

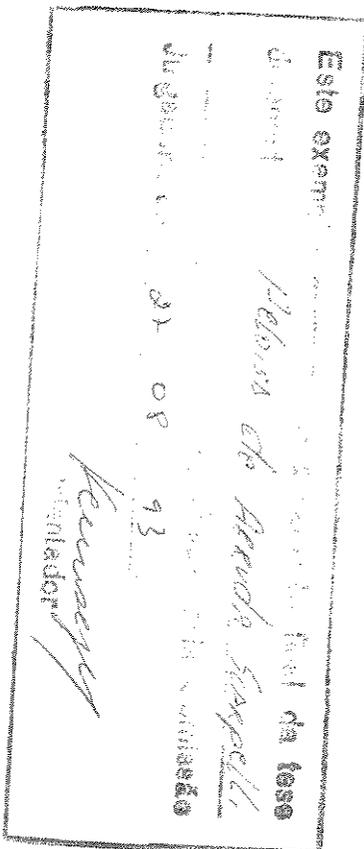
UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA ELÉTRICA  
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO  
INDUSTRIAL

**MODELAGEM, PROJETO E VERIFICAÇÃO DE  
BASES DE REGRAS NEBULOSAS VIA TEORIA DE  
REDES**

Por: Heloisa de Arruda Camargo [Scarpelli 76  
Orientador: Prof. Dr. Fernando Antonio Campos [Gomide t

Tese apresentada à Faculdade de Engenharia  
Elétrica da Unicamp como parte dos requisitos  
exigidos para a obtenção do título de Doutor em  
Engenharia Elétrica.

Campinas, agosto de 1993



## Resumo

Neste trabalho são propostos dois modelos formais para a representação de regras de produção nebulosas, como base em teoria de redes. A proposta desses modelos tem o objetivo de oferecer um mecanismo de representação onde os problemas relativos ao projeto, execução e verificação de bases de conhecimento nebulosas possam ser tratados de maneira uniforme e integrada. O modelo de Redes de Petri Nebulosas representa detalhadamente a execução da regra de inferência principal do raciocínio aproximado, chamada de regra composicional. O modelo de Redes de Petri Nebulosas de Alto Nível permite representar as regras nebulosas de um nível de abstração mais alto, gerando representações mais compactas e tratáveis. Os componentes de cada um desses dois modelos guardam uma relação bastante próxima, possibilitando que as informações sejam acessadas a partir do nível mais conveniente a cada fase do desenvolvimento de sistemas. A modelagem de vários tipos de regras como regras com múltiplos antecedentes, regras quantificadas e qualificadas, regras paralelas, regras conflitantes e fontes parciais de informação é discutida. Dois algoritmos de raciocínio aproximado são apresentados, utilizando as estratégias de encadeamento para frente e encadeamento para trás. A questão de verificação de inconsistência entre regras de produção nebulosas é discutida. Procedimentos para verificação da existência de conflitos em potencial, tanto a nível local como a nível global, são apresentados.

## **Abstract**

This work presents two formal models for the representation of fuzzy production rules, based on net theory. The proposal of the models aims at offering a mechanism where problems concerning design, execution and verification of fuzzy knowledge bases can be faced in an uniform and integrated manner. The Fuzzy Petri Net model represents in details the numerical components underlying the main inference rule in approximate reasoning, called compositional rule. The High Level Fuzzy Petri Net model allows to represent fuzzy rules in a higher level of abstraction, yielding more compact and manageable representations. The components of both systems are closely related, allowing the accessing of information from the most convenient level in each phase of systems development. The modeling of several rule types like rules with multiple antecedent, quantified and qualified rules, parallel rules, conflicting rules and partial sources of information has been discussed. Two different fuzzy reasoning algorithms have been presented, using forward chaining and backward chaining strategies. The issue of inconsistency checking among fuzzy production rules has also been approached. Procedures for verifying the existence of potential conflicts, both at local and global levels are included.

*para Moacir, Maíra e Alexandre*

Este trabalho contou com o apoio financeiro do programa PICD/CAPES, através da UFSCar, e da FAPESP, projeto n. 92/2151-8.

## **Agradecimentos**

Gostaria de agradecer inicialmente ao professor Fernando Gomide do Departamento de Engenharia de Computação e Automação Industrial da Faculdade de Engenharia Elétrica, Unicamp, pela dedicação e incentivo durante a orientação dessa tese.

Gostaria também de agradecer ao Departamento de Computação da Universidade Federal de São Carlos por ter dado condições ao desenvolvimento desse trabalho. Agradeço ao CNPq, Projeto RHAE, pela oportunidade de estágio no exterior para coletar subsídios à elaboração dessa tese.

Finalmente, gostaria de agradecer a minha família, em especial a meu marido e aos meus filhos que, direta ou indiretamente, contribuíram sensivelmente para a realização deste trabalho.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Formulação do Problema . . . . .	1
1.2	Objetivos . . . . .	2
1.3	Desenvolvimento da Tese . . . . .	3
<b>2</b>	<b>Conceitos Básicos</b>	<b>4</b>
2.1	Introdução . . . . .	4
2.2	Introdução ao Raciocínio Aproximado . . . . .	5
2.2.1	Subconjuntos Nebulosos . . . . .	5
2.2.2	Lógica Nebulosa . . . . .	11
2.2.3	Raciocínio Aproximado . . . . .	12
2.2.4	Qualificação de possibilidade . . . . .	22
2.2.5	Métodos de Inferência Nebulosa . . . . .	23
2.3	Introdução às Redes de Petri . . . . .	27
2.3.1	Métodos de Análise . . . . .	29
2.3.2	Redes de Alto Nível . . . . .	33
2.4	Conclusão . . . . .	37

<b>3</b>	<b>Redes de Petri Nebulosas</b>	<b>38</b>
3.1	Introdução . . . . .	38
3.2	Revisão da Literatura . . . . .	39
3.3	O Modelo de Redes de Petri Nebulosas . . . . .	41
3.4	Encadeamento de Regras . . . . .	48
3.5	Regras com Múltiplos Antecedentes . . . . .	50
3.6	Regras Quantificadas e Qualificadas . . . . .	55
3.6.1	Quantificadores no Antecedente . . . . .	55
3.6.2	Qualificação de Certeza . . . . .	57
3.7	Regras Paralelas . . . . .	58
3.8	Regras Conflitantes . . . . .	60
3.9	Matriz de Incidência e Equação de Estado . . . . .	62
3.10	Exemplo de aplicação . . . . .	64
3.11	Conclusão . . . . .	71
<b>4</b>	<b>Redes de Petri Nebulosas de Alto Nível</b>	<b>72</b>
4.1	Introdução . . . . .	72
4.2	O Modelo de Redes de Petri Nebulosas de Alto Nível . . . . .	73
4.3	Equivalência entre RPN e RPNAN . . . . .	77
4.4	Encadeamento de Regras . . . . .	78
4.5	Regras com Múltiplos Antecedentes . . . . .	79
4.6	Regras Quantificadas e Qualificadas . . . . .	82
4.7	Regras Paralelas . . . . .	82
4.8	Regras Conflitantes . . . . .	84
4.9	Fontes Parciais de Informação . . . . .	85

4.9.1	Representado Vários Fatos Sobre a Mesma Variável . . . . .	86
4.9.2	Representando Regras com a Mesma Variável no Consequente e Variáveis Diferentes no Antecedente. . . . .	87
4.10	Regras com Conjunção no Consequente . . . . .	88
4.11	Alcançabilidade . . . . .	90
4.12	Exemplo de aplicação . . . . .	91
4.13	Conclusão . . . . .	101
<b>5</b>	<b>Algoritmos de Raciocínio Aproximado</b>	<b>102</b>
5.1	Introdução . . . . .	102
5.2	Algoritmo de Encadeamento para Frente . . . . .	102
5.3	Aplicação do Algoritmo de Encadeamento para Frente . . . . .	107
5.4	Algoritmo de Encadeamento para Trás . . . . .	108
5.5	Aplicação do Algoritmo de Encadeamento para Trás - caso 1 . . . . .	111
5.6	Aplicação do Algoritmo com Encadeamento para Trás - caso 2 . . . . .	112
5.6.1	Fase 1 - Extração da subrede . . . . .	113
5.6.2	Fase 2 - Execução da subrede . . . . .	115
5.7	Conclusão . . . . .	120
<b>6</b>	<b>Verificação de Inconsistência</b>	<b>122</b>
6.1	Introdução . . . . .	122
6.2	Verificação de Bases de Conhecimento . . . . .	123
6.3	Conceitos Básicos . . . . .	124
6.3.1	RPNAN Hierárquica . . . . .	124
6.3.2	Matriz de Teste de Condições . . . . .	127
6.3.3	Composição Matricial . . . . .	129

6.4	Inconsistências na Lógica Nebulosa . . . . .	130
6.5	Verificação de Inconsistência a Nível Local . . . . .	135
6.6	Verificação de Inconsistência a Nível Global . . . . .	138
6.7	Conclusão . . . . .	141
<b>7</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>142</b>
7.1	Conclusões . . . . .	142
7.2	Trabalhos Futuros . . . . .	143

# Lista de Figuras

2.1	Exemplo de Disparo de transição. (a) Marcação antes do disparo de $t$ . (b) Marcação depois do disparo de $t$ . . . . .	29
2.2	Rede de Petri representando o problema de leitura e escrita. . . . .	30
2.3	Estrutura de Rede de Petri chamada conflito, escolha ou decisão. . . . .	30
2.4	Exemplo de disparo de transições em uma rede de alto nível. (a) antes do disparo. (b) Depois do disparo com substituição $\{a/x, b/y, c/z\}$ . (c) Depois do disparo com substituição $\{d/x, a/y, b/z\}$ . . . . .	36
2.5	Rede de Petri correspondente à rede de alto nível do exemplo de programação lógica. . . . .	36
3.1	Modelo de Rede de Petri Nebulosa para uma regra condicional. . . . .	43
3.2	Modelo de Rede de Petri Nebulosa Marcada representando o padrão de inferência básico. . . . .	44
3.3	RPN Marcada assumindo-se os conjuntos base $X = Y = \{1+2+3+4\}$ e o subconjunto nebuloso $A' = \{0.5/1 + 0.7/2 + 0.2/3\}$ . . . . .	46
3.4	Disparo de uma RPN Marcada. (a) Antes do disparo das transições $t_1, t_2, \dots, t_m$ . (b) Depois do disparo das transições $t_1, t_2, \dots, t_m$ . . . . .	49
3.5	RPN representando duas regras encadeadas. . . . .	50
3.6	RPN representando uma regra com conjunção no antecedente. Os pesos dos arcos foram omitidos. . . . .	52
3.7	RPN Marcada representando regras com conjunção no antecedente. . . . .	53

3.8	Disparo de uma RPN Marcada representando regras com conjunção no antecedente. (a) Após o disparo do primeiro grupo de transições. (b) Após o disparo do segundo grupo de transições. . . . .	54
3.9	Estrutura básica de RPN representando duas regras paralelas com $ X^i = Y =3$ , quando o princípio de combinação/projeção é realizado localmente. (a) Com transições do tipo 3. (b) Forma simplificada.	60
3.10	RPN representando regras conflitantes. (a) Regras com um único antecedente. (b) Regras com antecedentes múltiplos. . . . .	61
3.11	Representação hierárquica de regras paralelas. . . . .	66
3.12	RPN representando conjuntos encadeados de regras com transições do tipo 3. . . . .	67
3.13	RPN representando conjuntos encadeados de regras sem transições do tipo 3. . . . .	68
3.14	RPN representando conjuntos conflitantes de regras. . . . .	69
3.15	RPN representando a base de regras completa. . . . .	70
4.1	RPNAN representando uma regra condicional.(a) Antes do disparo das transições, onde $A' = \langle a'_1, a'_2, \dots, a'_n \rangle$ . (b) Depois do disparo das transições, onde $B' = \langle b'_1, b'_2, \dots, b'_m \rangle$ . . . . .	75
4.2	Exemplo de uma RPNAN. . . . .	77
4.3	RPNAN representando duas regras encadeadas. . . . .	79
4.4	RPNAN representando regras com antecedentes múltiplos. . . . .	80
4.5	RPNAN representando uma regra com duas condições no antecedente.	82
4.6	Estrutura básica de RPNAN representando regras paralelas quando o princípio de combinação/projeção é realizado localmente. . . . .	84
4.7	RPNAN com representação hierárquica para regras paralelas. (a) Com subrede interna. (b)Com bloco de regras paralelas. . . . .	84
4.8	Estrutura básica de RPNAN representando regras conflitantes. (a) Regras com um único antecedente.(b) Regras com antecedentes múltiplos.	85

4.9	RPNAN representando uma regra e dois fatos referentes à mesma variável. . . . .	87
4.10	RPNAN representado regras com a mesma variável no consequente. . . . .	88
4.11	RPNAN representando uma regra com conjunção no consequente. . . . .	90
4.12	Fluxo de produção de pistão, camisa e cabeçote . . . . .	93
4.13	RPNAN representando as regras da base de conhecimento do exemplo da manufatura. . . . .	100
5.1	Estrutura de árvores gerada pelo algoritmo de raciocínio para frente. . . . .	109
5.2	RPNAN representando um subconjunto das regras do exemplo de manufatura. . . . .	110
5.3	Sub-rede extraída pelo raciocínio de encadeamento para trás. . . . .	111
5.4	RPNAN representando a base de conhecimento do exemplo. . . . .	113
5.5	Subrede extraída pela primeira fase do raciocínio com encadeamento para trás. . . . .	115
5.6	Árvore de conhecimento contendo informação parcial sobre a variável $V_1$ . . . . .	116
5.7	Fórmulas inscritas na transição $t_4$ . . . . .	118
5.8	Fórmulas inscritas na transição $t_6$ . . . . .	119
5.9	Fórmulas inscritas na transição $t_7$ . . . . .	119
5.10	Subrede detalhada que substitui o lugar $p_7$ . . . . .	120
6.1	RPNAN representando a base de conhecimento do exemplo. . . . .	125
6.2	Representação hierárquica de regras com antecedentes múltiplos. (a)RPNAN com sub-rede detalhada. (b)RPNAN hierárquica com BAM. . . . .	126
6.3	Representação hierárquica de regras paralelas. (a)RPNAN com sub-redes detalhadas. (b)RPNAN com CRP. . . . .	126
6.4	Representação Hierárquica para fontes parciais de informação. (a)RPNAN com sub-rede detalhada. (b)RPNAN com BFP. . . . .	127

6.5 RPNAN nível1 representando a base de conhecimento do exemplo. . 128

# Lista de Tabelas

2.1	Valores linguísticos para a variável <i>altura</i> definidos como subconjuntos nebulosos de $X = \{150, 160, 170, 180, 190, 200\}$ . . . . .	13
2.2	Valores linguísticos para a variável <i>peso</i> definidos como subconjuntos nebulosos de $Y = \{60, 70, 80, 90, 100\}$ . . . . .	13
2.3	Valores de pertinência da relação <i>alto e pesado</i> . . . . .	14
2.4	Valores de pertinência da relação nebulosa <i>muito_alto ou leve</i> . . . . .	15
2.5	Valores de pertinência da relação nebulosa induzida pela regra SE-ENTÃO. . . . .	16
2.6	Valores linguísticos para a variável <i>tamanho</i> definidos como subconjuntos nebulosos de $W = \{1, 2, 3, 4, 5\}$ . . . . .	20
2.7	Relação induzida por uma regra com conjunção no antecedente. . . . .	21
3.1	Definição Formal de uma RPN Marcada. O elemento $\Gamma$ não está definido. . . . .	45
3.2	Exemplo de definição formal de uma RPN Marcada. . . . .	47
3.3	Exemplo de Regras Nebulosas para serem modeladas na RPN. . . . .	65
4.1	Valores linguísticos definidos como subconjuntos nebulosos. . . . .	78
4.2	Representação matricial da relação induzida por uma regra com antecedentes múltiplos. Em cada quadro, as linhas da matriz $5 \times 5$ representam elementos do conjunto base $X^1$ e as colunas, elementos do conjunto base $X^2$ . . . . .	81

4.3	Subconjuntos Nebulosos representando valores linguísticos. . . . .	95
4.4	Variáveis e seus significados. . . . .	96
4.5	Conjuntos Base . . . . .	96
4.6	Subconjunto da base de conhecimento do exemplo da manufatura. . .	97
4.7	Subconjunto da base de conhecimento do exemplo da manufatura. . .	98
5.1	Subconjuntos Nebulosos com rótulos <i>cerca de 60</i> e <i>cerca de 15</i> . . . . .	108
5.2	Matriz de incidência da rede de Petri associada a RPNAN do exemplo.	114
6.1	Exemplo de base de conhecimento. . . . .	125
6.2	Exemplo de matriz-TC. . . . .	129

# Capítulo 1

## Introdução

### 1.1 Formulação do Problema

Os Sistemas Baseados em Conhecimento possuem uma série de características que os diferenciam dos demais sistemas. Em função disso, apresentam problemas típicos a serem tratados durante as diversas fases de seu desenvolvimento, desde a aquisição de conhecimento até a implementação.

Nos sistemas de regras de produção a modularidade e uniformidade das regras são apontadas como algumas das vantagens desse formalismo. Entretanto esse esquema de representação traz o inconveniente de não oferecer nenhum tipo de estruturação da base de conhecimento. Assim as relações de dependência entre as informações não podem ser visualizadas claramente nem mesmo utilizadas no processo de raciocínio. Sendo a regra o único elemento básico de representação, o projeto modular da base de conhecimento e a posterior junção desses módulos nem sempre podem ser feitos adequadamente. A ausência de recursos para organizar hierarquicamente os elementos do mecanismo de representação limita a possibilidade de estruturação lógica das informações. Além disso, esquemas de representação do conhecimento em geral priorizam o poder expressivo, negligenciando métodos de verificação de integridade das bases de conhecimento. Isso dificulta a automatização de ferramentas para verificação, um componente frequentemente apontado como desejável em ambientes de desenvolvimento de sistemas baseados em conhecimento.

No âmbito de sistemas de regras nebulosas, especificamente, todas essas questões são também pertinentes, além de outras decorrentes das peculiaridades da área. A representação de conceitos imprecisos em termos de subconjuntos nebulosos aumenta a complexidade dos problemas. A representação das regras nebulosas exclusivamente

na forma SE-ENTÃO não prevê meios de explicitar o método e os operadores utilizados na aplicação da inferência. No caso da regra de inferência composicional, um grande número de cálculos numéricos deve ser executado quando essa regra é aplicada. Esses cálculos são usualmente representados matricialmente, em estruturas que não permitem ver claramente as relações entre proposições. Assim como para regras de produção convencionais, estudos sobre mecanismos para verificação não se equiparam àqueles desenvolvidos para outras áreas. O tratamento adequado de problemas como esses é essencial para se obter bons resultados em aplicações de sistemas nebulosos. Portanto, a evolução de sistemas nebulosos e a crescente aplicação dos mesmos em áreas diversas tem evidenciado a necessidade de ferramentas formais que ofereçam uma base uniforme para o tratamento dos problemas relacionados aqui.

## 1.2 Objetivos

Neste trabalho são propostos dois modelos formais para a representação de regras de produção nebulosas, baseados em teoria de redes. O objetivo dessa proposta é obter mecanismos de representação poderosos e ao mesmo tempo tratáveis, que forneçam meios de atacar os problemas relacionados na seção anterior, encontrados durante o desenvolvimento de sistemas nebulosos. O primeiro modelo, baseado em Redes de Petri Nebulosas, permite representar detalhadamente os cálculos numéricos que dão suporte à regra de inferência dentro do Raciocínio Aproximado. O segundo, chamado de modelo de Redes de Petri Nebulosas de Alto Nível, oferece uma maneira mais compacta de representação das regras nebulosas. Os dois modelos guardam uma relação muito próxima entre si, o que torna possível transpor de um para outro conforme necessário. Portanto é possível acessar as informações a partir de diferentes níveis de abstração, tal que seja evidenciado ora o aspecto linguístico, ora o aspecto numérico do raciocínio nebuloso. A representação hierárquica das estruturas, onde uma determinada parcela da rede é substituída por um único elemento, também é possível. Isso reforça significativamente a versatilidade do modelo durante o projeto de sistemas. As regras passam a ter uma representação estruturada. Assim as relações de dependência ficam evidentes e podem ser usadas nos processos de raciocínio e verificação.

### 1.3 Desenvolvimento da Tese

O conteúdo deste trabalho está organizado da seguinte forma. No capítulo 2 são introduzidos alguns conceitos básicos para o desenvolvimento dessa tese, relativos às áreas de Raciocínio Aproximado e Redes de Petri. O modelo de Redes de Petri Nebulosas proposto para representação de regras nebulosas será apresentado no capítulo 3. As técnicas de modelagem de diversos tipos e conjuntos de regras são abordadas. No capítulo 4 será apresentado o modelo baseado em Redes de Petri Nebulosas de Alto Nível, proposto para representar uma base de regras nebulosas em um nível de abstração mais alto que as Redes de Petri Nebulosas. A aplicação dessa ferramenta a uma base de conhecimento relacionada a sistemas de manufatura também é discutida. No capítulo 5, serão apresentados dois tipos de algoritmos para raciocínio aproximado. O algoritmo de raciocínio com encadeamento para frente é descrito com base no modelo de Redes de Petri Nebulosas. O algoritmo de raciocínio com encadeamento para trás é apresentado com base no modelo de Redes de Petri Nebulosas de Alto Nível. A aplicação desses algoritmos a alguns exemplos é abordada. A utilização dos modelos de Redes de Petri de Alto Nível para a verificação de sistemas baseados em regras é discutida no capítulo 6, onde um procedimento para detectar a existência de conflitos em potencial entre regras nebulosas é sugerido. As conclusões e os trabalhos futuros são, finalmente, discutidos no capítulo 7.

# Capítulo 2

## Conceitos Básicos

### 2.1 Introdução

Neste capítulo serão revistos os conceitos encontrados na literatura que são utilizados neste trabalho. Estes conceitos estão relacionados a duas grandes áreas do conhecimento, a saber, Raciocínio Aproximado e Redes de Petri. O objetivo desta apresentação não é rever completa e exaustivamente todas as linhas de pesquisa relacionadas a estas duas áreas, mas sim apresentar de forma resumida os aspectos mais relevantes, necessários ao entendimento do trabalho desenvolvido nos capítulos seguintes. Procurou-se entretanto introduzir os conceitos gradativamente, para facilitar o entendimento do assunto e permitir que leitores não muito familiarizados com essas áreas possam adquirir o embasamento necessário à compreensão do trabalho proposto.

Na seção 2.2 são apresentados os conceitos básicos de subconjuntos nebulosos e lógica nebulosa, além de uma introdução ao Raciocínio Aproximado com suas regras de representação e de inferência. Além das proposições condicionais da forma SE-ENTÃO, são apresentadas as proposições relacionais. A representação generalizada de métodos de raciocínio baseada em T-operadores é também abordada. Na seção 2.3 são definidos os principais tipos de Redes de Petri — comuns e de alto nível—, que serviram de base para o desenvolvimento dos modelos aqui sugeridos.

## 2.2 Introdução ao Raciocínio Aproximado

A afirmação de que o conhecimento humano é muitas vezes incompleto, incerto ou impreciso tem sido encontrada com frequência em diversas áreas de Inteligência Artificial preocupadas com formalismos de representação e raciocínio que permitam o tratamento apropriado a cada tipo de problema. A teoria de Raciocínio Aproximado em particular, fornece um método, baseado em subconjuntos nebulosos, tanto para representar como para raciocinar com informação imprecisa [82] [84] [85] [76]. A seguir são apresentados os fundamentos dessa teoria que fornece a base dos métodos de inferência generalizados representados nos modelos baseados em Redes de Petri propostos neste trabalho.

### 2.2.1 Subconjuntos Nebulosos

A teoria de subconjuntos nebulosos proposta inicialmente por Zadeh [79] em 1965 e desenvolvida por vários outros autores [30] oferece uma ferramenta bastante poderosa para se representar diferentes tipos de conhecimento humano. Na lógica convencional a classificação de um objeto qualquer segundo um certo critério é feita em uma entre duas categorias pré-determinadas, como por exemplo, par-ímpar, bom-mau, preto-branco, falso-verdadeiro. Entretanto, em diversas situações deparamo-nos com classes que não são definidas precisamente, como a classe dos homens altos, a classe de erros significativos, classe das matrizes esparsas e muitas outras. Esses conceitos, apesar de imprecisos, têm um significado óbvio considerando-se um determinado ambiente, e são usados em situações reais para se raciocinar e tomar decisões racionais. Na teoria de subconjuntos nebulosos a noção básica de conjunto é modificada, permitindo que os valores de pertinência de cada elemento ao conjunto variem no intervalo  $[0, 1]$ .

Formalmente, um subconjunto nebuloso  $A$  é definido [79] [30] em um universo de discurso (conjunto base)  $X$  e é caracterizado pela sua função característica

$$A : X \mapsto [0, 1]$$

onde  $A(x)$  expressa a extensão com que  $x$  se enquadra na categoria representada por  $A$ .

Um tópico de grande interesse na teoria de subconjuntos nebulosos é a seleção de operadores apropriados para a intersecção, união e complemento (ou negação) de subconjuntos, que representem uma generalização para o ambiente nebuloso dos operadores de conjuntos clássicos. Essas operações foram originalmente definidas como

- Intersecção:  $A \cap B = C$ , onde

$$C(x) = \min(A(x), B(x)) = A(x) \wedge B(x)$$

- União:  $A \cup B = D$ , onde

$$D(x) = \max(A(x), B(x)) = A(x) \vee B(x)$$

- Complemento:  $\bar{A}$ , onde

$$\bar{A}(x) = 1 - A(x)$$

Na tentativa de se determinar outros operadores adequados a intersecção, união e complemento além dos convencionais (min, max e complemento de um), pode-se considerar certas propriedades formais desejáveis desses operadores [77] [17] [48] [25]. As propriedades básicas geralmente esperadas desses operadores são comutatividade, associatividade e monotonicidade. Quando o conjunto base é o intervalo  $[0, 1]$ , as três propriedades acima junto com a independência ponto a ponto e mais a exigência de incluir o caso convencional caracterizam os operadores de intersecção e união. Uma classe de modelos para os conectivos lógicos de intersecção é formada pelas normas triangulares.

**Definição 2.2.1** *Um operador  $T : [0, 1] \times [0, 1] \mapsto [0, 1]$  é chamado um operador T-norma se*

- (i) *é não decrescente ponto a ponto*

$$T(a, b) \geq T(c, d) \text{ se } a \geq c \text{ e } b \geq d$$

- (ii) *é comutativo*

$$T(a, b) = T(b, a)$$

- (iii) *é associativo*

$$T(a, T(b, c)) = T(T(a, b), c)$$

- (iv) *satisfaz as condições limite*

$$T(0, a) = 0 \text{ e } T(a, 1) = a$$

O operador de união é generalizado pelo operador T-conorma ou S-norma.

**Definição 2.2.2** Um operador  $S : [0, 1] \times [0, 1] \mapsto [0, 1]$  é chamado operador T-conorma ou S-norma se

(i) é não decrescente ponto a ponto

$$S(a, b) \geq S(c, d) \text{ se } a \geq c \text{ e } b \geq d$$

(ii) é comutativo

$$S(a, b) = S(b, a)$$

(iii) é associativo

$$S(a, S(b, c)) = S(S(a, b), c)$$

(iv) satisfaz as condições limite

$$S(a, 0) = a \text{ e } S(a, 1) = 1$$

Pode ser mostrado [77] que o operador *min* é a maior de todas as T-normas possíveis e o operador *max* é a menor de todas as S-normas.

As propriedades dos operadores de negação ou complemento de subconjuntos nebulosos definem a classe das funções de negação.

**Definição 2.2.3** Um operador  $N : [0, 1] \mapsto [0, 1]$  é chamado um operador de negação se

(i)  $N(1) = 0$  ;  $N(0) = 1$

(ii) é involutivo

$$N(N(a)) = a$$

(iii) é de ordem reversa

$$\text{se } a > b \text{ então } N(a) \leq N(b)$$

A lei de DeMorgan, válida na teoria de conjuntos, pode ser generalizada para subconjuntos nebulosos. Seja  $T$  uma T-norma qualquer. Então o operador  $S$  definido por

$$S(a, b) = 1 - T(\bar{a}, \bar{b})$$

é uma S-norma, chamada dual de  $T$ .

Os operadores *max* e *min* são operadores duais.

Outro exemplo importante de T-norma é o operador de produto

$$T(a, b) = a \cdot b$$

e seu dual é a S-norma

$$S(a, b) = a + b - ab.$$

Outros exemplos de T-normas e de S-normas são dados a seguir.

$$T(x, y) = \frac{xy}{x + y - xy}$$

$$T(x, y) = \max(1 - ((1 - x)^p + (1 - y)^p)^{\frac{1}{p}}, 0)$$

$$S(x, y) = \frac{x + y - 2xy}{1 - xy}$$

$$S(x, y) = \min((x^p + y^p)^{\frac{1}{p}}, 1)$$

É importante observar a diferença entre o operador de negação e de antônimo de um subconjunto nebuloso. O antônimo relaciona conceitos como alto-baixo, grande-pequeno, velho-jovem, enquanto que a negação relaciona alto-não-alto, baixo-não-baixo, velho-não-velho. Se  $A$  é um subconjunto nebuloso definido sobre o conjunto base  $X$  e  $X$  é um conjunto ordenado

$$X = \{x_1, x_2, \dots, x_r\}$$

onde  $x_j > x_i$  quando  $j > i$  então o antônimo  $\hat{A}$  pode ser definido [77] por

$$\hat{A}(x_j) = A(x_{r-j+1}).$$

Outras propriedades podem ser associadas aos operadores de união e intersecção. Cada par de operadores duais pode ou não possuir essas propriedades. A propriedade de *idempotência* exige que

$$T(a, a) = a \text{ e } S(a, a) = a.$$

A propriedade da *lei do meio excluído*:

$$A \cup \bar{A} = X$$

e o princípio da não contradição:

$$A \cap \bar{A} = \emptyset$$

válidas na teoria de conjuntos convencional, geralmente não são satisfeitas pelos conectivos nebulosos.

Outra propriedade de interesse é a distributividade mútua:

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

No ambiente nebuloso, as propriedades de distributividade mútua e idempotência vêm juntas e se opõem à lei do meio excluído e da não contradição [17]. Logo, na seleção de um operador, deve-se escolher entre esses pares de propriedades.

Outros tópicos de interesse para esse trabalho relativos a subconjuntos nebulosos são os métodos para comparação de quantidades nebulosas. Duas medidas de proximidade entre subconjuntos nebulosos conhecidas são as medidas de possibilidade e necessidade (ou certeza) [82].

**Definição 2.2.4** *Sejam  $A$  e  $B$  subconjuntos nebulosos do conjunto base  $X$ . A possibilidade de  $A$  com relação a  $B$  é definida como*

$$Poss(A/B) = \max_{x \in X} [A(x) \wedge B(x)].$$

**Definição 2.2.5** *Sejam  $A$  e  $B$  subconjuntos nebulosos do conjunto base  $X$ . A certeza (necessidade) de  $A$  com relação a  $B$  é definida como*

$$Cert(A/B) = \min_{x \in X} [A(x) \vee (1 - B(x))].$$

As medidas de possibilidade e certeza se relacionam pela expressão

$$Cert(A/B) = 1 - Poss(\bar{A}/B).$$

A medida de possibilidade de  $A$  com relação a  $B$  reflete a extensão com que  $A$  e  $B$  se interceptam. A medida de certeza representa o grau com que  $B$  está contido em  $A$ .

Essas definições originais podem ser estendidas [48] [18] substituindo-se mínimo e máximo por T-norma e S-norma, respectivamente:<sup>1</sup>

$$Poss(A/B) = \sup_{x \in X} [A(x) \star B(x)]$$

$$Cert(A/B) = \inf_{x \in X} [A(x) \mathbf{s}(1 - B(x))]$$

O conceito de relação pode ser naturalmente estendido a subconjuntos nebulosos e tem um papel importante na teoria e aplicação dos mesmos. Uma relação nebulosa  $R$ , definida sobre o produto cartesiano  $X \times Y$  onde  $X$  e  $Y$  são conjuntos base, é um mapeamento tal que

$$R : X \times Y \mapsto [0, 1]$$

Logo, a cada par de elementos  $(x, y)$ ,  $x \in X$  e  $y \in Y$ , é atribuído um número no intervalo  $[0, 1]$ . Esse número representa o grau de ligação entre os elementos do par. Um exemplo simples é o conceito de *aproximadamente igual*.

**Exemplo 2.2.1** *Assuma  $X = Y = \{1 + 2 + 3 + 4\}$ . Seja  $R$  a relação  $x$  é aproximadamente igual a  $y$ . Esta relação pode ser definida<sup>2</sup> por*

$$R = \{1/(1,1) + 0.5/(1,2) + 0/(1,3) + 0/(1,4) + 0.5/(2,1) + 1/(2,2) + 0.5/(2,3) + 0/(2,4) + 0/(3,1) + 0.5/(3,2) + 1/(3,3) + 0.5/(3,4) + 0/(4,1) + 0/(4,2) + 0.5/(4,3) + 1/(4,4)\}$$

*Para conjuntos base finitos, a notação em forma de matriz para relações nebulosas é bastante útil. Neste caso a relação  $R$  é representada como uma matriz  $[R(x_i, y_j)]$  tal que*

$$R(x_i, y_j) = \begin{bmatrix} 1 & 0.5 & 0 & 0 \\ 0.5 & 1 & 0.5 & 0 \\ 0 & 0.5 & 1 & 0.5 \\ 0 & 0 & 0.5 & 1 \end{bmatrix}$$

A operação de produto cartesiano pode ser facilmente estendida para subconjuntos nebulosos. Sejam  $A$  e  $B$  subconjuntos nebulosos sobre os conjuntos base  $X$  e  $Y$ , respectivamente. O *produto cartesiano*  $A \times B$  é definido como uma relação nebulosa, tal que

$$A \times B(x, y) = \min[A(x), B(y)].$$

<sup>1</sup>Denota-se o operador T-norma por  $\star$  e o operador S-norma por  $\mathbf{s}$ .

<sup>2</sup>Adota-se aqui a notação de subconjuntos nebulosos onde  $+$  indica união e o elemento da forma  $a/b$  significa que o grau de pertinência do elemento  $b$  no conjunto é  $a$ .

## 2.2.2 Lógica Nebulosa

A lógica nebulosa pode ser vista em parte como uma extensão da lógica de valores múltiplos [84] [82]. É a lógica que trata de modelos de raciocínio aproximado. O poder expressivo da lógica nebulosa deriva do fato de conter como casos especiais não só os sistemas lógicos binários e de valores múltiplos, mas também teoria de probabilidades e lógica probabilística.

As principais características da lógica nebulosa, que a diferencia das lógicas tradicionais são [83] [80]

1. Os valores verdade podem ser subconjuntos nebulosos de um conjunto base  $T$ , usualmente o intervalo  $[0,1]$ , e denotados por termos linguísticos como *verdadeiro*, *muito verdadeiro*, *mais ou menos verdadeiro*, *não muito falso*, etc.
2. Os predicados podem ser precisos como na lógica clássica (*mortal*, *par*, *pai-de*), ou imprecisos (*cansado*, *grande*, *muito\_mais\_pesado\_que*, *amigo\_de*).
3. Os quantificadores podem ser de vários tipos como *a maioria*, *muitos*, *vários*, *frequentemente*, *cerca de 10*, *pelo menos 7*.
4. Os modificadores de predicado tais como *não*, *muito*, *mais\_ou\_menos*, *extremamente*, *levemente*, podem ser também representados.
5. As proposições podem ser qualificadas de três formas diferentes:
  - qualificação de verdade, como em  
(Maria é jovem) não é bem verdade.
  - qualificação de probabilidade, como em  
(Maria é jovem) é pouco provável.
  - qualificação de possibilidade, como em  
(Maria é jovem) é quase impossível.

A lógica nebulosa FL proposta por Zadeh [84] fornece a base para a teoria de raciocínio aproximado.

### 2.2.3 Raciocínio Aproximado

A teoria de Raciocínio Aproximado (RA) oferece um método baseado na teoria de subconjuntos nebulosos, para representar e raciocinar com informação imprecisa [84] [82] [76] [71]. Nesta seção serão apresentados alguns conceitos básicos do Raciocínio Aproximado. Considere as informações

A temperatura do paciente está alta,  
João é jovem,  
O tempo de processamento da operação é pequeno.

Pode-se dizer que declarações como essas tem a seguinte forma geral

O (atributo) do (objeto) é (valor).

As idéias de atributo e objeto podem ser combinadas no conceito de *variável* [71].

Assuma que  $V$  é uma variável que pode assumir valores no conjunto  $X$ . O conjunto  $X$  é chamado *conjunto base* ou *universo de discurso* de  $V$  e deve poder ser medido de alguma maneira. Ele fornece o conjunto de valores sobre o qual  $V$  pode assumir o seu valor. Por exemplo, para a variável *Idade de João*, o conjunto  $X$  pode ser  $X = \{1, 2, \dots, 100\}$ . Na teoria de Raciocínio Aproximado, o valor de  $V$  não precisa ser necessariamente um elemento de  $X$ . Os valores de tais variáveis podem ser valores linguísticos como *muito jovem*, *razoavelmente alto*. Variáveis que assumem valores linguísticos são chamadas *variáveis linguísticas*. O valor linguístico de uma variável  $V$  pode ser definido por um subconjunto nebuloso sobre o conjunto base  $X$  de  $V$ . No exemplo acima, o valor linguístico *jovem* pode ser definido como

$jovem = \{1/1+1/2+\dots+1/25+0.9/26+0.8/27+\dots+0.5/30+\dots+0/45+\dots+0/100\}$

**Definição 2.2.6** *Seja  $V$  uma variável sobre  $X$ . Se  $L$  é um valor linguístico definido em termos do subconjunto nebuloso  $A$  de  $X$ , diz-se que  $L$  é o rótulo de  $A$ .*

**Definição 2.2.7** *Uma proposição canônica é uma expressão da forma*

$V$  é  $A$

*onde  $A$  é um subconjunto nebuloso de  $X$ .*

<i>V (altura)</i>	<i>conjunto base X (em cm)</i>					
	150	160	170	180	190	200
<i>baixo</i>	1.0	0.8	0.2	0.0	0.0	0.0
<i>alto</i>	0.0	0.0	0.2	0.5	1.0	1.0
<i>muito_alto</i>	0.0	0.0	0.0	0.2	1.0	1.0

Tabela 2.1: Valores linguísticos para a variável *altura* definidos como subconjuntos nebulosos de  $X = \{150, 160, 170, 180, 190, 200\}$ .

<i>U (peso)</i>	<i>conjunto base Y (em Kg)</i>				
	60	70	80	90	100
<i>leve</i>	1.0	0.5	0.1	0.0	0.0
<i>meio_leve</i>	0.0	0.5	1.0	0.5	0.0
<i>pesado</i>	0.0	0.0	0.1	0.8	1.0

Tabela 2.2: Valores linguísticos para a variável *peso* definidos como subconjuntos nebulosos de  $Y = \{60, 70, 80, 90, 100\}$ .

**Exemplo 2.2.2** Considere como conjunto base o conjunto de alturas (em cm)  $X = \{150, 160, 170, 180, 190, 200\}$  e o conjunto de pesos (em kg)  $Y = \{60, 70, 80, 90, 100\}$ . Suponha que  $V$  é uma variável sobre  $X$  e  $U$  é uma variável sobre  $Y$ . A Tabela 2.1 mostra possíveis valores linguísticos para a variável  $V$  e suas definições como subconjuntos nebulosos de  $X$ . A Tabela 2.2 mostra possíveis valores linguísticos para  $U$  e suas definições como subconjuntos nebulosos de  $Y$ .

## Regras de Tradução

A teoria de Raciocínio Aproximado inclui dois componentes, que são as regras de tradução e as regras de inferência. As regras de tradução determinam como uma informação é traduzida na representação apropriada em termos de relações nebulosas, fornecendo portanto uma representação quantitativa da informação dada. As regras de inferência fornecem um meio de manipular a informação representada para se obter novas informações. Assuma que temos duas proposições  $V_1$  é  $A$  and  $V_2$  é  $B$ , onde  $A$  e  $B$  são subconjuntos nebulosos de  $X$  e  $Y$ , respectivamente. Uma das regras de tradução do RA determina que a conjunção

$$V_1 \text{ é } A \text{ e } V_2 \text{ é } B$$

induz uma relação nebulosa  $D$  sobre o conjunto  $X \times Y$  denotada por

$$(V_1, V_2) \text{ é } D,$$

onde  $(V_1, V_2)$  é chamada de *variável conjunta*. A disjunção

$$V_1 \text{ é } A \text{ ou } V_2 \text{ é } B$$

também induz uma relação  $E$  sobre o mesmo conjunto  $X \times Y$ .

A conjunção de proposições é convencionalmente traduzida na teoria de RA de Zadeh [84] através do operador de intersecção, como

$$D_{A,B}(x, y) = \min[A(x), B(y)] = A(x) \wedge B(y)$$

e a disjunção através do operador de união, como

$$D_{A,B}(x, y) = \max[A(x), B(y)] = A(x) \vee B(y).$$

**Exemplo 2.2.3** Considerando-se os conjuntos base  $X$  e  $Y$  e as variáveis  $V$  e  $U$  do exemplo 2.2.2, a conjunção

$$V \text{ é alto e } U \text{ é pesado}$$

induz uma relação  $D_{A,B}(x, y)$  sobre  $X \times Y$  dada pelos valores apresentados na Tabela 2.3.

<i>alto e pesado</i>	<i>conjunto base Y (em Kg)</i>				
<i>conjunto base X (em cm)</i>	60	70	80	90	100
150	0	0	0	0	0
160	0	0	0	0	0
170	0	0	0.1	0.2	0.2
180	0	0	0.1	0.5	0.5
190	0	0	0.1	0.8	1.0
200	0	0	0.1	0.8	1.0

Tabela 2.3: Valores de pertinência da relação *alto e pesado*.

*A disjunção*

<i>muito_alto ou leve</i>	<i>conjunto base Y (em Kg)</i>				
<i>conjunto base X (em cm)</i>	60	70	80	90	100
150	1.0	0.5	0.1	0	0
160	1.0	0.5	0.1	0	0
170	1.0	0.5	0.1	0	0
180	1.0	0.5	0.2	0.2	0.2
190	1.0	1.0	1.0	1.0	1.0
200	1.0	1.0	1.0	1.0	1.0

Tabela 2.4: Valores de pertinência da relação nebulosa *muito\_alto ou leve*.

$V$  é *muito\_alto* ou  $U$  é *leve*

induz uma relação  $D_{A,B}(x, y)$  dada pelos valores da Tabela 2.4.

Entre as proposições nebulosas compostas de várias variáveis, a mais usada e estudada é a proposição de implicação, que tem a forma

SE  $V$  é  $A$  ENTÃO  $U$  é  $B$ .

Proposições dessa forma são chamadas neste contexto de *regras de produção nebulosas* [43], ou simplesmente *regras*. Pelas regras de tradução, esta proposição induz uma relação nebulosa sobre  $X \times Y$ , convencionalmente definida na teoria de Zadeh por

$$D_{B/A}(x, y) = \min[1, 1 - A(x) + B(y)].$$

Essa expressão é conhecida como *soma limitada* e frequentemente denotada por  $\oplus$ .

**Exemplo 2.2.4** Considerando-se as variáveis  $V$  e  $U$  e os conjuntos base  $X$  e  $Y$  do exemplo 2.2.2, a regra

SE  $V$  é *alto* ENTÃO  $U$  é *pesado*

induz a relação  $D_{B/A}(x, y)$  dada pelos valores apresentados na Tabela 2.5.

Muitos outros métodos foram sugeridos para a representação de  $D$  [33] [35] [53] [5] [46] [71]. Alguns serão apresentados na seção 2.2.5.

$D_{B/A}(x, y)$	conjunto base $Y$ (em Kg)				
conjunto base $X$ (em cm)	60	70	80	90	100
150	1.0	1.0	1.0	1.0	1.0
160	1.0	1.0	1.0	1.0	1.0
170	0.8	0.8	0.9	1.0	1.0
180	0.5	0.5	0.6	1.0	1.0
190	0	0	0.1	0.8	1.0
200	0	0	0.1	0.8	1.0

Tabela 2.5: Valores de pertinência da relação nebulosa induzida pela regra SE-ENTÃO.

### Regras de Inferência

As regras de inferência do RA oferecem um meio de obter novas relações nebulosas a partir de relações nebulosas conhecidas. A primeira regra de inferência do RA é chamada de *princípio da consequência*. Este princípio permite inferir uma relação nebulosa mais restrita a partir de outra, dada como valor de uma variável  $V$ . A partir da proposição

$$V \text{ é } A$$

pode-se inferir

$$V \text{ é } B$$

para todo  $B$  tal que  $B(x) \geq A(x), \forall x, x \in X$ . Consequentemente é sempre possível inferir  $V \text{ é } X$ , onde  $X$  é o conjunto base de  $A$ .

A segunda regra de inferência é chamada de *princípio de projeção*. Ela diz que a partir da relação  $H(x, y)$  sobre o conjunto  $X \times Y$  pode-se inferir as relações marginais

$$E(x) = \max_{y \in Y} [H(x, y)]$$

e

$$F(y) = \max_{x \in X} [H(x, y)].$$

A terceira regra de inferência da teoria de RA é chamada de *princípio de conjunção*. Através dessa regra, dadas duas relações  $H_1(x)$  e  $H_2(y)$  sobre  $X$  e  $Y$  res-

pectivamente, obtém-se

$$H(x, y) = \min(H_1(x), H_2(y)).$$

A aplicação conjunta dos princípios de conjunção e projeção resulta na regra chamada *inferência composicional nebulosa*. Essa regra determina que, quando se tem as relações  $A(x)$  sobre o conjunto base  $X$  e  $H(x, y)$  sobre o conjunto  $X \times Y$  pode-se inferir a relação  $B(y)$  sobre  $Y$  tal que

$$B(y) = \max_{x \in X} [A(x) \wedge H(x, y)].$$

Uma importante aplicação dessa regra ocorre quando temos o seguinte tipo de informação:

$$\begin{array}{l} V \text{ é } A' \\ \text{SE } V \text{ é } A \text{ ENTÃO } U \text{ é } B \end{array}$$

onde  $A$  e  $A'$  são subconjuntos nebulosos sobre o conjunto base  $X$ ,  $V$  é uma variável sobre  $X$ ,  $B$  é um subconjunto nebuloso sobre o conjunto base  $Y$  e  $U$  é uma variável sobre  $Y$ . Proposições como  $V \text{ é } A'$  são chamadas neste contexto de *fatós*. Neste caso a inferência é chamada *modus ponens generalizado* ou *modus ponens composicional* e permite inferir proposições da forma  $U \text{ é } B'$ , onde

$$B' = A' \circ D_{B/A},$$

tal que

$$B'(y) = \max [A'(x) \wedge D_{B/A}(x, y)]. \quad (2.1)$$

$B'$  é um subconjunto nebuloso de  $Y$  e  $D_{B/A}$  é a relação induzida pela regra.

O resultado da expressão em (2.1) é encontrado facilmente pela aplicação da composição max-min nas matrizes das relações  $A'$  e  $D_{B/A}$ .

**Exemplo 2.2.5** *Considere-se a regra do exemplo 2.2.4. Assumindo-se conhecida a proposição*

$V$  é muito alto

e executando-se a composição max-min<sup>3</sup> das matrizes de relação correspondentes obtém-se

$$\begin{bmatrix} 0 & 0 & 0 & 0.2 & 1 & 1 \end{bmatrix} \circ \begin{bmatrix} 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0.8 & 0.8 & 0.9 & 1.0 & 1.0 \\ 0.5 & 0.5 & 0.6 & 1.0 & 1.0 \\ 0 & 0 & 0.1 & 0.8 & 1.0 \\ 0 & 0 & 0.1 & 0.8 & 1.0 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.2 \\ 0.2 \\ 0.8 \\ 1.0 \end{bmatrix}$$

O subconjunto nebuloso resultante  $B' = \{0.2/60 + 0.2/70 + 0.2/80 + 0.8/90 + 1.0/100\}$  pode ser interpretado linguisticamente como pesado.

Considerando-se a proposição

$V$  é baixo

o resultado da composição max-min é o subconjunto  $B' = \{1/60 + 1/70 + 1/80 + 1/90 + 1/100\}$ , que é o próprio conjunto base  $Y$  e pode ser interpretado como o valor linguístico desconhecido. Note que o valor baixo é o antônimo de alto (ver seção 2.2.1).

Nos exemplos apresentados, os conjuntos resultantes da composição max-min foram interpretados como valores linguísticos. O problema de encontrar um valor linguístico para  $U$  cujo significado mais se aproxima de um dado subconjunto nebuloso de  $Y$  é chamado de *aproximação linguística* [81]. As possíveis abordagens para tratamento deste importante problema não serão discutidas neste trabalho. As propriedades que o conjunto resultante  $B'$  deve apresentar em função da relação entre  $A$  e  $A'$  serão rapidamente discutidas na seção 2.2.5.

De forma geral, se  $V_1, V_2, \dots, V_k$  são variáveis que assumem valores nos conjuntos base  $X^1, X^2, \dots, X^k$  respectivamente, então a conjunção

$$V_1 \text{ é } A_1 \text{ e } V_2 \text{ é } A_2 \text{ e } \dots \text{ e } V_k \text{ é } A_k$$

induz uma relação  $H$  sobre  $X^1 \times X^2 \times \dots \times X^k$  tal que

$$H(x^1, x^2, \dots, x^k) = \min_{i=1, \dots, k} [A_i(x^i)]. \quad (2.2)$$

A proposição

---

<sup>3</sup>A composição max-min de duas matrizes é semelhante ao produto de duas matrizes, substituindo-se as operações de soma e multiplicação por máximo e mínimo, respectivamente.

$$V_1 \text{ é } A_1 \text{ ou } V_2 \text{ é } A_2 \text{ ou } \dots \text{ ou } V_k \text{ é } A_k$$

induz uma relação  $H$  sobre  $X^1 \times X^2 \times \dots \times X^k$  tal que

$$H(x^1, x^2, \dots, x^k) = \max_{i=1, \dots, k} [A_i(x^i)].$$

Se além disso  $U$  for uma variável com conjunto base  $Y$ , a regra

$$\text{SE } V_1 \text{ é } A_1 \text{ e } V_2 \text{ é } A_2 \text{ e } \dots \text{ e } V_k \text{ é } A_k \text{ ENTÃO } U \text{ é } B$$

induz uma relação  $F$  sobre  $X^1 \times X^2 \times \dots \times X^k \times Y$  tal que

$$F(x^1, x^2, \dots, x^k, y) = \min(1, 1 - H(x^1, x^2, \dots, x^k) + B(y))$$

onde  $H$  é dado por (2.2). Para fins de simplificação, assume-se que  $|X^1| = |X^2| = \dots = |X^k| = n$ . Assim, na forma matricial,  $H$  é uma matriz  $k$ -dimensional de ordem  $n^k$  e  $F$  como uma matriz  $(k+1)$ -dimensional de ordem  $n^{k+1}$ .

Sejam  $U_1, U_2, \dots, U_p$  variáveis com conjuntos base  $Y^1, Y^2, \dots, Y^p$ , respectivamente.

A regra

$$\text{SE } V_1 \text{ é } A_1 \text{ e } V_2 \text{ é } A_2 \text{ e } \dots \text{ e } V_k \text{ é } A_k \text{ ENTÃO } U_1 \text{ é } B_1 \text{ ou } U_2 \text{ é } B_2 \text{ ou } \dots \text{ ou } U_p \text{ é } B_p \quad (2.3)$$

induz a relação nebulosa sobre  $X^1 \times X^2 \times \dots \times X^k \times Y^1 \times \dots \times Y^p$  tal que

$$F(x^1, \dots, x^k, y^1, \dots, y^p) = \min(1, 1 - H(x^1, x^2, \dots, x^k) + G(y^1, \dots, y^p))$$

onde  $H$  é dado por (2.2) e

$$G(y^1, \dots, y^p) = \max_{i=1, \dots, p} [B_i(y^i)].$$

**Exemplo 2.2.6** *Sejam  $V_1$  e  $V_2$  variáveis sobre os conjuntos base  $X$  e  $Y$  do exemplo 2.2.2, respectivamente. Seja  $U$  uma variável sobre o conjunto base  $W = \{1, 2, 3, 4, 5\}$ . Alguns dos possíveis valores linguísticos para  $U$  e os conjuntos nebulosos correspondentes são apresentados na Tabela 2.6.*

*Considere a regra*

$$\text{SE } V_1 \text{ é alto e } V_2 \text{ é pesado ENTÃO } U \text{ é grande.}$$

$U$ (tamanho)	conjunto base $W$				
	1	2	3	4	5
<i>pequeno</i>	1	0.6	0.1	0	0
<i>médio</i>	0	0.5	1	0.5	0
<i>grande</i>	0	0	0.1	0.8	1

Tabela 2.6: Valores linguísticos para a variável *tamanho* definidos como subconjuntos nebulosos de  $W = \{1, 2, 3, 4, 5\}$ .

A conjunção do antecedente  $V_1$  é alto e  $V_2$  é pesado induz a relação nebulosa apresentada no exemplo 2.2.3. Logo, a regra acima induz uma relação nebulosa sobre  $X \times Y \times W$  dada pelos valores apresentados na Tabela 2.7, calculada por

$$F(x, y, w) = 1 \wedge [1 - D_{\text{alto, pesado}}(x, y) + \text{grande}(w)].$$

*Conhecidos os valores*

$V$  é alto e  $U$  é meio leve

a composição max-min resulta em  $B' = \{0.5/1 + 0.5/2 + 0.5/3 + 0.9/4 + 1.0/5\}$ , que pode ser interpretado como quase grande.

### Inferência com Proposições Relacionais

Além das proposições condicionais discutidas na seção anterior, outro tipo de proposição usada para representar informação são as chamadas proposições relacionais, como por exemplo [72] [84]

João e Maria tem quase a mesma idade,  
 $X$  e  $Y$  são aproximadamente iguais.

Proposições relacionais também induzem relações nebulosas sobre o produto cartesiano dos conjuntos base (veja seção 2.2.1). Por exemplo, sejam  $V_1$  e  $V_2$  duas variáveis sobre o mesmo conjunto base  $X = \{1+2+3+4\}$  e considere as proposições

$V_1$  e  $V_2$  são aproximadamente iguais  
 $V_1$  está bem próximo de 3

$w = 1$						
		$y$				
		60	70	80	90	100
$x$	150	1	1	1	1	1
	160	1	1	1	1	1
	170	1	1	0.9	0.8	0.8
	180	1	1	0.9	0.5	0.5
	190	1	1	0.9	0.2	0
	200	1	1	0.9	0.2	0

$w = 2$						
		$y$				
		60	70	80	90	100
$x$	150	1	1	1	1	1
	160	1	1	1	1	1
	170	1	1	0.9	0.8	0.8
	180	1	1	0.9	0.5	0.5
	190	1	1	0.9	0.2	0
	200	1	1	0.9	0.2	0

$w = 3$						
		$y$				
		60	70	80	90	100
$x$	150	1	1	1	1	1
	160	1	1	1	1	1
	170	1	1	1	0.9	0.9
	180	1	1	1	0.6	0.6
	190	1	1	1	0.3	0.1
	200	1	1	1	0.3	0.1

$w = 4$						
		$y$				
		60	70	80	90	100
$x$	150	1	1	1	1	1
	160	1	1	1	1	1
	170	1	1	1	1	1
	180	1	1	1	1	1
	190	1	1	1	1	0.8
	200	1	1	1	1	0.8

$w = 5$						
		$y$				
		60	70	80	90	100
$x$	150	1	1	1	1	1
	160	1	1	1	1	1
	170	1	1	1	1	1
	180	1	1	1	1	1
	190	1	1	1	1	1
	200	1	1	1	1	1

Tabela 2.7: Relação induzida por uma regra com conjunção no antecedente.

onde a relação *aproximadamente iguais* foi definida no exemplo 2.2.1 e a relação *próximo\_de\_3* é definida como

$$\text{próximo\_de\_3} = \{0/1 + 0.2/2 + 1/3 + 0.2/4\}$$

Logo, pode-se inferir  $V_2$  é  $B$  por

$$B = \text{próximo\_de\_3} \circ \text{próximo}$$

onde

$$B(y) = \max_{x \in X} [\text{próximo\_de\_3}(x) \wedge \text{próximo}(x, y)] .$$

Calculando-se a composição max-min das matrizes de relação correspondentes obtém-se

$$\begin{bmatrix} 0 & 0.2 & 1 & 0.2 \end{bmatrix} \circ \begin{bmatrix} 1 & 0.5 & 0 & 0 \\ 0.5 & 1 & 0.5 & 0 \\ 0 & 0.5 & 1 & 0.5 \\ 0 & 0 & 0.5 & 1 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.5 \\ 1 \\ 0.5 \end{bmatrix}$$

O resultado final pode ser interpretado como  $V_2$  está próximo de 3.

## 2.2.4 Qualificação de possibilidade

Uma das formas de se qualificar proposições em RA é a qualificação de possibilidade. Yager [73] [74] [75] sugere que proposições da forma

$$V \text{ é } A \text{ é possível}$$

sejam usadas para a representação de vários tipos de conhecimento de senso comum. A expressão acima é menos precisa que a proposição  $V$  é  $A$ . Com uma proposição com qualificação de possibilidade indica-se que a informação que se tem sobre o valor de  $V$  é que ele está em algum conjunto que possivelmente é o conjunto  $A$ . Formalmente esta proposição é traduzida em

$$V \text{ é } A^+$$

onde  $A^+$  é um subconjunto do conjunto potência de  $X$ , tal que  $B \in A^+$  se  $B \subset X$  e  $Poss[B/A] = 1$ . Em particular, para todo subconjunto nebuloso  $G$  de  $X$

$$A^+(G) = Poss[A/G] = \text{Max}_x [A(x) \wedge G(x)].$$

Uma variável *default*  $V$  é aquela que permite inferir o conseqüente baseado na possibilidade de  $V$  assumir um dos seus valores aceitáveis, e não na certeza. Se  $V$  é uma variável *default* com conjunto aceitável  $A$  pode-se representá-la em uma implicação incluindo-se no antecedente da regra uma proposição com qualificação de possibilidade. A expressão

tipicamente SE  $V$  é  $A$  ENTÃO  $U$  é  $B$

pode ser representada como

SE  $V$  é  $A$  e  $U$  é  $B$  é possível ENTÃO  $U$  é  $B$

Yager sugere um procedimento [74] que generaliza o processo de raciocínio aproximado para permitir a inclusão de regras com conhecimento *default*, representadas como na expressão acima. O referido procedimento foi implementado em [63] e aplicado a exemplos da área de sistemas de herança [68].

### 2.2.5 Métodos de Inferência Nebulosa

Considere novamente o padrão de inferência

$$\frac{V \text{ é } A' \quad \text{SE } V \text{ é } A \text{ ENTÃO } U \text{ é } B}{U \text{ é } B'}$$

(2.4)

onde  $A$  e  $B$  são subconjuntos nebulosos dos conjuntos base  $X$  e  $Y$ , e  $V$  e  $U$  são variáveis sobre  $X$  e  $Y$ , respectivamente. A expressão

$$B'(y) = \max[A'(x) \wedge D_{B/A}(x, y)],$$

que determina o conjunto  $B'$  da proposição resultante da inferência composicional, depende da forma como a relação  $D_{B/A}(x, y)$  é calculada e do operador aplicado na conjunção, que neste caso é o operador *min*. Serão discutidas a seguir generalizações sobre esse dois componentes. Observa-se que a expressão empregada na determinação de  $D_{B/A}$  é chamada uma *função de implicação* [25] ou um *operador de implicação* [18] e denotada de forma genérica por [25]

$$D_{A \rightarrow B}(x, y) = f_{\rightarrow}(A(x), B(y)).$$



A definição de uma função  $f_{\rightarrow}$  implica no que é usualmente chamado na literatura de *método de raciocínio aproximado* [35], [36], [25].

Muita discussão tem havido sobre a escolha apropriada da relação  $D(x, y)$ . Vários pontos de vista podem ser adotados para realizar esta escolha:

- um ponto de vista puramente algébrico,
- uma abordagem orientada à dedução ou
- uma investigação semântica das regras nebulosas.

Sob do ponto de vista algébrico, foram propostas as funções de implicação mais conhecidas e mais utilizadas principalmente na área de controle nebuloso [36]. Sejam  $\times, \cup, \cap, \oplus$  os operadores de produto cartesiano, união, intersecção e soma limitada de subconjuntos nebulosos, respectivamente. A proposição condicional do padrão de inferência em (2.4) pode derivar diversas relações sobre  $X \times Y$ . Por exemplo, Zadeh propôs os métodos

$$D_m = (A \times B) \cup (\bar{A} \times Y)$$

$$D_a = (\bar{A} \times Y) \oplus (X \times B).$$

Mamdani [33] propôs o método

$$D_c = A \times B.$$

Mizumoto et all. [35], [20], [36] sugeriram e testaram uma extensa variedade de operadores sob vários critérios. Outras propostas, análises e comparações de métodos de raciocínio aproximado podem ser encontradas em [53], [5], [46], [52] e [21]. Os métodos acima foram propostos considerando a regra de inferência composicional, na forma sugerida inicialmente por Zadeh, ou seja, a composição *max-min*. Outros tipos de composição foram estudados para serem usados na regra de inferência composicional [37] [52]. Uma flexibilidade muito maior pode ser obtida na representação de proposições condicionais com a generalização dos métodos de RA substituindo-se os operadores de *min* e *max* por *T-norma* e *S-norma*, respectivamente. A escolha de um operador é sempre uma questão de contexto, e depende principalmente do problema que está sendo modelado. É apropriado, portanto, usar o conceito geral de T-operadores para oferecer mais opções e maior flexibilidade na seleção de operadores que mais se adequam a um dado problema. Por exemplo, considerando a função de implicação de Mamdani, um método generalizado pode ser obtido fazendo-se

$$D_{A \rightarrow B}(x, y) = T(A(x), B(y)).$$

Usando-se os métodos de Zadeh, generaliza-se por

$$D_{A \rightarrow B}(x, y) = S(T(A(x), B(y)), N(A(x)))$$

ou

$$D_{A \rightarrow B}(x, y) = S(B(y), N(A(x))).$$

Extensões semelhantes podem ser feitas sobre outros métodos de implicação. Dubois e Prade [18] apresentam uma classificação bastante abrangente de operadores de implicação baseados em T-norma, agrupados em três famílias diferentes.

Uma generalização para T-norma semelhante pode ser aplicada ao operador *min* usado na composição *max-min* proposta por Zadeh (ver expressão 2.1), resultando numa composição generalizada da forma

$$B' = A' \bullet D$$

definida por

$$B'(y) = \sup(A'(x) \star D_{A \rightarrow B}(x, y)).$$

Resumindo, o padrão de inferência do RA de interesse neste trabalho é o padrão de inferência generalizado por T-operadores, que pode ser representado de forma esquemática como

$$\frac{V \text{ é } A' \quad (U, V) \text{ é } D_{A \rightarrow B}}{U \text{ é } B' = A' \bullet D}$$

onde  $D_{A \rightarrow B}$  é a relação nebulosa entre  $A$  e  $B$  dada por  $D(x, y) = f_{\rightarrow}(A(x), B(y))$  e  $\bullet$  é a composição de  $A'$  e  $D_{A \rightarrow B}$  definida por

$$A' \bullet D_{A \rightarrow B}(y) = \text{Sup}_x[A'(x) \star D_{A \rightarrow B}(x, y)].$$

A regra de inferência composicional tem sido estudada com o objetivo de se escolher o operador  $f_{\rightarrow}$  e o operador  $\star$  que aparece na composição, tal que  $B'$  satisfaça propriedades intuitivamente satisfatórias de acordo com a proximidade entre  $A$  e  $A'$ . Diferentes conjuntos de axiomas tem sido sugeridos, com claras contradições entre eles [18]. Por exemplo, um dos conjuntos de axiomas requer que  $B' \supseteq B$  (nada melhor do que a regra diz pode ser inferido), e o outro sugere que, ou

$$\text{muito}_A \circ D_{A \rightarrow B} = B$$

ou

$$\text{muito\_}A \circ D_{A \rightarrow B} = \text{muito\_}B .$$

Se for exigido que  $B' \supseteq B$ ,  $\text{muito\_}B$  nunca será deduzido de  $\text{muito\_}A$ . Entretanto, muitas vezes a regra  $\text{muito\_}A \circ D_{A \rightarrow B} = \text{muito\_}B$  parece ser natural. Por exemplo, de premissas como

*Se o tomate está vermelho então o tomate está maduro e  
O tomate está muito vermelho,*

parece natural concluir que

*O tomate está muito maduro.*

A compatibilidade com o modus ponens clássico ( $A \circ D_{A \rightarrow B} = B$ ) também tem sido incluída como um critério desejável. Este é satisfeito, por exemplo, pela abordagem de Mamdani [33] e não é satisfeita pela abordagem de Zadeh [84]. A escolha do método de raciocínio apropriado é uma questão de contexto onde essas considerações devem ser ponderadas.

De forma genérica, se  $V_1, V_2, \dots, V_k$  são variáveis assumindo valores nos conjuntos base  $X^1, X^2, \dots, X^k$  respectivamente, então a proposição

$$V_1 \text{ é } A_1 \text{ e } V_2 \text{ é } A_2 \text{ e } \dots \text{ e } V_k \text{ é } A_k$$

induz uma relação  $H$  sobre  $X^1 \times X^2 \times \dots \times X^k$  tal que

$$H(x^1, x^2, \dots, x^k) = \star_{i=1, \dots, k} [A_i(x^i)]. \quad (2.5)$$

Se além disso  $U$  for uma variável com conjunto base  $Y$ , a regra

$$\text{SE } V_1 \text{ é } A_1 \text{ e } V_2 \text{ é } A_2 \text{ e } \dots \text{ e } V_k \text{ é } A_k \text{ ENTÃO } U \text{ é } B \quad (2.6)$$

induz uma relação  $F$  sobre  $X^1 \times X^2 \times \dots \times X^k \times Y$  tal que

$$F(x^1, x^2, \dots, x^k, y) = f_{-}(H(x^1, x^2, \dots, x^k), B(y)) \quad (2.7)$$

onde  $H(x^1, x^2, \dots, x^k)$  está definido em(2.5).

Agora assumamos que temos os fatos  $V_1 \text{ é } A'_1, V_2 \text{ é } A'_2, \dots, V_k \text{ é } A'_k$ . Na representação esquemática de inferência, obtém-se

$$\begin{array}{l} V_1, V_2, \dots, V_k \text{ é } H' \\ (V_1, V_2, \dots, V_k, Y) \text{ é } F \\ \hline U \text{ é } B' = H' \bullet F \end{array}$$

onde

$$H'(x^1, x^2, \dots, x^k) = \star_{i=1, \dots, k} [A'_i(x^i)] \quad (2.8)$$

e  $F$  está definida em (2.7).

## 2.3 Introdução às Redes de Petri

As Redes de Petri são uma ferramenta de modelagem gráfica e matemática, aplicável a muitos sistemas. São adequadas a descrição de sistemas de processamento da informação que se caracterizam por serem concorrentes, assíncronos, distribuídos, paralelos, não determinísticos, e/ou estocásticos. Redes de Petri podem ser aplicadas a qualquer área ou sistema onde o fluxo de informação possa ser representado graficamente e que necessite algum meio de representar atividades concorrentes ou paralelas. Será apresentado aqui um resumo dos principais conceitos relativos às Redes de Petri. Para um estudo mais detalhado e profundo do assunto encontra-se na literatura uma extensa variedade de trabalhos como por exemplo [49], [39], [4].

Uma Rede de Petri pode ser representada graficamente por um tipo particular de grafo dirigido, ponderado e bipartido consistindo de dois tipos de nós, chamados de *lugares* e *transições*, onde os arcos vão ou de um lugar a uma transição ou de uma transição a um lugar. Lugares são denotados por círculos e transições por barras ou retângulos. Os arcos são rotulados por seus pesos (inteiros positivos), onde um arco com peso  $k$  pode ser interpretado como um conjunto de  $k$  arcos paralelos.

Uma *marcação* (estado) atribui a cada lugar um inteiro não negativo. Se uma marcação atribui a um lugar  $p$  um inteiro não negativo  $k$ , diz-se que  $p$  está marcado com  $k$  senhas. Senhas são representadas graficamente por pequenos círculos pretos. Uma marcação é denotada por  $M$ , um vetor de ordem  $m$  onde  $m$  é o número de lugares. O  $i$ -ésimo elemento de  $M$ , denotado por  $M(i)$ , é o número de senhas no lugar  $p$ .

**Definição 2.3.1** *Uma Rede de Petri é uma 5-tupla [39]*

$$RP = (P, T, F, W, M_0)$$

onde:

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

Uma estrutura de Rede de Petri  $N = (P, T, F, W)$  sem nenhuma marcação inicial específica é denotada por  $N$ .

As transições de uma Rede de Petri podem disparar, provocando uma mudança na marcação da rede. Para simular o comportamento dinâmico de um sistema, o estado ou marcação de uma rede muda de acordo com a seguinte regra de disparo de transições:

1. Uma transição  $t$  é dita *habilitada* se todo lugar de entrada  $p$  de  $t$  está marcado com pelo menos  $w(p, t)$  senhas, onde  $w(p, t)$  é o peso do arco que conecta  $p$  a  $t$ .
2. Uma transição habilitada pode ou não disparar (dependendo se o evento realmente acontece).
3. O disparo de uma transição habilitada  $t$  remove  $w(p, t)$  senhas de cada lugar de entrada  $p$  de  $t$ , e coloca  $w(p, t)$  senhas em cada um dos lugares de saída  $p$  de  $t$ , onde  $w(t, p)$  é o peso do arco que conecta  $t$  a  $p$ .

Essas regras de disparo assumem que todo lugar pode conter um número ilimitado de senhas. Tais Redes de Petri são chamadas de *redes com capacidade infinita*. As redes que consideram um limite superior para o número de senhas que um lugar pode conter é chamada uma *rede com capacidade finita*.

Muitos dos conceitos fundamentais de Redes de Petri foram definidos por pesquisadores diferentes de formas diferentes. A definição apresentada aqui foi extraída de [39] e caracteriza uma classe de redes chamadas de Redes Lugares/Transições [54]. Outras classes de modelos baseados em rede foram definidos, como por exemplo as Redes Elementares e as Redes de Alto Nível [4]. Classes como essas constituem elementos da *Teoria Geral de Redes* iniciada por Petri em 1962 [50].

A regra de disparo de transições é ilustrada na figura 2.1, onde a transição  $t$  está habilitada. Após o disparo de  $t$ , a marcação muda conforme apresentado na figura 2.1(b).

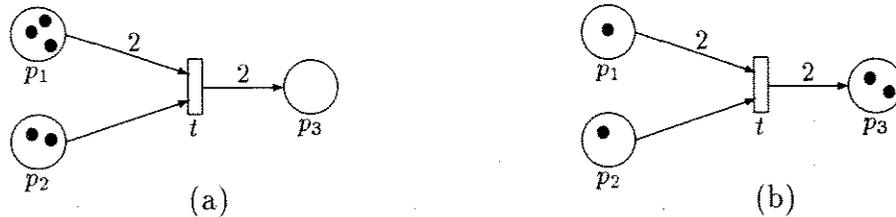


Figura 2.1: Exemplo de Disparo de transição. (a) Marcação antes do disparo de  $t$ . (b) Marcação depois do disparo de  $t$ .

**Exemplo 2.3.1** Para ilustrar a utilidade das Redes de Petri quanto à modelagem, é apresentado um problema clássico de sincronização conhecido como sistema de leitores e escritores. Um sistema como esse consiste de  $k$  processos que podem ler ou escrever em uma memória compartilhada. Até  $k$  processos podem estar lendo concorrentemente, mas quando um processo está escrevendo, nenhum outro processo pode estar lendo ou escrevendo. Na figura 2.2 os lugares  $p_1$ ,  $p_2$  e  $p_4$  representam estados dos processos e a memória é representada por  $p_3$ . Os pesos dos arcos  $w(t_4, p_3) = w(p_3, t_2) = k$  garantem a sincronização exigida pelo problema. Quando um determinado processo em  $p_1$  deve começar a escrever, a transição  $t_2$  deve disparar. Essa transição está habilitada ao disparo apenas quando o lugar  $p_2$  contém as  $k$ -senhas, significando que nenhum outro processo está lendo nem escrevendo. O disparo de  $t_2$  retira as  $k$  senhas de  $p_3$ , desabilitando o disparo de  $t_1$  bem como novos disparos de  $t_2$ . Quando o processo termina a escrita,  $t_4$  dispara e deposita as  $k$  senhas em  $p_3$ , permitindo que outros processos escrevam ou leiam.

Uma estrutura onde um lugar  $p$  tem duas ou mais transições de saída como mostrado na figura ?? é chamada de *conflito*, *decisão* ou *escolha*, dependendo da aplicação.

### 2.3.1 Métodos de Análise

Uma importante característica das Redes de Petri é a possibilidade de se analisar propriedades de sistemas concorrentes. Dois tipos de propriedades podem ser estudados com um modelo baseado em Redes de Petri: as propriedades de comportamento,

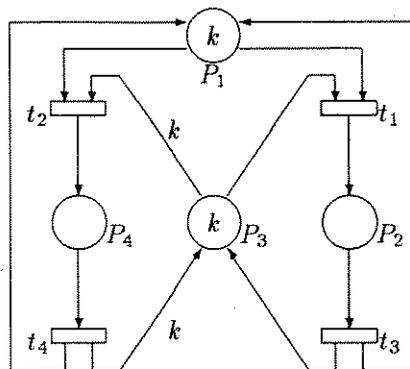


Figura 2.2: Rede de Petri representando o problema de leitura e escrita.

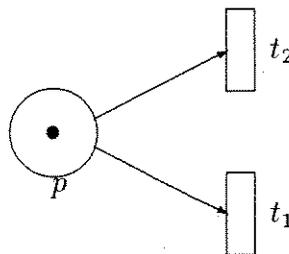


Figura 2.3: Estrutura de Rede de Petri chamada conflito, escolha ou decisão.

que dependem da marcação inicial, e as propriedades estruturais, que são independentes da marcação inicial. A seguir são discutidas brevemente as propriedades de comportamento da rede.

### Alcançabilidade

O conceito de alcançabilidade é fundamental para o estudo de propriedades dinâmicas da rede. Uma marcação  $M_n$  é dita ser *alcançável* de  $M_0$  se existe uma sequência de disparos que transforma  $M_0$  em  $M_n$ . O conjunto de todas as marcações possíveis de se alcançar a partir de  $M_0$  em uma rede  $(N, M_0)$  é denotado por  $R(M_0)$  e é chamado de *conjunto de alcançabilidade* de  $(N, M_0)$ . O problema de alcançabilidade para Redes de Petri é o problema de determinar se  $M_n \in R(M_0)$  para uma dada marcação

$M_n$  na rede  $(N, M_0)$ .

### Limitabilidade

Uma Rede de Petri  $(N, M_0)$  é dita ser  $k$ -limitada se o número de senhas em cada lugar não excede um número finito  $k$  para toda marcação atingível de  $M_0$ . Uma Rede de Petri  $(N, M_0)$  é dita *segura* se for 1-limitada.

### Vivacidade

Uma Rede de Petri  $(N, M_0)$  é dita ser *viva* se, qualquer que seja a marcação atingida a partir de  $M_0$ , é possível em algum momento disparar qualquer transição da rede através de alguma sequência de disparos. Vivacidade é uma propriedade bastante forte e sua verificação é de alto custo. Portanto, a condição de vivacidade é relaxada pela definição de diferentes níveis de vivacidade. Uma transição  $t$  em uma Rede de Petri  $(N, M_0)$  é dita ser:

1. *morta (L0-viva)* se  $t$  nunca pode ser disparada em nenhuma sequência de disparos.
2. *L1-viva (potencialmente disparável)* se  $t$  pode disparar pelo menos uma vez em alguma sequência de disparos.
3. *L2-viva* se, dado um inteiro positivo  $k$ ,  $t$  pode ser disparada pelo menos  $k$  vezes em alguma sequência de disparos.
4. *L3-viva* se  $t$  aparece infinitamente, frequentemente em alguma sequência de disparos.
5. *L4-viva ou viva* se  $t$  é L1-viva para toda marcação  $M$  em  $R(M_0)$ .

Uma Rede de Petri é dita ser  $Lk$ -viva se toda transição na rede é  $Lk$ -viva,  $k = 0, 1, 2, 3, 4$ .

### Conservação

Uma Rede de Petri  $(N, M_0)$  é dita ser *conservativa* com relação a um vetor de pesos  $w = \{w_1, \dots, w_n\}$ ,  $n = |P|$ ,  $w_i \geq 0$ , se para todo  $M$ ,  $M \in R(M_0)$ ,

$$\sum_i w_i \cdot M_0(p_i) = \sum_i w_i \cdot M(p_i)$$

Propriedades estruturais são aquelas que dependem da estrutura topológica das Redes de Petri. Elas são independentes da marcação inicial  $M_0$  no sentido de que essas propriedades valem para qualquer marcação inicial ou estão relacionadas com a existência de certas sequências de disparo a partir de alguma marcação inicial. As propriedades estruturais são semelhantes às propriedades comportamentais e não serão detalhadas aqui.

Os métodos de análise de Redes de Petri podem ser classificados em três grupos:

1. método de árvore de cobertura
2. equações matriciais
3. técnicas de decomposição ou redução

O primeiro método envolve essencialmente a enumeração de todas as marcações ou de suas marcações de coberturas. Pode potencialmente ser aplicado a qualquer tipo de rede, mas está limitado a pequenas redes devido à complexidade da explosão combinatorial do espaço de estados. Os outros dois métodos são poderosos mas em muitos casos são aplicáveis somente a subclasses especiais de redes de Petri.

### Árvore de Cobertura

A árvore de cobertura representa o conjunto de alcançabilidade de uma Rede de Petri. Dada uma Rede de Petri  $(N, M_0)$ , a partir da marcação inicial  $M_0$  pode-se obter tantas marcações novas quantas forem as transições habilitadas. De cada nova marcação, pode-se novamente alcançar mais marcações. Esse processo resulta em uma representação em árvore para as marcações. Os nós representam marcações geradas a partir da raiz  $M_0$  e de seus sucessores e cada arco representa o disparo de uma transição. Para manter a árvore finita, um símbolo especial  $w$  é introduzido, que pode ser considerado como *infinito*.

### Equações Matriciais

Uma segunda abordagem para análise de Redes de Petri é baseada na representação matricial dessas redes. Uma representação alternativa para as Redes de Petri é através de sua matriz de incidência.

**Definição 2.3.2** Para uma Rede de Petri  $N$  com  $n$  lugares e  $m$  transições, a matriz de incidência  $C = [c_{ij}]$  é uma matriz  $n \times m$  de inteiros onde cada elemento é dado

por

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

onde  $c_{ij}^+ = w(i, j)$  é o peso do arco da transição  $i$  para o lugar  $j$  e  $c_{ij}^-$  é o peso do arco do lugar  $j$  para a transição  $i$ .

Pode-se verificar que  $c_{ij}^-$ ,  $c_{ij}^+$  e  $c_{ij}$  representam respectivamente o número de senhas removidas, adicionadas e alteradas no lugar  $j$  quando a transição  $i$  dispara.

**Definição 2.3.3** Dada uma matriz de incidência  $C_{n \times m}$  correspondente à Rede de Petri  $(N, M_0)$ , um vetor  $m$ -dimensional  $p$  é chamado de  $S$ -invariante para a rede  $N$  se e somente se

$$p.C = 0 .$$

Um vetor  $q$  tal que

$$q.C^T = 0 .$$

é chamado de  $T$ -invariante.

A existência do vetor  $p$  implica numa lei de conservação entre as senhas da rede. Os invariantes fornecem ferramentas poderosas para o estudo de propriedades estruturais das Redes de Petri.

## Técnicas de Decomposição ou Redução

Nesta abordagem, o modelo de um certo sistema é reduzido a um mais simples, preservando as propriedades a serem analisadas. Diversas operações podem ser aplicadas para se reduzir sub-redes a simples transições ou lugares, com o objetivo de facilitar a análise de sistemas de grande porte.

### 2.3.2 Redes de Alto Nível

O uso de Redes de Petri para descrever sistemas concorrentes logo mostrou a necessidade de se encontrar tipos de redes mais poderosos, para descrever sistemas complexos de uma maneira mais tratável. Nas redes de Petri comuns é frequentemente necessário ter várias sub-redes idênticas. Isso motivou estudos para se encontrar uma generalização dessas redes que permitisse a representação de sub-redes idênticas de uma forma compacta. Redes de Petri Coloridas e redes Predicado/Transição são dois tipos de Redes de Petri de alto nível que generalizam os modelos de redes de

Petri, onde algum tipo de informação é vinculada às senhas. Uma discussão detalhada dessas redes pode ser encontrada em [23] e [28]. Aqui, apenas alguns aspectos elementares das redes Predicado/Transição serão discutidos informalmente, através de uma comparação com as Redes de Petri comuns. As Redes Predicado/Transição podem ser caracterizadas pelos seguintes pontos:

- A Rede Pr-T possui os mesmos elementos  $P$ ,  $T$  e  $F$  da Rede de Petri comum, onde os lugares representam predicados com extensões variáveis e transições representam classes de mudanças elementares nas extensões.
- Os arcos são rotulados com somas formais de  $n$ -uplas de variáveis; o comprimento de cada  $n$ -upla é a aridade do predicado conectado ao arco.
- As transições podem conter fórmulas inscritas construídas sobre uma estrutura  $\Sigma$  que define uma coleção de objetos tipados e operações e relações aplicáveis a esses objetos. As fórmulas inscritas em uma transição  $t$  determinam condições que devem ser satisfeitas para o disparo de  $t$ .
- Uma senha  $\langle a_1, \dots, a_r \rangle$  em um lugar  $p \in P$  denota o fato de que o predicado  $P(x_1, \dots, x_k)$  correspondente a esse lugar é verdadeiro para a particular instanciação dada pelos argumentos contidos na senha.

Para toda transição  $t \in T$ , os conjuntos

$$I(t) = \{p \in P \mid (p, t) \in F\}$$

$$O(t) = \{p \in P \mid (t, p) \in F\}$$

são chamados os conjuntos de lugares de entrada de  $t$  e o conjunto de lugares de saída de  $t$ . As definições de conjuntos de transições de entrada  $I(p)$  e de conjunto de transições de saída  $O(p)$  de um lugar  $p$  são análogas.

Uma transição  $t$  está habilitada quando

1. Todo lugar  $p \in I(t)$  contém pelo menos tantas senhas quantas são especificadas pelo rótulo de  $(p, t)$ .
2. As senhas dos lugares de entrada tem valores que satisfazem as fórmulas inscritas em  $t$  (se houver).

Uma transição habilitada  $t$  pode disparar removendo-se de cada lugar  $p \in I(t)$  o número de senhas especificado pelo rótulo do arco  $(p, t)$  e adicionando-se a cada lugar  $p' \in O(t)$  o número de senhas especificado pelo rótulo em  $(t, p')$ .

Como exemplo, considere-se a representação de programas lógicos mostrada na figura 2.4(a). A rede consiste de uma transição  $t$  e quatro lugares, sendo dois lugares de entrada  $p_1$  e  $p_2$  e dois lugares de saída  $p_3$  e  $p_4$ . Note que os quatro arcos estão rotulados com  $2x$ ,  $\langle x, y \rangle + \langle y, z \rangle$ ,  $\langle x, z \rangle$  e  $e$ . Os rótulos dos arcos determinam quantos e quais tipos de senhas coloridas serão removidas ou adicionadas a lugares. Por exemplo, quando a transição  $t$  na figura 2.4 dispara, acontece o seguinte:

- $p_1$  perde duas senhas da mesma cor  $x$ ;
- $p_2$  perde duas senhas de cores diferentes,  $\langle x, y \rangle$  e  $\langle y, z \rangle$ ;
- $p_3$  ganha uma senha da cor  $\langle x, z \rangle$ ; e
- $p_4$  ganha uma senha da cor  $e$ , uma constante.

A marcação inicial da rede consiste do seguinte:

- $p_1$  tem quatro senhas coloridas, dois  $a$ 's e dois  $d$ 's;
- $p_2$  tem três senhas coloridas  $\langle a, b \rangle$ ,  $\langle b, c \rangle$  e  $\langle d, a \rangle$ ;
- $p_4$  não tem senhas iniciais.

Nessa representação, variáveis são denotadas por  $x, y, z, \dots$  e constantes são denotadas por  $a, b, c, d, \dots$ . Para cada transição, uma variável de mesmo símbolo aparecendo em arcos de entrada e de saída denotam a mesma variável. Uma transição é dita habilitada se existirem senhas suficientes da cor certa em cada lugar de entrada de  $t$ . Por exemplo, na figura 2.4(a) a transição  $t$  está habilitada pois existem senhas suficientes nos seus lugares de entrada e existem substituições consistentes  $\{a/x, b/y, c/z\}$  e  $\{d/x, a/y, b/z\}$ . Portanto existem duas maneiras de disparar  $t$  com essas duas substituições diferentes. As redes apresentadas na figura 2.4(b) e (c) mostram as marcações após o disparo de  $t$  com cada uma das substituições acima, respectivamente. Uma rede de alto nível pode ser desdobrada em uma rede de Petri comum desdobrando-se cada lugar  $p$  em um conjunto de lugares, um para cada cor de senhas que  $p$  pode conter, e desdobrando-se cada transição  $t$  em um conjunto de transições, uma para cada maneira em que  $t$  pode disparar. Por exemplo, a rede da figura 2.4 pode ser desdobrada na rede mostrada na figura 2.5.

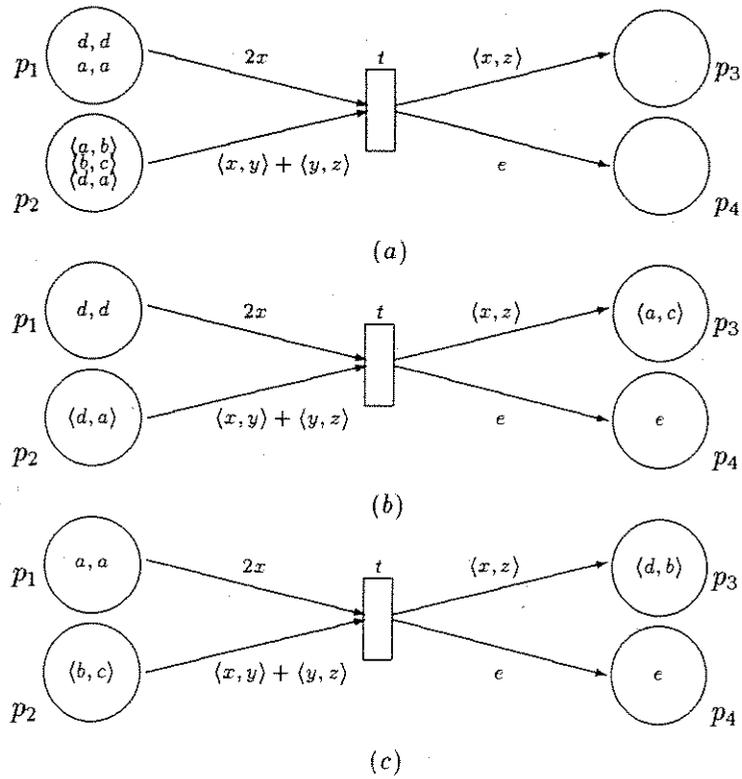


Figura 2.4: Exemplo de disparo de transições em uma rede de alto nível. (a) antes do disparo. (b) Depois do disparo com substituição  $\{a/x, b/y, c/z\}$ . (c) Depois do disparo com substituição  $\{d/x, a/y, b/z\}$ .

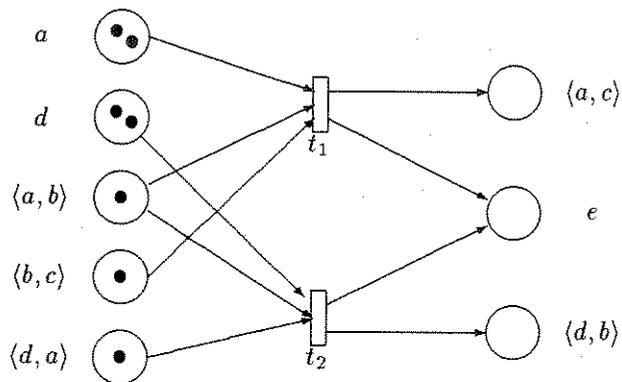


Figura 2.5: Rede de Petri correspondente à rede de alto nível do exemplo de programação lógica.

## 2.4 Conclusão

Os conceitos básicos para o desenvolvimento deste trabalho foram apresentados. Foram abordados conceitos da teoria de Subconjuntos Nebulosos, Lógica Nebulosa e teoria de Raciocínio Aproximado. Os operadores de intersecção e união e suas generalizações por T-norma e S-norma, respectivamente, foram apresentados juntamente com as suas propriedades. A regra de inferência do RA, chamada de modus ponens composicional, foi apresentada com exemplos ilustrativos. A generalização de métodos de inferência pela substituição dos operadores de min e max por T-norma e S-norma, respectivamente, foi abordada e uma representação genérica desses métodos foi introduzida. Os modelos básicos de Redes de Petri comum e de alto nível também foram apresentados. Esses temas não foram tratados exaustivamente, mas apenas uma seleção de conceitos pertencentes a cada uma dessas áreas foi introduzida.

# Capítulo 3

## Redes de Petri Nebulosas

### 3.1 Introdução

Neste capítulo, o modelo de Redes de Petri Nebulosas é introduzido como um modelo formal para representação de bases de regras nebulosas. A seção 3.2 contém uma descrição resumida das principais propostas encontradas na literatura, relacionadas a representação do conhecimento, usando estruturas do tipo de Redes de Petri. Estão aí incluídas abordagens tanto para representação de regras convencionais como de regras nebulosas. Na seção 3.3 apresenta-se a definição formal do modelo em questão, a modelagem do padrão básico de inferência, os diferentes tipos de transições com suas respectivas regras de disparo, e um exemplo numérico ilustrativo. A seguir, discute-se brevemente a questão do encadeamento de regras e sua representação no modelo proposto. A seção 3.5 descreve a representação de regras com antecedentes múltiplos. A modelagem das regras com quantificadores nebulosos e das regras com fatores de certeza é apresentada na seção 3.6. Conjuntos típicos de regras nebulosas como regras paralelas e regras conflitantes também são abordados. As duas interpretações de regras paralelas mais comumente encontradas na literatura e suas representações na Rede de Petri Nebulosa são apresentadas na seção 3.7. As regras conflitantes são discutidas na seção 3.8. Os conceitos tradicionais de matriz de incidência e de equações de mudança de estado são estendidos na seção 3.9 para que possam ser aplicados a Redes de Petri Nebulosas. No contexto de uma base de regras, uma atenção especial deve ser dada à construção do modelo global através da composição das estruturas básicas aqui introduzidas. Alguns casos críticos de projeto da rede quando existem certos tipos de relacionamentos entre as regras são discutidos na seção 3.10, onde uma pequena base de regras é usada para ilustração.

## 3.2 Revisão da Literatura

As Redes de Petri têm sido propostas como esquema de representação do conhecimento através de abordagens variadas. A maioria dessas propostas trata da representação do conhecimento em forma de regras. Nesta seção é apresentado um resumo dos principais trabalhos analisados durante a revisão da literatura. Obviamente não se trata de uma relação exaustiva, mas sim de uma coleção das propostas mais significativas, que traduzem tendências nas pesquisas. Algumas das propostas relacionadas aqui exerceram algum tipo de influência, em diferentes níveis, no trabalho desenvolvido. Outras estão incluídas nesta revisão apenas com o objetivo de situar os modelos propostos nesta tese em um panorama razoavelmente abrangente.

Do ponto de vista da representação de regras convencionais, a analogia entre Redes de Petri e sistemas de produção foi observada inicialmente por Zisman [86]. Em [24] encontra-se uma abordagem bastante abrangente, onde Giordana e Saitta propõem o uso de redes Predicado/ Transição para representar regras em um sistema de produção, envolvendo predicados binários. O objetivo dessa proposta é fornecer uma base uniforme de representação de conhecimento e de controle em sistemas especialistas. Giordana e Saitta também enfatizam a importância de se usar representações algébricas do modelo em rede para provar propriedades do sistema modelado, como por exemplo a questão da consistência da base de conhecimento (veja capítulo 6).

Enquanto o trabalho de Giordana e Saitta mostra como representar na rede Predicado/Transição diversas construções da lógica de primeira ordem usadas na representação do conhecimento, Murata e Zhang [38] apresentam um modelo baseado em redes Predicado/Transição para modelar programas lógicos em cláusulas de Horn [10]. Em [40] é mostrado como essa proposta pode ser estendida para ser aplicável ao raciocínio com a presença de inconsistências. O objetivo principal é o de estudar a execução paralela de consultas a programas lógicos. A representação de informação na forma de cláusulas de Horn também é tratada em [12]. A metodologia de modelagem baseada em redes é aí estendida por Chaudhury et al., através da proposta das chamadas redes proposicionais coloridas. Chaudhury et al. salientam que usando técnicas de análise de teoria de redes pode-se examinar inconsistências e contradições na base de conhecimento.

Outro modelo em rede para representação do conhecimento é introduzido por Deng e Chang em [16], chamado de modelo G-net. Neste modelo é possível representar tanto o conhecimento semântico estático como o conhecimento de controle dinâmico, usando-se estruturas básicas independentes. Essas estruturas básicas podem posteriormente ser combinadas em uma estrutura mista mais geral.

Uma proposta de representação de regras de produção na forma parâmetro-valor usando Redes de Petri ordinárias é apresentada por Agarwal e Tanniru [1]. O principal objetivo dessa proposta é o de usar as propriedades estruturais da rede para fazer verificação de integridade da base de conhecimento, tanto a nível local como a nível global.

Uma abordagem diferente, que introduz a incerteza e imprecisão nos modelos baseados em Redes de Petri, pode ser encontrada em [8] e [9]. O objetivo foi o de elaborar um modelo que combinasse Redes de Petri a Objetos, Teoria de Possibilidades e noções de conhecimento de tempo nebuloso, tendo em vista aplicações ao monitoramento de Sistemas Flexíveis de Manufatura a nível de coordenação global. Através do conceito de marcação imprecisa, é possível representar uma situação de ubiquidade, isto é, a incerteza relacionada a posição de um dado objeto no sistema. A noção de marcação imprecisa é traduzida formalmente com base nos conceitos de distribuição de possibilidade e de medida de possibilidade.

Algumas extensões às Redes de Petri convencionais têm sido propostas, para a representação de conhecimento impreciso, baseadas na teoria de subconjuntos nebulosos. Looney desenvolve uma proposta [32] considerando o raciocínio nebuloso via matrizes nebulosas. Nesse tipo de raciocínio, cada proposição é associada a um único valor entre zero e um que representa o grau de crença nessa proposição. A abordagem de Looney sugere uma representação em rede para o raciocínio aproximado por transformações de vetores de estado de valor verdade nebuloso. Um outro modelo de representação de conhecimento nebuloso é proposto por Chen et al. [13]. Este modelo representa regras de produção nebulosas com fatores de certeza. Um algoritmo de raciocínio fundamentado na estratégia de fatores de certeza é proposto, para determinar se existe uma relação do tipo antecedente-consequente entre duas proposições. O grau de verdade de cada proposição é representado por um único valor entre zero e um, associado às senhas. Essa abordagem tem sido usada na representação, planejamento de operações e recuperação automática de erros em um sistema de montagem de robôs [6] [7].

Uma nova abordagem para estender Redes de Petri para representar regras nebulosas foi proposta por Srinivasan e Gracanin [67]. Ao contrário da proposta anterior, uma senha é associada a uma função de pertinência de um valor linguístico. Uma transição pode disparar se a distância entre o valor da senha de entrada e o valor desejado é menor que um certo grau de tolerância. Essa distância determina a diferença entre as senhas de saída produzidas e os valores de saída especificados pela transição.

Outro modelo do tipo de Redes de Petri Nebulosas é proposto por Chun e Bien [14], juntamente com a descrição de um motor de inferência. Nesse trabalho é

mostrado que um sistema de produção com regras nebulosas pode ser transformado em uma Rede de Petri Nebulosa, que é por sua vez transformada em matrizes de acordo com o esquema proposto. Assim o motor de inferência é representado por uma fórmula algébrica apropriada à execução em paralelo.

Outra abordagem de Redes de Petri Nebulosas é sugerida por Matsumoto et. all. [34], com ênfase nas aplicações ao controle de sistemas nebulosos.

Na seção seguinte será apresentado o modelo de Redes de Petri Nebulosas proposto neste trabalho. Ao contrário das propostas descritas nesta seção, a modelagem é feita de acordo com a regra de inferência composicional.

### 3.3 O Modelo de Redes de Petri Nebulosas

Uma Rede de Petri Nebulosa (RPN) é um grafo bipartido que contém dois tipos de nós: *lugares* e *transições*, onde círculos representam lugares e barras representam transições. Cada lugar pode ou não conter uma senha associada com um valor de pertinência entre zero e um. As relações entre lugares e transições são representadas por arcos dirigidos. A cada arco está associado um valor de peso entre zero e um. Arcos que saem de um mesmo lugar podem estar ligados uns aos outros por um arco transversal, significando que este lugar está conectado a um conjunto de transições que devem disparar conjuntamente. Estes são chamados *arcos tipo e*. O conceito de Redes de Petri Nebulosas introduzido a seguir [56] [59], é derivado de Redes de Petri [39] [49] e de uma definição diferente de redes de Petri Nebulosas apresentada em [13].

**Definição 3.3.1** *Uma estrutura generalizada de Redes de Petri Nebulosas pode ser definida como uma 8-tupla:*

$$RPN = (P, T, S, F, w, \alpha, \beta, \Gamma)$$

onde

- $P = \{p_1, p_2, \dots, p_n\}$  é um conjunto finito de lugares,
- $T = \{t_1, t_2, \dots, t_m\}$  é um conjunto finito de transições,
- $P \cap T = \emptyset$ ,  $P$  e  $T$  são particionados em subconjuntos disjuntos,
- $S = X^1 \cup X^2 \cup \dots \cup X^s$  é uma união de conjuntos base  $X^i$  tal que  $X^i = \{x_1^i, x_2^i, \dots, x_{r_i}^i\}$  é um conjunto finito de elementos<sup>1</sup>, para  $i = 1, \dots, s$ ,

---

<sup>1</sup> $r_i$  é o número de elementos do conjunto base  $X^i$ .

- $F \subseteq (P \times T) \cup (T \times P)$  é um conjunto de arcos (relação de fluxo),
- $\omega : F \rightarrow [0, 1]$  é uma função de peso, um mapeamento de arcos a valores reais entre zero e um,
- $\alpha : P \rightarrow [0, 1]$  é uma função de associação, um mapeamento de lugares a valores reais entre zero e um,
- $\beta : P \rightarrow D$  é uma função de associação, um mapeamento bijetivo de lugares a elementos de conjuntos base,
- $\Gamma$  é uma tripla  $(\gamma, \delta, \tau)$  tal que  $\gamma$  e  $\delta$  são operadores  $T$ -norma e  $\tau$  é um operador  $T$ -norma ou  $T$ -conorma.

Sejam  $P^i$  e  $T^i$  elementos das partições de  $P$  e  $T$ , respectivamente. Seja  $l$  o número de elementos da partição de  $P$  e  $k$  o número de elementos da partição de  $T$ . Então temos

$$P = \cup_{i=1}^l P^i, P^i \cap P^j = \emptyset, \quad T = \cup_{i=1}^k T^i, T^i \cap T^j = \emptyset.$$

$\forall i, \forall j, i \neq j$ . Chamaremos  $P^i$  e  $T^j$  um grupo de lugares e um grupo de transições, respectivamente.

Na modelagem de regras de produção nebulosas de um sistema baseado em regras, grupos de lugares representam variáveis e conjuntos base. Grupos de transições juntamente com grupos de lugares representam regras. Os rótulos  $\omega(\langle p_i, t_j \rangle)$  dos arcos que conectam lugares a transições, são usualmente os elementos da matriz de relação induzida pela regra (ver seção 2.2.3). Grupos de transições e grupos de lugares podem ter significados diferentes quando formas mais complexas de regras são consideradas, como regras com múltiplos antecedentes e regras paralelas. Casos especiais como esses serão tratados nas próximas seções.

Seja  $X^i = \{x_1^i, x_2^i, \dots, x_r^i\}$  um dos conjuntos base e  $P^k = \{p_1, p_2, \dots, p_r\}, P^k \subset P$  um grupo de lugares. Se

$$\beta(p_j) = x_j^i$$

para  $j = 1, \dots, r$ , então diz-se que  $P^k$  está associado ao conjunto base  $X^i$ . Se  $f_i = \langle p_j, t_k \rangle$  e  $w(f_i) = d_{jk}$ , então diz-se que o arco  $f_i$  está associado ao peso (ou rótulo)  $d_{jk}$ . Se o grupo de transições  $T^i$  representa uma regra, diz-se que  $T^i$  está associado a essa regra.

Para maior clareza na notação, convencionou-se de agora em diante que, sempre que possível, conjuntos base de variáveis dos antecedentes das regras são denotados por  $X$  ou  $X^i$  e conjuntos base de variáveis dos consequentes são denotados por  $Y$  ou  $Y^j$ .

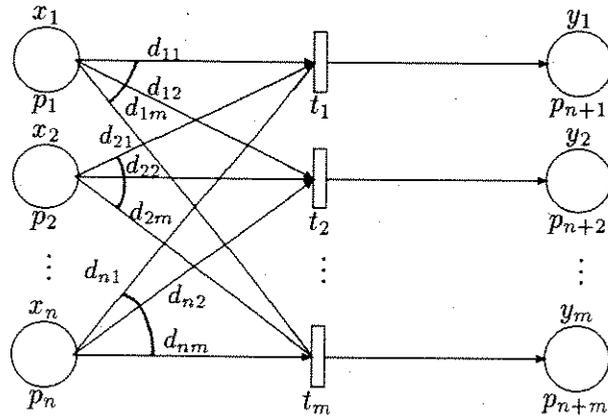


Figura 3.1: Modelo de Rede de Petri Nebulosa para uma regra condicional.

Considere-se a regra de produção nebulosa

$$\text{SE } V \text{ é } A \text{ ENTÃO } U \text{ é } B \quad (3.1)$$

onde  $A$  e  $B$  são subconjuntos nebulosos dos conjuntos base  $X^1 = X = \{x_1, x_2, \dots, x_n\}$  e  $X^2 = Y = \{y_1, y_2, \dots, y_m\}$  respectivamente, e a relação nebulosa entre  $A$  e  $B$  é dada por

$$D_{A \rightarrow B}(x_i, y_j) = d_{ij}.$$

Usando-se uma RPN, a regra em (3.1) pode ser modelada como mostra a figura 3.1. Neste caso, os lugares

$$p_1, p_2, \dots, p_n$$

estão associados ao conjunto base  $X$  e à variável  $V$ , e os lugares

$$p_{n+1}, p_{n+2}, \dots, p_{n+m}$$

estão associados ao conjunto base  $Y$  e à variável  $U$ . As transições

$$t_1, t_2, \dots, t_m$$

estão associadas a regra em (3.1).

**Definição 3.3.2** Uma Rede de Petri Nebulosa com alguns lugares contendo senhas é chamada uma Rede de Petri Nebulosa Marcada. A senha em um lugar  $p_i$  é representada por um ponto rotulado  $\bullet^{(\alpha(p_i))}$  e  $\alpha$  é chamada uma marcação nebulosa da Rede de Petri Nebulosa.

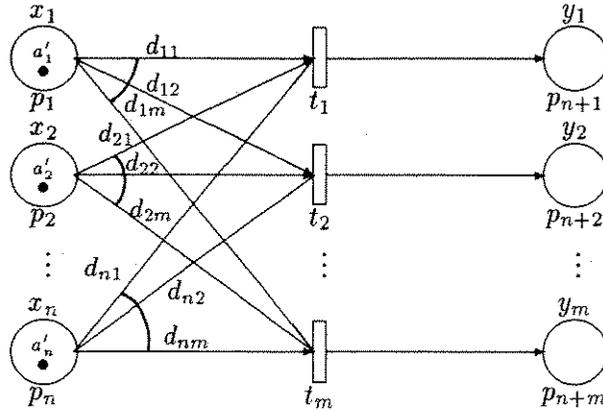


Figura 3.2: Modelo de Rede de Petri Nebulosa Marcada representando o padrão de inferência básico.

Quando se possui algum conhecimento sobre a variável  $V$  no antecedente, isto é, conhecemos a proposição

$$V \text{ é } A',$$

onde  $A'$  é um subconjunto nebuloso de  $X$ , dado por sua função de pertinência

$$A'(x_i) = a'_i, i = 1, \dots, n,$$

o padrão de inferência

$$V \text{ é } A' \tag{3.2}$$

$$\underline{\text{SE } V \text{ é } A \text{ ENTÃO } U \text{ é } B} \tag{3.3}$$

$$U \text{ é } B' \tag{3.4}$$

pode ser representado por uma Rede de Petri Nebulosa Marcada como a da figura 3.2, onde  $\alpha(p_i) = a'_i, i = 1, \dots, n$ . A existência de senhas nos lugares  $p_1, p_2, \dots, p_n$  e seus valores, denotam o fato  $V \text{ é } A'$ . A definição formal da RPN Marcada da figura 3.2 está na Tabela 3.1. Deve-se notar que a definição do mapeamento  $\omega$ , que dá os pesos dos arcos, depende da função de implicação usada no cálculo da relação induzida  $D_{A \rightarrow B}$ .

$$\begin{aligned}
P &= \{p_1, p_2, \dots, p_{n+m}\} \\
P &= P^1 \cup P^2 \text{ e } P^1 \cap P^2 = \emptyset \\
P^1 &= \{p_1, \dots, p_n\}, P^2 = \{p_{n+1}, \dots, p_{n+m}\} \\
T &= T^1 = \{t_1, \dots, t_m\} \\
S &= X \cup Y, X = \{x_1, \dots, x_n\}, Y = \{y_1, \dots, y_m\} \\
F &= \{\langle p_i, t_j \rangle \mid i = 1, \dots, n, j = 1, \dots, m\} \cup \{\langle t_j, p_{n+j} \rangle \mid j = 1, \dots, m\} \\
\omega(\langle p_i, t_j \rangle) &= d_{ij}, i = 1, \dots, n, j = 1, \dots, m \\
\omega(\langle t_j, p_{n+j} \rangle) &= 1, j = 1, \dots, m \\
\alpha(p_i) &= A'(x_i), i = 1, \dots, n \\
\beta(p_i) &= x_i, i = 1, \dots, n \\
\beta(p_{n+j}) &= y_j, j = 1, \dots, m
\end{aligned}$$

Tabela 3.1: Definição Formal de uma RPN Marcada. O elemento  $\Gamma$  não está definido.

**Exemplo 3.3.1** Como um exemplo mais concreto, suponha-se que o padrão de inferência dado em (3.2) e (3.3) será aplicado em um contexto tal que os conjuntos base são  $X = Y = \{1, 2, 3, 4\}$ , os subconjuntos nebulosos são definidos como  $A = \{0.6/2 + 1/3 + 0.5/4\}$ ,  $B = \{1/2 + 0.6/3 + 0.2/4\}$ ,  $A' = \{0.5/1 + 0.7/2 + 0.2/3\}$  e a relação nebulosa  $D_{A \rightarrow B}$  é obtida pela regra de tradução

$$D_{A \rightarrow B}(x, y) = 1 \wedge [1 - A(x) + B(y)],$$

resultando em

$$D_{A \rightarrow B} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0.4 & 1 & 1 & 0.6 \\ 0 & 1 & 0.6 & 0.6 \\ 0.5 & 1 & 1 & 0.7 \end{bmatrix}$$

A RPN que representa este caso particular é vista na figura 3.3, e sua definição formal é dada na Tabela 3.2.

O comportamento de um sistema baseado em regras nebulosas pode ser simulado mudando-se o estado ou marcação de uma Rede de Petri Nebulosa, de acordo com as regras de disparo de transições da rede. O estado de uma RPN é definido pela sua marcação nebulosa. Para introduzir a semântica do disparo de transições, uma definição adicional é necessária.

**Definição 3.3.3** Para cada transição  $t \in T$  e cada lugar  $p \in P$ , introduzimos:

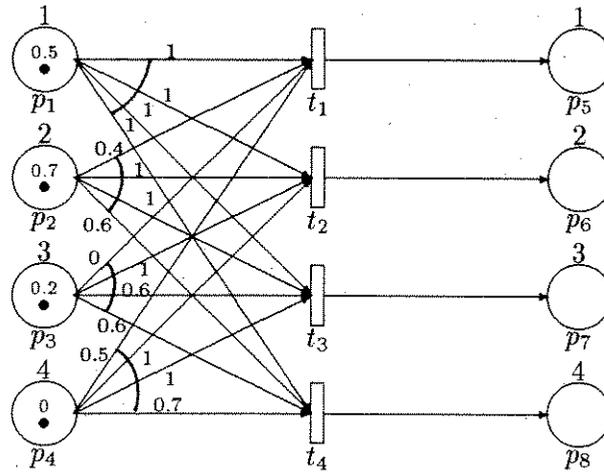


Figura 3.3: RPN Marcada assumindo-se os conjuntos base  $X = Y = \{1 + 2 + 3 + 4\}$  e o subconjunto nebuloso  $A' = \{0.5/1 + 0.7/2 + 0.2/3\}$ .

$$I(t) = \{p \in P \mid (p, t) \in F\}$$

$$I(p) = \{t \in T \mid (t, p) \in F\}$$

$$O(t) = \{p \in P \mid (t, p) \in F\}$$

$$O(p) = \{t \in T \mid (p, t) \in F\}$$

$I(t)$  e  $O(t)$  são os conjuntos de lugares de entrada e saída da transição  $t$ , respectivamente.  $I(p)$  e  $O(p)$  são os conjuntos de transições de entrada e saída do lugar  $p$ , respectivamente.

Em uma Rede de Petri convencional, uma transição  $t$  dispara removendo de cada um dos seus lugares de entrada o número de senhas especificado pelo peso do arco entre esse lugar de entrada e a transição e em seguida depositando uma senha em cada um de seus lugares de saída. Além disso, na interpretação convencional, dois ou mais arcos saindo de um mesmo lugar e indo para transições diferentes representam um *conflito* ou *escolha*. Em uma RPN, onde o número de senhas é sempre limitado a um, uma transição  $t$  está habilitada ao disparo quando todos os  $p_j \in I(t)$  tem uma senha. Porém, se a interpretação com relação a conflitos em redes convencionais fosse seguida para RPN, apenas uma transição de um mesmo

$$\begin{aligned}
P &= \{p_1, \dots, p_8\} \\
P &= P^1 \cup P^2, P^1 \cap P^2 = \emptyset \\
P^1 &= \{p_1, p_2, p_3, p_4\}, P^2 = \{p_5, p_6, p_7, p_8\} \\
T &= \{t_1, t_2, t_3, t_4\} \\
S &= X \cup Y = \{1, 2, 3, 4\} \\
F &= \{ \langle p_i, t_j \rangle \mid i, j = 1, \dots, 4 \} \cup \{ \langle t_j, p_{j+4} \rangle \mid j = 1, \dots, 4 \} \\
\omega(\langle p_i, t_j \rangle) &= d_{ij}, i, j = 1, \dots, 4 \\
\omega(\langle t_j, p_{j+4} \rangle) &= 1, j = 1, \dots, 4 \\
\alpha(p_1) &= 0.5, \alpha(p_2) = 0.7, \alpha(p_3) = 0.2, \alpha(p_4) = 0.0 \\
\beta(p_1) &= \beta(p_5) = 1 \\
\beta(p_2) &= \beta(p_6) = 2 \\
\beta(p_3) &= \beta(p_7) = 3 \\
\beta(p_4) &= \beta(p_8) = 4 \\
\Gamma &= (\gamma, \delta, \tau) \text{ tal que} \\
\gamma &= \delta = \min \text{ e } \tau = \max
\end{aligned}$$

Tabela 3.2: Exemplo de definição formal de uma RPN Marcada.

grupo poderia disparar, desabilitando as demais. Como o disparo de uma regra no sistema sendo modelado requer o disparo de um grupo de transições, define-se uma nova lei de disparos para grupos de transições. Arcos saindo de um lugar  $p_i$  e atingindo transições diferentes no mesmo grupo representam um fluxo paralelo de informações e devem ser conectados uns aos outros. Estes são chamados *arcos tipo e*. Isto significa que, quando existe uma senha no lugar  $p_i$ , um "pedaço" desta senha é mandado para cada um dos arcos que saem de  $p_i$  ou, equivalentemente, o número de senhas é multiplicado pelo número de arcos que saem desse lugar e cada senha segue por um desses arcos. Quando duas ou mais regras tem a mesma variável no antecedente mas não são paralelas, os lugares associados a essa variável possuirão dois ou mais grupos de *arcos tipo e* e conectados com grupos de transições diferentes e conflitantes. Este caso será melhor discutido na seção 3.8.

Cada grupo de transições de uma RPN pode ser de um entre quatro tipos, dependendo da operação que as transições desse grupo aplicam para encontrar o novo valor da senha. Entretanto, as regras que governam as mudanças nas posições das senhas são as mesmas para todos os tipos de transições. Até agora foram introduzidas as transições de *tipo 1*, que representam o disparo propriamente dito das regras na base de conhecimento. A seguir são apresentadas as regras para disparo dos quatro tipos de transições. Os significados das transições do tipo 2, 3 e 4, bem como suas aplicações serão dados nas seções 3.5 e 3.7.

**Definição 3.3.4** As transições de uma RPN disparam de acordo com as seguintes regras:

- Transições pertencentes ao mesmo grupo devem disparar em conjunto.
- O disparo de uma transição habilitada  $t$  remove  $1/n$  de senha de cada lugar de entrada  $p$  de  $t$  e adiciona uma senha a cada um de seus lugares de saída, onde  $n = |O(p)|$ .
- Seja  $t_j$  uma transição,  $O(t_j) = p_r$  e  $I(t_j) = \{p_1, \dots, p_n\}$ . O valor da senha em  $p_r$  depois do disparo de  $t_j$ , é dado por:

1. Se  $t_j$  é uma transição do tipo 1 e  $\alpha(p_i) = a'_i$ , então

$$\alpha(p_r) = b'_j = \sup_i (a'_i \gamma d_{ij})$$

onde  $d_{ij}$  é o peso do arco conectando o lugar  $p_i$  a transição  $t_j$ .

2. Se  $t_i$  é uma transição do tipo 2 e  $\alpha(p_i) = a^i_{si}$ ,  $si \in \{1, \dots, k\}$ , então

$$\alpha(p_r) = \delta_{i=1, \dots, n} (a^i_{si})$$

onde  $k$  é o número de elementos no conjunto base  $X^i$ .

3. Se  $t_i$  é uma transição do tipo 3, então

$$\alpha(p_r) = \alpha(p)$$

onde  $I(t_j) = \{p\}$ .

4. Se  $t_4$  é uma transição do tipo 4 e  $\alpha(p_i) = b^i_{si}$ ,  $si \in \{1, \dots, k\}$ , então

$$\alpha(p_r) = b'_j = \tau_{i=1, \dots, n} (b^i_{si})$$

onde  $k$  é o número de elementos no conjunto base  $Y$ .

As RPN Marcadas correspondentes aos estados anterior e posterior ao disparo de uma regra condicional como a de (3.3) são ilustradas nas figuras 3.4(a) e (b). Os elementos  $b'_j$  indicam o valor de pertinência de  $y_j$  em  $B'$ .

### 3.4 Encadeamento de Regras

O encadeamento de regras nebulosas requer a seguinte condição de consistência: duas regras nebulosas como

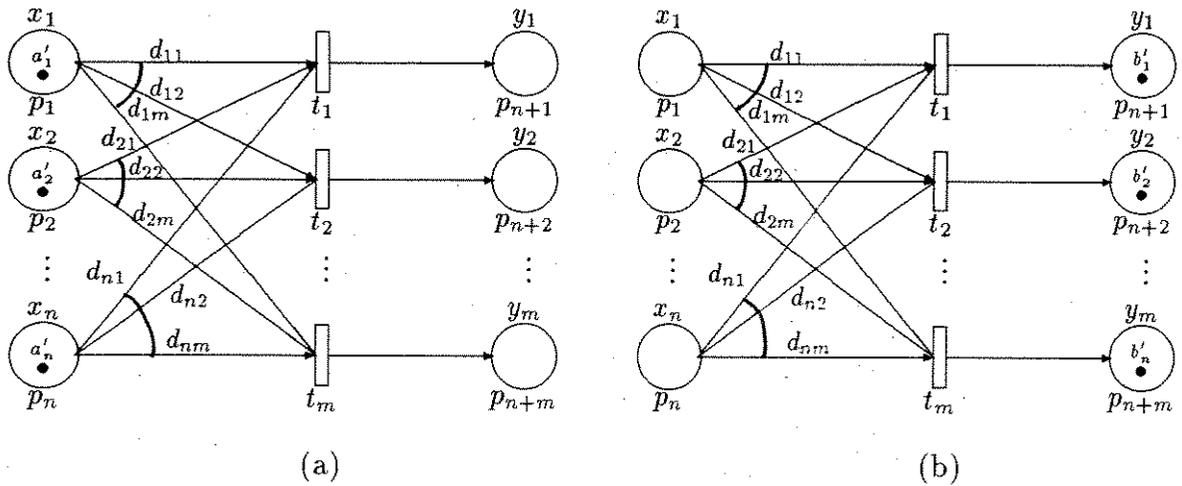


Figura 3.4: Disparo de uma RPN Marcada. (a) Antes do disparo das transições  $t_1, t_2, \dots, t_m$ . (b) Depois do disparo das transições  $t_1, t_2, \dots, t_m$ .

$$P_1 : \text{SE } V \text{ é } A \text{ ENTÃO } U \text{ é } B \text{ e}$$

$$P_2 : \text{SE } U \text{ é } B \text{ ENTÃO } W \text{ é } C$$

devem ser compatíveis com a regra nebulosa

$$P_3 : \text{SE } V \text{ é } A \text{ ENTÃO } W \text{ é } C$$

de acordo com algum critério pré-estabelecido. As propriedades dos métodos de inferência inclusive a satisfação da condição de consistência acima, tem sido investigadas por vários autores, por exemplo Mizumoto et all. [36]. Se a proposição  $P_3$  é dedutível das proposições  $P_1$  e  $P_2$ , então diz-se que o silogismo é válido. Estudos desenvolvidos em [36] mostram se o silogismo é verificado ou não sob vários métodos de raciocínio nebuloso.

Entretanto, conforme afirmado em [18], o fato de que em geral duas regras encadeadas não são equivalentes a uma única regra obtida eliminando-se a condição intermediária não é tão preocupante. O resultado obtido pelo disparo das regras encadeadas não está errado, mas apenas mais incerto que o resultado obtido por uma única regra. O modelo de RPN permite a representação de qualquer um dos métodos de inferência. A verificação da condição de consistência deve ser uma preocupação do projetista, dentro de cada domínio de aplicação. Este importante tópico de compatibilidade de regras encadeadas não será mais discutido neste trabalho por ser uma propriedade de cada método de raciocínio, e não uma característica da RPN em si.

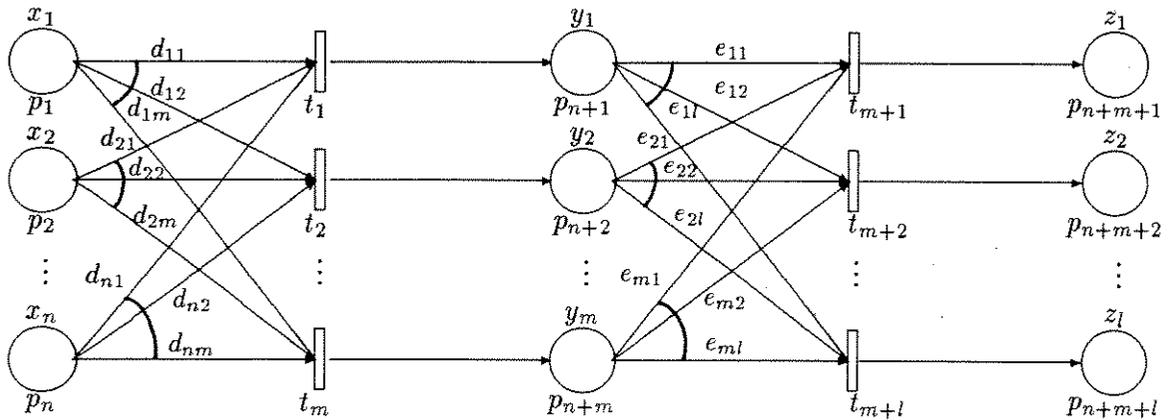


Figura 3.5: RPN representando duas regras encadeadas.

Em uma RPN, regras encadeadas são representadas associando-se o grupo de lugares que representam o consequente da primeira regra ao antecedente da segunda regra, e assim por diante. Duas regras encadeadas como  $P_1$  e  $P_2$  acima podem ser representadas em uma rede como a da figura 3.5. Assume-se aqui que  $Z = \{z_1, z_2, \dots, z_l\}$  é um conjunto base,  $W$  é uma variável sobre  $Z$ ,  $C$  é um subconjunto nebuloso de  $Z$  e a relação nebulosa entre  $B$  e  $C$  é dada por  $E_{B \rightarrow C}(y_j, z_k) = e_{jk}$ , além das mesmas condições assumidas para a representação da regra em (3.1). Nesta rede o disparo do primeiro grupo de transições habilita o segundo grupo de transições e assim por diante.

### 3.5 Regras com Múltiplos Antecedentes

Formas mais complexas de regras podem facilmente ser representadas nesta abordagem. Considere-se a regra

$$\text{SE } V_1 \text{ é } A_1 \text{ e } V_2 \text{ é } A_2 \text{ e } \dots \text{ e } V_k \text{ é } A_k \text{ ENTÃO } U \text{ é } B \quad (3.5)$$

onde  $V_1, V_2, \dots, V_k$  são variáveis assumindo valores nos conjuntos base  $X^1, X^2, \dots, X^k$  respectivamente e  $U$  é uma variável sobre o conjunto base  $Y$ . Inicialmente relembremos (seção 2.2.5) que a proposição composta do antecedente induz uma relação  $H$  sobre  $X^1 \times X^2 \times \dots \times X^k$  tal que

$$H(x^1, x^2, \dots, x^k) = \star_{i=1, \dots, k} [A_i(x^i)]. \quad (3.6)$$

e que a regra em (3.5) induz uma relação  $F$  sobre  $X^1 \times X^2 \times \dots \times X^k \times Y$  tal que

$$F(x^1, x^2, \dots, x^k, y) = f_{\rightarrow}(H(x^1, x^2, \dots, x^k), B(y)) \quad (3.7)$$

Este tipo de regra composta pode ser modelado na RPN usando-se um grupo adicional de lugares e um de transições, localizados entre os lugares associados ao antecedente e as transições associadas a regra, posição que chamamos informalmente de nível intermediário. Este novo grupo constitui-se de transições do tipo 2 (ver seção 3.3) e tem a função de agregar os conjuntos das proposições do antecedente. Por isso utilizam uma maneira diferente de calcular o valor das senhas por ocasião do disparo, conforme a definição (3.3.4). Os rótulos dos arcos entre os primeiros  $kn$  lugares e as transições do tipo 2 são definidos com o valor 1. Os rótulos dos arcos que conectam lugares no nível intermediário às transições do tipo 1 são definidos pelos valores da função de pertinência da relação nebulosa dada em (3.7). Assumindo  $|X^1| = |X^2| = \dots = |X^k| = n$  e  $|Y| = m$ , temos  $\omega(p_i, t_j) = 1$ , para  $i = 1, \dots, kn$  e  $j = 1, \dots, n^k$  e  $\omega(p_{kn+i}, t_{kn+j}) = F(x^1, x^2, \dots, x^k, y)$ , para  $i = 1, \dots, n^k$  e  $j = 1, \dots, m$ .

A modelagem da regra em (3.5) é apresentada na figura 3.6, quando  $n = 3$  e  $k = 2$ , onde as transições do tipo 2 são representadas por barras pretas.

Agora, assumamos que temos os fatos

$$V_1 \text{ é } A'_1, V_2 \text{ é } A'_2, \dots, V_k \text{ é } A'_k \quad (3.8)$$

Sabe-se que (seção 2.2.5) o valor obtido para  $U$  a partir das premissas em (3.8) e (3.5) é dado por

$$B' = H' \bullet F$$

onde

$$H'(x^1, x^2, \dots, x^k) = \star_{i=1, \dots, k} [A'_i(x^i)].$$

Como no caso de um único antecedente, representa-se os fatos em (3.8) na rede inserindo-se senhas com seus valores correspondentes nos lugares associados com as variáveis  $V_1, V_2, \dots, V_k$ , respectivamente. O  $i$ -ésimo elemento de  $A'_i$  é denotado por  $a_i^j$  (veja figura 3.7). O disparo do primeiro grupo de transições (transições do tipo 2) corresponde ao cálculo da relação  $H'$  envolvendo os subconjuntos nebulosos  $A'_i$ . Os valores das senhas nos lugares  $p_{kn+1}, p_{kn+2}, \dots, p_{kn+a}$  ( $a = n^k$ ) logo após este disparo são dados por (definição 3.3.4)

$$h_{s_1, s_2, \dots, s_k} = \star_{i=1, \dots, k} (a_{s_i}^i)$$

onde  $s_i = 1, \dots, n$ . A nova marcação da rede é ilustrada na figura 3.8(a). Esta marcação habilita o segundo grupo de transições (transições do tipo 1), cujo disparo corresponde ao disparo da regra propriamente dita. Os valores das senhas são obtidos pela equação já descrita no exemplo de regra simples (com um único antecedente). A marcação final está representada na figura 3.8(b).

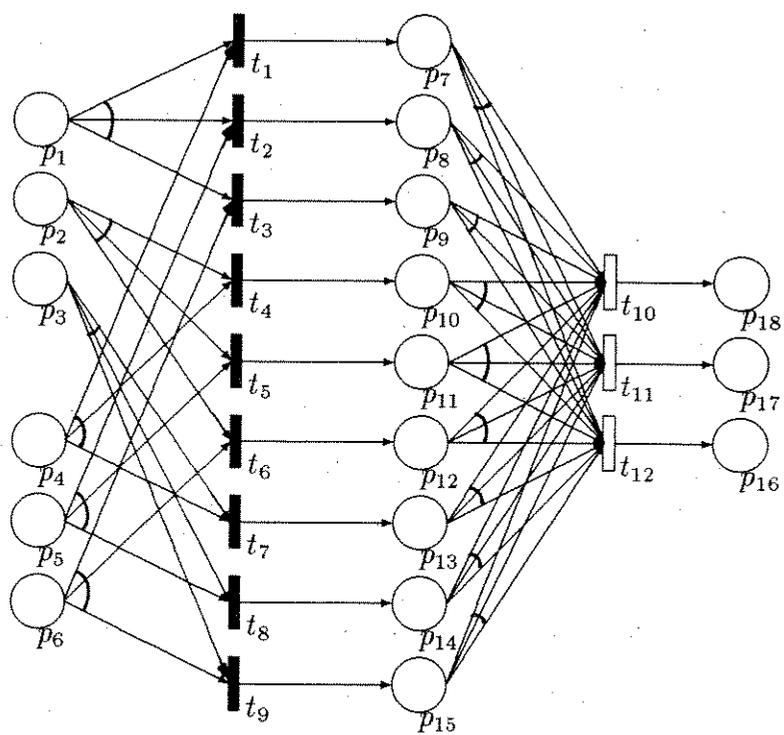


Figura 3.6: RPN representando uma regra com conjunção no antecedente. Os pesos dos arcos foram omitidos.

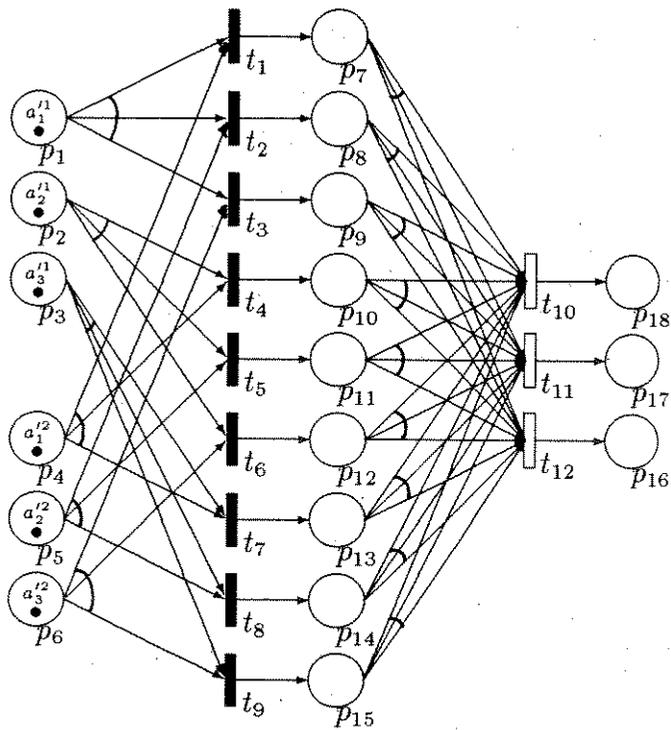


Figura 3.7: RPN Marcada representando regras com conjunção no antecedente.

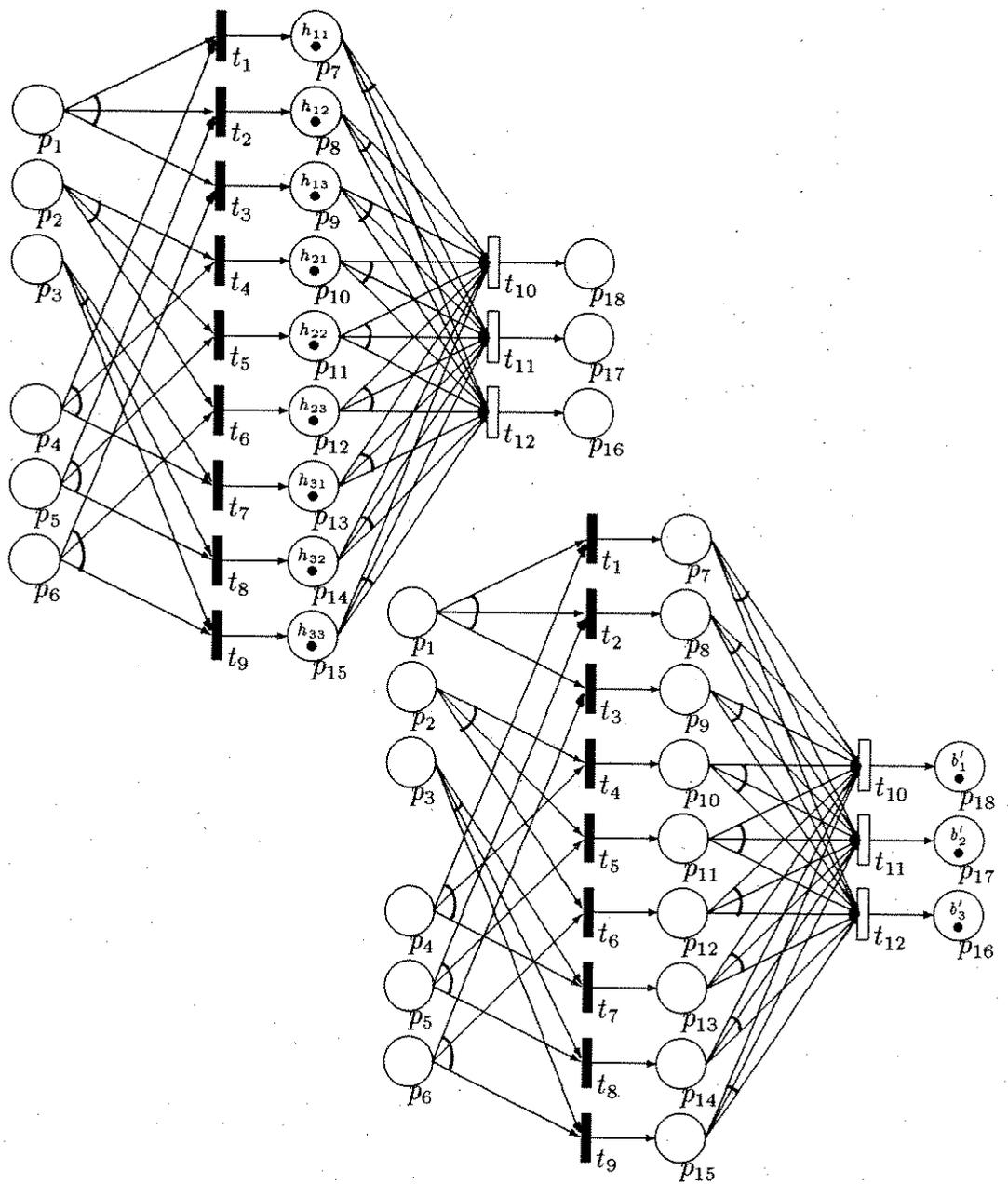


Figura 3.8: Disparo de uma RPN Marcada representando regras com conjunção no antecedente. (a) Após o disparo do primeiro grupo de transições. (b) Após o disparo do segundo grupo de transições.

## 3.6 Regras Quantificadas e Qualificadas

### 3.6.1 Quantificadores no Antecedente

Nesta seção será mostrado como uma extensão das idéias apresentadas previamente possibilita a representação de regras nebulosas mais sofisticadas que podem figurar em uma base de regras, tais como:

Se a maioria das condições  $V_1$  é  $A'_1$ ,  $V_2$  é  $A'_2$ , ...,  $V_k$  é  $A'_k$  está satisfeita então  $U$  é  $B$ .

Se pelo menos metade das condições  $V_1$  é  $A'_1$ ,  $V_2$  é  $A'_2$ , ...,  $V_k$  é  $A'_k$  está satisfeita então  $U$  é  $B$ .

Uma metodologia para representação de tais regras é fornecida por Yager [71], com base na representação de quantificadores sugerida por Zadeh. A seguir, as idéias fundamentais desta metodologia são brevemente descritas e, então, mostramos como este tipo de regra pode ser modelado usando-se o modelo de RPN.

A diferença fundamental entre esse tipo de regra e os tipos estudados nas seções anteriores é que, em vez de exigir-se que todas as condições do antecedente sejam satisfeitas, exige-se que apenas  $Q$  dessas condições sejam satisfeitas para que a regra dispare.

Como os outros tipos de regras condicionais, essas regras também induzem uma relação nebulosa  $F$  sobre o conjunto  $X^1 \times X^2 \times \dots \times X^k \times Y$ . Em particular, para todo ponto  $(x^1, x^2, \dots, x^k, y)$  onde  $x^i \in X^i$  e  $y \in Y$ ,

$$F(x^1, x^2, \dots, x^k, y) = f_-(H(x^1, x^2, \dots, x^k), B(y)).$$

A diferença essencial consiste na determinação da relação  $H(x^1, x^2, \dots, x^k)$ , o componente relacionado ao antecedente da regra.

De acordo com Yager [71], existem três tipos de quantificadores, dos quais os dois primeiros são de interesse para esta abordagem. Um quantificador de tipo 1 ou quantificador *absoluto* tal como *cerca de cinco* e *pelo menos sete* pode ser representado como um subconjunto nebuloso do conjunto dos números reais não negativos. Um quantificador do tipo 2 ou *relativo* é exemplificado por valores como *quase todos* e *pelo menos metade*. Estes podem ser representados por um subconjunto nebuloso do intervalo unitário. A discussão que se segue está restrita a estes dois tipos de quantificadores monotonicamente não decrescentes.

Assuma que  $Q$  é um quantificador do tipo 1. Para todo ponto

$$(x^1, x^2, \dots, x^k) \in X^1 \times X^2 \times \dots \times X^k$$

onde  $X^i$  é o conjunto base de  $A_i$  obtém-se  $H(x^1, x^2, \dots, x^k)$  pelo processo descrito a seguir. Seja

$$D(x^1, x^2, \dots, x^k) = \{A_1(x^1), A_2(x^2), \dots, A_k(x^k)\}.$$

Seja

$$D_i(x^1, x^2, \dots, x^k)$$

o  $i$ -ésimo maior elemento do conjunto  $D(x^1, x^2, \dots, x^k)$ .

Então, para qualquer quantificador absoluto  $Q_1$

$$H(x^1, x^2, \dots, x^k) = \max_{i=1, \dots, k} [Q_1(i) \wedge D_i(x^1, x^2, \dots, x^k)]. \quad (3.9)$$

Se  $Q_1$  for um quantificador relativo, substitui-se  $Q_1(i)$  por  $Q_1(i/n)$ .

Na RPN, os valores acima serão os rótulos dos arcos, isto é, os valores da função  $\omega$  (definição 3.3.1).

Além disso, se são conhecidos os valores

$$V_1 \text{ é } A'_1, V_2 \text{ é } A'_2, \dots, V_k \text{ é } A'_k$$

onde  $A'_i$  é um subconjunto nebuloso de  $X^i$ ,  $i = 1, \dots, k$ , obtém-se um valor para  $U$  como

$$U \text{ é } B'$$

onde

$$B'(y) = \max[F(x^1, x^2, \dots, x^k, y) \star H'(x^1, x^2, \dots, x^k)]$$

e

$$H'(x^1, x^2, \dots, x^k) = \star_{i=1, \dots, k} [A_i(x^i)].$$

De acordo com esta metodologia, a representação na RPN de uma regra com quantificadores no antecedente é feita utilizando-se a mesma estrutura básica do modelo de regras com antecedentes múltiplos. A única diferença resume-se a maneira como os rótulos  $\omega_i$  são calculados.

### 3.6.2 Qualificação de Certeza

O usuário de um sistema pode não estar completamente certo quanto aos valores que ele fornece para uma variável quando as regras de uma base de conhecimento estão sendo definidas. Logo, a informação pode ser dada da seguinte forma:

$V$  é  $A$  com certeza  $\mu$ .

onde a quantidade  $\mu$  é um número no intervalo unitário que expressa o grau com que o informante acredita que a sua informação é válida.

Yager([71]) desenvolveu uma abordagem para incluir esse tipo de proposição qualificada em sistemas baseados em regras, com base na suposição de que

$V$  é  $A$  com certeza  $\mu$

é equivalente a uma proposição

$V$  é  $C$

com grau de certeza igual a um. Esta abordagem pode ser aplicada a regras da seguinte maneira. Considere-se a regra

SE  $V$  é  $A$  ENTÃO  $U$  é  $B$  com certeza  $\mu$

onde  $A$  e  $B$  são subconjuntos nebulosos de  $X$  e  $Y$ , respectivamente. Esta regra transforma-se na relação

$$R(x, y) = (H(x, y) \wedge \mu) + (1 - \mu)$$

onde

$$H(x, y) = f_{-}(A(x), B(y)).$$

Quanto a representação em RPN, a mesma estrutura básica da rede que representa regras simples pode ser usada, sendo que a única diferença está na determinação da matriz de relação nebulosa entre  $A$  e  $B$ , ou seja, no cálculo dos rótulos dos arcos.

### 3.7 Regras Paralelas

Regras da forma

$$\text{SE } V \text{ é } A_i \text{ ENTÃO } U \text{ é } B_i \text{ para } i = 1, \dots, r$$

são chamadas *regras paralelas*. O ponto de vista mais comum na interpretação de regras paralelas é modelar cada regra por meio de um operador de implicação e agregar os resultados de todas as regras por um operador de união. Assim o conjunto de regras pode também ser escrito como

$$\cup_{i=1}^r \text{SE } V \text{ é } A_i \text{ ENTÃO } U \text{ é } B_i$$

Para implementar o conjunto de regras acima, uma função de implicação global é necessária. Suponha que a relação nebulosa entre  $A_i$  e  $B_i$  sobre o conjunto base  $X \times Y$  é representada por  $D_{A_i \rightarrow B_i}$ , tal que

$$D_{A_i \rightarrow B_i}(x, y) = f_{\rightarrow}(A_i(x), B_i(y))$$

A relação nebulosa global  $D$  é dada por

$$D(x, y) = \mathbf{s}_{i=1}^r(D_{A_i \rightarrow B_i}(x, y)).$$

Dado  $A'$ , o conseqüente  $B'$  é calculado por:

$$\begin{aligned} B' &= A' \bullet D \\ B'(y) &= \sup_x \star(A'(x), D(x, y)) \end{aligned}$$

ou seja,

$$B'(y) = \sup_x \star(A'(x), \mathbf{s}_{i=1}^r(f_{\rightarrow}(A_i(x), B_i(y)))).$$

Esta equação, que pode ser reescrita como

$$B' = A' \bullet \cup_{i=1}^r (A_i \rightarrow B_i),$$

representa que a inferência é realizada globalmente [18].

Uma forma alternativa para implementar regras paralelas é realizar cada inferência  $A', A_i \rightarrow B_i \vdash B'_i$  separadamente e então combinar os  $(B'_i)$ 's, como indicado por

$$B' = \cup_{i=1}^r A' \bullet (A_i \rightarrow B_i).$$

No caso em que as regras são modeladas especificamente via operador de mínimo e a coleção de regras é agregada através do operador de máximo temos:

$$A' \circ [\cup(A_i \times B_i)] = \cup_i [A'_i \circ (A_i \times B_i)].$$

Nos casos em que operadores diferentes são usados, o resultado obtido pela aplicação de inferência global pode não ser exatamente o mesmo que o resultado obtido pela aplicação da inferência local. A relação entre essas duas formas alternativas de interpretar regras paralelas não pertence ao escopo deste trabalho. O modelo de RPN permite a representação de ambos os casos e o tratamento mais apropriado deve ser definido considerando-se o contexto da aplicação específica.

Quando o princípio de combinação/projeção é executado globalmente, as regras são transformadas em uma única regra pela agregação das matrizes de relação  $D_{A_i \rightarrow B_i}$  em uma matriz de relação global  $D_{A \rightarrow B}$ . Logo, a RPN que representa esse conjunto de regras paralelas tem a mesma estrutura básica da rede que representa uma única regra, seja ela simples ou com múltiplos antecedentes, sendo que a única diferença está na forma de calcular os valores dos pesos dos arcos, como acontece com a representação de regras qualificadas, discutida na seção 3.6. Quando o princípio de combinação/projeção é realizado localmente, dois novos tipos de transições são necessárias. As transições do tipo 3 fazem reproduções de senhas, para que todas as regras paralelas tenham seu grupo de transições correspondente habilitados ao disparo. As transições do tipo 4 tem a função de aplicar o operador de agregação nos conjuntos  $B'_i$ ,  $i = 1, \dots, r$ , resultantes de cada inferência individual.

O conjunto de regras

$$\text{SE } V \text{ é } A_i \text{ ENTÃO } U \text{ é } B_i,$$

para  $i=1, 2$ , e  $|X^i|=|Y|=3$ , pode ser representado como mostra a figura 6.3(a). As transições  $t_1, t_2$  e  $t_3$  são transições do tipo 3 e o valor das senhas nos lugares de saída depois do disparo dessas transições é exatamente o mesmo que o valor das senhas no lugar de entrada antes do disparo das mesmas. Note-se que transições do tipo 3 tem sempre apenas um único lugar de entrada. Assim que o primeiro grupo de transições disparou, todos os grupos de transições que representam regras no conjunto de regras paralelas estão habilitados e podem ser disparados em paralelo. Os resultados fornecidos pelas  $r$  regras paralelas são então agregados por um operador do tipo T-norma ou T-conorma, implementado pelas transições do tipo 4 ( $t_{10}, t_{11}$  e  $t_{12}$ ).

A figura 6.3(b) ilustra uma representação de regras paralelas mais compacta e eficiente sem a utilização das transições do tipo 3. As senhas são enviadas diretamente para os grupos de regras paralelas, sendo que a multiplicação das mesmas

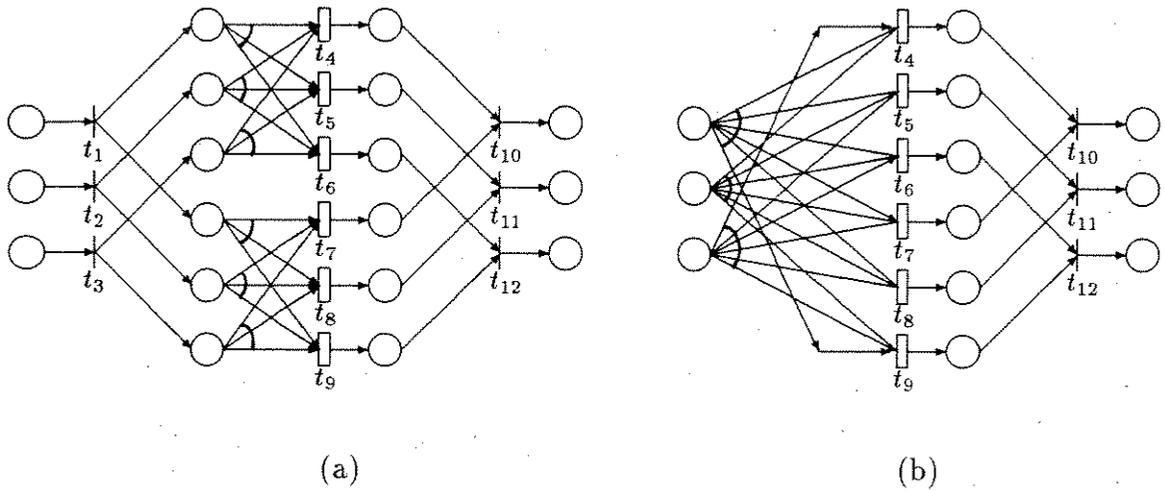


Figura 3.9: Estrutura básica de RPN representando duas regras paralelas com  $|X^i|=|Y|=3$ , quando o princípio de combinação/projeção é realizado localmente. (a) Com transições do tipo 3. (b) Forma simplificada.

está representada implicitamente pelos *arcos tipo e*. A estrutura básica da figura 6.3(a) foi incluída por sua utilidade do ponto de vista de projeto de bases de regras nebulosas, já que essa representação permite que os seus elementos mantenham uma relação de um para um com os elementos da rede de alto nível equivalente a ser definida no capítulo 4.

### 3.8 Regras Conflitantes

Regras conflitantes são aquelas que possuem a mesma variável no antecedente e variáveis diferentes no conseqüente. Além disso é exigido que essas regras dêem origem a caminhos de raciocínio independentes, isto é, caminhos que não se relacionam nas etapas posteriores de execução da rede. Este ponto será melhor discutido na seção 3.10.

Para apresentar o processo de modelagem de regras conflitantes na FPN, considere-se inicialmente o caso de regras com um único antecedente:

SE  $V$  é  $A$  ENTÃO  $U$  é  $B$   
SE  $V$  é  $C$  ENTÃO  $W$  é  $D$

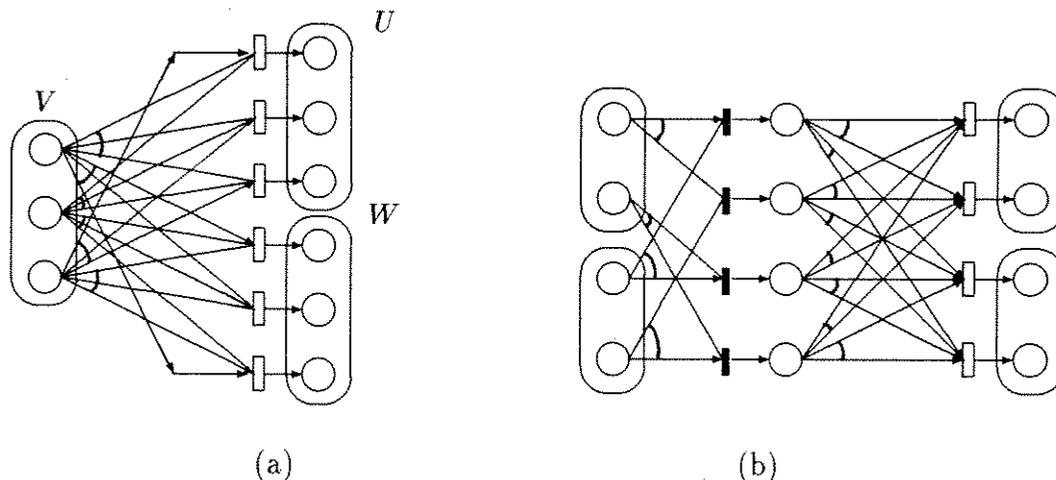


Figura 3.10: RPN representando regras conflitantes. (a) Regras com um único antecedente. (b) Regras com antecedentes múltiplos.

Modela-se esse tipo de regra colocando-se um único grupo de lugares representando o antecedente de ambas as regras, um grupo de lugares para cada um dos consequentes e um grupo de transições para cada regra. O grupo de lugares associado ao antecedente é conectado a cada grupo de transições por um conjunto independente de *arcos tipo e*, o que indica uma situação de conflito do ponto de vista de execução da rede. Escolhe-se apenas um dos conjuntos de *arcos tipo e* para propagar a informação trazida pela senha. A RPN representando o conjunto de regras acima é vista na figura 3.10(a). Dos  $n$  primeiros lugares, associados com a variável  $V$  dos antecedentes das regras, emergem dois conjuntos de *arcos tipo e*, cada um deles conectado a um dos grupos de transições associados às regras.

Com relação ao caso de antecedentes múltiplos, uma divisão dos caminhos de raciocínio também ocorre depois da execução da conjunção dos antecedentes. Como pode ser visto na figura 3.10(b), os lugares de saída das transições tipo 2, que representam os valores agregados dos antecedentes dão origem a dois conjuntos conflitantes de *arcos tipo e*. Na figura 3.10 os grupos de lugares foram circundados por ovais para maior clareza.

Neste ponto faz-se necessário esclarecer um aspecto importante. O termo conflitante foi adotado aqui com a finalidade de se manter a terminologia usada no contexto de Redes de Petri convencionais. Assim que uma situação (marcação) inicial é definida, uma particular sequência de disparos contém somente um desses grupos conflitantes. Na base de regras sendo modelada, um conflito representa o

início de caminhos de raciocínio diferentes. Entretanto, o algoritmo de raciocínio para frente gera automaticamente todos os possíveis caminhos a partir de um mesmo estado inicial.

Conjuntos de regras paralelas ou conflitantes implicam no problema de integridade entre as regras. A questão de verificação de inconsistência, um dos problemas que podem ocorrer em uma base de regras nebulosas, será tratada no capítulo 6.

Outro relevante conjunto de regras que pode aparecer em uma base de regras nebulosas são aquelas que possuem variáveis diferentes no antecedente e a mesma variável no conseqüente. Estas regras são chamadas de *fontes parciais de informação* e sua representação será discutida com base no modelo apresentado no capítulo 4.

### 3.9 Matriz de Incidência e Equação de Estado

Para Redes de Petri regulares, uma das abordagens de análise da rede é baseada em matrizes. As equações matriciais descrevem o comportamento dinâmico do sistema, governando as mudanças que ocorrem no número e na posição das senhas como resultado dos disparos das transições. O estado de uma RPN, no qual apenas uma senha pode estar presente em cada lugar ao mesmo tempo, é definido pela sua marcação  $\alpha$ , que fornece não apenas posições mas também valores das senhas. O disparo de um grupo de transições representa uma mudança na marcação da rede, alterando as posições e valores das senhas. Para descrever completamente o comportamento dinâmico de uma RPN seria necessário definir uma técnica que governasse ambos os aspectos da mudança na marcação, o que exigiria um ferramental matemático bastante poderoso. As equações matriciais definidas nesta seção captam uma parcela apenas das mudanças no comportamento da rede, relativas a posição das senhas. A matriz de incidência definida a seguir pode ser utilizada em algoritmos de raciocínio para trás.

**Definição 3.9.1** *Para uma RPN  $N$  com  $n$  lugares e  $m$  transições, a matriz de incidência  $C = [c_{ij}]$  é uma matriz  $n \times m$  de reais tal que cada linha corresponde a um lugar  $p$  de  $N$  e cada coluna corresponde a uma transição  $t$  de  $N$ . Seus elementos são dados por*

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

onde

$$c_{ij}^+ = \begin{cases} 1 & \text{se existe um arco da transição } i \text{ para seus lugares de saída } j \\ 0 & \text{caso contrário} \end{cases}$$

$$c_{ij} = \begin{cases} 1/b & \text{se existe um arco do lugar de entrada } j \text{ para a transição } i \\ 0 & \text{caso contrário} \end{cases}$$

onde  $b =$  número de arcos tipo  $e$  que saem do lugar  $j$  e incluem o arco  $(i, j)$ .

Para escrever equações matriciais, denota-se uma marcação  $M_k$  como um vetor  $m \times 1$ . A  $j$ -ésima posição de  $M_k$  representa o número de senhas no lugar  $j$  imediatamente após o  $k$ -ésimo disparo em alguma sequência de disparos. O  $k$ -ésimo vetor de controle ou de disparo  $u_k$  é um vetor  $m \times 1$  de  $m - l$  elementos iguais a zero e  $l$  elementos diferentes de zero,  $l = i, \dots, i + p$ , tal que as transições de  $i$  até  $i + p$  representam um grupo de transições. Os 1's nas posições  $i$  até  $i + p$  indicam que essas transições disparam juntas no  $k$ -ésimo disparo. Para simplificar a notação, assume-se que as transições de cada grupo estão nomeadas em sequência. As definições podem ser facilmente estendidas para o caso geral.

**Definição 3.9.2** Assumindo que a  $i$ -ésima coluna da matriz de incidência  $C$  denota a mudança de marcação que resulta do disparo da transição  $i$ , define-se a equação de estado para uma Rede de Petri Nebulosa como:

$$M_k = M_{k-1} + C \cdot u_k \quad k = 1, 2, \dots \quad (3.10)$$

**Definição 3.9.3** Suponha que uma marcação destino  $M_d$  é atingível de  $M_0$  por uma sequência de disparos  $u_1, u_2, \dots, u_d$ . Escrevendo-se a equação de estados (3.10) para  $i = 1, 2, \dots, d$  e somando-as, obtém-se a equação abaixo, chamada de Condição Necessária de Alcançabilidade:

$$M_d = M_0 + C \cdot \sum_{k=1}^d u_k$$

que pode ser reescrita como

$$Cx = \Delta M$$

onde  $\Delta M = M_d - M_0$  e  $x = \sum_{k=1}^d u_k$ . Aqui,  $x$  é um vetor coluna de inteiros não negativos  $n \times 1$ , e é chamado de vetor de contagem de disparos.

**Exemplo 3.9.1** Considere a RPN da figura 3.5. A equação de estados onde as transições  $t_1, t_2$  e  $t_3$  disparam resultando na marcação  $M_1 = (0, 0, 0, 1, 1, 1, 0, 0, 0)$  a

partir da marcação  $M_0 = (1, 1, 1, 0, 0, 0, 0, 0, 0)$ , é dada por:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1/3 & -1/3 & -1/3 & 0 & 0 & 0 \\ -1/3 & -1/3 & -1/3 & 0 & 0 & 0 \\ -1/3 & -1/3 & -1/3 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1/3 & -1/3 & -1/3 \\ 0 & 1 & 0 & -1/3 & -1/3 & -1/3 \\ 0 & 0 & 1 & -1/3 & -1/3 & -1/3 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Esta nova marcação habilita o grupo formado pelas transições  $t_4, t_5, t_6$  e o disparo resulta em  $M_2 = (000000111)$  de  $M_1$ .

### 3.10 Exemplo de aplicação

Nas seções anteriores foram definidas as estruturas básicas de modelagem de diferentes formas de regras. No contexto de uma base de conhecimento em que as regras estão relacionadas entre si, alguns detalhes devem ser considerados para integrar corretamente pequenos módulos em uma rede maior. Nesta seção serão apresentadas algumas situações em que essas estruturas básicas são combinadas para produzir a rede que representa a base de conhecimento completa. Com esse objetivo, será usada como exemplo uma base de conhecimento composta de regras nebulosas descritas na Tabela 3.3.

Considere-se agora as regras 7 a 13, da Tabela 3.3. As regras 7 e 8, assim como as regras 9 e 10, são regras paralelas com um único antecedente, a variável  $V_2$ . Os resultados obtidos por esses dois conjuntos de regras ( $V_3$  e  $V_4$  respectivamente), até aqui independentes, aparecem no antecedente das regras 11, 12 e 13. Por essa razão, os conjuntos de regras  $\{7,8\}$  e  $\{9,10\}$ , que quando isolados do restante da base de conhecimento seriam modelados como regras conflitantes, devem na verdade disparar em paralelo por estarem encadeadas com o mesmo conjunto de regras paralelas que tem  $V_3$  e  $V_4$  no antecedente. Caso contrário, as transições que agregam os antecedentes não estariam habilitadas, por não terem senhas em todos os lugares de entrada. A duplicação dos valores de  $V_2$  pode ser feita tanto com a utilização de transições do tipo 3 (veja figura 3.12) como com *arcos tipo* e (veja figura 3.13).

Como as duas formas alternativas para representação de regras paralelas já foram discutidas, a partir de agora adota-se uma representação hierárquica dessas regras

1)	SE $V_1$ é $S_1$ e $V_2$ é $S_2$ ENTÃO $V_6$ é $M_6$
2)	SE $V_1$ é $S_1$ e $V_2$ é $M_2$ ENTÃO $V_6$ é $S_6$
3)	SE $V_1$ é $M_1$ e $V_2$ é $L_2$ ENTÃO $V_6$ é $S_6$
4)	SE $V_1$ é $S_1$ e $V_2$ é $S_2$ ENTÃO $V_5$ é $M_5$
5)	SE $V_1$ é $M_1$ e $V_2$ é $M_2$ ENTÃO $V_5$ é $M_5$
6)	SE $V_1$ é $L_1$ e $V_2$ é $M_2$ ENTÃO $V_5$ é $L_5$
7)	SE $V_2$ é $S_2$ ENTÃO $V_3$ é $M_3$
8)	SE $V_2$ é $M_2$ ENTÃO $V_3$ é $S_3$
9)	SE $V_2$ é $M_2$ ENTÃO $V_4$ é $M_4$
10)	SE $V_2$ é $L_2$ ENTÃO $V_4$ é $S_4$
11)	SE $V_3$ é $S_3$ e $V_4$ é $M_4$ ENTÃO $V_7$ é $M_7$
12)	SE $V_3$ é $S_3$ e $V_4$ é $L_4$ ENTÃO $V_7$ é $L_7$
13)	SE $V_3$ é $M_3$ e $V_4$ é $M_4$ ENTÃO $V_7$ é $M_7$

Tabela 3.3: Exemplo de Regras Nebulosas para serem modeladas na RPN.

no modelo RPN, onde os detalhes da parte central da rede ficam escondidos. Usa-se aqui uma transição especial representada por uma retângulo chamado *bloco de regras paralelas*. Assim, regras paralelas podem ser representadas em um nível mais abstrato, como mostra a figura 3.11(b), onde o bloco retangular substitui a sub-rede central na figura 3.11(a).

Ao contrário da situação que foi discutida anteriormente, os conjuntos de regras  $\{1,2,3\}$  e  $\{4,5,6\}$  da Tabela 3.3 possuem o mesmo antecedente e consequentes diferentes e independentes. São portanto modelados como conjuntos conflitantes de regras (veja figura 3.14). A situação de conflito, portanto, não ocorre apenas entre regras individuais mas também entre conjuntos de regras.

A rede completa correspondente a todas as regras da Tabela 3.3 pode ser vista na figura 3.15.

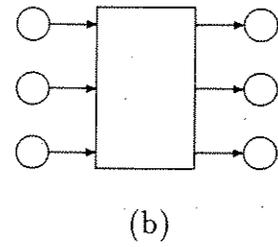
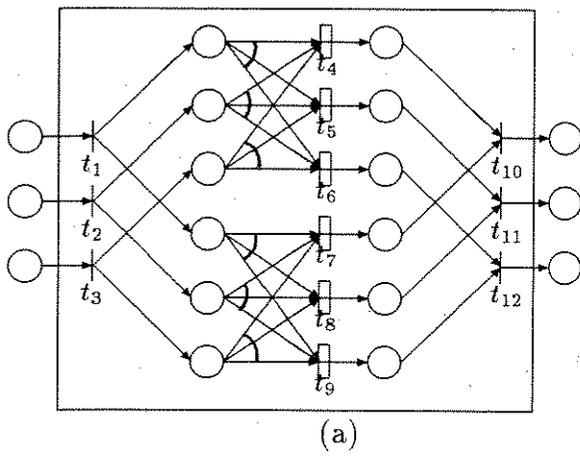


Figura 3.11: Representação hierárquica de regras paralelas.

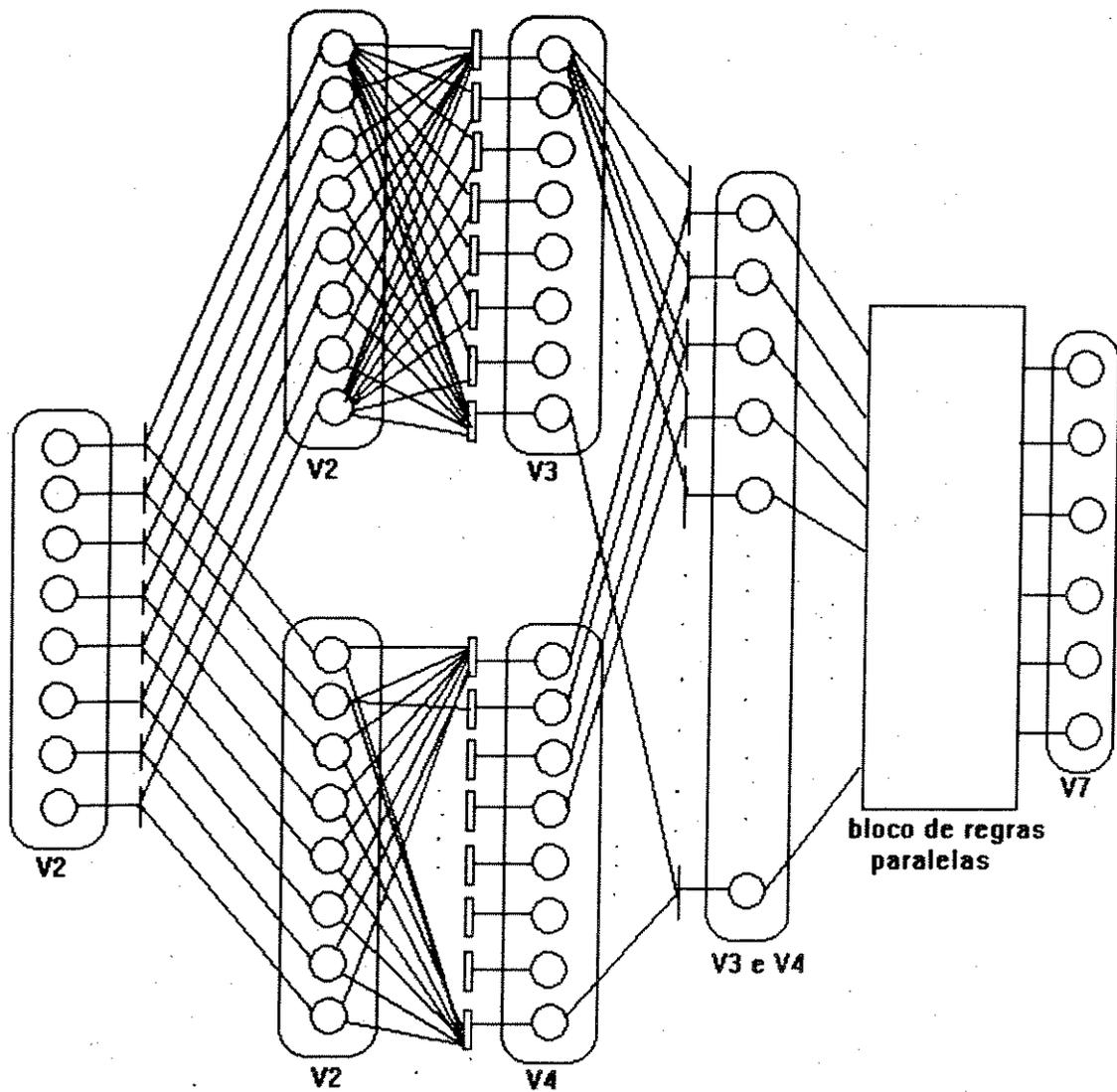


Figura 3.12: RPN representando conjuntos encadeados de regras com transições do tipo 3.

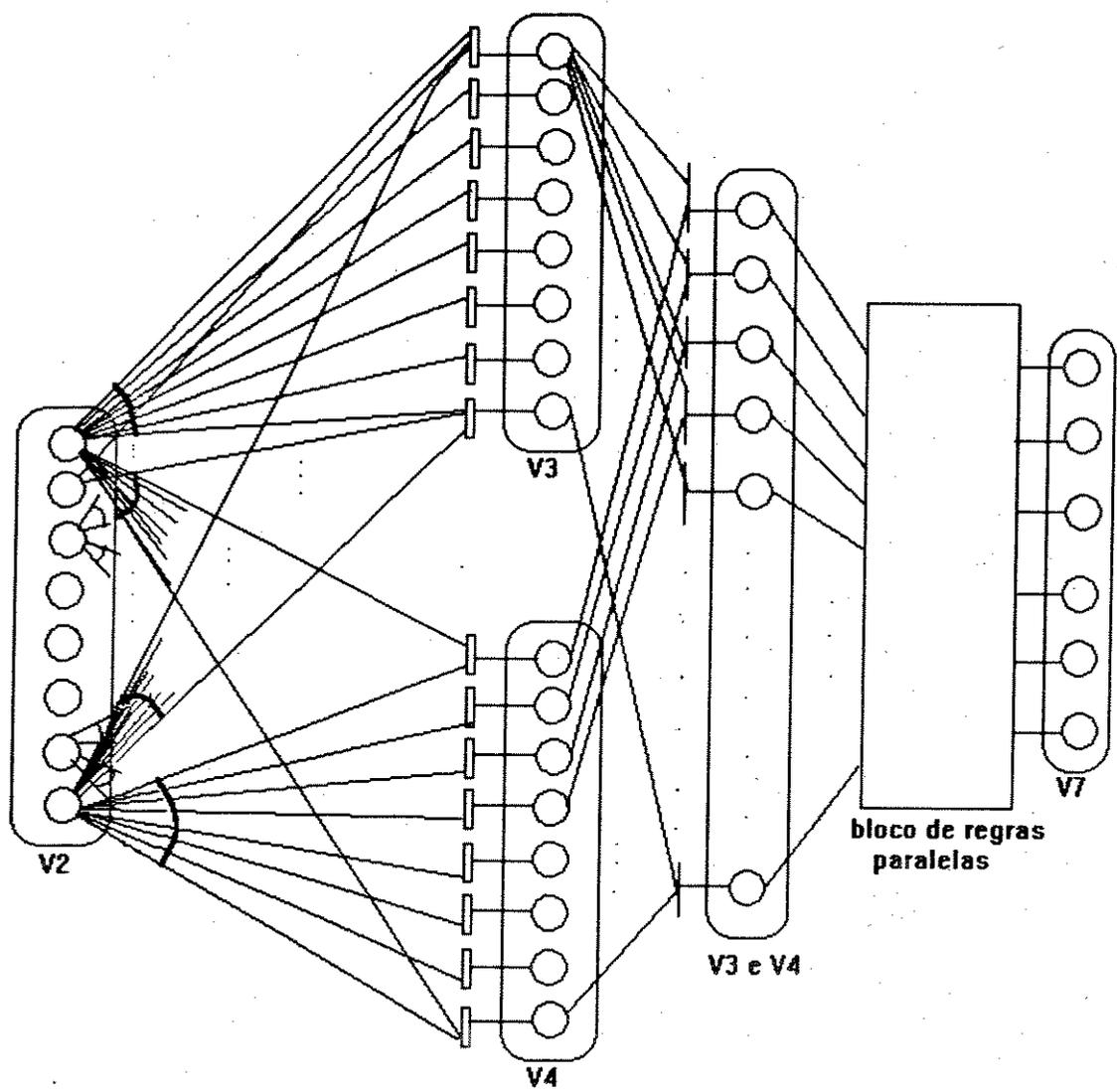


Figura 3.13: RPN representando conjuntos encadeados de regras sem transições do tipo 3.

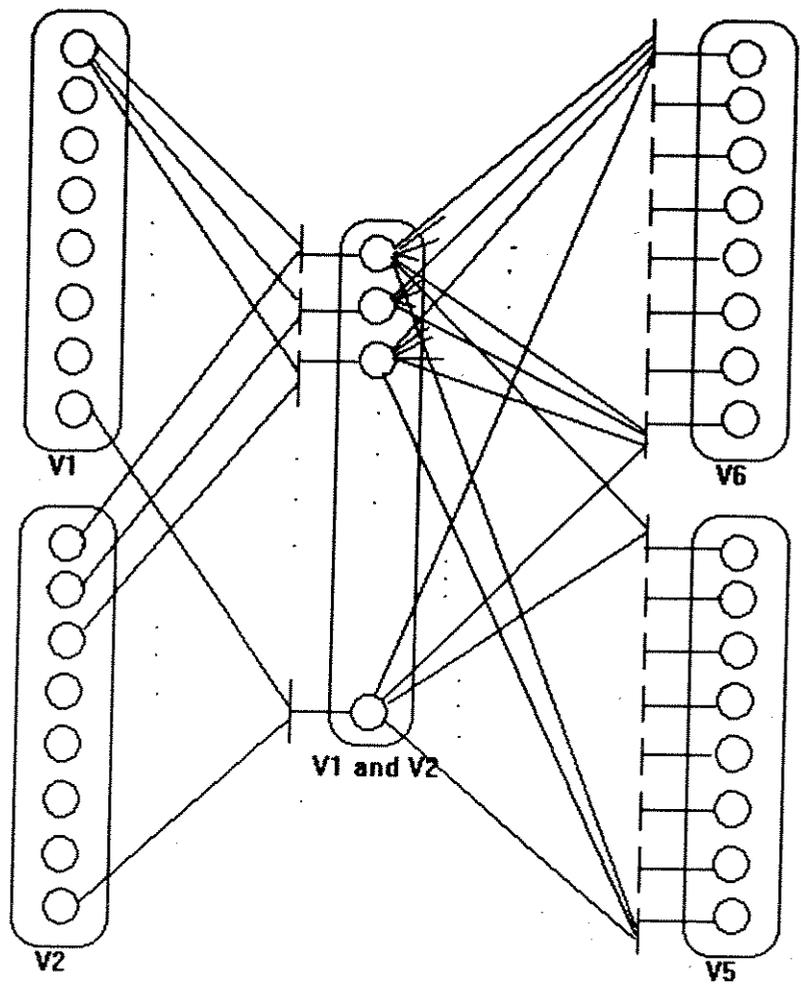


Figura 3.14: RPN representando conjuntos conflitantes de regras.

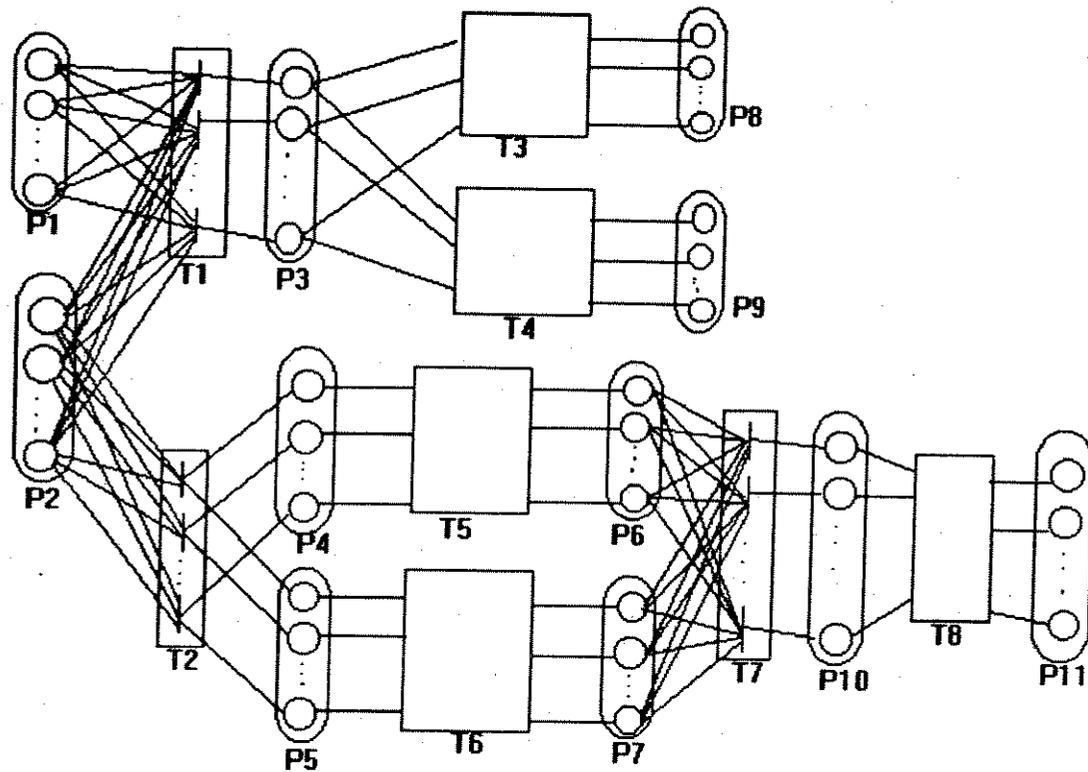


Figura 3.15: RPN representando a base de regras completa.

### 3.11 Conclusão

Neste capítulo foi apresentada a Rede de Petri Nebulosa como um modelo de representação de regras nebulosas. Vários tipos de regras podem ser modeladas na RPN, desde as mais simples até as mais complexas. A definição formal de Rede de Petri Nebulosa generalizada permite a modelagem de toda uma classe de métodos de inferência baseados em T-operadores. As questões de consistência no encadeamento de regras, característica de determinados métodos, não foi tratada neste contexto. Um exemplo de base de regras nebulosas foi utilizado na ilustração das técnicas de modelagem descritas. A RPN, apesar de ser um modelo matemático bastante poderoso, apresenta o inconveniente de que o número de lugares e transições aumenta exponencialmente em função do número de condições no antecedente das regras. Entretanto, trata-se de um problema decorrente do método de inferência utilizado (modus ponens composicional) e não da forma de modelagem proposta. Do ponto de vista de eficiência na implementação, esta desvantagem poderia ser contornada com a utilização de recursos de paralelismo, cuja discussão encontra-se fora do escopo de abrangência deste trabalho. Do ponto de vista de projeto de sistemas, a divisão dos conjuntos de lugares e transições em grupos, sugere uma representação em um nível de abstração mais alto, que torne os elementos da rede mais facilmente manuseáveis. Com este intuito, as Redes de Petri Nebulosas de Alto Nível serão introduzidas no próximo capítulo.

## Capítulo 4

# Redes de Petri Nebulosas de Alto Nível

### 4.1 Introdução

Apresenta-se neste capítulo o modelo de Redes de Petri Nebulosas de Alto Nível (RPNAN). Como já foi mencionado, as RPNAN permitem obter uma representação mais compacta e mais facilmente manuseável das RPN introduzidas no capítulo anterior, visualizando o sistema de regras em questão a partir de um nível de abstração mais elevado. As estruturas de RPNAN independem do número de elementos nos conjuntos base e portanto possibilitam uma visão gráfica mais clara da base de conhecimento sendo modelada.

Na seção 4.2 é dada a definição formal de RPNAN, os principais tipos de fórmulas que podem ser inscritas nas transições e a modelagem do padrão básico de inferência do raciocínio aproximado. Na seção 4.3 é apresentada informalmente a correspondência entre os elementos de uma RPNAN e de sua RPN equivalente. A seção 4.4 discute o encadeamento de regras. A representação de regras com antecedentes múltiplos é apresentada na seção 4.5. As regras quantificadas e qualificadas são brevemente discutidas na seção 4.6. Regras paralelas são abordadas na seção 4.7 e regras conflitantes na seção 4.8. Na seção 4.9 é apresentada a representação de fontes parciais de informação. A seção 4.10 contém uma descrição da modelagem de regras com mais de uma proposição no consequente. A utilização de conceitos conhecidos como alcançabilidade e S-invariantes aplicados a Rede de Petri associada a uma RPNAN está resumida na seção 4.11. Um exemplo de aplicação relativo à área de modelagem de sistemas de manufatura é apresentado na seção

4.12, para ilustrar as técnicas de modelagem no contexto de um conjunto de regras derivadas a partir de um problema real.

## 4.2 O Modelo de Redes de Petri Nebulosas de Alto Nível

As Redes de Petri Nebulosas de Alto Nível (RPNAN) [57], [62] derivam das redes Predicado/Transição [23] e das redes de Petri Coloridas [28]. Na RPNAN, os valores carregados pelas senhas são objetos estruturados representando subconjuntos nebulosos e os disparos de transições calculam novos conjuntos que são os valores das senhas que saem dessas transições.

**Definição 4.2.1** *Formalmente, uma RPNAN consiste de:*

1. *Um grafo dirigido bipartido, definido por uma tripla  $(P, T, F)$  tal que  $P$  e  $T$  são dois conjuntos disjuntos de vértices ( $P \cap T = \emptyset$ ) chamados lugares e transições respectivamente, e  $F$  é um conjunto de arcos dirigidos, cada um conectando um lugar  $p \in P$  a uma transição  $t \in T$  ou vice-versa ( $F \subset P \times T \cup T \times P$ ). Lugares correspondem a variáveis e transições representam cálculos de novos subconjuntos nebulosos.*
2. *Uma rotulação dos arcos com  $t$ -uplas de variáveis; o comprimento de cada  $t$ -upla é a aridade do conjunto base da variável conectada ao arco.*
3. *Uma estrutura  $\Sigma$ , definindo uma coleção de objetos tipados juntamente com algumas operações e relações aplicáveis a esses objetos. Fórmulas construídas em  $\Sigma$  podem ser usadas como inscrições de transições.*

Na modelagem de regras de produção nebulosas, transições juntamente com lugares representam regras e lugares representam variáveis e conjuntos base. Assim como no caso de RPN, quando regras mais complexas são modeladas, transições e lugares podem ter significados diferentes. Estes casos serão tratados nas próximas seções. Quando um lugar  $p$  representa uma variável  $V$  sobre o conjunto base  $X$ , diz-se que  $p$  está associado a  $V$  e a  $X$ . Uma transição que representa uma regra  $r$  é dita associada a essa regra.

As senhas representam fatos e são associadas a subconjuntos nebulosos.

**Definição 4.2.2** Uma distribuição de senhas sobre os lugares de uma RPNAN é chamada uma marcação nebulosa  $M$  da rede.

**Definição 4.2.3** Uma RPNAN com alguns lugares contendo senhas é chamada uma Rede de Petri Nebulosa de Alto Nível Marcada.

A seguir é dada a representação do padrão básico de inferência de raciocínio aproximado. Considere-se o padrão

$$V \text{ é } A' \quad (4.1)$$

$$\underline{\text{SE } V \text{ é } A \text{ ENTÃO } U \text{ é } B} \quad (4.2)$$

$$U \text{ é } B' \quad (4.3)$$

onde  $A$  e  $A'$  são subconjuntos nebulosos do conjunto base  $X = \{x_1, x_2, \dots, x_n\}$ ,  $B$  é um subconjunto nebuloso do conjunto base  $Y = \{y_1, y_2, \dots, y_m\}$  e a relação nebulosa induzida  $D_{A \rightarrow B}$  é dada por

$$D_{A \rightarrow B}(x_i, y_j) = d_{ij} .$$

O fato e a regra em (4.1) e (4.2) são modelados conforme apresentado na figura 4.1(a). Neste caso, o lugar  $p_1$  está associado com a variável  $V$  e com o conjunto base  $X$ . O lugar  $p_2$  está associado com a variável  $U$  e com o conjunto base  $Y$ . Os elementos  $d_{ij}$  da matriz de relação  $D_{A \rightarrow B}$  aparecem como constantes nas fórmulas inscritas na transição. O cálculo de tais constantes é feito de acordo com um método de inferência definido por  $f_{-}(A, B)$  (veja seção 2.2.5).

O fato

$$V \text{ é } A'$$

tal que  $A(x_i) = a'_i$ ,  $i = 1, \dots, n$ , é representado colocando-se uma senha  $\langle a'_1, a'_2, \dots, a'_n \rangle$  no lugar associado a  $V$ . A conclusão  $U \text{ é } B'$  é atingida disparando-se  $t$ . O disparo de uma transição muda o estado ou marcação de uma RPNAN, derivando uma nova marcação a partir da anterior. Para introduzir as regras de disparo de transições os conceitos de conjuntos de entrada e saída de lugares e de transições, já conhecidos para RPN, são também necessários .

**Definição 4.2.4** Para cada transição  $t \in T$  e cada lugar  $p \in P$ , define-se:

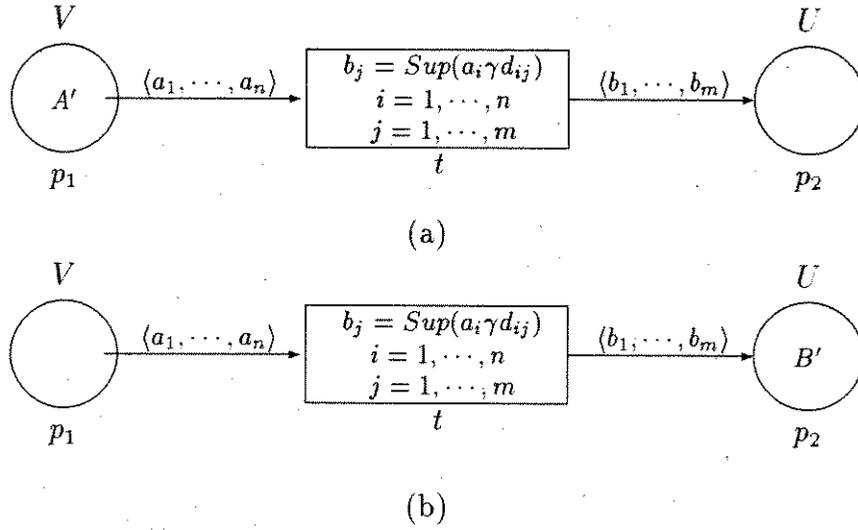


Figura 4.1: RPNAN representando uma regra condicional.(a) Antes do disparo das transiões, onde  $A' = \langle a'_1, a'_2, \dots, a'_n \rangle$ . (b) Depois do disparo das transiões, onde  $B' = \langle b'_1, b'_2, \dots, b'_m \rangle$ .

$$I(t) = \{p \in P \mid (p, t) \in F\}$$

$$I(p) = \{t \in T \mid (t, p) \in F\}$$

$$O(t) = \{p \in P \mid (t, p) \in F\}$$

$$O(p) = \{t \in T \mid (t, p) \in F\}$$

$I(t)$  e  $O(t)$  so os conjuntos de lugares de entrada e saıda da transio  $t$ , respectivamente.  $I(p)$  e  $O(p)$  so os conjuntos de transies de entrada e saıda do lugar  $p$ , respectivamente.

Uma transio  $t \in T$  esto habilitada quando todo lugar  $p \in I(t)$  contem uma senha da forma especificada pelo rtulo em  $\langle p, t \rangle$ . Uma transio habilitada  $t$  pode disparar removendo-se de cada lugar  $p \in I(t)$  uma senha da forma especificada pelo rtulo no arco  $\langle p, t \rangle$  e adicionando-se a cada lugar  $p' \in O(t)$  uma senha da forma especificada pelo rtulo no arco  $\langle t, p' \rangle$ . O valor da nova senha  calculado pelas frmulas inscritas em  $t$ . A ausncia de frmulas inscritas em uma transio significa que o valor da senha em  $p'$  permanece o mesmo que em  $p$ . As frmulas inscritas em  $t$  na figura 4.1 so as expresses usadas para calcular  $B'$  como

$$A' \bullet D_{A \rightarrow B}.$$

Quando é atribuída ao símbolo  $\gamma$  uma T-norma, um método específico de raciocínio nebuloso é definido. A marcação resultante do disparo da transição  $t$  está ilustrada na figura 4.1(b), onde cada  $b'_i$  representa o valor de pertinência de  $y_i$  em  $B'$ .

As formas gerais das fórmulas inscritas nas transições são normalmente de três tipos, dependendo da função exercida pela transição. Na figura 4.1 as fórmulas inscritas correspondem à aplicação da regra de inferência composicional. Transições que fazem a agregação de proposições de antecedentes ou consequentes de regras tem outros tipos de fórmulas inscritas, conforme apresentado na proposição 4.2.1.

**Proposição 4.2.1** *As inscrições dentro de transições são tipicamente de três tipos:*

- *expressões usadas para aplicar modus ponens composicional:*

$$b'_j = \sup_i (a'_i \gamma d_{ij}), i = 1, \dots, n, j = 1, \dots, m$$

onde  $d_{ij}$  são os elementos da matriz de relação  $D_{A \rightarrow B}$ .

- *expressões para realizar conjunção de antecedentes múltiplos:*

$$h_{s1, \dots, sk} = \delta_{i=1, \dots, k} (a_{si}^i), si = 1, \dots, n$$

onde  $k$  é o número de condições no antecedente e  $n$  é o número de elementos em cada conjunto base.

- *expressões para agregar os resultados de regras paralelas ou de fontes parciais de informação:*

$$b'_j = \tau_{i=1, \dots, r} (b_j^i), j = 1, \dots, m$$

onde  $r$  é o número de regras em questão.

As variáveis  $\gamma$  e  $\delta$  são operadores T-norma e  $\tau$  é um operador T-norma ou S-norma. A estrutura  $\Sigma$  (definição 4.2.1) deve incluir a definição desses T-operadores usados nas expressões da proposição 4.2.1 para cada rede em particular.

**Exemplo 4.2.1** *Considere-se a regra e o fato*

V é muito baixo  
SE V é baixo ENTÃO U é alto

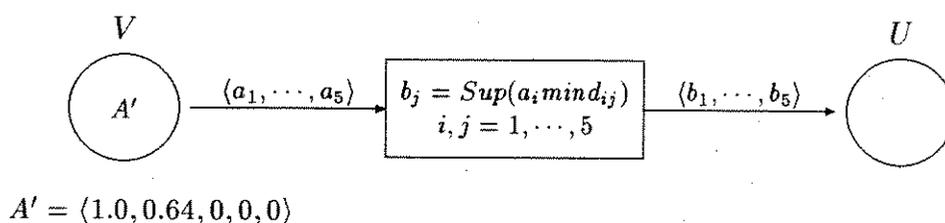


Figura 4.2: Exemplo de uma RPNAN.

onde  $V$  e  $U$  são variáveis sobre os conjuntos base  $X$  e  $Y$  respectivamente e  $X = Y = \{1, 2, 3, 4, 5\}$ . Os subconjuntos nebulosos rotulados pelos valores linguísticos baixo, muito\_baixo, médio e alto estão definidos sobre  $X$  conforme apresentado na Tabela 4.1. Suponha-se também que a relação  $D_{\text{baixo} \rightarrow \text{alto}}$  foi calculada pela expressão

$$f_{\rightarrow}(\text{baixo}(x), \text{alto}(y)) = 1 \wedge [1 - \text{baixo}(x) + \text{alto}(y)],$$

resultando em:

$$D = \begin{bmatrix} 0 & 0 & 0.1 & 0.8 & 1 \\ 0.2 & 0.2 & 0.3 & 1 & 1 \\ 0.9 & 0.9 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

A RPNAN marcada representando esse exemplo em particular é apresentada na figura 4.2 onde foi atribuído ao símbolo  $\gamma$  o operador min. Após o disparo da transição, a marcação muda colocando uma senha no lugar  $U$  com rótulo

$$\langle 0.2, 0.2, 0.3, 0.8, 1.0 \rangle$$

o qual pode ser interpretado linguisticamente como mais\_ou\_menos\_alto.

### 4.3 Equivalência entre RPN e RPNAN

Os modelos de RPNAN são generalizações de modelos de RPN. Portanto, são usados com o intuito de se obter representações mais claras das relações de dependência entre elementos da base de regras, quando os componentes numéricos não precisam estar representados individualmente. A cada elemento de uma RPNAN corresponde um elemento de uma RPN. A relação básica entre os dois modelos é que grupos de lugares e de transições na RPN são representados como um único lugar e uma única

Rótulo	Valor de Pertinência				
	1	2	3	4	5
baixo	1.0	0.8	0.1	0.0	0.0
muito baixo	1.0	0.64	0.0	0.0	0.0
médio	0.0	0.5	1.0	0.5	0.0
alto	0.0	0.0	0.1	0.8	1.0

Tabela 4.1: Valores linguísticos definidos como subconjuntos nebulosos.

transição na RPNAN, respectivamente. Informalmente pode-se descrever a relação entre RPN e RPNAN pelos seguintes pontos:

- Lugares e transições em uma RPNAN representam grupos de lugares e grupos de transições respectivamente na RPN correspondente.
- As transições de uma RPN podem ser de quatro tipos, dependendo da forma como o novo valor da senha é calculado com o disparo da transição. Transições em uma RPNAN não são classificadas em tipos. O valor da senha que sai da transição é calculado de acordo com as fórmulas inscritas nessa transição.
- Os rótulos dos arcos de uma RPN são os valores da matriz de relação induzida pela regra representada. Na RPNAN, os rótulos dos arcos são n-uplas de variáveis e os elementos da matriz de relação aparecem como constantes das fórmulas inscritas nas transições.
- O valor associado à senha em uma RPN é um único número entre zero e um, representando o valor de pertinência do subconjunto nebuloso. Na RPNAN, o valor associado a senha é uma n-upla de valores, representado a função de pertinência desse subconjunto.

## 4.4 Encadeamento de Regras

Duas regras encadeadas como

$$\begin{aligned}
 P_1 &: \text{SE } V \text{ é } A \text{ ENTÃO } U \text{ é } B \text{ e} \\
 P_2 &: \text{SE } U \text{ é } B \text{ ENTÃO } W \text{ é } C
 \end{aligned}$$

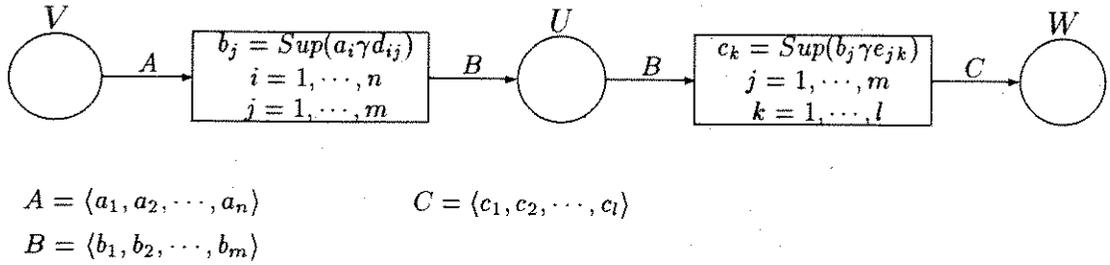


Figura 4.3: RPNAN representando duas regras encadeadas.

podem ser representadas em uma rede como a da figura 4.3, assumindo-se que  $A$  e  $B$  são subconjuntos nebulosos de  $X$  e  $Y$  respectivamente, a relação nebulosa  $D_{A \rightarrow B}$  é dada por  $D_{A \rightarrow B}(x, y) = d_{ij}$  e que  $Z = \{z_1, z_2, \dots, z_n\}$  é um conjunto base,  $W$  é uma variável sobre  $Z$ ,  $C$  é um subconjunto nebuloso de  $Z$  e a relação nebulosa entre  $B$  e  $C$  é dada por  $D_{B \rightarrow C}(y, z) = e_{ij}$ . O disparo da primeira transição deposita uma senha no lugar  $U$  e habilita o disparo da segunda transição.

A questão da condição de consistência para regras encadeadas discutidas no contexto de RPN (seção 3.4) é também pertinente ao contexto de RPNAN.

## 4.5 Regras com Múltiplos Antecedentes

A representação de regras com antecedentes múltiplos na RPNAN é obtida de forma semelhante à representação em RPN. Acrescenta-se um lugar e uma transição após os lugares associados às variáveis do antecedente da regra. As fórmulas inscritas nessa transição são da forma apropriada para agregar as proposições do antecedente (veja proposição 4.2.1).

Como no caso simples, representa-se os fatos

$$V_1 \text{ é } A'_1, V_2 \text{ é } A'_2, \dots, V_k \text{ é } A'_k$$

colocando-se senhas com rótulos  $\langle a_1^i, a_2^i, \dots, a_n^i \rangle$  nos lugares  $V_i$ . A figura 4.4 ilustra a representação do fato e da regra

$$\begin{aligned} &V_1 \text{ é } A'_1, V_2 \text{ é } A'_2, \dots, V_k \text{ é } A'_k \\ &\text{SE } V_1 \text{ é } A_1 \text{ e } V_2 \text{ é } A_2 \text{ e } \dots \text{ e } V_k \text{ é } A_k \text{ ENTÃO } U \text{ é } B \end{aligned}$$

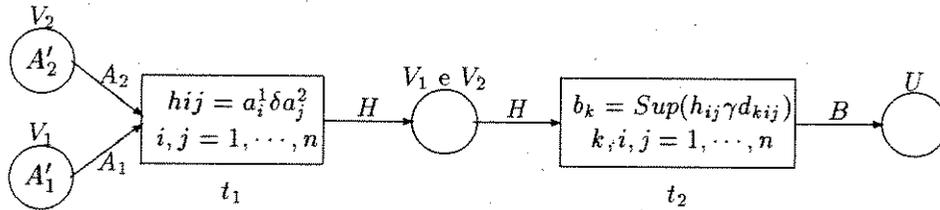
quando  $k = 2$ , assumindo-se que as variáveis  $V_1$ ,  $V_2$  e  $U$  são definidas sobre o mesmo conjunto base  $X = \{x_1, \dots, x_n\}$ . Por essas premissas infere-se  $U$  é  $B'$  tal que

$$B' = H' \bullet F$$

onde  $H'$  e  $F$  estão definidos pelas expressões em (2.8) e (2.7). O símbolo  $\delta$  na fórmula

$$h_{s1,s2,\dots,sk} = \delta(a_{si}^i)$$

inscrita na transição  $t_1$  representa o T-operador usado no cálculo de  $H'$ . Após o disparo da transição  $t_1$ , o lugar  $(V_1, V_2, \dots, V_n)$  fica com uma senha com valor  $H'$ . Essa nova marcação habilita a transição  $t_2$ , que corresponde à aplicação da inferência composicional.



$$\begin{aligned} A_1 &= \langle a_1^1, a_2^1, \dots, a_n^1 \rangle & H &= \langle h_{11}, h_{12}, \dots, h_{nn} \rangle & A_1' &= \langle a_1^{1'}, a_2^{1'}, \dots, a_n^{1'} \rangle \\ A_2 &= \langle a_1^2, a_2^2, \dots, a_n^2 \rangle & B &= \langle b_1, b_2, \dots, b_n \rangle & A_2' &= \langle a_1^{2'}, a_2^{2'}, \dots, a_n^{2'} \rangle \\ & & & & & d_{kij} \text{ é um elemento da relação nebulosa } F \end{aligned}$$

Figura 4.4: RPNAN representando regras com antecedentes múltiplos.

**Exemplo 4.5.1** Considere-se os subconjuntos nebulosos definidos na Tabela 4.1. A regra

SE  $V_1$  é baixo e  $V_2$  é baixo ENTÃO  $U$  é alto

induz uma relação nebulosa  $F$  sobre  $X^1 \times X^2 \times Y$ , onde  $X^1 = X^2 = Y = \{1, 2, 3, 4, 5\}$ . Assumindo-se que a relação nebulosa  $H$  induzida pelo antecedente da regra acima é dado por  $H(x_1, x_2) = \text{baixo}(x_1) \wedge \text{baixo}(x_2)$  e que a relação  $F$  definida em (2.7) é calculada por

$$F(y, x_1, x_2) = 1 \wedge [1 - H(x_1, x_2) + \text{alto}(y)],$$

obtem-se a representação de  $F$  na forma matricial conforme apresentado na Tabela 4.2. Cada quadro da tabela indica os valores da relação  $F$  para um particular valor de

$y = 1$
$\begin{bmatrix} 0 & 0.2 & 0.9 & 1 & 1 \\ 0.2 & 0.2 & 0.9 & 1 & 1 \\ 0.9 & 0.9 & 0.9 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$

$y = 2$
$\begin{bmatrix} 0 & 0.2 & 0.9 & 1 & 1 \\ 0.2 & 0.2 & 0.9 & 1 & 1 \\ 0.9 & 0.9 & 0.9 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$

$y = 3$
$\begin{bmatrix} 0.1 & 0.3 & 1 & 1 & 1 \\ 0.3 & 0.3 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$

$y = 4$
$\begin{bmatrix} 0.8 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$

$y = 5$
$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$

Tabela 4.2: Representação matricial da relação induzida por uma regra com antecedentes múltiplos. Em cada quadro, as linhas da matriz  $5 \times 5$  representam elementos do conjunto base  $X^1$  e as colunas, elementos do conjunto base  $X^2$ .

*y.* Cada um desses quadros apresenta uma matriz  $5 \times 5$  tal que cada linha representa um elemento do conjunto  $X^1$  e cada coluna representa um elemento do conjunto  $X^2$ .

Dados os fatos

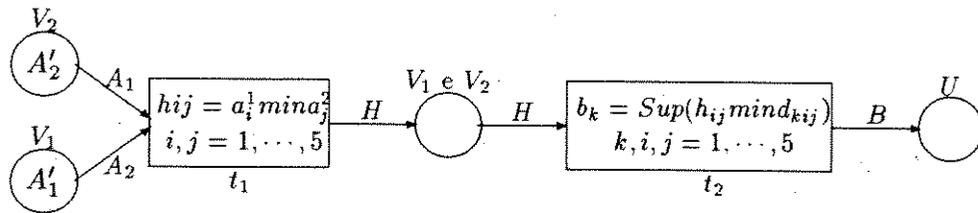
$V_1$  é baixo e  
 $V_2$  é mais\_ou\_menos\_baixo,

a relação nebulosa induzida  $H'$ , calculada pela equação

$$H'(x_1, x_2) = \min(\text{baixo}(x_1), \text{mais_ou_menos_baixo}(x_2))$$

é dada pela matriz

$$H' = \begin{bmatrix} 1 & 1 & 0.2 & 0 & 0 \\ 0.8 & 0.8 & 0.2 & 0 & 0 \\ 0.1 & 0.1 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



$$\begin{aligned}
 A_1 &= \langle a_1^1, a_2^1, \dots, a_5^1 \rangle & H &= \langle h_{11}, h_{12}, \dots, h_{55} \rangle & A'_1 &= \langle 1.0, 0.8, 0.1, 0, 0 \rangle \\
 A_2 &= \langle a_1^2, a_2^2, \dots, a_5^2 \rangle & B &= \langle b_1, b_2, \dots, b_5 \rangle & A'_2 &= \langle 1.0, 1.0, 0.2, 0, 0 \rangle \\
 d_{kij} & \text{ é um elemento da relação nebulosa } F
 \end{aligned}$$

Figura 4.5: RPNAN representando uma regra com duas condições no antecedente.

Este exemplo está ilustrado na figura 4.5. Após o disparo da transição  $t_1$ , uma nova marcação é obtida, com uma senha no lugar  $(V_1, V_2)$  cujo valor é  $\langle 1, 1, 0.2, 0, 0, 0.8, 0.8, 0.2, 0, 0, 0.1, 0.1, 0.1, 0, 0, 0, 0, 0, 0, 0, 0 \rangle$ . Tal marcação habilita a transição  $t_2$ , cujo disparo produz uma outra marcação colocando uma senha no lugar  $U$ , com valor  $\langle 0.2, 0.2, 0.3, 1, 1 \rangle$ , significando mais\_ou\_menos\_alto.

## 4.6 Regras Quantificadas e Qualificadas

A representação de regras com quantificadores e com qualificação de certeza, segue a metodologia fornecida por Yager [71], descrita na seção 3.6. A modelagem dessas regras na RPNAN dá-se de forma semelhante à modelagem das mesmas na RPN. As estruturas que representam regras com quantificadores no antecedente e regras com fatores de certeza são as mesmas das regras com conjunção no antecedente e regras simples, respectivamente, diferindo apenas na forma em que os rótulos dos arcos são calculados.

## 4.7 Regras Paralelas

Conforme a discussão de regras paralelas no contexto de RPN, existem duas formas alternativas de se representar regras da forma

$$\text{SE } V \text{ é } A_i \text{ ENTÃO } U \text{ é } B_i \text{ para } i = 1, \dots, r$$

na RPNAN, dependendo de como inferência é executada, seja a nível local ou global.

Quando a inferência é realizada globalmente, conforme a expressão

$$B' = A' \bullet \cup_{i=1}^r (A_i \rightarrow B_i),$$

a estrutura de RPNAN para regras paralelas é a mesma da regra simples. A única diferença está na forma de calcular os valores da matriz de relação global

$$D(x, y) = \mathbf{s}_{i=1}^r (D_{A_i \rightarrow B_i}(x, y)),$$

que são as constantes inscritas na transição.

Quando a inferência é realizada localmente conforme a expressão

$$B' = \cup_{i=1}^r A' \bullet (A_i \rightarrow B_i),$$

duas novas transições são incluídas. Uma delas copia o valor da senha para enviá-lo para todos os lugares representando antecedentes das regras paralelas, e é equivalente ao grupo de transições do tipo 3 com a mesma função na RPN. A outra serve para agregar os resultados individuais dos disparos de cada regra e é equivalente ao grupo de transições do tipo 4 na RPN. Como já foi dito anteriormente, as transições da RPNAN não são classificadas em tipos. As diferentes funções de uma transição são determinadas pelas fórmulas inscritas. Assim, a primeira das duas transições acrescentadas na representação de regras paralelas não tem fórmulas inscritas já que seu papel é apenas reproduzir o valor da senha que entrou, e a segunda dessas transições tem as fórmulas inscritas

$$b'_j = \tau_{i=1, \dots, r}(b_j^i), j = 1, \dots, m$$

onde  $r$  é o número de regras paralelas (ver proposição 4.2.1).

As regras da forma

$$\text{SE } V \text{ é } A_i \text{ ENTÃO } U \text{ é } B_i, \text{ para } i = 1, \dots, r,$$

podem ser representadas conforme a figura 4.6. A transição  $t_1$  não possui fórmulas inscritas. Assim que a primeira transição dispara, todas as transições representando regras do conjunto de regras paralelas estão habilitadas e podem disparar em paralelo. Os resultados obtidos individualmente são então agregados pelo operador T-norma ou S-norma que estiver substituindo o símbolo  $\tau$  na transição  $t_2$  em cada caso específico.

No contexto de RPNAN também será adotada uma representação hierárquica para regras paralelas, onde a sub-rede interna é substituída por uma única transição,

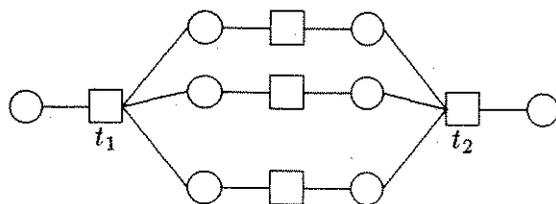


Figura 4.6: Estrutura básica de RPNAN representando regras paralelas quando o princípio de combinação/projeção é realizado localmente.

com o fim de simplificar a representação gráfica da rede. Essa transição será chamada *bloco de regras paralelas* e denotada por um duplo retângulo, como pode ser visto na figura 4.7(b) onde a sub-rede interna da figura 4.7(a) foi substituída.

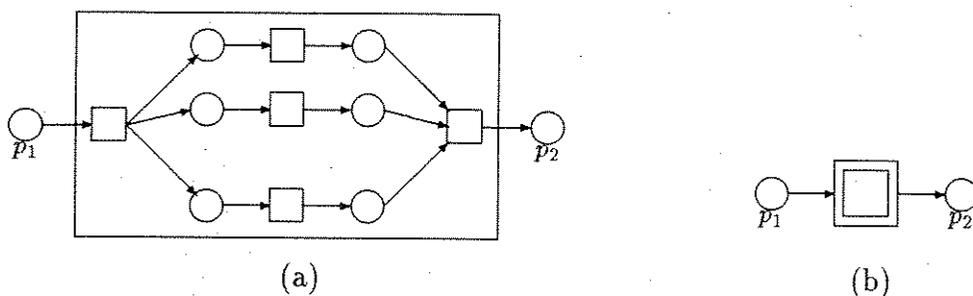


Figura 4.7: RPNAN com representação hierárquica para regras paralelas. (a) Com subrede interna. (b) Com bloco de regras paralelas.

## 4.8 Regras Conflitantes

Conforme definido da seção 3.8, regras conflitantes são aquelas que possuem a mesma variável no antecedente e variáveis diferentes no conseqüente. Para introduzir o processo de modelagem de regras conflitantes, considere-se inicialmente o caso de regras com um único antecedente:

$$\text{SE } V \text{ é } A \text{ ENTÃO } U_1 \text{ é } B_1$$

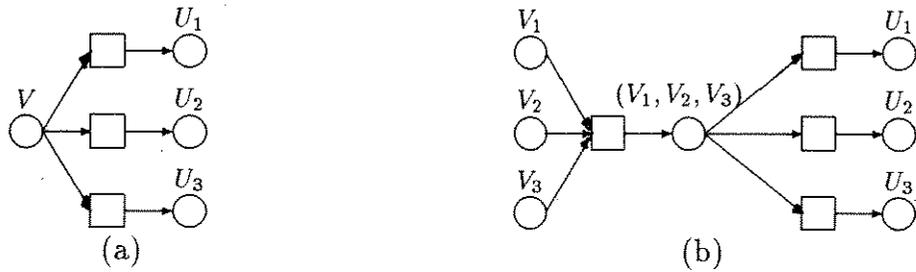


Figura 4.8: Estrutura básica de RPNAN representando regras conflitantes. (a) Regras com um único antecedente. (b) Regras com antecedentes múltiplos.

$$\begin{aligned} \text{SE } V \text{ é } C \text{ ENTÃO } U_2 \text{ é } B_2 \\ \text{SE } V \text{ é } D \text{ ENTÃO } U_3 \text{ é } B_3 \end{aligned}$$

Para representar o conjunto de regras acima na RPNAN, define-se um único lugar para os antecedentes de todas as regras, um lugar para cada um dos consequentes e uma transição para cada uma das regras. O lugar associado aos antecedentes é conectado a todas as transições. Portanto, assim como no caso de Redes de Petri convencionais e de RPN, mais de um arco saindo de um mesmo lugar representa uma situação de conflito do ponto de vista de execução da rede. O algoritmo de raciocínio para frente gera automaticamente todos os caminhos de raciocínio a partir de uma situação inicial.

O conjunto de três regras conflitantes acima é representado conforme a figura 4.8(a). Para um conjunto de regras conflitantes com antecedentes múltiplos  $V_1$ ,  $V_2$  e  $V_3$ , a RPNAN teria a forma apresentada na figura 4.8(b).

## 4.9 Fontes Parciais de Informação

Nesta seção será discutida a representação de fontes parciais de informação na RPNAN. Considera-se fontes parciais de informação proposições que fornecem informação sobre uma mesma variável. Estas podem ser de dois tipos básicos. O primeiro deles acontece quando se tem vários fatos sobre a mesma variável. O segundo se caracteriza pela existência de regras com a mesma variável no consequente e variáveis diferentes no antecedente.

### 4.9.1 Representado Vários Fatos Sobre a Mesma Variável

Em uma RPNAN, um fato  $V$  é  $A'$  é representado colocando-se uma senha  $\langle a'_1, \dots, a'_n \rangle$  no lugar  $V$ . Quando são fornecidos mais de um fato relacionados com a variável  $V$ , estes são representados na RPNAN colocando-se uma senha para cada um no mesmo lugar  $V$ . A semântica do disparo de transições para RPNAN deve ser estendida para ser aplicada a essa nova situação. Agora, uma transição  $t \in T$  está habilitada sempre que todo lugar  $p \in I(t)$  tiver pelo menos uma senha. Como é possível haver mais de uma senha em um lugar ao mesmo tempo, uma transição extra é incluída na representação para realizar a agregação dos valores das senhas. Além disso é necessário permitir que os rótulos dos arcos  $\langle p, t \rangle$  de lugares a transições sejam somas formais de n-uplas em vez de n-uplas apenas. Uma transição habilitada  $t$  pode disparar removendo de cada lugar  $p \in I(t)$  um conjunto de senhas conforme especificado pelo rótulo do arco  $\langle p, t \rangle$  e colocando uma senha em cada lugar  $p' \in O(t)$ .

**Exemplo 4.9.1** Considere a base de conhecimento  $K = \{d_1, d_2, I_1\}$  tal que

$$\begin{aligned}d_1 &: V \text{ é } A' \\d_2 &: V \text{ é } A'' \\I_1 &: \text{SE } V \text{ é } A \text{ ENTÃO } U \text{ é } B\end{aligned}$$

onde  $A$ ,  $A'$  e  $A''$  são subconjuntos nebulosos sobre o conjunto base  $X$ ,  $V$  é uma variável sobre  $X$ ,  $B$  é um subconjunto nebuloso sobre o conjunto base  $Y$  e  $U$  é uma variável sobre  $Y$ .

De acordo com as técnicas de modelagem descritas, os dados  $d_1$  e  $d_2$  do exemplo 4.9.1 são representados por senhas com valores  $\langle a'_1, a'_2, \dots, a'_n \rangle$  e  $\langle a''_1, a''_2, \dots, a''_n \rangle$ , respectivamente, colocadas inicialmente no lugar  $V$ . Suponha que a agregação dos valores  $A'$  e  $A''$  é interpretada como T-norma. A RPNAN obtida nesse caso é mostrada na figura 4.9. Após o disparo da transição  $t_1$ , o estado da rede muda e uma senha é colocada no lugar  $p_2$  com valor  $A_C = A' \delta A''$ . Essa nova marcação habilita a transição  $t_2$ , que pode disparar resultando numa marcação com uma senha no lugar  $U$ .

Em geral, quando se tem  $l$  fatos

$$d_1 : V \text{ é } A^1, d_2 : V \text{ é } A^2, \dots, d_l : V \text{ é } A^l,$$

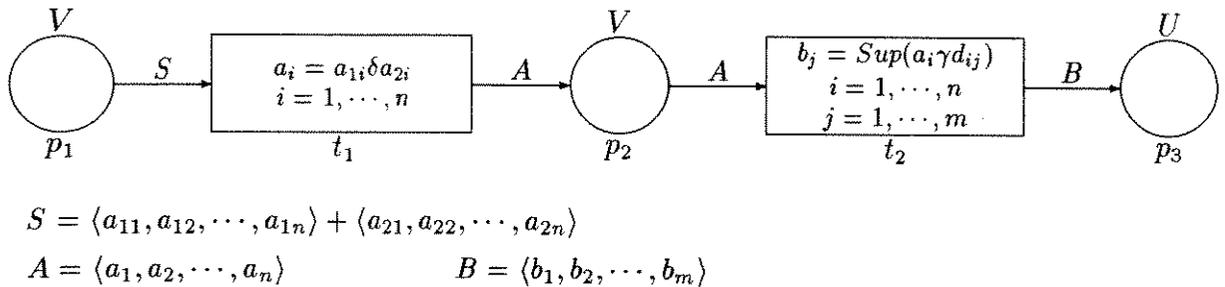


Figura 4.9: RPNaN representando uma regra e dois fatos referentes à mesma variável.

insere-se  $l$  senhas  $A^1, A^2, \dots, A^l$  no lugar  $p_1$ . A forma geral da fórmula inscrita na transição  $t_1$  é

$$a_i = \delta_{k=1, \dots, l} a_{ki}, i = 1, \dots, n.$$

Nesta representação pode-se tirar proveito da representação hierárquica das RPNaN, como foi feito com as regras paralelas. A sub-rede que tem a função de aplicar agregação de valores de senhas pode ser substituída por um único lugar, em um nível de abstração mais alto. Com isso obtém-se um modelo mais compacto sem se alterar as condições de estrutura e comportamento da rede determinadas pela definição formal. É importante observar que também é possível que a base de conhecimento contenha fatos referentes a variáveis dos consequentes das regras. O tratamento de casos como esse é semelhante ao que acabou de ser descrito aqui e será ilustrado na seção 5.6 com a aplicação do algoritmo de raciocínio para trás.

#### 4.9.2 Representando Regras com a Mesma Variável no Consequente e Variáveis Diferentes no Antecedente.

Um tipo diferente de fontes parciais de informação ocorre quando a base de conhecimento inclui mais de uma regra com a mesma variável no consequente e variáveis diferentes no antecedente. Quando o antecedente tem mais de uma variável, o conjunto de variáveis no antecedente de cada regra deve ser diferente, embora um subconjunto delas pode ser igual. Para representar conjuntos de regras como esse na RPNaN, acrescenta-se uma transição e um lugar para aplicar alguma operação de agregação entre os resultados individuais obtidos com cada uma das regras, para produzir um único resultado para a variável desejada.

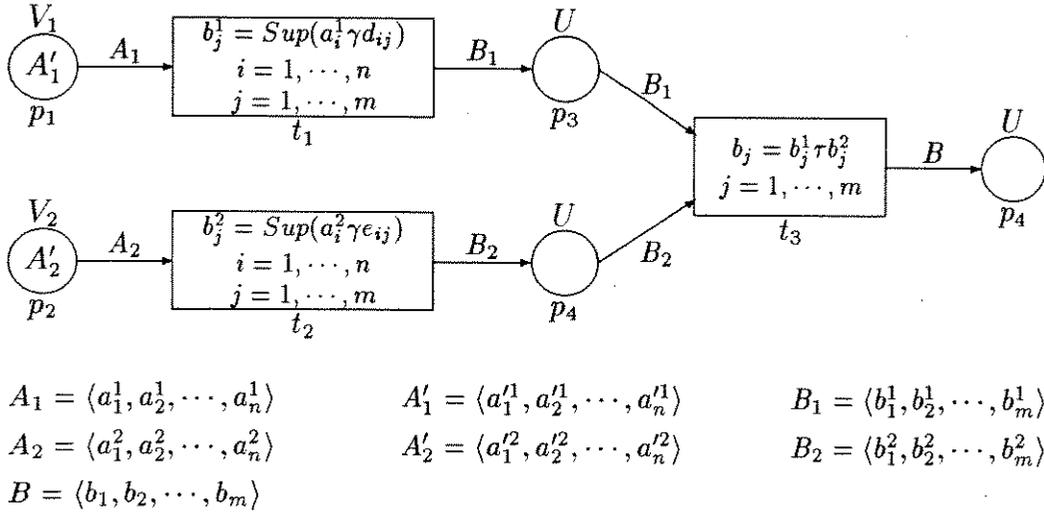


Figura 4.10: RPNAN representado regras com a mesma variável no conseqüente.

#### Exemplo 4.9.2 As regras e os fatos

$$\begin{aligned}
 d_1 &: V_1 \text{ é } A'_1 \\
 d_2 &: V_2 \text{ é } A'_2 \\
 I_1 &: \text{SE } V_1 \text{ é } A_1 \text{ ENTÃO } U \text{ é } B \\
 I_2 &: \text{SE } V_2 \text{ é } A_2 \text{ ENTÃO } U \text{ é } B
 \end{aligned}$$

onde os elementos da matriz de relação induzida são denotados por  $D_{A_1 \rightarrow B}(x_i, y_j) = d_{ij}$  e  $D_{A_2 \rightarrow B}(x_i, y_j) = e_{ij}$  podem ser representados pela rede apresentada na figura 4.10.

## 4.10 Regras com Conjunção no Conseqüente

Se o lado direito de uma regra de produção nebulosa contém apenas uma variável, então a aplicação da regra de inferência composicional nos dá o subconjunto nebuloso que é o valor dessa variável. Entretanto, se o lado direito da regra contém mais de uma variável, a aplicação da regra de inferência composicional nos dá uma relação nebulosa envolvendo todas as variáveis que aparecem no conseqüente da regra. Para obter o valor de uma única variável é então necessário aplicar o princípio de projeção.

**Exemplo 4.10.1** *Assuma que é dada a regra*

$$I : \text{SE } V_1 \text{ é } A \text{ ENTÃO } U_1 \text{ é } B \text{ e } U_2 \text{ é } C$$

e o fato

$$V_1 \text{ é } A'$$

*Sejam  $X^1$ ,  $X^2$  e  $X^3$  os conjuntos base de  $A$ ,  $B$  e  $C$  respectivamente.*

A conjunção

$$U_1 \text{ é } B \text{ e } U_2 \text{ é } C$$

do exemplo 4.10.1 induz uma relação nebulosa  $H(x_2, x_3)$  sobre  $X^2 \times X^3$ . A regra  $I$  do mesmo exemplo induz uma relação  $F$  sobre  $X^1 \times X^2 \times X^3$ , tal que

$$F(x_1, x_2, x_3) = f_{\rightarrow}(A(x_1), H(x_2, x_3)).$$

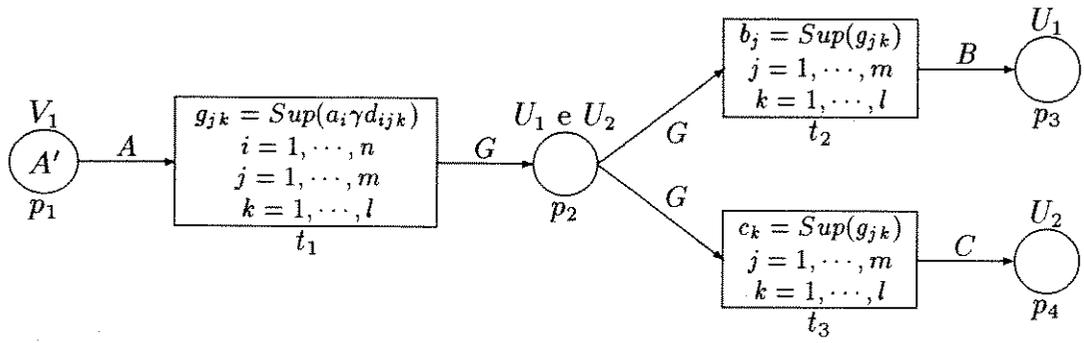
A regra de inferência composicional  $A' \circ F$  gera uma relação  $G$  sobre  $X^2 \times X^3$ . Aplica-se então o princípio de projeção para obter o subconjunto nebuloso que dá o valor de uma das duas variáveis do conseqüente da regra. No exemplo 4.10.1 obtém-se os valores das variáveis  $U_1$  e  $U_2$  por:

$$B(x_2) = \max_{x_3}[G(x_2, x_3)]$$

e

$$C(x_3) = \max_{x_2}[G(x_2, x_3)]$$

Para modelar regras com conjunção no conseqüente na RPNAN deve-se acrescentar tantas transições quantas forem as variáveis no conseqüente, para aplicar a respectiva operação de projeção. A RPNAN representando o exemplo 4.10.1 é mostrada na figura 4.11.



$$A = \langle a_1, a_2, \dots, a_n \rangle$$

$$A' = \langle a'_1, a'_2, \dots, a'_n \rangle$$

$$B = \langle b_1, b_2, \dots, b_m \rangle$$

$$C = \langle c_1, c_2, \dots, c_l \rangle$$

$$G = \langle g_{11}, g_{12}, \dots, g_{ml} \rangle$$

Figura 4.11: RPNAN representando uma regra com conjunção no conseqüente.

## 4.11 Alcunçabilidade

As Redes de Petri ordinárias e as Redes de Petri de alto nível são ferramentas de modelagem que apresentam a peculiaridade de permitir que propriedades algébricas da rede sejam usadas para analisar o sistema sendo modelado e eventualmente descobrir propriedades importantes desse sistema. Um método importante disponível para se investigar as propriedades de uma Rede de Petri é o método de *S-invariantes* [39] [49] [22] [26] [27].

Nas RPNAN não existe a circularidade e os valores das senhas são diferentes a cada disparo de transição. Em função disso, o método de *S-invariantes* não pode ser totalmente aproveitado para análises por não ser possível captar as mudanças ocorridas nos valores das senhas. Entretanto, usando-se as noções convencionais de alcançabilidade e *S-invariantes* aplicadas a rede de Petri associada [24] a uma RPNAN, é possível detectar aspectos relevantes quanto ao projeto da RPNAN, relativos à posição das senhas.

**Definição 4.11.1** Dada um RPNAN  $N$ , seja  $N^*$  a rede de Petri ordinária obtida de  $N$  por meio das seguintes substituições:

1. Os rótulos dos arcos em  $N^*$  são iguais a 1.
2. As senhas perdem a sua identidade e se tornam senhas anônimas comuns.

3. As fórmulas incritas são ignoradas.

$N^*$  é chamada a Rede de Petri Associada (RPA) de  $N$ .

**Definição 4.11.2** Dada uma RPA  $N^*$  associada à RPNAN  $N$ , uma marcação  $M$  de  $N^*$  e uma transição  $t$  habilitada em  $M$ , a marcação  $M_1 = r(M, t)$  obtida pelo disparo de  $t$  é dita imediatamente alcançável por  $M$ .

Se a marcação  $M_1$  é imediatamente alcançável de  $M_0$  e  $M_2$  é imediatamente alcançável de  $M_1$ , então  $M_2$  é alcançável de  $M_0$ . A relação de alcançabilidade é o fechamento transitivo da relação de alcançabilidade imediata.

**Definição 4.11.3** Dada uma RPA  $N^*$  e uma marcação  $M_0$  de  $N^*$ , o seguinte conjunto de alcançabilidade é definido:

$$R(M_0) = \{M \mid M \text{ é alcançável por } M_0\} .$$

**Definição 4.11.4** Dada a matriz de incidência  $C^*$  correspondente a uma RPA  $N^*$ , um vetor  $M$ -dimensional  $p$  será chamado um  $S$ -invariante de  $C^*$  se e somente se

$$p.C^* = 0 .$$

A existência do vetor  $p$  implica em uma lei de conservação entre as senhas da rede. De fato, dada uma marcação inicial  $M_0$  e outra marcação  $M \in R(M_0)$  prova-se facilmente [49] que existe um vetor  $N$ -dimensional  $v$  tal que

$$M - M_0 = C^*.v . \tag{4.4}$$

O método de  $S$ -invariante é muito útil na verificação da propriedade de alcançabilidade. Como as questões de inferência em uma base de regras se reduzem a questões de alcançabilidade quando modeladas na RPNAN, equações como a de (4.4), quando aplicadas a Redes de Petri Associadas, podem ser usadas para garantir que a RPNAN seja projetada corretamente.

## 4.12 Exemplo de aplicação

A seguir é apresentada uma RPNAN representando uma base de regras nebulosas real relacionada à modelagem de sistemas de manufatura. O sistema de manufatura

a ser estudado pertence a uma indústria que utiliza a filosofia de tecnologia de grupo para produção. Enfoca-se aqui uma célula de manufatura específica para a fabricação de componentes para motores diesel, a saber: pistão, camisa e cabeçote. A célula de manufatura consiste de

- 2 centros de usinagem cnc (CN\_14,CN\_24),
- um torno cnc (CN\_72), quatro furadeiras (B),
- três tornos revólver (TR),
- dois tornos verticais (TV),
- dois tornos paralelos (TP),
- uma plaina vertical (P),
- um forno para alívio de tensões (X\_58),
- uma bancada para limpeza e montagem (MDA),
- uma bancada para teste hidro-pneumático (MDE) e
- uma bancada para inspeção (MDI).

A figura 4.12 mostra o fluxo de produção de cada componente.

Um estudo de modelagem e simulação foi desenvolvido em [15], considerando-se o fluxo de produção do pistão, com o objetivo de se desenvolver estratégias para o gerenciamento da célula. No processo de manufatura do pistão, o torno cnc (CN\_72) realiza dois tipos de operações: aparar diâmetro interno (operação 7) e tornear diâmetro externo acabado (operação 9). Cada uma dessas operações exige uma operação de *set-up* (preparação de fixadores e ferramental). Um conjunto de regras de produção nebulosas foram derivadas em [15] para se construir um sistema baseado em conhecimento (SBC) para análise do comportamento da célula de manufatura. Este conjunto de regras nebulosas será representado aqui no modelo de RPNAN como ilustração das técnicas de modelagem apresentadas até agora [60] [64].

Vários aspectos do comportamento da célula de manufatura podem se analisados pelo SBC gerado. Por exemplo, quando as informações sobre tamanho do lote, tamanho do lote de transferência e tempo de processamento estão disponíveis, pode-se inferir dados a respeito de

- tempo final;

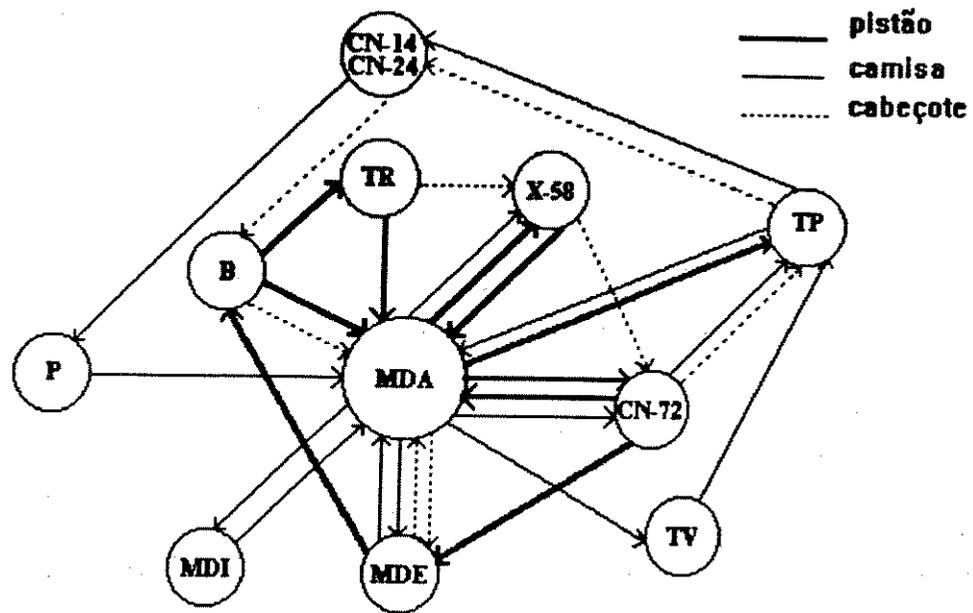


Figura 4.12: Fluxo de produção de pistão, camisa e cabeçote

- utilização da CN\_72 para operações de *set-up* 7 e 9 individualmente
- utilização da CN\_72 para operações de *set-up* em geral,
- a utilização da CN\_72 para operações de processamento 7 e 9 individualmente,
- a utilização da CN\_72 para operações de processamento em geral,
- utilização global da CN\_72,
- tamanho médio da fila para as operações 7 e 9,
- tamanho médio da fila total, etc.

As regras de produção nebulosas definem relações entre as variáveis envolvidas no problema em termos de valores linguísticos específicos como *pequeno*, *médio* e *grande*, os quais são interpretados como subconjuntos nebulosos sobre cada um dos conjuntos base. Denota-se por  $S_i$ ,  $M_i$  e  $L_i$  os subconjuntos com valores linguísticos *pequeno*, *médio* e *grande* sobre o conjunto base  $X^i$ , respectivamente. A Tabela 4.3 apresenta os subconjuntos nebulosos usados para se definir os valores linguísticos para cada um dos conjuntos base, que são conjuntos discretos e finitos. A Tabela 4.4 mostra as variáveis usadas pelas regras do exemplo e seus significados correspondentes. A

Tabela 4.5 apresenta os conjuntos base de cada uma das variáveis, enumerando os elementos. A base de regras completa é mostrada nas Tabelas 4.6 e 4.7, onde cada quadro representa um conjunto de regras paralelas. O significado de tais quadros pode ser compreendido através do seguinte exemplo: o elemento  $S_6$  na segunda linha e primeira coluna da Tabela 4.6(a) representa a regra

$$\text{SE } V_1 \text{ é } S_1 \text{ e } V_2 \text{ é } M_2 \text{ ENTÃO } V_6 \text{ é } S_6$$

que é uma interpretação formal da regra

*SE tamanho do lote de fabricação é pequeno e  
tamanho do lote de transferência é médio  
ENTÃO tempo final é pequeno.*

$X^1$	20	30	40	50	60	70	80	90	100	110	120	130
$S_1$	1	1	1	1	0.3	0.1	0	0	0	0	0	0
$M_1$	0	0	0	0.1	0.3	1	1	1	1	0.3	0.1	0
$L_1$	0	0	0	0	0	0	0	0	0.1	0.3	1	1

$X^2$	5	10	15	20	25	30	35	40
$S_2$	1	0	0	0	0	0	0	0
$M_2$	0	0.5	1	1	1	0.5	0	0
$L_2$	0	0	0	0	0	0	1	1

$X^3$	0.05	0.075	0.10	0.125	0.15	0.175	0.20	0.225	0.25
$S_3$	1	0.5	0	0	0	0	0	0	0
$M_3$	0	0.5	1	1	1	0.5	0	0	0
$L_3$	0	0	0	0	0	0.5	1	1	1

$X^6$	1	1.5	2	2.5	3	3.5	4	4.5
$S_6$	1	0.5	0	0	0	0	0	0
$M_6$	0	0.5	1	1	1	0.5	0	0
$L_6$	0	0	0	0	0	0.5	1	1

$X^7$	10	15	20	25	30	35	40	45	50	55
$S_7$	1	0.5	0	0	0	0	0	0	0	0
$M_7$	0	0.5	1	1	1	1	1	0.5	0	0
$L_7$	0	0	0	0	0	0	0	0.5	1	1

$X^9$	42	49	56	63	70	77	84	91	98	105
$S_9$	1	0.5	0	0	0	0	0	0	0	0
$M_9$	0	0.5	1	1	1	1	1	0.5	0	0
$L_9$	0	0	0	0	0	0	0	0.5	1	1

$X^{10}$	90	105	120	135	150	165	180	195	210
$S_{10}$	1	0.5	0	0	0	0	0	0	0
$M_{10}$	0	0.5	1	1	1	1	1	0.5	0
$L_{10}$	0	0	0	0	0	0	0	0.5	1

$X^{11}$	0.15	0.22	0.30	0.37	0.45	0.52	0.60	0.67	0.75
$S_{11}$	1	0.5	0	0	0	0	0	0	0
$M_{11}$	0	0.5	1	1	1	0.5	0	0	0
$L_{11}$	0	0	0	0	0	0.5	1	1	1

$X^{15}$	20	30	40	50	60	70	80	90	100
$S_{15}$	1	0.5	0	0	0	0	0	0	0
$M_{15}$	0	0.5	1	1	1	0.5	0	0	0
$L_{15}$	0	0	0	0	0	0.5	1	1	1

Tabela 4.3: Subconjuntos Nebulosos representando valores linguísticos.

$V_1$	tamanho do lote de fabricação
$V_2$	tamanho do lote de transferência
$V_3$	utilização da CN.72 na operação de <i>set-up</i> 7
$V_4$	utilização da CN.72 na operação de <i>set-up</i> 9
$V_5$	utilização da CN.72 em operações de <i>set-up</i>
$V_6$	tempo final
$V_7$	tamanho médio da fila na CN.72 para operação 7
$V_8$	tamanho médio da fila na CN.72 para operação 9
$V_9$	tempo de processamento para operação 7
$V_{10}$	tempo de processamento para operação 9
$V_{11}$	utilização da CN.72 na operação de processamento 7
$V_{12}$	utilização da CN.72 na operação de processamento 9
$V_{13}$	utilização da CN.72 em operações de processamento
$V_{14}$	tamanho médio da fila na CN.72
$V_{15}$	utilização total da CN.72

Tabela 4.4: Variáveis e seus significados.

variável	conjunto base	elementos	unidades
$V_1$	$X^1$	10,20,30,40,50,.....,140	peças
$V_2$	$X^2$	5,10,15,20,30,.....,40	peças
$V_3, V_4, V_5$	$X^3$	0.05,0.075,0.10,....,0.25	tempo
$V_6$	$X^6$	1,1.5,2,2.5,....,4.5	tempo $\times 10^4$
$V_7, V_8, V_{14}$	$X^7$	10,15,20,25,.....,60	peças
$V_9$	$X^9$	42,49,56,.....,105	tempo
$V_{10}$	$X^{10}$	90,105,120,135,.....,210	tempo
$V_{11}, V_{12}, V_{13}$	$X^{11}$	0.15,0.22,0.30,.....,0.75	tempo
$V_{15}$	$X^{15}$	20,30,40,50,.....,100	porcentagem

Tabela 4.5: Conjuntos Base

(a) Consequente: tempo final( $V_6$ ).  
Antecedentes: tamanho do lote( $V_1$ )  
e tamanho do lote de transfer.( $V_2$ ).

		$V_1$		
		$S_1$	$M_1$	$L_1$
$V_2$	$S_2$	$M_6$	$M_6$	$L_6$
	$M_2$	$S_6$	$M_6$	$M_6$
	$L_2$	$S_6$	$S_6$	$M_6$

(b) Consequente: tamanho médio da fila  
na CN.72 para op. 7( $V_7$ ).

Antecedentes: tamanho do lote( $V_1$ )  
e tamanho do lote de transfer.( $V_2$ ).

		$V_1$		
		$S_1$	$M_1$	$L_1$
$V_2$	$S_2$	$M_7$	$M_7$	$L_7$
	$M_2$	$S_7$	$M_7$	$L_7$
	$L_2$	$S_7$	$M_7$	$M_7$

(c) Consequente: util. da CN.72  
em set-up para op. 7( $V_3$ ).

Antecedente: tam. lote transfer.( $V_2$ ).

$V_2$		
$S_2$	$M_2$	$L_2$
$M_3$	$S_3$	$S_3$

(d) Consequente: util. da CN.72  
em set-up para op. 9( $V_4$ ).

Antecedente: tam. lote transf.( $V_2$ ).

$V_2$		
$S_2$	$M_2$	$L_2$
$L_3$	$M_3$	$S_3$

(e) Consequente: util. da CN.72  
em op. de set-up( $V_5$ ).

Antecedentes: util. CN.72 set-up op.7( $V_3$ ).  
e util.da CN.72 em set-up op.9( $V_4$ ).

		$V_4$		
		$S_3$	$M_3$	$L_3$
$V_3$	$S_3$	$S_3$	$M_3$	$L_3$
	$M_3$	$M_3$	$M_3$	$L_3$

(f) Consequente: util. da CN.72  
em processamento para op. 7( $V_{11}$ )

Antecedentes: tam. lote transf.( $V_2$ )  
e tempo de process. para op.7( $V_9$ ).

		$V_2$		
		$S_2$	$M_2$	$L_2$
$V_9$	$S_9$	$S_{11}$	$S_{11}$	$S_{11}$
	$M_9$	$M_{11}$	$M_{11}$	$M_{11}$
	$L_9$	$M_{11}$	$M_{11}$	$M_{11}$

Tabela 4.6: Subconjunto da base de conhecimento do exemplo da manufatura.

(g) Consequente: util. da CN.72 em processamento para op.9( $V_{12}$ )  
 Antecedentes: tam. lote transf.( $V_2$ ) e tempo de process. para op.9( $V_{10}$ ).

		$V_2$		
		$S_2$	$M_2$	$L_2$
$V_{10}$	$S_{10}$	$M_{11}$	$M_{11}$	$L_{11}$
	$M_{10}$	$L_{11}$	$L_{11}$	$L_{11}$
	$L_{10}$	$L_{11}$	$L_{11}$	$L_{11}$

(i) Consequente: tam. médio da fila na CN.72 para op. 9( $V_8$ ).

Antecedente: tam. lote transf.( $V_2$ ).

$V_2$		
$S_2$	$M_2$	$L_2$
$S_7$	$S_7$	$M_7$

(h) Consequente: util. da CN.72 em op. de proc.( $V_{13}$ ).

Antecedentes: util. CN.72 em proc. op.7( $V_{11}$ ) e util. CN.72 proc. op.9( $V_{12}$ ).

		$V_{12}$	
		$M_{11}$	$L_{11}$
$V_{11}$	$S_{11}$	$M_{11}$	$L_{11}$
	$M_{11}$	$L_{11}$	$L_{11}$

(j) Consequente: tam. médio da fila na CN.72( $V_{14}$ ).  
 Antecedentes: tam. med. da fila na CN.72 op.7( $V_7$ ) e tam. médio da fila na CN.72 op.9( $V_8$ ).

		$V_7$		
		$S_7$	$M_7$	$L_7$
$V_8$	$S_7$	$S_7$	$M_7$	$L_7$
	$M_7$	$M_7$	$L_7$	$L_7$

(k) Consequente: util. total da CN.72 da CN.72 ( $V_{15}$ ).

Antecedentes: util. CN.72 em op. de set-up ( $V_5$ ) e util. da CN.72 em op. de proc. ( $V_{13}$ ).

		$V_5$		
		$S_{15}$	$M_{15}$	$L_{15}$
$V_{13}$	$M_{13}$	$S_{15}$	$M_{15}$	$L_{15}$
	$L_{13}$	$M_{15}$	$L_{15}$	$L_{15}$

Tabela 4.7: Subconjunto da base de conhecimento do exemplo da manufatura.

Considere-se as Tabelas 4.6(c), (d) e (e). Os conjuntos de regras das Tabelas 4.6(c) e (d) são conjuntos de regras paralelas com um único antecedente ( $V_2$ ). Os resultados obtidos independentemente por esses dois conjuntos de regras, são os antecedentes das regras da Tabela 4.6(e). Logo, os conjuntos de regras das Tabelas 4.6(c) e (d) não podem ser conflitantes, o que requer a introdução de uma transição sem fórmulas inscritas após o lugar que representa  $V_2$ , para fazer cópias das senhas (transição  $t_2$  da figura 4.13).

Considerando-se as regras das Tabelas 4.6 (a) e (b), nota-se que os valores das variáveis  $V_1$  e  $V_2$  são requeridos para se concluir dados sobre tempo final e tamanho médio da fila, valores que pertencem a caminhos de raciocínio independentes. Portanto, esses dois conjuntos de regras paralelas (figuras 4.6 (a) e (b)) devem ser modelados como conjuntos conflitantes de regras, como pode ser observado nas transições  $t_3$  e  $t_4$  da figura 4.13.

No capítulo 5 são apresentadas aplicações dos algoritmos de raciocínio para frente e para trás utilizando-se o exemplo descrito acima.



## 4.13 Conclusão

O modelo de RPNAN foi aqui definido, como um modelo matemático para representação de raciocínio aproximado que complementa o modelo de RPN apresentado no capítulo anterior. Uma das vantagens oferecidas pelo modelo de RPNAN é o fato de permitir representações mais facilmente manipuláveis por tratar-se de uma abstração sobre o modelo detalhado de RPN. Foram discutidas as representações das regras modeladas com RPN no capítulo 3 e outras como fontes parciais de informação e regras com conjunção no consequente. Através do exemplo de aplicação apresentado foi possível abordar detalhes importantes a serem considerados na modelagem de um conjunto de regras de maior porte. A relação entre os elementos dos modelos de RPN e RPNAN foi brevemente discutida. A equivalência entre os dois modelos sugere ser possível optar por um ou outro modelo a qualquer momento durante o processo de definição de um sistema conforme a conveniência, sem problemas de compatibilidade entre representações. Cada um dos modelos apresentados oferece vantagens e características diferentes e se complementam na modelagem, projeto e implementação de sistemas nebulosos.

Nos próximos capítulos será demonstrado como as estruturas de RPNAN e de RPN podem ser utilizadas na definição de algoritmos de raciocínio e técnicas de verificação de integridade.

# Capítulo 5

## Algoritmos de Raciocínio Aproximado

### 5.1 Introdução

Neste capítulo, dois tipos de algoritmos para raciocínio aproximado, baseados nos modelos de RPN e RPNAN são definidos. Na seção 5.2 é apresentado o algoritmo de raciocínio com encadeamento para frente, com base no modelo de RPN. O algoritmo é também aplicável a RPNAN, desde que os procedimentos sejam detalhados adequadamente. A execução desse algoritmo é exemplificada com um subconjunto da base de regras apresentada na seção 4.12, relativo a modelagem de sistemas de manufatura. O algoritmo de raciocínio com encadeamento para trás é apresentado na seção 5.4 e sua execução é ilustrada na seção 5.5, também sobre o exemplo da manufatura. Na seção 5.6 um novo exemplo de execução é apresentado, com casos de fontes parciais de informação. Neste segundo exemplo, as duas etapas do algoritmo são descritas detalhadamente.

### 5.2 Algoritmo de Encadeamento para Frente

O algoritmo de encadeamento para frente será descrito nesta seção com base no modelo de RPN. Em seguida será demonstrado que a mesma descrição, com pequenas alterações é também aplicável ao modelo de RPNAN. Inicialmente são introduzidas algumas definições adicionais, a maior parte delas sendo extensões de conceitos já definidos.

**Definição 5.2.1** Seja  $P^i = \{p_1^i, \dots, p_r^i\}$  um grupo de lugares e  $\alpha(p_j^i) = a_j^i, \forall p_j^i \in P^i$ . Seja  $X^i$  um conjunto base,  $X^P = \{X^1, X^2, \dots, X^N\}$  e  $P^P = \{P^1, P^2, \dots, P^n\}$ . Defina-se  $\alpha^*$  e  $\beta^*$  como extensões de  $\alpha$  e de  $\beta$ , respectivamente:

$$\begin{aligned}\alpha^* : P^P &\mapsto [0, 1]^n \\ \alpha^*(P^i) &= \langle a_1^i, a_2^i, \dots, a_n^i \rangle, P^i \in P^P \\ \beta^* : P^P &\mapsto X^P \\ \beta^*(P^i) &= X^i\end{aligned}$$

**Definição 5.2.2** Para todo grupo de transições  $T^i \subset T$ , um conjunto de entrada  $I^*(T^i)$  e um conjunto de saída  $O^*(T^i)$  são introduzidos:

$$\begin{aligned}I^*(T^i) &= \{P^l \mid \forall p_j^l \in P^l, \exists t_k^i \in T^i, \langle p_j^l, t_k^i \rangle \in F\} \\ O^*(T^i) &= \{P^l \mid \forall t_k^i \in T^i, \exists p_j^l \in P^l, \langle t_k^i, p_j^l \rangle \in F\}\end{aligned}$$

**Definição 5.2.3** Para todo grupo de lugares  $P^i \subset P$ , um conjunto de entrada  $I^*(P^i)$  e um conjunto de saída  $O^*(P^i)$  são introduzidos:

$$\begin{aligned}I^*(P^i) &= \{T^l \mid \forall t_j^l \in T^l, \exists p_k^i \in P^i, \langle t_k^i, p_j^l \rangle \in F\} \\ O^*(P^i) &= \{T^l \mid \forall p_j^l \in P^l, \exists t_k^i \in T^i, \langle p_j^l, t_k^i \rangle \in F\}\end{aligned}$$

**Definição 5.2.4** Seja  $P^i$  um grupo de lugares e  $O^*(P^i) = \{T^{i1}, \dots, T^{il}\}$ . O conjunto de grupos de lugares imediatamente alcançável de  $P^i$  é o conjunto

$$IRG(P^i) = \cup_{j=1}^l O^*(T^{ij}).$$

**Definição 5.2.5** Seja  $P^i$  um grupo de lugares e  $T^{ik} \in O^*(P^i)$ . O conjunto de grupos de lugares imediatamente alcançável de  $P^i$  por  $T^{ik}$  é o conjunto

$$IRG(P^i, T^{ik}) = O^*(T^{ik}).$$

**Definição 5.2.6** Seja  $T^k$  um grupo de transições,  $P^i$  e  $P^j$  grupos de lugares. Se  $P^i \in I(T^k)$  e  $P^j \in I(T^k)$  então  $P^i$  e  $P^j$  são chamados grupos de lugares adjacentes com relação a  $T^k$ .

No algoritmo descrito nesta seção é usada uma estratégia de encadeamento para frente. A rede é marcada inicialmente com senhas que representam os fatos na base de conhecimento. Lugares que contenham uma senha inicial são chamados *lugares iniciais*. Os grupos de lugares que contenham lugares iniciais são chamados *grupos iniciais*. O algoritmo gera automaticamente todos os caminhos de raciocínio a partir dos lugares iniciais, até que não haja mais transições habilitadas a disparar. Novos valores de senhas são calculados conforme as transições disparam. Lugares marcados com uma senha logo após o término da execução do algoritmo são chamados *lugares objetivos*. Grupos de lugares contendo lugares objetivos são chamados *grupos objetivos*.

Suponha-se que todos os conjuntos base tem o mesmo número  $n$  de elementos e que os fatos na base de conhecimento são da forma

$$V_i \text{ é } A'_i, i = 1, \dots, q .$$

Logo, deve-se ter  $q$  grupos de lugares  $P^i$  com uma marcação nebulosa inicial  $\alpha(p_k^i) = a_k^i, i = 1, \dots, q$  e  $k = 1, \dots, n$ .

O algoritmo de encadeamento para frente pode ser representado por uma estrutura de árvore. Cada nó na árvore pode ser denotado por uma  $n$ -upla

$$N = (I_1, I_2, \dots, I_q) .$$

Cada elemento  $I_i$  de  $N$  contém informações referentes a um grupo de lugares marcados na rede. Logo, a raiz da árvore terá sempre tantos elementos  $I_i$  quantos forem os grupos iniciais. Cada  $I_i \in N$  é uma tripla

$$(P^i, \alpha^*(P^i), L_i)$$

onde  $P^i$  é um grupo de lugares e  $\alpha^*(P^i)$  é uma  $n$ -upla de valores de senhas nesse grupo de lugares. O terceiro elemento  $L_i$  é um conjunto de pares que indica quais os grupos de transições e de lugares que são alcançáveis a partir de  $P^i$ . O primeiro elemento do par é um grupo de transições imediatamente alcançável de  $P^i$ . O segundo elemento do par é o conjunto de grupos de lugares imediatamente alcançáveis dessa transição.  $L_i$  é definido por:

$$\{(T^{i1}, IRG(T^{i1}, P^i)), (T^{i2}, IRG(T^{i2}, P^i)), \dots, (T^{il}, IRG(T^{il}, P^i))\}$$

onde  $\{T^{i1}, \dots, T^{il}\} = O^*(P^i)$ .

Seja  $Type(T^k)$  o tipo da transição  $T^k$  e  $R_k^i = IRG(T^k, P^i)$ .

Seja  $[P^i \times P^j]$  a matriz cujos elementos são os pesos dos arcos conectando lugares do grupo  $P^i$  com as transições de  $T^{ik}$ , tal que  $IRG(P^i, T^{ik}) = P^j$ . Logo, a cada linha de  $[P^i \times P^j]$  corresponde um lugar de  $P^i$  e a cada coluna de  $[P^i \times P^j]$  corresponde uma transição de  $T^{ik}$ .

Seja  $AP^{ij}$  o conjunto de grupos adjacentes a  $P^i$  com relação a  $P^j$ ,  $P^j \in IRG(P^i)$ .

Supondo que os grupos de lugares  $P^1, P^2, \dots, P^q$  são grupos iniciais, o algoritmo de encadeamento para frente pode ser descrito como segue.

**Algoritmo 5.2.1** *Raciocínio de encadeamento para frente*

**Entrada:** Rede RPNAN e marcação inicial

**Saída:** Árvore de execução

1. Faça o nó da raiz  $(I_1, I_2, \dots, I_q)$  um nó não terminal, onde cada  $I_i$  é uma tripla  $(P^i, M^i, L_i)$  tal que

- $P^i$  é um grupo inicial, para  $i = 1, \dots, q$ .
- $M^i = \alpha^*(P^i) = A_i^!$ .
- $L_i$  é um conjunto de pares da forma definida acima.

2. Selecione um nó não terminal  $N = (I_1, I_2, \dots, I_q)$ .

Para cada  $I_i = (P^i, M^i, L_i)$  faça:

SE  $O^*(P^i) = \emptyset$

ENTÃO marque o nó  $N$  como terminal

SENÃO Para todo par  $(T^k, R_k^i) \in L_i$  faça

SE  $R_k^i = \emptyset$

ENTÃO crie um novo nó marcado como nó de corte

SENÃO BEGIN (Construa a  $n$ -upla  $Next$  para ser o próximo nó da árvore)

Faça  $Next = []$  (lista vazia)

SE  $T^k$  é um BRP

ENTÃO Aplique recursivamente o algoritmo

a partir da raiz  $N = (I_i)$  e faça

$Next =$  nó terminal resultante dessa aplicação.

SENÃO SE  $Type(T^k) = 3$

ENTÃO Para todo  $P^j \in R_k^i$  faça

BEGIN

Call evaluate(3,  $P^j, M^i, [], I_j$ )

insira  $I_j$  em  $Next$

END

SENÃO BEGIN

Faça  $P^j$  ser o único membro de  $R_k^i$ .

SE  $AP^{ij} \neq \emptyset$

ENTÃO Para todo  $P^l \in AP^{ij}$  faça

BEGIN

SE  $P^l$  não foi calculado

ENTÃO remova  $P^j$  de  $R_k^i$   
no nó  $N$

END

evaluate( $Type(T^k), P^j, M^i, AP^{ij}, I_j$ )

insira  $I_j$  em  $Next$

END

crie um novo nó com rótulo  $Next$  e um arco

com rótulo  $T^k$ , direcionado do nó  $N$  ao

nó  $Next$ , marque esse nó como não terminal

END

3. SE não houverem mais nós não terminais, pare, senão vá para o passo 2.

Seja  $M^l$  a marcação nebulosa de  $P^l, \forall P^l \in AP^{ij}$ . Seja  $Q = \{l_1, \dots, l_k\}$  o conjunto de índices dos elementos  $P^l \in AP^{ij}$ . O procedimento *evaluate* que calcula os valores das senhas, é definido como:

**Algoritmo 5.2.2** Cálculo de uma nova marcação.

**Entrada:** Tipo da transição  $T^k$ , Grupo  $P^j$ , Marcação  $M^i$ , conjunto de lugares adjacentes  $AP^{ij}$ .

**Saída:** Tripla  $I_j$

evaluate( $Type(T^k), P^j, M^i, AP^{ij}, I_j$ )

BEGIN

Build\_list( $P^j, L_j$ )

case Type of

1:  $M^j = \sup(M^i \gamma [P^i \times P^j])$

2:  $M = (\delta_{l \in Q} M^l) \delta M^i$

3:  $M^j = M^i$

4:  $M^j = (\tau_{l \in Q} M^l) \tau M^i$

crie a tripla  $I_j = (P^j, M^j, L_j)$

END

O procedimento  $Build\_List(P^j, L_j)$  monta a lista  $L_j$  para o grupo de lugares  $P^j$ , consultado a matriz de incidência da rede.

### 5.3 Aplicação do Algoritmo de Encadeamento para Frente

Para ilustrar a execução do algoritmo de encadeamento para frente será considerada novamente a base de regras nebulosas da Tabela 3.3 do capítulo 3, cuja RPN encontra-se na figura 3.15. Nessa figura os grupos de lugares e os grupos de transições são identificados por um rótulo  $P^i$  ou  $T^i$ . Os conjuntos de regras paralelas são representados por um *bloco de regras paralelas*, conforme definido na seção 3.7. Note-se que essas regras são um subconjunto da base de regras do exemplo de modelagem do sistema de manufatura, mostrada nas Tabelas 4.6 e 4.7.

Suponha que os seguintes fatos são conhecidos:

O tamanho do lote é cerca de 60,  
O tamanho do lote de transferência é cerca de 15.

ou, de uma maneira mais formal,

$$V_1 \text{ é } A'_1 \text{ e } V_2 \text{ é } A'_2$$

onde  $A'_1$  e  $A'_2$  são subconjuntos nebulosos dos conjuntos base  $X^1$  e  $X^2$  respectivamente, com os valores de pertinência mostrados na Tabela 5.1. Os grupos de lugares  $P^1$  e  $P^2$ , associados com  $V_1$  e  $V_2$  respectivamente são marcados com senhas que levam os valores  $M^1 = (0, 0, 0, 0, 0.5, 1, 0.5, 0, 0, 0, 0, 0)$  e  $M^2 = (0, 0.1, 1, 0.1, 0, 0, 0, 0)$ . Esta situação está representada na raiz da estrutura de árvore da figura 5.1.

Com a execução do algoritmo, todas as transições habilitadas disparam sucessivamente, até que não haja mais transições habilitadas. A árvore que representa a execução do algoritmo se expande como na figura 5.1, onde as marcações (segundo elemento da tripla associada ao nó) foram omitidas, para simplificação.

No estado final, a rede tem senhas nos lugares associados às variáveis

*tempo final* ( $V_6$ ),  
*utilização da CN.72* ( $V_5$ ) e  
*tamanho médio da fila* ( $V_7$ ),

$X_1$	10	20	30	40	50	60	70	80	90	100	110	120	130
$A'_1$ (cerca de 60)	0	0	0	0	0.5	1	0.5	0	0	0	0	0	0

$X_2$	5	10	15	20	25	30	35	40
$A'_2$ (cerca de 15)	0	0.1	1	0.1	0	0	0	0

Tabela 5.1: Subconjuntos Nebulosos com rótulos *cerca de 60* e *cerca de 15*.

cujos valores podem ser interpretados linguisticamente por

*O tempo final é quase médio,  
A utilização da CN-72 é pequena e  
O tamanho médio da fila está entre pequeno e médio.*

Este estado final está representado na árvore gerada pelo algoritmo, que tem 3 nós terminais, cujos rótulos são as triplas referentes aos grupos de lugares  $P^8$ ,  $P^9$  e  $P^{11}$ , associados a  $V_6$ ,  $V_5$  e  $V_7$ , respectivamente.

A execução do algoritmo de encadeamento para frente pode ser também acompanhada considerando-se a RPNAN. Por exemplo, a figura 5.2 mostra a estrutura básica da RPNAN equivalente à RPN da figura 3.15, na qual foi aplicado o algoritmo. Cada lugar  $p_i$  e transição  $t_j$  da figura 5.2 corresponde ao grupo de lugares  $P^i$  e grupo de transições  $T^j$  da figura 3.15, respectivamente. Dados os fatos, os lugares  $p_1$  e  $p_2$  são marcados com as senhas correspondentes. A representação em árvore do algoritmo considerando-se a RPNAN é a mesma da representação para RPN, apenas substituindo-se os grupos de lugares e grupos de transições por lugares e transições da RPNAN.

## 5.4 Algoritmo de Encadeamento para Trás

O algoritmo descrito nesta seção usa uma estratégia de encadeamento para trás. Este algoritmo é usado para responder questões que pedem o valor de uma variável específica, como

$V$  é ?

O último lugar com rótulo  $V$  — com relação a orientação da RPNAN — é considerado o lugar objetivo. A partir deste, procura-se sub-objetivos e fatos que dêem



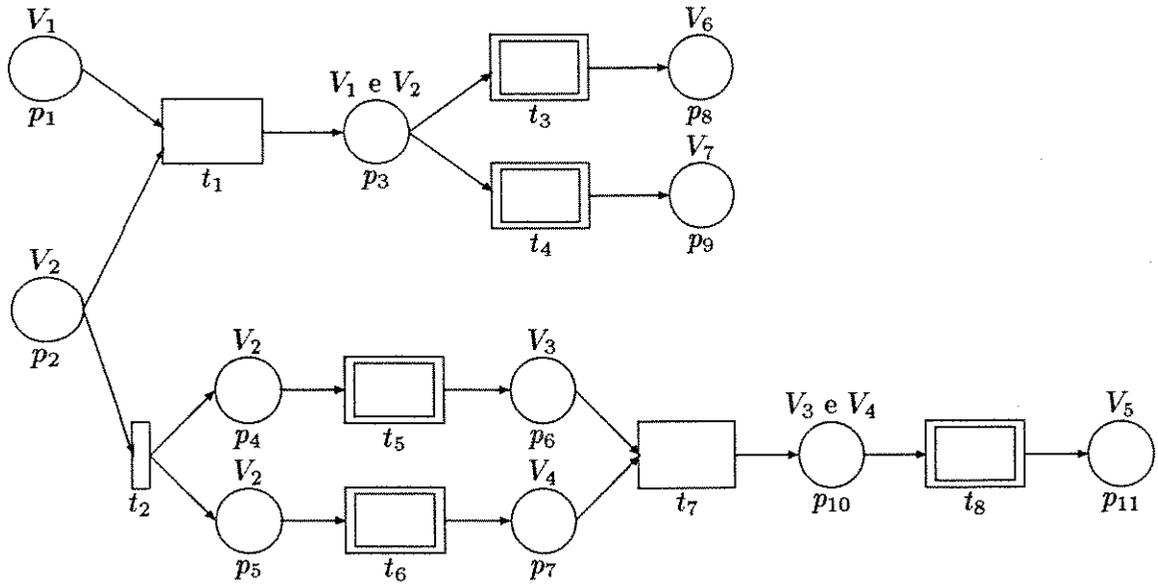


Figura 5.2: RPNAN representando um subconjunto das regras do exemplo de manufatura.

O algoritmo utiliza duas estruturas de fila, para acompanhar os lugares e transições que pertencem a sub-rede desejada. A expressão  $p := delete(Q)$  representa a operação de eliminar o primeiro elemento da fila  $Q$  e atribuí-lo a  $p$ . A expressão  $insert(x, Q)$  representa a operação de inserir  $x$  como o último elemento da fila  $Q$ .

Suponha-se que a consulta

$V$  é ?

foi colocada. O lugar  $p$  com rótulo  $V$  é o *lugar objetivo*. Suponha também que a base de conhecimento está modelada em uma RPNAN  $N$ . Na fase 1, o algoritmo encontra a sub-rede  $N'$  contendo um subconjunto de lugares  $P'$  e um subconjunto de transições  $T'$ . Para isso, a matriz de incidência  $C^*$  da RPA  $N^*$  (veja definição 4.11.1) é pesquisada para se encontrar conexões entre  $P'$  e  $T'$ . Na fase 2,  $N'$  é executada.

O algoritmo pode ser descrito como segue.

**Algoritmo 5.4.1** *Raciocínio com encadeamento para trás*

**Entrada:** rede RPNAN, consulta  $V$  é ?

**Saída:** valor inferido para  $V$

• Fase 1:

1. Insira o lugar  $p$  no conjunto de lugares  $P'$  e na fila  $Q_p$ .
2. Enquanto  $Q_p \neq \text{empty}$  faça  
 $p_1 := \text{delete}(Q_p)$   
 Seja  $t$  o elemento de  $I(p_1)$   
 Se  $t$  não está em  $T'$  então insira  $t$  em  $T'$  e  $\text{insert}(t, Q_t)$
3. Enquanto  $Q_t \neq \text{empty}$  faça  
 $t_1 := \text{delete}(Q_t)$   
 Seja  $C_1$  a coleção de elementos em  $I(t)$   
 Para todo  $p \in C_1$  faça  
 Se  $p$  não está em  $P'$  então insira  $p$  em  $P'$  e  $\text{insert}(p, Q_p)$
4. Se  $Q_p \neq \text{empty}$  então vá para o passo 2, caso contrário continue.
5. Considere o subconjunto de  $C$  contendo somente os lugares de  $P'$  e as transições de  $T'$ .

• Fase 2:

1. Aplique o algoritmo de encadeamento para frente descrito na seção 5.2 [56] à sub-rede derivada na fase 1.

## 5.5 Aplicação do Algoritmo de Encadeamento para Trás - caso 1

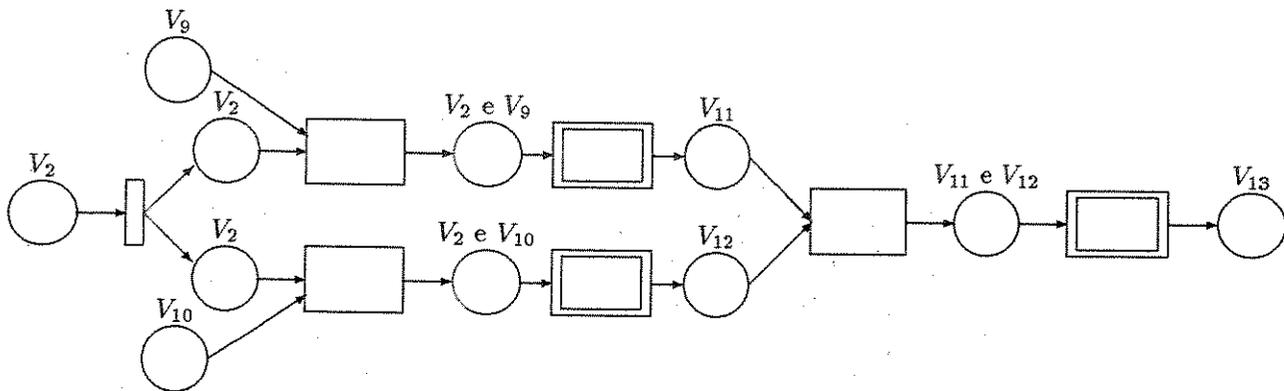


Figura 5.3: Sub-rede extraída pelo raciocínio de encadeamento para trás.

Considere-se novamente o exemplo de modelagem do sistema de manufatura descrito na seção 4.12. Suponha que a questão

*Qual é o valor da variável  $V_{13}$ ,*

ou

*Qual é a utilização da CN.72 em operações de processamento?*

foi colocada. O lugar  $V_{13}$  da figura 4.13 é considerado o lugar objetivo. Na fase 1 do algoritmo, a rede é pesquisada de trás para frente e a sub-rede extraída contendo a informação relativa a  $V_{13}$  é mostrada na figura 5.3. Os nós iniciais dessa sub-rede são

*tamanho do lote de transferência ( $V_2$ ),  
tempo de processamento para operação 7 ( $V_9$ ) e  
tempo de processamento para operação 9 ( $V_{10}$ ).*

Portanto, os fatos que fornecem os valores iniciais das variáveis  $V_2$ ,  $V_9$  e  $V_{10}$  devem estar disponíveis para que se possa encontrar o valor da variável  $V_{13}$ . Neste caso, os respectivos lugares são marcados com os subconjuntos nebulosos apropriados e o algoritmo de encadeamento para frente é aplicado na segunda fase do raciocínio de encadeamento para trás.

## 5.6 Aplicação do Algoritmo com Encadeamento para Trás - caso 2

Um novo exemplo será fornecido para aplicação do algoritmo com encadeamento para trás, que contém casos de fontes parciais de informação.

**Exemplo 5.6.1** *Assuma que uma base de conhecimento  $K$  consiste da seguinte informação:*

*Fatos:*

$d_1:V_1 \text{ é } A_1$     $d_2:V_1 \text{ é } B_1$     $d_3:V_2 \text{ é } A_2$   
 $d_4:V_3 \text{ é } A_3$     $d_5:V_4 \text{ é } A_4$     $d_6:V_5 \text{ é } A_5$

*Regras:*

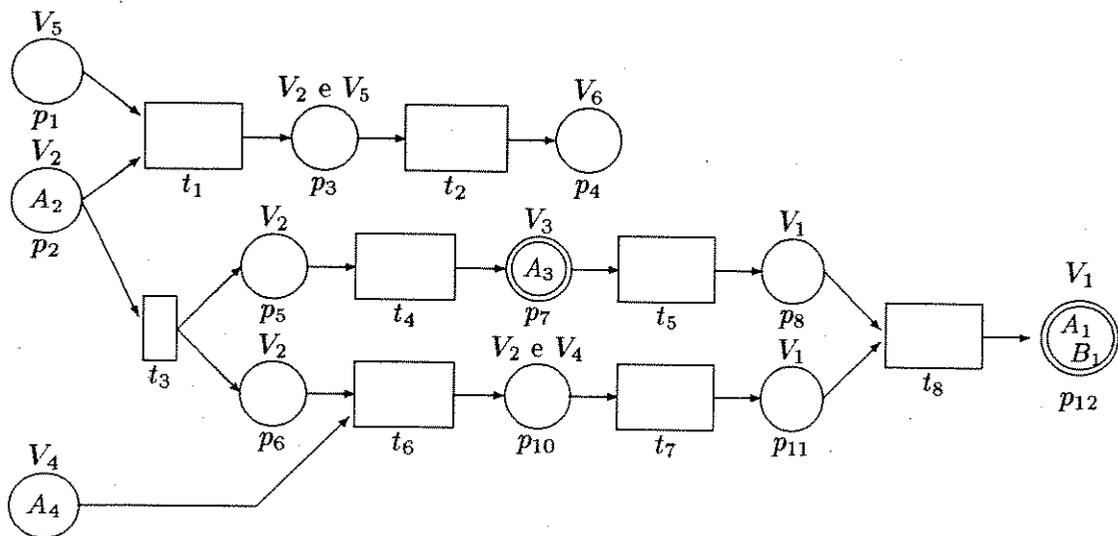


Figura 5.4: RPNAN representando a base de conhecimento do exemplo.

- $I_1 : SE V_2 \acute{e} C_2 ENT\tilde{A}O V_3 \acute{e} C_3$   
 $I_2 : SE V_2 \acute{e} D_2 E V_4 \acute{e} D_4 ENT\tilde{A}O V_1 \acute{e} D_1,$   
 $I_3 : SE V_3 \acute{e} E_3 ENT\tilde{A}O V_1 \acute{e} E_1$   
 $I_4 : SE V_2 \acute{e} F_2 E V_5 \acute{e} C_5 ENT\tilde{A}O V_6 \acute{e} C_6$

Suponha que a consulta  $V_1 \acute{e} ?$  foi colocada para essa base de regras.

O modelo de RPNAN representando a base de conhecimento do exemplo 5.6.1 pode ser visto na figura 5.4. Os lugares associados às variáveis para as quais são fornecidos um ou mais fatos, são representados por um círculo duplo. Isso indica que, durante a execução da rede, esses lugares são substituídos por sub-redes que executam a agregação dos valores das senhas que se encontram nesses lugares.

A matriz de incidência  $C^*$  da rede de Petri associada a RPNAN da figura 5.4 é dada na Tabela 5.2.

### 5.6.1 Fase 1 - Extração da subrede

A questão  $V_1 \acute{e} ?$  inicia a extração da subrede  $N'$ , que contém todas as fontes parciais de informação relacionadas a  $V_1$

	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$
$p_1$	-1	0	0	0	0	0	0	0
$p_2$	-1	0	-1	0	0	0	0	0
$p_3$	1	-1	0	0	0	0	0	0
$p_4$	0	1	0	0	0	0	0	0
$p_5$	0	0	1	-1	0	0	0	0
$p_6$	0	0	1	0	0	-1	0	0
$p_7$	0	0	0	1	-1	0	0	0
$p_8$	0	0	0	0	1	0	0	-1
$p_9$	0	0	0	0	0	-1	0	0
$p_{10}$	0	0	0	0	0	1	-1	0
$p_{11}$	0	0	0	0	0	0	1	-1
$p_{12}$	0	0	0	0	0	0	0	1

Tabela 5.2: Matriz de incidência da rede de Petri associada a RPNAN do exemplo.

O passo 1 insere o lugar  $p_{12}$  em  $P'$  e faz

$$Q_p = \{p_{12}\}.$$

No passo 2 usa-se  $p_{12}$ , o único elemento em  $Q_p$  para pesquisar a matriz de incidência  $C^*$  da RPA  $N^*$  associada à  $N$ . Como  $C^*(p_{12}, t_8) = 1$ , ou  $I(p_{12}) = \{t_8\}$ , insere-se  $t_8$  em  $T'$  e faz-se  $Q_t = \{t_8\}$ .

No passo 3 obtém-se  $I(t_8) = \{p_8, p_{11}\}$ , o conjunto  $P'$  é atualizado para  $P' = \{p_8, p_{11}, p_{12}\}$  e faz-se  $Q_p = \{p_8, p_{11}\}$ .

No passo 4, como  $Q_p \neq \emptyset$ , volta-se ao passo 2.

Aplicando-se sucessivamente os passos 2 e 3, termina-se com

$$P' = \{p_2, p_5, p_9, p_6, p_7, p_8, p_{10}, p_{11}, p_{12}\}$$

e

$$T' = \{t_3, t_4, t_6, t_5, t_7, t_8\}.$$

Consultando a Tabela 5.2 encontram-se os arcos que conectam os lugares e transições em  $P'$  e  $T'$  e finalmente obtém-se a subrede  $N'$ , conforme apresentado na figura 5.5.

A árvore de conhecimento obtida pelo procedimento proposto em [70] para o mesmo exemplo é mostrada na figura 5.6.

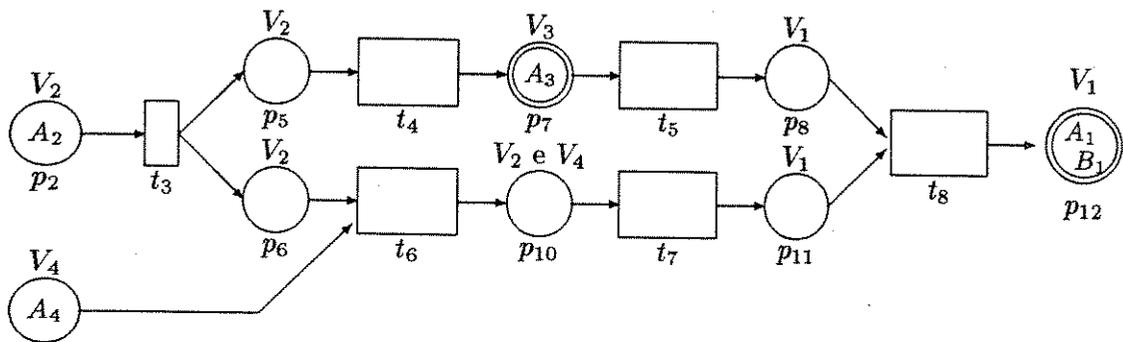


Figura 5.5: Subrede extraída pela primeira fase do raciocínio com encadeamento para trás.

A grande vantagem do algoritmo proposto aqui é a estrutura de rede, com sua matriz de incidência correspondente, que permite acessar diretamente, a cada passo do procedimento, as proposições envolvidas com a variável da consulta.

### 5.6.2 Fase 2 - Execução da subrede

A fase 2 do algoritmo consiste da execução da sub-rede extraída pela fase 1, pela aplicação do algoritmo com encadeamento para frente apresentado na seção 5.2.

Desde que a questão colocada é da forma  $V_1$  é ?, com uma única variável, a sub-rede obtida na fase 1 representa apenas um caminho de raciocínio, o que significa que não há situações de conflito na subrede. Os lugares iniciais são aqueles que têm conjunto de entrada vazio. No caso de haver algum lugar inicial sem marcação, isto é, não são fornecidos fatos relativos a variável associada a este lugar, insere-se nesse lugar uma senha cujo valor é o conjunto base da variável associada. Tal senha tem o significado de *nada sabemos sobre a variável V*.

Assuma-se que os subconjuntos nebulosos que aparecem na base de regras do exemplo 5.6.1 tem os seguintes conjuntos base:

$$\begin{aligned} X_1 &= \{a, b, c\}, & X_3 &= \{l, m\}, \\ X_2 &= \{1, 2, 3\}, & X_4 &= \{p, q\}. \end{aligned}$$

Subconjuntos de  $X_1$ :

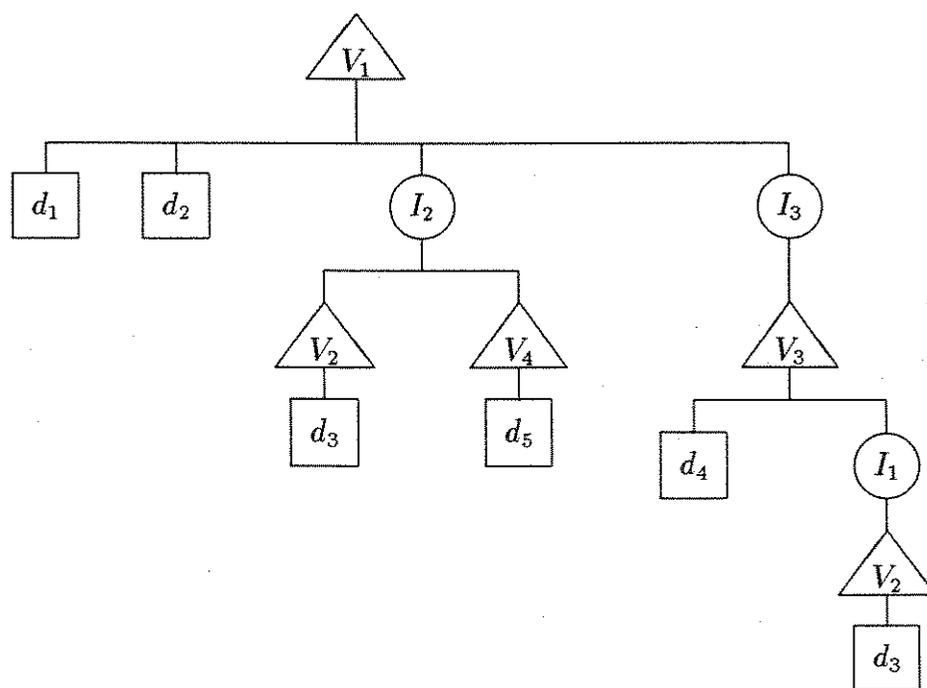


Figura 5.6: Árvore de conhecimento contendo informação parcial sobre a variável  $V_1$ .

	a	b	c
$A_1$	1	1	0.3
$B_1$	1	0.6	0.9
$D_1$	1	0.7	0.5
$E_1$	1	0.3	0.2

Subconjuntos de  $X_2$ :

	1	2	3
$A_2$	1	0.8	0.5
$C_2$	1	0.6	0
$D_2$	1	1	0.3

Subconjuntos de  $X_3$ :

	l	m
$A_3$	0.2	1
$C_3$	0.5	1
$E_3$	0.1	1

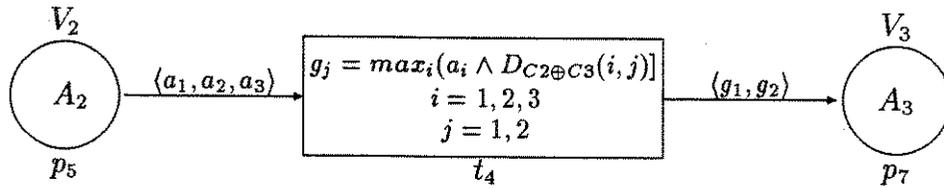
Subconjuntos de  $X_4$ :

	p	q
$A_4$	1	0.5
$D_4$	1	0.4

Assuma também que a função de implicação  $f_{\rightarrow}(A, B)$  usada neste exemplo é a *soma limitada*, definida por

$$D_{A \oplus B}(x, y) = 1 \wedge [1 - A(x) + B(y)].$$

Inicialmente, apenas a transição  $t_3$  está habilitada. A transição  $t_3$  não tem fórmulas inscritas, é uma transição de duplicação. Logo após o seu disparo, são inseridas senhas nos lugares  $p_5$  e  $p_6$ , ambas com o mesmo rótulo da senha no lugar de entrada da transição  $t_3$  antes do disparo. Nesse novo estado, as transições  $t_4$  e  $t_6$  estão habilitadas. A transição  $t_4$  tem as fórmulas inscritas mostradas em detalhe na figura



$$A_2 = \langle 1, 0.8, 0.5 \rangle \quad A_3 = \langle 0.2, 1 \rangle$$

Figura 5.7: Fórmulas inscritas na transição  $t_4$ .

5.7, onde  $D_{C2 \oplus C3}(i, j)$  denota o elemento na linha  $i$  e coluna  $j$  da matriz  $D_{C2 \oplus C3}$  dada a seguir, representando a relação sobre  $X^2 \times X^3$ , induzida pela implicação  $I_1$ :

$$D_{C2 \oplus C3} = \begin{bmatrix} 0.5 & 1 \\ 0.9 & 1 \\ 1 & 1 \end{bmatrix}$$

De acordo com as regras de disparo da RPNAN, as variáveis  $a_1$ ,  $a_2$ , e  $a_3$  da figura 5.7 recebem os valores 1, 0.8 e 0.5 respectivamente. O cálculo das fórmulas de  $t_4$  resultam em  $g_1 = 0.8$  e  $g_2 = 1$ , que são os valores da senha inserida em  $p_7$ . O lugar  $p_7$  tem agora duas senhas: uma correspondente ao fato  $d_4:V_3$  é  $A_3$  e uma correspondente ao cálculo da implicação  $I_1$ . Isto significa que a execução da rede neste ponto requer que o lugar  $p_7$  seja substituído por uma sub-rede, para realizar conjunção dos valores das senhas antes de prosseguir. Este processo será discutido mais adiante.

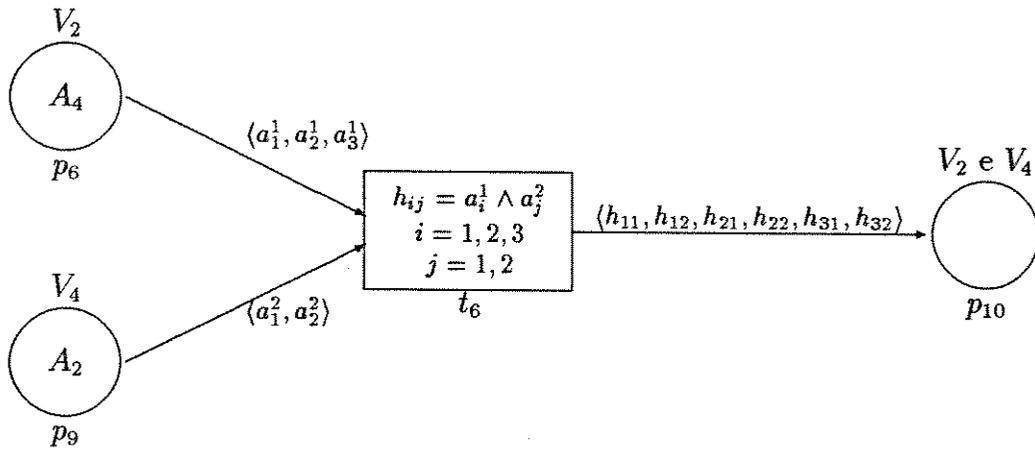
As fórmulas inscritas na transição  $t_6$  realizam a conjunção dos fatos nos seus lugares de entrada, os quais correspondem aos fatos sobre as variáveis  $V_2$  e  $V_4$ , que aparecem no antecedente da implicação  $I_2$ . As fórmulas são mostradas em detalhe na figura 5.8. O disparo da transição  $t_6$  muda o estado da rede para um novo estado onde uma senha com rótulo  $\langle 1, 0.8, 0.5, 0.5, 0.5, 0.5 \rangle$  é colocada no lugar  $p_{10}$  representando a relação  $A_2 \cap A_4$ .

A transição  $t_7$  agora está habilitada. Suas fórmulas inscritas são apresentadas na figura 5.9. Observe-se inicialmente que a implicação  $I_2$  induz uma relação nebulosa sobre  $X^2 \times X^4 \times X^1$  dada por

$$D_{D2, D4 \oplus D1}(v_2, v_4, v_1) = 1 \wedge [1 - D_{D2, D4}(v_2, v_4) + D_1(y)] \quad (5.1)$$

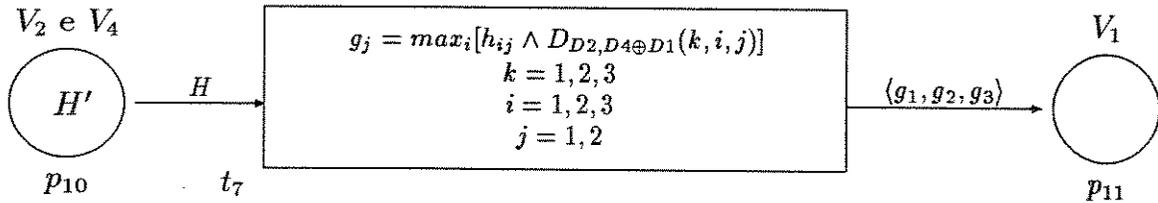
onde

$$D_{D2, D4} = \begin{bmatrix} 1 & 0.4 \\ 1 & 0.4 \\ 0.3 & 0.3 \end{bmatrix}.$$



$$A_2 = \langle 1, 0.8, 0.5 \rangle \quad A_4 = \langle 1, 0.5 \rangle$$

Figura 5.8: Fórmulas inscritas na transição  $t_6$ .



$$H' = \langle 1, 0.8, 0.5, 0.5, 0.5, 0.5 \rangle$$

$$H = \langle h_{11}, h_{12}, h_{21}, h_{22}, h_{31}, h_{32} \rangle$$

Figura 5.9: Fórmulas inscritas na transição  $t_7$ .

A fórmula de (5.1) produz os seguintes valores, apresentados na forma matricial:

$y = a$	$y = b$	$y = c$
1 1	0.7 1	0.5 1
1 1	0.7 1	0.5 1
1 1	1 1	1 1

O resultado do disparo da transição  $t_7$  é uma marcação onde uma senha com rótulo  $\langle 1, 0.5, 0.5 \rangle$  é colocada no lugar  $p_{11}$ .

Agora o caso do lugar  $p_7$  será discutido. Este lugar deve ser substituído por uma sub-rede como a da figura 5.10. A transição  $t_{7,1}$  realiza conjunção de valores de

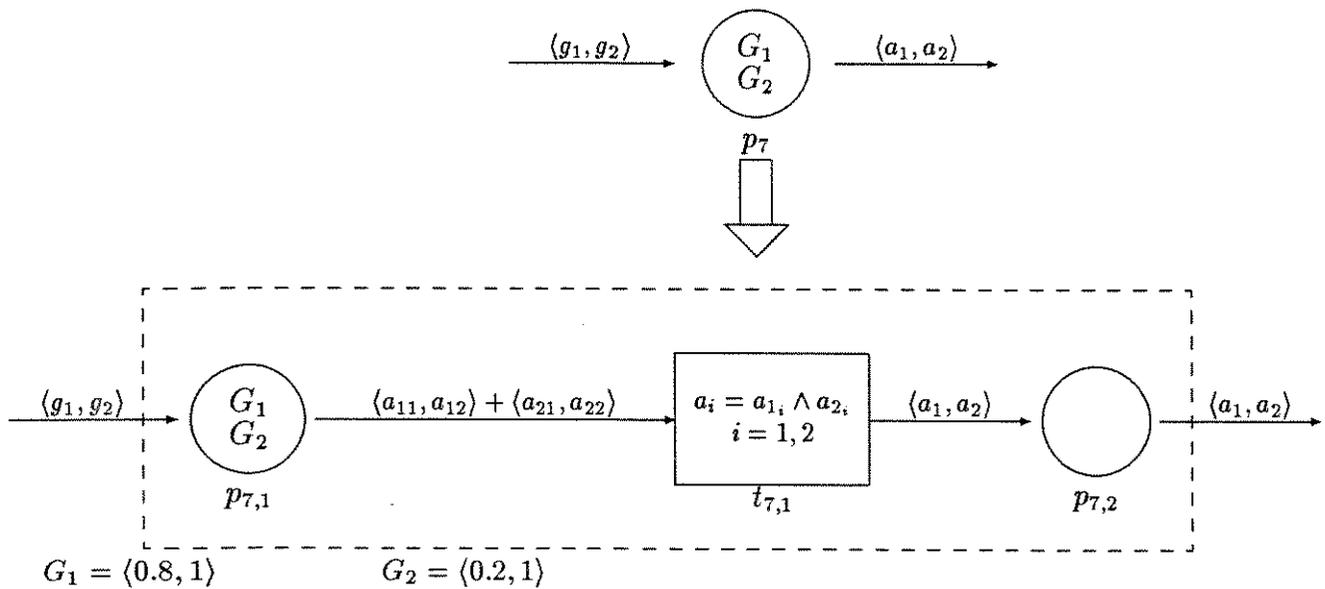


Figura 5.10: Subrede detalhada que substitui o lugar  $p_7$ .

senhas no seu lugar de entrada. Note-se que o rótulo  $\langle a_{11}, a_{12} \rangle + \langle a_{21}, a_{22} \rangle$  requer que existam pelo menos duas senhas no lugar  $p_{7,1}$  para que a transição  $t_{7,1}$  esteja habilitada. Por esta razão, a transição  $t_5$  não está habilitada antes que  $t_4$  tenha disparado, mesmo que exista uma senha inicial em  $p_7$ . O disparo da transição  $t_{7,1}$  produz a senha  $\langle 0.2, 1 \rangle$ , que é colocada no lugar  $p_{7,2}$  pois o rótulo do arco  $\langle a_1, a_2 \rangle$  recebe os valores  $a_1 = 0.2$  e  $a_2 = 1$ . A transição  $t_5$  agora está habilitada.

O disparo da transição  $t_5$  é semelhante ao da transição  $t_7$ , e o resultado é a colocação da senha  $\langle 1, 0.5, 0.5 \rangle$  no lugar  $p_8$ .

Finalmente, a transição  $t_8$  fornece um único valor para  $V_1$ , pela agregação dos dois resultados parciais para  $V_1$  obtidos através de  $I_1$  e  $I_2$ , e insere uma senha no lugar  $p_{12}$ . Como o lugar  $p_{12}$  já possui duas outras senhas devido a existência de dois fatos relativos a  $V_1$ , o mesmo procedimento aplicado ao lugar  $p_7$  deve ser utilizado para se encontrar um valor final para  $V_1$ , que é  $V_1 = \{1/a, 0.3/b, 0.2/c\}$ .

## 5.7 Conclusão

Neste capítulo foram apresentados dois tipos de algoritmos de raciocínio aproximado, que utilizam as estratégias de encadeamento para frente e encadeamento para trás.

O algoritmo com encadementamento para frente pode ser representado por uma estrutura de árvore. A partir de uma dada situação inicial, todos os caminhos de raciocínio possíveis são gerados. O algoritmo com encadementamento para trás é executado em duas fases. Na primeira fase, uma sub-rede é extraída da RPNAN completa e, na segunda fase, essa sub-rede é executada pelo algoritmo com encadementamento para frente. O tratamento dado a diferentes casos de fontes parciais de informação também foi abordado. Ambos os algoritmos utilizam alguma representação na forma matricial da rede para obter as informações necessárias a sua execução. Isso mostra que a representação de uma base de regras nebulosas nos modelos de RPN e RPNAN auxilia a execução dos algoritmos, por fornecer uma organização das regras e fatos onde as relações de dependência entre proposições são facilmente obtidas.

# Capítulo 6

## Verificação de Inconsistência

### 6.1 Introdução

Neste capítulo é abordado o problema de verificação de integridade de bases de regras nebulosas. Os problemas que podem ocorrer em uma base de conhecimento e comprometer sua integridade são usualmente classificados como problemas de redundância, inconsistência e incompletude. O estudo apresentado aqui restringe-se ao problema de inconsistência. Será apresentada uma técnica para verificação de inconsistências em uma base de regras nebulosas a nível local e global, baseada no modelo de RPNAN. Serão adotados um conceito particular de inconsistência e uma metodologia para encontrar condições necessárias para a existência de conflitos em potencial apresentados em [78]. Uma vez determinadas essas condições, os algoritmos de verificação local e global utilizam as propriedades estruturais da RPNAN para agilizar o processo de verificação.

Na seção 6.2 é apresentado um breve resumo sobre os problemas encontrados no campo de verificação de integridade de sistemas especialistas e, em especial, de sistemas nebulosos. A classificação dos problemas que podem ocorrer também é abordada. Na seção 6.3 são apresentados os conceitos de matriz de testes de condições, RPNAN nível 1 e composição matricial, usados nos algoritmos. O método de *reflexão na entrada*, que fornece a base dos procedimentos propostos neste capítulo é introduzido na seção 6.4, que inclui também a discussão da aplicação desse método a casos novos, não analisados em [78]. Finalmente, as seções 6.5 e 6.6 apresentam os algoritmos de verificação de inconsistência a nível local e global, respectivamente, com exemplos de aplicação.

## 6.2 Verificação de Bases de Conhecimento

Apesar do desenvolvimento de métodos de verificação para “software” procedural convencional, vários problemas são encontrados quando se tenta aplicá-los a sistemas baseados em conhecimento. Alguns desses problemas são devidos a separação entre inferência e conhecimento e a natureza não procedural dos sistemas especialistas. A necessidade de métodos para verificação e validação de sistemas baseados em conhecimento tem sido constantemente apontada na literatura. Vários estudos bastante abrangentes sobre as abordagens a esse problema podem ser encontrados, como por exemplo [1], [11], [42], [45]. Grande parte das técnicas propostas fornece um mecanismo automático para verificação de integridade da base de regras e o problema de assegurar a integridade de uma base de regras tem se tornado cada vez mais tratável com cada uma dessas técnicas. Algumas das propostas mais significativas encontram-se em [1], [44], [87], [19]. A maior parte dos trabalhos sobre esse assunto se concentra na validação de conhecimento baseado em regras.

Enquanto algumas das abordagens enfocam a validade externa dos sistemas, isto é, o grau de correspondência entre as respostas do especialista e as recomendações do sistema, outras tratam da validade interna do conhecimento, isto é, a correção lógica das regras. O escopo da presente discussão está restrito à verificação de validade interna do sistema, a nível local e global. A verificação a nível local é aquela realizada entre pares de regras individuais, que têm o mesmo consequente. A verificação a nível global envolve o encadeamento de duas ou mais regras.

Vários tipos de problemas podem ser introduzidos nas bases de conhecimento conforme estas são modificadas com o tempo. Nazareth [42] descreve uma taxonomia exaustiva de erros lógicos. Agarwall [1] classifica erros com base no fato da ocorrência ser a nível local ou global. A cada nível podem ocorrer problemas como redundância, inconsistência e incompletude. Neste trabalho é investigada a questão da inconsistência.

O processo de localização de inconsistências em uma base de regras nebulosas é difícil e deve ser diferente daquele para sistemas baseados em regras tradicionais. Em [11], a validação de sistemas baseados em regras nebulosas é visto como sendo uma extensão ou generalização do processo para sistemas não nebulosos. Um tópico fundamental nesse contexto é como determinar o grau de semelhança entre duas expressões nebulosas [69]. Com o intuito de definir formas de medir a proximidade de subconjuntos nebulosos, uma medida de afinidade é criada em [31] com base na medida de similaridade. Com essa medida é possível, de acordo com [31], aplicar as técnicas para verificação de inconsistência de contextos não nebulosos a contextos nebulosos. A inconsistência de regras de controle foi formalmente estudada em [48],

onde as diferenças entre relações nebulosas ou subconjuntos nebulosos são representadas pela medida de possibilidade para se definir o índice de inconsistência entre um par de regras e entre uma regra com as restantes.

Uma abordagem formal para a verificação de inconsistência foi sugerida em [76], com base em uma espécie de inferência para trás aplicada à base de conhecimento, chamada de *reflexão na entrada*. A ocorrência de inconsistências foi discutida no contexto de diversos cálculos para tratamento de incerteza, inclusive lógica nebulosa. Esse método fornece as bases da abordagem apresentada neste capítulo.

Uma regra é dita *estruturada* quando o antecedente consiste somente de conjunções e o conseqüente conclui o valor de apenas um parâmetro [47]. O conceito de regra estruturada pode ser obviamente estendido para regras de produção nebulosas. Neste capítulo assume-se que as regras de produção nebulosas estão na forma estruturada.

## 6.3 Conceitos Básicos

Inicialmente serão introduzidas as noções básicas necessárias à definição do procedimento a ser proposto neste capítulo. Os conceitos apresentados serão ilustrados sobre a BC exibida na Tabela 6.1. O modelo de RPNAN representando esta BC pode ser visto na figura 6.1. As regras paralelas 4, 5 e 6 da Tabela 6.1 foram denotadas na figura 6.1 pela transição  $t_2$ , que é um *bloco de regras paralelas* (veja seção 4.7).

### 6.3.1 RPNAN Hierárquica

A técnica de hierarquização de RPNAN é aplicada sempre que uma parte da rede não provê nenhuma informação relevante à análise em curso. Assim, manter a representação completa com suas respectivas matrizes exigiria mais cálculos sem oferecer nenhum benefício. A substituição de uma sub-rede por um lugar ou uma transição tem a finalidade de explicitar determinadas relações entre elementos da rede em um nível de abstração mais alto.

A seguir serão introduzidas algumas formas de hierarquização de RPNAN, usadas na definição de RPNAN nível\_1, consultada nos procedimentos de verificação propostos aqui.

1)	SE $V_1$ é $A_1$ e $V_2$ é $C_1$ ENTÃO $V_5$ é $B_1$
2)	SE $V_1$ é $A_2$ e $V_2$ é $C_2$ ENTÃO $V_6$ é $J_1$
3)	SE $V_6$ é $J_2$ e $V_7$ é $D_1$ e $V_8$ é $E_1$ ENTÃO $V_{10}$ é $F_1$
4)	SE $V_2$ é $C_2$ ENTÃO $V_9$ é $G_1$
5)	SE $V_2$ é $C_4$ ENTÃO $V_9$ é $G_2$
6)	SE $V_2$ é $C_5$ ENTÃO $V_9$ é $G_3$
7)	SE $V_3$ é $H_1$ e $V_4$ é $I_1$ ENTÃO $V_9$ é $G_4$

Tabela 6.1: Exemplo de base de conhecimento.

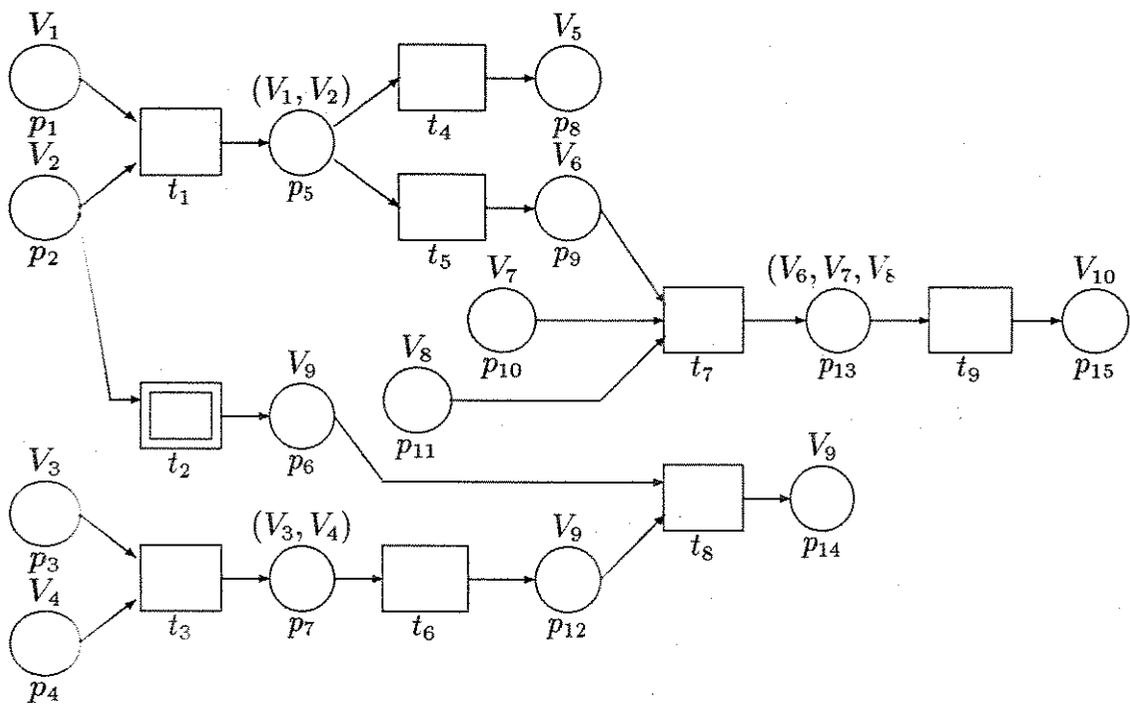


Figura 6.1: RPNAN representando a base de conhecimento de exemplo.

Com relação às regras com antecedentes múltiplos, introduzidas na seção 4.5, a sub-rede interna pode ser substituída por uma única transição que é chamada de *bloco de antecedentes múltiplos (BAM)*. BAM são denotados graficamente por um retângulo com uma borda mais larga, como na figura 6.2(a) and (b).

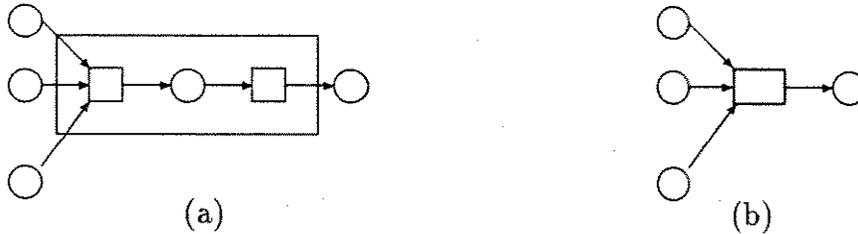


Figura 6.2: Representação hierárquica de regras com antecedentes múltiplos. (a)RPNAN com sub-rede detalhada. (b)RPNAN hierárquica com BAM.

Regras paralelas podem ser representadas conforme apresentado na figura 6.3(a), onde a transição  $t_1$  realiza uma multiplicação e a transição  $t_{n+2}$  realiza agregação. Em uma representação hierárquica, as sub-redes iniciais e finais podem ser substituídas por um único lugar chamado *círculo de regras paralelas (CRP)* (veja figura 6.3(b)). O objetivo de tal representação é agrupar todos os lugares associados com a mesma variável em um único lugar, para captar somente a informação referente às relações do tipo antecedente-consequente. Os CRP são representado por círculos duplos.

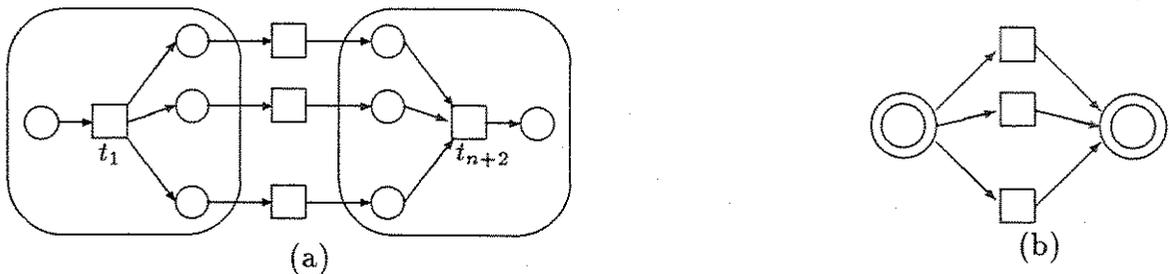


Figura 6.3: Representação hierárquica de regras paralelas. (a)RPNAN com sub-redes detalhadas. (b)RPNAN com CRP.

Fontes parciais de informação (seção 4.9) também podem ser representadas de forma hierárquica, onde a sub-rede final é substituída por um único lugar. Este

lugar é chamado de *bloco de fontes parciais (BFP)* e é representado por um círculo triplo. Um exemplo pode ser visto na figura 6.4(a) e (b).

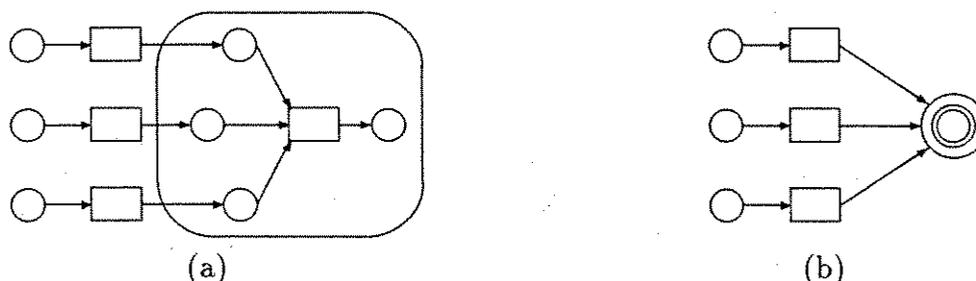


Figura 6.4: Representação Hierárquica para fontes parciais de informação. (a)RPNAN com sub-rede detalhada. (b)RPNAN com BFP.

Um tipo particular de RPNAN hierárquica é definido a seguir.

**Definição 6.3.1** *Seja  $K$  uma BC modelada pela RPNAN  $\mathcal{N}$ . Suponha que todos os conjuntos de regras paralelas de  $\mathcal{N}$  estão representados por círculos de regras paralelas, regras com antecedentes múltiplos estão representadas por blocos de antecedentes múltiplos e todos os conjuntos de fontes parciais de informação estão representados por círculos de fontes parciais. Então  $\mathcal{N}$  é chamada uma RPNAN hierárquica de nível\_1, ou simplesmente RPNAN nível\_1.*

Por exemplo, a RPNAN nível\_1 relativa a BC da Tabela 6.1 é ilustrada na figura 6.5. Note que em uma RPNAN nível\_1  $\mathcal{N}$  existe uma relação de um para um entre o conjunto de variáveis da BC e o conjunto de lugares de  $\mathcal{N}$  e uma relação de um para um entre o conjunto de regras na BC e o conjunto de transições de  $\mathcal{N}$ . Deve-se também observar que na figura 6.5, o bloco de regras paralelas que aparece na figura 6.1 como transição  $t_2$  foi expandido e as três transições associadas a cada uma das regras do conjunto de regras paralelas está representada explicitamente.

### 6.3.2 Matriz de Teste de Condições

O conceito de *matriz de teste de condições* é essencial para o procedimento apresentado nesta seção.

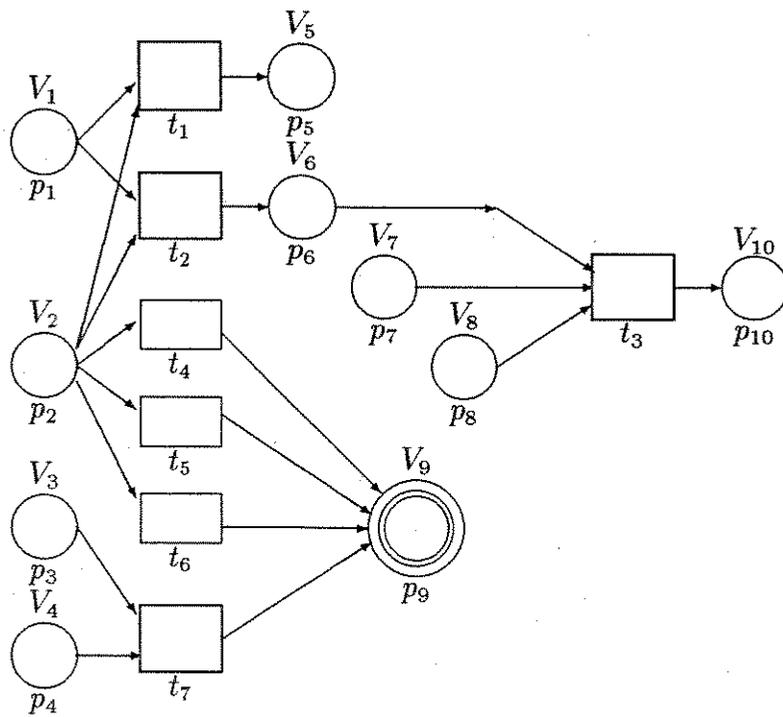


Figura 6.5: RPNAN nível 1 representando a base de conhecimento do exemplo.

		(1)	(2)	(3)	(4)	(5)	(6)	(7)
		$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$
(V <sub>1</sub> )	$p_1$	$-A_1$	$-A_2$	0	0	0	0	0
(V <sub>2</sub> )	$p_2$	$-C_1$	$-C_2$	0	$-C_3$	$-C_4$	$-C_5$	0
(V <sub>3</sub> )	$p_3$	0	0	0	0	0	0	$-H_1$
(V <sub>4</sub> )	$p_4$	0	0	0	0	0	0	$-I_1$
(V <sub>5</sub> )	$p_5$	$B_1$	0	0	0	0	0	0
(V <sub>6</sub> )	$p_6$	0	$J_1$	$-J_2$	0	0	0	0
(V <sub>7</sub> )	$p_7$	0	0	$-D_1$	0	0	0	0
(V <sub>8</sub> )	$p_8$	0	0	$-E_1$	0	0	0	0
(V <sub>9</sub> )	$p_9$	0	0	0	$G_1$	$G_2$	$G_3$	$G_4$
(V <sub>10</sub> )	$p_{10}$	0	0	$F_1$	0	0	0	0

Tabela 6.2: Exemplo de matriz-TC.

**Definição 6.3.2** Considere uma RPAN nível 1  $\mathcal{N}1$  correspondente a uma BC qualquer. Suponha que cada transição de  $\mathcal{N}1$  está associada com uma regra  $R$  tal como

$$R : SE V_1 \text{ é } A_1 \text{ e } \dots \text{ e } V_k \text{ é } A_k \text{ ENTÃO } U \text{ é } B \quad (6.1)$$

A matriz de teste de condições (matriz-TC)  $C$  de  $\mathcal{N}1$  é definida como uma matriz  $n \times m$  tal que cada linha corresponde a um lugar  $p$  de  $\mathcal{N}1$  e cada coluna corresponde a uma transição  $t$  de  $\mathcal{N}1$ . O elemento  $c(p_i, t)$  é definido por

$$c(p_i, t) = \begin{cases} -A_i & \text{se } p_i \in I(t) \text{ e está associado com } V_i \text{ em } R \\ B & \text{se } p_i \in O(t) \text{ e está associado com } U \text{ em } R \\ 0 & \text{caso contrário} \end{cases}$$

Por exemplo, a matriz-TC da RPAN nível 1 da figura 6.5 é mostrada na Tabela 6.2.

### 6.3.3 Composição Matricial

A composição matricial é fundamental para a aplicação dos algoritmos propostos nas seções 6.5 e 6.6.

**Definição 6.3.3** Sejam  $A = (a_{ij}), i = 1, \dots, n, j = 1, \dots, m$  e  $B = (b_{jk}), j = 1, \dots, m, k = 1, \dots, l$  matrizes cujos elementos são subconjuntos nebulosos sobre

conjuntos base finitos. Então a composição matricial é definida tal que  $A * B = (c_{ik}), i = 1, \dots, n, k = 1, \dots, l$ , onde

$$c_{ik} = \sum_{j=1}^m S(a_{ij}b_{jk})$$

e

$$S(a, b) = \begin{cases} Poss(A/B) & \text{se } A \text{ e } B \text{ são conjuntos não nulos com mesmo sinal} \\ 0 & \text{caso contrário} \end{cases}$$

O símbolo  $\sum$  indica uma soma formal de valores.

No contexto de verificação de inconsistência, a composição matricial será usada para comparar novas regras com regras já existentes na BC. Portanto,  $A$  é sempre um vetor linha e  $C$  é um vetor coluna. Em cada linha de  $C$ , o número de componentes diferentes de zero na soma formal indica quantas proposições da regra sendo acrescentada na BC interferem com as proposições das regras existentes.

## 6.4 Inconsistências na Lógica Nebulosa

Aborda-se nesta seção o problema de verificação de inconsistência quando usa-se lógica nebulosa para representar conhecimento. Em [78], foi proposta uma metodologia para encontrar conflitos em potencial em uma base de conhecimento, fazendo-se uma espécie de inferência para trás, chamada *reflexão na entrada*. Suponha-se que  $V$  é uma variável de entrada de uma parte de uma BC de um sistema especialista com domínio  $D$ . Se nenhum valor de entrada para  $V$  é fornecido e o valor de  $V$  é pedido, o processo de *reflexão na entrada* é aplicado para se executar o sistema no sentido contrário. Foi demonstrado em [78] que, quando a base de conhecimento contém conflitos em potencial, o valor inferido para  $V$ , em vez de ser todo o domínio de  $V$ , é um subconjunto deste. Este tipo de resposta a inconsistências em potencial ocorre usando-se diferentes métodos para representação de incerteza .

A seguir é descrito como essa metodologia é aplicada em um ambiente de lógica nebulosa considerando-se diferentes casos de inconsistências locais e globais, a maioria deles não tratada em [78]. Tais casos fornecem a base para os procedimentos apresentados na seção 6.5 e 6.6 para encontrar conflitos em potencial usando o modelo de RPNAN. Antes de prosseguir, o conceito de inconsistência com relação a uma variável deve ser definido. Um subconjunto nebuloso  $A$  é considerado inconsistente se for subnormal [78].

**Definição 6.4.1** O grau de inconsistência de um subconjunto nebuloso  $F$  de um conjunto base  $Y$  é definido como

$$c = Cert(U \text{ é } \emptyset / U \text{ é } F) .$$

onde  $\emptyset$  denota o conjunto vazio.

Como

$$Cert = (U \text{ é } \emptyset / U \text{ é } F) = 1 - Poss(Y/F) .$$

então

$$Cert = (U \text{ é } \emptyset / U \text{ é } F) = 1 - \max_y [F(y)] .$$

O grau de inconsistência é uma medida de quão distante o valor de uma variável está de ser um subconjunto normal. Logo, se  $F$  é normal não há inconsistência, isto é,  $c = 0$ .

Na discussão que segue, subconjuntos nebulosos aparecendo em fatos são chamados *conjuntos de entrada*.

Assuma que temos as proposições

$$P_1 : SE V \text{ é } A_1 \text{ ENTÃO } U \text{ é } B_1$$

$$P_2 : SE V \text{ é } A_2 \text{ ENTÃO } U \text{ é } B_2$$

$$P_3 : V \text{ é } D$$

De acordo com conceitos introduzidos na seção 2.2.3, a conjunção dessas três proposições resulta em

$$(U, V) \text{ é } H$$

onde

$$H(x, y) = (\bar{A}_1(x) \vee B_1(y)) \wedge (\bar{A}_2(x) \vee B_2(y)) \wedge D(x) .$$

Denotando o valor de  $U$  como a projeção de  $H$  no conjunto base  $Y$ , pode-se mostrar que [60]

$$F(y) = (Poss(\bar{A}_1 \wedge \bar{A}_2 / D) \vee (Poss(\bar{A}_1 / D) \wedge E_2(y)) \vee (Poss(\bar{A}_2 / D) \wedge B_1(y)) \vee (B_1(y) \wedge B_2(y)) .$$

O processo de reflexão na entrada, quando aplicado a um ambiente de regras nebulosas como as proposições acima pode ser resumido como segue. Para obter um grau de inconsistência de  $\alpha$  para a variável assumindo o valor  $F$ , deve ocorrer

$$F(y) \leq 1 - \alpha, \forall y$$

Então tem-se

$$B_1(y) \wedge B_2(y) \leq 1 - \alpha, \forall y$$

isto é,

$$Poss(B_1/B_2) \leq 1 - \alpha .$$

Além disso é também necessário que, para todo  $y$ ,

$$(Poss(\bar{A}_1/D) \wedge B_2(y)) \vee (Poss(\bar{A}_2/D) \wedge B_1(y)) \leq 1 - \alpha$$

o que garante que

$$Poss(\bar{A}_1 \wedge \bar{A}_2/D) \leq 1 - \alpha .$$

É possível provar facilmente que [78] essa condição implica em

$$Poss(A_1/A_2) \geq \alpha .$$

Portanto temos duas condições necessárias para obter qualquer inconsistência:

1.  $Poss(B_1/B_2) \leq 1 - \alpha < 1$
2.  $Poss(A_1/A_2) \geq \alpha > 0$

para algum  $\alpha \in (0, 1]$ .

Considere-se agora o mesmo conjunto de proposições, excluindo-se o fato  $V$  é  $D$ . A conjunção dessas regras resulta em

$$(V, U) \text{ é } H$$

tal que

$$H(x, y) = (\bar{A}_1(x) \vee B_1(y)) \wedge (\bar{A}_2(x) \vee B_2(y)) .$$

Se for feita a reflexão em  $V$  pela projeção, obtém-se  $E$  tal que

$$E(x) = \max_y [H(x, y)] .$$

Com as suposições

$$Poss(B_1 \cap B_2) = \alpha_1 < 1$$

e

$$Poss(A_1 \cap A_2) = \alpha_2 > 0,$$

é possível demonstrar que existe um  $x$  tal que

$$E(x) < 1.$$

Este resultado atua como um indicador da existência de um conflito em potencial entre as proposições  $P_1$  e  $P_2$  acima.

Um resultado semelhante pode ser obtido quando existem várias proposições no antecedente. Assuma que  $V_1, \dots, V_k$  são  $k$  variáveis com conjuntos base  $X^1, \dots, X^k$ , respectivamente. Seja  $X = X^1 \times \dots \times X^k$ . Assuma que temos as regras

$$P_1 : \text{SE } V_1 \text{ é } A_1 \text{ e } \dots \text{ e } V_k \text{ é } A_k \text{ ENTÃO } U \text{ é } B_1$$

$$P_2 : \text{SE } V_1 \text{ é } D_1 \text{ e } \dots \text{ e } V_k \text{ é } D_k \text{ ENTÃO } U \text{ é } B_2$$

Supondo  $A = A_1 \times A_2 \times \dots \times A_k$  e  $D = D_1 \times D_2 \times \dots \times D_k$ , as condições necessárias para conflitos em potencial são

$$1. \text{ Poss}(B_1/B_2) \leq 1 - \alpha$$

$$2. \text{ Poss}(A/D) \geq \alpha.$$

O caso acima é na verdade um caso particular de

$$P_1 : \text{SE } V_1 \text{ é } A_1 \text{ e } \dots \text{ e } V_k \text{ é } A_k \text{ ENTÃO } U \text{ é } B_1$$

$$P_2 : \text{SE } V_1 \text{ é } D_1 \text{ e } \dots \text{ e } V_l \text{ é } D_l \text{ ENTÃO } U \text{ é } B_2$$

onde  $l \leq k$ .

A ausência das variáveis  $V_{l+1}, \dots, V_k$  na proposição  $P_2$  significa que os valores dessas variáveis não são conhecidos. Pode-se então completar as proposições restantes do antecedente de  $P_2$  com as proposições  $V_{l+1}$  é  $X^{l+1}$  e  $\dots$  e  $V_k$  é  $X^k$ . As condições necessárias para existência de conflito em potencial encontradas nesse caso são

$$1. \text{ Poss}(B_1/B_2) \leq 1 - \alpha$$

$$2. \text{ Poss}(A/D') \geq \alpha$$

onde  $A = A_1 \times A_2 \times \dots \times A_k$  e  $D' = D_1 \times D_2 \times \dots \times D_l \times X^{l+1} \times \dots \times X^k$

Observe-se que uma das condições acima envolve variáveis conjuntas, o que pode, as vezes, tornar os procedimentos de verificação de inconsistência bastante complexos. O método de reflexão na entrada aplicado a um antecedente individual  $V_j$ , entretanto, pode não produzir um conjunto base menor que  $X^j$ . Como o interesse maior está em encontrar condições que exijam cálculos razoavelmente simples para construir um procedimento de verificação, esta exigência não será imposta em casos como esses. Para derivar as condições usa-se então o fato de que, se

$$Poss(A/D) \geq \alpha$$

então

$$Poss(A_1/D_1) \geq \alpha, Poss(A_2/D_2) \geq \alpha, \dots, Poss(A_k/D_k) \geq \alpha.$$

As condições derivadas são condições mais fracas, mas também são condições necessárias para a existência de conflitos em potencial e não requerem cálculos sobre conjuntos compostos.

Sejam  $V$ ,  $U$  e  $W$  variáveis com conjuntos base  $X$ ,  $Y$  e  $Z$ . Considere-se as duas regras

$$\begin{aligned} &SE\ V\ \acute{e}\ A\ ENT\tilde{A}O\ U\ \acute{e}\ B_1 \\ &SE\ W\ \acute{e}\ D\ ENT\tilde{A}O\ U\ \acute{e}\ B_2 \end{aligned}$$

A condição  $Poss(B_1/B_2) \leq 1 - \alpha$  é suficiente para garantir um conflito em potencial.

O mesmo tipo de análise pode ser aplicada à um contexto onde existe uma inconsistência global, isto é, após a ocorrência de um encadeamento de regras. Considere-se as regras

$$\begin{aligned} P_1 &: SE\ V\ \acute{e}\ A_1\ ENT\tilde{A}O\ U\ \acute{e}\ B_1 \\ P_2 &: SE\ U\ \acute{e}\ G_2\ ENT\tilde{A}O\ W\ \acute{e}\ S_2 \\ P_3 &: SE\ V\ \acute{e}\ A_3\ ENT\tilde{A}O\ W\ \acute{e}\ S_3 \end{aligned}$$

As condições necessárias para existência de conflitos em potencial são

1.  $Poss(S_2/S_3) \leq 1 - \alpha$
2.  $Poss(A_1/A_3) \geq \alpha$
3.  $Cert(G_2/B_1) \geq \alpha$ .

Uma situação mais complicada pode aparecer em uma BC, como:

$$\begin{aligned}
 P_1 &: \text{SE } V_1 \text{ é } A_1 \text{ e } V_2 \text{ é } A_2 \text{ ENTÃO } U_1 \text{ é } B_1 \\
 P_2 &: \text{SE } U_1 \text{ é } B_2 \text{ e } U_2 \text{ é } B_3 \text{ e } U_3 \text{ é } B_4 \text{ ENTÃO } W \text{ é } C_2 \\
 P_3 &: \text{SE } V_2 \text{ é } A_3 \text{ e } U_3 \text{ é } B_5 \text{ ENTÃO } W \text{ é } C_3
 \end{aligned}$$

onde  $V_1$  e  $V_2$  são variáveis sobre  $X$ ,  $A_1$ ,  $A_2$  e  $A_3$  são subconjuntos nebulosos de  $X$ ,  $U_1$ ,  $U_2$  e  $U_3$  são variáveis sobre  $Y$ ,  $B_1$ ,  $B_2$ ,  $B_3$ ,  $B_4$  e  $B_5$  são subconjuntos nebulosos de  $Y$ ,  $W$  é uma variável sobre  $Z$ ,  $C_2$  e  $C_3$  são subconjuntos nebulosos de  $Z$ .

A aplicação do método para esse caso leva às condições

1.  $Poss(C_2/C_3) \leq 1 - \alpha$
2.  $Poss(A_2/A_3) \geq \alpha$
3.  $Poss(B_4/B_5) \geq \alpha$
4.  $Cert(B_2/B_1) \geq \alpha$ .

## 6.5 Verificação de Inconsistência a Nível Local

Para a apresentação dos algoritmos de verificação [61] [66], assume-se ter uma BC  $K$ , consistente de acordo com o conceito de consistência adotado neste contexto e modelada em uma RPNAN  $\mathcal{N}$ . Uma nova regra  $R$  da forma

$$\text{SE } V_1 \text{ é } A_1 \text{ e } \dots \text{ e } V_k \text{ é } A_k \text{ ENTÃO } U \text{ é } B$$

deve ser adicionada a  $K$ . Deseja-se então verificar se esta atualização de conhecimento pode ou não levar a uma situação de inconsistência. A RPNAN nível 1 correspondente a  $\mathcal{N}$  é denotada por  $\mathcal{N}1$  e a matriz-TC por  $C$ . Os procedimentos são descritos em pseudo-código, cujas estruturas básicas e palavras chaves tem o significado convencional de programação estruturada. Assume-se também que as regras estão na forma estruturada.

O primeiro passo no processo de verificação local é extrair da matriz-TC as colunas correspondentes a transições associadas com regras que tenham no conseqüente a mesma variável que a regra  $R$ . Suponha que  $p$  é o lugar de  $\mathcal{N}1$  associado a  $U$ , a variável do conseqüente de  $R$ .

**Algoritmo 6.5.1** Verificação de inconsistência a nível local

**Entrada:**  $C_{n \times m}$ , regra  $R$ , grau de inconsistência  $\alpha \in (0, 1]$ .

**Saída:** Mensagem

*Passo 1. (Construa uma matriz-TC  $C'$ , a partir de  $C$ , correspondente a regras com a variável  $U$  no consequente.)*

*PARA  $j = 1, \dots, m$  FAÇA*

*SE  $c(p, t_j) \neq 0$  ENTÃO inclua a coluna  $t_j$  em  $C'$ .*

*Passo 2. (Construa um vetor de teste  $V$  para a regra  $R$ .)*

*PARA  $i = 1, \dots, n$  FAÇA*

*SE a regra  $R$  tem a proposição  $V_i$  e  $G$*

*ENTÃO SE  $V_i$  e  $G$  está no antecedente de  $R$*

*ENTÃO FAÇA  $V(i) = -G$*

*SENÃO FAÇA  $V(i) = G$*

*SENÃO FAÇA  $V(i) = 0$ .*

*Passo 3. (Aplique composição matricial entre  $C'$  e  $V$ , resultando  $F$ .)*

*FAÇA  $F = V * C'$ ,*

*(onde  $*$  denota a composição matricial da definição 6.3.3.)*

*Passo 4. (Analise os valores de  $F$  comparando-os com  $\alpha$ .)*

*PARA cada elemento  $F(j) = s_1 + s_2 + \dots + s_n$  FAÇA*

*Encontre  $F'(j) = r_1, \dots, r_k$ ,  $k \geq 1$ , com todos e somente os elementos não nulos de  $F(j)$ .*

*SE  $r_k \leq 1 - \alpha$  e  $r_1 > \alpha$  e  $\dots$  e  $r_{k-1} > \alpha$*

*ENTÃO retorne a mensagem As condições necessárias para a existência de conflitos em potencial entre a nova regra  $R$  e a regra  $j$  estão satisfeitas.*

*SENÃO retorne a mensagem As condições necessárias para a existência de conflitos em potencial entre a nova regra  $R$  e a regra  $j$  não estão satisfeitas.*

Por exemplo, suponha que a regra

$SE V_2 \text{ é } C_6 \text{ ENTÃO } V_9 \text{ é } G_5$

deve ser inserida na BC da Tabela 6.1. A variável  $V_9$  está associada com o lugar  $p_9$  de  $\mathcal{N}1$ , a RPNAN nível.1 da figura 6.5. A matriz  $C'$  é criada com o mesmo número de linhas que  $C$  e inclui apenas as quatro últimas colunas de  $C$  (Tabela 6.2), pois as regras 4, 5, 6 e 7 da Tabela 6.1 têm  $V_9$  no consequente.

O vetor de teste de uma regra tem tantos elementos quantas são as variáveis da BC, que é também o número de linhas da matriz-TC. Seus elementos são definidos como na matriz-TC. Para a regra 8, o vetor de teste é

$$V = [0 \quad -C_6 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad G_5 \quad 0].$$

A composição matricial resulta em

$$[0 \quad -C_6 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad G_5 \quad 0] * \begin{bmatrix} 0 & 0 & 0 & 0 \\ -C_3 & -C_4 & -C_5 & 0 \\ 0 & 0 & 0 & -H_1 \\ 0 & 0 & 0 & -I_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ G_1 & G_2 & G_3 & G_4 \\ 0 & 0 & 0 & 0 \end{bmatrix} = F$$

onde

$$F = \begin{bmatrix} 0 + Poss(C_6/C_3) + 0 + 0 + 0 + 0 + 0 + 0 + Poss(G_5/G_1) + 0 \\ 0 + Poss(C_6/C_4) + 0 + 0 + 0 + 0 + 0 + 0 + Poss(G_5/G_2) + 0 \\ 0 + Poss(C_6/C_5) + 0 + 0 + 0 + 0 + 0 + 0 + Poss(G_5/G_3) + 0 \\ 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + Poss(G_5/G_4) + 0 \end{bmatrix}$$

Cada elemento do vetor coluna resultante é uma soma formal de valores de possibilidades onde o último elemento diferente de zero corresponde à medida de possibilidade entre conjuntos dos consequentes das regras comparadas e os demais correspondem a medida de possibilidade entre conjuntos dos antecedentes. A última linha do vetor  $F$  do exemplo acima indica que não existem variáveis comuns entre os antecedentes das regras 7 e 8, pois apenas um elemento da soma formal é diferente de zero. Como exemplo, suponha que todas as variáveis têm o mesmo conjunto base  $X = \{1, 2, 3\}$  e sejam

$$\begin{array}{lll} C_3 = \{1/1, 1/2, 0.5/3\} & C_4 = \{0.7/1, 0.3/2, 1/3\} & C_5 = \{1/1, 0.2/2, 0/3\} \\ C_6 = \{0.6/1, 0.4/2, 1/3\} & G_1 = \{1/1, 0.5/2, 0.6/3\} & G_2 = \{0/1, 0.2/2, 1/3\} \\ G_3 = \{0.8/1, 0.7/2, 0.2/3\} & G_4 = \{1/1, 0.4/2, 0.3/3\} & G_5 = \{1/1, 0.7/2, 0.6/3\} \end{array}$$

Neste caso, o vetor coluna resultante seria

$$\begin{bmatrix} 0 + 0.6 + 0 + 0 + 0 + 0 + 0 + 0 + 1 + 0 \\ 0 + 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0.2 + 0 \\ 0 + 0.6 + 0 + 0 + 0 + 0 + 0 + 0 + 0.8 + 0 \\ 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 + 0 \end{bmatrix}$$

A segunda linha indica a existência de um conflito em potencial entre as regras 5 e 8 se for considerado qualquer valor de  $\alpha \leq 0.8$ , pois

$$\begin{aligned} Poss(C_6/C_4) &= 1 \geq \alpha \\ Poss(G_4/G_2) &= 0.2 \leq 1 - \alpha \end{aligned}$$

## 6.6 Verificação de Inconsistência a Nível Global

O procedimento descrito a seguir [61] [66] realiza primeiro a composição matricial entre a nova regra e as regras da sequência de encadeamentos com a qual esta nova regra interfere, para verificar a extensão em que antecedentes e consequentes se sobrepõem. A verificação positiva dessas condições já é um indício de que um conflito pode existir, mas ainda há uma condição necessária adicional a ser verificada: a que indica com que extensão o conjunto do antecedente da segunda regra na sequência de encadeamento da BC está contido no conjunto do consequente da primeira regra da sequência de encadeamento, usando a medida de certeza entre conjuntos nebulosos (veja seção 6.4). A verificação da referida condição requer nova consulta à matriz-TC para obtenção das informações necessárias além do cálculo de expressões em termos de complementos de conjuntos, já que

$$Cert(J_1/J_2) = 1 - Poss(\overline{J_1}/J_2).$$

O resultado da verificação pode fortalecer a evidência de conflitos em potencial (se positivo) ou eliminar essa evidência (se negativo). Para testar a integridade de uma BC a nível global é necessário identificar inicialmente a parte relevante da RPNAN completa que é afetada pela regra, isto é, a *cobertura mínima* da nova regra. A *cobertura mínima* de uma regra é definida como a menor parte não desconexa da RPNAN global que contém regras com variáveis que aparecem na nova regra [1].

**Algoritmo 6.6.1** *Verificação de inconsistência a nível global*

**Entrada:**  $C_{n \times m}$ , regra  $R$ , grau de inconsistência  $\alpha \in (0, 1]$ .  
**Saída:** Mensagem

*Passo 1. (Construa a matriz-TC  $C'$  a partir de  $C$ , correspondente a cobertura mínima da nova regra  $R$ .)*

*Passo 2. (Construa um vetor de teste  $V$  para a regra  $R$ .)*

*PARA  $i = 1, \dots, n$  FAÇA*  
*SE a regra  $R$  tem a proposição  $V_i$  e  $G$*   
*ENTÃO SE  $V_i$  e  $G$  está no antecedente de  $R$*   
*ENTÃO FAÇA  $V(i) = -G$*   
*SENÃO FAÇA  $V(i) = G$*   
*SENÃO FAÇA  $V(i) = 0$ .*

*Passo 3. (Aplique composição matricial entre  $C'$  e  $V$ , resultando  $F$ .)*

*FAÇA  $F = V * C'$ ,*  
*(onde  $*$  denota a composição matricial da definição 6.3.3.)*

*Passo 4. (Análise os valores de  $F$  comparando-os com  $\alpha$ .)*

*Encontre  $F' = r_1, \dots, r_k$ , com todos e somente os elementos não nulos de todas as somas formais  $F(j)$ .*  
*SE  $r_k \leq 1 - \alpha$  e  $r_1 > \alpha$  e  $\dots$  e  $r_{k-1} > \alpha$*   
*ENTÃO VA PARA o passo 5*  
*SENÃO retorne a mensagem As condições necessárias para a existência de conflitos em potencial entre a nova regra  $R$  e a regra  $j$  não estão satisfeitas e pare.*

*Passo 5. (Verifique o grau de inclusão dos valores das variáveis de ligação.)*

*Sejam  $p_i, i = 1, \dots, l$  as linhas de  $C'$  com elementos não nulos, com sinais opostos  $S_i$  e  $Q_i$ .*  
*SE  $Cert(S_1/Q_1) \geq \alpha$  e  $\dots$  e  $Cert(S_l/Q_l) \geq \alpha$*   
*ENTÃO retorne a mensagem As condições necessárias para a existência de conflitos em potencial entre a nova regra  $R$  e a regra  $j$  estão satisfeitas.*  
*SENÃO retorne a mensagem As condições necessárias para a existência de conflitos em potencial entre a nova regra  $R$  e a regra  $j$  não estão satisfeitas.*

Para ilustrar o procedimento acima, assumo que a base de conhecimento da

Tabela 6.1 será atualizada pela inserção da regra

$$\text{SE } V_2 \text{ é } C_6 \text{ e } V_9 \text{ é } D_2 \text{ ENTÃO } V_{10} \text{ é } F_2$$

A cobertura mínima dessa regra é a sub-rede que inclui as duas regras ligadas pela variável  $V_6$  : regras 2 e 3. A matriz-TC  $C'$  correspondente contém somente as colunas  $t_2$  e  $t_3$  da Tabela 6.2. O vetor de teste da regra 9 é dado por

$$V = [ 0 \quad -C_6 \quad 0 \quad 0 \quad 0 \quad 0 \quad -D_2 \quad 0 \quad 0 \quad F_2 ]$$

A composição matricial resulta em

$$[ 0 \quad -C_6 \quad 0 \quad 0 \quad 0 \quad 0 \quad -D_2 \quad 0 \quad 0 \quad F_2 ] * \begin{bmatrix} -t_2 & 0 \\ -C_2 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ J_1 & -J_2 \\ 0 & -D_1 \\ 0 & -E_1 \\ 0 & 0 \\ 0 & -F_1 \end{bmatrix} = F$$

tal que

$$F = \begin{bmatrix} 0 + Poss(C_6/C_2) + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 \\ 0 + 0 + 0 + 0 + 0 + 0 + Poss(D_2/D_1) + 0 + 0 + Poss(F_2/F_1) \end{bmatrix}$$

Sejam

$$\begin{aligned} C_6 &= \{0.5/1, 0.4/2, 1/3\} & C_2 &= \{0.4/1, 1/2, 0.5/3\} & I_1 &= \{0.8/1, 1/2, 0/3\} \\ D_2 &= \{0.7/1, 1/2, 0.8/3\} & F_1 &= \{0.3/1, 1/2, 0/3\} & F_2 &= \{0.2/1, 0/2, 1/3\} \end{aligned}$$

Então,

$$Poss(C_6/C_2) = 0.5 \geq \alpha,$$

$$Poss(D_2/D_1) = 1 \geq \alpha$$

e

$$Poss(F_2/F_1) = 0.2 \leq 1 - \alpha,$$

para  $\alpha = 0.5$ .

No passo seguinte verifica-se, por consulta à matriz-TC, que  $V_6$  é o nó de ligação entre as regras 2 e 3 e, assumindo-se

$$J_1 = \{1/1, 0.8/2, 0.7/3\} \quad J_2 = \{0/1, 0.7/2, 1/3\} \quad \bar{J}_1 = \{0/1, 0.2/2, 0.3/3\}$$

obtém-se

$$Cert(J_1/J_2) = 1 - Poss(\bar{J}_1/J_2) = 0.7 \geq \alpha$$

Logo tem-se um indicador de que um conflito em potencial existe com a adição da regra 9 na BC  $K$ .

## 6.7 Conclusão

Uma nova abordagem para verificação de inconsistências de bases de regras nebulosas foi apresentada. Dois algoritmos para verificação de inconsistências a nível local e global foram definidos com base no modelo de RPNAN. A técnica proposta assume um conceito de inconsistência entre regras nebulosas e utiliza-se do método de reflexão na entrada para determinar condições necessárias para a existência de conflitos em potencial. O referido método foi inicialmente aplicado a vários casos de inconsistência a nível local e global, para que as condições fossem determinadas. A seguir, as propriedades estruturais da RPNAN foram utilizadas na definição dos procedimentos. Os algoritmos propostos usam a representação de matriz de teste de condições, definida sobre uma forma específica de RPNAN hierárquica. Essa representação hierárquica reforça a versatilidade do modelo em rede, que permite visualizar a base de conhecimento do nível de abstração adequado à obtenção da informação necessária a cada situação.

# Capítulo 7

## Conclusões e Trabalhos Futuros

Algumas conclusões e considerações gerais sobre este trabalho serão discutidas. Os principais pontos para investigação futura, levantados durante o desenvolvimento da abordagem proposta nesta tese serão relacionados.

### 7.1 Conclusões

Uma abordagem para representação de regras nebulosas usando teoria de redes foi sugerida neste trabalho. Dois modelos de representação foram propostos, um a nível numérico e outro a nível lógico. O primeiro modelo, baseado em Redes de Petri Nebulosas, permite representar detalhadamente os cálculos numéricos executados durante a aplicação da regra de inferência do Raciocínio Aproximado, chamada *modus ponens composicional*. O segundo modelo, baseado em Redes de Petri Nebulosas de Alto Nível, oferece uma ferramenta formal a um nível mais abstrato, permitindo representações mais compactas e de fácil tratamento. Os modelos propostos são bastante gerais, pois prevêm a representação de uma classe de métodos de raciocínio caracterizada por funções de implicação definidas com T-operadores.

A versatilidade dos modelos pode ser demonstrada com a discussão sobre técnicas de modelagem de vários tipos de regras. Além do modelo das regras básicas do raciocínio aproximado, foram apresentados os modelos que representam regras com antecedentes múltiplos, regras quantificadas, regras com fatores de certeza, regras paralelas, regras conflitantes, fontes parciais de informação e regras com conjunção no consequente. As técnicas de modelagem propostas aqui permitem que diferentes situações da base de regras nebulosas sejam representadas por estruturas modulares também diferentes na rede. Assim é possível tirar proveito dos métodos de análise

conhecidos, aplicados à Rede de Petri Associada, como um recurso auxiliar para projetar corretamente a base de conhecimento.

Os modelos de RPN e RPNAN são formas equivalentes de se representar uma mesma base de regras, em níveis de abstração diferentes. A correspondência entre componentes dos dois modelos foi discutida informalmente. Com isso, é possível usar um ou outro modelo, dependendo da conveniência e das exigências de cada uma das etapas de desenvolvimento de um sistema.

A hierarquização das redes RPN e RPNAN foi diversas vezes explorada neste trabalho, como um meio de se aproveitar ao máximo as vantagens da representação em rede. Do ponto de vista de projeto da base de regras, as redes hierárquicas somam-se aos dois modelos propostos como mais uma opção quanto ao nível de abstração na representação. A flexibilidade obtida com esse recurso traz vantagens imediatas nas etapas de modelagem, execução e de verificação da base de conhecimento.

Dois tipos de algoritmos para raciocínio aproximado foram apresentados, usando estratégias de encadeamento para frente e para trás. Os algoritmos de raciocínio podem fazer uso das relações de dependência entre as proposições, representadas na forma matricial, para acessar mais diretamente a informação necessária à sua execução.

No campo de verificação de integridade, mostrou-se também a utilidade dos modelos propostos. A questão de verificação de inconsistências, tanto a nível local como global, foi abordada. Com a definição de um tipo específico de redes hierárquicas e de sua representação matricial cujos elementos são subconjuntos nebulosos, é possível extrair da rede apenas as informações relevantes quanto à dependência entre antecedentes e consequentes. Assim, as medidas de possibilidade entre conjuntos são encontradas através de equações matriciais. Com isso estabeleceu-se a base para a sugestão dos procedimentos para se constatar condições que indicam a existência de conflitos em potencial.

Enfim, conclui-se que os modelos de RPN e RPNAN são ferramentas formais poderosas que oferecem uma base concreta e uniforme para tratamento dos problemas encontrados durante a representação, projeto, execução e verificação de bases de regras nebulosas.

## 7.2 Trabalhos Futuros

As investigações desenvolvidas durante as várias etapas desse trabalho fizeram surgir diversas questões interessantes que merecem ser melhor exploradas.

A representação de regras com fatores de certeza segue uma metodologia proposta na literatura, onde esse fator é absorvido pela função de implicação. Outras formas de tratar fatores de certeza podem ser analisadas futuramente.

Optou-se neste contexto por definir o disparo de transições de maneira que a similaridade entre o antecedente da regra e o fato não é considerada como uma condição. Nos modelos propostos as regras disparam desde que haja informação disponível. A inclusão de patamares mínimos como requisito para o disparo de transições é outro possível foco de pesquisa futura.

O conceito de matriz de incidência para Redes de Petri convencionais foi estendido para ser aplicado a RPN. A equação matricial que rege as mudanças de estados permite verificar a relação de alcançabilidade entre lugares, apenas com relação as posições das senhas. A investigação de outros formalismos matemáticos que permitam incluir as alterações de valores ds senhas na equação de estados pode trazer ganhos significativos à capacidade de análise da RPN.

Os algoritmos apresentados, com estratégias de encadeamento para frente e encadeamento para trás, respondem consultas que podem ser chamadas consultas de valor. Nessas consultas, os resultados procurados são valores de uma ou mais variáveis. Nas bases de regras nebulosas é possível também aplicar consultas que solicitam o grau de confirmação de uma dada informação. O tratamento desse tipo de consulta, nas RPN ou RPNAN, chamadas questões de verdade [72] [43], demanda o estudo de possíveis extensões dos modelos propostos.

O método básico para verificação de inconsistências adotado aqui, o método de reflexão na entrada, pressupõe o uso de uma função de implicação particular, que é baseada na implicação da lógica clássica ( $\bar{A} \vee B$ ). A aplicação do método consiste em usar as propriedades dos operadores *min* e *max* para reduzir as equações resultantes da conjunção e projeção a expressões que representem as medidas clássicas de possibilidade e certeza entre subconjuntos. As medidas de possibilidade e certeza podem ser estendidas, usando-se T-norma no lugar de mínimo. Um tópico de pesquisa futura é analisar a possibilidade de aplicar o mesmo tipo de método usando-se funções de implicações generalizadas por T-norma, procurando reduzir as expressões as medidas de possibilidade e certeza também estendidas.

O método de verificação de inconsistência que oferece a base para os procedimentos propostos neste trabalho é um método formal, que parte de uma definição de inconsistência e deduz algebricamente as condições a serem verificadas. Outras propostas, formais ou informais para o mesmo fim também podem obter ganhos significativos fazendo uso das estruturas de rede.

Neste trabalho foi abordado o problema de verificação de inconsistência. Outros

tipos de problemas como redundância e completude exigem um tratamento especial no contexto de regras nebulosas. Os estudos desses problemas e sua verificação nas RPN e RPNAN constituem um campo fértil para pesquisa futura.

As redes de Petri são adequadas a representação de sistemas concorrentes. A modelagem de regras nebulosas por RPN e RPNAN permite evidenciar o paralelismo existente numa base de conhecimento desse tipo em diferentes níveis. Tanto o paralelismo entre as regras quanto o paralelismo entre os cálculos numéricos da regra de inferência composicional estão representados claramente nos modelos propostos. Um bom aproveitamento desse poder de representação das redes em algoritmos de raciocínio exige estudos adicionais quanto à análise de concorrência.

A implementação dos modelos de RPN e RPNAN pode vir a ser feita utilizando-se recursos de paralelismo, ou por simulação dos mesmos em máquinas sequenciais. Uma perspectiva ampla de trabalhos futuros advém da implementação desses modelos.

A representação de conhecimento default em regras de produção nebulosas pelo uso de qualificação de possibilidade exige que sejam consideradas regras de prioridade na aplicação do princípio de conjunção/projeção. Portanto, a modelagem de tais regras pelo uso de RPN ou RPNAN não pode ser feita da mesma forma que as regras de produção nebulosas comuns. Redes de Petri são adequadas à representação de sistemas que exigem sincronização entre os eventos. A exploração dessa possibilidade visando uma possível extensão dos modelos propostos para incluir a representação de conhecimento default pode ser abordada futuramente.

O recurso de representar os modelos de forma hierarquizada foi usado em alguns pontos desse trabalho. Muitas evidências apontam na direção de que uma ferramenta muito poderosa para projeto de sistemas nebulosos pode ser obtida pela formalização de operações como essa [51] [29] [3]. Quanto à hierarquização, é necessário definir com maior precisão que tipos de sub-redes podem ser substituídas por lugares e transições. Quanto à modularização, é preciso analisar condições a serem satisfeitas por módulos de redes para que possam posteriormente ser integrados em um conjunto maior. Assim, esses recursos podem vir a ser incluídos como operadores formais dos modelos e usados no desenvolvimento de ferramentas para desenvolvimento de sistemas nebulosos.

Abordagens alternativas à inferência nebulosa são encontradas com frequência na literatura [41] [2]. Muitas delas se propõem a evitar o problema de explosão no número de estados, típica do modus ponens composicional. O estudo de formas de representação de outros métodos nos modelos de RPN e RPNAN, pode originar trabalhos interessantes. Com isso, aumenta-se ainda mais a versatilidade dos

modelos propostos quanto ao projeto e representação e ampliam-se as opções de implementações a curto prazo em equipamentos que não disponham de recursos reais de paralelismo.

Uma fase primordial no desenvolvimento de sistemas baseados em conhecimento, sejam eles nebulosos ou não, é a fase de aquisição do conhecimento. A investigação da questão da aquisição de conhecimento no contexto de RPN e RPNAN, através de meios convencionais ou de formas automáticas de aprendizado, abre um vasto campo para pesquisas futuras.

# Bibliografia

- [1] AGARWAL,R. ; TANNIRU,M. - *A Petri-Net Based Approach for Verifying the Integrity of Production Systems*, Int. J. Man-Machine Studies, 36, 1992, pp. 447-468.
- [2] BALDWIN, J. - *A New Approach to Approximate Reasoning using a Fuzzy Logic*, Fuzzy Sets and Systems, 2, 1979, pp. 309-325.
- [3] BEST, E. - *Design Methods Based on Nets*, in Advances in Petri Nets 1989, G. Rozenberg (ed.), LNCS 424, Springer-Verlag, 1989.
- [4] BRAUER, W. ; REISIG, W. ; ROZENBERG, G. (eds.) - *Petri Nets: Central Models and Their Properties*, Advances in Petri Nets 1986, Part I, Lecture Notes in Computer Science 254, Springer-Verlag, 1987.
- [5] CAO, Z. ; KANDEL, A. - *Applicability of Some Fuzzy Implication Operators*, Fuzzy Sets and Systems, 31, 1989, pp. 151-186.
- [6] CAO,T. ; SANDERSON,A. - *Task Sequence Planning Using Fuzzy Petri Nets*, Proceedings IEEE International Conf. Systems, Man and Cybernetics, Charlottesville, VA. Oct. 1991, pp.349-354.
- [7] CAO,T. ; SANDERSON,A. - *Sensor-based Error Recovery for Robotic Task Sequences Using Fuzzy Petri Nets*, Proceedings IEEE International Conf. Robotics and Automation, Nice, France, May 1992, pp.1063-1069.
- [8] CARDOSO,J. - *Sur les Reseaux de Petri avec Marquages Flous*, These présentée au Laboratoire D'Automatique et D'Analyse des Systemes du CNRS en vue de l'obtention du titre de docteur de l'universite Paul Sabatier de Toulouse, 1990.
- [9] CARDOSO, J. ; VALETTE, R. ; DUBOIS, D. - *Petri Nets with Uncertain Markings*, in Advances in Petri Nets 1990, G. Rosenberg (ed.), LNCS 483, springer-Verlag. 1990, pp. 64-78.

- [10] CHANG, C. ; LEE, R. - *Symbolic Logic and Mechanical Theorem Proving*, Academic Press, 1973.
- [11] CHANG, A. ; HALL, L. - *The validation of fuzzy knowledge-based systems*, in *Fuzzy Logic and the Management of Uncertainty*, L. Zadeh and J. Kacprzyk, eds., John Wiley & Sons, 1992, pp.589-604.
- [12] CHAUDHURY, H. ; MARINESCU, D. ; WHINSTON, A. - *Net-Based Computaional Models of Knowledge-Processing Systems*, IEEE Expert, April, 1993, pp. 79-86.
- [13] CHEN,S., KE,J. S. ; CHANG,J. - *Knowledge Representation using Fuzzy Petri Nets*, IEEE Trans. Knowledge and Data Engineering, 2, 3, 1990, pp. 311-319.
- [14] CHUN, M. ; BIEN, Z. - *A Knowledge Network and Inference Engine for Fuzzy Expert Systems Based on Fuzzy Petri Net*, Fifth IFSA World Congress, Seoul, Korea, 1993, pp. 203-206.
- [15] DALTRINI,A. ; GOMIDE,F. - *Fuzzy Simulation in Manufacturing Systems*, The fourth International Symposium on Robotics and Manufacturing, New Mexico, USA, november,1992.
- [16] DENG, Y. ; CHANG, S. - *A G-net Nodel for Knowledge Representation and Reasoning*, IEEE Trans. on Knowledge and Data Engineering, 2, 3, 1990, pp. 295-310.
- [17] DUBOIS, D. ; PRADE, H. - *A review of fuzzy sets aggregation connectives*, Information Sciences, 36, 1985, pp. 85-121.
- [18] DUBOIS, D. ; PRADE, H. - *Fuzzy sets in approximate reasoning, Part 1: Inference with possibility distributions*, Fuzzy Sets and Systems, 40, 1991, pp. 143-202.
- [19] ENAND, R. ; KAHN, G. - *A Methodology for Validating Large Knowledge Bases*, Int. J. Man-Machine Studies, 33, 1990, pp. 361-371.
- [20] FUKAMI, S. ; MIZUMOTO, M. ; TANAKA, K. - *Some Considerations on Fuzzy Conditional Inference*, Fuzzy Sets and Systems, 4, 1980, pp. 243, 273.
- [21] FULLER. R. ; Zimmermann, H. - *On Computation of the Compositional Rule of Inference Under Triangular Norms*, Fuzzy Sets and Systems, 51, 1992, pp. 267-275.

- [22] GENRICH, H. J. ; LAUTENBACH, K. - *System Modelling with High-Level Petri Nets*, Theoretical Computer Science, 13, 1981, pp. 109-136.
- [23] GENRICH, H. J. - *Predicate/Transition Nets*, in W. Brauer, W. Reisig ; G. Rozenberg(eds.), *Petri Nets: Central Models and Their Properties*, Lecture Notes in Computer Science 254, Springer-Verlag, 1986, pp.207-247.
- [24] GIORDANA, A. ; SAITTA, L. - *Modeling Production Rules by Means of Predicate Transition Networks*, Information Sciences, 35, 1, 1985, pp.1-41.
- [25] GUPTA, M. M. ; QI, J. - *Theory of T-norms and fuzzy inference methods*, Fuzzy Sets and Systems, 40, 1991, pp. 431-450.
- [26] JENSEN, K. - *Coloured Petri Nets and the Invariant- Method*, Theoretical Computer Science, 14, 1981, pp. 317-336.
- [27] JENSEN, K. - *How to Find Invariants for Coloured Petri Nets*, in *Mathematical Foundations of Computer Science 1981*, J. Gruska, M. Chytil (eds.), LNCS 118, Springer-Verlag, 1981, pp. 327-338.
- [28] JENSEN, K. - *Coloured Petri Nets*, in W. Brauer, W. Reisig ; G. Rozenberg(eds.), *Petri Nets: Central Models and Their Properties*, Lecture Notes in Computer Science 254, Springer-Verlag, 1986, pp.248-299.
- [29] JENSEN, K. - *Coloured Petri Nets : A High Level Language for System Design and Analysis*, in G. Rosenberg (ed.), *Advances in Petri Nets 1990*, LNCS 483, Springer-Verlag, 1990.
- [30] KAUFMANN, A. - *Introduction to the Theory of Fuzzy Subsets*, Academic Press, 1975.
- [31] LEUNG, K. S. ; SO, Y. T. - *Inconsistency in Fuzzy Rule-based Expert Systems*, *Proceedings of the International Conference on Fuzzy Logic & Neural Networks*, Japan, 1990, pp.849-852.
- [32] LOONEY, C. - *Fuzzy Petri Nets for Rule-Based Decision Making*, *IEEE trans. on Systems, Man and Cybernetics*, 18, 1, 1988, pp. 178-183.
- [33] MAMDANI, E. H. - *Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis*, *IEEE Trans. on Computers*, 26, 12, 1977, pp. 1182-1191.

- [34] MATSUMOTO, T. ; SAKAGUCHI, A. ; TSUJI, K. - *Fuzzy Petri Nets and Their Applications to Fuzzy Reasoning Systems Control*, Fifth IFSA World Congress, Seoul, Korea, 1993.
- [35] MIZUMOTO, M. ; FUKAMI, S. ; TANAKA, K. - *Some Methods of Fuzzy Reasoning*, in *Advances in Fuzzy Set Theory and Applications*, M. M. Gupta et al. (eds.), North-Holland Publishing Company, 1979, pp. 117-136.
- [36] MIZUMOTO, M. ; ZIMMERMANN, H. - *Comparison of Fuzzy Reasoning Methods*, *Fuzzy Sets and Systems*, 8, 1982, pp.253-283.
- [37] MIZUMOTO, M. - *Fuzzy Inference using Max- $\wedge$  Composition in the Compositional Rule of Inference*, in *Approximate reasoning in Decision Analysis*, M. M. Gupta and E. Sanchez (eds.), North-Holland Publishing Company, 1982, pp.67-76.
- [38] MURATA, T. ; ZHANG, D. - *A Predicate-Transition Net Model for Parallel Interpretation of Logic Programs*, *IEEE Trans. on Software Engineering*, 14, 4, 1988, pp. 481-497.
- [39] MURATA, T. - *Petri Nets: Properties, Analysis and Applications*, *Proceedings IEEE*, 77, 4, 1989, pp.541-580.
- [40] MURATA, T. ; SUBRAHMANIAN, V. S. ; WAKAYAMA, T. - *A Petri Net Model for Reasoning in the Presence of Inconsistency*, *IEEE Trans. on Knowledge and Data Engineering*, 3, 3, 1991, pp. 281-292.
- [41] NAFARIEH, A. ; KELLER, J. - *A New Approach to Inference in Approximate Reasoning*, *Fuzzy Sets and Systems*, 41, 1991, pp. 17-37.
- [42] NAZARETH, D. - *Issues in the verification of knowledge in rule-based systems*, *Int. J. Man-Machine Studies*, 30, 1989, pp.255-271.
- [43] NEGOITA, C. V. - *Expert Systems and Fuzzy Systems*, Benjamin/Cummings, 1985.
- [44] NGUYEN, T. ; PERKINS, W. ; LAFFEY, T. ; PECORA, D. - *Knowledge Base Verification*, *AI Magazine*, summer 1987, pp. 69-75.
- [45] O'KEEFE, R. ; O'LEARY, D. - *Expert System Verification and Validation: a Survey and Tutorial*, *Artificial Intelligence Review*, 7, 3, 1993, pp. 3-42.

- [46] PARK, D. ; CAO, Z. ; KANDEL, A. - *Investigations on the Applicability of Fuzzy Inference*, Fuzzy Sets and Systems, 49, 1992, pp. 151-169.
- [47] PEDERSEN, K. - *Well-Structured Knowledge Bases - Part I*, AI Expert, April, 1989, pp. 44-55.
- [48] PEDRYCZ, W. - *Fuzzy Control and Fuzzy Systems*, John Wiley & Sons Inc., 1989.
- [49] PETERSON, J. L. - *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, 1981.
- [50] PETRI, C. - *Introduction to General Net Theory*, in W. Brauer, ed., Net Theory and Applications, Lecture Notes in Computer Science 84, Springer-Verlag, 1980.
- [51] PINCI, V. ; SHAPIRO, R. - *An Integrated Software Development Methodology Based on Hierarchical Colored Petri Nets*, in advances in Petri Nets 1991, G. Rosenberg (Ed.), LNCS 524, Springer-Verlag, 1991.
- [52] PISKUNOV, A. - *Fuzzy Implication in Fuzzy Systems Control*, Fuzzy Sets and Systems, 45, 1992, pp. 25-35.
- [53] REDDY, P. ; BABU, M. - *Some Methods of Reasoning for Fuzzy Conditional Propositions*, Fuzzy Sets and Systems, 52, 1992, pp. 229-250.
- [54] REISIG, W. - *Place/Transition Systems*, in W. Brauer, W. Reisig, G. Rozenberg (eds.), Petri Nets: Central Models and Their Properties, Lecture Notes in Computer Science 254, Springer-Verlag pp. 117-141, 1987.
- [55] SCARPELLI,H. ; GOMIDE,F. - *Theory and Design of Knowledge Bases*, Technical Report RT-DCA 007/91, DCA/ FEE/UNICAMP, 1991.
- [56] SCARPELLI,H. ; GOMIDE,F. - *Modeling Fuzzy Reasoning using Fuzzy Petri Nets*, Technical Report RT-DCA 020/92, DCA/ FEE/UNICAMP, 1992.
- [57] SCARPELLI,H. ; GOMIDE,F. ; PEDRYCZ, W. - *Modeling Fuzzy Reasoning using High Level Fuzzy Petri Nets*, Technical Report RT-DCA 023/92, DCA/ FEE/UNICAMP, 1992. (submetido à publicação)

- [58] SCARPELLI,H.; GOMIDE,F.; YAGER, R. - *A Backward Reasoning Algorithm for High Level Fuzzy Petri Nets*, Technical Report RT-DCA 004/93, DCA/FEE/UNICAMP, 1993. (submetido à publicação)
- [59] SCARPELLI,H. ; GOMIDE,F. - *Fuzzy Reasoning and Fuzzy Petri Nets*, Fifth IFSA World Congress, Seoul, Korea, July, 5-9, 1993.
- [60] SCARPELLI,H. ; GOMIDE,F. - *Fuzzy Reasoning and Fuzzy Petri Nets in Manufacturing Systems Modeling*, Journal of Intelligent and Fuzzy Systems, 1, 3, July, 1993.
- [61] SCARPELLI,H.; GOMIDE,F. - *A High Level Fuzzy Petri Net Approach for Discovering Potential Inconsistencies in Fuzzy Knowledge Bases*, Technical Report RT-DCA 006/93, DCA/FEE/UNICAMP, 1993. (submetido à publicação)
- [62] SCARPELLI,H. ; GOMIDE,F. - *Fuzzy Reasoning and High Level Fuzzy Petri Nets*, First European Congress on Fuzzy and Intelligent Technologies, Aachen, Germany, September, 7-10, 1993.
- [63] SCARPELLI,H. ; GOMIDE,F. ; YAGER, R. - *Default Knowledge Representation in the Theory of Approximate Reasoning: An Implementation*, em *Advances in Fuzzy Systems: applications and Theory*, P. Z. Wang ; K. F. Loe (eds.), World Scientific Publishing Company, 1993.
- [64] SCARPELLI,H. ; GOMIDE,F. - *Raciocínio Aproximado e Redes de Petri de Alto Nível na Modelagem de Sistemas de Manufatura*. (submetido à publicação).
- [65] SCARPELLI,H. ; GOMIDE,F. - *High Level Fuzzy Petri Nets and Backward Reasoning*. (submetido à publicação).
- [66] SCARPELLI,H. ; GOMIDE,F. - *Consistency Checking Based on High Level Fuzzy Petri Nets*. (submetido à publicação).
- [67] SRINIVASAN, P. ; GRACANIN, D. - *Approximate Reasoning with Fuzzy Petri Nets*, IEEE Int. Conf. on Fuzzy Sustersms, San Francisco, California, USA, 1993, pp. 396-401.
- [68] TOURETZKY, D. - *The Mathematics of Inheritance Systems*, Morgan Kaufmann, 1986.
- [69] XUECHENG, L. - *Entropy, Distance Measure and Similarity Measure of Fuzzy Sets and Their Relations*, Fuzzy Sets and Systems, 52, 1992, pp. 305-318.

- [70] YAGER, R. R. - *Querying Knowledge Base Systems with Linguistic Information via Knowledge Trees.*, Int. J. Man-Machine Studies, 19, 1983, pp.73-95.
- [71] YAGER, R. R. - *Approximate Reasoning as a Basis for Rule-Based Expert Systems*, IEEE Trans. on SMC, 14, 4, 1984, pp.636-643.
- [72] YAGER, R. R. - *Knowledge Trees in Complex Knowledge Bases*, Fuzzy Sets and Systems, 15, 1985, pp. 45-64.
- [73] YAGER, R. R. - *Using Approximate Reasoning to Represent Default Knowledge*, Artificial Intelligence, 31, 1987, pp.99-112.
- [74] YAGER, R. R. - *Nonmonotonic Inheritance Systems*, IEEE Trans. on SMC, 18, 6, 1988, pp.1028-1034.
- [75] YAGER, R. R. - *On the Representation of Commonsense Knowledge by possibilistic Reasoning*, Int. J. Man-Machine Studies, 31, 1989, pp.587-610.
- [76] YAGER, R. R. - *Deductive Approximate Reasoning Systems*, IEEE Trans. on Knowledge and Data Engineering, 3, 4, 1991, pp.399-414.
- [77] YAGER, R. R. - *Connectives and Quantifiers in Fuzzy Sets*, Fuzzy Sets and Systems, 40, 1991, pp.39-75.
- [78] YAGER,R. ; LARSEN,H. - *On Discovering Potential Inconsistencies in Validating Uncertain Knowledge Bases by Reflecting on the Input*, IEEE Trans. on Systems, Man and Cybernetics, 21, 4, july/august 1991, pp. 790-801.
- [79] ZADEH, L. A. - *Fuzzy Sets*, Information and Control, 8, 1965, pp. 338-353.
- [80] ZADEH,L. - *Fuzzy Logic and Approximate Reasoning*, Syntese, 30, 1975, pp. 407-428.
- [81] ZADEH,L. - *The Concept of a Linguistic Variable and its Application to Approximate Reasoning*, Part I, Information Sciences, 8, 199-249; Part II, Information Sciences, 8, 301-347; Part III, Information Sciences, 9, 43-80.
- [82] ZADEH,L. A. - *Fuzzy Sets as a Basis for a Theory of Possibility*, Fuzzy Sets and Systems, 1, 1978, pp.3-20.
- [83] ZADEH,L. A. - *Fuzzy Logic*, Computer, April 88, 1988, pp.83-93.

- [84] ZADEH, L. A. - *A Theory of Approximate Reasoning*. Machine Intelligence 9, Hayes, Michie & Kulich, (eds.), 9, 1979, pp.149-194.
- [85] ZADEH, L. A. - *PRUF - a Meaning Representation Language for Natural Languages*, in *Fuzzy Sets and Applications: Selected Papers by L. A. Zadeh, R. R. Yager, S. Ovchinnikov, R. M. Tong, H. T. Nguyen* (eds.), John Wiley & Sons, 1987, pp. 499-568.
- [86] ZISMAN, M. D. - *Use of Production Systems for Modeling Asynchronous, Concurrent Processes*, in *Pattern-Directed Inference systems*, D. A. Waterman, F. Hayes-Roth, (eds.), Academic Press, 1978, pp. 53-68.
- [87] ZLATAREVA, N. - *Truth Maintenance Systems and Their Application for Verifying Expert System Knowledge Bases*, Artificial Intelligence Review,

13  
15