

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**UM MÉTODO DE REDUÇÃO DA ÁRVORE DE
ALCANÇABILIDADE DE SISTEMAS MODELADOS POR
REDES DE PETRI PREDICADO/TRANSIÇÃO,
MANTENDO A CAPACIDADE DE ANÁLISE**

**DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE
FEDERAL DE SANTA CATARINA PARA A OBTENÇÃO
DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA.**



0.188.271-1

UFSC-BU

Florianópolis, março de 1991.

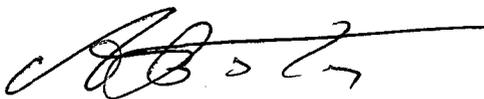
**UM MÉTODO DE REDUÇÃO DA ÁRVORE DE ALCANÇABILIDADE
DE SISTEMAS MODELADOS POR REDES DE PETRI PREDICADO/TRANSIÇÃO,
MANTENDO A CAPACIDADE DE ANÁLISE**

MARCOS MOECKE

**ESTA DISSERTAÇÃO FOI JULGADA PARA A OBTENÇÃO DO TÍTULO DE
MESTRE EM ENGENHARIA**

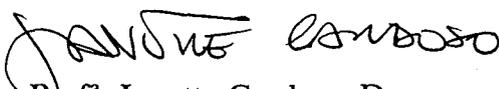
**ESPECIALIDADE ENGENHARIA ELÉTRICA, ÁREA DE CONCENTRAÇÃO
SISTEMAS DE CONTROLE E AUTOMAÇÃO INDUSTRIAL, E APROVADA EM SUA
FORMA FINAL PELO CURSO DE PÓS-GRADUAÇÃO**


Prof. Jean-Marie Farines, Dr. Ing.
Orientador


Prof. João Pedro Assumpção Bastos, Dr. D'Etat
Coordenador do Curso de Pós-Graduação em Engenharia Elétrica

BANCA EXAMINADORA:


Prof. Jean-Marie Farines, Dr. Ing.
Orientador


Profª. Janette Cardoso, Dra.


Prof. Aloisio de Castro Pinto Pedrosa, Dr.

*" O SENHOR é o meu pastor:
nada me faltará.*

*Ele me faz repousar em pastos verdejantes,
leva-me para junto das águas de descanso,
refrigera-me a alma,
guia-me pelas veredas da justiça
por amor do seu nome.*

*Ainda que eu ande pelo vale da sombra da morte,
não temerei mal nenhum,
porque Tu estás comigo:
a Tua vara e o Teu cajado me consolam. "*

Salmo 23

**A você,
Elisa,
porque eu te amo.**

AGRADECIMENTOS

Ao término deste trabalho, quero agradecer a todas aquelas pessoas que participaram de forma direta ou indireta nesta etapa da minha vida.

Inicialmente quero agradecer aos colegas de curso, "bairros" e do LCMI pela amizade e companheirismo que pudemos cultivar, tornando agradável o tempo que juntos passamos. Agradeço ao professor Jean-Marie Farines pela orientação, incentivo e confiança, que em muito ajudaram na elaboração e conclusão deste trabalho. Aos membros da Banca Examinadora agradeço pelas sugestões, críticas e outras formas de contribuição dadas. Aos professores Herve Eric Garnousset e Michel Diaz (LAAS/CNRS - França) pelas sugestões dadas durante a realização deste trabalho.

Ao povo brasileiro agradeço o apoio financeiro dado em forma de bolsa durante todo o trabalho, através do CNPQ.

A minha querida esposa, um agradecimento muito especial, pois durante todo este tempo, ela sempre esteve ao meu lado, dando o apoio, carinho, amor e compreensão que eu necessitava.

Finalmente agradeço a Deus Pai, por ter me capacitado e por ter me guiado e amparado ao longo de todo o trabalho.

RESUMO

A análise de alcançabilidade tem sido um dos métodos de análise mais utilizado para as redes de Petri, por ser aplicável a qualquer classe de rede. A sua principal limitação está na explosão dos estados que tem impedido o seu uso para sistemas de maior porte.

Neste trabalho descrevemos um novo método de reduzir a árvore de alcançabilidade das Redes Predicado/Transição (Pr/T), usando suas simetrias. O princípio básico deste método é a remoção das subárvores análogas, preservando a capacidade de análise do modelo. A recuperação das informações suprimidas na Árvore de Alcançabilidade Reduzida (AAR) é possível através do uso das funções de simetria encontradas durante a redução.

A determinação das simetrias é realizada usando um procedimento de generalização de marcações, que é descrito neste trabalho. A partir disso conseguimos uma automatização parcial da construção da árvore de alcançabilidade reduzida para o caso geral, e um automatização completa para alguns casos especiais de rede que denominamos de redes *Redes com Árvore de Alcançabilidade Diretamente Redutível (RDR)*.

Também é mostrado como é possível determinar a alcançabilidade de uma marcação da rede usando a AAR, e é apresentado um procedimento para determinar as seqüências de disparo de transições que permitem atingir esta marcação.

Comparando com outros métodos, os resultados obtidos mostram um decréscimo do tamanho da árvore de alcançabilidade e uma construção e análise mais simples.

Palavras chave: Redes Pr/T, Árvore de Alcançabilidade, Análise de Comportamentos, Funções Simétricas, Generalização.

ABSTRACT

The Reachability Tree (R-Tree) analysis has been the most used technique for the Petri-Nets. The problem with this analysis technique is that as the system to be analysed grows, the number of possible states tends to explode. We try to solve this problem with a new method which reduces the Reachability Tree of Pr/T-Nets using their symmetries.

This method is based on only removing the analogous subtree, preserving the analysis capability of the model. The symmetry determination is carried out by using a marking generalization procedure which is also described. From that, we can reach the partial automatization of the reduced R-Tree construction in the general cases and the complete automatization in some special nets denominated *Nets with Directly Reducible Reachability Tree (DR-Nets)*.

We also show that the constructed RR-Tree allows to determine if a marking is reachable, and then recover the path to reach this marking.

In comparison with other approaches the obtained results show the decrease of the Reachability Tree and the facility of its construction and analysis.

Keyword: Pr/T-Nets, Reachability Trees, Behaviour Analysis of Nets, Symmetry Functions, Generalization.

SUMÁRIO

Agradecimentos	v
Resumo	vi
Abstract	vii
Sumário	viii
Índice de Figuras	xi
Índice de Tabelas	xii
Notações	xiii

CAPÍTULO I

Introdução	1
------------------	---

CAPÍTULO II

Introdução as Propriedades das Redes de Petri e aos Métodos de Análise e Verificação	3
---	---

2.1 Introdução	3
2.2 Propriedades das redes de Petri	3
2.2.1 Propriedades dinâmicas	4
2.2.2 Propriedades estruturais	13
2.3 Métodos de análise	17
2.3.1 Análise de alcançabilidade	18
2.3.2 Análise algébrica	18
2.4 Métodos de verificação	18
2.4.1 Verificação lógica	19
2.4.2 Verificação por abstração (observacional)	19
2.5 Conclusão	20

CAPÍTULO III

Métodos de Redução da Árvore de Alcançabilidade	21
---	----

3.1 Introdução	21
----------------------	----

3.2	Árvore e grafo de alcançabilidade: definições e problemas	21
3.2.1	Árvore de alcançabilidade	21
3.2.2	Grafo de alcançabilidade	22
3.2.3	O problema da exploração dos estados	23
3.3	Métodos de geração da árvore de alcançabilidade	24
3.3.1	Redução da árvore de alcançabilidade usando a equivalência de marcações	24
3.3.2	Árvore de alcançabilidade parametrizada	25
3.3.3	Grafo de alcançabilidade simbólico	26
3.3.4	Redução do espaço de estados através de conjuntos "Stubborn"	27
3.3.5	Geração do espaço de estados através da composição	28
3.4	Método de redução proposto	29
3.4.1	Princípios básicos usados para a redução da árvore de alcançabilidade	29
3.4.2	Definição da marcação equivalente	31
3.4.3	O uso das funções de simetria	32
3.5	Conclusão	35

CAPÍTULO IV

Determinação da Relação de Equivalência entre Duas Marcações

4.1	Introdução	36
4.2	PASSO 1 - Classificação das posições das tuplas	37
4.2.1	Posições das tuplas relacionadas por valor	37
4.2.2	Posições das tuplas com valor irrelevante	39
4.3	PASSO 2 - Classificação das marcações de acordo com as condições de equivalência	39
4.3.1	A cardinalidade da marcação	40
4.3.2	Cardinalidade do conjunto de substituições de habilitação	40
4.3.3	Funções de transformação	41
4.3.4	Três condições necessárias para a equivalência	42
4.4	PASSO 3 - Determinação das funções de transformação	42
4.4.1	Generalização de indivíduos	43
4.4.2	Composição consistente de generalizações de indivíduos	43
4.4.3	Generalização de tuplas	44

4.4.4	Composição de generalizações de tuplas	45
4.4.5	Grupos de tuplas compatíveis	45
4.4.6	Forma normal da marcação	46
4.4.7	A generalização de grupos de tuplas	46
4.4.8	A generalização de marcações	47
4.4.9	Algoritmo de generalização	48
4.5	PASSO 4 - Verificação da simetria das funções	52
4.5.1	Caso particular - Rede com Árvore de Alcançabilidade Diretamente Redutível (RDR).....	52
4.5.2	Caso geral	53
4.6	Algoritmo para determinação da equivalência de marcações no caso geral das redes Pr/T	53
4.7	Conclusão	55
 CAPÍTULO V		
	A Árvore de Alcançabilidade Reduzida (AAR).....	57
5.1	Introdução	57
5.2	O método de construção da árvore de alcançabilidade reduzida (AAR)	57
5.2.1	Tipos de nós da AAR	58
5.2.2	Algoritmo para construção da árvore de alcançabilidade reduzida	59
5.2.3	Exemplo de aplicação do algoritmo de geração da AAR	64
5.3	Equivalência entre a árvore de alcançabilidade completa (AA) e a árvore de alcançabilidade reduzida (AAR)	65
5.3.1	Determinação da alcançabilidade de uma marcação	66
5.3.2	Determinação da Seqüência de Disparos de Transições	69
5.3.3	Determinação dos Indivíduos Capturados pelas Transições	70
5.4	Comparação da AAR com a AA	70
5.5	Conclusão	73
 CAPÍTULO VI		
	Conclusões	75
	Referências bibliográficas	78

ÍNDICE DE FIGURAS

Figura 1.	Ilustração das propriedades dinâmicas das redes de Petri.....	12
Figura 2.	Ilustração das propriedades estruturais das redes de Petri.....	17
Figura 3.	Transformação de uma árvore de alcançabilidade em um grafo de alcançabilidade	22
Figura 4.	Rede modelando k-processos concorrentes que não interagem.....	23
Figura 5.	O almoço dos cinco filósofos - modelado em rede Pr/T.....	29
Figura 6.	Árvore de alcançabilidade para o almoço dos cinco filósofos - modelado em rede Pr/T	30
Figura 7.	Árvore de alcançabilidade reduzida para o almoço dos cinco filósofos.....	30
Figura 8.	Simetrias em árvore de alcançabilidade.....	33
Figura 9.	Redução da árvore de alcançabilidade usando as simetrias totais.....	33
Figura 10.	Redução da árvore de alcançabilidade usando as simetrias totais e parciais	34
Figura 11.	Determinação das posições relacionadas por valor e posições com valor irrelevante em redes Pr/T	38
Figura 12.	Exemplo de generalização de marcações.....	43
Figura 13.	AAR almoço dos cinco filósofos.....	65
Figura 14.	Grafo de alcançabilidade reduzido (GAR) do almoço dos cinco filósofos	67
Figura 15.	O almoço dos cinco filósofos - modelado em RDR	71
Figura 16.	O protocolo para a manutenção de múltiplas cópias de um banco de dados - modelado em RDR.....	71
Figura 17.	Número de marcações armazenadas em função do valor N, para $k = 3$	73

ÍNDICE DE TABELAS

Tabela 1.	Propriedades dinâmicas das redes de Petri da figura 1.....	13
Tabela 2.	Propriedades estruturais das redes de Petri da figura 2.....	16
Tabela 3.	Número de nós e arcos da AA para $k = 3$	23
Tabela 4.	Número de nós e arcos da AA para $N = 2$	24
Tabela 5.	Número marcações das AA e AAR para os exemplos clássicos	72
Tabela 6.	Número de nós e arcos da AAR para $k = 3$	72
Tabela 7.	Número de nós e arcos da AAR para $N = 2$	73

NOTAÇÕES

\mathbf{N}	. Conjunto dos números naturais $\{1, 2, 3 \dots\}$
\mathbf{Z}	. Conjunto dos números inteiros $\{\dots -3, -2, -1, 0, 1, 2, 3 \dots\}$
\mathbf{Z}^+	. Conjunto dos números inteiros positivos $\{0, 1, 2, 3 \dots\}$
$ X $. Cardinalidade do conjunto X
R	. Rede de Petri Lugar/Transição ou Predicado/Transição
$P = \{p_1, p_2 \dots p_n\}$. Conjunto de lugares da rede R onde $n = P $
$T = \{t_1, t_2 \dots t_m\}$. Conjunto de transições da rede R onde $m = T $
$\bullet p$. Conjunto das transições de entrada do lugar p
$p \bullet$. Conjunto das transições de saída do lugar p
$M(p_j)$. Fichas ou tuplas do lugar p_j (Marcação do lugar p_j)
$M = (M(p_1) \dots M(p_n))$. Marcação da rede R
M_0	. Marcação inicial da rede R
(R, M_0)	. Rede R com marcação inicial M_0
$\sigma = t_1 t_2 \dots t_k$. Seqüência de transições da rede R
$[\sigma >$. Disparo da seqüência de transições $[t_1 t_2 \dots t_k >$
$M [t >$. A transição t é habilitada por M
$M [\sigma >$. A seqüência de transições σ é habilitada por M
sh	. Substituição de habilitação da transição
$t.sh$. A transição t é habilitada pela substituição sh
$SH(t, M)$. Conjunto de substituições de habilitação da transição t na marcação M
$[M_0 >$. Conjunto de marcações alcançáveis a partir de M_0
Tp_i	. Tuplas da rede Pr/T
$p(Tp_i)$. Lugar da tupla Tp_i na rede Pr/T
$A_T(t)$. Fórmula seletora da transição t na rede Pr/T
$\langle a_1, a_2 \dots a_p \rangle$. Tupla com p indivíduos, onde a_i é a i ésima posição da tupla
$\langle V_1, V_2 \dots V_p \rangle$. Tupla com p variáveis, onde V_i é a i ésima posição da tupla
Pos_{ipj}	. Posição i das tuplas do lugar p_j
$G(p_i)$. Grupo de tuplas compatíveis que estão no lugar p_i
$NT(p_i)$. Número de tuplas contidas no lugar p_i
$f: (x \rightarrow x')$. Função de transformação de x em x'
$FS = \{fs_1 \dots fs_k\}$. Conjunto de funções de simetria fs_i
$Sub\grave{a}rvore(M_x)$. Sub\grave{a}rvore cuja raiz é M_x
$M \sim M'$. A marcação M é equivalente a M'
$x \setminus x'$. Generalização de x e x'

$x x' \circ y y'$. Composição de duas generalizações
$CompI$. Composição de indivíduos
$CompG$. Composição de grupos de tuplas
MG	. Matriz de generalização
$CompM$. Composição de generalizações de marcações
$\{X \setminus X'\}$. Conjunto de generalizações
GAR	. Grafo de alcançabilidade reduzido
$M_0 [\pi > M_k$. Caminho orientado $M_0 [t_1.sh_1 > M_1 \dots M_{k-1} [t_k.sh_k > M_k$
$Pré_{M_x} = \{M_1, \dots, M_k\}$. Conjunto de todas as marcações a partir das quais M_x é alcançável
FSV_{M_x}	. Conjunto de funções de simetria válidas para a marcação M_x
$M_{GAR} = \{M_1, \dots, M_k\}$.	. Conjunto das marcações do GAR

CAPITULO I

INTRODUÇÃO

Para sistemas de grande complexidade, tais como os Sistemas de Automação Industrial e os Sistemas Informáticos Distribuídos, é necessário o uso de *Técnicas de Descrição Formal* (TDF) que permitem a *especificação* destes sistemas, a *validação* e a posterior *implementação*. O uso desses modelos formais deve permitir que o sistema seja especificado de forma *clara, completa e sem ambigüidades*. O seu poder de expressão deve ser suficiente para descrever as características do sistema. A especificação deve ter um bom nível de abstração tornando-a independente da implementação. Finalmente é desejável que o modelo possa ser validado automaticamente [DIAZ 89, FARI 89].

A *validação* da especificação é geralmente dividida em duas categorias [DIAZ 89, BOUR 87]: a *análise das propriedades gerais* e a *verificação das propriedades específicas*.

Vários são os formalismos que tem sido utilizados para a especificação de sistemas, entre eles citamos: as *máquinas de estado finitas*, as *redes de Petri*, as *gramáticas formais*, a *abordagem algébrica CCS*, as *abordagens híbridas*, os *tipos de dados algébricos abstratos* e a *lógica temporal* [COUR 87].

Neste trabalho utilizamos o formalismo redes de Petri, proposto originalmente por Carl Adam Petri em 1962 [PETR 62]. Desde então ele tem sido amplamente utilizado. Dentre as diversas extensões ao modelo de base (Lugar/Transição) utilizamos as redes *Predicado/Transição* (Pr/T), que consistem de um dobramento dos lugares e transições através de anotações que utilizam variáveis e predicados lógicos de primeira ordem [GENR 87].

Para a análise das propriedades gerais das redes de Petri as técnicas que tem sido mais utilizadas são a *análise de alcançabilidade* e a *análise algébrica*. A *análise de alcançabilidade* baseada na construção da *árvore de alcançabilidade* (AA), tem uma aplicação geral servindo também como base de diversas técnicas de verificação. O seu principal problema está na *explosão dos estados alcançáveis* de um sistema que limita o seu emprego a modelos de sistemas de pequeno porte.

Para resolução deste problema diversas abordagens tem sido apresentadas, nas quais tem se buscado reduzir o número de estados a serem armazenados na árvore. No entanto a maioria das abordagens resulta em uma árvore de alcançabilidade sobre a qual apenas algumas propriedades podem ser analisadas diretamente.

CAPÍTULO II

INTRODUÇÃO AS PROPRIEDADES DAS REDES DE PETRI E AOS MÉTODOS DE ANÁLISE E VERIFICAÇÃO

2.1 Introdução

Neste capítulo definiremos algumas das propriedades das redes de Petri, e descreveremos sucintamente alguns métodos de análise e de verificação.

Inicialmente apresentaremos as propriedades das redes de Petri, dividindo-as em propriedades *dinâmicas* e em propriedades *estruturais*. A sua apresentação será feita usando a forma como elas geralmente são definidas na literatura citada. A maioria dessas definições são aplicadas às redes de Petri Ordinárias (Lugar/Transição), podendo ser interpretadas para o caso das redes de Petri de Predicado/Transição (Pr/T) e outras redes de Petri de Alto Nível (como por exemplo Redes Coloridas e Redes de Petri com Objetos).

Em segundo lugar serão apresentadas as duas principais técnicas de análise: *análise de alcançabilidade* e *análise algébrica*. Ambas são aplicáveis às redes Pr/T, sendo que o nosso interesse se restringirá à *análise de alcançabilidade* por permitir obter diversas destas propriedades e ser aplicável a vários tipos de rede de Petri.

Finalmente serão apresentadas duas técnicas de verificação baseadas no grafo de alcançabilidade: a *verificação lógica* e a *verificação por abstração*.

2.2 Propriedades das redes de Petri

As propriedades das redes de Petri são classicamente divididas em *dinâmicas* (ou de *comportamento*) e *estruturais* [MURA 89, COUR 87].

As *propriedades dinâmicas* são aquelas que além de dependerem da estrutura da rede, dependem também do seu estado (marcação) inicial, e estão ligadas com a evolução da rede. O estudo dessas propriedades é realizado utilizando-se tanto a *análise de alcançabilidade* como a *análise algébrica*. Entre estas propriedades existem algumas que estão ligadas

aos estados alcançáveis, e outras que tem relação também com as seqüências de disparo de transições [VAUT 86, BRAM 83]. Uma vez que o número de seqüências de disparo possíveis de um sistema é geralmente muito maior que o número de estados alcançáveis, estas são mais difíceis de serem analisadas.

As *propriedades estruturais* são aquelas que dependem somente da estrutura da rede, sendo assim independentes da marcação inicial M_0 . O seu estudo é realizado através da *análise algébrica* que trabalha com a matriz de incidência $A = O - I$, onde O é a matriz de saída e I é a matriz de entrada da rede, que são dependentes unicamente da estrutura da rede (veja em [PETE 81, BRAM 83, MURA 89]). Para redes de Petri Ordinárias os elementos da matriz são os pesos dos arcos, e no caso das redes de Petri de Alto Nível os elementos da matriz são funções [JENS 81] ou variáveis [GENR 87].

A análise das propriedades dinâmicas e estruturais, quando aplicadas ao modelos de sistemas, permitem provar duas classes de propriedades relativas ao funcionamento do sistema [DIAZ 82, COUR 87]:

- as *propriedades gerais* que a princípio devem ser satisfeitas para "todos" os sistemas (limitação, vivacidade, reversibilidade, etc.);
- as *propriedades específicas* que dependem das características funcionais do sistema e do significado de cada lugar e transição da rede. Estas propriedades são muitas vezes expressas através de invariantes.

A seguir apresentaremos as propriedades dinâmicas e estruturais das redes de Petri.

2.2.1 Propriedades dinâmicas

Nesta seção apresentaremos as propriedades dinâmicas das redes de Petri. Para ilustrar o uso das definições das propriedades utilizaremos nesta seção as redes apresentadas na Figura 1.

As principais notações a serem utilizadas para a definição das propriedades estão reunidas em um glossário apresentado no início deste trabalho.

a) Alcançabilidade

A propriedade da *alcançabilidade* ("reachability") é a base fundamental para estudar as outras propriedades dinâmicas de qualquer sistema.

Definição 2.1 A marcação M é alcançável se $\exists \sigma \mid M_0[\sigma > M$ [BRAM 83, MURA 89]

Exemplo 2.1 Na rede (e) da Figura 1 podemos verificar que a marcação $M = (1, 1, 1, 1, 1)$ é alcançável de $M_0 = (2, 0, 0, 0, 0)$ através do disparo da seqüência de transições $[\sigma > = [t_1 t_2 t_1 t_2 t_1 t_3 >$

O problema da determinação da alcançabilidade de uma marcação $M \in (R, M_0)$ é decidível mesmo para redes *ilimitadas*, embora a sua verificação seja pelo menos de complexidade exponencial em espaço (e tempo) [MURA 89].

b) Limitação

A *limitação* ("boundness") da rede (R, M_0) é uma propriedade que garante a finitude do sistema físico que esta sendo modelado, sendo por isso uma condição necessária para que um sistema possa ser implementado [ESTE 85]. A limitação também é condição necessária para a utilização da técnica de *análise de alcançabilidade*.

Definição 2.2 Seja $k \in \mathbf{N}$. Um lugar p de uma rede (R, M_0) é *k-limitado* se:
 $\forall M \in [M_0 > \Rightarrow M(p) \leq k.$

Definição 2.3 Uma rede (R, M_0) é *k-limitada* se:
 $\forall p \in P, \text{ o lugar } p \text{ é } k\text{-limitado}$ [PETE 81, BRAM 83].

Definição 2.4 Uma rede (ou lugar) que é *1-limitada* é chamada de *binária* ("safe").

Exemplo 2.2 Na rede (e) da Figura 1 os lugares p_1 e p_2 são *k-limitados*, p_4 e p_5 são *binários* e p_3 é *ilimitado*.

c) Conservabilidade

Uma rede é dita *conservativa* se nenhuma ficha é criada ou destruída durante a evolução da rede [PETE 81]. Para sistemas nos quais as fichas representam recursos, a propriedade de conservação é muito importante.

Definição 2.5 Uma rede (R, M_0) é *estritamente conservativa* se:
 $\forall M \in [M_0 > \Rightarrow \sum_{i=1 \dots n} M(p_i) = \text{cte.}$

Definição 2.6 Uma rede (R, M_0) é *conservativa* se:
 $\exists z = (z_1, z_2 \dots z_n)$ onde $z_i \in \mathbf{N} \mid \forall M \in [M_0 > \Rightarrow \sum_{i=1 \dots n} z_i * M(p_i) = \text{cte.}$

Exemplo 2.3 Na rede (c) da Figura 1, a soma $\sum_{i=1 \dots 4} z_i * M(p_i)$ é igual a 2 para $z = (2, 1, 1, 1)$. Logo, podemos concluir que essa rede é *conservativa*.

d) Vivacidade

O conceito de vivacidade ("liveness") é fortemente ligado a ausência de *marcações poço* ou *bloqueios mortais* ("deadlocks") no sistema. A vivacidade é uma propriedade importante para muitos sistemas e está relacionada com o comportamento futuro dos caminhos da árvore de marcações alcançáveis [LAM 84]. Muitas vezes a sua verificação não é possível na prática, devido ao alto custo em espaço (e/ou tempo) para a sua determinação.

Existem vários conceitos de vivacidade, que podem ser classificados em níveis [MURA 89, PETE 81]:

Definição 2.7 Uma transição t de uma rede (R, M_0) é dita:

- 1) *Viva-N0* (*morta*) se t nunca pode ser disparada, i.e., $\neg \exists M \in [M_0 > \mid M [t >$.
- 2) *Viva-N1* (*quase-viva* [BRAM 83] ou *potencialmente disparável*) se t pode ser disparada pelo menos uma vez, i.e., $\exists M' \in [M_0 > \mid M' [t >$.
- 3) *Viva-N2* se $\forall k \in \mathbf{N}$, $\exists \sigma \mid t$ pode ser disparada pelo menos k vezes na sequência de disparos σ .
- 4) *Viva-N3* se t pode ser disparada indefinidamente somente em situações particulares de evolução da rede, i.e., $\exists \sigma$ infinita $\mid t$ pode ser disparada infinitas vezes em σ .
- 5) *Viva-N4* (*viva*) se t pode ser disparada indefinidamente qualquer que seja a evolução da rede, i.e., $\forall M \in [M_0 >$, $\exists \sigma$ e $\exists M' \mid M [\sigma > M'$ e $M' [t >$. Uma outra forma de expressar a vivacidade de uma transição em nível *N4* é: t é *viva* se $\forall M \in [M_0 >$, t é *quase-viva* na rede (R, M) [BRAM 83].

Proposição 2.8 Se uma transição t é *viva-Nk*, então ela é *viva-N(k-1)* para $k = 2, 3, 4$.

Exemplo 2.4 Na rede (f) da Figura 1, a transição t_4 é *viva-N0* (*morta*), t_1 é *viva-N1* (*quase-viva*), t_3 é *viva-N3* pois pode ser disparada indefinidamente, t_2 é *viva-N2* pois pode ser disparada k vezes, onde k é o número de vezes que t_3 é disparada. Na rede (d) dessa mesma figura as transições t_3 e t_5 são *vivas-N2*, t_1 e t_2 são *vivas-N3* e t_4 é *viva-N4* (*viva*).

Geralmente uma transição que não possui pelo menos a *vivacidade-N1* é inútil para o funcionamento do sistema modelado. A exceção ocorre quando a transição modela um erro. Este tipo de transição é chamada de *fato* ("fact"), e obviamente não deve nunca ser habilitada (ver o uso de fatos em [GENR 87]).

A vivacidade de uma rede é determinada a partir da vivacidade das suas transições. Os vários níveis de vivacidade também podem ser definidas para a rede conforme enunciado abaixo:

Definição 2.9 Uma rede de Petri (R, M_0) é dita *viva-Nk*, se:

$$\forall t \in T, t \text{ é viva-N}k \text{ (} k = 0, 1, 2, 3, 4 \text{)}.$$

A vivacidade de nível N4 de uma rede de Petri também pode ser definida através de duas formas equivalentes, onde cada uma enfoca um aspecto diferente da vivacidade da rede.

Definição 2.10 Uma rede (R, M_0) é dita *viva* se qualquer uma das seguintes situações ocorrer:

- 1) $\forall t \in T, t$ é viva [BRAM 83, MURA 89].
- 2) M_0 é viva [VALK 87] (ver a seguir a definição 2.12).

e) Vivacidade de marcação

A vivacidade de uma marcação é uma outra forma de determinação da vivacidade da rede.

Definição 2.11 A marcação M é *viva* se:

$$\forall t \in T \text{ e } M' \in [M >, \exists M'' \in [M' > \mid M'' [t > \text{ [VALK 87].}$$

Definição 2.12 Uma marcação M é chamada de *marcação poço (morta)* se:

$$\neg \exists t \in T \mid M [t > \text{ [BRAM 83].}$$

A existência de marcações poço nem sempre está associada a erro, pois pode representar um situação desejada, como por exemplo no caso da redes com terminação.

Exemplo 2.5 Na rede (a) da Figura 1 considere a marcação M' tal que $M_0 [t_2 t_3 t_4 > M'$; esta marcação M' é uma *marcação poço*.

f) Pseudo-vivacidade

Uma rede R é *pseudo-viva* quando ela está em permanente funcionamento [BEST 87, SOUS 90]. Esta propriedade não leva em conta o funcionamento individualizado de cada transição. O conceito de pseudo-vivacidade é mais fraco do que o conceito de *vivacidade-N4*, e também não se enquadra em nenhum dos outros níveis de vivacidade definidos anteriormente.

Definição 2.13 Uma rede (R, M_0) é *pseudo-viva* se:

$$\forall M \in [M_0 > \exists t \in T \mid M [t > \text{ (i.e. não existem marcações poço na rede).}$$

Um subconjunto de marcações da rede (R, M_0) formam um *laço "livelock"* se após uma certa marcação M da rede, ela evolui indefinidamente e somente as marcações desse subconjunto podem ser alcançadas.

Exemplo 2.6 Na rede (d) da Figura 1, considere a marcação M' tal que $M_0 [t_1 t_1 > M'$; esta marcação M' é um *laço* no qual apenas t_4 esta sempre habilitada.

g) Terminação

Em muitos sistemas, o seu bom funcionamento pode ser caracterizado pela sua capacidade de parar em estados (marcações) terminais bem definidos.

Definição 2.14 Uma rede (R, M_0) *termina* se:

$$\forall M \in [M_0 >, \exists \sigma \text{ finita e } \exists M' \mid M [\sigma > M', \text{ onde } M' \text{ é marcação poço [DIMI 90].}$$

Definição 2.15 Uma rede com um lugar de entrada p_e e lugar de saída p_s é dita *normalizada*, se somente o lugar p_e é marcado na marcação inicial M_0 e somente o lugar de saída é marcado na marcação final M_f . Esta rede (R, M_0) é chamada de *rede de terminação própria* se $\forall M \in [M_0 >, M_f \in [M >$ [BRAM 83, PETE 81].

h) Reiniciação e estados de recepção

A *reiniciação* (ou *reversibilidade*) de uma rede garante a existência de uma seqüência de disparos que pode levar o sistema de qualquer marcação alcançável de volta para a marcação inicial.

Definição 2.16 Uma rede (R, M_0) é *reiniciável* (ou *reversível*) se:

$$\forall M \in [M_0 >, \exists \sigma \mid M [\sigma > M_0 \text{ [MURA 89]}$$

A vivacidade de uma rede reiniciável é facilmente determinável.

Teorema 2.17 Se uma rede (R, M_0) é *reiniciável* e *quase-viva* então ela é *viva* [ESTE 86].

Uma rede também pode possuir estados aos quais ela sempre retorna, sendo alcançáveis qualquer que seja a evolução da rede. Estes estados são chamados de *estados de recepção* ("home-states") [BRAM 83].

Definição 2.18 Uma marcação M_r de uma rede (R, M_0) é chamada de *estado de recepção* se:

$$\forall M \in [M_0 >, M_r \in [M > \text{ [BEST 87, MURA 89].}$$

Teorema 2.19 Se a marcação inicial M_0 da rede é um *estado de recepção* então a rede (R, M_0) é *reiniciável* [BEST 87].

Exemplo 2.7 Na rede (d) da Figura 1 considere a marcação M' tal que $M_0 [t_1 t_2 > M'$; esta marcação M' é um *estado de recepção*. Nas redes (b) e (c) as marcações M_0 também são *estados de recepção*, logo, essas redes são *reiniciáveis*.

i) Cobertura

Definição 2.20 Seja (R, M_0) uma rede de Petri, e sejam duas marcações M e $M' \in [M_0>$. Então M é *coberta* por M' sse [LIND 89]:

1) $M(p_i) = M'(p_i) \forall p_i \in P$ (*interpretação estrita*) ou

2) $M(p_i) \geq M'(p_i) \forall p_i \in P$ (*interpretação fraca*).

Definição 2.21 Uma marcação M é dita *coberta*¹ se:

$\exists M' \in [M_0> \mid \forall p_i \in P, M(p_i) \leq M'(p_i) \text{ e } M \neq M'$ [BRAM 83, MURA 89].

A cobertura na sua interpretação fraca pode ser utilizada na verificação de outras propriedades como mostram os teoremas abaixo:

Teorema 2.22 Se M é a mínima marcação necessária para habilitar uma transição t , então t é *morta* sse M não é coberta (t é *quase-viva* se M é coberta) [MURA 89].

Teorema 2.23 Uma rede é *ilimitada* se $\exists M \mid M'$ é coberta por M e $M \in [M'>$ [ESTE 86].

Exemplo 2.8 Na rede (a) da Figura 1 considere a marcação M' tal que $M_0 [t_2 t_3 t_4> M'$; esta marcação M' é coberta por M_0 . Considere a marcação M'' da rede (e) tal que $M_0 [t_1 t_2> M''$. A marcação M_0 é coberta por M'' , logo, a rede é *ilimitada*.

j) Repetição

Uma seqüência de disparos de transições é *repetitiva* se existe a possibilidade de ela estar indefinidamente ativa, indicando a possibilidade de evolução permanente da rede.

Definição 2.24 A seqüência de disparos de transições σ é *repetitiva* se:

$\exists M \text{ e } \exists M' \mid M [\sigma> M' \text{ e } M' \geq M$ [BRAM 83].

Corolário 2.25 Se σ é repetitiva, $M [\sigma> M'$ e $M' > M$, então a rede é *ilimitada* [BRAM 83].

Exemplo 2.9 Na rede (a) da Figura 1 a seqüência $M_0 [t_2 t_1> M'$ é *repetitiva* e $M_0 = M'$. Na rede (e) a seqüência $M_0 [t_1 t_2> M''$ é *repetitiva* e $M'' > M_0$, logo a rede é *ilimitada*.

k) Comutatividade

Definição 2.26 Duas seqüências de disparo σ' e σ'' constituídas do mesmo número de disparos de cada transição são ditas *comutativas* se:

$\forall M \text{ e } \forall M', \exists \sigma' \mid M [\sigma'> M' \text{ sse } \exists \sigma'' \mid M [\sigma''> M'$ [BRAM 83].

Exemplo 2.10 Na rede (b) da Figura 1 as seqüências $[t_2 t_3>$ e $[t_3 t_2>$ *comutam*.

¹ Aqui é utilizada a *interpretação fraca* de cobertura.

l) Concorrência e conflito

Definição 2.27 Duas transições t e t' são ditas *habilitadas concorrentemente*, se elas estão habilitadas na marcação M e o disparo de uma delas não desabilita a outra [BEST 87].

Corolário 2.28 Se duas transições t e t' são *habilitadas concorrentemente* pela marcação M , então $M [t t' > M'$ e $M [t' t > M'$ [BEST 87].

Definição 2.29 Se duas transições t e t' estão habilitadas em uma marcação M mas não estão habilitadas concorrentemente, então elas estão em *conflito* [BEST 87].

Exemplo 2.11 Na rede (a) da Figura 1 as transições t_1 e t_3 estão em *conflito*. Na rede (b) da mesma figura as transições t_2 e t_3 estão *habilitadas concorrentemente* na marcação M' (onde $M_0 [t_1 > M'$).

m) Persistência e confluência

Uma rede é dita persistente se para quaisquer duas transições t e t' habilitadas em uma marcação M , o disparo de uma transição não desabilita a outra.

Definição 2.30 Uma rede (R, M_0) é *persistente* se $\forall M \in [M_0 >$ e $\forall t$ e $\forall t' \in T \mid M [t >$ e $M [t' >$ então t e t' são *habilitadas concorrentemente* [BRAM 83].

Corolário 2.31 Se em uma rede (R, M_0) , $\forall M \in [M_0 >$, as transições não estão em *conflito* então a rede é *persistente*.

Corolário 2.32 Se a rede (R, M_0) é persistente e existem duas seqüências de disparo σ_1 e σ_2 que estão habilitadas pela marcação $M \in [M_0 >$, então também existem duas seqüências de disparo σ_1' e σ_2' que *confluem* para a mesma marcação M' , i.e. $M [\sigma_1 \sigma_1' > M'$ e $M [\sigma_2 \sigma_2' > M'$ [BRAM 83].

n) Justiça

Na literatura existem propostos diversos conceitos de *justiça* ("fairness"). Apresentaremos aqui os dois mais comumente usados: *justiça incondicional (global)* e *L-justiça*.

Exemplo 2.12 A noção de justiça pode ser ilustrada no exemplo do almoço dos filósofos (ver Figura 5) quando dois filósofos f_1 e f_3 agem injustamente contra um terceiro f_2 . Isto ocorre sempre que eles depositam os palitos de tal modo que o terceiro nunca tenha simultaneamente na mesa os dois palitos que necessita para comer [VALK 87].

Definição 2.33 Uma seqüência de disparo de transições σ é *incondicionalmente justa* se σ é finita, ou então $\forall t \in T$, t é disparado infinitas vezes em σ infinita [VALK 87, MURA 89].

Definição 2.34 Uma rede é *incondicionalmente justa* se toda seqüência de disparo de transições σ a partir de M_0 é *incondicionalmente justa* [MURA 89, TUOM 86].

Definição 2.35 Duas transições t e t' estão em uma relação de *L-Justiça* (Justiça-Limitada), se o número máximo de vezes que cada transição pode ser disparada sem que a outra seja disparada é limitado [MURA 89].

Definição 2.36 Uma rede (R, M_0) é *L-justa* se todos os pares de transições da rede estão em uma relação de *L-Justiça* [MURA 89].

Exemplo 2.13 Na rede (e) da Figura 1 as transições t_1 e t_2 estão em uma relação de *L-justiça* pois t_1 só pode ser disparada 2 vezes sem que t_2 seja disparada. As transições t_2 e t_3 não estão em uma relação de *L-justiça* pois t_2 pode ser disparada infinitas vezes sem que t_3 seja disparada, de onde podemos concluir que a rede não é *L-justa*.

o) Distância síncrona

A distância síncrona é uma medida relacionada com o grau de dependência mútua entre duas transições da rede.

Definição 2.37 Sejam t_1 e t_2 duas transições. A *distância síncrona* entre elas é dada por $d_{12} = \text{Max}_{\sigma^2} |\text{nr}(\sigma, t_1) - \text{nr}(\sigma, t_2)|$, onde σ é uma seqüência de disparos começando em qualquer marcação $M \in [M_0>$, e $\text{nr}(\sigma, t_i)$ é o número de vezes que a transição t_i é disparada em σ .

Exemplo 2.14 Na rede (a) da Figura 1 o valor de $d_{12} = 1$ e $d_{13} = \infty$. Isto é, cada disparo da transição t_1 é seguido de um disparo da transição t_2 e a transição t_1 pode ser disparada infinitas vezes sem que t_3 seja disparada.

A tabela 1 resume a análise das propriedades dinâmicas das redes apresentadas, onde o símbolo "+" indica que a rede possui a propriedade, e "-" indica a ausência da propriedade.

² Max_{σ} é o valor máximo obtido de todas as seqüências de disparo σ .

Tabela 1. Propriedades dinâmicas das redes de Petri da figura 1

Propriedades dinâmicas	(a)	(b)	(c)	(d)	(e)	(f)
Limitada	+	+	+	+	-	-
Estritamente Conservativa	-	+	-	+	-	-
Conservativa	-	+	+	+	-	-
Terminação	+	-	-	-	-	+
Reiniciação	-	+	+	-	+	-
Viva-N1 (quase viva)	+	+	+	+	+	-
Viva-N2	-	+	+	+	+	-
Viva-N3	-	+	+	+	+	-
Viva-N4 (viva)	-	+	+	-	+	-
Pseudo-viva	-	+	+	+	+	-
Persistência	-	+	-	-	+	-
L-justa	-	+	-	-	-	-
Justa	-	+	-	-	-	-

2.2.2 Propriedades estruturais

A influência da estrutura da rede no seu comportamento é de particular importância uma vez que esta permanece a mesma, qualquer que seja a marcação inicial. A estrutura pode ser analisada independentemente da dinâmica da rede.

A sua análise é baseada na álgebra linear aplicada ao estudo da *equação de estado* $M = M_0 + A^T * v$, onde A é a matriz de incidência da rede de ordem $m * n$ (com m o número de transições e n é o número de lugares da rede), v é o vetor característico da seqüência de disparo de ordem $m * 1$ e M é o vetor marcação de ordem $n * 1$. Para o estudo das propriedades estruturais consideraremos apenas as redes que são puras³.

Para ilustrar as propriedades estruturais utilizaremos as redes apresentadas na Figura 2.

a) Invariante de lugar

Definição 2.38 Um vetor de números inteiros Il é chamado de *invariante de lugar* ("R-Invariant") se $A * Il = 0$ [MURA 89]

Corolário 2.39 Toda combinação linear de um invariante de lugar é também um invariante de lugar [LAUT 87].

Teorema 2.40 Se o vetor Il é um *invariante de lugar* então

$$\forall M_0 \text{ e } \forall M \in [M_0 >, M^T * Il = M_0^T * Il \text{ [LAUT 87, MURA 89].}$$

³ Uma rede é *pura* se não existe nenhum lugar que seja um lugar de entrada e de saída de uma transição qualquer da rede.

Notação 2.41 O conjunto de lugares que participam de um invariante de lugar é denominado de *componente conservativa* da rede [BRAM 83].

b) Invariante de transição

Definição 2.42 Um vetor de números inteiros It é chamado de *invariante de transição* ("T-Invariant") se $A \cdot T^* It = 0$ [MURA 89]

Corolário 2.43 Toda combinação linear de um invariante de transição é também um invariante de transição [LAUT 87].

Teorema 2.44 Seja o vetor It um *invariante de transição*, então $\exists M_0$ e $\exists \sigma \mid M_0 \{\sigma\} > M_0$ [LAUT 87].

Notação 2.45 O conjunto das transições que participam de um invariante de transição é denominado de *componente repetitiva* da rede [BRAM 83].

c) Sifão e armadilha

Definição 2.46 Um conjunto de lugares $P' \neq \emptyset$ de uma rede Ordinária é denominado de *sifão* ("siphon") se cada transição que tem um lugar de saída em P' tem também um lugar de entrada em P' , i.e. $\bullet P' \subseteq P' \bullet$ [MURA 89].

Teorema 2.47 Se um *sifão* ficar sem fichas em uma determinada marcação M , então $\forall M' \in [M >$ a marcação M' permanecerá sem ficha [MURA 89, ALAI 85].

Essa característica dos *sifões* permite em algumas subclasses e extensões da rede de Petri Ordinária mostrar que a existência de um *sifão* pode levar a um eventual bloqueio total (*marcação poço*) ou parcial (*laço*) de uma rede.

Definição 2.48 Um conjunto de lugares $P'' \neq \emptyset$ de uma rede Ordinária é denominado de *armadilha* ("trap") se cada transição que tem um lugar de entrada em P'' tem também um lugar de saída em P'' , i.e. $P'' \bullet \subseteq \bullet P''$ [MURA 89].

Teorema 2.49 Se uma *armadilha* ficar com alguma ficha em uma determinada marcação M , então $\forall M' \in [M >$ a marcação M' permanecerá com a ficha [MURA 89, ALAI 85].

Exemplo 2.15 Na rede (f) da Figura 2 para $P' = \{p_2, p_3\}$ temos que $\bullet P' = \{t_2\}$ e $P' \bullet = \{t_1, t_2\}$, logo $\bullet P' \subseteq P' \bullet$ e P' é um *sifão*. Na rede (e) da mesma figura para $P'' = \{p_2, p_3\}$ temos que $\bullet P'' = \{t_1, t_2\}$ e $P'' \bullet = \{t_2\}$, logo $P'' \bullet \subseteq \bullet P''$ e P'' é uma *armadilha*.

d) **Vivacidade estrutural**

Definição 2.50 A rede R é *estruturalmente viva* se $\exists M_0 \mid (R, M_0)$ é viva [MURA 89].

A vivacidade estrutural é uma propriedade para a qual não existe ainda uma completa caracterização no caso geral das redes de Petri.

e) **Controlabilidade**

Definição 2.51 A rede R é *completamente controlável* se $\forall M$ e $\forall M' \mid M \leq M'$ [MURA 89].

Teorema 2.52 Se uma rede R com n lugares é *completamente controlável* então $\text{Rank}^4(A) = n$ [MURA 89].

f) **Limitação estrutural**

Definição 2.53 Uma rede R é *estruturalmente limitada* se $\forall M_0, (R, M_0)$ é limitada [MURA 89].

Teorema 2.54 Uma rede R é *estruturalmente limitada* sse $\exists v = (v_1, v_2 \dots v_m)^T$ onde $v_i \in \mathbf{Z}^+ \mid A * v \leq 0$ [MURA 89].

g) **Conservabilidade estrutural**

Uma rede R é *estruturalmente conservativa* se a soma balanceada das fichas é constante para todas marcações pertencentes ao conjunto de marcações alcançáveis a partir de M_0 , qualquer que seja a marcação inicial M_0 . Esta propriedade é um caso especial da limitação estrutural.

Definição 2.55 A rede R é *estruturalmente conservativa* se $\forall M_0, (R, M_0)$ é conservativa.

Teorema 2.56 Uma rede R é *estruturalmente conservativa* sse $\exists v = (v_1, v_2 \dots v_m)^T$ onde $v_i \in \mathbf{Z}^+$ e $\exists v_j \neq 0 \mid A * v = 0$ [MURA 89].

h) **Repetitividade**

Definição 2.57 Uma rede R é *repetitiva* se existe uma marcação M_0 e uma seqüência de disparo de transições σ a partir de M_0 , tal que toda transição ocorre infinitas vezes em σ [MURA 89].

Teorema 2.58 Uma rede R é *repetitiva* sse $\exists w = (w_1, w_2 \dots w_n)^T$ onde $w_i \in \mathbf{Z}^+$ e $\exists w_j \neq 0 \mid A^T * w \geq 0$.

⁴ O Rank de uma matriz é dado pelo número de colunas ou linhas linearmente independentes.

i) **Consistência**

Definição 2.59 Uma rede R é *consistente* se existe uma marcação M_0 e uma seqüência de disparos σ da marcação M_0 para M_0 , tal que toda transição ocorre pelo menos uma vez em σ [MURA 89].

Esta propriedade é um caso especial de repetitividade.

Teorema 2.60 Uma rede R é *consistente* sse $\exists w = (w_1, w_2 \dots w_n)^T$ onde $w_i \in \mathbf{Z}^+$ e $\exists w_j \neq 0 \mid A^T * w = 0$.

j) **Justiça estrutural**

Definição 2.61 Uma rede R é *estruturalmente L-justa* se ela é *L-justa* para qualquer que seja a marcação inicial M_0 .

k) **Finitude e conexão forte**

A rede de Petri por ser um grafo orientado bipartido, formado pelos conjuntos P (*lugares*) e T (*transições*), também pode ser estudado segundo as propriedades dos grafos.

Definição 2.62 Uma rede R é *finita* se o número de lugares (n) e o número de transições (m) da rede é finito [BEST 87].

Definição 2.63 Uma rede R é *fortemente conexa* sse:

$$\forall x \text{ e } \forall y \in T \cup P, \text{ existe um caminho orientado de } x \text{ para } y \text{ [BEST 87].}$$

A tabela 2 resume as propriedades estruturais das redes apresentadas, onde o símbolo "+" indica que a rede possui a propriedade, e "-" indica a ausência da propriedade.

Tabela 2. Propriedades estruturais das redes de Petri da figura 2

Propriedades estruturais	(a)	(b)	(c)	(d)	(e)	(f)
Controlabilidade	-	-	+	-	-	-
Limitação estrutural	+	-	-	+	-	+
Conservabilidade estrutural	-	-	-	+	-	-
Repetitividade	-	+	+	+	+	-
Consistência	-	-	+	+	-	-
Conexão forte	-	-	-	+	-	-
Vivacidade estrutural	-	+	+	+	-	-
L-justiça estrutural	+	+	-	+	+	+

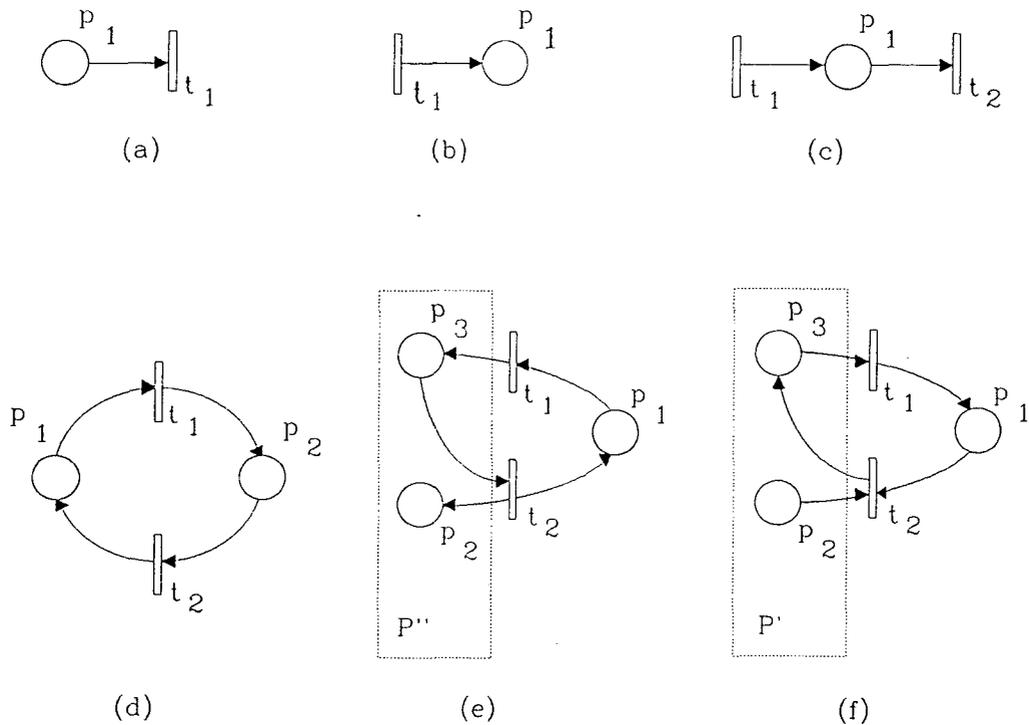


Figura 2 Ilustração das propriedades estruturais das redes de Petri

2.3 Métodos de análise

Os métodos de análise mais usados para o caso das redes de Petri são o de *análise de alcançabilidade* (enumeração dos estados alcançáveis) e o de *análise algébrica* (procura dos invariantes).

O primeiro método consiste em enumerar todas as marcações alcançáveis de uma rede. Este é um método que pode ser aplicado a qualquer classe de rede de Petri, mas o seu uso tem sido dificultado devido a *explosão combinatória de estados* que ocorre com o crescimento da rede a ser analisada.

Os métodos de análise algébrica por outro lado, apesar de muito poderosos, tem sua aplicação limitada a algumas classes de rede de Petri, ou então em algumas situações especiais.

Nesta seção descreveremos brevemente cada um destes métodos, buscando mostrar as vantagens e desvantagens de cada um deles, permitindo assim ao final do capítulo realizar uma escolha de um destes métodos.

2.3.1 Análise de alcançabilidade

Na análise de alcançabilidade as marcações alcançáveis são organizadas em uma estrutura de árvore, onde cada nó tem associado uma marcação e cada arco uma transição que realiza a transformação da marcação do nó origem na marcação do nó destino. Esta árvore contém todas as marcações alcançáveis e seqüências de disparo de transições possíveis.

No caso de a rede ser ilimitada, a árvore de alcançabilidade cresce indefinidamente. Para que o seu tamanho seja mantido finito utiliza-se o número ω que possui as seguintes propriedades: $\omega > k$, $\omega \pm k = \omega$, $\omega \geq \omega$, onde $k \in \mathbf{Z}^+$. Quando uma marcação M é maior que uma marcação M' que a precede (i.e., $M' \leq_2 M$ e $M \geq M'$), então em cada lugar $p \mid M(p) > M'(p)$, $M(p)$ é substituído por ω . A árvore assim construída é denominada de *árvore de cobertura* e permite provar as seguintes propriedades: limitação, quase-vivacidade e a possibilidade de alcançabilidade (i.e., se uma marcação M é alcançável de M_0 então existe um nó M' tal que $M' \geq M$). No caso geral a alcançabilidade e a vivacidade não podem ser provadas através do uso da *árvore de cobertura*.

No nosso trabalho lidaremos apenas com redes limitadas, e por isso utilizaremos a *árvore de alcançabilidade*. Pela inspeção dessa árvore é possível verificar as *propriedades dinâmicas* e algumas *propriedades específicas* (veja a seguir na seção 2.4).

2.3.2 Análise algébrica

Esta técnica utiliza equações matriciais, e permite observar de forma simples e rápida o comportamento das redes de Petri mesmo quando em *redes ilimitadas*, para uma dada marcação inicial. No entanto o seu uso não é geral pois nem sempre existe uma solução para as equações, devido aos indeterminismos das redes de Petri e as restrições de soluções com números inteiros não negativos.

A partir da análise da equação de estado pode-se determinar muitas das propriedades estruturais da rede, e também em certas situações provar a existência de algumas propriedades dinâmicas tais como: quase-vivacidade [JENS 87], distância síncrona [MURA 89], estados de recepção e reiniciação [MENN 87].

2.4 Métodos de verificação

A análise das propriedades gerais é um passo importante da validação de um modelo de sistema no que diz respeito ao seu funcionamento. No entanto essa análise não per-

mite concluir a respeito da correção do sistema, i.e., não permite saber se o sistema realiza a tarefa para a qual ele foi projetado.

Propriedades específicas são definidas para cada sistema e refletem o objetivo para o qual ele foi projetado. Um grande número de técnicas tem sido descrito na literatura para a verificação das propriedades específicas. Neste trabalho citaremos apenas aquelas que se baseiam na árvore de alcançabilidade do sistema. Estas técnicas são divididas em *verificação lógica* e *verificação por abstração* [VERN 89, AZEM 89].

2.4.1 Verificação lógica

Na *verificação lógica* de um sistema, as propriedades que o modelo deve possuir são expressas através do uso de formalismos lógicos que possuam uma forma conhecida de prova da veracidade das *fórmulas* ou asserções.

Um dos formalismos lógicos muito utilizado é o de lógica temporal, na qual o ordenamento temporal dos eventos é descrito através de predicados (*é possível, após, etc*) e de operadores (*sempre, inevitavelmente, etc*). O resultado da descrição das propriedades do sistema são fórmulas, cuja veracidade é verificada através de algoritmos gerais de prova. Uma condição necessária para este tipo de verificação é a de que o grafo de estados alcançáveis seja finito [QUEI 82, CLAR 83].

Neste tipo de verificação existem dois passos básicos [AZEM 89]:

- . Os requisitos do sistema são expressos em termos de fórmulas de lógica temporal.
- . A validade das fórmulas é conferida sobre o grafo de estados alcançáveis do sistema.

O uso das fórmulas de lógica temporal, torna esta verificação *segura, e sem ambiguidades*. No entanto elas tem a desvantagem de não serem explicativas [VERN 89].

2.4.2 Verificação por abstração (observacional)

Este tipo de verificação se baseia numa representação observacional do comportamento do sistema. O grafo de estados alcançáveis do sistema é reduzido a partir da divisão dos eventos do sistema em *observáveis* (visíveis) e *não observáveis* (invisíveis). Por exemplo, na modelagem de protocolos a seleção dos eventos observáveis pode corresponder a visão abstrata do serviço que o protocolo deve oferecer, considerando-se os demais eventos como invisíveis.

A verificação por abstração é geralmente realizada da seguinte forma [AZEM 89]:

- . São escolhidos os critérios de observação, ou seja é feita a definição dos eventos observáveis e não observáveis.
- . São calculadas as classes equivalentes (dois estados são equivalentes sse eles não podem ser distinguidos por nenhum evento observável)
- . O grafo de estados é reduzido, resultando em um modelo abstrato do sistema.
- . O modelo abstrato é analisado.

Uma das grandes vantagens dessa técnica é permitir a extração de um *modelo explicativo* onde os detalhes insignificantes são omitidos. A escolha de diferentes critérios de observação permite a visualização de diferentes partes do sistema, validando certos aspectos do modelo e sinalizando os erros [VERN 89].

2.5 Conclusão

Neste capítulo definimos as propriedades dinâmicas e estruturais das redes de Petri, apresentando também alguns teoremas que permitem determinar de forma simples estas propriedades. Dentre as propriedades dinâmicas apresentadas, as mais importantes são a da alcançabilidade, limitação, vivacidade e terminação.

Apresentamos também vários métodos de análise e verificação, e escolhemos a análise de alcançabilidade como método de base pelos seguintes motivos:

- 1) Ser um método de uso geral, isto é para qualquer tipo de rede de Petri;
- 2) Possibilitar a análise das suas propriedades dinâmicas;
- 3) Servir de base para muitos métodos de verificação das propriedades específicas do sistema modelado.

O método de análise de alcançabilidade é baseado na construção de uma árvore de alcançabilidade.

Nos próximos capítulos descreveremos um método que permite construir a árvore de forma reduzida, preservando a sua capacidade de análise de propriedades dinâmicas e verificação de propriedades específicas.

CAPÍTULO III

MÉTODOS DE REDUÇÃO DA ÁRVORE DE ALCANÇABILIDADE

3.1 Introdução

Este capítulo visa introduzir alguns conceitos que serão utilizados no método de redução da árvore de alcançabilidade (AA) descrito neste trabalho.

Serão também descritos alguns dos métodos propostos na literatura para resolver o problema da explosão combinatória dos estados alcançáveis, entre os quais destacamos o método proposto em [HUBE 86], no qual, a árvore de alcançabilidade é reduzida, preservando-se a capacidade de análise, conforme o objetivo visado neste trabalho.

Finalmente serão descritos os princípios básicos utilizados no método de redução da árvore de alcançabilidade proposto neste trabalho, dando destaque à determinação da equivalência entre marcações e ao uso das simetrias parciais.

3.2 Árvore e grafo de alcançabilidade: definições e problemas

Neste seção definiremos formalmente a árvore de alcançabilidade, e mostraremos como o problema da explosão combinatória dos estados dificulta (e as vezes impossibilita) a sua construção de forma completa.

3.2.1 Árvore de alcançabilidade

A árvore de alcançabilidade é um grafo orientado acíclico, no qual cada vértice é uma marcação e cada arco representa o disparo da transição que leva de uma marcação a outra. A marcação inicial é a raiz da árvore de alcançabilidade.

Para manter finito o tamanho da árvore de alcançabilidade, é utilizada uma relação de cobertura (conforme Definição 2.20), evitando assim a construção das subárvores que são cobertas por outras.

Neste trabalho utilizaremos a relação de *cobertura* na sua interpretação estrita, e para tanto trabalharemos apenas com redes que possuam árvores de alcançabilidade finita.

Definição 3.1 Seja (R, M_0) uma rede Pr/T, e sh uma substituição de habilitação da transição t . Então a sua árvore de alcançabilidade completa será denotada por $AA = (Nós, Arcos)$ onde:

- 1) A marcação inicial $M_0 \in Nós$.
- 2) $\forall M \in Nós \exists t.sh \mid M [t.sh > M'$:
 - a) Se $\neg \exists M_x \in Nós \mid M' = M_x$, então $M' \in Nós$ e o rótulo $(M, t.sh, M') \in Arcos$.
 - b) Se $\exists M_x \in Nós \mid M' = M_x$, então $nó_x \in Nós$, o rótulo $(M, t.sh, nó_x) \in Arcos$ e $nó_x$ tem o rótulo " $=M_x$ ".
- 3) $\forall M \in Nós$ e $\neg \exists t.sh \mid M [t.sh >$, então M tem o rótulo "poço".

3.2.2 Grafo de alcançabilidade

Um grafo de alcançabilidade contém as mesmas informações da árvore. A diferença principal é o fechamento da árvore através da eliminação dos *nós duplicados* (que tem um rótulo " $=M_x$ "). Nesta modificação os arcos cujos destinos são *nós duplicados*, têm o seu destino modificado para o nó M_x (anotado no rótulo " $=M_x$ ").

Exemplo 3.1 A árvore (a) da Figura 3 pode ser transformada no grafo (b) através da eliminação dos nós 3 e 4 e da conseqüente modificação do destino dos arcos 3 e 4.

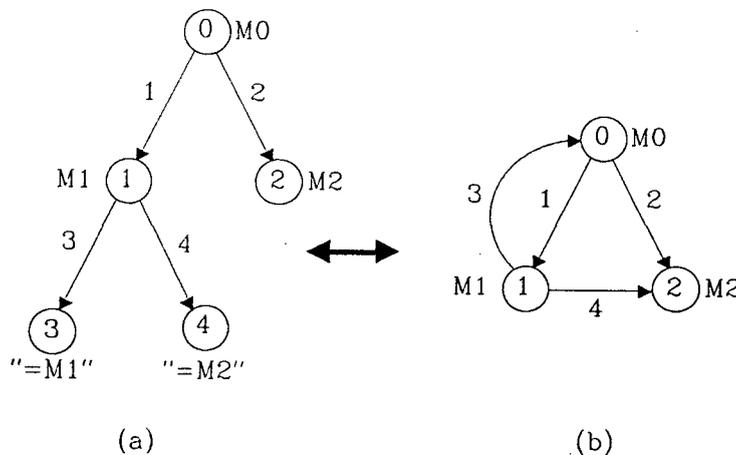


Figura 3. Transformação de uma árvore de alcançabilidade em um grafo de alcançabilidade

3.2.3 O problema da exploração dos estados

Em sistemas complexos o tamanho da AA a ser gerada pode ser tão grande que inviabilize a sua construção completa. Isto se deve ao problema da *explosão combinatória dos estados* que surge por causa da *concorrência*, do *número de indivíduos* que podem habilitar as transições, e da própria *estrutura da rede*.

Exemplo 3.2 Considere a rede ordinária R mostrada na Figura 4 formada por k subredes. As subredes são isoladas uma das outras, e todas tem inicialmente N fichas no lugar p_{1j} ($j = 1 \dots k$).

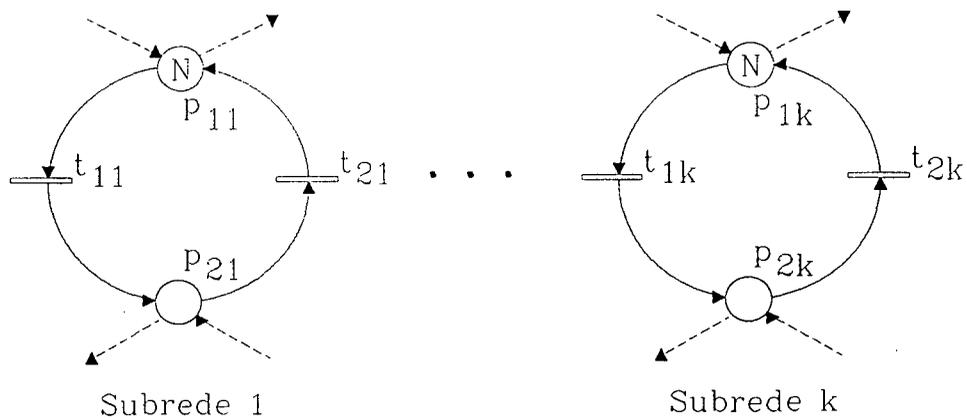


Figura 4. Rede modelando k-processos concorrentes que não interagem

A tabela 3 mostra como ocorre o crescimento do número de transições disparadas (arcos) e marcações armazenadas (nós) do grafo em função do número k de subredes. A tabela 4 mostra o mesmo crescimento mas em função do número N de indivíduos.

Tabela 3. Número de nós e arcos da AA para $k = 3$

N	Nós	Arcos
1	8	24
2	27	108
3	64	288
4	125	600
5	216	1080

Tabela 4. Número de nós e arcos da AA para $N = 2$

K	Nós	Arcos
1	3	4
2	9	24
3	27	108
4	81	432
5	243	1620

Nestes exemplos o número de arcos e nós da árvore é determinado pelas equações:

$$\text{Número de nós} = (N + 1)^k \quad (1)$$

$$\text{Número de arcos} = ((N + 1)^k - (N + 1)^{k-1}) * 2 * k \quad (2)$$

onde N é o número de indivíduos e k o número de subredes.

Exemplo 3.3 Se usarmos esta rede com $N = 9$ e $k = 10$, teremos um número de nós e arcos de 10^{10} e $18 * 10^{10}$ respectivamente. Considerando que seja possível armazenar as informações de cada nó e de cada arco em 1 byte de memória, seriam necessários aproximadamente 200 Giga bytes de memória para o armazenamento completo da árvore.

A partir deste exemplo fica claro a necessidade de uma redução da árvore de alcançabilidade.

3.3 Métodos de geração da árvore de alcançabilidade

Nesta seção serão descritas e analisadas de forma sucinta, algumas das abordagens apresentadas na literatura para as redes de Petri de Alto Nível.

3.3.1 Redução da árvore de alcançabilidade usando a equivalência de marcações

A redução da árvore de alcançabilidade através de uma relação de equivalência entre as marcações foi sugerido inicialmente por Huber [HUBE 86] para as redes de Petri Coloridas, e foi posteriormente estendido por Lindqvist [LIND 90] para as redes Pr/T.

Essa redução pode ser vista como um generalização da redução que ocorre através das marcações duplicadas, de forma que para cada grupo de marcações equivalentes, apenas uma das subárvores é construída. Uma marcação M_1 é dita equivalente a uma outra mar-

cação M_2 quando existe uma função de simetria fs , que permita a transformação de M_1 em M_2 ($fs(M_1) = M_2$).

A relação de equivalência deve ser determinada pelo usuário¹. A correção dessa relação é assegurada desde que [LIND 89]:

- 1) A marcação inicial seja simétrica, isto é, a marcação inicial é insensível à aplicação das funções de simetria ($\forall fs_i, M_0 = fs_i(M_0)$)
- 2) Seja possível inverter a ordem entre a aplicação da função de simetria sobre um indivíduo e o disparo das transições.

Exemplo 3.4 Para ilustrar a necessidade da segunda condição, considere-se o disparo de uma transição t_i que transforma um indivíduo I_1 em um outro indivíduo I_2 . Se a aplicação da função de simetria fs_j no indivíduo I_2 resultar em um indivíduo I_2' , então a aplicação da mesma função de simetria fs_j sobre o indivíduo I_1 seguida do disparo da transição t_i deve resultar no mesmo indivíduo I_2' (i.e., $I_1 [t_i > I_2$ sse $fs_j(I_1) [t_i > I_2'$ e $fs_j(I_2) = I_2'$).

A definição da relação de equivalência é feita através da partição dos indivíduos da rede em conjuntos de indivíduos atômicos e indivíduos relacionados [HUBE 86]. A cada conjunto de indivíduos atômicos é associado um conjunto de funções de simetria do tipo *rotação*, *permutação* ou *identidade*. As simetrias dos indivíduos relacionados decorrem diretamente daquelas dos indivíduos atômicos.

A necessidade da definição dessas partições e das funções de simetrias pelo usuário, antes do início da construção da árvore de alcançabilidade se torna uma limitação frente ao objetivo de automatização fixado neste trabalho. Além disso esta abordagem perde a sua eficiência nos casos em que existem assimetrias na rede.

Entretanto esta abordagem apresenta uma grande vantagem: a possibilidade de provar as propriedades do sistema a partir da AA reduzida, pois o uso das relações de equivalência para a redução não resulta em perda de informação.

3.3.2 Árvore de alcançabilidade parametrizada

A parametrização da árvore de alcançabilidade foi sugerida em [LIND 90] para as redes Pr/T, e busca realizar a redução dessa árvore sem que haja a necessidade de intervenção do usuário na definição das partições e das simetrias. Além disso a parametrização permite que *redes assimétricas* (com marcação inicial não simétrica) também sejam reduzidas.

Este método baseia-se na *parametrização das marcações e dos disparos de transições*. Uma marcação é parametrizada através da substituição dos indivíduos das tuplas das mar-

¹ Chamaremos de *usuário* a pessoa que analisa o sistema.

cações por parâmetros (variáveis), de forma que cada marcação parametrizada passa a representar um conjunto de marcações. O casamento (i.e. determinação da igualdade) entre duas marcações parametrizadas é realizado através de um processo de unificação, onde se busca através da substituição dos parâmetros por outros parâmetros ou indivíduos, determinar se existe uma relação de cobertura ou igualdade entre duas marcações.

A substituição dos indivíduos por parâmetros pode levar a uma marcação parametrizada resultante que represente um grupo de marcações mais extenso do que aquelas marcações que a originaram. Com isso surge o problema de que a AA parametrizada pode incluir marcações que não pertencem a AA completa, impedindo assim o seu uso "direto" para a análise de alcançabilidade, e tornando necessário realizar o desdobramento da árvore através da substituição dos parâmetros por indivíduos. Esta é uma desvantagem do método, que se reflete em uma dificuldade de determinação das *marcações poço*, uma vez que a parametrização da marcação pode unir marcações poço com outras marcações.

A parametrização do disparo das transições permite que novas marcações sejam geradas a partir de marcações antigas. Essa parametrização do disparo das transições é conseguido atribuindo-se também parâmetros às variáveis que estão nas tuplas das fórmulas seletoras. Um arco da AA parametrizada representa o conjunto de todos os disparos que levam as marcações representadas pela marcação parametrizada da origem do arco, para a marcação do destino do arco.

3.3.3 Grafo de alcançabilidade simbólico

O grafo de alcançabilidade simbólico foi introduzido em [HADD 87] para uma subclasse de redes Coloridas denominadas *Redes Regulares*, na qual foram feitas certas restrições sobre o domínio das cores e das funções dos arcos. A construção deste grafo é realizada de maneira totalmente automática, sem que o usuário necessite introduzir informações heurísticas para decidir quanto as agregações de marcações a serem feitas.

Esta abordagem consegue desta forma *explorar as simetrias* da rede para reduzir o grafo de forma automática, sem o conhecimento explícito das simetrias do modelo a ser estudado. O grafo simbólico é constituído de marcações simbólicas e de arcos que representam os disparos das transições habilitadas simbolicamente.

O princípio da representação simbólica da marcação nas *Redes Regulares* consiste em agregar em subclasses todos os objetos (fichas coloridas) que pertencem a um marcação, de forma que a permutação de dois objetos dentro de uma subclasse não a modifique [HADD 87]. Na marcação simbólica a identidade dos objetos dentro de uma subclasse pode ser desprezada, levando-se em conta *apenas a cardinalidade* (número de objetos) de cada subclasse.

A marcação simbólica passa a representar uma classe de marcações equivalentes, sendo que essa equivalência é entendida como uma relação que associa duas marcações que podem ser obtidas uma da outra através da permutação dos objetos de cada classe (cor).

Para a construção do grafo é definida uma regra de disparo simbólica das transições. Um transição está habilitada simbolicamente se e somente se existe uma marcação ordinária² que habilite essa transição. A marcação ordinária obtida pelo disparo da transição (com cores) pertence a marcação simbólica obtida do disparo simbólico da transição.

A principal vantagem dessa abordagem é a equivalência entre a habilitação simbólica de transições e da alcançabilidade de marcações em relação ao grafo de alcançabilidade completo, permitindo assim a determinação das propriedades dinâmicas tais como: alcançabilidade, estados de recepção, quase-vivacidade, e reinicialização. O grande problema dessa abordagem está na recuperação do valor dos objetos para o seu uso em técnicas de verificação das propriedades específicas do sistema modelado [HADD 87].

3.3.4 Redução do espaço de estados através de conjuntos "Stubborn"

A teoria de conjuntos "*stubborn*" (obstinados), foi proposta em [VALM 88] e utiliza a noção de *concorrência* do sistema para agrupar as transições. Essa teoria foi desenvolvida para sistemas chamados de Variável/Transição que são uma abstração das redes de Petri [VALM 89] e também para as redes de Petri Lugar/Transição.

Um conjunto "*stubborn*" consiste num grupo de transições que possuem interações umas com as outras, isto é, a desabilitação de uma transição pelo disparo de uma outra indica que elas pertencem ao mesmo conjunto "*stubborn*".

Essa noção é dinâmica e depende do estado (marcação) corrente do sistema, de forma que os conjuntos "*stubborn*" devem ser determinados a cada nova marcação gerada.

A redução do grafo de alcançabilidade proposta em [VALM 89] é baseada na observação de que a ordem de execução das ações em processos concorrentes que não interagem, não altera o resultado final.

A redução do grafo de marcações alcançáveis é conseguido, expandindo-se em cada marcação apenas as transições habilitadas em um dos conjuntos "*stubborn*". Quando um desses conjuntos contém um número de transições menor que o total de transições da rede, consegue-se evitar a necessidade de disparo de várias transições, permitindo-se assim agregar di-

² Nas Redes Regulares as marcações que correspondem a distribuição das cores nos lugares (i.e., não são simbólicas) são chamadas de marcações ordinárias.

retamente aquelas marcações que possuem informações redundantes em relação a outras já representadas.

Este método preserva todos as *marcações poço* e as possibilidades de não determinismo. Além disso, todas as seqüências de disparo de transição que levam a uma marcação poço são representadas diretamente ou então correspondem a uma permutação das transições desta seqüência.

O uso dessa abordagem é muito útil na validação de sistemas com terminação, porém em sistemas sem terminação o seu uso é limitado devido a possibilidade de que algumas transições habilitadas possam ser ignoradas. Em [VALM 89] é proposto um algoritmo para considerar estas transições, no entanto o seu custo em tempo é muito alto. No caso de não haver transições ignoradas, ou no caso em que tenham sido corrigidas pelo algoritmo citado, as propriedades de vivacidade são conservadas no grafo reduzido.

Como o princípio dessa abordagem (concorrência) está ligado a uma característica diferente das anteriores (simetrias), Valmari sugere que o uso das duas abordagens em conjunto resulta numa redução ainda maior do grafo, desde que o sistema analisado possua características de simetrias e de concorrência.

3.3.5 Geração do espaço de estados através da composição

Este tipo de abordagem foi proposto também por Valmari [VALM 90] para as redes de Petri. A técnica consiste em dividir o sistema a ser analisado em subsistemas, gerar o espaço de estados de cada uma delas, condensar o espaço de estados de acordo com um critério de equivalência escolhido antecipadamente, e compor os subsistemas e os seus espaços de estados condensados.

As propriedades a serem analisadas devem ser definidas antes da aplicação da técnica, pois a escolha do critério de equivalência determinará a forma de condensação dos espaços de estados de cada parte do sistema. Em conseqüência, uma das desvantagens desse método é que o seu uso para a análise de diversas propriedades pode resultar na geração de vários espaços de estados.

Um outro problema está relacionado com a dificuldade de se dividir o sistema, o que pode levar a obter espaços globais de tamanho e forma diferentes. Entretanto, no caso do sistema ter sido construído através do método de refinamentos, este problema é minorado. Essa técnica possui a vantagem de poder utilizar a análise dos subsistemas para analisar o sistema global.

3.4 Método de redução proposto

3.4.1 Princípios básicos usados para a redução da árvore de alcançabilidade

No método proposto neste trabalho optamos por uma *redução da Árvore de Alcançabilidade (AA) sem perda da capacidade de análise*. Escolhemos como princípio básico de redução a remoção das subárvores com *comportamentos análogos* àquelas subárvores já construídas na Árvore de Alcançabilidade Reduzida (AAR), como em [HUBE 86].

Definição 3.2 Os *comportamentos* de duas subárvores são *análogos* se as mesmas seqüências de disparo de transições são possíveis a partir de raiz de ambas.

Neste caso, a única diferença entre as seqüências de disparos das subárvores é devida aos indivíduos que habilitam as transições.

Para preservar as informações relacionadas com as subárvores removidas, é utilizada uma *anotação* que possibilita a recuperação das informações a partir da subárvore análoga na AAR. Estas *anotações* contêm as seguintes informações: a marcação da Rede de Petri que é raiz da *subárvore análoga* na AAR: as *funções* que transformam esta subárvore naquela que foi removida.

Para ilustrar o método proposto, apresentaremos inicialmente a construção da AA e AAR para o "Almoço dos Cinco Filósofos" modelado em uma rede Predicado/Transição conforme a Figura 5 [LIND 89].

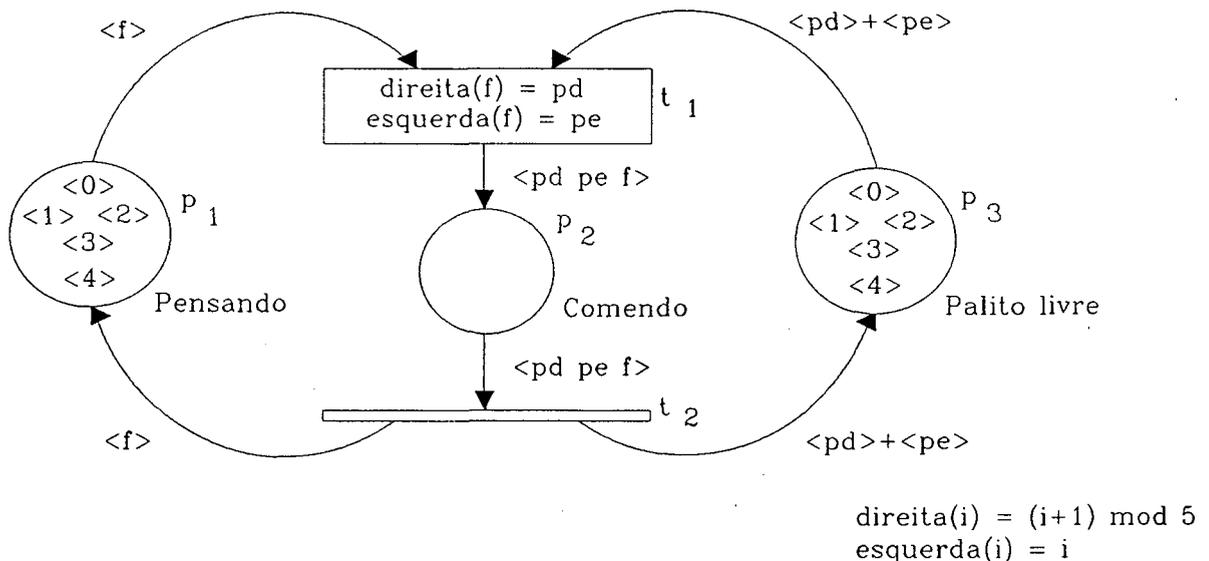


Figura 5. O almoço dos cinco filósofos - modelado em rede Pr/T

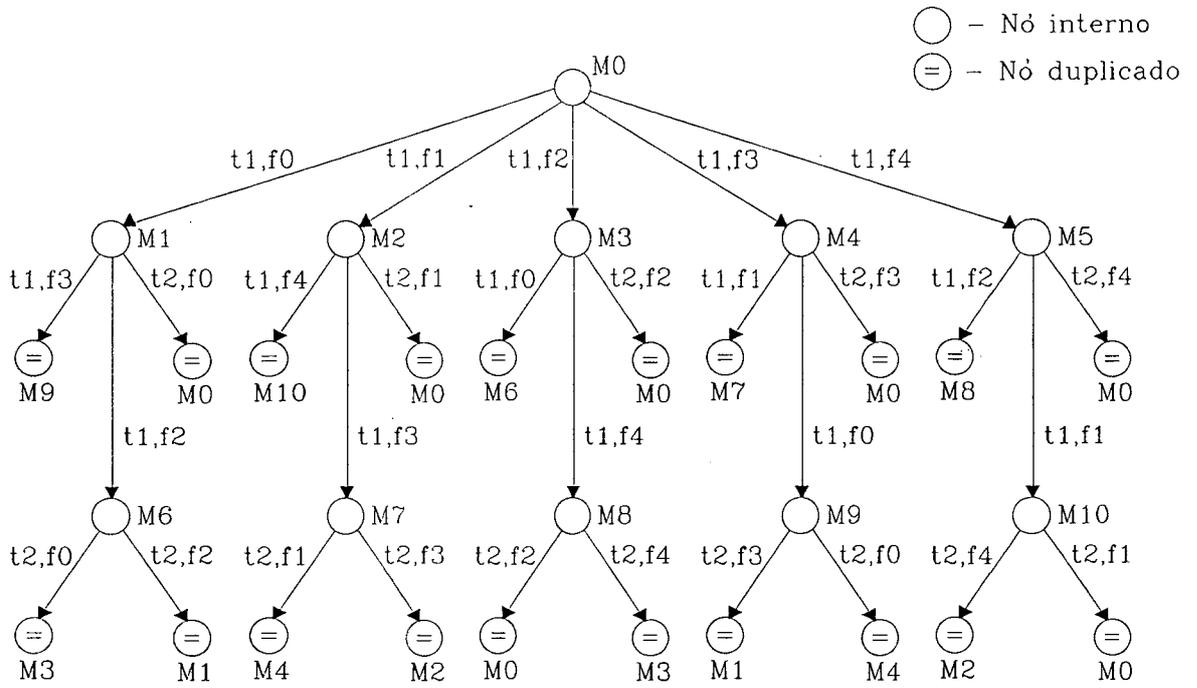


Figura 6. Árvore de alcançabilidade para o almoço dos cinco filósofos - modelado em rede Pr/T

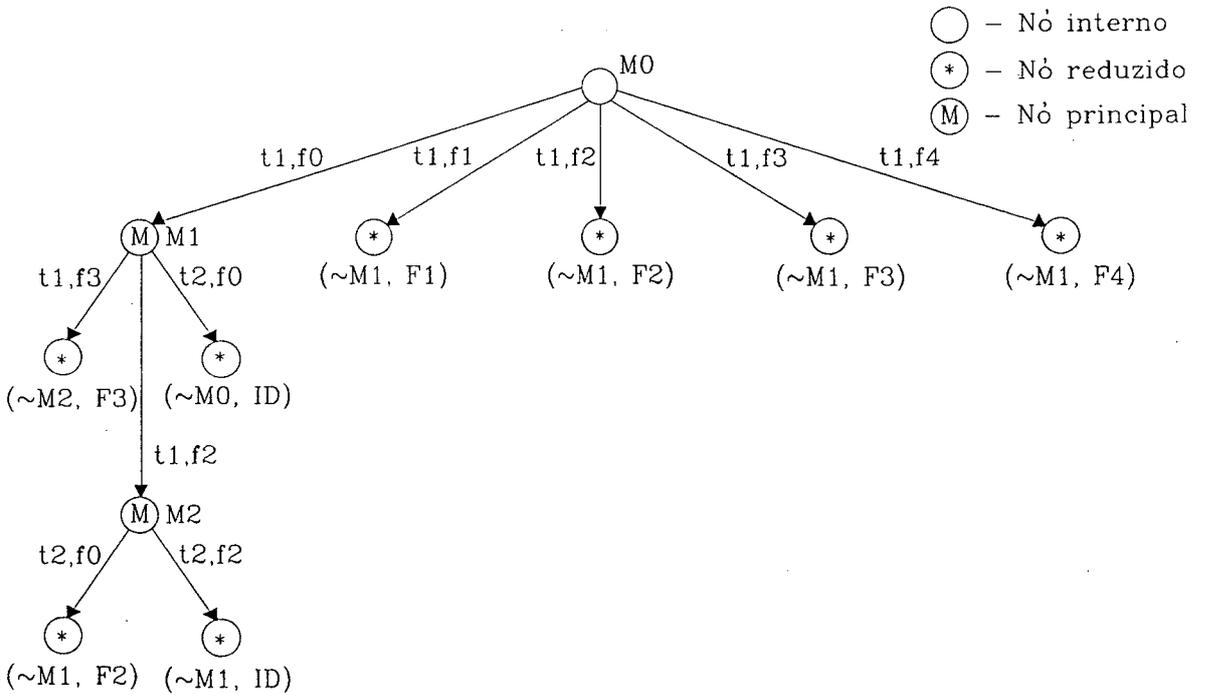


Figura 7. Árvore de alcançabilidade reduzida para o almoço dos cinco filósofos

Na AA do modelo mostrada na Figura 6 podemos observar que as subárvores cujas raízes são as marcações M_1 à M_5 , tem comportamentos análogos. Também podemos verificar que apenas três estados entre os onze representam situações diferentes, que correspondem aos seguintes casos: nenhum filósofo almoçando (M_0), apenas um filósofo almoçando (M_1 à M_5), e dois filósofos que não são vizinhos estão almoçando (M_6 à M_{10}). As subárvores com raiz em M_1 à M_5 possuem as mesmas seqüências de disparo de transição, tendo assim comportamentos análogos.

Pode-se então obter uma redução neste exemplo através da eliminação de quatro das cinco subárvores com comportamentos análogos (M_1 à M_5). A AA reduzida é mostrada na Figura 7.

Notação 3.3 Denotaremos a subárvore cuja raiz é M_x por *Subárvore*(M_x).

Chamaremos de *subárvores análogas*, àquelas que representam a AA após a redução; os *nós principais* (representados graficamente por "M") correspondem às raízes das subárvores análogas e contém os valores das marcações. As *subárvores removidas* são representadas pelo caractere "*" no nó *reduzido* correspondente a sua raiz. A anotação (M_x, F_y) no nó *reduzido* contém as seguintes informações: M_x indica a marcação correspondente ao nó *principal* da AAR, e F_y são as funções que relacionam as subárvores removidas com a *Subárvore*(M_x) da AAR.

Embora as marcações correspondentes aos nós reduzidos sejam geradas durante a fase de construção da AAR, elas não são armazenadas na árvore. De fato, os nós *reduzidos* não contém uma marcação, mas apenas uma *anotação*.

A recuperação da *Subárvore*(M_2) que foi removida, é feita a partir da *Subárvore*(M_1) representada na AAR, usando-se as funções F_j ; o mesmo é válido para as outras subárvores removidas.

3.4.2 Definição da marcação equivalente

Para podermos reduzir drasticamente o tempo relacionado com a comparação das marcações, será necessário evitar a construção das subárvores análogas na AAR. Por isso, para cada nova marcação gerada, procuraremos uma relação de equivalência com outras já armazenadas na AAR.

Definição 3.4 Seja $[M_0>$ o conjunto das marcações alcançáveis a partir de M_0 , e duas marcações $M_x, M_y \in [M_0>$. Uma *relação de equivalência* existe entre elas se e somente se as subárvores com raiz em cada uma dessas marcações tem *comportamentos análogos*.

Notação 3.5 Duas marcações equivalentes M_x e M_y serão denotadas por $M_x \sim M_y$

O conceito de marcação equivalente foi inicialmente introduzido por Huber para as Redes de Petri Coloridas [HUBE 86] conforme vimos na Seção 3.3.1. Esta relação de equivalência deve-se a existência de uma função de simetria que permite a transformação consistente de uma marcação em outra, através da simples troca de indivíduos.

Definição 3.6 Duas marcações M_x e M_y são equivalentes sse existe uma função de simetria fs tal que $M_x = fs(M_y)$ [HUBE 86].

A partir desta definição o problema da *determinação da equivalência* entre duas marcações pode ser substituído pela *determinação das funções de simetria*. A procura da solução deste problema receberá uma atenção especial neste trabalho.

3.4.3 O uso das funções de simetria

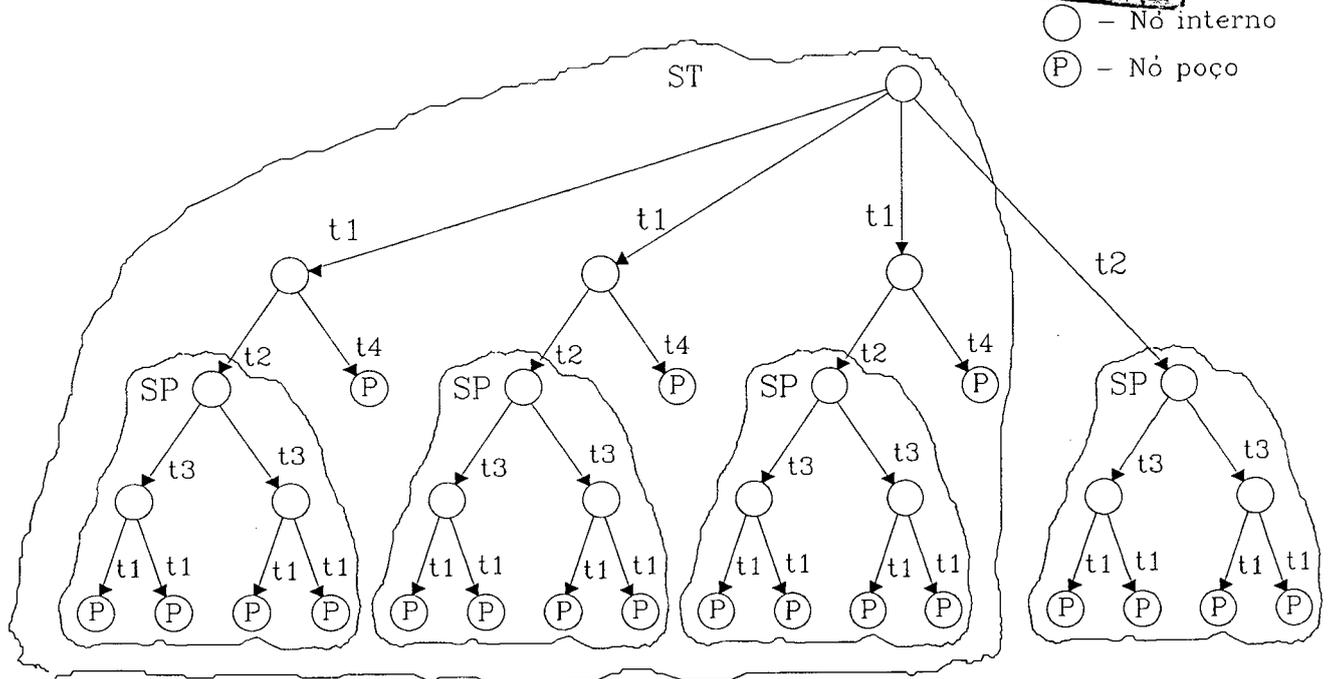
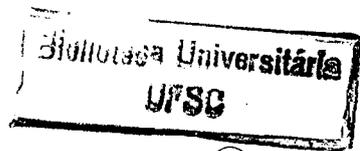
No método proposto em [HUBE 86] a definição das funções de simetria deve ser realizada pelo usuário antes do início da construção da AA. Neste método a marcação inicial deve ser obrigatoriamente simétrica.

Como já comentamos anteriormente isto faz com que apareçam duas principais limitações:

1. As simetrias que existem na marcação inicial (chamadas de *simetrias totais*) são as únicas usadas; por isso, todas as simetrias que surgem durante a construção da AA (chamadas *simetrias parciais*), não são levadas em conta.
2. Para que seja alcançada a maior redução possível, o usuário deve achar todas as simetrias [HUBE 86]. Esta tarefa pode ser muito difícil em sistemas complexos.

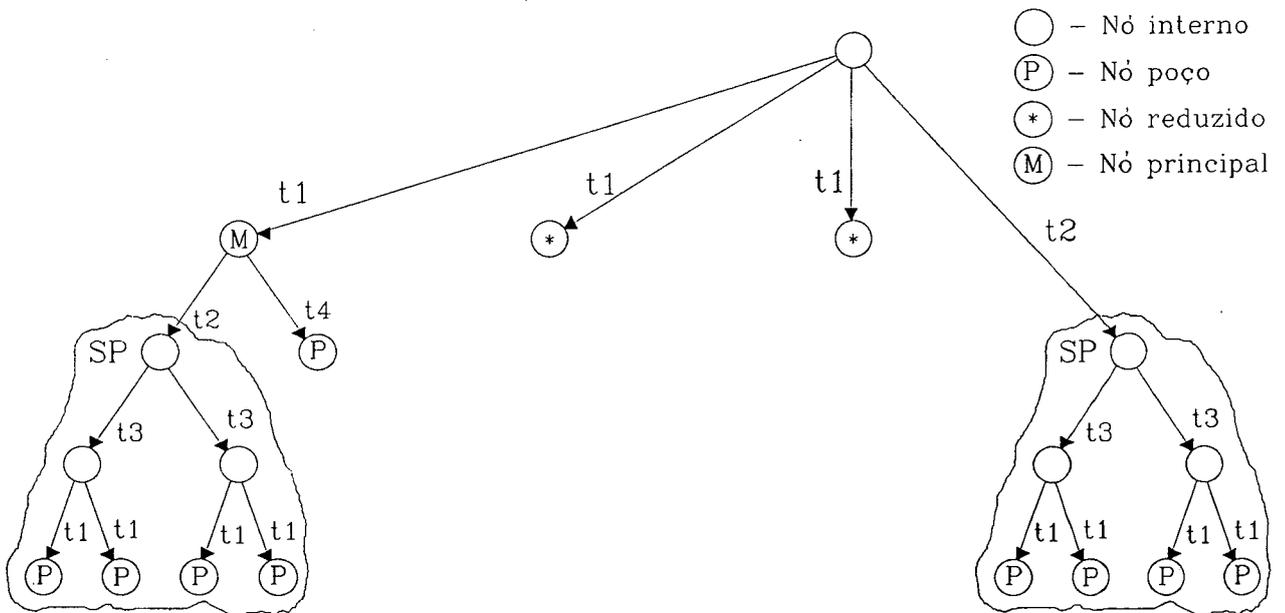
Na Figura 8 é mostrada uma árvore de alcançabilidade que possui os dois tipos de simetria. Usando-se apenas as *simetrias totais* na redução da AA, como nos métodos citados anteriormente, obtém-se a árvore de alcançabilidade mostrada na Figura 9. Não há dúvidas que o uso das simetrias parciais durante a fase de redução, permite uma redução maior da árvore, como mostra a Figura 10. O uso das *simetrias parciais* é particularmente importante quando lidamos com redes não reiniciáveis.

Na abordagem proposta usaremos os dois tipos de simetria. Estas deverão ser encontradas durante a construção da AAR, uma vez que as simetrias parciais podem ser válidas para as marcações de uma determinada subárvore e não serem válidas naquelas de outra subárvore. Sendo assim a determinação da simetria deve ser feita de modo dinâmico, a medida que cada nova marcação é gerada, contrariamente ao método de [HUBE 86].



- - Nó interno
- Ⓟ - Nó poço

Figura 8. Simetrias em árvore de alcançabilidade



- - Nó interno
- Ⓟ - Nó poço
- * - Nó reduzido
- Ⓜ - Nó principal

Figura 9. Redução da árvore de alcançabilidade usando as simetrias totais

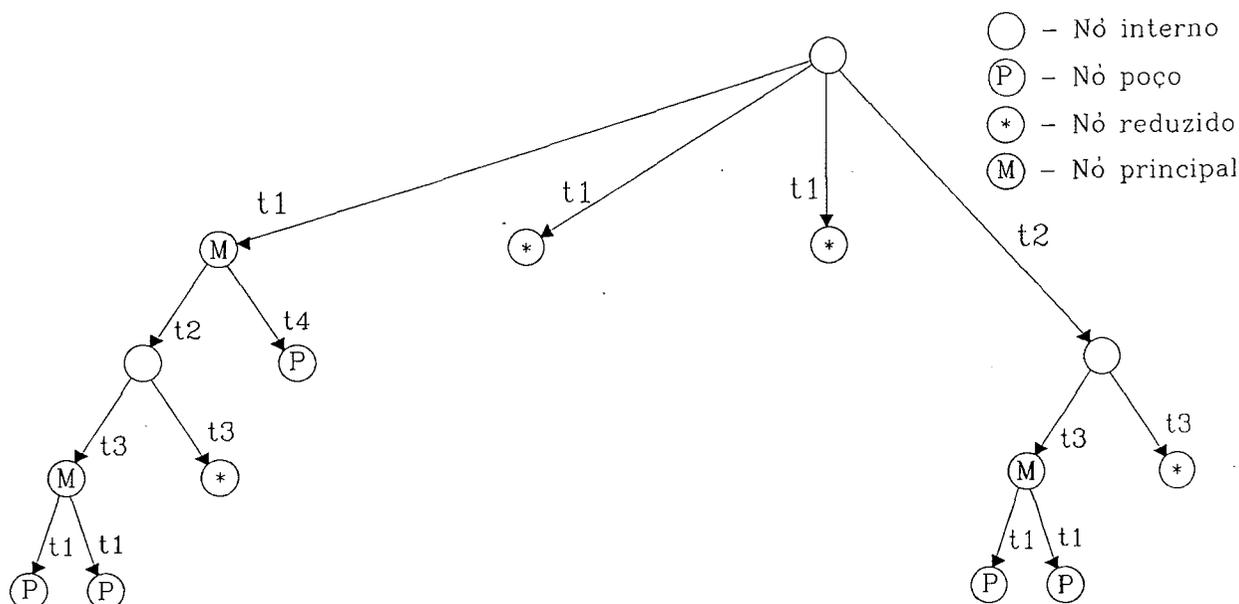


Figura 10. Redução da árvore de alcançabilidade usando as simetrias totais e parciais

Através da diminuição do número de comparações de marcações, tentaremos reduzir o custo em tempo introduzido pela determinação dinâmica das simetrias. Por isso, necessitamos determinar e separar, quando da geração das marcações, aquelas que não podem ser equivalentes, através do agrupamento das marcações que possuem as mesmas características necessárias para a equivalência, como veremos em detalhes no Capítulo 4.

As funções de simetria serão encontradas através de um procedimento de generalização de marcações, o qual permitirá encontrar o conjunto de funções que inclui todas as possíveis simetrias para cada marcação gerada. Uma vez determinadas essas funções é necessário distinguir quais dentre elas são simétricas e quais não são.

Na abordagem proposta, assim como no método descrito em [HUBE 86] a tarefa de realizar esta separação cabe ao usuário. Contudo aqui a sua tarefa é facilitada uma vez que o conjunto de funções que precisam ser analisadas é menor. Além disso a única resposta requerida do usuário em relação as funções obtidas, é se elas são simétricas ou não. Um outro fator importante é que a intervenção do usuário também só ocorre em algumas das marcações geradas.

Em *alguns casos particulares* que correspondem a uma subclasse de redes Pr/T, é possível encontrar diretamente as funções de simetria através do procedimento *Generalização-de-Marcações*. Estas redes são denominadas de *Redes com Árvore de Alcançabilidade Diretamente Redutível (RDR)*, pois a redução pode ser realizada de forma completamente automática. Dois exemplos clássicos que pertencem a esta subclasse são o *Almoço dos Cinco Filóso-*

fos [LIND 90] e o *Protocolo para a Manutenção de Múltiplas Cópias em um Bando de Dados* [GENR 87].

3.5 Conclusão

Apresentamos neste capítulo alguns métodos que permitem resolver parcialmente o problema da explosão de estados da árvore de alcançabilidade das redes de Petri.

Entretanto estes métodos possuem algumas limitações:

- . Não é automático [HUBE 86]
- . Exigem estrutura simétrica das redes [HUBE 86] e [LIND 89].
- . Não mantêm a capacidade de análise das propriedades [LIND 90, VALM 89, VALM 90].
- . Dificuldade na recuperação dos objetos para uso na validação [HADD 87]
- . Restrição da aplicação a subclasses das redes de Alto Nível [HADD 87] e [LIND 90]

Apresentamos o método proposto neste trabalho, cujo objetivo é a redução da árvore de alcançabilidade mantendo a sua capacidade de análise. Mostramos que o uso das simetrias parciais permite uma redução maior do que o uso somente das simetrias totais (presentes na marcação inicial da rede) como em alguns dos métodos estudados.

A determinação da equivalência entre marcações que é a base fundamental deste método de redução, foi também brevemente discutido neste capítulo, e foi mostrado que esta determinação é equivalente a encontrar as funções de simetria que transformam uma marcação na outra.

Concluindo este capítulo resumiremos aqui o método a ser usado na construção da AAR. Este consiste em duas etapas, que se repetem a cada nova marcação gerada.

1. Determinação da equivalência das marcações.
2. Redução da árvore de alcançabilidade.

O estudo detalhado de cada uma destas etapas será realizado nos próximos capítulos.

CAPÍTULO IV

DETERMINAÇÃO DA RELAÇÃO DE EQUIVALÊNCIA ENTRE DUAS MARCAÇÕES

4.1 Introdução

Neste capítulo apresentaremos a metodologia proposta para a *determinação da relação de equivalência* entre duas marcações, a ser usada na redução da Árvore de Alcançabilidade (AA). Esta determinação consiste em um ponto chave da redução, pois evita a construção das subárvores análogas.

A determinação da equivalência deve ser realizada a cada nova marcação gerada, e é equivalente a *encontrar as funções simetria* desta marcação em relação as outras marcações já existentes na árvore, conforme já visto no capítulo 3.

No capítulo anterior, mostrou-se como as funções de simetria podem ser usadas na redução da AA. Neste capítulo será mostrado como estas simetrias podem ser determinadas.

A determinação da equivalência entre duas marcações pode ser separada em quatro passos, conforme mostrado a seguir:

Passo 1. Classificação das posições das tuplas em dois tipos de subconjuntos de posições: *relacionadas por valor*, e *com valor irrelevante*. Essa classificação possibilita a realização da troca dos indivíduos de uma maneira consistente.

Passo 2. Classificação das marcações de acordo com as condições de equivalência, e separação das marcações em subgrupos, o que possibilita uma determinação mais rápida da equivalência.

Passo 3. Determinação das funções de transformação utilizando um procedimento de generalização de marcações. Este procedimento automatiza parcialmente a determinação das funções de simetria no caso geral das redes Pr/T, e em alguns casos possibilita a automação completa da redução da árvore de alcançabilidade.

Passo 4. Verificar quais das funções de transformação obtidas são simétricas. *Esta é a única tarefa que cabe ao usuário.*

A seguir detalharemos cada um desses passos.

4.2 PASSO 1 - Classificação das posições das tuplas

As tuplas da rede Pr/T indicam uma relação dinâmicas existente entre seus indivíduos. Por exemplo na Figura 5 a tupla $\langle 2, 1, 1 \rangle$ representa a seguinte relação dinâmica: "O filósofo 1 está utilizando os palitos 1 e 2".

Neste trabalho será necessário distinguir entre as posições dos elementos nas tuplas, por isso introduziremos o conceito de "*posição de tupla*" (ou simplesmente *posição*).

Notação 4.1 A posição i das tuplas do lugar p_j será denotada por Pos_{ip_j} .

Na nossa abordagem será necessário considerar também a relação que existe devido ao uso de diferentes posições das tuplas em uma transição em uma comparação. Neste caso o valor relativo entre os indivíduos destas posições das tuplas é um fator muito importante para a determinação do comportamento dinâmico do sistema modelado pela rede de Petri.

A análise automática de cada transição antes do início da construção da árvore permite determinar as relações existentes entre as posições de cada tupla, e classifica-las de acordo com o tipo: subconjuntos de *posições relacionadas por valor* (SPRV) e subconjunto de *posições com valor irrelevante* (SPVI).

4.2.1 Posições das tuplas relacionadas por valor

Definição 4.2 Seja $t \in T$ uma transição da rede de Petri (R, M_0) . Duas posições das tuplas (no mesmo lugar ou em lugares diferentes) estão *relacionadas por valor* e pertencem ao mesmo subconjunto SPRV se:

- 1) As variáveis destas posições são usadas em uma mesma fórmula seletora dessa transição t , ou
- 2) Uma variável X está em mais de um arco de entrada da transição t .

Em ambas as situações, quando modificamos o indivíduo de uma posição da tupla, a modificação do indivíduo da outra posição relacionada deve também ocorrer para que seja mantido o mesmo comportamento.

Exemplo 4.1 Considere a rede da Figura 11. Se o indivíduo de Pos_{2p_1} for modificado para 7, então o indivíduo de Pos_{2p_3} também deverá ser modificado para 7, uma vez que são capturados pela mesma variável Z . De forma semelhante, se modificarmos o indivíduo de Pos_{1p_1} for

modificado para 5, então o indivíduo de Pos_{Ip2} deverá ser modificado para 4, para que a transição t permaneça habilitada, uma vez que as variáveis que capturam estes indivíduos são utilizadas na mesma fórmula seletora.

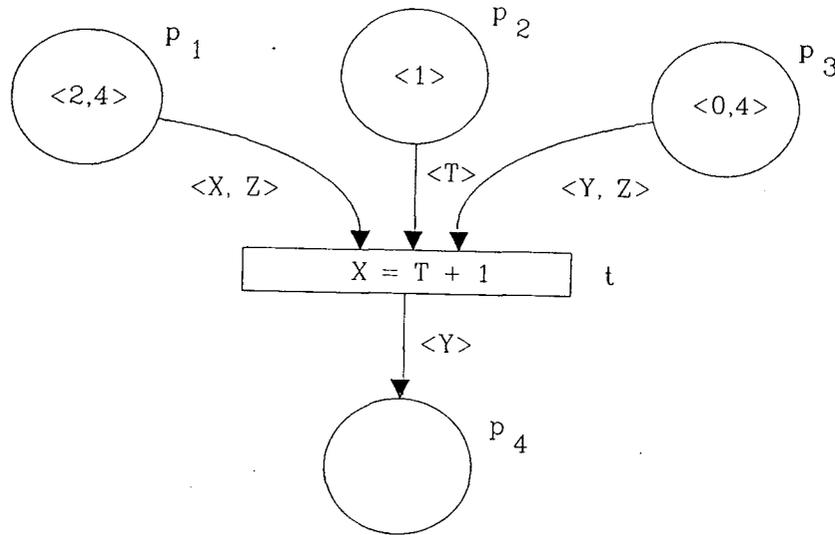


Figura 11. Determinação das posições relacionadas por valor e posições com valor irrelevante em redes Pr/T

Proposição 4.3 A situação 2 da definição 4.2 pode ser transformada na situação 1 através da troca de todas as ocorrências da variável X nos arcos de entrada dessa transição por variáveis distintas $X_1, X_2 \dots X_k$ e pela introdução das fórmulas seletoras $X_1 = X_2, X_2 = X_3 \dots X_{k-1} = X_k$.

Corolário 4.4 Se existirem dois subconjuntos de posições relacionadas por valor $SPRV_1$ e $SPRV_2$ que contém uma mesma posição de tupla, então estes subconjuntos devem ser unidos em um único subconjunto $SPRV$:

$$\exists Pos_{ipj} \mid Pos_{ipj} \in SPRV_1 \text{ e } Pos_{ipj} \in SPRV_2 \Rightarrow SPRV = SPRV_1 \cup SPRV_2.$$

Exemplo 4.2 Considere a transição t_1 da Figura 5. Pela análise das fórmulas seletoras $direita(f) = p_d$ e $esquerda(f) = p_e$ podemos constatar que as posições Pos_{Ip1} e Pos_{Ip3} estão relacionadas por valor. Portanto $SPRV_1 = \{Pos_{Ip1}, Pos_{Ip3}\}$. Da posição das variáveis f, p_d e p_e nas tuplas anotadas nos arcos dessa transição, podemos determinar ainda que $SPRV_2 = \{Pos_{Ip1}, Pos_{3p2}\}$ e $SPRV_3 = \{Pos_{Ip3}, Pos_{Ip2}, Pos_{2p2}\}$. Como Pos_{Ip1} pertence aos subconjuntos $SPRV_1$ e $SPRV_2$, e Pos_{Ip3} pertence aos subconjuntos $SPRV_1$ e $SPRV_3$, podemos determinar que o subconjunto das tuplas relacionadas por valor neste caso é $SPRV = SPRV_1 \cup SPRV_2 \cup SPRV_3 = \{Pos_{Ip1}, Pos_{Ip2}, Pos_{2p2}, Pos_{3p2}, Pos_{Ip3}\}$.

A relação por valor entre duas posições das tuplas é geralmente uma noção que o próprio usuário possui durante a modelagem do sistema. Se utilizarmos essa informação do usuário então é possível realizar a detecção prévia de erros do modelo, tais como atribuições erradas de valores e comparações entre indivíduos de posições das tuplas pertencente a subconjuntos diferentes.

Essa noção de posições das tuplas *relacionadas por valor* será posteriormente utilizada para permitir a modificação consistente dos indivíduos durante o Procedimento *Generalização-de-Marcações*.

4.2.2 Posições das tuplas com valor irrelevante

Definição 4.5 Uma posição de tupla é de *valor irrelevante*, quando a variável dessa posição não aparece em nenhuma fórmula seletora A_T , nem em mais de um arco de entrada de alguma transição.

Em outras palavras, estas *posições das tuplas* permanecem sempre com o mesmo indivíduo durante toda construção da árvore de alcançabilidade. Elas também não são usadas em comparações entre indivíduos de posições das tuplas.

Exemplo 4.3 Na Figura 11, Pos_{1p3} é um posição de *valor irrelevante*.

Esse tipo de posição de tupla é somente usada como simples fichas ou então para conter "tags"¹. A modificação dos indivíduos dessa posição de tupla pode ser realizada de forma aleatória sem que isso altere o comportamento da rede.

As posições das tuplas com *valor irrelevante* simplificam a redução, uma vez que seus indivíduos podem ser desprezados durante a comparação das marcações.

4.3 PASSO 2 - Classificação das marcações de acordo com as condições de equivalência

O problema da determinação da equivalência de marcações pode ser definida da seguinte forma:

Definição 4.6 Seja f_s uma função simétrica, M_x uma marcação, e $[M_\theta >$ o conjunto de marcações alcançáveis a partir da marcação inicial M_θ . A determinação da equi-

¹ Tags são um tipo especial de indivíduos que permitem fazer a distinção entre as tuplas que representam cópias de um mesmo objeto (ver em [GENR 87] na seção 3.4).

valência entre a marcação M_x e uma das marcações de $[M_0>$, equivale a verificação da relação:

$$\exists M_y \in [M_0> \text{ e } \exists fs \mid M_x = fs(M_y)$$

Notação 4.7 A relação de equivalência será denotada por " \sim ".

O tamanho do conjunto de marcações da árvore de alcançabilidade cresce continuamente durante a construção desta. Por isso, subdividiremos este conjunto em grupos de marcações de acordo com as condições necessárias para a sua equivalência.

As duas primeiras condições apresentadas a seguir serão usadas unicamente com este propósito, enquanto que a terceira condição além desse propósito, possibilitará a determinação do conjunto de funções que contém todas as funções simétricas para duas marcações dadas.

4.3.1 A cardinalidade da marcação

A seguir introduziremos a noção de cardinalidade da marcação que será usada na determinação da primeira condição de equivalência.

Definição 4.8 Seja M uma marcação, e $P = \{p_1, p_2 \dots p_n\}$ o conjunto de lugares da rede. A *cardinalidade da marcação* $|M|$ é definida pela lista de números de tuplas $NT(p_j)$ contidos em cada lugar $p_j \in P$, i.e. $|M| = (NT(p_1), NT(p_2), \dots, NT(p_n))$

Para assegurar que duas marcações possuam as mesmas substituições de variáveis por indivíduos, é inicialmente necessário (mas não suficiente) que as duas marcações possuam o mesmo número de tuplas em cada lugar. Por isso, podemos afirmar que uma condição necessária para a equivalência entre duas marcações é a igualdade entre a cardinalidade de cada lugar.

Teorema 4.9 Para que duas marcações M_x e M_y possam ser equivalentes é necessário que:
 $|M_x| = |M_y|$ (CONDIÇÃO 1).

4.3.2 Cardinalidade do conjunto de substituições de habilitação

A transição de uma rede Pr/T pode ser disparada usando diferentes substituições de grupos de indivíduos pelas variáveis de entrada.

Definição 4.10 O conjunto de substituições de habilitação $SH(M_x) = \{t_1.sh_1, t_1.sh_2 \dots t_1.sh_k, t_2.sh_{k+1} \dots t_n.sh_1\}$ na marcação M_x representa todas as possibilidades de evolução da rede a partir de M_x .

Definição 4.11 A *cardinalidade do conjunto de substituições de habilitação* é a lista do número de substituições de habilitação ($NSHt_i$) de cada transição $t_i \in T$ na marcação M_x :

$$|SH(M_x)| = (NSHt_1, NSHt_2 \dots NSHt_n)$$

A partir desse conceito de *conjunto de substituições de habilitação* podemos estabelecer outra condição necessária para a equivalência entre duas marcações. No capítulo 3 estabelecemos que as subárvores de marcações equivalentes devem descrever comportamentos análogos, logo podemos concluir que o número de substituições de habilitação para cada transição da rede de duas marcações equivalentes deve ser igual. Esta conclusão pode ser formalizada pelo seguinte teorema:

Teorema 4.12 Seja $SH(M)$ o conjunto de substituições de habilitação conforme a Definição 4.10. Para que duas marcações M_x e M_y possam ser equivalentes é necessário que:

$$|SH(M_x)| = |SH(M_y)| \text{ (CONDIÇÃO 2)}$$

4.3.3 Funções de transformação

Para que duas marcações possam ser equivalentes deve existir uma função que permita a transformação de uma marcação em outra através da troca de indivíduos.

Notação 4.13 Uma função que transforma uma marcação em outra, através da troca de indivíduos, será chamada de *função de transformação*.

Teorema 4.14 Seja f uma *função de transformação* da marcação M_x na marcação M_y . Para que M_x possa ser equivalente a M_y é necessário que exista uma função de transformação f que seja bijetora, i. e.:

$$\exists f \mid f(M_x) = M_y \text{ e } M_x = f^{-1}(M_y) \text{ (CONDIÇÃO 3)}$$

A partir desta condição podemos determinar o conjunto de todas as funções de transformação que mudam os indivíduos das posições das tuplas de duas marcações de forma a tornar essas marcações iguais. As funções de simetria que procuramos também modificam os indivíduos, porém de uma forma que não altera o comportamento da rede, isto é, mantém habilitadas as mesmas seqüências de disparo de transição. Conseqüentemente essas funções de simetria estarão incluídas neste conjunto de funções.

Exemplo 4.4 Sejam $M_1 = (\langle a \rangle \langle b \rangle \langle d \rangle)$ e $M_2 = (\langle a \rangle \langle d \rangle \langle d \rangle)$ duas marcações e seja $f: (b \rightarrow d)$ uma função que transforma M_1 em M_2 . Podemos notar que $f(M_1) = M_2$, mas $f^{-1}(M_2) = (\langle a \rangle \langle b \rangle \langle b \rangle) \neq M_1$ pois $f^{-1}: (d \rightarrow b)$. Por isso podemos concluir que M_1 não é equivalente a M_2 .

4.3.4 Três condições necessárias para a equivalência

Concluindo esta seção uniremos em um único teorema as três condições que são necessárias para que possa existir a equivalência entre duas marcações.

Teorema 4.15 Sejam M_x e M_y duas marcações. Para que M_x possa ser equivalente a M_y é necessário que as três condições abaixo sejam satisfeitas:

- 1) $|M_x| = |M_y|$
- 2) $|SH(M_x)| = |SH(M_y)|$
- 3) $\exists f \mid f(M_x) = M_y$ e $M_x = f^{-1}(M_y)$

A verificação das duas primeiras condições é trivial. A terceira condição será determinada pelo procedimento descrito pelo passo a seguir.

4.4 PASSO 3 - Determinação das funções de transformação

Veremos nesta seção, como a verificação da existência das funções de transformação para duas marcações pode ser realizada através da generalização das marcações, a qual também será descrita aqui.

O procedimento de generalização usado neste trabalho é realizado da forma inversa ao procedimento de unificação [KNIG 89]. Embora a generalização possa ser aplicada a várias marcações ao mesmo tempo, neste trabalho ela sempre será aplicada sobre pares de marcações.

Quando duas marcações M_x e M_y são generalizadas, os seus indivíduos são substituídos por variáveis, de tal forma que uma marcação generalizada comum é obtida.

Essas generalizações podem ser consistentes ou não, conforme definido a seguir:

Definição 4.16 Uma *generalização* é *consistente* se ela dá origem a uma função f que transforma a marcação M_x na marcação M_y e f é bijetora (ou seja, a terceira condição do Teorema 4.15 é verdadeira).

Exemplo 4.5 Considere-se a Figura 12 e as funções $f_{12}: (b \rightarrow d)$ e $f_{23}: (a \rightarrow d, d \rightarrow b)$. A marcação generalizada Mg_{12} é inconsistente porque $M_1 \neq f_{12}^{-1}(M_2)$, enquanto que a marcação generalizada Mg_{23} é consistente porque $f_{23}(M_2) = M_3$ e $M_2 = f_{23}^{-1}(M_3)$.

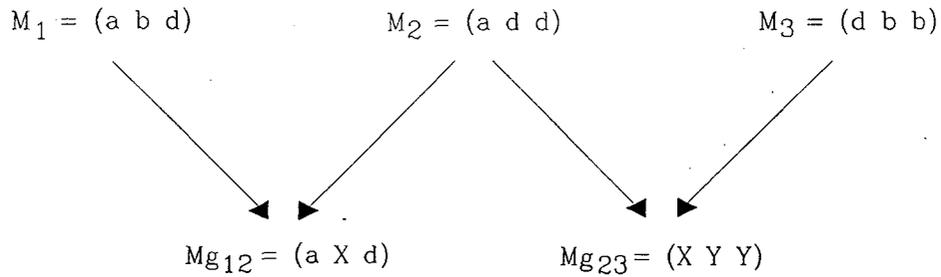


Figura 12. Exemplo de generalização de marcações

No procedimento de generalização usual, quando os valores são iguais eles não precisam ser substituídos por uma variável. No entanto, no nosso método a substituição por variável sempre será feita com o propósito de poder obter diretamente as *funções de transformação*.

O procedimento de generalização será apresentado passo-à-passo, iniciando pela generalização dos indivíduos, seguido das tuplas e finalizando com a generalização das marcações.

4.4.1 Generalização de indivíduos

Definição 4.17 Dados dois indivíduos a e a' , a variável V generaliza estes indivíduos.

Não utilizaremos explicitamente as variáveis pois estamos interessados não na marcação generalizada, mas na modificação que necessária para transformar uma marcação na outra. Cada generalização de dois indivíduos dá origem diretamente a uma função de transformação. Adotaremos a seguinte notação:

Notação 4.18 A generalização dos indivíduos a e a' será denotado por $a \setminus a'$.

Definição 4.19 Sejam a e a' dois indivíduos, e seja $a \setminus a'$ a sua generalização. Então a função $f: (a \rightarrow a')$ transforma o indivíduo a no indivíduo a' , e f é bijetora.

4.4.2 Composição consistente de generalizações de indivíduos

A generalização de indivíduos pode ser estendida para a generalização de dois grupos de indivíduos através da composição de suas generalizações.

Notação 4.20 O símbolo " \circ " representará a composição de generalizações.

Definição 4.21 Sejam $a_1 \setminus a_1', a_2 \setminus a_2' \dots a_n \setminus a_n'$ generalizações de indivíduos. Então a sua composição $CompI = a_1 \setminus a_1' \circ a_2 \setminus a_2' \circ \dots \circ a_n \setminus a_n'$ representa a generalização simultânea de todos os pares de indivíduos (a_i, a_i') .

Notação 4.22 Se $CompI = a_1 \setminus a_1' \circ a_2 \setminus a_2' \circ \dots \circ a_n \setminus a_n'$ então cada generalização $a_i \setminus a_i'$ será chamada de *componente* da composição $CompI$.

Em alguns casos, a composição pode levar a uma generalização de um indivíduo com diferentes indivíduos ao mesmo tempo. Se estes indivíduos estão em posições das tuplas que são *relacionadas por valor* então uma inconsistência aparece.

Exemplo 4.6 Considere as tuplas $Tp_1 = (a \ b)$ e $Tp_2 = (c \ c)$, e as generalizações $a \setminus c$ e $b \setminus c$ respectivas a primeira e segunda posição das tuplas. Se os indivíduos dessas posições aparecerem em uma fórmula seletora de uma transição, então uma inconsistência existe nessa composição de generalizações ($a \neq b$ mas $c = c$).

Definição 4.23 Sejam a_i e a_j dois indivíduos de posições das tuplas que estão *relacionadas por valor*. Uma composição $CompI = a_1 \setminus a_1' \circ a_2 \setminus a_2' \circ \dots \circ a_n \setminus a_n'$ é *consistente* sse $\forall a_i \setminus a_i'$ e $\forall a_j \setminus a_j'$, se $a_i = a_j$ então $a_i' = a_j'$ e reciprocamente.

Quando ocorre uma situação onde $a_i = a_j$ e $a_i' = a_j'$, então a composição pode ser simplificada pela eliminação de uma das generalizações $a_i \setminus a_i'$ iguais.

4.4.3 Generalização de tuplas

A partir deste conceito de composição consistente podemos definir a generalização de tuplas. A generalização de indivíduos sempre ocorre com indivíduos situados nas mesmas posições das tuplas.

Definição 4.24 Sejam duas tuplas $Tp_1 = \langle a_{11}, a_{12} \dots a_{1k} \rangle$ e $Tp_2 = \langle a_{21}, a_{22} \dots a_{2k} \rangle$, e as suas generalizações $a_{11} \setminus a_{21}, a_{12} \setminus a_{22} \dots a_{1k} \setminus a_{2k}$ (onde o índice i indica a tupla a qual os indivíduos a_{ij} pertencem e o índice j indica a sua posição dentro da tupla). As tuplas Tp_1 e Tp_2 podem ser generalizadas sse a composição $a_{11} \setminus a_{21} \circ a_{12} \setminus a_{22} \circ \dots \circ a_{1k} \setminus a_{2k}$ é consistente.

Notação 4.25 A generalização das tuplas Tp_1 e Tp_2 será denotada por $Tp_1 \setminus Tp_2$.

Da mesma forma que foi feito com a generalização de indivíduos, podemos obter uma função que transforma a tupla Tp_1 na tupla Tp_2 .

Definição 4.26 Sejam duas tuplas Tp_1 e Tp_2 generalizadas por $Tp_1 \setminus Tp_2 = a_{11} \setminus a_{21} \circ a_{12} \setminus a_{22} \circ \dots \circ a_{1k} \setminus a_{2k}$. A função $f: (a_{11} \rightarrow a_{21}, a_{12} \rightarrow a_{22} \dots a_{1k} \rightarrow a_{2k})$ transforma Tp_1 em Tp_2 , e f é bijetora.

4.4.4 Composição de generalizações de tuplas

A partir das generalizações de tuplas podemos obter uma composição destas, da seguinte forma:

Definição 4.27 Sejam $Tp_1 \setminus Tp_2 = a_{11} \setminus a_{21} \circ \dots \circ a_{1k} \setminus a_{2k}$ e $Tp_1' \setminus Tp_2' = a_{11}' \setminus a_{21}' \circ \dots \circ a_{1l}' \setminus a_{2l}'$ duas generalizações de tuplas. Então a sua composição $CompT = Tp_1 \setminus Tp_2 \circ Tp_1' \setminus Tp_2'$ corresponde a composição de suas generalizações de indivíduos $a_{11} \setminus a_{21} \circ \dots \circ a_{1k} \setminus a_{2k} \circ a_{11}' \setminus a_{21}' \circ \dots \circ a_{1l}' \setminus a_{2l}'$.

Definição 4.28 Uma composição de tuplas é consistente sse a composição de suas generalizações de indivíduos é consistente.

Exemplo 4.7 Considere-se as generalizações de tuplas $Tp_1 \setminus Tp_2 = a \setminus b \circ b \setminus d$ e $Tp_1' \setminus Tp_2' = d \setminus a \circ a \setminus c$. A consistência da composição $Tp_1 \setminus Tp_2 \circ Tp_1' \setminus Tp_2'$ deve ser verificada pela consistência de $a \setminus b \circ b \setminus d \circ d \setminus a \circ a \setminus c$. A partir da Definição 4.23 podemos concluir que $a \setminus b$ é inconsistente com $a \setminus c$, e portanto, a partir da Definição 4.28 podemos concluir também que $Tp_1 \setminus Tp_2 \circ Tp_1' \setminus Tp_2'$ é inconsistente.

4.4.5 Grupos de tuplas compatíveis

Queremos aqui lembrar que a nossa meta ao realizarmos a generalização é determinar a equivalência entre duas marcações. Sabemos também que as condições 1 e 2 do Teorema 4.15 devem ser satisfeitas para que possa existir uma equivalência entre duas marcações, e por isso, somente faz sentido a generalização de tuplas que estejam no mesmo lugar. Conseqüentemente podemos dividir as tuplas em grupos dentro dos quais as generalizações de tuplas devem ocorrer. Estes grupos e a sua similaridade são definidos a seguir.

Definição 4.29 Para uma marcação M_x os grupos de tuplas compatíveis $G(p)$ são definidos como sendo os conjuntos de tuplas que estão no mesmo lugar p , e são denotadas por:

$$G(p) = \{Tp_1, Tp_2 \dots Tp_n\} \mid \forall Tp_i \text{ e } \forall Tp_j \in G, p(Tp_i) = p(Tp_j).$$

Definição 4.30 Sejam $G(p_x) = \{Tp_1, Tp_2 \dots Tp_k\}$ e $G(p_y) = \{Tp_1', Tp_2' \dots Tp_l'\}$ dois grupos de tuplas compatíveis obtidas respectivamente a partir das marcações M e M' . Sejam k e l o número de tuplas em cada grupo. Então dizemos que os grupos são similares sse $p_x = p_y$ e $k = l$.

4.4.6 Forma normal da marcação

A partir desta definição podemos passar a representar as marcações das redes em uma forma normalizada, onde as tuplas estão organizadas por lugar.

Definição 4.31 Uma marcação M_x de uma rede R está na *forma normal* se as tuplas da marcação estão ordenadas por lugar conforme o padrão abaixo:

$$M = ((p_1 Tp_{11} Tp_{12} \dots Tp_{1h}) (p_2 Tp_{21} Tp_{22} \dots Tp_{2k}) \dots (p_n Tp_{n1} Tp_{n2} \dots Tp_{nl})) \mid \forall i = 1, 2 \dots n, Tp_{ij} \text{ está contida no lugar } p_i .$$

Exemplo 4.8 A marcação inicial da Figura 5 representada na *forma normal* é dada por:

$$M_0 = ((p_1 <0> <1> <2> <3> <4>) (p_2) (p_3 <0> <1> <2> <3> <4>))$$

4.4.7 A generalização de grupos de tuplas

Nos lugares das Redes de Petri as tuplas não estão organizadas. Conseqüentemente, a ordem em que as generalizações de tuplas são realizadas dentro de cada grupo não é importante. Por esta razão, a generalização dos grupos de tuplas deve levar em conta todas as possíveis combinações, resultando em mais de uma solução.

Exemplo 4.9 Considere-se os grupos de tuplas compatíveis $G = \{Tp_1, Tp_2\}$ e $G' = \{Tp_1', Tp_2'\}$, onde $Tp_1, Tp_2 \in M$ e $Tp_1', Tp_2' \in M'$. Neste caso as generalizações de tuplas que devem ser analisadas são $Tp_1 \setminus Tp_1', Tp_1 \setminus Tp_2', Tp_2 \setminus Tp_1'$ e $Tp_2 \setminus Tp_2'$. Como as tuplas Tp_1 e Tp_2 (Tp_1' e Tp_2') pertencem à mesma marcação, a análise da generalização $Tp_1 \setminus Tp_2$ ($Tp_2 \setminus Tp_2'$) não é necessária, uma vez que o objetivo de realizarmos a generalização de duas marcações é o de obter as funções de transformação para duas marcações dadas.

Estas generalizações são realizadas através da construção de uma matriz que contém todas as combinações de tuplas entre os dois grupos.

Definição 4.32 Sejam dois grupos similares de tuplas compatíveis $G = \{Tp_1, Tp_2 \dots Tp_k\}$ e $G' = \{Tp_1', Tp_2' \dots Tp_k'\}$. Então a Matriz de Generalização MG de ordem $k * k$ é definida por:

MG	Tp_1'	Tp_2'	...	Tp_k'
Tp_1	mg_{11}	mg_{12}	...	mg_{1k}
Tp_2	mg_{21}	mg_{22}	...	mg_{2k}
\vdots	\vdots	\vdots	\ddots	\vdots
Tp_k	mg_{k1}	mg_{k2}	...	mg_{kk}

se Tp_i e Tp_j' podem ser generalizadas então $mg_{ij} = Tp_i \setminus Tp_j'$, senão $mg_{ij} = 0$.

A partir das generalizações de tuplas dentro de um grupo podemos obter o conjunto das composições válidas dessas generalizações.

Definição 4.33 Uma composição de generalizações de tuplas é denominada *válida* se ela contém uma e somente uma tupla de cada grupo.

Exemplo 4.10 No exemplo anterior as composições válidas são $Tp_1 \setminus Tp_1' \circ Tp_2 \setminus Tp_2'$ e $Tp_1 \setminus Tp_2' \circ Tp_2 \setminus Tp_1'$. A composição $Tp_1 \setminus Tp_1' \circ Tp_1 \setminus Tp_2'$ não é válida pois a tupla Tp_1 é representada duas vezes e a tupla Tp_2 não é representada. O mesmo ocorre com a composição $Tp_2 \setminus Tp_1' \circ Tp_2 \setminus Tp_2'$.

Definição 4.34 Sejam G e G' dois grupos similares de tuplas compatíveis. Então suas *generalizações* são as *composições válidas consistentes* das generalizações de tuplas obtidas em cada elemento mg_{ij} da matriz MG . O conjunto destas generalizações será denotado por $\{G \setminus G'\}$.

Pelo menos uma composição consistente deve existir para que dois grupos de tuplas compatíveis possam ser generalizados. Por isso podemos afirmar que:

Teorema 4.35 Os grupos G e G' podem ser generalizados sse $\{G \setminus G'\} \neq \emptyset$.

Devido a necessidade de cada tupla de cada grupo ser generalizada com uma tupla do outro grupo, podemos concluir que:

Corolário 4.36 Se a matriz MG tem alguma linha ou coluna nula, então G e G' não podem ser generalizadas.

A partir das definições acima, podemos determinar o número limite de composições válidas.

Teorema 4.37 Se nenhum elemento mg_{ij} da matriz MG for nulo, então o número de composições válidas (consistentes ou não) será $N!$, onde N é o número de tuplas em cada grupo.

4.4.8 A generalização de marcações

Uma vez que as marcações tenham sido divididas em grupos de tuplas compatíveis e as generalizações desses grupos já foram obtidas, podemos encontrar a generalização dessas marcações da seguinte forma:

Definição 4.38 Sejam M e M' duas marcações que satisfazem as condições 1 e 2 do Teorema 4.15. Sejam $\{G_1, G_2 \dots G_k\}$ e $\{G_1', G_2' \dots G_k'\}$ grupos de tuplas compatíveis correspondentes a M_1 e M_2 respectivamente. Sejam ainda $\{G_1 \setminus G_1'\}$, $\{G_2 \setminus G_2'\} \dots \{G_k \setminus G_k'\}$ os conjuntos de generalizações correspondentes aos

grupos similares das duas marcações. Então as generalizações das marcações M e M' são dadas pelo conjunto de composições $\{M \setminus M'\} = \{Comp_1 M, Comp_2 M \dots Comp_k M\}$, onde cada $Comp_i M$ é a composição consistente de uma composição de tuplas ($CompT$) de cada $\{G_i \setminus G'_i\}$.

Corolário 4.39 As marcações M e M' podem ser generalizadas por qualquer uma das composições $Comp_i M \in \{M \setminus M'\}$.

Exemplo 4.11 Considere-se as marcações $M = \{G_1, G_2\}$ e $M' = \{G'_1, G'_2\}$. Considere-se também que as condições 1 e 2 do Teorema 4.15 são satisfeitas e que as generalizações de grupos são $\{G_1 \setminus G'_1\} = \{a \setminus b \circ b \setminus c, b \setminus a\}$ e $\{G_2 \setminus G'_2\} = \{b \setminus a\}$. Então as composições a serem analisadas são $Comp_1 M = a \setminus b \circ b \setminus c \circ b \setminus a$ e $Comp_2 M = b \setminus a \circ b \setminus a$. Como $Comp_1 M$ é inconsistente, concluímos que $\{M \setminus M'\} = \{b \setminus a\}$.

Uma vez determinadas as generalizações das marcações M_1 e M_2 , podemos afirmar que:

Teorema 4.40 Sejam M e M' duas marcações. A terceira condição do Teorema 4.15 é verdadeira sse $\{M \setminus M'\} \neq \emptyset$.

Corolário 4.41 Se $\{M \setminus M'\} = \emptyset$ então M e M' não são equivalentes.

Para encontrar a função que transforma a marcação M na marcação M' , devemos estender a definição dada para o caso das tuplas da seguinte forma:

Definição 4.42 Sejam M e M' duas marcações e uma composição $CompM = a_1 \setminus a'_1 \circ a_2 \setminus a'_2 \circ \dots \circ a_k \setminus a'_k$. Então $\forall CompM \in \{M \setminus M'\}$, $\exists f: (a_1 \rightarrow a'_1, a_2 \rightarrow a'_2 \dots a_k \rightarrow a'_k)$ bijetora que transforma a marcação M na marcação M' (i.e. $f(M) = M'$ e $f^{-1}(M') = M$).

4.4.9 Algoritmo de generalização

A partir das definições acima é possível determinar a generalização de duas marcações M e M' . No entanto, esta tarefa pode ser um pouco árdua para marcações com grande quantidade de indivíduos. Por isso estabelecemos aqui algumas regras que tornam o procedimento de generalização de duas marcações M e M' mais rápido e eficiente.

REGRA 1 A generalização de duas marcações M e M' deve ser sempre iniciada com o menor grupo de tuplas compatíveis, com o objetivo de diminuir a complexidade da generalização de grupos de tuplas (veja o Teorema 4.37)

REGRA 2 Sejam dois grupos de tuplas $G \in M$ e $G' \in M'$. Se $\{G \setminus G'\}$ tem mais de uma composição e alguma das generalizações de indivíduos $a \setminus a'$ é comum a todas as composições,

então esta generalização $a \setminus a'$ é *obrigatória*, e deve ser aplicada a todos os indivíduos contidos nas posições das tuplas dos outros grupos de tuplas que pertencem ao mesmo subconjunto de posições relacionadas por valor (*SPRV*). Esta regra é devida a necessidade de consistência nas composições de marcações (veja a Definição 4.38).

REGRA 3 Da regra anterior podemos concluir que se $\{G \setminus G'\}$ tem apenas uma composição então todas as suas generalizações de indivíduos desta composição são obrigatórias.

REGRA 4 Uma vez conhecendo-se um conjunto de generalizações obrigatórias, todas as generalizações obtidas das marcações M e M' devem ser consistentes com as generalizações obrigatórias (ver a Definição 4.38).

Baseado nestas regras, definições e teoremas anteriormente descritos, sugerimos um procedimento que permite encontrar as generalizações de duas marcações M_1 e M_2 .

Algoritmo 1

ENTRADA: . As marcações M_1 e M_2 descritas na sua forma normal, e satisfazendo as condições 1 e 2 do Teorema 4.15.

SAÍDA: . O conjunto de generalizações $\{M_1 \setminus M_2\}$.

NOTAÇÕES:

$Gen1$. Generalizações obrigatórias já usadas
 $Gen2$. Generalizações obrigatórias ainda não usadas
 $\{MG\}$. Conjunto das matrizes MG não resolvidas
 $\{OMG\}$. Conjunto com as ordens das matrizes MG não resolvidas
 $NTpGi$. Número de tuplas por grupo que ainda precisam ser generalizadas
 G . Grupo de tuplas compatíveis

Procedimento Generalização-de-Marcações (Entr. M_1, M_2 ; Sai. $\{M_1 \setminus M_2\}$)

* PASSO 0

. $Gen1 = \emptyset$; $Gen2 = \emptyset$;

PARA TODO par (G_{1x}, G_{2x}) , onde $G_{1x} \in M_1$ e $G_{2x} \in M_2$

.Construa a matriz MG_x ; Determine $NTpG_x$;

. $\{MG\} = \{MG_1, MG_2 \dots MG_k\}$; $\{OMG\} = \{NTpG_1, NTpG_2 \dots NTpG_k\}$;

ENQUANTO $\{MG\} \neq \emptyset$

.Selecione a matriz MG_x com o menor $NTpG_x$;

* PASSO 1

.Determine $\{G_{1x} \setminus G_{2x}\}$;

SE $\{G_{1x} \setminus G_{2x}\} = \emptyset$ ENTÃO SAÍDA 1

. $Gen2 = \{a \setminus a' \mid a \setminus a' \text{ é uma componente de } Comp_i G, \forall Comp_i G \in \{G_{1x} \setminus G_{2x}\}\}$

*** PASSO 2**

PARA TODO $a \setminus a' \in Gen2$

.Remova $a \setminus a'$ de $\{G_{1x} \setminus G_{2x}\}$;

.Remova MG_x de $\{MG\}$; Remova $NTpG_x$ de $\{OMG\}$;

ENQUANTO $Gen2 \neq \emptyset$

PARA TODO $MG_x \in \{MG\}$

.Zere todos mg_{ij} que correspondem às generalizações de tuplas que são inconsistentes com as generalizações $a \setminus a' \in Gen2$, respeitando os subgrupos *SPRV*;

.Inclua $Gen2$ em $Gen1$; $Gen2 = \emptyset$;

PARA TODO $MG_x \in \{MG\}$

*** PASSO 2.1**

PARA TODA $coluna_j$ e $linha_i$ de MG_x pegue mg_{ij}

SE $NTpG_x = 0$

ENTÃO Remova MG_x de $\{MG\}$; Remova $NTpG_x$ de $\{OMG\}$;

SENÃO

CASO o número de mg_{ij} não nulos =

0 ENTÃO SAÍDA 1;

1 ENTÃO

.Encontre a $\{Tp_1 \setminus Tp_2\}$ correspondente a mg_{ij} ;

.Elimine a $coluna_j$ e $linha_i$;

$NTpG_x = NTpG_x - 1$;

.Inclua $\{Tp_1 \setminus Tp_2\} - Gen1$ em $Gen2$;

FIM CASO;

FIM SE;

SE $\{MG\} = \emptyset$ ENTÃO Inclua $Gen2$ em $Gen1$; $Gen2 = \emptyset$;

FIM PARA;

FIM PARA;

FIM ENQUANTO;

FIM ENQUANTO;

.Encontre $\{M_1 \setminus M_2\}$ a partir das composições dos elementos de $Gen1$ com as composições de $\{G_{1x} \setminus G_{2x}\} \neq \emptyset$;

FIM;

SAÍDA 1

$\{M_1 \setminus M_2\} = \emptyset$;

Aplicação do algoritmo de generalização de marcações em um exemplo

Para facilitar o entendimento da generalização de marcações, descreveremos a generalização de duas marcações pertencentes ao conjunto de marcações alcançáveis de M_0 da rede de Petri modelada na Figura 5. Consideremos as marcações:

$$M_1 = ((p_1 \langle 0 \rangle \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle) (p_2 \langle 2 \ 1 \ 1 \rangle) (p_3 \langle 0 \rangle \langle 3 \rangle \langle 4 \rangle)) \text{ e}$$

$$M_2 = ((p_1 \langle 1 \rangle \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle) (p_2 \langle 1 \ 0 \ 0 \rangle) (p_3 \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle))$$

obtidas por $M_0 [t_1.sh_1 > M_1$ e $M_0 [t_1.sh_2 > M_2$, onde $sh_1: (2 \rightarrow p_d, 1 \rightarrow p_e, 1 \rightarrow f)$ e $sh_2: (1 \rightarrow p_d, 0 \rightarrow p_e, 0 \rightarrow f)$. Estas marcações satisfazem a condição 1 ($|M_1| = |M_2|$) e a condição 2 ($|SH(M_1)| = |SH(M_2)|$) do teorema 4.15, e todas as posições de tupla pertencem ao mesmo subgrupo *SPRV*.

Para generalizar estas marcações serão construídas as três matrizes de generalização MG_1 , MG_2 e MG_3 respectivamente contendo as tuplas dos lugares p_1, p_3 e p_2 .

A generalização da marcação é iniciada pela matriz MG_3 , da qual obtemos a generalização de tuplas $\{1 \setminus 2 \circ 0 \setminus 1\}$ que por ser obrigatória (ver REGRA 3), permite eliminar os elementos $mg_{11} = 1 \setminus 0, mg_{13} = 1 \setminus 3, mg_{14} = 1 \setminus 4, mg_{22} = 2 \setminus 2, mg_{32} = 3 \setminus 2$ e $mg_{42} = 4 \setminus 2$ de MG_1 , por serem todos inconsistentes com a generalização de valores $1 \setminus 2$. Como $mg_{12} = 1 \setminus 2$ é consistente com $1 \setminus 2 \circ 0 \setminus 1$, podemos eliminar a linha 1 e a coluna 2 dessa matriz mudando a sua ordem para 3.

MG_1	$\langle 0 \rangle$	$\langle 2 \rangle$	$\langle 3 \rangle$	$\langle 4 \rangle$
$\langle 1 \rangle$	$1 \setminus 0$	$1 \setminus 2$	$1 \setminus 3$	$1 \setminus 4$
$\langle 2 \rangle$	mg_{21}	$2 \setminus 2$	mg_{23}	mg_{24}
$\langle 3 \rangle$	mg_{31}	$3 \setminus 2$	mg_{33}	mg_{34}
$\langle 4 \rangle$	mg_{41}	$4 \setminus 2$	mg_{43}	mg_{44}

MG_2	$\langle 0 \rangle$	$\langle 3 \rangle$	$\langle 4 \rangle$
$\langle 2 \rangle$	mg_{11}	mg_{12}	mg_{13}
$\langle 3 \rangle$	mg_{21}	mg_{22}	mg_{23}
$\langle 4 \rangle$	mg_{31}	mg_{32}	mg_{33}

MG_3	$\langle 2 \ 1 \ 1 \rangle$
$\langle 1 \ 0 \ 0 \rangle$	$1 \setminus 2 \circ 0 \setminus 1$

Na seqüência do algoritmo são feitas as generalizações das tuplas da matriz MG_1 ou MG_2 , que são idênticas. Como resultado obtemos as generalizações mostradas abaixo:

MG	$\langle 0 \rangle$	$\langle 3 \rangle$	$\langle 4 \rangle$
$\langle 2 \rangle$	$2 \setminus 0$	$2 \setminus 3$	$2 \setminus 4$
$\langle 3 \rangle$	$3 \setminus 0$	$3 \setminus 3$	$3 \setminus 4$
$\langle 4 \rangle$	$4 \setminus 0$	$4 \setminus 3$	$4 \setminus 4$

Dessa matriz determinamos a generalização desse grupo como sendo o conjunto das composições válidas $G_2 \setminus G_1 = \{2 \setminus 0 \circ 3 \setminus 3 \circ 4 \setminus 4, 2 \setminus 0 \circ 3 \setminus 4 \circ 4 \setminus 3, 2 \setminus 3 \circ 3 \setminus 0 \circ 4 \setminus 4, 2 \setminus 4 \circ 3 \setminus 0 \circ 4 \setminus 3, 2 \setminus 3 \circ 3 \setminus 4 \circ 4 \setminus 0, 2 \setminus 4 \circ 3 \setminus 3 \circ 4 \setminus 0\}$.

Ao término do algoritmo obteremos as generalizações das marcações $\{M_2 \setminus M_1\} = \{0 \setminus 1 \circ 1 \setminus 2 \circ 2 \setminus 0 \circ 3 \setminus 3 \circ 4 \setminus 4, 0 \setminus 1 \circ 1 \setminus 2 \circ 2 \setminus 0 \circ 3 \setminus 4 \circ 4 \setminus 3, 0 \setminus 1 \circ 1 \setminus 2 \circ 2 \setminus 3 \circ 3 \setminus 0 \circ 4 \setminus 4, 0 \setminus 1 \circ 1 \setminus 2 \circ 2 \setminus 4 \circ 3 \setminus 0 \circ 4 \setminus 3, 0 \setminus 1 \circ 1 \setminus 2 \circ 2 \setminus 3 \circ 3 \setminus 4 \circ 4 \setminus 0, 0 \setminus 1 \circ 1 \setminus 2 \circ 2 \setminus 4 \circ 3 \setminus 3 \circ 4 \setminus 0\}$, que dão origem a seis funções de transformação f_i tal que $f_i(M_2) = M_1$.

4.5 PASSO 4 - Verificação da simetria das funções

Uma vez obtidas as funções de transformações, devemos determinar quais delas são simétricas.

Definição 4.43 Uma função $f: (M \rightarrow M')$ é simétrica se ela não altera o comportamento possível a partir destas marcações. Isto é, $Sub\grave{a}rvore(M)$ é análogo a $Sub\grave{a}rvore(M')$.

Uma função simétrica quando aplicada sobre as marcações e transições da subárvore com raiz em M , transforma esta subárvore na subárvore com raiz em M' .

Corolário 4.44 Se f é uma função simétrica então $f(Sub\grave{a}rvore(M)) = Sub\grave{a}rvore(M')$.

Uma vez definido o significado da simetria da função, podemos utilizá-la para determinar a equivalência entre a marcações.

Teorema 4.45 Sejam duas marcações $M, M' \in [M_0 >$, e $\{M \setminus M'\}$ o conjunto das suas generalizações. Então $M \sim M' \Leftrightarrow \exists f_i$ obtida de $\{M \setminus M'\} \mid f_i$ é simétrica.

O teorema acima fornece a *condição necessária e suficiente* para determinar a equivalência entre duas marcações que são alcançáveis a partir da marcação inicial M_0 .

4.5.1 Caso particular - Rede com Árvore de Alcançabilidade Diretamente Redutível (RDR)

Em alguns casos particulares a terceira condição do Teorema 4.15 é suficiente para a equivalência de marcações. Para estes casos definiremos um nova subclasse de Rede Pr/T.

Definição 4.46 Uma rede Pr/T é denominada de *Rede com Árvore de Alcançabilidade Diretamente Redutível (RDR)*, se as seguintes condições são satisfeitas:

- 1) Nenhuma transição modifica os indivíduos das posições das tuplas;
- 2) Todas as fórmulas seletoras A_T são comparações de indivíduos na forma de relações de *igualdade* ($I_1 = I_2$) ou *desigualdade* ($I_1 \neq I_2$).

Teorema 4.47 Seja M_0 a marcação inicial de uma *RDR*, e duas marcações $M, M' \in [M_0>$. Então $M \sim M' \Leftrightarrow \{M \setminus M'\} \neq \emptyset$.

As redes que pertencem a esta subclasse são detectadas automaticamente por uma análise de transições.

Uma vez que a generalização de duas marcações é determinada, as funções de simetria podem também ser obtidas de um modo automático.

Teorema 4.48 Sejam as marcações M e M' de uma *RDR*. Se M e M' podem ser generalizadas então todas as funções f_i obtidas a partir de $\{M \setminus M'\}$ são simétricas.

Assim a determinação da equivalência e das funções simétricas para as *RDR* está resolvido de forma completamente automatizada, uma vez que todas as funções de transformação encontradas são simétricas.

4.5.2 Caso geral

No caso geral das Redes Pr/T necessitamos de um meio de encontrar as simetrias das funções obtidas a partir das generalizações das marcações. Esta informação pode ser obtida a partir da estrutura da rede e das marcações. No método proposto em [HUBE 86] e em [LIND 89] esta tarefa fica por conta do usuário, que deve definir previamente as funções de simetria. O método proposto por estes autores limita o conjunto de funções simétricas que podem ser usadas àquelas que existem na marcação inicial (simetrias totais).

Na nossa abordagem as funções de simetria são definidas de forma interativa durante a construção da AAR permitindo o uso das simetrias totais e parciais. Entretanto, o usuário apenas precisa verificar quais das funções de transformações encontradas automaticamente pelo Algoritmo 1 são simétricas.

Essa verificação pode ser feita utilizando-se por exemplo a inversão da aplicação das funções de simetria e do disparo das transições como em [HUBE 86].

4.6 Algoritmo para determinação da equivalência de marcações no caso geral das redes Pr/T

A seguir apresentaremos um algoritmo que permite determinar a equivalência de uma marcação M_x com as marcações da AAR, durante o processo de redução.

Para utilizar eficientemente este algoritmo é necessário que as marcações da AAR estejam divididas em subgrupos SG_x de marcações com a mesma cardinalidade $|M_{SG}|$ e o mesmo número de substituições de habilitação $|SH(M_{SG})|$.

OBS: Para as redes da subclasse RDR todas as funções simétricas são obtidas diretamente a partir de $\{M_x \setminus M_y\}$ e este algoritmo não é necessário.

Algoritmo 2

ENTRADA: . A marcação M_x descrita na forma normal.
 . A lista $Nós = \{SG_1, SG_2 \dots SG_l\}$.

SAÍDA: . A marcação $M_y \in Nós$ que é equivalente a M_x
 . O conjunto de funções simétricas $FS = \{f_1, f_2 \dots f_n\}$ associadas a M_x . Se $FS = \emptyset$ então M_x não é equivalente a nenhuma das marcações da lista $Nós$.

Procedimento Determinação-da-Equivalência-de-Marcações (Entr: $M_x, Nós$; Sai: M_y, FS)

* PASSO 0

$FS = \emptyset$;
 .Determinar $|M_x|$ e $|SH(M_x)|$;
 .SE $\exists SG_i \in Nós \mid |M_x| = |M_{SG}|$ e $|SH(M_x)| = |SH(M_{SG})|$;
 .ENTÃO
 .PARA TODO $M_y \in SG_i$

* PASSO 1

.Procedimento Generalização-de-Marcações (Entr: M_x, M_y , Sai: $\{M_x \setminus M_y\}$);
 .SE $\{M_x \setminus M_y\} \neq \emptyset$
 .ENTÃO

* PASSO 2

.PARA TODA f_i obtida a partir de $\{M_x \setminus M_y\}$
 .Determinar se f_i é simétrica (INTERVENÇÃO DO USUÁRIO);
 .SE f_i é simétrica ENTÃO Inclua f_i em FS ;
 .FIM PARA;

.SE $FS \neq \emptyset$ ENTÃO FIM;

.FIM SE;

.FIM PARA;

.FIM SE;

.FIM;

Aplicação do algoritmo de determinação da equivalência de marcações em um exemplo

Utilizaremos novamente o exemplo da Figura 5, para o qual consideraremos que a AAR está parcialmente construída, com a lista $Nós = \{(SG_1 M_0) (SG_2 M_1)\}$, onde $|M| = (5 \ 0 \ 5)$ e $|SH(M)| = (5 \ 0)$ para SG_1 e $|M| = (4 \ 1 \ 3)$ e $|SH(M)| = (2 \ 1)$ para SG_2 .

Queremos determinar se existe alguma marcação armazenada na lista $Nós$, que seja equivalente a marcação M_2 . As marcações M_0 , M_1 e M_2 representadas na forma normal são mostradas abaixo:

$$M_0 = ((p_1 \langle 0 \rangle \langle 1 \rangle \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle) (p_2) (p_3 \langle 0 \rangle \langle 1 \rangle \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle))$$

$$M_1 = ((p_1 \langle 0 \rangle \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle) (p_2 \langle 2 \ 1 \ 1 \rangle) (p_3 \langle 0 \rangle \langle 3 \rangle \langle 4 \rangle))$$

$$M_2 = ((p_1 \langle 1 \rangle \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle) (p_2 \langle 1 \ 0 \ 0 \rangle) (p_3 \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle))$$

Inicialmente determinamos $|M_2| = (4 \ 1 \ 3)$ e $|SH(M_2)| = (2 \ 1)$, e verificamos que M_2 só pode ser equivalente as marcações do subgrupo SG_2 (conforme o Teorema 4.15).

Em seguida é determinada a generalização da marcação M_2 com as marcações do subgrupo SG_2 (no caso apenas M_1). Como já foi no exemplo de aplicação do Algoritmo 1, $\{M_2 \setminus M_1\} = \{0 \setminus 1 \circ 1 \setminus 2 \circ 2 \setminus 0 \circ 3 \setminus 3 \circ 4 \setminus 4, 0 \setminus 1 \circ 1 \setminus 2 \circ 2 \setminus 0 \circ 3 \setminus 4 \circ 4 \setminus 3, 0 \setminus 1 \circ 1 \setminus 2 \circ 2 \setminus 3 \circ 3 \setminus 0 \circ 4 \setminus 4, 0 \setminus 1 \circ 1 \setminus 2 \circ 2 \setminus 4 \circ 3 \setminus 0 \circ 4 \setminus 3, 0 \setminus 1 \circ 1 \setminus 2 \circ 2 \setminus 3 \circ 3 \setminus 4 \circ 4 \setminus 0, 0 \setminus 1 \circ 1 \setminus 2 \circ 2 \setminus 4 \circ 3 \setminus 3 \circ 4 \setminus 0\}$ que correspondem a seis funções de transformação f_1, f_2, f_3, f_4, f_5 e f_6 respectivamente.

Em seguida é solicitado que o usuário determine quais dessas funções são simétricas para as marcações M_1 e M_2 . Pela análise do comportamento da rede o usuário pode verificar que somente a função de transformação $f_5 = \{0 \rightarrow 1, 1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 4, 4 \rightarrow 0\}$ é simétrica.

Assim como resultado do procedimento obtemos que M_1 é equivalente a M_2 , e $FS = \{f_5\}$.

4.7 Conclusão

Neste capítulo formalizamos um método que possibilita a determinação da relação de equivalência entre duas marcações que pertençam ao conjunto de marcações alcançáveis da rede (R, M_0) . Esta determinação é a chave para a redução da árvore de alcançabilidade sem que se perca a sua capacidade de análise, uma vez que evita a construção das subárvores análogas para comparar seus comportamentos. Mostramos também que essa determinação é equivalente a determinação das funções de simetria.

O método proposto apresenta várias vantagens em relação ao método proposto em [HUBE 86] que também preserva a mesma capacidade de análise da árvore de alcançabilidade completa:

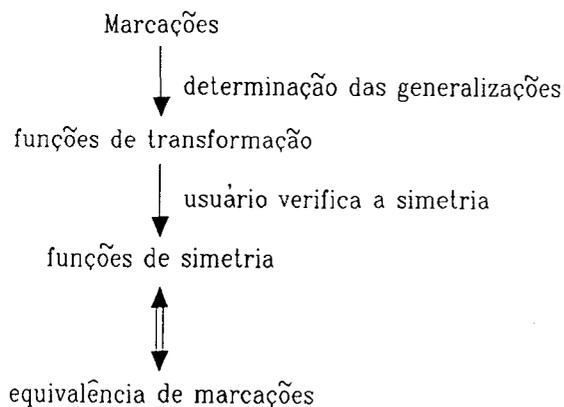
. As simetrias parciais da rede também são utilizadas na redução da AA.

- . A classificação das posições das tuplas é realizada pela análise das transições sem a intervenção do usuário.
- . No caso geral das redes Pr/T a tarefa do usuário na determinação das funções de simetria é simplificada, uma vez que ela consiste em apenas verificar a simetria de um conjunto de funções dado.

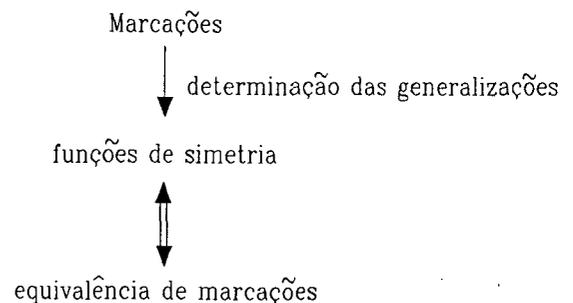
Introduzimos também neste capítulo o uso da *cardinalidade do conjunto de substituições de habilitação* para a separação das marcações, diminuindo assim consideravelmente o número de marcações a serem comparadas a cada nova marcação gerada. Esta melhoria pode ser aplicada diretamente aos métodos citados que utilizam as simetrias da rede para a redução.

No caso particular da subclasse de redes Pr/T chamadas de *Redes com Árvore de Alcançabilidade Diretamente Redutível (RDR)* definidas neste capítulo, a *determinação das funções de simetria é realizada de forma automática* pelo procedimento de *Generalização-de-Marcações*.

Resumindo, a obtenção da equivalência entre duas marcações é realizada de acordo com um dos seguintes esquemas:



Caso Geral (Pr/T)



Caso Particular (RDR)

No próximo capítulo utilizaremos essa determinação de equivalência entre marcações, na obtenção da AAR.

CAPÍTULO V

A ÁRVORE DE ALCANÇABILIDADE REDUZIDA (AAR)

5.1 Introdução

Neste capítulo descrevemos inicialmente como será construída a árvore de alcançabilidade reduzida (AAR) usando a equivalência de marcações conforme visto no capítulo anterior, e apresentamos um algoritmo para construí-la. Também é descrito um procedimento simples que transforma a AAR em um grafo de alcançabilidade reduzido (GAR).

A seguir nos interessa demonstrar a equivalência entre a AAR e a AA. Para tal descrevemos como pode ser determinada a alcançabilidade das marcações da rede a partir da AAR. Também é mostrado que na AAR estão presentes todas as seqüências de disparo de transições da árvore de alcançabilidade completa (AA). Finalmente, mostramos como recuperar as informações referentes aos indivíduos capturados pelas transições das seqüências de disparo.

5.2 O método de construção da árvore de alcançabilidade reduzida (AAR)

Para construir a árvore de alcançabilidade reduzida (AAR), estruturamos o conjunto de marcações alcançáveis ($\{M_0\}$) a partir da marcação inicial M_0 em forma de árvore. Uma *marcação* (ou uma *anotação* no caso do nó reduzido) é associada a cada nó da árvore, e uma *transição* bem como os *indivíduos* da *substituição de habilitação* dessa transição são associados a cada arco direcionado, o qual representa a transformação que leva da marcação associada ao seu nó origem para a marcação associada ao seu nó destino.

Esta árvore contém parte das marcações alcançáveis a partir de M_0 e todas as seqüências de disparo de transição, uma vez que todos os comportamentos possíveis estão representados na AAR. Pela sua inspeção podemos responder a um grande número de questões sobre o comportamento do sistema modelado, e também realizar a validação do modelo através da análise das suas propriedades gerais e da verificação das propriedades específicas.

Neste trabalho lidaremos apenas com redes limitadas e por isso utilizaremos a *relação de cobertura na sua interpretação estrita* para a geração da árvore, conforme definido no Capítulo 2 (Definição 2.20).

A definição da árvore de alcançabilidade apresentada no Capítulo 3 (Definição 3.1), precisa agora ser modificada para incluir a redução pelo uso das *relações de equivalência*.

Definição 5.1 Seja (R, M_0) uma rede Pr/T, sh a substituição de habilitação da transição t , e FS o conjunto de funções de simetria que permitem transformar uma marcação M_x na marcação M_y quando $M_x \sim M_y$. Então a árvore de alcançabilidade reduzida de (R, M_0) será denotada por $AAR = (Nós, Arcos)$ onde:

- 1) A marcação inicial $M_0 \in Nós$.
- 2) $\forall M \in Nós \exists t.sh \mid M [t.sh > M'$:
 - a) Se $\neg \exists M_x \in Nós \mid M' \sim M_x$ então
 $M' \in Nós$ e o rótulo $(M, t.sh, M') \in Arcos$.
 - b) Se $\exists M_x \in Nós \mid M' \sim M_x$ então
 $nó_x \in Nós$, o rótulo $(M, t.sh, nó_x) \in Arcos$ e $nó_x$ tem a anotação $(\sim M_x, FS)$ onde $FS = \{fs_1, fs_2 \dots fs_n\} \mid \forall fs_i \Rightarrow fs_i(M') = M_x$.
- 3) $\forall M \in Nós$ e $\neg \exists t.sh \mid M [t.sh >$ então M tem o rótulo "poço".

Na definição acima, a noção de nós duplicados foi substituída pela de nós reduzidos.

5.2.1 Tipos de nós da AAR

Para podermos identificar melhor os nós da AAR, classificamos estes nós nas seguintes categorias: *nó poço*, *nó interno*, *nó terminal*, *nó principal*, *nó duplicado* e *nó reduzido*. Estas categorias de nós são utilizadas durante a construção da AAR e o significado de cada uma delas é descrita a seguir.

Nó poço

O *nó poço* é um nó que possui associado a ele uma *marcação poço*, na qual nenhuma transição está habilitada. Nas representações gráficas este tipo de nó será denotado por "P".

Nó interno

O nó ao qual está associada uma marcação, cujo *conjunto de substituições de habilitação* ($SH(M_x)$) já foi explorado é chamado de *nó interno*. Um *nó interno* pode ser ao mesmo tempo um *nó principal*. Nas representações gráficas este tipo de nó será denotado por " ".

Nó terminal

Um nó terminal é aquele que possui associado a ele uma marcação que ainda não teve todas as suas *substituições de habilitação* exploradas. Este tipo de nó só existe durante a execução do algoritmo e corresponde aos nós armazenados na lista *Nós-a-explorar*.

Nó principal

A raiz da subárvore que é análoga as subárvores removidas é denominada de *nó principal*. A marcação raiz é associada a este nó. Nas representações gráficas este tipo de nó será denotado por "M".

Nó reduzido e nó duplicado

O nó reduzido corresponde à raiz da subárvore removida. Este nó não tem associado a ele nenhuma marcação, mas somente uma *anotação* (M_x, FS), onde M_x indica a marcação correspondente ao nó principal da AAR que lhe é equivalente, e $FS = \{fs_1, fs_2 \dots fs_n\}$ indica o conjunto de funções simétricas. Tais funções permitem a recuperação da subárvore removida usando como base a subárvore cuja raiz é M_x (denotada por $Subárvore(M_x)$) construída na AAR. Nas representações gráficas este tipo de nó será denotado por "*".

O *nó duplicado* é um caso especial deste tipo de nó, onde a função de simetria fs é igual a função identidade ID ¹.

5.2.2 Algoritmo para construção da árvore de alcançabilidade reduzida**a) Descrição geral**

A AAR é construída a partir da marcação inicial (M_0), explorando-se em cada *nó terminal* todo o conjunto de *substituições de habilitação* e criando-se assim novos nós. A cada novo nó criado verifica-se se este corresponde a uma *marcação poço* ou a uma *marcação equivalente* já existente na AAR. Se o nó não possuir nenhum desses dois tipos de marcação, então ele será utilizado como nó terminal. A terminação da construção da árvore ocorre quando não existir mais nenhum *nó terminal*.

Esse algoritmo de construção da AAR determinará as *simetrias totais e parciais* durante a redução, usando como base o procedimento de *Generalização-de-Marcações* (ver Seção 4.4). A determinação dessas funções de simetria equivale a determinar a equivalência entre as marcações, conforme mostrado no Capítulo 4.

¹ A função identidade não realiza mudança de indivíduos, i.e. $ID: (I_x \rightarrow I_x)$

Os princípios básicos que serão usados neste algoritmo são enunciados a seguir:

1. Para cada marcação equivalente, somente uma das subárvores será construída. As subárvores removidas serão indicadas como *nós reduzidos*, sendo a subárvore análoga indicada pela sua marcação raiz (associada a um *nó principal*).
2. A cada *nó interno*, *nó principal* e *nó poço* será associada uma marcação, enquanto que a cada *nó reduzido* e *duplicados* será associada apenas uma *anotação* do tipo (M_x, FS) . Os *nós poço* além da marcação receberão um *rótulo* de "poço".
3. Um *rótulo* $(t.sh)$ será associado a cada arco direcionado, onde: sh denota o conjunto de substituições de habilitação da transição t que realiza a transformação da marcação M_x (do nó origem do arco) na marcação M_y (do nó destino do arco).
4. As marcações da AAR serão organizadas em subgrupos $\{SG_1, SG_2 \dots SG_l\}$, de tal forma que todas as marcações que pertencem ao subgrupo SG_x terão a mesma *cardinalidade de marcação* $|M_{SG}|$ e o mesmo número de *substituições de habilitação* $|SH(M_{SG})|$.

A AAR que resulta deste algoritmo, conterà todas as informações da AA completa, estando algumas destas informações representadas diretamente nos *nós* e *arcos* da árvores, enquanto outras estarão representadas indiretamente através das anotações nos *nós reduzidos*.

b) Apresentação do algoritmo

Algoritmo 3

ENTRADA: . A Rede Pr/TR.
 . A marcação inicial M_0 descrita na sua *forma normal*.
 SAÍDA: . A Árvore de Alcançabilidade Reduzida - AAR.

Procedimento Gera-AAR (Entr: R, M_0 ; Sai: AAR)

* PASSO 0

.Classificar as posições das tuplas;
 .Criar lista AAR;

* PASSO 1

.Criar $nó_0$ para a marcação M_0 ;
 .Determinar $SH(M_0)$;
 .Determinar $|SH(M_0)|$ e $|M_0|$;
 .Incluir marcação M_0 em $nó_0$;
 .SE $SH(M_0) = \emptyset$ ENTÃO Incluir rótulo "poço" em $nó_0$; SAÍDA;
 $Nós-a-explorar = \{M_0\}$;
 .ENQUANTO $Nós-a-explorar \neq \emptyset$

* PASSO 2

.Selecionar e remover uma marcação M_x de *Nós-a-explorar*;

ENQUANTO $SH(M_x) \neq \emptyset$

.Selecionar e remover um par (t, sh) de $SH(M_x)$;

.Disparar transição t com o conjunto de indivíduos da substituição de habilitação sh transformando a marcação M_x na marcação M_y ;

.Criar $nó_y$ para a nova marcação M_y ;

.Armazenar *arco* com o rótulo $(M_x, t, sh, nó_y)$;

* PASSO 2.1

.Determinar $SH(M_y)$;

.Determinar $|SH(M_y)|$ e $|M_y|$;

SE $SH(M_y) = \emptyset$ ENTÃO

.Incluir rótulo "poço" em $nó_y$;

.Incluir marcação M_y em $nó_y$;

SENÃO

.Procedimento Determinação-da-Equivalência-de-Marcações (Entr: $M_y, AAR, Sai:$ M_z, FS)

SE $FS \neq \emptyset$ ENTÃO

.Incluir a *anotação* (" $\sim M_z$ ", FS) em $nó_y$;

SENÃO

.Incluir marcação M_x em $nó_y$;

.Incluir marcação M_y em *Nós-a-explorar*;

FIM SE;

FIM SE;

FIM ENQUANTO;

FIM ENQUANTO;

FIM;

O algoritmo foi dividido em passos para facilitar o seu entendimento.

O PASSO 0 corresponde à iniciação do algoritmo, durante o qual é realizada a classificação das posições das tuplas e é criada a lista *AAR* que conterà, no término do algoritmo, as informações da árvore de alcançabilidade reduzida.

Durante o PASSO 1, é realizado o tratamento da marcação inicial M_0 . O Passo 2 só será executado se M_0 não for uma marcação do tipo "poço".

A cada vez que o PASSO 2 for completado, uma marcação é removida da lista *Nós-a-explorar* e todas as marcações resultantes do disparo das transições do conjunto de substituições de habilitação (SH) terão sido exploradas. Este passo é executado até que não haja mais nenhum *nó terminal* na árvore ($lista\ Nós-a-explorar = \emptyset$).

O PASSO 2.1 determina o tipo de cada marcação gerada pelo disparo da transição t habilitada pela substituição de indivíduos sh , onde $sh \in SH$. As novas marcações geradas so-

mente serão incluídas na lista *Nós-a-explorar* se não forem do tipo "poço" e também se não existir nenhuma marcação armazenada na lista *AAR*, que lhe seja equivalente.

c) Descrição das operações do algoritmo

Cada uma das operações realizadas pelo algoritmo será descrita a seguir, exceto o procedimento *Determinação-da-Equivalência-de-Marcações* já descrito no Capítulo 4.

Classificar as posições das tuplas

A classificação das posições das tuplas é feita conforme descrito na seção 4.2, dividindo-as em subgrupos de *posições relacionadas por valor* e de *posições de valor irrelevante*. Esta classificação é realizada analisando-se cada transição individualmente, sendo considerado nessa análise as variáveis de entrada e de saída dos arcos e as fórmulas seletoras A_T .

Criar a lista AAR

Esta operação cria a lista *AAR* que conterà as listas dos *nós*, *arcos* e *funções de simetria* da árvore.

Criar a lista nó ($nó_x$)

Esta operação cria a lista $nó_x$ que é uma *marcação* (caso dos *nós internos*), uma *marcação* e um *rótulo* (caso dos *nós poço*), ou então apenas uma *anotação* (" $\sim M_x, FS$ ") (no caso dos *nós reduzidos*). Por uma questão de otimização de memória, as funções de simetria $FS = \{fs_1, fs_2 \dots fs_n\}$ devem ficar armazenadas em uma lista separada, uma vez que a mesma função de simetria pode ser associada a diversos *nós reduzidos*. Desta forma a anotação apenas conterà *apontadores* para as funções armazenadas na lista *funções de simetria*.

As marcações armazenadas nos nós são divididas em subgrupos de acordo com a sua *cardinalidade* $|M_x|$ e o número de *substituições de habilitação* $|SH(M_x)|$.

Armazena arco ($M_x, t.sh, nó_y$)

Esta operação armazena as informações referentes ao nó de origem do arco (aquele que contém a marcação M_x), a transição t disparada, o conjunto de substituições de indivíduos sh , e o nó de destino do arco ($nó_y$).

As listas dos arcos podem ser armazenadas na lista *AAR* na ordem em que são geradas, uma vez que durante a construção da árvore de alcançabilidade reduzida elas não precisam mais ser recuperadas.

Determinar $SH(M_x)$

Esta operação *determina as substituições de habilitação das transições* para a marcação M_x , fornecendo uma lista de pares $(t.sh)$, onde t corresponde à transição habilitada e sh corresponde à substituição dos indivíduos nas variáveis de entrada da transição t .

OBS: Uma possível implementação deste procedimento pode ser feito usando-se o algoritmo de filtragem do *Simulador de redes de Petri com Objetos (SRPO)*, desenvolvido no Laboratório de Controle e Micro-Informática da UFSC [GARN 89, KAES 89, CANT 90].

Determinar $|SH(M_x)|$ e $|M_x|$

A determinação de $|SH(M_x)|$ é imediata a partir de $SH(M_x)$. Consiste em contar o número de vezes (Nt_i) que cada transição t_i aparece em $SH(M_x)$ e formar a lista $(Nt_1, Nt_2 \dots Nt_m)$.

A determinação de $|M_x|$ é também imediata, e consiste em contar o número de tuplas (Ntp_i) contidas em cada lugar p_i da marcação M_x , e formar a lista $(Ntp_1, Ntp_2 \dots Ntp_n)$.

Selecionar e remover de lista

Esta operação escolhe um dos elementos de uma lista, retirando-o segundo uma política FIFO², LIFO³ por exemplo.

O uso de uma política do tipo FIFO na seleção das marcações da lista *nós-a-explorar* resulta em uma exploração da árvore em *largura*, enquanto que o uso da política do tipo LIFO resulta em uma exploração em *profundidade*.

Incluir (X, L)

Esta operação inclui o elemento X na lista L , e é utilizada para incluir *rótulos, anotações e marcações* nas listas *Nós*, e também para incluir as marcações que correspondem a *nós terminais* na lista *nós-a-explorar*. Uma marcação quando é incluída em um nó, recebe um identificador único, de forma que a primeira marcação armazenada passa a ser M_0 , seguida de $M_1, M_2 \dots$.

2 Numa FIFO ("First In First Out") sempre é primeiro retirado o primeiro elemento inserido.

3 Numa LIFO ("Last In First Out") sempre é primeiro retirado o último elemento inserido.

k) Disparar transição

Esta operação realiza as modificações na marcação M_x transformando-a em uma nova marcação M_y , disparando a transição t com os indivíduos da substituição sh , de acordo com a regra de disparo de transições das redes Predicado/Transição.

5.2.3 Exemplo de aplicação do algoritmo de geração da AAR

Para estudar a construção da árvore de alcançabilidade reduzida, voltaremos ao exemplo apresentado na Figura 5. Já determinamos anteriormente que as posições das tuplas desta rede pertencem todas a um mesmo subconjunto *SPRV*.

Em primeiro lugar a marcação inicial $((p_1 \langle 0 \rangle \langle 1 \rangle \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle) (p_2 \langle p_3 \langle 0 \rangle \langle 1 \rangle \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle))$ é incluída na lista *Nós* dentro do subgrupo SG_0 (com $|M_{SG_0}| = (5 \ 0 \ 5)$ e $|SH(M_{SG_0})| = (5 \ 0)$) da AAR com o identificador M_0 .

Determinada a $SH(M_0)$ encontramos $\{t_1.sh_0, t_1.sh_1, t_1.sh_2, t_1.sh_3, t_1.sh_4\}$, onde $sh_i: ((i+1) \bmod 5 \rightarrow p_d, i \rightarrow p_e, i \rightarrow f)$. O disparo da transição t_1 com as substituições sh_i , gera cinco marcações diferentes.

A marcação $((p_1 \langle 1 \rangle \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle) (p_2 \langle 1 \ 0 \ 0 \rangle) (p_3 \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle))$ resultante do disparo de $t_1.sh_0$ (onde $sh_0: (1 \rightarrow p_d, 0 \rightarrow p_e, 0 \rightarrow f)$) não é equivalente a M_0 por ter $|M| \neq |M_0|$. Assim ela é incluída na lista *Nós* no subgrupo SG_1 (com $|M_{SG_1}| = (4 \ 1 \ 3)$ e $|SH(M_{SG_1})| = (2 \ 1)$) com o identificador M_1 .

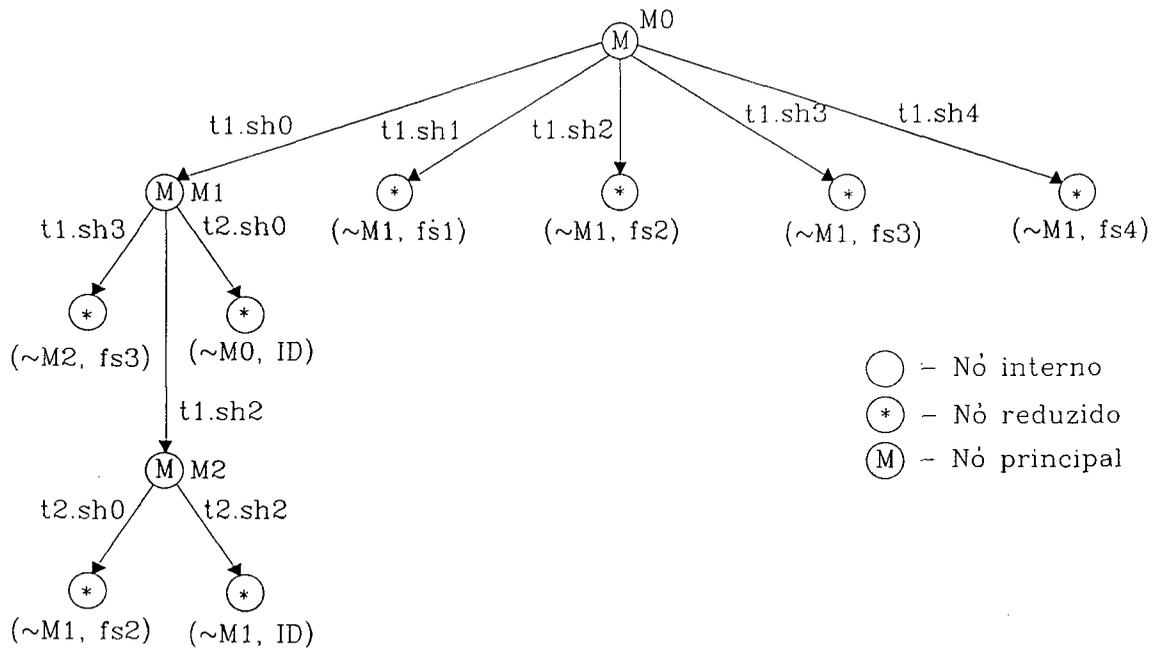
A marcação $((p_1 \langle 0 \rangle \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle) (p_2 \langle 2 \ 1 \ 1 \rangle) (p_3 \langle 0 \rangle \langle 3 \rangle \langle 4 \rangle))$ resultante do disparo de $t_1.sh_1$ (onde $sh_1: (2 \rightarrow p_d, 1 \rightarrow p_e, 1 \rightarrow f)$) não é armazenada pois é equivalente a M_1 , conforme já mostrado, podendo ser recuperada pela função de simetria $fs_1: (0 \rightarrow 1, 1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 4, 4 \rightarrow 0)$. Neste caso o nó recebe apenas a anotação $(\sim M_1, fs_1)$.

As marcações resultantes do disparo de $t_1.sh_2, t_1.sh_3$ e $t_1.sh_4$ também são equivalentes a M_1 , sendo recuperáveis respectivamente pelas funções de simetria

$$fs_2: (0 \rightarrow 2, 1 \rightarrow 3, 2 \rightarrow 4, 3 \rightarrow 0, 4 \rightarrow 1),$$

$$fs_3: (0 \rightarrow 3, 1 \rightarrow 4, 2 \rightarrow 0, 3 \rightarrow 1, 4 \rightarrow 2),$$

$$fs_4: (0 \rightarrow 4, 1 \rightarrow 0, 2 \rightarrow 1, 3 \rightarrow 2, 4 \rightarrow 3).$$



shi: $((i+1) \bmod 5 \rightarrow pd, i \rightarrow pe, i \rightarrow f)$ para $i = 0,1,2,3,4$

$M_0 = ((p_1 \langle 0 \rangle \langle 1 \rangle \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle) (p_2) (p_3 \langle 0 \rangle \langle 1 \rangle \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle))$

$M_1 = ((p_1 \langle 1 \rangle \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle) (p_2 \langle 1 \ 0 \ 0 \rangle) (p_3 \langle 2 \rangle \langle 3 \rangle \langle 4 \rangle))$

$M_2 = ((p_1 \langle 1 \rangle \langle 3 \rangle \langle 4 \rangle) (p_2 \langle 1 \ 0 \ 0 \rangle \langle 3 \ 2 \ 2 \rangle) (p_3 \langle 4 \rangle))$

$fs_1 = (0 \rightarrow 1, 1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 4, 4 \rightarrow 0)$

$fs_2 = (0 \rightarrow 2, 1 \rightarrow 3, 2 \rightarrow 4, 3 \rightarrow 0, 4 \rightarrow 1)$

$fs_3 = (0 \rightarrow 3, 1 \rightarrow 4, 2 \rightarrow 0, 3 \rightarrow 1, 4 \rightarrow 2)$

$fs_4 = (0 \rightarrow 4, 1 \rightarrow 0, 2 \rightarrow 1, 3 \rightarrow 2, 4 \rightarrow 3)$

Figura 13. AAR almoço dos cinco filósofos

Repetindo o Passo 2 do algoritmo até que a lista *Nós-a-explorar* fique vazia, obteremos ao final do algoritmo, a AAR da Figura 13.

5.3 Equivalência entre a árvore de alcançabilidade completa (AA) e a árvore de alcançabilidade reduzida (AAR)

É necessário demonstrar que a AAR obtida através do Procedimento *Gera-AAR* descrito anteriormente contém todas as informações da árvore de alcançabilidade (AA). Desta forma mostraremos que o princípio de redução utilizado não resulta em perda da capacidade de análise da árvore.

Embora todas as informações da AA estejam representadas na AAR, algumas delas estão armazenadas na forma direta nos nós e arcos, enquanto outras estão contidas indiretamente nas *anotações dos nós reduzidos*.

Entre as informações que podem estar representadas de forma direta ou indireta na AAR, destacamos as marcações alcançáveis e os indivíduos capturados durante as seqüências de disparo das transições. As seqüências de disparo das transições (sem considerar os indivíduos capturados) estão todas representadas diretamente na AAR.

Nas seções seguintes mostraremos como as *anotações dos nós reduzidos* permitem recuperar todas as informações da árvore de alcançabilidade completa.

5.3.1 Determinação da alcançabilidade de uma marcação

Para podermos determinar a alcançabilidade de uma marcação M_x qualquer da rede R , devemos primeiramente verificar se ela está representada em algum nó árvore. Caso não esteja, verificar se ela está em alguma das subárvores suprimidas na redução, conforme mostrado a seguir.

a) Obtenção do grafo de alcançabilidade reduzido

A determinação da alcançabilidade de uma marcação da rede será realizada através do uso do grafo de alcançabilidade reduzido (GAR). Para a sua obtenção é necessário que as *funções de simetria* associadas aos *nós reduzidos* da AAR sejam associadas às marcações dos *nós principais*.

Notação 5.2 O conjunto dos nós reduzidos que possuem a anotação $(\sim M_y, FS_i)$ será denotado por $NóR_{My} = \{nóR_1, nóR_2 \dots nóR_k\}$.

Definição 5.3 O conjunto das funções de simetria ligadas aos nós principais da AAR é definido por:

$$FS_{My} = \bigcup_{i=1 \dots k} FS_i \cup ID \mid FS_i \text{ é a anotação do } nóR_i \in NóR_{My}$$

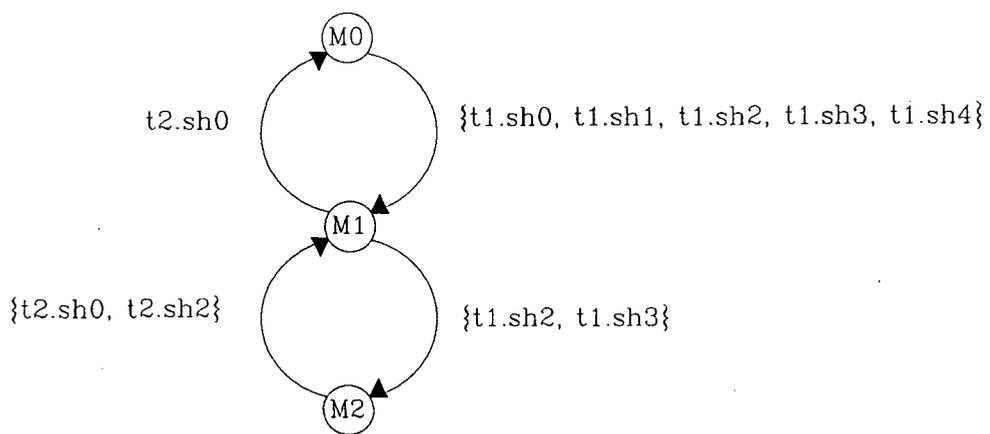
Exemplo 5.1 No exemplo da Figura 13 os conjuntos de funções de simetria são $FS_{M0} = \{ID\}$, $FS_{M1} = \{fs_1, fs_2, fs_3, fs_4, ID\}$ e $FS_{M2} = \{fs_3\}$

A partir da definição desse conjunto de funções de simetria ligadas aos nós principais da AAR, podemos *eliminar os nós reduzidos* e os arcos cujo destino é um destes nós. Esta transformação resulta no grafo alcançabilidade reduzido .

Definição 5.4 A árvore de alcançabilidade reduzida (AAR) representada conforme a definição 5.1 pode ser transformada em um grafo de alcançabilidade reduzido (GAR), fazendo com que:

- 1) cada arco $(M_x, t.sh, nóR_i)$ da árvore onde $nóR_i$ é anotado com $(\sim M_y, FS_i)$ seja transformado em um arco $(M_x, t.sh, M_y)$ no grafo.
- 2) se existem dois arcos $(M_x, t.sh, M_y)$ e $(M_x, t'.sh', M_y)$ com a mesma origem M_x e o mesmo destino M_y , então eles são unidos em um arco $(M_x, \{t.sh, t'.sh'\}, M_y)$.

Exemplo 5.2 A AAR da Figura 13 pode ser transformado no GAR da Figura 14 utilizando-se a definição acima.



$$\begin{aligned}
 FS_{M0} &= \{ID\} \\
 FS_{M1} &= \{ID, fs1, fs2, fs3, fs4\} \\
 FS_{M2} &= \{fs3\} \\
 FSV_{M0} &= FSV_{M1} = FSV_{M2} = \{ID, fs1, fs2, fs3, fs4\}
 \end{aligned}$$

Figura 14. Grafo de alcançabilidade reduzido (GAR) do almoço dos cinco filósofos

b) Utilização do GAR para a determinação da alcançabilidade da marcação

Com o objetivo de determinar a alcançabilidade de uma marcação da rede R introduzimos aqui, o conceito de *funções de simetria válidas para cada marcação M_x* do GAR.

Notação 5.5 O caminho orientado $M_0 [t_1.sh_1 > M_1 \dots M_{k-1} [t_k.sh_k > M_k$ será denotado por $M_0 [\pi > M_k$

Definição 5.6 O conjunto de todas as marcações a partir das quais M_x é alcançável é definido por:

$$Pré_{M_x} = \{M_1, M_2 \dots M_k\} \cup M_x \text{ onde } M_i \in Pré_{M_x} \Leftrightarrow \exists \pi \mid M_i [\pi > M_x$$

Definição 5.7 Cada marcação do GAR tem associado um conjunto de funções de simetria válidas, definidas por:

$$FSV_{M_x} = \bigcup_{i=1 \dots k} FSV_{M_i} \mid M_i \in Pré_{M_x}$$

Exemplo 5.3 Para o mesmo exemplo da Figura 13 podemos obter as funções de simetria válidas para cada marcação: $Pré_{M_x} = \{M_0, M_1, M_2\}$ e $FSV_{M_x} = \{fs_1, fs_2, fs_3, fs_4, ID\}$ para $i = 1, 2, 3$.

Obtidas as funções de simetria válidas podemos determinar a alcançabilidade de uma marcação qualquer da rede realizando apenas testes sobre as marcações da AAR e suas funções de simetria.

Notação 5.8 O conjunto das marcações do grafo de alcançabilidade reduzido será denotado por $M_{GAR} = \{M_1, M_2 \dots M_k\}$.

Teorema 5.9 Seja M_x uma marcação da rede R , e sejam $M_{GAR} = \{M_1, M_2 \dots M_k\}$ e $FSV = \{FSV_{M_1}, FSV_{M_2} \dots FSV_{M_k}\}$, onde FSV_{M_i} é o conjunto de funções de simetria válidas para M_i . Então podemos afirmar que:

$$M_x \in [M_0 > \Leftrightarrow \exists M_y \in M_{GAR} \mid fs(M_y) = M_x \text{ onde } fs \in FSV_{M_y}$$

Esse teorema apenas formaliza que para uma marcação M_x ser alcançável na rede (R, M_0) , é necessário que esta marcação esteja representada diretamente nos nós da GAR, ou então que ela pertença a uma das subárvores suprimidas durante a redução.

O procedimento para determinar a alcançabilidade de uma marcação a partir de M_0 pode ser realizado de forma automática através do seguinte algoritmo:

Algoritmo 4

- ENTRADA:**
- . A marcação M_x descrita na forma normal.
 - . A lista $Nós = \{SG_1, SG_2 \dots SG_l\}$ da AAR.
- SAÍDA:**
- . Se M_x é alcançável então $Flag = 1$, e o algoritmo apresenta a marcação $M_y \in Nós$ que é equivalente a M_x e a função simétrica $fs_1 \mid fs_1(M_y) = M_x$.
 - . Se M_x não é alcançável então $Flag = 0$.

Procedimento Determinação-Alcançabilidade (Entr: $M_x, Nós$; Sai: $Flag, M_y, fs$)

.Flag = 0;
 .Determinar $|M_x|$ e $|SH(M_x)|$;
 .SE $\exists SG_i \in N\acute{o}s \mid |M_x| = |M_{SG_i}|$ e $|SH(M_x)| = |SH(M_{SG_i})|$; ENTÃO
 .PARA TODO $M_y \in SG_i$
 .PARA TODA $fs \in FS_{M_y}$
 .SE $fs(M_y) = M_x$ ENTÃO Flag = 1; SAÍDA;
 .FIM PARA;
 .FIM PARA;
 .FIM SE;
 .FIM;

Exemplo 5.4 Seja $M = ((p_1 \langle 0 \rangle \langle 1 \rangle \langle 2 \rangle) (p_2 \langle 4 \ 3 \ 3 \rangle \langle 0 \ 4 \ 4 \rangle) (p_3 \langle 2 \rangle))$ e $M' = ((p_1 \langle 0 \rangle \langle 1 \rangle \langle 3 \rangle) (p_2 \langle 3 \ 2 \ 2 \rangle \langle 0 \ 4 \ 4 \rangle) (p_3 \langle 1 \rangle))$. Para determinar se estas marcações são alcançáveis na rede da Figura 5 utilizaremos a AAR obtida na Figura 13. Podemos assim concluir que M não é alcançável em (R, M_0) pois $M \notin N\acute{o}s$ e $\neg \exists M_y \in N\acute{o}s \mid fs(M_y) = M$. A marcação M' é alcançável pois $M' \in N\acute{o}s$ mas $\exists M_y \in N\acute{o}s \mid fs_2(M_y) = M'$, onde $fs_2: (0 \rightarrow 2, 1 \rightarrow 3, 2 \rightarrow 4, 3 \rightarrow 0, 4 \rightarrow 1)$ e $M_y = M_2 = ((p_1 \langle 1 \rangle \langle 3 \rangle \langle 4 \rangle) (p_2 \langle 1 \ 0 \ 0 \rangle \langle 3 \ 2 \ 2 \rangle) (p_3 \langle 4 \rangle))$

5.3.2 Determinação da Seqüência de Disparos de Transições

Uma vez determinada a alcançabilidade de uma marcação, é possível determinar as transições que necessitam ser disparadas para atingí-la. A determinação da seqüência de disparo de transições que leva a uma marcação determinada pode ser dividida em duas etapas.

- 1) Determinar as transições da seqüência de disparos.
- 2) Determinar os indivíduos capturados por cada uma dessas transições.

A primeira etapa, que é a determinação das transições da seqüência é obtida diretamente da AAR.

Teorema 5.10 Se $M_x \in [M_0 \rangle$ então $\exists \pi = M_0 [t_1 sh_1 \rangle M_1 \dots M_{x-1} [t_x sh_x \rangle M_x \Leftrightarrow \exists \pi' = M_0 [t_1 sh_1' \rangle M_1' \dots M_{x-1}' [t_x sh_x' \rangle M_x' \mid M_i' \in N\acute{o}s$ (para $i = 1 \dots x$) e $M_x = fs(M_x')$ onde $fs \in FSV_{M_x'}$.

Como os *caminhos orientados* π e π' diferem apenas nas substituições de indivíduos das transições disparadas, podemos obter diretamente do caminho π' representado na AAR, as transições disparadas para transformar M_0 em M_x .

Esta característica da AAR indica que a *linguagem* da rede (R, M_0) é diretamente representada na árvore de alcançabilidade reduzida (AAR).

Definição 5.11 O conjunto de todas as seqüências de disparo de transição para uma rede (R, M_0) , a partir de M_0 é denominada a *linguagem* da rede R a partir da marcação M_0 sobre o alfabeto T .

Teorema 5.12 A linguagem representada diretamente pela árvore de alcançabilidade reduzida (AAR) é igual a linguagem representada pela árvore de alcançabilidade completa.

A grande vantagem do uso da AAR na determinação da linguagem da rede (R, M_0) , provém do fato das redundâncias existentes na AA causadas pelas simetrias da rede terem sido eliminadas.

5.3.3 Determinação dos Indivíduos Capturados pelas Transições

A determinação dos indivíduos capturados pelas transições da seqüência de disparos que transformam a marcação inicial em uma marcação M_x alcançável, pode ser realizada através do seguinte procedimento:

Algoritmo 5

ENTRADA: . A marcação alcançável M_x
 . A marcação $M_x' \in \text{Nós} \mid M_x \sim M_x'$.
 . A lista AAR
 . A função de simetria $fs \mid fs(M_x') = M_x$
 SAÍDA: . Um caminho orientado $\pi \mid M_0[\pi > M_x$

Procedimento Determinação-Caminho (Entr: M_x, M_x', AAR, fs ; Sai: π)

.Encontrar $M_y \in \text{Nós} \mid M_0[\pi_1' > M_y, \pi_2' > M_x'$ e $\exists n \in R_i$ anotado com $(\sim M_y, FS_i) \mid fs \in FS_i$;
 .Determinar $\pi_1 \mid M_0[\pi_1' > n \in R_i$
 .Determinar $\pi_2 = fs(\pi_2') \mid M_y[\pi_2' > M_x$
 .Determinar $\pi = \pi_1 \cdot \pi_2$
 .FIM;

5.4 Comparação da AAR com a AA

Para podermos avaliar o método sugerido neste trabalho, construímos a árvore de alcançabilidade completa (AA) e a árvore de alcançabilidade reduzida (AAR) para alguns exemplos.

A construção da AA foi realizada usando um algoritmo similar ao Algoritmo 4 descrito neste capítulo. A troca do procedimento *Determinação-da-Equivalência-de-Marcações* por um teste de igualdade de marcações consiste na principal diferença entre os algoritmos.

No exemplo do *Almoço dos Cinco Filósofos* [LIND 90] a rede da Figura 5 foi transformada em uma *Rede com Árvore Diretamente Redutível (RDR)* para permitir a determinação automática da AAR. Essa transformação consistiu em associar a cada tupla de filósofo um

indivíduo que indica a posição do palito que está a sua direita, eliminando assim a fórmula seletora $direita(f) = p_d$ da transição t_1 . A RDR obtida é mostrada na Figura 15.

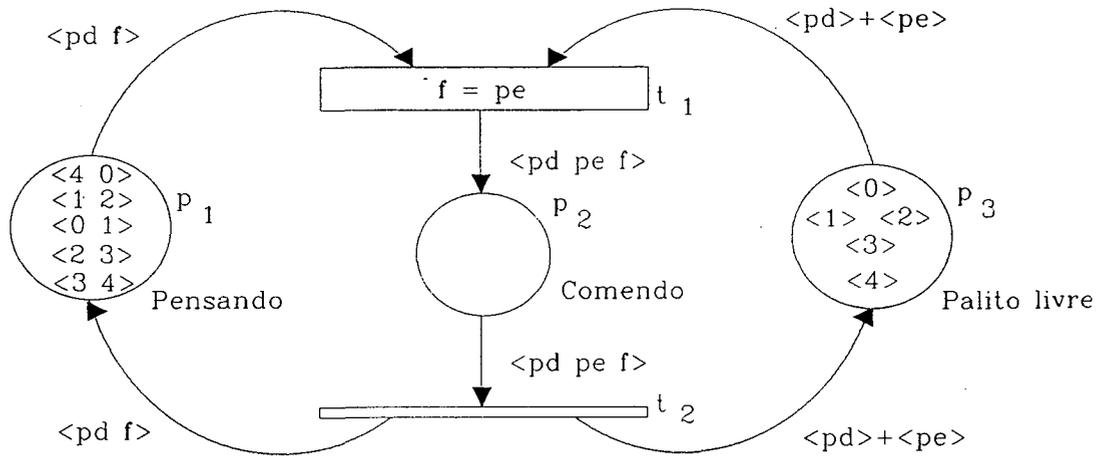
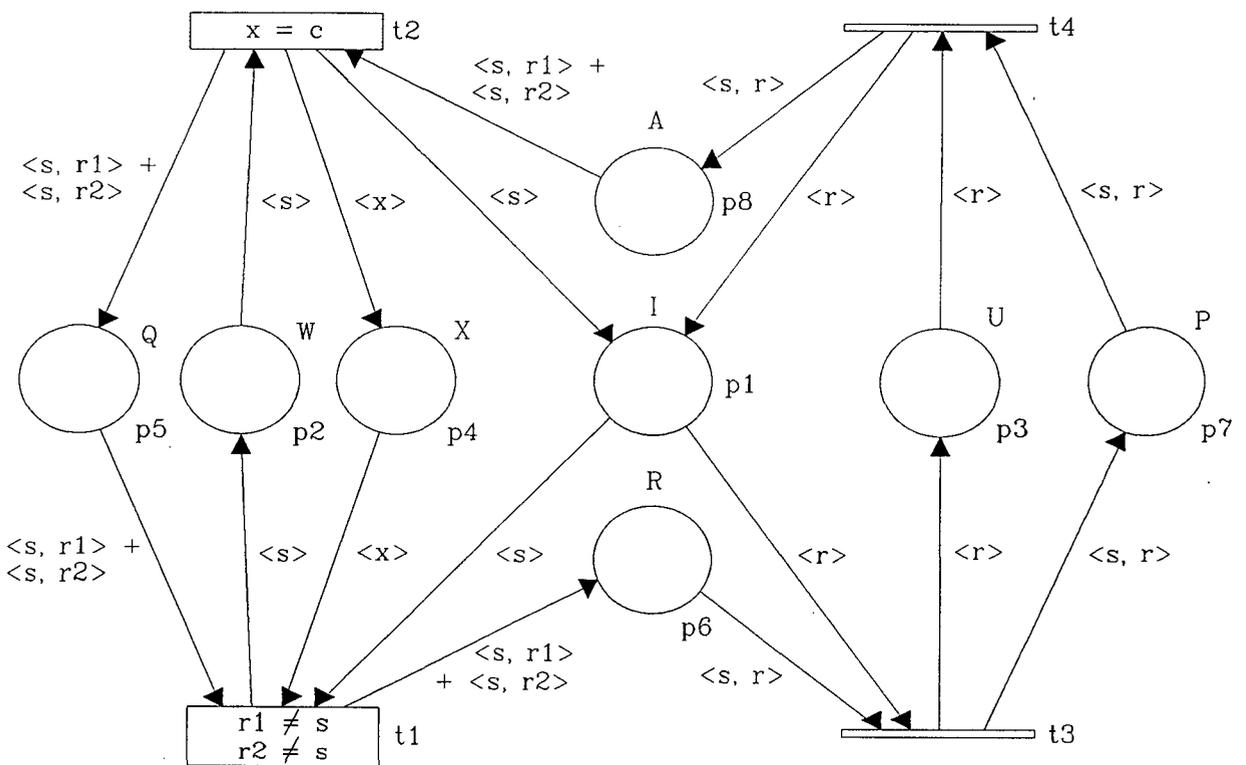


Figura 15. O almoço dos cinco filósofos - modelado em RDR



$M_0 = Q \langle a_1, a_2 \rangle \langle a_1, a_3 \rangle \langle a_2, a_1 \rangle \langle a_2, a_3 \rangle \langle a_3, a_1 \rangle \langle a_3, a_2 \rangle$
 $I \langle a_1 \rangle \langle a_2 \rangle \langle a_3 \rangle$
 $X \langle c \rangle$

Figura 16. O protocolo para a manutenção de múltiplas cópias de um banco de dados - modelado em RDR

Para o exemplo do *Protocolo para a Manutenção de Múltiplas Cópias de um Banco de Dados* [GENR 87] (Figura 16) foram construídas árvores de alcançabilidade para 3, 4 e 5 gerenciadores de banco de dados, aos quais denominaremos respectivamente de DATABASE3, DATABASE4, DATABASE5.

A Tabela 5 resume os resultados obtidos da construção da AA e AAR nos casos citados. Os resultados marcados com o caractere "@" foram retirados de [HUBE 86]. Os resultados obtidos são similares aos fornecidos por [HUBE 86] contudo na nossa abordagem a construção da AAR é realizada de forma completamente automática para estes exemplos.

Tabela 5. Número marcações das AA e AAR para os exemplos clássicos

Exemplos Clássicos	Marcações da AA		Marcações da AAR		Índice de Redução ⁶
	Armazenadas ⁴	Geradas ⁵	Armazenadas ⁴	Geradas ⁵	
FILÓSOFO	11	30	3	10	3
DATABASE 3	28	42	7	12	3.5
DATABASE 4	-	224@	11	25	9
DATABASE 5	-	>1400@	16	46	>30

Para o exemplo dos "*k-Processos Concorrentes*" apresentado na Figura 4 do Capítulo 3, construímos também a AAR de forma automática. Os resultados mostrados nas tabelas 6 e 7 revelam que a redução não ocorreu somente em função do número de indivíduos mas também em função do grau de paralelismo. Os resultados obtidos também mostram que a tendência de crescimento exponencial $O(N^k)$ do número de marcações geradas e armazenadas da AA, torna-se polinomial $O(N + k)^x$ conforme pode ser visto na Figura 17, onde estão mostrados graficamente o número de marcações armazenadas da AA e da AAR.

Tabela 6. Número de nós e arcos da AAR para $k = 3$

Exemplos com $k = 3$	Marcações da AA		Marcações da AAR		Índice de Redução ⁶
	Armazenadas ⁴	Geradas ⁵	Armazenadas ⁴	Geradas ⁵	
$N = 1$	8	24	4	12	2
$N = 2$	27	108	10	40	2.7
$N = 3$	64	288	20	90	3.2
$N = 4$	125	600	35	168	3.6

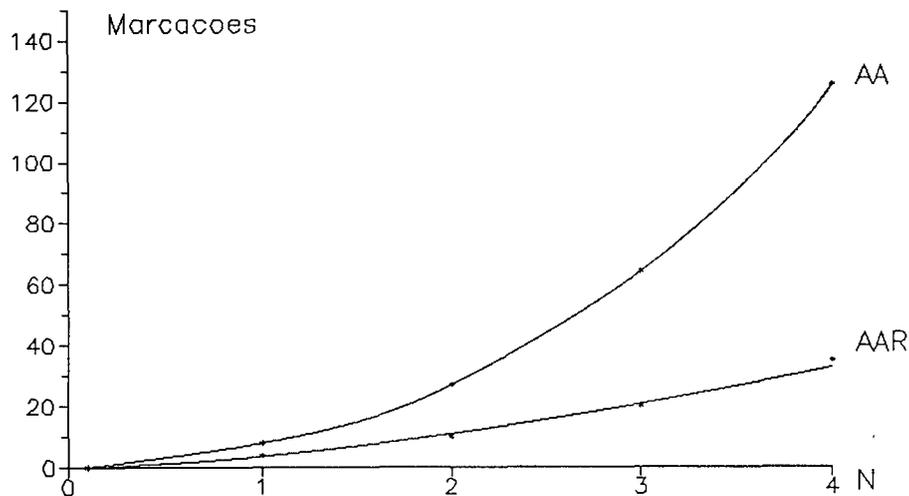
⁴ As marcações armazenadas correspondem aos nós da AA e da AAR.

⁵ As marcações geradas correspondem aos arcos da AA e da AAR.

⁶ índice de redução = marcações geradas AA / marcações geradas AAR.

Tabela 7. Número de nós e arcos da AAR para $N = 2$

Exemplos com $N = 2$	Marcações da AA		Marcações da AAR		Índice de Redução ⁶
	Armazenadas ⁴	Geradas ⁵	Armazenadas ⁴	Geradas ⁵	
$k = 1$	3	4	3	4	0
$k = 2$	9	24	6	16	1.5
$k = 3$	27	108	10	40	2.7
$k = 4$	81	432	15	80	5.4

Figura 17. Número de marcações armazenadas em função do valor N , para $k = 3$

Os resultados acima apresentados deixam claro que a redução ocorreu não somente no número de *marcações armazenadas* mas também ocorreu proporcionalmente no número de *marcações geradas*.

Uma outra observação sobre estes resultados é que o índice de redução aumenta com o número de indivíduos. Isto se deve ao índice de redução crescente.

5.5 Conclusão

A AAR construída conforme o procedimento descrito neste capítulo fornece a base para provar as propriedades dinâmicas descritas no Capítulo 2.

O método de construção da AAR proposto permite obter uma árvore na qual a *alcançabilidade* de uma marcação qualquer da rede R pode ser determinada facilmente utilizando-se apenas das marcações da AAR e das funções de simetria encontradas.

Utilizando-se as funções de simetria para modificar os indivíduos capturados pelas transições é também possível recuperar todos os *caminhos* que levam a qualquer marcação alcançável da rede.

Como a AAR representa diretamente todas as possíveis seqüências de disparo de transição, é possível analisar a linguagem da rede (R, M_0) diretamente sobre ela. Neste sentido a AAR traz a vantagem adicional de reduzir as redundâncias que dificultam a análise.

Como o número de transições representadas na AAR é menor do que na AA torna-se mais fácil analisar as propriedades que estão ligadas as seqüências de disparo, tais como vivacidade, justiça, persistência, etc. As propriedades com marcações poço, reiniciação, e outras também tem a sua determinação facilitada através do uso do grafo de alcançabilidade reduzido.

Para os exemplos clássicos apresentados em [HUBE 86], produzimos os mesmos resultados com a vantagem de obtê-los de forma completamente automática.

Finalmente os resultados obtidos mostraram ainda que o problema da explosão combinatória de estados fica parcialmente resolvido, uma vez que as tendências de crescimento exponencial de estados passam para um crescimento polinomial.

CAPÍTULO VI

CONCLUSÕES

Neste trabalho introduzimos um novo método para construir a árvore de alcançabilidade reduzida que preserva a capacidade de análise. Este método é baseado nas simetrias da rede.

O princípio de redução utilizado pode ser expresso pela seguinte sentença: "*Removeremos somente as subárvores que têm as mesmas seqüências de disparo de transição (comportamentos análogos) que outras já construídas na AAR*".

O método proposto traz as seguintes melhorias em relação aos métodos nos quais são também preservadas as propriedades, apresentados em [HUBE 86] e em [LIND 89]:

- Os métodos citados permitem somente a redução de árvores simétricas através do uso das *simetrias totais*, enquanto que a nossa abordagem pode ser também aplicada na construção de AAR em casos assimétricos (como por exemplo redes não reiniciáveis) usando-se as *simetrias totais e parciais*. Isto resulta em uma AAR de menor tamanho e menor custo de tempo de construção.
- O uso das *cardinalidades das substituições de habilitação* das transições é uma melhoria em relação ao uso somente da *cardinalidade da marcação* (como nos métodos citados), resultando em uma melhor separação das marcações, permitindo uma diminuição do número de comparações de marcações, uma vez que permite eliminar as marcações que não possuem as mesmas condições de equivalência. O uso dessa condição adicional de equivalência pode ser aplicado aos métodos citados para melhorá-los.
- Nos métodos citados a definição das simetrias é de responsabilidade do usuário, e deve ser realizada antes do início da construção da árvore de alcançabilidade. Na nossa abordagem todas as *funções de transformação* são determinadas automaticamente. O usuário intervém somente para verificar a simetria de cada uma das funções apresentadas.
- Nossa abordagem é ainda mais fácil e eficiente para a solução de alguns casos particulares, tais como a subclasse de Redes Pr/T denominada de *Redes com Árvore de Alcançabilidade Diretamente Redutíveis (RDR)*. Nestes casos a redução é completamente automática, pois todas as funções de transformação determinadas são simétricas. Além disso, existem regras que possibilitam a transformação de vários tipos de Redes Pr/T em RDR.

Na AAR construída de acordo com a nossa abordagem, podemos obter diretamente as *propriedades dinâmicas* tais como marcações mortas, laços de marcações, transições mortas, estados de recepção reiniciação e vivacidade. Também é possível determinar a alcançabilidade de uma marcação qualquer da rede (R, M_0) .

Além disso, nossa abordagem facilita a *verificação da equivalência de linguagem*, que também pode ser obtida diretamente da AAR, uma vez que ela contém todas as seqüências de disparo de transição. A determinação das seqüências de disparo entre duas marcações também é facilmente determinável.

Embora o método tenha sido descrito para as redes Pr/T, ele é facilmente extensível para outras redes de Petri de Alto Nível, tais como as Redes de Petri com Objetos [SIBE 84] e as Redes Coloridas [JENS 87].

Um prolongamento desse trabalho é importante, visando ampliar e facilitar ainda mais o uso do método aqui proposto. A seguir citaremos alguns dos pontos que ao nosso modo de ver deveriam ser primordialmente trabalhados:

- . Analisar os diversos tipos de *propriedades* e realizar a *verificação lógica* e ou a *verificação observacional* sobre a AAR.
- . Incluir a *interpretação fraca da relação de cobertura*, de forma a poder trabalhar também com *redes ilimitadas*, produzindo uma árvore de cobertura reduzida, a qual possua a mesma capacidade da árvore de cobertura completa.
- . Determinar outras condições necessárias para a equivalência de marcações, de modo a aumentar a automatização do método no caso geral das redes Pr/T.
- . Estudar a possibilidade de ampliação da subclasse de *Redes com Árvore Diretamente Reduzível (RDR)*.
- . Verificar se não existe alguma forma de reutilizar a análise das funções de simetria feitas pelo usuário, de forma a poder reduzir a necessidade de intervenção deste.
- . Estudar o uso do mecanismo de compilação de regras do *Simulador de Redes de Petri* desenvolvido no laboratório de Controle e Micro-Informática da UFSC [GARN 89, CANT 90], para realizar uma redução sem a determinação explícita das simetrias.

REFERÊNCIAS BIBLIOGRÁFICAS

- [ALAI 85] Alaiwan, H.; Toudic, J. M.: "Recherche des semi-flots, des verrous et des trapes dans les réseaux de Petri". Technique es Science Informatiques, 4(1):103-112, (1985).
- [AZEM 89] Azema, P. & Vernadat, F.: "Requirement analysis for communication protocols". Raport LAAS/CNRS, Toulouse, France, (1989), p:1-15.
- [BEST 87] Best, E.: "Structure Theory of Petri Nets: the Free Choice Hiatus". Lecture Notes in Computer Science, 254:168-206, (1987).
- [BOUR 87] Bourguet, A.: "A Petri net tool for service validation in protocol". Protocol Specitication, Testing and Verification VI, (1987), p:281-292.
- [BRAM 83] Brams, G. W.: Réseaux de Petri: Théorie et pratique, Masson, Paris, (1983).
- [CANT 90] Cantú, E.: "Uma abordagem para a representação, simulação e implementação de sistemas baseada na rede de Petri a Objetos". Dissertação M.Sc, Universidade Federal de Santa Catarina, Florianópolis, (1990).
- [COUR 87] Courtiat, J. P.: "Contribution a la description formelle de protocoles". These Docteur-Ingenieur, Universite Paul Sabatier de Toulouse, France, (1987).
- [DIAZ 82] Diaz, M.: "Modelling and analysis of communication and cooperation protocols using Petri Net based models". Protocol Specitication, Testing and Verification II, (1982), p:465-510.
- [DIAZ 89] Diaz, M.: "Environnements Logiciels pour la conception des Protocoles dans les Systemes distribués". Anais do Seminário Franco Brasileiro em Sistemas Informáticos Distribuídos, Florianópolis, Brasil, (set. 1989), p:28-35.
- [DIMI 90] Dimitrovici, C.; Hummert, V.; Petrucci, L.: "The Properties of Algebraic Nets Schemes in some Semantics". Proceedings of the 11th International Conference on Aplication and Theory of Petri Nets, Paris, France, (jun. 1990), p:180-203.
- [ESTE 85] Esteban, P.: "Sur la recherche d'algorithmes simplifiés d'analyse des reseaux de Petri". These Docteur-Ingenieur, Universite Paul Sabatier de Toulouse, France, (1985),

- [ESTE 86] Esteban, P.; Valette, R.; Courvoisier, M.: "Simplified algorithms for Petri net analysis". IECON'86 12th Annual Conference of IEEE Industrial Electronics Society, (nov. 1986), p:437-442.
- [FARI 89] FARINES, J. M.; Cantú, E.; Garnousset, H. E.; Maziero, C. A.: "ARP : Uma Ferramenta para Desenvolvimento de Software em aplicações distribuídas". Anais do Seminário Franco Brasileiro em Sistemas Informáticos Distribuídos, Florianópolis, Brasil, (set. 1989), p:73-79.
- [GARN 89] Garnousset, H. E.; Farines, J. M.; Cantú, E.: "Efficient Tools for Analysis and Implementation of Manufacturing Systems Modelled by Petri Nets with Objects: A production Rules Compilation Based Approach". IECON'89 15th Annual Conference of IEEE Industrial Electronics Society, Philadelphia, USA, (nov. 1989), p:543-549.
- [GENR 87] Genrich, H. J.: "Predicate/Transition Nets". Lecture Notes in Computer Science, 254, (1987).
- [HADD 87] Haddad, S.: "Une categorie Reguliere de reseau de Petri de Haut Niveau: definition, proprietes et reductions". These Docteur-Informatique, Universite Paul et Marie Curie, Paris, France, (1987).
- [HUBE 86] Huber, P.; Jensen, A. M.; Jepsen, L. O.; Jensen, K.: "Reachability Trees for High-level Petri Nets". Theoretical Computer Science, 45:261-292, (1986).
- [JENS 81] Jensen, K.: "Coloured Petri Nets and the Invariant-Method". Theoretical Computer Science, 14:317-336, (1981).
- [JENS 87] Jensen, K.: "Coloured Petri Nets". Lecture Notes in Computer Science, 254, (1987).
- [KAES 89] Kaestner, C. A.: "Contribuição ao Estudo e Desenvolvimento de um Sistema de Regras de Produção". Dissertação M.Sc, Universidade Federal de Santa Catarina, Florianópolis, (1989).
- [KARP 69] Karp, R. M.; Miller, R. E.: "Parallel program schemata". Journal of Computer and Systems Science, 3(2), (1969).
- [KNIG 89] Knight, K.: "Unification: A Multidisciplinary Survey". ACM Computing Surveys, 21(1):93-124, (1989).
- [LAUT 87] Lautenbach, K.: "Linear Algebraic Techniques for P/T Nets". Lecture Notes in Computer Science, 254, (1987).

- [LIND 89] Lindqvist, M.: "Parametrized Reachability Trees for Predicate/Transition Nets". Acta Polytechnica Scandinavica - Mathematics and Computer Science Series, 54:1-120, (1989).
- [LIND 90] Lindqvist, M.: "Parametrized Reachability Trees for Predicate/Transition Nets". Proceedings of the 11th International Conference on Application and Theory of Petri Nets, Paris, France, (jun. 1990), p:22-42.
- [MEMM 87] Memmi, G.; Vautherin, J.: "Analysing Nets by the Invariant Method". Lecture Notes in Computer Science, 254:300-336, (1987).
- [MOEC 91] Moecke, M.; Farines, J. M.: "On the reduction of the reachability Tree, Preserving the model properties". Submitted to "12th International Conference on Application and Theory of Petri Nets" (1991).
- [MURA 89] Murata, T.: "Petri Nets: Properties, Analysis and Applications". Proceeding of the IEEE, 77(4):541-580, (1989).
- [PETE 81] Peterson, J. L.: Petri net theory and the modeling of systems. Prentice-Hall, Englewood Cliffs, USA, (1981).
- [PETR 62] Petri, C. A.: "Kommunikation mit Automaten". Schriften des Rheinisch Westfälischen Institutes für Instrumentelle Mathematik Nr. 3, Universität Bonn, (1962).
- [SIBE 84] Sibertin-Blanc, C.: "Petri Nets with Individuals or Objects Instead of Tokens". Internal Report n° 229, Université des Sciences Sociales, UER Informatique, Toulouse, France, (1984), p:1-35.
- [SOUI 90] Souissi, Y.: "On Liveness Preservation by Composition of Nets via a set of Places". Proceedings of the 11th International Conference on Application and Theory of Petri Nets, Paris, France, (jun. 1990), p:104-122.
- [TUOM 86] Tuominen, H.: "Temporal logic as a query language for Petri Net Reachability Graphs". Proceedings of the 7th International Conference on Application and Theory of Petri Nets, Oxford, England, (jul. 1986), p:327-342.
- [VALK 87] Valk, R.: "Infinite Behaviour and Fairness". Lecture Notes in Computer Science, 254:377-396, (1987).
- [VALM 89] Valmari, A.: "Stubborn Sets for Reduced State Space Generation". Proceedings of the 10th International Conference on Application and Theory of Petri Nets, Bonn, West Germany, (jun. 1989), v.2, p:1-22.

- [VALM 90] Valmari, A.: "Compositional State Space Generation". Proceedings of the 11th International Conference on Application and Theory of Petri Nets, Paris, France, (jun. 1990), p:43-62.
- [VAUT 86] Vautherin, J.: "Parallel Systems Specifications with Coloured Petri Nets and Algebraic Abstract Data Types". Proceedings of the 7th International Conference on Application and Theory of Petri Nets, Oxford, England, (jul. 1986), p:5-23.
- [VERN 89] Vernadat, F.: "Verification formelle d'applications repartiées caracterisation logique d'une equivalence de comportement". These Docteur-Informatique, Universite Paul Sabatier de Toulouse, France, (1989).