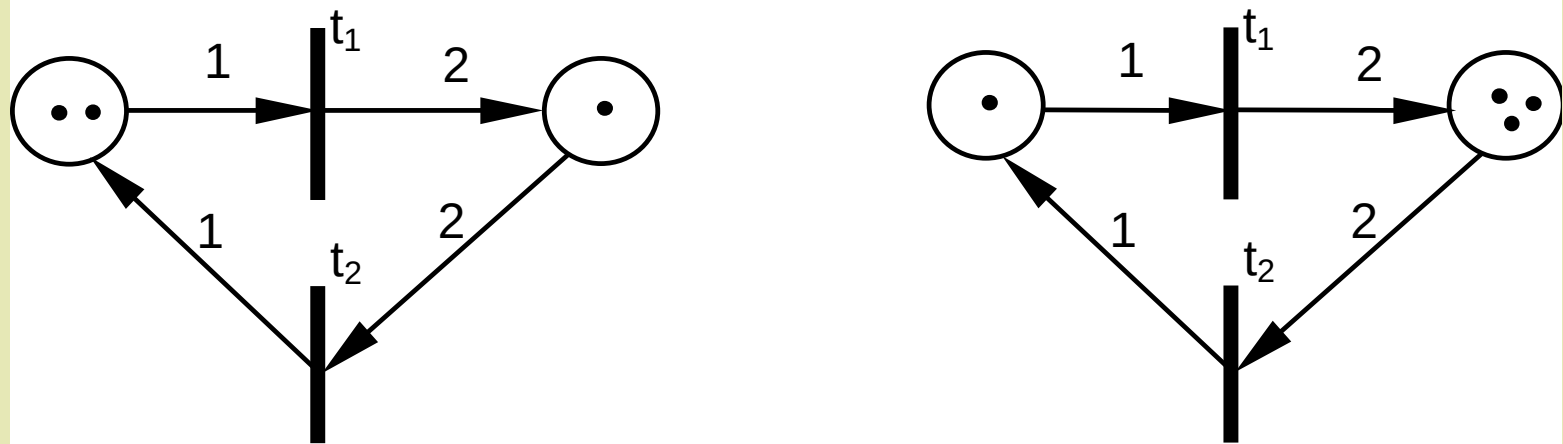
Na Figura, um único disparo é possível; retira 2 marcas de p_1 e põe 3 marcas em p_2 .



Execução das Redes de Petri

Exemplo 4:

Na Figura, a seqüência de disparos $t_1 t_2 t_1 t_2 t_1 \dots$ gera apenas os dois diferentes estados mostrados.



Execução das Redes de Petri

Alguns Sistemas a Eventos e Suas RPs

Exemplo 4: Terminal de caixa bancário automático

□ *Condições do Sistema (Posições):*

- a) p_1 Caixa com cliente encontra-se à espera do serviço (em prontidão)
- b) p_2 Caixa encontra-se à espera da senha
- c) p_3 Caixa encontra-se em apresentação de menu de serviços (saldo, saque ou extrato)
- d) p_4 Caixa em espera para data do extrato
- e) p_5 Caixa em espera para valor do saque
- f) p_6 Caixa executando o serviço

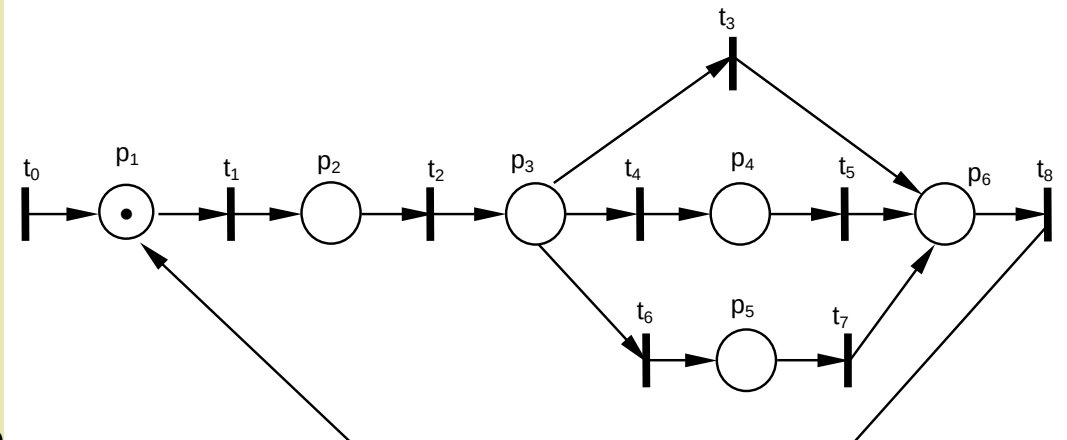
Execução das Redes de Petri

Alguns Sistemas a Eventos e Suas RPs

Exemplo 4: Terminal de caixa bancário automático – *continuação*

➤ Eventos do sistema

- a) t_0 Entrada do usuário no caixa
- b) t_1 Passagem do cartão magnético
- c) t_2 Digitação da senha
- d) t_3 Digitação da operação de saldo
- e) t_4 Digitação da operação de extrato
- f) t_5 Digitação da data do extrato
- g) t_6 Digitação da operação de saque
- h) t_7 Digitação do valor do saque
- i) t_8 Digitação de fim de operação

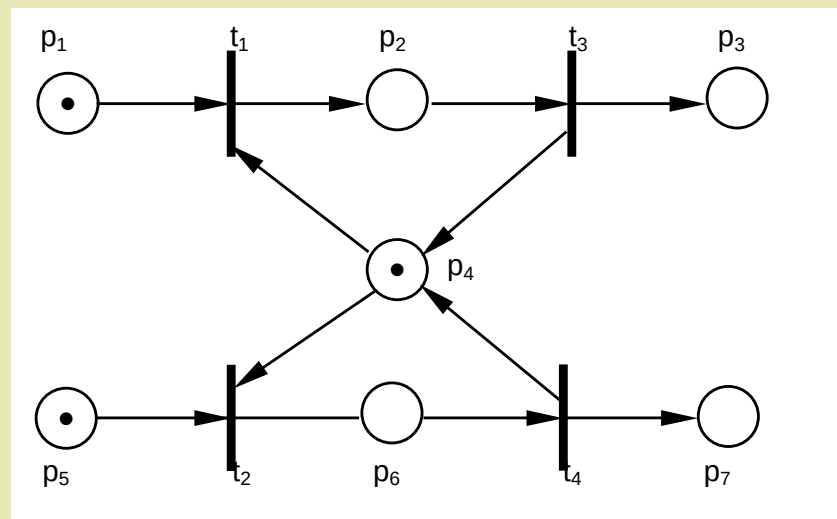


Execução das Redes de Petri

Alguns Sistemas a Eventos e Suas RPs

Exemplo 2: Recursos compartilhados

□ Um robo transportador disponível é representado por marca na posição p_4 da RP da Figura; a correia participa de ambas as linhas de produção (p_1, p_2, p_3) e (p_5, p_6, p_7). Ocorrendo a marcação da figura, tanto t_1 quanto t_2 estão habilitadas, e o disparo de uma inabilita a outra; tem-se um conflito-confusão.



SISTEMAS DINÂMICOS A EVENTOS DISCRETOS

Análise de Desempenho

Índices

As propriedades usualmente examinadas por motivos operacionais, são:

- **Limitação e Segurança;**
- **Conservação;**
- **Vivacidade e Conflito;**
- **Alcançabilidade;**
- **Reversibilidade.**

Análise de Desempenho

Uma RP é **viva**, dado um estado inicial x_0 , se e somente se todas as suas transições são vivas.

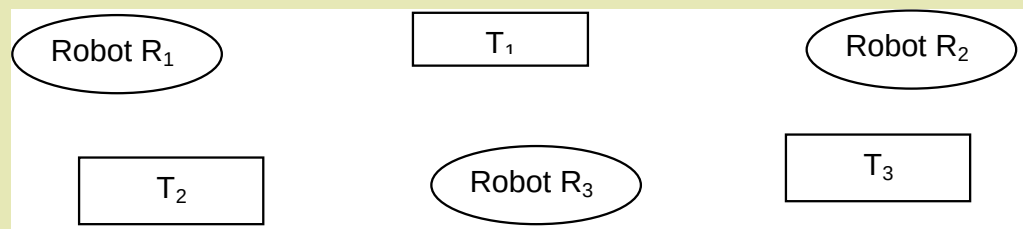
Conflito Confusão

Um problema operacional compartilhamento de recursos em sistemas a eventos discretos é o **conflito confusão**; são as desabilitações subsequentes; implicando em perda parcial da vivacidade, de uma forma particular. Eliminar conflitos confusão requer usualmente modificar a RP (e o sistema real), por meio de adequados estados e transições adicionais, normalmente priorizando ações para uma determinada limitação do hardware/equipamentos que operam em compartilhamento.

Análise de Desempenho

Exemplo 2: Célula de montagem flexível [Desrochers, 1995]

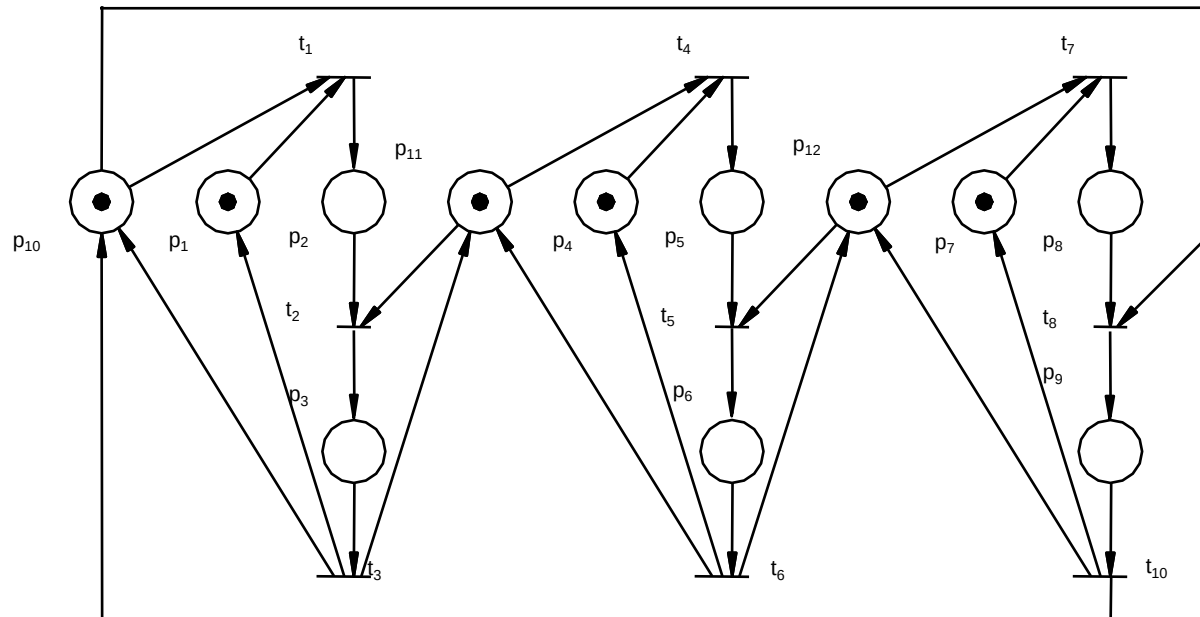
- Considere a célula de montagem automática da Figura: cada tarefa de montagem requer um transportador T e dois robots R, adjacentes a T. Cada transportador T requisita o robot a sua esquerda (esquerda de quem olha o sistema a partir do centro) e depois o robot de sua direita.



- Realizada a tarefa de montagem, o transportador libera os dois robots simultaneamente.

Análise de Desempenho

Exemplo : Célula de montagem flexível [Desrochers, 1995]



RP completa do sistema

Análise de Desempenho

Uma RP é **viva**, dado um estado inicial x_0 , se e somente se todas as suas transições são vivas.

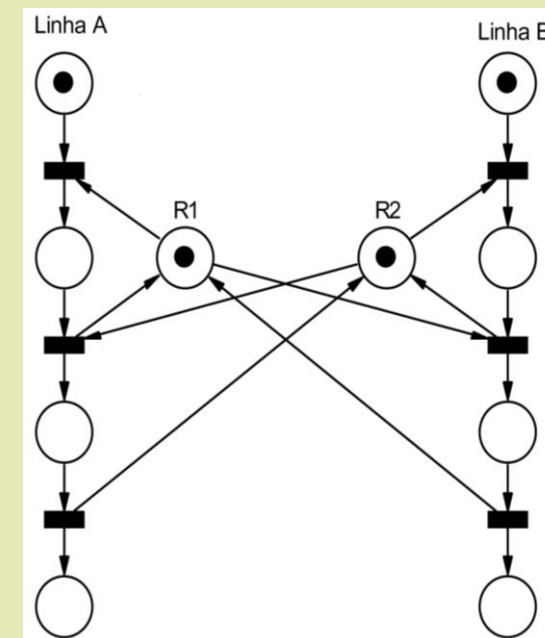
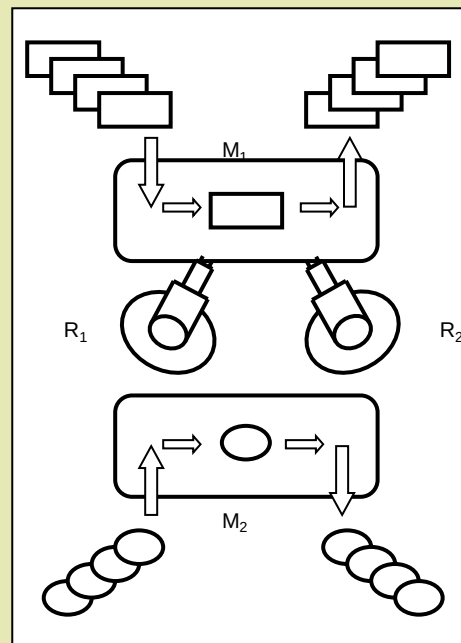
Conflito mortal (deadlock)

O principal problema operacional em sistemas a eventos discretos é o **conflito mortal** (deadlock); é a parada total de habilitações subsequentes; implica em perda da vivacidade, de uma forma particular. Eliminar conflitos mortais requer usualmente modificar a RP (e o sistema real), por meio de adequados estados e transições adicionais.

Análise de Desempenho

Exemplo 1:

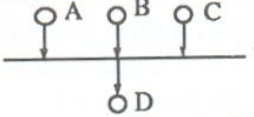
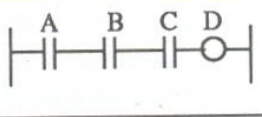
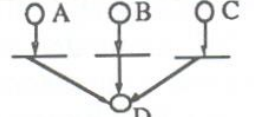
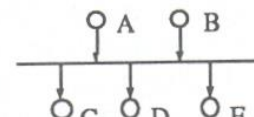
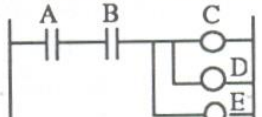
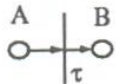
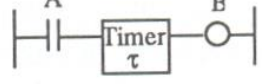
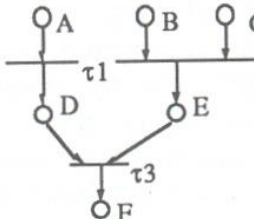
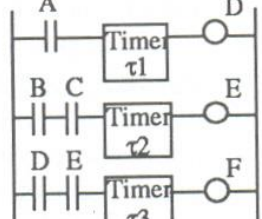
O cliente A prende o recurso R1 enquanto não recebe o recurso R2; o cliente B prende R2 enquanto não recebe R1; nessas condições, nem A nem B podem ser atendidos por qualquer dos dois recursos, e o sistema pára.



Relações entre Redes de Petri, Diagramas Lader e Linguagem Lader

Redes de Petri: Objetivos da Rede Petri na Automação Industrial

- 1) Permitir ao engenheiro a visão correta para a programação do controlador indicando todas as memórias físicas e virtuais necessárias ao programa aplicativo (ladder/sfc/fb)
- 2) Permitir que o programa opere normalmente diante dos conflitos confusão e evite os conflitos mortais (deadlock).

Logic constructs	Petri nets	Ladder logic diagrams
Condition or the status of a system element	○ Place	No explicit representation
An activity	—— Transition	No explicit representation
Flow of information or material.	——> Directed arc	No explicit representation
Objects such as machines, robots, pallets, etc.	○ Token(s) in place(s)	No explicit representation
<u>Logical AND</u> IF A = 1 and B = 1 and C = 1 THEN D = 1		
<u>Logical OR</u> IF A = 1 or B = 1 or C = 1 THEN D = 1		
<u>Concurrency</u> IF A = 1 and B = 1 THEN C = 1 and D = 1 and E = 1		
<u>Time delay</u> IF A = 1 THEN delay "τ time units" B = 1		
<u>Synchronization</u> IF A = 1 THEN delay "τ1 time units"; D = 1 IF B = 1 and C = 1 THEN delay "τ2 time units"; E = 1 IF D = 1 and E = 1 THEN delay "τ3 time units"; F = 1		

SISTEMAS DINÂMICOS A EVENTOS DISCRETOS

O que há de comum nos equipamentos discretos ?

- Os estados normalmente são dois: ligado / desligado
aberto/ fechado
0/1

Ex: disjuntores, contadores, relés
servoválvulas, valv pneumáticas

- Alguns casos necessitam definição mais completa para caracterizar os estados possíveis:
 - Motor ligado, desligado, acelerando, frenando, em falha, sobreaquecido. etc.

Sistemas a Eventos Discretos

Exemplo : Automação eletro-pneumática

- Em manufaturas automatizadas é usual que pistões sejam utilizados para carregar/descarregar peças da mesa de trabalho, mover ferramentas de corte em máquinas-ferramenta, etc.
- Um sistema de pistões consiste de 3 pistões pneumáticos (A, B, C) que são acionados por eletro-válvulas.
- Quando energizada a bobina de uma eletro-válvula, esta comanda a ida do pistão correspondente para a direita (movimento designado, por exemplo, com A+); e quando desenergizada, para a esquerda (A-).
- Cada pistão tem os seus finais da excursão desejada sinalizados por chaves fim-de-curso; por exemplo, quando o pistão A chega ao final à esquerda, o contato a0 é fechado; analogamente, à direita, a1.

Sistemas a Eventos Discretos

Exemplo : Automação eletro-pneumática - continuação

➤ A especificação do sistema automatizado será realizar ciclicamente as operações:

□ [Partida, A+, B+, {C+, A-}, {B-, C-}, Partida,...],

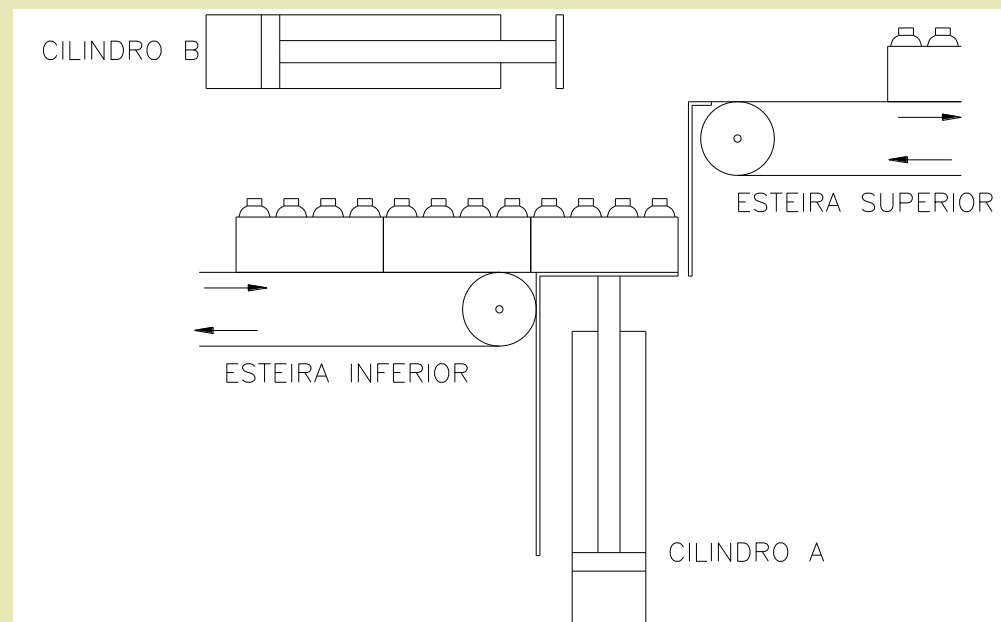
onde {C+, A-} representa que os movimentos C+ e A- devem ocorrer ao mesmo tempo.

➤ Há 3 botões de comando manual: ON liga o sistema; OFF desliga o sistema, após o fim do ciclo que está em curso, E desliga o sistema em emergência, instantaneamente.

Sistemas a Eventos Discretos

Exemplo : Célula de transporte de material

➤ Sistema automático de transferência de containeres: a esteira inferior permanentemente em movimento coloca as caixas na plataforma do cilindro A que ao receber o sinal da presença da caixa expande o embolo colocando a caixa no nível da esteira superior, quando então o cilindro B desloca a caixa para a esteira superior.

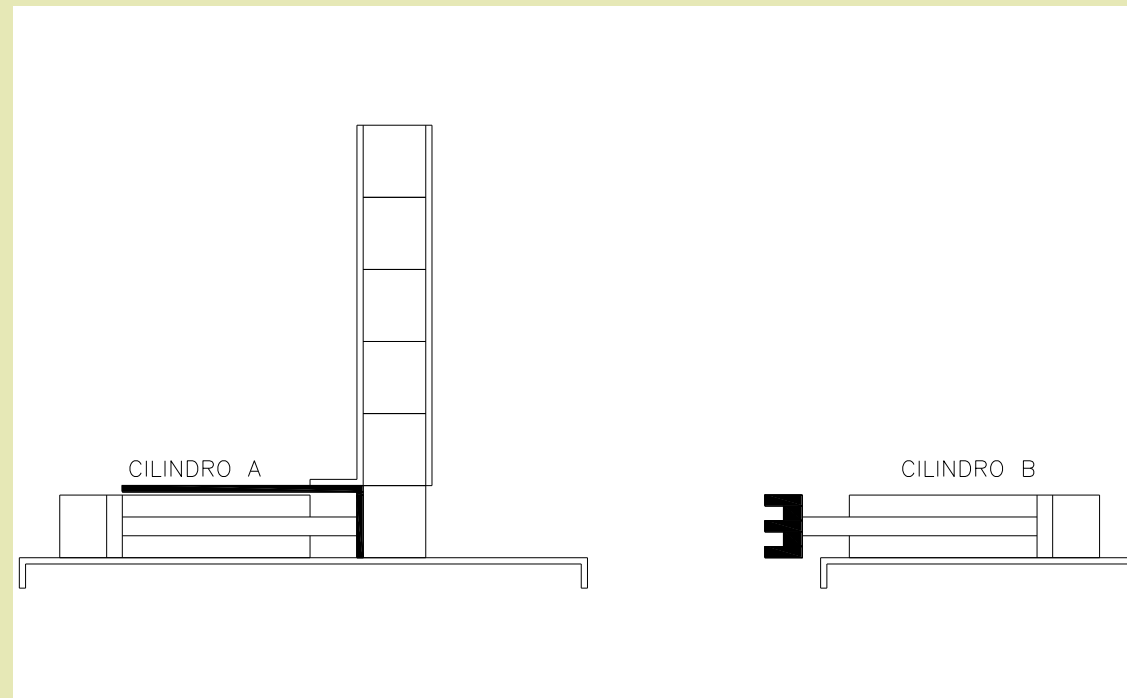


Sequência dos acionamentos: A+;B+;A-;B-

Sistemas a Eventos Discretos

Exemplo : **Célula de manufatura para marcação de peças**

➤ Sistema automático de marcação (carimbo) de peças é realizado pela expansão do cilindro A que empurra a peça da pilha colocando-a em frente ao cilindro B. Este é acionado carimbando a peça e retornando a posição inicial. O cilindro A é recolhido, a pilha desce e ao empurrar a segunda peça esta joga a anterior para baixo ficando pronta para ser carimbada.

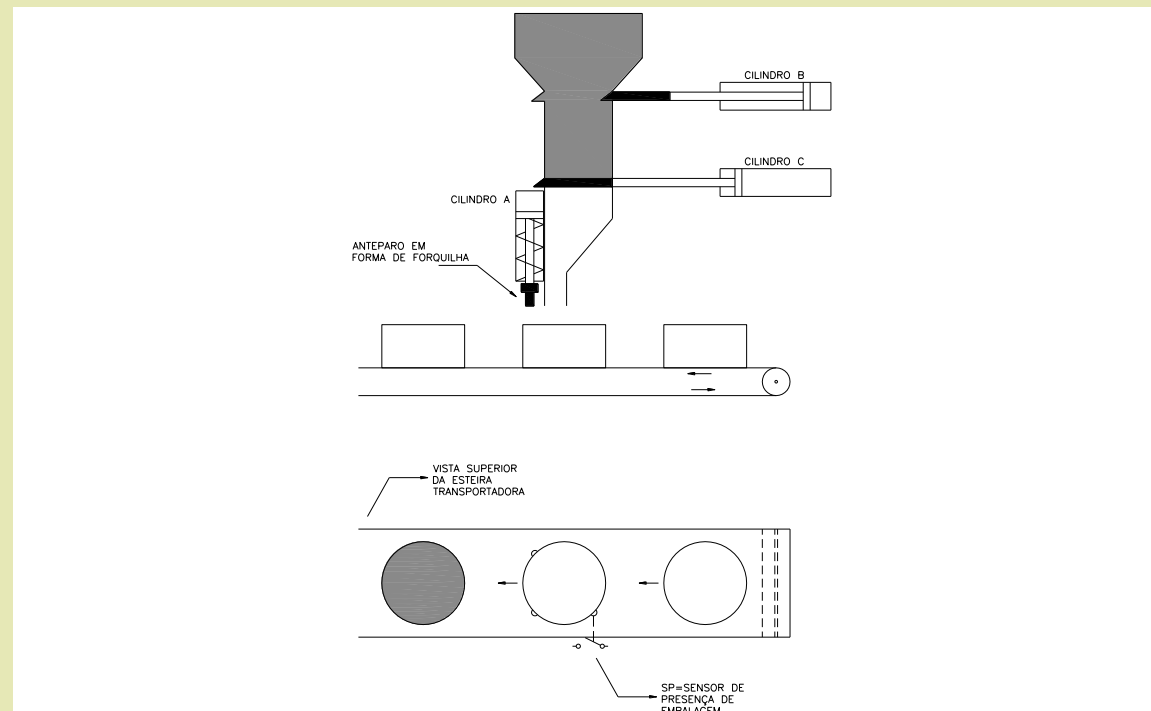


Sequência dos acionamentos: A+;B+;B-;A-

Sistemas a Eventos Discretos

Exemplo : Célula dosadora

➤ Sistema automático de enchimento: quando o sensor de presença detecta o recipiente sob o dosador, o cilindro A expandindo fixa o recipiente com a pinça. O cilindro B expande, o C é recolhido permitindo o enchimento do recipiente, retornando a posição original. Finalmente o cilindro A é recolhido liberando o recipiente na esteira e o B tb. é recolhido permitindo a reposição do material no silo.



Sequência dos acionamentos: A+;B+;C-;C+;A-;B-

SISTEMAS DINÂMICOS A EVENTOS DISCRETOS

Transferência de Matéria Prima em Esteiras Transportadoras:

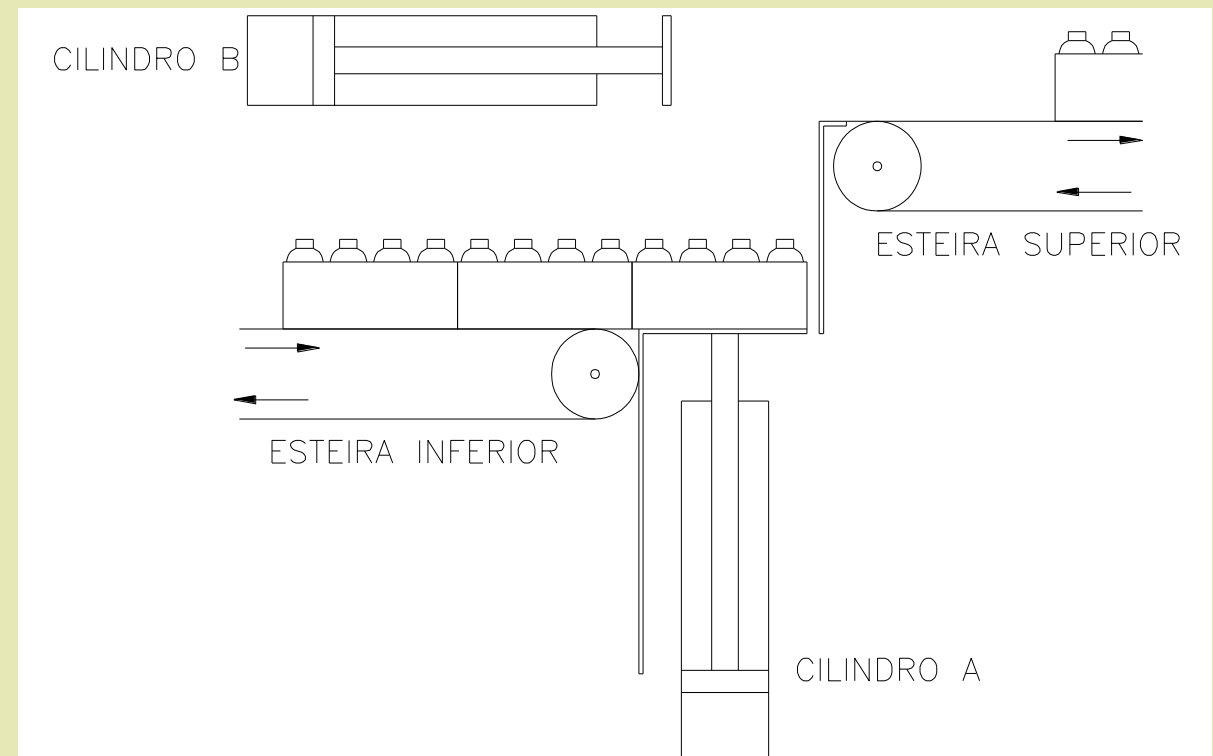
Sequência: $A+B+A-B-$

Sequência de Operações

Diagrama de Trajeto e Passos

Modelagem pela Rede de Petri

Programa Lader



Metodologia para Análise e Síntese dos Sistemas de Automação

Análise por Simulação Digital

Há vários softwares disponíveis para simular pequenas RPs (menos de 100 posições):

Visual object Net++ VON

Este também abrange um tipo de redes híbridas, isto é, com blocos dinâmicos “contínuos” além dos blocos lógicos temporizados; não inclui redes estocásticas.

Endereço: www.systemtechnik.tu-ilmenau.de/~drath

Autor: Dipl-Ing. Rainer Drath, Ilmenau Univ. of Technology, Ilmenau, Alemanha.

SISTEMAS DINÂMICOS A EVENTOS DISCRETOS

Servo Válvula:

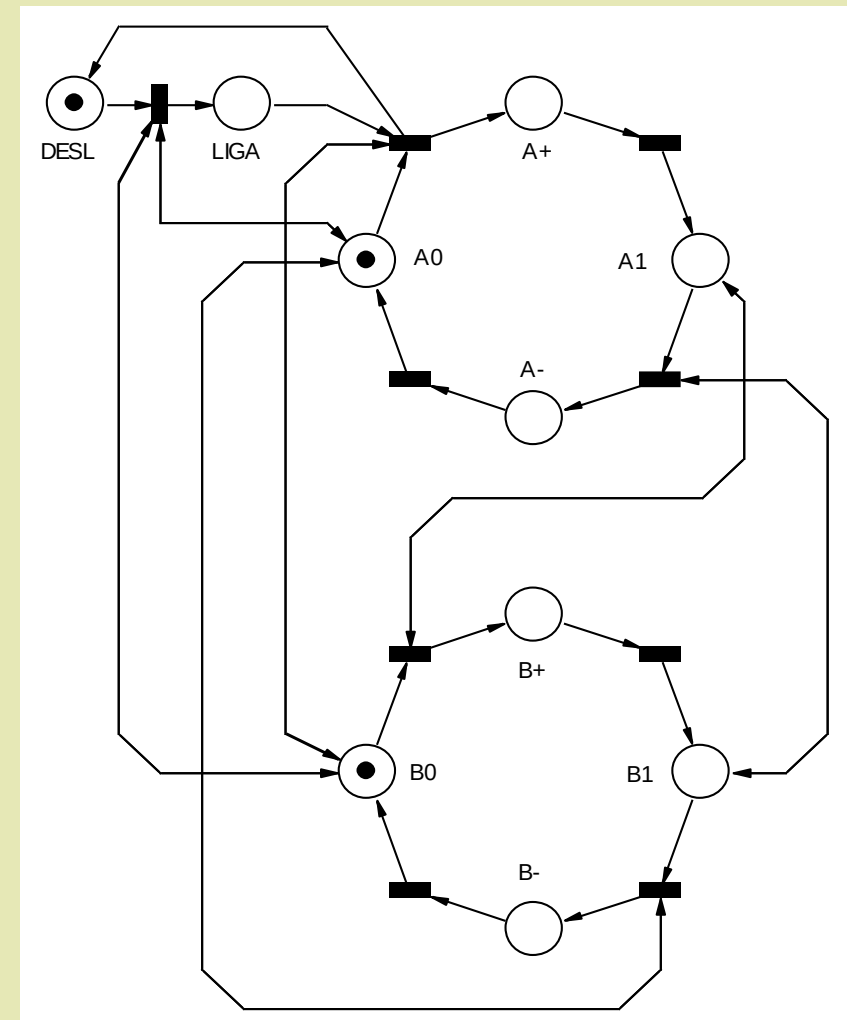
Sequência de Operações

Diagrama de Trajeto e Passos

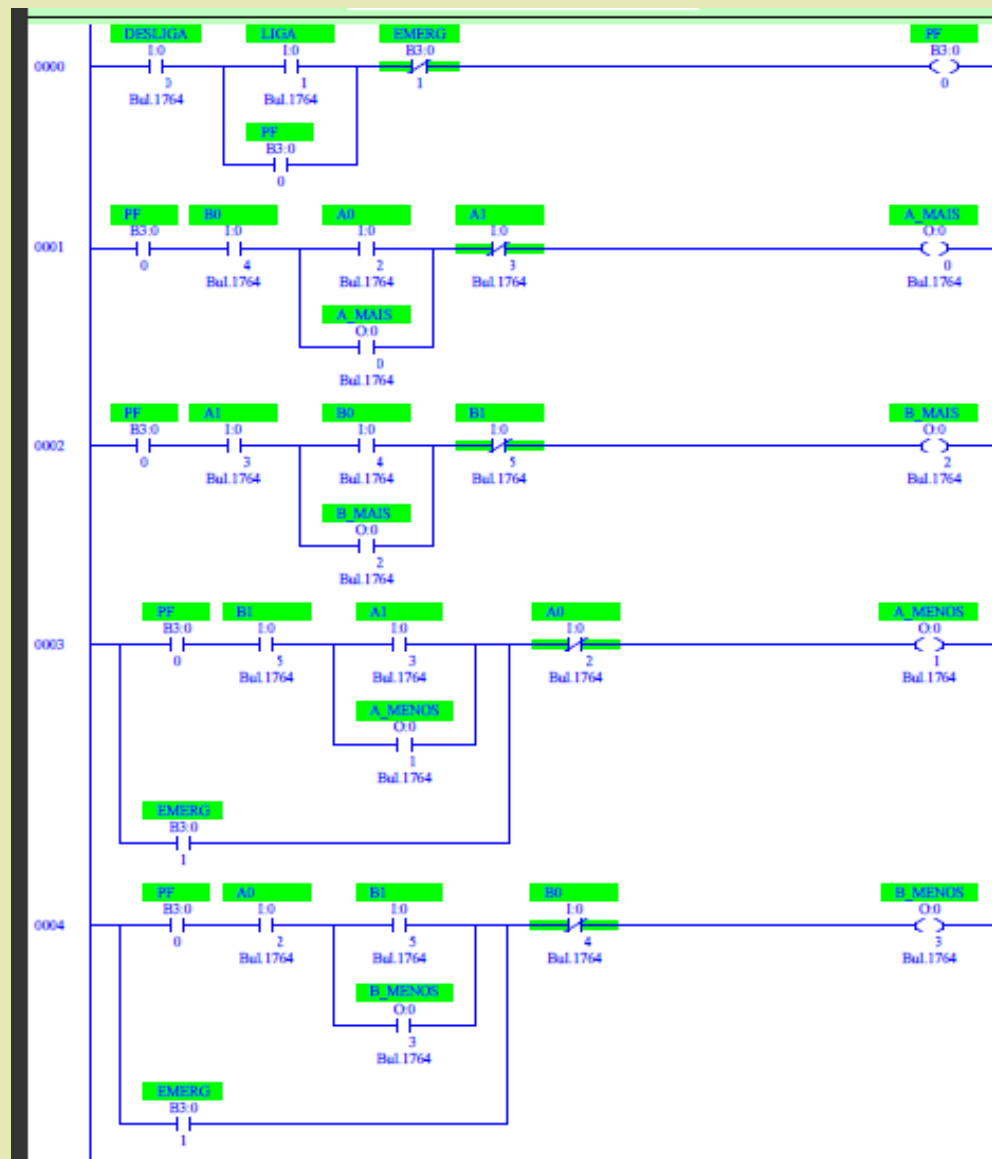
Modelagem pela Rede de Petri

Programa Lader

Sequência: A+B+A-B-



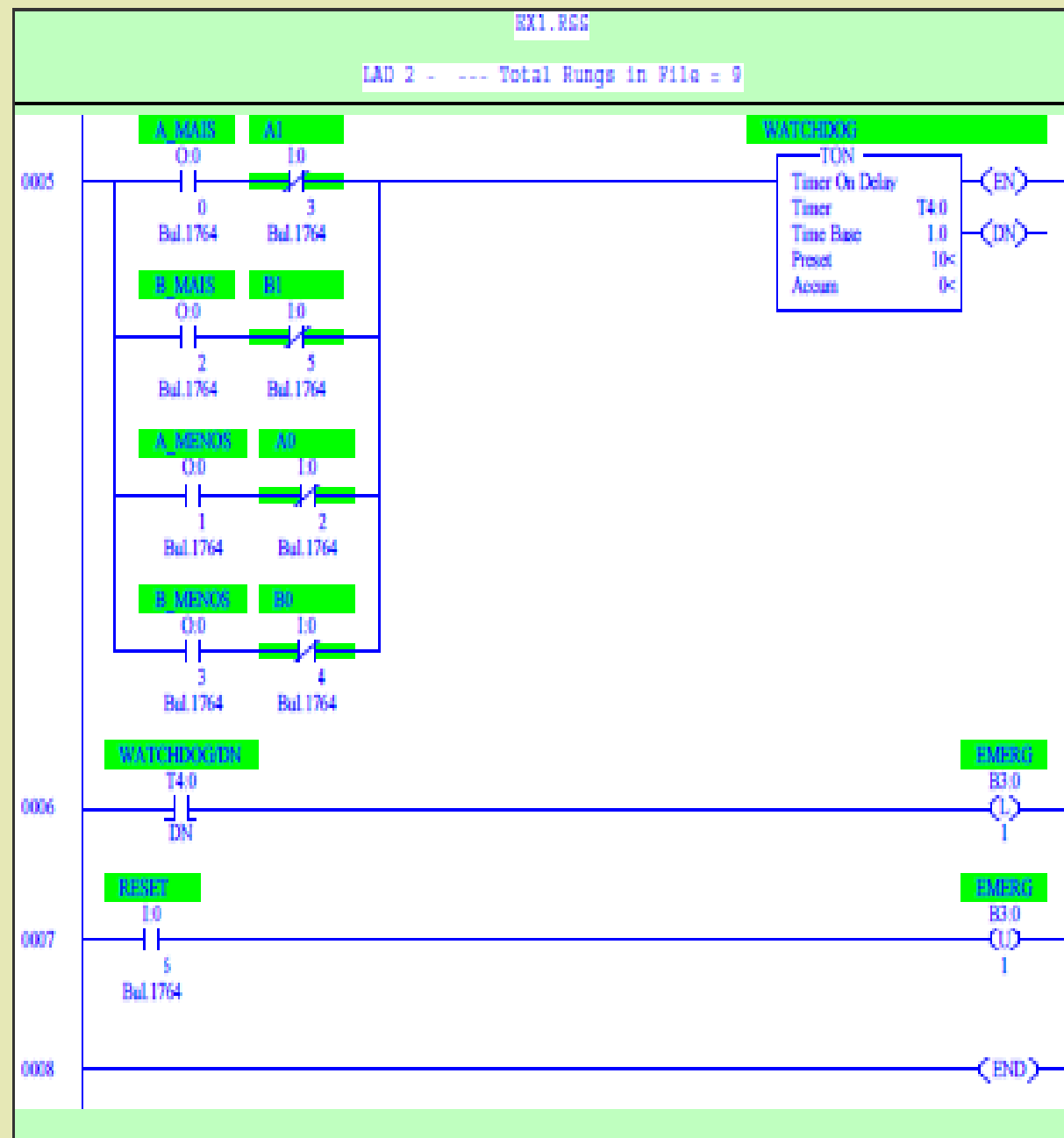
SISTEMAS DINÂMICOS A EVENTOS DISCRETOS





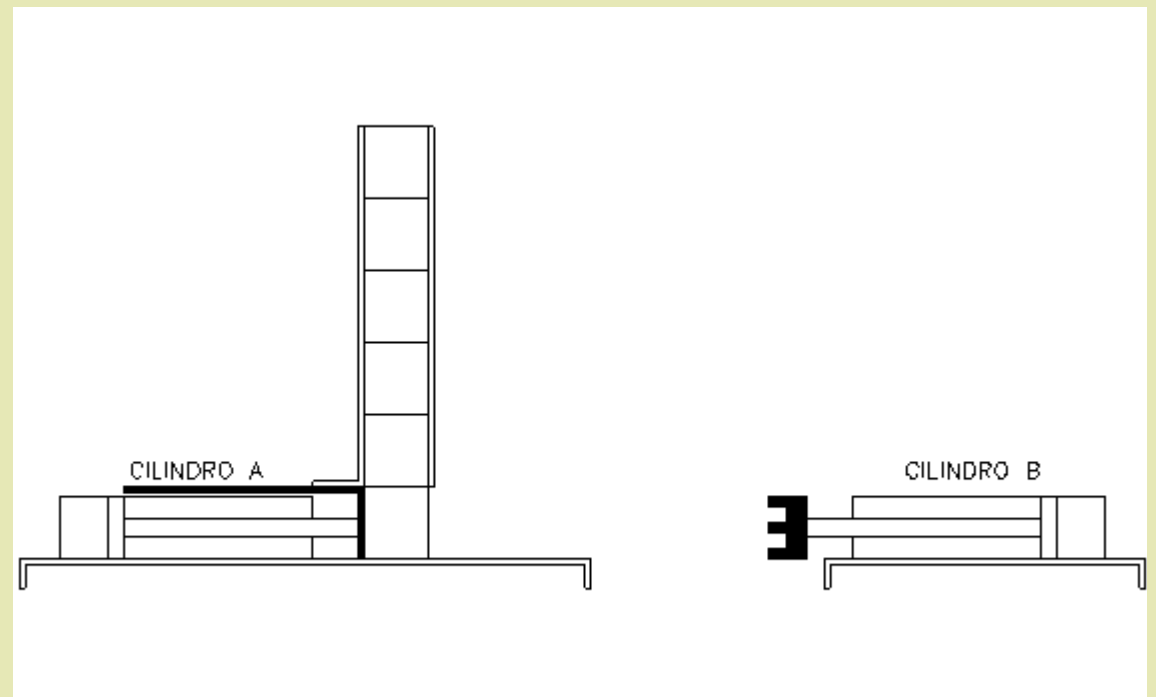
EPUSP

Modelagem de Sistemas a Eventos Discretos: Redes de Petri



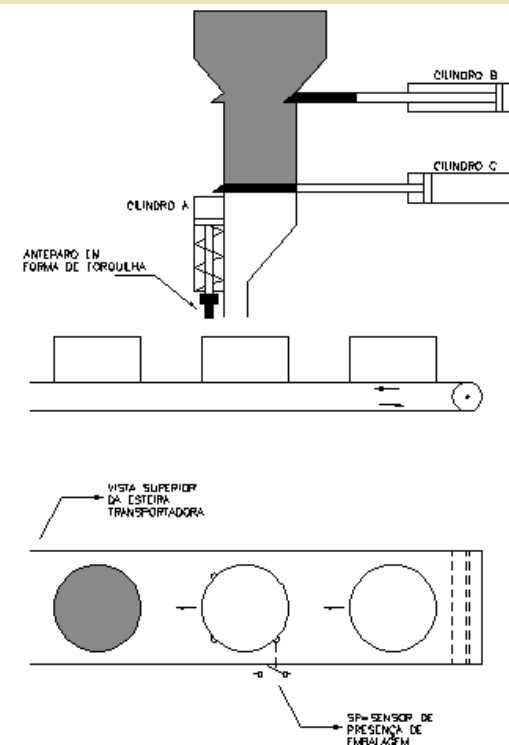
Processo de Rotulação em Produtos de Auto Peças: Sequência: $A+B+B-A-$

Servo Válvula:
Sequência de Operações
Diagrama de Trajeto e Passos
Modelagem pela Rede de Petri
Programa Lader



Enchimento de Material Alimentício em Vasilhames/Caixas: Sequência: $A+B+C-C+B-A-$

Servo Válvula:
Sequência de Operações
Diagrama de Trajeto e Passos
Modelagem pela Rede de Petri
Programa Lader



Redes de Petri

Técnicas de Modelamento de Sistemas

- a) *Agrupamento, “bottom-up”, de baixo para cima* - reúne gradualmente sub-redes representativas de partes do sistema;
- b) *Refinamento, “top-down”, de cima para baixo* - parte das características mais gerais ou “macro” do sistema, e depois as detalha passo a passo.

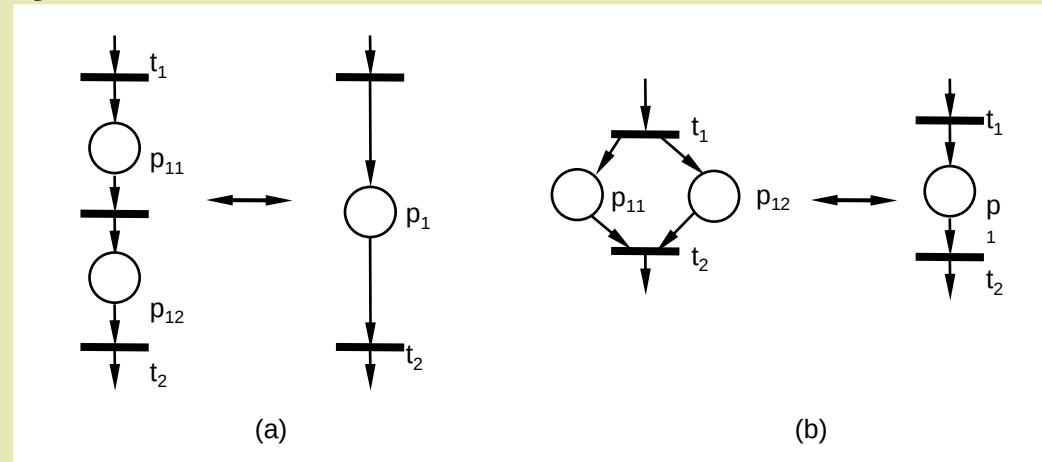
Construção de Modelos por Agrupamento (Bottom-up)

- Primeiramente constroem-se redes para descrever os subsistemas simples, agrupando-os, depois, e formando o modelo final.

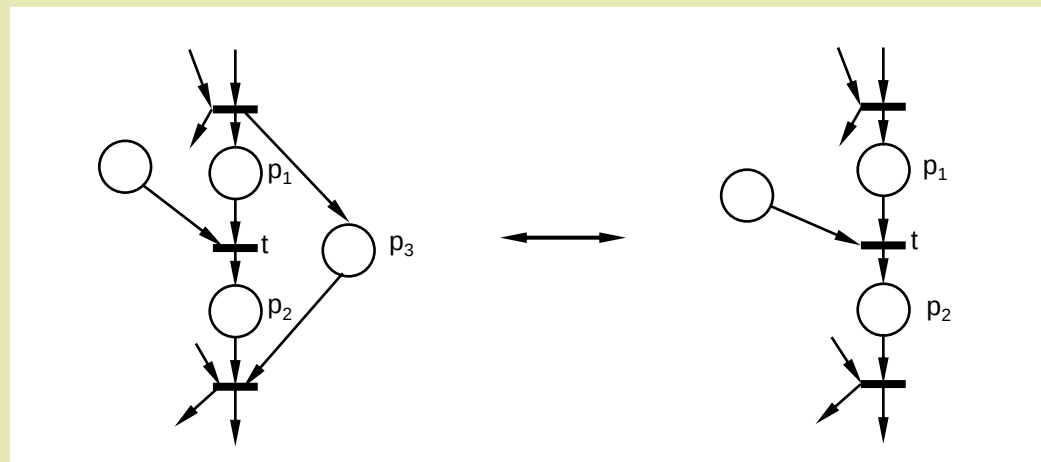
Redes de Petri

Transformações de posições

Simplificação



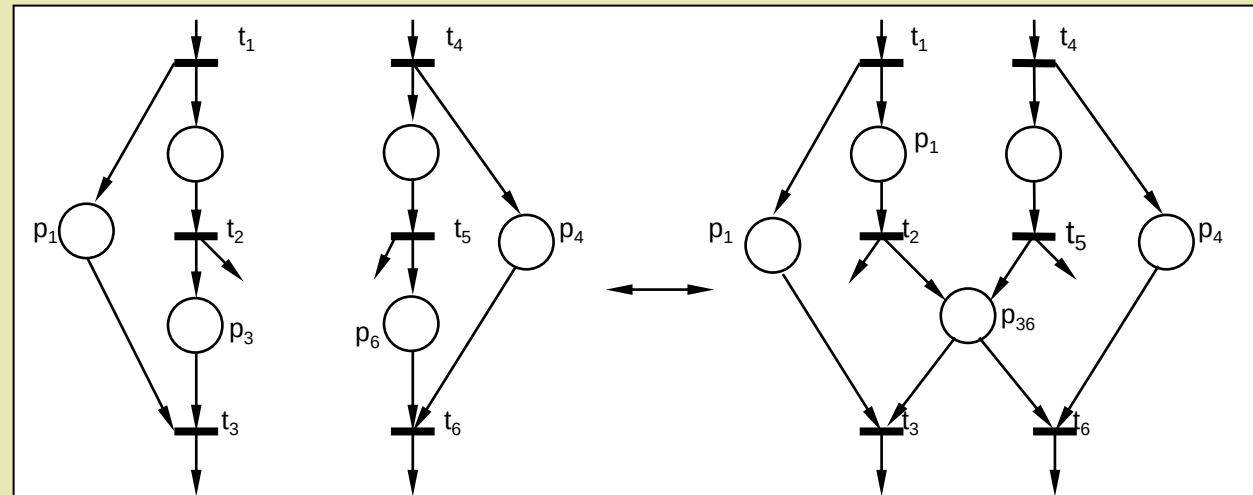
Simplificação de Posições Redundantes



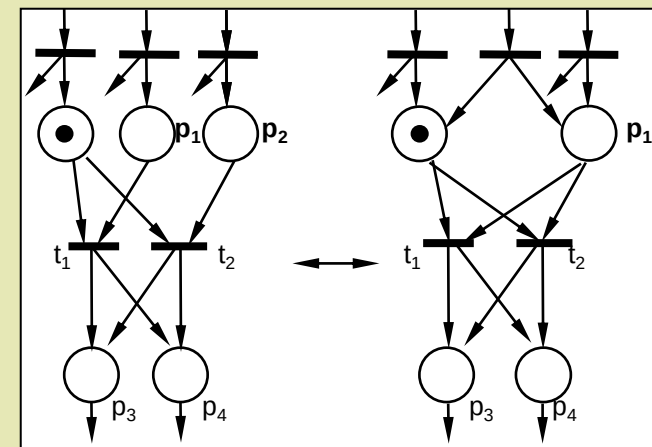
Redes de Petri

Transformações de posições - *continuação*

Fusão de Posições Duplicadas



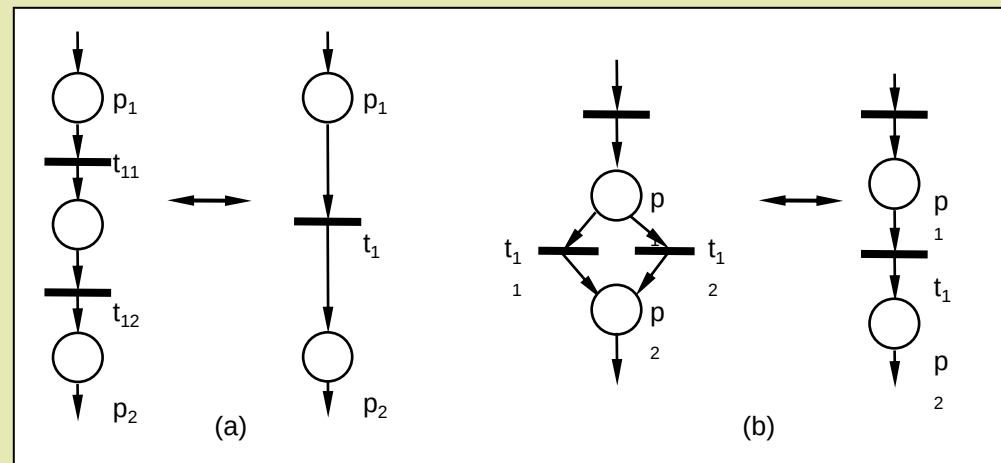
Fusão de posições equivalentes



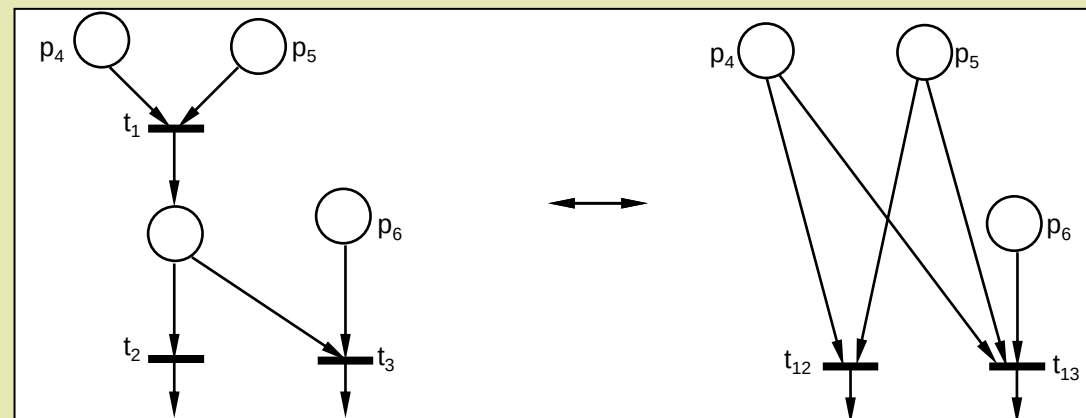
Redes de Petri

Transformações de posições - *continuação*

Transformações de transições



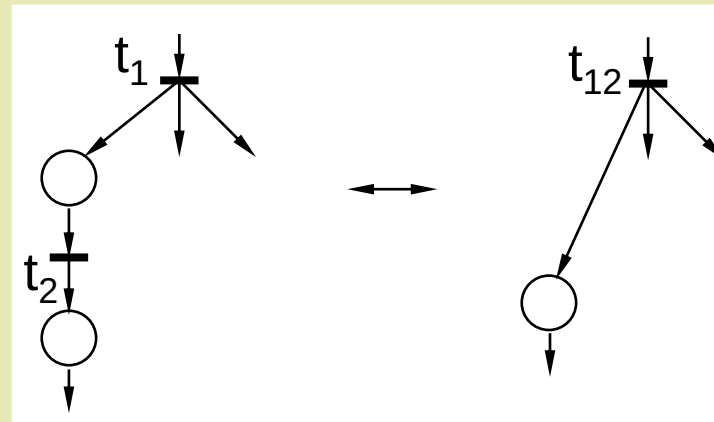
Pré-fusão



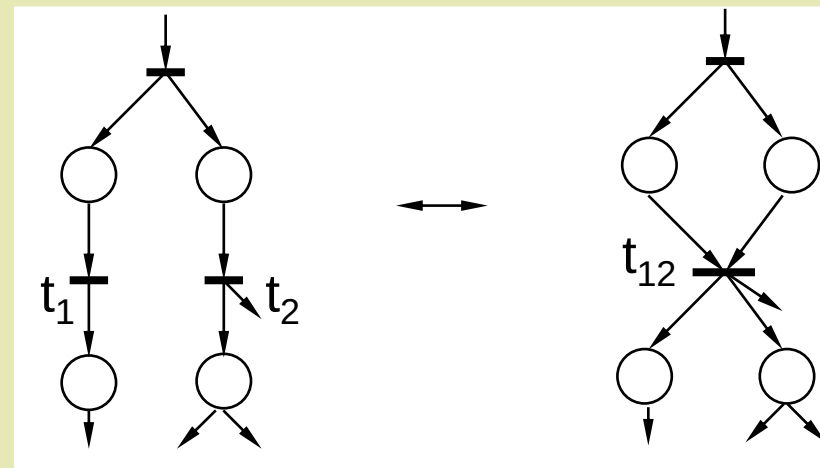
Redes de Petri

Transformações de posições - *continuação*

Pós-fusão



Fusão lateral



Redes de Petri

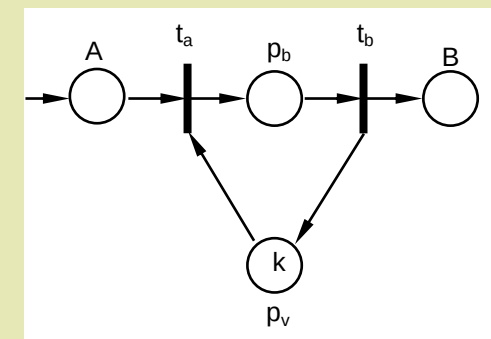
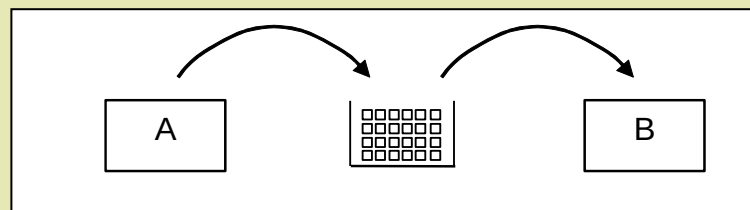
Construção de Modelos por Agrupamento (Bottom-up)

Primeiramente constroem-se redes para descrever os subsistemas simples, agrupando-os, depois, e formando o modelo final.

Modelos para processos de manufatura

a) Estoque Intermediário Limitado (Buffer)

Cada vez que a Máquina A tem uma peça pronta/disponível, a RP impõe que a transição t_a só possa disparar se houver alguma marca em p_v .

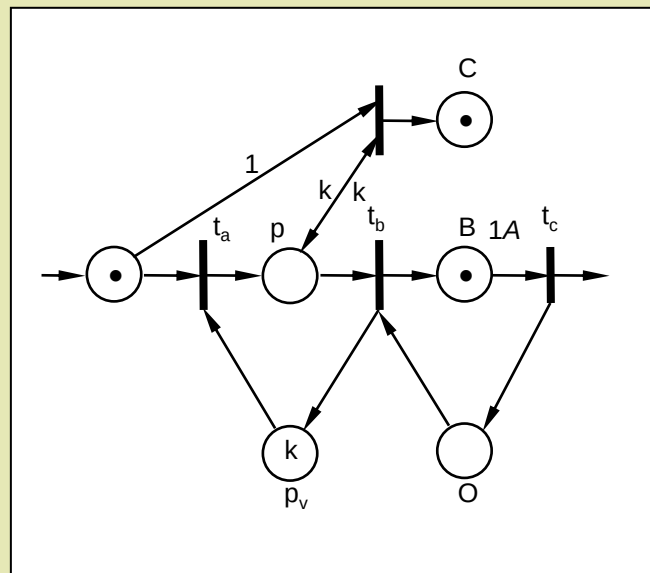


Redes de Petri

Construção de Modelos por Agrupamento (Bottom-up)

b) Overflow

Na figura, quando a produção de B se atrasa e o buffer atinge k peças, $p_b = k$ e $O = 0$; se nova peça surge na saída de A, os arcos de peso 1 e de peso k habilitam a transição que alimenta C, e a nova peça é enviada a C.

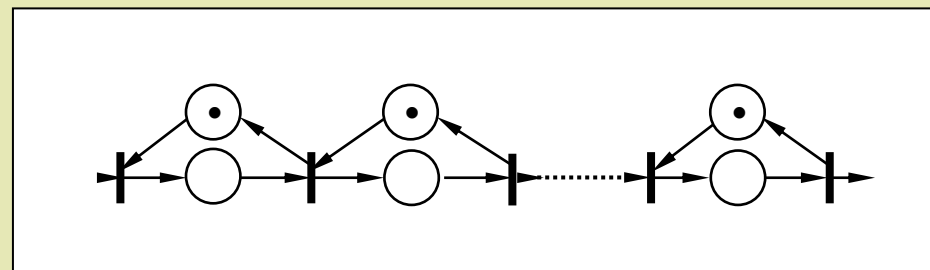
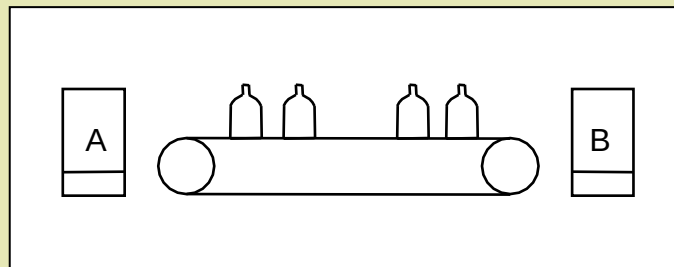


Redes de Petri

Construção de Modelos por Agrupamento (Bottom-up)

c) Buffer First-in First-out

A esteira da figura pode ser considerada como um *buffer* FIFO.

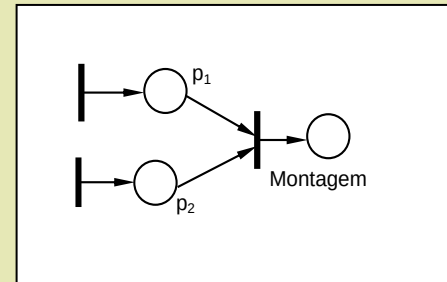


Redes de Petri

Construção de Modelos por Agrupamento (Bottom-up)

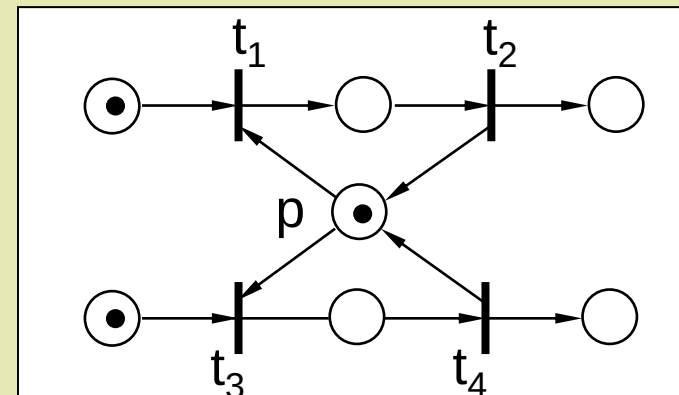
d) Concorrência ou paralelismo

A montagem, na figura, só pode ocorrer quando existem peças de ambos os tipos produzidas (marcas nos estoques p_1 e p_2).



e) Recursos compartilhados

Tem-se um conflito tipo confusão: t_1 e t_3 estão habilitadas, mas o disparo de uma inabilita a outra.

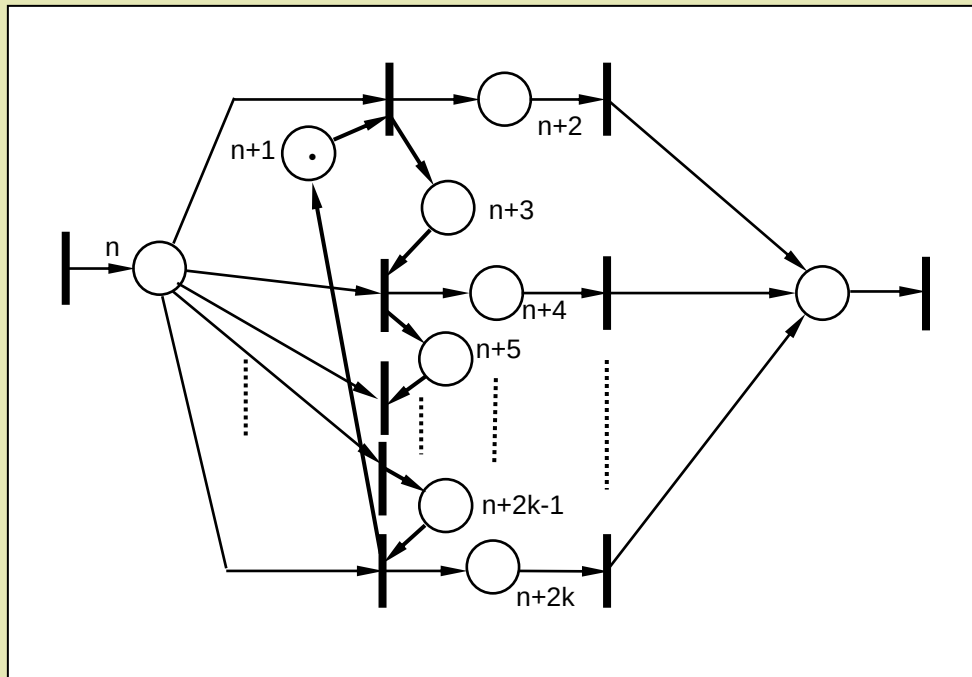


Redes de Petri

Construção de Modelos por Agrupamento (Bottom-up)

f) Rede de escolha automática de servidor ou de sincronização[ZC]

Na RP da figura, as posições ($n + \text{número par}$) representam k máquinas. As posições do tipo ($n + \text{número ímpar}$) representam um sistema de controle que decide qual máquina recebe a peça nova ou chegada à posição n :

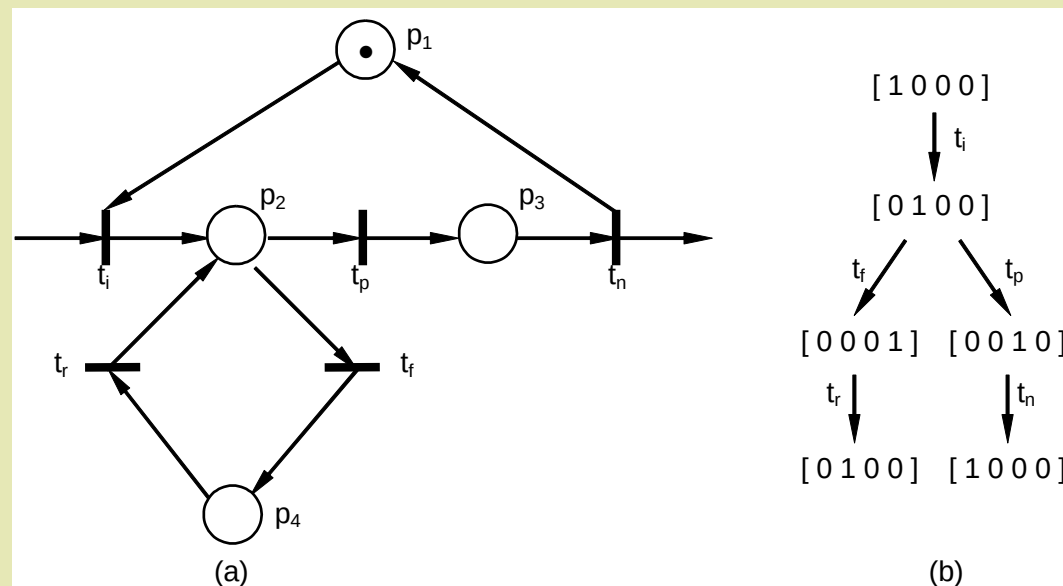


Redes de Petri

Construção de Modelos por Agrupamento (Bottom-up)

g) Uma máquina com falhas

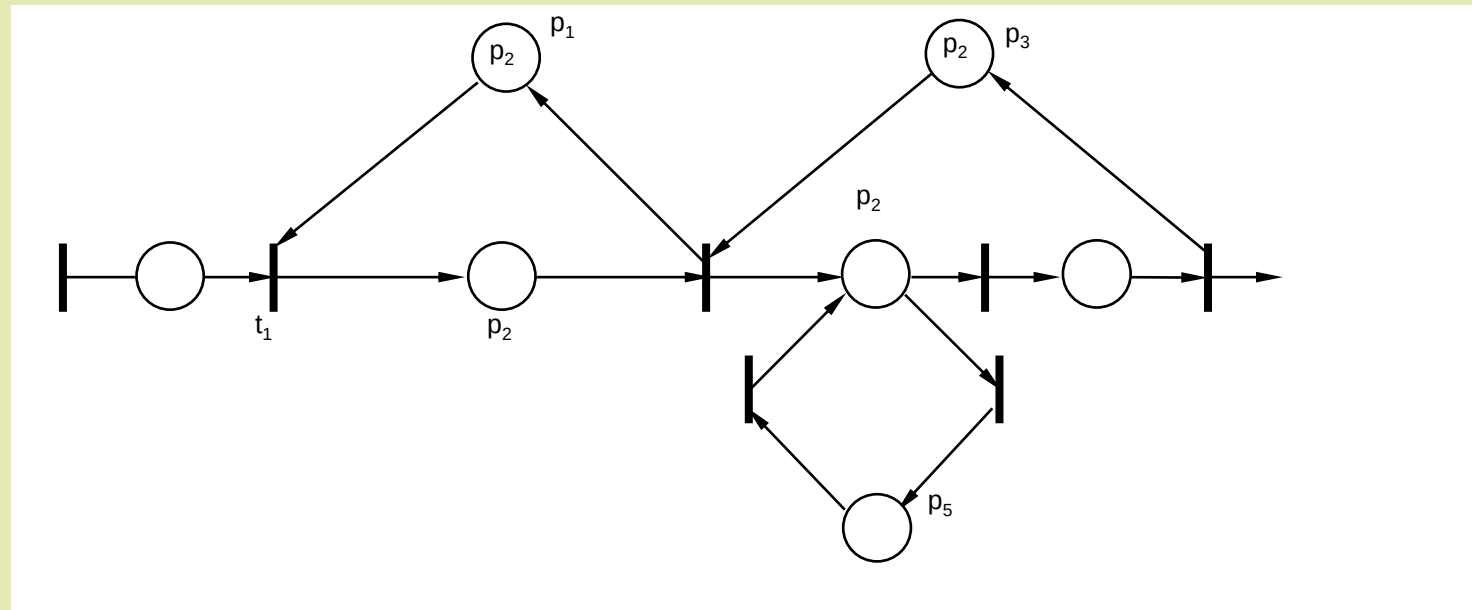
- A RP abaixo mostra um conflito do tipo confusão, em p_2 ; para suprimi-lo seria necessário modelar o processo interno a p_2 o qual decide habilitar t_f ou t_r .
- As peças a processar são marcas liberadas pela transição fonte.



Redes de Petri

Construção de Modelos por Agrupamento (Bottom-up)

h) Duas máquinas, a segunda com falhas, nenhum estoque intermediário



Redes de Petri

Construção de Modelos por Agrupamento (Bottom-up)

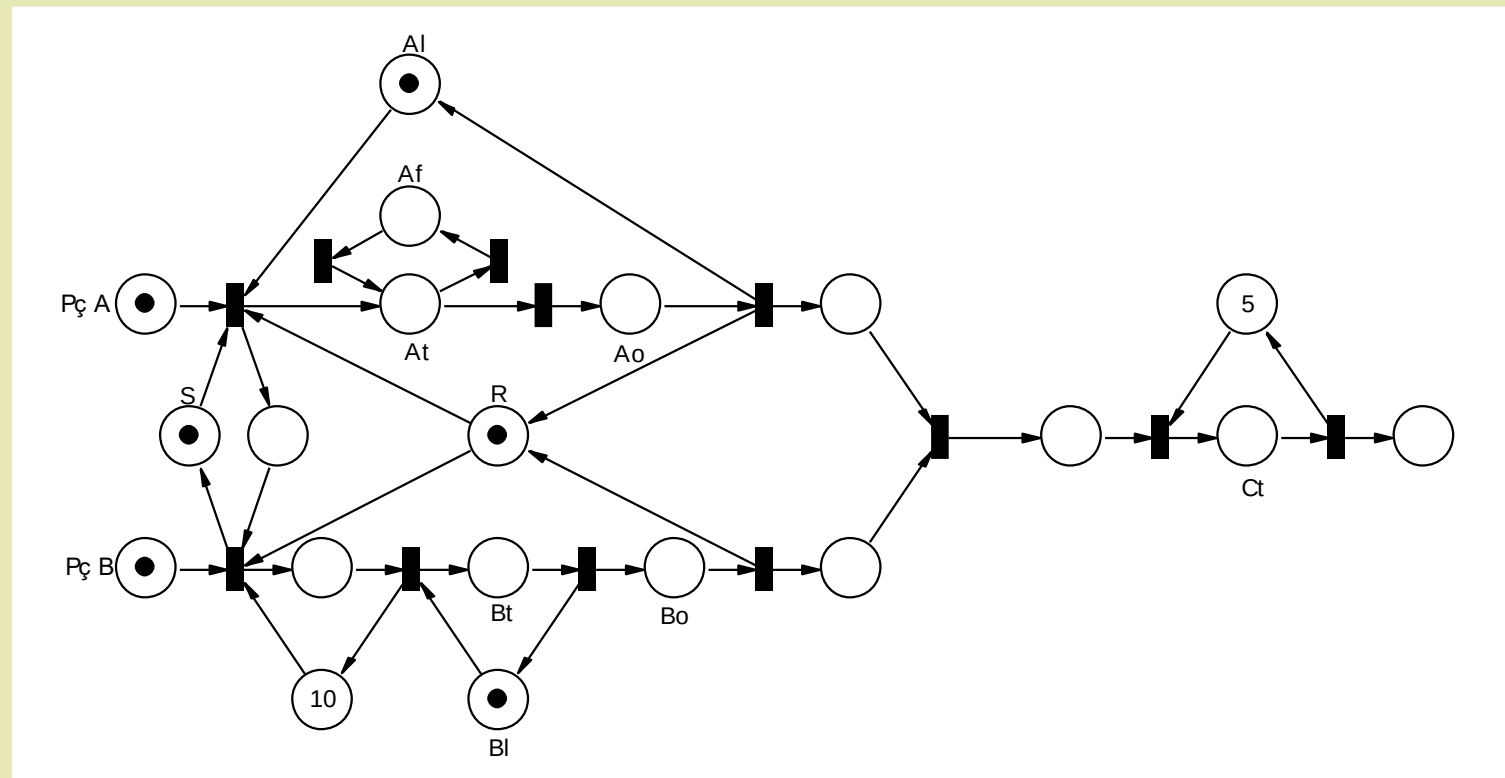
Exemplo de modelamento *bottom-up*

- Duas máquinas A e B alimentam a máquina montadora C, com suas peças. As duas máquinas A e B são preparadas por um robô que coloca e retira as peças em cada uma.
- Máquina A: Posições: Parada, recebendo peça (Ap_1) / Parada, entregando peça (Ap_2) / Defeituosa (Af) / Trabalhando (At).
- Máquina B: Posições: Parada, recebendo peça (Bp_1) / Parada entregando peça (Bp_2) / Trabalhando (Bt) / Buffer na entrada (m)
- Máquina C: Posição Montagem, com buffer (k)
- Robô: Posições: Robô disponível (R); Sincronizador do robot (S).

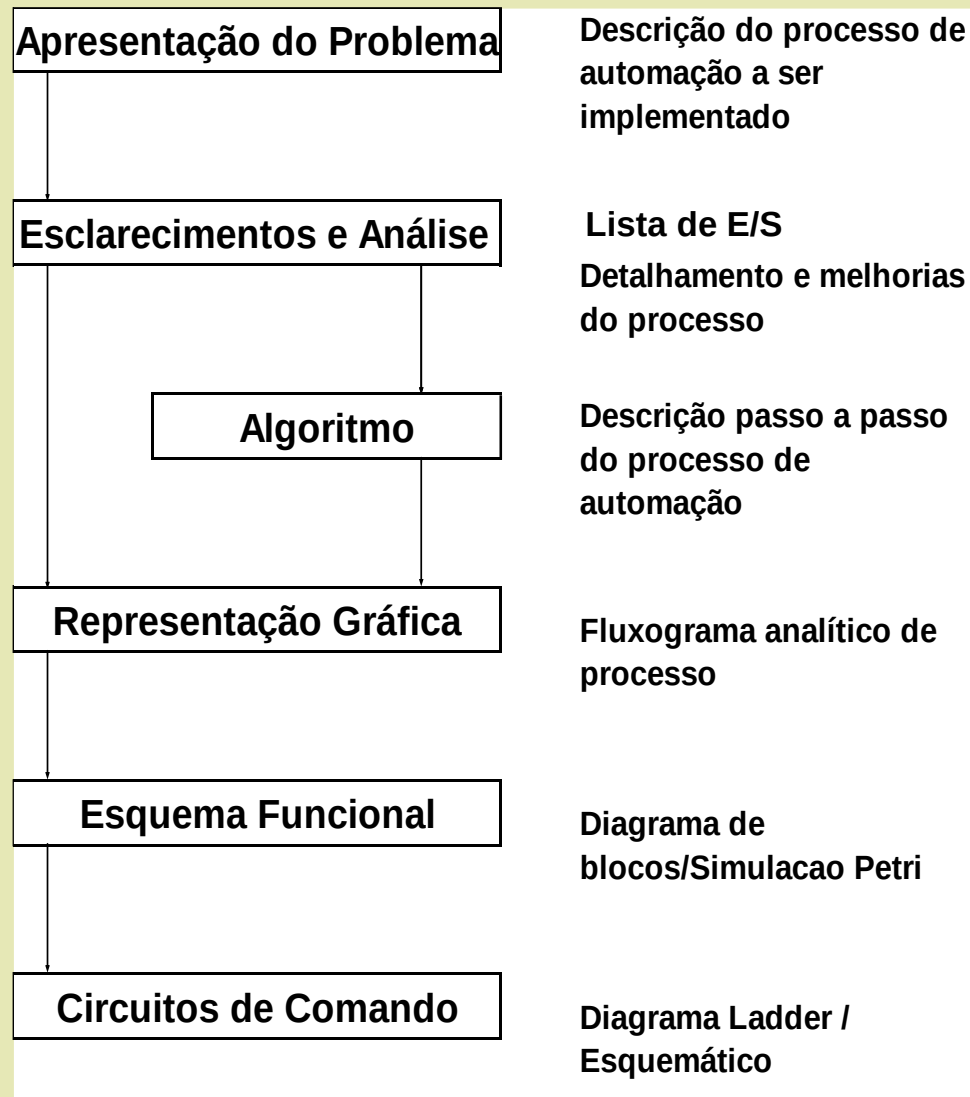
Redes de Petri

Construção de Modelos por Agrupamento (Bottom-up)

Exemplo de modelamento *bottom-up*



Metodologia de análise e projeto para programação de CLP's



Metodologia de análise e projeto para programação de CLP's

Descrição do problema

Vt – velocidade constante do tubo. Quando o tubo atinge Pr (ponto de referência), o carrinho sai de P0 e atinge P1 com velocidade $vc=vt$. A morsa prende o tubo e a serra circular executa o corte, em seguida o acionamento é invertido através de frenagem.

O carro atinge uma posição P2 quando o acionamento é desligado e volta a P0 pela inércia

Análise

Variáveis de entrada

- C1: tubo na posição para partir o carro
- C2: posição inicial do carrinho P0
- C3: posição P1 que informa fechamento da morsa
- C4: fim do curso inferior da serra
- C5: fim do curso superior da serra
- C6: fechamento da morsa
- C7: desligamento do acionamento do carro em P2

Variáveis de saída

- CF: carro a frente
- CR: carro a re
- M: fecha morsa
- SD: Serra desce
- SS: Serra sobe

Metodologia de análise e projeto para programação de CLP's

Algoritmo: Etapas

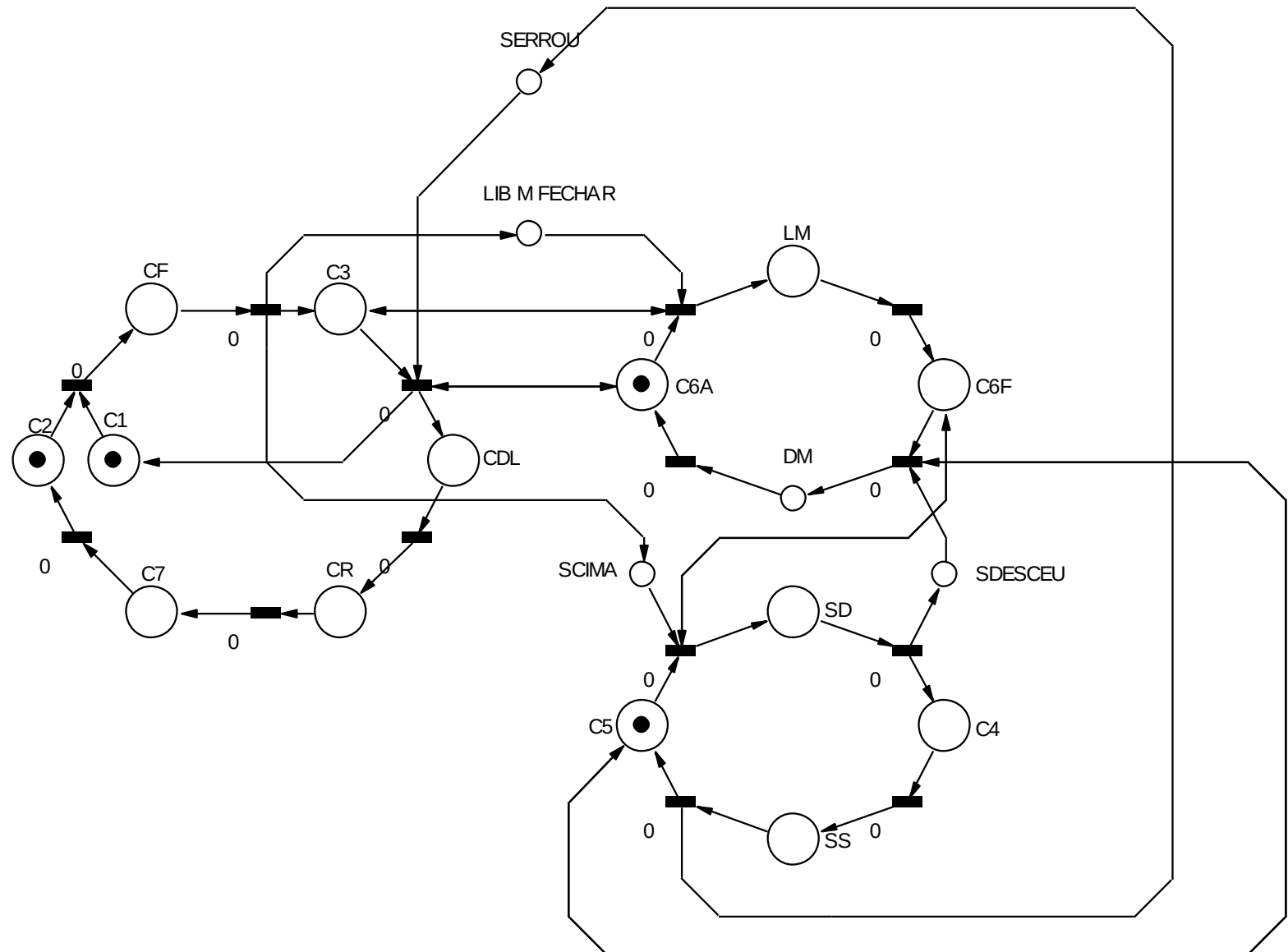
- 1) Posição do carro em C2
- 2) Posição do tubo em C1 – Acionar o carro, ligar o acionamento
- 3) Posição do carro em C3 – Acionar a morsa
- 4) Acionar o fechamento da morsa – C6
- 5) Descida da serra até o acionamento da chave C4
- 6) Subida da serra até o acionamento da chave C5
- 7) Frenagem do acionamento e inversão da velocidade
- 8) Desligamento do acionamento em P2 – C7



EPUSP

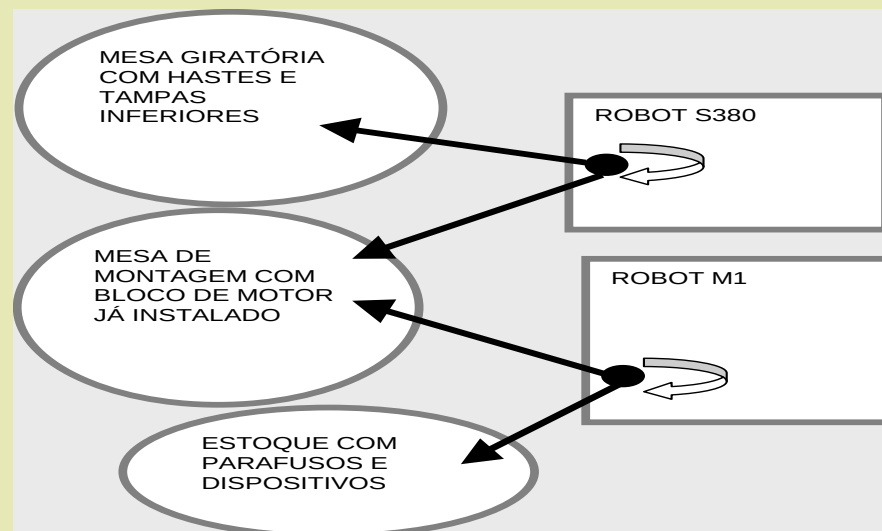
Modelagem de Sistemas a Eventos Discretos: Redes de Petri

Metodologia de análise e projeto para programação de CLP's



Refinamento Célula de Montagem de Pistões em Blocos de Motor

A representação por RP começa descrevendo o sistema da forma mais simplificada possível:



- p4 = S380 posiciona e encaixa o virabrequim para alinhamento;
- p5 = Os dois robôs trabalham simultaneamente, com duas atividades temporárias na entrada e saída;
- p6 = M1 trabalha sozinho em três etapas consecutivas.

Célula de Montagem de Pistões em Blocos de Motor

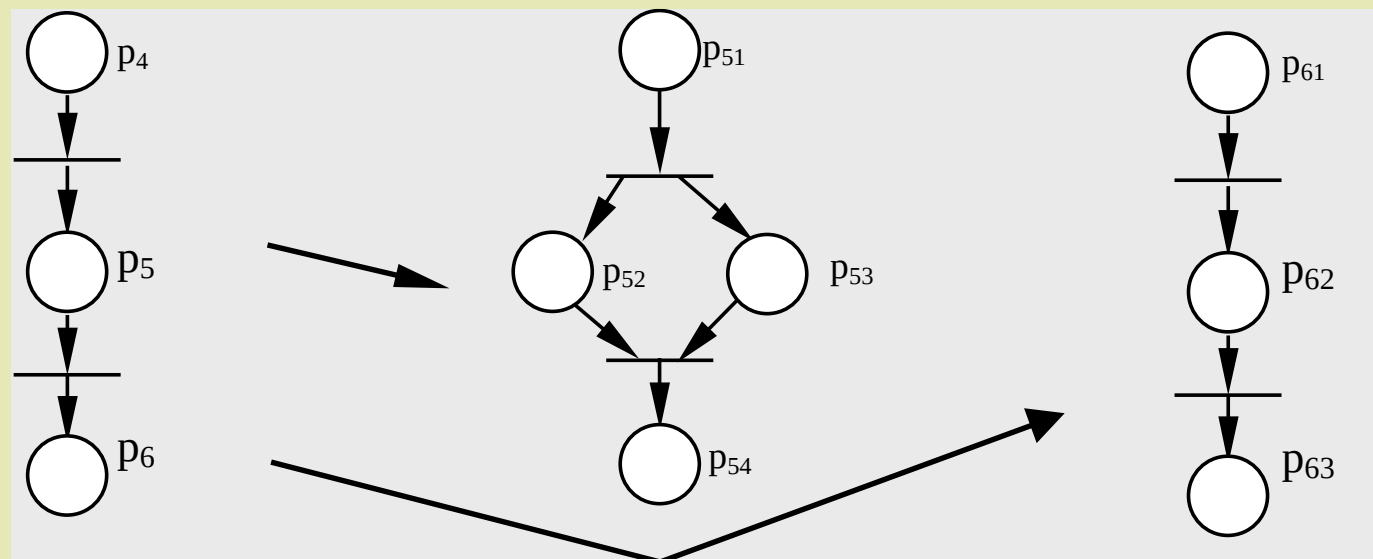
A seguir substitui-se p_3 por um bloco sequencial de 3 posições:

p_4 = S380 posiciona e encaixa o virabrequim para alinhamento;

p_5 = os dois robôs trabalham simultaneamente, com duas atividades temporárias na entrada e saída;

p_6 = M1 trabalha sozinho em três etapas consecutivas.

Resulta:



Processos de Modelamento por Redes de Petri

Construção de Modelos por Refinamento (Top-down)

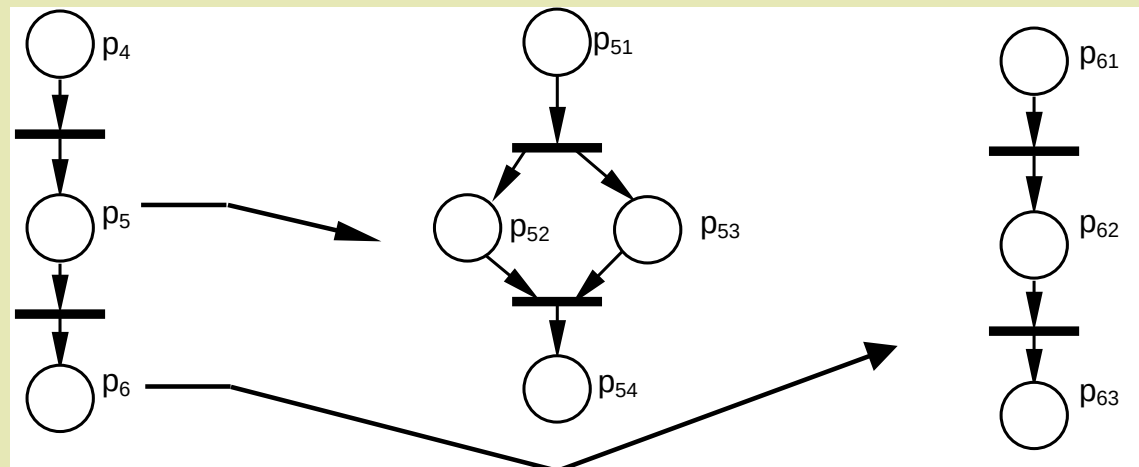
➤ Exemplo: **Célula de Montagem de Pistões** - *continuação*

- A seguir substitui-se p_6 por 3 posições sequenciais:

p_{61} = S380 empurra o pistão no bloco e devolve o dispositivo;

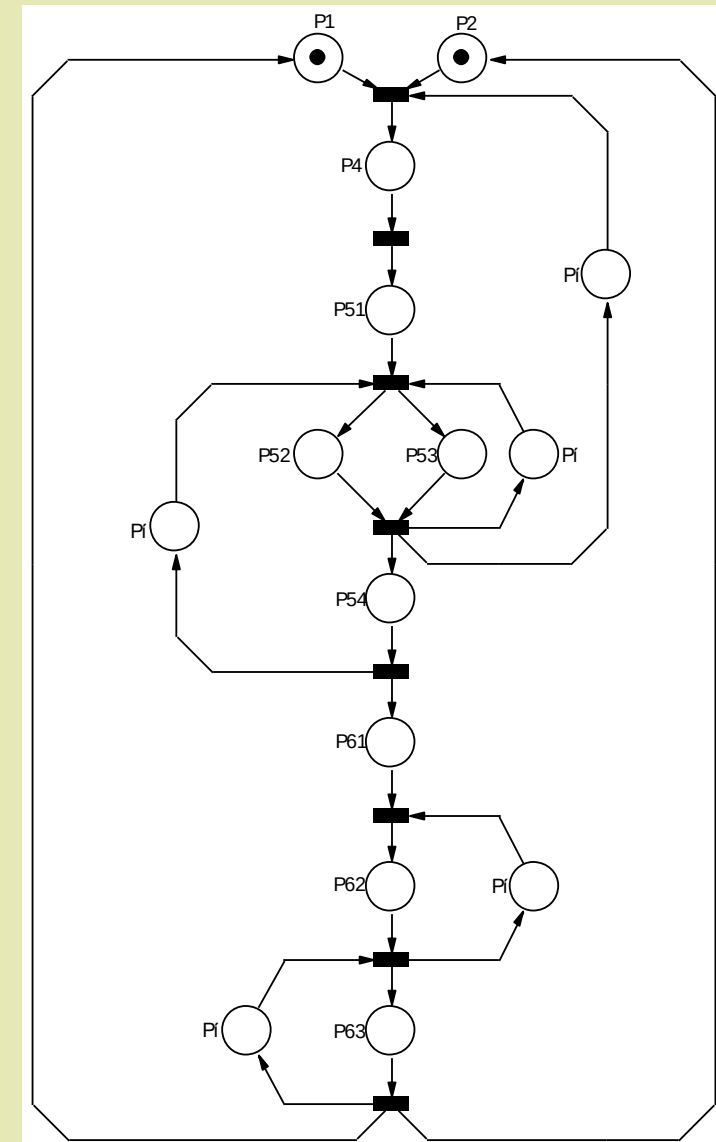
p_{62} = S380 pega uma tampa e coloca no bloco;

p_{63} = M1 pega 2 parafusos e instala no bloco. Figura abaixo, à direita:



Célula de Montagem de Pistões em Blocos de Motor

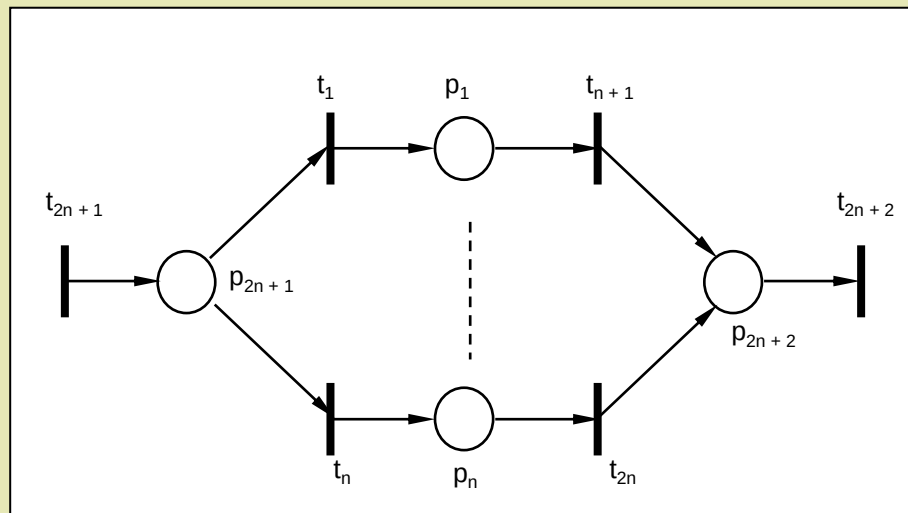
- R1 opera P4 e P5 Completas
- R2 opera P5 e P6 Completas
- Buffer de uma peça para :
P52,P62 e P63



Refinamento de transições

Algumas sub-redes que, ao substituírem uma transição, introduzem detalhes e mantêm as qualidades V/S/R da rede:

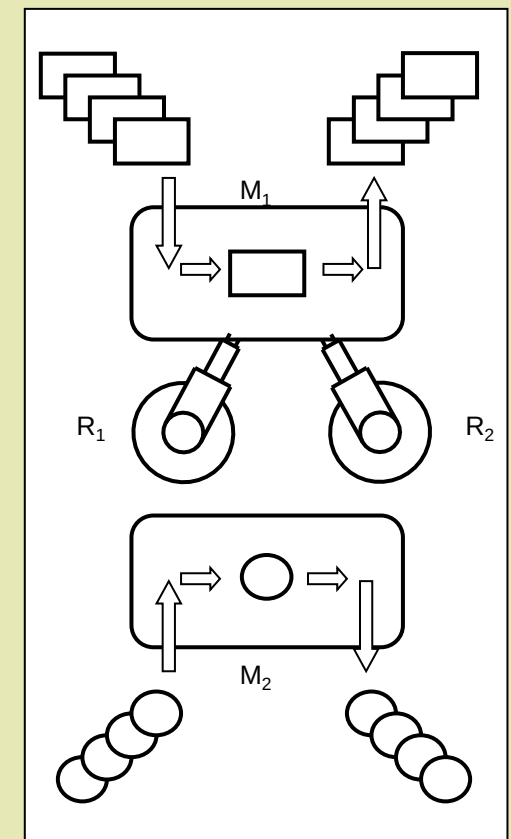
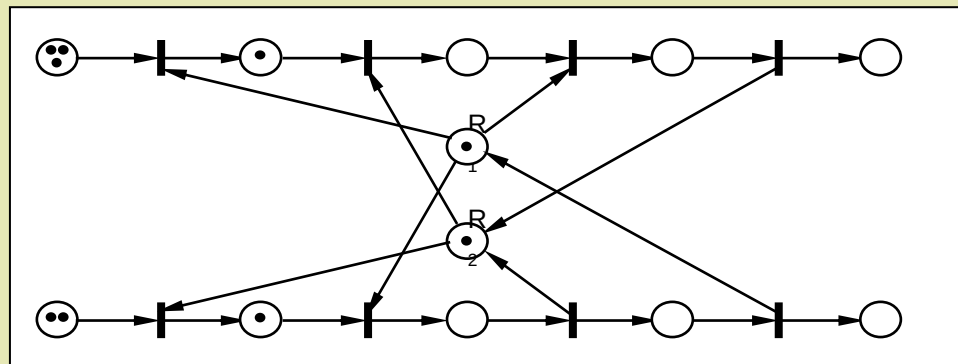
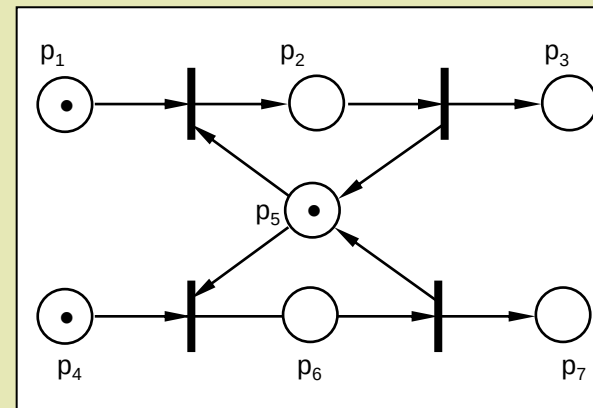
- a) módulo sequencial ($n + 2$ posições e $2(n+1)$ transições);
- b) módulo paralelo ($n + 2$ posições e $2(n+1)$ transições).



A RP resultante é limitada, segura ou viva se e somente se a RP original é, respectivamente, limitada, segura ou viva.

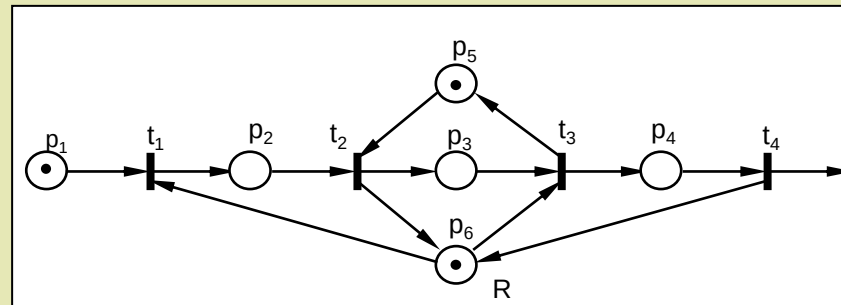
Síntese

Mútua Exclusão Paralela (MEP): é uma sub-rede como abaixo:

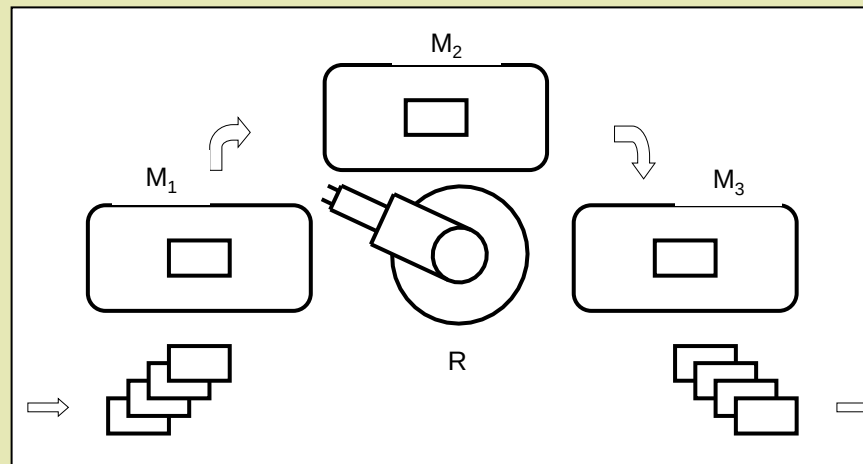


Síntese

Mútua Exclusão Sequencial (MES) é uma sub-rede como abaixo:

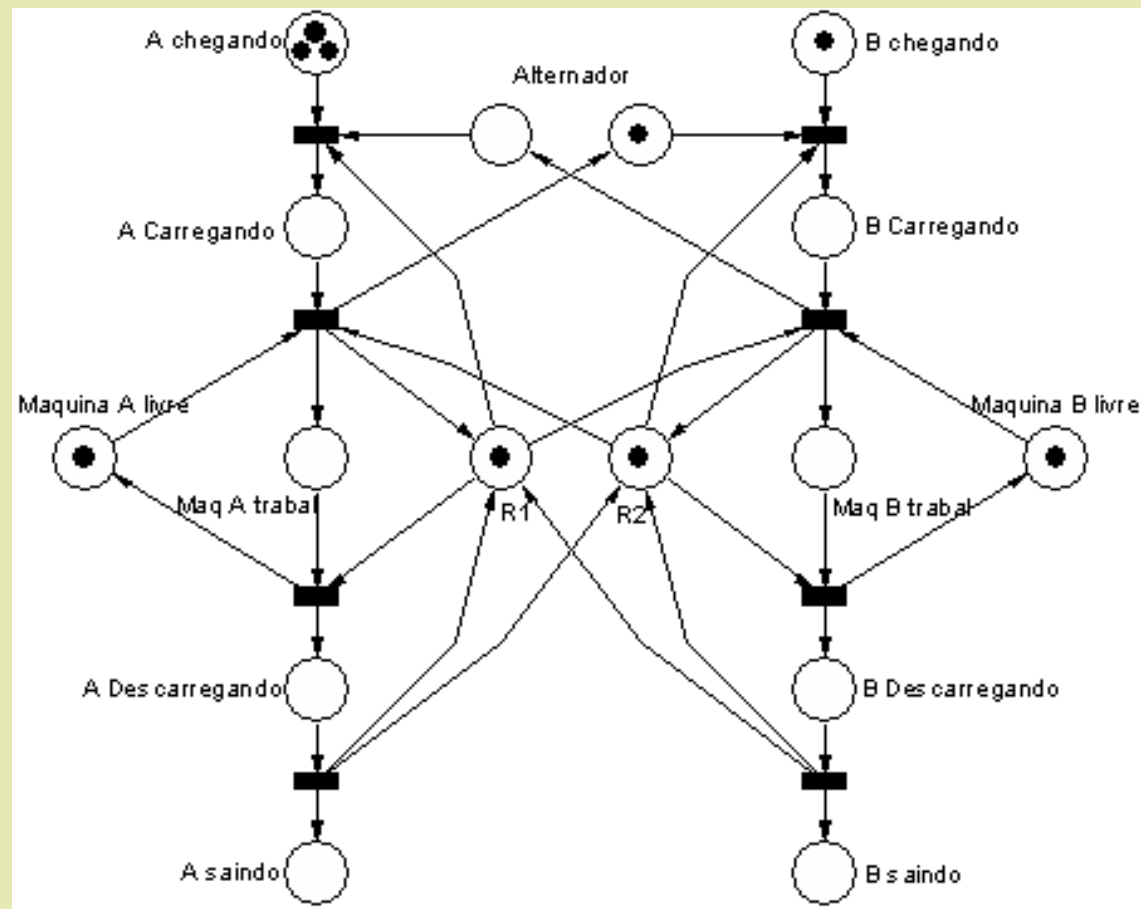


Exemplo: - um processo de produção completo envolve M_1 , M_2 e M_3 em sequência; um único robô R alimenta de partes os processos M_1 e M_3 ; M_2 carrega-se e descarrega-se por si.



Síntese Híbrida

Combinação envolvendo mútuas exclusões série e paralela também acontece frequentemente nas modelagens de sistemas de manufaturas: por exemplo



Síntese Híbrida

Chama-se **capacidade de marcação** $C(t, t')$, entre duas transições t e t' de uma rede, ao maior número de disparos de t que são possíveis a partir da marcação inicial da rede, sem que ocorra qualquer disparo de t' .

Propriedade:

→ numa MES, quando as marcações iniciais das posições de recursos variáveis são menores ou iguais à capacidade de marcação entre as subredes sequenciadas, o partilhamento do recurso fixo mantém as propriedades (V, S ou R);

→ em consequência, a introdução dessa MES em qualquer outra rede V/S/R não lhe altera as propriedades V/S/R.

Algoritmo híbrido

Diz-se ***híbrido*** o procedimento de modelamento descrito a seguir, porque recomenda duas fases: a inicial por refinamento e a final, que aborda fontes de recursos variáveis, por agrupamento.

Passo 1: Formule um modelo em nível macro que capture a partir do momento inicial as interações do tipo recursos compartilhados e redes de escolha/sincronização, e que seja vivo, limitado e reversível.

Passo 2: Refine (*top-down*) as posições e as transições.

Passo 3: Adicione (*bottom-up*) posições de recursos. Primeiro as de recursos dedicados, depois as de compartilhados entre processos paralelos.