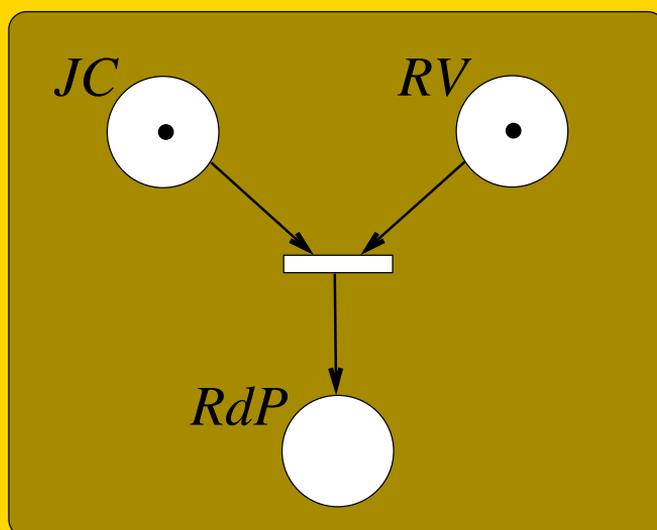


REDES DE PETRI



JANETTE CARDOSO
ROBERT VALETTE

A rede de Petri é um modelo matemático com representação gráfica que vem sendo amplamente utilizado, há mais de 30 anos, em vários domínios de atuação, entre os quais destacam-se os sistemas de manufatura, de comunicação, de transporte, de informação, logísticos e, de forma geral, todos os sistemas a eventos discretos. Especificar, analisar o comportamento lógico, avaliar o desempenho e implementar esses tipos de sistemas são as principais motivações para o uso da Rede de Petri.

Este livro trata, numa primeira parte, do modelo básico da rede de Petri com suas definições, propriedades e a análise destas. Na segunda parte, apresentam-se as extensões da rede de Petri que permitem tratar os dados, o tempo e a interação com o ambiente externo. Finalizando, um último capítulo aborda temas recentes de pesquisa sobre o uso da Rede de Petri associada às lógicas nebulosa e linear e a sua aplicação ao estudo de sistemas híbridos.

Através deste livro, os autores colocam sua larga experiência de ensino e pesquisa em rede de Petri à disposição dos professores, estudantes e engenheiros que pretendem se familiarizar com este assunto, apresentando um suporte completo e atualizado para disciplinas de graduação e pós-graduação de cursos de informática, engenharias de automação, elétrica e de produção entre outros.

Jean-Marie Farines

Janette Cardoso

Robert Valette

Redes de Petri

Florianópolis
1997

*Para Thomas
Janette*

*Para Marly, Fabien e Aline
Robert*

Conteúdo

LISTA DE FIGURAS	5
PREFÁCIO	7
I Modelo Básico	10
1 VOCABULÁRIO E CONCEITOS	11
1.1 Sistemas discretos	11
1.2 Noções básicas	12
1.2.1 Conceitos utilizados na modelagem	12
1.2.2 Paralelismo, cooperação, competição	13
1.3 Máquina de estados finitos	13
1.3.1 Processo seqüencial único	13
1.3.2 Vários processos seqüenciais	14
1.4 Exemplo de sistema discreto paralelo	15
1.4.1 Apresentação do exemplo	15
1.4.2 Modelagem usando máquinas de estado	16
1.5 Requisitos da modelagem	17
1.6 Apresentação informal da rede de Petri	17
1.6.1 Elementos básicos	17
1.6.2 Comportamento dinâmico	18
1.7 Modelando diferentes interações entre processos	19
1.7.1 Seqüência	19
1.7.2 Evoluções síncronas e assíncronas	20
1.7.3 Variantes e caminhos alternativos	21
1.7.4 Repetição	22
1.7.5 Alocação de recursos	23
1.8 Notas	27
1.9 Exercícios	28
2 DEFINIÇÕES	30
2.1 Conceitos	31
2.1.1 Rede de Petri	31
2.1.2 Rede marcada	31
2.1.3 Grafo associado e notação matricial	32
2.1.4 Rede de Petri pura	33

2.1.5	Transição sensibilizada	33
2.1.6	Disparo de uma transição	34
2.1.7	Conflito e paralelismo	35
2.1.8	Seqüência de disparo	36
2.1.9	Conjunto de marcações acessíveis	39
2.2	Rede de Petri e sistema de regras	39
2.2.1	Sistema de regras	40
2.2.2	Gramática	41
2.3	Propriedades do modelo	42
2.3.1	Rede marcada k-limitada	42
2.3.2	Rede marcada viva	44
2.3.3	Rede marcada reiniciável	46
2.4	Propriedades estruturais	47
2.4.1	Componentes conservativos, invariantes de lugar	47
2.4.2	Componentes repetitivos, invariantes de transição	49
2.5	Notas	51
2.6	Exercícios	51
3	ANÁLISE DAS PROPRIEDADES	53
3.1	Análise por enumeração de marcações	53
3.1.1	Decidibilidade da propriedade k-limitada	54
3.1.2	Procura das outras propriedades	55
3.2	Análise estrutural	56
3.2.1	Componentes conservativos, invariantes de lugar	57
3.2.2	Componentes repetitivos, invariantes de transição	59
3.3	Análise através de redução	60
3.3.1	Lugar substituível	60
3.3.2	Lugar implícito	61
3.3.3	Transição neutra ou identidade	64
3.3.4	Transições idênticas	65
3.4	Relação entre os diversos métodos de análise	65
3.5	Resultados particulares: subclasses	68
3.6	Notas	70
3.7	Exercícios	70

II Dados, Tempo e Ambiente Externo **72**

4	REDES INTERPRETADAS	73
4.1	O que é a interpretação?	74
4.2	Análise	77
4.3	Validação por simulação	79
4.4	Modelagem com rede de Petri interpretada	79
4.5	Exemplo	80
4.5.1	Descrição do processo	80

4.5.2	Modelo do sistema	83
4.6	Notas	83
4.7	Exercícios	83
5	REDES DE ALTO NÍVEL	84
5.1	Características gerais	85
5.2	Os diferentes modelos de RPAN	88
5.2.1	Rede de Petri colorida	88
5.2.2	Rede de Petri predicado-transição	92
5.2.3	Rede de Petri a objetos	96
5.3	Características dos modelos	101
5.3.1	A ficha como elemento de informação	101
5.3.2	Dobramento das transições e dos lugares	101
5.4	Escolha do modelo	102
5.5	Notas	104
5.6	Exercícios	104
6	REDES DE PETRI E A REPRESENTAÇÃO DO TEMPO	105
6.1	Rede de Petri temporizada	105
6.1.1	Tempo associado ao lugar	105
6.1.2	Tempo associado à transição	106
6.2	Rede de Petri temporal	107
6.2.1	Representação do <i>watchdog</i>	108
6.2.2	Comparação entre os dois modelos	109
6.3	Rede de Petri estocástica	109
6.3.1	Limite das redes temporizada e temporal	109
6.3.2	Duração de sensibilização estocástica	110
6.3.3	Obtenção de uma cadeia de Markov	111
6.4	Notas	112
7	MÉTODOS DE IMPLEMENTAÇÃO	113
7.1	Abordagem procedimental	114
7.2	Abordagem não procedimental	114
7.2.1	Princípio	114
7.2.2	Comparação com a abordagem procedimental	115
7.3	Abordagem descentralizada	116
8	REDES DE PETRI, LÓGICAS NÃO CLÁSSICAS E SISTEMAS HÍBRIDOS	118
8.1	Redes de Petri nebulosas	118
8.1.1	Requisitos para modelos de sistemas dinâmicos	118
8.1.2	Combinando redes de Petri e conjuntos nebulosos	119
8.2	Redes de Petri como semântica para lógica linear	122
8.2.1	Lógica linear: noções de base	122
8.2.2	Descrição da rede de Petri usando lógica linear	123
8.2.3	Seqüência de tiro na lógica linear	124
8.3	Redes de Petri para sistemas híbridos	126
8.3.1	Sistema de produção híbrido	126

8.3.2	Técnicas de modelagem	127
8.4	Notas	131
A	GRAFOS	132
A.1	Definições formais e notação	132
A.2	Conectividade	133
A.3	Notas	136
B	CÁLCULO DOS COMPONENTES	137
B.1	Princípio do cálculo de uma base	137
B.2	Exemplo	140
B.3	Algoritmo simplificado e procura das soluções positivas	141
	BIBLIOGRAFIA	144
	ÍNDICE	149

Lista de Figuras

1.1	Sistemas: a) discretizado; b) discreto; c) a eventos discretos	12
1.2	a) Sistema de triagem; b) Máquina de estados	14
1.3	Explosão combinatória do número de estados	14
1.4	Triagem de objetos pesados com estoque intermediário	16
1.5	Conjunto de máquinas comunicantes	17
1.6	Rede de Petri	18
1.7	Seqüência de processos	20
1.8	a) Divisão; b) Junção	21
1.9	a) Caminhos alternativos; b) Repetição	22
1.10	Partilhamento de um recurso	23
1.11	a) Sistema de transporte; b) Modelo do circuito N_0	24
1.12	Modelo dos circuitos N_1 e N_2	25
1.13	a) Sistema tipo batelada; b) Modelo rede de Petri	26
1.14	Grafo do sistema tipo batelada	27
1.15	Célula de fabricação	29
2.1	Rede de Petri	32
2.2	Rede de Petri não pura	33
2.3	Seqüência de disparo de transições	36
2.4	Rede com seqüência não disparável	38
2.5	Grafo de marcações acessíveis	40
2.6	Rede de Petri não limitada	43
2.7	Rede parênteses	43
2.8	Transição quase viva e não viva	45
2.9	Grafo das marcações (transição quase viva)	45
2.10	Transição quase viva e seqüência infinita	46
2.11	a) Rede não reiniciável; b) Grafo associado	47
2.12	Rede com propriedades dependentes da marcação inicial	47
2.13	Invariantes	48
2.14	Exercício 3	51
2.15	Exercício 4	52
3.1	a) Lugar substituível p_4 ; b) Simplificação do lugar p_4	61
3.2	a) Lugar implícito p_1 ; b) p_1 simplificado; c) Contra-exemplo	62
3.3	a) Lugares idênticos; b) Lugar implícito degenerado	63
3.4	Transição neutra a) t ; b) d (não simplificável)	64
3.5	a) Transições t_1 e t_2 idênticas; b) Simplificação de t_2	65
3.6	Caracterização das marcações	66

3.7	Rede com conjunto de marcações diferentes	67
3.8	a) Grafo de eventos; b) Circuitos elementares	69
3.9	Análise por redução	71
4.1	Disparo de transições	73
4.2	Associação de uma atividade a uma transição	75
4.3	Interação da rede de Petri com o ambiente externo	77
4.4	Marcações acessíveis	78
4.5	Estação de coleta de petróleo	80
4.6	Modelo do controle	81
4.7	Modelo da planta	82
5.1	Comportamento: a) detalhado; b) um só processo; c) geral	86
5.2	Leitores e escritores: rede completa	87
5.3	Leitores e escritores: rede dobrada	87
5.4	Rede de Petri colorida	89
5.5	Rede de Petri predicado-transição	93
5.6	Rede de Petri subjacente	96
6.1	Temporização da rede de Petri	106
6.2	O tempo e a rede de Petri	107
6.3	<i>Watchdog</i>	108
6.4	a) Rede de Petri estocástica; b) Grafo $GA(R; M)$	112
7.1	Princípio do jogador	115
8.1	a) $P \rightarrow Q \wedge R$, b) $(P \rightarrow Q) \wedge (P \rightarrow R)$	122
8.2	Recursos e seqüência	125
8.3	Reator	128
8.4	Receita de fabricação	130
8.5	Explicitação do tempo	130
8.6	Duração não calculável <i>a priori</i>	130
A.1	a) Grafo conexo; b) Grafo não conexo	134
A.2	Grafo a) fortemente conexo; b) fracamente conexo	135

PREFÁCIO

Histórico

A rede de Petri é uma ferramenta gráfica e matemática que se adapta bem a um grande número de aplicações em que as noções de eventos e de evoluções simultâneas são importantes.

Esta teoria é muito jovem, pois nasceu da tese, intitulada *Comunicação com autômatos*, defendida por Carl Adam Petri em 1962 na Universidade de Darmstadt, Alemanha. Carl Adam Petri, nascido em 1926 em Leipzig, é professor na Universidade de Bonn. Anatol W. Holt foi seduzido por este trabalho e sob sua impulsão um grupo de pesquisadores do *Massachusetts Institute of Technology-MIT*, Estados Unidos, lança as bases, entre 1968 e 1976, do que se tornou as *redes de Petri*. Entre estes pioneiros destacam-se F. Commoner e M. Hack.

Entre as aplicações pode-se citar: avaliação de desempenho, análise e verificação formal em sistemas discretos, protocolos de comunicação, controle de oficinas de fabricação, concepção de *software* tempo real e/ou distribuído, sistemas de informação (organização de empresas), sistemas de transporte, logística, gerenciamento de base de dados, interface homem-máquina e multimídia.

Em relação ao controle de sistemas de fabricação automatizada, a aplicação de redes de Petri efetuou-se de início na França, sob a forma um pouco modificada da norma *Grafset*, para a programação de autômatos programáveis industriais. Esta norma foi inicialmente proposta por uma comissão da *Afcet-Association française des sciences et technologie de l'information et des systèmes* em 1977 e se tornou uma norma industrial (C03.190-UTE) na França, em 1980, e em seguida em nível europeu através do escritório central da CEI sob referência IEC 848.

A complexidade dos sistemas a eventos discretos, em particular no caso de sistemas de fabricação automatizada, leva a uma decomposição hierárquica com vários níveis de controle. Em geral são utilizados cinco níveis: *planejamento*, *escalonamento*, *coordenação global*, *coordenação de sub-sistemas* e *controle direto* (autômatos programáveis diretamente conectados aos sensores e aos atuadores). Considerando a utilização da rede de Petri no nível de coordenação, os países mais ativos são a Alemanha (Sociedade PSI em Berlim) e o Japão (Sociedade Hitachi). Na França pode-se citar a Sociedade IXI (Toulouse).

As vantagens da utilização da rede de Petri podem ser resumidas pelas considerações seguintes:

- pode-se descrever uma ordem parcial entre vários eventos, o que possibilita levar-se em conta a flexibilidade;
- os estados, bem como os eventos, são representados explicitamente;

- uma única família de ferramentas é utilizada através da especificação, da modelagem, da análise, da avaliação do desempenho e da implementação;
- uma única família de ferramentas é utilizada nos diversos níveis da estrutura hierárquica do controle, o que facilita a integração destes níveis;
- uma descrição precisa e formal das sincronizações torna-se possível, o que é essencial para alcançar-se a necessária segurança de funcionamento.

Organização do livro

Este livro corresponde às notas de aula utilizadas, desde 1991, na disciplina *Sistemas de tempo real I* do curso de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.

O livro é organizado em duas partes. A primeira parte trata do modelo básico, chamado também de rede de Petri ordinária. O capítulo 1 faz uma introdução, iniciando com a definição de sistemas a eventos discretos, situando de modo informal o modelo, sua importância e áreas de aplicação.

No capítulo 2 são apresentados os aspectos básicos do modelo rede de Petri: definições e propriedades. Como realizar a análise destas é descrito no capítulo 3. É interessante complementar o estudo deste capítulo com o *software* ARP de análise de redes de Petri, desenvolvido no LCMI - Laboratório de Controle e Microinformática do Departamento de Engenharia Elétrica da UFSC (já distribuído para algumas universidades, inclusive da França, Bélgica e Inglaterra).

A segunda parte deste livro trata da interação entre a rede de Petri e os dados, o tempo e o ambiente externo. O capítulo 4 apresenta as redes de Petri interpretadas, que permitem representar a interação com o ambiente externo. No capítulo 5 são apresentadas as redes de alto nível, que permitem individualizar a ficha, permitindo que esta transporte informação, aumentando o nível de abstração do modelo. As diferentes maneiras de explicitar o tempo são tratadas no capítulo 6.

O capítulo 7 trata dos métodos de implementação de sistemas representados por redes de Petri.

O capítulo 8 se refere a aplicações mais recentes do modelo. Trata da utilização conjunta da rede de Petri com outras teorias como a lógica nebulosa e a lógica linear. A primeira permite modelar a marcação imprecisa, representando, assim, situações reais em que a informação sobre o sistema é incompleta. A segunda permite expressar a noção de recursos, que a lógica clássica não permite. Este capítulo trata também da utilização de redes de Petri em sistemas híbridos, como os sistemas de tipo *batelada* (*batch*, em inglês), que contém uma parte contínua e uma parte dirigida por eventos discretos.

Durante todo o livro serão apresentadas aplicações da rede de Petri e suas extensões na modelagem e na implementação, destacando-se a área de automação da manufatura.

Este livro pode também ser empregado num curso de redes de Petri de um semestre na graduação, utilizando os capítulos 1 a 4 e parte dos capítulos 6 e 7.

Agradecimentos

Os autores agradecem aos alunos e colegas que, através de perguntas, dúvidas e sugestões, trouxeram contribuições a este trabalho e a Brigitte Pradin-Chézalviel que participou da elaboração do item *Redes de Petri como semântica para lógica linear*.

Agradecem também a UFSC, a CAPES, ao CNPq e ao LAAS por terem oportunizado este livro e o apoio da equipe do Laboratório de Controle e Microinformática LCMI/EEL/UFSC que ajudou a torná-lo possível.

Em particular, agradecem a Eduardo Souza (esms) que, durante a versão final, com profissionalismo e gentileza, prestou inestimável auxílio nas emergências *informáticas*, aos colegas Carlos Alberto Maziero e Guilherme Bittencourt que acudiram aos pedidos de socorro *latexianos*, a Hugo Leonardo Gosmann pela ajuda na editoração de parte da versão final e a Maria Joana Zucco, Eda Brustolin e Ana Lúcia do Amaral pelo apoio e rapidez na revisão.

J. C.
R. V.

Florianópolis, SC
Maio de 1997

Parte I

MODELO BÁSICO

Capítulo 1

VOCABULÁRIO E CONCEITOS

O modelo de rede de Petri foi proposto por Carl Petri para modelar a comunicação entre autômatos, utilizados, na época¹, para representar sistemas a eventos discretos. Para situar o leitor nesta área de atuação, é apresentada inicialmente a caracterização de tais sistemas, e os conceitos básicos utilizados na sua modelagem. A máquina de estados finitos, bastante usada para representar tais sistemas, é discutida, apresentando-se suas limitações.

Antes da definição formal da rede de Petri no capítulo 2, é realizada, neste capítulo, uma apresentação informal. O objetivo é fazer com que o leitor possa, desde o início, realizar a modelagem de alguns sistemas que lhe são familiares. Poderá verificar, assim, o quanto este modelo é poderoso para representar os diferentes processos existentes num sistema, permitindo estruturar, de forma organizada, a modelagem. Mesmo sem estudar ainda as técnicas de análise das propriedades, apresentadas no capítulo 3, poderá observar que o formalismo do modelo auxilia, inclusive, a detectar eventuais erros presentes na especificação informal.

1.1 Sistemas discretos

De um modo geral, um sistema discreto é um sistema no qual as mudanças de estado ocorrem em instantes precisos. Costuma-se situar os sistemas discretos em oposição aos sistemas contínuos. Esta classificação depende do ponto de vista em que se coloca o observador e depende do grau de abstração desejado. Por exemplo, considere uma fresadora num sistema de manufatura. Do ponto de vista da operação de fresagem, o sistema deve ser modelado por um modelo contínuo. Do ponto de vista da coordenação do sistema de manufatura, considerando os eventos *início de fresagem* e *fim de fresagem*, o sistema deve ser modelado por um modelo a eventos discretos. Pode-se, de fato, encontrar diversas definições de tais sistemas, representados na FIG. 1.1, que serão enumerados a seguir.

Sistemas discretizados: são sistemas estudados somente em instantes precisos. Trata-se, portanto, de sistemas contínuos *observados* em instantes discretos (sistemas amostrados). As variáveis de estado evoluem de maneira contínua, sem mudança brusca

¹Recentemente, foi proposta a teoria de autômatos híbridos, para modelar os sistemas que possuem um componente contínuo e um componente discreto (ver capítulo 8).

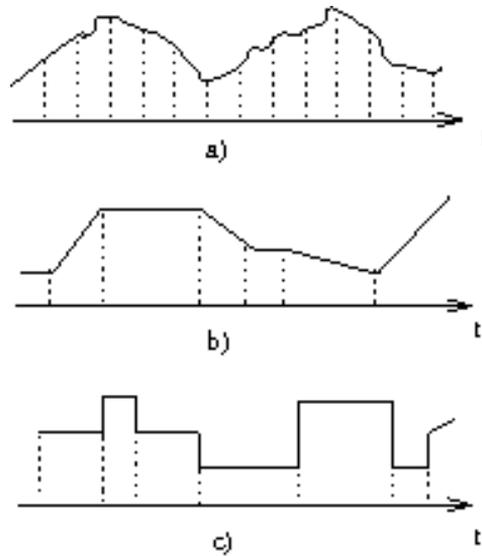


Figura 1.1: Sistemas: a) discretizado; b) discreto; c) a eventos discretos

de comportamento, mas é somente a instantes discretos do tempo que há interesse em conhecer seu valor.

Sistemas discretos: são sistemas para os quais os valores das variáveis de estado, ou ao menos de algumas delas, variam bruscamente a certos instantes. Entretanto, estes instantes não podem necessariamente ser previstos e o conhecimento do estado a um instante dado não permite que, *sem cálculo*, se conheça o estado seguinte.

Sistemas a eventos discretos: são sistemas modelados de tal sorte que as variáveis de estado variam bruscamente em instantes determinados e *que os valores das variáveis nos estados seguintes* podem ser calculados diretamente a partir dos valores precedentes e *sem ter que considerar o tempo entre estes dois instantes*. É esta classe de sistemas que será estudada neste livro.

1.2 Noções básicas

1.2.1 Conceitos utilizados na modelagem

Os conceitos básicos utilizados na modelagem de um sistema baseada numa abordagem por eventos discretos são os seguintes:

Eventos: são os instantes de observação e de mudança de estado do sistema.

Atividades: são as caixas-pretas utilizadas para recuperar e esconder a evolução do sistema físico entre dois eventos. Portanto, os eventos correspondem em geral ao início e ao fim de uma atividade.

Processos: são *seqüências* de eventos e de atividades interdependentes. Por exemplo, um evento *provoca* uma atividade, que *provoca* um evento de fim de atividade, que por sua vez pode *provocar* uma outra atividade e assim por diante.

1.2.2 Paralelismo, cooperação, competição

A evolução dos processos num sistema pode se dar de forma simultânea ou não. Se esta se dá de forma simultânea, os processos podem ser completamente independentes ou relativamente independentes.

Esta independência relativa significa que certas atividades são totalmente independentes entre si, enquanto que outras atividades necessitam de pontos de *sincronização*, isto é, de eventos comuns a várias evoluções. Existem diferentes formas de interação entre processos:

Cooperação: os processos concorrem a um objetivo comum; procura-se descrever uma independência de processos antes de um ponto de sincronização.

Competição: os processos devem ter acesso a um dado recurso para realizar sua tarefa. Se existisse um número suficiente de recursos, os processos seriam completamente independentes. Trata-se, portanto, de um partilhamento de recursos resolvido, em geral, por exclusões mútuas. Procura-se portanto descrever uma exclusão entre dois processos a partir de um ponto de sincronização.

Pseudo-paralelismo: o paralelismo é apenas aparente e os eventos, mesmo independentes, nunca serão simultâneos. Eles serão ordenados por um relógio comum. É o caso de várias tarefas informáticas sendo executadas num único processador. Este executa somente uma instrução por vez.

Paralelismo verdadeiro: os eventos podem ocorrer simultaneamente. Isto significa que não existe uma escala de tempo comum suficientemente precisa para determinar qual evento precedeu o outro. Ocorre quando várias tarefas informáticas são executadas num computador paralelo, com um processador alocado para cada tarefa independente.

1.3 Máquina de estados finitos

1.3.1 Processo seqüencial único

A representação clássica de um sistema a eventos discretos, cujo número de estados é finito, consiste em enumerar todos os *estados* possíveis e a descrever os eventos do tipo *mudanças de estado*, isto é, descrever o próximo estado a partir de cada estado.

O modelo matemático $M = (E, A, \theta, E_0)$ é chamado *máquina de estados finitos* e consiste em um conjunto finito de estados E com um estado inicial E_0 , um alfabeto de entrada A e uma função de transição de estado $\theta : E \times A \rightarrow E$ que associa a cada par estado-entrada o próximo estado. Um grafo é associado ao modelo, cujo conjunto de nós é o conjunto de estados E , e o arco que leva de um estado E_i a um estado E_j é etiquetado pelo evento $a \in A$, tal que $\theta(E_i, a) = E_j$.

Este modelo matemático exprime bem a noção de evento, e parcialmente a de atividade (um estado entre dois eventos); não exprime, entretanto, a noção de processo (evolução simultânea de diversos processos paralelos). Uma máquina de estados finitos descreve, de fato, apenas um único processo seqüencial.

Considere um sistema simples de triagem numa esteira T em movimento. Os objetos pesados devem ser retirados por um operador. Um sensor P indica a presença de um objeto pesado; quando isto acontece, a esteira deve parar para que o operador possa retirar o objeto. Uma vez retirado, a esteira entra novamente em movimento (FIG. 1.2.a).

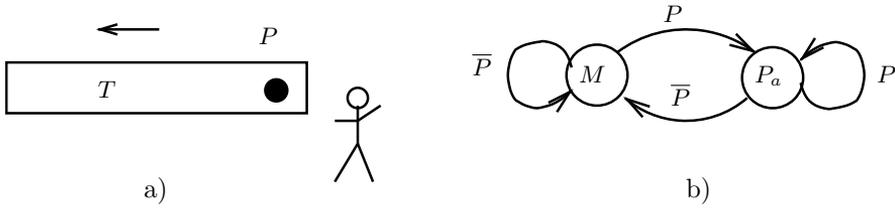


Figura 1.2: a) Sistema de triagem; b) Máquina de estados

A FIG. 1.2.b mostra a máquina de estado deste sistema: $E = \{M, P_a\}$ é o conjunto de estados. O estado inicial M representa a esteira em movimento, e o estado P_a , a esteira parada. O alfabeto $A = \{P, \bar{P}\}$ é formado pelos eventos possíveis de ocorrer no sistema. O evento P indica a presença de um objeto pesado, e o evento \bar{P} , a ausência. Se o estado atual do sistema é M , a ocorrência do evento P leva-o ao estado P_a . Este comportamento é traduzido pela transição de estado $\theta(M, P) = P_a$. As demais transições de estado são dadas por: $\theta(P_a, \bar{P}) = M$, $\theta(P_a, P) = P_a$, $\theta(M, \bar{P}) = M$.

1.3.2 Vários processos seqüenciais

Quando é necessário descrever vários processos seqüenciais, a solução mais simples é representar o sistema por um conjunto de máquinas de estado finito. Se estas máquinas são independentes, não existe nenhum problema. Entretanto, quando existe competição, e principalmente quando existe cooperação entre diversos processos, não existe independência entre as máquinas que deverão sincronizar-se entre si. Têm-se, então, os *processos seqüenciais comunicantes*. A sincronização é descrita fazendo intervir na função θ de uma máquina, os estados $E_i \in E$ de outras máquinas.

A FIG. 1.3 mostra as máquinas de estado M_1 e M_2 (ambas com $n = 2$) e a composição de ambas numa máquina de estado M_G que descreve o comportamento global do sistema, com $n = 4$ (os sinais compartilhados não são mostrados).

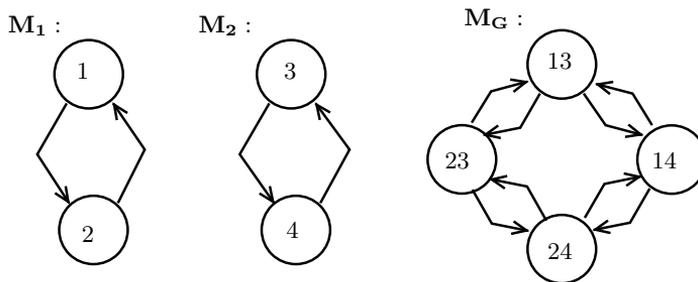


Figura 1.3: Explosão combinatória do número de estados

Os processos seqüenciais comunicantes podem ser decompostos de diversas maneiras, podendo tal decomposição ser material ou funcional (no exemplo da FIG. 1.2.b a decomposição é material). Qualquer que seja o método utilizado, a representação das comunicações entre as máquinas difere da representação interna do seqüenciamento de uma máquina. É por esta razão que as escolhas de decomposição iniciais serão dificilmente

colocadas em questão. Portanto, é impossível utilizar uma abordagem por refinamentos sucessivos (*top-down*). É necessário desde o início escolher uma decomposição que não será colocada em causa *a posteriori*.

Explosão combinatória de estados

Se existem informações partilhadas ou troca de sinais entre as máquinas, é necessário analisar o comportamento global do sistema. Esta análise só poderá ser feita recalculando uma máquina de estado que descreva o sistema global. Neste caso, ocorre a inevitável *explosão combinatória* do número de estados. De fato, a composição de k máquinas, tendo cada uma n estados, produz uma máquina de n^k estados.

Na FIG. 1.3, $n = 2$ para as máquinas M_1 e M_2 e a máquina composta M_G possui $n = 4$. Observe na tabela abaixo que à medida que n e k aumentam ocorre a explosão combinatória:

n	2	2	3	3	4	4	4	6	6	10	6	6
k	3	4	3	4	2	3	4	2	3	3	4	5
n^k	8	16	27	81	16	64	256	36	216	10^3	1296	7776

Não-independência de submáquinas, bloqueio mortal

Ora, a análise do comportamento global do sistema é indispensável, pois um certo número de problemas graves podem aparecer em sistemas paralelos. O mais conhecido é o *bloqueio mortal*. Em certas configurações, nenhuma máquina de estado pode evoluir, pois cada máquina espera uma determinada evolução de uma outra máquina. Tal evolução não pode ocorrer, pois a outra máquina está igualmente num estado de espera. Este fenômeno será ilustrado através de um exemplo.

1.4 Exemplo de sistema discreto paralelo

1.4.1 Apresentação do exemplo

Considere um sistema de triagem com estoque intermediário que utiliza esteiras rolantes. Tal sistema é representado na FIG. 1.4. Os objetos são transportados por uma única esteira T_1 e os objetos pesados devem ser retirados e encaminhados para a esteira T_2 . Supõe-se que a cadência dos objetos chegando em T_1 é, às vezes, maior que a dos objetos saindo de T_2 , o que exige um armazenamento intermediário (*buffer*).

A esteira T_A é utilizada como intermediária para armazenar temporariamente os objetos pesados, evitando assim que o fluxo de chegada de objetos em T_1 seja interrompido. Supõe-se que o comprimento de T_A é suficiente para evitar que esta seja totalmente ocupada, o que permite considerá-la com uma capacidade de armazenamento infinita. A esteira T_A é controlada por dois sinais A e R que a fazem avançar e recuar de uma posição (comando do tipo passo a passo).

O robô R_b é utilizado para deslocar os objetos da esteira T_1 para a esteira T_A (atividade P_1) e de T_A para T_2 (atividade P_2). Um sensor F indica o fim dos movimentos P_1 ou P_2 .

Utilizando uma decomposição ligada ao material (*hardware*), cada subsistema (esteiras T_1 e T_A e robô R_b) é descrito por uma máquina de estados finitos. O sistema é então

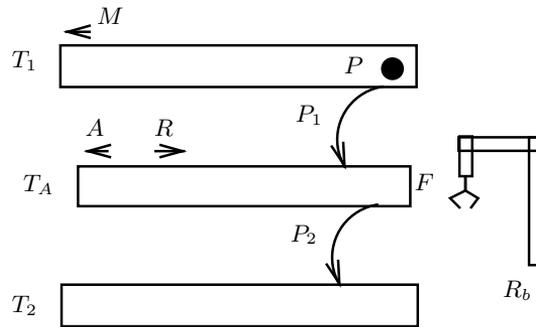


Figura 1.4: Triagem de objetos pesados com estoque intermediário

descrito por três máquinas de estados finitos: T_1 , T_A e RB . A máquina T_1 corresponde ao controle da esteira T_1 , a máquina T_A ao controle da esteira T_A e a máquina RB corresponde ao controle do robô Rb . O mecanismo de funcionamento é o seguinte:

- se um objeto pesado é detectado pelo sensor P (é gerado o evento $P = 1$), a esteira T_A deve avançar de um passo (A); a seguir o sinal $CP1$ é enviado ao controle do robô, o que provoca a execução da atividade P_1 (estado 6);
- se a esteira T_2 está livre e a esteira T_A contém ao menos um objeto (condição OK), o robô executa a atividade P_2 e a esteira T_A deve recuar um passo (envio da ordem CA para o controle do robô e execução da atividade R por T_A).

1.4.2 Modelagem usando máquinas de estado

As três máquinas de estados finitos que descrevem este sistema de triagem são dadas na FIG. 1.5. Os círculos representam os estados e os arcos, as transições entre os estados.

As etiquetas associadas aos estados descrevem as atividades, ou as informações utilizadas por outras máquinas. Por exemplo, M associada ao estado 3 da máquina T_1 significa que, neste estado, a esteira T_1 avança (em movimento). A etiqueta CR (controle R), associada ao estado 5 da máquina RB (robô), indica que esta espera que T_A realize a atividade R (estado de espera).

As etiquetas associadas aos arcos descrevem os eventos que provocam a mudança de estado (passagem do estado atual ao próximo estado). Por exemplo, P da máquina T_1 corresponde à condição $P = 1$ que indica a chegada de um objeto pesado. A etiqueta CR associada a um arco da máquina T_A indica que a mudança de estado (de 1 para 7) ocorre quando a máquina RB está no estado 5.

As comunicações entre as máquinas T_A e P (T_1 é independente) são feitas através de CR , de $CP1$ e do sensor F . Considere o sistema no seguinte estado:

- máquina T_A no estado 1, RB em 2 e T_1 em 3;
- a esteira T_A contém objetos pesados e a esteira T_2 acaba de liberar um lugar (condição OK).

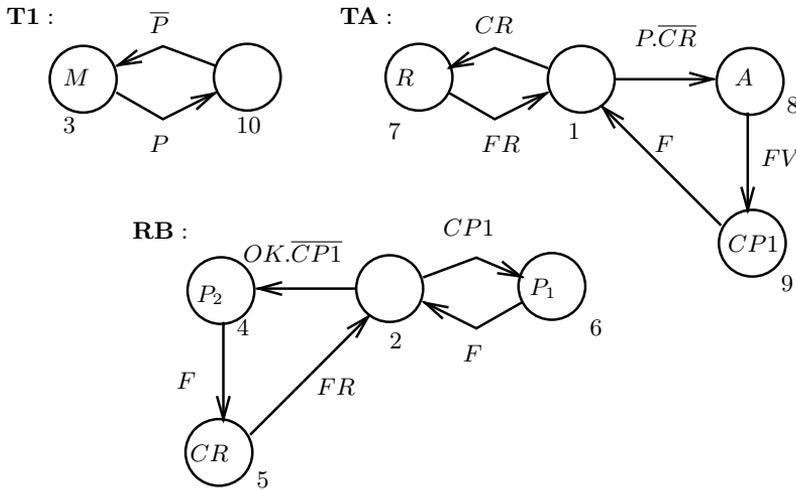


Figura 1.5: Conjunto de máquinas comunicantes

A máquina *RB* vai passar do estado 2 ao estado 4 e a atividade *P2* vai ser executada. Se durante este tempo chega um objeto pesado, a máquina *T1* vai passar para o estado 10 e *TA* para o estado 8. Após a execução de *P2* (sinal *F*), a máquina *RB* vai passar para o estado 5; após *A* (sinal *FA*), a máquina *TA* vai para o estado 9. O sistema estará então totalmente bloqueado (*bloqueio mortal*), pois *TA* espera que *RB* execute a atividade *CP1* enquanto *RB* por sua vez espera que *TA* execute *R*! \diamond^2

1.5 Requisitos da modelagem

Existem técnicas na teoria de máquinas de estados finitos que permitem evitar o bloqueio mortal. Entretanto, um ponto importante é que a estrutura do sistema é completamente perdida. Dois arcos (transições) saindo do mesmo estado podem representar uma decisão entre duas opções diferentes ou dois eventos independentes. Deve-se ressaltar que o principal propósito de uma especificação clara de um sistema a eventos discretos é explicitar as interações entre os estados do processo e o sistema de tomada de decisão que irá controlá-lo. Outro ponto importante é que a introdução de modificações, mesmo pequenas, implica a construção de uma nova máquina de estados.

O modelo de rede de Petri que introduziremos a seguir oferece, além do conhecimento comportamental sobre o sistema, também o conhecimento estrutural como será visto ao longo deste livro.

1.6 Apresentação informal da rede de Petri

1.6.1 Elementos básicos

Os elementos básicos que permitem a definição de uma rede de Petri, em número de três, são polivalentes e em grande medida podem ser interpretados livremente. Estes

²O símbolo \diamond marca o final de um exercício resolvido.

elementos são os seguintes:

- **Lugar** (representado por um círculo): pode ser interpretado como uma condição, um estado parcial, uma espera, um procedimento, um conjunto de recursos, um estoque, uma posição geográfica num sistema de transporte, etc. Em geral, todo lugar tem um predicado associado, por exemplo, *máquina livre*, *peça em espera* (FIG. 1.6);
- **Transição** (representada por barra ou retângulo): é associada a um evento que ocorre no sistema, como o evento *iniciar a operação* (transição *t* na FIG. 1.6);
- **Ficha** (representado por um ponto num lugar): é um indicador significando que a condição associada ao lugar é verificada. Pode representar um objeto (recurso ou peça) numa certa posição geográfica (num determinado estado), ou ainda uma estrutura de dados que se manipula. Por exemplo, uma ficha no lugar *máquina livre* indica que a máquina está livre (predicado verdadeiro). Se não tem fichas neste lugar, o predicado é falso, por conseguinte a máquina não está livre. Se no lugar *peças em espera* houvesse três fichas, indicaria que existem três peças em espera.

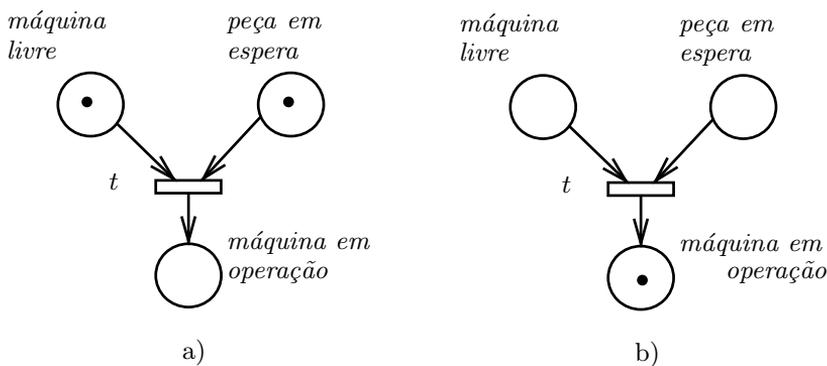


Figura 1.6: Rede de Petri

A primeira observação a fazer é que estas diversas interpretações dos lugares e fichas são bastante variadas. Podem ser utilizadas para descrever entidades abstratas como condições ou estados, mas também entidades físicas como peças ou depósitos. Pode-se também chegar a uma banalização completa, no nível descritivo, entre as peças, ferramentas e, de modo geral, entre todos os recursos utilizados na fábrica. É esta grande *banalização* que permite uma visão sintética do sistema a ser modelado e que autoriza certos procedimentos de análise.

1.6.2 Comportamento dinâmico

O estado do sistema é dado pela repartição de fichas nos lugares da rede de Petri, cada lugar representando um estado parcial do sistema. A cada evento que ocorre no sistema, é associada uma transição no modelo de rede de Petri. A *ocorrência* de um

evento no sistema (que faz com que este passe do estado atual ao próximo estado) é representado, no modelo, pelo *disparo* da transição ao qual este está associado.

O disparo de uma transição consiste em dois passos:

- retirar as fichas dos lugares de entrada, indicando que esta condição não é mais verdadeira após a ocorrência do evento, e
- depositar fichas em cada lugar de saída, indicando que estas atividades estarão, após a ocorrência do evento, sendo executadas.

Por exemplo, a ocorrência do evento *iniciar a operação*, associado à transição t (FIG. 1.6.a), só pode acontecer se houver (ao menos) uma ficha no lugar *máquina livre* e (ao menos) uma ficha no lugar *peça em espera*. A ocorrência do evento *iniciar a operação*, no sistema, equivale ao tiro da transição t na rede de Petri: é retirada uma ficha do lugar *máquina livre* e uma ficha do lugar *peça em espera*, e é colocada uma ficha no lugar *máquina em operação* (FIG. 1.6.b).

Vale aqui uma observação sobre os termos “retirada” (ou *desaparecimento*) e “colocação” (ou *criação*) de fichas quando do disparo de uma transição t à qual é associado um evento e do sistema que está sendo modelado.

O disparo de t corresponde à ocorrência do evento e no sistema real, que o faz passar de um estado atual E_i ao próximo estado E_{i+1} (ver item 1.3). O estado E_i é representado na rede pela distribuição de fichas nos lugares, chamada marcação M_i . Do mesmo modo que o sistema só atingirá o estado E_{i+1} após a ocorrência do evento e se estiver no estado E_i , assim também a transição t só será disparada se a marcação for M_i (marcação em que, em particular, os lugares de entrada de t estão marcados). A marcação M_{i+1} , correspondente ao estado E_{i+1} , será atingida após o disparo da transição t .

O desaparecimento das fichas nos lugares de entrada de t indica que as condições ou predicados associados àqueles lugares não são mais verdadeiros, e o surgimento de fichas nos lugares de saída indica que os predicados associados a estes lugares são verdadeiros. O comportamento dinâmico do sistema é, assim, traduzido pelo comportamento da rede de Petri.

1.7 Modelando diferentes interações entre processos

Como foi visto no item 1.2.1, os processos podem evoluir em cooperação, em competição e em paralelo. Dependendo do nível de detalhamento da modelagem, um processo pode ser representado por somente um lugar, ou uma atividade. Os processos podem ainda evoluir em seqüência, de forma repetida, etc. Estas diferentes interações entre os processos, num sistema a eventos discretos, serão modeladas, a seguir, de modo informal, utilizando a rede de Petri.

1.7.1 Seqüência

Considere a rede da FIG. 1.7. Esta mesma rede pode representar:

- Seqüência de um processo de fabricação. Os lugares P_1 , P_2 e P_3 representam as diferentes fases da operação sobre a peça, que devem ser encadeadas em seqüência. As transições t_1 , t_2 e t_3 descrevem os eventos de *passagem de uma fase a outra* e

as fichas correspondem aos artigos. Um artigo está sendo usinado em P_1 (fase J_1), enquanto que um outro artigo está, neste momento, na fase J_2 , tendo já passado pela fase J_1 (lugares P_1 e P_2).

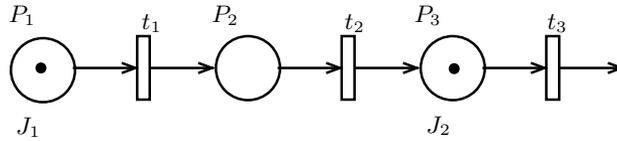


Figura 1.7: Seqüência de processos

- Trecho do itinerário de um sistema de transporte. Este sistema é baseado em veículos que seguem circuitos pré-traçados no solo. Neste caso, os lugares descrevem as seções; as transições correspondem à passagem do veículo de uma seção a outra (passagem por um sensor no solo) e as fichas representam os veículos. Um veículo na seção S_i é representado na rede de Petri por uma ficha no lugar P_i . O evento *sair da seção S_i e entrar na seção S_{i+1}* é associado à transição t_i . Assim, a marcação da rede da FIG. 1.7 representa um veículo em P_1 e outro veículo em P_3 , indicando que é possível a ocorrência (em paralelo!) dos eventos *sair de S_1 e sair de S_3* .

Nos dois casos acima a rede de Petri é a mesma, o que exprime que, fazendo-se abstração dos detalhes e considerando-se apenas a *estrutura dos encadeamentos de eventos e atividades*, estes dois sistemas são idênticos.

1.7.2 Evoluções síncronas e assíncronas

Se a rede de Petri da FIG. 1.7 representa o modelo de um processo de fabricação, as fichas em P_1 e P_3 exprimem que, *simultaneamente*, existe um artigo que está na fase J_1 e outro artigo que está na fase J_2 . Do mesmo modo, se esta figura representa o trecho de um sistema de transporte, então, vê-se que *simultaneamente* dois veículos circulam no mesmo itinerário, em seções diferentes (P_1 e P_3). A evolução destas duas fichas (artigos ou veículos), tais como elas são descritas pela rede, são *independentes* e se desenvolvem de maneira totalmente *assíncrona*: não há nenhuma correlação entre o fim da fase J_1 e o fim da fase J_2 .

No caso da FIG. 1.8.a é descrito um procedimento de divisão ou separação. O fim da operação P_1 consiste em separar um artigo (neste caso J_1) para criar dois novos artigos correspondendo a processos de fabricação diferentes. Portanto, os dois artigos J_2 e J_3 são criados, *simultaneamente*, de modo síncrono. Deste ponto de vista, eles não são independentes. Entretanto, após sua criação, evoluem independentemente um do outro de modo assíncrono. O evento *desaparecimento* de um artigo J_1 e a *criação* de *mais* um artigo J_2 e J_3 é descrito de modo claro e sintético pela transição t_2 . A marcação da FIG. 1.8.a indica que o artigo J_1 está sofrendo a operação P_1 (uma ficha em P_1) e que, ao mesmo tempo, os artigos J_2 e J_3 também estão em operação (lugares P_2 e P_3 marcados).

No caso da FIG. 1.8.b, os artigos evoluem independentemente, paralelamente e de modo assíncrono (transições t_1 e t_2) nos diferentes processos exceto na transição t_3 que

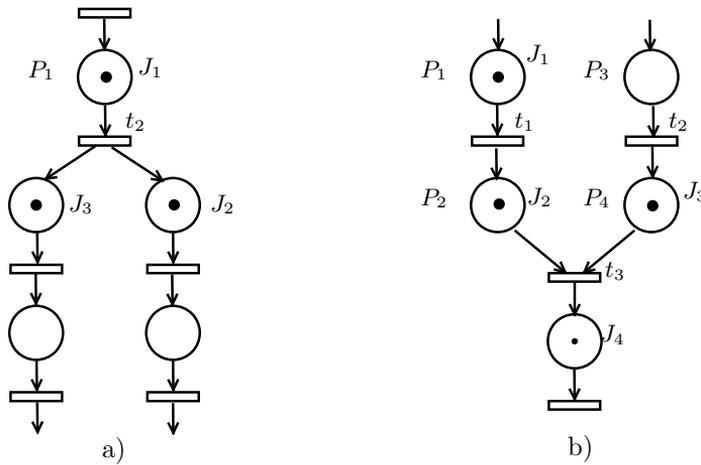


Figura 1.8: a) Divisão; b) Junção

requer, para iniciar, a presença simultânea de uma ficha em P_2 e uma ficha em P_4 . A esta transição pode estar associado, por exemplo, o evento *iniciar a montagem*. O lugar P_2 pode representar, neste caso, uma máquina livre, e o lugar P_4 , peças esperando num depósito, a quantidade sendo dada pelo número de fichas no lugar. O *desaparecimento* destas duas fichas (quando do tiro de t_3) será síncrono e criará no mesmo instante uma nova ficha que sofrerá a primeira operação de seu processo de fabricação (J_4). A passagem destas evoluções assíncronas a uma evolução síncrona implica necessariamente uma *espera* nos lugares P_2 ou P_4 , dependendo de qual condição é verdadeira em primeiro lugar: a peça espera que a máquina se libere, ou a máquina espera que haja uma peça no depósito.

1.7.3 Variantes e caminhos alternativos

A FIG. 1.9.a representa o caso em que, após uma fase (ou operação) P_1 , tem-se a escolha entre as seqüências P_2-P_3 ou P_4-P_5 . Em seguida, executa-se P_6 . Para executar a seqüência P_2-P_3 , a transição t_2 deve ser disparada, e para executar P_4-P_5 , a transição t_3 deve ser disparada. Embora as duas transições estejam sensibilizadas, apenas uma delas pode ser disparada, pois a ficha é retirada do lugar P_1 durante o disparo. A partir deste momento, a outra transição não pode mais disparar. Entretanto, a estrutura da rede de Petri não dá nenhuma informação sobre o mecanismo de tomada de decisões para a escolha da alternativa a efetuar (t_2 ou t_3). Ela se contenta de indicar que esta informação deve estar disponível no final da fase P_1 .

Se a rede modela o trecho do itinerário de um sistema de transporte — com o lugar P_i associado à travessia da seção S_i — o lugar P_1 representa o veículo numa seção que permite dois caminhos alternativos: passar pelas seções S_2 e S_3 ou pelas seções S_4 e S_5 antes de seguir pela seção S_6 .

Comparando-se a FIG. 1.9.a e a FIG. 1.8.a pode-se observar que:

- um lugar com mais de um arco de saída (P_1 na FIG. 1.9.a) corresponde ao início de um conjunto de caminhos alternativos;

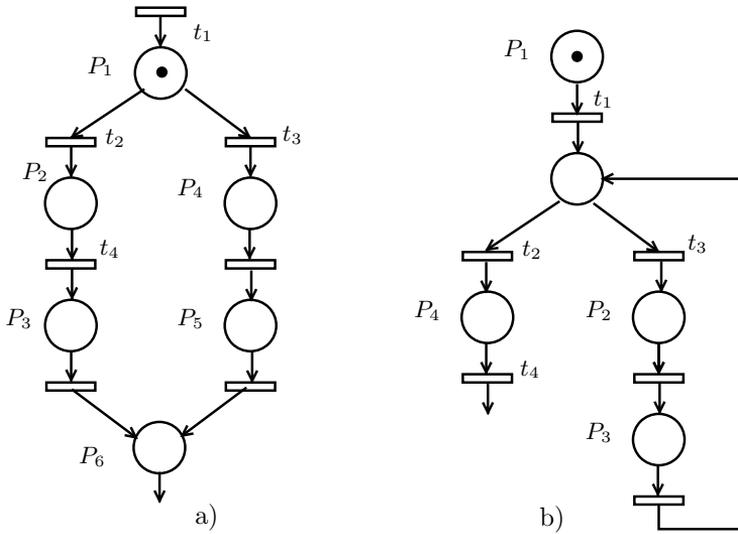


Figura 1.9: a) Caminhos alternativos; b) Repetição

- uma transição com mais de um arco de saída (t_2 na FIG. 1.8.a) corresponde a uma divisão ou início de evoluções paralelas.

1.7.4 Repetição

Outro comportamento que é necessário modelar quando se deseja representar sistemas a eventos discretos é a repetição de uma atividade (ou seqüência de atividades) enquanto uma condição for verdadeira. Por exemplo, um veículo deve repetir a passagem pelo circuito formado pelas seções S_2 e S_3 até que venha uma ordem de mudar o itinerário, ou que seja necessário carregar a bateria.

Considere a rede de Petri da FIG. 1.9.b (novamente, a cada lugar P_i é associado a travessia da seção S_i). Esta rede modela a possibilidade de, após P_1 , efetuar-se a seqüência P_2 - P_3 um certo número de vezes, antes de executar P_4 . Mais uma vez não há nenhuma indicação, no nível do grafo, sobre o teste que deve ser efetuado para decidir entre a repetição ou a finalização da seqüência. Sabe-se somente que o resultado deve estar disponível após o fim da operação P_3 (que pode precisamente ser a execução deste teste).

Considere novamente o exemplo do itinerário do sistema de transporte. O veículo deve realizar algumas vezes o circuito C_1 que passa pelas seções S_2 e S_3 , e após realizar uma vez o circuito C_2 , que passa pelas seções S_4 e S_5 , voltando então à seção inicial S_1 . Este comportamento pode ser modelado pela rede da FIG. 1.9.b: basta adicionar um lugar P_5 (seção S_5) como saída de t_4 , ligado ao lugar P_1 através de uma transição t_5 . A indicação de qual circuito (C_1 ou C_2) o veículo deve seguir é dada pelo nível de coordenação, e não faz parte da estrutura da rede.³ Esta deve apenas indicar as duas possibilidades existentes para o veículo.

³Este tipo de informação faz parte da *interpretação* e será visto no capítulo 4.

1.7.5 Alocação de recursos

Sem dúvida, a utilização de recursos, e principalmente o seu compartilhamento, é um dos pontos mais importantes na modelagem de um sistema. Um veículo que deve levar uma dentre diferentes peças num dado momento, um robô que pode levar uma peça do depósito para a máquina e da máquina para a saída pode executar apenas uma atividade de cada vez. Uma vez ocupado com uma delas, não pode estar disponível para a outra. Embora a idéia seja trivial, a verificação nem sempre o é. A modelagem do compartilhamento de recursos é, pois, fundamental para a representação correta de um sistema.

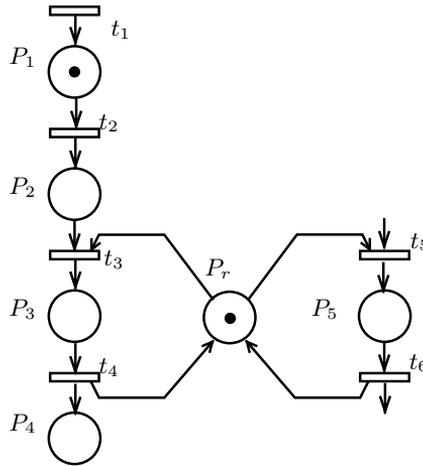


Figura 1.10: Partilhamento de um recurso

Observando a FIG. 1.10, considere que, após uma atividade P_1 , é preciso executar uma operação que necessita a utilização de um recurso r representado pelo lugar P_r na figura. P_r marcado corresponde ao estado parcial *recurso disponível*. A transição t_3 exprime a tomada do recurso e o início da fase P_3 .

Para modelar separadamente o *final de atividade* P_1 e o *início de* P_3 , é preciso introduzir a transição t_2 associada ao fim da atividade P_1 . O lugar P_2 corresponde à espera do recurso associado ao lugar P_r , se este não está disponível (este lugar faz um papel similar aos lugares P_2 e P_4 da rede da FIG. 1.8.b). O lugar P_2 permite representar o estado do sistema em que a atividade P_1 já foi executada, esperando que o recurso se libere para executar a atividade P_3 .

A ausência de t_2 e P_2 nesta rede modelaria um comportamento diferente. Como t_3 só pode disparar se o lugar P_r está marcado, a espera do recurso bloquearia, neste caso, o fim da atividade P_1 .

Cada utilização do recurso corresponde à execução de uma *malha*, que inicia com a *ocupação* do recurso (tiro de t_3 na FIG. 1.10) e termina com a sua *liberação* (tiro de t_4 na FIG. 1.10). Existem tantas malhas diferentes passando pelo lugar P_r quantas utilizações possíveis do recurso.

A rede FIG. 1.10 pode modelar o comportamento do robô descrito no item 1.4.1, que deve executar duas ações: i) retirar um objeto da esteira T_1 e colocar na esteira T_A ; ii) retirar um objeto de T_A e colocar em T_2 . Observe que o sensor não será representado na rede. O evento *retirar objeto de* T_1 é associado à transição t_3 , e o evento *retirar objeto*

de T_A é associado à transição t_5 . As transições t_4 e t_6 estão associadas, respectivamente, aos eventos *colocar objeto em T_A* e *colocar objeto em T_2* . Os lugares P_3 e P_5 representam a execução das ações i) e ii), respectivamente. As malhas modelando a ocupação e a liberação do robô são duas: t_3 - P_3 - t_4 - P_r e t_5 - P_5 - t_6 - P_r .

O recurso r é tratado exatamente como um artigo e a operação P_3 é análoga a uma operação de *montagem* do artigo em questão com r e depois de uma *divisão*. Assim, os conceitos de artigo e recurso, representados por fichas, são banalizados, bem como as operações com os estoques (lugar P_2) e as condições do tipo *recurso livre* (lugar P_r). Se o recurso r faz parte de um conjunto de recursos, então o lugar P_r representa diretamente este conjunto. Basta colocar inicialmente tantas fichas neste lugar quantos recursos elementares existem.

Exemplo resolvido 1:

Considere o itinerário de um sistema de transporte, baseado em veículos guiados auto-

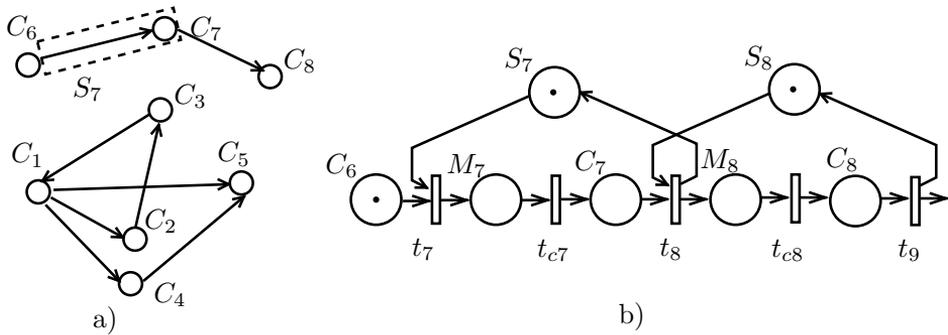


Figura 1.11: a) Sistema de transporte; b) Modelo do circuito N_0

maticamente (FIG. 1.11.a). Os veículos seguem automaticamente determinados circuitos pré-traçados; suas localizações são conhecidas somente nos pontos C_i , chamados *contatos*. Os comandos para *parar*, *continuar* e *mudar de itinerário* são enviados aos veículos quando estes estão sobre o contato.

Para evitar colisões, os circuitos são decompostos em seções de tal modo que pode haver somente um veículo por seção. Portanto, antes de entrar na próxima seção do circuito, é necessário verificar se esta está livre: a seção é, pois, considerada como um recurso a ser repartido entre os diferentes veículos. A seção aparece tracejada na figura, e possui o mesmo nome do contato.

Considere apenas o circuito N_0 , formado pelas seções S_6, S_7 e S_8 , modelado pela rede de Petri da FIG. 1.11.b. As seções são consideradas como recursos; a cada proposição S_i livre corresponde um lugar S_i . Existem dois predicados ainda associados à seção: *veículo em movimento*, associado ao lugar M_i e *veículo parado no contato*, associado ao lugar C_i . O evento *entrar na seção S_{i+1}* é associado à transição t_{i+1} . Sua ocorrência se dá quando um veículo está no contato da seção S_i (lugar C_i) e a próxima seção S_{i+1} está livre. Após o tiro da transição t_{i+1} , o veículo libera a seção S_i , e se encontra em movimento na seção S_{i+1} (lugar M_{i+1}). O evento *parar no contato C_i* é associado à transição t_{ci} . O estado inicial do sistema é um veículo no contato da seção S_6 e seções S_7 e S_8 livres (lugares C_6, S_7 e S_8 marcados).

Os circuitos podem possuir 4 tipos de seções: convergentes, divergentes, que se cruzam e em seqüência. Neste caso, além da noção de seção livre, é necessário introduzir a noção de célula livre. Deve-se, então, além de verificar se a próxima seção S_{i+1} do circuito está livre, verificar se a outra seção S' da célula está livre também. Caso contrário, durante a travessia da seção S_{i+1} , o veículo poderia cruzar um outro veículo que atravessa a seção S' .

Considere apenas o circuito N_1 , formado pelas seções S_1, S_2 e S_3 . O modelo é similar ao da FIG. 1.11.b; como este circuito é fechado, após atravessar a seção S_3 o veículo retorna à seção S_1 . Se considerado de forma independente, o circuito N_2 , formado pelas seções S_1, S_4 e S_5 , é também modelado por uma rede similar. Entretanto, há uma célula de cruzamento, formada pelas seções S_3 e S_5 , que deve ser considerada como um recurso. Antes de entrar numa seção desta célula, deve-se verificar se ela está livre, isto é, se não há nenhum veículo em movimento em S_3 e em S_5 . Pode, entretanto, haver um veículo parado no contato da outra seção, pois neste caso não haverá colisão. Por este motivo, a célula se libera quando o veículo chega no contato.

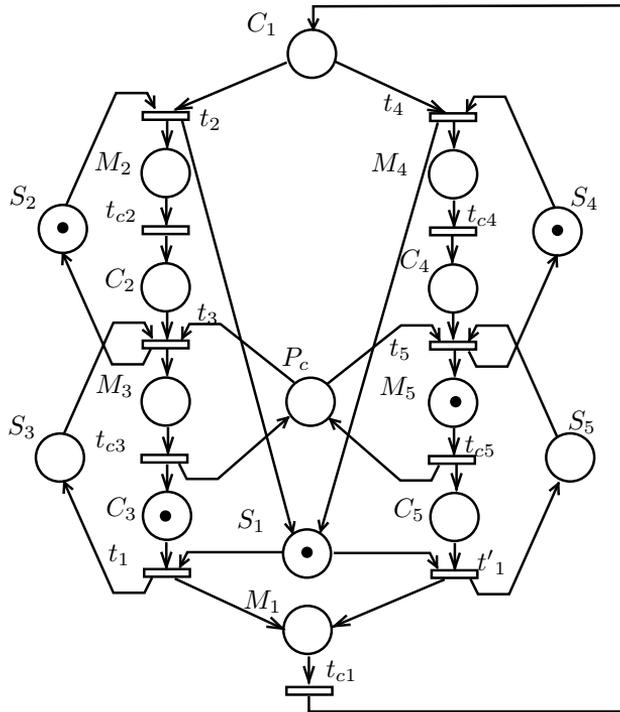


Figura 1.12: Modelo dos circuitos N_1 e N_2

A FIG. 1.12 representa o modelo completo dos circuitos N_1 e N_2 do sistema de transporte, em que o lugar P_c corresponde à célula de cruzamento. Observe que estando em C_1 (contato no final da seção S_1) o veículo pode seguir o circuito N_1 ou N_2 : a estrutura da rede indica apenas esta opção (transições t_1 e t'_1 em conflito). A marcação inicial da rede representa o estado do sistema em que um veículo está atravessando a seção S_5 enquanto o outro está parado no contato da seção S_3 . \diamond

Exemplo resolvido 2:

Um sistema do tipo *batch* pode produzir dois produtos (Pr_1 e Pr_2), utilizando dois reatores em modo concorrente, como representado na FIG. 1.13.a. O produto Pr_1 pode ser produzido pelo reator R_1 ou pelo reator R_2 , devendo ser, previamente, armazenado (respectivamente) no *buffer* B_1 ou B_2 . Já o produto P_2 pode ser produzido apenas pelo reator R_2 , onde é diretamente carregado. O comportamento do sistema é repetitivo: uma vez o produto pronto, cada reator é liberado e pode recomeçar uma nova atividade. O reator R_2 , embora possa tratar dois tipos de produtos, o faz um de cada vez. O modelo do comportamento do sistema é dado pela rede de Petri da FIG. 1.13.b. Os reatores R_1 e R_2 , representados pelos lugares p_8 e p_9 , respectivamente, são considerados como recursos, sendo R_2 partilhado entre os dois lotes de produto Pr_1 e Pr_2 .

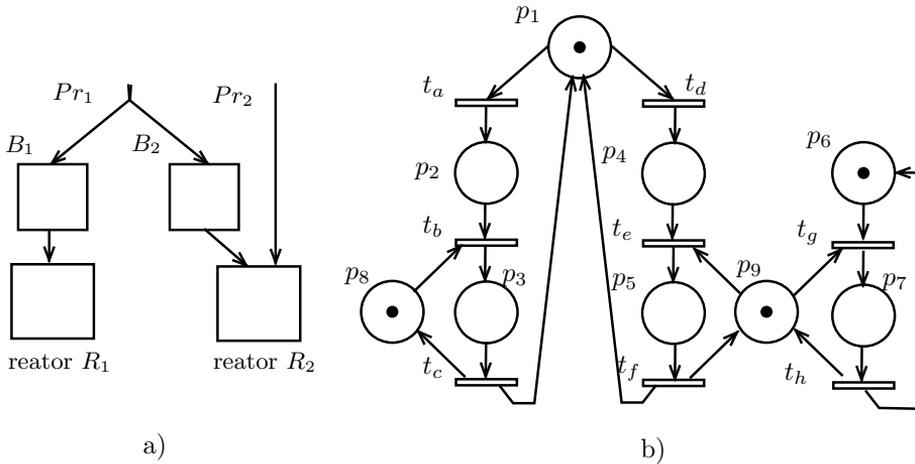


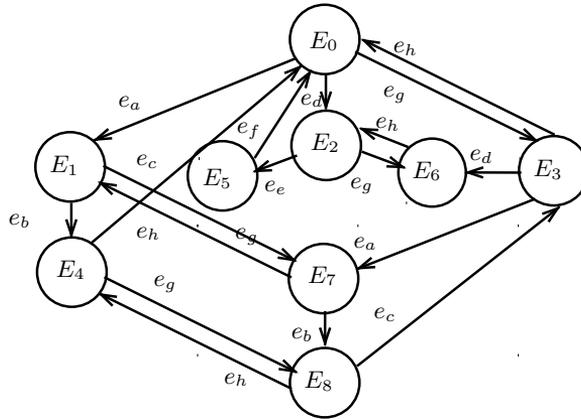
Figura 1.13: a) Sistema tipo *batch*; b) Modelo rede de Petri

Observe que, quando o lugar p_1 está marcado, indicando que há um lote do produto Pr_1 a ser produzido, há dois caminhos alternativos, representados pelas transições t_a e t_d , associadas, respectivamente, aos eventos *armazenar em B_1* e *armazenar em B_2* , pois o produto Pr_1 (lugar p_1) pode ser armazenado em ambos os *buffers*. O lugar p_2 (p_4) representa a atividade *produto Pr_1 armazenado no *buffer* B_1 (B_2)*, para posteriormente ser processado, respectivamente, pelo reator R_1 (lugar p_8) ou pelo reator R_2 (lugar p_9). A transição t_b (t_e) está associada ao evento *início de operação do reator R_1 (R_2) sobre o lote do produto Pr_1* .

A transição t_g está associada ao evento *início de operação do reator R_2 sobre o lote do produto Pr_2* e exprime a ocupação do recurso R_2 (lugar p_9), enquanto a transição t_h exprime a sua liberação (arco para o lugar p_9) e permissão para que um novo lote de Pr_2 possa ser processado. As transições t_e e t_f indicam também ocupação e liberação de R_2 , mas em relação ao produto Pr_1 .

Do ponto de vista do lugar p_9 , há também dois caminhos alternativos: processar o produto Pr_1 (tiro de t_e) ou processar o produto Pr_2 (tiro de t_g). A estrutura da rede modela apenas os comportamentos possíveis para o sistema, sem indicar qual escolha será feita.

Este sistema pode também ser modelado por uma máquina de estados finitos global, representada na FIG. 1.14. Os nós $E_i, i = 0 \dots 8$ são os diferentes estados do sistema, e



- E_0 : Pr_1 e Pr_2 esperando, reatores livres;
 E_1 : Pr_1 em B_1 , Pr_2 esperando, reatores livres;
 E_2 : Pr_1 em B_2 , Pr_2 esperando, reatores livres;
 E_3 : Pr_1 esperando, R_2 processando Pr_2 , R_1 livre;
 E_4 : R_1 processando Pr_1 , Pr_2 esperando, R_2 livre;
 E_5 : R_2 processando Pr_1 , Pr_2 esperando, R_1 livre;
 E_6 : Pr_1 em B_2 , R_2 processando Pr_2 , R_1 livre;
 E_7 : Pr_1 em B_1 , R_2 processando Pr_2 , R_1 livre;
 E_8 : R_1 processando Pr_1 , R_2 processando Pr_2 .

Figura 1.14: Grafo do sistema tipo batelada

os arcos indicam, para um estado E_i , qual o próximo estado E_{i+1} . Os eventos e_a, \dots, e_h , que etiquetam os arcos, são os mesmos eventos associados às transições t_a, \dots, t_h da rede.

Observe, como foi discutido no item 1.5, que a estrutura do sistema foi completamente perdida. Dois arcos saindo de um mesmo nó (estado) podem representar uma decisão entre duas opções ou dois eventos independentes. Por exemplo, os arcos do nó E_1 são etiquetados por e_b (alocação de R_1 por Pr_1) e e_g (alocação de R_2 por Pr_2), sendo estes eventos independentes, podendo então, ocorrer ao mesmo tempo. Já os eventos e_a e e_d , que etiquetam os arcos de saída do estado E_0 , não podem ocorrer simultaneamente, e são concorrentes.

No modelo rede de Petri, a escolha entre as duas opções de fabricação para o produto Pr_1 é claramente indicada na rede: o lugar p_1 possui duas transições de saída (t_a e t_d), indicando as duas alternativas (armazenamento em B_1 ou B_2). O mesmo acontece para o reator R_2 , que pode processar Pr_1 ou Pr_2 , representado pelas alternativas t_e e t_g . Em ambos os casos, estas transições estão em concorrência. Já as transições t_b e t_g , quando os lugares p_2 , p_6 , p_8 e p_9 estão marcados, podem ser disparadas de forma independente. Portanto, a rede de Petri indica, na própria estrutura, quando há paralelismo ou concorrência entre os eventos. É interessante retomar este exemplo após a leitura do capítulo 2, onde as noções de conflito e paralelismo são definidas formalmente.

◇

1.8 Notas

O conceito de sistema a eventos discretos é apresentado em CASSANDRAS (1993).

Para uma introdução à teoria de máquinas de estado finito, ver ROSEN (1991) e JOHNSONBAUGH (1993). Para um estudo completo e aprofundado, ver CARROL & LONG (1989). Na área de sistemas concorrentes, ANDREWS (1987) faz uma boa introdução para quem não conhece o assunto.

Leitura complementar para toda a Parte I, que trata do modelo básico: PETERSON (1981), BRAMS (1983), COURVOISIER & VALETTE (1986) e DAVID (1989).

1.9 Exercícios

Os exercícios abaixo são propostos a título de reflexão. Se encontrar dificuldades neste momento, deixe para fazê-los após a leitura do capítulo 2.

1. Modelar o sistema de triagem descrito no item 1.4.1 usando uma única rede de Petri. Tente modelar o sistema realizando uma decomposição material. Faça o mesmo usando uma decomposição funcional.
2. Um sistema de manufatura possui duas máquinas M_1 e M_2 . Dois tipos de processos de fabricação são previstos para uma peça, segundo a ordem de passagem nas máquinas: $P_1 = O_{11}O_{12}$ e $P_2 = O_{21}O_{22}$, em que O_{ij} é a j -ésima operação do processo de fabricação P_i . A máquina M_1 realiza as operações O_{11} e O_{22} , e a máquina M_2 , as operações O_{12} e O_{21} . Modele este sistema utilizando rede de Petri, considerando dois casos diferentes: a) antes de realizar a segunda operação de seu processo de fabricação, a máquina da primeira operação deve ser liberada; b) a máquina da primeira operação é liberada somente após o início da segunda operação de seu processo de fabricação.
3. Numa pequena cidade com um hidrante em cada rua, há dois bombeiros. Quando ocorre um incêndio, um dos bombeiros se coloca ao lado do hidrante, e outro ao lado do incêndio. O bombeiro ao lado do hidrante enche o balde e vai na direção do incêndio; o bombeiro ao lado do incêndio, uma vez esvaziado o balde, vai na direção do hidrante. Quando ambos se encontram, trocam de balde e de direção (o que vinha do hidrante volta para encher o balde vazio do colega; o que vinha do incêndio volta com o balde cheio que trazia o colega). Considere que, mesmo se o bombeiro que enche o balde seja i) muito rápido, ele não chegará ao incêndio antes que o outro esvazie o seu; ii) muito lento, ele já terá enchido o balde antes que seu colega chegue ao hidrante com o balde vazio. a) Modele, usando rede de Petri, o comportamento dos dois bombeiros; b) Considere que um terceiro bombeiro venha ajudar, ficando no meio do caminho para minimizar o deslocamento dos dois outros. Faça o modelo do novo sistema.
4. Modele, utilizando rede de Petri, a estrutura do controle de velocidade de um motor utilizando tiristores. Inicialmente deve ser feito a inicialização do sistema de controle (inicializar variáveis, conversores analógico-digitais e digital-analógicos, etc.). O ciclo de controle consiste em realizar as medidas de temperatura e de fluxo (de forma independente), e calcular após a lei de controle. No caso particular da medida de temperatura, se esta ultrapassar um limite L_T , um alarme deve ser acionado. Ao mesmo tempo que o ângulo de disparo é calculado e enviado depois

para o conversor digital-analógico, é realizado um cálculo para uma estatística posterior. Uma vez os cálculos terminados, o ciclo recomeça.

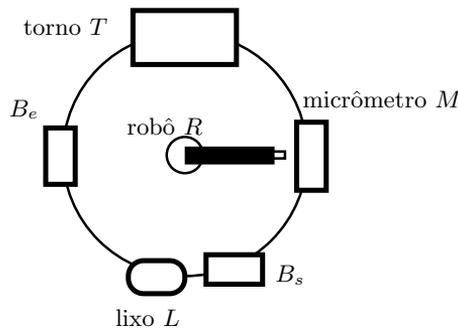


Figura 1.15: Célula de fabricação

5. Descrever, utilizando rede de Petri, uma célula de fabricação composta de um robô R , um torno T e um micrômetro laser M (FIG. 1.15), ligados entre si por uma rede de comunicação, cujo funcionamento é o seguinte:

- as peças são depositadas no *buffer* de entrada B_e ;
- o robô pega a peça de B_e e a coloca no torno T . Um programa deve ser carregado para tornear a peça;
- após torneada, o robô pega a peça do torno e a deposita no micrômetro M para inspeção;
- após realizada a inspeção, o robô R coloca a peça no *buffer* de saída B_s se está de acordo com as normas, ou no lixo L em caso contrário.

a) Considere que há somente um tipo de peça, e que o *buffer* tem capacidade para armazenar três peças; b) como ficaria a rede se houvesse dois tipos de peças?

Sugestão: antes de desenhar a rede de Petri, detalhe:

- as atividades e os eventos existentes no sistema;
- os recursos a serem compartilhados;
- quais as atividades que podem ser realizadas em paralelo e quais são realizadas de forma concorrente.

Capítulo 2

DEFINIÇÕES

Este capítulo apresenta a rede de Petri enquanto um modelo formal, através de três visões:

- um grafo com dois tipos de nós e um comportamento dinâmico;
- um conjunto de matrizes de inteiros positivos ou nulos cujo comportamento dinâmico é descrito por um sistema linear;
- um sistema de regras baseado numa representação do conhecimento sob a forma *condição* \rightarrow *ação*.

Os conceitos de marcação e tiro de transição, apresentados informalmente no capítulo anterior, são aqui definidos formalmente, para cada uma das visões acima.

As visões gráfica e matricial, apresentadas no item 2.1, diferenciam-se apenas pela forma de representação. Ambas permitem verificar se as transições são paralelas ou em conflito, se uma transição está ou não sensibilizada, bem como disparar uma transição e fazer evoluir a rede. O grafo é utilizado pelo projetista que, num relance, pode fazer todas estas verificações, e ter uma idéia global do sistema modelado. A representação matricial é uma representação natural, utilizada, pelo computador, na verificação automática.

Já a representação sob um sistema de regras tem o objetivo de compatibilizar a representação da rede de Petri com técnicas de Inteligência Artificial. Na especificação de um sistema complexo, utilizando uma hierarquia de controle em vários níveis, a modelagem do nível de coordenação dos subsistemas pode ser feita por redes de Petri, e do nível de planejamento, por um sistema perito que forneça a ordem de passagem das peças nas máquinas.

Embora a representação gráfica seja uma vantagem da rede de Petri, como o leitor já teve a possibilidade de constatar, a característica mais importante deste modelo é o fato de ser formal. A vantagem da rede de Petri em relação a outros modelos, como o SADT, que também oferece uma boa ferramenta gráfica de especificação, é que, sendo formal, é possível obter informações sobre o comportamento do sistema modelado, através da análise de suas propriedades, gerais ou estruturais. As propriedades gerais são apresentadas no item 2.3. Os componentes conservativos e repetitivos estacionários, que permitem a análise das propriedades estruturais, são definidas no item 2.4. Os métodos de análise destas propriedades são apresentados e discutidos mais tarde no capítulo 3.

2.1 Conceitos

2.1.1 Rede de Petri

Uma rede de Petri é uma quádrupla

$$R = \langle P, T, Pre, Post \rangle \quad (2.1)$$

onde:

- P é um conjunto finito de lugares de dimensão n ;
- T é um conjunto finito de transições de dimensão m ;
- $Pre : P \times T \rightarrow \mathbb{N}$ é a aplicação de *entrada* (lugares precedentes ou incidência anterior), com \mathbb{N} sendo o conjunto dos números naturais;
- $Post : P \times T \rightarrow \mathbb{N}$ é a aplicação de *saída* (lugares seguintes ou incidência posterior).

A quádrupla $R = \langle P, T, Pre, Post \rangle$ com $P = \{p_1, p_2, p_3\}$, $T = \{a, b, c, d\}$ e os valores das aplicações de entrada e saída dados por: $Pre(p_2, c) = 3$, $Pre(p_1, b) = Pre(p_2, a) = Pre(p_3, d) = 1$, $Post(p_2, d) = 3$ e $Post(p_1, a) = Post(p_2, b) = Post(p_3, c) = 1$, é uma rede de Petri.

2.1.2 Rede marcada

Uma rede marcada N é uma dupla

$$N = \langle R, M \rangle \quad (2.2)$$

onde:

- R é uma rede de Petri,
- M é a marcação inicial dada pela aplicação

$$M : P \rightarrow \mathbb{N}. \quad (2.3)$$

$M(p)$ é o número de marcas ou fichas (*jetons* em francês, *tokens* em inglês) contidas no lugar p . A marcação M é a distribuição das fichas nos lugares, sendo representada por um vetor coluna cuja dimensão é o número de lugares e elementos $M(p)$.

A dupla $N = \langle R, M \rangle$ com R sendo a rede de Petri do exemplo acima e marcação $M^T = [0 \ 3 \ 0]$ (T é o transposto do vetor) é uma rede de Petri marcada.

2.1.3 Grafo associado e notação matricial

Pode-se associar a uma rede de Petri um grafo com dois tipos de nós: nós lugares e nós transições. Um arco liga um lugar p a uma transição t se e somente se $Pre(p, t) \neq 0$. Um arco liga uma transição t a um lugar p se e somente se $Post(p, t) \neq 0$.

A partir dos elementos $a_{ij} = Pre(p_i, t_j)$ que indicam o peso do arco ligando o lugar de entrada p_i à transição t_j , define-se a matriz de incidência anterior Pre de dimensão $n \times m$: o número de linhas é igual ao número de lugares e o número de colunas é igual ao número de transições. Da mesma forma, a matriz de incidência posterior $Post$ de dimensão $n \times m$ é definida a partir dos elementos $b_{ij} = Post(p_i, t_j)$.

Os valores não nulos das matrizes Pre e $Post$ são associados aos arcos do grafo como etiquetas. Se o valor é unitário, não é necessário etiquetar o arco correspondente no grafo. Da mesma forma, se nada é indicado no grafo, o valor correspondente na matriz é unitário.

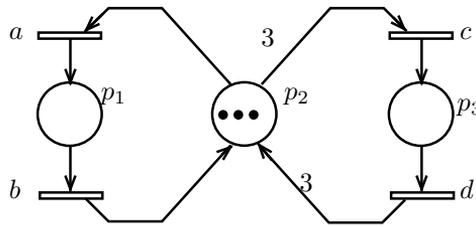


Figura 2.1: Rede de Petri

O grafo associado à rede de Petri marcada N do item anterior está representada na FIG. 2.1. Esta rede representa o partilhamento de um conjunto de recursos (três), representado pelo lugar p_2 , entre duas atividades, representadas pelos lugares p_1 e p_3 . A atividade correspondente a p_1 necessita de apenas um recurso de cada vez (o peso do arco (p_2, a) vale 1, ou $Pre(p_2, a) = 1$). Já a atividade correspondente a p_3 necessita de todos os três recursos ao mesmo tempo. Observe também que há uma exclusão mútua entre p_1 e p_3 : a partir do tiro de c , a transição a não pode mais disparar, e vice-versa. Entretanto, após o tiro de a , esta pode ainda disparar mais duas vezes. Pode, portanto, haver três atividades p_1 sendo executadas ao mesmo tempo.

A notação matricial desta rede é dada por:

$$\begin{aligned}
 \bullet \quad Pre &= \begin{matrix} & a & b & c & d \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & p_1 \\ & & & & p_2 \\ & & & & p_3 \end{matrix} \\
 \bullet \quad Post &= \begin{matrix} & a & b & c & d \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \end{bmatrix} & p_1 \\ & & & & p_2 \\ & & & & p_3 \end{matrix} .
 \end{aligned}$$

A partir de Pre e $Post$ defini-se a matriz de incidência C

$$C = Post - Pre \tag{2.4}$$

que fornece o *balanço* das fichas na rede quando do tiro das transições.

$$\text{Neste exemplo, tem-se } C = \begin{matrix} & \begin{matrix} a & b & c & d \end{matrix} \\ \begin{matrix} p_1 \\ p_2 \\ p_3 \end{matrix} & \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & -3 & 3 \\ 0 & 0 & 1 & -1 \end{bmatrix} \end{matrix}$$

Utiliza-se a notação $Pre(\cdot, t)$ para a coluna da matriz Pre associada a uma transição t . A dimensão deste vetor é dada pelo número de lugares. Da mesma forma, define-se os vetores coluna $Post(\cdot, t)$ e $C(\cdot, t)$ em relação às matrizes $Post$ e C , respectivamente.

2.1.4 Rede de Petri pura

Uma rede de Petri R é pura se e somente se

$$\forall p \in P \quad \forall t \in T \quad Pre(p, t)Post(p, t) = 0. \tag{2.5}$$

O grafo equivalente não possui nenhuma *malha* elementar, isto é, nenhuma transição possui um mesmo lugar como entrada e saída ao mesmo tempo.

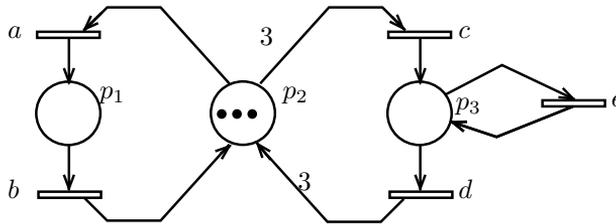


Figura 2.2: Rede de Petri não pura

Por exemplo, a rede de Petri da FIG. 2.1 é pura. Entretanto, a rede da FIG. 2.2 não o é, pois o lugar p_3 é lugar de entrada e também de saída da transição e : $Pre(p_3, e)Post(p_3, e) = 1$.

2.1.5 Transição sensibilizada

Uma transição t está sensibilizada ou habilitada (*franchissable* em francês, *enabled* em inglês) se e somente se:

$$\forall p \in P, \quad M(p) \geq Pre(p, t). \tag{2.6}$$

Isto é, se o número de fichas em cada um dos lugares de entrada for maior (ou igual) que o peso do arco que liga este lugar à transição.

A equação acima pode ser escrita na forma

$$M \geq Pre(\cdot, t)$$

onde o vetor coluna $Pre(\cdot, t)$ é a coluna da matriz Pre referente à transição t e M , o vetor marcação inicial.

Existem outras notações que exprimem que t está sensibilizada para uma dada marcação M :

$$\begin{aligned} M &\geq Pre(\cdot, t) \\ M(t &> \\ M &\xrightarrow{t} \cdot \end{aligned}$$

Na rede de Petri da FIG. 2.1, para uma marcação inicial:

$$M = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix}$$

com

$$Pre(\cdot, a) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad Pre(\cdot, c) = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix}$$

as transições a e c estão sensibilizadas, pois

$$M > Pre(\cdot, a) \text{ e } M = Pre(\cdot, c).$$

2.1.6 Disparo de uma transição

Se t é sensibilizada por uma marcação M , uma nova marcação M' é obtida através do tiro ou disparo (*tir* ou *franchissement* em francês, *firing* em inglês) de t tal que:

$$\forall p \in P, \quad M'(p) = M(p) - Pre(p, t) + Post(p, t). \quad (2.7)$$

A nova marcação M' é dada pela equação

$$M' = M - Pre(\cdot, t) + Post(\cdot, t) = M + C(\cdot, t). \quad (2.8)$$

O vetor coluna $Pre(\cdot, t_i)$ referente à transição t_i da matriz Pre , pode ser escrito como $Pre[1]_i$, produto da matriz Pre pelo vetor coluna $[1]_i$ (vetor com todos os elementos nulos, exceto o da linha i). A equação acima pode, então, ser reescrita como

$$M' = M - Pre[1]_i + Post[1]_i = M + C[1]_i. \quad (2.9)$$

As seguintes notações são utilizadas para representar a marcação obtida com o disparo de t :

$$\begin{aligned} M(t &> M' \\ M &\xrightarrow{t} M'. \end{aligned}$$

O disparo de t é uma operação que consiste em retirar $Pre(p, t)$ fichas de cada lugar precedente p (peso do arco de entrada) e colocar $Post(p, t)$ fichas em cada lugar seguinte p . Modificando a marcação, o disparo de t representa, no modelo, a mudança de estado ocorrida no sistema devido à ocorrência do evento associado à transição t .

Na rede de Petri da FIG. 2.1, após o tiro de a a partir da marcação inicial M , obtém-se, utilizando-se a equação 2.9, a marcação seguinte M' :

$$\begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}.$$

Se, a partir da marcação inicial M , a transição c fosse disparada, obter-se-ia a marcação $M'' = [0 \ 0 \ 1]^T$. Vê-se pela equação 2.9 que a matriz C faz um balanço das fichas quando do tiro da transição t . Cada elemento $C(p_j, t_i) < 0$ do vetor coluna $C[1]_i$ indica quantas fichas são retiradas de cada lugar p_j da rede quando do disparo de t . O elemento $C(p_j, t_i) > 0$ indica o número de fichas adicionadas à marcação de cada lugar p_j , quando do disparo de t . Se $C(p_j, t_i) = 0$, a marcação de p_j não é modificada.

2.1.7 Conflito e paralelismo

As noções de conflito e paralelismo de transições, apresentadas informalmente no item 1.7.2, serão definidas a seguir. Ter bem claro como uma e outra noção é representada na rede de Petri torna mais simples a tarefa de modelagem de um sistema. Uma vez feita a identificação da interação entre as atividades, basta fazer a tradução para o modelo.

O conflito pode existir na estrutura do modelo, sem todavia poder ser efetivado: é necessário que a marcação possibilite de fato sua ocorrência. O mesmo ocorre com o paralelismo. Por ocorrência do conflito entende-se a existência, num dado estado, de duas (ou mais) possibilidades, excludentes, de evolução. Já a ocorrência do paralelismo implica que todas as atividades poderão ser executadas ao mesmo tempo.

Conflito estrutural: duas transições t_1 e t_2 estão em conflito estrutural se e somente se elas têm ao menos um lugar de entrada em comum:

$$\exists p \in P, \quad Pre(p, t_1)Pre(p, t_2) \neq 0. \quad (2.10)$$

Conflito efetivo: duas transições t_1 e t_2 estão em conflito efetivo para uma marcação M se e somente se ambas estão em conflito estrutural e estão sensibilizadas:

$$M \geq Pre(., t_1) \text{ e } M \geq Pre(., t_2). \quad (2.11)$$

Paralelismo estrutural: duas transições t_1 e t_2 são paralelas estruturalmente se não possuem nenhum lugar de entrada em comum:

$$\forall p \in P \quad Pre(p, t_1)Pre(p, t_2) = 0 \quad (2.12)$$

que equivale a verificar se $Pre(., t_1)^T \times Pre(., t_2) = 0$, onde \times é o produto escalar de dois vetores.

Paralelismo efetivo: Duas transições t_1 e t_2 são paralelas para uma marcação dada M se e somente se são paralelas estruturalmente e:

$$M \geq Pre(., t_1) \text{ e } M \geq Pre(., t_2). \quad (2.13)$$

Assim, na rede de Petri da FIG. 2.1, as transições a e c estão em conflito estrutural, pois $Pre(p_2, a)Pre(p_2, c) = 3$. Para a marcação inicial M , as transições a e c estão em conflito efetivo, pois estão em conflito estrutural e $M_1 > Pre(., a)$ e $M_1 > Pre(., c)$.

Já as transições b e d são paralelas estruturalmente. Com efeito, como

$$Pre(., b) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \text{ e } Pre(., d) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

tem-se $Pre(., b)^T \times Pre(., d) = 0$.

Na mesma estrutura da rede da FIG. 2.1, mas agora com a marcação

$$M' = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix},$$

as transições b e d são efetivamente paralelas.

Na rede da FIG. 1.13.b, há um conflito entre as transições t_e e t_g , que é efetivo quando os lugares p_4 e p_6 estão marcados (existência de um lote de Pr_1 no *buffer* B_1 e um lote de Pr_2 esperando para serem processados pelo reator R_2). Este conflito modela o compartilhamento do recurso R_2 (lugar p_9).

O modelo de rede de Petri permite representar o paralelismo verdadeiro. Entretanto, no momento da implementação ou simulação, este pode se transformar num pseudoparalelismo. O importante é que o modelo permite tal representação. De qualquer forma, transições paralelas podem ser disparadas de forma independente entre si; o disparo de uma não impedindo o disparo da(s) outra(s) como no caso das transições em conflito.

2.1.8 Seqüência de disparo

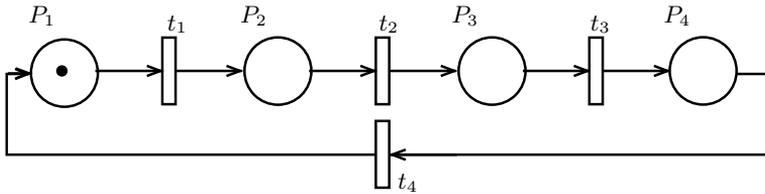


Figura 2.3: Seqüência de disparo de transições

Observe a rede de Petri da FIG. 2.3. A transição t_1 está sensibilizada pela marcação $M_0 = [1\ 0\ 0\ 0]$ e pode disparar, levando à marcação $M_1 = [0\ 1\ 0\ 0]$, $M_0 \xrightarrow{t_1} M_1$. Após seu disparo, a transição t_2 por sua vez pode disparar, levando à marcação $M_2 = [0\ 0\ 1\ 0]$, $M_1 \xrightarrow{t_2} M_2$. O disparo em seqüência das transições t_1 e t_2 é chamado *seqüência de disparo* e é notado, neste caso, por $s = t_1 t_2$. Diz-se que a seqüência $s = t_1 t_2$ é disparável a partir de M_0 , com a seguinte notação:

$$M_0 \xrightarrow{t_1 t_2} M_2$$

ou ainda:

$$M_0(t_1 t_2 > M_2).$$

Considere novamente a rede de Petri da FIG. 2.1. O tiro da seqüência $s = aab$, a partir da marcação M_0 , leva a rede à marcação M_1 , $M_0 \xrightarrow{aab} M_1$, passando sucessivamente pelas marcações:

$$\begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix}_{M_0} \xrightarrow{a} \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}_{M_1} \xrightarrow{a} \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix}_{M_2} \xrightarrow{b} \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}_{M_1}.$$

Vetor característico

Considere o vetor \mathbf{s} onde cada componente $s(t)$ representa o número de ocorrências da transição t numa seqüência de disparo s . Este vetor é chamado *vetor característico* da seqüência s . Sua dimensão é igual ao número de transições da rede de Petri.

No exemplo anterior, o vetor característico \mathbf{s} da seqüência $s = aab$ é dado por:

$$\mathbf{s} = \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} a \\ b \\ c \\ d \end{matrix}$$

$s(a) = 2$ indica que a transição a foi disparada duas vezes e $s(b) = 1$, que a transição b foi disparada somente uma vez na seqüência s . Os valores nulos de $s(c)$ e $s(d)$ indicam que as transições c e d não foram disparadas nesta seqüência.

A evolução da marcação de uma rede de Petri, com marcação inicial M , para uma seqüência $s = t_i t_j \dots t_l$, aplicando sucessivamente a equação 2.9 é dada por:

$$\begin{aligned} M' &= (\dots ((M + C[1]_i) + C[1]_j) + \dots + C[1]_l) \\ &= M + C([1]_i + [1]_j + \dots + [1]_l). \end{aligned} \quad (2.14)$$

A soma dos vetores $[1]_i + [1]_j + \dots + [1]_l$ corresponde exatamente ao vetor característico \mathbf{s} da seqüência s , indicando quantas vezes cada transição da rede foi disparada. Tem-se então:

$$M' = M + C\mathbf{s} \quad \text{com } M \geq 0, \mathbf{s} \geq 0. \quad (2.15)$$

Esta equação é chamada *equação fundamental* da rede de Petri.

Considerações sobre a existência de s

A caracterização da marcação na teoria de rede de Petri por meio de um vetor M é precisa (fornece toda a informação necessária sobre o estado). Já a representação de uma seqüência s através do vetor \mathbf{s} não considera a ordem do disparo das transições. De fato, o vetor \mathbf{s} é dado pela soma $[1]_i + [1]_j + \dots + [1]_l$; sendo a soma uma operação comutativa, qualquer permutação na ordem de disparo das transições $t_i, t_j \dots t_l$ dentro da seqüência será representado pelo mesmo vetor \mathbf{s} . A ordem de disparo é perdida, existindo, por conseguinte, uma perda de informação a respeito da evolução da rede. Somente é descrita a transformação da marcação.

Portanto, não deve causar surpresa que a existência de um vetor característico \mathbf{s} , solução da equação 2.15, não garanta que a seqüência s possa ser realmente disparada e, portanto, que exista de fato M' . Com efeito, é necessário que a marcação inicial seja tal que as transições sejam de fato disparadas para cada marcação intermediária. Basta observar a rede da FIG. 2.3: para a marcação M_0 da figura, a seqüência $t_1 t_2$, representada pelo vetor $\mathbf{s}^T = [1 \ 1 \ 0]$, é disparável (leva à marcação M_2), enquanto a seqüência $t_2 t_1$, representada pelo mesmo vetor \mathbf{s} , não pode ser disparada a partir de M_0 .

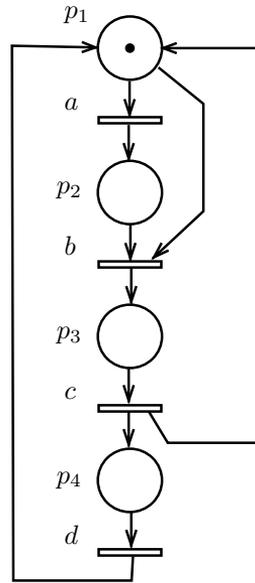


Figura 2.4: Rede com seqüência não disparável

Considere a rede de Petri da FIG. 2.4. A matriz C é dada por

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

Deseja-se saber se a seqüência $s = abcd$, representada pelo vetor \mathbf{s} , é disparável a partir da marcação inicial M ,

$$M = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{s} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Aplicando-se a equação 2.15, obtém-se a marcação

$$M' = M + C\mathbf{s} = [1 \ 0 \ 0 \ 0]^T.$$

Embora o valor de M' seja positivo, esta marcação não existe, pois a seqüência $abcd$ não pode disparar a partir de M . Basta usar a equação 2.7 para verificar que, a partir de M , após o tiro de a , nenhuma outra transição pode disparar. Assim, a existência de uma marcação $M' > 0$ através da equação 2.15 não implica a existência da seqüência s que, de fato, a geraria. Se $M = [2 \ 0 \ 0 \ 0]^T$ na rede da FIG. 2.4, obtém-se, aplicando-se a equação 2.15 com o mesmo vetor \mathbf{s} acima, a marcação $M' = [1 \ 1 \ 0 \ 0]^T$. Neste caso, a seqüência $s = abcd$ (nesta ordem) é, efetivamente, disparável, $M \xrightarrow{abcd} M'$. Entretanto, qualquer outra seqüência, obtida permutando-se a ordem das transições, não o é.

As questões que se colocam são as seguintes:

- dada uma marcação inicial M e uma seqüência s , existe uma marcação M' tal que $M \xrightarrow{s} M'$? Se $M' < 0$ (equação 2.15) não existe tal seqüência. Se $M' > 0$, tampouco se tem garantia da existência de s . Na melhor das hipóteses não se conhece a ordem de disparo das transições. No exemplo da FIG. 1.7, para $M_0 = [1\ 0\ 0]^T$ a seqüência $s = t_1 t_2 t_3$ é disparável, mas $s = t_2 t_3 t_1$ não o é. E ambas são caracterizadas pelo mesmo vetor $\mathbf{s}^T = [1\ 1\ 1]$;
- dada uma marcação inicial M e uma marcação qualquer M' , existe s tal que $M \xrightarrow{s} M'$? Se $s < 0$ não existe tal seqüência. Mas se $s > 0$ não se tem garantia da existência de s (ver invariantes no item 2.4 em que s é tal que $Cs = 0$; tem-se o caso particular em que $M' = M$);
- dada uma marcação inicial M e uma marcação M' conhecidas, a existência de um vetor $s > 0$ implica a existência de todas as seqüências s (obtidas permutando a ordem de tiro das transições) representáveis por este vetor? Como a ordem não é conhecida, algumas seqüências — para uma dada marcação — poderão existir e outras não.

A caracterização das seqüências de disparo na teoria de rede de Petri não é precisa. O capítulo 8 mostra como obter uma caracterização precisa de uma seqüência, utilizando a lógica linear.

2.1.9 Conjunto de marcações acessíveis

O conjunto de marcações acessíveis $A(R, M)$ de uma rede de Petri marcada é o conjunto das marcações que podem ser atingidas a partir da marcação inicial, através de uma seqüência de disparo:

$$A(R, M) = \{M_i, \exists s\ M \xrightarrow{s} M_i\}. \quad (2.16)$$

Se este conjunto é finito, pode-se representá-lo sob a forma de um grafo $GA(R, M)$, cujos nós são as marcações acessíveis do conjunto $A(R, M)$. Um arco orientado liga dois nós M_i e M_j se existe uma transição t sensibilizada que permite passar de uma marcação M_i a uma outra M_j : $M_i \xrightarrow{t} M_j$. Este arco é etiquetado pela transição t .

De fato, o grafo de marcações acessíveis é a máquina de estados *equivalente* à rede de Petri. Entretanto, é necessário observar que, tendo desaparecido a noção de processo, não se pode mais fazer, no grafo, a diferença entre as transições paralelas e as transições em conflito para uma dada marcação.

A FIG. 2.5 representa o grafo das marcações acessíveis para a rede de Petri da FIG. 2.1 e a marcação inicial $M = [0\ 3\ 0]^T$.

2.2 Rede de Petri e sistema de regras

A rede de Petri pode ser vista sob o aspecto gráfico ou matricial, como apresentado no item precedente. Dada a sua característica de fazer evoluir o estado, a partir de regras representadas por transições, a rede de Petri pode igualmente ser considerada como um sistema de regras de produção baseado em uma representação da forma

se condição então ação.

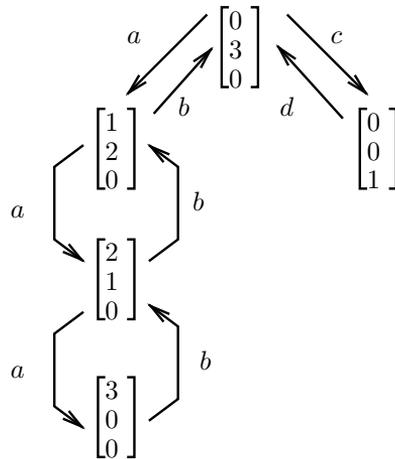


Figura 2.5: Grafo de marcações acessíveis

2.2.1 Sistema de regras

Um sistema de regras de produção de conhecimento, ou simplesmente sistema de regras, é formado de:

- uma base de fatos, representando o conhecimento disponível sobre o sistema;
- uma base de regras, que permite deduzir novos fatos;
- um mecanismo de inferência, chamado *motor de inferência*, que permite realizar novas deduções, aplicando as regras aos fatos.

O mecanismo de inferência mais elementar consiste em dispor o conjunto de regras numa lista, e percorrê-la seqüencialmente. Se a parte *condição* da regra é verdadeira no contexto atual (corresponde a um fato da base de fatos), a regra é aplicada ou disparada. Se nenhuma regra é aplicável ou se um fato terminal (conclusão que se deseja deduzir) é deduzido, o mecanismo de inferência pára.

Na maioria das vezes, em um dado contexto, várias regras podem ser aplicadas. O resultado da dedução, naturalmente, pode ser diferente segundo a regra escolhida. O mecanismo de inferência mais elementar escolhe a primeira regra, sem verificar se existem outras que podem também ser aplicadas.

Uma situação onde diversas regras podem ser aplicáveis constitui um *conflito*. A resolução de um conflito é realizado pelo *controle* do motor de inferência, que pode ser bastante complexo.

A correspondência entre a rede de Petri e os sistemas de regras é baseada nas correspondências seguintes:

- o conjunto de transições da rede, com suas condições e ações, representadas, respectivamente, pelos vetores $Pre(., t)$ e $Post(., t)$ (item 2.1.5), constituem a base de regras;
- a marcação inicial corresponde à base de fatos inicial, ou contexto inicial;

- o controle é dado pela estrutura da rede: se as transições são paralelas, a ordem de disparo é indiferente; se as transições estão em conflito efetivo, apenas uma poderá ser disparada.

A vantagem da utilização da rede de Petri para representar um sistema de regras é devida ao seu formalismo, que permite a análise. Um sistema de regras, diferente de um sistema descrito utilizando a lógica clássica, não permite a verificação formal. Entretanto, permite representar sistemas baseados em conhecimento e obter resultados quando não existe um algoritmo conhecido, como é o caso de sistemas peritos para prospecção de petróleo, diagnóstico médico, etc.

2.2.2 Gramática

Uma gramática é um sistema de reescrita de palavras de um dado alfabeto. Seja $S = \langle \mathbb{P}, Q \rangle$ a gramática associada à rede de Petri $R = \langle P, T, Pre, Post \rangle$, definida por:

- um alfabeto \mathbb{P} cujos símbolos são os lugares $p \in P$, e
- um conjunto Q de regras de reescrita

$$t : \mu(Pre(., t)) \rightarrow \mu(Post(., t)). \quad (2.17)$$

Tome M como sendo o conjunto de todas as marcações da rede. A aplicação $\mu : M \rightarrow \mathbb{P}^*$ associa, a um vetor marcação da rede, uma palavra do vocabulário \mathbb{P}^* . \mathbb{P}^* é o conjunto de seqüências finitas de elementos de P , inclusive a seqüência vazia λ . Para cada marcação $M(p) = i$, $i \in \mathbb{N}$, dos lugares $p \in P$ da rede, é associada a ocorrência de um símbolo de \mathbb{P}^* ; a palavra $\mu(M)$ é obtida através da concatenação dos símbolos p^i . Por abuso de linguagem, esta palavra é, neste livro, também representada por M .

De modo geral, uma aplicação $\mu : \mathbb{N}^m \rightarrow \mathbb{P}^*$ associa, a um vetor v de \mathbb{N}^m , uma palavra $\mu(v)$ de \mathbb{P}^* .

Uma transição t está sensibilizada por uma marcação M se e somente se:

$$\mu(Pre(., t)) \subseteq \mu(M). \quad (2.18)$$

O tiro de t leva a uma nova marcação M' , dada por:

$$\mu(M') = \mu(M) \cup \mu(Post(., t)) \setminus \mu(Pre(., t)). \quad (2.19)$$

A expressão $v \cup w$ indica, aqui, a concatenação de palavras v e w , e $v \setminus w$, a exclusão da palavra w da palavra v .

Considere a rede de Petri da FIG. 2.1. A gramática correspondente é definida pelo:

- alfabeto $P = \{p_1, p_2, p_3\}$, e

$$\bullet \text{ conjunto de regras de reescrita } Q = \begin{cases} a : p_2 \rightarrow p_1 \\ b : p_1 \rightarrow p_2 \\ c : p_2^3 \rightarrow p_3 \\ d : p_3 \rightarrow p_2^3 \end{cases}.$$

À marcação inicial $M = [0\ 3\ 0]^T$ da rede, corresponde o axioma $\mu(M) = p_2^3$ da gramática, a partir do qual novas palavras podem ser derivadas (por abuso de linguagem, será utilizada a notação $M = p_2^3$).

A equação 2.18 permite verificar que, para a marcação $M = p_2^3$, apenas as transições a e c estão sensibilizadas, pois $\mu(\text{Pre}(\cdot, a)) = p_2 \subseteq M$ e $\mu(\text{Pre}(\cdot, c)) = p_2^3 \subseteq M$. Se a transição a é disparada, obtém-se, aplicando a equação 2.19: $M' = p_2^3 \cup p_2 \setminus p_1 = p_2^2 p_1 = p_1 p_2^2$. Vale aqui uma observação: a ordem de escrita das letras nas palavras do vocabulário \mathbb{P}^* não é importante. Para facilidade de leitura, costuma-se escrever os lugares em ordem alfanumérica.

Utilizando-se sucessivamente a equação 2.19, obtém-se o conjunto de marcações acessíveis:

$$A(R, M) = \{p_2^3, p_1 p_2^2, p_1^2 p_2, p_1^3, p_3\}.$$

Uma seqüência de disparo s também pode ser representada como uma palavra de um vocabulário T^* , formado a partir de um alfabeto de transições T . Diferentemente das palavras do vocabulário \mathbb{P}^* , numa palavra $s \in T^*$, a ordem das letras $t \in T$ é importante, pois indica a ordem de disparo de t na seqüência s . Para a rede da FIG. 2.1, a partir da marcação inicial M , as seqüências a^3 , $a^2 b$, $abab$, cd (entre outras) fazem parte do vocabulário efetivo da rede, mas as seqüências ba , dc , ac , ca (entre outras) são proibidas.

Embora a regra de reescrita $a : p_1 \rightarrow p_2 p_3$ seja bastante similar à fórmula da lógica proposicional $p_1 \rightarrow p_2 \wedge p_3$, a rede de Petri não pode ser utilizada como semântica para a lógica clássica. Este assunto é tratado no capítulo 8.

2.3 Propriedades do modelo

Neste item serão definidas as propriedades relativas à rede de Petri marcada: vivacidade, limitabilidade e reiniciabilidade, reagrupadas sob o nome genérico de *boas propriedades*. Suas definições implicam considerações sobre o conjunto de marcações acessíveis a partir da marcação inicial. As redes de Petri marcadas não permitem a obtenção direta de algoritmos capazes de determinar se a propriedade é ou não verificada, pois o conjunto de marcações acessíveis não é sempre finito. Os métodos de análise destas propriedades baseados no grafo de marcações serão apresentados no capítulo 3.

As propriedades relativas às redes de Petri não marcadas (que independem, portanto, da sua marcação inicial) permitem derivar métodos de cálculo, diretamente das definições, através da resolução de um sistema de equações lineares. Estas propriedades serão definidas no item 2.4 no que concerne aos componentes conservativos e repetitivos estacionários.

2.3.1 Rede marcada k-limitada

Lugar k-limitado e lugar binário

Um lugar p de uma rede marcada N é k -limitado se e somente se:

$$\forall M' \in A(R, M), \quad M'(p) \leq k. \quad (2.20)$$

Se $k = 1$ diz-se que o lugar é *binário* (*safe* em inglês).

Na rede de Petri da FIG. 2.1, cujo grafo de marcações é representado na FIG. 2.5, para a marcação inicial $M = p_2^3$ (ou $M = [0 \ 3 \ 0]^T$), o lugar p_3 é binário, enquanto os lugares p_1 e p_2 são 3-limitados.

Rede de Petri marcada k-limitada e rede marcada binária

Uma rede marcada N é k -limitada (*bounded*) se e somente se todos os seus lugares são k -limitados.

Uma rede marcada N é *binária* se e somente se todos os seus lugares são binários. Diz-se também salva ou segura.

A rede de Petri da FIG. 2.1, para a marcação $M = p_2^3$, é 3-limitada, ou, limitada com $k = 3$. Se sua marcação inicial fosse $M = p_2$ a rede seria binária (as transições c e d entretanto não seriam nunca disparadas).

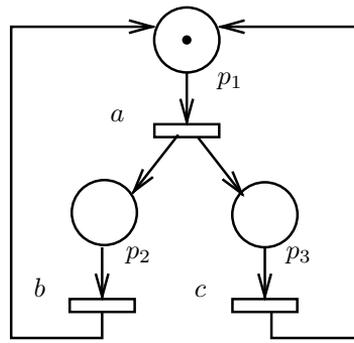


Figura 2.6: Rede de Petri não limitada

Considere a rede de Petri da FIG. 2.6, com a marcação inicial $M = p_1$. A cada vez que a seqüência $s = ab$ é disparada, uma ficha é introduzida no lugar p_3 . Este lugar, portanto, é não limitado e por sua vez a rede também não o é. Os lugares p_1 e p_2 são também não limitados.

Um exemplo de rede não limitada que descreve um mecanismo clássico é dado pela FIG. 2.7. Se a abertura de um parêntese (parêntese esquerdo) é associada ao tiro da transição a e o fechamento de um parêntese (parêntese direito) ao tiro da transição b , as seqüências de tiros de transição que levam, a partir da marcação inicial $M = p_1$, à mesma marcação M , descrevem todas as expressões “parenteseadas” permitidas. É claro que esta rede não é limitada pois sempre se pode abrir um parêntese (disparar a) e portanto colocar mais uma ficha no lugar p_1 .

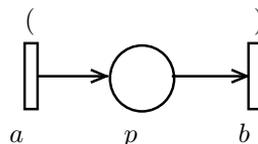


Figura 2.7: Rede parênteses

A propósito da propriedade da rede ser limitada, ou em particular, salva, deve-se salientar que:

- a rede de Petri considerada aqui é sempre uma rede com marcação inicial dada (rede marcada);
- o conceito de rede limitada corresponde ao fato de que um sistema físico é sempre limitado. Entretanto, pode-se ser levado a utilizar uma rede de Petri não limitada quando se quer avaliar o desempenho de um sistema independentemente dos limites de seus elementos de armazenamento intermediários.
- o conceito de rede salva apresenta o interesse seguinte: se os lugares representam condições lógicas, a presença de mais de uma ficha num lugar significa que uma incoerência foi introduzida no modelo. Em geral, trata-se de uma condição lógica colocada a 1 durante um ciclo de funcionamento, que não foi utilizada (colocada a 0) e que é recolocada a 1 no ciclo seguinte. Numa rede binária, apenas um processo é representado; as transições paralelas (efetivamente) que aparecem são devidas às divisões existentes em algum momento, mas que se sincronizam em seguida (ver discussão seção 2.1.7);

2.3.2 Rede marcada viva

Transição quase viva

Uma transição t de uma rede marcada $N = \langle R, M \rangle$ é quase viva se e somente se existe uma seqüência de disparo s tal que

$$\exists s \mid M \xrightarrow{s} M' \text{ e } M' \xrightarrow{t} . \quad (2.21)$$

Ou seja, a transição t é quase viva se é possível sensibilizá-la por uma marcação M' do grafo de marcações acessíveis, obtida, a partir da marcação inicial M , através do tiro de uma seqüência de transições s , eventualmente vazia ($s = \lambda$). Diz-se que t pode ser sensibilizada a partir de M através do disparo da seqüência s . A equação acima pode ser reescrita sob a forma $\exists s \mid M \xrightarrow{st}$.

Transição viva

Uma transição t de uma rede marcada $N = \langle R, M \rangle$ é viva se e somente se:

$$\forall M' \in A(R, M), \exists s \mid M' \xrightarrow{st} . \quad (2.22)$$

Para ser viva, a transição t deve poder ser sensibilizada a partir de qualquer marcação M' do grafo de marcações acessíveis, através de uma seqüência s de disparo.

Considere a rede de Petri da FIG. 2.8. O grafo das marcações acessíveis associado à marcação inicial $M = p_3 p_4$ é dado pela FIG. 2.9. Vê-se facilmente que a transição d é quase viva, pois pode ser disparada uma vez (para a marcação $M = p_3 p_4$). Entretanto, não é viva: após este disparo, d é bloqueada, não sendo mais sensibilizada por nenhuma marcação. Já as transições a , b , c , e e f são vivas, pois para qualquer marcação acessível a partir de M , é possível encontrar uma seqüência s de disparo que leva a uma marcação que as sensibiliza.

É necessário tomar cuidado porque uma transição pode, sem ser viva, aparecer uma infinidade de vezes em seqüências infinitas de disparo de transição. É o caso da transição g na FIG. 2.10. Ela pode ser sempre disparada antes do disparo de d , mas não poderá mais ser disparada após o disparo de d .

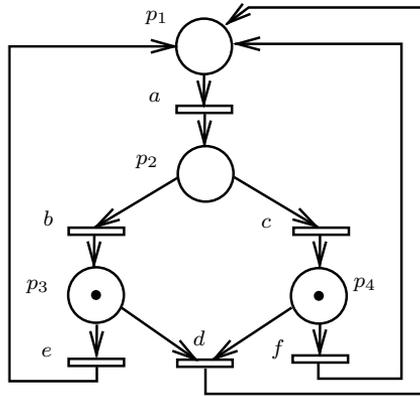


Figura 2.8: Transição quase viva e não viva

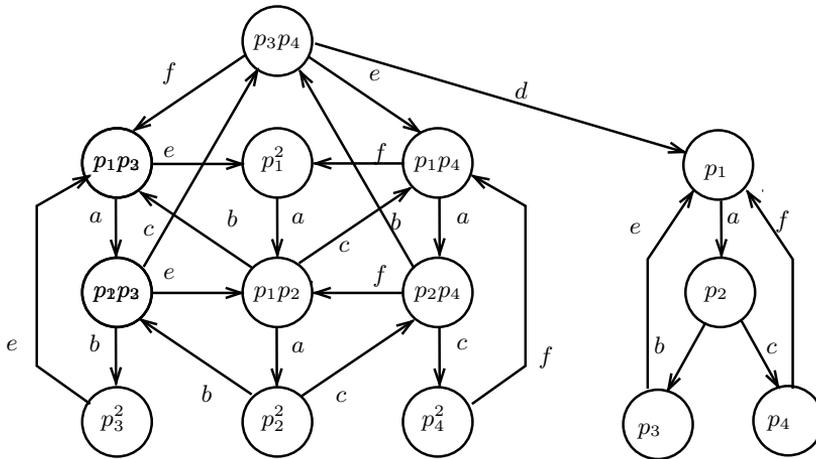


Figura 2.9: Grafo das marcações (transição quase viva)

Rede marcada viva

Uma rede de Petri marcada $N = \langle R, M \rangle$ é viva se e somente se todas as suas transições são vivas:

$$\forall t \in T, \forall M' \in A(R, M), \exists s \mid M' \xrightarrow{st} . \tag{2.23}$$

É importante observar que:

- somente é considerada aqui a rede de Petri marcada;
- uma rede de Petri viva garante que nenhum bloqueio pode ser provocado pela estrutura da rede. No entanto, ela não prova a ausência de eventuais bloqueios provocados por uma má interação entre a rede de Petri e seu ambiente externo (ver mais adiante no capítulo 4 os problemas ligados à interpretação da rede de Petri);

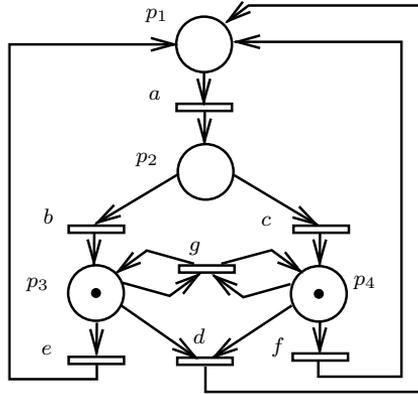


Figura 2.10: Transição quase viva e seqüência infinita

- uma rede de Petri viva garante também a ausência de partes mortas (nunca atingidas), valendo aqui a mesma observação a respeito da interpretação.

Por exemplo, a rede de Petri da FIG. 2.1 é viva para a marcação inicial $M = [0\ 3\ 0]^T$. Já a rede da FIG. 2.8, para $M = p_3p_4$ não o é, pois a transição d não é viva (é quase viva).

Uma rede de Petri pode muito bem ser não limitada e viva: é o caso, por exemplo, da rede da FIG. 2.7 (a seqüência $a\ b$ é sempre disparável qualquer que seja a marcação).

2.3.3 Rede marcada reiniciável

Uma rede marcada $N = \langle R, M \rangle$ é reiniciável (ou própria) se e somente se:

$$\forall M' \in A(R, M), \quad \exists s \mid M' \xrightarrow{s} M. \tag{2.24}$$

É possível, portanto, a partir de qualquer marcação acessível M' de $GA(R, M)$, encontrar uma seqüência de disparo s que leve a rede de volta à marcação inicial M . A maioria dos sistemas possuem funcionamentos repetitivos e, portanto, as redes de Petri utilizadas para representá-los são reiniciáveis.

Considere a rede marcada da FIG. 2.11.a, cujo grafo de marcações acessíveis é dado pela FIG. 2.11.b. Esta rede é não reiniciável, pois não existe nenhuma seqüência de tiro que permita voltar à marcação inicial $M = p_1p_4$ após o tiro da transição a . No entanto, é interessante observar que a rede é viva, pois a partir de qualquer marcação acessível, existe uma seqüência de disparo levando a uma marcação que sensibiliza cada uma das transições da rede.

Mas para qualquer uma das seguintes marcações iniciais: $M = p_2p_4$, $M = p_1p_3$ ou $M = p_2p_3$, a rede da FIG. 2.11.a é, ao mesmo tempo, viva e reiniciável. Inversamente, a rede de Petri da FIG. 2.8 é reiniciável para a marcação $M = p_1$ sem ser viva, pois a transição d nunca será disparada.

Mais uma vez, é importante salientar que as propriedades que foram definidas são fortemente ligadas à marcação inicial. Considere, por exemplo, a rede de Petri da FIG. 2.12. Para a marcação $M = p_1p_3$ a rede é binária, viva e reiniciável. Se uma ficha é adicionada no lugar p_4 a rede deixa de ser limitada (ver a seqüência $abab\dots$). Enfim para a marcação $M = p_1$ a rede é binária e não viva (nenhuma transição é disparável).

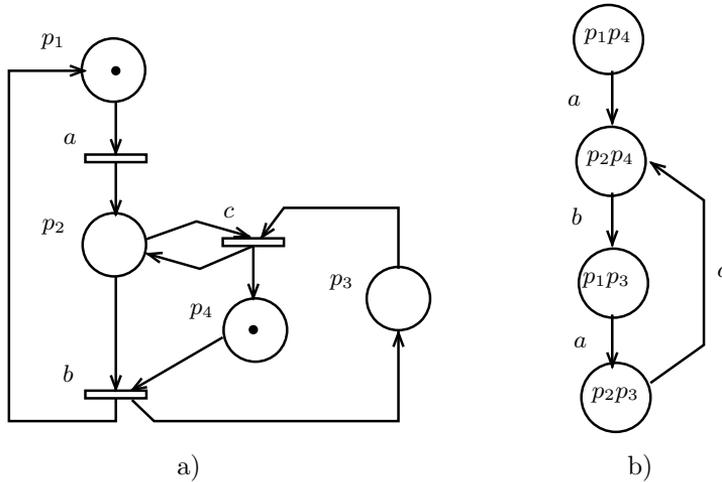


Figura 2.11: a) Rede não reiniciável; b) Grafo associado

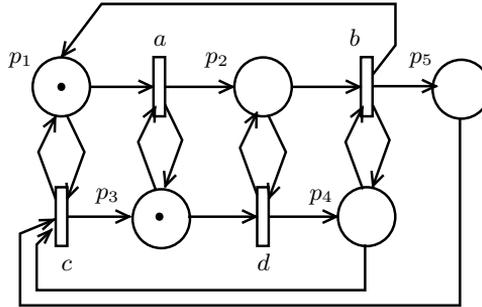


Figura 2.12: Rede com propriedades dependentes da marcação inicial

2.4 Propriedades estruturais

Nesta seção serão consideradas as propriedades derivadas diretamente da *estrutura* da rede de Petri e que, portanto, não dependem da sua marcação inicial. Estas propriedades são definidas através dos componentes conservativos de lugar e dos componentes repetitivos estacionários. A partir destes elementos estruturais, pode-se utilizar também a informação sobre a marcação, definindo-se, assim, os invariantes de lugar e de transição, que permitem algumas informações adicionais sobre o comportamento dinâmico da rede de Petri.

2.4.1 Componentes conservativos, invariantes de lugar

Considere a rede de Petri da FIG. 2.13. Observe o circuito formado pelos lugares p_1 e p_2 e pelas transições a e b . A soma:

$$M(p_1) + M(p_2)$$

vale 1 para a marcação inicial $M_0 = [1\ 0\ 3\ 0\ 1]^T$. O tiro de a não modifica em nada esta soma, da mesma forma que o de b , embora a marcação de cada lugar seja modificada a

cada tiro de transição. O tiro das outras transições da rede (c e d) também não modifica esta soma. Para este exemplo, pode-se verificar que, para todas as marcações acessíveis a partir da marcação inicial, tem-se $M(p_1) + M(p_2) = 1$, ou de modo mais geral:

$$\forall M \in A(R, M_0), \quad M(p_1) + M(p_2) = M_o(p_1) + M_o(p_2).$$

A forma linear $M(p_1) + M(p_2) = M_o(p_1) + M_o(p_2)$ é chamada *invariante linear de lugar*, pois a soma (em geral, ponderada) das fichas se conserva para estes lugares. O conjunto de lugares p_1 e p_2 forma um *componente conservativo* da rede.

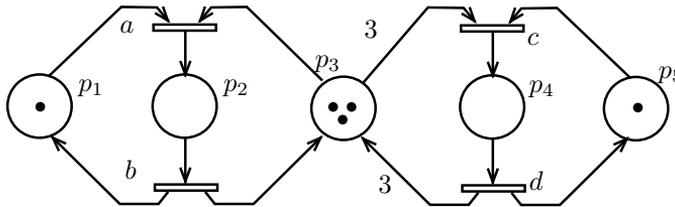


Figura 2.13: Invariantes

Para esta mesma rede, considerando agora o circuito formado pelos lugares p_2 , p_3 e p_4 e pelas transições a , b , c e d , pode-se verificar que o conjunto p_2 , p_3 e p_4 forma um componente conservativo, com o invariante de lugar

$$\forall M \in A(R, M_0), \quad M(p_2) + M(p_3) + 3.M(p_4) = 3.$$

Um *invariante linear* de lugar é uma função linear da marcação dos lugares cujo valor é uma constante que depende apenas da marcação inicial da rede. Ele corresponde a uma restrição sobre os estados e as atividades do sistema que será sempre verificada, quaisquer que sejam suas evoluções.

A equação que permite determinar a evolução da rede é a equação fundamental (equação 2.15). Para obter-se uma soma ponderada das marcações, pré-multiplica-se a equação por um vetor \mathbf{f}^T :

$$\mathbf{f}^T M' = \mathbf{f}^T M + \mathbf{f}^T C s. \quad (2.25)$$

É fácil verificar que a única maneira de tornar a equação independente das seqüências de disparo s é anular o produto $\mathbf{f}^T C$.

Um *componente conservativo* de uma rede de Petri é o conjunto de lugares $p_i \in P$ correspondentes aos elementos não nulos f_i do vetor coluna \mathbf{f} , solução da equação:¹

$$\mathbf{f}^T C = 0 \quad \text{com } \mathbf{f} > 0. \quad (2.26)$$

Do ponto de vista gráfico, um *componente conservativo* define uma *sub-rede* de Petri. Consideram-se somente os lugares, para os quais o componente correspondente de \mathbf{f} é não nulo, e as transições de entrada e de saída destes lugares. As transições possuem o mesmo número de arcos de entrada e de saída (nos casos simples).

Uma rede de Petri é dita *conservativa* se todos os lugares da rede pertencem a um componente conservativo.

¹Veja apêndice B para o cálculo da solução \mathbf{f} .

Se \mathbf{f} é solução da equação 2.26, então a função linear:

$$\mathbf{f}^T M = \mathbf{f}^T M_0 \quad \forall M \in A(R, M_0), \quad (2.27)$$

é o *invariante linear de lugar* correspondente.

O invariante linear de lugar, obtido através da equação 2.27, depende da marcação inicial, enquanto que o componente conservativo, obtido a partir da equação 2.26, é completamente independente. A matriz C da rede da FIG 2.11.a (somente a estrutura!) é dada por:

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

Utilizando-se a equação 2.26, encontra-se $\mathbf{f}^1 = [1 \ 1 \ 0 \ 0]^T$ e $\mathbf{f}^2 = [0 \ 0 \ 1 \ 1]^T$. Portanto, a rede é conservativa, pois $f(p) \neq 0$ para todo lugar p . Substituindo-se o valor de \mathbf{f}^1 e \mathbf{f}^2 na equação 2.27, para a marcação inicial $M_0 = [0 \ 1 \ 0 \ 1]^T$, encontram-se os invariantes lineares de lugar

$$\begin{aligned} I_1 : \quad & M(p_1) + M(p_2) = M_0(p_1) + M_0(p_2) = 1 \\ I_2 : \quad & M(p_3) + M(p_4) = M_0(p_3) + M_0(p_4) = 1. \end{aligned}$$

O invariante I_1 indica que, para qualquer marcação acessível a partir de M_0 , a soma das fichas em p_1 e p_2 será sempre 1. Portanto, numa marcação M em que $M(p_2) = 1$, o número de fichas em p_1 será nulo. Além disto, sabe-se que nunca haverá mais de uma ficha em p_1 e em p_2 . O invariante de lugar permite, sem enumerar todas as marcações acessíveis, obter algumas informações sobre a propriedade de limitabilidade da rede de Petri, como será visto no capítulo 3.

2.4.2 Componentes repetitivos, invariantes de transição

Considere, novamente na FIG. 2.13, a sub-rede formada pelas transições c e d juntamente com seus lugares de entrada ou saída (p_3, p_4 e p_5). Observe que o disparo da seqüência $s = cd$ a partir da marcação inicial leva de volta à mesma marcação.

A seqüência $s = cd$ é um *invariante de transição*, pois o tiro de tal seqüência não modifica a marcação da rede. O invariante de transição corresponde a uma seqüência cíclica de eventos que pode ser repetida indefinidamente. O conjunto das transições c e d do invariante forma um *componente repetitivo estacionário* da rede. A seqüência $s = ab$ é também um invariante de transição da rede da FIG. 2.13.

Para encontrar o conjunto de transições que, uma vez disparadas a partir de uma marcação da rede, faz retornar a esta mesma marcação, utiliza-se novamente a equação fundamental (2.15). É fácil verificar que, para obter-se $M' = M$, a seqüência s deve ser tal que o vetor \mathbf{s} verifique:

$$C\mathbf{s} = 0. \quad (2.28)$$

Toda solução \mathbf{s} desta equação é chamada *componente repetitivo estacionário*, e a seqüência s é dita invariante de transição. Uma rede de Petri é dita *repetitiva* se todas as transições $t \in T$ pertencem a um componente repetitivo estacionário. Um componente repetitivo estacionário define uma sub-rede em que se considera somente as transições para as quais

o componente correspondente de \mathbf{s} é não nulo, com seus lugares de entrada e de saída. As outras transições ligadas a estes lugares não são consideradas.

Aplicando-se a matriz C da rede de Petri da FIG 2.11.a na equação 2.28 encontra-se o vetor $\mathbf{s}^T = [1 \ 1 \ 1]$. A rede é, pois, repetitiva.

Se \mathbf{s} é o vetor característico de uma seqüência s efetivamente disparada a partir de uma marcação acessível, então esta seqüência s é um *invariante de transições*.

O componente repetitivo estacionário não depende da marcação inicial, ao contrário do invariante de transição correspondente de quem sua existência depende .

Exemplo resolvido 2: (continuação)

Considere o sistema tipo batelada do exercício resolvido no capítulo 1 (FIG. 1.13). Os componentes conservativos desta rede são dadas pelos vetores $\mathbf{f}^1 = [1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0]^T$, $\mathbf{f}^2 = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0]^T$, $\mathbf{f}^3 = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1]^T$ e $\mathbf{f}^4 = [0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0]^T$, que definem, respectivamente, as seguintes sub-redes, formadas pelos:

- lugares p_1, p_2, p_3, p_4 e p_5 , e transições t_a, t_b, t_c, t_d, t_e e t_f : esta sub-rede representa as duas opções de fabricação do produto Pr_1 ;
- lugares p_6 e p_7 , com as transições t_g e t_h : esta sub-rede representa a fabricação do produto Pr_2 ;
- lugares p_3 e p_8 , com as transições t_b e t_c : esta sub-rede representa as evoluções de estado do reator R_1 ; (se ocupado, lugar p_3 marcado; se livre, lugar p_8 marcado);
- lugares p_5, p_7 e p_9 , com transições t_e, t_f, t_g e t_h : esta sub-rede representa as evoluções de estado do reator R_2 e sua possível alocação a um dos dois produtos.

Esta rede de Petri descreve o sistema como uma coleção de processos comunicantes (as sub-redes definidas acima), e, além do funcionamento do sistema, mostra também a sua estrutura. Portanto, fornece, ao mesmo tempo, um modelo comportamental e estrutural do sistema.

Os invariantes de lugar I_1 a I_4 , obtidos a partir dos vetores \mathbf{f}^1 a \mathbf{f}^4 e da marcação inicial $M_0 = [1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1]^T$ permitem verificar que de fato o sistema se comporta como esperado:

- I_1 : $M(p_1) + M(p_2) + M(p_3) + M(p_4) + M(p_5) = 1$: o lote Pr_1 pode estar em espera (lugar p_1), em B_1 (p_2), em B_2 (p_4), sendo processado pelo reator R_1 (lugar p_3) ou reator R_2 (lugar p_5). O fato de ter somente uma ficha num dos lugares $p_1 \dots p_5$ indica que o sistema efetivamente estará em apenas um destes estados;
- I_2 : $M(p_6) + M(p_7) = 1$: existe um lote Pr_2 esperando (p_6) ou sendo processado pelo reator R_2 (p_7);
- I_3 : $M(p_3) + M(p_8) = 1$: o reator R_1 está livre (p_8) ou processando Pr_1 (p_3);
- I_4 : $M(p_5) + M(p_7) + M(p_9) = 1$: o reator R_2 está livre (p_9), processando Pr_1 (p_5) ou processando Pr_2 (p_7).

As informações fornecidas pelos invariantes são importantes para uma rede de maior tamanho, onde nem sempre é evidente verificar, sem fazer cálculos, se dois ou mais lugares que não podem estar marcados ao mesmo tempo, de fato não o estão.

As soluções da equação 2.28 são os vetores $\mathbf{s}^1 = [1\ 1\ 1\ 0\ 0\ 0\ 0\ 0]^T$, $\mathbf{s}^2 = [0\ 0\ 0\ 1\ 1\ 1\ 0\ 0]^T$ e $\mathbf{s}^3 = [0\ 0\ 0\ 0\ 0\ 0\ 1\ 1]^T$, que formam as sub-redes com as transições:

- t_a, t_b e t_c , com os lugares p_1, p_2, p_3 e p_8 : uma vez que o lote Pr_1 passa pelo *buffer* e é processado por R_1 , um novo ciclo pode recomeçar;
- t_d, t_e e t_f , com os lugares p_1, p_4, p_5 e p_9 : mesmo comportamento em relação a Pr_2 e R_2 ;
- t_g e t_h , com os lugares p_6, p_7 e p_9 : indica a seqüência de eventos início de processamento de Pr_2 e seu fim.

A rede é repetitiva pois $s(t) \neq 0$ para todas as transições $t \in T$. Os invariantes de transição correspondentes são as seqüências $s_1 = t_a t_b t_c$, $s_2 = t_d t_e t_f$ e $s_3 = t_g t_h$.

O grafo de marcações acessíveis desta rede é o mesmo da FIG. 1.14 (capítulo 1). Escrivendo as marcações como palavras do vocabulário \mathcal{IP}^* , tem-se a seguinte equivalência; $E_0 = p_1 p_6 p_8 p_9$, $E_1 = p_2 p_6 p_8 p_9$, $E_2 = p_4 p_6 p_8 p_9$, $E_3 = p_1 p_7 p_8$, $E_4 = p_3 p_6 p_9$, $E_5 = p_5 p_6 p_8$, $E_6 = p_4 p_7 p_8$, $E_7 = p_2 p_7 p_8$ e $E_8 = p_3 p_7$.

2.5 Notas

BITTENCOURT (1996) apresenta, de forma clara, os sistemas de regras e a lógica proposicional, além de outros conceitos deste campo de conhecimento. O método de especificação de sistemas SADT pode ser encontrado no livro de JALOTE (1991).

2.6 Exercícios

1. Revise os exercícios de modelagem propostos no capítulo anterior. Verifique se os recursos, as atividades paralelas e concorrentes (em conflito) foram corretamente considerados. Para cada modelo, encontre: a) a representação matricial; b) a representação gramatical (alfabeto \mathcal{IP} e regras de reescrita \mathcal{Q}); c) o conjunto de marcações acessíveis, utilizando a equação 2.9; d) os componentes conservativos e repetitivos estacionários; e) os invariantes de lugar e transição.
2. Para a rede da FIG. 2.1, verifique quais as seqüências de fato disparáveis. Mude a ordem de algumas delas (de modo que o vetor característico seja o mesmo) e verifique quais não são disparáveis.

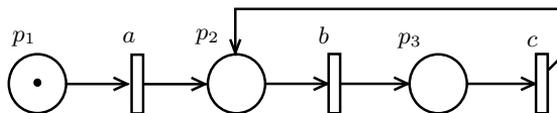


Figura 2.14: Exercício 3

3. Verifique, usando o método da enumeração de marcações, se a rede da FIG. 2.14 é viva e limitada. A partir do cálculo dos componentes repetitivos, verifique se a rede é repetitiva.

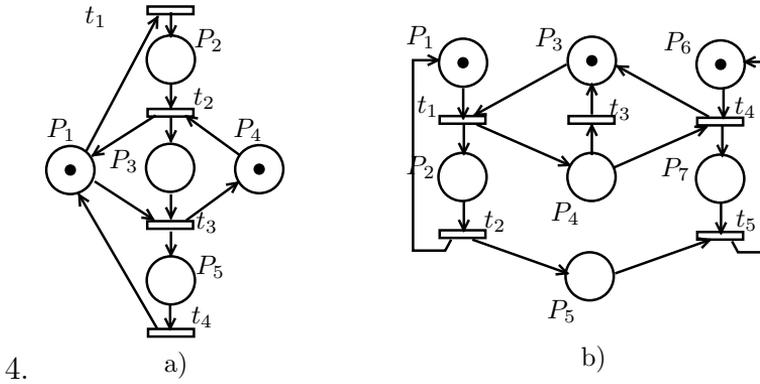


Figura 2.15: Exercício 4

- a) Encontre os componentes conservativos e repetitivos (e respectivos invariantes de lugar e transição) das redes da FIG. 2.15; b) Encontre o conjunto de marcações acessíveis de cada rede. A rede da FIG. 2.15.a descreve um robô (livre, se P_1 está marcado) utilizado para carregar (P_2) e descarregar (P_5) uma fresadeira (livre se P_4 está marcado); a operação de fresagem está associada ao lugar P_3 .

Capítulo 3

ANÁLISE DAS PROPRIEDADES

Este capítulo apresenta três formas de análise do comportamento de uma rede de Petri: por enumeração das marcações, estrutural e por redução. Para o primeiro método, serão apresentados os algoritmos que permitem determinar se uma dada rede de Petri possui as *boas* propriedades. De fato, embora as propriedades de rede viva e reiniciável tenham sido mostradas como decidíveis, nenhum algoritmo que seja prático foi encontrado no caso geral. Serão dados apenas os princípios da prova de decidibilidade relativa à propriedade de rede limitada, e após será apresentado um método para determinar as outras propriedades. Em seguida, outros métodos que permitem evitar parcialmente a enumeração das marcações serão apresentados.

A análise estrutural é realizada através do cálculo dos componentes conservativos e repetitivos estacionários e dos invariantes correspondentes. Este método, embora nem sempre concludente, permite, em alguns casos, obter, de forma mais simples e rápida, as propriedades da rede.

Quando não é possível obter as propriedades da rede através da análise estrutural, e a rede é tal que é muito custoso enumerar as marcações, é possível obter uma rede reduzida através do método de redução. A análise é, então, realizada sobre o modelo reduzido.

Por fim, serão descritas as relações entre os diferentes métodos, e a ordem em que se deve utilizá-los para obter as propriedades do sistema modelado.

3.1 Análise por enumeração de marcações

Este método é válido somente para a rede marcada. Portanto, para analisar o comportamento do modelo para outra marcação inicial, é necessário construir um novo grafo. A vantagem deste método é que, se é possível construir o grafo, todas as propriedades podem ser encontradas, como será visto a seguir.

É necessário inicialmente observar que é equivalente dizer que a rede marcada é limitada e que o número de suas marcações acessíveis é finito, pois o número de lugares é finito.

Da mesma forma, é equivalente dizer que o conjunto de marcações acessíveis não é finito e que existe ao menos uma seqüência infinita de disparos que, quando aplicada, não leva a nenhuma marcação da rede já calculada. Tal seqüência é construída a partir de uma seqüência finita, mas a cada aplicação desta, uma nova marcação é alcançada. Portanto, o conjunto de marcações acessíveis não é finito. Não se deve confundir uma

seqüência infinita de disparos com uma seqüência finita que se repete indefinidamente, levando sempre, a cada aplicação, a uma mesma marcação do conjunto de marcações acessíveis.

3.1.1 Decidibilidade da propriedade k-limitada

Ao enumerar-se as marcações acessíveis, se este número é finito, ocorrerá a parada do procedimento de cálculo. Entretanto, se o número de marcações é infinito, o cálculo jamais terminará. Não é, portanto, evidente, *a priori*, que a propriedade de k-limitada seja decidível. A decidibilidade decorre das seguintes propriedades:

1. Monotonia:

$$\exists M, \exists M', \exists s \mid M \xrightarrow{s} e M' > M \Rightarrow M' \xrightarrow{s} .$$

Quando uma evolução é possível a partir de uma marcação dada, ela continua *possível* se a marcação é “aumentada” (fichas são adicionadas em certos lugares).

2. Se $\exists M, M' \in A(R, M) \mid M \xrightarrow{s} M'$ e $M' > M$ então a rede de Petri marcada não é k-limitada. De fato, deduz-se imediatamente do ponto precedente que a seqüência s pode ser repetida tantas vezes quantas se deseja, e a cada vez, o conteúdo de fichas em pelo menos um lugar é aumentado, pois M' é estritamente maior que M .
3. Lema de Karp e Miller:

em toda seqüência infinita de vetores formados por inteiros positivos ou nulos $v_1, v_2, \dots, v_k, \dots$ é tal que ela contém ao menos dois elementos (de fato, uma infinidade de pares) v_i e v_j com $i < j$ tais que $v_i < v_j$ ou $v_i = v_j$.

O resultado é evidente para os vetores que possuem somente um componente: é impossível existir uma seqüência infinita de inteiros positivos ou nulos que seja estritamente decrescente.

4. Considere agora uma seqüência infinita s de disparos de transições. A seqüência das marcações é uma seqüência infinita de vetores de inteiros positivos ou nulos. O lema anterior implica, portanto, a existência de uma subseqüência s_1 tal que

$$M \xrightarrow{s_1} M' \text{ e } M' \geq M$$

e que pode ser repetida indefinidamente.

As seqüências definidas no ponto 2 são, portanto, características das seqüências de comprimento infinitas não estacionárias (que nunca passam pelas mesmas marcações). Elas são características do fato de uma rede não ser limitada, isto é, o conjunto de marcações acessíveis $A(R, M)$ contém uma infinidade de elementos.

Árvore de cobertura

O algoritmo que permite decidir se uma rede de Petri marcada é k -limitada é baseado na construção de uma árvore chamada *árvore de cobertura*. Parte-se da marcação inicial e cada transição sensibilizada por esta marcação dá origem a um ramo. As marcações obtidas através do disparo das transições são calculadas e o processo recomeça para cada marcação obtida. A construção de um ramo é interrompida desde que seja encontrada uma marcação:

- igual a uma outra já calculada e para a qual todos os sucessores já foram ou serão calculados. O segundo caso corresponde a ramos pendentes que serão explorados em seguida (senão uma subárvore já calculada seria reexplorada);
- estritamente superior a uma marcação *do ramo que está sendo explorado*.

No segundo caso, pára-se a exploração da árvore, pois a rede marcada pode não ser limitada. O valor $M'(p)$ que torna a marcação estritamente superior é notado w . Senão continua-se até que todos os ramos tenham sido explorados. Desta forma pode-se afirmar com certeza que a rede marcada é limitada e que o conjunto das marcações acessíveis é conhecido. Este procedimento caracteriza um algoritmo, pois:

- não pode ter nenhum ramo de comprimento infinito (lema de Karp e Miller);
- o número de ramos é finito, pois para cada marcação o número de transições sensibilizadas é finito (inferior ao número de transições da rede de Petri).

3.1.2 Procura das outras propriedades

Viu-se que a propriedade de limitabilidade é decidível para uma rede de Petri marcada. A construção da árvore de cobertura fornece um algoritmo que responde a esta questão.

Não existe algoritmo geral para as outras propriedades, embora elas sejam, do ponto de vista teórico, decidíveis. Entretanto, na maior parte do tempo, deseja-se que a rede construída possua simultaneamente todas as *boas* propriedades. O primeiro passo é verificar se a rede é k -limitada, e construir o grafo de marcações acessíveis. Depois, utilizar os algoritmos de procura em grafos (ver apêndice A para verificar se a rede é reiniciável ou viva).

Reiniciabilidade

Uma rede marcada $N = \langle R, M \rangle$ é reiniciável se e somente se seu grafo de marcações acessíveis $GA(R, M)$ for fortemente conexo:

$$\forall M_i, M_j \in GA(R, M), \exists s \mid M_i \xrightarrow{s} M_j.$$

Como a marcação inicial M faz parte do grafo, e o grafo é fortemente conexo, existe um caminho (seqüência s de transições) entre qualquer par de nós (marcações). No caso, o que interessa é um caminho entre qualquer marcação $M_i \in GA(R, M)$ e a marcação inicial M .

Vivacidade

Uma rede marcada $N = \langle R, M \rangle$ é viva se e somente se o grafo de marcações acessíveis $GA(R, M)$ for fortemente conexo e todas as transições $t \in T$ etiquetarem ao menos um arco do grafo:

$$\forall t \in T \exists M_i, \exists M_j \in GA(R, M), \mid M_i \xrightarrow{t} M_j.$$

Portanto, se a rede é reiniciável, basta provar que é quase viva. Se toda transição $t \in T$ etiqueta um arco (M_i, M_j) (rede quase viva), basta verificar se existe uma seqüência que, após M_j , permita novamente chegar a M_i , e disparar t . Ora, se o grafo é fortemente conexo (rede reiniciável), existe um caminho entre (M_j, M) e (M, M_i) , e a rede é, portanto, viva.

Se o grafo não for conexo (rede não reiniciável), a rede é viva se, em cada componente pendente fortemente conexo (ou subgrafo conexo), todas as transições $t \in T$ etiquetarem ao menos um arco.

Os passos para realizar a análise a partir da enumeração de marcações são os seguintes:

1. verificar se a rede marcada é limitada pela construção da árvore de cobertura;
2. se a rede marcada é limitada, construir o grafo de marcações acessíveis;
3. mostrar que o grafo de marcações acessíveis é fortemente conexo;
4. verificar que toda transição aparece ao menos uma vez como etiqueta de um arco do grafo de marcações acessíveis.

Este encaminhamento permite verificar se a rede marcada é ao mesmo tempo limitada, reiniciável e viva.

Considere o grafo de marcações da FIG. 2.9, referente à rede da FIG. 2.8. A rede não é reiniciável, pois o grafo não é fortemente conexo. As transições a, b, c, e e f são vivas pois figuram nos dois componentes fortemente conexos pendentes (subgrafos). A transição d é quase viva, pois não aparece em nenhum dos subgrafos. A rede, portanto, não é viva (é quase viva).

Já a rede da FIG. 2.11.a, cujo grafo de marcações acessíveis é dado pela FIG. 2.11.b, é viva, pois dentro do componente fortemente conexo pendente formado pelas marcações $M = p_2p_4$, $M = p_1p_3$ e $M = p_2p_3$, todas as transições da rede (a, b e c) etiquetam um arco.

3.2 Análise estrutural

O método de análise estrutural, a partir dos componentes repetitivos e conservativos, vistos no item 2.4, permite obter informações sobre o comportamento da rede não marcada. A vantagem é que os resultados são válidos para qualquer marcação. Entretanto, não é possível obter uma informação concludente quando $f(p) = 0$ e quando $s(t) \neq 0$.

Os invariantes de lugar e transição, obtidos, respectivamente, a partir dos componentes conservativos e repetitivos estacionários, dependem da marcação inicial. Entretanto, seu cálculo é simples e rápido, bastando substituir o valor de M_0 na equação 2.27 e escrever as seqüências a partir dos componentes repetitivos. Mas vale a mesma observação acima: o resultado nem sempre é concludente.

3.2.1 Componentes conservativos, invariantes de lugar

Os componentes conservativos permitem considerar apenas a estrutura da rede de Petri, isto é, a rede não marcada. São as seguintes as informações obtidas acerca da propriedade de limitabilidade:

- qualquer lugar que pertença a um componente conservativo é limitado;
- mesmo que um lugar p não pertença a nenhum componente conservativo ($f(p) = 0$), p pode ser limitado; portanto, uma rede não conservativa pode ser limitada;
- um lugar não limitado não pertence a nenhum componente conservativo; portanto, uma rede não limitada é não conservativa;
- uma rede conservativa é limitada (para qualquer marcação) já que todos os seus lugares são limitados.

Embora possa, à primeira vista, parecer estranho que uma propriedade aparentemente tão ligada à marcação possa ser obtida a partir apenas da estrutura da rede, lembre-se que a matriz C na equação 2.26 é quem faz o balanço das fichas na rede, indicando o número de fichas colocadas e retiradas de cada lugar.

A análise estrutural, do ponto de vista da conservação do número de fichas na rede, fornece os dois resultados a seguir:

Cobertura de lugar \times limitada:

Uma rede de Petri para a qual existe uma cobertura de componentes conservativos¹ $f^T > 0$ é k -limitada *qualquer que seja sua marcação inicial*.

Invariantes de lugar \times valor do limite:

A forma linear $f^T M = f^T M_0$ permite calcular um limite para cada lugar. Com efeito, como os componentes de f e as marcações são números não negativos, tem-se para um lugar p qualquer:

$$f(p)M(p) \leq f^T M_0$$

e por conseqüência, o seguinte limite (que não é necessariamente atingido):

$$M(p) \leq \frac{f^T M_0}{f(p)}. \quad (3.1)$$

Como foi visto no item 2.4.1, os invariantes de lugar da rede da FIG 2.11.a, para a marcação inicial $M_0 = [0 \ 1 \ 0 \ 1]^T$ são dados por:

$$\begin{aligned} I_1 : \quad & M(p_1) + M(p_2) = 1 \\ I_2 : \quad & M(p_3) + M(p_4) = 1. \end{aligned}$$

O invariante I_1 mostra que os lugares p_1 e p_2 são binários, pois para qualquer marcação acessível a partir de M_0 , o número máximo de fichas é 1. Outra informação fornecida

¹Uma *cobertura* de componentes conservativos (respectivamente, repetitivos estacionários) é formada por um componente *positivo* — ou um conjunto de componentes elementares *positivos* — que passa por todos os lugares (respectivamente transições).

por este invariante é que, quando o lugar p_1 está marcado, o número de fichas em p_2 é nulo (o mesmo raciocínio vale para o invariante I_2 e os lugares p_3 e p_4). Esta informação vai além da noção de limitabilidade: ela mostra a relação que existe entre as marcações dos lugares. Para estabelecer esta relação entre todos os lugares da rede, basta calcular a combinação linear $\mathbf{f} = \mathbf{f}^1 + \mathbf{f}^2 = [1 \ 1 \ 1 \ 1]^T$, e encontrar o invariante linear de lugar correspondente (para $M_0 = [0 \ 1 \ 0 \ 1]^T$):

$$M(p_1) + M(p_2) + M(p_3) + M(p_4) = 2.$$

A rede é conservativa, pois todos os lugares pertencem a um componente conservativo.

Exemplo resolvido 2: (continuação)

Considere o sistema tipo batelada do exercício resolvido no capítulo 1 (FIG. 1.13) cujos componentes conservativos foram calculados no capítulo 2.

O suporte dos componentes conservativos é dado por $\mathbf{f} = \mathbf{f}^1 + \mathbf{f}^2 + \mathbf{f}^3 + \mathbf{f}^4 = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]^T$. Como todos os lugares pertencem ao suporte (de fato, $\forall p \in P, f(p) \neq 0$), a rede é conservativa. Portanto, sem realizar o grafo de marcações acessíveis, a análise estrutural permite provar que todos os lugares são limitados, e que a rede é limitada. Não é necessário conhecer a marcação inicial da rede; este resultado é válido qualquer que seja a marcação inicial.

Os invariantes I_1 a I_4 (exercício resolvido do item 2.4.2) permitem calcular um limite superior para a marcação de cada lugar. Este limite pode não acontecer nunca; o que se garante é que a marcação será sempre menor ou igual. O cálculo dos invariantes é válido somente para a marcação inicial utilizada na equação 2.27. A partir do invariante I_1 , para $M_0 = [1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1]^T$, obtém-se a equação:

$$\forall M \in A(R, M), \quad M(p_1) + M(p_2) + M(p_3) + M(p_4) + M(p_5) = 1$$

e pode-se verificar que, para qualquer marcação acessível, o limite máximo de fichas, para os lugares p_1, p_2, p_3, p_4 e p_5 , é unitário. Mais uma vez, sabe-se que o limite é unitário, mas não se pode garantir que este será atingido, isto é, se estes lugares serão de fato marcados para alguma marcação. Analisando os outros invariantes, pode-se verificar que a rede é salva, pois o número máximo de fichas para todos os lugares da rede é igual a um. \diamond

É necessário entretanto salientar que uma rede pode ser limitada para uma dada marcação, sem possuir uma cobertura de componentes conservativos. É o caso da rede de Petri da FIG. 2.12, com

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

cujos componentes conservativos são: $\mathbf{f}^1 = [1 \ 1 \ 0 \ 0 \ 0]$ e $\mathbf{f}^2 = [0 \ 0 \ 0 \ 1 \ 1]$. Não existe uma cobertura de lugar, pois qualquer que seja a combinação linear entre \mathbf{f}^1 e \mathbf{f}^2 , $f(p_3) = 0$. Portanto, a rede é não conservativa. Entretanto, esta rede é viva e binária para a marcação inicial $M = p_1 p_3$, o que é facilmente verificável calculando as marcações acessíveis. Já para a marcação inicial $M = p_1 p_3 p_4$, a rede é não limitada. Portanto, não é possível concluir sobre o limite de fichas em um lugar p para o qual $f(p) = 0$.

3.2.2 Componentes repetitivos, invariantes de transição

As propriedades de vivacidade e limitabilidade podem ser também definidas a partir dos componentes repetitivos estacionários:

- toda rede de Petri viva, limitada e reiniciável é repetitiva (a recíproca é falsa);
- se a rede é não repetitiva ($\exists t, s(t) = 0$) então ou ela é não viva ou é não limitada;
- uma rede repetitiva ($\forall t, s(t) > 0$) pode ser não viva.

Cobertura de transição \times limitada e viva:

Toda rede de Petri que é, ao mesmo tempo, viva e limitada para ao menos uma marcação inicial é tal que existe uma cobertura de componentes repetitivos estacionários $\bar{s} > 0$. Isto decorre dos seguintes fatos:

- limitada \Rightarrow existe um número finito de marcações sensibilizando uma dada transição;
- viva \Rightarrow existe seqüências de comprimento infinito;
- limitada e viva \Rightarrow existe uma seqüência repetitiva estacionária.

É fácil verificar que a recíproca do ponto dois é falsa. Basta em geral escolher uma marcação para a qual todos os lugares são vazios.

A rede de Petri da FIG 2.11.a é limitada, viva e reiniciável para a marcação $M = p_2p_4$, como se pode ver no grafo da FIG 2.11.b (item 2.3.3), utilizando a análise por enumeração de marcações. A rede deve ser, portanto, repetitiva como de fato o é, pois $\mathbf{s} = [1\ 1\ 1]^T$, conforme visto no item 2.4.2. Entretanto, a recíproca não é verdadeira, pois esta rede, repetitiva, não é reiniciável para a marcação $M = p_1p_4$, embora seja limitada e viva.

Considere a rede de Petri da FIG. 2.6, cuja matriz C é dada por:

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

O único valor de \mathbf{s} que satisfaz a equação 2.28 é $\mathbf{s} = [0\ 0\ 0]^T$, e a rede é não repetitiva. Ela é, portanto, não viva ou não limitada. Construindo-se o grafo de marcações acessíveis, pode-se verificar que a rede é não limitada.

Exemplo resolvido 2: (continuação)

O suporte dos componentes repetitivos é dado por $\mathbf{s} = \mathbf{s}^1 + \mathbf{s}^2 + \mathbf{s}^3 = [1\ 1\ 1\ 1\ 1\ 1\ 1]^T$. Como todos as transições pertencem ao suporte (de fato, $\forall t \in T, s(t) \neq 0$), a rede é repetitiva. Nada se pode afirmar sobre a vivacidade da rede. Se existisse um elemento $s(t) = 0$, então a rede seria não viva ou não limitada. \diamond

A rede da FIG. 2.4 também é repetitiva, com $\mathbf{s} = [1\ 1\ 1\ 1]^T$ mas não é viva para a marcação $M = p_1p_2$, como foi visto no item 2.1.8. É, pois, importante observar que a existência de um componente repetitivo não garante a vivacidade.

Por que um elemento $s(t) = 0$ implica que a rede é não viva ou não limitada? A idéia do componente repetitivo, e em particular, do invariante de transição correspondente, é que o disparo desta seqüência não modifica a marcação inicial, como indica a equação 2.28. Ora, se uma transição t não pertence a um componente repetitivo, então não pertence a nenhuma seqüência que leve novamente à marcação de partida. Existem dois motivos para tal:

- a rede é não limitada, é impossível voltar à marcação inicial porque existe um crescimento de fichas a cada tiro de t ;
- a rede é não viva, porque a transição t tem um bloqueio mortal, sendo assim impossível voltar à marcação inicial.

Por que um elemento $s(t) \neq 0$ não permite determinar se a rede é viva? Este fato é consequência da limitação da matriz C para representar a rede de Petri,² como foi extensivamente discutido no item 2.1.8.

Embora a análise estrutural da rede não permita, em muitos casos, uma resposta conclusiva sobre as propriedades do modelo, seu cálculo é simples, sem os custos de tempo e memória necessários ao cálculo do grafo de marcações acessíveis.

3.3 Análise através de redução

O método de análise por enumeração de marcações, apresentado no item 3.1, é extremamente pesado a implementar quando o tamanho do conjunto de marcações acessíveis torna-se grande, devido à explosão combinatória do número de estados. Uma solução consiste em aplicar regras de redução, de modo tal que a rede de Petri inicial e a rede reduzida sejam equivalentes do ponto de vista das propriedades. Deste modo, pode-se aplicar o método de análise por enumeração a uma rede de tamanho razoável, e deduzir as propriedades da rede original.

A noção de equivalência não implica necessariamente que as duas redes possuam as mesmas seqüências de tiro de transições ou os mesmos invariantes de lugar ou de transição. O objetivo é que as propriedades gerais, se verificadas na rede reduzida, possam ser consideradas como as da rede inicial. Entretanto, em certos casos particulares, a aplicação de algumas regras, necessárias para reduzir a rede, não conserva todas as boas propriedades.

3.3.1 Lugar substituível

Informalmente, um lugar substituível é um lugar que serve unicamente de etapa intermediária entre duas transições (ou dois conjuntos de transições). O disparo da transição (ou de uma das transições) de entrada é condição suficiente para poder disparar a transição (ou uma das transições) de saída. A simplificação consiste em fazer a fusão das transições de entrada e das transições de saída, duas a duas.

Se t_s é uma transição de saída do lugar p e t_e uma transição de entrada, é necessário que:

$$\forall q \in P, \text{Pre}(q, t_s) \leq \text{Post}(q, t_e)$$

²Ver capítulo 8, onde a lógica linear é utilizada para caracterizar a seqüência de tiro.

e o par $t_e t_s$ é substituído pela transição t_{es} tal que:

$$\begin{aligned} \forall q \in P, \text{Pre}(q, t_{es}) &= \text{Pre}(q, t_e) \quad \text{e} \\ \text{Post}(q, t_{es}) &= \text{Post}(q, t_e) - \text{Pre}(q, t_s) + \text{Post}(q, t_s) \end{aligned}$$

eliminando assim o lugar p .

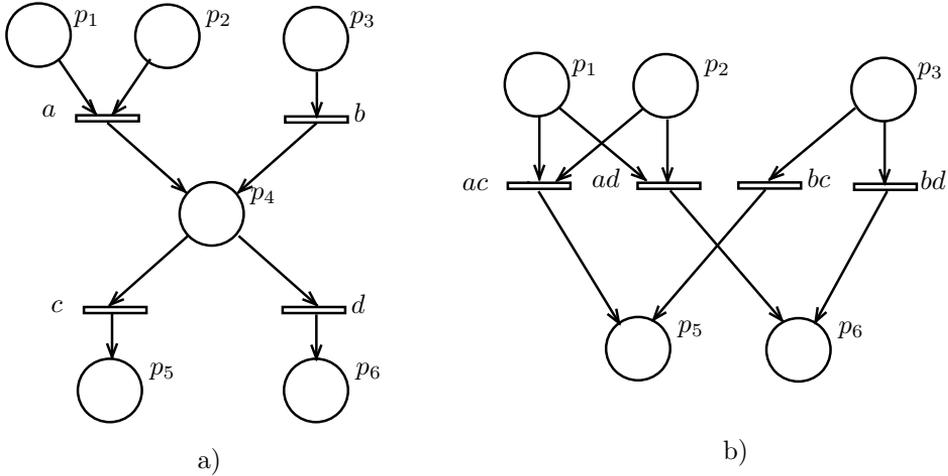


Figura 3.1: a) Lugar substituível p_4 ; b) Simplificação do lugar p_4

Considere, por exemplo, a rede de Petri da FIG. 3.1.a. O tiro de uma das duas transições a ou b é uma condição suficiente para poder disparar a transição c ou d (ambas estão em conflito). O lugar p_4 serve apenas como passagem e pode ser suprimido. Sua supressão produz a rede de Petri da FIG. 3.1.b.

O caso mais comum corresponde a transições t_s que possuem somente um lugar de entrada. Uma longa seqüência, como a da FIG. 3.2.b, será compactada em uma única transição t_{abc} com lugar de entrada p_4 e lugar de saída p_5 .

Dois pontos importantes devem ser salientados:

- se o lugar substituível contém fichas, é necessário disparar a transição de saída antes de suprimi-lo. Mas, neste caso, a rede reduzida e a rede original não são mais equivalentes do ponto de vista da propriedade de reiniciabilidade;
- se a transição de saída do lugar substituível não possui lugares de saída, a rede reduzida e a rede original não são mais equivalentes em relação à propriedade de k -limitada.

3.3.2 Lugar implícito

Um lugar implícito é redundante, do ponto de vista do tiro de sua transição de saída. Com efeito, sua marcação é uma combinação linear da marcação de um conjunto de lugares E e, em relação a suas transições de saída, tal lugar não introduz nenhuma condição suplementar de disparo.

O fato de a marcação de um lugar implícito ser uma combinação linear da marcação de um conjunto E liga esta noção àquela de componente conservativo, como pode ser visto no exemplo a seguir.

Considere a rede de Petri da FIG. 3.2.a. O objetivo é mostrar que o lugar p_1 é implícito em relação ao conjunto de lugares $\{p_2, p_3\}$. Para tal, deve-se procurar os componentes conservativos.

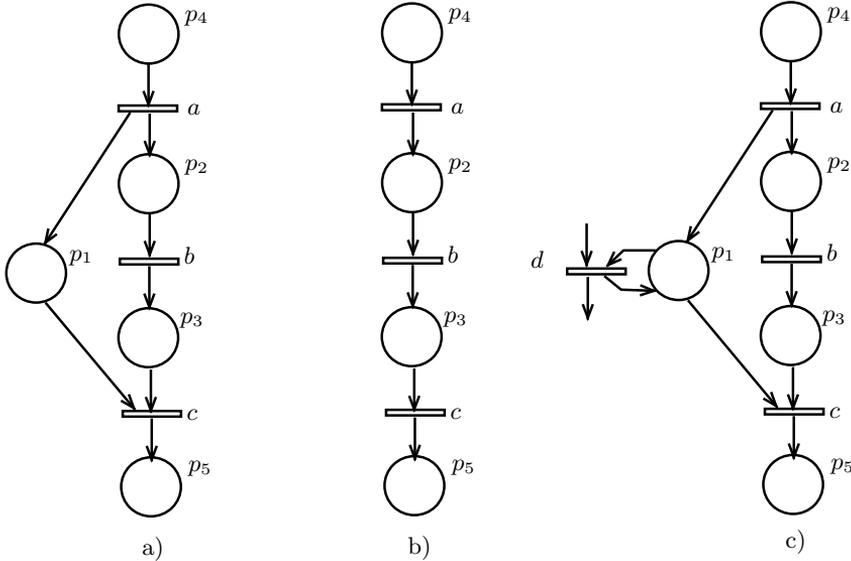


Figura 3.2: a) Lugar implícito p_1 ; b) p_1 simplificado; c) Contra-exemplo

O sistema de equações gerado por $f^T C = 0$ é:

$$\begin{aligned} f_1 + f_2 - f_4 &= 0 \\ -f_2 + f_3 &= 0 \\ -f_1 - f_3 + f_5 &= 0. \end{aligned}$$

A base B de vetores linearmente independentes é formada pelos vetores $\mathbf{f}^1 = [0 \ 1 \ 1 \ 1 \ 1]^T$ e $\mathbf{f}^2 = [1 \ 0 \ 0 \ 1 \ 1]^T$. O objetivo aqui não é o de simplesmente encontrar a base B , mas sim, um vetor \mathbf{f} , combinação destes vetores, tal que seus componentes f_4 e f_5 sejam nulos, já que os lugares de interesse são p_1, p_2 e p_3 . Tal vetor é obtido através da combinação linear $\mathbf{f} = \mathbf{f}^1 - \mathbf{f}^2 = [-1 \ 1 \ 1 \ 0 \ 0]^T$.

Aplicando-se \mathbf{f} na equação 2.27, com $M_0 = [0 \ 0 \ 0 \ 0 \ 0]^T$, obtém-se o invariante linear de lugar:

$$M(1) = M(2) + M(3).$$

Pode-se, portanto, calcular a qualquer instante a marcação do lugar p_1 em função da marcação dos lugares p_2 e p_3 .

A única transição de saída do lugar p_1 é a transição c . Ora, para que c esteja sensibilizada, é necessário que o lugar p_3 contenha ao menos uma ficha, o que implica, como visto acima, que o lugar p_1 contenha também uma ficha. Portanto, o lugar p_1

pode ser suprimido sem modificar as seqüências de disparo das transições. Tem-se assim a rede reduzida da FIG. 3.2.b. Naturalmente, do ponto de vista da representação dos estados do sistema, este lugar pode ter um papel importante. Por exemplo, o lugar p_1 , marcado, indica que o processo formado pelas atividades executadas entre o tiro de a e o de c ainda está em execução.

É necessário prestar atenção a dois fatores. Primeiro, a marcação inicial da rede modifica a equação que permite calcular a marcação do lugar implícito. Por exemplo, se o lugar p_2 (ou p_3) estiver marcado, tem-se o novo invariante:

$$M(1) = M(2) + M(3) - 1.$$

Conseqüentemente, o fato de o lugar p_2 conter uma ficha não é mais suficiente para afirmar que o lugar p_1 contém ao menos uma ficha.

O segundo problema a ser observado é que não é suficiente que a marcação de um lugar seja função da marcação de um subconjunto E para que ele seja implícito. Um contra-exemplo é dado pela FIG. 3.2.c. Como a transição d não é transição de saída do lugar p_2 nem do lugar p_3 , o lugar p_1 não pode ser suprimido. Com efeito, nesta rede, o tiro de d só pode ocorrer após o tiro de a (e antes do tiro de c). Se o lugar p_1 é retirado, esta condição para o disparo de d não mais existe, o que modifica o comportamento da rede.

Lugares idênticos e lugares implícitos degenerados:

Dois tipos de lugares implícitos, ilustrados pelas FIG. 3.3.a e 3.3.b, são facilmente reconhecíveis: os lugares idênticos e os lugares implícitos degenerados:

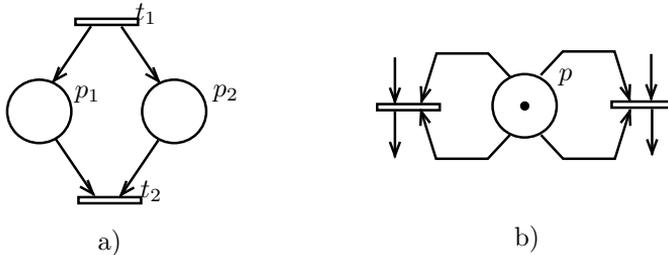


Figura 3.3: a) Lugares idênticos; b) Lugar implícito degenerado

- dois lugares p_1 e p_2 são idênticos se possuem as mesmas transições de entrada e de saída, com os mesmos pesos nos arcos. Diz-se, então, que p_1 é implícito em relação a p_2 ;
- o lugar p é implícito degenerado se for conectado à rede de Petri somente por malhas elementares: p é implícito em relação ao conjunto vazio, sua marcação sendo, então, constante.

3.3.3 Transição neutra ou identidade

Uma transição neutra é uma transição que, se retirada, não modifica o comportamento da rede. Sendo conectada à rede de Petri somente por malhas elementares, seu disparo não modifica a marcação.

Uma transição t é *neutra* (diz-se também que t é uma transição *identidade*) se e somente se:

$$Pre(\cdot, t) = Post(\cdot, t). \tag{3.2}$$

Para poder suprimir uma transição neutra t , conservando a propriedade de rede marcada viva, é necessário estar certo de que t é viva. Caso contrário, o que pode estar sendo suprimida da rede é uma parte morta desta. A rede reduzida pode, neste caso, ser viva, enquanto a rede original não o é.

Para suprimir a transição sem modificar a vivacidade da rede, é necessário que exista uma outra transição que, sendo viva na rede reduzida, garanta que t é viva na rede inicial. Existem dois casos evidentes, em que tal transição:

- produz uma marcação que sensibiliza t :

$$\exists t_1 \mid Post(\cdot, t_1) = Pre(\cdot, t);$$

- possui as mesmas condições de disparo que t :

$$\exists t_2 \mid Pre(\cdot, t_2) = Pre(\cdot, t)$$

que são ilustrados na FIG. 3.4.a. É suficiente que uma das condições seja verificada. Um contra-exemplo é dado pela rede de Petri da FIG. 3.4.b. De fato, a transição d , embora satisfaça a equação 3.2, não pode ser eliminada, já que nenhuma das condições acima é satisfeita. A rede reduzida, obtida com a supressão de d , é viva, enquanto a rede original é quase viva. Por outro lado, na rede de Petri da FIG. 2.10, a transição g é neutra e pode ser eliminada, pois a transição d possui as mesmas condições de disparo.

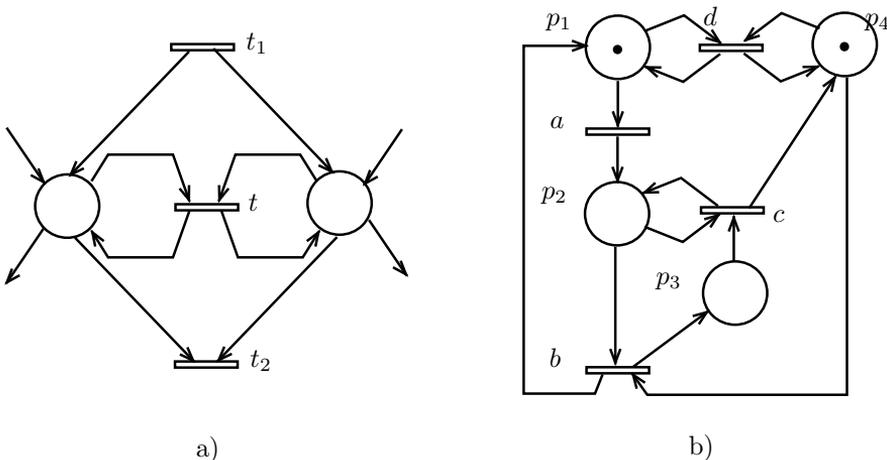


Figura 3.4: Transição neutra a) t ; b) d (não simplificável)

3.3.4 Transições idênticas

Duas transições são idênticas se e somente se as colunas correspondentes das matrizes Pre e $Post$ são idênticas:

$$\begin{aligned} Pre(\cdot, t_1) &= Pre(\cdot, t_2) \\ Post(\cdot, t_1) &= Post(\cdot, t_2). \end{aligned}$$

É claro que uma das duas pode ser eliminada sem mudar as propriedades da rede de Petri.

A rede de Petri da FIG. 3.5.a apresenta um exemplo de duas transições t_1 e t_2 idênticas; a rede simplificada em relação à transição t_2 é mostrada na FIG. 3.5.b.

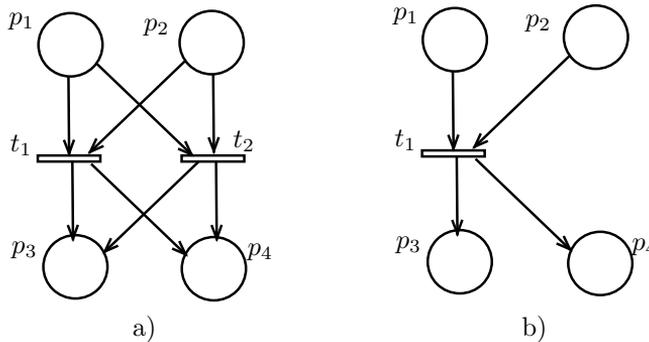


Figura 3.5: a) Transições t_1 e t_2 idênticas; b) Simplificação de t_2

3.4 Relação entre os diversos métodos de análise

Existem situações em que não é possível utilizar o método de análise baseado na enumeração das marcações acessíveis, devido: i) problema da explosão combinatória do número de estados; ii) sabe-se que existem lugares não limitados (caso dos *buffers*, por exemplo). Deste modo, não é possível mostrar que uma rede marcada não limitada é viva ou reiniciável. Pode também ocorrer que não seja necessário ou interessante construir o conjunto de marcações acessíveis. Resta, então, utilizar um dos dois outros métodos para obter informações sobre o sistema: o de análise estrutural e o de redução.

Através do método de análise por redução (seção 3.3), é possível reduzir o tamanho da rede. Tal método permite, em certos casos, aplicando a regra do lugar substituível, transformar uma rede não limitada em uma rede limitada, permitindo, posteriormente, a análise de vivacidade e reiniciabilidade pelo método de enumeração de estados. De forma geral, uma vez obtida a rede reduzida, se esta não possui as *boas* propriedades para uma dada marcação inicial, é às vezes difícil corrigi-la. De fato, a análise por redução permite evitar a enumeração das marcações, mas não produz, em caso de anomalia, a seqüência de disparo de transições que leva a uma marcação problema (bloqueio mortal ou marcação superior a uma marcação já obtida).

A análise estrutural, através do cálculo dos componentes conservativos e repetitivos estacionários fornece, então, as informações complementares que podem ser úteis. De

fato, viu-se no item 3.2 que todo lugar que pertence a um componente conservativo é limitado, embora o lugar possa ser limitado sem pertencer a um componente conservativo. Se existe uma transição que não pertence a nenhum componente repetitivo estacionário, então a rede pode ser não viva ou não limitada.

A utilização dos invariantes de lugar permite mostrar que certos lugares são limitados e calcular seu limite, sem enumerar o conjunto de marcações acessíveis. Por outro lado, os invariantes de transições fornecem apenas as condições necessárias, mas não suficientes.

Caracterização das marcações

Existem diferentes maneiras de se caracterizar um conjunto de marcações. A primeira maneira consiste em enumerar as marcações de fato *acessíveis* a partir de uma marcação inicial dada, usando a equação 2.8, construindo-se, assim, as seqüências permitidas de disparo de transições (são disparadas somente as transições de fato sensibilizadas).

A segunda maneira de caracterizar um conjunto de marcações é a de considerar a equação fundamental (2.15) e calcular todas as marcações positivas M' , obtidas substituindo-se todos os vetores positivos $\mathbf{s} > 0$ (correspondentes às diferentes seqüências de tiro). Obtém-se, assim, um conjunto que inclui necessariamente o primeiro, mas que pode ser maior que este. De fato, pode-se encontrar vetores $\mathbf{s} > 0$ que não correspondem à nenhuma seqüência s disparável, conforme discutido no item 2.1.8. Sabe-se, entretanto, que se a marcação M' resultante for negativa, tal seqüência não é disparável.

Enfim, o conjunto dos componentes conservativos $\{\mathbf{f}^i\}$ associado a uma marcação inicial M define igualmente um conjunto de marcações M' . Como os vetores \mathbf{f}^i formam um espaço vetorial, basta considerar uma base B de vetores linearmente independentes, e calcular as marcações utilizando a equação 2.27. Este conjunto inclui aquele calculado a partir da equação fundamental, mas pode ser estritamente maior.

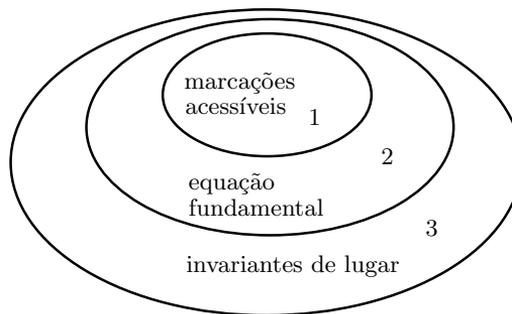


Figura 3.6: Caracterização das marcações

A FIG. 3.6 ilustra a relação entre os diferentes conjuntos de marcações, obtidos em cada um dos três casos. O conjunto “1” corresponde às marcações acessíveis $A(R, M)$, o conjunto “2”, às marcações positivas M' , soluções da equação fundamental para $\mathbf{s} > 0$, e o conjunto “3” é aquele definido pelo conjunto de invariantes lineares de lugar.

Estas diferentes maneiras de caracterizar as marcações podem ser ilustradas através

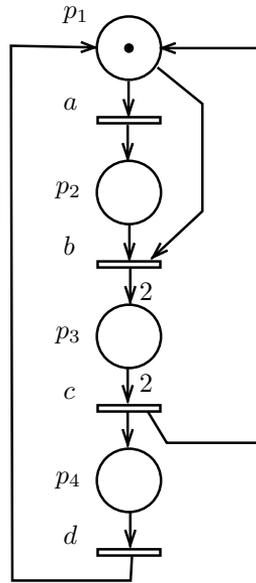


Figura 3.7: Rede com conjunto de marcações diferentes

de um exemplo, como o da rede da FIG. 3.7, cuja matriz C é dada por:

$$C = \begin{bmatrix} -1 & -1 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 2 & -2 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}.$$

O conjunto das marcações de fato acessíveis, obtidas através do cálculo do grafo de marcações acessíveis, é dado por $\{M_0, M_1\}$, com $M_0 = [1\ 0\ 0\ 0]^T$ e $M_1 = [0\ 1\ 0\ 0]^T$, e corresponde ao conjunto “1” da FIG. 3.6.

As marcações positivas³ obtidas substituindo-se na equação 2.15 vetores $\mathbf{s} > 0$ para a marcação inicial $M_0 = [1\ 0\ 0\ 0]^T$, são as seguintes:

- para $\mathbf{s}^1 = [1\ 0\ 0\ 0]^T$, obtém-se $M_1 = [0\ 1\ 0\ 0]^T$;
- para $\mathbf{s}^2 = [1\ 1\ 1\ 0]^T$, obtém-se $M_2 = [0\ 0\ 0\ 1]^T$;
- para $\mathbf{s}^3 = [\lambda\ \lambda\ \lambda\ \lambda]^T$, obtém-se $M_3 = [1\ 0\ 0\ 0]^T = M_0$;
- para $\mathbf{s}^4 = [(\lambda + 1)\ \lambda\ \lambda\ \lambda]^T$, obtém-se $M_4 = [0\ 1\ 0\ 0]^T = M_1$.

Estas marcações formam o conjunto $\{M_0, M_1, M_2\}$, que corresponde ao conjunto “2” na FIG. 3.6. Os vetores \mathbf{s}^i correspondem às seqüências $s_1 = a$, $s_2 = abc$, $s_3 = (abcd)^*$ e $s_4 = (abcd)^*a$, bem como a todas as seqüências obtidas permutando-se a ordem das transições em s_2 e s_3 .⁴ Os vetores $\mathbf{s} > 0$, correspondentes às seqüências b , c , d , ab $(abcd)^*ab$ (e suas permutações), produzem marcações negativas.

³Outros vetores \mathbf{s} , como $[0\ 1\ 0\ 0]^T$, $[0\ 0\ 1\ 0]^T$, $[0\ 0\ 0\ 1]^T$ e $[1\ 1\ 0\ 0]^T$, produzem marcações negativas quando substituídos na equação 2.15.

⁴O * indica que a seqüência pode se repetir zero, uma ou infinitas vezes.

Esta rede possui um único componente conservativo $\mathbf{f}^T = [1 \ 1 \ 1 \ 1]$ (ver método de cálculo no apêndice **A**). Para a marcação $M_0 = [1 \ 0 \ 0 \ 0]$, obtém-se, utilizando-se a equação 2.27, o invariante linear de lugar

$$M(p_1) + M(p_2) + M(p_3) + M(p_4) = 1.$$

As únicas marcações que satisfazem esse invariante são as seguintes: $M_0 = p_1$, $M_1 = p_2$, $M_2 = p_4$ e $M_3 = p_3$, pois a soma das marcações de todos os lugares deve ser igual a 1. O conjunto das marcações $\{M_0, M_1, M_2, M_3\}$ corresponde ao conjunto “3” da FIG. 3.6.

Esse exemplo mostra bem que o único modo de conhecer o verdadeiro conjunto de marcações acessíveis é o de calcular o grafo, aplicando-se sucessivamente a equação 2.8. Entretanto, em numerosos casos práticos, os conjuntos “1”, “2” e “3” são idênticos, podendo-se obter assim, informações sobre o comportamento dinâmico da rede de Petri de modo mais rápido e menos custoso.

3.5 Resultados particulares: subclasses

Nesta seção serão estudadas três subclasses de redes com ponderação unitária para as quais existem resultados particulares do ponto de vista da análise. Para as outras redes com ponderação unitária (além destas subclasses) é necessário utilizar os métodos vistos anteriormente.

Rede binária, Grafcet

As matrizes *Pre*, *Post* e *C* são definidas no conjunto $\{0,1\}$. Os lugares correspondem a variáveis booleanas.

Esta classe corresponde aos automatismos lógicos (e, portanto, também ao Grafcet). É também a única classe para a qual pode-se prever uma implementação.

A árvore de cobertura é mais simples, pois o algoritmo pára desde que um lugar contém mais de uma ficha.

Máquinas de estado

A subclasse de rede de Petri com ponderação unitária chamada máquina de estado (não confundir com o conceito de máquina de estados finitos) é definida da seguinte maneira:

- toda transição possui um e somente um arco de entrada;
- toda transição possui um e somente um arco de saída.

Uma rede de Petri do tipo máquina de estado forma um componente conservativo que passa por todos os lugares. Ela é, portanto, limitada qualquer que seja sua marcação inicial, e todas as propriedades são decidíveis. Esta subclasse modela o conflito, mas não modela nem a concorrência nem a sincronização.

Como as transições possuem somente um arco de entrada, uma rede do tipo máquina de estado é viva e reiniciável se ela contém ao menos uma ficha em cada componente conservativo e se cada um destes componentes é fortemente conexo (ver apêndice A).

Grafo de eventos

A subclasse de rede de Petri com ponderação unitária chamada grafo de eventos é definida do seguinte modo:

- todo lugar possui um e somente um arco de entrada;
- todo lugar possui um e somente um arco de saída.

Uma rede de Petri do tipo grafo de eventos forma um componente repetitivo estacionário.

Todo circuito do grafo *associado* obtido conservando como nó apenas as transições (isto é possível uma vez que os lugares possuem somente um arco de entrada e um de saída) é um componente conservativo.

Uma condição necessária e suficiente para que um grafo de eventos seja limitado (ele será, portanto, limitado para qualquer marcação inicial) é que todo lugar pertença a, pelo menos, um *circuito* do grafo associado. Um circuito é a representação gráfica de uma seqüência de transições $s = t_{j_1}t_{j_2} \dots t_{j_k}$ tal que $t_{j_1} = t_{j_k}$ e:

$$\forall(t_{j_r}, t_{j_{r+1}}) \exists p_{i_r} \mid p_{i_r} \in Post(t_{j_r}) p_{i_r} \in Pre(t_{j_{r+1}}).$$

Um circuito é um caminho de comprimento não nulo, partindo de t_{j_1} e voltando a t_{j_1} .

Uma condição necessária e suficiente para que um grafo de eventos seja vivo para uma marcação inicial dada é que a marcação seja tal que todo circuito elementar do grafo associado contenha ao menos uma ficha. Portanto, as propriedades de vivacidade e limitabilidade são decidíveis numa rede de Petri do tipo grafo de eventos.

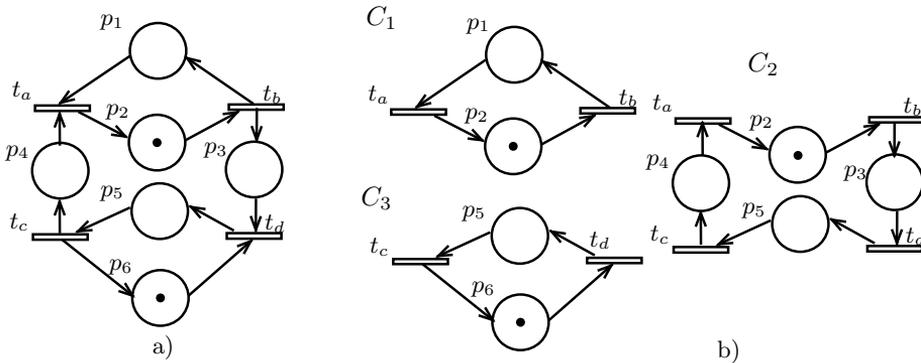


Figura 3.8: a) Grafo de eventos; b) Circuitos elementares

O número de fichas num circuito do grafo de eventos não muda com os disparos. Em consequência:

- a rede é viva se e somente se o número de fichas em cada circuito é maior ou igual a 1;
- a rede é salva se e somente se todo lugar que pertence a um circuito contém somente uma ficha.

A rede da FIG. 3.8.a é um grafo de eventos pois todo lugar possui somente um arco de entrada e um arco de saída. Os três circuitos elementares C_1 , C_2 e C_3 , mostrados na FIG. 3.8.b, são as sub-redes obtidas dos componentes conservativos $\mathbf{f}^1 = [1\ 1\ 0\ 0\ 0\ 0]$, $\mathbf{f}^2 = [0\ 1\ 1\ 1\ 1\ 0]$ e $\mathbf{f}^3 = [0\ 0\ 0\ 0\ 1\ 1]$. Em cada um dos circuitos existe uma ficha, portanto, a rede é viva. Todos os lugares pertencem a algum circuito, portanto, a rede é limitada.

3.6 Notas

É interessante utilizar o programa ARP de análise de redes de Petri disponível no LCMI para computadores compatíveis IBM-PC.⁵ Este programa realiza análise através do grafo de marcações, análise estrutural, simulação e avaliação de desempenho. Ao utilizar este programa, lembre-se que ele apenas indica se a rede é viva, limitada, reiniciável ou não (análise através do grafo) e os invariantes lineares de lugar e transição (análise estrutural). É fundamental que estes resultados sejam interpretados pelo usuário. O ARP auxilia durante a fase de especificação, pois, em geral, se a rede não possui as boas propriedades é porque a modelagem está errada e deve ser revista.

3.7 Exercícios

1. A partir dos resultados obtidos no exercício 1 do capítulo 2, analise o comportamento de cada rede. Interprete os resultados obtidos, principalmente os invariantes.
2. Analise as redes de Petri da FIG. 3.9 usando o método de redução. Detalhe cada passo, dando o nome da regra e justificando sua aplicação.
3. Considere a rede de Petri da FIG. 2.15.a com a marcação inicial $M_0 = P_1P_4$ e da FIG. 2.15.b com a marcação inicial $M_0 = P_1P_6$ e seus respectivos componentes conservativos e repetitivos, calculados no exercício 4 proposto no capítulo 2. a) O que se pode deduzir acerca das propriedades da rede? A rede é conservativa? É repetitiva? b) Para a marcação inicial M_0 e os invariantes já calculados, quais são as informações adicionais acerca das propriedades? É possível determinar se a rede é viva e limitada? c) Encontre a árvore de cobertura desta rede. Caso exista bloqueio mortal, dê a seqüência que o produz, a partir da marcação inicial; d) para a rede da FIG. 2.15.b, encontre a seqüência que permite disparar t_3 e compare com a questão a).
4. A transição g da FIG. 2.10 é neutra?

⁵O programa ARP (MAZIERO, 1990) encontra-se disponível, sob o nome `arp2-4.zip`, por FTP anônimo no endereço `ftp.lcmi.ufsc.br`, diretório `/pub/automacao/petri`. Cópias em disquete 3,5" podem ser obtidas através do endereço eletrônico `sysadmin@lcmi.ufsc.br`, colocando *Redes de Petri - ARP* no subject.

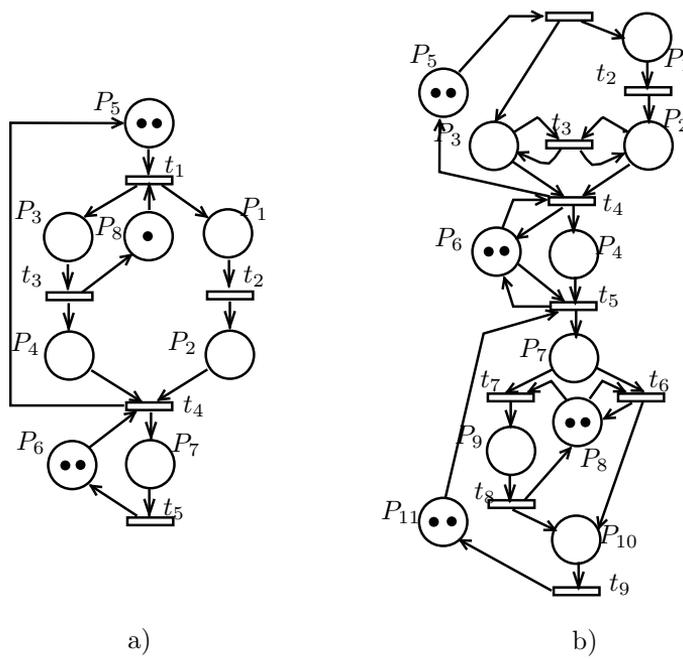


Figura 3.9: Análise por redução

Parte II

DADOS, TEMPO E AMBIENTE EXTERNO

Capítulo 4

REDES INTERPRETADAS

A primeira parte deste livro definiu o modelo básico da rede de Petri, também chamada de rede ordinária. Apenas a estrutura é representada, mas os conflitos, por exemplo, não são resolvidos. Na rede binária, a ficha indica apenas se a proposição associada ao lugar é verdadeira ou não. De forma geral, a ficha atua como um contador.

A rede de Petri pode ser utilizada para descrever *encadeamentos* de tratamentos atuando sobre estruturas de dados complexas, como, por exemplo, sinais que permitem a comunicação com o ambiente externo (do tipo sensores e acionadores) e restrições temporais explícitas. Esta estrutura de dados pode ser também uma hierarquia de classes de objetos (por exemplo, a classe *peça*, com os atributos: ordem de operação, máquinas onde será operada, etc.). Neste caso, tem-se uma rede de Petri de alto nível, objeto do capítulo 5.

No caso da rede de Petri interpretada, são associadas variáveis às transições da rede — representando condições e ações existentes no sistema. Tais variáveis podem indicar o estado de atuadores, sensores, etc., permitindo, assim, modelar a interação com o ambiente externo.

Numa rede de Petri ordinária, a transição é disparada desde que esteja sensibilizada. A sensibilização é condição necessária e suficiente para o disparo: se não houver conflito, a transição dispara.

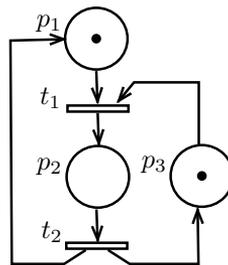


Figura 4.1: Disparo de transições

A rede da FIG. 4.1 é viva, limitada e reiniciável. A transição t_1 está sensibilizada para a marcação inicial $M = p_1p_3$ e dispara (o disparo é indivisível e de duração nula), levando à nova marcação $M' = p_2$. Por sua vez, t_2 é sensibilizada e dispara, voltando a rede à marcação inicial.

Suponha que o lugar p_1 represente *peças em espera*; p_2 , *peça sendo torneada*; p_3 , *máquina livre*; que a transição t_1 represente *início de operação* e t_2 , *final de operação*. Desde que os lugares p_1 e p_3 estejam marcados, os eventos *início* e *fim* da operação, correspondentes à seqüência $t_1 t_2$, irão ocorrer.

No sistema real, o tempo entre o início da operação, quando o lugar p_2 é marcado (disparo de t_1) e o final da operação, quando a transição t_2 vai ser disparada, corresponde à duração da operação e, portanto, é não nulo. Para representar corretamente o comportamento de um sistema onde o tempo transcorrido é importante, é necessário que o modelo considere a informação temporal.

Há diferentes maneiras de representar a ocorrência dos eventos associados ao início e ao fim da operação, seja para fins de simulação, emulação ou implementação.

- Tempo transcorrido: o tempo pode ser considerado implicitamente no modelo (rede de Petri temporal ou temporizada, capítulo 6) ou explicitamente (através de variáveis do tipo monoestável). Se $M(p_1) = 1$, a transição t_2 irá disparar τ segundos após o disparo de t_1 , onde τ é a duração da operação. É feito o cálculo do tempo transcorrido previsto, mas não há nenhuma informação proveniente do sistema.
- Informação do ambiente externo: a informação proveniente do sistema (real ou simulado) — um sensor associado à transição t_2 indicando o final da operação, por exemplo — pode ser representada por uma variável (correspondendo ao sensor) utilizada como condição suplementar de tiro. A condição $M(p_2) = 1$ é necessária, mas não suficiente: se a variável associada ao sensor não tiver valor verdadeiro, a transição não irá disparar.

Através do exemplo da FIG. 4.1 pode-se ver os diferentes níveis de representação de um sistema. A rede ordinária, constituída somente pela estrutura (lugares e transições) e marcação, modela apenas a relação de causalidade entre as ações: o evento *início da operação* acontece antes do evento *final da operação*.

Para que a modelagem seja mais fiel, é necessário *interpretar* a rede, o que implica adicionar condições suplementares de disparo. Para representar o tempo na rede da FIG. 4.1, pode-se, por exemplo, associar uma duração d à atividade *peça sendo trabalhada* (lugar p_2). A transição só pode disparar, se, além do lugar p_2 marcado, o tempo d for transcorrido. Outra forma ainda de representar a interação com o sistema é associar uma variável booleana F à transição t_2 que indica o estado de um sensor *final de operação*. Neste caso, t_2 só pode disparar se, além do lugar p_2 marcado, o valor de F for verdadeiro. Uma rede interpretada é uma rede não autônoma que introduz o tempo, a interação com o ambiente ou dados contidos na ficha. A rede da FIG. 4.1 é uma rede autônoma, pois o disparo das transições depende apenas da estrutura da rede e de sua marcação.

4.1 O que é a interpretação?

Semântica dos lugares, transições e fichas

Interpretar uma rede de Petri implica antes de tudo dar um sentido concreto a um modelo matemático, associando, aos lugares, transições e fichas, os elementos existentes no sistema.

Como visto no capítulo 1, os lugares podem ser interpretados como as atividades de um sistema a eventos discretos, como estoques de um sistema de manufatura, condições lógicas de um sistema lógico seqüencial, tarefas sendo executadas num sistema informático, etc.

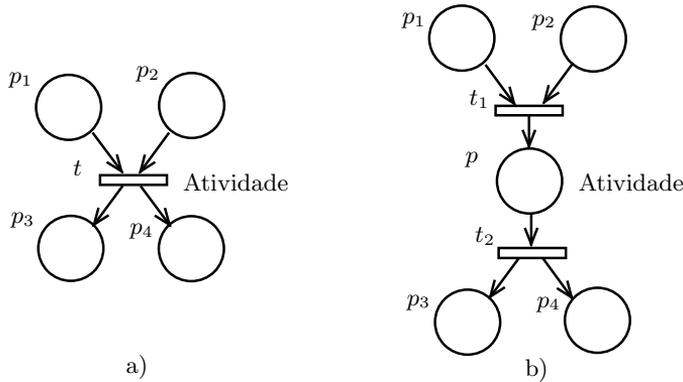


Figura 4.2: Associação de uma atividade a uma transição

Pode-se fazer corresponder a uma transição, um evento considerado de duração nula na escala de tempo considerada como indicado na FIG. 4.2.a. Pode-se também lhe associar uma atividade ou tarefa, desde que esta seja *indivisível*, ou ainda, *não interrompível*. Tal transição é, neste caso, chamada de *abreviação* de uma seqüência elementar, formada de uma transição de início, um lugar descrevendo a atividade ou a tarefa, e uma transição de fim, como indicado na FIG. 4.2.b. As duas redes da figura são equivalentes, pois o lugar p é um lugar *substituível* (ver regras de redução no capítulo 3) e as propriedades da rede não são modificadas.

As fichas podem ser interpretadas como objetos físicos, informações, estruturas de dados ou recursos. Tais entidades são submetidas a eventos (disparo de transições) que lhes fazem mudar de estado (passar de um lugar a outro).

Interação com os dados e o ambiente

Os tratamentos realizados por um sistema devem ser especificados durante a modelagem, complementando, assim, as informações contidas na estrutura da rede de Petri. É por este motivo que, além de um simples nome, deve-se associar a especificação dos tratamentos às transições ou aos lugares. No primeiro caso, supõe-se que o tratamento será executado de maneira não interrompível quando do disparo da transição. No segundo, considera-se que se trata de uma atividade que se desenrola durante todo o tempo em que o lugar está marcado.

A estrutura da rede de Petri descreve os encadeamentos dos tratamentos a serem efetuados, definindo a *estrutura de controle* do sistema modelado.

Às vezes certos tratamentos não são sistematicamente executados imediatamente após o término dos tratamentos precedentes. Suas execuções podem ser submetidas a condições que tratam com certos dados (instruções condicionais do tipo “if”, por exemplo). Tal fenômeno será descrito por um lugar com várias transições de saída. Do ponto de vista da teoria de rede de Petri, isto caracteriza uma situação de *conflito*. O disparo

de uma das transições em conflito ocorre quando a sua *condição suplementar de disparo*, cujo valor depende dos dados, for verdadeira.

No caso de sistemas que interagem com seu ambiente, tem-se, igualmente, condições suplementares de disparo associadas às transições. As ações são associadas às transições (ou eventualmente aos lugares). Tais ações (que são tratamentos particulares) e condições fazem intervir dados ou eventos exteriores como sensores, acionadores, recepções ou emissões de mensagens. Muitas vezes o relógio tempo real desempenha um papel importante no ambiente e o tempo intervém explicitamente sob a forma de uma *temporização* através de variáveis booleanas cujo valor depende do tempo (monoestáveis, *watchdog*, etc.).

Controle, dados e ambiente externo

Finalmente, pode-se dizer que modelar um sistema *aberto* (sistema comunicando-se em permanência com o ambiente externo) utilizando a rede de Petri consiste em estruturá-lo em duas partes: controle e dados.

A parte *controle* descreve *todos* os encadeamentos potenciais de eventos e de atividades. Esta *parte controle* (ou simplesmente *controle*) é descrita pela rede de Petri não interpretada.

A parte *dados* (chamada também *parte operativa*) descreve, ao mesmo tempo, as estruturas de dados internas ao sistema e os cálculos que são feitos sobre estes dados, sem especificar em quais instantes eles serão realizados. Além dos dados internos, nestes cálculos intervém o tempo e as informações provenientes dos mundo exterior. Dois tipos de cálculo podem aparecer:

- as condições;
- os tratamentos ou ações.

A interpretação da rede de Petri especifica as ligações entre a parte *controle* de um lado e a parte *dados* e ambiente externo de outro lado. A FIG. 4.3 descreve esta estruturação. A interpretação exprime-se da seguinte forma:

- associação de condições às transições;
- associação de ações às transições;
- associação de tratamento aos lugares.

Tanto as condições C_i como as ações A_i podem ser expressas como disjunção ou conjunção de variáveis booleanas, e são associadas à transição t_i . Graficamente, são representadas na forma (C_i, A_i) ao lado de t_i .

A rede de Petri interpretada oferece uma boa estruturação da parte *controle*. Entretanto, não existe um método para estruturar os dados. O capítulo 5 apresenta modelos de rede de Petri que propõem a estruturação dos dados: rede predicado-transição, rede colorida e rede a objetos. Tais redes permitem uma descrição compacta e estruturada, a partir da noção de individualização da ficha.

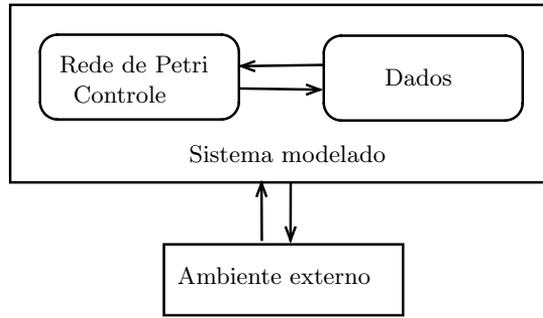


Figura 4.3: Interação da rede de Petri com o ambiente externo

Tempo considerado explicitamente

A consideração explícita do tempo se faz através de variáveis booleanas aparecendo tanto nas condições quanto nas ações associadas às transições e cujos valores dependem do tempo.

Seja $mon(\theta)$ uma variável booleana; se seu valor inicial é “0”, ela automaticamente assume o valor “1” após uma duração θ , semelhante a um monoestável. Suponha que quando do disparo da transição t_1 na FIG. 4.1, tal variável assumo o valor “0”. Se uma transição t_2 , imediatamente sensibilizada pelo disparo de t_1 possui a condição

$$mon(\theta) = 1,$$

então o intervalo de tempo entre o disparo de t_1 e o de t_2 será exatamente de θ . Este intervalo corresponde à duração da atividade associada ao lugar p_2 .

O tempo pode também ser considerado implicitamente no próprio modelo, de modo rigoroso e sistemático, associando uma duração às transições (ou lugares). Alguns destes modelos de rede de Petri — temporizado, temporal e estocástico — serão apresentados no capítulo 6.

4.2 Análise

A rede de Petri descreve apenas a *parte controle* do sistema. A marcação da rede de Petri fornece, portanto, apenas o *estado do controle*. O *estado do sistema*, isto é, o estado da rede de Petri interpretada é descrito pela marcação associada ao estado dos dados.

$$\text{estado rede interpretada} = \text{marcação} + \text{estado dados}$$

$$\text{estado dados} = \text{estado variáveis internas} + \text{tempo transcorrido}$$

Por outro lado, é necessário observar que as evoluções das marcações da rede de Petri interpretada são restrições de evoluções das marcações da rede antes de sua interpretação. Isto significa que o conjunto de marcações acessíveis após a interpretação está incluído

no conjunto de marcações acessíveis antes da interpretação. De fato, para que uma transição possa ser disparada na rede de Petri interpretada, deve estar sensibilizada na rede não interpretada. Mas a interpretação pode ser tal que uma transição, sensibilizada pela marcação, pode não ser disparada. Por exemplo, o tempo associado à atividade ainda não foi transcorrido, ou o valor verdade da variável auxiliar não foi verificado. Portanto, uma transição que é viva na rede subjacente (rede sem a interpretação) pode ser não viva na rede interpretada. Como consequência, algumas marcações nunca serão alcançadas na rede interpretada, e o conjunto de marcações acessíveis pode ser menor que o da rede subjacente, como ilustrado na FIG. 4.4.



Figura 4.4: Marcações acessíveis

Tem-se, portanto, os seguintes resultados:

Rede não interpretada		Rede interpretada
k-limitada para M	\implies	k-limitada para $M + \text{dados}$
f invariante de lugar	\implies	f invariante de lugar

Por outro lado:

Rede não interpretada		Rede interpretada
viva	\neq	viva
reiniciável	\neq	reiniciável

Pode-se, então, questionar qual o interesse de analisar uma rede de Petri não interpretada já que quase sempre será necessário interpretá-la e que suas propriedades podem então ser modificadas.

A primeira razão é que as propriedades das redes de Petri interpretadas são não decidíveis. De fato, a complexidade da verificação destas propriedades é a mesma da prova formal de programa. Mas certas propriedades (em particular os invariantes de lugar) são conservadas, o que facilita raciocínios posteriores relativos à rede interpretada.

A segunda razão é que a concepção e a validação de um sistema complexo deve decompor-se em etapas. O estudo da rede de Petri não interpretada, mais fácil, permite desde cedo colocar em evidência as incoerências e evitar o alto custo provocado pelos erros na fase de especificação. Não teria sentido construir uma especificação correta do ponto de vista da rede interpretada que estivesse fundada sobre uma rede adjacente, tendo uma estrutura incoerente, como bloqueio mortal, por exemplo.

4.3 Validação por simulação

A rede de Petri autônoma, estudada na Parte I, pode ser analisada, permitindo a obtenção de resultados sobre o comportamento dinâmico do sistema modelado.

A rede de Petri interpretada, como foi visto no item 4.2, pode ter apenas a rede subjacente analisada.

Para obter maiores informações sobre o comportamento dinâmico da rede interpretada, é necessário simulá-la. Embora a simulação seja um método de validação não exaustivo (pois é impossível prever *todas* as configurações possíveis das variáveis), é possível, ao menos, testar o comportamento do sistema modelado para as configurações previstas.

A maneira mais simples de simular um sistema de controle é o usuário fornecer, passo a passo, os valores das variáveis associados às transições. Entretanto, é mais interessante simular o conjunto formado pelo sistema de controle e a planta. Se a simulação é realizada utilizando o tempo de resposta real de cada sistema, têm-se uma *emulação*. Nesta fase de validação, o sistema é previamente simulado e a seguir emulado, pois, eventualmente, o comportamento pode mudar ao utilizar os tempos reais de resposta. O sistema pode depois ser conectado diretamente à planta (ver capítulo 7).

No caso de sistemas a eventos discretos, em particular sistemas complexos como os sistemas de manufatura, é fundamental que o projetista disponha de um ambiente integrado de ferramentas de especificação, validação (por análise e simulação) e também de implementação. A rede de Petri, sendo uma técnica de descrição formal, permite esta integração.

4.4 Modelagem com rede de Petri interpretada

Os passos para modelar um sistema utilizando a rede de Petri interpretada são os seguintes:

1. Encontrar a estrutura da rede de Petri (ou rede subjacente), representando as atividades concorrentes, paralelas, em seqüência, etc. Descrever ao mesmo tempo, quais os eventos associados às transições e como representá-los, indicando, em particular os eventos que exigem uma interação com o ambiente externo. Estas informações não fazem parte da estrutura, mas farão parte dos dados da rede de Petri interpretada.
2. Analisar a rede subjacente, verificando se possui as boas propriedades, como ausência de bloqueio mortal. Se necessário, corrigir tanto a parte estrutural quanto a de dados.
3. Simular a rede interpretada (controle e dados) para tentar obter o comportamento do sistema modelado. Lembre-se que uma simulação apresenta resultados apenas sobre o conjunto de valores de entrada escolhido (marcação inicial e valores iniciais das variáveis) e que, na quase totalidade das vezes, não é possível simular todas as configurações de entrada possíveis, seja por falta de tempo ou por esquecimento. É impossível, pois, garantir que o sistema se comportará sempre da mesma forma.

As delimitações entre o que deve ser modelado na parte controle e na parte dados são imprecisas e pressupõem escolhas delicadas a fazer durante a concepção.

Passa-se progressivamente de um *modelo* a uma especificação de concepção (descrição detalhada) afinando a descrição através do enriquecimento da interpretação.

4.5 Exemplo

Para ilustrar a modelagem de um sistema utilizando redes de Petri interpretadas, será apresentado o exemplo tratado em SOARES (1994), da automação de uma estação de coleta de petróleo similar às existentes nos campos terrestres explorados pela Petrobrás no Rio Grande do Norte.

A função de uma estação coletora é receber o petróleo bruto proveniente de poços localizados em sua circunvizinhança, testar a vazão de cada um desses poços e transferir o volume produzido, através de bombas de transferências, por oleodutos que conduzem a estações de armazenamento de maior capacidade.

4.5.1 Descrição do processo

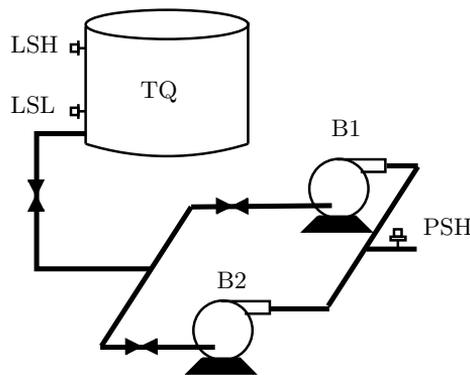


Figura 4.5: Estação de coleta de petróleo

O sistema representado na FIG. 4.5 possui um tanque de armazenamento (TQ) que recebe o petróleo continuamente através de uma tubulação ligada aos poços produtores. Quando o óleo atinge um determinado nível no tanque (nível alto), aciona-se uma moto-bomba (B_1) que faz a transferência do óleo para uma estação de coleta central. Uma segunda moto-bomba *stand by* (B_2) deve ser acionada, caso a primeira não seja ligada dentro de um intervalo de 2 segundos. Quando o nível do óleo no tanque cair até um determinado ponto (nível baixo), deve-se desligar a moto-bomba para evitar que a mesma corra o risco de sugar em vazio (o que poderia danificá-la). Também deve ocorrer o desligamento da bomba quando houver um aumento excessivo de pressão na descarga da mesma (o que caracteriza um bloqueio na tubulação a jusante). A sinalização dos níveis alto e baixo é feita através de chaves de nível localizadas nessas posições: LSH (nível alto) e LSL (nível baixo); e a sinalização da pressão alta na descarga da bomba é feita através do pressostato PSH.

Para automatizar este sistema é necessário que haja:¹

- intertravamento das chaves de nível e do pressostato com as bombas;
- medição do volume produzido através da contagem dos pulsos elétricos enviados por um medidor de vazão (não representado na FIG. 4.5);
- aquisição e envio dos dados para uma estação central, onde é feito o controle supervisorio de várias estações.

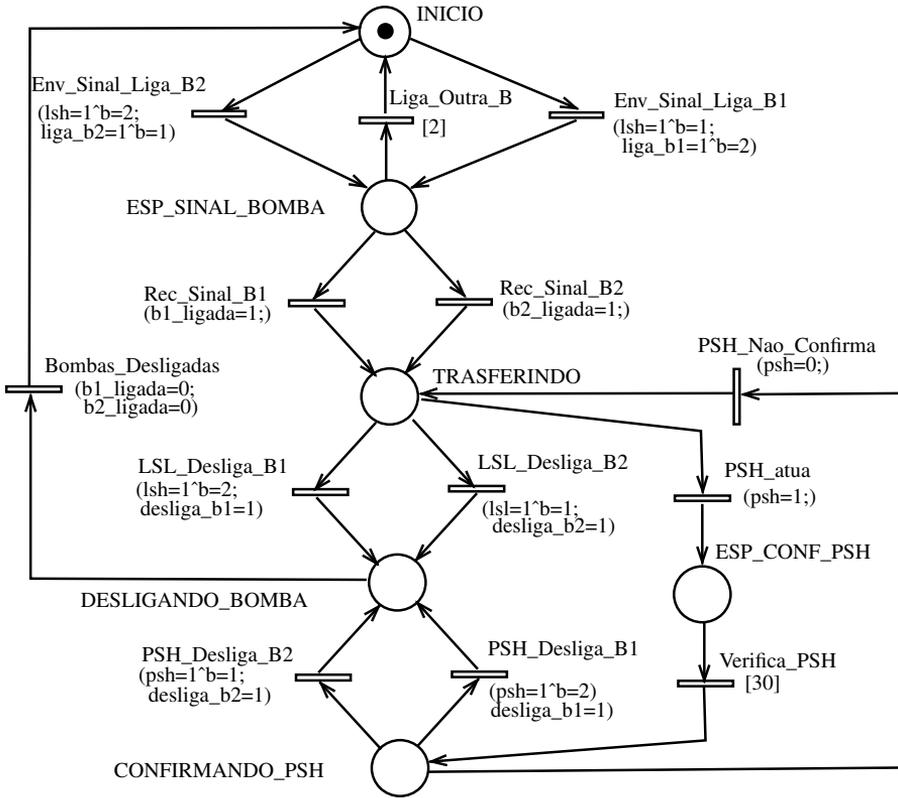


Figura 4.6: Modelo do controle

Lógica de intertravamento

As bombas B_1 e B_2 são acionadas alternadamente para cada transferência, ou seja, se na primeira transferência do dia for acionada B_1 , na segunda será acionada B_2 e assim por diante. Para que o CLP acione uma bomba é necessário que o tanque tenha atingido o nível alto (LSH=1). Na finalização da transferência, o sinal de entrada que provoca

¹Na prática estas tarefas são implementadas através de controladores lógicos programáveis (CLPs) localizados em cada estação coletora. Apesar da linguagem de programação dos CLPs ser familiar e de fácil entendimento para o pessoal técnico envolvido, não permite a validação do programa de aplicação, o que normalmente acaba sendo feito através de testes com o sistema real.

o desligamento da bomba acionada é o de nível baixo (LSL=1). Ao acionar uma determinada bomba, o programa aguarda, durante 2 segundos, pelo sinal de confirmação de bomba ligada (proveniente de um contato auxiliar do contactor da mesma); se, transcorridos os 2 segundos, o programa não receber essa confirmação, a bomba é desligada, sendo enviado um outro sinal para o acionamento da segunda bomba. O sinal de pressão alta (PSH) enviado pelo pressostato na descarga das bombas, durante a transferência, provoca o desligamento da bomba que estiver em operação. Esse sinal deverá ser ignorado se ocorrer em um tempo inferior a 30 segundos após o acionamento da bomba (para evitar que os transientes de pressão alta, que ocorrem normalmente nessa situação, provoquem uma interrupção desnecessária da transferência).

As FIG. 4.6 e FIG. 4.7 representam, respectivamente, o modelo do sistema de controle e da planta. Os termos [2] e [30] associados às transições Liga_Outra_B e Verifica_PSH são variáveis do tipo monoestável para representar o tempo transcorrido. Após este tempo, se a ficha ainda estiver no lugar de entrada, a transição é disparada. As outras transições t_i possuem uma interpretação associada do tipo (C_i, A_i) . A interpretação associada às transições $t_1 \dots t_8$ estão indicadas na tabela abaixo:

t_i	Nome	(C_i, A_i)
t_1	B2_Resp_Desligada	(; B2_ligada=0^lsl=0)
t_2	B2_Rec_Sinal_Liga	(liga_B2=1;liga_B2=0)
t_3	B2_Resp_Ligada	(; B2_ligada=1^lsh=0)
t_4	B2_Rec_Sinal_Desl	(desl_B2=1;desl_B2=0)
t_5	B1_Resp_Ligado	(; B1_ligada=1^lsh=0)
t_6	B1_Rec_Sinal_Liga	(liga_B1=1;liga_B1=0)
t_7	B1_Resp_Desligada	(; B1_ligada=0^lsl=0)
t_8	B1_Rec_Sinal_Desl	(desl_B1=1; desl_B1=0)

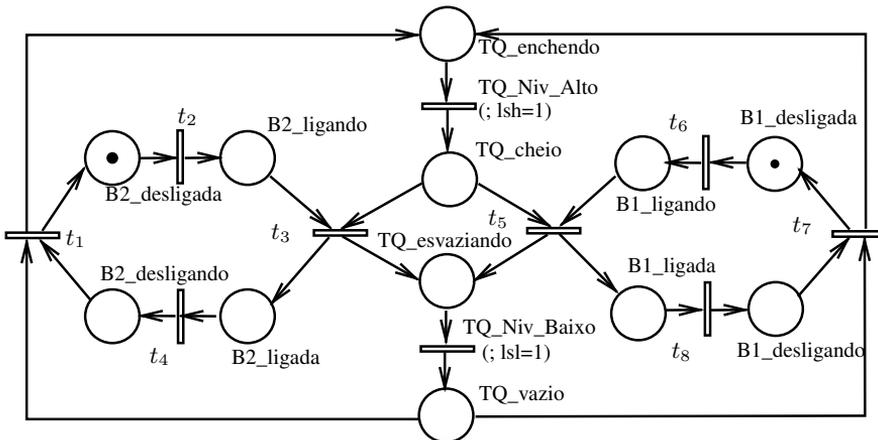


Figura 4.7: Modelo da planta

Para realizar a validação deste sistema, o primeiro passo é analisar a rede subjacente. Neste exemplo, a rede do sistema de comando é viva, reiniciável e binária, assim como a rede do sistema comandado.

4.5.2 Modelo do sistema

Para simular o sistema, é necessário estabelecer alguns cenários, como por exemplo:

- Cenário 1: fazer $PSH = 1$, simulando pressão alta na descarga das bombas;
- Cenário 2: fazer $PSH = 1$, e após o disparo da transição que indica o recebimento do sinal do pressostato (PSH_Atua), fazer $PSH = 0$.

4.6 Notas

O ambiente integrado de ferramentas AIF,² proposto em OLIVEIRA (1994), integra ferramentas de especificação, validação (por análise e simulação) e implementação com o objetivo de auxiliar o projeto de sistemas a eventos discretos. Estão integradas neste ambiente as ferramentas: ARP (Analisador de redes de Petri), SPP (Sistema de produção proposicional), que traduz uma rede de Petri na forma de regras num programa em código C, e a ferramenta de simulação propriamente dita, especificada e implementada de modo a simular o controle e a planta interagindo. Todas estas ferramentas foram desenvolvidas no LCMI.

4.7 Exercícios

1. Modele, usando o modelo de rede de Petri interpretada, os exercícios propostos no capítulo 1.
2. Proponha outros cenários de simulação para o exemplo apresentado no item 4.5.

²Este ambiente pode ser obtido através do endereço sysadmin@lcmi.ufsc.br, colocando *Redes de Petri - AIF* no *subject*.

Capítulo 5

REDES DE ALTO NÍVEL

Os modelos descritos a seguir propõem uma estruturação da parte de dados do sistema, diferenciando-se, por exemplo, os dados globais daqueles locais. Por dados locais, entendem-se dados que só intervêm em certas condições ou ações, e que só são acessíveis em certos instantes. As redes coloridas, predicado-transição e a objetos associam, de forma estruturada, uma parte dos dados às fichas.

Neste capítulo serão apresentados os elementos principais que estão na origem destes modelos: o conceito de dobramento de lugares e transições, e o da ficha como portadora de informações. Serão dadas definições simplificadas com o objetivo de diferenciar os modelos entre si, e de diferenciá-los da rede ordinária. As definições formais podem ser encontradas nas referências apresentadas no final do capítulo.

A delimitação entre controle e dados num sistema não é algo inerente ao sistema descrito. Ela resulta de uma escolha do projetista. Num extremo, praticamente todo o controle pode ser integrado às condições de execução dos tratamentos, e assim desaparecer da estrutura da rede de Petri. No outro extremo, o controle pode ser desenvolvido de forma exagerada, resultando em redes de Petri de grande tamanho, dificilmente manipuláveis.

Sistemas complexos, como os sistemas de manufatura, caracterizam-se pela capacidade de fabricar vários tipos de produtos simultaneamente. Existe, portanto, um componente de dados importante (tipo de peça a ser fabricada, estado atual da peça, etc.), em que a dinâmica tem um papel fundamental (seqüência de operações possíveis sobre um processo de fabricação, por exemplo). A repartição dos recursos (máquinas, sistema de transporte) e a complexidade dos mecanismos de alocação destes põem em evidência o paralelismo (tanto a cooperação como a concorrência) existente no sistema. Devido a estas características, a rede de Petri é considerada como uma ferramenta indicada para a especificação do controle de um sistema de manufatura. Além disso, seu formalismo permite a validação pela análise, como visto no capítulo 3, e pela simulação.

Entretanto, em tais sistemas complexos, alguns problemas aparecem. Esta complexidade significa, às vezes, a composição de vários processos semelhantes. Neste caso, quando se utiliza a rede de Petri ordinária (com a marcação dos lugares dada por fichas indiferenciadas, e com os lugares comportando-se como contadores), tem-se duas escolhas:

- modelar o comportamento geral sem precisar a identidade de cada processo, mas somente seu número;

- modelar, individualmente, cada um dos processos que constituem o sistema e modelar a interação existente entre eles, o que consiste, muitas vezes, em desdobrar o modelo que representa o comportamento geral.

No primeiro caso, obtém-se uma descrição compacta, mas não detalhada o suficiente: há uma falta de informação. No segundo caso, o modelo obtido pode ser pouco prático de se trabalhar, seja pelo tamanho da rede, seja pelo número de interações existentes. É necessário, portanto, estruturar parte dos dados do sistema fora da estrutura da rede.

Vários modelos foram propostos, chamados neste livro de Redes de Petri de alto nível (RPAN). Esses modelos, embora mais ou menos equivalentes, possuem, cada um, as suas próprias características. Na próxima seção serão apresentadas as características gerais das RPAN, e a seguir, as características principais de cada modelo. Finalmente, os modelos serão comparados ressaltando-se os pontos comuns.

5.1 Características gerais

Para melhor discutir as características de uma RPAN, será apresentado um exemplo de coordenação da passagem de peças em diferentes máquinas num sistema de manufatura.

Exemplo resolvido 3:

O sistema possui três máquinas M_1 , M_2 , M_3 . As peças que entram na oficina a um dado momento podem pertencer a diferentes processos de fabricação, cada processo sendo caracterizado por uma seqüência de operações. Se a máquina à qual a peça foi destinada está livre, a operação se inicia. Após seu término, a máquina é liberada e a peça está pronta para a próxima operação. Para simplificar a modelagem, considera-se que a entrada/saída das peças da oficina é feita por uma máquina de carga/descarga.

A FIG. 5.1.a mostra a rede de Petri (ordinária) no caso de uma só máquina e uma só peça. A FIG. 5.1.b representa o caso mais complexo de três peças, pertencentes a dois processos de fabricação diferentes, Pr_1 e Pr_2 . Cada processo é composto de três operações, executadas por diferentes máquinas (M_1 , M_2 e M_3). A ordem de passagem das peças nas máquinas é $M_1-M_2-M_3$ para o processo Pr_1 e $M_2-M_1-M_3$ para o processo Pr_2 . ◇

Este é um caso típico de sistema em que os processos têm um comportamento semelhante. Para simplificar a modelagem, pode-se representar somente o comportamento geral do sistema, o número de processos sendo dado pelo número de fichas no lugar *máquinas livres* (FIG. 5.1.c). Esta passagem de um modelo detalhado, que mostra todos os processos, a um modelo que representa apenas o comportamento geral é chamado de *dobrar* a rede. De fato, o conjunto de processos com uma mesma estrutura (ou estruturas próximas umas das outras) é *dobrado* num único componente conservativo.

Exemplo resolvido 4:

Um exemplo bem conhecido de um sistema com vários processos de igual comportamento é o dos leitores e escritores, modelado pela rede da FIG. 5.2. Os lugares L_i indicam que o processo de leitura está inativo. Quando há um pedido de leitura (tiro da transição t_{pi}), o processo está em espera (lugares L_{e_i}). Se o recurso estiver livre neste

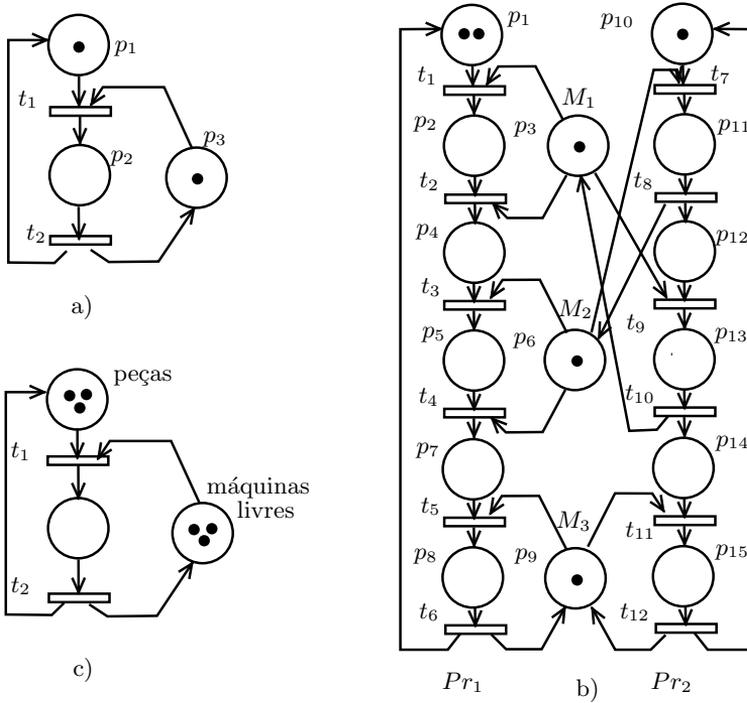


Figura 5.1: Comportamento: a) detalhado; b) um só processo; c) geral

momento, um dos processos poderá ler (tiro de uma das transições de início de leitura, t_{i_i}), passando para o estado ativo (lugar L_{a_i}). Uma vez a leitura acabada, a transição t_{f_i} correspondente dispara, e o sistema retorna à marcação original. Este sistema pode ler até três processos ao mesmo tempo, mas possui exclusão mútua entre os processos de leitura e escrita. As transições e lugares do processo de escrita são análogos aos dos processos de leitura; os arcos (R, t_{i_e}) e (R, t_{f_e}) têm peso 3 para impedir que um leitor possa ler uma zona de memória ao mesmo tempo em que está sendo feita a escrita.

Como os processos de leitura são similares, eles podem ser representados por um único processo, com três fichas (uma para cada processo), como representado na FIG. 5.3. Os lugares L_i são dobrados sobre o lugar L , os lugares L_{e_i} são dobrados sobre o lugar L_e , e os lugares L_{a_i} são dobrados sobre o lugar L_a . As transições t_{p_i} são dobradas sobre t_{p_l} , as transições t_{i_i} sobre t_{i_l} e as transições t_{f_i} sobre t_{f_l} . A rede torna-se mais simples, pois os três processos *leitores* são descritos a partir de uma mesma estrutura (encadeamento de lugares e transições). Entretanto, quando acontece um evento implicando um leitor, a transição t_{p_l} dispara, mas não se sabe de qual processo-leitor se trata. \diamond

Desta forma, um modelo mais compacto e abstrato é obtido, mas perdem-se informações. O simples *dobramento* de uma rede pode não ser suficiente e, inclusive, representar o sistema de modo errôneo. É o que acontece também no caso da FIG. 5.1.c: sabe-se o número de máquinas livres, mas não se sabe quais estão livres. Como existem três fichas no lugar *peças à espera*, e três fichas em *máquinas livres*, a transição t_1 (*iniciar operação*) pode ser disparada sem que se possa, entretanto, verificar qual máquina é afetada à peça.

Portanto, o fato de dobrar a rede é insuficiente. Além de tornar o modelo mais com-

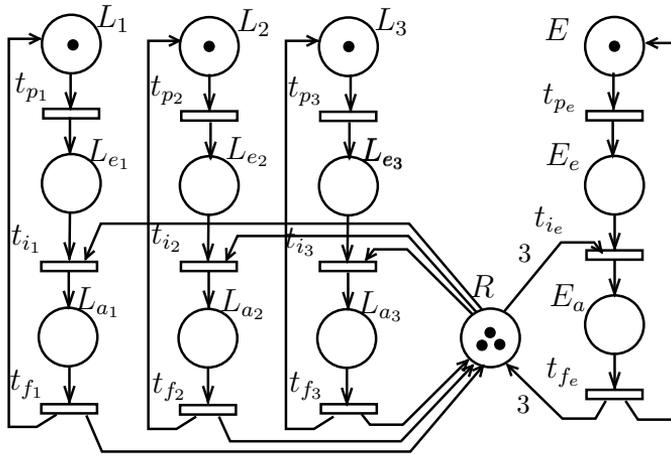


Figura 5.2: Leitores e escritores: rede completa

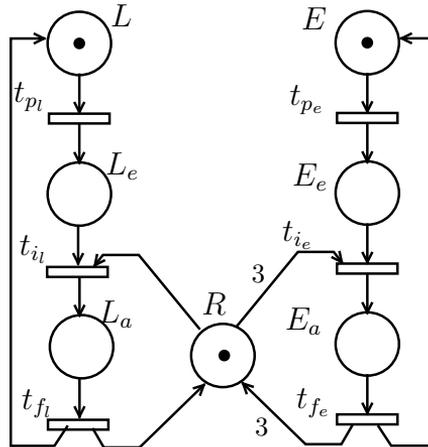


Figura 5.3: Leitores e escritores: rede dobrada

pacto, é necessário, sobretudo, poder diferenciar cada ficha, fazendo com que ela possa transmitir uma informação. A idéia subjacente das redes de Petri com individualização das fichas é, precisamente, a de poder realizar este dobramento sem perda de informação e sem perder a visualização gráfica da estrutura dos processos. Deste modo, com as noções de dobramento de uma rede e de individualização das fichas, pode-se aumentar o poder de expressão da rede de Petri.

A estas duas características, junta-se mais uma: a utilização de variáveis ou funções no disparo das transições. Agora que cada ficha é um *indivíduo*, é necessário escolher quais delas vão sensibilizar a transição e fazer evoluir o sistema.

A seguir são apresentados os diferentes modelos de rede de Petri. Todos se baseiam na idéia de dobramento da rede com individualização das fichas. Segundo a forma que assume esta individualização, tem-se o modelo de rede de Petri colorida, rede predicado-transição e rede a objetos.

5.2 Os diferentes modelos de RPAN

5.2.1 Rede de Petri colorida

Definição formal (simplificada)

Uma rede de Petri colorida associada a uma marcação inicial é uma sêxtupla dada por:

$$N_c = \langle P, T, C_{or}, C_{sc}, W, M_0 \rangle$$

onde:

- P é um conjunto finito de lugares;
- T é um conjunto finito de transições;
- C_{or} é um conjunto finito de cores;
- C_{sc} é a função *sub-conjunto de cores* que a cada lugar e a cada transição associa um sub-conjunto de C_{or} (as cores possíveis para este lugar ou esta transição):
 $C_{sc} : P \cup T \longrightarrow \mathcal{P}(C_{or})$;
- W é a função de incidência (equivalente a $C = Post - Pre$); cada elemento $W(p, t)$ de W é também uma função
 $W(p, t) : C_{sc}(t) \times C_{sc}(p) \longrightarrow \mathbb{N}$;
- M_0 é a marcação inicial que associa, para cada lugar e para cada cor possível neste lugar, um número de fichas
 $M_0(p) : C_{sc}(p) \longrightarrow \mathbb{N}$.

Associação de cores às fichas

Com o objetivo de diferenciar as fichas, são associadas cores (números inteiros ou conjunto de etiquetas) a estas. Como consequência, a cada *lugar* se associa o conjunto de cores das fichas que podem pertencer a este lugar. A cada *transição* se associa um conjunto de cores que corresponde às diferentes maneiras de disparar uma transição. Nos casos mais simples, quando todos os processos possuem rigorosamente a mesma estrutura e são independentes uns dos outros, as cores das transições são diretamente associadas aos processos, e o conjunto de cores dos lugares e das transições são idênticos.

Por exemplo, no caso dos leitores e escritores da FIG. 5.3, os processos correspondem às identidades dos leitores. Portanto, aos lugares L , L_e e L_a e às transições t_p , t_i e t_f se associa o conjunto de cores $\{L_1, L_2, L_3\}$. Isto não acontece quando os lugares são comuns a vários processos. É necessário, então, introduzir *cores compostas*.

O conjunto de cores de uma transição indica as diferentes maneiras de como ela pode disparar. Cada cor de transição corresponde, neste caso, a uma das transições da rede ordinária equivalente. Dito de outra forma, cada transição da rede ordinária que é dobrada numa transição da rede colorida equivalente, vai corresponder a uma cor do conjunto de cores desta última.

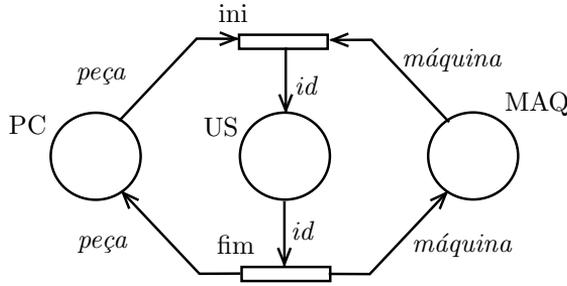


Figura 5.4: Rede de Petri colorida

Exemplo resolvido 3 (rede colorida): (continuação)

Considere novamente o exemplo do sistema de manufatura da FIG. 5.1, constituído por três máquinas e três peças. A FIG. 5.4 representa o modelo de rede de Petri colorida deste sistema.

O conjunto de cores do lugar *PC* é o conjunto $\{pc_1, pc_2, pc_3\}$ dos nomes das peças e o conjunto de cores do lugar *MAQ* é o conjunto dos nomes das máquinas $\{maq_1, maq_2, maq_3\}$. Já as cores do lugar *US* formam o conjunto $\{pc_1.maq_1, pc_2.maq_1, pc_3.maq_1, pc_1.maq_2, \dots, pc_3.maq_3\}$, que corresponde às diferentes operações de usinagem e que serão denotadas de forma simplificada por $u_{11}, u_{21}, u_{31}, u_{12}, \dots, u_{33}$. As cores das transições *ini* e *fim* são as mesmas de *US*.

O modelo completo da rede colorida da FIG. 5.4 é dado por: Voltando ao exemplo da FIG. 5.4, tem-se:

- $C_{or} = \{pc_1, pc_2, pc_3, maq_1, maq_2, maq_3, u_{11}, u_{12}, u_{13}, \dots, u_{32}, u_{33}\};$
- $C_{sc}(PC) = \{pc_1, pc_2, pc_3\},$
 $C_{sc}(MAQ) = \{maq_1, maq_2, maq_3\},$
 $C_{sc}(US) = C_{sc}(ini) = C_{sc}(fim) = \{u_{11}, u_{12}, \dots, u_{32}, u_{33}\};$

- $W = \begin{matrix} PC \\ US \\ MAQ \end{matrix} \begin{matrix} ini & fim \\ \left[\begin{array}{cc} -peça & peça \\ id & -id \\ -máquina & máquina \end{array} \right] \end{matrix}$

cujas funções são dadas por: $\begin{cases} id(u_{ij}) = u_{ij} \\ máquina(u_{ij}) = maq_j; \\ peça(u_{ij}) = pc_i; \end{cases}$

- $M_0 = \begin{bmatrix} pc_1 & pc_2 & pc_3 & maq_1 & maq_2 & maq_3 & u_{11} & u_{12} & u_{13} & u_{21} & \dots & u_{33} \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix},$

ou, sob a forma de um vetor de somas formais:

$$M_0 = \begin{bmatrix} pc_1 + pc_2 + pc_3 \\ 0 \\ maq_1 + maq_2 + maq_3 \end{bmatrix}. \quad \diamond$$

Associação de funções aos arcos

Numa rede colorida, as etiquetas dos arcos não são mais números inteiros como no caso da rede ordinária, mas funções que representam matrizes de inteiros. Para cada cor de uma transição (maneira de dispará-la) é necessário descrever quais cores de fichas serão retiradas dos lugares de entrada (pré-condição) e quais cores de fichas serão colocadas nos lugares de saída (pós-condição).

A função associada ao arco tem como domínio o produto cartesiano entre o conjunto de cores da transição e o conjunto de cores do lugar (de entrada ou saída da transição, segundo o arco). O conjunto imagem da função Pre é o conjunto \mathbb{N} dos inteiros positivos que descreve quantas fichas de cada cor devem ser retiradas dos lugares de entrada quando a transição for disparada:

$$Pre(p, t) : C_{sc}(t) \times C_{sc}(p) \longrightarrow \mathbb{N},$$

e o conjunto imagem da função $Post$ é o conjunto \mathbb{N} que descreve quantas fichas de cada cor devem ser colocadas no lugar de saída quando a transição for disparada (ver definição simplificada mais adiante):

$$Post(p, t) : C_{sc}(t) \times C_{sc}(p) \longrightarrow \mathbb{N}.$$

A partir das matrizes Pre e $Post$, define-se a matriz $W(p, t) = Post(p, t) - Pre(p, t)$.

As matrizes Pre , $Post$ e W são matrizes de matrizes, pois seus elementos (as etiquetas dos arcos), descritos por funções, são de fato matrizes cujas linhas são as maneiras de disparar as transições, e as colunas, as cores possíveis nos lugares.

Exemplo resolvido 3 (rede colorida): (continuação)

A rede de Petri colorida representada na FIG. 5.4 possui três funções: *peça*, *máquina* e *id*. Esta última não modifica as cores das fichas.

A matriz Pre é dada por:

$$Pre = \begin{array}{cc} & \begin{array}{cc} ini & fim \end{array} \\ \begin{array}{c} PC \\ US \\ MAQ \end{array} & \begin{bmatrix} Pre(PC, ini) & Pre(PC, fim) \\ Pre(US, ini) & Pre(US, fim) \\ Pre(MAQ, ini) & Pre(MAQ, fim) \end{bmatrix} \end{array} \quad \begin{array}{c} PC \\ US \\ MAQ \end{array} \quad \diamond$$

A função $Pre(PC, ini)$, que etiqueta o arco que liga o lugar de entrada PC à transição ini é dada pela matriz $Pre(PC, ini) : US \times PC \rightarrow \mathbb{N}$, onde $US = C_{sc}(ini)$ e $PC = C_{sc}(PC)$. A função $Post(MAQ, ini)$ que etiqueta o arco que liga a transição ini ao lugar de saída MAQ é dada pela matriz $Post(MAQ, ini) : US \times MAQ \rightarrow \mathbb{N}$, onde $MAQ = C_{sc}(MAQ)$. As funções $Pre(PC, ini)$ e $Post(MAQ, ini)$ são, respectivamente, descritas pelas matrizes:

$$\begin{array}{c} pc_1 \\ pc_2 \\ pc_3 \end{array} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{21} & u_{22} & \cdots & u_{33} \\ 1 & 1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 1 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}, \quad \begin{array}{c} maq_1 \\ maq_2 \\ maq_3 \end{array} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{21} & u_{22} & \cdots & u_{33} \\ 1 & 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

A rede colorida se baseia principalmente na notação matricial para tornar a rede de Petri mais compacta. Se por um lado tem-se um ganho do ponto de vista visual, pois a representação é bem compacta, por outro lado, continua-se a tratar com a mesma complexidade do sistema *desdobrado* formado pela rede de Petri ordinária.

Uma maneira abreviada de representar a função é indicar diretamente a cor que deve ser retirada (ou colocada) do (no) lugar. Neste caso, o domínio da função é o conjunto de cores da transição, e a imagem, o conjunto de cores do lugar:

$$Pre(p, t) : C_{sc}(t) \rightarrow C_{sc}(p)$$

$$Post(p, t) : C_{sc}(t) \rightarrow C_{sc}(p),$$

e o nome da função é colocado como etiqueta do arco no grafo.

Exemplo resolvido 3 (rede colorida): (continuação)

No exemplo da FIG. 5.4, a função $Pre(PC, ini)$ é dada por *peça*: $US \rightarrow PC$, que etiqueta o arco (PC, ini) . Esta função, aplicada às cores u_{ij} do conjunto de cores US da transição ini , indica que uma ficha de cor pc_i deve ser retirada do lugar PC , $peça(U_{ij}) = pc_i$. Esta mesma função, etiquetando o arco (fim, PC) , aplicada ao conjunto de cores US da transição fim , indica que uma ficha de cor pc_i deve ser colocada no lugar PC . \diamond

Análise

A rede de Petri colorida foi introduzida devido à dificuldade de encontrar os invariantes no modelo predicado-transição (seção 5.2.2). Há duas maneiras de encarar a análise:

- trabalhar com a rede de Petri ordinária obtida *desdobrando-se* a rede colorida, ou
- utilizar diretamente a matriz de funções lineares entre os conjuntos de cores.

No primeiro caso, se o número de cores é finito, é possível enumerar as marcações acessíveis. A análise por enumeração de marcações pode ser feita exatamente do mesmo modo como para uma rede de Petri ordinária. Mas a explosão combinatória é muito grande, pois é necessário diferenciar duas fichas de cores diferentes.

No segundo caso, existe algoritmo de procura da árvore de marcações, baseado na noção de marcação equivalente. Quando as cores e as funções apresentam uma certa simetria, é possível definir classes de equivalência entre as marcações acessíveis. É necessário estruturar os conjuntos de cores com uma simetria do tipo rotação. Duas marcações M_1 e M_2 são equivalentes se e somente se existe uma rotação ϕ tal que $M_1 = \phi(M_2)$. A árvore obtida deste modo é de dimensão menor que a obtida a partir do modelo *desdobrado*.

A explosão combinatória é então reduzida e a análise torna-se mais próxima da análise da rede *subjacente* (rede ordinária obtida substituindo-se as funções associadas aos arcos por pesos).

No que concerne ao cálculo dos componentes conservativos e repetitivos estacionários, pode-se realizá-lo de modo semelhante ao utilizado pelas redes de Petri ordinárias.

Os invariantes de lugares são obtidos por uma seqüência de transformações que visam a reduzir a matriz de incidência sem mudar o conjunto de invariantes. As regras são

baseadas no método de eliminação de Gauss. Entretanto, ao invés de realizar produto de inteiros, deve-se fazer composição de funções (funções de funções). Cada vez que se consegue fazer aparecer uma linha (ou uma coluna) de zeros na matriz W , obtém-se uma componente. No entanto, isto não é sempre possível. Por outro lado, o conjunto de componentes não forma necessariamente um espaço vetorial.

Embora não exista um algoritmo geral para resolver a matriz, muitas vezes a matriz reduzida, obtida após aplicação das regras, é tal que os invariantes podem ser encontrados por inspeção.

Retornando ao exemplo anterior, pode-se encontrar dois componentes conservativos. De fato, multiplicando a segunda linha por *peça* e observando que $peça(id) = peça$, obtém-se:

$$W_1 = \begin{bmatrix} 0 & 0 \\ peça & -peça \\ -máquina & máquina \end{bmatrix} \begin{bmatrix} PC + peça(US) \\ peça(US) \\ MAQ \end{bmatrix}.$$

Encontra-se assim o invariante de lugar:

$$M(PC) + peça(M(US)) = M_0(PC) + peça(M_0(US)).$$

Multiplicando a segunda linha da matriz W por *máquina* e somando com a terceira, obtém-se o invariante de lugar:

$$máquina(M(US)) + M(MAQ) = máquina(M_0(US)) + M_0(MAQ).$$

A interpretação destes invariantes é simples com a utilização de somas formais. Suponha que a usinagem u_{12} esteja sendo realizada, a peça pc_2 e a máquina maq_1 estão em espera. Tem-se $M(PC) = pc_2$, $M(US) = u_{12}$ e $M(MAQ) = maq_1$. O primeiro invariante descreve a conservação das peças:

$$pc_2 + peça(u_{12}) = pc_2 + pc_1,$$

e o segundo, descreve a conservação das máquinas

$$máquina(u_{12}) + maq_1 = maq_2 + maq_1.$$

5.2.2 Rede de Petri predicado-transição

Noção de variáveis, semi-unificação

No caso da rede de Petri colorida, o poder de descrição é aumentado substituindo-se os inteiros da matriz de incidência de uma rede de Petri ordinária por funções. No caso das redes predicado-transição, as transições de uma rede de Petri ordinária são consideradas como regras num sistema de lógica proposicional (sem variáveis), e o poder de descrição é aumentado substituindo-se por regras da lógica de primeira ordem (regras com variáveis). Assim, não somente a representação é mais concisa, mas o modelo permite estudar melhor as propriedades estruturais e comportamentais do sistema.

Uma regra (transição) descreve, então, uma família de eventos e não mais somente um evento. A família é definida pelo conjunto de substituições possíveis de variáveis por valores. Ao invés de regras do tipo

- se uma peça pc_1 e máquina M_2 , fazer usinagem; u_{12}
- se uma peça pc_2 e máquina M_2 , fazer usinagem; u_{22}
- se uma peça pc_i e máquina M_j , fazer usinagem; u_{ij}
- ...

tem-se regras do tipo

se uma peça $\langle x \rangle$ e uma máquina $\langle y \rangle$, fazer usinagem $\langle u \rangle$.

Estas variáveis $\langle x \rangle$, $\langle y \rangle$ e $\langle u \rangle$ assumirão, respectivamente, valores no conjunto de constantes $\{pc_i\}$ descrevendo as peças em espera, o conjunto $\{M_j\}$ das máquinas livres, e o conjunto $\{u_{ij}\}$ das operações a serem realizadas. Estas constantes têm um papel análogo ao das cores na rede colorida, mas variáveis são associadas aos arcos da rede de Petri predicado-transição, no lugar das funções.

No modelo predicado-transição, a transição *dobrada* é vista como um esquema (*scheme*) que possui, como instâncias, todas as transições (de uma rede ordinária) que representam o mesmo evento.

Exemplo resolvido 3 (rede predicado-transição): (continuação)

No exemplo apresentado no início do capítulo (FIG. 5.1), as transições t_1 , t_3 e t_5 (Pr_1) podem ser consideradas como *instâncias* de um só *esquema-transição* (verificar se a máquina é livre e se a peça está esperando; se for o caso, iniciar a operação), como se pode ver na FIG. 5.5.

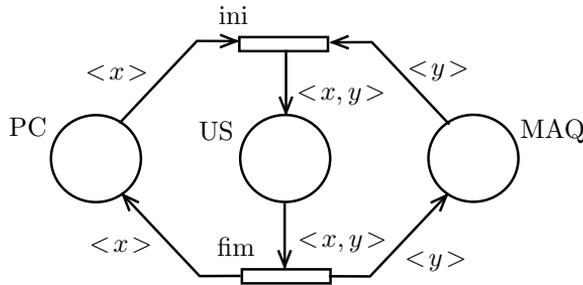


Figura 5.5: Rede de Petri predicado-transição

Os lugares p_3 , p_6 e p_9 , referentes ao Pr_1 (máquinas M_1 , M_2 e M_3 livres), também podem ser vistos como um *esquema lugar* (lugar *dobrado* MAQ da FIG. 5.5), representado pelo predicado *Máquina M_i livre*. O mesmo se passa com os lugares p_2 , p_5 e p_8 — representados pelo predicado *Máquina M_i realiza operação sobre peça j* (lugar dobrado US) — e com os lugares p_1 e p_{10} — representados pelo predicado *peça j à espera* (lugar dobrado PC). ◇

Como pode ser visto neste exemplo, os lugares da rede representam as relações dinâmicas do sistema, indicadas por predicados. Por este motivo, o modelo é chamado predicado-transição.

Não é necessário enumerar *a priori* todas as fichas autorizadas nos lugares PC, US e MAQ. Pode-se ignorar as diferentes maneiras de disparar as transições, pois basta apenas saber que todos os elementos de uma mesma família serão tratados da mesma maneira. Aplicam-se as regras sobre uma marcação atual. Ao buscar todas as substituições

possíveis para as variáveis, sabe-se, a um dado momento, de quais modos uma transição pode ser disparada. Este mecanismo de substituição de variáveis por constantes (fichas) é chamado de *unificação*. Mais precisamente, de semi-unificação, pois as variáveis são sempre substituídas diretamente por constantes e nunca por outras variáveis.

Quando o mecanismo de unificação produz mais de uma solução, diz-se (no vocabulário de Inteligência Artificial) que existe um conflito. O espaço de conflitos compreende todas as transições (regras) disparáveis para uma marcação dada (base de fatos), e para cada transição o conjunto de substituições possíveis. Esta noção de conflito não é a mesma que aquela, mais restritiva, que concerne às transições em uma rede de Petri (ver capítulo 2, seção 2.1.7).

Para realizar a escolha das fichas e das transições a disparar, são associadas condições suplementares de disparo às transições (por exemplo, para especificar que certas máquinas só podem tratar certas peças). Tais condições podem ser escritas como fórmulas lógicas utilizando as variáveis dos arcos de entrada, bem como operadores e predicados a elas associados. Entretanto, estas condições só se aplicam às constantes associadas às fichas susceptíveis de serem deslocadas, quando substituídas às variáveis no disparo da transição. Assim, uma instância do *esquema-transição* é gerado por uma substituição consistente das variáveis pelos indivíduos que constituem a marcação.

Da mesma forma, as ações associadas às transições são escritas utilizando-se as variáveis associadas aos arcos de entrada e saída (os arcos são etiquetados por somas formais de variáveis).

Vale salientar que na rede de Petri colorida a escolha específica de uma cor do conjunto de fichas é descrito por funções associadas aos arcos. Na rede predicado-transição, estas escolhas são expressas por condições associadas às transições que as variáveis devem obedecer (exceto no caso simples de igualdade de constantes que pode ser exprimido utilizando a mesma variável associada a mais de um arco de entrada de uma transição). As redes de Petri predicado-transição são, portanto, deste ponto de vista, mais próximas das redes de Petri interpretadas.

Noção de n-upla de constantes e de variáveis

No disparo da transição *ini* (FIG. 5.5), é retirada uma constante representando uma peça do lugar *PC*, uma outra representando uma máquina no lugar *MAQ* e ambas são colocadas no lugar *US* codificando-se, assim, o fato de que uma peça e uma máquina estão temporariamente em relação. No momento do disparo da transição *fim*, significando o fim desta relação, é feito o contrário. Ao invés de codificar esta relação temporária de duas informações, elas são justapostas numa lista, como num *record* em linguagem Pascal. A diferença é que se trata de algo temporário, criado e destruído durante o disparo das transições. Deste modo, pode-se exprimir uma *relação dinâmica* entre as constantes. Estas listas de constantes chamam-se de *n-uplas*.

Após o disparo de *ini*, a identidade da máquina utilizada não desaparece, mas se mantém como elemento da n-upla $\langle pc_i, maq_j \rangle$ no lugar *US*, representando a relação dinâmica *peça pc_i sofrendo operação na máquina maq_j* . No caso da rede de Petri colorida (FIG. 5.4), o lugar *US* contém a ficha de cor u_{ij} .

A marcação deixa de ser uma distribuição de constantes nos lugares para se tornar uma distribuição de n-uplas de constantes. A n-upla $\langle pc_1, maq_2 \rangle$ difere da soma $\langle pc_1 \rangle + \langle maq_2 \rangle$, pois no segundo caso $\langle pc_1 \rangle$ e $\langle maq_2 \rangle$ serão utilizadas no disparo

da transição de maneira independente, enquanto que a n -upla $\langle pc_1, maq_2 \rangle$ é substituída às variáveis, em bloco. Portanto, as expressões associadas aos arcos são também n -uplas de variáveis formais.

Definição da rede predicado-transição (simplificada)

Uma rede de Petri predicado-transição marcada é uma quintupla:

$$N_{pt} = \langle P, T, V, Pre, Post, C_{onst}, A_{tc}, A_{ta}, M_0 \rangle$$

onde P é o conjunto de lugares, T é o conjunto de transições, e

- V é um conjunto de variáveis formais que serão substituídas por constantes de C_{onst} quando do disparo da transição;
- Pre é a aplicação *lugar precedente* que associa, a cada arco, uma soma formal de n -uplas de elementos de V ;
- $Post$ é a aplicação *lugar seguinte* que associa, a cada arco, uma soma formal de n -uplas de elementos de V ;
- C_{onst} é um conjunto de constantes (conjunto \mathcal{N} , por exemplo);
- $A_{tc} : T \longrightarrow L_c(C_{onst}, V)$ é uma aplicação que associa, a cada transição, uma condição sob a forma de um predicado utilizando tanto as constantes como as variáveis formais;
- $A_{ta} : T \longrightarrow L_a(C_{onst}, V)$ é uma aplicação que associa, a cada transição, uma ação sob a forma de uma seqüência de afetações de valores às variáveis formais;
- M_0 é a marcação inicial, associando a cada lugar p de P uma soma formal de n -uplas de constantes C_{onst} .

As relações dinâmicas aparecem como lugares da rede. As funções e relações da parte estática, ou mais precisamente, seus nomes, vão aparecer nas anotações. As anotações das condições e ações são expressões que *resumem* as sentenças afirmativas da linguagem natural. Os operadores (símbolos da função) e predicados (símbolos da relação) formam o vocabulário da linguagem. A linguagem usada é a da lógica de predicados de primeira ordem mais a classe de expressões algébricas simples para denotar combinações lineares.

A definição das linguagens L_c e L_a , que expressam as condições e ações, determina a classe particular da rede de Petri predicado-transição.

A aplicação Pre (e $Post$) é tal que seu módulo (a soma de todos os elementos sem distingui-los e desenvolvendo as n -uplas) é igual ao peso do arco da rede subjacente: $\|Pre(p, t)\| = Pre(p, t)$. Isto significa que não há nenhuma contradição entre uma rede de Petri predicado-transição N_{pt} e sua rede de Petri *subjacente* R do ponto de vista do número de fichas deslocadas.

Exemplo resolvido 3 (rede predicado-transição): (continuação)

O sistema de coordenação de passagem das máquinas, descrito pela rede predicado-transição da FIG. 5.5, é definido por:

- $C_{const} = \{pc_1, pc_2, pc_3, maq_1, maq_2, maq_3\}$;
- $V = \{x, y\}$;
- não existe nem condição nem ação neste caso;

$$\bullet W = Post - Pre = \begin{array}{c} PC \\ US \\ MAQ \end{array} \begin{array}{cc} ini & fim \\ \left[\begin{array}{cc} - \langle x \rangle & \langle x \rangle \\ \langle x, y \rangle & - \langle x, y \rangle \\ - \langle y \rangle & \langle y \rangle \end{array} \right] \end{array};$$

$$\bullet M_0 = \begin{array}{c} PC \\ US \\ MAQ \end{array} \left[\begin{array}{c} \langle pc_1 \rangle + \langle pc_2 \rangle + \langle pc_3 \rangle \\ 0 \\ \langle maq_1 \rangle + \langle maq_2 \rangle + \langle maq_3 \rangle \end{array} \right]. \quad \diamond$$

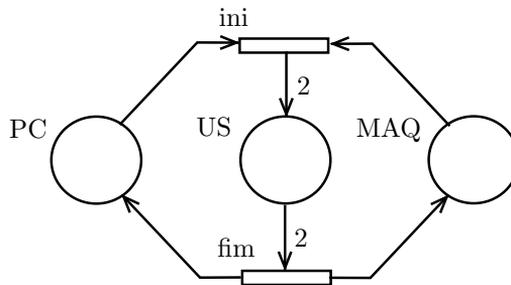


Figura 5.6: Rede de Petri subjacente

Na definição de rede de Petri predicado-transição surge a noção de rede subjacente que é obtida substituindo-se as variáveis por um peso unitário e desfazendo-se as n-uplas. A rede da FIG. 5.6 é a rede subjacente da rede da FIG. 5.5. O peso dois associado aos arcos (*ini*, *US*) e (*US*, *fim*) é obtido substituindo-se uma ficha a cada variável da etiqueta $\langle x, y \rangle$. Esta rede representa a estrutura de *controle* do sistema, enquanto que a rede predicado-transição é uma maneira estruturada de descrever o conjunto *controle e dados*.

5.2.3 Rede de Petri a objetos

Noção de objeto

O conceito de objeto, cada vez mais utilizado em engenharia de programação, consiste em estruturar uma aplicação em torno das entidades envolvidas, encapsulando, ao mesmo tempo, estruturas de dados — sob a forma de uma lista de atributos —, e métodos de transformação destes dados. Esta abordagem opõe-se ao procedimento funcional segundo o qual definem-se de maneira separada, as estruturas de dados e as funções que o sistema deve executar.

Uma *classe de objetos* é definida por um conjunto de *atributos* (também chamados *propriedades*) e um conjunto de *operações* ou *métodos* que permitem manipular os valores dos atributos. Pode-se definir uma classe a partir de outra por *herança*. A nova classe — chamada subclasse — *herda* as definições de atributos da classe anteriormente definida. Estas definições podem ser completadas por atributos e operações específicas.

As classes são apenas definições. Um objeto particular é uma *instância de classe de objeto*. Pode-se atribuir valores aos atributos destes objetos e executar suas operações. O encapsulamento confere aos objetos uma certa autonomia e uma certa remanência. Um objeto *nasce* (instanciação) e *desaparece* (destruição) dinamicamente durante a execução do programa, mas tais eventos devem ser raros em relação ao número de eventos correspondendo à execução de operações sobre os atributos.

A rede a objetos pode ser considerada como uma utilização da rede de Petri predicado-transição no contexto de uma abordagem a objetos. As fichas não são mais constantes, mas sim instâncias de n-uplas de classes de objetos. Além dos atributos definidos pela classe, um atributo *implícito* contém o nome do lugar em que o objeto está localizado. As operações são associadas às transições, e correspondem às pré-condições (filtros) que operam sobre os atributos dos objetos situados nos lugares de entrada, e às ações que modificam estes valores. Uma operação associada a uma transição t só poderá ser executada por um objeto se este estiver localizado num lugar de entrada de t .

De um certo ponto de vista, esta abordagem é menos estruturante que a abordagem a objeto clássica, porque uma operação é definida no quadro de um conjunto de classes de objetos (as classes dos objetos podendo estar nos lugares de entrada) e não de uma única classe. De um outro ponto de vista, tal abordagem é mais estruturante, pois introduz uma noção de *controle*. Para que uma operação seja aplicável sobre os atributos de uma instância de objeto, é necessário que esta esteja num certo *estado*, isto é, que esteja localizada num certo lugar.

No exemplo que está sendo tratado, os objetos são associados diretamente aos objetos físicos: peças e máquinas. Cada um destes objetos pode possuir um certo número de atributos como:

- o nome da instância de objeto;
- a data de entrega da peça;
- a operação que deve ser executada na peça;
- a lista de operações que podem ser feitas por uma máquina.

Definição da rede de Petri a objeto (simplificada)

Uma rede de Petri a objeto é definida pela 9-upla:

$$N_o = \langle P, T, C_{class}, V, Pre, Post, A_{tc}, A_{ta}, M_0 \rangle,$$

onde:

- C_{class} é um conjunto finito de classes de objetos, eventualmente organizado em uma hierarquia e definindo para cada classe um conjunto de atributos;
- P é um conjunto finito de lugares cujos tipos são dados por C_{class} ;

- T é um conjunto finito de transições;
- V é um conjunto de variáveis cujos tipos são dados por C_{lass} ;
- Pre é a função *lugar precedente* que a cada arco de entrada de uma transição faz corresponder uma soma formal de n-uplas de elementos de V ;
- $Post$ é a função *lugar seguinte* que a cada arco de saída de uma transição faz corresponder uma soma formal de n-uplas de elementos de V ;
- A_{tc} é uma aplicação que a cada transição associa uma condição fazendo intervir as variáveis formais associadas aos arcos de entrada e aos atributos das classes correspondentes;
- A_{ta} é uma aplicação que a cada transição associa uma ação fazendo intervir as variáveis formais associadas aos arcos de entrada e aos atributos das classes correspondentes;
- M_0 é a marcação inicial que associa a cada lugar uma soma formal de n-uplas de instâncias de objetos (os objetos devem ser representados por identificadores, como por exemplo, seu nome).

Exemplo resolvido 3 (rede a objetos): (continuação)

A representação gráfica do modelo rede de Petri a objetos do sistema de coordenação é a mesma da FIG. 5.5.

Para esta rede são definidos:

- o conjunto de classes $C_{lass} = \{peça, máquina\}$ com:
 - $máquina = \begin{cases} nome : \text{identificador} \\ operações : \text{lista de operações possíveis} \\ manutenção : \text{data de parada para manutenção;} \end{cases}$
 - $peça = \begin{cases} nome : \text{identificador} \\ operação : \text{operação a ser executada} \\ data : \text{data de entrega;} \end{cases}$
- as variáveis x , de tipo $peça$ e y , de tipo $máquina$: $tipo(x) = peça$ e $tipo(y) = máquina$;
- os lugares possuem os tipos: $tipo(PC) = peça$, $tipo(MAQ) = máquina$ e $tipo(\langle PC, MAQ \rangle) = (peça, máquina)$;
- as matrizes Pre e $Post$ são dadas por:

$$Pre = \begin{bmatrix} \langle x \rangle & 0 \\ 0 & \langle x, y \rangle \\ \langle y \rangle & 0 \end{bmatrix} \quad Post = \begin{bmatrix} 0 & \langle x \rangle \\ \langle x, y \rangle & 0 \\ 0 & \langle y \rangle \end{bmatrix};$$

- um exemplo de condição pode ser:

$$A_{tc}(ini) = x.operação \in y.operações;$$

- a marcação inicial é definida por:

$$M_0 = \begin{bmatrix} \langle pc_1 \rangle + \langle pc_2 \rangle + \langle pc_3 \rangle \\ 0 \\ \langle maq_1 \rangle + \langle maq_2 \rangle + \langle maq_3 \rangle \end{bmatrix},$$

em que os objetos pc_1, pc_2, pc_3 são da classe *peça*, e os objetos maq_1, maq_2, maq_3 são da classe *máquina*. \diamond

Análise

As redes de Petri predicado-transição e a objetos são definidas sob a forma de redes de Petri ordinárias (redes subjacentes) munidas de inscrições. A análise é feita primeiramente sobre estas redes, da mesma forma como no caso das redes interpretadas. Os invariantes da rede de Petri subjacente fornecem resultados sobre a conservação de números de objetos independentemente de suas classes.

Se a solução da equação $f^T W = 0$ (em que $W = Post - Pre$ é a matriz de incidência condensada do sistema) é sem variáveis, — o que nem sempre é o caso — tal solução pode ser utilizada como um invariante de lugar. Para transformar a matriz W de modo a eliminar variáveis, pode-se utilizar o método de *projeção* de lugares.

Projeção de lugares

Considere $R = \{\langle a, b \rangle, \langle a, c \rangle, \langle b, c \rangle\}$ uma relação binária num conjunto $D = \{a, b, c\}$. Se não estamos interessados em conhecer quais indivíduos ocorrem na segunda posição da dupla $\langle x, y \rangle$, podemos escrever $R = \{\langle a, - \rangle, \langle a, - \rangle, \langle b, - \rangle\}$, ou $R = \{2 \langle a, - \rangle, \langle b, - \rangle\}$. Podemos eliminar a i -ésima posição de R , fazendo uma *projeção* ao longo da i -ésima posição, denotada $|R|_i$. Assim, $|R|_2 = \{2 \langle a \rangle, \langle b \rangle\}$ e $||R|_2|_1 = \{3 \langle \rangle\}$.

N-uplas que diferem somente na i -ésima posição (localizadas num mesmo lugar) não são mais distinguíveis nas suas ocorrências, mas são contadas se $| \cdot |_i$ é aplicado ao lugar. Ao fazer a projeção de um lugar, consideram-se todos os arcos adjacentes a este lugar. Se a rede predicado-transição é uniforme, a projeção total é uma rede ordinária.

Fazer uma projeção é, de uma certa forma, construir uma rede de Petri subjacente relativa a apenas uma classe de objeto (ou um conjunto de classes). A informação obtida, mais precisa que a obtida com a rede subjacente, aproxima-se daquela obtida com as redes de Petri coloridas.

No caso das matrizes $Post, Pre$ e W , se estamos interessados apenas numa classe $class_i$ de C_{lass} , todas as outras classes são projetadas em primeiro lugar, projetando por último $class_i$. Desta forma, todas as variáveis correspondentes aos indivíduos que não se deseja observar são suprimidos da matriz.

No exemplo do sistema de coordenação, para procurar os invariantes relativos às peças, faz-se primeiro uma projeção em relação às máquinas, conservam-se somente as variáveis $\langle x \rangle$; depois faz-se uma última projeção em relação às peças. Assim, x é substituída por “1” e y por “0”. Para os invariantes relativos às máquinas, faz-se o contrário (primeiro as peças, depois as máquinas).

Para a matriz de incidência W da rede da FIG. 5.5:

$$\begin{array}{l} PC \\ US \\ MAQ \end{array} \begin{array}{cc} ini & fim \\ \left[\begin{array}{cc} -\langle x \rangle & \langle x \rangle \\ \langle x, y \rangle & -\langle x, y \rangle \\ -\langle y \rangle & \langle y \rangle \end{array} \right], \end{array}$$

fazendo-se primeiramente a projeção sobre as máquinas, obtém-se:

$$|W|_y = \begin{bmatrix} -\langle x \rangle & \langle x \rangle \\ \langle x, 1 \rangle & -\langle x, 1 \rangle \\ 1 & 1 \end{bmatrix},$$

e depois sobre as peças, obtém-se, finalmente,

$$\|W\|_y|_x = \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ 0 & 0 \end{bmatrix}.$$

Fazendo a projeção em relação à peça e depois à máquina, obtém-se, respectivamente¹,

$$|W|_x = \begin{bmatrix} 0 & 0 \\ \langle y \rangle & -\langle y \rangle \\ -\langle y \rangle & \langle y \rangle \end{bmatrix} \text{ e } \|W\|_x|_y = \begin{bmatrix} 0 & 0 \\ 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

Obtém-se, assim, nos dois casos, uma matriz de incidência C de inteiros naturais e pode-se calcular os componentes como na rede de Petri ordinária, utilizando a equação $f.C = 0$. O invariante de lugar é, agora, uma função linear não mais da marcação dos lugares, mas de sua projeção, isto é, do número de objetos da classe $class_i$ contidos no lugar.

O componente conservativo da matriz obtida após a projeção sobre a peça é:

$$f_{pe\zeta a} = [1 \quad 1 \quad 0]$$

e o invariante de lugar correspondente é dado por:

$$\|M(PC)\|_{pe\zeta a} + \|M(US)\|_{pe\zeta a} = \text{constante}.$$

O componente conservativo da matriz obtida após a projeção sobre a máquina é:

$$f_{máquina} = [0 \quad 1 \quad 1]$$

com o invariante de lugar correspondente dado por:

$$\|M(US)\|_{máquina} + \|M(MAQ)\|_{máquina} = \text{constante}.$$

Desta forma, obtém-se resultados análogos aos encontrados com as redes de Petri coloridas.

Outra forma de análise é introduzir transições *erro* no modelo da rede predicado-transição. O projeto é correto se e somente se a transição é não viva. As transições não vivas exprimem asserções invariantes.

¹ Para simplificar, diz-se que foi feita uma projeção em relação à $class_i$ quando esta é a última projeção realizada.

5.3 Características dos modelos

As duas grandes vantagens das redes de Petri que são, de um lado, seu poder descritivo (como a facilidade de modelar o paralelismo e a concorrência), e de outro lado, a possibilidade de análise, serão utilizadas como critérios de comparação entre os modelos de RPN vistos no item anterior. Esta comparação deve ser feita dentro do âmbito do sistema que se deseja modelar, no nosso caso, os sistemas de manufatura.

Embora todos os modelos possuam as características gerais vistas no item 5.1, cada um introduziu-as de maneira diferente. Estas diferenças serão ressaltadas e a seguir serão apresentadas as características particulares introduzidas por cada modelo.

5.3.1 A ficha como elemento de informação

A primeira diferença entre os modelos concerne à maneira de diferenciar as fichas. Ou cada ficha representa um indivíduo passando de um lugar a outro, ou representa uma informação que se interpreta de acordo com o lugar que a contém.

A ficha representa uma *informação*

No modelo de rede de Petri colorida, à cada ficha é associada uma cor. Assim, a cada lugar é associado explicitamente um conjunto de cores. Essa cor representa uma informação. Por exemplo, na FIG. 5.4, que mostra o caso de um sistema de manufatura modelado por uma rede colorida, as cores do lugar *MAQ* correspondem diretamente às identidades das máquinas. No entanto, as cores do lugar *US* correspondem às operações de usinagem, quer dizer, uma combinação de uma peça e uma máquina.

A ficha representa um *indivíduo*

No modelo de redes de Petri predicado-transição, a ficha representa um indivíduo pertencente a um conjunto de constantes.

O objetivo central do desenvolvimento da rede predicado-transição é, justamente, a introdução, de uma maneira formal, do conceito de indivíduos com propriedades e relações modificáveis, na teoria de redes. Este fato é análogo à introdução de indivíduos e suas estruturas na lógica, o que leva da lógica proposicional à lógica de predicados de primeira ordem.

Voltando ao lugar *US* do exemplo, as fichas são n -uplas $\langle x, y \rangle$ em que x representa uma peça e y uma máquina. Em consequência, essas fichas não somente guardam a identidade dos indivíduos, mas também representam claramente relações dinâmicas entre estes.

No modelo de redes de Petri a objetos, representado na FIG. 5.5, as fichas são consideradas como indivíduos cujo valor é variável. A cada ficha é associada uma estrutura de dados, representada por um objeto. Cada objeto pertence a uma classe de objetos à qual são definidas propriedades (atributos). Cada classe pode ser subdividida em subclasses.

5.3.2 Dobramento das transições e dos lugares

Como foi explicitado no item 5.1, todos os modelos são baseados na idéia de *dobrar* redes ordinárias. Mas esse dobramento pode ser feito com duas filosofias diferentes.

No caso das redes predicado-transição e a objetos, a transição é vista como uma regra e a idéia é de passar de regras sem variáveis a regras com variáveis, seguindo a idéia mencionada no item 5.2.2 de passagem da lógica proposicional à lógica de predicados de primeira ordem. No caso da rede colorida o enfoque é dado na visão matricial da rede de Petri e, portanto, a abordagem é a de passar de uma matriz de inteiros a uma matriz de funções, cada função sendo representada por uma matriz de inteiros. Estes pontos serão detalhados a seguir.

Uso de variáveis

Na rede de Petri predicado-transição, o lugar *dobrado* é visto como um *esquema lugar*, representado por um predicado $P(x)$ (ou $P(x, y)$ se existe uma relação dinâmica). A transição *dobrada* é vista como um *esquema transição*, representada por uma regra do tipo *se* $P_1(x)$ e \dots $P_2(y)$ *então* $Q_1(z)$ e \dots $Q_2(w)$ onde $P_i(\cdot)$ e $Q_j(\cdot)$ indicam respectivamente os predicados relativos aos lugares de entrada e saída, e x, y, z, w , as variáveis que etiquetam os arcos. Na rede de Petri a objetos, os lugares pertencem a classes de lugares que indicam as classes de objetos que podem pertencer ao lugar. Neste caso, as variáveis pertencem a classes de variáveis que indicam as classes de objetos que podem substituí-las.

Uso de funções

No modelo de rede de Petri colorida (FIG. 5.4), a cada lugar é associado, explicitamente, um conjunto de cores. O mesmo é feito com cada transição. Os arcos, em lugar de serem etiquetados por somas formais de variáveis, o são por funções das cores da transição nas cores do lugar. Esta é uma diferença fundamental em relação à rede predicado-transição.

5.4 Escolha do modelo

Um sistema de manufatura caracteriza-se principalmente pela capacidade de fabricar vários tipos de produtos (processos de fabricação) simultaneamente e também por permitir rapidamente a modificação do plano de fabricação, seja quanto ao número de peças de cada processo, seja do próprio processo.

Numa boa modelagem, a relação entre os objetos físicos e as entidades matemáticas correspondentes é a mais clara e a mais direta possível. É por esta razão que as classes de redes de Petri em que a individualização das fichas é baseada na noção de *indivíduo* são mais convenientes que aquelas que utilizam a noção de cor representando uma simples *informação*. Deste modo, os elementos físicos do sistema, tais como as peças, as máquinas, as ferramentas, etc., são diretamente descritos através de fichas.

Mais um motivo do interesse desta forma de representação, ainda do ponto de vista da ficha, é o fato de que as n-uplas representam fielmente relações dinâmicas existentes, num dado momento, entre elementos físicos do sistema.

Dentre as classes de redes em que as fichas representam indivíduos, o modelo das redes a objetos apresenta uma característica muito importante para este tipo de aplicação: a associação de uma estrutura de dados a cada ficha.

Embora na rede predicado-transição é possível também representar uma estrutura de dados através de uma n-upla, esta representação é mais cômoda na rede a objetos.

Isto se torna mais claro a partir do momento em que se deseja representar muitos dados sobre a ficha: o valor de uma n -upla pode ocupar toda uma linha...

Além disso, a noção de classe permite guardar sempre em mente as características dos objetos, e permite, também, utilizar sobre as redes somente os atributos que nos interessam no momento. A noção de subclasse contribui, também, para uma descrição mais concisa e mais estruturada, além de mais legível. De fato, num sistema de manufatura, a cada peça é associada uma série de informações como: processo de fabricação ao qual ela pertence, ordem de operação dentro do processo (atual ou a próxima), máquina à qual está afetada. Da mesma forma, cada máquina é capaz de tratar um conjunto de operações, pode estar livre ou ocupada, etc.

A noção de classe e subclasse na rede a objetos também é extremamente útil quando da modelagem por refinamentos: partindo-se de uma rede inicial, com uma determinada estrutura de dados, vai-se refinando, em paralelo, a rede e as classes de objetos.

A introdução de novas peças num sistema de manufatura é um bom exemplo para ilustrar a adequação da rede a objeto. No caso da rede colorida, é preciso acrescentar novas cores no conjunto de cores e, portanto, modificar as funções sobre os arcos. No caso das redes predicado-transição e a objetos, há modificações somente quando as novas peças pertencem a novos processos de fabricação: novas classes de objetos (ou subclasses) devem ser definidas.

Isto não quer dizer que a conclusão será a mesma para qualquer aplicação. Por exemplo, se a noção de rotação (simetria entre as cores) existe de maneira evidente dentro do sistema a ser modelado, é aconselhado o uso da rede colorida.

A escolha da classe pode depender não somente das características do sistema a ser modelado, mas também do ambiente geral de trabalho, quer dizer, das outras ferramentas com que se trabalha eventualmente. Por exemplo, no caso de sistemas de manufatura que utilizam técnicas da Inteligência Artificial para tratar do controle no nível da planificação e do ordenamento em tempo real, tem-se uma razão a mais para escolher a rede predicado-transição ou a rede a objetos. De fato, a filosofia destas classes é bem próxima de um sistema de regras, ao contrário do caso da rede colorida.

Comparação com a rede de Petri ordinária

A modelagem de Sistemas de Manufatura a fim de conceber o comando e a supervisão implica modelar a parte do banco de dados técnicos que evoluem em tempo real.

Existem basicamente duas opções, utilizando:

- rede de Petri ordinária interpretada: a rede ordinária modela o controle, e a estrutura de dados evolui, separadamente, em paralelo. A interação entre ambas dá-se através da interpretação;
- rede de Petri de alto nível: o controle é modelado pela rede (subjacente) e a estrutura de dados é modelada na própria ficha, através de objetos ou constantes (rede a objetos e predicado-transição) ou através das cores (rede colorida).

Ambas as opções permitem igualmente a modelagem. A escolha vai depender principalmente de critérios como a complexidade da estrutura de controle e da estrutura de dados. É isto, tanto do ponto de vista da modelagem quanto do ponto de vista da implementação (ver capítulo 7).

5.5 Notas

Para uma definição formal da rede de Petri colorida ver JENSEN (1986); para a rede predicado-transição, ver GENRICH (1987); para a rede a objetos, ver SIBERTIN-BLANC (1985). Existem outros modelos ainda de rede de Petri de alto nível, como a rede de Petri a indivíduos, definida por REISIG (1985).

5.6 Exercícios

1. Considere o mesmo problema dos bombeiros da lista de exercícios do capítulo 1, mas agora com quatro bombeiros: um perto do fogo, um perto do hidrante e dois que estão no caminho para auxiliar. O mecanismo de sincronização é o mecanismo clássico de *pipeline*. Modele o sistema utilizando: a) rede de Petri colorida, dando o conjunto de cores dos lugares e transições, bem como as funções etiquetando os arcos; b) rede de Petri a objetos, indicando as variáveis nos arcos e as classes utilizadas.
2. Considere, no problema do torno da lista de exercícios do capítulo 1 (FIG. 1.15), que existem três tipos de peças, com vários exemplares de cada tipo na célula. É necessário, portanto, três tipos de programas diferentes para o torno; o micrômetro deve inspecionar as peças segundo seus tipos. Para simplificar o problema, considere que o robô se adapte automaticamente ao tamanho das peças. Construa uma rede de Petri a objetos que modele este sistema. Sugestão: utilize um lugar que contenha os programas para o torno e para o micrômetro.

Capítulo 6

REDES DE PETRI E A REPRESENTAÇÃO DO TEMPO

A rede de Petri permite descrever causalidade (um evento é consequência de um outro), decisão e independência entre eventos: o evento a é causa do evento b , evento a precede evento b , a e b são ordenados no tempo. É uma maneira de descrever uma ordem parcial num conjunto de eventos e, portanto, de introduzir o tempo de modo *quantitativo*.

Vários trabalhos foram realizados no sentido de utilizar *explicitamente* o tempo, como um parâmetro contínuo e quantificável, seja para avaliação do desempenho, seja para a verificação formal. Estando associado diretamente à rede de Petri, de um certo ponto de vista, o tempo faz parte do *controle* ao invés de ficar na parte de *dados* de forma não estruturada.

Existem duas grandes classes de modelo: rede de Petri temporal e rede de Petri temporizada.

No modelo temporal é associado a cada transição um par de datas $(\theta_{min}, \theta_{max})$. θ_{min} indica a duração mínima de sensibilização da transição antes do disparo, enquanto θ_{max} permite calcular a duração máxima de sensibilização. A transição deve disparar neste intervalo de tempo.

No segundo modelo, é associada uma duração de tiro às transições. A rede é chamada de rede de Petri com transição temporizada. Outros modelos da mesma família são as redes com lugares temporizados e as redes com arcos temporizados.

6.1 Rede de Petri temporizada

6.1.1 Tempo associado ao lugar

Se um lugar representa uma atividade A , trata-se simplesmente de colocar θ como a duração desta atividade. Assim, quando a ficha chega na data τ no lugar p , ela só pode deixá-lo após $\tau' = \tau + \theta$ instantes.

O lugar p pode desdobrar-se numa seqüência *lugar-transição-lugar*, como representado na FIG. 6.1.a, pois as duas redes da figura são equivalentes, já que o lugar p_1 é substituível (item 3.3.1). O lugar p_1 corresponde à atividade sendo executada, a transição t' corresponde ao evento *tempo transcorrido* e o lugar p_2 corresponde a uma eventual espera (sincronização com outras atividades) após o final da atividade A . Enquanto a atividade está sendo executada, a ficha não pode ser utilizada para disparar a transição. A ficha é

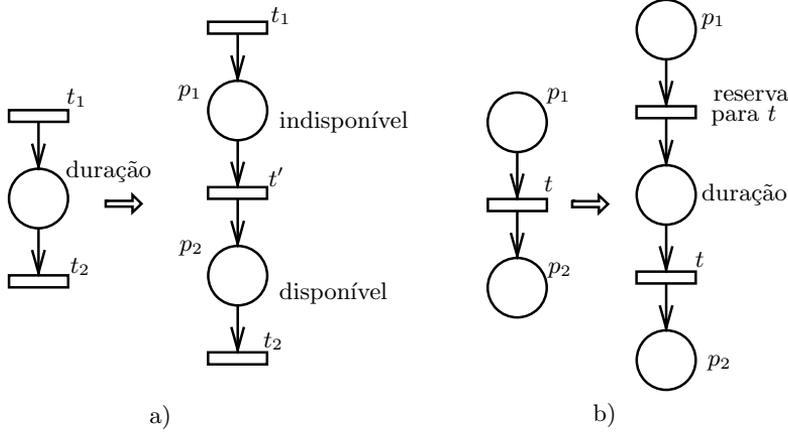


Figura 6.1: Temporização da rede de Petri

dita *indisponível*. Desde que a atividade estiver acabada, a ficha se torna *disponível* e t_2 pode eventualmente ser disparada. Assim, a ficha pode estar em dois estados: disponível ou indisponível. Somente as fichas disponíveis podem sensibilizar a transição.

Sendo $M_d(p)$ o conjunto de fichas disponíveis no lugar p , uma transição t está sensibilizada se e somente se $M_d(p) \geq Pre(p, t)$.

6.1.2 Tempo associado à transição

A rede de Petri temporizada é obtida associando-se a cada transição da rede ordinária uma duração de tiro. Sua semântica é uma noção de atraso durante a qual as fichas utilizadas para disparar a transição *não são disponíveis* ou *visíveis* em nenhum lugar. A partir do momento que a ficha se torna visível num lugar, ela pode ser utilizada imediatamente por qualquer transição de saída deste lugar.

Portanto, o disparo não é instantâneo, mas possui uma duração. Esta associação de tempo à transição só tem sentido se a transição é interpretada como uma atividade e não como um evento instantâneo. Supõe-se, implicitamente, que esta atividade não é interrompível em relação ao comportamento global do sistema modelado ou, então, que se trata simplesmente de uma abreviação.

Pode-se obter uma rede equivalente, substituindo-se a transição por uma seqüência *transição-lugar-transição* (FIG. 6.1.b). A primeira transição corresponde ao evento *instantâneo* de início de atividade (são retiradas as fichas referentes ao peso do arco). O lugar serve para memorizar a *atividade* sendo executada, e a última transição corresponde ao evento *instantâneo* de fim de atividade (fichas colocadas nos lugares de saída).

Esta equivalência é mostrada na FIG. 6.1.b. O tiro da primeira transição após p_1 corresponde à reserva das fichas (as fichas reservadas não podem ser utilizadas para disparar uma outra transição diferente de t). Após o tiro da segunda transição as fichas são liberadas.

As fichas possuem então dois estados possíveis: *reservado* (M_r) ou *não reservado* (M_{nr}). Somente as não reservadas podem sensibilizar a transição. Portanto, a marcação a cada momento é dada por

$$M = M_r + M_{nr}.$$

A rede de Petri temporizada pode associar o tempo ao lugar ou à transição. As fichas *desaparecem* ou ficam *indisponíveis* ou *reservadas* durante um certo tempo, e reaparecem após na marcação. A FIG. 6.1 mostra que as duas abordagens são equivalentes.

Definição:

Uma rede de Petri temporizada é um par $N_t = \langle N, \Theta_f \rangle$ onde:

- N é uma rede de Petri $\langle P, T, Pre, Post \rangle$ com uma marcação inicial M_0 ;
- $\Theta_f : T \rightarrow \mathcal{Q}^+$ é a função *duração de disparo*, que associa a cada transição um número racional positivo que descreve a duração do disparo.

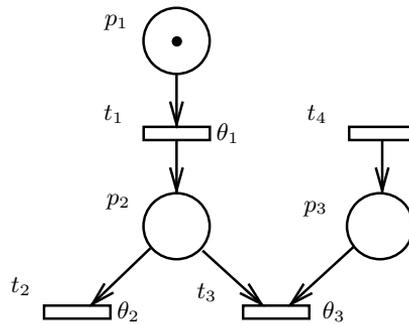


Figura 6.2: O tempo e a rede de Petri

Exemplo resolvido 5:

Considere a rede de Petri representada na FIG. 6.2, com o lugar p_1 marcado no tempo $\tau = 0$. Considere que esta rede é temporizada, com o tempo associado às transições. A cada transição t_i é associada uma duração de disparo θ_i .

A transição t_1 é disparada no tempo τ_1 , mas a ficha somente será visível em p_2 no tempo $\tau_1 + \theta_1$, sensibilizando neste momento a transição t_2 . Se a transição t_4 (não temporizada) é disparada no tempo τ_2 , p_3 é imediatamente marcado e as transições t_2 e t_3 estão sensibilizadas e em conflito após o tempo $\max[(\tau_1 + \theta_1), \tau_2]$. \diamond

6.2 Rede de Petri temporal

A rede de Petri temporal é obtida associando-se a cada transição uma duração de sensibilização. O tiro é instantâneo, mas a transição deve estar sensibilizada durante o intervalo de tempo dado.

Um intervalo $(\theta_{min}, \theta_{max})$ é associado a cada transição; a duração de sensibilização deve ser maior que θ_{min} e menor que θ_{max} (a transição deve disparar antes de θ_{max}).

Diferente do modelo rede de Petri temporizada, durante a sensibilização de uma transição, as fichas continuam a ser “vistas” por outras transições de saída do lugar.

Pode-se associar um intervalo $[a, a]$ a uma transição para representar uma duração a . Se a transição estiver sensibilizada no tempo τ , ela irá disparar no tempo $\tau + a$, caso continue, nesta data, sensibilizada pela marcação.

Definição:

Uma rede de Petri temporal é um par $N_t = \langle N, I \rangle$ onde:

- N é uma rede de Petri $\langle P, T, Pre, Post \rangle$ com uma marcação inicial M_0 ;
- $\theta(t) = [\theta_{min}(t), \theta_{max}(t)]$ é uma função que, a cada transição t , associa um intervalo fechado racional que descreve uma *duração de sensibilização*.

Exemplo resolvido 5: (continuação)

Considere a rede de Petri representada na FIG. 6.2, com o lugar p_1 marcado no tempo $\tau = 0$. Considere que esta rede é temporal. A cada transição t_i é associada uma duração de sensibilização, dada pelo intervalo $\theta_i = (\theta_{min}(t_i), \theta_{max}(t_i))$.

Se a transição t_1 disparar no tempo τ_1 , $\tau_1 \in (\theta_{min}(t_1), \theta_{max}(t_1))$, o lugar p_2 estará marcado nesta data. A transição t_2 pode, então, disparar em $\tau_1 + \theta_2$. Se a transição t_4 for disparada no tempo τ_2 , marcando o lugar p_3 , a transição t_3 só poderá disparar no tempo $max(\tau_1, \tau_2) + \theta_3$.

O conflito entre t_2 e t_3 existirá em função da relação entre as datas em que os lugares p_2 e p_3 são marcados, e dos valores de θ_2 e θ_3 . Se $\theta_3 = 0$, $\theta_2 \neq 0$ e $\tau_2 < \tau_1$, t_3 já está sensibilizada a partir da data τ_1 , enquanto t_2 , nesta data, está sensibilizada apenas pela marcação. Portanto, t_3 dispara no tempo τ_1 e não existe conflito entre t_2 e t_3 .

Entretanto, se $\theta_3 \neq 0$, o conflito existe durante a intersecção dos intervalos $(\tau_1 + \theta_2)$ e $max(\tau_1, \tau_2) + \theta_3$. \diamond

6.2.1 Representação do *watchdog*

Certos mecanismos implicam que uma ficha esteja visível somente para certas transições a um dado instante. Um caso típico é o *watchdog*. Considere a FIG. 6.3. O lugar *espera* permite que o sistema esteja receptivo à chegada de um evento sob a forma de uma ficha no lugar *condição* (disparo da transição fim_1). Mas se este lugar não estiver marcado ao fim do tempo θ , é acionado um alarme (disparo da transição fim_2).

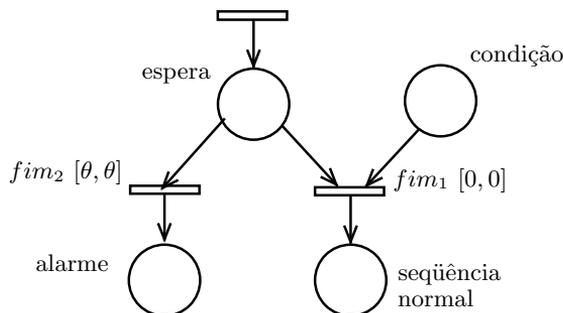


Figura 6.3: *Watchdog*

As duas formas de associar o tempo na rede de Petri temporizada (lugar ou transição) não permitem uma representação correta do *watchdog* pois:

- um tempo de atividade associado ao lugar *espera* atrasaria o disparo de fim_1 mesmo se o lugar *condição* contivesse uma ficha mais cedo;

- um tempo de atividade associado à transição fim_2 faria com que a ficha contida no lugar *espera* fosse imediatamente reservada (de modo irreversível) e que o alarme fosse acionado após o tempo θ , mesmo se a ficha tivesse, durante este tempo, chegado no lugar *condição*.

A solução é dada pela rede temporal, que associa uma *duração de sensibilização* $\theta(t)$ a cada transição t . Se esta transição se tornar sensibilizada (pela marcação) no tempo τ , ela continuará sensibilizada durante todo o intervalo $(\tau + \theta(t))$, desde que a marcação que a sensibiliza não for modificada pelo tiro de outra transição. A diferença em relação à *duração de disparo* é que, durante todo este intervalo, as fichas estão disponíveis nos lugares de entrada de t e podem, eventualmente, ser utilizadas por uma transição em conflito com t , como no exemplo da FIG. 6.2.

Voltando à FIG. 6.3, basta associar à transição fim_2 uma duração θ igual ao valor do *watchdog* e à transição fim_1 , uma duração de sensibilização nula.

O domínio principal de utilização deste modelo (validação de protocolos de comunicação) fez com que, desde o início, fosse definida uma duração *imprecisa* de sensibilização de t sob a forma de um intervalo $\theta(t) = [\theta_{min}(t), \theta_{max}(t)]$. A transição t só poderá ser disparada depois que o tempo $\theta_{min}(t)$ tenha transcorrido e não poderá ser disparada após o tempo $\theta_{max}(t)$. Todos os valores de $\theta(t)$:

$$\theta_{min}(t) \leq \theta(t) \leq \theta_{max}(t)$$

correspondem a durações de sensibilização para a transição t .

6.2.2 Comparação entre os dois modelos

O modelo de rede de Petri temporal é mais geral que o de rede de Petri temporizada, pois permite descrever o *watchdog*, o que a rede temporizada não permite. Neste caso, basta, na FIG. 6.2, usar θ_2 como a duração do *watchdog* e fazer $\theta_3 = 0$, para que a transição t_3 , que representa o fim da operação normal, seja disparada assim que sensibilizada pela marcação.

Para passar de uma rede de Petri temporizada a uma rede de Petri temporal, basta substituir a transição t da primeira por uma seqüência $t_1 t_2$ com:

- $\theta_{min}(t_1) = \theta_{max}(t_1) = 0$
- e $\theta_{min}(t_2) = \theta_{max}(t_2) = \theta(t)$.

Quanto às redes de Petri ordinárias (sem explicitação do tempo), correspondem simplesmente ao caso em que todos os intervalos de sensibilização são iguais a $[0, \infty)$.

6.3 Rede de Petri estocástica

6.3.1 Limite das redes temporizada e temporal

A utilização das redes de Petri temporizadas permite a construção de modelos realistas para avaliar sistemas a eventos discretos para *simulação*. De fato, nada impede modificar a função Θ_f para associar às transições o resultado de uma tiragem aleatória

ao invés de um valor constante. Entretanto, as propriedades de uma rede temporizada, ou de uma rede interpretada de modo geral, podem ser diferentes daquelas obtidas sobre a rede de Petri subjacente, como discutido no item 4.4.

A utilização da rede de Petri temporal permite facilmente descrever o mecanismo de *watchdog* e a construção de um grafo de cobertura dos estados acessíveis em casos simples em que a rede de Petri subjacente é limitada, e os intervalos de tempo associados às transições são dados sob a forma de inteiros. Este modelo tem servido para verificar formalmente um certo número de protocolos. Entretanto, o tamanho do grafo de cobertura explode rapidamente quando os intervalos de tempo são muito diferentes.

O problema principal vem do fato de que a descrição do estado destas redes deve compreender não somente a marcação, mas também as informações temporais.

Para poder utilizar a potência da análise markoviana é necessário que os sistemas sejam sem memória do passado, isto é, se um evento produzir um disparo da transição t e transformar a marcação M_1 em M_2 , a evolução futura das transições que estavam sensibilizadas por M_1 antes do disparo de t deverá ser idêntica àquela que as transições sofreriam se viessem a ser sensibilizadas por M_2 .

Somente as distribuições geométricas e exponenciais verificam este fato. As redes de Petri estocásticas são, portanto, as redes em que as durações de sensibilização associadas às transições são definidas por tais distribuições, a fim de poder construir um processo markoviano equivalente e assim analisar o comportamento da rede.

6.3.2 Duração de sensibilização estocástica

No caso da rede de Petri temporal, a duração de sensibilização é uma variável que assume um valor dentro do intervalo $(\theta_{min}, \theta_{max})$, sendo estes valores igualmente distribuídos. Na rede de Petri estocástica, esta duração de sensibilização é uma variável estocástica com uma distribuição de probabilidade exponencial

$$Pr_{[\theta \leq \tau]} = 1 - e^{-\lambda\tau}.$$

A função $Pr_{\theta(\tau)}$ descreve a probabilidade que o disparo da transição t aconteça antes da data τ e, portanto, que a duração de sensibilização seja inferior a esta data.

Seja λ a taxa de transição; o valor médio da duração de sensibilização é dado por:

$$\bar{\theta} = \int_0^{\infty} (1 - Pr_{\theta(\tau)}) d\tau = \int_0^{\infty} e^{-\lambda\tau} d\tau = \frac{1}{\lambda}.$$

Definição:

Uma rede de Petri estocástica é um par $N_{ts} = \langle N, \Lambda \rangle$

- N é uma rede de Petri com uma marcação inicial;
- Λ é uma função que associa, a cada transição t , uma taxa de transição $\lambda(t)$.

Isto equivale a associar a cada transição um intervalo contínuo de sensibilização $[0, \infty)$ com uma distribuição exponencial. Isto explica porque o conjunto de marcações acessíveis é o mesmo que o da rede subjacente. A duração média de sensibilização é $\theta_s(t) = \frac{1}{\lambda(t)}$.

6.3.3 Obtenção de uma cadeia de Markov

Considere duas marcações acessíveis M_i e M_j :

- se existe apenas uma transição tal que $M_i \xrightarrow{t_k} M_j$, a taxa de transição do estado M_i ao estado M_j é dada por $\Lambda(t)$;
- se existem duas transições t_k e t_m tal que $M_i \xrightarrow{t_k} M_j$ e $M_i \xrightarrow{t_m} M_j$, a taxa de transição do estado M_i ao estado M_j é dada por $\Lambda(t_k) + \Lambda(t_m)$.

A partir da rede de Petri estocástica $N_{ts} = \langle N, \Lambda \rangle$, pode-se contruir uma cadeia de Markov. Os estados são as marcações acessíveis do conjunto $A(R, M_0)$. A matriz Q de taxas de transição é escrita diretamente a partir da função Λ . A coluna j de Q descreve a evolução da probabilidade π_j da marcação M_j no decorrer do tempo. O termo q_{ij} é a taxa de transição para chegar no estado M_j a partir de M_i . Seu valor é negativo pois descreve a taxa de transição que permite abandonar o estado M_i .

O regime estacionário do processo markoviano é dado pelo vetor Π^* , solução da equação:

$$\Pi^{*T}Q = 0 \quad \text{com} \quad \sum_i \pi_i^* = 1. \quad (6.1)$$

Seja B uma propriedade que é verificada por um conjunto de marcações EM ; tem-se:

$$Pr_B = \sum_{M_i \in EM} \pi_i.$$

Esta equação permite calcular, por exemplo, o valor médio do número de fichas num dado lugar.

O fato de poder construir uma cadeia de Markov a partir da rede de Petri estocástica permite realizar a análise desta rede. Tem-se os seguintes resultados:

- se a rede de Petri é reiniciável, a solução da equação 6.1 é única;
- o conjunto dos lugares que pertencem a um componente conservativo positivo é análogo a uma subcadeia de Markov fechada;
- um lugar não limitado corresponde a uma cadeia aberta. Devido ao problema de complexidade, é feita a restrição de redes de Petri limitadas ou possuindo no máximo um ou dois lugares não limitados (para uma dada marcação inicial);
- uma equação particular do sistema 6.1 é descrita pela coluna j da matriz Q e corresponde à aplicação do teorema dos cortes em torno do nó correspondente à marcação M_j . Em regime estacionário, a derivada da probabilidade desta marcação é nula.

Exemplo resolvido 6:

Considere a rede de Petri estocástica da FIG. 6.4.a, com as seguintes taxas de transição:

$$\begin{aligned} \Lambda(t_1) &= \lambda_1, \quad i = 1..4 \\ \Lambda(t_5) &= M(p_3)\lambda_5. \end{aligned}$$

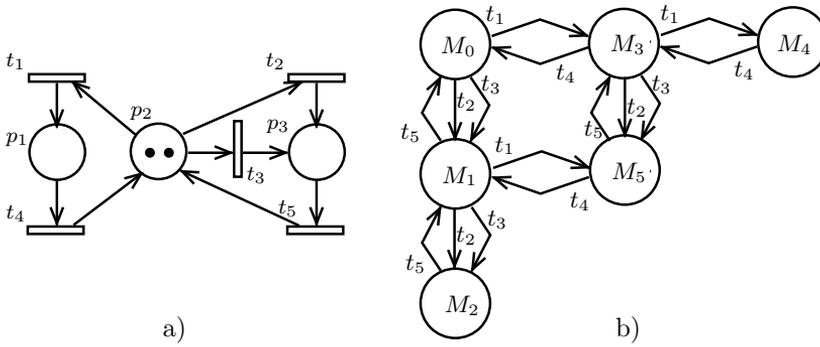


Figura 6.4: a) Rede de Petri estocástica; b) Grafo $GA(R; M)$

O grafo de marcações acessíveis é dado pela FIG. 6.4.b, com $M_0 = p_2^2$, $M_1 = p_1p_3$, $M_2 = p_1^2$, $M_3 = p_1p_2$, $M_4 = p_1^2$ e $M_5 = p_1p_3$. A matriz Q de taxas de transições, cujas linhas e colunas correspondem às marcações M_0 a M_5 é dada por:

$$\begin{bmatrix}
 -\lambda_1 - \lambda_2 - \lambda_3 & \lambda_2 + \lambda_3 & 0 & \lambda_1 & 0 & 0 \\
 \lambda_5 & a & \lambda_2 + \lambda_3 & 0 & 0 & \lambda_1 \\
 0 & 2.\lambda_5 & -2.\lambda_5 & 0 & 0 & 0 \\
 \lambda_4 & 0 & 0 & b & \lambda_1 & \lambda_2 + \lambda_3 \\
 0 & 0 & 0 & \lambda_4 & -\lambda_4 & 0 \\
 0 & \lambda_4 & 0 & \lambda_5 & 0 & -\lambda_4 - \lambda_5
 \end{bmatrix}$$

onde $a = (-\lambda_1 - \lambda_2 - \lambda_3 - \lambda_5)$ e $b = (-\lambda_1 - \lambda_2 - \lambda_3 - \lambda_4)$. A primeira coluna da matriz Q permite escrever a equação

$$\frac{d\pi_0}{dt} = -(\lambda_1 + \lambda_2 + \lambda_3)\pi_0 + \lambda_5\pi_1 + \lambda_4\pi_3$$

em que π_i é a probabilidade da marcação M_i . Observe que a soma dos elementos de uma linha de Q é nula, pois os arcos saindo de um estado M_i são exatamente os mesmos arcos que chegam a M_j a partir de M_i .

◇

6.4 Notas

A rede de Petri temporal foi introduzida por MERLIN (1974) em seu trabalho de doutoramento; a rede de Petri temporizada por RAMCHANDANI (1974) e a rede de Petri estocástica, por MOLLOY (1981). Em todos estes modelos, o tempo é utilizado explicitamente. Trabalhos fundamentados nestes modelos são utilizados para avaliar o desempenho de sistemas representados por rede de Petri, como em FREEDMAN (1988), ou para verificação formal, como em BERTHOMIEU (1983).

Para uma definição de cadeia de Markov, ver CASSANDRAS (1993).

Capítulo 7

MÉTODOS DE IMPLEMENTAÇÃO

Pode-se classificar todos os métodos de implementação de uma concepção baseada em rede de Petri segundo duas séries de escolhas principais.

A primeira escolha é se a rede de Petri é implementada como um sistema de regras ou não. Segundo esta escolha, a implementação é feita seguindo-se:

- a abordagem procedimental: consiste em desenvolver métodos que permitem escrever os programas usando linguagens procedimentais como Pascal ou ADA, cujo comportamento seja idêntico ao especificado pela rede de Petri;
- a abordagem declarativa: consiste em considerar que uma rede de Petri é um sistema de regras (ver item 2.2) e basear a implementação na utilização de um mecanismo de inferência especializado chamado *jogador de rede de Petri*. O programa que realiza este motor de inferência não depende de uma aplicação específica. Ele implementa as definições correspondentes ao comportamento dinâmico da rede de Petri (transição sensibilizada, disparo de transição, etc.).

Outra escolha fundamental concernente à implementação da rede será efetuada:

- de modo centralizado;
- de modo descentralizado ou mesmo distribuído.

É evidente que estas escolhas serão de fato efetuadas quando das últimas etapas da concepção. Estas escolhas não são totalmente independentes, pois não seria realista tentar realizar uma implementação centralizada e procedimental de um sistema descrito por uma rede de Petri possuindo um alto grau de paralelismo (grande número de transições sensibilizadas e paralelas para uma dada marcação).

Por outro lado, no caso da implementação descentralizada, pode-se implementar de modo procedimental certas partes do sistema e, de modo não procedimental, as outras partes.

7.1 Abordagem procedimental

A rede descreve um processo seqüencial

A rede corresponde a um componente conservativo que contém apenas uma ficha. Trata-se, portanto, de uma rede de Petri pertencente à subclasse das máquinas de estado.

A ficha pode ser vista como o registro de instrução do programa seqüencial que deve ter o mesmo comportamento.

Os procedimentos (*procedures*) são associados às transições ou aos lugares, segundo o tempo é associado às transições ou aos lugares. As duas formas são equivalentes.

O programa é escrito diretamente; os lugares compartilhados correspondem às instruções *if* ou *case of* ou *go to*.

Caso de uma rede qualquer

Calcula-se o grafo de marcações acessíveis da rede de Petri, o que equivale a transformar a rede numa máquina de estado equivalente. O problema é a explosão combinatória do número de estados...

7.2 Abordagem não procedimental

7.2.1 Princípio

Como foi dito no capítulo 2, a rede de Petri pode ser considerada como um sistema de regras de produção.

Cada transição é então considerada como uma regra de transformação do estado; a partir de um estado dado (marcação), deve-se procurar uma regra aplicável para passar ao estado seguinte.

A abordagem não procedimental consiste em escrever um motor de inferência especializado capaz de *jogar* a rede de Petri, isto é, de deslocar as fichas de modo a respeitar as regras de disparo das transições. No contexto da engenharia de programação, esta abordagem pode também ser considerada como uma *execução da especificação da concepção*.

Quando da definição da interpretação de uma rede de Petri, viu-se que as transições podem ser associadas a eventos externos ou a eventos internos do sistema que se procura implementar.

O motor de inferência especializado, freqüentemente chamado *jogador* de rede de Petri, tem seu funcionamento descrito pela FIG. 7.1. Este deve ser tal que o disparo das transições externas deve *sincronizar-se* com a ocorrência do evento associado. Isto corresponde à recepção de mensagens de atualização do conhecimento ou, então, a requisições provenientes do sistema.

Por outro lado, a partir de uma marcação dada, todas as transições internas sensibilizadas devem ser disparadas antes de considerar o evento externo seguinte. Elas correspondem a conclusões (decisões) internas e se traduzem, eventualmente, por emissões de mensagens (comandos ou respostas) para o exterior.

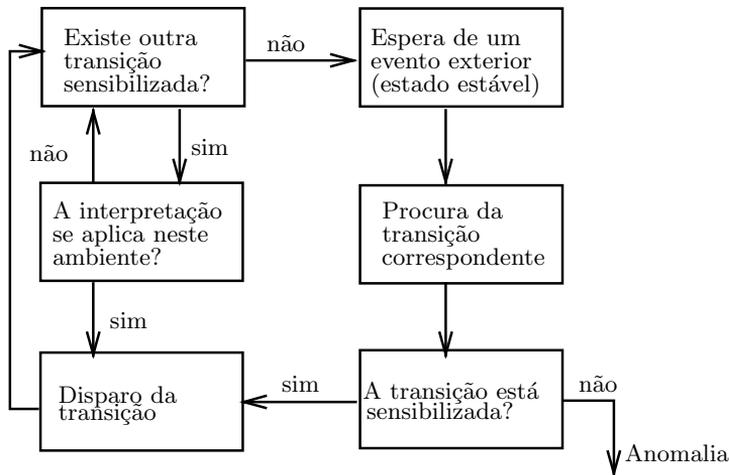


Figura 7.1: Princípio do jogador

O *estado estável* de uma rede é uma marcação para a qual somente as transições associadas a eventos externos estão sensibilizadas. O jogador deve então se colocar num estado de “escuta” do ambiente externo.

Quando uma mensagem é recebida, é necessário procurar a transição que lhe é associada e dispará-la. Em seguida, a partir da nova marcação, é necessário disparar todas as transições internas sensibilizadas a fim de alcançar um novo estado estável. Disparando todas as transições internas, as ações associadas são executadas, o que provoca operações sobre os dados (parte operativa) e envios de mensagens para o ambiente externo do sistema.

7.2.2 Comparação com a abordagem procedimental

Na abordagem não procedimental, tem-se um programa executável que não depende da aplicação ao menos no que concerne ao controle. Claro que resta ainda o tratamento de dados da parte operativa.

A rede é considerada pelo jogador como uma estrutura de dados. Obtém-se assim uma programação automática da parte controle, de onde vem a analogia com a execução de especificações.

A complexidade desta abordagem varia pouco em função do número de lugares e de transições da rede de Petri. Ela depende principalmente do número de transições e de fichas implicadas nos conflitos. As implementações do tipo procedimental apresentam uma complexidade que cresce com o tamanho da rede de Petri. Esta abordagem é preferível nos casos simples. Nos casos complexos e quando as restrições de tempo de resposta não são críticas, é preferível a abordagem não procedimental.

Outro ponto a salientar é que, estando próxima das técnicas de Inteligência Artificial, é possível utilizá-la juntamente com o mecanismo do jogador. Dois casos são interessantes:

- resolução de conflitos;
- diagnóstico em caso de anomalia.

O primeiro caso consiste em ter uma estratégia eficaz para escolher uma transição (regra) e uma n -upla de fichas (fatos) quando do funcionamento em encadeamento para frente (*forward*) que corresponde ao disparo de uma transição interna. Pode-se privilegiar o tempo de resposta e organizar os dados e os procedimentos de procura para que a primeira solução encontrada seja a boa. Ou então, construir todo o espaço de conflitos e fazer apelo a regras de escolha exteriores à rede de Petri (eventualmente a um operador humano) para chegar a uma decisão final.

O segundo caso decorre do seguinte problema. O mecanismo do jogador é um instrumento eficaz para detectar toda incoerência entre a recepção de uma atualização do conhecimento e o último estado conhecido, mas não oferece uma ferramenta de diagnóstico de incidente. Se a rede de Petri é armazenada na forma de uma estrutura de dados sobre a qual opera o jogador, ela pode igualmente ser considerada como um diagrama semântico para um outro sistema de dedução que será, este sim, encarregado do diagnóstico. A rede de Petri é então um modo cômodo de conhecer o estado presumido do conhecimento (a marcação), bem como todas as possibilidades coerentes para fazê-la evoluir (as seqüências de transições).

7.3 Abordagem descentralizada

A concepção do sistema deve ter sido feita utilizando a noção de objeto, ou melhor ainda, de processos seqüenciais como elementos de estruturação. Esta abordagem implica na decomposição da rede de Petri em um certo número de processos que deverão ser implementados sob a forma de processos seqüenciais comunicando-se entre si. Estes processos seqüenciais serão implementados por tarefas executadas em paralelo sobre um processador ou, de modo distribuído, sobre uma rede local.

Os objetos que são processos seqüenciais poderão ser implementados diretamente, enquanto que os outros deverão ser decompostos mais finamente, ou transformados em máquinas de estado finito, ou implementados de modo não procedimental.

As comunicações entre as tarefas são baseadas nos princípios clássicos de:

- comunicação assíncrona;
- *rendez-vous*;
- execução de procedimento distante.

Mas, para a implementação destes mecanismos, novas escolhas são necessárias e, segundo estas escolhas, vão aparecer restrições. Estas escolhas e restrições estão evidentemente muito ligadas à linguagem de implementação escolhida e ao sistema operacional multi-tarefas (ou às primitivas de comunicação oferecidas pela rede local) do sistema informático alvo.

Um dos problemas importantes concerne ao conhecimento nominal mútuo que devem ter ou não as tarefas que se comunicam entre si. Se as tarefas A e B devem comunicar-se ou simplesmente sincronizar-se, existem quatro possibilidades:

1. A conhece B , e B não conhece A ;
2. B conhece A , e A não conhece B ;

3. A e B se conhecem mutuamente;
4. A e B não se conhecem, mas:
 - conhecem um objeto global específico a esta troca;
 - são conhecidas de um objeto global, o *configurador*, tendo o conhecimento da troca e a possibilidade de concorrer à sua realização.

Os casos 1 e 2 são diferentes no caso de uma comunicação assimétrica como a comunicação assíncrona.

Para poder aumentar a autonomia dos objetos e favorecer sua reutilização em outra parte da aplicação ou mesmo em outro projeto, tem-se sempre interesse em que os objetos conheçam nominalmente um número mínimo de outros objetos. A escolha da solução 3 é, portanto, sem interesse. Por outro lado, a solução 4 é interessante mas é, às vezes, difícil de implementar.

Capítulo 8

REDES DE PETRI, LÓGICAS NÃO CLÁSSICAS E SISTEMAS HÍBRIDOS

8.1 Redes de Petri nebulosas

Procedimentos de diagnóstico e recuperação para sistemas a eventos discretos tais como sistemas de manufatura implicam raciocínios em cima de objetos e recursos, e suas mudanças de estado. Além disso, muitas vezes é necessário tratar com dados cujo caráter pode ser vago ou incerto. A lógica clássica não é muito indicada para tais tarefas, principalmente quando se trata de mudanças, pois os problemas de *frame* e ramificação tornam qualquer raciocínio difícil. Por outro lado, as redes de Petri são indicadas para representar estados de sistemas dinâmicos constituídos de processos concorrentes que repartem recursos, mas não é adequada para tratar com a incerteza do sistema, caso exista.

A teoria de redes de Petri nebulosas é recente e a grande maioria dos trabalhos vem sendo publicada em revistas técnicas e conferências.

8.1.1 Requisitos para modelos de sistemas dinâmicos

O desenvolvimento da tecnologia computacional possibilitou uma variedade de métodos, técnicas e ferramentas para resolver problemas associados com o projeto e operação de sistemas dinâmicos. Uma classe importante de tais sistemas são os dirigidos por eventos instantâneos, os sistemas a eventos discretos, chamados de SED.

Existem dois aspectos fundamentais em tais sistemas: a evolução no tempo e a incerteza concernente às informações com as quais eles tratam. Em geral, por um lado, trabalhos sobre a representação do aspecto evolutivo raramente consideram o aspecto incompleto ou de incerteza da descrição do mundo e sua evolução. Por outro lado, o aspecto dinâmico da informação não é, na maioria das vezes, considerado em trabalhos que tratam com o aspecto de incerteza. Estes aspectos são intrinsecamente multidisciplinares, utilizando conceitos de Engenharia de Programação, Análise de Sistemas, Inteligência Artificial, Controle e Base de Dados. Vários modelos foram desenvolvidos, mas nenhum deles é de propósito geral, pois nenhum considera todos os aspectos envolvidos no problema.

Redes de Petri

O aspecto dinâmico de um SED é capturado pela rede de Petri de um modo formal e ao mesmo tempo natural. Como já vimos na primeira parte deste livro, este modelo permite representar o paralelismo verdadeiro, concorrência, restrições de precedência, etc. A análise das assim chamadas boas propriedades (como limitada e viva) e as propriedades estruturais (invariantes de lugar e transição) são uma ajuda eficiente durante as fases de especificação e projeto no ciclo de vida de um SED, como foi visto no capítulo 3. A especificação obtida pode, também, ser simulada e implementada diretamente por um *jogador* de rede de Petri, como visto no capítulo 7.

A rede de Petri ordinária pode somente representar se o sistema está (ou não) num dado estado. Quando o sistema é complexo, e consiste em um conjunto de processos, pode ser necessário usar redes de Petri de alto nível (capítulo 5). Em ambos os casos, somente são consideradas informações conhecidas com precisão.

Se a rede de Petri é um modelo de um sistema físico e é implementada em tempo real de modo concorrente à sua evolução, as transições são somente disparadas quando se tem certeza de que o evento associado de fato ocorreu (capítulos 4 e 7). Se o evento associado à transição acontece e a transição não está sensibilizada pela marcação, é detectada uma inconsistência no modelo. É possível representar o tempo e a comunicação com o meio externo, mas não é possível, entretanto, representar uma informação incompleta, como, por exemplo, “a duração da operação é de *aproximadamente* 5 minutos”.¹

Conjuntos nebulosos

Os conjuntos nebulosos foram introduzidos por Lotfi Zadeh, da Universidade de Berkeley, em 1965. Tais conjuntos permitem tratar com a representação de classes cujos limites não são bem determinados, através de funções características que assumem valores no intervalo $[0, 1]$. Os conjuntos nebulosos são uma ferramenta poderosa para a representação do conhecimento, pois permitem considerar categorias vagas ou flexíveis que são utilizadas por peritos em regras como “*se X é A então Y é B*”, em que *A* e *B* são conjuntos que podem ser nebulosos. Este tipo de regra é usado em sistemas de controle nebuloso e sistemas especialistas nebulosos.

8.1.2 Combinando redes de Petri e conjuntos nebulosos

As diversas abordagens

Vários autores têm demonstrado as relações entre redes de Petri e programas lógicos, como MURATA (1988), e redes de Petri e sistemas de Regras de Produção, como VALLETTE & ATABAKHCHE (1987) e ZISMAN (1978). É natural, portanto, a idéia de introduzir a lógica nebulosa no modelo de rede de Petri. Desde o trabalho pioneiro de LOONEY (1988) vários autores, da comunidade de redes de Petri e de Inteligência Artificial, têm proposto diferentes tipos de redes de Petri nebulosas (RPN). Sob o mesmo nome, estes modelos são baseados em diferentes noções e são mais ou menos consistentes com a teoria de redes de Petri.

¹Uma informação incompleta pode ser expressa pelos termos vago e impreciso. Em geral, *vago* é associado com a dificuldade de realizar distinções rígidas ou precisas; a imprecisão é associada com relações *one-to-many*.

Fichas e marcações

Embora não tenha sido formalmente definido em nenhum artigo, pode-se imaginar que marcações nebulosas poderiam ser definidas associando a cada lugar um número nebuloso indicando o número de fichas no lugar. Entretanto, isto deveria ser consistente com os invariantes de lugar, senão o número de marcações cresceria infinitamente (quando a imprecisão aumenta), resultando num sistema não limitado.

Por exemplo, no modelo de RPN de CAO & SANDERSON (1993), a marcação nebulosa é atribuída a um lugar, para quantificar a incerteza da existência de uma ficha neste. Em CARDOSO (1991), a marcação nebulosa (de uma rede de Petri a objetos) é definida atribuindo a cada ficha sua localização nebulosa, caracterizada por um conjunto nebuloso de lugares (os lugares onde é possível que a ficha esteja). Em outros trabalhos em que uma proposição lógica nebulosa é associada ao lugar, uma função de pertencimento é associada à ficha. O valor da ficha indica o grau de verdade das proposições e muda com o disparo das transições, como nos trabalhos de CHEN (1990) e SCARPELLI (1996).

Transições

Embora no modelo clássico de redes de Petri as fichas sejam removidas no disparo das transições, este não é sempre o caso quando a rede de Petri é usada com técnicas de Inteligência Artificial. Como em lógica uma proposição (antecedente de uma regra) permanece com valor verdadeiro após o disparo da regra, a ficha no lugar correspondente não é removida, como por exemplo em MURATA (1988). Já em outros trabalhos, como CARDOSO (1991), as fichas representam entidades e os lugares são os possíveis estados destas entidades. Se uma ficha f está num lugar p , isto significa que a proposição “ficha f está no estado p ” é verdadeira.

No caso em que as RPNs representam regras de produção nebulosa, as fichas são removidas. Isto não implica que a proposição se torne falsa, mas sim que a informação correspondente foi usada na realização de um cenário de prova.

A maioria dos trabalhos tem um fator de certeza, confiança ou valor verdade associado à transição. Como a transição na RPN corresponde a uma regra num sistema de produção, o disparo de uma seqüência de transições (considerando o fator de confiança de cada uma) define uma seqüência de disparo nebulosa que é mais ou menos possível de ser disparada.

Outra abordagem consiste em associar uma função de autorização à transição: é definido então um *pseudo* disparo quando existe uma informação imprecisa ou incerta. Durante a operação normal, a RPN se comporta como uma rede de Petri clássica. Uma generalização deste trabalho é a associação de uma data nebulosa a cada transição. Esta data é comparada com o tempo corrente para verificar se a transição ainda *não foi* disparada, ou se ela já *foi com certeza*, ou se é *possível* que tinha sido (*possível* no sentido da teoria de Possibilidades).

Considerações sobre a consistência

A rede de Petri é caracterizada por uma dinâmica: é possível *executar* uma rede para analisar o comportamento do sistema modelado. O modo como as funções de pertencimento dos conjuntos nebulosos são atribuídas às fichas (ou lugares) e os fatores

de certeza são associados às transições só será promissor se for consistente com a teoria de redes de Petri.

Um ponto importante na teoria de redes de Petri é que através da programação linear (capítulo 4) é possível extrair sub-redes específicas. Um invariante de lugar é uma sub-rede em que o número total de fichas contidas nos lugares desta é constante. Um invariante de transição é uma sub-rede em que a transformação de marcações obtidas pelo disparo das transições desta sub-rede é nula (a marcação final é igual à inicial). É, pois, importante explorar pelo menos uma destas noções. Assim, a utilização dos invariantes de transição permite encontrar, em redes de Petri que modelam programas lógicos, uma prova *ótima*: prova em que todas as deduções intermediárias são de fato necessárias para obter a conclusão (o que não é sempre verdade de modo geral). No caso de RPN que modela processos físicos, para cada ficha (que é uma instância de objeto numa rede de Petri a objetos) existe um invariante de lugar que a contém com *certeza* (a localização da ficha pode ser imprecisa ou nebulosa, mas está contida no invariante de lugar).

Representar regras por transições numa rede de Petri é uma velha idéia, mas é uma questão que permanece aberta e certamente não é resolvida quando regras nebulosas são consideradas. Isto é uma conseqüência direta do fato de as fichas serem consumidas e produzidas pelo disparo das transições numa rede, enquanto na lógica clássica, devido à propriedade da monotonicidade, uma dedução sempre adiciona novos valores-verdades, sem tornar falsos os que já foram utilizados (veja discussão sobre lógica linear na próxima seção).

Como foi visto no capítulo 2, a gramática $S = \langle IP, Q \rangle$ associada à rede R é definida pelo seu vocabulário IP e pelo conjunto Q de regras de reescrita $t_i : \mu(Pre(., t_i)) \rightarrow \mu(Post(., t_i))$. A uma marcação inicial M da rede corresponde um axioma $\mu(M)$ da gramática, a partir da qual novas palavras são deduzidas.

Usando esta notação, considere, por exemplo, a regra de reescrita $\mu(t) : P \rightarrow Q \wedge R$, cuja rede de Petri está representada na FIG. 8.1.a. Isto é bastante similar à fórmula da lógica proposicional $P \rightarrow Q \wedge R$. Esta fórmula é equivalente a $(P \rightarrow Q) \wedge (P \rightarrow R)$ ou $P \rightarrow Q$ e $P \rightarrow R$. Considerando, agora, regras de reescrita similares a estas duas fórmulas, tem-se $\mu(t_1) : P \rightarrow Q$ e $\mu(t_2) : P \rightarrow R$ cuja rede está representada na FIG. 8.1.b. Na lógica clássica, estas duas fórmulas são equivalentes, porque ambas levam à mesma conclusão: se o fato P é *verdadeiro*, pode-se deduzir que Q e R são verdadeiros. Mas as duas redes de Petri da FIG. 8.1 são diferentes. Na FIG. 8.1.a ambos os lugares Q e R são marcados, enquanto na FIG. 8.1.b apenas um destes lugares será marcado.

De modo geral, a aplicação das RPNs, no caso específico de controle e supervisão de sistemas de manufatura, pode se dar em duas linhas, segundo sua utilização como:

- um modelo para o raciocínio: as transições representam regras de diagnóstico, cada seqüência de disparo de transições representa uma explicação que poderá ser usada para recuperação do erro. A RPN como um todo representa o sistema perito de diagnóstico;
- um modelo do sistema físico: as transições representam as mudanças de estado possíveis; as seqüências de transições representam os comportamentos possíveis. A RPN é uma ferramenta para atualizar o estado do sistema no nível de supervisão, utilizando informação *mal* conhecida (com imprecisão ou incerteza), sem obter estados inconsistentes.

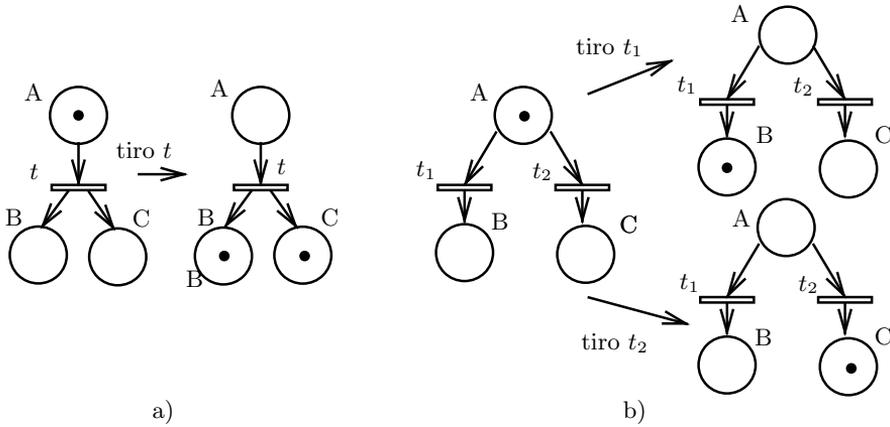


Figura 8.1: a) $P \rightarrow Q \wedge R$, b) $(P \rightarrow Q) \wedge (P \rightarrow R)$

8.2 Redes de Petri como semântica para lógica linear

Como vimos na seção anterior, um problema ao representar a rede de Petri, utilizando a lógica clássica, é que esta não permite trabalhar diretamente com a noção de recurso. Já a lógica linear o permite, e por isto a rede de Petri pode ser utilizada diretamente para dar uma semântica à lógica linear.

Além da noção de recurso, outra motivação para utilizar a lógica linear é a representação mais precisa de uma seqüência de disparos. De fato, um problema na teoria da rede de Petri é que a caracterização da marcação através de um vetor M é *completa e precisa*, o que não é o caso da seqüência s , como foi visto na seção 2.1.8. A existência de um vetor característico s que é solução de $M' = M + C \cdot s$ não garante que a seqüência s possa efetivamente ser disparada. Além de desconsiderar a ordem da seqüência, somente a transformação da marcação é descrita.

8.2.1 Lógica linear: noções de base

A lógica linear, proposta por GIRARD (1987), é de grande interesse quando se deseja manipular recursos consumíveis; esta difere da lógica clássica devido ao fato de que, dentro das regras usuais, duas não são válidas: a regra de contração e a de enfraquecimento (*weakening*). A lógica linear possui três conjuntos de conectivos:

- conectivos multiplicativos:
 - \otimes , chamado *vezes (times)*, expressa AND de recursos,
 - \multimap , chamado *implicação linear*, expressa dependência causal entre recursos,
 - $\bar{}$, chamado *par*, quando usado juntamente com a negação, expressa noções de precedência;
- os conectivos aditivos $\&$, chamado *com (with)*, e \oplus , chamado *mais (plus)*, que permitem expressar escolha interna e externa;

- os conectivos exponenciais ! (*naturalmente/of course*) e ? (*por que não/why not*). ! A indica que o recurso A pode ser *usado* tanto quanto necessário (é ilimitado) e ? A indica que o recurso A pode ser *produzido* tanto quanto necessário. Estes conectivos permitem reintroduzir as noções de contração e enfraquecimento, se necessário, mas para fórmulas específicas e de uma forma controlada.

A lógica linear também manipula a negação, mas com uma interpretação diferente daquela da lógica clássica: a negação linear é denotada \perp e não expressa a verdade/falsidade, mas conceitos de ação/reação, como, por exemplo, produção e consumo.

Assim, em lógica linear, se A e B são recursos, $A \otimes B$ representa um recurso consistindo de A e B simultaneamente, e $A \multimap B$ representa a ação de usar A de maneira a produzir B (dependência causal).

Se se dispõe do recurso A , pode-se obter o recurso B através do cálculo de seqüente entre as fórmulas A e $A \multimap B$. Diferentemente da lógica clássica, a partir deste momento não se dispõe mais do recurso A ; a dedução representa o consumo deste recurso e a produção de B .

Portanto, fórmulas bem formadas descrevendo o estado dos recursos são consumidas quando utilizadas na prova de um seqüente (quando um recurso ocioso é alocado a uma tarefa, ele se torna ocupado).

8.2.2 Descrição da rede de Petri usando lógica linear

Qualquer rede de Petri pode ser descrita por uma coleção de fórmulas bem-formadas da lógica linear: marcações são descritas por fórmulas consumíveis e transições por fórmulas não consumíveis (utilizando o conectivo !). Uma fórmula do tipo $A \otimes D$ descreve a distribuição de fichas (marcação) tal que os lugares A e D contêm, cada um, uma ficha. Esta marcação pode ser uma marcação parcial. Esta fórmula é correta, porque as fichas podem ser utilizadas apenas uma vez.

Por sua vez, uma transição t com $Pre(t) = \{A\}$ e $Post(t) = \{B, C\}$ é representada pela fórmula

$$!(A \multimap B \otimes C).$$

Como as transições não são consumidas pelo disparo (se uma nova ficha chega no lugar de entrada, ela pode ser imediatamente disparada), é necessário usar o conectivo !. Entretanto, numa prova o conectivo ! não aparece pois a transição é disparada uma só vez para uma dada marcação.

Uma coleção de fórmulas da lógica linear correspondendo a uma descrição de rede de Petri com uma marcação inicial será da forma $(A_k$ descreve uma ficha no lugar A_k , e $m_k.A_k$ indica que existe m_k fichas em A_k):

- para a marcação inicial

$$\left(\bigotimes_{A_k \in \text{conj. de lugares marcados}} m_k.A_k \right); \tag{8.1}$$

- para cada transição t_j da rede:

$$!\left(\left(\bigotimes_{\substack{A_i \in \text{lug. ent.} \\ \text{de } t_j}} m_i.A_i \right) \multimap \left(\bigotimes_{\substack{A_o \in \text{lug. saída} \\ \text{de } t_j}} m_o.A_o \right) \right). \tag{8.2}$$

Tiro de transição

Uma transição t de uma rede de Petri está sensibilizada se $M > Pre(\cdot, t)$, sendo M o vetor marcação e $Pre(\cdot, t)$, a função entrada. O tiro consiste em retirar fichas dos lugares de entrada e colocá-las nos lugares de saída usando a equação

$$M' = M + C.t$$

sendo $C = Post - Pre$ a matriz de incidência.

O tiro de uma transição corresponde em lógica linear a provar o seqüente:

$$\frac{M, t}{M'}$$

em que a fórmula M é a marcação inicial na forma da equação 8.1 e a fórmula t é a transição na forma da equação 8.2. Pode-se também escrever este seqüente na forma: $M, t \vdash M'$.

O tiro só é possível se todos os átomos do lado esquerdo da fórmula t estiverem presentes na fórmula M . No caso da transição t tratada acima, se a marcação da rede é A , tem-se, então,

$$\frac{A, (A \multimap B \otimes C)}{B \otimes C}. \quad (8.3)$$

O recurso A foi consumido, e foi produzido o recurso $B \otimes C$. Em termos de redes de Petri, corresponde a tirar as fichas dos lugares de entrada de t e colocá-las nos lugares de saída de t .

8.2.3 Seqüência de tiro na lógica linear

Nesta seção será apresentada a maneira pela qual uma seqüência de tiro pode ser caracterizada por uma fórmula bem formada da lógica linear. A seguir é feita a comparação com a seqüência obtida pelo cálculo de transformação de marcação usado na teoria de rede de Petri. A caracterização de uma seqüência s em rede de Petri é Cs , usada na equação fundamental

$$M' = M + Cs. \quad (8.4)$$

Cada componente $s(t)$ de \mathbf{s} indica o número de vezes que a transição t dispara na seqüência s . O vetor \mathbf{s} não indica a ordem de disparo das transições, e mesmo uma solução $\mathbf{s} > 0$ não implica a existência de tal seqüência (condição necessária, mas não suficiente).

Seqüências ordenadas

Se a seqüência $t_1 t_2$ é uma seqüência ordenada, então t_1 deve disparar antes de t_2 . Todos as fichas produzidas por t_1 e que podem ser consumidas por t_2 devem de fato ser consumidas. Isto significa eliminar ao máximo o conjunto de proposições que são membros da parte direita da fórmula que representa a primeira transição da seqüência, e da parte esquerda da fórmula que representa a segunda transição da seqüência. Corresponde, em rede de Petri, à definição de tiro: todas as fichas presentes nos lugares de

entrada de t que a sensibilizam devem ser retiradas e a condição de sensibilização é a marcação mínima que permite o disparo. Assim procedendo, tem-se ao mesmo tempo a transformação de marcação e a mínima marcação necessária para disparar a seqüência.

Considere a rede de Petri da FIG. 8.2. Sua estrutura (a marcação inicial não será usada aqui) corresponde ao seguinte conjunto de fórmulas da lógica linear:

$$\begin{aligned} t_1 & : !(A \multimap B) \\ t_2 & : !(B \otimes \alpha \multimap C \otimes \beta) \\ t_3 & : !(C \otimes \beta \multimap D \otimes \alpha). \end{aligned}$$

Considere, por exemplo, a seqüência $s_1 = t_1 t_2 t_3$. Calcula-se o seqüente entre as fórmulas de t_1 e t_2 , e deste com t_3 :

$$\frac{\frac{(A \multimap B), (B \otimes \alpha \multimap C \otimes \beta)}{(A \otimes \alpha \multimap C \otimes \beta)}, (C \otimes \beta \multimap D \otimes \alpha)}{(A \otimes \alpha \multimap D \otimes \alpha)}.$$

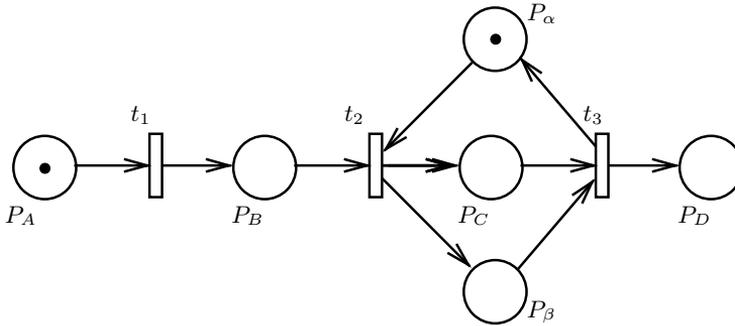


Figura 8.2: Recursos e seqüência

A fórmula

$$s_1 : !(A \otimes \alpha \multimap D \otimes \alpha) \tag{8.5}$$

representa a seqüência $s_1 = t_1 t_2 t_3$.

Considere agora a seqüência $s_2 = t_3 t_1 t_2$. A representação desta seqüência é dada por:

$$\frac{\frac{(C \otimes \beta \multimap D \otimes \alpha), (A \multimap B)}{(C \otimes \beta \otimes A \multimap D \otimes \alpha \otimes B)}, (B \otimes \alpha \multimap C \otimes \beta)}{(C \otimes \beta \otimes A \multimap D \otimes C \otimes \beta)}.$$

A fórmula

$$s_2 : !(C \otimes \beta \otimes A \multimap C \otimes \beta \otimes D) \tag{8.6}$$

representa a seqüência $s_2 = t_3 t_1 t_2$.

Recursos

O primeiro ponto a observar, considerando seqüência s_1 , é que o átomo α permanece presente nos dois lados da implicação linear e não pode ser simplificado.

Entretanto, o átomo β não aparece porque, no disparo da seqüência s , o recurso α é consumido antes de ser produzido, enquanto β é produzido antes de ser consumido. A fórmula 8.5 mostra tanto a transformação da marcação (uma ficha que se desloca de P_A a P_D) quanto a marcação mínima necessária para possibilitar o tiro da seqüência (marcação $A \otimes \alpha$ neste exemplo). Quanto à seqüência s_2 (fórmula 8.6), a marcação mínima necessária é $(C \otimes \beta \otimes A)$. Portanto, para a marcação indicada na FIG. 8.2, o disparo desta seqüência não é possível.

Assim, tem-se ao mesmo tempo a transformação da marcação e a marcação mínima necessária para disparar a seqüência, enquanto C_s indica somente a transformação da marcação provocada pela seqüência s e não garante nem mesmo a sua existência (o componente relativo ao lugar α é nulo).

A caracterização de uma seqüência na lógica linear depende da ordem de tiro. Assim, as seqüências $s_1 = t_1 t_2 t_3$ e $s_2 = t_3 t_1 t_2$ não são representadas pela mesma fórmula. A primeira é representada por (8.5) e a segunda é representada por (8.6). Na teoria de redes de Petri, ambas são representadas pelo mesmo vetor característico $\mathbf{s} = [1 \ 1 \ 1]^T$, que, aplicado na equação 8.4 com a marcação da figura 8.2 (lugares P_A e P_α marcados), fornece a mesma marcação (lugares P_D e P_α marcados). Entretanto, é fácil verificar que a seqüência s_1 pode ser disparada para esta marcação inicial, enquanto a seqüência s_2 não pode ser disparada.

8.3 Redes de Petri para sistemas híbridos

8.3.1 Sistema de produção híbrido

De forma geral, o modelo de um sistema dinâmico é híbrido se as variáveis de estado forem de dois tipos: contínuas (valores reais) e discretas (valores de conjunto finito, por exemplo, dos inteiros). Como foi visto no capítulo 1, considerando a natureza das variáveis de estado e o tempo, tem-se quatro tipos de modelos:

- sistema contínuo: tempo e variáveis de estado contínuos;
- sistema discretizado: tempo discreto e variáveis de estado contínuas;
- sistema discreto: tempo contínuo e variáveis de estado discretas;
- sistema a eventos discretos: tempo e variáveis de estado discretos.

No caso de sistemas híbridos, as variáveis de estado, assim como o tempo, podem ser dos dois tipos (discreto e contínuo). Assim, o tempo é também híbrido, o que implica utilizar dois tipos de modelos: um modelo contínuo e um modelo a eventos discretos.

Processo de fabricação híbrida

Modelos híbridos são frequentemente necessários nos sistemas de produção do tipo batelada. Por exemplo, num sistema de manufatura, no nível do controle local, os

sistemas de regulação de motores (comando dos eixos) impõem modelos contínuos (ou discretizados se o controle é numérico). Mas o encadeamento das operações e a pilotagem do sistema é baseado num modelo a eventos discretos. Entretanto, estas duas visões não devem ser combinadas num sistema híbrido, pois raramente estes seriam ligados por relações de dependência.

No caso de um processo industrial que trate um fluxo de matéria-prima contínuo funcionando em regime permanente como uma refinaria de petróleo, o modelo matemático mais simples é o contínuo formado de equações algébricas. Mas se existe um interesse nos lançamentos e paradas da instalação, é necessário introduzir equações diferenciais para representar o comportamento dinâmico. Pode também ser necessário introduzir variáveis discretas para descrever as diferentes fases do regime transitório e os controles associados (abertura e fechamento de válvulas, aquecimento, etc.). No caso de modelar o comportamento quando da abertura de uma válvula, com vistas à análise da estabilidade, é necessário um modelo híbrido com tempo contínuo.

Os sistemas híbridos de produção, chamados *batch systems* em inglês, correspondem a um outro caso. Trata-se de processos industriais que transformam matéria-prima contínua e que são formados por dois tipos de equipamentos:

- um primeiro tipo, por exemplo os trocadores de calor utilizados para esquentar ou resfriar a matéria-prima, funciona de modo contínuo. Para que haja eficácia e que a temperatura da matéria-prima seja correta, o fluxo que a atravessa deve ser constante em regime permanente. Durante um intervalo de tempo dt , um volume dV de matéria sofre completamente a transformação físico-química. Por exemplo, sua temperatura passa de θ_0 a θ_1 ;
- um segundo tipo de equipamento, por exemplo os reatores, funciona de modo descontínuo. Após uma fase de encher, todo um lote (*batch*) de matéria sofre a transformação. Se um reator é utilizado para esquentar um lote de volume V , durante o intervalo de tempo dt todo o volume V passa de uma temperatura θ a $\theta + d\theta$.

Sistemas do tipo batelada (ou por lotes) impõem, em geral, a utilização de um modelo híbrido. A receita de fabricação que descreve o encadeamento das transformações a efetuar sobre os equipamentos é naturalmente um modelo a eventos discretos (como um processo de fabricação numa indústria manufatureira), mas a unidade *matéria transformada* não é um lote de peças, mas uma quantidade de matéria-prima que pode variar segundo o tamanho dos reatores, e que não é fisicamente delimitada nos equipamentos funcionando de modo contínuo. A todo momento deve-se verificar se os equipamentos obrigados a funcionar em regime permanente estão corretamente aprovisionados (matéria-prima na entrada e lugar disponível na saída), o que impõe um balanço de matéria (equações implicando variáveis contínuas). Portanto, é difícil evitar a manipulação simultânea de estados discretos e de eventos para descrever o encadeamento de fases e de variáveis contínuas tomando seus valores no conjunto dos reais para descrever os volumes, massas molares, composições químicas, etc.

8.3.2 Técnicas de modelagem

O problema apresentado pelos sistemas híbridos vem de que, do ponto de vista matemático, os cálculos sobre os inteiros (e portanto, as variáveis discretas) são muito diferentes (em geral mais complexos) que os cálculos sobre os reais. Não existem derivadas, e muitas vezes, existem apenas procedimentos enumerativos custosos.

As abordagens desenvolvidas podem ser agrupadas em quatro classes: modelos contínuos, nos quais se adicionam variáveis discretas; modelos discretos, nos quais se adicionam variáveis contínuas; modelos integrando o contínuo e o discreto num único formalismo ou nos dois formalismos comunicantes.

Estas abordagens serão ilustradas em um exemplo de um reator que deve ser enchido e esvaziado (FIG. 8.3).

Modelo contínuo estendido

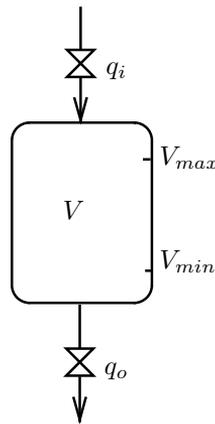


Figura 8.3: Reator

No caso do reator da FIG. 8.3, o modelo contínuo é uma simples equação diferencial. O volume V (única variável de estado) contido no reator é calculado a partir do débito de entrada q_i e do débito de saída q_o (consideradas aqui como sendo parâmetros) por:

$$\frac{dV}{dt} = q_i - q_o. \quad (8.7)$$

Se o modelo é utilizado em tempo real sobre um processador para calcular o volume num dado instante, então os instantes t_n de discretização devem ser considerados, e obtém-se (supõe-se que entre dois instantes de amostragem a variação de q_i e q_o é desprezível):

$$V(t_{n+1}) - V(t_n) = (q_i - q_o)(t_{n+1} - t_n). \quad (8.8)$$

Há duas válvulas (de entrada e de saída) no sistema da FIG. 8.3; como cada uma pode estar aberta ou fechada, existem quatro configurações de funcionamento neste sistema. A equação 8.7 descreve o comportamento do sistema quando ambas as válvulas estão abertas. Para descrever as quatro configurações numa única equação, a equação 8.7 é acrescida de duas variáveis booleanas b_i e b_o , obtendo-se assim a equação:

$$\frac{dV}{dt} = b_i \cdot q_i - b_o \cdot q_o. \quad (8.9)$$

A variável booleana b_i corresponde ao predicado *válvula de entrada aberta* (vale 1 quando a válvula de entrada está aberta e 0 quando está fechada) e a variável booleana b_o , ao predicado *válvula de saída aberta* (vale 1 quando aberta e 0 quando está fechada).

A equação 8.9 é válida para as quatro fases possíveis do sistema permitindo, então, simular o comportamento geral do sistema (parte contínua e parte discreta, num mesmo modelo):

- o reator pode estar isolado: $b_i = 0$ e $b_o = 0$;
- o reator está sendo enchido: $b_i = 1$ e $b_o = 0$;
- o reator está sendo esvaziado: $b_i = 0$ e $b_o = 1$;
- o reator está sendo enchido e esvaziado (assume o papel de um equipamento de armazenamento para desacoplar dois equipamentos funcionando continuamente): $b_i = 1$ et $b_o = 1$.

Existem três variáveis de estado (V , b_i e b_o) e dois parâmetros (q_i e q_o). A variável V descreve a parte contínua do estado, enquanto as duas outras correspondem à parte discreta do estado. Elas realizam uma codificação das quatro fases e fazem com que a equação 8.9 seja válida qualquer que seja o estado do reator. Entretanto, esta equação não descreve os eventos que provocam a passagem de uma fase a outra. É necessário adicionar equações que permitem calcular as variáveis booleanas em função dos limiares da variável contínua V :

$$\begin{array}{ll} \text{Se } (V < V_{max}) \text{ então } b_i = 1 & \text{Se } (V \geq V_{max}) \text{ então } b_i = 0 \\ \text{Se } (V > V_{min}) \text{ então } b_o = 1 & \text{Se } (V \leq V_{min}) \text{ então } b_o = 0. \end{array}$$

Esta solução só é possível se o sistema possuir poucas mudanças de fase. Além disto, um primeiro problema aparece quando o sentido da variação da grandeza contínua influi no cálculo das variáveis booleanas, como por exemplo,

$$\begin{array}{l} \text{Se } (V = V_{alto}) \text{ quando } V \text{ é crescente então } b_i = 0 \\ \text{Se } (V = V_{medio}) \text{ quando } V \text{ é crescente então } b_i = 1. \end{array}$$

Um segundo problema surge quando existem várias fases *em processo de enchimento* numa receita de fabricação. Por exemplo, quando for necessário colocar inicialmente 1000 litros do produto A no reator, e depois colocar 500 litros do produto B . As variáveis booleanas e os limiares crescem rapidamente.

Modelo a eventos discretos estendido

Os modelos a eventos discretos (autômatos finitos ou redes de Petri) são baseados em estados discretos e em tempo discreto. O tempo é explicitado sob a forma de uma seqüência de eventos que são as mudanças de estado (função *estado seguinte* em um autômato, ou disparo de transição na rede de Petri).

Um modelo a eventos discretos é particularmente bem adaptado para descrever a sucessão de fases de um sistema híbrido necessárias à fabricação de um produto (*receita*). No caso do reator, uma receita elementar pode ser descrita pela rede de Petri da FIG. 8.4.

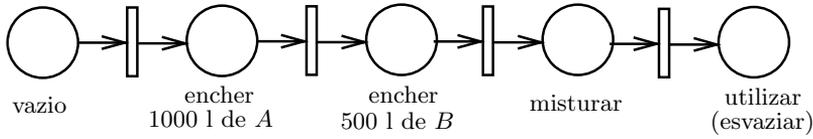


Figura 8.4: Receita de fabricação

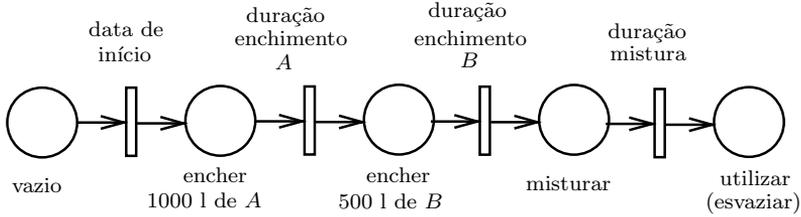


Figura 8.5: Explicitação do tempo

A partir do reator vazio, são colocados 1000 litros do produto *A* e depois são colocados 500 litros de *B*. Estes são, então, misturados, e finalmente o produto é utilizado. Cada fase é um lugar da rede.

Introduzir o tempo no modelo da FIG. 8.4 significa passar de um modelo a eventos discretos a um modelo discreto (estados discretos, tempo contínuo). No caso da rede de Petri temporal, uma duração de sensibilização é associada à transição.

O fato de associar uma duração de enchimento do reator com o produto *A* à transição de saída do lugar *encher 1000 l de A* obriga as fichas circulando na rede a ficarem no lugar durante o tempo especificado (rede da FIG. 8.5). Em casos simples, esta duração pode ser calculada como $\frac{1000}{q_{i_A}}$ onde q_{i_A} é o débito de entrada. O valor de *V* pode ser calculado a qualquer momento; assim, por simples interpolação o estado híbrido do reator (fase e *V*) pode ser reconstituído a cada instante (tempo contínuo).

O modelo híbrido é então formado de um modelo discreto (rede de Petri) completado com uma variável contínua: o tempo. Todas as outras variáveis de estado contínuas devem ser funções lineares do tempo, calculáveis a todo instante a partir do estado discreto.

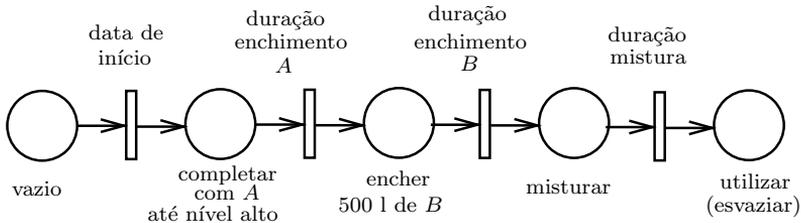


Figura 8.6: Duração não calculável *a priori*

Mas a modelagem de sistemas híbridos por redes de Petri temporais possui certas limitações. Considere por exemplo a FIG. 8.6. Se o nível inicial V_0 do produto contido no reator não é sempre o mesmo no início da execução da receita, e se a operação a ser

executada é *completar com o produto A até o nível V_{alto}* , então a duração da operação é

$$\frac{V_{alto} - V_0}{q_{i_A}}. \quad (8.10)$$

Esta data depende do valor do componente contínuo V de estado quando do início da fase correspondente ao lugar. Assim, a informação contida no modelo não é mais suficiente para simular o comportamento do sistema.

Este problema pode ser resolvido utilizando-se uma rede de Petri de alto nível (capítulo 6), em que um certo número de atributos pode ser atribuído às fichas. Pode-se, assim, associar variáveis de estado contínuas a certas fichas. Por exemplo, a variável V pode ser associada à ficha que circula de lugar em lugar sob a forma de um atributo. No início da operação *encher com 1000 l do produto A* este atributo conterà o valor V_0 , o que permite calcular a duração usando a fórmula 8.10. O modelo híbrido é então obtido representando as fases por lugares (variáveis de estado discretas) e as variáveis de estado contínuas por atributos associados às fichas.

8.4 Notas

Livros básicos sobre a teoria dos conjuntos nebulosos e teoria de possibilidades: DU-BOIS & PRADE (1988), KLIR (1988) e BOUCHON-MEUNIER (1994), além dos artigos ZADEH (1977,1978).

A lógica linear é uma teoria bem recente desenvolvida por GIRARD (1987,1990). Para detalhes sobre cálculo de seqüentes ver FITTING (1990). Maiores detalhes sobre redes de Petri e lógica linear ver PRADIN (1993).

O modelamento de sistemas híbridos usando um único modelo é também recente. Alguns trabalhos usando redes de Petri: DAUBAS (1994), ANDREU (1994,1995).

Apêndice A

GRAFOS

A teoria de grafos é um tema antigo com aplicações modernas. O primeiro resultado foi dado por Euler em 1736 com a solução do problema da cidade de Koenigsberg: como atravessar as sete pontes da cidade passando apenas uma única vez por cada uma.

Até 1946 a teoria de grafos se restringe à Matemática; as contribuições mais importantes foram as dos matemáticos Kirchoff, Hamilton, Sylvester, Kempe, Lucas, Petersen e Tarry no século XIX e Poincaré, Sainte-Lagu, Kuratowski, Hall, Polya, Köning, Whitney, Tutte na primeira metade do século XX.

A *pesquisa operacional*, surgida das pesquisas militares ligadas à Segunda Guerra, provocou um desenvolvimento da teoria de grafos como modelo de sistemas concretos. Entre os grandes especialistas desta nova orientação pode-se citar Kühn, Dantzig, Roy, Faure, Kaufmann. Um novo salto foi dado nos anos 60 com o desenvolvimento das Ciências da Informação e da Comunicação, com contribuições de informáticos como Dijkstra, Knuth, Wirth, Sakharovitch, Gondran e outros (tanto no desenvolvimento de algoritmos quanto na modelagem de problemas).

No estudo de grafos, é importante conhecer os algoritmos adaptados aos principais tipos de problemas de grafos: procura do caminho de menor custo, passar uma única vez por todos os arcos do grafo, etc.

A.1 Definições formais e notação

Definição A.1.1 (Grafo) *Um grafo ou grafo não ordenado $G = (V, E)$ consiste em um conjunto V de vértices (ou nós) e um conjunto E de arcos tal que cada arco $e \in E$ é associado com um par não ordenado de nós.*

Definição A.1.2 (Grafo direcionado) *Um grafo direcionado ou ordenado $G = (V, E)$ consiste em um conjunto V de vértices (ou nós) e um conjunto E de arcos tal que cada arco $e \in E$ é associado com um par ordenado de nós.*

Definição A.1.3 (Arco incidente) *Um arco (direcionado ou não) associado com um par de nós v e w é dito incidente em v e w , e os nós são ditos adjacentes.*

Definição A.1.4 (Arcos Paralelos) *Os arcos são ditos paralelos se estão associados com os mesmos pares de vértices.*

Definição A.1.5 (Malha) *Uma malha é um arco incidindo num único vértice.*

Definição A.1.6 (Vértice isolado) *Considera-se um vértice isolado quando não existe nenhum arco incidente.*

Definição A.1.7 (Vértice pendente) *O vértice é dito pendente quando existe apenas um arco incidente.*

Definição A.1.8 (Grafo simples) *Um grafo é simples quando se apresenta sem malhas nem nós paralelos.*

Definição A.1.9 (Grau) *O grau de um nó é o número de arcos incidindo no nó. No caso de uma malha, contam-se dois.*

Definição A.1.10 (Aplicação Multívoca Γ) *A aplicação multívoca Γ é definida por $\Gamma : V \rightarrow \mathcal{P}(V)$, onde Γ_i é o conjunto de sucessores do nó i , e Γ_i^{-1} é o conjunto de predecessores do nó i .*

A.2 Conectividade

Caminhos e circuitos

Caminhos e circuitos são definidos num grafo *orientado*.

Definição A.2.1 (Caminho) *Considere v_o e v_n nós em um grafo. Um caminho de v_o a v_n de comprimento n é uma seqüência alternada de $n+1$ nós e n arcos começando em v_o e terminando em v_n .*

Definição A.2.2 (Caminho elementar) *Um caminho elementar de v a w é um caminho que passa uma única vez por cada nó.*

Definição A.2.3 (Circuito) *Um circuito é um caminho de comprimento não-nulo de v a v que passa uma única vez por cada arco.*

Definição A.2.4 (Circuito elementar) *Um circuito elementar é um ciclo de v a v passando uma única vez por cada nó (exceto $v_o = v_n = v$).*

Cadeias e ciclos

Cadeias e ciclos são definidos para grafos não orientados; cadeias correspondem aos caminhos e ciclos correspondem aos circuitos para grafos orientados nas definições acima.

Definição A.2.5 (Fechamento transitivo) *O fechamento transitivo da aplicação Γ é a aplicação $\hat{\Gamma}$ definida por*

$$\hat{\Gamma}_i = i \cup \Gamma_i \cup \Gamma_i^2 \dots \cup \Gamma_i^{n-1}$$

Γ_i^k é o conjunto de nós que se pode atingir a partir do nó i por caminhos tendo exatamente k arcos. $\hat{\Gamma}_i$ é o conjunto de sucessores de i . Do mesmo modo, Γ_i^{-1} representa o conjunto de predecessores de i .

Definição A.2.6 (Grafo conexo) Um grafo é conexo se para qualquer par de nós v e w de G existe uma cadeia de v a w .

Por exemplo, é possível construir um grafo representando os estados brasileiros, com um arco ligando os estados que possuem uma fronteira comum.

Um grafo conexo é único, enquanto que um grafo não conexo possui vários *pedaços*, chamados *subgrafos* ou *componentes*.

Definição A.2.7 (Subgrafo) $G' = (V', E')$ é um subgrafo de $G = (V, E)$ se: a) $V' \subseteq V$ e $E' \subseteq E$; b) $\forall e' \in E'$, se e' incide em v' e w' , então v' e $w' \in V'$ (a restrição b) garante que G' é um grafo).

Definição A.2.8 (Componente) Dado um grafo G e um nó v de G , o subgrafo G' de G formado por todos os arcos e nós de G contidos nas cadeias começando em v é chamado componente de G contendo v .

Um grafo é conexo se tiver um só componente. O grafo da FIG. A.1.a (JOHNSON-BAUGH (1993)) é conexo, pois dado qualquer par de vértices v e w , existe um caminho de v a w . Já o da FIG. A.1.b não é conexo; por exemplo, não há um caminho de v_2 a v_5 .

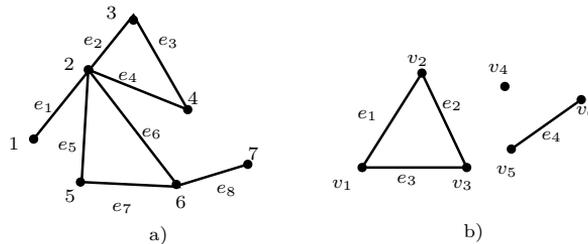


Figura A.1: a) Grafo conexo; b) Grafo não conexo

Definição A.2.9 (Grafo fortemente conexo) Um grafo direcionado é fortemente conexo se para qualquer par de nós v e w de G existe um caminho de v a w e um caminho de w a v , $\forall w, v \in V$.

Definição A.2.10 (Grafo fracamente conexo) Um grafo direcionado é fracamente conexo se para qualquer par de nós v e w do grafo subjacente existe uma cadeia de v a w .

Na FIG. A.2, o grafo G é fortemente conexo (existe um caminho entre quaisquer pares de vértices). G é também fracamente conexo. Já o grafo H não é fortemente conexo, pois não existe caminho entre a e b ; entretanto, é fracamente conexo, pois existe uma cadeia entre quaisquer pares de vértices no grafo subjacente.

Definição A.2.11 (Ciclo de Euler) Um ciclo em G que inclui todos os arcos e todos os nós é um ciclo de Euler.

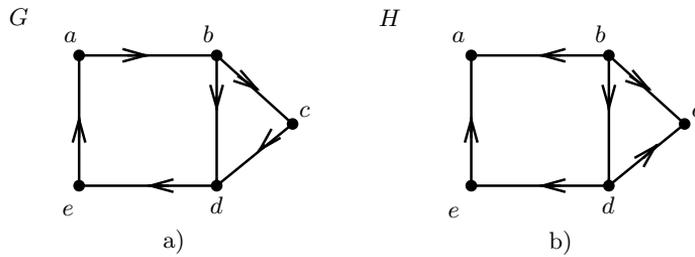


Figura A.2: Grafo a) fortemente conexo; b) fracamente conexo

Se G consiste em um nó v , sem arcos, o caminho (v) é chamado um ciclo euleriano em G .

Teorema A.2.1 *Se um grafo G possui um ciclo de Euler, então G é conexo e todos os nós têm grau par.*

Se o nó possui número ímpar de vértices incidentes, é impossível *sair* do nó sem passar pelo arco duas vezes.

Teorema A.2.2 *Se G é um grafo conexo e todos os nós têm grau par, então G tem um ciclo euleriano.*

Teorema A.2.3 (Aperto de mão) *Se G é um grafo não direcionado com m arcos e n vértices, então a soma dos graus de todos os nós, $\sum_{i=1}^n \delta(v_i) = 2m$, é par.*

Corolário A.2.1 *Em qualquer grafo, existe um número par de vértices com grau ímpar.*

Teorema A.2.4 *Um grafo G possui um caminho sem arcos repetidos de v a w ($v \neq w$) contendo todos os arcos e nós se e somente se é conexo e v e w são os únicos nós com grau ímpar.*

Teorema A.2.5 *Se G contém um ciclo de v a v , G contém um ciclo elementar de v a v .*

Algoritmo de construção de ciclos de Euler

procedimento *Euler* (G : grafo conexo, $\forall v \in V, \text{grau}(v) = \text{par}$)

ciclo := ciclo em G começando num nó qualquer

H := G com os arcos deste ciclo removidos

enquanto H tem arcos

início

subciclo := ciclo em H começando num nó em H que também é um nó de *ciclo*

H := H com arcos de *subciclo* e nós isolados removidos

ciclo := *ciclo* com *subciclo* inserido no nó apropriado

fim {*ciclo* é um ciclo de Euler}.

A.3 Notas

Para maiores detalhes sobre grafos ver JOHNSONBAUGH (1993), GONDRAM & MINOUX (1985) e ROSEN (1991).

Apêndice B

CÁLCULO DOS COMPONENTES

B.1 Princípio do cálculo de uma base

O método será explicado para a procura dos componentes conservativos. Basta apenas calcular a transposta da matriz C (inverter lugares e transições) para encontrar os componentes repetitivos estacionários.

Procuram-se as soluções da equação:

$$f^T C = 0$$

que também pode ser escrita como

$$C^T f = 0.$$

O número de variáveis desta equação corresponde ao número de lugares da rede (n componentes de f ou n linhas de C) e o número de equações corresponde ao número de transições (m colunas de C).

Esta equação matricial é equivalente ao sistema de equações:

$$\left. \begin{array}{l} c_{11}f_1 + \dots + c_{n1}f_n = 0 \\ \dots + \dots + \dots = 0 \\ c_{1m}f_1 + \dots + c_{nm}f_n = 0 \end{array} \right\} m \text{ equações.}$$

Trata-se de um sistema de equações lineares cujo segundo membro é nulo. O conjunto de soluções forma, portanto, um espaço vetorial. A solução degenerada $f^T = 0$ existe, mas não apresenta nenhum interesse. O que interessa é procurar uma base do espaço de soluções.

Como será visto em seguida, se r é o posto (*rank*) de C e n o número de lugares, a dimensão deste espaço vetorial será:

$$\dim_p = n - r$$

e a dimensão dos componentes repetitivos estacionários

$$\dim_t = m - r$$

onde m é o número de transições.

Existe um método sistemático de resolução de equações lineares, chamado *método de Gauss*. Segundo esse método, as soluções de um sistema de equações lineares não são modificadas se as seguintes operações forem efetuadas nas linhas (ou colunas):

Este sistema possui uma única solução em relação às r variáveis f'_1, \dots, f'_r que é a solução degenerada. Assim, as soluções na forma f' são tais que os r primeiros componentes deste vetor são sempre iguais a zero. Por outro lado, os outros $\dim_p = n - r$ componentes podem ser escolhidos livremente, pois foram eliminados do sistema de equações (as últimas $n - r$ linhas de C' são nulas).

A obtenção de uma base do espaço vetorial das soluções, sob esta forma, é imediata. Basta apenas escolher as soluções para as quais uma e somente uma das componentes de f' é não nula:

$$\begin{bmatrix} (0) \\ \hline f'_{r+1} = 1 \\ 0 \\ \dots \\ 0 \end{bmatrix}, \quad \begin{bmatrix} (0) \\ \hline 0 \\ f'_{r+2} = 1 \\ \dots \\ 0 \end{bmatrix}, \quad \dots \quad \begin{bmatrix} (0) \\ \hline 0 \\ 0 \\ \dots \\ f'_n = 1 \end{bmatrix},$$

motivo pelo qual a dimensão do espaço vetorial de soluções é $n - r$.

Para poder escrever esta base na forma inicial da equação

$$f^T C = 0$$

basta saber construir a matriz F . Obtém-se assim uma base de componentes conservativos.

Como já foi dito, o método de triangularização de Gauss consiste em fazer aparecer zeros, combinando as linhas da matriz. Considere o caso em que as $i - 1$ primeiras linhas e colunas tenham sido triangularizadas. Considere a linha i (uma variável) e a coluna i (uma equação). Os coeficientes desta coluna que correspondem às variáveis $i + 1$ a n são anulados multiplicando a linha i por c_{ki} , a linha k por c_{ii} e eliminando a linha i da linha k .

Se o coeficiente c_{ii} é nulo, tal operação é impossível. Neste caso, devem ser realizadas permutações entre as últimas $m - i$ colunas de C (que estão sendo transformadas) de modo a fazer aparecer um c_{ii} não nulo. Se isto é impossível, significa que a linha i possui apenas zeros pois as primeiras $i - 1$ colunas de C já foram tratadas. A variável f_i já foi eliminada do sistema de equações e produz um elemento da base do espaço vetorial das soluções. As últimas $n - i$ linhas de C são permutadas de modo a produzir uma linha que não contenha apenas zeros. Se isto é impossível, o algoritmo termina; C foi escrita sob a forma C' .

Substituir a “linha k ” por “ c_{ii} vezes a linha k menos c_{ki} vezes a linha i ” consiste em fazer a seguinte mudança de variável (permitido para $c_{ii} \neq 0$):

$$\begin{aligned} f_i &= f'_i - c_{ki} f'_k \\ f_k &= c_{ii} f'_k \\ f_j &= f'_j \quad \forall j \neq i, k. \end{aligned}$$

Assim a equação j (coluna j de C), por exemplo, escrita como:

$$c_{1j} f_1 + \dots + c_{ij} f_i + \dots + c_{kj} f_k + \dots + c_{nj} f_n = 0$$

torna-se:

$$c_{1j}f'_1 + \dots + c_{ij}(f'_i - c_{ki}f'_k) + \dots + c_{kj}c_{ii}f'_k + \dots + c_{nj}f'_n = 0$$

ou:

$$c_{1j}f'_1 + \dots + c_{ij}f'_i + \dots + (c_{ii}c_{kj} - c_{ki}c_{ij})f'_k + \dots + c_{nj}f'_n = 0.$$

Se se considera a equação i ($j = i$) verifica-se que a variável f'_k é assim eliminada da equação. A normalização da linha i na equação i (para que o coeficiente s_{ii} de S seja igual a 1) implica normalmente fazer a seguinte mudança:

$$f_i = \frac{1}{c_{ii}}f'_i.$$

De fato, esta normalização é inútil, pois ela não modifica em nada o princípio do método e pode introduzir números fracionários nos cálculos.

A matriz F pode também ser construída através de transformações sucessivas. Parte-se de uma matriz identidade efetuando-se *sobre as colunas de F* as mesmas operações que sobre as linhas de C . No caso precedente, consiste em substituir a coluna “ k ” por “ c_{ii} vezes a coluna k menos c_{ki} vezes a coluna i ” na matriz F . Tal conduta é justificada, uma vez que essa mudança de variável é descrita por:

$$f = \begin{matrix} 1 \\ i \\ k \\ n \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -c_{ki} & 0 \\ 0 & 0 & c_{ii} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} f'$$

e que a matriz F da mudança de variáveis resultante do encadeamento de mudanças de variáveis elementares A_1, A_2, \dots, A_l é igual a:

$$F = A_1 A_2 \dots A_l.$$

B.2 Exemplo

Considere a rede de Petri da FIG. 2.13. Inicialmente tem-se:

$$C_0 = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ -1 & 1 & -3 & 3 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \quad F_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

A eliminação da linha 1 em relação à primeira coluna é realizada multiplicando-se a primeira linha $C(1, \cdot)$ por -1 , e substituindo-se $C(2, \cdot)$ por $C(2, \cdot) + C(1, \cdot)$ e $C(3, \cdot)$ por $C(3, \cdot) - C(1, \cdot)$. A linha 1 não será mais tratada, e vai corresponder à primeira linha de S ; ela não fará parte da base geradora do espaço vetorial de soluções. Obtém-se então:

$$C_1 = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -3 & 3 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \quad F_1 = \begin{bmatrix} -1 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Se é impossível fazer aparecer um 1 numa linha de zeros, pode-se continuar, neste caso, trocando entre si as linhas 2 e 5 assim como as colunas b e d . A reordenação das colunas (ordem de escrita das equações) não altera F . Tem-se então:

$$C_2 = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 3 & -3 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad F_2 = \begin{bmatrix} -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

A linha 2 pode agora ser eliminada do espaço de soluções substituindo-se:

- $C_2(3, \cdot)$ por $C_2(3, \cdot) - 3C_2(2, \cdot)$
- e $C_2(4, \cdot)$ por $C_2(4, \cdot) + C_2(2, \cdot)$.

Obtém-se então:

$$C_3 = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad F_3 = \begin{bmatrix} -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & -3 & 1 & 0 \end{bmatrix}.$$

As três últimas linhas sendo nulas, a triangularização está pronta. O espaço vetorial das soluções é de dimensão 3 e as três últimas colunas de F_3 são uma base deste espaço. Elas correspondem aos três últimos componentes das soluções em f' , isto é, após a mudança de variáveis.

Os componentes conservativos são:

$$\begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \\ -3 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

o que fornece os seguintes invariantes lineares para a marcação inicial indicada na FIG. 2.13:

$$\begin{aligned} -M(1) + M(3) - 3M(5) &= -1 \\ M(1) + M(2) &= 1 \\ M(4) + M(5) &= 1. \end{aligned}$$

B.3 Algoritmo simplificado e procura das soluções positivas

De fato, apenas o cálculo das colunas da matriz F que correspondem às $n - r$ componentes da base das soluções f é útil. As outras colunas de F podem ser eliminadas.

A cada passo (eliminação de uma variável), a coluna de C (equação que vai ser colocada na base das equações) que serviu para eliminar uma variável (todos os elementos

da coluna são nulos exceto um) é eliminada, assim como a linha que corresponde a esta variável. Como a variável não faz parte da base de soluções, a coluna correspondente de F é também apagada.

Enfim, ao invés de construir F explicitamente, as operações efetuadas sobre as linhas de C são memorizadas sob a forma de somas formais. Cada coluna de F é memorizada sob a forma de uma combinação linear das colunas de origem ($f_i + f_j$ descreve uma coluna obtida através da soma das colunas i e j de F_0).

Por outro lado, as soluções positivas são, em geral, as mais interessantes (o que é evidente para os componentes repetitivos estacionários). É útil, portanto, favorecer as combinações de linhas positivas no momento da eliminação.

Obtém-se então o algoritmo simplificado:

1. Procurar uma coluna que possua um só termo diferente de zero. Ela corresponde a uma equação da base de equações e é apagada, sendo a variável correspondente ao termo não nulo eliminada, pois como a única solução possível é nula, não pode fazer parte da base de soluções. Apagar a linha de C associada a esta variável e a coluna correspondente de F . Repetir a etapa **1** enquanto for possível, ir para **2**.
2. Procurar uma coluna tendo apenas um componente não nulo com um sinal dado; os outros devem ser nulos ou de sinal contrário (um positivo e todos os outros negativos ou nulos, por exemplo). A variável correspondendo a esta linha é eliminada efetuando-se apenas combinações positivas de linhas de modo a provocar o aparecimento de zeros na coluna considerada. Após esta operação, eliminar esta coluna, a linha de C que corresponde à variável eliminada e a coluna de F associada. Se tal coluna é encontrada, voltar a **1** senão ir a **3**.
3. Procurar uma coluna tendo $i \geq 2$ componentes não nulos positivos e $j \geq 2$ componentes não nulos negativos. Com a ajuda de um dos componentes não nulos positivos, anular $j - 1$ componentes negativos. Voltar à etapa **2** para eliminar a variável que corresponde a um único componente negativo não anulado. É preciso salientar que foi escolhido um componente positivo entre i e um componente negativo entre j . Existem, portanto, $i \times j$ maneiras de proceder. Esta escolha pode ter consequência sobre a forma da base obtida. Se nenhuma coluna que responda ao critério acima é encontrada, ir para o passo **4**.
4. Se existe uma coluna tendo todos os componentes de mesmo sinal, prosseguir a eliminação de Gauss para esta coluna (mas a base não possuirá apenas soluções positivas); em seguida, ir a **1**, senão o algoritmo termina. Ou todas as colunas são nulas, ou a matriz não possui mais nenhum elemento. As combinações não apagadas que correspondem à construção de colunas de F fornecem diretamente os vetores da base.

Considere novamente o exemplo da FIG. 2.13. Inicia-se com:

$$C_0 = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ -1 & 1 & -3 & 3 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix}.$$

Observe a primeira coluna de C_0 , cuja equação correspondente é selecionada para fazer parte da base de equações. Para eliminar a variável f_2 , basta somar as linhas correspondentes às variáveis f_1 e f_3 à linha correspondente à f_2 . Com estas novas variáveis, a equação terá apenas um termo não nulo. A linha f_2 é eliminada (f_2 não fará parte da base de soluções), assim como a primeira coluna de C_0 . Obtém-se desta forma:

$$C_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -3 & 3 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} f_1 + f_2 \\ f_2 + f_3 \\ f_4 \\ f_5 \end{bmatrix}.$$

A segunda coluna de C_1 é utilizada para colocar f_4 fora da base de soluções, o que é obtido somando três vezes a linha “ f_4 ” à linha “ $f_2 + f_3$ ” e uma vez à “ f_5 ”. A segunda coluna de C_1 é eliminada, assim como a linha correspondente à variável f_4 . Isto dá:

$$C_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} f_1 + f_2 \\ f_2 + f_3 + 3f_4 \\ f_4 + f_5 \end{bmatrix}.$$

Resta apenas zeros e o algoritmo termina. A base obtida corresponde às somas figurando à direita da matriz e descreve as manipulações efetuadas nas linhas de C e colunas de F . Substituindo os símbolos f_i pela coluna i de F_0 , obtém-se a base:

$$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \\ 1 \\ 3 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

BIBLIOGRAFIA

- ANDREU, D., PASCAL, J. C., VALETTE, R. Interaction of discrete and continuous parts of a batch process control system. In: WORKSHOP ON ANALYSIS AND DESIGN OF EVENT-DRIVEN OPERATIONS IN PROCESS SYSTEMS, 1995, London.
- ANDREU, D., PASCAL, J. C., PINGAUD, H., VALETTE, R. Batch process modelling using Petri nets. In: IEEE INTERNATIONAL CONFERENCE ON SMC, 1994, San Antonio, USA. 314-319p.
- ANDREWS, G. R. *Concurrent programming: principles and practice*. California: The Benjamin/Cummings Publishing Company, Inc., 1987. 637p. ISBN 0-8053-0086-4.
- BERTHOMIEU, B., MENASCHE, M. An enumerative approach for analysing time Petri nets. *Information processing*, p. 41-46, 1983.
- BITTENCOURT, G. *Inteligência Artificial: ferramentas e teoria*. In: ESCOLA DE COMPUTAÇÃO, 10 a, 1996, Campinas. Campinas: Instituto de Computação, UNICAMP, 1996. 240p.
- BOUCHON-MEUNIER, B. *La logique floue*. 2.ed. Paris: Presses Universitaires de France, 1994. 128p. ISBN 2 13 045007-5
- BLANCHARD, M. *Comprendre maîtriser et appliquer le Grafcet*, França: Cepadues Éditions, 1979. 174p.
- BRAMS, G.W. *Réseaux de Petri: théorie et pratique*; Tome 1 théorie et analyse. França: Éditions Masson, 1982. 184p. ISBN 2-903-60712-5.
- BRAMS, G.W. *Réseaux de Petri: théorie et pratique*; Tome 2 Modélisation et applications. França: Éditions Masson, 1983. 160p. ISBN 2-903-60713-3.
- BRAUER W. (Ed.) *Net theory and applications*; Proceedings of the advanced course on general net theory of processes and systems. Hamburg: Springer-Verlag, 1980. (Lecture Notes in Computer Science 84).
- BRAUER W. (Ed.): *Petri Nets: central models and their properties*. Hamburg: Springer-Verlag, 1987. (Lecture Notes in Computer Science 254).
- BRAUER W. (Ed.): *Petri Nets: applications and relationships to other models of concurrency*. Hamburg: Springer-Verlag, 1987. (Lecture Notes in Computer Science 255).

- CAO, T., SANDERSON, A. C. A Fuzzy Petri net approach to reasoning about uncertainty in robotic systems. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 1993. *Proceedings of ...* vol. 1, p. 317-322.
- CARDOSO, J., VALETTE, R., DUBOIS, D. Petri nets with uncertain markings. In: ROZENBERG, G. (Ed.). *Lecture Notes in Computer Science; Advances in Petri nets 1990*. Berlin: Springer Verlag, 1991. v. 483, p. 64-78.
- CARDOSO, J., VALETTE, R. and PRADIN-CHEZALVIEL., B. Linear logic for imprecise firings in fuzzy Petri nets, In B. BOUCHON-MEUNIER, L. ZADEH and R. YAGER (Eds). *Fuzzy logic and soft computing*. World Scientific, 1995. p. 119-128.
- CARROL, J., LONG, D. *Theory of finite automata*. Englewood Cliffs, NJ: Prentice-Hall Int. Ed., 1989. 438p. ISBN 0-13-913815-3.
- CASSANDRAS, C. G. *Discrete event systems: modeling and performance analysis*. Boston: Aksen Assoc. Inc. Publishers, 1993. 790p. ISBN 0-256-11212-6.
- CHEN, Shyi-Ming, KE, Jyh-Sheng, CHANG, Jin-Fu. Knowledge representation using fuzzy Petri nets. *IEEE Trans. on Knowledge and Data Engineering*, local, v. 2, n. 3, p. 311-319, Sept. 1990.
- COURVOISIER, M., VALETTE, R. *Commande des Procédés Discontinus - Logique Séquentielle*. Bordas: Dunod Université, 1986. 181p. ISBN 2-04-016423-5.
- DAVID, R., ALLA, H.: *Du Grafet aux Réseaux de Petri*. Paris: Éditions Hermes, 1989. 424p.
- DAUBAS, B., PAGES, A., PINGAUD, H.: *Combined simulation of hybrid processes*. IEEE INTERNATIONAL CONFERENCE ON SMC, 1994, San Antonio, USA. p. 314-319.
- DUBOIS, D., PRADE, H. *Possibility theory: an approach to computerized processing of uncertainty*. New York: Plenum Press, 1988. 263p. ISBN 0-306-42520-3.
- FITTING, M. *First-order logic and automated theorem proving*. New York: Springer-Verlag, 1990. 241p.
- FREEDMAN, P., MALOWANY A. *Sage: a decision support system for the optimization of repetitive workcell sequencing problems*. Report TR-CIM-88-16, McGill University, 1988.
- GENRICH, H. J. predicate/transition nets. In: JENSEN, J. and ROZENBERG, G. (Eds). *High-level Petri nets: theory and application*. Berlin: Springer-Verlag, 1991. p. 3-43. ISBN 3-540-54125-X, ISBN 0-387-54125-X.
- GIRARD, J.Y. Linear logic. *Theoretical Computer Science*, n. 50, p. 1-102, 1987.
- GIRARD, J.Y. La logique linéaire. *Pour la Science*, n. 150, p. 74-85, avril 1990.

- KLIR, G. J., FOLGER, T. A. *Fuzzy sets, uncertainty, and information*. New Jersey: Prentice Hall, 1988. 355p. ISBN 0-13-345984-5.
- GONDRAM, M., MINOUX, M. *Graphes et algorithmes*. Paris: Editions Eyrolles, 1985. 545p.
- HACK, M. H.-T. *Analysis of production schemata by Petri nets*. Massachussets: MIT, 1972. (Thesis, Master of Science in Computer Science).
- HEUSER, C. A. Modelagem conceitual de sistemas: redes de Petri. In: ESCOLA BRASILEIRO-ARGENTINA DE INFORMÁTICA, 5º, 1991, Nova Friburgo. Campinas: R. Vieira Gráfica e Editora Ltda, 1991. 150p.
- HOLT, A., COMMONER, F. *Events and conditions*. Massachussets: Record of the Project MAC Conference on Concurrent Systems and Parallel Computation (MIT), June 1970. p. 3-33.
- INTERNATIONAL CONFERENCE ON APPLICATION AND THEORY OF PETRI NETS, 1980-. *Proceedings . . .* (Os melhores artigos são submetidos e publicados pela Springer-Verlag na série *Lecture Notes in Computer Science: Application and Theory of Petri nets*).
- INTERNATIONAL WORKSHOP ON PETRI NETS AND PERFORMANCE MODELS, 1985-. *Proceedings . . .* Whashington: IEEE Computer Society Press.
- JALOTE, P. *An integrated approach to software engineering*. New York: Springer-Verlag, 1991. 375p. ISBN 0-387-97561-6.
- JENSEN, K. A high-level language for system design and analysis. In: JENSEN, J. and ROZENBERG, G. (Eds). *High-level Petri nets: theory and application*. Berlin: Springer-Verlag, 1991. p. 44-119. ISBN 3-540-54125-X, ISBN 0-387-54125-X.
- JOHNSONBAUGH, R. *Discrete mathematics*. 3.ed. New York: Macmillan Publishing Company, 1993. 800p.
- LOONEY, C. G. Fuzzy Petri nets for rule-based decision making. *IEEE Trans. on Systems Man and Cybernetics*. v. 18, n. 1, p. 178-183, jan./feb. 1988.
- MAZIERO, C. A. *Um ambiente para a análise e simulação de sistemas modelados por redes de Petri*. Florianópolis: Universidade Federal de Santa Catarina, 1990. 96p. (Dissertação, Mestrado em Sist. Cont. e Automação Industrial).
- MERLIN, P. *A study of the recoverability of computer systems*. Irvine: University of California, 1974. (Thesis, Phd in Computer Science).
- MIYAGI, P. E. *Controle programável: fundamentos do controle de sistemas a eventos discretos*. São Paulo: Editora Edgard Blücher Ltda., 1995. 204p.
- MOLLOY, M.: *On the integration of delay and throughput measures in distributed processing models*. Report csd-810021, University of California, 1981.

- MURATA, T., ZHANG, D. A predicate-transition net model for parallel interpretation of logic programs. *IEEE Transaction on Software Engineering*, v.14, n.4, p. 481-497, abril 1988.
- MURATA, T. Petri nets: properties, analysis and applications, *Proceedings of the IEEE*, local, v.77, n.4, p. 541-580, abril 1989.
- OLIVEIRA, F. A. S. *Um ambiente integrado de ferramentas para o projeto de sistemas a eventos discretos*. Florianópolis: Universidade Federal de Santa Catarina, 1994. 84p. (Dissertação, Mestrado em Sist. Cont. e Automação Industrial).
- PRADIN CHÉZALVIEL, B., VALETTE, R. Petri nets and Linear logic for process oriented diagnosis, In: *IEEE/SMC INT. CONF. ON SYSTEMS, MAN AND CYBERNETICS: SYSTEMS ENGINEERING IN THE SERVICE OF HUMANS*, 1993, Le Touquet, França. *Proceedings of* v.2, p.264-269.
- PETERSON, J.L. *Petri net theory and the modelling of systems*. Englewood Cliffs, NJ: Prentice-Hall, 1981. 286p.
- PETRI, C. A. Communication with automata. Tech. report n. RAD-TR-65-377, 1966. Versão inglesa.
- RAMCHANDANI, C.: *Analysis of asynchronours concurrent systems by timed Petri nets*. Project mac tr 120, Massachussets Institute of Technology, 1974.
- REISIG, W. *Petri nets: an introduction*, Berlin: Springer Verlag, 1985. 161p. (EATCS Monographs on Theoretical Computer Science). ISBN 0-387-13723-8.
- REUTENAUER, C. *The mathematics of Petri nets*, Hertfordshire: Prentice Hall Int. Ltda, 1990. 117p.
- ROSEN, K. H. *Discrete mathematics and its applications*. 2.ed. local: McGraw-Hill Int. Editions, 1991. 793p. ISBN 0-07-112788-7.
- SCARPELLI, H., GOMIDE, F., YAGER, R. A Reasoning algorithm for high level fuzzy Petri nets, *IEEE Trans. on Fuzzy Systems*, v.4, n.3, p.282-294, August 1996.
- SIBERTIN-BLANC, C. High-level Petri nets with data structures. In: EUROPEAN WORKSHOP ON APPLICATION AND THEORY OF PETRI NETS, 6º, 1985, Helsinki, Finland. Helsinki: Ed. Digital Systems Laboratory, p.141-170.
- SILVA, M.: *Las redes de Petri en la automática y la informática*. Madri: Ed. AC, 1985. 190p.
- SILVA, M., VALETTE, R. Petri nets and flexible manufacturing. In: GOOS, G., HARTMARIS, J.: *Lecture Notes in Computer Science*; Advances in Petri nets. Berlin: Springer Verlag, 1990. v. 424, p. 374-417.
- SOARES, F. A., CARDOSO, J., CURY, J. E. An integrated environment of tools for the design of manufacturing systems, In: WORKSHOP INTELLIGENT MANUFACTURING SYSTEMS, 1994, Viena, Áustria.

- VALETTE, R., ATABAKHCHE. H. Petri nets for sequence constraint propagation in knowledge based approaches. In: VOSS, K., GENRICH, H. J., ROZENBERG, G.: *Concurrency and Petri Nets*, Berlin: Springer Verlag, 1987. p. 555-569. ISBN 3-540-18057-5.
- R. VALETTE, D. ANDREU, J. CARDOSO, J.C. PASCAL. Fuzzy Petri nets and their application in CIME, Special Issue on Recent Applications of Petri Nets of the *Transactions of the Institute of Electrical Engineers of Japan*, v. 114-C, n. 9, p. 876-880. 1994.
- ZISMAN, M.D. Use of production systems for modeling asynchronous concurrent processes. In: WATTERMAN, D.A., HAYES-ROTH, F. (Ed.). *Pattern Directed Inference systems*. London: Academic Press, 1978, p. 53-68.
- ZURAWSKI, R., ZHOU, M. Petri nets and industrial applications: a tutorial, *IEEE Trans. on Industrial Electronics*, v.41, n.6, p. 567-583, 1994.
- ZADEH, L.A. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and Systems*, n. 1, p. 3-28, 1978.
- ZADEH, L.A. Theory of fuzzy sets. In BELZER, J., HOLZMAN, A., KENT A. (Ed.), *Encyclopedia of Computer Science and Technology*, New York, 1977.

Índice

- árvore de cobertura, 55
- AIF*, 83
- alfabeto
 - de lugares, 41
 - de transições, 42
- análise
 - através de redução, 60–65
 - da rede de Petri a objetos, 99
 - da rede de Petri colorida, 91
 - da rede de Petri predicado-transição, 99
 - da rede interpretada, 77
 - estrutural, 56–60, 65
 - por enumeração de marcações, 53
 - relação entre os métodos, 65–68
- ANDREU, 131
- ANDREWS, 28
- aplicação
 - de entrada, 31
 - de saída, 31
- ARP*, 70, 83
- atividades, 12
- axioma da gramática, 42

- BERTHOMIEU, 112
- BITTENCOURT, 51
- bloqueio mortal, 15, 17
- boas propriedades, *ver* rede de Petri, propriedades da, 60
- BOUCHON-MEUNIER, 131
- BRAMS, 28

- C*, *ver* matriz, de incidência
- cadeia de Markov, 111
- caminhos alternativos, 21
- CARROL, 28
- CASSANDRAS, 27, 112
- causalidade, 105
- circuito, 69, 133
- cobertura de componentes
 - conservativos, 57, 58
 - repetitivos estacionários, 57
- COMMONER, 7
- competição entre processos, 13
- componentes
 - conservativos, 47–50, 57
 - repetitivos estacionários, 47, 49–51, 59
- conectivos da lógica linear, 122
- conflito
 - efetivo, 35
 - estrutural, 35
- conjunto de marcações acessíveis, 39
- conjunto de marcações acessíveis, 42
- conjuntos nebulosos, 119
- cooperação entre processos, 13
- COURVOISIER & VALETTE, 28

- DAUBAS, 131
- DAVID, 28
- decomposição material, 15
- disparo, *ver* transição, tiro de
 - duração de, 106
- DUBOIS & PRADE, 131

- equação fundamental, 37–39, 48, 49
- estado estável, 115
- eventos, 12, 18
 - concorrência entre, 27
 - ocorrência de, 19
 - paralelismo entre, 27
 - seqüências de, 12
- exemplos:
 - cálculo invariante lugar, 140
 - estação de coleta de petróleo, 80
 - leitores e escritores, 85
 - oficina com 3 máquinas, 85, 89
 - reator, 128
 - seqüência de fabricação, 19, 21, 27, 28
 - sistema de transporte, 20–22, 24

- sistema de triagem, 13, 15–17, 23
sistema tipo batelada, 26, 50, 58
- explosão combinatória de estados, *ver* estados, explosão combinatória
- ficha, 18, 20, 24, 31, 75
colocação da, 19
desaparecimento da, 21
disponível, 106
indisponível, 106, 107
individualização da, 87
não reservada, 106
na rede nebulosa, 120
portadora de informação, 84
reservada, 106
retirada da, 19
- fichas, 69
balanço das, 33, 35, 57
cores associadas às, 88
limite de, 58
- FITTING, 131
- FREEDMAN, 112
- FTP anônimo*, 70
- função transição de estado, 13
- funções associadas aos arcos, 90
- GENRICH, 104
- GIRARD, 131
- GONDRAM & MINOUX, 136
- Grafcet, 7
- grafo
componente de um, 134
conexo, 134
etiquetado, 32
eventos, 69
fortemente conexo, 134
fracamente conexo, 134
marcações acessíveis, 39, 44, 46, 51
- gramática, 41
- HACK, 7
- hierarquia de controle, 7
- HOLT, 7
- implementação
centralizada, 113
declarativa, 113
descentralizada, 113
- não procedimental, 114
procedimental, 113, 114
- incidência anterior, *ver* aplicação de entrada
- incidência posterior, *ver* aplicação de saída
- interação entre dados e ambiente, 75–76
- interpretação, 74
- invariante linear
algoritmo, 141
cálculo, 137
de lugar, 47–50, 57, 91
de transição, 47, 49–51
- JALOTE, 51
- JENSEN, 104
- jogador de rede de Petri, 114–116
- JOHNSONBAUGH, 28, 136
- KLIR, 131
- lógica linear, 39, 122
- λ , 41
- lema de Karp e Miller, 54
- limitabilidade, *ver* rede de Petri, limitada
- lugar, 18, 20
binário, 42, 57
implícito, 61
degenerado, 63
limitado, 42
predicado associado ao, 24
substituível, 60
tempo associado, 105
- lugares
dobramento de, 84, 86
idênticos, 63
projeção de, 99
- máquina de estado, 68
- máquinas de estados finitos, 13, 17
composição de, 14
comunicação entre, 16
exemplo de, 14
independentes, 14
sincronização de, 14
- método de Gauss, 137
- malha elementar, 33
- marcação, 31
cálculo da próxima, 34

- caracterização da, 66–68
- evolução da, 37
- inicial, 31, 33
- limite superior da, 58
- mínima, 126
- na rede nebulosa, 120
- transformação da, 37
- trasformação da, 126
- usando lógica linear, 123
- matriz
 - de incidência, 32
 - de incidência anterior, 32
 - notação da, 33
 - de incidência posterior, 32
- MAZIERO, 70
- MERLIN, 112
- MOLLOY, 112
- n-upla, 94
- normas, 7
- OLIVEIRA, 83
- palavra
 - do vocabulário T^* , 42
 - do vocabulário P^* , 41
 - $\mu(M)$, 41
- paralelismo, 13, 22
 - efetivo, 35
 - estrutural, 35
 - verdadeiro, 36
- PETERSON, 28
- PETRI, 7, 11
 - Post*, 31, *ver* matriz, de incidência posterior, 90
 - Pré*, 31, *ver* matriz, incidência anterior, 90
 - Pre*[1]_{*i*}, *ver* vetor coluna, *Pre*[1]_{*i*}
- PRADIN, 131
- processos, 12
 - competitivos, *ver* competição entre processos
 - cooperantes, *ver* cooperação entre processos
 - evolução assíncrona de, 20
 - evolução de, 13
 - evolução síncrona de, 20
 - independência de, 13
 - interação entre, 13, 19
 - noção de, 39
 - paralelos, *ver* paralelismo
 - pseudo-paralelos, *ver* pseudo-paralelismo
 - repetição de, 22
 - seqüência de, 19
 - seqüenciais comunicantes, 14
- propriedades
 - da rede de Petri, *ver* rede de Petri, propriedades da
 - estruturais, 47
- pseudo-paralelismo, 13
- RAMCHANDANI, 112
- reator, 127
- receita de fabricação, 129
- recurso, 24
 - alocação de, 23
 - disponível, 23
 - e lógica linear, 126
 - partilhamento de, 32
 - utilização do, 23
- rede de Petri
 - a objetos, 96–103
 - aplicações da, 7
 - as três visões da, 30
 - autônoma, 74
 - binária, 43, 44, 68, 69
 - colorida, 88–94, 99–103
 - conservativa, 48, 58
 - de alto nível, 84–103
 - definição formal da, 31
 - descrição usando lógica linear, 123
 - elementos básicos da, 17
 - estocástica, 109–112
 - grafo associado à, 32
 - e sistemas híbridos, 126
 - interpretada, 73–83
 - exemplo da, 80
 - modelagem da, 79
 - validação da, 79
 - e lógica linear, 122–126
 - limitada, 42–44, 55, 57–59, 69
 - decidibilidade da, 54
 - marcada, 31
 - não conservativa, 58
 - não limitada, 43, 46, 58, 60

- não reiniciável, 46
- não viva, 46, 60
- nebulosa, 118–121
- notação matricial da, 32
- ordinária, 73, 88, 91, 103
- própria, *ver* rede de Petri, reiniciável
- predicado-transição, 92–96, 99–103
- propriedades da, 30, 42, 46
- pura, 33
- quase viva, 46
- reiniciável, 42, 46, 55
- repetitiva, 49, 51, 59
- representação gráfica da, 30
- salva, *ver* rede de Petri, binária
- segura, *ver* rede de Petri, binária
- subjacente, 82, 91, 99, 103
- temporal, 107–110
- temporizada, 105–107, 109
- vantagens da, 7
- viva, 42, 45, 46, 56, 59, 69
- redes de Petri
 - implementação, *ver* implementação
- regras de reescrita, 41, 42
- reiniciabilidade, *ver* rede de Petri, reiniciável
- REISIG, 104
- ROSEN, 28, 136
- seqüência
 - caracterização da, 39
 - ordenada, 124
 - usando lógica linear, 124, 126
- seqüência s , *ver* transições, seqüência de tiros de
 - existência da, 37, 38
 - infinita, 54
- seqüente, 124
- SIBERTIN-BLANC, 104
- sistema
 - a eventos discretos, 12
 - contínuo, 11
 - de regras, 30
 - discretizado, 11
 - discreto, 11, 12
 - híbrido, 126
 - modelagem de, 127
 - repetitivo, 26
 - tipo batelada, 127
- SOARES, 80
- sub-rede, 48–51
- subclasses de redes de Petri, 68
- subgrafo, 134
- suporte, 58
- tempo (explícito), 77
- tempo associado
 - à transição, *ver* transição, tempo associado
 - ao lugar, *ver* lugar, tempo associado
- transição, 18, 20, 32
 - bloqueio mortal de, 44, 45
 - disparo, *ver* transição, tiro de
 - identidade, *ver* transição, neutra
 - interna, 114
 - não viva, 44
 - na rede nebulosa, 120
 - neutra, 64
 - quase viva, 44
 - sensibilizada, 33, 36, 41, 42
 - notações de, 34
 - tempo associado, 106
 - tiro de, 19, 34, 41
 - notações do, 34
 - usando lógica linear, 124
 - viva, 44
- transições
 - conflito de, 21, 25, 26, 35, 36
 - cores associadas às, 88
 - dobramento de, 84, 86
 - idênticas, 65
 - paralelismo de, 35
 - seqüência de tiros de, 36, 42
- trocadores de calor, 127
- variável tipo monoestável, 74
- vetor característico s , 37, 50
- vetor coluna
 - $Pre[1]_i$, 34
 - $C[1]_i$, 35, 37
- vivacidade, *ver* rede de Petri, viva
- vocabulário, 41
- ZADEH, 131

Janette Cardoso,
catarinense, doutora em
Informática Industrial pela
Université Paul Sabatier de
Toulouse (França) em 1990,
ingressou no Departamento de
Engenharia Elétrica da UFSC
em 1982 e atualmente é
professora no Departamento de
Automação e Sistemas.
Desenvolve suas atividades de
pesquisa no LCMI/UFSC –
Laboratório de Controle e
Microinformática. Sua
área de interesse principal
consiste no desenvolvimento
de uma abordagem consistente
combinando rede de Petri, lógica
possibilista e lógica linear, com
aplicações a sistemas a eventos
discretos e sistemas híbridos.

Robert Valette,
francês, doutor em
Informática Industrial pela
Université Paul Sabatier de
Toulouse (França) em 1976, é
Directeur de Recherche no
LAAS/CNRS – *Laboratoire*
d'Architecture et d'Analyse de
Systèmes, em Toulouse, França.
Sua área de interesse em
pesquisa inclui a aplicação da
teoria de redes de Petri em
sistemas de manufatura e o
estudo das relações entre redes
de Petri e lógica, com aplicações
à supervisão de sistemas de
produção.