

# Supervisão de Sistemas de Produção Baseada em Redes de Petri

Tomaz de Carvalho Barros

Tese de Doutorado submetida à Coordenação dos Cursos de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Paraíba - Campus II como parte dos requisitos necessários para obtenção do grau de Doutor em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Processamento da Informação

Angelo Perkusich

Orientador

Campina Grande, Paraíba, Brasil

©Tomaz de Carvalho Barros, 18 de Fevereiro de 2000



B277s Barros, Tomaz de Carvalho  
Supervisao de sistemas de producao baseada em redes de petri / Tomaz de Carvalho Barros. - Campina Grande, 2000. 152 f.

Tese (Doutorado em Engenharia Eletrica) - Universidade Federal da Paraiba, Centro de Ciencias e Tecnologia.

1. Redes de Petri 2. Sistemas de Producao 3. Sistemas Híbridos 4. Tese - Engenharia Eletrica I. Perkusich, Angelo II. Universidade Federal da Paraiba - Campina Grande (PB)

CDU 681.3.02(043)

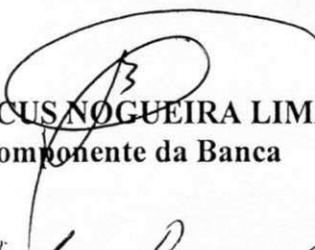
**SUPERVISÃO DE SISTEMAS DE PRODUÇÃO BASEADA EM REDES DE PETRI**

**TOMAZ DE CARVALHO BARROS**

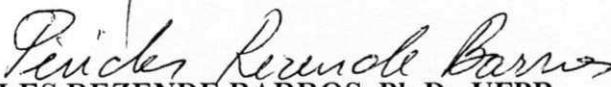
Tese Aprovada em 18.02.2000



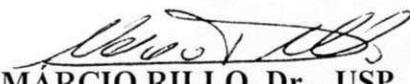
**ANGELO PERKUSICH, D.Sc., UFPB**  
Orientador



**ANTONIO MARCUS NOGUEIRA LIMA, Dr., UFPB**  
Componente da Banca



**PÉRICLES REZENDE BARROS, Ph.D., UFPB**  
Componente da Banca



**MÁRCIO RILLO, Dr., USP**  
Componente da Banca



**RAFAEL SANTOS MENDES, Dr., UNICAMP**  
Componente da Comissão

CAMPINA GRANDE - PB  
Fevereiro - 2000

## Dedicatória

Dedico esta tese a minha querida esposa Jussara e filhos Maxwell, Marconi e Mayara que durante todos estes anos sempre me apoiaram e incentivaram. Dedico também esta tese em memória do meu querido pai Roque. ✓

## Agradecimentos

A Deus que até agora tem nos fortalecido e ajudado.

Ao meu amigo e orientador Professor Angelo Perkusich e esposa Lígia.

A minha querida esposa Jussara e filhos Maxwell, Marconi e Mayara.

A minha querida mãe Nubia e irmãs Telma e Telina.

Ao meu querido irmão Tadeu e esposa Zélia.

A minha amiga Francinete do setor de bolsas da CAPES/UFPE.

A CAPES pelo suporte financeiro.

Aos professores integrantes da banca de avaliação.

Aos amigos, colegas e professores que conviveram comigo durante o doutorado.

## Resumo

Este trabalho trata da supervisão dos sistemas de produção empregando redes de Petri. No contexto desta tese, o termo sistemas de produção refere-se aos sistemas de manufatura e de produção em lotes. Os problemas centrais tratados estão relacionados com a integração e a adaptabilidade de tais sistemas às mudanças nos requisitos operacionais motivadas pela demanda de mercado, requisitos de planejamento, disponibilidade de matéria prima, entre outros. Além disso, considera-se a tolerância a faltas e a verificação de ordens de produção para sistemas de manufatura e receitas para a produção de lotes, em nível de supervisão.

Devido a estas adaptações, a estrutura original do modelo e o supervisor podem ser modificados. Entretanto, a preservação de tais estruturas é importante, pois possibilita agilizar as adaptações. Assim como outros tipos de sistemas, os sistemas de produção estão sujeitos a falhas e portanto a perdas de produtividade. Portanto, os aspectos relacionados com a adaptabilidade e a agilidade dos sistemas de produção são considerados nesta tese.

Na fase de projeto devemos provar que os sistemas de produção são ou não limitados. No caso de sistemas de manufatura, tal procedimento possibilita por exemplo o dimensionamento adequado de áreas de armazenamento intermediárias no chão de fábrica, o que pode implicar em redução de custos. No caso dos sistemas de produção em lote, busca-se por exemplo evitar o transbordo de áreas de armazenamento, resultando em potenciais perdas da produção. Para evitar esta situação indesejável, antes de iniciar a produção devemos provar que a receita é limitada.

Nesta tese introduz-se um procedimento de modelagem sistemático e estruturado para a obtenção tanto de modelos como do supervisor para sistemas de produção. As soluções introduzidas permitem generalizar a estrutura do supervisor, desta forma proporcionando maior agilidade para adaptações.

## Abstract

This work deals with the supervision of production systems using Petri Nets. In this dissertation production systems refer to both, manufacturing systems and batch production systems. The main problems tackled are related to the integration and adaptability of such systems to changes in the operational requirements. These changes may be motivated by market demand, planning requirements, availability of raw materials, among others. Furthermore, we take into account fault tolerance aspects as well as the verification of production orders for manufacturing systems and recipes for batch production systems.

Due to the adaptations, the original structure of the model and the supervisor may need to be changed. On the other hand, the preservation of such structures is important so that it promotes the adaptation in an agile way. As well as other kind of systems, production systems have to be fault tolerant in order to guarantee productivity. Therefore, fault tolerance aspects as well as agility are taken into account in this research.

During the design phase, one has to prove that the production systems are bounded. In the case of manufacturing systems boundedness properties can be used, for example, to adequate buffers and storages in the shop floor, that may lead, for example, to cost reduction. In the case of batch production systems, overflows must be avoided, because they may result in production loss. In order to avoid such undesired situations, it is important to prove that the recipe is bounded.

In the research we introduce a systematic and structured modeling procedure to obtain both, the model of the system and the supervisor for production systems. The introduced solutions allow the generalization of the structure of the supervisor, therefore promoting a higher level of adaptability in case of changes in the operational requirements.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Sistemas Flexíveis de Manufatura . . . . .	2
1.2	Sistemas de Produção em Lotes . . . . .	5
1.3	Arquitetura e Funcionalidade do Controle . . . . .	7
1.4	Apresentação das Redes de Petri . . . . .	10
1.5	Objetivos da Tese . . . . .	14
1.5.1	Problema da Integração do Sistema . . . . .	14
1.5.2	Problema da Adaptabilidade do Sistema . . . . .	15
1.5.3	Problema da Tolerância a Faltas . . . . .	16
1.5.4	Problema da Verificação de Receitas . . . . .	16
1.6	Utilização das Redes de Petri Nesta Tese . . . . .	17
1.7	Organização da Tese . . . . .	17
<b>2</b>	<b>Conceitos Básicos</b>	<b>19</b>
2.1	Introdução às Redes de Petri . . . . .	19
2.2	Análise de Modelos de Redes de Petri . . . . .	22
2.2.1	Enumeração do Espaço de Estados . . . . .	24
2.2.2	Equações Algébricas . . . . .	27
2.2.3	Um Sistema Multi Robô . . . . .	27
2.3	Redes de Petri Temporizadas . . . . .	31
2.4	Redes de Petri Coloridas . . . . .	34
2.4.1	Operação de uma Máquina . . . . .	37
2.4.2	Redes de Petri Coloridas Hierárquicas . . . . .	39
2.5	Tolerância a Faltas . . . . .	40

2.6	Considerações Finais . . . . .	42
<b>3</b>	<b>Supervisão de Sistemas Flexíveis de Manufatura</b>	<b>44</b>
3.1	Definição para Sistemas Flexíveis de Manufatura . . . . .	44
3.1.1	Exemplo de um Sistema Flexível de Manufatura . . . . .	46
3.1.2	Sistema de Transporte . . . . .	47
3.2	Especificação do Supervisor . . . . .	50
3.3	Especificação do Supervisor em Redes de Petri Coloridas . . . . .	57
3.3.1	Descrição da Página <i>fluxo de ordens de serviço</i> . . . . .	58
3.3.2	Descrição da Página <i>be para bec</i> . . . . .	62
3.4	Funcionalidade e Análise do Supervisor . . . . .	63
3.5	Considerações Finais . . . . .	67
<b>4</b>	<b>Supervisão de Sistemas Flexíveis de Manufatura Tolerantes a Faltas</b>	<b>68</b>
4.1	Os Casos de Re-Escalonamento de Rotas de Produção . . . . .	68
4.1.1	Caso 1 . . . . .	70
4.1.2	Caso 2 . . . . .	70
4.1.3	Caso 3 . . . . .	70
4.1.4	Caso 4 . . . . .	71
4.2	Especificação do Supervisor . . . . .	71
4.2.1	Descrição da Página <i>fluxo de ordens de serviço</i> . . . . .	72
4.2.2	Descrição da Página <i>re-escalonamento bem</i> . . . . .	77
4.2.3	Descrição da Página <i>re-escalonamento bec</i> . . . . .	79
4.2.4	Descrição da Página <i>re-escalonamento be</i> . . . . .	80
4.2.5	Descrição da Página <i>re-escalonamento bsc</i> . . . . .	80
4.2.6	Descrição da Página <i>re-escalonamento bsm</i> . . . . .	81
4.3	Exemplo de Re-Escalonamento de Rotas de Produção . . . . .	81
4.4	Considerações Finais . . . . .	83
<b>5</b>	<b>Supervisão de Sistemas de Produção em Lotes</b>	<b>85</b>
5.1	Definição para Sistemas de Produção em Lotes . . . . .	85
5.1.1	Receitas de Baixo Nível . . . . .	87
5.1.2	Tanques Depósitos de Saída . . . . .	90

---

5.1.3	Tanques Depósitos de Entrada . . . . .	92
5.2	Especificação do Supervisor . . . . .	94
5.3	Especificação do Supervisor em Redes de Petri Coloridas . . . . .	97
5.3.1	Supervisão Fixa . . . . .	97
5.3.2	Supervisão Dinâmica . . . . .	100
5.4	Alocação dos Recursos de Manutenção . . . . .	104
5.4.1	Arquitetura do Controle Considerando os Recursos de Manutenção	104
5.4.2	Descrição do Modelo do Supervisor . . . . .	106
5.5	Funcionalidade e Análise do Supervisor . . . . .	108
5.6	Considerações Finais . . . . .	110
<b>6</b>	<b>Verificação de Receitas para Sistemas de Produção em Lotes</b>	<b>111</b>
6.1	Determinação dos Estados Estáveis do Sistema . . . . .	112
6.2	Determinação dos Vetores de Estado Contínuo . . . . .	117
6.3	Algoritmos para Cálculo e Verificação de Capacidade . . . . .	121
6.4	Aplicação dos Resultados Obtidos . . . . .	123
6.5	Considerações Finais . . . . .	132
<b>7</b>	<b>Conclusão e Perspectivas</b>	<b>134</b>
7.1	Sumário . . . . .	135
7.2	Contribuições da Tese . . . . .	136
7.3	Perspectivas . . . . .	136

# Lista de Figuras

1.1	Célula de manufatura 1 . . . . .	3
1.2	Célula de manufatura 2 . . . . .	4
1.3	Re-escalonamento da rota de produção $rota_i$ . . . . .	5
1.4	Exemplo de um sistema de produção em lotes . . . . .	7
1.5	Arquitetura do controle hierárquico . . . . .	8
1.6	Funcionalidade do supervisor . . . . .	9
1.7	Utilização das redes de Petri nas fases de concepção e implementação dos sistemas de produção . . . . .	11
1.8	Modelo da célula de manufatura 1 . . . . .	12
1.9	Utilização das redes de Petri nesta tese . . . . .	18
2.1	Rede de Petri utilizada para exemplificar as regras de transição . . . . .	21
2.2	(a) Modelagem de atividades paralelas (b) Conflito ou decisão . . . . .	22
2.3	Redes utilizadas para exemplificar os níveis de vivacidade . . . . .	24
2.4	(a) Árvore de cobertura (b) Grafo de cobertura . . . . .	26
2.5	Sistema multi robô . . . . .	28
2.6	Modelo do sistema multi robô . . . . .	29
2.7	Árvore de alcançabilidade do sistema multi robô . . . . .	31
2.8	Grafo de alcançabilidade do sistema multi robô . . . . .	32
2.9	Evolução de uma rede de Petri P-temporizada . . . . .	33
2.10	Evolução em máxima velocidade . . . . .	34
2.11	Exemplo de uma máquina . . . . .	37
2.12	Modelos da máquina . . . . .	38
2.13	(a) Estufa (b) Estufa tolerante a faltas . . . . .	42

3.1	Exemplo de um SFM . . . . .	46
3.2	Grafo de atividades para o SFM da Figura 3.1 . . . . .	47
3.3	(a) VAG bidirecional (b) VAG unidirecional . . . . .	48
3.4	Esquema distribuído aplicado ao SFM da Figura 3.1 . . . . .	48
3.5	Fases de transporte . . . . .	50
3.6	Refinamento dos níveis de abstrações de um SFM . . . . .	51
3.7	Modelo das Atividades Fora das Células . . . . .	52
3.8	Modelo das Atividades Dentro das Células . . . . .	53
3.9	Modelo dos Recursos de Produção . . . . .	55
3.10	Modelo do SFM apresentado na Figura 3.1 . . . . .	56
3.11	Estrutura de alto nível . . . . .	57
3.12	Páginas do supervisor . . . . .	58
3.13	Página <i>fluxo de ordens de serviço</i> . . . . .	60
3.14	Página <i>be para bec</i> . . . . .	63
3.15	Declaração de variáveis e cores . . . . .	64
3.16	Manufatura de porcas . . . . .	65
3.17	Estado de bloqueio . . . . .	67
4.1	Re-escalamento para o Caso 1 . . . . .	70
4.2	Re-escalamento para o Caso 2 . . . . .	71
4.3	Re-escalamento para o Caso 3 . . . . .	72
4.4	Re-escalamento para o Caso 4 . . . . .	73
4.5	Páginas do supervisor . . . . .	74
4.6	Página <i>fluxo de ordens de serviço</i> . . . . .	75
4.7	Declaração de variáveis e cores . . . . .	76
4.8	Falha do recurso $r_4$ . . . . .	76
4.9	Página <i>re-escalamento bem</i> . . . . .	78
4.10	Páginas <i>re-escalamento bem</i> e <i>fluxo de ordens de serviço</i> . . . . .	78
4.11	Página <i>re-escalamento bec</i> . . . . .	79
4.12	Página <i>re-escalamento be</i> . . . . .	80
4.13	Página <i>re-escalamento bsc</i> . . . . .	81
4.14	Página <i>re-escalamento bsm</i> . . . . .	82

4.15	SFM tolerante a faltas . . . . .	82
4.16	Grafo de atividades considerando a falha dos recursos $r_1$ , $r_4$ , $r_6$ e $r_8$ . . . . .	83
5.1	(a) Visão geral de um SPL (b) Arquitetura de um reator . . . . .	86
5.2	(a) Exemplo de um SPL (b) Visão detalhada do reator $s_5$ . . . . .	88
5.3	Exemplo de um SPL utilizando os tanques depósitos de saída . . . . .	90
5.4	Exemplo de um SPL utilizando os tanques depósitos de entrada . . . . .	92
5.5	Fluxo de produção dos reatores ilustrado na Figura 5.4 . . . . .	94
5.6	Modelo do supervisor para o sistema ilustrado na Figura 5.2 . . . . .	95
5.7	Modelo do supervisor para o sistema ilustrado na Figura 5.4 . . . . .	96
5.8	Estrutura de alto nível . . . . .	98
5.9	Modelo do supervisor para o sistema ilustrado na Figura 5.3 . . . . .	99
5.10	SPL utilizado para exemplificar a supervisão dinâmica . . . . .	100
5.11	Páginas do supervisor . . . . .	101
5.12	Página <i>fluxo de ordens de serviço</i> . . . . .	102
5.13	Página <i>alternativas</i> . . . . .	103
5.14	Arquitetura considerando a alocação dos recursos de manutenção do sistema	105
5.15	Reator tolerante a faltas . . . . .	106
5.16	Página <i>alternativas</i> modificada . . . . .	107
5.17	Marcação inicial e final do modelo apresentado na seção 5.3.2 . . . . .	110
6.1	Utilização das redes de Petri para tratar com o problema da verificação de receitas . . . . .	111
6.2	Sistema tanque-válvulas . . . . .	113
6.3	Modelo do sistema tanque-válvulas . . . . .	115
6.4	Representação das marcações estáveis do sistema tanque-válvulas . . . . .	116
6.5	Variação de volume de um lote para o sistema tanque-válvulas . . . . .	122
6.6	Sistema de produção de bebidas . . . . .	125
6.7	Modelo do supervisor para o sistema da Figura 6.6 . . . . .	126
6.8	Variação de volume no reator $s_5$ . . . . .	132

# Capítulo 1

## Introdução

Devido ao mercado competitivo e globalizado da atualidade, tem sido cada vez maior o interesse da ciência e dos empresários no campo industrial em sistemas de produção mais produtivos e automatizados. No contexto desta tese, o termo sistema de produção é utilizado para referenciar tanto sistemas de manufatura como de produção em lotes. A operação de tais sistemas envolve um conjunto de tomadas de decisões complexas, tornando-se mais críticas à medida que tais decisões envolvem diretamente os dispositivos físicos ligados ao processo. Com o objetivo de gerenciar tal complexidade, estrutura-se a gerência destes sistemas em níveis hierárquicos denominados *planejamento*, *escalonamento*, *supervisão*, *coordenação* e *comando local*. Nesta tese, focalizamos o nível de supervisão considerando os problemas da integração, adaptabilidade, tolerância a faltas e limitação de tais sistemas. Na Seção 1.3 apresentamos cada um destes níveis hierárquicos e na Seção 1.5 estes problemas.

Um aspecto relevante no contexto de sistemas de produção é a necessidade de adaptá-los para atender a demanda de mercado, preferência dos consumidores, requisitos de planejamento, disponibilidade de matéria prima, etc. O problema é que devido a estas adaptações, o supervisor de tais sistemas pode não ser preservado, seja em sua estrutura, em suas funcionalidades ou ambas. Entretanto, preservar o supervisor é importante, visto que torna os sistemas de produção mais *ágeis* [Zho95] e, quanto mais ágil é um sistema, mais rápido este pode ser adaptado.

Outro aspecto relevante a considerar é a possibilidade da ocorrência de falhas nos mais diversos componentes do sistema. Tais falhas podem implicar degradação funcional, de

desempenho ou ambas. Tais degradações podem ser intoleráveis e podem resultar em prejuízos das mais diversas naturezas, incluindo, por exemplo, perdas de produtividade. Desta forma é imprescindível estabelecer técnicas e mecanismos objetivando minimizar ou mesmo evitar tais perdas. A solução é implementar tolerância a faltas [Jal94].

Além disso, os sistemas de produção devem ser limitados, ou seja, os limites de armazenamento devem ser respeitados durante todo o processo de produção. No caso de sistemas de manufatura, a limitação está relacionada com o dimensionamento adequado de áreas de armazenamento intermediárias no chão de fábrica, o que pode implicar em redução de custos. No caso dos sistemas de produção em lotes busca-se, por exemplo, evitar o transbordo de áreas de armazenamento, resultando em potenciais perdas da produção. Para evitar esta situação indesejável, antes de iniciar a produção é imperativo provar que não ocorrerá transbordamentos, ou seja, que as receitas em questão são limitadas. A seguir apresentamos os sistemas de produção considerados nesta tese.

## 1.1 Sistemas Flexíveis de Manufatura

Os sistemas flexíveis de manufatura (SFM) são sistemas distribuídos, assíncronos e seqüenciais podendo exibir sincronismos, concorrência e conflitos [COKG96; HSBM96; KC97b; Jen97a; Val90; LD94; Zur94; FMMT97; ZDR92; Rig93]. Tais sistemas podem ser constituídos de subsistemas flexíveis denominados células de manufatura. As células, interligadas por um sistema de transporte, podem possuir uma ou mais máquinas e seus recursos de produção, como ferramentas. Por exemplo, numa máquina furadeira, a ferramenta broca é o recurso de produção da máquina. Os recursos de transporte, robôs e Veículos Auto Guiados (VAGs) são utilizados para transportar produtos, subprodutos ou a matéria-prima dentro ou fora das células do sistema [PMK97; SCF94; BM97; EC97; SV94; LD94]. O sistema, as células e as máquinas possuem depósitos (*buffers*) utilizados para armazenar temporariamente produtos, subprodutos ou matéria-prima. Denominamos por *rota de produção* a seqüência finita de recursos de produção utilizados para transformar a matéria-prima em um determinado produto [HWL95; FWBD96; ZBH96; Zim95; BPF97b; KD97; EC97].

Na Figura 1.1 ilustra-se uma célula de manufatura formada por dois robôs e duas máquinas. O *robô 1* transporta a matéria-prima do depósito de entrada da célula para os

depósitos de entrada das máquinas, ou seja, carrega as máquinas. O *robô 2* transporta subprodutos do depósito de saída das máquinas para o depósito de saída da célula, ou seja, descarrega as máquinas. Neste exemplo, as rotas de produção são  $rota_1$  e  $rota_2$ .

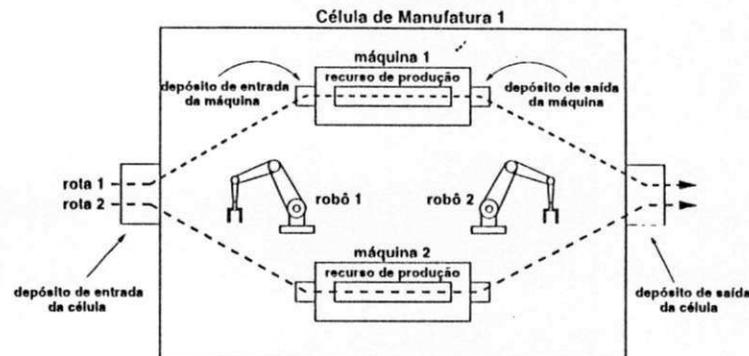


Figura 1.1: Célula de manufatura 1

Os SFMs são caracterizados como sendo Sistemas a Eventos Discretos (SEDs) [KCG97; RK97; ZZ94; BLP96; LMH94; DA94; Hey89; Cap93]. Em tais sistemas o espaço de estados é discreto e a causa da mudança de estados é a ocorrência de eventos. Além disso, os recursos de um SFM são compartilhados por diversas atividades. Este compartilhamento deve ser controlado ou sincronizado para garantir a correta operação do sistema como um todo.

Alguns eventos podem ser dependentes da ocorrência de outros eventos, definindo portanto uma dependência causal entre eles. No exemplo apresentado, a ocorrência do evento *iniciar processamento na máquina 1* depende da ocorrência do evento *fim de carregamento da máquina 1*. Da mesma forma, o evento *iniciar descarregamento da máquina 1* depende da ocorrência do evento *fim de processamento na máquina 1*.

Nos SFMs, eventos freqüentemente ocorrem de forma assíncrona. Por exemplo, o evento *iniciar descarregamento da máquina 1* ocorrerá provavelmente em intervalos de tempo irregulares e diferentes com relação a ocorrência de outros eventos.

Os SFMs são seqüenciais com respeito a um dado conjunto de eventos. Dois eventos  $\alpha$  e  $\beta$  são ditos seqüenciais nesta ordem, se o evento  $\beta$  sempre ocorrer após o evento  $\alpha$ . Por exemplo, o evento *fim de processamento na máquina 1* sempre ocorrerá após o evento *iniciar processamento na máquina 1*.

Dois eventos  $\alpha$  e  $\beta$  são ditos concorrentes se o evento  $\alpha$  pode ocorrer antes ou após o evento  $\beta$ . Por exemplo, os eventos *iniciar processamento na máquina 1* e *iniciar pro-*

cessamento na máquina 2 são concorrentes. Isto porque, as máquinas são programadas para executar atividades independentes começando em instantes de tempo provavelmente diferentes. Assim a máquina 1 pode começar o processamento antes, em paralelo ou após a máquina 2.

Dois eventos estão em conflito se a ocorrência de um evento impede a ocorrência do outro evento e vice-versa. Considerando ainda o exemplo mostrado na Figura 1.1, os eventos *iniciar carregamento da máquina 1* e *iniciar carregamento da máquina 2* estão em conflito pois, o robô 1 não pode carregar ao mesmo tempo a máquina 1 e a máquina 2.

Além de resolver conflitos, o controle deve assegurar a exclusão mútua de cada recurso. Por exemplo, o robô 1 é um recurso compartilhado pelas máquinas, logo, ele somente deve ser alocado para carregar uma máquina se estiver livre para executar a operação. Portanto, enquanto ele não terminar a operação, voltar ao estado livre, não poderá ser alocado para outra operação.

Outro problema a ser considerado nos SFMs é que a ocorrência de falhas pode comprometer a funcionalidade ou o desempenho do sistema, resultando, por exemplo, em perdas de produtividade. Como exemplo, considerando a célula de manufatura mostrada na Figura 1.2 formada por três robôs e três máquinas, temos:

1. O robô 1 carrega a máquina 1.
2. O robô 2 descarrega a máquina 1 e carrega a máquina 2 ou a máquina 3.
3. O robô 3 descarrega a máquina 2 ou a máquina 3.

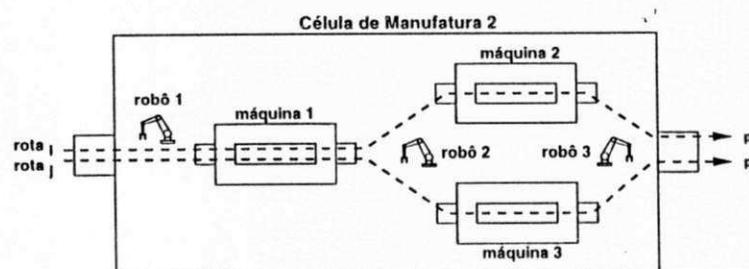


Figura 1.2: Célula de manufatura 2

Conforme é ilustrado na Figura 1.2, as rotas de produção consideradas são  $rota_i$  e  $rota_j$  resultando nos produtos  $p_i$  e  $p_j$ , respectivamente. Suponha os seguintes casos: (a)

a máquina 1 falhou e (b) somente a máquina 2 falhou e que a máquina 3 é a réplica da máquina 2, ou seja, a máquina 3 é capaz de executar as mesmas atividades da máquina 2. Observe que, ocorrendo o caso (a), ilustrado na Figura 1.3(a), a produção dos produtos  $p_i$  e  $p_j$  pode ser interrompida. Isto porque, devido à falha da máquina 1, o robô 2 não poderá carregar mais a máquina 2 e a máquina 3. Ocorrendo o caso (b), ilustrado na Figura 1.3(b), apenas a produção do produto  $p_i$  pode ser interrompida. Entretanto, a máquina 3 pode ser usada para assegurar também a produção da máquina 2. Observe que, para o caso (b), as rotas de produção são agora  $rota_k$  e  $rota_j$ . Isto porque, devido à falha da máquina 2, a rota de produção  $rota_i$  foi re-escalorada resultando na rota de produção  $rota_k$ . O processo de transformar uma rota de produção em outra diferente para o mesmo produto e livre de bloqueios será denominado de *re-escalonamento de rota de produção*.

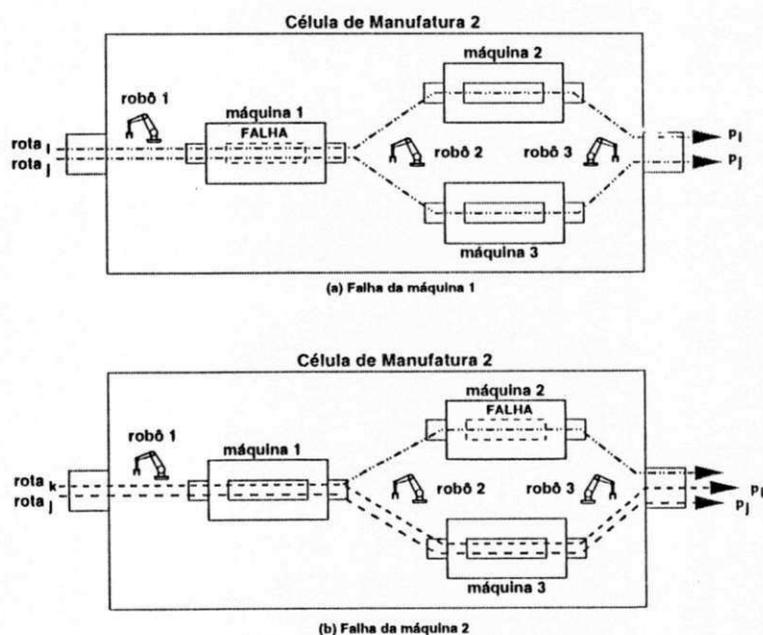


Figura 1.3: Re-escalonamento da rota de produção  $rota_i$

## 1.2 Sistemas de Produção em Lotes

Os sistemas de produção em lotes são formados por diversos recursos de armazenamento, produção e transporte. Os recursos de armazenamento, denominados tanques, armazenam temporariamente matéria-prima, subprodutos ou produtos. Os recursos de produção, encontrados nos reatores, são utilizados para produzir os lotes. Um lote é a quantidade de um

subproduto produzida em um reator do sistema. Esta quantidade, expressa em unidades de massa ou volume, varia e depende da receita de cada produto. Os recursos de transporte permitem a transferência dos lotes entre os tanques do sistema. Devido a transferência dos lotes, há um fluxo entre os recursos, denominado fluxo de produção.

Como exemplo de um sistema de produção em lotes considere uma máquina de vendas de café, ilustrada na Figura 1.4. A máquina é formada por dois tanques de entrada, um reator, um conjunto de válvulas e um tanque de saída (copo). Nos tanques de entrada estão armazenados o café solúvel e o leite em pó solúvel. Desta forma, podemos ter duas receitas, uma para a produção de café puro e outra para a produção de café com leite. O reator é formado por um tanque, a válvula 4 de abastecimento de água, a válvula 5 do dreno do tanque do reator, um aquecedor de água, dutos e um motor de acionamento do misturador. O transporte entre os tanques de entrada e o reator é feito por meio das válvulas 1 e 2, e o transporte entre o reator e o copo é feito por meio da válvula 3. Para produzir o café puro, uma certa quantidade de café solúvel, especificada na receita, é transferida por meio da válvula 1 para o tanque de reator. Terminada a fase de transferência, a fase de processamento é iniciada. Nesta fase é adicionada água quente ao café solúvel. Enquanto isto, o motor do misturador é acionado. Terminada a fase de processamento, o café puro é transferido para o copo por meio da válvula 3. Semelhantemente, é produzido o café com leite. Após a execução de uma receita, o reator deve ser preparado para a execução da receita seguinte. Para o exemplo ilustrado na Figura 1.4, esta preparação é feita lavando-se o tanque do reator por meio da válvula 4 de abastecimento de água e da válvula dreno 5 do tanque do reator.

Da mesma forma que os sistemas flexíveis de manufatura, os sistemas de produção em lotes (SPLs) são sistemas distribuídos, assíncronos e seqüenciais podendo exibir sincronismos, concorrência e conflitos [YPK90; Han92]. Em contraste, os sistemas de produção em lotes podem ser descritos por um conjunto de equações diferenciais. Sob este ponto de vista, estes são considerados como sendo sistemas híbridos [APPV94; Bra95; VPPP95] pois podem ser representados por variáveis discretas e contínuas.

Assim, como no caso dos sistemas de manufatura, a ocorrência de faltas também pode comprometer a produção dos lotes. Para evitar isto, os reatores podem ser formados por partes redundantes e que podem ser utilizadas no lugar de outras defeituosas assegurando assim a continuidade da produção. Por exemplo, o aquecedor do reator da máquina de

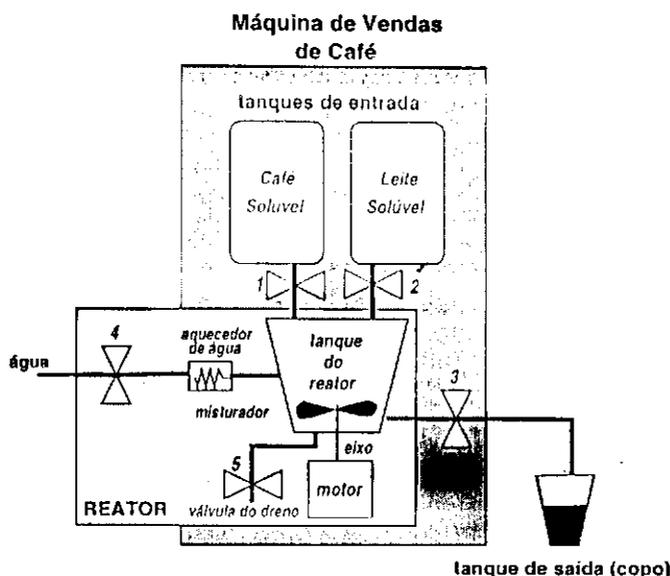


Figura 1.4: Exemplo de um sistema de produção em lotes

vendas de café poderia ser duplicado, desta forma disponibilizando uma réplica.

### 1.3 Arquitetura e Funcionalidade do Controle

Como dito no início deste capítulo, o controle de um sistema de produção envolve um conjunto de tomadas de decisões complexas, envolvendo desde o planejamento da produção até o controle dos dispositivos físicos ligados diretamente ao processo. Além disso, tais processos de decisão podem ser a longo prazo ou a curto prazo, tornando-se mais críticos à medida que se aproximam do processo físico. Assim, para gerenciar a complexidade, tanto lógica como temporal, o gerenciamento de sistemas de produção é dividido em níveis hierárquicos denominados *planejamento*, *escalonamento*, *supervisão*, *coordenação* e *comando local* [GHSM86; MQFK88; Bar90; ZD93; Per94; STVP98; VCD85]. É evidente que, a correta funcionalidade do sistema dependerá da interação e da integração de todos estes níveis, sendo mais crítica nos três últimos níveis (supervisão, coordenação e comando local) devido as exigências de operação em tempo real. Baseado nesta decomposição, apresenta-se na Figura 1.5 a arquitetura do controle hierárquico, descrita a seguir.

O nível de comando local tem como função o controle dos dispositivos físicos ligados diretamente ao processo. Por exemplo, o controle do atuador de abertura e fechamento de

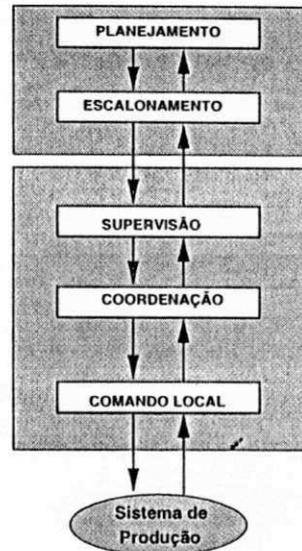


Figura 1.5: Arquitetura do controle hierárquico

uma válvula, o controle do freio da placa de um torno mecânico, o controle do acionamento do motor de uma esteira, etc.

O nível de coordenação, implementado pelos gerenciadores locais de recursos (*GLRs*), tem como função controlar subsistemas em resposta às requisições de serviço enviadas pelo nível de supervisão do sistema. Cada requisição de serviço corresponde a uma atividade de transporte ou produção do sistema.

O nível de supervisão, implementado pelo *supervisor*, toma decisões cada vez que uma *requisição de serviço* é executada. Denomina-se *ordem de serviço* o conjunto de requisições de serviços requeridas para um determinado produto. Denomina-se *ordem de serviço executada* a ordem de serviço que possui todas suas requisições atendidas. O supervisor para alocar e gerenciar concorrentemente os recursos compartilhados do sistema, deve ser concebido para:

1. interagir com os gerenciadores locais de recursos por meio da troca de mensagens,
2. reconhecer a qualquer instante todos os recursos livres do sistema,
3. determinar as atividades concorrentes de todos os recursos alocados,
4. resolver conflitos assegurando a exclusão mútua dos recursos alocados e
5. reconhecer e tomar decisões após o término de cada atividade.

Os níveis de planejamento e escalonamento possuem funções de controle relacionadas com a gestão da produção. Estes dois níveis não levam em conta exigências de tempo real.

Com base na arquitetura hierárquica de controle, na Figura 1.6 mostra-se como o supervisor interage com os recursos do sistema. Para exemplificar a interação, considere o carregamento da *máquina 1* ilustrada na Figura 1.1. Neste caso, o supervisor interage com o *robô 1* da seguinte forma:

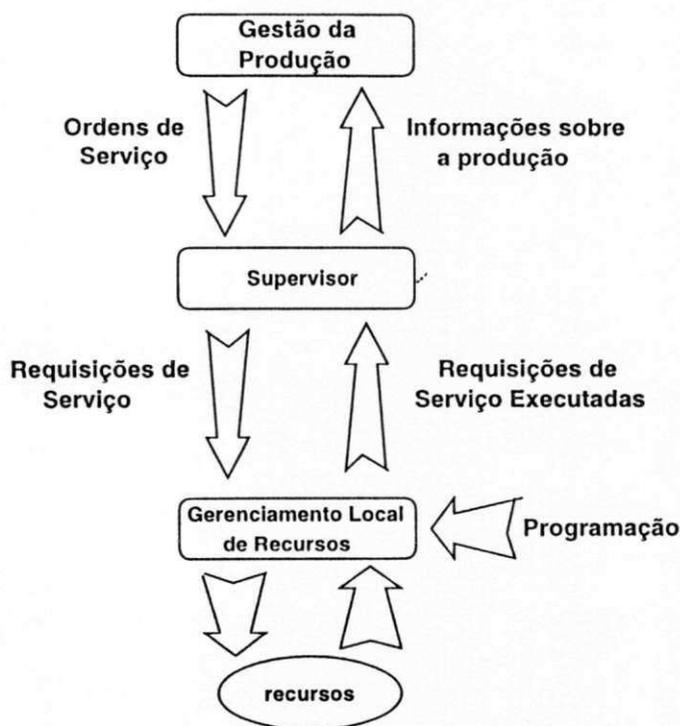


Figura 1.6: Funcionalidade do supervisor

1. o supervisor envia uma requisição de serviço para o gerenciador local do robô solicitando o carregamento da máquina,
2. após receber a requisição de serviço, o gerenciador local começa controlar o robô para fazer o carregamento da máquina e,
3. após o término do serviço, o gerenciador local informa ao supervisor que o serviço foi executado, ou seja, que o robô está livre para a próxima alocação.

Como um segundo exemplo, considere a máquina de vendas de café ilustrada na Figura 1.4 e a execução da receita de café puro. Neste caso, o supervisor interage com o reator da seguinte forma:

1. o supervisor envia uma requisição de serviço para o gerenciador local do reator solicitando a produção do café,
2. após receber a requisição de serviço, o gerenciador local começa controlar os dispositivos internos do reator para produzir o café e,
3. após o término do serviço, o gerenciador local informa ao supervisor que o serviço foi executado, ou seja, que o café está armazenado no copo.

Portanto, os gerenciadores locais são programados para controlar localmente os recursos do sistema isto em atendimento às requisições de serviço enviadas pelo supervisor. A programação é feita com base nas receitas ou rotas de produção do sistema.

## 1.4 Apresentação das Redes de Petri

Uma rede de Petri [Mur89] é um grafo direcionado, ponderado e bipartido consistindo de dois tipos de nós, denominados lugares e transições, onde arcos podem ser direcionados de um lugar para uma transição ou de uma transição para um lugar. A cada arco pode ser associado um número inteiro  $k$  que é denominado de peso ou ponderação do arco. Na representação gráfica, lugares são representados por círculos e transições por barras ou retângulos. A marcação atribuída a cada lugar um inteiro não negativo representado por marcas denominadas fichas.

As redes de Petri têm sido usadas para a modelagem e controle dos sistemas de produção notadamente, dos sistemas flexíveis de manufatura [PX97; DAJ94]. Na Figura 1.7 mostra-se como as redes de Petri podem ser utilizadas neste contexto específico de pesquisa [ZD93]. Na fase de concepção, o sistema é modelado e, após, o modelo é analisado. Na fase de implementação, o sistema e o supervisor são implementados.

Assim como as redes de Petri, outras metodologias tem sido usadas para a modelagem e controle de sistemas de produção, como, por exemplo, a teoria dos autômatos finitos, a teoria de controle supervisório [RW89] e os processos recursivos finitos [IV88; Hoa85]. Entretanto, a aplicação destas teorias pode ser limitada pelo crescimento exponencial do espaço de estados do sistema. Os processos recursivos finitos são baseados nos processos seqüenciais de comunicação de Hoare. Dado um conjunto de eventos, um processo é formado por três componentes: um conjunto de caminhos (*traces*) que o processo

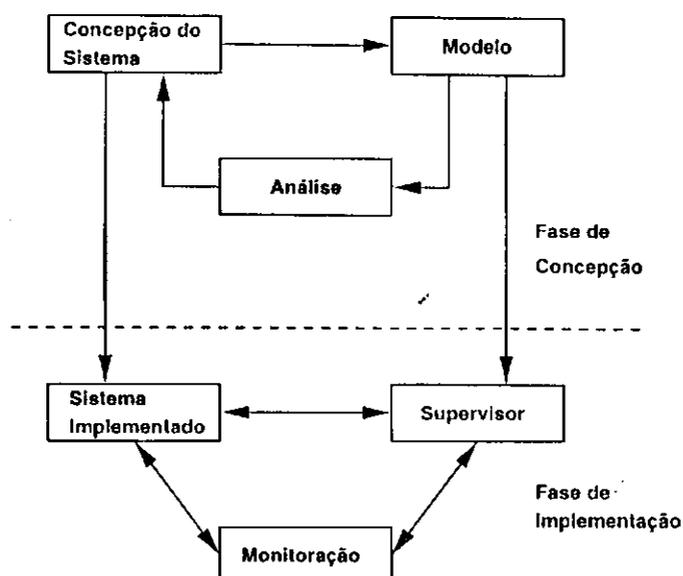


Figura 1.7: Utilização das redes de Petri nas fases de concepção e implementação dos sistemas de produção

pode executar, uma função de eventos e uma função de terminação. Uma das características dos processos recursivos finitos é que cada processo pode ser descrito por um conjunto de equações recursivas. Assim, um sistema pode ser descrito com base nestas equações. Entretanto, alguns problemas permanecem em aberto, como a utilização destas equações para a especificação de supervisores para sistemas em tempo real.

Com relação a estas metodologias, as redes de Petri apresentam as seguintes vantagens [ZD93; ST96]:

1. facilidade para modelar sistemas exibindo concorrência, sincronismo e assincronismo, conflito, exclusão mútua, seqüenciamento de eventos e bloqueios,
2. possibilitam a utilização de abordagens do tipo refinamento (*top-down*) [ZDR92; HLMM95], do tipo composição modular (*bottom-up*) [FCP96; HDS96; KC97a; Jen97b; PWX97] e do tipo híbrido [PWX97; ZDR92], para a obtenção do modelo. A discussão sobre a utilização de cada um destes métodos pode ser encontrada em [HMN96; JX97; ZD93; Jen97a],
3. possibilidade de especificar o supervisor diretamente do modelo do sistema;
4. possibilidade de provar, por meio da análise do modelo, as propriedades do sistema, tais como bloqueio e limitação,

5. simulação dos eventos discretos diretamente a partir do modelo e
6. informação do estado atual do sistema que permite monitoração em tempo real.

Maiores detalhes sobre as redes de Petri estão apresentados no Capítulo 2. Para exemplificar o uso das redes de Petri, na Figura 1.8 mostra-se o modelo da célula de manufatura apresentada na Figura 1.1, onde lugares e transições possuem o seguinte significado:

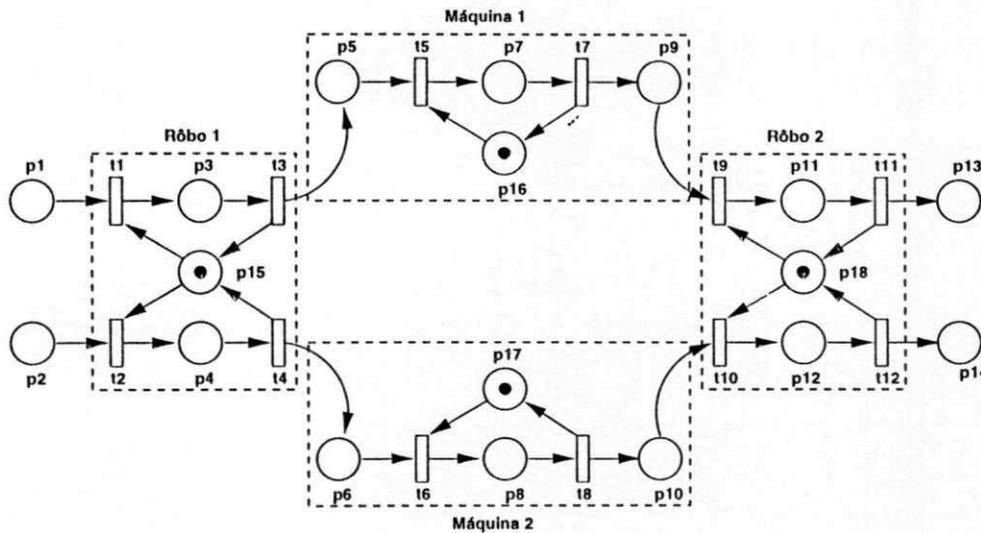


Figura 1.8: Modelo da célula de manufatura 1

Lugares:

- $p_1(p_2)$ : modela ordens de serviço requisitando o carregamento da *máquina 1(2)*,
- $p_3(p_4)$ : modela o *robô 1* carregando a *máquina 1(2)*,
- $p_5(p_6)$ : modela ordens de serviço requisitando a alocação da *máquina 1(2)*,
- $p_7(p_8)$ : modela a *máquina 1(2)* processando,
- $p_9(p_{10})$ : modela ordens de serviço requisitando o descarregamento da *máquina 1(2)*,
- $p_{11}(p_{12})$ : modela o *robô 2* descarregando a *máquina 1(2)*,
- $p_{13}(p_{14})$ : modela ordens de serviço executadas relativo a rota de produção  $rota_1(rota_2)$ ,
- $p_{16}(p_{17})$ : modela a *máquina 1(2)* livre,
- $p_{15}(p_{18})$ : modelo o *robô 1(2)* livre.

Transições:

$t_1(t_2)$ : modela o início do carregamento da *máquina 1(2)*,

$t_3(t_4)$ : modela o fim do carregamento da *máquina 1(2)*,

$t_5(t_6)$ : modela o início da alocação da *máquina 1(2)*,

$t_7(t_8)$ : modela o fim da alocação da *máquina 1(2)*,

$t_9(t_{10})$ : modela o início do descarregamento da *máquina 1(2)*,

$t_{11}(t_{12})$ : modela o fim do descarregamento da *máquina 1(2)*.

Dentro dos retângulos, exibidos em linha tracejada, temos os conjuntos de lugares e transições que modelam os componentes (robôs e máquinas) do sistema. Observe que, aspectos como concorrência, conflito, seqüenciamento de eventos e exclusão mútua são modelados naturalmente por meio das redes de Petri, como detalhado a seguir:

1. *Concorrência*: Por exemplo, as transições  $t_5$  e  $t_6$  são concorrentes, pois, a ocorrência da primeira (segunda) independe da ocorrência da segunda (primeira). Isto porque uma máquina pode começar a processar independentemente da outra máquina.
2. *Conflito*: Por exemplo, as transições  $t_1$  e  $t_2$  representam um conflito, pois, a ocorrência da primeira (segunda) impossibilita a ocorrência da segunda (primeira). Isto porque, o robô 1 somente pode carregar uma máquina de cada vez.
3. *Seqüenciamento de eventos*: Por exemplo, as transições  $t_5(t_6)$  e  $t_7(t_8)$  representam eventos seqüenciais, pois, a ocorrência da transição  $t_7(t_8)$  sempre ocorrerá após a ocorrência da transição  $t_5(t_6)$ . Isto porque, o evento *fim da alocação da máquina 1(2)* sempre ocorrerá após a ocorrência do evento *início da alocação da máquina 1(2)*.
4. *Exclusão mútua*: Visto que os recursos são compartilhados por diversas atividades, estes devem executar uma atividade de cada vez. Os lugares  $p_{15}, p_{16}, p_{17}$  e  $p_{18}$  são usados para assegurar a exclusão mútua dos recursos. Por exemplo, a ocorrência de  $t_1$  remove a ficha de  $p_{15}$  desabilitando assim a ocorrência de  $t_2$ . Isto assegura que, o robô 1 sempre será alocado para executar uma atividade de cada vez.

As redes de Petri podem ser classificadas em redes de baixo ou alto nível. Exemplo de redes de alto nível são as redes de Petri Predicado/Transição [GL79] e as redes de Petri Coloridas (CPN) [Jen92]. Nesta tese, utilizamos as redes de Petri Coloridas pelas seguintes razões:

1. com relação aos modelos de baixo nível, os modelos de rede de Petri Coloridas podem ser mais compactos,
2. existe a possibilidade de preservar a estrutura original do modelo obtido e portanto o supervisor face as adaptações do sistema,
3. as informações contidas nas fichas permitem a resolução de conflitos e
4. a existência de uma ferramenta computacional, denominada *Design/CPN*, que permite a edição e análise de modelos hierárquicos [Jen96b; Jen96a].

Maiores detalhes sobre as redes de Petri Coloridas podem ser encontrados no Capítulo 2.

## 1.5 Objetivos da Tese

Nesta tese introduz-se, utilizando-se redes de Petri, um procedimento de modelagem sistemático e estruturado para a obtenção tanto de modelos como do supervisor para os sistemas de produção aqui considerados, ou seja, os sistemas flexíveis de manufatura e os sistemas de produção em lotes. As soluções introduzidas permitem, em ambos os casos, generalizar a estrutura do supervisor possibilitando assim tratar com um conjunto de problemas relacionados com a integração, adaptabilidade, tolerância a faltas e limitação de tais sistemas. Estes problemas são descritos a seguir.

### 1.5.1 Problema da Integração do Sistema

Os sistemas de produção podem ser formados por um número arbitrário de subsistemas (células de manufatura ou reatores) interligados por um sistema de transporte. Entretanto, o problema é que, quanto maior é o número de subsistemas, maior é o número de atividades e interações entre estes subsistemas e portanto, mais difícil é a descrição global do sistema.

As redes de Petri têm sido usadas para a modelagem e controle de células de manufatura isoladamente [FH97; HC92; Wan96; ZWD95]. Feng, C. [FH97] apresenta um procedimento sistemático para analisar, modelar e simular o controle de uma célula manufatura utilizando redes de Petri estocásticas. A célula é parte integrante de um sistema de manufatura integrado por computador utilizado para a produção de pastilhas de circuitos integrados. À análise do modelo é feita com o objetivo de encontrar bloqueios e determinar as limitações de armazenamento da célula. Huang, H.P. [HC92] utiliza as redes de Petri Coloridas temporizadas para a modelagem e controle de uma célula de manufatura. Com base no modelo obtido, o controlador da célula é implementado. Wang, L.-C. [Wan96] utiliza as redes de Petri orientada a objetos para tratar com o escalonamento e o controle de operações de uma célula de manufatura. Zhou, Q. [ZWD95] apresenta uma metodologia para determinar as estratégias de controle para otimizar a produção de uma célula de manufatura. A metodologia tem como base as redes de Petri estocásticas e os processos de decisão markovianos. Entretanto, para controlar sistemas com integração celular, o problema é que não basta apenas conhecer o comportamento interno de uma célula, mas também como elas estão interligadas para formar o sistema.

### 1.5.2 Problema da Adaptabilidade do Sistema

Os sistemas de produção são adaptados para atender às mudanças nos requisitos operacionais. O problema é que, devido às adaptações, a estrutura original do modelo de rede de Petri de um sistema pode ser modificada. Conseqüentemente, o supervisor do sistema adaptado não é preservado. Entretanto, preservar o supervisor torna os sistemas de produção mais ágeis e quanto mais ágil é um sistema de produção, mais rápido este pode ser adaptado. Por exemplo, para obter a célula de manufatura 2 ilustrada na Figura 1.2, incorporamos uma máquina e um robô à célula de manufatura 1, ilustrada na Figura 1.1. Evidentemente, a estrutura do modelo apresentado na Figura 1.8 deve ser modificada. Como um segundo exemplo, suponha que para atender a preferência do consumidor, a máquina de vendas de café, ilustrada na figura 1.4, seja adaptada para produzir também chocolate quente. Neste caso, a estrutura original do modelo da máquina deve ser modificada.

Dentro deste contexto, as redes de Petri tem sido aplicadas para modelar tais sistemas,

conforme pode ser visto em [EC97; Zim95; YPK90; Tit95]. Ezpeleta, J. [EC97], utilizando as redes de Petri Coloridas, apresenta uma metodologia de modelagem para os sistemas flexíveis de manufatura. Com base nesta metodologia, o modelo da planta e o modelo das rotas de produção do sistema são obtidos. Estes modelos são integrados automaticamente resultando em um único modelo de redes de Petri Coloridas. Assim como Ezpeleta, J., Zimmermann, A. [Zim95] utiliza uma abordagem semelhante para a modelagem dos sistemas flexíveis de manufatura. Yamalidou, E.C. [YPK90] e Tittus, M. [Tit95] utilizam as redes de Petri para a modelagem dos sistemas de produção em lotes. Diferente de Yamalidou, E.C., Tittus, M. utiliza as redes de Petri orientada a objetos. Entretanto, diferente destes trabalhos, apresentamos aqui modelos de alto nível possuindo estruturas que podem ser preservadas, possibilitando modelar qualquer sistema de produção definido no contexto desta tese.

### 1.5.3 Problema da Tolerância a Faltas

Este problema tem como solução encontrar mecanismos para aumentar o grau de confiança no funcionamento e a capacidade de assegurar a continuidade dos serviços dos sistemas de produção. Neste contexto, conforme pode ser visto em [RV94; ST96], as redes de Petri têm sido utilizadas. Ramaswamy [RV94] emprega as *EPNs* (do inglês *Extended Petri Nets*) para a modelagem, análise e simulação de falhas em sistemas flexíveis de manufatura. Esta classe de rede possui multiplicidade dos arcos e várias representações para lugares e fichas. Silva [ST96] discute alguns métodos que combinam a teoria dos códigos corretores de erros e as redes de Petri. A idéia é considerar a marcação como uma palavra e o conjunto de todas as marcações alcançáveis como um código. Observa-se que, a inclusão de lugares implícitos ou testes ao modelo aumenta a distância de Hamming do código, aumentando assim a probabilidade para a detecção e correção de erros [Sif79; SV85; dFdFPdA97]. Em contraste com estes trabalhos, utilizamos aqui as redes de Petri Coloridas e o re-escalonamento de rotas de produção para tratar com tolerância a faltas.

### 1.5.4 Problema da Verificação de Receitas

Em um sistema de produção, os limites de armazenamento do sistema devem ser respeitados. Por exemplo, na máquina de vendas de café apresentada na Figura 1.4, o volume de

café produzido pode exceder a capacidade do copo. Se ocorrer *transbordamentos* parte do café é perdida. Para evitar esta situação indesejável, antes de iniciar a produção do café, devemos provar que a receita é limitada, ou seja, que não ocorrerá transbordamentos. A dificuldade para resolver este problema utilizando redes de Petri, é que a marcação de uma rede de Petri é um vetor de inteiros positivos e as grandezas associadas a tais sistemas são números reais. Portanto, a marcação de um lugar não pode representar a variação de volume em um tanque, uma vez que esta grandeza é um número real.

Para tratar com os aspectos contínuos e discretos do sistema, as redes de Petri híbridas [DA94; DA92], as redes de Petri híbridas de alto nível [Wie96; WS95] ou as redes de Petri Globais [Rez96; RIL95] podem ser utilizadas. Entretanto, a desvantagem da utilização destas redes com relação as redes de Petri é a falta de um formalismo matemático definido para análise. Já o formalismo para analisar as redes de Petri é conhecido e pode ser encontrado, por exemplo, em [Mur89]. Por esta razão, utilizamos aqui as redes de Petri para tratar este tipo de problema.

## 1.6 Utilização das Redes de Petri Nesta Tese

Na Figura 1.9 mostra-se de que forma as redes de Petri são utilizadas nesta tese. Considerando um sistema de produção e os problemas descritos acima, construímos o modelo de tal forma que as ocorrências de transições, modelando o início de qualquer atividade do sistema, sejam conforme as rotas de produção/receita dos produtos. Isto porque, o fluxo de fichas nos lugares é regulado pela ocorrência de transições em uma rede de Petri marcada e é este fluxo de fichas que define as atividades do sistema [ZD93]. O modelo analisado é a especificação do supervisor do sistema considerado.

## 1.7 Organização da Tese

Além deste capítulo a apresentação desta tese está organizada em seis capítulos.

No Capítulo 2, apresentamos com mais detalhes as redes de Petri, bem como algumas considerações sobre tolerância a faltas.

No Capítulo 3, tratamos da supervisão dos sistemas flexíveis de manufatura utilizando redes de Petri.

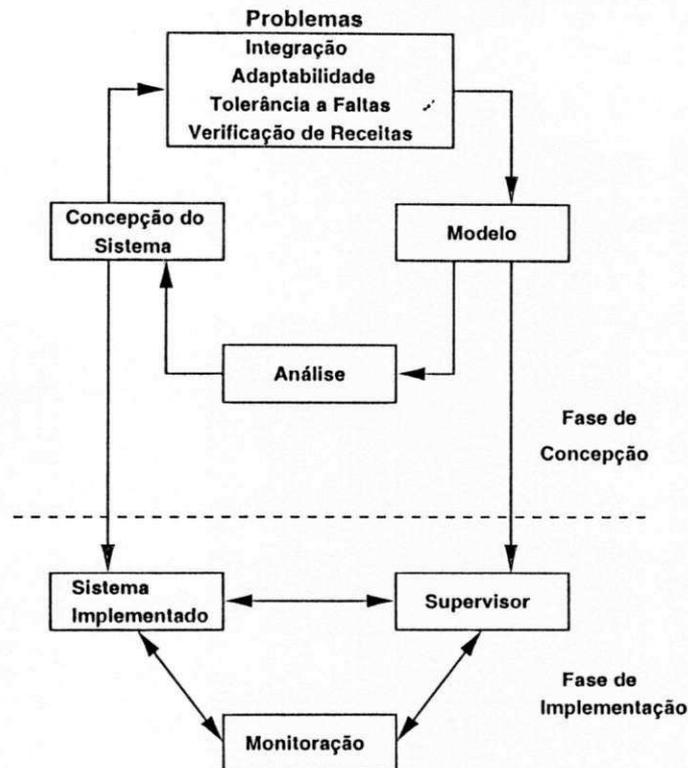


Figura 1.9: Utilização das redes de Petri nesta tese

No Capítulo 4, apresentamos, baseado nos resultados obtidos no Capítulo 3, a modelagem do supervisor para os sistemas flexíveis de manufatura tolerantes a faltas.

No Capítulo 5, tratamos da supervisão dos sistemas de produção em lotes utilizando redes de Petri.

No Capítulo 6, utilizamos as redes de Petri e as equações diferenciais para provar as limitações dos sistemas de produção em lotes.

No Capítulo 7, apresentamos um sumário das contribuições e resultados alcançados bem como perspectivas de desenvolvimento.

# Capítulo 2

## Conceitos Básicos

Este capítulo tem como objetivo apresentar o suporte teórico para o formalismo encontrado nesta tese. Por esta razão apresentamos aqui as redes de Petri e os métodos de análise conhecidos. Além disso, apresentamos também as redes de Petri Coloridas e as redes de Petri temporizadas. No final do capítulo, apresentamos alguns aspectos relacionados com tolerância a faltas.

### 2.1 Introdução às Redes de Petri

Como dito no Capítulo 1, as redes de Petri são uma ferramenta de modelagem matemática com uma representação gráfica. Estas podem ser utilizadas para descrever e estudar sistemas de processamento da informação caracterizados como sendo concorrentes, assíncronos, distribuídos, paralelos, não determinísticos e/ou estocásticos. Formalmente, uma rede de Petri [Mur89] é a quintupla  $PN = (P, T, F, W, M_0)$  onde:

1.  $P = \{p_1, p_2, \dots, p_m\}$  é um conjunto finito de lugares,
2.  $T = \{t_1, t_2, \dots, t_n\}$  é um conjunto finito de transições,
3.  $F \subseteq (P \times T) \cup (T \times P)$  é um conjunto de arcos,
4.  $W : F \rightarrow \{1, 2, 3, \dots\}$  é a função de ponderação,
5.  $M_0 : P \rightarrow \{0, 1, 2, 3, \dots\}$  é a marcação inicial,
6.  $P \cap T = \emptyset$  e  $P \cup T \neq \emptyset$ .

Uma rede de Petri é um grafo direcionado, ponderado e bipartido consistindo de dois tipos de nós, denominados lugares e transições, onde arcos podem ser direcionados de um lugar para uma transição ou de uma transição para um lugar [Pet81]. A cada arco pode ser associado um número inteiro positivo que é denominado peso ou ponderação do arco. Na representação gráfica, lugares são representados por círculos e as transições por barras ou retângulos. A marcação atribuí a cada lugar um inteiro não negativo. Se a marcação atribuí a um lugar  $p$  um inteiro não negativo  $k$ , então  $p$  é marcado com  $k$  fichas. Graficamente, estas fichas são representadas por  $k$  pontos localizados dentro do lugar  $p$ . A marcação, denotada por  $M$ , é um vetor de  $m$  componentes, onde  $m$  é o número total de lugares da rede. Denota-se  $M(p)$ , o número de fichas no lugar  $p$ .

Na modelagem, usando o conceito de condições e de eventos, lugares representam condições e transições eventos. Uma transição (evento) tem um certo número de lugares de entrada e de lugares de saída representando as pré-condições e pós-condições do evento, respectivamente. A presença de uma ficha em um lugar indica que a condição associada aquele lugar é verdadeira. Numa outra interpretação,  $k$  fichas em um lugar indicam que  $k$  recursos ou itens de dados estão disponíveis. Para simular o comportamento dinâmico de um sistema, o estado ou marcação em uma rede de Petri é modificado de acordo com as seguinte regra de transição:

1. Uma transição  $t$  é dita habilitada se cada lugar de entrada  $p$  de  $t$  é marcado com pelo menos  $w(p,t)$  fichas, sendo  $w(p,t)$  o peso do arco direcionado de  $p$  para  $t$ .
2. Uma transição habilitada pode ou não ocorrer, dependendo se o evento ocorreu.
3. A ocorrência de uma transição  $t$  retira  $w(p,t)$  fichas de cada lugar de entrada  $p$  de  $t$  e adiciona  $w(t,p)$  fichas a cada lugar de saída  $p$  de  $t$ , sendo  $w(t,p)$  o peso do arco direcionado de  $t$  para  $p$ .

Uma transição sem qualquer lugar de entrada é denominada transição fonte e uma sem qualquer lugar de saída é denominada transição poço. Observe que uma transição fonte está incondicionalmente habilitada e que a ocorrência de uma transição poço consome fichas mas não produz nenhuma. O par formado por um lugar  $p$  e uma transição  $t$  é um *auto-laço* se  $p$  é simultaneamente lugar de entrada e saída de  $t$ . Uma rede de Petri é dita

*pura* se ela não possuir nenhum *auto-laço*. Uma rede de Petri é dita *ordinária* se o peso de todos os seus arcos são iguais a 1.

Para ilustrar as regras de transição, na Figura 2.1 apresenta-se o modelo de rede de Petri para a conhecida reação química de formação da água, ou seja:  $2H_2 + O_2 \rightarrow 2H_2O$ .

Duas fichas em cada lugar de entrada, veja a Figura 2.1(a), modelam que duas moléculas de  $H_2$  e uma de  $O_2$  estão disponíveis, portanto a transição  $t$  está habilitada. Após a ocorrência de  $t$ , a marcação será modificada para aquela mostrada na Figura 2.1(b), onde, a transição  $t$  não está mais habilitada. As seguintes notações são usadas para representar os conjuntos de pré e pós-condições de uma rede de Petri, são elas:

1.  $\bullet t = \{p \mid (p, t) \in F\}$  é o conjunto de lugares de entrada de  $t$ ,
2.  $t^\bullet = \{p \mid (t, p) \in F\}$  é o conjunto de lugares de saída de  $t$ ,
3.  $\bullet p = \{t \mid (t, p) \in F\}$  é o conjunto de transições de entrada de  $p$ ,
4.  $p^\bullet = \{t \mid (p, t) \in F\}$  é o conjunto de transições de saída de  $p$ .

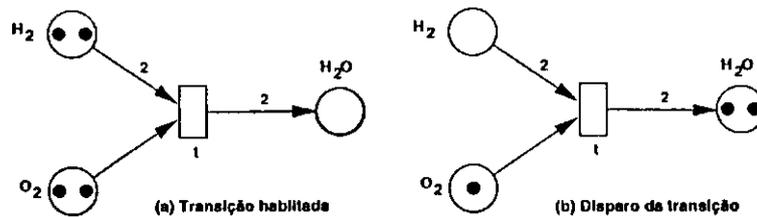


Figura 2.1: Rede de Petri utilizada para exemplificar as regras de transição

As redes de Petri têm sido utilizadas em diversas áreas. Por exemplo, por meio das redes de Petri podemos modelar atividades paralelas ou concorrentes [Hey89], protocolos de comunicação [JRMRT97; BD91], o controle de sincronização em sistemas distribuídos [EHI97; HC92; Wan96; HWL95; ZWZ95; PMK97], sistemas híbridos [CP97; PD94], sistemas tolerantes a faltas [Pro91; ZD90; LFK91; BPF97b; RV94], a base de conhecimento de sistemas especialistas [WX97; Yao94], conflitos [GGa96]. Por exemplo, ver Figura 2.2(a), as atividades paralelas ou concorrentes representadas pelas transições  $t_2$  e  $t_3$  começam após a ocorrência da transição  $t_1$  e terminam após a ocorrência da transição  $t_4$ . Portanto, as transições  $t_1$  e  $t_4$  sincronizam o início e o fim das atividades paralelas ou concorrentes representadas pelas transições  $t_2$  e  $t_3$ , respectivamente. Duas transições

são ditas concorrentes se uma transição pode ocorrer antes, em paralelo ou após a outra transição. Note que, cada lugar da rede da Figura 2.2(a) possui apenas uma transição de entrada e uma de saída. Redes de Petri com esta propriedade são denominadas grafos marcados. Esta classe de rede permite modelar apenas a sincronização e a concorrência de atividades paralelas, mas não conflitos ou tomadas de decisões [RH80; HP89; LS96; Mur89]. Dois eventos estão em conflito se a ocorrência de um evento desabilita a ocorrência do outro. Esta situação pode ser modelada naturalmente por redes de Petri. Por exemplo, conforme mostra-se na Figura 2.2(b), as transições  $t_1$  e  $t_2$  estão em conflito. Isto porque, a ocorrência da transição  $t_1$  desabilita a ocorrência da transição  $t_2$ . A causa deste conflito pode ser o compartilhamento de um recurso. Suponha que um robô é utilizado para carregar duas máquinas  $m_1$  e  $m_2$ . Note que, o robô é um recurso compartilhado pelas máquinas e portanto este somente pode carregar uma máquina de cada vez. Neste caso, as transições  $t_1$  e  $t_2$ , ilustradas na Figura 2.2(b), podem modelar a ocorrência dos eventos *iniciar carregamento da máquina  $m_1$*  e *iniciar carregamento da máquina  $m_2$* , respectivamente.

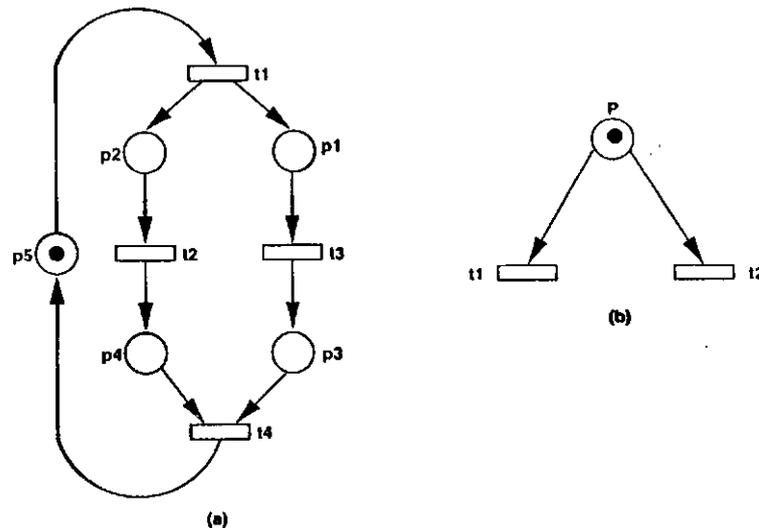


Figura 2.2: (a) Modelagem de atividades paralelas (b) Conflito ou decisão

## 2.2 Análise de Modelos de Redes de Petri

Analisando o modelo de rede de Petri de um sistema, podemos determinar se o sistema modelado apresenta ou não as propriedades de *alcançabilidade*, *limitação/segurança*, *con-*

*servação, vivacidade ou reversibilidade/recorrência* [ZZ94]. Tais propriedades são descritas a seguir:

1. *Alcançabilidade*: Uma marcação  $M_i$  é dita alcançável a partir da marcação inicial  $M_0$  se existir uma seqüência de ocorrências de transições  $\sigma$  que transforma  $M_0$  em  $M_i$ , denotado por  $M_0[\sigma]M_i$ . Denota-se por  $R(M_0)$  o conjunto de todas as marcações alcançáveis a partir de  $M_0$  e por  $L(M_0)$  ao conjunto de todas as seqüências de ocorrências de transições. Então o problema de determinar a existência de uma marcação específica  $M_i$ , consiste em determinar se  $M_i \in R(M_0)$  [IIM97].
2. *Limitação/Segurança*: A propriedade de limitação está relacionada com a capacidade de armazenamento de um sistema [Pet81]. Uma rede de Petri é dita *k-limitada* se o número de fichas em qualquer lugar  $p \in P$  é sempre menor ou igual a  $k$  para cada marcação  $M_i$  alcançável a partir de  $M_0$ , onde  $M_i \in R(M_0)$  e  $k$  é um inteiro positivo. Uma rede de Petri é dita *segura* se for 1- limitada.
3. *Conservação*: Uma rede de Petri é dita conservativa se, para qualquer marcação alcançável, a soma das fichas de cada lugar da rede, ponderada pelas componentes de um vetor  $W = (w_1, w_2, \dots, w_j, \dots, w_m)$ , é constante, onde  $m$  é o número de lugares da rede e  $w_j > 0, \forall j = 1, 2, \dots, m$ . Uma rede de Petri é dita *estritamente conservativa* se cada componente do vetor  $W$  é a unidade [ZZ94].
4. *Vivacidade*: A propriedade de vivacidade está relacionada com a ocorrência de bloqueios em um sistema [JP97; FMT97; HSBM96; FM97]. Uma transição  $t \in T$  é dita viva se para qualquer marcação  $M_i \in R(M_0)$  existir, a partir de  $M_i$ , uma seqüência de ocorrência  $\sigma$  contendo  $t$ . Uma rede de Petri é dita viva se todas as suas transições são vivas. Entretanto, na prática, esta propriedade é considerada ideal. Desta forma, diferentes níveis de vivacidade são definidos, com o objetivo de tornar este conceito o mais abrangente possível. Uma transição  $t$  de uma rede de Petri é dita:
  - $L_0$ -viva: se  $t$  não puder ocorrer em qualquer seqüência de ocorrência  $\sigma \in L(M_0)$ .
  - $L_1$ -viva: se  $t$  puder ocorrer pelo menos uma vez em alguma seqüência de ocorrência  $\sigma \in L(M_0)$ .

- $L_2$ -viva: se, dado um inteiro positivo  $k$ ,  $t$  puder ocorrer pelo menos  $k$  vezes em alguma seqüência de ocorrência  $\sigma \in L(M_0)$ .
- $L_3$ -viva: se  $t$  puder ocorrer infinitamente em alguma seqüência de ocorrência  $\sigma \in L(M_0)$ .
- $L_4$ -viva: se  $t$  for  $L_1$ -viva para cada marcação  $M \in R(M_0)$ .

Uma rede de Petri é  $L_k$ -viva se cada transição da rede for  $L_k$ -viva, isto para,  $k=0,1,2,3,4$ . Note que, conforme o conceito de vivacidade, uma rede de Petri é viva se for considerada  $L_4$ -viva. Por exemplo, a rede mostrada na Figura 2.3(a) é estritamente  $L_1$ -viva pois cada transição somente pode ocorrer, nesta seqüência  $t_2t_4t_5t_1t_3$ , apenas uma vez. Note que, as transições  $t_0, t_1, t_2, t_3$  na Figura 2.3(b) são estritamente  $L_0$ -viva (morta),  $L_1$ -viva,  $L_2$ -viva e  $L_3$ -viva, respectivamente.

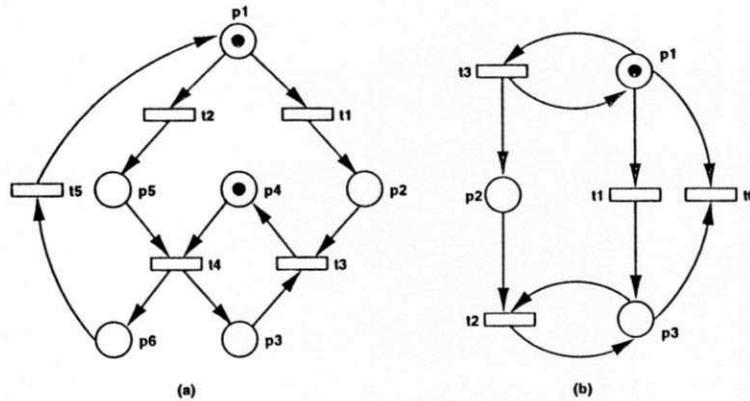


Figura 2.3: Redes utilizadas para exemplificar os níveis de vivacidade

5. *Reversibilidade/Recorrência*: Uma rede de Petri, para uma dada marcação inicial  $M_0$ , é dita reversível se para cada marcação  $M_i \in R(M_0)$ ,  $M_0$  pode ser alcançada a partir de  $M_i$ . Uma marcação  $M_j$  é um *estado recorrente* se para cada marcação  $M_i \in R(M_0)$ ,  $M_j$  é alcançável a partir de  $M_i$  [ZZ94].

Os métodos de análise estão baseados na enumeração do espaço de estado do sistema ou na utilização de equações algébricas [Zur94], fato mostrado a seguir.

### 2.2.1 Enumeração do Espaço de Estados

No que se refere a enumeração do espaço de estado, o conjunto de marcações  $R(M_0)$  de uma rede de Petri pode ser infinito ou finito. Denomina-se *árvore de cobertura*, a árvore

obtida para o caso infinito e por *árvore de alcançabilidade*, a árvore obtida para o caso finito. O *grafo de cobertura* de uma rede de Petri é o grafo orientado  $G = (V, E)$ , onde o conjunto de nós  $V$  é o conjunto de todas as marcações distintas obtidas da árvore de cobertura e  $E$  é o conjunto de arcos. Cada arco é rotulado por uma transição  $t_k$  cuja ocorrência transforma uma marcação  $M_i$  na marcação  $M_j$  onde,  $M_i$  e  $M_j \in V$ . No caso de uma rede de Petri limitada, ou seja, aquela cujo conjunto de marcações  $R(M_0)$  é finito, o grafo de cobertura é chamado de *grafo de alcançabilidade*, pois o conjunto de nós  $V$  possui todos os elementos de  $R(M_0)$ . Desde que a árvore de cobertura representa um conjunto infinito de marcações, devemos sistematizar a construção deste tipo de árvore por meio do seguinte algoritmo:

*Algoritmo de Construção da Árvore de Cobertura*

*Passo 1)* Denominar de *nova* a raiz da árvore  $M_0$ .

*Passo 2)* Enquanto existir marcações do tipo *nova* fazer o seguinte:

*Passo 2.1)* Selecionar uma marcação *nova*  $M$ .

*Passo 2.2)* Se  $M$  é idêntica a outra marcação na árvore, então denominar *velha* a marcação  $M$  e selecionar outra marcação.

*Passo 2.3)* Se nenhuma transição está habilitada em  $M$ , denominar esta marcação de *terminal*.

*Passo 2.4)* Para cada transição  $t$  habilitada na marcação  $M$  fazer o seguinte:

*Passo 2.4.1)* Obter a marcação  $M'$  que resulta da ocorrência de  $t$  em  $M$ .

*Passo 2.4.2)* Se no caminho da raiz até  $M$ , existir uma marcação  $M''$  tal que  $M'(p) \geq M''(p)$  para cada lugar  $p \in P$ , e  $M'(p) \neq M''(p)$ , então substitua  $M'(p)$  por  $\omega$  para cada lugar  $p$  onde  $M'(p) > M''(p)$ .

*Passo 2.4.3)* Introduzir a marcação  $M'$  na árvore, desenhando um arco com rótulo  $t$  de  $M$  para  $M'$  e denominar *nova* a marcação  $M'$ .

As propriedades podem ser provadas enumerando-se o espaço de estados e analisando estados alcançáveis e seqüências de eventos (ocorrências de transições). Por exemplo, aparecendo o símbolo  $\omega$  em qualquer nó da árvore, a rede é ilimitada, visto que  $\omega$  representa

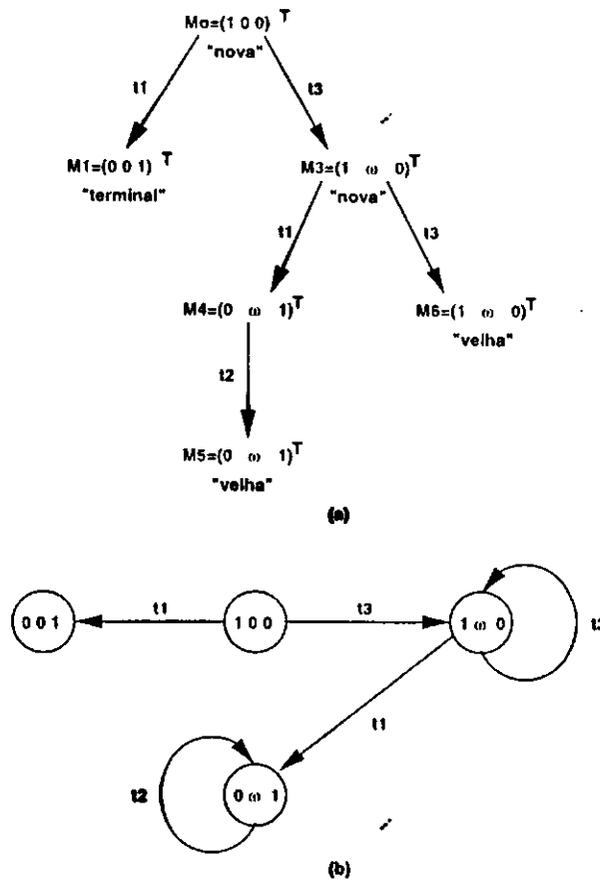


Figura 2.4: (a) Árvore de cobertura (b) Grafo de cobertura

um inteiro positivo arbitrariamente grande. Caso contrário, a rede é limitada. Se cada nó da árvore apresentar somente zeros e uns, então a rede é segura. Uma transição é dita morta caso ela não apareça como rótulo em nenhum arco da árvore. Note que, para uma rede limitada, a árvore contém todas as possíveis marcações alcançáveis a partir da marcação inicial. Para exemplificar o uso do algoritmo acima (ver Figura 2.4(a)), considere a rede mostrada na Figura 2.3(b). Na marcação inicial  $M_0 = (1, 0, 0)^T$ , as transições  $t_1$  e  $t_3$  estão habilitadas. A ocorrência de  $t_1$  transforma  $M_0 = (1, 0, 0)^T$  em  $M_1 = (0, 0, 1)^T$  que é na verdade um nó terminal da árvore, pois, nenhuma transição está habilitada nesta marcação. Ocorrendo  $t_3$  em  $M_0$  resulta em  $M'_3 = (1, 1, 0)^T$ . Note que, esta marcação cobre  $M_0$ . Portanto, a nova marcação introduzida é  $M_3 = (1, \omega, 0)^T$ , onde, as transições  $t_1$  e  $t_3$  estão novamente habilitadas. Ocorrendo  $t_1$  transforma  $M_3$  em  $M_4 = (0, \omega, 1)$ , onde,  $t_2$  pode ocorrer, resultando na marcação *velha*  $M_5 = M_4$ . Ocorrendo  $t_3$  em  $M_3$  resulta na marcação *velha*  $M_6 = M_4$ . O grafo de cobertura é mostrado na Figura 2.4(b).

### 2.2.2 Equações Algébricas

No que se refere as equações algébricas, seja uma rede de Petri formada por  $n$  transições e  $m$  lugares. A *matriz de incidência*  $A = [a_{ij}]$  é uma matriz  $n \times m$  de inteiros tal que,

$$a_{ij} = a_{ij}^+ - a_{ij}^-$$

onde,  $a_{ij}^+ = w(i, j)$  é o peso do arco direcionado da transição  $i$  para seu lugar de saída  $j$  e  $a_{ij}^- = w(j, i)$  é o peso do arco direcionado do lugar  $j$  para sua transição de saída  $i$ . Portanto, uma transição  $i$  está habilitada em uma marcação  $M$  se

$$a_{ij}^- \leq M(p_j), j = 1, 2, \dots, m$$

A equação de estado para uma rede de Petri é:

$$M_k = M_0 + A^T x$$

onde,  $x$  é um vetor coluna  $n \times 1$  formado de inteiros não negativos denominado *vetor de ocorrência* de transições. A  $i$ -ésima componente deste vetor indica o número de vezes que a transição  $i$  deve ocorrer para transformar a marcação  $M_0$  na marcação  $M_k$ . A matriz de incidência pode ser usada para o cálculo dos invariantes de lugar (P-invariantes) ou invariantes de transição (T-invariantes) de uma rede de Petri. O vetor  $y$  de inteiros é um P-invariante se  $Ay = 0$ . As componentes diferente de zero em um P-invariante representam os pesos associados com os respectivos lugares, tal que, a soma ponderada de fichas destes lugares é constante para qualquer marcação alcançável a partir da marcação inicial. O vetor  $x$  de inteiros é um T-invariante se  $A^T x = 0$ . Um T-invariante é o *vetor de ocorrência* de retorno a marcação inicial. Denota-se por suporte  $\|y\|/\|x\|$ , ao subconjunto de lugares/transições correspondente as componentes diferente de zero de um P-invariante/T-invariante, respectivamente [WTY92; BA96; BBC97; BBKC97; Pro91; Bar90].

### 2.2.3 Um Sistema Multi Robô

Para exemplificar os métodos de análise, considere sistema multi robô ilustrado na Figura 2.5. Este sistema é formado por duas máquinas  $M_1$  e  $M_2$ , dois robôs  $R_1$  e  $R_2$ , e um depósito  $B$ . O robô  $R_1$  opera descarregando a máquina  $M_1$ , ou seja, transportando subprodutos de

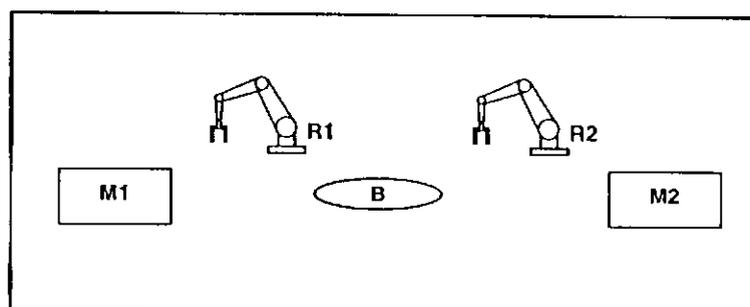


Figura 2.5: Sistema multi robô

$M_1$  para o depósito  $B$ , enquanto que, o robô  $R_2$  opera carregando  $M_2$ . O modelo para este exemplo é mostrado na Figura 2.6, onde lugares e transições possuem o seguinte significado:

Lugares:

$p_1(p_4)$ : Robô  $R_1(R_2)$  operando fora da área do depósito

$p_2(p_5)$ : Robô  $R_1(R_2)$  esperando para acessar o depósito

$p_3(p_6)$ : Robô  $R_1(R_2)$  acessando o depósito

$p_7$ : exclusão mútua

$p_8(p_9)$ : número de posições vazias(chcias) do depósito

Transições:

$t_1(t_4)$ : Robô  $R_1(R_2)$  requisita entrar na área do depósito

$t_2(t_5)$ : Robô  $R_1(R_2)$  entra na área do depósito

$t_3(t_6)$ : Robô  $R_1(R_2)$  deixa a área do depósito

Neste modelo, os lugares  $p_1, p_2, p_3$  e as transições  $t_1, t_2, t_3$  modelam o comportamento do robô  $R_1$ , enquanto que, os lugares  $p_4, p_5, p_6$  e as transições  $t_4, t_5, t_6$  modelam o comportamento do robô  $R_2$ . Note que,  $t_1$  e  $t_4$  representam atividades concorrentes de  $R_1$  e  $R_2$ , respectivamente. Conforme é ilustrado na Figura 2.5, a área do depósito  $B$  é compartilhada pelos robôs. Portanto, para evitar colisão dos robôs, o acesso ao depósito  $B$  requer sincronização, ou seja, apenas um robô pode acessar o depósito de cada vez. A sincronização assegurando exclusão mútua é modelada por meio dos lugares  $p_3, p_6, p_7$  e as transições  $t_2, t_3, t_5, t_6$ . A ocorrência de  $t_2$  desabilita  $t_5$  e vice versa. Assumimos que a

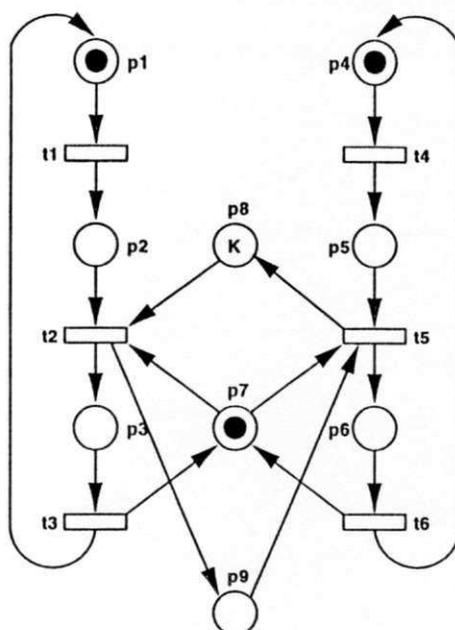


Figura 2.6: Modelo do sistema multi robô

capacidade do depósito é  $k$ , onde,  $k > 0$ . Portanto,  $k$  fichas em  $p_8$  indicam que no depósito podem ser armazenados  $k$  itens. Em contraste,  $k$  fichas em  $p_9$  indicam que o depósito está completamente cheio. Portanto, se  $p_8$  estiver sem fichas,  $t_2$  não poderá ocorrer impedindo que o robô  $R_1$  descarregue a máquina  $M_1$ . Da mesma forma,  $R_2$  não poderá carregar  $M_2$  caso o lugar  $p_9$  esteja vazio. A árvore de alcançabilidade, considerando a capacidade do depósito unitária, é mostrada na Figura 2.7.

A matriz de incidência para este exemplo é:

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

Os P-invariantes são:

$$y_1 = (1, 1, 1, 0, 0, 0, 0, 0, 0)^T \Rightarrow M(p_1) + M(p_2) + M(p_3) = 1 \quad (2.1)$$

$$y_2 = (0, 0, 0, 1, 1, 1, 0, 0, 0)^T \Rightarrow M(p_4) + M(p_5) + M(p_6) = 1 \quad (2.2)$$

$$y_3 = (0, 0, 1, 0, 0, 1, 1, 0, 0)^T \Rightarrow M(p_3) + M(p_6) + M(p_7) = 1 \quad (2.3)$$

$$y_4 = (0, 0, 0, 0, 0, 0, 0, 1, 1)^T \Rightarrow M(p_8) + M(p_9) = 1 \quad (2.4)$$

onde, os suportes para cada invariante são:

$$\|y_1\| = \{p_1, p_2, p_3\}, \|y_2\| = \{p_4, p_5, p_6\}, \|y_3\| = \{p_3, p_6, p_7\} \text{ e } \|y_4\| = \{p_8, p_9\}$$

Analisando o modelo de rede de Petri mostrado na Figura 2.6, podemos determinar que o sistema multi robô apresenta as seguintes propriedades:

1. *Limitação/Segurança*: O modelo é limitado, pois o símbolo  $\omega$  não aparece na árvore de alcançabilidade ilustrada na Figura 2.7. Além disso, a rede é *segura*, pois cada marcação alcançada a partir de  $M_0$  contém apenas zeros e uns. A propriedade de limitação pode também ser verificada a partir dos suportes dos invariantes. Isto porque, as Equações 2.1 até 2.4 garantem que a soma ponderada das fichas dos lugares de cada suporte é sempre igual a unidade e que cada lugar do modelo aparece pelo menos uma vez nos suportes dos invariantes. Por exemplo, a equação 2.1 garante que a soma ponderada das fichas nos lugares do suporte  $\|y_1\|$ , ou seja,  $p_1, p_2, p_3$  é sempre igual a unidade. Isto significa dizer que, para qualquer seqüência de ocorrência de transições, há somente uma ficha marcando qualquer um destes lugares. Análise semelhante pode ser feita para os demais suportes dos invariantes.
2. *Conservação*: O modelo para o exemplo apresentado é conservativo. Com base na árvore de alcançabilidade prova-se que o modelo é conservativo com respeito ao vetor  $W = (1, 1, 2, 1, 1, 2, 1, 1, 1)$ . Isto porque, a soma ponderada das fichas para cada marcação alcançada a partir de  $M_0$  é constante e igual a quatro. Desta forma pode-se afirmar que os recursos do sistema são conservados.
3. *Vivacidade*: Na Figura 2.8 mostra-se o grafo de alcançabilidade para o exemplo dado. Analisando o grafo, vemos que o modelo é  $L_1$  vivo, posto que para qualquer marcação alcançada a partir de  $M_0$  é sempre possível ocorrer qualquer transição dada. Isto assegura que não *bloqueios* durante a operação do sistema.
4. *Reversibilidade*: Analisando o grafo de alcançabilidade mostra-se que o modelo é reversível, visto que podemos alcançar  $M_0$  de qualquer marcação  $M \in R(M_0)$ .

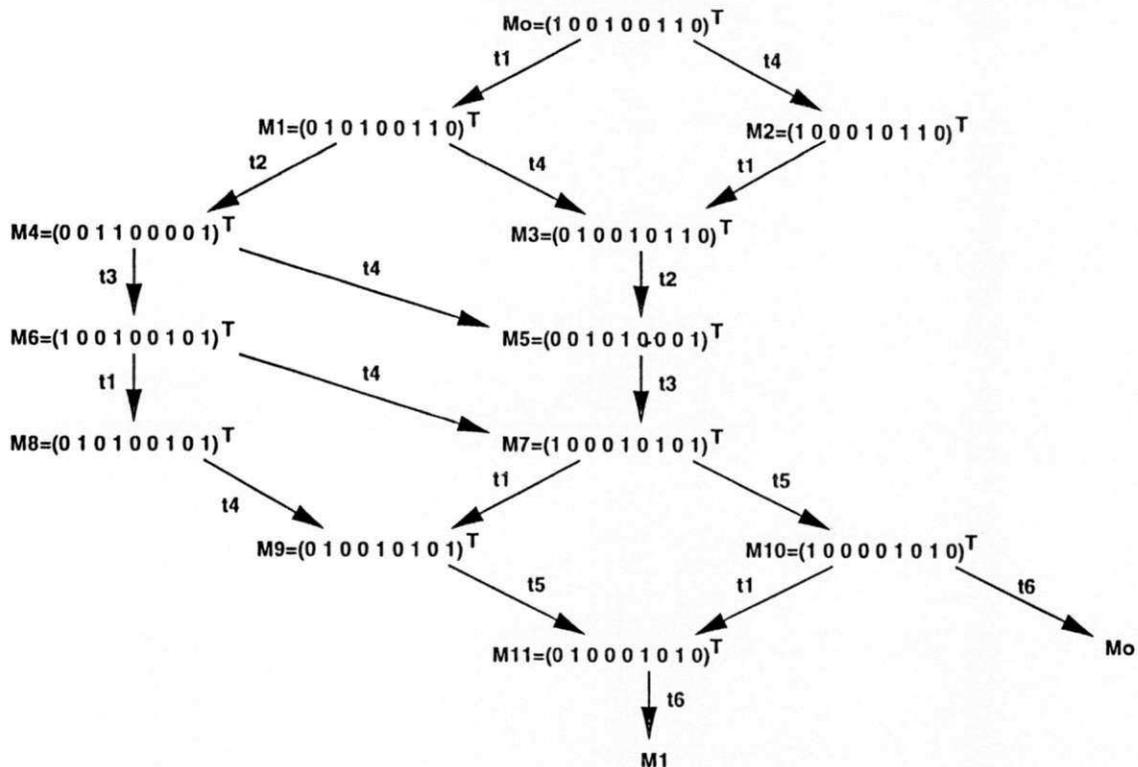


Figura 2.7: Árvore de alcançabilidade do sistema multi robô

## 2.3 Redes de Petri Temporizadas

Na definição original de redes de Petri, o conceito de tempo não é introduzido. Entretanto, para tratar problemas de escalonamento ou avaliação de desempenho é necessário introduzir o conceito de atraso de tempo associado a lugares ou transições [LS96; FM97; JJH97; KCG97; SCF94; LD94; HP89; HWL95; TWY92]. Uma rede de Petri é dita P-temporizada [SCF94; KC97a; DA92], caso o atraso esteja associado exclusivamente aos lugares da rede e T-temporizada, caso o atraso esteja associado exclusivamente às transições da rede. As redes de Petri temporizadas são ditas determinísticas, caso o atraso seja conhecido ou determinado, ou estocásticas, caso o atraso seja probabilístico. Numa rede de Petri estocástica, cada transição está associada a uma variável aleatória exponencialmente distribuída que expressa os atrasos de tempo para a ocorrência das transições. Exemplos do uso das redes estocásticas podem ser encontrados em [MBC<sup>+</sup>95; LFK91; FH97; LW95; ZBH96; HWL95]. Outra extensão temporal da redes de Petri, denominadas *redes de Petri com Temporização Nebulosa*, utiliza uma abordagem baseada na teoria dos conjuntos nebulosos. Neste modelo, as fichas carregam uma função nebulosa que indica a possibilidade

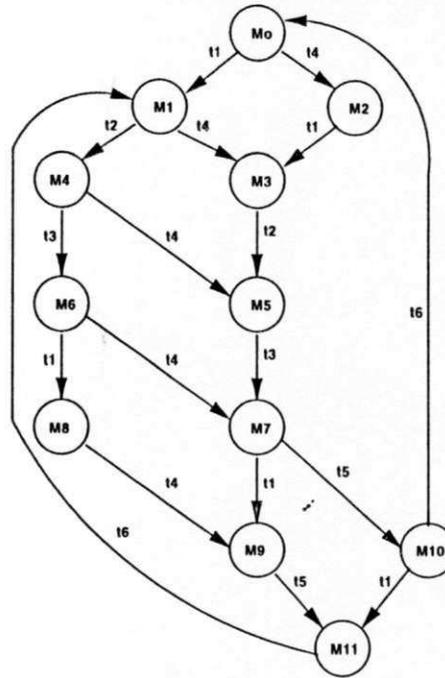


Figura 2.8: Grafo de alcançabilidade do sistema multi robô

de sua existência em um determinado lugar em um dado instante de tempo. Os intervalos nebulosos de tempo são associados com as transições, propiciando assim a representação das restrições temporais [Fig94].

Formalmente, uma rede de Petri P-temporizada determinística [DA92] é a dupla  $(PN, \delta)$ , onde,

1.  $PN$  é uma rede de Petri marcada  $(P, T, F, W, M_0)$ .
2.  $\delta : P \rightarrow [0, +\infty)$  é a função atraso.  $\delta(p_i) = d_i$ , onde  $d_i \in [0, +\infty)$  é o atraso de tempo associado ao lugar  $p_i \in P$  da rede.

Seja uma rede de Petri com  $n$  transições e  $m$  lugares. Denomina-se vetor atraso o vetor  $V_d = (d_1, d_2, \dots, d_i, \dots, d_m)^T$ , onde, cada componente  $d_i$  é igual a  $\delta(p_i)$ . Portanto, dados  $V_d$  e o modelo do sistema, podemos modelar especificações do tipo: *máquina  $M_1$  deve processar subproduto  $X$  por  $d_i$  unidades de tempo* ou *abrir válvula  $V_1$  por  $d_i$  unidades de tempo*. As regras de funcionamento das redes P-temporizadas são:

1. Toda ficha depositada em um lugar  $p_i$  torna-se indisponível por um tempo  $\tau \leq \delta(p_i)$ .
2. A ocorrência de transições segue as regras de transições das redes de Petri não temporizadas, onde o atraso de tempo das ocorrências é considerado nulo.

Para ilustrar as regras de funcionamento considere a Figura 2.9, onde, uma ficha indisponível é simbolizada por  $\circ$  e disponível por  $\bullet$ . A ocorrência de  $t_1$  adiciona uma ficha no lugar  $p_1$ , onde, permanecerá indisponível por um tempo  $d_1$ . Note que, somente após o término deste tempo a transição  $t_2$  será habilitada. A ocorrência desta adiciona uma ficha no lugar  $p_2$ , onde, permanecerá indisponível por um tempo  $d_2$ , e assim por diante. O tempo que  $t_2$  e  $t_3$  ficam habilitadas é arbitrário.

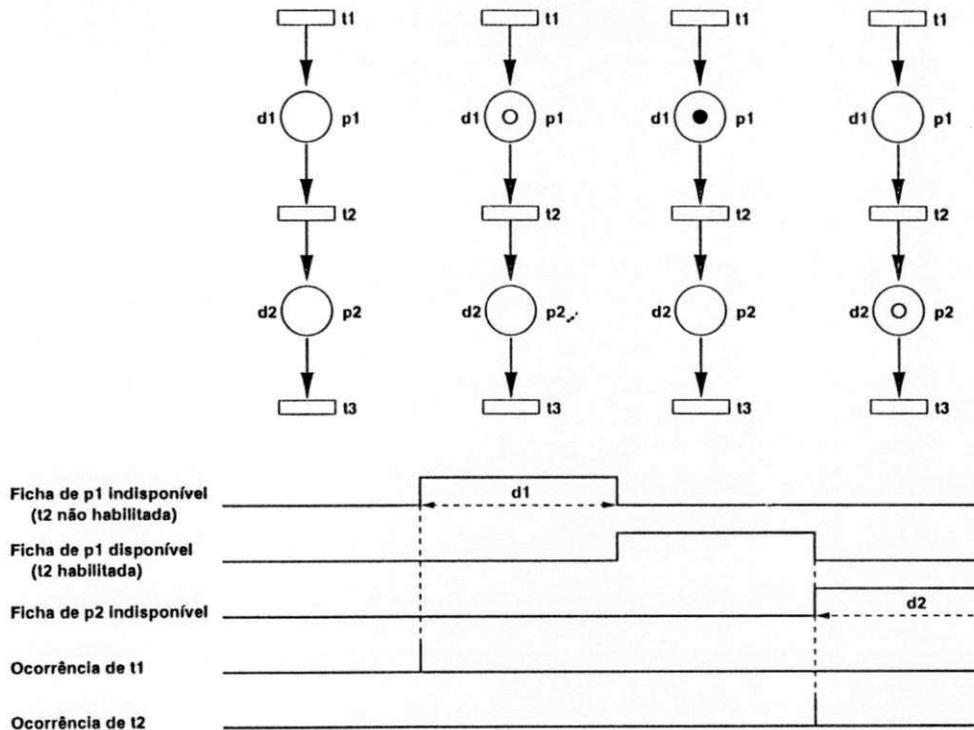


Figura 2.9: Evolução de uma rede de Petri P-temporizada

No segundo exemplo, ilustrado na Figura 2.10,  $x$  e  $y$  são inteiros positivos representando valores arbitrários de tempo. Inicialmente, a transição  $t_1$  está habilitada no intervalo  $[0, x)$ . No instante  $x$  a transição  $t_1$  ocorre adicionando uma ficha no lugar  $p_2$ , onde torna-se indisponível no intervalo  $[x, x + 3)$ , ou seja, por 3 unidades de tempo. No instante  $x + 3$ ,  $t_2$  torna-se habilitada e permanece assim até o instante  $x + 3 + y$ , ou seja, no intervalo  $[x + 3, x + 3 + y)$ . No instante  $x + 3 + y$  esta ocorre adicionando uma ficha no lugar  $p_1$ , onde, torna-se indisponível no intervalo  $[x + 3 + y, x + 3 + y + 2)$ , ou seja, por 2 unidades de tempo. No fim deste intervalo,  $t_1$  torna-se novamente habilitada, e o ciclo se repete. A Figura 2.10(b) mostra o caso de *máxima velocidade* [DA92], ou seja, neste caso  $x$  e  $y$  são nulos. Nesta situação, as transições ocorrem nos instantes em que estas tornam-se

habilitadas. A notação  $M_1[t_i/d_k]M_2$  indica que a ocorrência de  $t_1$ , com um atraso de  $d_k$  unidades de tempo, transforma a marcação  $M_1$  em  $M_2$ . Baseado nesta notação, o grafo de cobertura (operação em máxima velocidade) é mostrado na Figura 2.10(c).

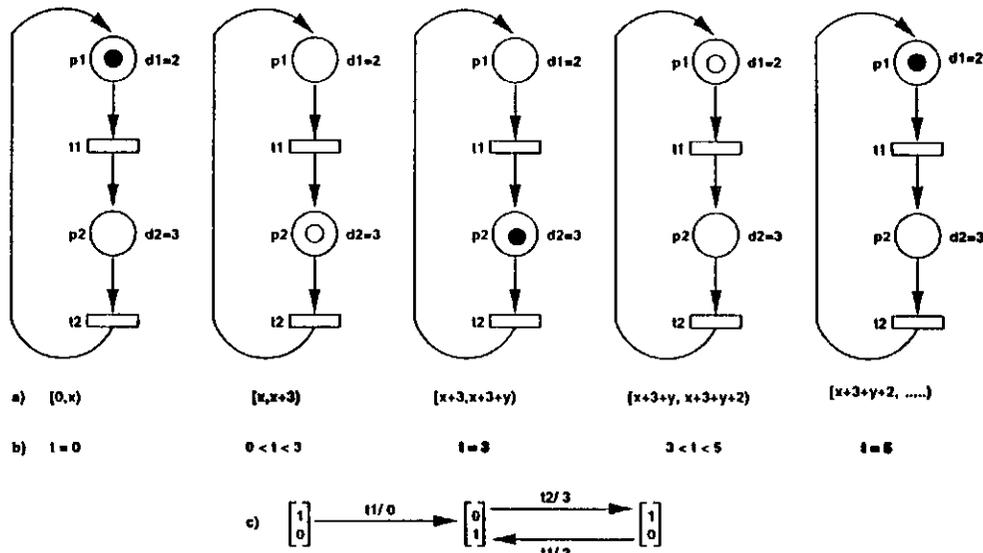


Figura 2.10: Evolução em máxima velocidade

## 2.4 Redes de Petri Coloridas

Sistemas de produção freqüentemente apresentam subsistemas similares. Neste caso, o modelo de rede de Petri de um sistema apresenta estruturas similares e que podem ser representadas por uma única estrutura de alto nível. Desta forma, é possível representar modelos de grandes dimensões utilizando-se estruturas de alto nível relativamente menores. O desenvolvimento e aplicação das redes de Petri de alto nível, tais como as redes de Petri Coloridas (CPNs) [Jen92] e as redes de Petri Predicado/Transição [BG91; GL79], constituiu um avanço significativo nesta direção. As redes de Petri de alto nível, mais especificamente as CPNs, têm o mesmo poder de descrição e análise das redes de Petri de baixo nível. Assim, todos os sistemas modelados por redes de Petri podem ser modelados por CPNs, mas de forma muito mais compacta. A representação mais compacta das CPNs é conseguida através de informações associadas às fichas, ou seja, numa CPN cada ficha possui uma cor que a diferencia das demais, indicando assim sua identidade. Formalmente, uma CPN não hierárquica [Jen92] é a tupla  $\langle \Sigma, P, T, A, N, C, G, E, I \rangle$ , onde,

1.  $\Sigma$  é um conjunto finito e não vazio de *cores*,
2.  $P$  é um conjunto finito de *lugares*,
3.  $T$  é um conjunto finito de *transições*,
4.  $A$  é um conjunto finito de *arcos* tal que  $P \cap T = P \cap A = T \cap A = \emptyset$ ,
5.  $N : A \rightarrow P \times T \cup T \times P$  é a *função nó*,
6.  $C : P \rightarrow \Sigma$  é a *função cor*,
7.  $G : T \rightarrow EXPR$  é a *função de guarda*,
8.  $E : A \rightarrow EXPR$  é a *função de arco*,
9.  $I : P \rightarrow EXPR$  é a *função de inicialização*.

onde,  $EXPR$  é um conjunto de expressões definidas no domínio do conjunto das cores. Cada componente de uma CPN tem o seguinte significado:

- (1) O conjunto de cores determina os tipos, operações e funções que podem ser usadas nas inscrições das rede, ou seja, nas expressões de arco, guardas e expressões de inicialização,
- (2)+(3)+(4) *Lugares, transições e arcos* são representados pelos conjuntos  $P, T$  e  $A$ . As redes CPNs podem apresentar estrutura vazia, ou seja,  $P \cup T \cup A = \emptyset$ . Isto permite ao usuário definir e verificar sintaticamente o conjunto de cores sem que seja necessário desenhar a rede,
- (5) A *função nó* associa cada arco a um par, onde o primeiro elemento do par é o nó fonte (lugar/transição) e o segundo o nó destino (transição/lugar),
- (6) A *função de cor*  $C$  associa cada lugar da rede a uma cor pertencente ao conjunto de cores. Isto significa dizer que, as fichas dos lugares devem possuir tipos pertencentes ao conjunto de cores,
- (7) A *função de guarda*  $G$  associa cada transição da rede a uma expressão do tipo booleana, onde, as variáveis das expressões devem possuir tipos pertencentes ao conjunto de cores. A guarda pode ser omitida se for o caso,

- (8) A *função de arco*  $E$  associa cada arco da rede a uma expressão, onde, a avaliação de uma expressão de arco é um multi conjunto sobre o conjunto de cores,
- (9) A *função de inicialização*  $I$  mapeia cada lugar da rede a uma expressão onde, cada expressão é um multi conjunto sobre o conjunto de cores.

Desta forma, uma CPN é composta essencialmente por uma estrutura, um conjunto de inscrições e um conjunto de declarações. Como as redes de Petri, as CPNs são também grafos direcionados e bipartidos. Entretanto, ao invés de pesos inteiros, aos arcos são associadas inscrições que determinam dinamicamente quantas e quais fichas devem ser removidas ou adicionadas aos lugares associados, na ocorrência de uma transição. Inscrições, denominadas guardas, podem ser também associadas às transições. Guardas restringem a ocorrência de transições a determinadas condições. O estado inicial de uma CPN também é determinado por inscrições associadas aos lugares. Cada inscrição é, em geral, uma expressão construída a partir de constantes, variáveis e operadores previamente definidos. Uma CPN também possui um conjunto de declarações para indicar a natureza dos elementos citados nas diversas inscrições, à semelhança de uma área de declarações de uma linguagem de programação qualquer.

As inscrições e declarações de uma CPN podem, *a priori*, ser escritas em praticamente qualquer linguagem com sintaxe e semântica bem definidas. Em geral, as CPNs têm sido utilizadas em associação com uma linguagem denominada CPN-ML [Jab], derivada da linguagem funcional *Standard ML* [Ull93], cuja sintaxe é bastante semelhante à usada por linguagens de programação convencionais [Güc97]. Atualmente as CPNs utilizam a linguagem *Standard ML'97* [Ull98; Jaa].

Por razões históricas, na teoria das CPNs usa-se a expressão *conjunto de cores* (*colour set*) em substituição a tipos de dados e, por conseqüência, cada valor é denominado cor (*colour*), que pode ser de um tipo arbitrário de dados (inteiro, real, lista, etc.). Desta forma, cada lugar na estrutura interna é associado a um conjunto de cores, que indica o tipo de fichas que o lugar pode conter, i.e., para um dado lugar, todas as fichas devem ter cores que pertencem a um mesmo tipo. A linguagem CPN-ML dispõe de mecanismos que permitem a definição de conjuntos de cores relativamente complexos.

*Variáveis de Transição* referem-se ao conjunto de variáveis presentes nas inscrições dos arcos e na guarda da referida transição. Uma *ligação* (*binding*) é a substituição de cada

variável da transição por uma cor (valor). É requerido, entretanto, que as cores pertençam aos conjuntos de cores apropriados e que impliquem a avaliação da guarda como verdadeira [Gue97].

Em cada marcação, a ocorrência de uma transição sob uma determinada ligação é dita habilitada se todos os seus lugares de entrada tiverem fichas suficientes para satisfazer às expressões dos arcos. Cada expressão deve ser devidamente avaliada segundo as substituições determinadas pela ligação, a fim de determinar quantas e quais fichas são requeridas nos lugares de entrada. Caso a transição ocorra, então são retiradas fichas dos lugares de entrada e depositadas novas fichas nos lugares de saída. A quantidade de fichas é determinada também pela avaliação das expressões dos arcos segundo as substituições implicadas pela ligação [Gue97].

Exemplos do uso de CPNs podem ser encontrados em [Sha91; CC97; Lin97; EC97; TBT94]. Outros detalhes referentes a semântica, técnicas de análise, bem como aplicações adicionais podem ser encontrados em [Jen92; Jen95; Jen97c; Jen98; Jen99].

### 2.4.1 Operação de uma Máquina

Para exemplificar o uso CPNs, considere a máquina ilustrada na Figura 2.11, possuindo um recurso de produção  $r$ , um depósito de entrada e outro de saída. Suponha que a máquina fornece produtos do tipo A ou B. A matéria-prima é armazenada no depósito de entrada e os produtos são armazenados no depósito de saída da máquina. O modelo de baixo nível da máquina é mostrado na Figura 2.12(a) e o modelo de alto nível (CPN) é mostrado na Figura 2.12(b). Os lugares e transições da Figura 2.12(a) possuem o seguinte significado,

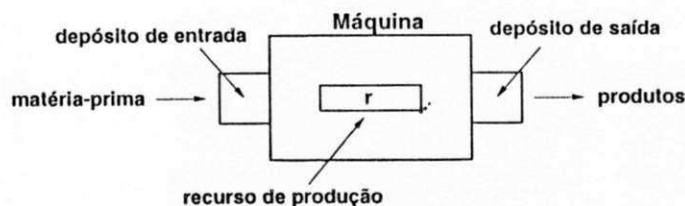


Figura 2.11: Exemplo de uma máquina

Lugares:

$p_1(p_A)$ : produto A(B) aguarda processamento,

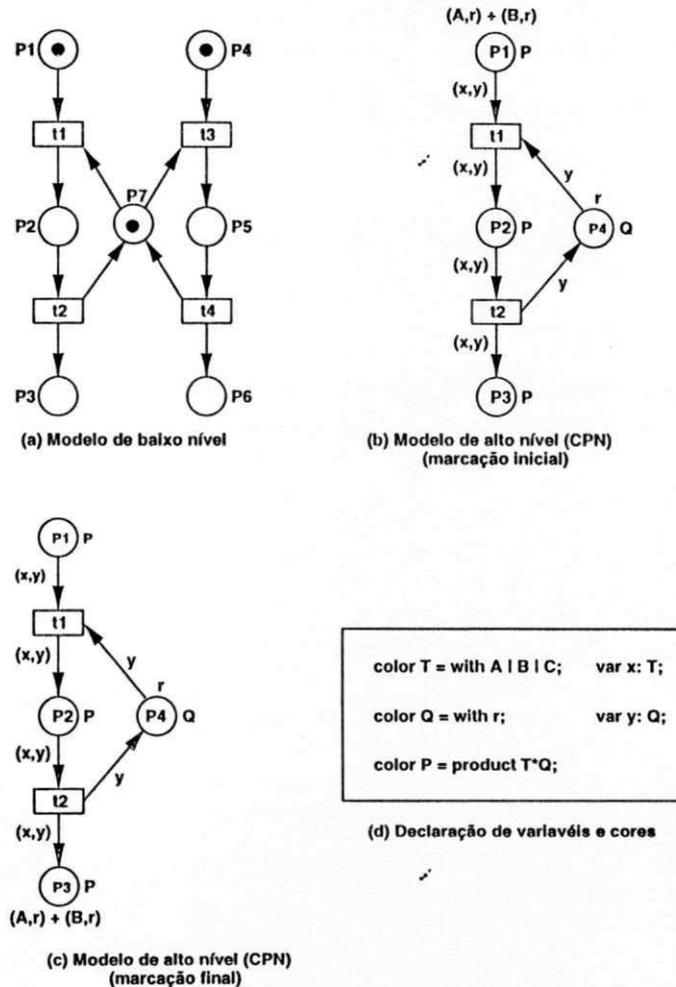


Figura 2.12: Modelos da máquina

$p_2(p_5)$ : produto A(B) está sendo processado,

$p_3(p_6)$ : produto A(B) foi processado,

$p_7$ : recurso disponível.

Transições:

$t_1(t_3)$ : início da manufatura de A(B),

$t_2(t_4)$ : término da manufatura de A(B).

Note que, a ocorrência da transição  $t_1(t_3)$  desabilita a ocorrência de  $t_3(t_1)$ , garantindo assim a exclusão mútua do recurso para cada processo de produção. Os lugares e transições da Figura 2.12(b) possuem o seguinte significado,

Lugares:

- $p_1$ : produto A ou B aguarda processamento,
- $p_2$ : produto A ou B está sendo processado,
- $p_3$ : produto A ou B foi processado,
- $p_4$ : recurso disponível.

Transições:

- $t_1$ : início da manufatura de A ou B,
- $t_2$ : término da manufatura de A ou B.

Note que, na marcação inicial ilustrada na Figura 2.12(b), duas fichas estão depositadas no lugar  $p_1$  e uma no lugar  $p_4$ . As fichas em  $p_1$  definem o tipo de produto e o recurso utilizado. Após a manufatura dos produtos, a marcação final do modelo é mostrada na Figura 2.12(c). A declaração de variáveis e cores está mostrada na Figura 2.12(d), onde  $P$  é cor dos lugares  $p_1$ ,  $p_2$  e  $p_3$  e  $Q$  é a cor do lugar  $p_4$ . A variável  $x$  é do tipo  $T$  e a variável  $y$  é do tipo  $Q$ . Com relação aos modelos apresentados na Figura 2.12, observa-se:

1. o modelo de alto nível possui menos lugares e transições do que o modelo de baixo nível. Portanto, modelos de alto nível são relativamente menores do que modelos de baixo nível, e
2. se for preciso adaptar a máquina para produzir outros tipos de produtos, não será necessário modificar a estrutura original do modelo de alto nível, então esta estrutura é preservada. O mesmo fato não ocorre com o modelo de baixo nível, visto que existe um conjunto de lugares e transições relacionados com a manufatura de cada produto.

Desta forma, baseado nestes fatos, torna-se aparente a vantagem da utilização de CPNs com relação as redes de Petri de baixo nível.

### 2.4.2 Redes de Petri Coloridas Hierárquicas

A idéia básica das redes de Petri Coloridas Hierárquicas (CPNH) é possibilitar a construção de um modelo através da combinação de um conjunto de redes relativamente menores denominadas *páginas*, de forma análoga à construção de um programa a partir de um conjunto de módulos e funções [Jen92]. O poder de modelagem de uma rede de Petri,

de uma CPN e de uma CPNH são equivalentes. É sempre possível traduzir uma CPNH para uma não hierárquica, que por sua vez pode ser traduzida para uma rede de Petri. Em termos de linguagem de programação, podemos comparar as redes de Petri com as linguagens de máquina, as CPNs com a introdução de elementos de dados estruturados na programação e as CPNHs com o uso de módulos e funções.

As CPNHs são construídas utilizando-se o conceito de *lugares de fusão e transições de substituição*. Lugares de fusão são estruturas que permitem especificar um conjunto de lugares como funcionalmente um único lugar, isto é, se uma ficha é removida ou adicionada de um dos lugares, uma ficha idêntica é adicionada ou removida de todos os outros lugares pertencentes ao conjunto. Um conjunto de lugares de fusão é denominado conjunto de fusão (*fusion set*). Uma transição de substituição pode ser vista como uma transição de mais alto nível que se relaciona a uma rede mais complexa e que fornece maiores detalhes das atividades representadas pela transição de substituição. A página que contém a transição de substituição é denominada *superpágina* e a que contém uma visão mais detalhada é denominada *subpágina*. Cada transição de substituição é denominada *supernó* da subpágina correspondente. Uma transição de substituição se relaciona com sua subpágina através da utilização de um tipo de conjunto de fusão de dois membros denominados portas e *sockets*. Estas estruturas descrevem a interface entre a transição de substituição e a subpágina. *Sockets* são atribuídos aos lugares conectados à transição de substituição, e portas são associadas a determinados lugares na subpágina tal que um par *socket/porta* forma um conjunto de fusão. Dessa forma, quando uma ficha é depositada num *socket*, ela aparece também na porta associada aquele *socket*, permitindo assim a conexão entre a superpágina e a subpágina. É sempre possível traduzir uma CPNH para sua correspondente não hierárquica. Para isso, basta substituir cada transição de substituição e arcos conectados, por sua respectiva subpágina “colando” cada *socket* com sua respectiva porta. A definição formal de CPNH pode ser encontrada em [Jen92]. Exemplo do uso de CPNHs pode ser visto em [Sha91].

## 2.5 Tolerância a Falhas

Como qualquer outro tipo de sistema, os sistemas de manufatura estão sujeitos a falhas e, dependendo da natureza do sistema, as conseqüências devido a estas falhas podem variar

de uma simples inconveniência até uma catástrofe. Desta forma, os aspectos relacionados com o grau de confiança no funcionamento correto do sistema e a capacidade de assegurar a continuidade dos serviços ganharam uma grande importância nos últimos anos. Motivados por este fato, apresentamos nesta seção alguns conceitos básicos sobre o assunto, mostrando as diferenças existentes entre os termos denominados *faltas*, *erros* e *falhas*.

Uma *falha* do sistema ocorre quando o mesmo não se comporta conforme as especificações, ou seja, um sistema falha quando este não pode prover o serviço esperado [AL81; Lap85; Lap92; Rob82]. A falha ocorre pois o sistema está em um estado de *erro*. A causa de um erro é uma *falta*. Portanto, um erro é então a manifestação de uma falta no sistema, e uma falha é o efeito de um erro no serviço esperado.

A detecção de erros em algum estado do sistema sempre implica a ocorrência de faltas, entretanto, a ocorrência de faltas não implica necessariamente a manifestação de erros. Dizemos que um sistema é tolerante a faltas se o mesmo possuir mecanismos para mascarar faltas por meio de algum tipo de redundância [Avi76; TMB80a; TMB80b]. Em outras palavras, um sistema é tolerante a faltas se o comportamento do sistema, mesmo na presença de faltas dos seus componentes, é consistente com as especificações. Para tolerar faltas, as seguintes fases são executadas:

1. *Deteção de erros*: nesta fase, os erros são detectados. A ocorrência de erros indica a presença de faltas no sistema.
2. *Recuperação de erros*: nesta fase, o erro deve ser removido do estado do sistema, isto é, o sistema deve ser levado a um estado livre de erro. Caso o erro não seja removido, este poderá se propagar levando assim o sistema a falhar. Duas técnicas podem ser usadas para recuperação de erro, elas são: (a) recuperação para frente (*forward error recovery*), ou (b) recuperação para trás (*backward error recovery*). No caso (a), durante o processamento normal os dados relativos ao estado do sistema são periodicamente armazenados em pontos de verificação (*checkpoints*). Se ocorrer uma falta no sistema após o estabelecimento do último ponto, a restauração dos dados relativos a este ponto removerá os erros detectados. Isto porque, a restauração dos dados do mais recente ponto de verificação causa o sistema retroceder a um estado livre de erro. No caso (b) não existem pontos de verificação e a remoção de erros é feita por meio de ações corretivas.

3. *Tratamento de faltas:* Esta fase é dividida em duas subfases que são: (a) localização da falta e (b) reparo do sistema. Uma vez localizada a falta, o sistema deve ser reparado. Isto pode ser feito, empregando-se um componente redundante para operar no lugar do componente defeituoso. Terminada esta operação, o sistema deve continuar operando como se nada tivesse acontecido.

Por exemplo (ver Figura 2.13(a)), suponha um simples sistema para ligar/desligar automaticamente uma lâmpada aquecedora de uma estufa. Durante o dia a estufa é aquecida pela luz do sol e a noite pela lâmpada. A chave acionadora  $E$  da lâmpada  $L_1$  possui dois estados:  $D$ (dia) e  $N$ (noite). No estado  $D$  a lâmpada deve ser apagada e no estado  $N$  ligada. Suponha que a lâmpada esteja defeituosa (filamento partido), ou seja, ocorreu uma falta no sistema. No estado  $D$  nenhum erro é detectado, isto porque a lâmpada não é ligada durante o dia. Observe que, o erro somente será detectado quando a chave mudar para o estado  $N$ . Neste caso o sistema falha, pois não comporta-se conforme o especificado. A Figura 2.13(b) mostra a estufa tolerante a faltas formada por um relé  $R$  pelas lâmpadas de aquecimento  $L_1$  e  $L_2$ , onde  $L_2$  é o componente redundante de  $L_1$ . A chave  $s$  do relé  $R$  é normalmente fechada. Não ocorrendo falha em  $L_1$ , a corrente elétrica aciona o relé  $R$  que abre a chave  $s$  desligando  $L_2$ . Esta permanecerá apagada durante todo o tempo de aquecimento da estufa. Se o filamento de  $L_1$  queimar, o relé não é acionado e portanto  $L_2$  permanecerá acesa até que  $L_1$  seja trocada.

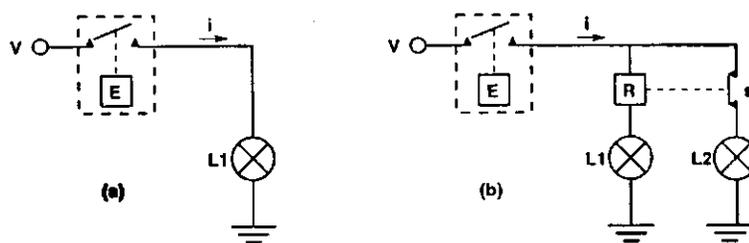


Figura 2.13: (a)Estufa (b)Estufa tolerante a faltas

## 2.6 Considerações Finais

Neste capítulo, apresentamos as redes de Petri e os métodos de análise. Através de um exemplo, mostramos as vantagens da utilização das redes de alto nível com relação as redes de baixo nível. Posto que, nesta tese tratamos com os aspectos contínuos dos sistemas de

produção, apresentamos também as redes Petri temporizadas. Além disso, apresentamos alguns conceitos básicos encontrados no estudo da tolerância a faltas.

## Capítulo 3

# Supervisão de Sistemas Flexíveis de Manufatura

Neste capítulo, tratamos da supervisão dos sistemas flexíveis de manufatura (SFMs) utilizando redes de Petri. Definimos tais sistemas em termos de seus componentes físicos e das rotas de produção [BPF98]. Apresentamos um procedimento de modelagem sistemático baseado em refinamentos sucessivos dos diferentes níveis abstração do SFM. No mais alto nível, representamos as células de manufatura e refinando este nível, representamos as máquinas de cada célula e assim por diante. Para obter o modelo do sistema, fazemos a fusão de lugares comuns aos modelos dos níveis obtidos. Tratamos da supervisão dos recursos de transporte isto, levando em conta os riscos de colisões nas proximidades dos depósitos do sistema.

### 3.1 Definição para Sistemas Flexíveis de Manufatura

Os SFMs possuem recursos de produção, recursos de transporte e recursos de armazenamento de material denominados depósitos. Um recurso de produção é a parte da máquina empregada para a manufatura da matéria-prima ou subprodutos. Os recursos de transporte, geralmente robôs e VAGs [BK93; LD94; Pet91], são empregados para o transporte de material entre os diversos depósitos do sistema. Estes depósitos são:

1. *be* e *bs*: depósitos de entrada e saída do sistema, respectivamente,
2. *bec* e *bsc*: depósitos de entrada e saída de uma célula, respectivamente,

3. *bem* e *bsm*: depósitos de entrada e saída de uma máquina, respectivamente.

Em um SFM, as máquinas podem ser agrupadas em diferentes locais, denominados *células de manufatura*. Tais sistemas, são empregados para transformar a matéria prima em produtos onde, para cada produto está associado uma rota de produção. Um SFM é definido [BPF97c] a seguir.

**Definição 3.1** Um Sistema Flexível da Manufatura é a tupla  $\langle P, RE, C, M, H \rangle$ , onde:

1.  $P$  é um conjunto finito e não vazio de produtos,
2.  $RE = Q \cup R \cup B$  é um conjunto finito e não vazio de recursos, onde,  $Q$  representa os recursos de transporte,  $R$  os recursos de produção e  $B$  os recursos de armazenamento,  $Q \cap R \cap B = \emptyset$ ,
3.  $C$  é um conjunto finito e não vazio de células,
4.  $M$  é um conjunto finito e não vazio de máquinas,
5.  $H$  é um conjunto finito e não vazio de rotas de produção.

As atividades de um SFM podem ocorrer fora ou dentro das células. Estas atividades são:

*Atividades Fora das Células:*

1. O transporte da matéria-prima do depósito de entrada do sistema para os depósitos de entrada de qualquer célula,
2. O transporte de subprodutos do depósito de saída de uma célula para o depósito de entrada de outra célula ou para o depósito de saída do sistema.

*Atividades Dentro das Células:*

1. O transporte de subprodutos do depósito de entrada da célula  $c_i$  para o depósito de entrada de uma máquina  $m_j$ ,
2. O transporte de subprodutos do depósito de saída da máquina  $m_j$  para o depósito de entrada de uma máquina  $m_k$  ou para o depósito de saída da célula  $c_i$ .
3. A manufatura de subprodutos.

onde, as máquinas  $m_j$  e  $m_k$  estão na célula  $c_i$ . Estas atividades podem ser representadas por um grafo de atividades definido a seguir.

**Definição 3.2** O grafo de atividades de um SFM é a dupla  $\langle B, A \rangle$ , onde

1.  $B = \{b_1, b_2, \dots, b_{2(1+\#(C)+\#(M))}\}$  é o conjunto de depósitos do sistema,
2.  $A \subseteq (B \times B)$  é o conjunto de arcos que representam as atividades do sistema.

onde  $\#(C)$  e  $\#(M)$  denotam as cardinalidades dos conjuntos  $C$  e  $M$  respectivamente.

### 3.1.1 Exemplo de um Sistema Flexível de Manufatura

Na Figura 3.1 mostra-se um SFM formado por quatro células  $c_1, c_2, c_3$  e  $c_4$  e oito máquinas  $m_1, m_2, \dots, m_8$ . Os recursos de produção das máquinas são  $r_1, r_2, \dots, r_8$ . Assumimos que, o sistema de transporte é formado por VAGs empregados para o transporte fora das células, e por quatro robôs utilizados para o transporte dentro das células. Para obter maiores detalhes sobre o sistema de transporte, veja a Seção 3.1.2. Neste exemplo, as rotas de produção são  $h_1, h_2$  e  $h_3$  dos produtos  $p_1, p_2$  e  $p_3$ , respectivamente.

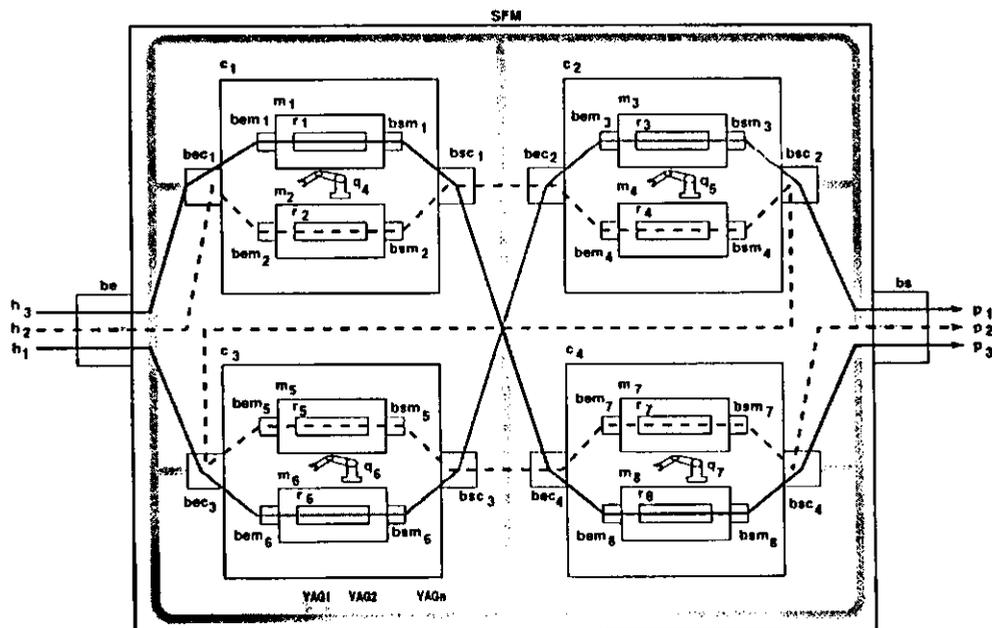


Figura 3.1: Exemplo de um SFM

O grafo de atividades para o exemplo ilustrado na Figura 3.1 é mostrado na Figura 3.2. Por exemplo, o arco  $(be, bec_1)$  representa o transporte da matéria-prima do depósito fonte

$be$  para o depósito destino  $bec_1$ , o arco  $(bec_1, bem_1)$  representa o transporte de subprodutos do depósito fonte  $bec_1$  para o depósito destino  $bem_1$ , e assim por diante. A partir da definição 3.1, o SFM da Figura 3.1 é a tupla  $\langle P, RE, C, M, H \rangle$ , onde:

1.  $P = \{p_1, p_2, p_3\}$ ,
2.  $RE = \{VAG_1, \dots, VAG_n, q_4, \dots, q_7, r_1, r_2, \dots, r_8, be, bec_1, bem_1, \dots, bs\}$ ,
3.  $C = \{c_1, c_2, c_3, c_4\}$ ,
4.  $M = \{m_1, m_2, \dots, m_8\}$ ,
5.  $H = \{h_1, h_2, h_3\}$

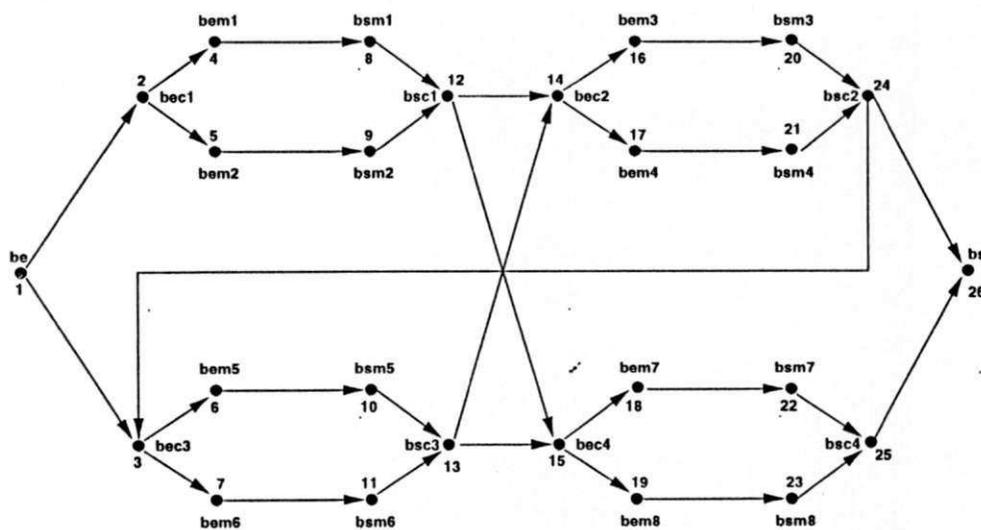


Figura 3.2: Grafo de atividades para o SFM da Figura 3.1

### 3.1.2 Sistema de Transporte

Sejam  $b_i$  e  $b_j$  depósitos fonte e destino de um SFM, respectivamente. Portanto, relativamente aos depósitos,  $b_i$  é uma área de descarregamento e  $b_j$  uma área de carregamento de material. O trecho entre estes depósitos pode ser bidirecional ou não, conforme é ilustrado na Figura 3.3. No caso bidirecional, ilustrado na Figura 3.3(a), os VAGs podem trafegar para frente e para trás dentro do mesmo trecho. No caso unidirecional ilustrado na Figura

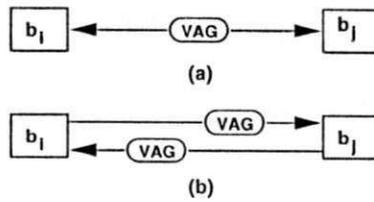


Figura 3.3: (a) VAG bidirecional (b) VAG unidirecional

3.3(b), devem existir dois trechos independentes e os VAGs podem trafegar apenas em um único sentido.

Quanto à localização, os VAGs podem ser centralizados ou distribuídos [LD94]. No esquema centralizado, o sistema possui uma única *estação de despachos de VAGs*. Esta estação é o local de partida e chegada de todos os VAGs do sistema. No esquema distribuído os VAGs possuem diversas estações de despachos localizadas próximas a locais de carregamento ou descarregamento de material. Por exemplo, o esquema distribuído considerando o sistema da Figura 3.1, está ilustrado na Figura 3.4 onde,  $E1$ ,  $E2$  e  $E3$  são as estações de despachos de VAGs. Cada estação pode possuir um ou mais VAGs dedicados ao transporte de material entre um determinado grupo de depósitos do sistema.

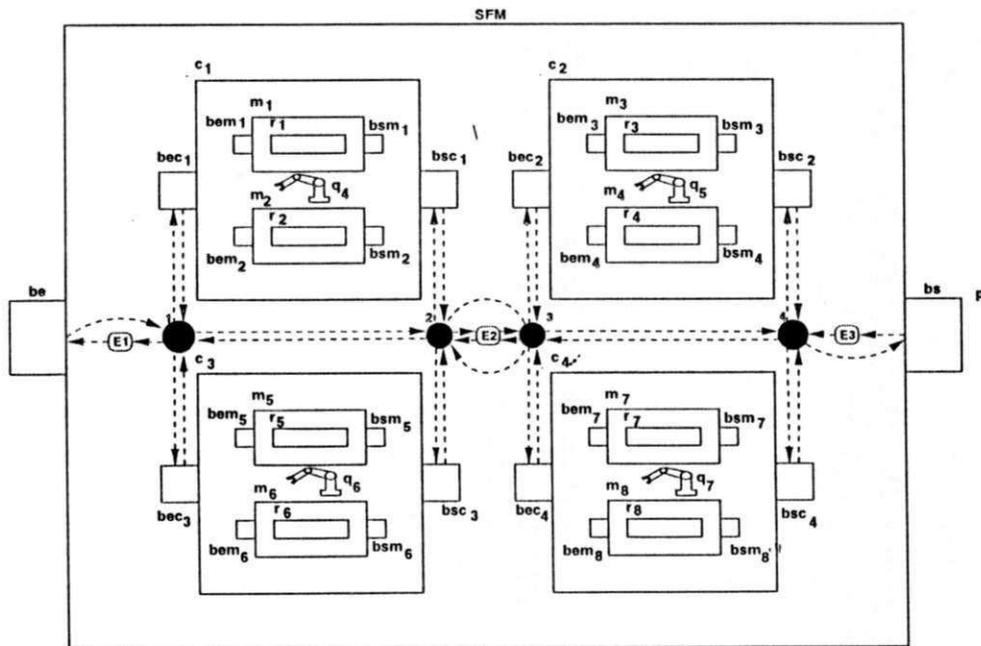


Figura 3.4: Esquema distribuído aplicado ao SFM da Figura 3.1

Observe que, um trecho pode ser compartilhado por diferentes VAGs executando diferentes atividades. Portanto, para evitar colisão entre VAGs, apenas um VAG pode ocupar

um trecho de cada vez. Se o trecho é muito longo, este pode ser dividido em subtrechos, onde, cada subtrecho pode ser ocupado apenas por um VAG. Nas proximidades de um depósito podem ocorrer colisões e danos ao equipamento. Por exemplo, o depósito  $b_{c_1}$  ilustrado na Figura 3.4 pode ser carregado por VAGs do trecho  $(1, b_{c_1})$  e descarregado pelo robô  $q_4$  da célula  $c_1$ . Portanto, para evitar colisões entre VAGs e robôs, o tráfego também nas proximidades dos depósitos deve ser controlado. O transporte entre um depósito fonte  $b_i$  e um depósito destino  $b_j$  pode ser feito em quatro fases distintas denominadas de *fases de transporte* [BP99]. Estas fases são:

*Fase A:* Esta fase inicia quando um VAG (robô), situado fora da área do depósito  $b_j$ , é despachado com destino ao depósito  $b_i$ . Esta fase termina quando o mesmo entra na área do depósito  $b_i$ .

*Fase B:* Nesta fase, o VAG (robô) descarrega o depósito  $b_i$ . Esta fase inicia quando o VAG (robô) entra na área do depósito  $b_i$  e termina quando o mesmo deixa a área deste depósito.

*Fase C:* Esta fase inicia quando um VAG (robô), situado fora da área do depósito  $b_i$ , é despachado com destino ao depósito  $b_j$ . Esta fase termina quando o mesmo entra na área do depósito  $b_j$ .

*Fase D:* Nesta fase, o VAG (robô) carrega o depósito  $b_j$ . Esta fase inicia quando o VAG (robô) entra na área do depósito  $b_j$  e termina quando o mesmo deixa a área deste depósito.

Observe que, o término de uma fase é seguido do começo da fase seguinte. Após o término da fase *D*, a fase *A* poderá ser iniciada novamente. Para exemplificar as diferentes fases de transporte, considere a Figura 3.5. O  $VAG_1$  descarrega o depósito  $b_i$  e carrega o depósito  $b_j$  e o  $VAG_2$  descarrega o depósito  $b_j$ . Na Figura 3.5, um arco direcionado acima da representação de um VAG indica que o mesmo está em movimento. Os círculos tracejados representam os limites da área de cada depósito. As Figuras 3.5(a), (b) e (c) ilustram as fases de transporte *A*, *B* e *C*, do  $VAG_1$ , respectivamente. Na Figura 3.5(d), o  $VAG_1$  está aguardando a saída do  $VAG_2$  da área do depósito  $b_j$  para iniciar a sua fase de transporte seguinte. Na Figura 3.5(e), o  $VAG_1$  está executando a fase *D*, visto que, a área do depósito  $b_j$  está livre.

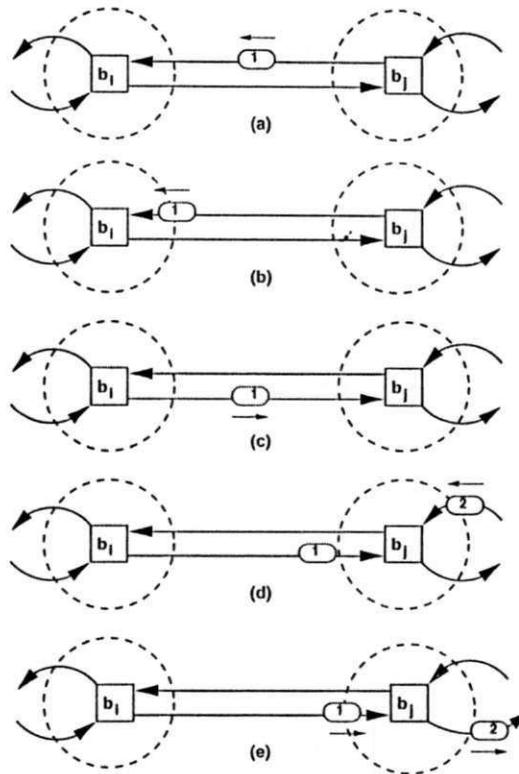


Figura 3.5: Fases de transporte

## 3.2 Especificação do Supervisor

Quanto maior é a integração de um sistema de produção, mais difícil é a descrição do sistema como um todo. Para facilitar esta descrição, podemos refinar sucessivamente os níveis de abstração do sistema partindo do nível mais alto até alcançar níveis mais elementares. Este procedimento, denominado *top-down*, é geral e pode ser aplicado aos sistemas de produção. Por exemplo, um SFM pode ser visto em três níveis de abstração ilustrados na Figura 3.6. No *nível 1* estão as células de manufatura, no *nível 2* estão as máquinas que compõem cada célula e no *nível 3* estão os recursos de produção de cada máquina do sistema. Entretanto, outros procedimentos podem ser utilizados como o do tipo composição modular, denominada *bottom-up*, e do tipo híbrido que combina o procedimento *top-down* e o *bottom-up*. A discussão e as vantagens de utilização de cada um destes métodos pode ser encontrada em [ZD93].

O procedimento de modelagem introduzido nesta tese baseia-se na construção dos modelos de cada nível de abstração, denominados *modelo das atividades fora das células*, *modelo das atividades dentro das células* e *modelo dos recursos de produção*, relativos aos

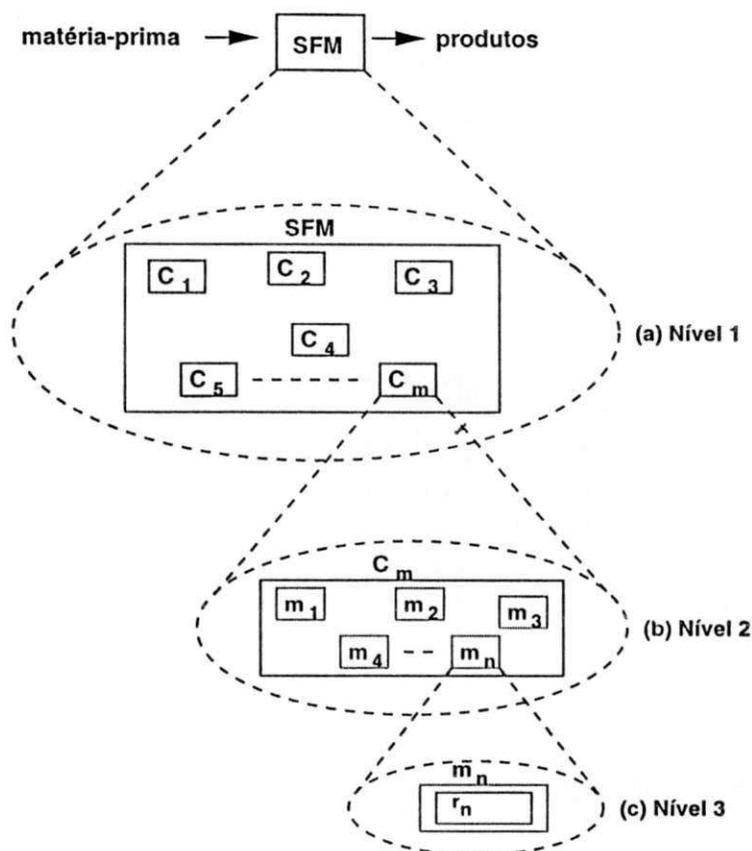


Figura 3.6: Refinamento dos níveis de abstrações de um SFM

níveis 1, 2 e 3, respectivamente. O modelo do sistema é construído fazendo-se a fusão de lugares comuns (lugares com mesmo nome) aos modelos de cada nível. A construção segue os seguintes passos:

1. Elaborar os modelos das atividades fora das células, das atividades dentro das células e dos recursos de produção,
2. Fazer a fusão de lugares do modelo das atividades foras das células com os lugares do modelo das atividades dentro das células,
3. Fazer a fusão de lugares do modelo obtido no passo anterior com os lugares do modelo dos recursos de produção, obtendo assim o modelo do sistema considerado.

Note que, em cada passo o modelo é refinado, ou seja, apresenta mais detalhes sobre o sistema. Para exemplificar este procedimento de modelagem, considere o exemplo da Figura 3.1. Os modelos para este exemplo são descritos a seguir:

- O modelo das atividades fora das células está apresentado na Figura 3.7. Neste modelo, lugares e transições são os seguintes:

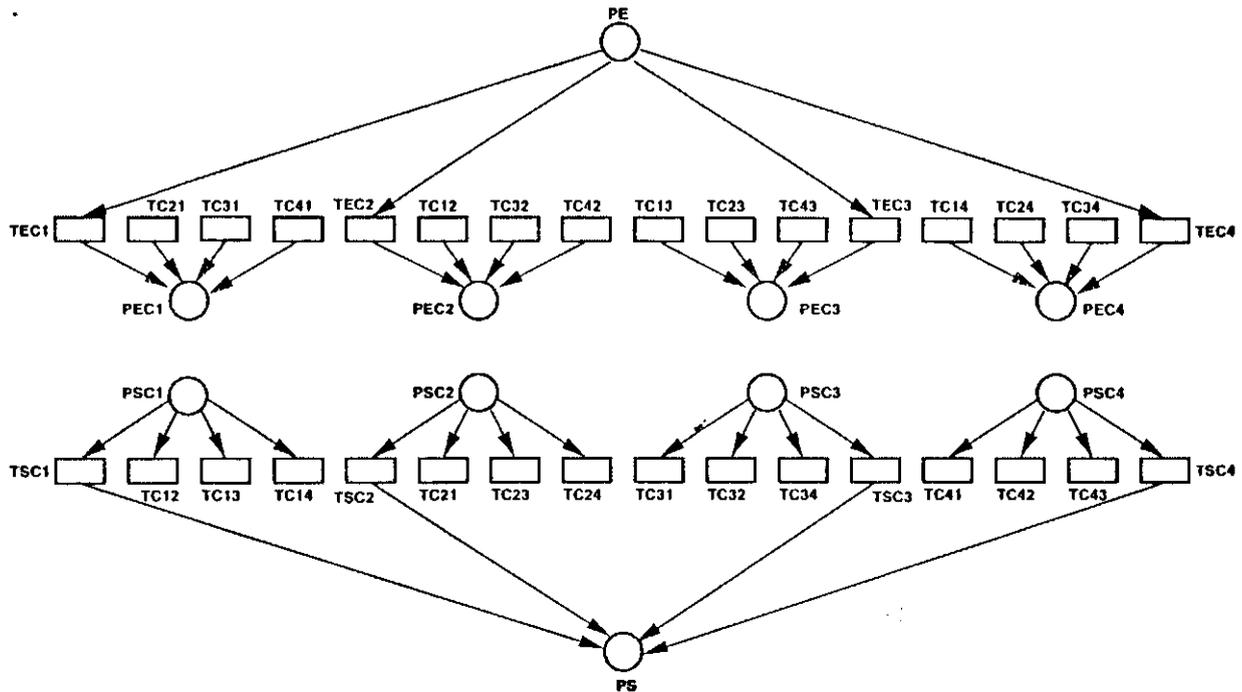


Figura 3.7: Modelo das Atividades Fora das Células

Lugares:

*PE*: modela ordens de serviço requisitando o transporte da matéria-prima do depósito de entrada do sistema para o depósito de entrada de uma célula  $c_i$ ,

*PS*: modela ordens de serviço executadas.

*PEC<sub>i</sub>*: modela ordens de serviço requisitando o transporte de subprodutos do depósito de entrada de uma célula  $c_i$  para o depósito de entrada de uma máquina dentro da célula,

*PSC<sub>i</sub>*: modela ordens de serviço requisitando o transporte de subprodutos do depósito de saída de uma célula  $c_i$  para o depósito de entrada de outra célula  $c_j$  ou para o depósito de saída do sistema,

Transições:

$TEC_i$ : modela o transporte da matéria-prima do depósito de entrada do sistema para o depósito de entrada da célula  $c_i$  onde,  $\bullet(TEC_i) = \{PE\}$  e  $(TEC_i)^\bullet = \{PEC_i\}$

$TSC_i$ : modela o transporte de subprodutos do depósito de saída da célula  $c_i$  para o depósito de saída do sistema onde,  $\bullet(TSC_i) = \{PSC_i\}$  e  $(TSC_i)^\bullet = \{PS\}$

$TC_{ij}$ : modela o transporte de subprodutos do depósito de saída de uma célula  $c_i$  para o depósito de entrada de uma célula  $c_j$  onde,  $\bullet(TC_{ij}) = \{PSC_i\}$  e  $(TC_{ij})^\bullet = \{PEC_j\}$

- O modelo das atividades dentro das células está apresentado na Figura 3.8. Neste modelo, lugares e transições são os seguintes:

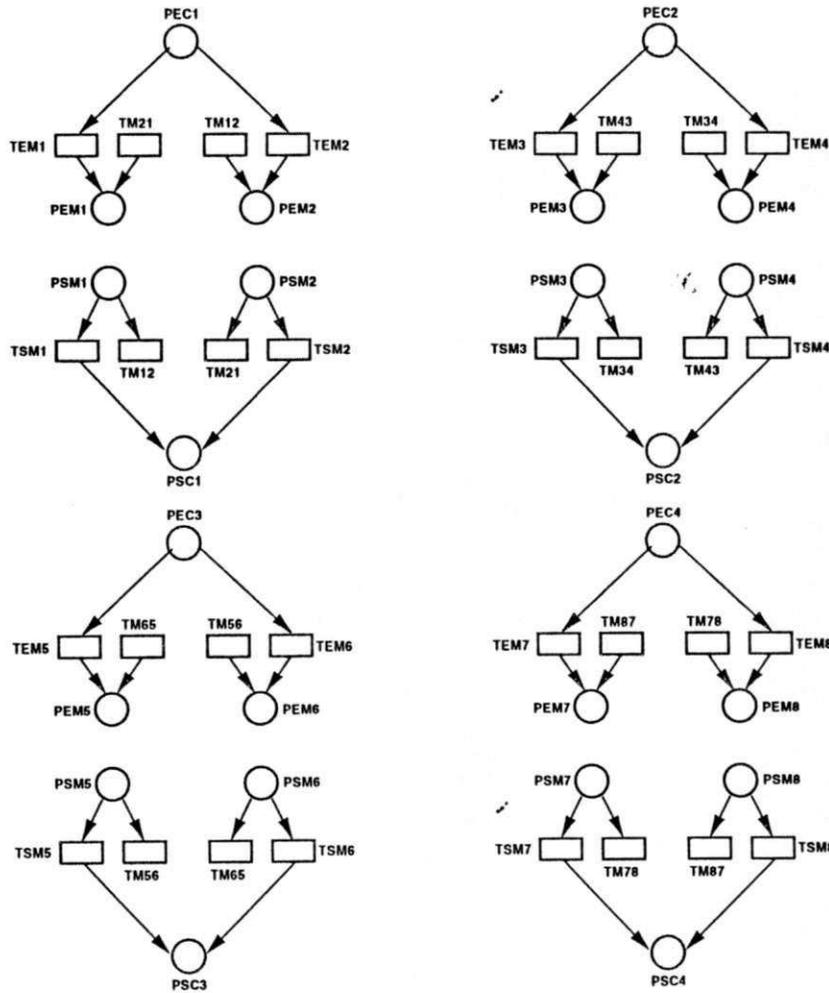


Figura 3.8: Modelo das Atividades Dentro das Células

Lugares:

$PEC_i$ : modela ordens de serviço requisitando o transporte de subprodutos do depósito de entrada de uma célula  $c_i$  para o depósito de entrada de uma máquina dentro da célula,

$PSC_i$ : modela ordens de serviço requisitando o transporte de subprodutos do depósito de saída de uma célula  $c_i$  para o depósito de entrada de outra célula  $c_j$  ou para o depósito de saída do sistema,

$PEM_j$ : modela ordens de serviço requisitando a manufatura de subprodutos contidos no depósito de entrada de uma máquina  $m_j$ ,

$PSM_j$ : modela ordens de serviço requisitando o transporte de subprodutos do depósito de saída de uma máquina  $m_j$  para o depósito de entrada de outra máquina  $m_k$  dentro de uma célula  $c_i$  ou para o depósito de saída da célula,

Transições:

$TEM_j$ : modela o transporte de subprodutos do depósito de entrada da célula  $c_i$  para o depósito de entrada de uma máquina  $m_j$  onde,  $\bullet(TEM_j) = \{PEC_i\}$  e  $(TEM_j)^\bullet = \{PEM_j\}$

$TSM_j$ : modela o transporte de subprodutos do depósito de saída da máquina  $m_j$  para o depósito de saída da célula  $c_i$  onde,  $\bullet(TSM_j) = \{PSM_j\}$  e  $(TSM_j)^\bullet = \{PSC_i\}$

$TM_{jk}$ : modela o transporte de subprodutos do depósito de saída da máquina  $m_j$  para o depósito de entrada de uma máquina  $m_k$  onde,  $\bullet(TM_{jk}) = \{PSM_j\}$  e  $(TM_{jk})^\bullet = \{PEM_k\}$

onde, as máquinas  $m_j$  e  $m_k$  estão na célula  $c_i$ .

- O modelo dos recursos de produção está apresentado na Figura 3.9. Neste modelo, lugares e transições são o seguinte:

Lugares:

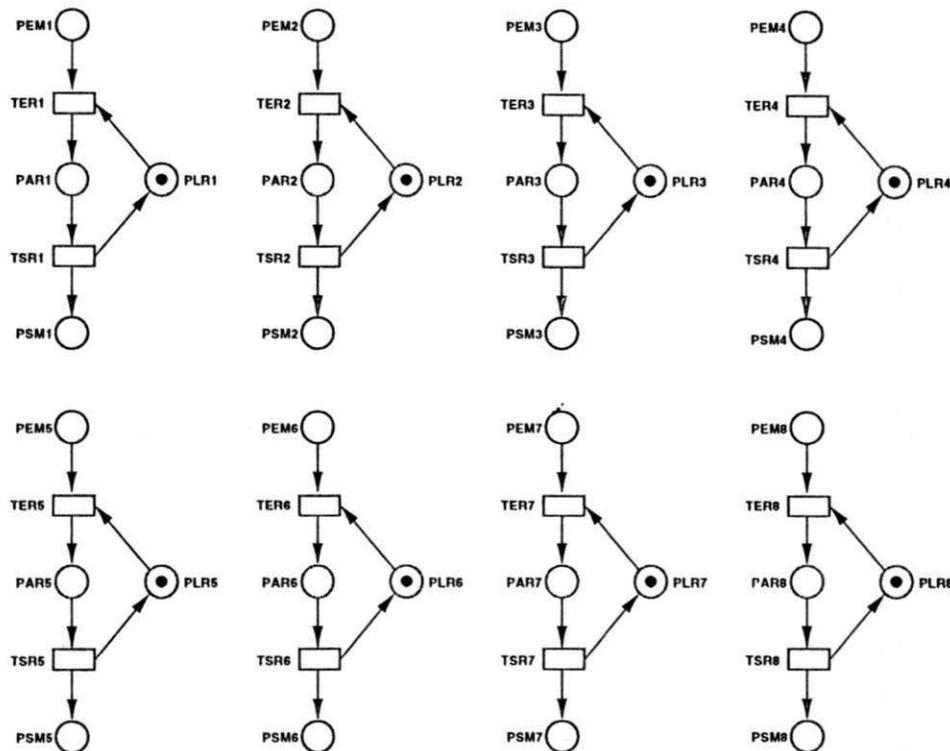


Figura 3.9: Modelo dos Recursos de Produção

$PEM_j$ : modela ordens de serviço requisitando a manufatura de subprodutos contidos no depósito de entrada de uma máquina  $m_j$ ,

$PSM_j$ : modela ordens de serviço requisitando o transporte de subprodutos do depósito de saída de uma máquina  $m_j$  para o depósito de entrada de outra máquina  $m_k$  dentro de uma célula  $c_i$  ou para o depósito de saída da célula,

$PAR_j$  e  $PLR_j$ : modela um recurso de produção  $r_j$  ocupado ou livre, respectivamente.

Transições:

$TER_j$ : modela recurso de produção  $r_j$  inicia processamento onde,  $\bullet(TER_j) = \{PEM_j, PLR_j\}$  e  $(TER_j)^\bullet = \{PAR_j\}$

$TSR_j$ : modela recurso de produção  $r_j$  termina processamento onde,  $\bullet(TSR_j) = \{PAR_j\}$  e  $(TSR_j)^\bullet = \{PSM_j, PLR_j\}$

O modelo do SFM considerado está apresentado na Figura 3.10. Nesta seção apresentamos um procedimento de modelagem sistemático para os SFMs. Note que, devido

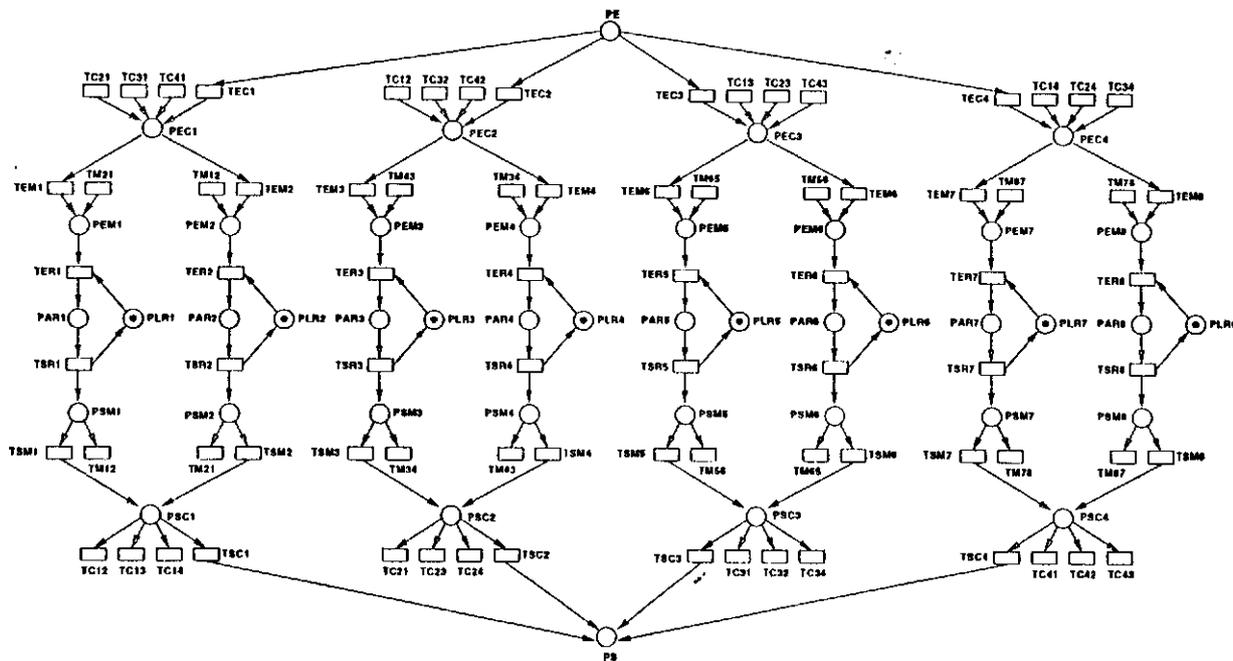


Figura 3.10: Modelo do SFM apresentado na Figura 3.1

a este procedimento, os modelos são formados por estruturas idênticas e que podem ser representadas pela estrutura de alto nível ilustrada na Figura 3.11 onde:

1. O lugar  $PE$  desta estrutura representa o lugar  $PE$  do modelo global obtido (ver Figura 3.10), o lugar  $PEC$  representa todos os lugares  $PEC_i$  e assim por diante,
2. A transição  $TEC$  desta estrutura representa todas as transições  $TEC_i$  do modelo global obtido (ver Figura 3.10), a transição  $TEM$  representa todas as transições  $TEM_i$  e assim por diante,
3. As transições  $TEC, TPC$  e  $TSC$  estão relacionadas com o transporte fora das células e, as transições  $TEM, TPM$  e  $TSM$  estão relacionadas com o transporte dentro das células do sistema.

Esta estrutura de alto nível pode ser utilizada para modelar qualquer SFM definido no contexto desta tese e resolver conflitos por meio das informações contidas nas fichas. No caso do modelo apresentado na Figura 3.10 as transições  $TEC_1, TEC_2, TEC_3$  e  $TEC_4$  estão em conflito. Para resolver este conflito, no caso deste modelo seria necessário adicionar lugares de controle nas entradas destas transições. Estes lugares receberiam fichas, depositadas por uma outra estrutura de controle, de modo a decidir a ocorrência

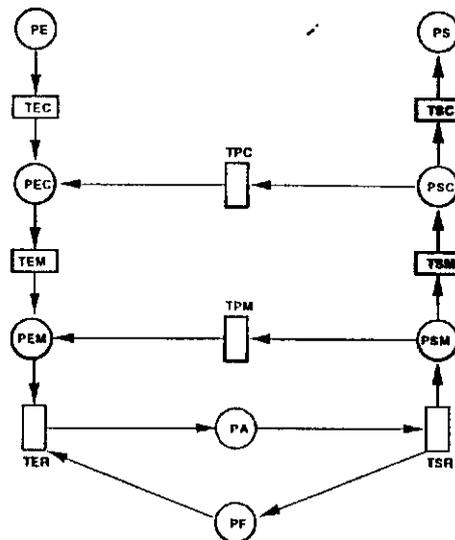


Figura 3.11: Estrutura de alto nível

destas transições. Como mostraremos nas seções seguintes, o uso de CPN, juntamente com informações de controle nas fichas, possibilita resolver tais conflitos de forma mais simples e eficiente, pois não será necessário definir uma outra estrutura para a tomada de decisões, tornando o modelo mais compacto. Além disso, a solução para a especificação do supervisor deve atender aos requisitos de agilidade inerentes aos sistemas de produção, logo, como também será introduzido, a solução com CPN irá satisfazer tais requisitos, enquanto a especificação apresentada na Figura 3.10 obviamente não satisfaz. Entretanto, tal solução foi apresentada com o intuito de enfatizar o procedimento sistemático e estruturado que norteou a obtenção do modelo em CPN a partir do procedimento apresentado nesta seção.

### 3.3 Especificação do Supervisor em Redes de Petri Coloridas

Para exemplificar a utilização da estrutura de alto nível, especificamos o supervisor do sistema apresentado na Figura 3.1. Para construir e analisar o modelo utilizamos a ferramenta Design/CPN, que possibilita editar e analisar modelos CPN [Jab; Jaa]. O modelo foi construído segundo a hierarquia de páginas<sup>1</sup> ilustrada na Figura 3.12. Este é formado por sete páginas onde, a primeira página, denominada *fluxo de ordens*

<sup>1</sup>O conceito de páginas de um modelo CPN hierárquico está apresentado na Seção 2.4.2.

*de serviço*, é utilizada para o controle do fluxo de ordens de serviço segundo as rota de produção do sistema. As demais páginas são empregadas para a supervisão do transporte entre os diversos depósitos do sistema. Estas páginas são descritas a seguir:

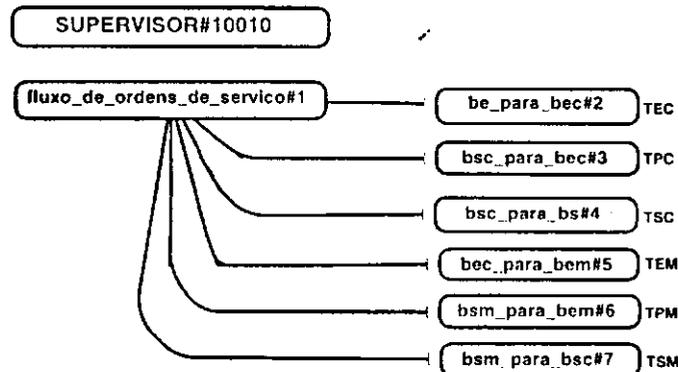


Figura 3.12: Páginas do supervisor

### 3.3.1 Descrição da Página *fluxo de ordens de serviço*

Na descrição das páginas a seguir, as seguintes operações com listas são utilizadas. Sejam as listas  $l_1 = [\alpha_1 \alpha_2 \dots \alpha_m]$  e  $l_2 = [\beta_1 \beta_2 \dots \beta_n]$ . Uma lista pode ser dividida em duas partes, denominadas cabeça e cauda. Por exemplo,  $\alpha_1$  é a cabeça de  $l_1$  e  $\alpha_2 \dots \alpha_m$  é a cauda da lista. Na expressão  $l_1 = (x :: List)$ , as variáveis  $x$  e  $List$  representam a cabeça e a cauda da lista, respectivamente.

A página *fluxo de ordens de serviço*, apresentada na Figura 3.13, possui a mesma estrutura apresentada na Figura 3.11 onde, lugares e transições possuem o seguinte significado:

Lugares:

*PE*: modela ordens de serviço requisitando o transporte da matéria-prima do depósito de entrada do sistema para o depósito de entrada de uma célula,

*PSC*: modela ordens de serviço requisitando o transporte de subprodutos do depósito de saída de uma célula para o depósito de entrada de outra célula ou para o depósito de saída do sistema,

*PS*: modela ordens de serviço executadas.

*PEC*: modela ordens de serviço requisitando o transporte de subprodutos do depósito de entrada de uma célula para o depósito de entrada de uma máquina dentro da célula,

*PEM*: modela ordens de serviço requisitando a manufatura de subprodutos contidos no depósito de entrada das máquinas,

*PSM*: modela ordens de serviço requisitando o transporte de subprodutos do depósito de saída de uma máquina para o depósito de entrada de outra máquina dentro de uma célula ou para o depósito de saída da célula,

*PR*: modela um recurso de produção ocupado,

*PF*: modela um recurso de produção livre.

Transições:

*TEC*: modela o transporte de subprodutos do depósito *be* para os depósitos *bec*,

*TSC*: modela o transporte de subprodutos dos depósitos *bse* para o depósito *bs*,

*TEM*: modela o transporte de subprodutos dos depósitos *bec* para os depósitos *bem*,

*TSM*: modela o transporte de subprodutos dos depósitos *bsm* para os depósitos *bse*,

*TPC*: modela o transporte de subprodutos dos depósitos *bse* para os depósitos *bec*,

*TPM*: modela o transporte de subprodutos dos depósitos *bsm* para os depósitos *bem*,

*TER*: modela o início de uma atividade de produção,

*TSR*: modela o término de uma atividade de produção.

Na Figura 3.13, mostra-se a marcação inicial desta página. Note que, temos oito fichas no lugar *PF*, ou seja,  $r_1 + r_2 + \dots + r_8$  indicando que existem oito recursos de produção disponíveis. O lugar *PE* pode conter um número arbitrário de fichas, ou seja, ordens de serviço  $(p_i, [lista_i])$  onde, *lista<sub>i</sub>* é a lista de todas as *requisições de serviço* empregadas para a manufatura do produto  $p_i$ . Estas *requisições de serviço* indicam que recursos devem ser alocados e os depósitos utilizados. Para o exemplo mostrado na Figura 3.1, consideramos duas ordens de serviço, uma para a produção do produto  $p_2$  e outra para a produção do produto  $p_3$ , ou seja,  $(p_2, [lista_2]) + (p_3, [lista_3])$ . Para construir as listas (*lista<sub>2</sub>* e *lista<sub>3</sub>*), assumimos que o sistema de transporte possui três VAGs  $q_1$ ,  $q_2$  e  $q_3$  pertencentes as estações de despachos *E1*, *E2* e *E3*, respectivamente, e quatro robôs  $q_4$ ,  $q_5$ ,  $q_6$  e  $q_7$ , onde:

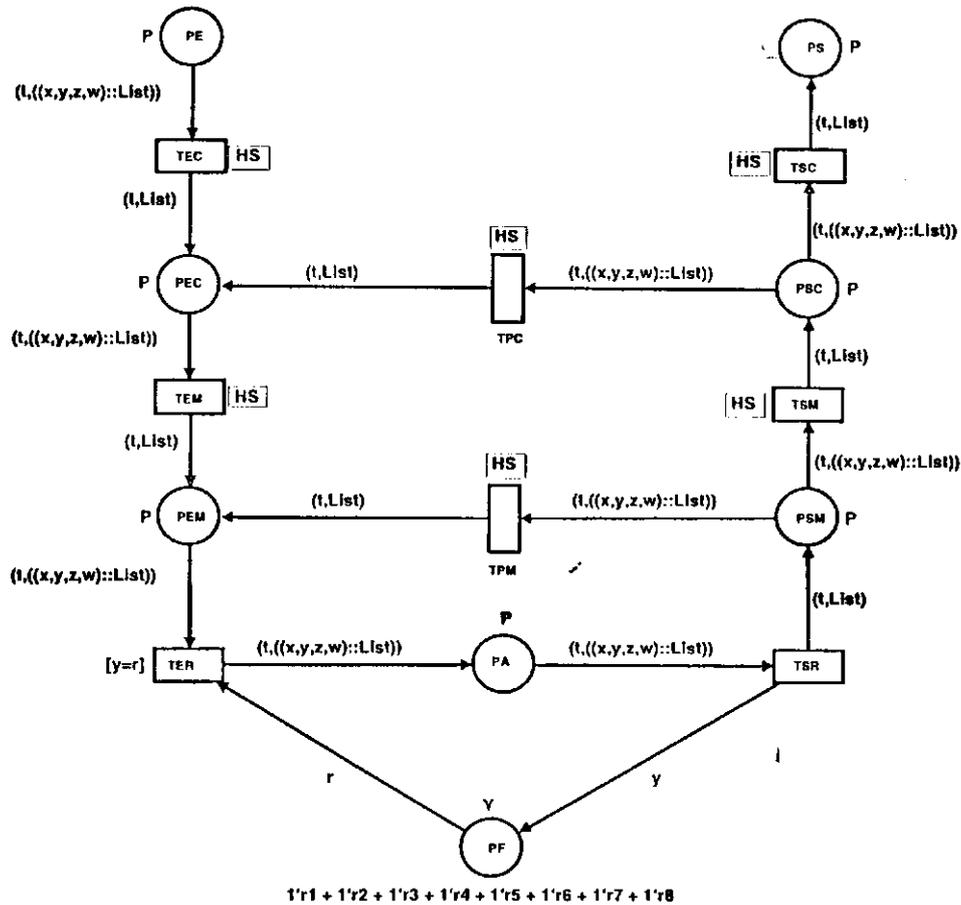


Figura 3.13: Página fluxo de ordens de serviço

1. o VAG  $q_1$  é dedicado ao transporte da matéria-prima do depósito de entrada do sistema para os depósitos de entrada das células,
2. o VAG  $q_2$  é dedicado ao transporte de subprodutos dos depósitos de saída das células para os depósitos de entrada de outras células,
3. o VAG  $q_3$  é dedicado ao transporte de subprodutos dos depósitos de saída das células para para o depósito de saída do sistema, e
4. os robôs  $q_4$ ,  $q_5$ ,  $q_6$  e  $q_7$  são empregados para o transporte dentro da células  $c_1$ ,  $c_2$ ,  $c_3$  e  $c_4$ , respectivamente.

As listas foram construídas utilizando-se o grafo de atividades mostrado na Figura 3.2. Note que, para cada depósito representado no grafo, associamos arbitrariamente um número inteiro positivo. Isto porque, na declaração de variáveis do modelo, as variáveis que representam os depósitos são números inteiros. Desta forma, o arco (1,2) corresponde

ao arco  $(be, bec_1)$  e assim por diante. Por exemplo, a  $lista_3$  é formada por uma seqüência de nove requisições de serviço ( $rs$ ), ou seja,

$$lista_3 = [rs_1, rs_2, rs_3, rs_4, rs_5, rs_6, rs_7, rs_8, rs_9]$$

onde,

$$rs_1 = (0, q_1, 1, 2), rs_2 = (0, q_4, 2, 4), rs_3 = (0, r_1, 4, 8)$$

$$rs_4 = (2, q_4, 8, 12), rs_5 = (3, q_2, 12, 15), rs_6 = (0, q_7, 15, 19)$$

$$rs_7 = (0, r_8, 19, 23), rs_8 = (2, q_7, 23, 25), rs_9 = (4, q_3, 25, 26,)$$

A primeira requisição de serviço  $rs_1 = (0, q_1, 1, 2)$  especifica que o VAG  $q_1$  deve ser alocado para transportar a matéria-prima do depósito  $be$  (1) de entrada do sistema para o depósito  $bec_1$  (2) de entrada da célula  $c_1$ . O segunda requisição de serviço  $rs_2 = (0, q_4, 2, 4)$  especifica que o robô  $q_4$  deve ser alocado para transportar subprodutos do depósito  $bec_1$  (2) para o depósito  $bem_1$  (4) de entrada da máquina  $m_1$  e assim por diante. Da mesma forma, pode ser construída a  $lista_2$  ou qualquer número de listas desde que, sejam baseadas nas rotas de produção do sistema.

As expressões de arco desta página são do tipo  $(t, ((x, y, z, w) :: List))$  ou  $(t, List)$ , onde  $t$  especifica o produto a ser manufaturado,  $x$  usada para resolver conflitos,  $y$  especifica um recurso,  $z$  especifica um depósito fonte,  $w$  especifica um depósito destino e  $List$  especifica a cauda da lista. Note que, devido a expressão de arco  $(t, List)$ , a cabeça da lista  $(x, y, z, w)$  é removida, após a ocorrência de algumas transições. Isto deve ser feito pois, requisições de serviços atendidas, devem ser removidas da lista. Desta forma, o lugar  $PS$  deve conter fichas  $(p_i, [])_i$ , ou seja, deve conter ordens de serviço executadas do produto  $p_i$  onde,  $[]$  representa a lista vazia.

Note que, as transições  $TPM$  e  $TSM$  estão em conflito. Este conflito surge porque subprodutos contidos no depósito de saída de uma máquina podem ser transportados para outra máquina ou para a saída da célula que contém estas máquinas. Este conflito é resolvido através da expressão de guarda  $[x = 1]$  associada a  $TPM$  e pela expressão de guarda  $[x = 2]$  associada a  $TSM$ . Se  $x = 1$  então  $TPM$  ocorre senão,  $TSM$  ocorre. Da mesma forma, as transições  $TPC$  e  $TSC$  estão em conflito. Este conflito surge porque

subprodutos contidos no depósito de saída de uma célula podem ser transportados para a entrada de outra célula ou para a saída do sistema. Este conflito é resolvido através da expressão de guarda [ $x = 3$ ] associada a *TPC* e pela expressão de guarda [ $x = 4$ ] associada a *TSC*. Se  $x = 3$  então *TPC* ocorre, caso contrário, *TSC* ocorre. Por exemplo, a variável  $x$  assume o valor quatro na última requisição de serviço da *lista*<sub>3</sub>. Isto inibe a ocorrência da transição *TPC*, portanto habilitando a transição *TSC*. A ocorrência desta aloca o VAG  $q_3$  para transportar subprodutos do depósito *bsc*<sub>4</sub> (25) de saída da célula  $c_4$  para o depósito *bs* (26) de saída do sistema.

### 3.3.2 Descrição da Página *be para bec*

A página *be para bec*, ilustrada na Figura 3.14, é utilizada para a supervisão do transporte entre os depósitos de entrada do sistema *be* e das células *bec*. Nesta página, lugares e transições possuem o seguinte significado:

Lugares:

1. *PE* e *PEC*: ver descrição feita na página *fluxo de ordens de serviço*,
2. *PA2*, *PB2*, *PC2* e *PD2*: indicam que fase de transporte *A*, *B*, *C* ou *D* está sendo executada, respectivamente.
3. *PBE* e *PBEC*: indicam que a área dos depósitos de entrada do sistema ou das células estão livres, respectivamente.
4. *PRTE*: modela a disponibilidade dos recursos de transporte fora das células.

Transições:

1. *TA2*, *TB2*, *TC2* e *TD2*: a ocorrência destas transições indicam início da fase de transporte *A*, *B*, *C* ou *D*, respectivamente.
2. *TE2*: a ocorrência desta transição indica término da fase de transporte *D*, respectivamente.

Note que, a marcação inicial do lugar *PRTE* é  $q_1 + q_2 + q_3$ . A escolha de apenas um VAG em cada estação não é um fator limitante na especificação do supervisor desde que, podemos marcar o lugar *PRTE* com um número arbitrário de fichas sem que seja necessário

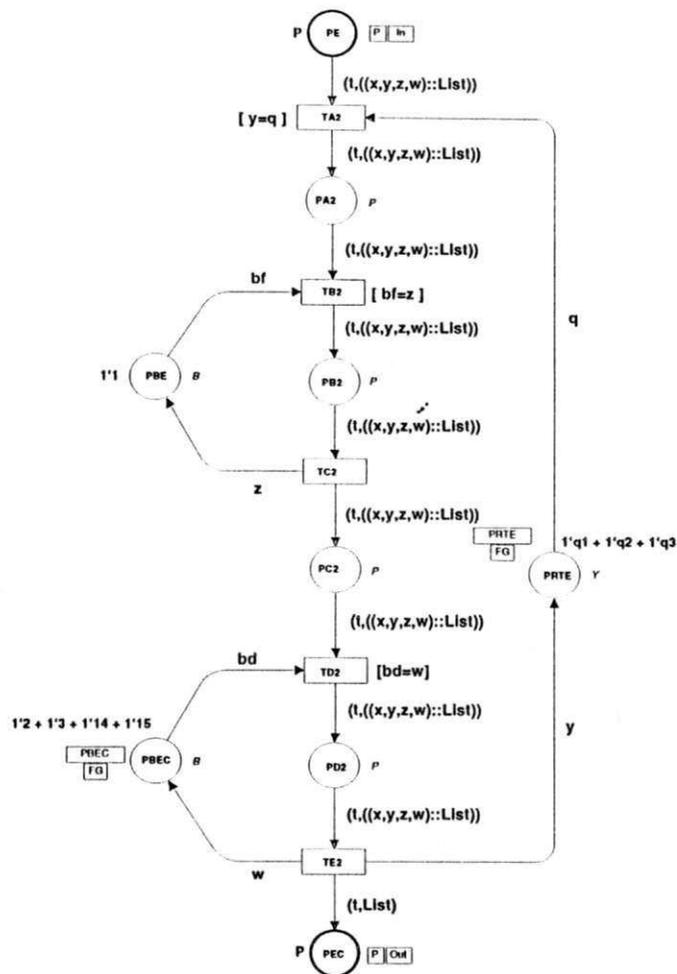


Figura 3.14: Página *be para bec*

redesenhar o modelo obtido. As demais páginas possuem a mesma estrutura da página *be para bec* portanto, a ilustração destas páginas é omitida. A declaração de variáveis e cores do modelo é mostrada na Figura 3.15.

### 3.4 Funcionalidade e Análise do Supervisor

Para alocar e gerenciar concorrentemente recursos compartilhados, o supervisor interage com os gerenciadores locais do sistema enviando mensagens requisitando serviços e recebendo mensagens que informam o estado dos recursos<sup>2</sup>. Estas mensagens são:

1. Mensagens enviadas pelo supervisor:

<sup>2</sup>A funcionalidade do supervisor está baseada nos conceitos apresentados na Seção 1.3

```

color T = with p2lp3;
color B = int;
color X = int;
color Y = with r1r2r3r4r5r6r7r8lq1lq2lq3lq4lq5lq6lq7;
color Z = int;
color W = int;
color D = product X*Y*Z*W;
color E = list D;
color P = product T*E;

var t: T;
var x: X;
var y,r,q: Y;
var z,bf,w,bd: B;
var List: E;

```

Figura 3.15: Declaração de variáveis e cores

- (a) Mensagens para inicializar os recursos de produção: sincronizadas com a ocorrência da transição  $TER$ ,
- (b) Mensagens para inicializar a fase de transporte  $A$ : sincronizadas com a ocorrência de transições do tipo  $TA_i$ ,
- (c) Mensagens para inicializar a fase de transporte  $B$ : sincronizadas com a ocorrência de transições do tipo  $TB_i$ ,
- (d) Mensagens para inicializar a fase de transporte  $D$ : sincronizadas com a ocorrência de transições do tipo  $TD_i$ .

## 2. Mensagens recebidas pelo supervisor:

- (a) Mensagens informando o término de uma atividade de produção: a ocorrência da transição  $TSR$  está sincronizado com o recebimento deste tipo de mensagem,
- (b) Mensagens informando o término da fase de transporte  $A$ ,
- (c) Mensagens informando o término da fase de transporte  $B$ : a ocorrência de transições do tipo  $TC_i$  está sincronizado com o recebimento deste tipo de mensagem,
- (d) Mensagens informando o término da fase de transporte  $C$ ,
- (e) Mensagens informando o término da fase de transporte  $D$ : a ocorrência de transições do tipo  $TE_i$  está sincronizado com o recebimento deste tipo de mensagem.

para  $i = 2, 3, \dots, 7$ . Por exemplo, quando a transição  $TER$  ocorre, uma requisição de serviço, contida no lugar  $PEM$ , é enviada a um recurso de produção do sistema. Após

o envio desta mensagem, o supervisor fica aguardando o recebimento da mensagem de serviço executado. Recebida esta mensagem, a transição *TSR* então ocorre, liberando assim o recurso para a próxima alocação. Para ilustrar, considere que a rota de produção  $h_2$ , exibida em linha tracejada na Figura 3.1, corresponde a manufatura de porcas. Conforme mostra a Figura 3.16, a produção possui cinco estágios ou etapas de manufatura descritos a seguir:

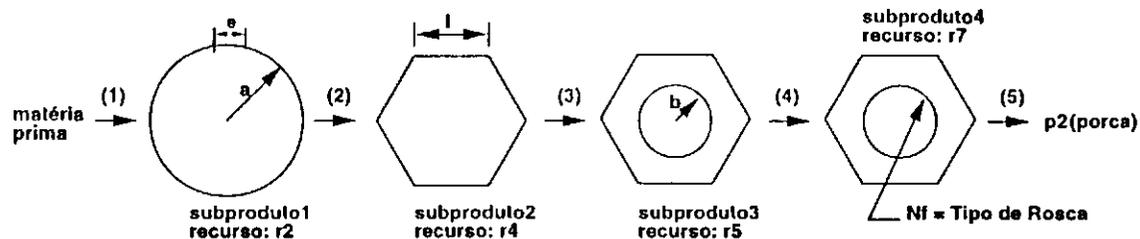


Figura 3.16: Manufatura de porcas

1. A matéria-prima é levada ao recurso  $r_2$  dando origem ao *subproduto*<sub>1</sub>, cuja especificação é: (*Raio Externo da Porca* =  $a$ , *Largura da Porca* =  $e$ );
2. O *subproduto*<sub>1</sub> é levado ao recurso  $r_4$  dando origem ao *subproduto*<sub>2</sub>, cuja especificação é: (*Comprimento da Face da Porca* =  $l$ );
3. O *subproduto*<sub>2</sub> é levado ao recurso  $r_5$  dando origem ao *subproduto*<sub>3</sub>, cuja especificação é: (*Raio Interno da Porca* =  $b$ );
4. O *subproduto*<sub>3</sub> é levado ao recurso  $r_7$  dando origem ao *subproduto*<sub>4</sub>, cuja especificação é: (*Tipo de Rosca da Porca* =  $N_f$ );
5. O *subproduto*<sub>4</sub> é levado para a saída do sistema.

Por exemplo, suponha que a matéria-prima está no depósito de entrada da máquina  $m_2$ . Neste caso, o supervisor envia (ocorrência da transição *TER*) para o gerenciador da máquina, uma requisição de serviço solicitando a manufatura da porca. Após receber esta mensagem, a máquina inicia a transformação da matéria-prima no *subproduto*<sub>1</sub>. Após o término do processamento, uma mensagem de requisição de serviço executado é enviada ao supervisor. Após o receber esta mensagem a transição *TSR* então ocorre indicando que o recurso de produção  $r_2$  está livre.

Da mesma forma, quando uma transição  $TA_i$  ocorre, uma requisição de serviço é enviada a um VAG (robô) para executar a fase de transporte  $A$  entre dois depósitos do sistema (depósitos fonte e destino). Após o envio desta mensagem, o supervisor fica aguardando a mensagem de término da fase  $A$ . Recebida esta mensagem, a transição  $TB_i$  somente poderá ocorrer caso a área do depósito fonte esteja livre. Devido a ocorrência da transição  $TB_i$ , o VAG é então liberado para a execução da fase de transporte  $B$ . Após o recebimento da mensagem de término da fase de transporte  $B$ , a transição  $TC_i$  então ocorre, indicando que o VAG iniciou a fase de transporte  $C$  onde, o supervisor fica aguardando a mensagem de término desta fase. Recebida esta mensagem, a transição  $TD_i$  somente poderá ocorrer caso a área do depósito destino esteja livre. Devido a ocorrência da transição  $TD_i$ , o VAG é então liberado para a execução da fase de transporte  $D$ . Após o recebimento da mensagem de término da fase de transporte  $D$ , a transição  $TE_i$  então ocorre, indicando que o VAG está livre para a próxima alocação, ou seja, iniciar outra vez a fase de transporte  $A$ .

Como anteriormente, utilizamos a ferramenta *Design/CPN* para editar e analisar o modelo apresentado. Com base nos resultados da análise, mostra-se que o grafo de ocorrência<sup>3</sup> possui 1552 estados, 1300 arcos e que foi construído completamente dentro de 11 segundos onde, o estado de número 1552, apresentado na Figura 3.17, é um estado de bloqueio.

Investigando a marcação do lugar  $PS$  no estado de bloqueio, encontramos  $(p_2, []) + (p_3, [])$ , isto é, todas as ordens de serviço foram executadas. Note que, o estado de bloqueio é alcançado mediante a ocorrência da transição  $TE_4$  da página *bsc para bs* isto para  $(t = p_3, x = 4, y = q_3, z = 25, w = 26, List = [])$  ou  $(t = p_2, x = 4, y = q_3, z = 24, w = 26, List = [])$ . Além disso, as transições  $TA_6$ ,  $TB_6$ ,  $TC_6$ ,  $TD_6$  e  $TE_6$  da página *bsm para bem* não ocorreram. Isto porque, conforme as rotas de produção  $h_2$  e  $h_3$  ilustradas na Figura 3.1, não existe atividades de transporte entre os depósitos de saída e entrada das máquinas do sistema.

<sup>3</sup>O grafo de ocorrência para uma CPN é similar a grafo de alcançabilidade para as redes de baixo nível. A principal diferença é que os arcos relacionando marcações não correspondem a ocorrência das transições, mas sim às ligações da transição, o leitor interessado em detalhes pode consultar [Jen92].

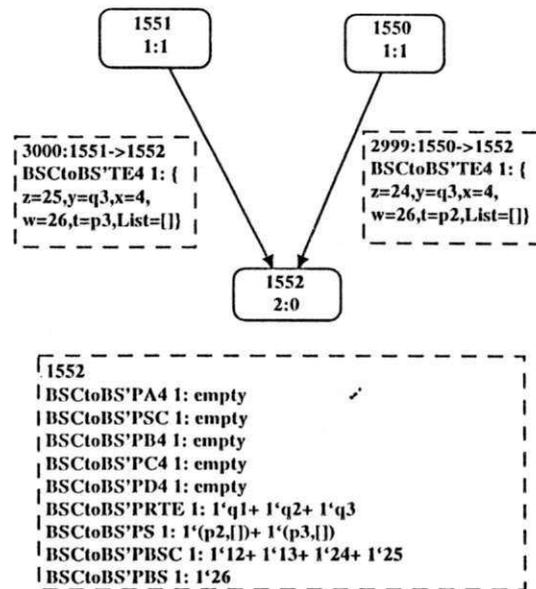


Figura 3.17: Estado de bloqueio

### 3.5 Considerações Finais

Neste capítulo, definimos os sistemas flexíveis de manufatura (SFM) assim como, apresentamos um procedimento de modelagem sistemático e estruturado para tais sistemas. Apresentamos modelos de alto nível possuindo uma estrutura que pode ser usada para modelar qualquer SFM definido no contexto desta tese e resolver conflitos por meio das informações contidas nas fichas. O modelo do supervisor é composto por sete páginas onde, a primeira página modela o fluxo de ordens de serviço e as demais páginas modelam a supervisão dos recursos de transporte do sistema. Com base nos resultados aqui obtidos, apresentaremos no capítulo seguinte a supervisão de sistemas flexíveis de manufatura tolerantes a faltas.

## Capítulo 4

# Supervisão de Sistemas Flexíveis de Manufatura Tolerantes a Faltas

Com base nos resultados obtidos no capítulo anterior, neste capítulo tratamos da supervisão dos SFMs tolerantes a faltas. Para tanto, utilizamos tanto recuperação de erros para frente como para trás. No caso da técnica de recuperação de erros para frente a ação corretiva é o re-escalonamento de rotas de produção. Já a técnica de recuperação de erros para trás é utilizada em nível de gerenciamento local de recursos. O modelo do supervisor apresentado neste capítulo é uma extensão ao modelo de rede de Petri Colorida apresentado no Capítulo 3 [BPF97d; BPF97a].

### 4.1 Os Casos de Re-Escalamento de Rotas de Produção

Como qualquer outro tipo de sistema, os sistemas de produção estão sujeitos a falhas e conseqüentemente a diminuição da produção. Desta forma, os aspectos relacionados com o grau de confiança na operação do sistema e a capacidade de assegurar a continuidade dos serviços tem ganhado uma grande importância nos últimos anos [Jal94; Lap92]. Um sistema tolerante a faltas possui algum tipo de redundância. Em outras palavras, um sistema é tolerante a faltas se o comportamento do sistema, mesmo falhando alguns de seus componentes, é consistente com as especificações. No contexto deste trabalho, consideramos a existência de recursos de produção redundantes, denominados réplicas.

Por exemplo, o recurso  $r_j$  é dito réplica do recurso  $r_i$ , se  $r_j$  for capaz de executar todas as atividades de  $r_i$ . Para tolerar faltas, as seguintes fases são executadas:

1. *Fase da detecção de erros*: nesta fase, os erros são detectados. A ocorrência de erros indica a presença de faltas no sistema,
2. *Fase da recuperação de erros*: nesta fase, o erro é removido. Para tanto, o sistema deve ser levado a um estado livre de erro. Caso o erro não seja removido, este poderá se propagar levando assim o sistema a falhar. Duas técnicas podem ser usadas para recuperação de erro, elas são: (a) recuperação para trás (*backward error recovery*) e (b) recuperação para frente (*forward error recovery*). No caso (a), durante o processamento normal, os dados relativos ao estado do sistema são periodicamente armazenados em pontos de verificação (*checkpoints*). Se ocorrer uma falta no sistema, a restauração dos dados relativos ao mais recente ponto de verificação removerá os erros detectados. No caso (b) não existem pontos de verificação e a remoção de erros é feita por meio de ações corretivas, como, por exemplo, o re-escalonamento de rotas de produção. Para exemplificar como é feito o re-escalonamento, considere dois recursos  $r_i$  e  $r_j$  de uma célula  $c_i$ , onde  $r_j$  é réplica de  $r_i$ . Um robô opera transportando subprodutos do depósito de entrada de  $c_i$  para o recurso  $r_i$ . Neste caso a recuperação para frente é: antes de alocar o robô para executar o transporte, o supervisor verifica se  $r_i$  falhou. Se este não falhou, nenhuma ação corretiva é tomada, senão, a rota é re-escalorada, ou seja, desviada para a réplica  $r_j$  garantindo assim a continuidade do serviço. Nesta tese, empregamos a técnica (a) em nível de gerenciamento local e a técnica (b) em nível de supervisão,
3. *Fase do tratamento de faltas*: nesta fase a falta é localizada e o componente redundante assume as atividades do componente defeituoso, assegurando assim a continuidade do serviço.

A recuperação de erros para frente pode ser aplicada nos casos [BPF97d] descritos a seguir.

### 4.1.1 Caso 1

Seja  $r_i$  um recurso de produção de uma célula  $c_i$  e  $r_j$  sua réplica. Com relação a  $r_i$ , o recurso  $r_j$  pode estar localizado dentro ou fora da célula  $c_i$ . Suponha que  $r_i$  falhou. No primeiro caso, conforme é ilustrado na Figura 4.1, o recurso  $r_j$  está dentro da célula  $c_i$ . A rota de produção  $h$  é re-escalada no depósito de entrada  $bec_i$  da célula  $c_i$  e no depósito de entrada  $bem_i$  da máquina  $m_i$  para a réplica  $r_j$ . Na Figura 4.1 e nas seguintes (Figuras 4.2, 4.3 e 4.4), os arcos de seta cheia representam as atividades de transporte do sistema e os arcos tracejados a ausência destas atividades. Já os arcos de seta vazia representam o fluxo de ordens de serviço associado ou não com as atividades de transporte do sistema. Detalhes sobre a representação destes arcos são apresentados na Seção 4.2.2.

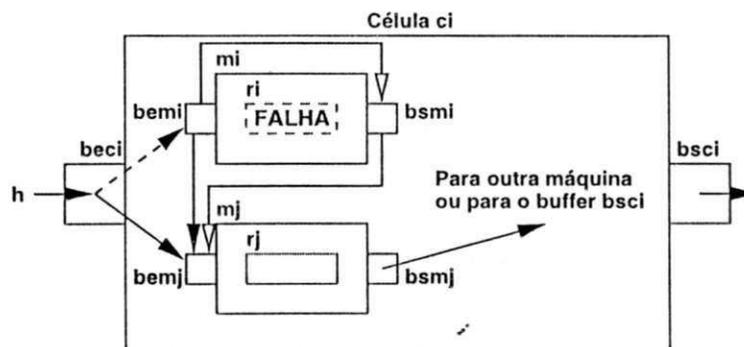


Figura 4.1: Re-escalamento para o Caso 1

### 4.1.2 Caso 2

Neste caso, conforme é ilustrado na Figura 4.2, o recurso  $r_j$  também está dentro da célula  $c_i$  mas a rota de produção  $h$  é re-escalada no depósito de saída  $bsm_k$  da máquina  $m_k$  e no depósito de entrada  $bem_i$  da máquina  $m_i$  para a réplica  $r_j$ .

### 4.1.3 Caso 3

Neste caso, temos dois cenários: No primeiro cenário, conforme é ilustrado na Figura 4.3(a), o recurso  $r_j$  está fora da célula  $c_i$  e a rota de produção  $h$  é re-escalada no depósito de saída  $bsc_k$  da célula  $c_k$  e no depósito de entrada  $bem_i$  da máquina  $m_i$  para a réplica  $r_j$ . No segundo cenário, conforme é ilustrado na Figura 4.3(b), o recurso  $r_j$  está também fora da

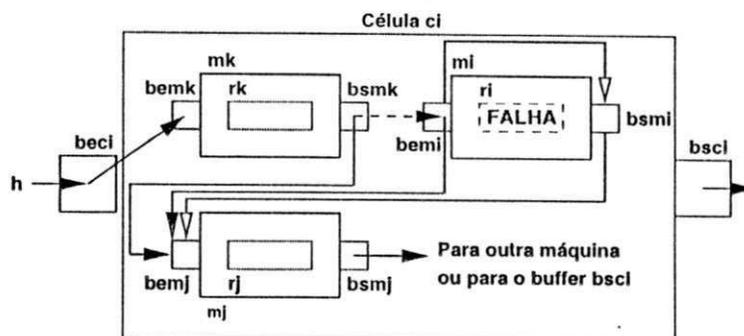


Figura 4.2: Re-escalamento para o Caso 2

célula  $c_i$ , mas a rota de produção  $h$  é re-escalada no depósito de entrada  $be$  do sistema e no depósito de entrada  $bem_i$  da máquina  $m_i$  para a réplica  $r_j$ .

#### 4.1.4 Caso 4

Neste caso, conforme é ilustrado na Figura 4.4, o recurso  $r_j$  está também fora da célula  $c_i$ , mas a rota de produção  $h$  é re-escalada no depósito de saída  $bsm_k$  da máquina  $m_k$  e no depósito de entrada  $bem_i$  da máquina  $m_i$  para a réplica  $r_j$ .

## 4.2 Especificação do Supervisor

Apresentamos a seguir, o modelo do supervisor considerando os casos descritos acima. O modelo foi construído conforme a hierarquia de páginas<sup>1</sup> ilustrada na Figura 4.5. O modelo possui seis páginas, onde, a primeira página, ilustrada na Figura 4.6, modela o fluxo de ordens de serviço conforme as rotas de produção do sistema. A segunda página, denominada *re-escalamento bem*, modela o re-escalamento das rotas de produção no depósito de entrada das máquinas, a terceira página, denominada *re-escalamento bec*, modela o re-escalamento das rotas de produção no depósito de entrada das células, e assim por diante. A composição do modelo é feita por meio da fusão de lugares comuns a todas as páginas.

<sup>1</sup>O conceito de páginas de um modelo CPN hierárquico está apresentado na Seção 2.4.2.

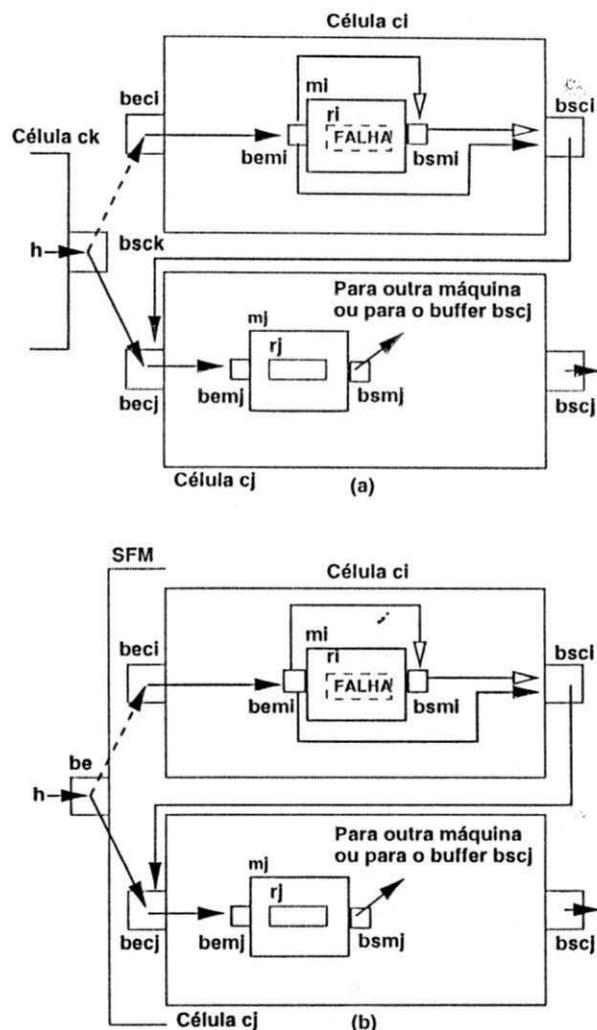


Figura 4.3: Re-escalamento para o Caso 3

#### 4.2.1 Descrição da Página *fluxo de ordens de serviço*

A primeira página, apresentada na Figura 4.6, possui lugares e transições com o seguinte significado:

Lugares:

*PE* e *PE1*: modelam ordens de serviço para o transporte da matéria-prima do depósito de entrada do sistema para o depósito de entrada de uma célula,

*PSC* e *PSC1*: modelam ordens de serviço para o transporte de subprodutos do depósito de saída para o depósito de entrada de uma célula ou para o depósito de saída do sistema,

*PS*: modela ordens de serviço executadas.

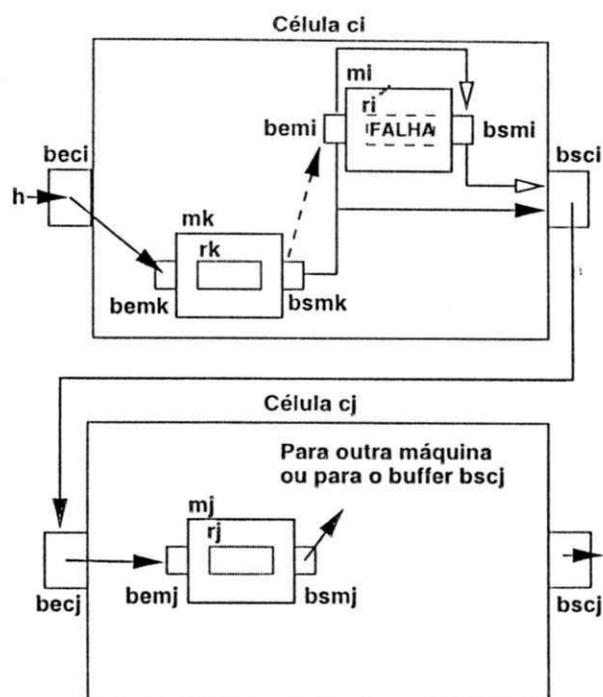


Figura 4.4: Re-escalamento para o Caso 4

*PEC e PEC1:* modelam ordens de serviço para o transporte de subprodutos do depósito de entrada de uma célula para o depósito de entrada de uma máquina dentro da célula,

*PEM:* modela ordens de serviço para a manufatura de subprodutos contidos no depósito de entrada das máquinas,

*PSM e PSM1:* modelam ordens de serviço para o transporte de subprodutos do depósito de saída para o depósito de entrada de uma máquina dentro de uma célula ou para o depósito de saída da célula,

*PAR e PVS:* modelam um recurso de produção ocupado,

*PF:* modela um recurso de produção livre.

A ocorrência das transições corresponde aos seguintes eventos:

Transições:

*TEC:* modela o transporte de subprodutos do depósito *be* para os depósitos *bec*,

*TSC:* modela o transporte de subprodutos dos depósitos *bsc* para o depósito *bs*,

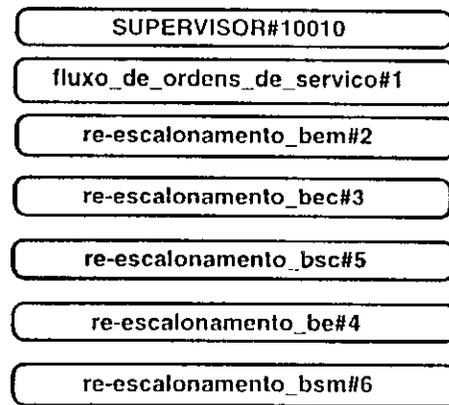


Figura 4.5: Páginas do supervisor

*TEM*: modela o transporte de subprodutos dos depósitos *bec* para os depósitos *bem*,

*TSM*: modela o transporte de subprodutos dos depósitos *bsm* para os depósitos *bsc*,

*TPC*: modela o transporte de subprodutos dos depósitos *bsc* para os depósitos *bcc*,

*TPM*: modela o transporte de subprodutos dos depósitos *bsm* para os depósitos *bem*,

*TER*: modela o início de uma atividade de produção,

*TFO*: modela o término de uma atividade de produção (recurso ocupado),

*TSR*: modela o término de uma atividade de produção (recurso livre).

Na marcação inicial desta página, o lugar *PF* pode conter um número arbitrário de fichas,  $r_1 + r_2 + \dots + r_k$ , indicando que existem  $k$  recursos de produção disponíveis. Da mesma forma, lugar *PE* pode conter um número arbitrário de fichas modelando ordens de serviço  $(p_i, [lista_i])$  onde,  $lista_i$  é a lista de todas as *requisições de serviço* empregadas para a manufatura do produto  $p_i$ .

As expressões de arco são  $E_0 = (t, ((x, y, z, w, s) :: List))$  ou  $E_1 = (t, List)$  onde,  $t$  especifica o produto a ser manufaturado,  $x$  é utilizada para resolver conflitos,  $y$  especifica um recurso,  $z$  especifica um depósito fonte,  $w$  especifica um depósito destino,  $s$  especifica o estado dos recursos e  $List$  especifica a cauda da lista. Note que, devido a expressão de arco  $(t, List)$ , a cabeça da lista  $(x, y, z, w, s)$  é removida, após a ocorrência de algumas transições. Isto deve ser feito pois recursos já utilizados, ou seja, requisições já atendidas devem ser removidas da lista. Desta forma, o lugar *PS* deve conter fichas  $(p_i, [])$ , ou seja, ordens de serviço executadas do produto  $p_i$  onde,  $[]$  representa a lista vazia.

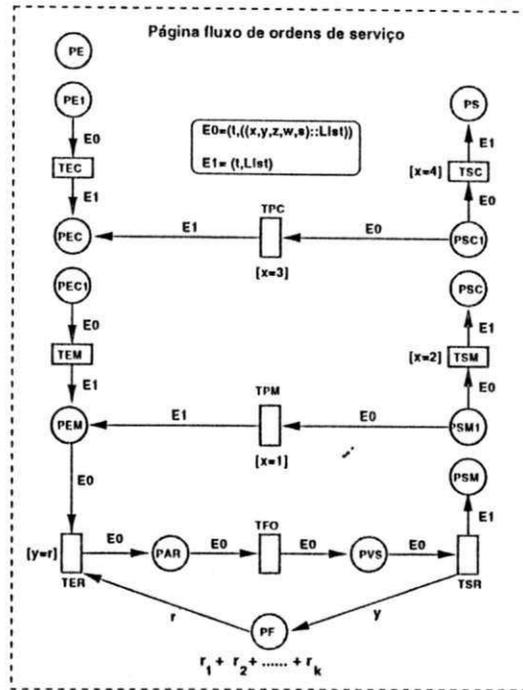


Figura 4.6: Página *fluxo de ordens de serviço*

Note que, as transições *TPM* e *TSM* estão em conflito. Este conflito surge porque subprodutos contidos no depósito de saída de uma máquina podem ser transportados para outra máquina ou para a saída da célula que contém estas máquinas. Este conflito é resolvido através da expressão de guarda  $[x = 1]$  associada a transição *TPM* e pela expressão de guarda  $[x = 2]$  associada a transição *TSM*. Se  $x = 1$  então *TPM* ocorre senão, *TSM* ocorre. Da mesma forma, as transições *TPC* e *TSC* estão em conflito. Este conflito surge porque subprodutos do depósito de saída de uma célula podem ser transportados para a entrada de outra célula ou para a saída do sistema. Este conflito é resolvido através da expressão de guarda  $[x = 3]$  associada a transição *TPC* e pela expressão de guarda  $[x = 4]$  associada a transição *TSC*. Se  $x = 3$  então *TPC* ocorre senão, *TSC* ocorre. A declaração de variáveis e cores do modelo do supervisor está ilustrada na Figura 4.7.

Para tolerar faltas, o supervisor deve possuir mecanismos para a detecção e recuperação de erros. Um erro é detectado dependendo do valor que a variável *s* assume no lugar *PVS*. Portanto, o valor desta variável indica se um recurso especificado pela variável *y* falhou ou não. Por exemplo, suponha que em um dado instante o recurso  $r_i$  está processando, isto é, o lugar *PAR* contém uma ficha. Após o término da atividade, ocorre a transição *TFO* adicionando uma ficha ao lugar *PVS* (veja Figura 4.6). Se o funcionamento do recurso foi

```

color T = with p1|p2;
color X = int;
color Y = with r1|z2|x3|x4|x5|x6|x7|x8|q1|q2|q3|q4|q5|q6|q7;
color Z = int;
color W = int;
color S = int;
color M = int;
color N = int;
color FLAG = int;
color E = product X*Y*Z*W*S;
color L = list E;
color P = product T*L;
color D = product Y*M*N*FLAG*L*L;

var m:M;
var n:N;
var flag: FLAG;
var t,t1: T;
var x,x1,x2,x3: X;
var y,y1,y2,y3: Y;
var z,z1,z2,z3: Z;
var w,w1,w2,w3: W; var s,s1,s2,s3: S;
var List,List1,List2,List3: L;

```

Figura 4.7: Declaração de variáveis e cores

normal a variável  $s$  assume o valor zero senão, esta variável assume o valor igual a 1 (um) indicando que o recurso falhou. Neste caso, o supervisor deve fazer o re-escalonamento das rotas de produção passando pelo recurso. Uma vez consertado o recurso, este pode entrar novamente em operação a partir dos dados armazenados no seu mais recente ponto de verificação. Para exemplificar este procedimento, considere a manufatura de porcas ilustrada Figura 4.8. Suponha que, o recurso  $r_1$  falhou no início do corte da face  $d$  de uma porca, conforme é ilustrado na figura 4.8. Uma vez que registra-se periodicamente em pontos de verificação os dados relativos ao último serviço executado, o recurso poderá então ser inicializado a partir deste dados, assegurando assim a continuidade do serviço requisitado. Neste caso, o corte das faces  $d$ ,  $e$ ,  $f$  do material.

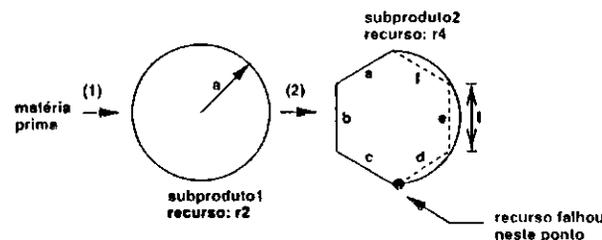


Figura 4.8: Falha do recurso  $r_4$

Para descrever as páginas seguintes de re-escalonamento, as seguintes notações são empregadas. Sejam as listas  $l_1 = [\alpha_1\alpha_2 \cdots \alpha_m]$  e  $l_2 = [\beta_1\beta_2 \cdots \beta_n]$ . O símbolo  $\wedge$  indica concatenação de listas. Por exemplo,  $l_1 \wedge l_2 = [\alpha_1\alpha_2 \cdots \alpha_m\beta_1\beta_2 \cdots \beta_n]$ . A função  $nthtail(l, k)$  remove os  $k$  primeiros elementos da lista  $l$  e retorna a cauda da lista. Por exemplo,  $nthtail(l_1, 2) = \alpha_3\alpha_4 \cdots \alpha_m$ . Em conseqüência,  $nthtail(l_1, m) = []$ , onde,  $[]$  representa lista vazia. Sejam  $x_1$  e  $x_2$  variáveis. Na regra *if* ( $x_1 \neq x_2$ ) *then*  $A$  *else*  $B$ , a expressão  $A$  é executada para todo  $x_1$  diferente de  $x_2$ , senão a expressão  $B$  é executada.

Os lugares e transições das páginas de re-escalonamento descritas a seguir, possuem as seguintes características:

1. As transições não estão relacionadas com qualquer atividade do sistema,
2. Com exceção dos lugares *PD* e *PD1*, todos lugares das páginas de re-escalonamento são comuns à página *fluxo de ordens de serviço* na Seção 4.2.1. Portanto, a descrição destes lugares comuns será omitida.

As informações contidas nas fichas do lugar *PD*, são:

1. O recurso de produção representado pela variável *y*,
2. A variável *m* especificando o número de elementos a ser removido da lista *List1* para ser concatenada com a lista *List2*, ou seja,  $List2 \wedge \wedge nhtail(List1, m)$ ,
3. A variável *n* especificando o número de elementos a ser removido da lista *List1* para ser concatenada com a lista *List3*, ou seja,  $List3 \wedge \wedge nhtail(List1, n)$ ,
4. A variável *flag* especificando um inteiro associado aos depósitos do sistema onde,
  - flag* = 1: indica o depósito de entrada do sistema,
  - flag* = 2: indica o depósito de entrada das células,
  - flag* = 3: indica o depósito de saída das máquinas
  - flag* = 4: indica o depósito de saída das células,
5. As listas *List2* e *List3*

#### 4.2.2 Descrição da Página *re-escalonamento bem*

Esta página, apresentada na Figura 4.9, modela o re-escalonamento das rotas de produção no depósito de entrada das máquinas. Nesta página, a ocorrência da transição *TET* remove fichas do lugar *PEM* e adiciona fichas ao lugar *PSM1* indicando que um recurso falhou. As fichas adicionadas são definidas pela expressão  $(t_1, List3 \wedge \wedge nhtail(List1, n))$ .

Na Figura 4.10, mostra-se de que forma esta página está relacionada com os casos de re-escalonamento descrito na Seção 4.1. Cada depósito do sistema corresponde a um lugar

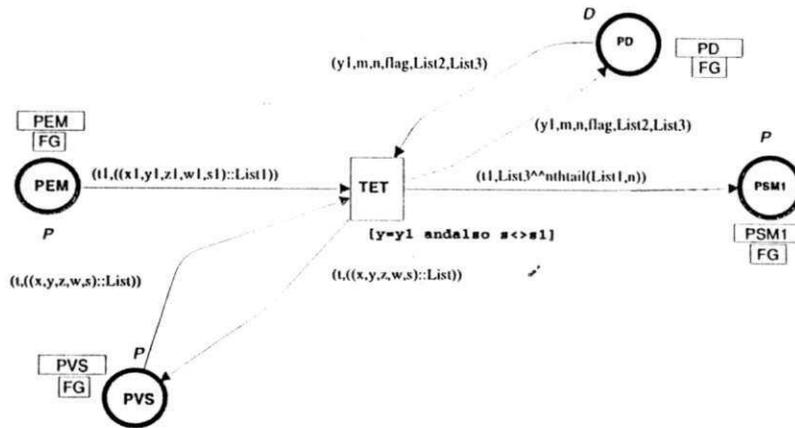


Figura 4.9: Página re-escalonamento bem

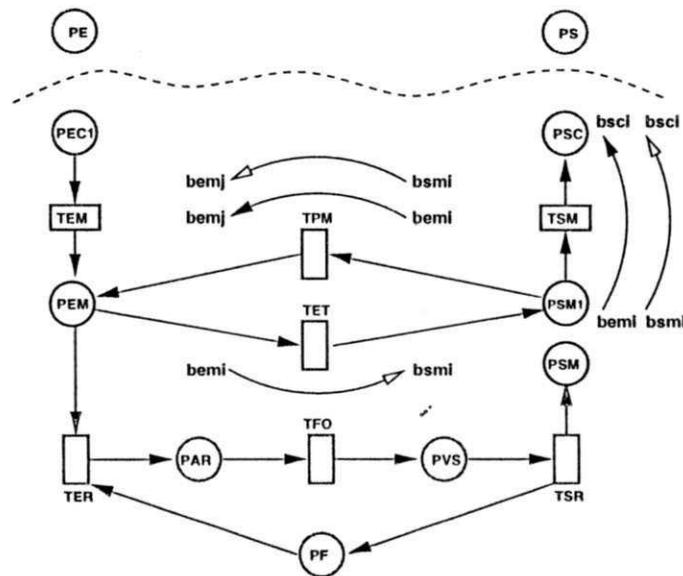


Figura 4.10: Páginas re-escalonamento bem e fluxo de ordens de serviço

no modelo. É o fluxo de ordens de serviço entre estes lugares que determina as atividades do sistema. Entretanto, não foi considerado na concepção do modelo, o transporte entre os depósitos ( $bem_i, bem_j$ ) ou ( $bem_i, bsc_i$ ), o que deve ser feito se um recurso falha. Para resolver este problema, as ordens de serviço, mediante a ocorrência da transição *TET*, são removidas do lugar *PEM* e adicionadas ao lugar *PSM1*, isto sem nenhum comando para o sistema de transporte. Note que uma ordem de serviço no lugar *PEM1* habilita as transições *TPM* ou *TSM*. O transporte entre os depósitos ( $bem_i, bem_j$ ) ou ( $bem_i, bsc_i$ ) é modelado pela ocorrência das transições *TPM* ou *TSM* respectivamente. Portanto, os arcos de seta vazia, representando o fluxo de ordens de serviço entre os depósitos ( $bsm_i, bem_j$ )

ou  $(bsm_i, bsc_i)$ , correspondem ao transporte entre os depósitos  $(bem_i, bem_j)$  ou  $(bem_i, bsc_i)$  respectivamente. Já os arcos de seta vazia, representando o fluxo de ordens de serviço entre os depósitos  $(bem_i, bsm_i)$ , não estão relacionados com nenhuma atividade de transporte. Na Figura 4.10 mostra-se como a página *re-escalonamento bem* está relacionada com a página *fluxo de ordens de serviço* descrita na Seção 4.2.1. Por simplicidade, os lugares *PD* e *PVS* foram omitidos nesta representação.

### 4.2.3 Descrição da Página *re-escalonamento bec*

Esta página, apresentada na Figura 4.11, modela o re-escalonamento das rotas de produção no depósito de entrada das células. Nesta página, a ocorrência da transição *TET1* remove fichas do lugar *PEC* e adiciona fichas ao lugar *PEC1* sem alterar as informações contida nas fichas.

Já a ocorrência da transição *TET2* remove fichas do lugar *PEC* e adiciona fichas ao lugar *PEC1*, mas se o recurso falhou, as fichas adicionadas são definidas pela expressão  $(t_1, List2 \wedge \wedge nthtail(List1, m))$ . Caso contrário, as fichas adicionadas não são alteradas. Esta página está relacionada ao primeiro caso de re-escalonamento descrito na Seção 4.1.1.

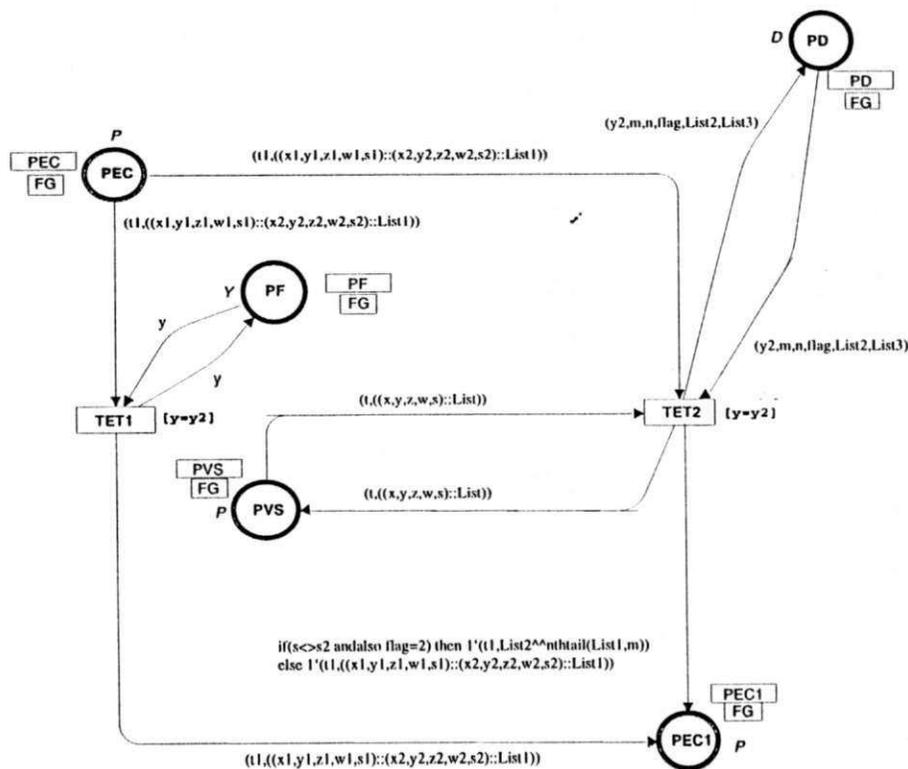


Figura 4.11: Página *re-escalonamento bec*

#### 4.2.4 Descrição da Página *re-escalonamento be*

Esta página, apresentada na Figura 4.12, modela o re-escalonamento das rotas de produção no depósito de entrada do sistema. Nesta página, a ocorrência da transição *TET3* remove fichas do lugar *PE* e adiciona fichas ao lugar *PE1* sem alterar as informações contida nas fichas.

Já a ocorrência da transição *TET4* remove fichas do lugar *PE* e adiciona fichas ao lugar *PE1*, mas se o recurso falhou, as fichas adicionadas são definidas pela expressão  $(t_1, List2 \wedge \wedge nthtail(List1, m))$ . Caso contrário, as fichas adicionadas não são alteradas. Esta página está relacionada ao terceiro caso de re-escalonamento descrito na Seção 4.1.3.

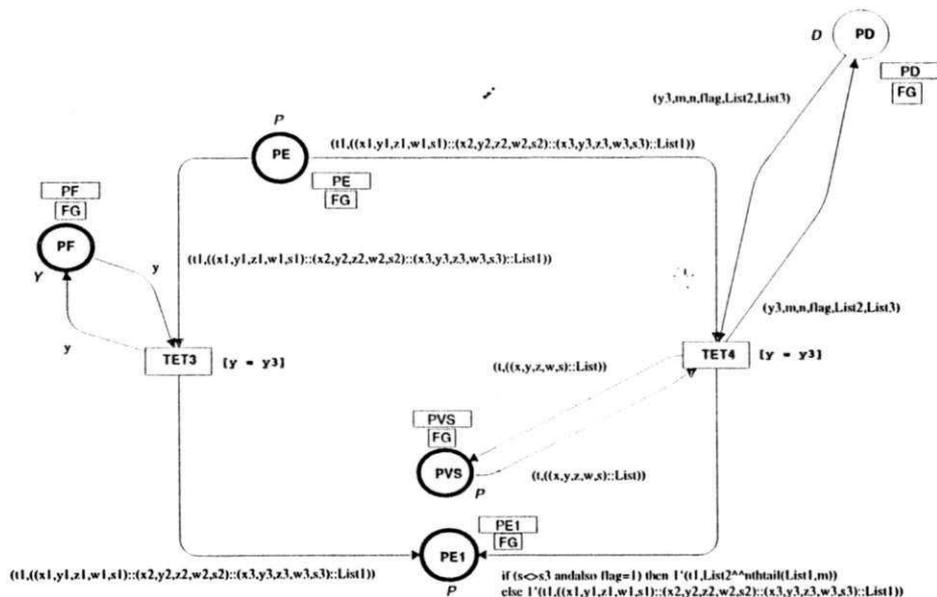


Figura 4.12: Página *re-escalonamento be*

#### 4.2.5 Descrição da Página *re-escalonamento bsc*

Esta página, apresentada na Figura 4.13, modela o re-escalonamento das rotas de produção no depósito de saída das células. Nesta página, a ocorrência das transição *TET5* ou *TET7* remove fichas do lugar *PSC* e adiciona fichas ao lugar *PSC1* sem alterar as informações contida nas fichas.

Já a ocorrência da transição *TET6* remove fichas do lugar *PSC* e adiciona fichas ao lugar *PSC1*, mas se o recurso falhou, as fichas adicionadas são definidas pela expressão  $(t_1, List2 \wedge \wedge nthtail(List1, m))$ . Caso contrário, as fichas adicionadas não são alteradas.

Esta página está relacionada ao terceiro caso de re-escalonamento descrito na Seção 4.1.3.

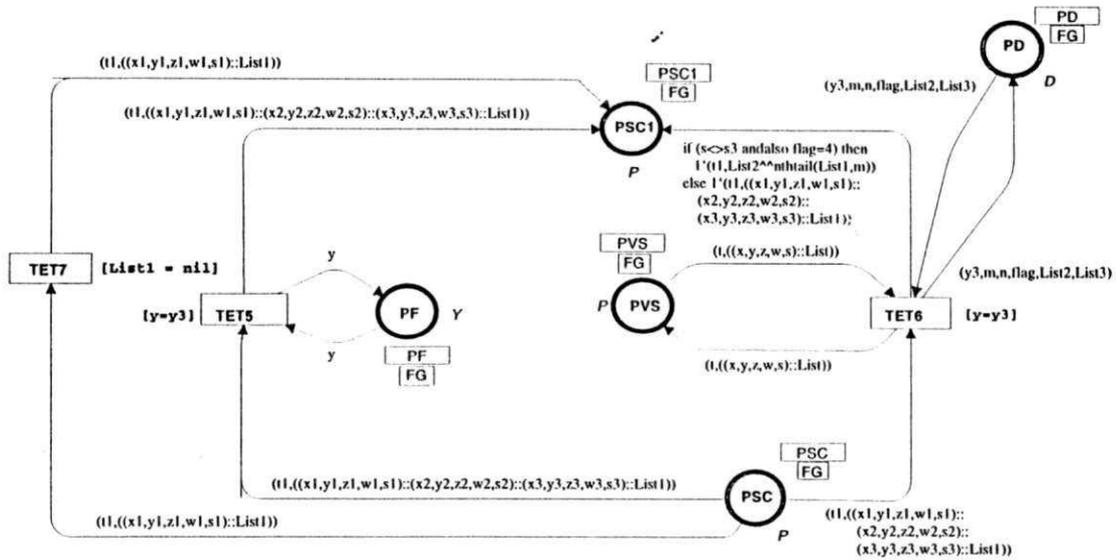


Figura 4.13: Página *re-escalonamento bsc*

#### 4.2.6 Descrição da Página *re-escalonamento bsm*

Esta página, apresentada na Figura 4.14, modela o re-escalonamento das rotas de produção no depósito de saída das máquinas. Nesta página, a ocorrência das transição *TET9* ou *TET0* remove fichas do lugar *PSM* e adiciona fichas ao lugar *PSM1* sem alterar as informações contida nas fichas.

Já a ocorrência da transição *TET8* remove fichas do lugar *PSM* e adiciona fichas ao lugar *PSM1*, mas se o recurso falhou, as fichas adicionadas são definidas pela expressão  $(t_1, List2 \wedge \text{nthtail}(List1, m))$ . Caso contrário, as fichas adicionadas não são alteradas. Esta página está relacionada ao segundo e quarto casos de re-escalonamento descritos nas Seções 4.1.2 e 4.1.4, respectivamente.

### 4.3 Exemplo de Re-Escalonamento de Rotas de Produção

Seja o sistema flexível de manufatura (SFM) ilustrado na Figura 4.15 onde, o recurso de produção  $r_2$  é réplica dos recursos de produção  $r_1$  e  $r_6$ , o recurso de produção  $r_5$  é réplica

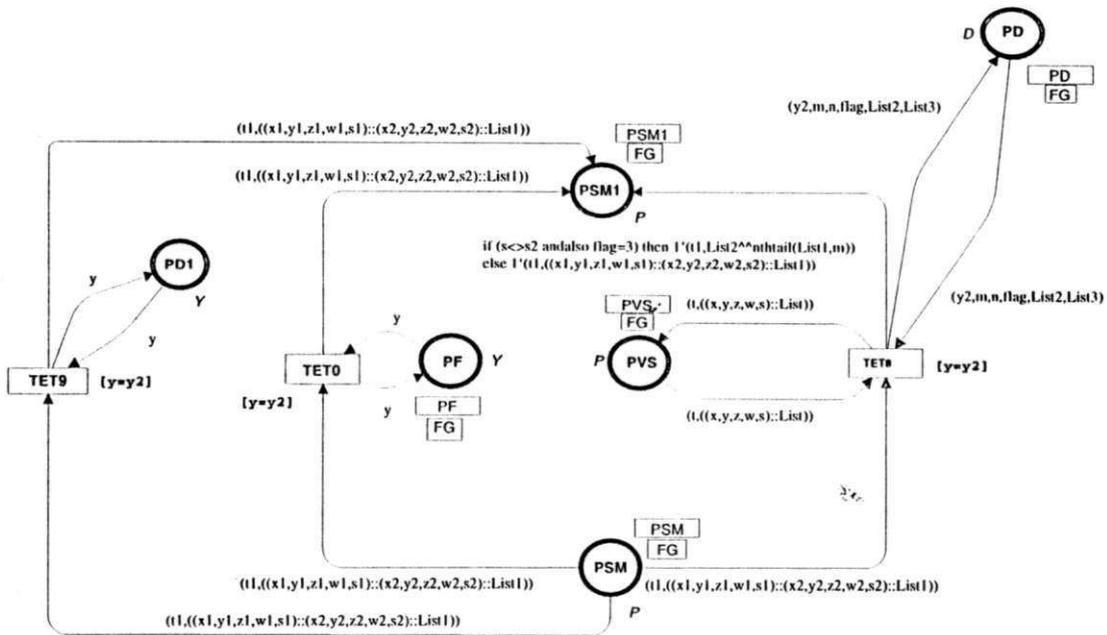


Figura 4.14: Página re-escalonamento bsm

dos recursos de produção  $r_4$  e  $r_8$ . As rotas de produção são  $h_1$  e  $h_2$  dos produtos  $p_1$  e  $p_2$ , respectivamente.

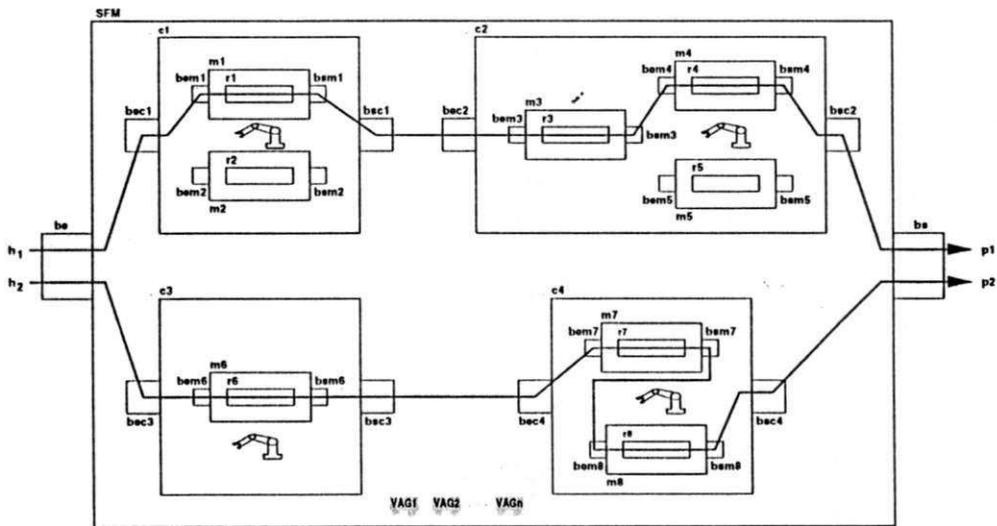


Figura 4.15: SFM tolerante a falhas

O grafo de atividades deste sistema, considerando a falha dos recursos  $r_1$ ,  $r_4$ ,  $r_6$  e  $r_8$ , é mostrado na Figura 4.16. Para exemplificar o re-escalonamento das rotas de produção, considere como exemplo, que o recurso  $r_1$  falhou. Devido a isto, os arcos (2, 4), (4, 5) e (5, 3) estão representados em linha tracejada na Figura 4.16, indicando que a máquina  $m_1$

não é carregada ou descarregada pelo sistema de transporte. Note que, o recurso  $r_2$  (réplica do recurso  $r_1$ ) está localizado na mesma célula do recurso  $r_1$ , logo o re-escalonamento deve ser feito no depósito de entrada da máquina  $m_1$  e no depósito de entrada da célula  $c_1$ . Veja a descrição do caso de re-escalonamento descrito na Seção 4.1.1. O re-escalonamento é feito por meio da concatenação de duas listas. Desta forma, para fazer o re-escalonamento no depósito de entrada da máquina  $m_1$ , as listas associadas às seqüência de arcos  $(4, 6)$ ,  $(6, 7)$ ,  $(7, 3)$  e  $(3, 8), \dots, (12, 13), (13, 9), (9, 26)$  devem ser concatenadas. Esta seqüência de arcos está representada na Figura 4.16. Da mesma forma, para fazer o re-escalonamento no depósito de entrada da célula  $c_1$ , as listas associadas as seqüências de arcos  $(2, 6)$ ,  $(6, 7)$ ,  $(7, 3)$  e  $(3, 8), \dots, (12, 13), (13, 9), (9, 26)$  devem ser concatenadas.

Para analisar o modelo do sistema apresentado na Figura 4.15, utilizamos a ferramenta *Design/CPN*. Baseado nos resultados da análise, mostra-se que as ordens de serviço são executadas mesmo ocorrendo faltas no sistema.

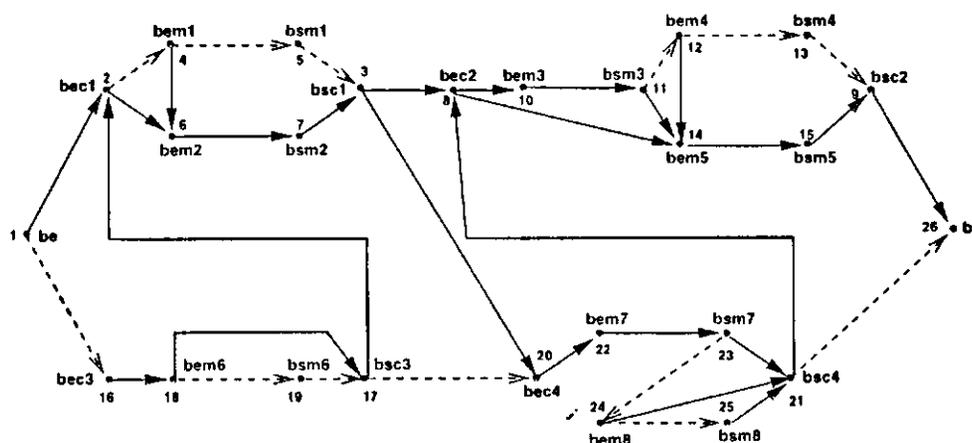


Figura 4.16: Grafo de atividades considerando a falha dos recursos  $r_1, r_4, r_6$  e  $r_8$

## 4.4 Considerações Finais

Neste capítulo, tratamos da supervisão dos sistemas flexíveis de manufatura tolerantes a faltas. Para tanto, utilizamos as técnicas de recuperação de erros para frente e para trás. A primeira técnica é baseada no re-escalonamento de rotas de produção. Já a técnica de recuperação de erros para trás, utilizada em nível de gerenciamento local de recursos, está relacionada com o estabelecimento dos chamados pontos de verificação (*checkpoints*).

Com base nos resultados obtidos no Capítulo 3, especificamos o supervisor considerando a ocorrência de faltas. O modelo obtido é formado por seis páginas onde, a primeira página modela o fluxo de ordens de serviço e as demais páginas modelam o re-escalonamento de rotas de produção. No final deste capítulo, apresentamos um exemplo do re-escalonamento onde o recurso e a réplica estão localizados na mesma célula. Devido à semelhança no procedimento, a exemplificação considerandos os outros casos foi omitida.

## Capítulo 5

# Supervisão de Sistemas de Produção em Lotes

Neste capítulo, tratamos da supervisão fixa e dinâmica dos sistemas de produção em lotes utilizando redes de Petri [BP98a]. Para tanto, começamos o capítulo definindo os sistemas de produção em lotes e com base na definição, apresentamos alguns exemplos de tais sistemas. Na supervisão fixa, as requisições de serviço para a produção de um lote são previamente determinadas, logo o supervisor não possui alternativas de produção. Já no caso dinâmico, o supervisor conta com uma ou mais alternativas de produção e o processo de escolha é baseado na disponibilidade dos recursos. Com base no conceito de supervisão dinâmica, tratamos da alocação dos recursos de manutenção do sistema.

### 5.1 Definição para Sistemas de Produção em Lotes

Os sistemas de produção em lotes são formados por diversos recursos de armazenamento, produção e transporte. Os recursos de armazenamento, denominados tanques, armazenam temporariamente a matéria-prima, subprodutos ou produtos. Os recursos de produção, encontrados nos reatores, são utilizados para produzir os lotes. Denominamos por lote a quantidade de um subproduto produzida em um reator do sistema. Esta quantidade, expressa em unidades de massa ou volume, varia e depende da receita de cada produto. Os recursos de transporte permitem a transferência dos lotes entre os tanques do sistema. Devido a transferência dos lotes, surge um fluxo entre os recursos, denominado fluxo de

produção.

Para exemplificar estes conceitos, na Figura 5.1(a) ilustra-se um sistema de produção em lotes formado por  $m$  tanques e  $n$  reatores onde, os arcos orientados representam o *fluxo de produção*, os círculos os tanques e os retângulos os reatores do sistema.

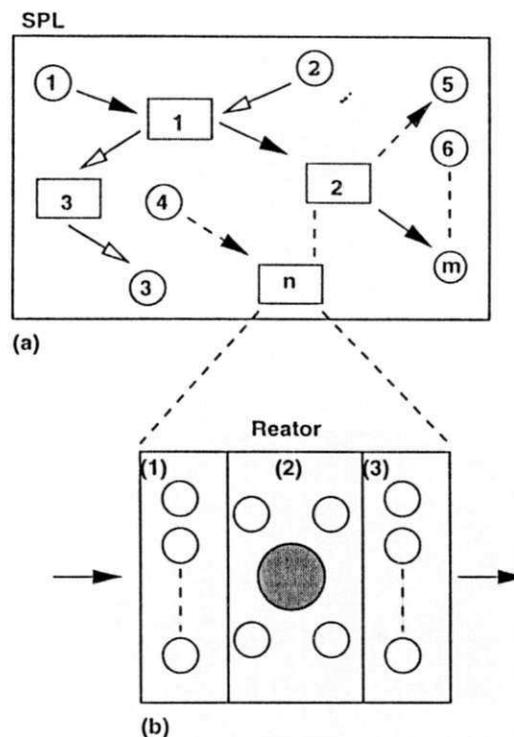


Figura 5.1: (a) Visão geral de um SPL (b) Arquitetura de um reator

Na Figura 5.1(b) mostra-se detalhes de um reator do sistema. Este pode ser formado por três partes denominadas área de armazenamento de entrada (1), área de produção (2) e área de armazenamento de saída do reator (3). Localizados na área (1), os tanques depósitos de entrada armazenam o *refluxo de produção*, ou seja, o excesso do material reutilizável produzido em outros reatores do sistema. Na área (2) está localizado o recurso de produção do reator. Este recurso é formado por um tanque principal (círculo maior), tanques auxiliares e equipamentos utilizados para a produção dos lotes. Os recursos de produção são programados para atender a produção de cada lote. Localizados na área (3), os tanques depósitos de saída armazenam os lotes da área de produção do reator. Os tanques das áreas de armazenamento estão conectados, por meio de válvulas e dutos, ao tanque principal assim como, a entrada e saída do reator. Baseado nestas considerações, definimos os sistemas de produção em lotes da seguinte forma:

**Definição 5.1** Um Sistema de Produção em Lotes é a quadrupla  $\langle U, L, R, S \rangle$ , onde

1.  $U = U_a \cup U_p \cup U_t$  é um conjunto finito e não vazio de recursos onde,  $U_a$  representa os recursos de armazenamento,  $U_p$  os recursos de produção e  $U_t$  os recursos de transporte do sistema,
2.  $L = \{\text{lote}_1, \dots, \text{lote}_i, \dots, \text{lote}_n\}$  é um conjunto finito e não vazio de lotes onde,  $\text{lote}_i$  é o lote a ser produzido conforme a receita  $R_i \in R$ ,
3.  $R = \{R_1, \dots, R_i, \dots, R_n\}$  é um conjunto finito e não vazio de receitas onde,  $R_i$  é a receita do  $\text{lote}_i \in L$ .
4.  $S$  é um conjunto finito e não vazio de reatores.

Entretanto, para produzir um  $\text{lote}_i$  devemos especificar o fluxo de produção entre os recursos utilizados e a funcionalidade destes recursos.

**Definição 5.2** A receita  $R_i$  do  $\text{lote}_i$  é a dupla  $\langle RAN_i, RBN_i \rangle$  onde,

1.  $RAN_i$ , denominada *Receita de Alto Nível*, especifica o fluxo de produção entre os recursos necessários para produzir o  $\text{lote}_i$ ,
2.  $RBN_i$ , denominada *Receita de Baixo Nível*, determina como devem ser utilizados os recursos especificados na receita  $RAN_i$ .

Utilizaremos daqui por diante o símbolo  $\rightarrow$  para representar o fluxo de produção entre os recursos do sistema. Os lotes são produzidos por etapas denominadas *fases de produção*. Em cada fase o lote é transferido para a área de produção de um reator, processado e então transferido para fora da área de produção do reator. Portanto, uma fase de produção é formada por três subfases distintas, ou seja, uma subfase de transferência para dentro, uma subfase de processamento e uma subfase de transferência para fora da área de produção do reator, nesta ordem. Com base na definição 5.1, alguns exemplos de sistemas de produção em lotes são apresentados a seguir.

### 5.1.1 Receitas de Baixo Nível

Para exemplificar as receitas de baixo nível, considere que o sistema ilustrado na Figura 5.2(a) produz lotes de refrigerante de limão e uva, onde:

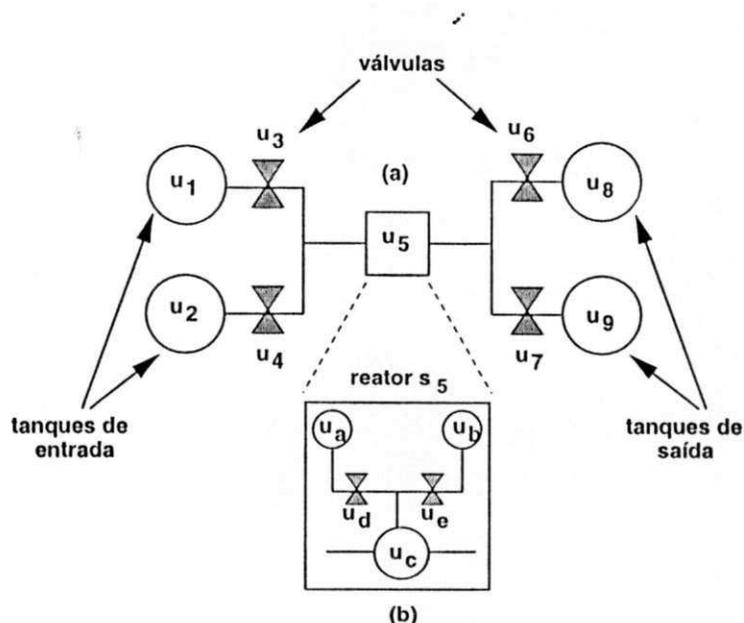


Figura 5.2: (a) Exemplo de um SPL (b) Visão detalhada do reator  $s_5$

1.  $U_a = \{u_1, u_2, u_8, u_9\}$ , onde o tanque  $u_1$  armazena o suco concentrado de limão, o tanque  $u_2$  armazena o suco concentrado de uva, o tanque  $u_8$  armazena um lote de refrigerante de limão e o tanque  $u_9$  armazena um lote de refrigerante de uva.
2.  $U_p = \{u_5\}$ , onde  $u_5$  produz os lotes de limão e uva. Desta forma,  $u_5$  representa o recurso de produção do reator  $s_5$ ,
3.  $U_t = \{u_3, u_4, u_6, u_7\}$ , onde a válvula  $u_3$  permite a passagem do suco de limão para  $u_5$ , a válvula  $u_4$  permite a passagem do suco de uva para  $u_5$ , a válvula  $u_6$  permite a passagem do refrigerante de limão para  $u_8$  e a válvula  $u_7$  permite a passagem do refrigerante de uva para  $u_9$
4.  $L = \{lote_1, lote_2\}$ , onde o  $lote_1$  é de refrigerante de limão e o  $lote_2$  é de refrigerante de uva,
5.  $R = \{R_1, R_2\}$ , onde  $R_1 = \{RAN_1, RBN_1\}$  é a receita para produzir o  $lote_1$  e  $R_2 = \{RAN_2, RBN_2\}$  é a receita para produzir o  $lote_2$ .
6.  $S = \{s_5\}$  é o reator do sistema.

Na Figura 5.2(b) mostra-se detalhes da área de produção do reator  $s_5$ . Note que, o reator é um subsistema formado por um tanque principal  $u_c$  onde, são produzidos os lotes e dois tanques auxiliares  $u_a$  e  $u_b$  onde,  $u_a$  armazena água e  $u_b$  armazena o adoçante.

A válvula  $u_d$  permite a passagem da água para o tanque  $u_c$  e a válvula  $u_e$  permite a passagem do adoçante. Controlando a abertura destas válvulas podemos determinar a taxa de transferência da mistura água adoçante para produzir cada lote. Os lotes são produzidos conforme as seguinte fases de produção:

1. *Fase de Produção A*: O suco de limão armazenado no tanque  $u_1$  é transferido, por meio da válvula  $u_3$ , para o reator  $s_5$  e processada no tanque  $u_c$ . O processamento consiste em adicionar, por meio as válvulas  $u_d$  e  $u_e$ , ao suco de limão a mistura água adoçante dos tanques  $u_a$  e  $u_b$ . Após o término desta fase o lote é transferido, por meio da válvula  $u_6$ , para o tanque  $u_8$ .
2. *Fase de Produção B*: O suco de uva armazenado no tanque  $u_2$  é transferido, por meio da válvula  $u_4$ , para o reator  $s_5$  e processada no tanque  $u_c$ . O processamento consiste em adicionar, por meio das válvulas  $u_d$  e  $u_e$ , ao suco de uva a mistura água adoçante dos tanques  $u_a$  e  $u_b$ . Após o término desta fase o lote é transferido, por meio da válvula  $u_7$ , para o tanque  $u_9$ .

O reator  $s_5$  não possui tanques depósitos de saída visto que, os lotes podem ser transferidos diretamente para os tanques de saída do sistema. Note que, a área de produção do reator é um recurso compartilhado, logo, somente poderá ser ocupada por um lote de cada vez. Além disso devemos sincronizar a abertura das válvulas conforme cada receita do sistema. Por exemplo, para transferir o suco de limão devemos abrir a válvula  $u_3$  e manter fechada a válvula  $u_4$ . Da mesma forma, para transferir o lote de refrigerante de limão devemos abrir a válvula  $u_6$  e manter fechada a válvula  $u_7$ . As receitas para a produção de bebidas são:

1. Receita  $RAN_1 = \{(u_1 \mapsto u_5 \mapsto u_8)\}$

2. Receita  $RBN_1$ :

- (a) O suco de limão deve ser transferido para o tanque  $u_c$  dentro de  $\tau_{1,1}$  unidades de tempo a uma taxa de  $\xi_{1,1}$  unidades de volume/unidades de tempo
- (b) A mistura água adoçante deve ser transferida para o tanque  $u_c$  dentro de  $\tau_{1,2}$  unidades de tempo a uma taxa de  $\xi_{1,2}$  unidades de volume/unidades de tempo

(c) O refrigerante de limão deve ser transferido para o tanque  $u_8$  dentro de  $\tau_{1,3}$  unidades de tempo a uma taxa de  $\xi_{1,3}$  unidades de volume/unidades de tempo

1. Receita  $RAN_2 = \{(u_2 \rightarrow u_5 \rightarrow u_9)\}$

2. Receita  $RBN_2$ :

(a) O suco de uva deve ser transferido para o tanque  $u_c$  dentro de  $\tau_{2,1}$  unidades de tempo a uma taxa de  $\xi_{2,1}$  unidades de volume/unidades de tempo

(b) A mistura água-adoçante deve ser transferida para o tanque  $u_c$  dentro de  $\tau_{2,2}$  unidades de tempo a uma taxa de  $\xi_{2,2}$  unidades de volume/unidades de tempo

(c) O refrigerante de uva deve ser transferido para o tanque  $u_9$  dentro de  $\tau_{2,3}$  unidades de tempo a uma taxa de  $\xi_{2,3}$  unidades de volume/unidades de tempo

### 5.1.2 Tanques Depósitos de Saída

Para exemplificar como pode ser utilizado os tanques depósitos de saída de um reator, considere o sistema de produção em lotes dos produtos A e B ilustrado na Figura 5.3, onde:

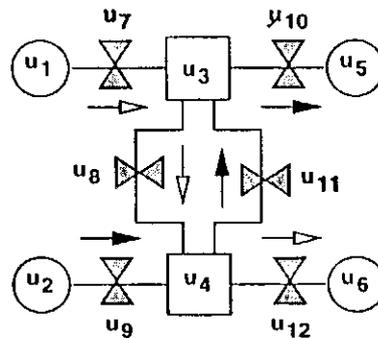


Figura 5.3: Exemplo de um SPL utilizando os tanques depósitos de saída

1.  $U_a = \{u_1, u_2, u_5, u_6\}$

2.  $U_p = \{u_3, u_4\}$

3.  $U_t = \{u_7, u_8, u_9, u_{10}, u_{11}, u_{12}\}$

4.  $L = \{lote_1, lote_2\}$  onde, o  $lote_1$  é do produto A e o  $lote_2$  é do produto B,

5.  $R = \{R_1, R_2\}$  onde,

(a)  $R_1 = \{\{u_1 \rightarrow u_3 \rightarrow u_4 \rightarrow u_6\}, RBN_1\}$

(b)  $R_2 = \{\{u_2 \rightarrow u_4 \rightarrow u_3 \rightarrow u_5\}, RBN_2\}$

6.  $S = \{s_3, s_4\}$

Os arcos de seta vazia ilustrados na Figura 5.3 indicam, com base na receitas  $RAN_1$ , o fluxo de produção para o  $lote_1$  e os arcos de seta cheia indicam, com base na receita  $RAN_2$ , o fluxo de produção para o  $lote_2$ . Os lotes podem ser produzidos da seguinte forma:

1. *Fase de Produção A*: A matéria prima do tanque  $u_1$  é transferida, por meio da válvula  $u_7$ , para a área de produção do reator  $s_3$  e processada. Após o processamento, o lote é então transferido para um tanque depósito (1) de saída do reator  $s_3$ .
2. *Fase de Produção B*: A matéria prima do tanque  $u_2$  é transferida, por meio da válvula  $u_9$ , para a área de produção do reator  $s_4$  e processada. Após o processamento, o lote é então transferido para um tanque depósito (2) de saída do reator  $s_4$ .
3. *Fase de Produção C*: O lote do tanque depósito de saída (1) do reator  $s_3$  é transferido, por meio da válvula  $u_8$ , para a área de produção do reator  $s_4$  e processado. Após o processamento, o lote do produto A é então transferido, por meio da válvula  $u_{12}$ , para o tanque de saída do sistema  $u_6$ .
4. *Fase de Produção D*: O lote do tanque depósito de saída (2) do reator  $s_4$  é transferido, por meio da válvula  $u_{11}$ , para a área de produção do reator  $s_3$  e processado. Após o processamento, o lote do produto B é então transferido, por meio da válvula  $u_{10}$ , para o tanque de saída do sistema  $u_5$ .

Se os reatores não possuem tanques depósitos de saída, não seria possível produzir os lotes pois,

1. A matéria prima contida no tanque  $u_1$  é transferida, por meio da válvula  $u_7$ , para a área de produção do reator  $s_3$  e processada.
2. A matéria prima contida no tanque  $u_2$  é transferida, por meio da válvula  $u_9$ , para a área de produção do reator  $s_4$  e processada.

3. O lote contido no reator  $s_3$  não poderá ser transferido, por meio da válvula  $u_8$ , para a área de produção do reator  $s_4$  visto que, o reator  $s_4$  está ocupado.
4. O lote contido no reator  $s_4$  não poderá ser transferido, por meio da válvula  $u_{11}$ , para a área de produção do reator  $s_3$  visto que, o reator  $s_3$  está ocupado.

Baseado nas considerações acima, sem os tanques depósitos de saída, o sistema poderia evoluir para um estado de bloqueio e toda produção seria perdida.

### 5.1.3 Tanques Depósitos de Entrada

Para exemplificar como podem ser utilizado os tanques depósitos de entrada de um reator, considere o sistema de produção em lotes dos produtos X e Y ilustrado na Figura 5.4, onde:

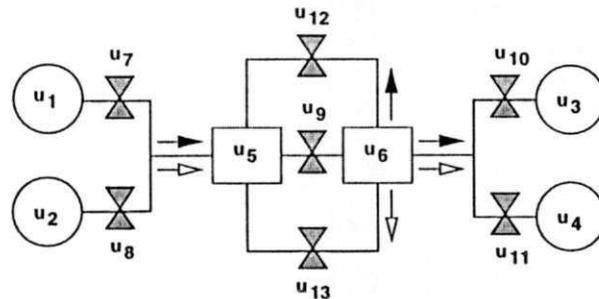


Figura 5.4: Exemplo de um SPL utilizando os tanques depósitos de entrada

1.  $U_a = \{u_1, u_2, u_3, u_4\}$
2.  $U_p = \{u_5, u_6\}$
3.  $U_t = \{u_7, u_8, u_9, u_{10}, u_{11}, u_{12}, u_{13}\}$
4.  $L = \{lote_1, lote_2\}$  onde, o  $lote_1$  é do produto X e o  $lote_2$  é do produto Y,
5.  $R = \{R_1, R_2\}$  onde,
  - (a)  $R_1 = \{\{u_1 \rightarrow u_5 \rightarrow u_6 \rightarrow (u_3 \text{ e } u_5)\}, RBN_1\}$
  - (b)  $R_2 = \{\{u_2 \rightarrow u_5 \rightarrow u_6 \rightarrow (u_4 \text{ e } u_5)\}, RBN_2\}$
6.  $S = \{s_5, s_6\}$

Os arcos de seta cheia ilustrados na Figura 5.3 indicam o fluxo de produção para o *lote*<sub>1</sub> e os arcos de seta vazia indicam o fluxo de produção para o *lote*<sub>2</sub>. A notação  $u_6 \rightarrow (u_3 \text{ e } u_5)$  indica que o lote produzido no reator  $s_6$  é transferido para os reatores  $s_3$  e  $s_5$ . O mesmo raciocínio é válido para a notação  $u_6 \rightarrow (u_4 \text{ e } u_5)$ . Neste exemplo, existe refluxo de produção do reator  $s_6$  para o reator  $s_5$ . Isto significa que, parte de um lote produzido no reator  $s_6$  é transferida de volta para o reator  $s_5$  para ser outra vez utilizada. Os lotes podem ser produzidos da seguinte forma:

1. *Fase de Produção A*: A matéria prima (*lote*<sub>1</sub>) do tanque  $u_1$  é transferida, por meio da válvula  $u_7$ , para a área de produção do reator  $s_5$  e processada com o material disponível do tanque depósito de entrada (1). Após o processamento, o lote é então transferido para o tanque depósito de saída (2),
2. *Fase de Produção B*: A matéria prima (*lote*<sub>2</sub>) do tanque  $u_2$  é transferida, por meio da válvula  $u_8$ , para a área de produção do reator  $s_5$  e processada com o material disponível do tanque depósito de entrada (3). Após o processamento, o lote é então transferido para o tanque depósito de saída (4),
3. *Fase de Produção C*: O lote do tanque depósito de saída (2) do reator  $s_5$  é transferido, por meio da válvula  $u_9$ , para a área de produção do reator  $s_6$  e processado. Após o processamento, o lote do produto  $X$  e o volume reutilizável são então transferidos, por meio da válvula  $u_{10}$  e  $u_{12}$ , para o tanque de saída  $u_3$  e de volta (refluxo) para o tanque depósito de entrada (1) do reator  $s_5$  respectivamente,
4. *Fase de Produção D*: O lote do tanque depósito de saída (4) do reator  $s_5$  é transferido, por meio da válvula  $u_9$ , para a área de produção do reator  $s_6$  e processado. Após o processamento, o lote do produto  $Y$  e o volume reutilizável são então transferidos, por meio da válvula  $u_{11}$  e  $u_{13}$ , para o tanque de saída  $u_4$  e de volta (refluxo) para o tanque depósito de entrada (3) do reator  $s_5$  respectivamente.

Na Figura 5.5 mostra-se o fluxo de produção dos reatores  $s_5$  e  $s_6$  em cada fase. Os arcos rotulados com o símbolo  $a$  representam o fluxo de produção da fase A, os arcos rotulados com o símbolo  $b$  representam o fluxo de produção da fase B e assim por diante.

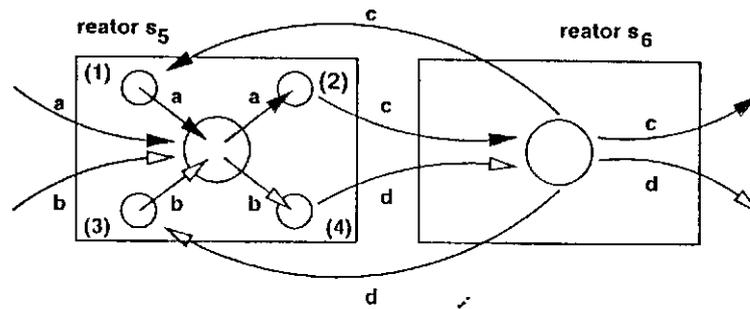


Figura 5.5: Fluxo de produção dos reatores ilustrado na Figura 5.4

## 5.2 Especificação do Supervisor

Nesta seção, apresentamos como modelar os sistemas de produção em lotes utilizando redes de Petri. O procedimento é o seguinte:

1. Modela-se o fluxo de produção de cada receita de alto nível do sistema,
2. Obtém-se o modelo do supervisor, por meio da fusão de lugares comuns a cada modelo obtido.

O modelo do supervisor possui lugares que representam as atividades de cada fase de produção, ordens de serviço ou disponibilidade dos recursos e transições que representam o início ou o fim destas atividades. Estes lugares e transições são do tipo:

Descrição dos lugares:

- $os_{i,j}$ : modela ordens de serviço do  $lote_i$  requisitando a ocupação do reator  $s_j$ ,
- $ose_i$ : modela ordens de serviço executadas do  $lote_i$ ,
- $pul_j$ : modela o recurso de produção  $u_j$  do reator  $s_j$  livre,
- $puo_{i,j}$ : modela uma fase de produção do  $lote_i$  sendo executada no reator  $s_j$ ,
- $pqi$ : modela a quantidade de ordens de serviço do  $lote_i$  a ser atendida.

Descrição das transições:

- $tio_{i,j}$ : modela o início de uma fase de produção do  $lote_i$  no reator  $s_j$ ,
- $tfo_{i,j}$ : modela o fim de uma fase de produção do  $lote_i$  no reator  $s_j$ .

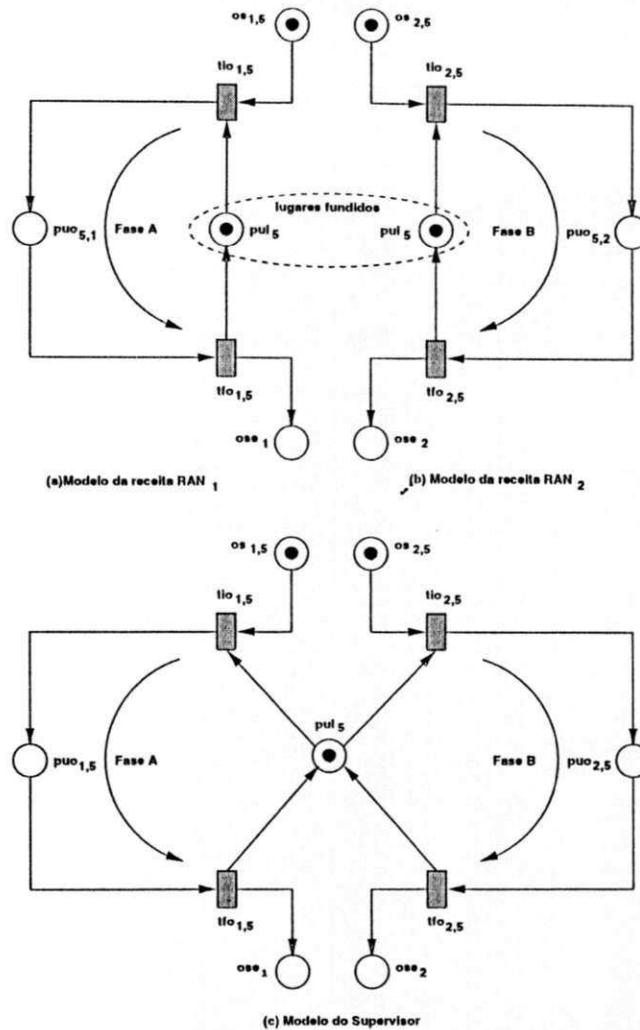


Figura 5.6: Modelo do supervisor para o sistema ilustrado na Figura 5.2

Para exemplificar o procedimento de modelagem, considere o sistema para a produção de bebidas ilustrado na Figura 5.2. Para este exemplo, os modelos das receitas  $RAN_1$  e  $RAN_2$  estão apresentados nas Figuras 5.6(a) e 5.6(b), respectivamente. Note que, o lugar  $pul_5$  está em ambos os modelos, logo devem ser fundidos. Fazendo a fusão destes lugares, obtivemos o modelo ilustrado na Figura 5.6(c). Entre ocorrências das transições  $tio_{1,5}$  e  $tfo_{1,5}$  a fase produção *A*, descrita na Seção 5.1.1, do  $lote_1$  é executada no reator  $s_5$ . Da mesma forma, entre as ocorrências das transições  $tio_{2,5}$  e  $tfo_{2,5}$  a fase de produção *B* do  $lote_2$  é executada.

Como um segundo exemplo, o modelo do supervisor, para o sistema da Figura 5.4, é mostrado na Figura 5.7. No modelo, os arcos de seta cheia estão associados ao fluxo de produção do  $lote_1$  e os arcos de seta vazia ao fluxo de produção do  $lote_2$ . Note que, na

marcação inicial os lugares  $pq_1$  e  $pq_2$  estão marcados, indicando que os tanques depósitos de entrada (1) e (3), ilustrados na Figura 5.5, estão inicialmente cheios. Ao término das fases de produção  $A$  ou  $B$  (ocorrências das transições  $tfo_{1,5}$  ou  $tfo_{2,5}$ ), estes são esvaziados e, devido ao refluxo de produção, enchidos ao término das fases de produção  $C$  ou  $D$  (ocorrência das transições  $tfo_{1,6}$  ou  $tfo_{2,6}$ ). Estas fases estão descritas na Seção 5.1.3. Portanto, somente podemos iniciar outra vez as fases  $A$  ou  $B$ , após o término das fases  $C$  ou  $D$ , respectivamente. Devido a este fato, os lugares  $pq_1$  e  $pq_2$  são lugares de entrada das transições  $tio_{1,5}$  e  $tio_{2,5}$  e lugares de saída das transições  $tfo_{1,6}$  e  $tfo_{2,6}$ , respectivamente. Note que, entre as ocorrências das transições ( $tio_{1,5}$  e  $tfo_{1,5}$ ) e ( $tio_{1,6}$  e  $tfo_{1,6}$ ) as fases de produção  $A$  e  $C$  do lote<sub>1</sub> são executadas nos reatores  $s_5$  e  $s_6$ , respectivamente. Da mesma forma, entre as ocorrências das transições ( $tio_{2,5}$  e  $tfo_{2,5}$ ) e ( $tio_{2,6}$  e  $tfo_{2,6}$ ) as fase de produção  $B$  e  $D$  do lote<sub>2</sub> são executadas nos reatores  $s_5$  e  $s_6$ , respectivamente.

A descrição dos lugares e transições dos modelos foi omitida pois, segue a mesma descrição feita no início desta seção.

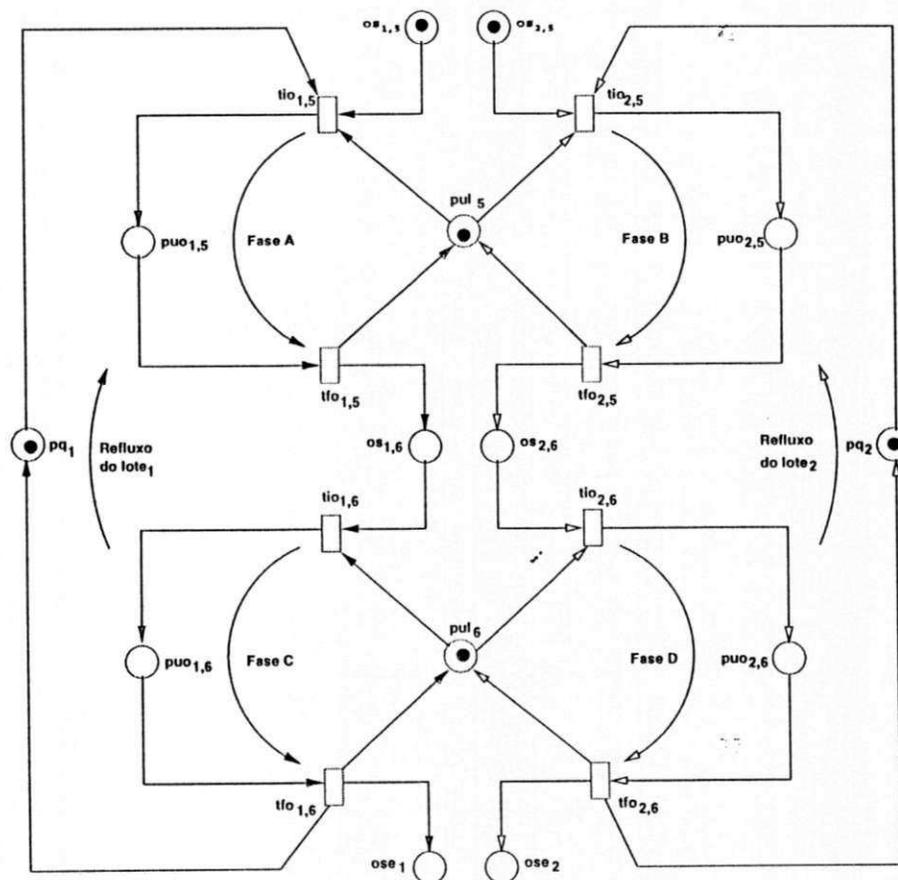


Figura 5.7: Modelo do supervisor para o sistema ilustrado na Figura 5.4

## 5.3 Especificação do Supervisor em Redes de Petri Coloridas

Nesta seção, tratamos da supervisão dos sistemas de produção em lotes utilizando as redes de Petri Coloridas. Para tanto, consideramos os conceitos de supervisão fixa e dinâmica do sistema. Diferente da supervisão fixa, a supervisão dinâmica conta com uma ou mais alternativas de produção.

### 5.3.1 Supervisão Fixa

Na seção 5.2, apresentamos um procedimento de modelagem sistemático para os sistemas de produção em lotes. Como no caso dos sistemas flexíveis de manufatura, os modelos obtidos possuem uma característica comum, ou seja, são formados por estruturas idênticas e que podem ser representadas pela estrutura de alto nível ilustrada na Figura 5.8 onde,

1. O lugar *OS1* desta estrutura representa todos os lugares do tipo  $os_{i,j}$ , o lugar *OS2* representa todos os lugares  $ose_i$  ou  $os_{i,k}$ , o lugar *PQ* representa todos os lugares  $pq_i$ , o lugar *PUL* representa todos os lugares  $pul_j$  e o lugar *PUO* representa todos os lugares  $puo_{i,j}$ ,
2. A transição *TIO* desta estrutura representa todas as transições do tipo  $tio_{i,j}$  e a transição *TFO* desta estrutura representa todas as transições do tipo  $tfo_{i,j}$ .

Note que, podemos utilizar a estrutura de alto nível para modelar qualquer sistema de produção em lotes definido no contexto desta tese e resolver conflitos por meio das informações contidas nas fichas.

### Descrição do Modelo do Supervisor

Para exemplificar o conceito de supervisão fixa, especificamos o supervisor para o sistema apresentado na Figura 5.3. Para tanto, utilizamos a estrutura de alto nível apresentada na Figura 5.8 e as informações contidas nas fichas para resolver conflitos. O modelo obtido, mostrado na Figura 5.9, possui lugares e transições com o seguinte significado:

Lugares:

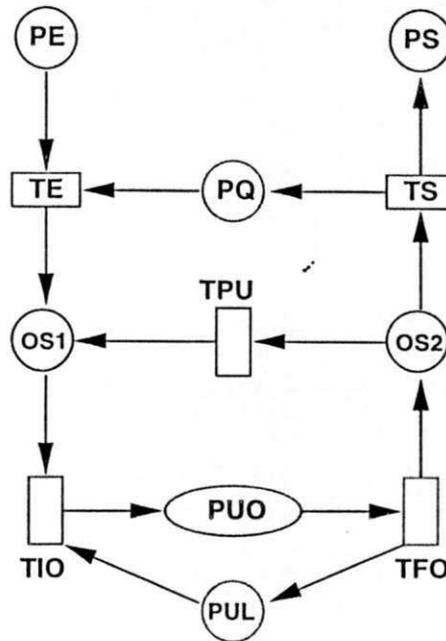


Figura 5.8: Estrutura de alto nível

1. *PE* e *OS1*: modelam ordens de serviço,
2. *OS2*: modela ordens de serviço executadas ou não,
3. *PS*: modela ordens de serviço executadas,
4. *PQ*: modela a quantidade de cada lote que pode ser produzida no sistema,
5. *PUO*: modela um recurso de produção ocupado,
6. *PUL*: modela um recurso de produção livre.

Transições:

1. *TE* e *TPU*: modelam o recebimento de ordens de serviço,
2. *TS*: modela o envio de ordens de serviço executadas,
3. *TIO*: modela o início da ocupação de um recurso de produção do sistema,
4. *TFO*: modela o fim da ocupação de um recurso de produção do sistema,

A marcação inicial do lugar *PUL* é  $u_3 + u_4$ , indicando que há dois recursos de produção disponíveis. A marcação inicial do lugar *PE* contém fichas  $(lote_i, [lista_i])$ , ou seja, ordens de serviço para a produção do  $lote_i$  onde,  $lista_i$  é a lista dos recursos de produção para produzir o  $lote_i$ . As expressões de arco são  $(t, (x :: List))$  ou  $(t, List)$  onde,  $t$  é igual a  $lote_i$ ,

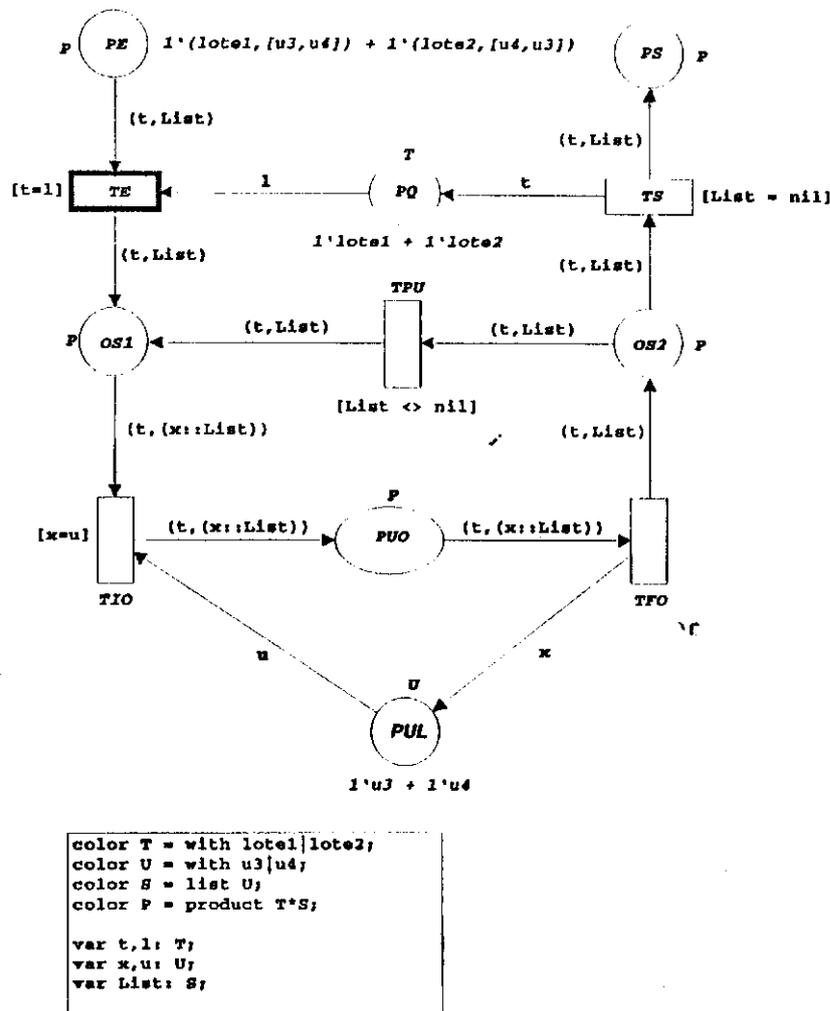


Figura 5.9: Modelo do supervisor para o sistema ilustrado na Figura 5.3

$x$  especifica um recurso de produção e  $List$  especifica a cauda da lista. Note que, devido a expressão de arco  $(t, List)$ , a cabeça da lista  $x$  é removida, após a ocorrência da transição  $TFO$ . Isto deve ser feito, pois requisições de serviço já atendidas devem ser removidas da lista. Desta forma, a marcação do lugar  $PS$  deve conter fichas  $(lote_i, [])$ , ou seja, ordens de serviço executadas do  $lote_i$  onde,  $[]$  representa a lista vazia. Note que, as transições  $TS$  e  $TPU$  estão em conflitos. Este conflito surge porque uma ordem de serviço pode estar ou não executada. Este conflito é resolvido associando expressões de guarda  $[List = nil]$  e  $[List \neq nil]$  a estas transições. Se a lista está vazia ( $[List = nil]$ ), a transição  $TS$  ocorre indicando que todas as requisições de serviço foram atendidas. Se a lista não está vazia  $[List \neq nil]$  a transição  $TPU$  é ocorre indicando que há requisições de serviço ainda pendentes.

### 5.3.2 Supervisão Dinâmica

Nesta seção, tratamos da supervisão dinâmica dos sistemas de produção em lotes. Para exemplificar a supervisão dinâmica, considere o sistema de produção em lotes ilustrado na Figura 5.10, onde:

1.  $U_p = \{u_3, u_4, u_5, u_6\}$ ,
2.  $L = \{lote_1, lote_2\}$ ,
3.  $R = \{R_1, R_2\}$  onde,
  - (a)  $R_1 = \{\{u_1 \rightarrow u_3, u_3 \rightarrow (u_5 \text{ ou } u_6), u_5 \rightarrow u_7, u_6 \rightarrow u_9\}, RBN_1\}$ .
  - (b)  $R_2 = \{\{u_2 \rightarrow u_4, u_4 \rightarrow (u_5 \text{ ou } u_6), u_5 \rightarrow u_8, u_6 \rightarrow u_{10}\}, RBN_1\}$ .

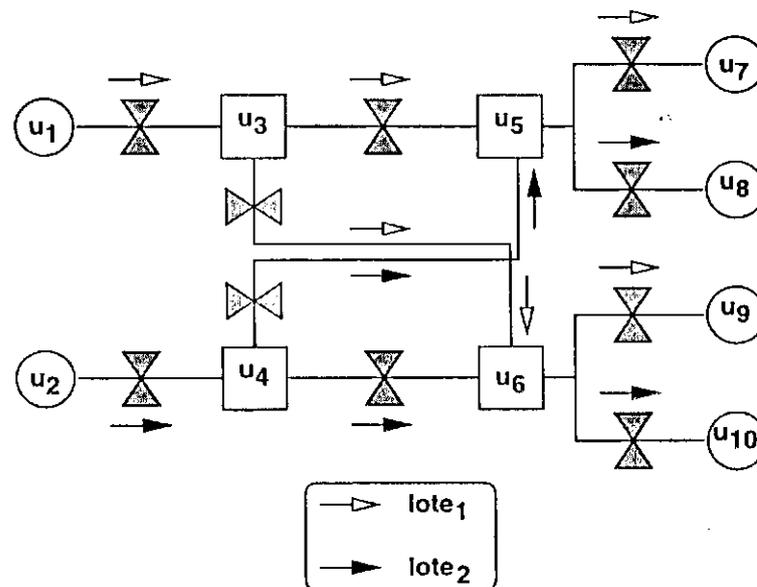


Figura 5.10: SPL utilizado para exemplificar a supervisão dinâmica

Os arcos de seta vazia e seta cheia representam o fluxo de produção dos *lote1* e *lote2*, respectivamente. Note que, os lotes podem ocupar o reator  $s_5$  ou o reator  $s_6$ . Por exemplo, suponha a seguinte situação: um *lote* está pronto para ser transferido do reator  $s_3$ , o reator  $s_5$  está ocupado e o reator  $s_6$  está livre. Portanto, desde que o reator  $s_6$  está livre, este poderá ser ocupado pelo *lote* do reator  $s_3$ . A vantagem da supervisão dinâmica com relação a fixa é que podemos aproveitar a disponibilidade dos recursos de produção do sistema. Para esta mesma situação, se a supervisão fosse fixa, o supervisor teria que esperar a

desocupação do reator  $s_5$  para transferir o *lote*. Isto porque, no caso fixo o supervisor não possui alternativas de produção. Por exemplo, a notação  $u_3 \mapsto (u_5 \text{ ou } u_6)$  indica alternativas para a produção do *lote*<sub>1</sub>, ou seja, saindo do reator  $s_3$  este pode ocupar o reator  $s_5$  ou o reator  $s_6$ . Para lidar com o caso dinâmico, o supervisor deve reconhecer as alternativas e tomar decisões baseado na disponibilidade dos recursos.

### Descrição da Página *fluxo de ordens de serviço*

Conforme é mostrado na Figura 5.11, o modelo possui duas páginas<sup>1</sup> onde, a primeira página, denominada *fluxo de ordens de serviço*, modela o controle do fluxo de ordens de serviço e a outra página, denominada *alternativas*, modela a supervisão dinâmica do sistema. A página *fluxo de ordens de serviço*, apresentada na Figura 5.12, possui lugares e transições com o seguinte significado:

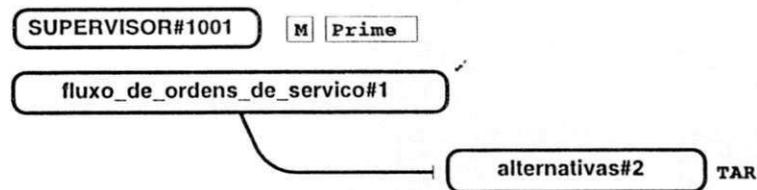


Figura 5.11: Páginas do supervisor <sup>10</sup>

Descrição dos lugares:

*PE e OS1*: modelam ordens de serviço,

*OS2*: modela ordens de serviço executadas ou não,

*PS*: modela ordens de serviço executadas,

*PQ*: modela a quantidade de cada lote que pode ser produzida no sistema,

Descrição das transições:

*TE e TPU*: modelam o recebimento de ordens de serviço,

*TS*: modela o envio de ordens de serviço executadas,

*TAR*: modela o fim da ocupação de um recurso de produção do sistema,

<sup>1</sup>O conceito de páginas de um modelo CPN hierárquico está apresentado na Seção 2.4.2.

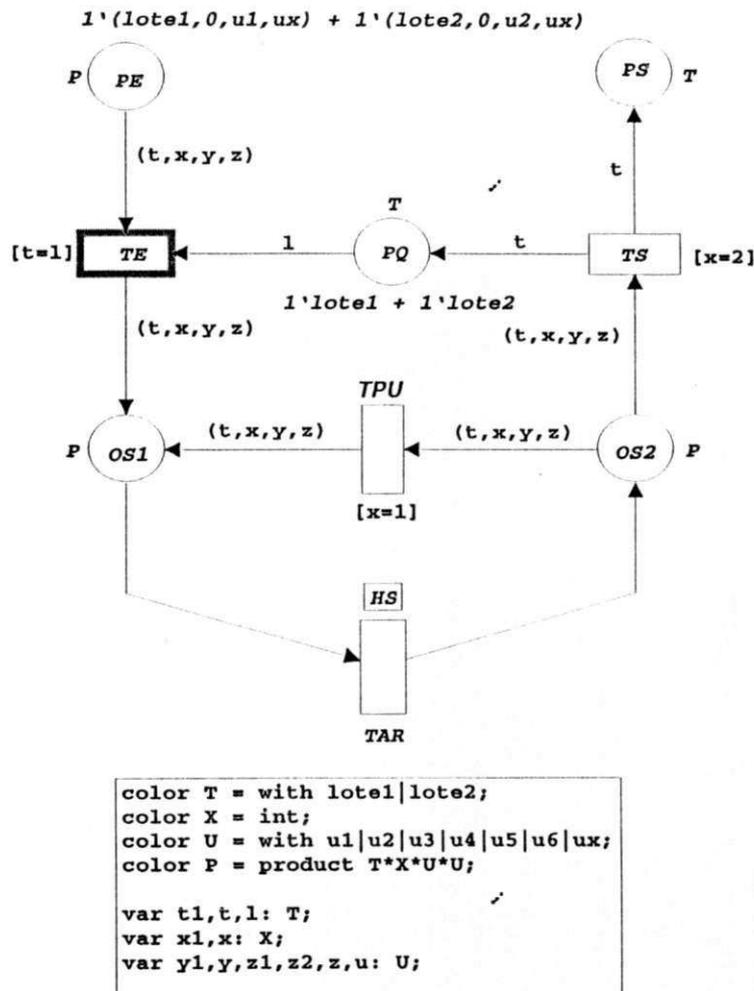


Figura 5.12: Página *fluxo de ordens de serviço*

Na marcação inicial, o lugar *PE* contém fichas especificando os lotes a serem produzidos, um número inteiro, representado pela variável  $x$ , para resolver o conflito entre as transições *TPU* e *TS*, os recursos fonte e destino. Este conflito é resolvido associando as expressões de guarda  $[x = 1]$  e  $[x = 2]$  a estas transições. Se o valor de  $x$  em uma ficha do lugar *OS2* for igual a 1, a transição *TPU* ocorre, indicando que a ficha é uma ordem de serviço não executada. Se o valor de  $x$  em um ficha do lugar *OS2* for igual a 2, a transição *TS* ocorre, indicando que a ficha é uma ordem de serviço executada. Já a marcação do inicial do lugar *PQ* modela a quantidade de lotes que pode ser produzida no sistema.

Na Figura 5.12 mostra-se a marcação inicial desta página para o exemplo apresentado na Figura 5.10 assim como, a declaração de variáveis e cores. O lugar *PE* contém duas fichas, ou seja, duas ordens de serviço, uma para a produção do *lote<sub>1</sub>* e outra para a produção

do  $lote_2$ . O valor da variável  $x$  inicialmente é zero, indicando que as fichas são ordens de serviço não executadas. A origem do  $lote_1$  é o tanque  $u_1$  e a origem do  $lote_2$  é o tanque  $u_2$ . A marcação do lugar  $PQ$  indica que apenas um lote pode ser produzido de cada vez.

### Descrição da Página *alternativas*

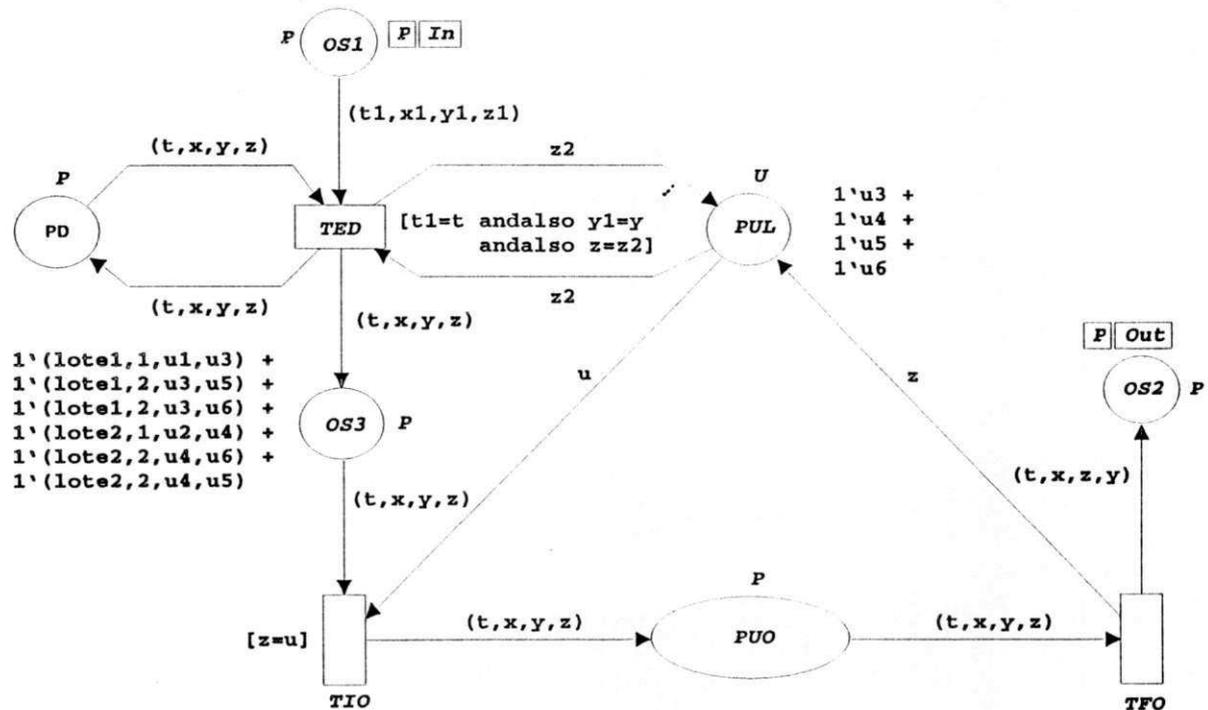


Figura 5.13: Página *alternativas*

A página *alternativas* está apresentada na Figura 5.13. Nesta página, os lugares e transições são os seguintes:

Descrição dos lugares:

$OS1$  e  $OS3$ : modelam ordens de serviço,

$OS2$ : modela as ordens de serviço executadas ou não,

$PD$ : modela as alternativas de produção,

$PU0$ : modela os recursos de produção ocupados,

$PUL$ : modela os recursos de produção livres.

Descrição das transições:

*TED*: modela a escolha das alternativas de produção,

*TIO*: modela o início da ocupação de um recurso de produção do sistema,

*TFO*: modela o fim da ocupação de um recurso de produção do sistema,

Na marcação inicial, o lugar *PD* contém fichas representando as alternativas para a produção de cada lote. Para o sistema apresentado na Figura 5.10, temos três fichas alternativas para o *lote*<sub>1</sub> e três fichas alternativas para o *lote*<sub>2</sub>. Na marcação inicial, o lugar *PUL* contém fichas representando os recursos de produção do sistema. Para o exemplo considerado esta marcação é  $u_3 + u_4 + u_5 + u_6$ .

Note que, a transição *TED* somente pode ocorrer se para cada lote alocado em um recurso fonte, o recurso de produção destino escolhido deste lote esteja livre. Neste caso, a ocorrência desta transição remove fichas do lugar *OS1* e adiciona fichas no lugar *OS3* do modelo.

## 5.4 Alocação dos Recursos de Manutenção

Periodicamente, é feita a *manutenção preventiva* dos recursos do sistema visando prevenir eventuais quebras de partes consideradas críticas para o correto funcionamento do sistema. A troca de partes é feita obedecendo datas e nestas datas os recursos podem ficar indisponíveis. Após o término da manutenção de cada recurso, as datas e o estoque de material devem ser atualizadas. Para o caso dos sistemas de produção em lotes, um reator pode estar parado para fazer concertos no seu tanque principal, a troca de juntas para prevenir possíveis vazamentos, etc. Apesar da manutenção preventiva, a falha de um recurso é inevitável. Isto porque, por questões de custos, todas as partes de um recurso não são trocadas e também aquelas que foram trocadas na manutenção preventiva podem eventualmente quebrar. Em todo caso, a manutenção preventiva pode reduzir a ocorrência de faltas mas não pode evitá-las.

### 5.4.1 Arquitetura do Controle Considerando os Recursos de Manutenção

Na Figura 5.14 mostra-se a arquitetura do controle hierárquico considerando a alocação dos recursos de manutenção do sistema. Para o caso da manutenção preventiva, a alocação

pode ser feita como detalhado a seguir.

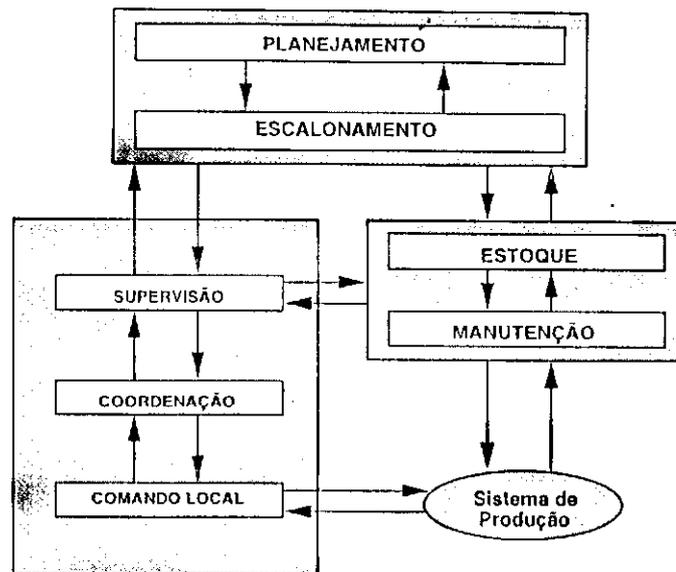


Figura 5.14: Arquitetura considerando a alocação dos recursos de manutenção do sistema

1. O controle de planejamento/escalonamento do sistema, obedecendo datas e o estoque de material, informa ao supervisor que um recurso deve ser entregue à manutenção preventiva,
2. O supervisor envia uma requisição solicitando a manutenção preventiva do referido recurso,
3. Atendida a requisição de serviço, o supervisor informa ao controle de planejamento/escalonamento do sistema, que a manutenção preventiva foi executada.

Entretanto, a alocação da manutenção corretiva é um indicativo da ocorrência de faltas. Para ser tolerante a estas faltas, um sistema de produção em lotes deve possuir redundância de seus componentes. Para exemplificar este fato, considere que o reator, ilustrado na Figura 5.15(a), possui um agitador de mistura dentro de seu tanque principal (1) e que o eixo do agitador quebrou durante o processamento de um lote. Neste caso, para fazer o concerto do eixo, o lote pode ser transferido, sob o controle do gerenciador local, para o tanque réplica (2) do reator ilustrado na 5.15(b). Após o término do concerto, o tanque principal pode ser colocado novamente em operação.

Portanto, para o caso da manutenção corretiva, a alocação pode ser feita da seguinte forma:

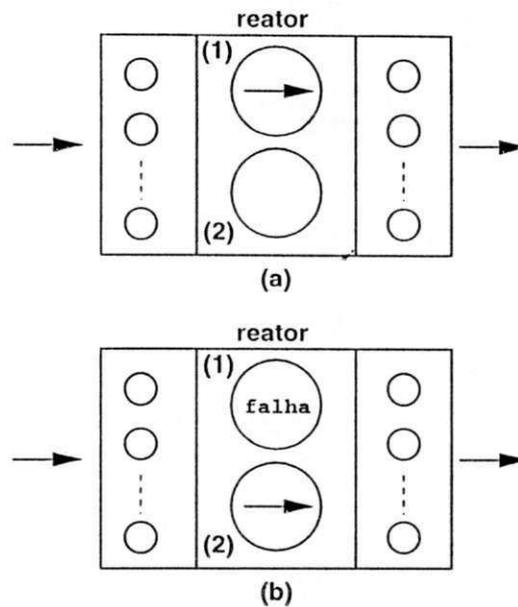


Figura 5.15: Reator tolerante a falhas

1. Ocorrendo a falha de um recurso, o gerenciador local informa ao supervisor, que o recurso deve ser entregue à manutenção corretiva,
2. O supervisor envia uma requisição de serviço solicitando a manutenção corretiva do referido recurso,
3. Atendida a requisição de serviço, o supervisor informa ao gerenciador local, que a manutenção corretiva foi executada.

A seguir, utilizando a supervisão dinâmica, apresentamos a especificação do supervisor considerando a alocação dos recursos de manutenção do sistema.

#### 5.4.2 Descrição do Modelo do Supervisor

O modelo do supervisor é o mesmo descrito na Seção 5.3.2 exceto que, a página *alternativas* apresenta os seguintes lugares e transições:

Descrição dos lugares:

- P1*: modela ordens de serviço requisitando a alocação da manutenção preventiva,
- P2*: modela ordens de serviço executada,
- P3*: modela ordens de serviço requisitando a alocação da manutenção corretiva,

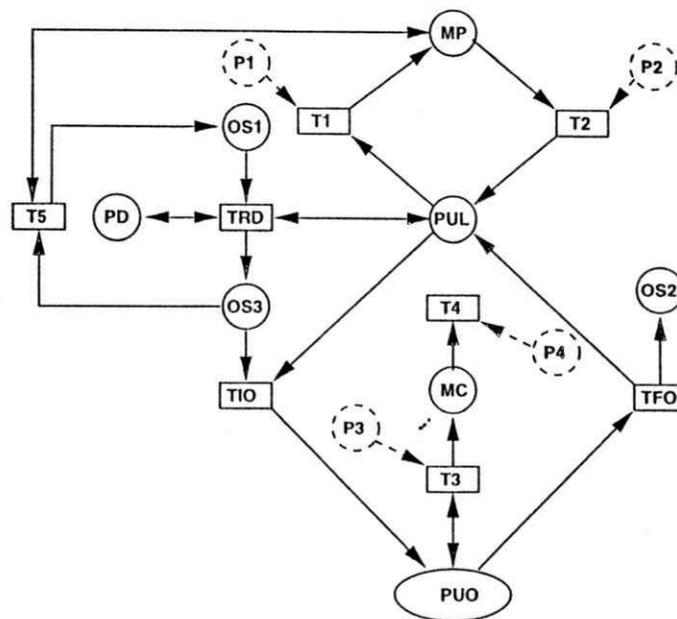


Figura 5.16: Página *alternativas* modificada

*P4*: modela ordens de serviço executada,

*MP*: modela os recursos entregues a manutenção preventiva,

*MC*: modela os recursos entregues a manutenção corretiva,

Descrição das transições:

*T1*: modela o início da manutenção preventiva,

*T2*: modela o término da manutenção preventiva,

*T3*: modela o início da manutenção corretiva,

*T4*: modela o término da manutenção corretiva,

*T5*: modela alterações das ordens de serviço.

Observe que, a ocorrência da transição *T5* remove fichas do lugar *OS3* e adiciona fichas no lugar *OS1*. Isto significa que, as ordens de serviço do lugar *OS3* devem ser modificadas. Note que, a marcação dos lugares *MP* e *MC* representam os recursos que estão em manutenção preventiva e corretiva, respectivamente. Na verdade, as informações representadas por fichas destes lugares podem ser usadas nos processos de decisões dos níveis superiores de controle do sistema.

## 5.5 Funcionalidade e Análise do Supervisor

Ao longo deste capítulo, tratamos da supervisão dos sistemas de produção em lotes empregando redes de Petri. Em cada modelo obtido, fichas nos lugares representam ordens de serviço ou atividades e a ocorrência de transições representam o início e o término destas atividades.

O envio de requisições de serviço está sincronizado com a ocorrência de transições que representam o início de atividades. Já a ocorrência das transições que representam o fim das atividades, está sincronizado com o recebimento de requisições de serviço atendidas. Desta forma, o envio de uma requisição de serviço está sincronizado com a ocorrência da transição *TIO*. Após o atendimento da requisição de serviço, a transição *TFO* então ocorre, veja Figuras 5.9 e 5.13.

Para exemplificar a funcionalidade do supervisor<sup>2</sup>, considere o sistema apresentado na Figura 5.4 e o modelo da Figura 5.9. Neste caso, a marcação do lugar *PE* seria  $(lote_1, [u_5, u_6]) + (lote_2, [u_5, u_6])$  e a do lugar *PUL* seria  $u_5 + u_6$ . Suponha que, após a ocorrência da transição *TE* a marcação do lugar *OS1* seja  $(lote_1, [u_5, u_6])$  logo, a transição *TIO* é habilitada. Ocorrendo esta transição, uma requisição de serviço é enviada ao gerenciador local do reator  $s_5$  solicitando a execução da fase de produção *A*, veja descrição das fases na Seção 5.1.3. Terminada esta fase, a transição *TFO* ocorre e a cabeça da lista  $u_5$  é removida da lista. Neste caso, a marcação do lugar *OS2* seria  $(lote_1, [u_6])$ , onde, somente a transição *TPU* está habilitada pois, a lista não está vazia. Ocorrendo esta transição, o lugar *OS1* é marcado possibilitando assim outra ocorrência da transição *TIO*. Ocorrendo esta transição, uma requisição de serviço é enviada o gerenciador local do reator  $u_6$  solicitando a execução da fase de produção *C*. Terminada esta fase, a transição *TFO* ocorre e a cabeça da lista  $u_6$  é removida da lista. Neste caso, a marcação do lugar *OS2* é  $(lote_1, [])$  onde, somente a transição *TS* está habilitada pois, a lista está vazia. Ocorrendo esta transição, o lugar *PS* é marcado indicando que uma ordem de serviço foi executada. Além disso, o lugar *PQ* é marcado liberando assim a produção de outro lote. Note que, as fases de produção podem ser executadas concorrentemente. Por exemplo, a fase de produção *C* pode ser executada durante a execução da fase de produção *B*. A alocação dos recursos de manutenção do sistema pode ser feita da seguinte forma, veja Figura 5.16:

<sup>2</sup>A funcionalidade do supervisor está baseada nos conceitos apresentados na Seção 1.3

*Manutenção Preventiva*

1. O controle de planejamento/escalonamento do sistema, informa ao supervisor que um recurso deve ser entregue á manutenção preventiva (Marcação do lugar  $P1$ ),
2. O supervisor envia uma requisição de serviço, solicitando a manutenção preventiva do referido recurso (Ocorrência da transição  $T1$ ),
3. Atendida a requisição de serviço, a manutenção informa ao supervisor que o serviço foi executado (Marcação do lugar  $P2$ ),
4. O supervisor informa ao controle de planejamento/escalonamento do sistema que a manutenção preventiva foi executada (Ocorrência da transição  $T2$ ).

*Manutenção Corretiva*

1. O gerenciador local de um recurso informa ao supervisor que um recurso deve ser entregue á manutenção corretiva (Marcação do lugar  $P3$ ),
2. O supervisor envia uma requisição de serviço, solicitando a manutenção corretiva do referido recurso (Ocorrência da transição  $T3$ ),
3. Atendida a requisição de serviço, a manutenção informa ao supervisor que o serviço foi executado (Marcação do lugar  $P4$ ),
4. O supervisor informa ao gerenciador local do recurso que a manutenção corretiva foi executada (Ocorrência da transição  $T4$ ).

Para construir e analisar os modelos obtidos, utilizamos também a ferramenta *Design/CPN*. Por exemplo, para analisar o modelo descrito na Seção 5.3.2, assumimos a produção de dois lotes e que o reator  $s_5$  está indisponível. Para simular esta situação, retiramos a ficha, representante deste reator, do lugar  $PUL$ . Para este cenário, o grafo de ocorrência possui 99 estados, 174 arcos e que foi construído completamente. Por outro lado, o estado final é uma marcação de bloqueio. Observando a marcação do lugar  $PS$  neste estado, concluímos que, todas as ordens de serviço foram executadas. Para comprovar este fato, compare a marcação deste lugar nos estados inicial e final ilustrados na Figura 5.17.

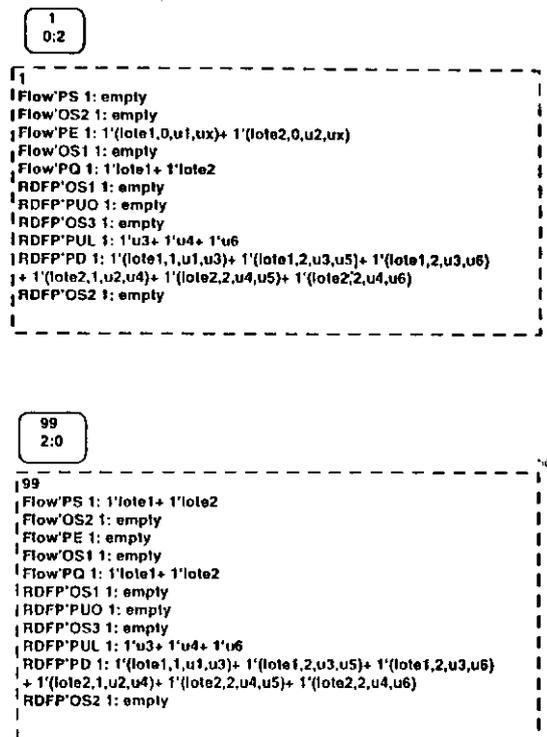


Figura 5.17: Marcação inicial e final do modelo apresentado na seção 5.3.2

Portanto, devido a supervisão dinâmica e sem contar com o reator  $s_5$ , os lotes foram produzidos. A mesma conclusão, considerando-se a indisponibilidade do reator  $s_6$ , pode ser obtida. De forma semelhante foi feita a análise considerando a supervisão fixa.

## 5.6 Considerações Finais

Neste capítulo tratamos da supervisão fixa e dinâmica dos sistemas de produção em lotes. Diferente da supervisão fixa, na supervisão dinâmica o supervisor conta com alternativas de produção e o processo de escolha depende da disponibilidade dos recursos. Com base no conceito de supervisão dinâmica, especificamos o supervisor considerando a alocação dos recursos de manutenção do sistema. Neste capítulo, contribuimos apresentando modelos cuja estrutura pode ser empregada para modelar qualquer sistema de produção em lotes definido no contexto desta tese.

## Capítulo 6

# Verificação de Receitas para Sistemas de Produção em Lotes

Neste capítulo, tratamos da verificação de receitas para sistemas de produção em lotes. Isto porque, o volume de um lote a ser produzido pode exceder a capacidade de um reator do sistema. Caso exceda, ou seja, ocorrer *transbordamentos*, parte do lote é perdida. Para tratar com este problema empregamos aqui a teoria das redes de Petri e das equações diferenciais [BP98b]. Com base nas receitas, obtemos o modelo híbrido do sistema, ou seja, o modelo de rede de Petri temporizado associado as equações diferenciais, veja Figura 6.1. Uma vez obtido o modelo este é analisado. No escopo desta tese, o objetivo da análise é provar se as receitas são limitadas para o sistema de produção apresentado. Com base nos resultados da análise, as receitas ou eventualmente os recursos podem ser ajustados para evitar os transbordamentos. Para analisar o modelo híbrido apresentamos algoritmos para o cálculo e verificação de capacidade. Os resultados apresentados são exemplificados através da aplicação a um sistema de produção de bebidas.

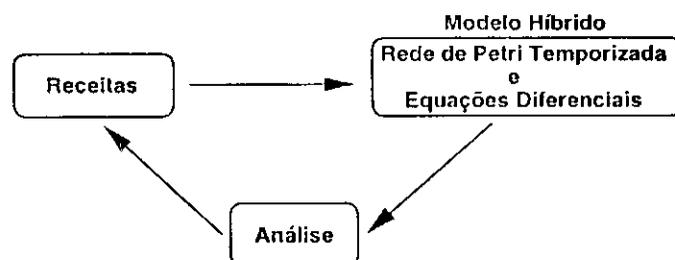


Figura 6.1: Utilização das redes de Petri para tratar com o problema da verificação de receitas

## 6.1 Determinação dos Estados Estáveis do Sistema

Os estados discretos de um sistema de produção em lotes mudam devido a ocorrência de eventos. Para cada estado alcançado, o sistema evolui dinamicamente no tempo obedecendo certas leis físicas descritas por um conjunto de equações diferenciais da forma  $\dot{v} = \xi(t)$  onde,  $v$  é a variável de estado contínua representando o volume de um lote e  $\xi(t)$  é o campo vetor da equação. Os eventos ocorrem assincronamente no tempo. Suponha que os eventos  $e_a$  e  $e_b$  ocorram nos instantes de tempo  $t_a$  e  $t_b = t_a + \Delta$ , respectivamente. Note que, durante o intervalo  $[t_a, t_b)$  o sistema não muda de estado. Neste caso, dizemos que o sistema entrou em um estado estável naquele intervalo de tempo.

Para exemplificar o conceito de estados estáveis, suponha um sistema, denominado daqui por diante por *sistema tanque-válvulas*, formado por uma válvula de enchimento  $q_1$ , um tanque inicialmente vazio e uma válvula de esvaziamento  $q_2$ , veja Figura 6.2. A taxa de variação de volume no tanque é  $\dot{v}$ . As especificações para este sistema são:

1. Tempo de transferência do lote para o tanque: 5 unidades de tempo,
2. Tempo de transferência do lote do tanque: 5 unidades de tempo,
3. Tempo de espera do lote para ser transferido do tanque: 2 unidades de tempo,
4. Taxa de transferência do lote para o tanque:  $\xi_1$  unidades de volume/unidades de tempo,
5. Taxa de transferência do lote do tanque:  $-\xi_2$  unidades de volume/unidades de tempo.
6. A capacidade máxima do tanque é  $v_c = 6k$  unidades de volume.

Conforme ilustrado na Figura 6.2, o sistema tanque-válvulas possui cinco estados, descritos a seguir:

1. Estado inicial  $E_0$ : o lote está aguardando ser transferido para o tanque (Figura 6.2(a)). Neste estado as válvulas  $q_1$  e  $q_2$  estão fechadas e  $\dot{v} = 0$ .
2. Estado  $E_1$ : o lote está sendo transferido para o tanque (Figura 6.2(b)). Neste estado a válvula  $q_1$  está aberta e a válvula  $q_2$  está fechada e  $\dot{v} = \xi_1$ .

3. Estado  $E_2$ : o lote está aguardando ser transferido do tanque (Figura 6.2(c)). Neste estado as válvulas  $q_1$  e  $q_2$  estão fechadas e  $\dot{v} = 0$ .
4. Estado  $E_3$ : o lote está sendo transferido do tanque (Figura 6.2(d)). Neste estado a válvula  $q_1$  está fechada e a válvula  $q_2$  está aberta e  $\dot{v} = -\xi_2$ .
5. Estado final  $E_4$ : o lote foi transferido do tanque (Figura 6.2(e)). Neste estado as válvulas  $q_1$  e  $q_2$  estão fechadas e  $\dot{v} = 0$ .

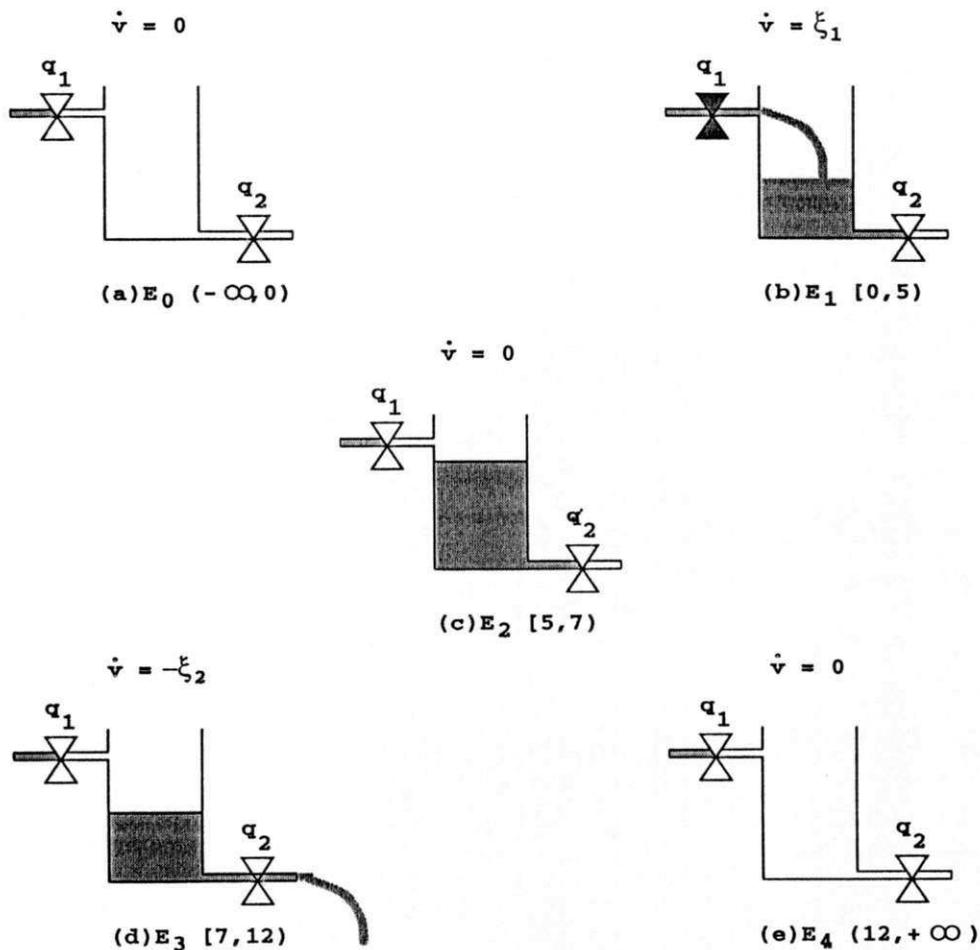


Figura 6.2: Sistema tanque-válvulas

Note que, o estado  $E_1$  é estável no intervalo  $[0, 5)$ , o estado  $E_2$  é estável no intervalo  $[5, 7)$  e o estado  $E_3$  é estável no intervalo  $[7, 12)$ . O estado  $E_0$  é estável no intervalo  $(-\infty, 0)$  e o estado  $E_4$  em  $[12, +\infty)$ . A mudança do estado  $E_0$  para o estado  $E_1$  ocorre devido à ocorrência, em  $t = 0$ , do evento *abrir válvula  $q_1$*  e do estado  $E_1$  para o estado  $E_2$  devido à ocorrência, no instante  $t = 5$ , do evento *fechar válvula  $q_1$* . A mudança do estado  $E_2$  para o

estado  $E_3$  ocorre devido à ocorrência, em  $t = 7$ , do evento *abrir válvula*  $q_2$  e do estado  $E_3$  para o estado  $E_4$  devido à ocorrência, no instante  $t = 12$ , do evento *fechar válvula*  $q_2$ .

Visto que, o sistema muda de estado devido à ocorrência de eventos, os campos vetores das equações devem ser alterados conforme a ocorrência destes eventos. Por exemplo, a ocorrência do evento *abrir válvula*  $q_1$ , altera o campo vetor da equação de zero para  $\xi_1$ , ou seja,  $\dot{v} = 0 \rightarrow \xi_1$ . A ocorrência do evento *fechar válvula*  $q_1$ , altera o campo vetor da equação de  $\xi_1$  para zero, ou seja,  $\dot{v} = \xi_1 \rightarrow 0$ . A ocorrência do evento *abrir válvula*  $q_2$ , altera o campo vetor da equação de zero para  $-\xi_2$ , ou seja,  $\dot{v} = 0 \rightarrow -\xi_2$  e finalmente a ocorrência do evento *fechar válvula*  $q_2$ , altera o campo vetor da equação de  $-\xi_2$  para zero, ou seja,  $\dot{v} = -\xi_2 \rightarrow 0$ . A seguir apresentamos algumas definições relativas aos estados do sistema e ao estado do modelo de redes de Petri

**Definição 6.1** Denota-se  $S_E = \{E_0, E_1, \dots, E_i, \dots, E_n\}$  o conjunto finito e não vazio dos estados estáveis do sistema.

Para o sistema tanque-válvulas este conjunto é  $S_E = \{E_0, E_1, E_2, E_3, E_4\}$ . As marcações de uma rede de Petri representam os estados do sistema. Uma marcação é dita estável se esta representa um estado estável do sistema logo,

**Definição 6.2** Denota-se  $S_M = \{M_0, M_1, \dots, M_i, \dots, M_n\}$  o conjunto finito e não vazio das marcações estáveis onde,  $M_i$  representa o estado estável  $E_i \in S_E$ .

Para exemplificar o conceito de marcações estáveis, considere ainda o sistema tanque-válvulas ilustrado na Figura 6.2. O modelo deste sistema, apresentado na Figura 6.3, é uma rede de Petri temporizada, onde, os atrasos estão associados aos lugares. Neste modelo, fichas nos lugares e ocorrência de transições possuem o seguinte significado.

Lugares:

$p_0$ : modela que o lote está pronto para ser transferido para o tanque,

$p_1$ : modela que o lote está sendo transferido para o tanque (atraso de 5 unidades de tempo),

$p_2$ : modela que o lote está aguardando transferência (atraso de 2 unidades de tempo),

$p_3$ : modela que o lote está sendo transferido do tanque (atraso de 5 unidades de tempo),

$p_5$ : modela que o lote foi transferido do tanque.

Transições:

$\lambda_1$  e  $\lambda_2$ : modelam a abertura e o fechamento da válvula  $q_1$ , respectivamente,

$\lambda_3$  e  $\lambda_4$ : modelam a abertura e o fechamento da válvula  $q_2$ , respectivamente.

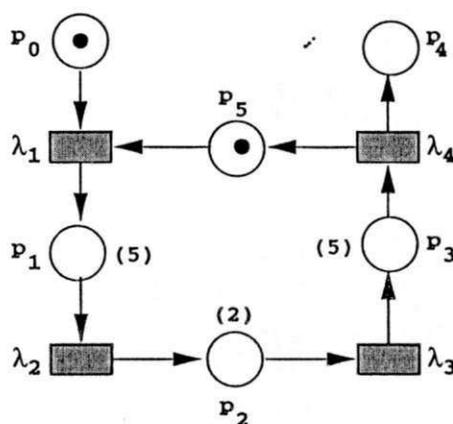


Figura 6.3: Modelo do sistema tanque-válvulas

As marcações estáveis são:

$$S_M = \{M_0, M_1, M_2, M_3, M_4\}$$

onde,  $M_0 = (100001)^T$ ,  $M_1 = (010000)^T$ ,  $M_2 = (001000)^T$ ,  $M_3 = (000100)^T$  e  $M_4 = (000011)^T$ . Estas marcações estão apresentadas na Figura 6.4. Note que,  $M_0$  é estável no intervalo  $(-\infty, 0)$ ,  $M_1$  é estável no intervalo  $[0, 5)$ ,  $M_2$  é estável no intervalo  $[5, 7)$ ,  $M_3$  é estável no intervalo  $[7, 12)$  e  $M_4$  é estável no intervalo  $[12, +\infty)$ . A ocorrência da transição  $\lambda_1$  transforma a marcação inicial  $M_0$  na marcação  $M_1$ , a sequência de ocorrências de transições  $\lambda_1\lambda_2$  transforma a marcação inicial  $M_0$  na marcação  $M_2$ , e assim por diante. Os elementos de  $S_M$  podem ser obtidos por meio da equação de estado

$$M_i = M_0 + A^T x_i \quad (6.1)$$

onde,  $A^T$  é a matriz de incidência e  $x_i$  é o vetor ocorrência da seqüência de transições que transforma a marcação inicial  $M_0$  na marcação  $M_i$ . Por exemplo, o vetor  $x_1$  para a ocorrência da transição  $\lambda_1$  é  $x_1 = (1000)^T$  logo,  $M_1 = M_0 + A^T x_1$ . Podemos calcular as demais marcações utilizando o mesmo procedimento.

**Definição 6.3** Denota-se  $S_x = \{x_1, \dots, x_i, \dots, x_n\}$  o conjunto finito e não vazio dos vetores de ocorrência das seqüências de transições, onde:  $M_0[x_i]M_i$  e  $M_0, M_i \in S_M$ .

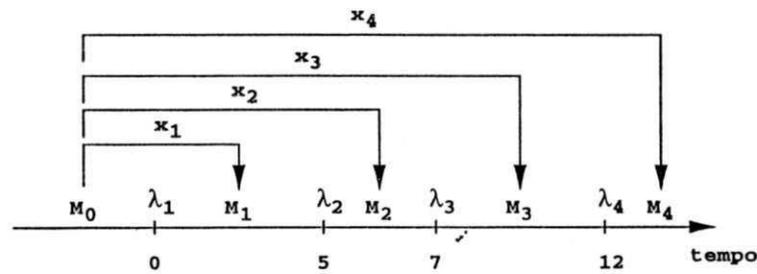


Figura 6.4: Representação das marcações estáveis do sistema tanque-válvulas

Visto que  $M_1$  é estável no intervalo  $[0, 5)$ , o vetor  $x_1$  é utilizado para calcular  $M_1$  dentro deste intervalo isto é, resolvendo a Equação (6.1). Da mesma forma, o vetor  $x_2$  é utilizado para calcular  $M_2$  dentro do intervalo  $[5, 7)$ , e assim por diante.

**Definição 6.4** Denota-se função ocorrência de transições a função  $f : [0, +\infty) \rightarrow S_x$ .

Baseado nesta definição, a função ocorrência para o sistema tanque- válvulas é:

$$f(t) = \begin{cases} x_1 = (1000)^T & \text{se } t \in [0, 5) \\ x_2 = (1100)^T & \text{se } t \in [5, 7) \\ x_3 = (1110)^T & \text{se } t \in [7, 12) \\ x_4 = (1111)^T & \text{se } t \in [12, +\infty) \end{cases}$$

**Teorema 6.1** Seja  $M_i \in S_M$ , uma marcação estável, em algum intervalo  $[t_a, t_b)$  e  $t$  definido neste intervalo, então,

$$M_i = M_0 + A^T f(t) \quad (6.2)$$

**Prova:**

Se  $M_i \in S_M$  então existe um vetor  $x_i \in S_x$  que transforma a marcação  $M_0$  na marcação  $M_i$ . Portanto, a partir da Equação (6.1) temos  $M_i' = M_0 + A^T f(t)$ .  $\square$

A Equação (6.2) pode ser utilizada para determinar as marcações estáveis do sistema em qualquer instante de tempo considerado. Por exemplo, para o modelo apresentado na Figura 6.3 temos:

$$M_i = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 1 \end{bmatrix} \cdot f(t) \quad (6.3)$$

Por exemplo, para  $t \in [0, 5)$ ,  $f(t) = x_1 = (1000)^T$ , logo, obtemos a marcação  $M_1 = (010000)^T$ , como mostrado na Equação (6.4).

$$M_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (6.4)$$

## 6.2 Determinação dos Vetores de Estado Contínuo

Os estados de um sistema de produção em lotes mudam devido a ocorrência de eventos, onde para cada estado alcançável, a dinâmica contínua do sistema pode ser especificada por um determinado conjunto de equações diferenciais. Na modelagem de sistemas utilizando redes de Petri, a ocorrência de transições corresponde a ocorrência destes eventos e as marcações obtidas representam os estados do sistema. Desta forma, a marcação de certos lugares podem ser utilizada para determinar este conjunto de equações diferenciais. Estes lugares são definidos a seguir.

**Definição 6.5** Um lugar  $p$  do modelo é dito lugar de chaveamento se:

1. para qualquer marcação alcançável, este lugar modela alguma atividade física do sistema e
2. para qualquer seqüência de ocorrência de transições  $M(p) \leq 1$ .

Por exemplo, os lugares  $p_1$  e  $p_3$  do modelo apresentado na Figura 6.3, são lugares de chaveamento pois, calculando os invariantes de lugar podemos mostrar que

$$M(p_1) + M(p_2) + M(p_3) \leq 1 \quad (6.5)$$

logo, para qualquer seqüência de ocorrência de transições a soma das marcação destes lugares é sempre menor ou igual a unidade. Além disso, estes lugares representam atividades físicas do sistema. Veja a descrição destes lugares na Seção 6.1. A marcação dos lugares de chaveamento são utilizadas como variáveis lógicas para determinar as equações diferenciais que descrevem o sistema. Por exemplo, para o sistema tanque-válvulas temos que

$$\dot{v} = M(p_1)\xi_1 - M(p_3)\xi_2 \quad (6.6)$$

Note que, a Equação (6.6) representa a dinâmica contínua do sistema para qualquer seqüência de ocorrência de transições pois,

1. Na marcação inicial  $M_0$ ,  $M(p_1) = M(p_3) = 0$ , logo  $\dot{v} = 0$ ,
2. A ocorrência da transição  $\lambda_1$  no instante  $t = 0$  (evento abrir válvula  $q_1$ ), transforma a marcação inicial  $M_0$  na marcação  $M_1$ , onde  $M(p_1) = 1$  e  $M(p_3) = 0$ , logo  $\dot{v} = \xi_1$ ,
3. A ocorrência da transição  $\lambda_2$  no instante  $t = 5$  (evento fechar válvula  $q_1$ ), transforma a marcação  $M_1$  na marcação  $M_2$ , onde  $M(p_1) = M(p_3) = 0$ , logo  $\dot{v} = 0$ ,
4. A ocorrência da transição  $\lambda_3$  no instante  $t = 7$  (evento abrir válvula  $q_2$ ), transforma a marcação  $M_2$  na marcação  $M_3$ , onde  $M(p_1) = 0$  e  $M(p_3) = 1$ , logo  $\dot{v} = -\xi_2$ ,
5. A ocorrência da transição  $\lambda_4$  no instante  $t = 12$  (evento fechar válvula  $q_2$ ), transforma a marcação  $M_3$  na marcação  $M_4$ ,  $M(p_1) = M(p_3) = 0$ , logo  $\dot{v} = 0$ .

Seja  $\#(S)$  o número de reatores de um sistema de produção em lotes. Neste caso, temos  $\#(S)$  variáveis de estado contínuas descrevendo o volume de cada reator do sistema. Portanto,  $v_k(t)$  representa o volume do  $k$ -ésimo reator do sistema em qualquer instante de tempo considerado,  $k = 1, 2, \dots, \#(S)$ . Baseado neste fato, e assumindo que o modelo do sistema possui  $m$  lugares, temos as seguintes definições:

**Definição 6.6** Denota-se vetor chaveamento, o vetor

$$\dot{\mathbf{V}}_i = (\dot{v}_1, \dot{v}_2, \dots, \dot{v}_k, \dots, \dot{v}_{\#(S)})^T$$

onde,  $\dot{v}_k$  representa a taxa de variação de volume no  $k$ -ésimo reator do sistema.

**Definição 6.7** Denota-se matriz de campo vetores, a matriz  $B = [b_{kj}]_{[\#(S) \times m]}$  onde,  $b_{kj}$  é igual a zero se o lugar  $p_j$  não é um lugar de chaveamento de  $\dot{v}_k$  senão,  $b_{kj} = \xi_{kj}$ .

**Teorema 6.2** O vetor chaveamento é dado por

$$\dot{\mathbf{V}}_i = C^T f(t) \quad (6.7)$$

onde,  $\dot{\mathbf{V}}_i = B M_i$  e  $C^T = B A^T$ .

**Prova:** Para obter a Equação (6.7) basta multiplicar ambos os membros da Equação (6.2) pela matriz de campos vetores  $B$ . O produto  $B M_0$  é um vetor nulo, pois a marcação inicial de todos os lugares de chaveamento é zero indicando ausência de qualquer atividade física no sistema.  $\square$

A Equação (6.7) pode ser utilizada para determinar o chaveamento entre os diversos campos vetores das equações diferenciais que descrevem a dinâmica contínua do sistema. Uma vez obtido  $\dot{\mathbf{V}}_i$  para algum intervalo  $[t_a, t_b)$ , podemos integrar cada componente deste vetor no intervalo considerado, obtendo assim o vetor de estados contínuos  $\mathbf{V}_i$ , logo:

**Definição 6.8** Denota-se vetor de estados contínuos, o vetor

$$\mathbf{V}_i = (v_1, v_2, \dots, v_k, \dots, v_{\#(S)})^T$$

onde,

$$v_k(t) = \int_{t_a}^t \xi(u) du + v_k(t_a), \forall t \in [t_a, t_b) \quad (6.8)$$

Portanto, para obter o vetor  $V_i$  basta integrar, utilizando a Equação (6.8), cada componente do vetor  $\dot{V}_i$  no intervalo de tempo considerado, ou seja

$$V_i = \int_{t_a}^t \dot{V}_i du + V_i(t_a), \forall t \in [t_a, t_b] \quad (6.9)$$

Por exemplo, para o sistema tanque-válvulas, a matriz de campos vetores e a Equação (6.7) são as seguintes:

$$B = \begin{bmatrix} 0 & \xi_1 & 0 & -\xi_2 & 0 & 0 \end{bmatrix} \quad (6.10)$$

$$\begin{bmatrix} \dot{v} \end{bmatrix} = \begin{bmatrix} \xi_1 & -\xi_1 & -\xi_2 & \xi_2 \end{bmatrix} f(t) \quad (6.11)$$

onde,  $\dot{V}_i = [\dot{v}]$  e  $C^T = \begin{bmatrix} \xi_1 & -\xi_1 & -\xi_2 & \xi_2 \end{bmatrix}$ . Note que, o produto  $BM_0$  é um vetor nulo.

Para exemplificar a utilização da 6.11, considere que  $\xi_1 = \xi_2 = k$  unidades de volume/unidade de tempo e que a capacidade máxima do tanque é  $v_c = 6k$  unidades de volume. Neste caso, os vetores de estado contínuos são:

1. No intervalo  $t \in [0, 5)$ ,  $f(t) = x_1$  e  $\dot{V}_1 = [k]$  dado pela Equação (6.11), e resolvendo a Equação (6.8) para estes valores temos:

$$v(t) = k \int_0^t du + v(0) \implies v(t) = kt$$

portanto,  $V_1(t) = [kt]$ .

2. No intervalo  $t \in [5, 7)$ ,  $f(t) = x_2$  e  $\dot{V}_2 = [0]$  dado pela Equação (6.11), e resolvendo a Equação (6.8) para estes valores temos:

$$v(t) = v(5) \implies v(t) = 5k$$

portanto,  $V_2 = [5k]$ .

3. No intervalo  $t \in [7, 12)$ ,  $f(t) = x_3$  e  $\dot{V}_3 = [-k]$  dado pela Equação (6.11), e resolvendo a Equação (6.8) para estes valores temos:

$$v(t) = -k \int_7^t du + v(7) \implies v(t) = k(12 - t)$$

portanto,  $V_3(t) = [k(12 - t)]$ .

4. No intervalo  $t \in [12, +\infty)$ ,  $f(t) = x_4$  e  $V_4 \doteq [0]$  dado pela Equação (6.11), e resolvendo a Equação (6.8) para estes valores temos:

$$v(t) = v(12) \implies v(t) = 0$$

portanto,  $V_4 = [0]$ .

### 6.3 Algoritmos para Cálculo e Verificação de Capacidade

Na seção anterior obtivemos cada vetor  $V_i$  para o sistema tanque-válvulas. No estado inicial  $E_0$ , temos que o volume do tanque é  $V_0 = [0]$ . No instante  $t = 0$  a válvula  $q_1$  é aberta e o sistema muda para o estado  $E_1$  e o volume do tanque é dado por  $V_1 = [kt]$  no intervalo  $[0, 5)$ . No instante  $t = 5$  a válvula  $q_1$  é fechada e o sistema muda para o estado  $E_2$  e o volume do tanque é dado por  $V_2 = [5k]$  no intervalo  $[5, 7)$  e, assim por diante.

**Definição 6.9** *Seja  $E_i$  um estado estável no intervalo  $[t_a, t_b)$  e  $t$  definido neste intervalo. Denota-se função indexação, a função  $I : [0, +\infty) \rightarrow \{1, 2, \dots\}$  onde,  $I(t) = i \forall t \in [t_a, t_b)$ .*

Por exemplo, considerando o sistema tanque-válvulas,  $I(t) = 1$  para  $t \in [0, 5)$  pois  $E_1$  é estável neste intervalo,  $I(t) = 2$  para  $t \in [5, 7)$  pois  $E_2$  é estável neste intervalo e, assim por diante. A função de indexação é utilizada nos seguintes algoritmos:

**ALGORITMO:** Cálculo de Capacidade

1.  $t = 0$ ;

**Enquanto**  $t < t_{max}$  **faça**

2. Obter o vetor  $\dot{V}_i$ ;
3. Obter o vetor  $V_i$ ;
4.  $i \leftarrow I(t)$ ;

**Enquanto**  $i = I(t)$  **faça**

- 5.1 Avaliar  $V_i(t)$ ;
- 5.2  $t = t + 1$ ;

FimEnquanto

FimEnquanto

Fim

Aplicando o algoritmo para o cálculo de capacidade ao sistema tanque- válvulas, obtemos a curva apresentada na Figura 6.5, onde  $t_{max} = 12$  unidades de tempo. Note que no instante  $t = 5$  o volume do lote é máximo, ou seja,  $5k$  unidades de volume. Portanto, a capacidade do tanque deve ser maior do que  $5k$  senão, o lote não poderá ser transferido para o tanque.

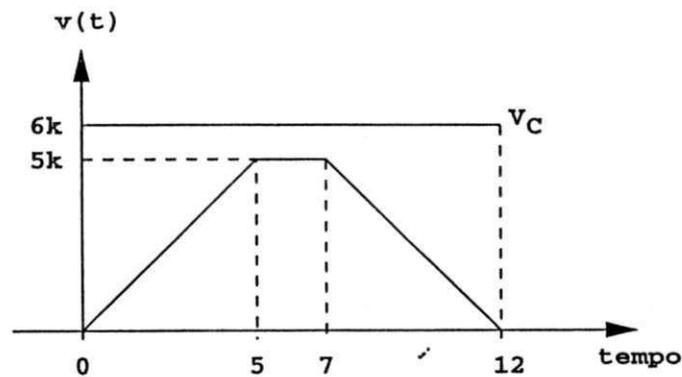


Figura 6.5: Variação de volume de um lote para o sistema tanque-válvulas

**Definição 6.10** Denota-se vetor capacidade, o vetor

$$V_C = (v_{c_1}, v_{c_2}, \dots, v_{c_k}, \dots, v_{c_{\#(S)}})^T$$

onde,  $v_{c_k}$  representa a capacidade em unidades de volume do  $k$ -ésimo reator do sistema.

**Definição 6.11** Um sistema de produção em lotes é dito  $V_C$ -limitado se  $V_i(t) \leq V_C \forall t \in [0, +\infty)$  onde,  $i = I(t)$ .

Por exemplo, para o sistema tanque-válvulas  $V_C = [6k]$ . Podemos utilizar o algoritmo a seguir para verificar os limites de capacidade do sistema no intervalo de tempo  $[0, t_{max}]$ .

**ALGORITMO:** Verificação dos Limites de Capacidade

1.  $t = 0$ ;