Instituto de Ciências Matemáticas de São Carlos

ISSN - 0109-2569

Propriedades de máquinas de estado finito relevantes para critérios de geração de seqüências de teste

KAREN KIOMI NAKAZATO
JOSÉ CARLOS MALDONADO
SANDRA C. P. F. FABBRI
PAULO CESAR MASIERO

Nº 27

RELATÓRIOS TÉCNICOS DO ICMSC

SÃO CARLOS
Out. / 1994

SYSNO 872262
DATA / /
ICMC - SBAB

Propriedades de Máquinas de Estado Finito relevantes para Critérios de Geração de Seqüências de Teste

Karen Kiomi Nakazato[1]
José Carlos Maldonado[2]
Sandra C. P. F. Fabbri[3]
Paulo Cesar Masiero[2]

1- Mestranda do ICMSC-USP-São Carlos; Profa. do CEETPS-FATEC-Taquaritinga

2-Professores do ICMSC-USP-São Carlos-Cx.P. 668

3-Doutoranda do IFSC-USP-São Carlos; Profa. do Depto de Computação-UFSCar-Cx.P.676

Conteúdo

1. Introdução 1
2. Máquinas de Estado Finito 2
3. Propriedades e Características de MEF 6
4. Aspectos de Implementação 15
    4.1. Principais Estruturas de Dados utilizadas nos Algoritmos 16
        4.1.1. Conjunto de Estados 16
        4.1.2. Conjunto de Símbolos de Entrada 16
        4.1.3. Conjunto de Símbolos de Saída 16
        4.1.4. Tabela de Transição 16
            4.1.4.1. Subtabela Zv 17
            4.1.4.2. Subtabela Sv+1 17
        4.1.5. Tabelas Pk 18
        4.1.6. Rótulo de Origem 20
        4.1.7. Índice da Seqüência Minimal de Diagnose 20
    4.2. Algoritmos de Geração e Identificação 21
        4.2.1. Geração de Tabelas Pk 21
            4.2.1.1. Geração da Tabela P1 25
            4.2.1.2. Identificação da Última Tabela Pk 27
            4.2.1.3. Inclusão da Origem 29
            4.2.1.4. Obtenção do Índice 30
            4.2.1.5. Identificação de Linhas Semelhantes 31
            4.2.1.6. Identificação de Listas Semelhantes 32
            4.2.1.7. Geração de uma Seqüência Minimal de Diagnose 33
            4.2.1.8. Geração do Índice da Seqüência Minimal de Diagnose 35
        4.2.2. Identificação de uma Máquina Completamente Especificada 38
        4.2.3. Identificação de uma Máquina Minimal 39
        4.2.4. Identificação de uma Máquina Determinística 40
        4.2.5. Identificação de uma Máquina de Mealy 41
5. Conclusão 43
Bibliografia 44

Lista de Tabelas

Tabela 1 : Forma Geral de uma Tabela de Transição 5
Tabela 2 : Tabela de Transição da Máquina M1. 6
Tabela 3 : Aplicação de Seqüências de Entrada nos Estados 5 e 7 da Máquina M1. .. 7
Tabela 4 : Aplicação de Seqüências de Entrada de Comprimento 3
nos Estados 3 e 7 da Máquina M1. 7
Tabela 5 : Tabela P1 da Máquina M1. 10
Tabela 6 : Tabela P2 da Máquina M1. 10
Tabela 7 : Tabela P3 da Máquina M1. 12
Tabela 8 : Tabela P4 da Máquina M1. 12
Tabela 10 : Aplicação de uma Seqüência Minimal de Diagnose
nos Estados 2 e 6 da Máquina M1. 14

Lista de Figuras

Figura 1 : Máquina M1 5

Propriedades de Máquinas de Estado Finito relevantes para Critérios de Geração de Seqüências de Teste

Sumário

Este trabalho introduz conceitos fundamentais de Máquinas de Estado Finito (MEF) e propriedades básicas. Esses aspectos são relevantes para a proposição e implementação de uma ferramenta de geração de seqüências de teste baseados em MEF. Os algoritmos para verificar se uma determinada MEF satisfaz ou não tais propriedades são implementados.

1. Introdução

Este trabalho tem por objetivo introduzir alguns conceitos e propriedades básicas de Máquinas de Estado Finito pertinentes ao estabelecimento de critérios de geração de seqüências de teste baseados em Máquinas de Estado Finito [FUJI91]. Neste relatório não existe a preocupação de se ressaltar a relevância do estudo e da aplicação de Máquinas de Estado Finito na produção de software. Os conceitos, terminologia e algoritmos apresentados a seguir são extraídos essencialmente de [GILL62].

Segundo GILL [GILL62], a teoria de Máquinas de Estado Finito trata de modelos matemáticos que servem como aproximação de fenômenos físicos ou abstratos. Assim, seus modelos não são

limitados a uma determinada área científica, mas são aplicáveis praticamente em todo campo de investigação.

Este trabalho está organizado em cinco seções. Nesta seção foram apresentados os principais elementos abordados e os objetivos. A Seção 2 apresenta a técnica de modelagem de sistemas denominada Máquinas de Estado Finito (MEF). Na Seção 3 são introduzidas algumas propriedades e características que as MEF devem satisfazer para a aplicação de critérios de geração de seqüências de teste. A Seção 4 apresenta os algoritmos de identificação das propriedades e o algoritmo de geração das Tabelas Pk, definidas na Seção 3, e na Seção 5 é apresentada a conclusão deste trabalho.

2. Máquinas de Estado Finito

De forma esquemática, um sistema pode ser representado por uma "**caixa preta**" , com um número finito de terminais de acesso :

a) Os terminais de entrada, que representam as variáveis de entrada, isto é, os estímulos gerados pelo ambiente externo do sistema em questão.

b) Os terminais de saída, que representam as variáveis de saída, isto é, os aspectos de comportamento do sistema que são uma conseqüência dos estímulos.

Um sistema é considerado **síncrono** se verifica as seguintes hipóteses:

i) Todo sistema representável pelo modelo básico de Máquinas de Estado Finito é controlado por uma origem de sincronização. Dessa forma as variáveis do sistema não são medidas continuamente, mas somente em instantes discretos de tempo nos quais um certo evento especificado, chamado sinal de sincronização, é exibido pela origem .

ii) O comportamento do sistema num instante é independente do intervalo entre esse instante e o anterior; assim uma variável do sistema é relacionada a um valor num determinado instante e não depende do valor do tempo contínuo, mas do valor do instante discreto de tempo medido.

Cada variável pode assumir um número finito de valores distintos (que podem ser numéricos ou não) . O conjunto de valores de uma variável é chamado de **Alfabeto** dessa variável e cada elemento do Alfabeto é chamado de **símbolo**.

Definição 1: **Alfabeto de entrada** X é um conjunto finito de símbolos que descreve as variáveis de entrada do sistema.

Exemplo 1 : A MEF M1 da Figura 1 apresenta o Alfabeto de entrada $X=\{\alpha,\beta,\gamma\}$.

Definição 2: **Alfabeto de saída** Z é o conjunto finito de símbolos que descreve as variáveis de saída do sistema.

Exemplo 2 : A MEF M1 da Figura 1 apresenta o Alfabeto de saída $Z=\{0, 1\}$.

O "efeito combinado" entre as variáveis de entrada e de saída é chamado de **estado do sistema**, e está sujeito às hipóteses de sincronização, dessa forma pode-se denotar o estado no instante t_v pór S_v. O conceito de estado é definido pelas seguintes regras :

1) O símbolo de saída no presente instante v é determinado unicamente pelo símbolo de entrada e pelo estado no presente instante .
2) O estado no próximo instante é determinado unicamente pelo símbolo de entrada e pelo estado no presente instante.

Definição 3 : S é o **Conjunto de estados** do sistema.

Exemplo 3 : A MEF M1 da Figura 1 apresenta o Conjunto de estados $S=\{1,2,3,4,5,6,7,8,9\}$.

Definição 4 : Uma **MEF** M é uma classe de sistemas com um alfabeto de entrada $X=\{x_1,x_2,x_3,\ldots,x_p\}$, um alfabeto de saída $Z=\{z_1,z_2,z_3,\ldots,z_q\}$, um conjunto de estados $S=\{s_1,s_2,s_3,\ldots,s_n\}$ e um par de funções de caracterização f_z e f_s dados por:

i) $z_v = f_z(s_v,x_v)$ e

ii) $s_{v+1} = f_s(s_v,x_v)$

onde x_v, z_v e s_v são respectivamente o símbolo de entrada, o símbolo de saída e o estado da MEF M no instante t_v e s_{v+1} é o estado no instante t_{v+1}.

As MEF podem ser representadas pelas **Tabelas de Transição** através das funções de caracterização f_z e f_s, que representam os símbolos de saída e o próximo estado da MEF em relação aos símbolos de entrada. Estas tabelas representam todos os possíveis pares ordenados (x_v, s_v) de M, onde $x_v \in X$ e $s_v \in S$, e é formada por duas subtabelas Z_v e S_{v+1}, que representam f_z e f_s, respectivamente. As duas subtabelas têm a coluna mais à esquerda em comum que representa todos os possíveis estados e os cabeçalhos das colunas que representam todos os possíveis símbolos de entrada. Graficamente, as MEF podem ter seu comportamento representado através de **Grafos Direcionados** , sendo cada estado representado por um vértice e cada transição representada por uma aresta direcionada, onde cada aresta possui um rótulo x_i / z_i, com $f_z = z_i$ e o estado destino da aresta representando f_s (a partir deste ponto sempre que se referir a um Grafo Direcionado de uma MEF usa-se o termo Máquina). Na Tabela 1 apresenta-se a forma geral de uma Tabela de Transição, onde:

i) o elemento da subtabela Z_v que possui linha s_v e coluna x_v representa $f_z = (s_v, x_v)$, isto é, representa o símbolo de saída gerado por M quando é aplicado o símbolo de entrada x_v sobre M no estado s_v.

ii) o elemento da subtabela S_{v+1} que possui linha s_v e coluna x_v representa $f_s = (s_v, x_v)$, isto é, representa o próximo estado de M após a aplicação do símbolo de entrada x_v sobre M no estado s_v.

Tabela 1 : Forma Geral de uma Tabela de Transição.

	Z_v			S_{v+1}		
S_v \ X_v	x_1	...	x_p	x_1	...	x_p
s_1	$f_z(s_v, x_v)$			$f_s(s_v, x_v)$		
s_2						
...						
s_n						

Apresenta-se na Figura 1 a MEF M1 e sua respectiva Tabela de Transição na Tabela 2, retirada de [GILL62] .

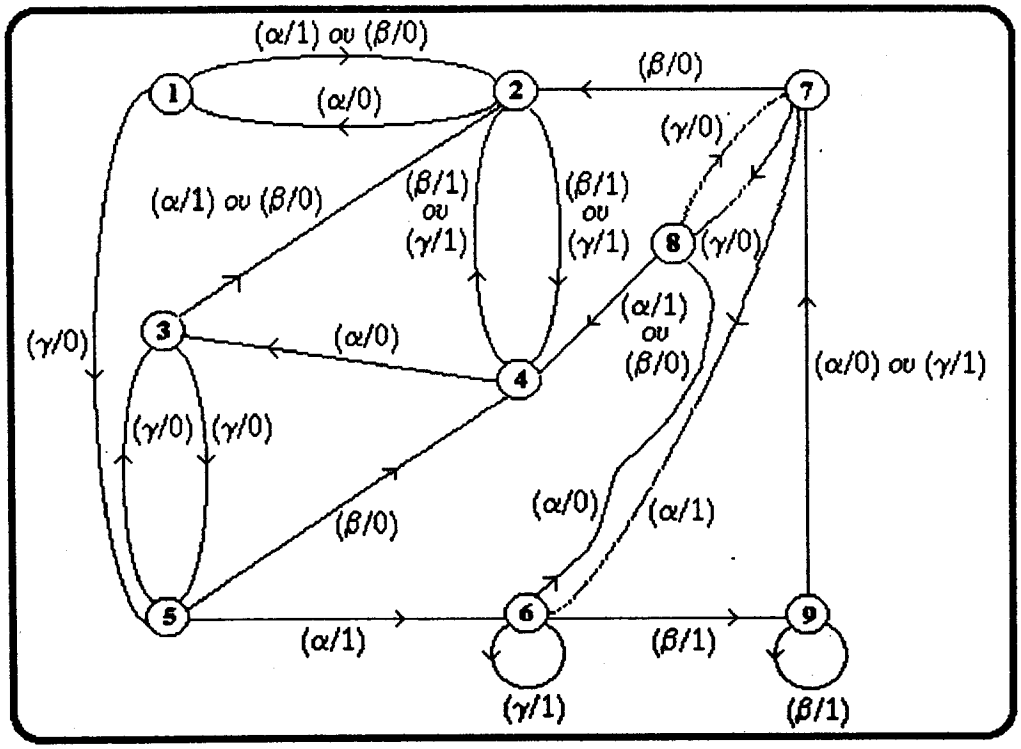


Figura 1 : Máquina M1.

Tabela 2 : Tabela de Transição da Máquina M1.

SV \ XV	α	β	γ	α	β	γ
1	1	0	0	2	2	5
2	0	1	1	1	4	4
3	1	0	0	2	2	5
4	0	1	1	3	2	2
5	1	0	0	6	4	3
6	0	1	1	8	9	6
7	1	0	0	6	2	8
8	1	0	0	4	4	7
9	0	1	1	7	9	7

Exemplo 4 : M1 é uma MEF que tem as suas funções de caracterização representadas na Tabela 2 .

Após a definição e as formas de representação de MEF, introduz-se na próxima seção, as características e propriedades das mesmas.

3. Propriedades e Características de MEF

Além das características e propriedades, são apresentados nessa seção, alguns conceitos necessários para o estabelecimento de critérios de geração de seqüências de teste baseados em MEF; assim, é usada a seguinte notação : $M|_{S_i}$ representa a MEF M no estado S_i.

Alguns dos principais conceitos apresentados a seguir serão as classes de equivalência e as Tabelas Pk, que geram todas as classes de equivalência da Máquina M.

Definição 5 : O estado S_i da MEF M1 e o estado S_j da MEF M2 são denominados **estados equivalentes** se $M1|_{S_i}$ e $M2|_{S_j}$ quando exercitados por qualquer seqüência de entrada, produzem saídas idênticas; caso contrário, são chamados **estados distinguíveis**. M1 e M2 podem representar a mesma Máquina.

Exemplo 5 : A MEF M1 apresenta dois estados equivalentes, 5 e 7, uma vez que estes estados sempre fazem parte de um mesmo grupo

nas Tabelas Pk (definição 7) de M1; Assim quando se aplica qualquer seqüência de entrada nesses estados, as seqüências de saída serão as mesmas para os dois estados, o que está exemplificado na Tabela 3.

Tabela 3 : Aplicação de Seqüências de Entrada nos Estados 5 e 7 da Máquina M1.

	$\alpha\beta\alpha\gamma$	$\alpha\alpha\gamma$	$\beta\beta\gamma$	$\gamma\beta\gamma$	$\alpha\alpha$
5	1100	100	011	001	10
7	1100	100	011	001	10

Definição 6 : Dois estados quaisquer da MEF M , S_i e S_j são chamados de **estados k-equivalentes** se quando exercitados por qualquer seqüência de comprimento k, produz seqüências de saída idênticas , caso contrário são chamados **estados k-distinguíveis** .

Exemplo 6 : A MEF M1 apresenta dois estados que são 3-equivalentes, 3 e 5 , uma vez que estes estados fazem parte de um mesmo grupo nas Tabelas P1, P2 e P3 de M1, representadas respectivamente pelas Tabelas 1, 2 e 3. Assim, quando se aplica qualquer seqüência de comprimento 3 sobre esses estados, as seqüências de saída serão as mesmas para cada um dos estados. Isto está exemplificado na Tabela 4.

Tabela 4 : Aplicação de Seqüências de Entrada de Comprimento 3 nos Estados 3 e 7 da Máquina M1.

	$\alpha\alpha\alpha$	$\beta\alpha\gamma$	$\gamma\beta\beta$	$\alpha\beta\gamma$	$\alpha\gamma\alpha$
3	101	000	001	111	110
7	101	000	001	111	110

Definição 7 : Uma MEF M pode ser dividida em partições de estado, que são **classes de equivalência**, dentro das quais têm-se os seguintes critérios :

(1) Todos os estados que se encontram na mesma partição devem ser k-equivalentes .

(2) Todos os estados que se encontram em partições diferentes devem ser k-distinguíveis .

Essas partições são chamadas de partições de k-equivalência e denotadas por Pk, podendo ser representadas pelas Tabelas Pk, com $k \geq 1$.

Exemplo 7 : As classes de equivalência da MEF M1 são representadas por todos os grupos da última Tabela Pk gerada, isto é, a Tabela P4. As classes de 3-equivalência de M1 são representadas por todos os grupos da Tabela P3. As Tabelas P4 e P3 são apresentadas mais adiante, pelas Tabelas 8 e 7, respectivamente.

Definição 8 : O estado em que a MEF M1 se encontra, depois de ser aplicada uma seqüência de entrada de comprimento k em $M1|_{S_i}$ é chamado de **k-ésimo sucessor** de S_i em relação a essa seqüência de entrada.

Exemplo 8 : O estado 5 da MEF M1 é um 4-ésimo sucessor do estado 1 de M1 em relação à seqüência de entrada $\alpha\beta\alpha\gamma$.

Definição 9 : Uma **Tabela Pk** de uma MEF M é essencialmente a mesma que a subtabela S_{v+1} dessa Máquina , com as seguintes modificações :

(1) Se $\{ S_{i_1}, S_{i_2}, \cdots, S_{i_r} \}$ é uma classe de Pk, as linhas $S_{i_1}, S_{i_2}, \cdots, S_{i_r}$ são colocadas juntas em **grupos**, sendo cada grupo separado dos adjacentes por uma "faixa". A ordem dos grupos e a ordem das linhas dentro de cada grupo é arbitrária. Linhas que pertencem ao mesmo grupo e, logo, pertencem à mesma classe de k-equivalência são chamadas **linhas associadas** e linhas que pertencem a diferentes grupos são chamadas de **linhas disjuntas**.

(2) Uma coluna Σ é adicionada, com **rótulos** que são atribuídos a cada grupo de linhas da Tabela Pk. Os rótulos são arbitrários e podem ser escolhidos de maneira independente em cada Tabela Pk.

(3) A todo estado gerado pela função f_s, na Tabela Pk, é justaposto o rótulo que foi atribuído ao grupo ao qual esse estado pertence. Assim, se a linha S_i pertence ao grupo rotulado por "a",

então, a toda ocorrência do estado S_i nessa Tabela Pk é justaposto o rótulo "a".

A construção da Tabela P1 é feita da seguinte forma:

(1) Verifique a Tabela de Transição de M e atribua um único rótulo a todos os estados que apresentam linhas idênticas na subtabela S_{v+1}, caracterizando grupos distintos de estados na Tabela P1.

(2) Associe a cada componente da subtabela S_{v+1}, o rótulo (atribuído em (1)) correspondente ao grupo da própria Tabela P1, ao qual aquele estado pertence.

A geração das Tabelas Pk ($k \geq 2$) segue o mesmo raciocínio utilizado na geração da Tabela P1, com a diferença de que agora, as informações não são mais extraídas da Tabela de Transição, mas sim da Tabela Pk-1.

Assim, pode-se descrever o procedimento utilizado para a geração das Tabelas Pk (≥ 2) da seguinte maneira :

(1) Se todos os estados de cada grupo da Tabela Pk-1 possuem os mesmos rótulos justapostos aos seus respectivos componentes da subtabela S_{v+1} da Tabela Pk-1, obecendo-se os símbolos de entrada, então não é possível gerar a Tabela Pk; caso contrário, siga para o passo (2).

(2) Verifique a Tabela Pk-1 e separe em grupos aqueles estados para os quais os rótulos justapostos a cada componente da subtabela S_{v+1} da Tabela Pk-1, nas linhas correspondentes a esses estados, são idênticos, obedecendo-se os símbolos de entrada;

Obs: ao separar-se os estados em novos grupos, os componentes da subtabela S_{v+1} da Tabela Pk-1, correspondentes a cada estado, devem ser transportados para a Tabela Pk sem o rótulo que foi justaposto quando da construção da Tabela Pk-1.

(3) Atribua, a cada um desses novos grupos gerados no passo (2), um rótulo;

(4) Justaponha, a cada componente da subtabela S_{v+1}, que compõe a Tabela Pk, o rótulo (atribuído no passo(3)) correspondente ao grupo da própria Tabela Pk, ao qual aquele estado pertence.

Exemplo 9 : As Tabelas Pk da MEF M1 são apresentadas a seguir, onde Tabela 5, Tabela 6, Tabela 7 e Tabela 8 representam respectivamente Tabela P1, Tabela P2, Tabela P3 e Tabela P4[GILL62].

Tabela 5 : Tabela P1 da Máquina M1.

Σ	SV \ XV	α	β	γ
a	1	2b	2b	5a
	3	2b	2b	5a
	5	6b	4b	3a
	7	6b	2b	8a
	8	4b	4b	7a
b	2	1a	4b	4b
	4	3a	2b	2b
	6	8a	9b	6b
	9	7a	9b	7a

Tabela 6 : Tabela P2 da Máquina M1.

Σ	SV \ XV	α	β	γ
a	1	2b	2b	5a
	3	2b	2b	5a
	5	6b	4b	3a
	7	6b	2b	8a
	8	4b	4b	7a
b	2	1a	4b	4b
	4	3a	2b	2b
	6	8a	9c	6b
c	9	7a	9c	7a

Tabela 7 : Tabela P3 da Máquina M1.

Σ	sv \ xv	α	β	γ
a	1	2b	2b	5a
	3	2b	2b	5a
	5	6c	4b	3a
	7	6c	2b	8a
	8	4b	4b	7a
b	2	1a	4b	4b
	4	3a	2b	2b
c	6	8a	9d	6c
d	9	7a	9d	7a

Tabela 8 : Tabela P4 da Máquina M1.

Σ	sv \ xv	α	β	γ
a	1	2b	2b	5c
	3	2b	2b	5c
	8	4b	4b	7c
b	2	1a	4b	4b
	4	3a	2b	2b
c	5	6d	4b	3a
	7	6d	2b	8a
d	6	8a	9e	6d
e	9	7c	9e	7c

Definição 10 : Uma **Seqüência de Diagnose** de dois estados S_i e S_j da MEF M é uma seqüência de símbolos de entrada que, quando aplicada em M no estado S_i e em M no estado S_j , produzem duas seqüências de símbolos de saída diferentes.

Exemplo 10 : A seqüência de símbolos de entrada βγααα é uma seqüência de diagnose para os estados 2 e 6 da Máquina M1 da

Figura 1 com sua Tabela de Transição na Tabela 2, uma vez que essa seqüência de entrada produz seqüências de saída diferentes para cada um dos estados 2 e 6 de M1, exemplificado na Tabela 9.

Tabela 9 : Aplicação de uma Seqüência de Diagnose nos Estados 2 e 6 da Máquina M1.

si \ sd	β γ α α α
2	1 1 0 1 0
6	1 1 1 0 1

Definição 11 : Uma seqüência de diagnose de dois estados S_i e S_j de uma MEF M que possui menor número de símbolos de entrada é chamada de **Seqüência Minimal de Diagnose** de S_i e S_j.

A geração de uma seqüência minimal de diagnose, denotada pela seqüência de símbolos de entrada $X_{u1}, X_{u2}, \ldots, X_{uk}$, de dois estados S_{i0} e S_{j0} da Máquina M pode ser definida da seguinte forma :

(1) Construa as Tabelas Pk da Máquina M.
(2) Calcule L tal que S_{i0} e S_{j0} sejam linhas associadas na Tabela P(L-1) e linhas disjuntas na Tabela PL. Seja k=1.
(3.a) Se (L-k) > 0 então siga para o passo **(4)**.
(3.b) Se (L-k) = 0 então X_{uk} será o símbolo de entrada da subTabela Z_v tal que as linhas $S_{i(k-1)}$ e $S_{j(k-1)}$ nesta coluna são diferentes.
(4) X_{uk} é o cabeçalho de qualquer coluna na Tabela P(L-k) tal que as linhas $S_{i(k-1)}$ e $S_{j(k-1)}$ nessa coluna, apresentem estados com rótulos diferentes. Esses estados são S_{ik} e S_{jk} respectivamente. Incremente k de 1 e retorne para **(3.a)**.

Exemplo 11 : Para os estados 2 e 6 da Máquina M1 da Figura 1 com sua Tabela de Transição na Tabela 2, calcula-se L=3 pois os estados 2 e 6 são linhas associadas na Tabela P2 representada pela Tabela 6 e são linhas disjuntas na Tabela P3 representada pela Tabela 7. Na

Tabela P2, observando as linhas dos estados 2 e 6, apenas o símbolo de entrada β apresenta os rótulos dos grupos diferentes, isto é, 4b e 9c; então o primeiro símbolo da seqüência minimal de diagnose é β. Na Tabela P1, observando as linhas dos estados 4 e 9, apenas o símbolo de entrada γ apresenta os rótulos dos grupos diferentes, isto é, 2b e 7a; então o segundo símbolo da seqüência minimal de diagnose é γ. Na subtabela Zv da Tabela de Transição da Máquina M1 representada pela Tabela 2, os estados 2 e 7 têm saídas diferentes em relação ao símbolo de entrada α ; então o terceiro símbolo da seqüência minimal de diagnose é α. Assim a seqüência minimal de diagnose para os estados 2 e 6 da Máquina M1 é βγα.

Tabela 10 : Aplicação de uma Seqüência Minimal de Diagnose nos Estados 2 e 6 da Máquina M1.

sd / e	β γ α
2	1 1 0
6	1 1 1

Definição 12 : Uma MEF M é **Completamente Especificada** se para cada estado existe uma transição para cada símbolo de entrada [KOHA78].

Exemplo 12 : A MEF M1 é Completamente Especificada, pois o número de linhas de sua Tabela de Transição (Tabela 2) é igual ao número de estados de M1 e todas as posições da Tabela de Transição estão preenchidas.

Definição 13 : Uma MEF M é **Minimal** se nenhum par de estados de M é equivalente [GILL62].

Exemplo 13 : A MEF M1 não é Minimal pois a Tabela P4, representada pela Tabela 8, não apresenta um único estado em cada grupo.

Definição 14 : Uma MEF M é **Determinística** se para cada entrada, existe no máximo uma transição definida em cada estado de M [BOYD91].

Exemplo 14 : A MEF M1 é Determinística pois o número de linhas de sua Tabela de Transição (Tabela 2) é igual ao número de estados de M1 e os elementos da subtabela S_{v+1} são todos distintos, em relação a cada símbolo de entrada.

Definição 15 : Uma MEF M é de **Mealy** se produz uma saída para cada transição [KOHA78].

Exemplo 15 : A MEF M1 é de Mealy pois o número de linhas de sua Tabela de Transição (Tabela 2) é igual ao número de estados de M1 e a subtabela Z_v está totalmente preenchida.

Depois de introduzidas as definições das características e propriedades relevantes para a geração de seqüências de teste, apresenta-se, a seguir, os algoritmos de verificação das mesmas, sendo em grande parte retirados de [GILL62].

4. Aspectos de Implementação

Nesta seção são descritos os algoritmos para a identificação das propriedades definidas na Seção 3 em relação à MEF M, além da descrição desses algoritmos e dos procedimentos usados nos mesmos.

Esses algoritmos foram baseados na representação da MEF M através de sua Tabela de Transição, mas isso não limita o domínio de aplicabilidade das MEF, uma vez que a geração de uma Tabela de Transição tendo como origem qualquer tipo de especificação da MEF é uma tarefa relativamente simples.

4.1. Principais Estruturas de Dados utilizadas nos Algoritmos

São apresentadas nesta seção, as estruturas de dados utilizadas na geração das Tabelas Pk e na identificação das propriedades de MEF definidas anteriormente na seção 3.

4.1.1. Conjunto de Estados

O conjunto S de estados da MEF M é representado pelo vetor S[1..n], onde n é o número de estados da Máquina M.

S =	S[1]	S[2]	...	S[n]

4.1.2. Conjunto de Símbolos de Entrada

O conjunto X de símbolos de entrada da MEF M é representado pelo vetor X[1..p], onde p é o número de símbolos de entrada da Máquina M.

A =	A[1]	A[2]	...	A[p]

4.1.3. Conjunto de Símbolos de Saída

O conjunto Z de símbolos de saída da MEF M é representado pelo vetor Z[1..q], onde q é o número de símbolos de saída da Máquina M.

Z =	Z[1]	Z[2]	...	Z[q]

4.1.4. Tabela de Transição

A tabela de transição da MEF M é formada por duas subtabelas Z_v e S_{v+1} que serão representadas respectivamente pelas matrizes FZ e FS.

4.1.4.1. Subtabela Zv

A matriz FZ[1..n,1..p] tem cada elemento FZ[i,j] representando o símbolo de saída gerado pela Máquina M no estado S[i] após a aplicação do símbolo de entrada A[j]. Assim a matriz FZ pode ser representada da seguinte forma :

FZ =

	1	2	3	...	p
1	FZ[1,1]	FZ[1,2]	FZ[1,3]	...	FZ[1,p]
2	FZ[2,1]	FZ[2,2]	FZ[2,3]	...	FZ[2,p]
...	...	...	...	...	...
...	...	...	...	...	...
n	FZ[n,1]	FZ[n,2]	FZ[n,3]	...	FZ[n,p]

4.1.4.2. Subtabela Sv+1

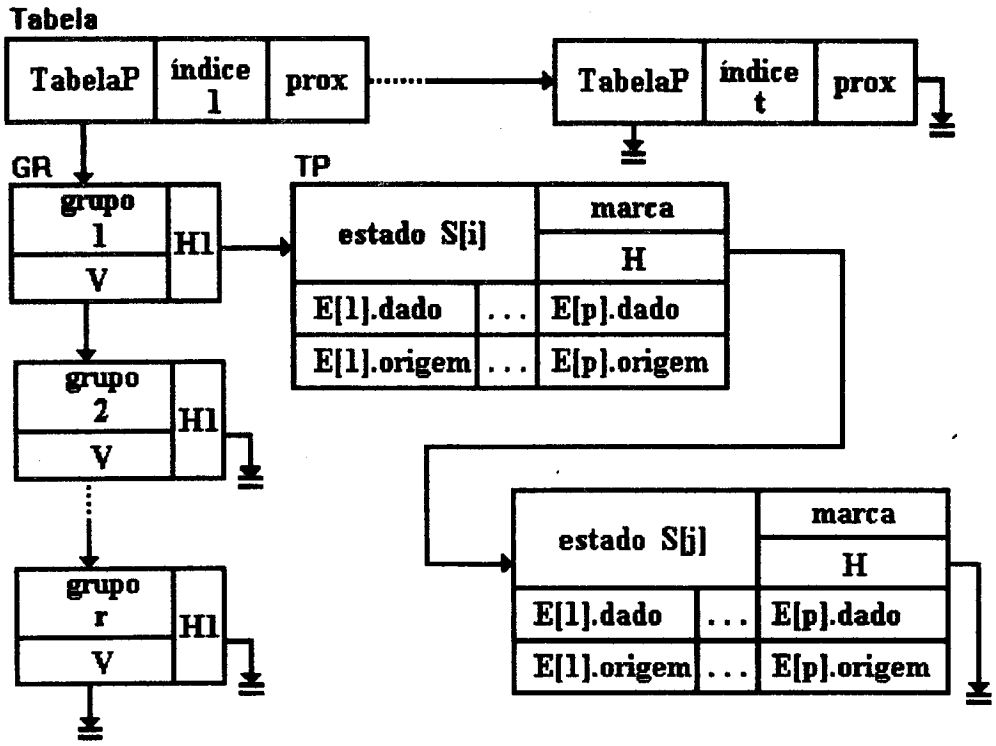
A matriz FS[1..n,1..p] tem cada elemento FS[i,j] indicando o próximo estado da Máquina M após a aplicação do símbolo de entrada A[j] sobre a Máquina M no estado S[i]. Assim, a matriz FS pode ser representada da seguinte forma :

FS =

	1	2	3	...	p
1	FS[1,1]	FS[1,2]	FS[1,3]	...	FS[1,p]
2	FS[2,1]	FS[2,2]	FS[2,3]	...	FS[2,p]
...	...	...	...	...	...
...	...	...	...	...	...
n	FS[n,1]	FS[n,2]	FS[n,3]	...	FS[n,p]

4.1.5. Tabelas Pk

As Tabelas Pk serão representadas por uma estrutura dinâmica denominada Tabela que pode ser esquematizada da seguinte forma :



É definida a seguir, a notação para cada um dos campos de :

A) Estrutura Tabela :

TabelaP = ponteiro para a lista de grupos.
índice = índice da Tabela Pk
prox = ponteiro para a próxima Tabela Pk da lista de Tabela

B) Estrutura GR :

grupo = rótulo que identifica cada grupo.
H1 = ponteiro para a lista de estados que fazem parte de um mesmo grupo.
V = ponteiro para o próximo grupo da lista de grupos.

C) Estrutura TP :

estado = identificação do estado.
marca = "flag" que indica se o estado já foi ou não incluído na Tabela Pk+1.
H = ponteiro para o próximo estado da lista de estados.
E[i].dado = próximo estado da Máquina M após a aplicação da entrada A[i] no estado identificado pelo campo estado.
E[i].origem = identificação do grupo de origem a que E[i].dado pertence na Tabela Pk.

4.1.6. Rótulo de Origem

A estrutura utilizada para auxiliar na rotulação dos estados da Tabela Pk em relação ao grupo de origem a que pertence cada um dos estados será representada pelo vetor C[1..n] e pode ser esquematizada da seguinte forma :

C =

C[1].dado	C[1].origem
C[2].dado	C[2].origem
...	...
...	...
C[n].dado	C[n].origem

Os campos do vetor C tem a seguinte notação :

C[i].dado = identificação do estado.

C[i].origem = identificação do grupo de origem do estado identificado em C[i].dado.

4.1.7. Índice da Seqüência Minimal de Diagnose

Esta estrutura é usada para auxiliar na localização do índice L das Tabelas Pk para a geração de uma seqüência minimal de diagnose de um par de estados (s_i,s_j) da Máquina M, e é denominada LOCAL.

LOCAL =

e1
e2

Os campos da estrutura LOCAL têm a seguinte notação:

e1 = rótulo de um estado s_i na Tabela Pk.

e2 = rótulo de um estado s_j na Tabela Pk.

4.2. Algoritmos de Geração e Identificação

São apresentadas nesta seção os seguintes itens dos algoritmos :

- Parâmetros de Entrada.
- Parâmetros de Saída.
- Descrição de Variáveis Locais.
- Procedimentos Utilizados.
- Procedimentos que o utilizam.
- Descrição em Pseudocódigo.

4.2.1. Geração de Tabelas Pk

O algoritmo de geração das Tabelas Pk utiliza o procedimento GeraçãoP1 para cálculo da Tabela P1, uma vez que esta é elaborada de forma diferente das demais e utiliza o procedimento GeraçãoPk de forma recursiva para o cálculo das Tabelas Pk, $k \geq 2$. Dentro deste algoritmo deve-se destacar os seguintes aspectos :

1) Parâmetro de Entrada :

TAB representando a Tabela Pk da Máquina M.
S1 representando o conjunto de estados da Máquina M.

2) Parâmetro de Saída :

TAB1 representando a Tabela Pk+1 da Máquina M.

3) Descrição de Variáveis Locais :

TAB : Tabela Pk corrente.
TAB1 : Tabela Pk+1.
GRP : grupo da Tabela Pk.
TPE : estado da Tabela Pk.
AUX1,
AUX5 : variáveis auxiliares de acesso aos grupos da Tabela Pk.
AUX2,
AUX3,
AUX4 : variáveis auxiliares de acesso aos estados de cada grupo da Tabela Pk.
I : contador de laço.

N : número de estados da Máquina M.
P : número de símbolos de entrada da Máquina M.
IDENT : variável auxiliar para índice dos grupos da Tabela Pk.
VAZIA : variável dinâmica ainda não inicializada.
C1 : vetor de auxílio para rotulação dos estados da Tabela Pk corrente.
CABECA_ORD : ponteiro para o primeiro grupo da Tabela P1.

4)Procedimentos Utilizados :

a)GeraçãoP1 b)ÚltimaTabela
c)IncluiOrigem d)ListaSemelhante
e)ObterÍndice

5)Procedimentos que o utilizam :

Nenhum.

6)Descrição do Algoritmo em Pseudocódigo :

```
Algoritmo GeraçãoPk(TAB)
início
  se (TAB = VAZIA)
    então
      /* Geração da Tabela P1 */
      GeraçãoP1(CABECA_ORD)
      Aloque espaço para TAB
      TAB.TabelaP:=CABECA_ORD
      TAB.índice:=1
      TAB.prox:=VAZIA
  fim se
  se não ÚltimaTabela(TAB)
    então
      para I := 1 até N faça
        C1[I].origem:=0
      fim para
      Aloque espaço para TAB1
      TAB.prox:=TAB1
      TAB1.prox:=VAZIA
      TAB1.índice:=TAB.índice+1
      /* Alocação para o primeiro grupo */
      Aloque espaço para GRP
      IDENT=1
      TAB.P:=GRP
      AUX1:=TAB.TabelaP
      enquanto (AUX1 ≠ VAZIA) faça
        /* Inclusão de um grupo */
        AUX2:=AUX1.H
        AUX2.marca:=1
        GRP.grupo:=IDENT
```

```
IDENT:=IDENT+1
GRP.V:=VAZIA
/*  Inclusão do primeiro estado de um grupo  */
Aloque espaço para TPE
GRP.H:=TPE
TPE.estado:=AUX2.estado
IncluiOrigem(AUX2.estado,GRP.grupo,C1)
TPE.marca:=0
para I :=1 até P faça
  TPE.E[I].dado:=AUX2.E[I].dado
fim para
TPE.H:=VAZIA
repita
  AUX3:=AUX2.H
  repita
    se ListaSemelhante(AUX2,AUX3)
      então
        /* Inclusão de um estado num grupo */
        AUX3.marca:=1
        AUX4:=TPE
        Aloque espaço para TPE
        AUX4.H:=TPE
        TPE.estado:=AUX3.estado
        IncluiOrigem(AUX3.estado,GRP.grupo,C1)
        TPE.marca:=0
        para I:=1 até P faça
          TPE.E[I].dado:=AUX.E[I].dado
        fim para
        TPE.H:=VAZIA
    fim se
    AUX3:=AUX3.H
  até(AUX3=VAZIA)
  fim repita
  repita
    AUX2:=AUX2.H
  até(AUX2.marca=0  ou  AUX2=VAZIA)
  fim repita
  se(AUX2.marca=0)
     então
       /*  Inclusão de um novo grupo resultante de quebra
           de um grupo    */
       AUX5:=GRP
       Aloque espaço para GRP
       AUX5.V:=GRP
       GRP.grupo:=IDENT
       IDENT:=IDENT+1
       GRP.V:=VAZIA
       /*  Inclusão do primeiro estado do novo grupo  */
       Aloque espaço para TPE
       GRP.H:=TPE
       TPE.estado:=AUX2.estado
       IncluiOrigem(AUX2.estado,GRP.grupo,C1)
       TPE.marca:=0
       para I:=1 até P faça
         TPE.E[I].dado:=AUX2.E[I].dado
       fim para
       TPE.H:=VAZIA
       AUX2:=AUX2.H
  fim se
até(AUX2=VAZIA)
```

```
            fim repita
            AUX1:=AUX1.V
            AUX5:=GRP
            Aloque espaço para GRP
            AUX5.V:=GRP
          fim enquanto
          AUX1:=Tab1.P
          repita
            /* Rotulação dos estados com suas origens  */
            AUX2:=AUX1.H1
            repita
              para I:=1 até P faça
                ObterÍndice(AUX2.E[I].dado,índice,C1,S1)
                AUX2.E[I].origem:=indice1
              fim para
            até(AUX2=VAZIA)
            fim repita
            AUX1:=AUX1.V
          até(AUX1=VAZIA)
          fim repita
    fim se
     se não ÚltimaTabela(TAB1)
       então
         /* Geração da próxima Tabela Pk  */
         GeraçãoPk(TAB1)
     fim se
fim GeraçãoPk
```

Neste algoritmo são utilizados alguns procedimentos que serão descritos também em pseudocódigo nas seções a seguir.

4.2.1.1. Geração da Tabela P1

O procedimento de geração da Tabela P1 é descrito em separado porque ele utiliza como base as informações da Tabela de Transição da Máquina M e não as informações da Tabela Pk anterior, como no procedimento para o cálculo das Tabelas Pk, k≥2.

1)Parâmetro de Entrada :

Subtabela Z_v da Máquina M representada por FZ1.
Conjunto de estados da Máquina M representado por S1.

2)Parâmetro de Saída :

Tabela P1 da Máquina M representada por CABECA_ORD.

3) Descrição de Variáveis Locais :

GRP : grupo da Tabela P1.
TPE : estado da Tabela P1.
AUX1 : variável auxiliar de acesso aos grupos da Tabela P1.
AUX : variável auxiliar de acesso aos estados de cada grupo da Tabela P1.
I,K : contadores de laço.
N : número de estados da Máquina M.
P : número de símbolos de entrada da Máquina M.
IDENT : variável auxiliar para índice dos grupos da Tabela P1.
VAZIA : variável dinâmica ainda não inicializada.
C1 : vetor de auxílio para rotulação dos estados da Tabela P1.
INDICE : rótulo do estado da Tabela P1.
CABECA : variável auxiliar para identificar grupos da Tabela P1.

4)Procedimentos Utilizados :

a)LinhasSemelhantes b)ObterÍndice

5)Procedimentos que o utilizam :

GeraçãoPk.

6)Descrição do Algoritmo em Pseudocódigo :

```
Procedimento GeraçãoP1(S1,FS1,CABECA_ORD)
início
  CABECA:=VAZIA
  para I:=1 até N faça
    C1[I].estado:=0
  fim para
  I:=1
  IDENT:=1
  /*  Alocação do primeiro grupo   */
  Aloque espaço para GRP
  CABECA:=GRP
  GRP.V:=VAZIA
  enquanto(I<N) faça
    se(C1[I].estado=0)
      então
        /*  Inclusão de um novo grupo e do
            primeiro estado desse grupo      */
        C1[I].estado:=S[I]
        Aloque espaço para TPE
        TPE.marca:=0
        TPE.estado:=S[I]
        GRP.grupo:=IDENT
        C1[I].origem:=GRP.grupo
        GRP.H:=TPE
        IDENT:=IDENT+1
        para K:=1 até P faça
          TPE.E[K].dado:=FS1[I,K]
        fim para
        AUX1:=GRP
        /*  Alocação de um novo grupo     */
        Aloque espaço para GRP
        AUX1.V:=GRP
        GRP.V:=VAZIA
    fim se
    para J:=I+1 até n faça
      se(C1[J].estado=0)
        então
          se LinhasSemelhantes(I,J)
            então
              /*  Inclusão de um estado num  grupo   */
              C1[J].estado:=S[J]
              C1[J].origem :=AUX1.grupo
              AUX:=TPE
              Aloque espaço para TPE
              TPE.estado:=S[J]
              TPE.marca:=0
              para K:=1 até P faça
                TPE.E[K].dado:=FS1[J,K]
              fim para
              AUX.H:=TPE
              TPE.H:=VAZIA
          fim se
    fim para
    I=I+1
  fim enquanto
  AUX1:=CABECA_ORD
  /*  Rotulação  dos estados  com suas origens  */
```

```
    repita
      AUX:=AUX1.H
      repita
        para I:=1 até P faça
          ObterÍndice(AUX.E[I].dado,INDICE,C1,S1)
          AUX.E[I].origem:=INDICE
        fim para
        AUX:=AUX.H
      até(AUX=VAZIA)
      AUX1:=AUX1.V
    até(AUX1=VAZIA)
  fim GeraçãoP1
```

4.2.1.2. Identificação da Última Tabela Pk

Neste procedimento identifica-se se é possível gerar mais uma Tabela Pk+1 através da Tabela Pk corrente.

1)Parâmetro de Entrada :

Tabela Pk da Máquina M representada por TAB1.

2)Parâmetro de Saída :

Nenhum.

3) Descrição de Variáveis Locais :

SUCESSO : variável de caracterização da Tabela Pk.
AUX1 : variável auxiliar de acesso aos grupos da Tabela Pk.
AUX2
AUX3 : variáveis auxiliares de acesso aos estados de cada grupo da Tabela Pk.
I : contador de laço.
P : número de símbolos de entrada da Máquina M.
VAZIA : variável dinâmica ainda não inicializada.

4)Procedimentos Utilizados :

Nenhum.

5)Procedimentos que o utilizam :

GeraçãoPk

6)Descrição do Algoritmo em Pseudocódigo :

```
Procedimento ÚltimaTabela(TAB1)
início
  SUCESSO:=verdadeiro
  AUX1:=TAB1.P
  repita
    AUX2:=AUX1.H
    repita
      se (AUX2.H ≠ VAZIA)
        então
          AUX3:=AUX2.H
          para I:=1 até P faça
            se(AUX2.E[I].origem ≠ AUX3.E[I].origem)
              então SUCESSO:=falso
            fim se
          fim para
      fim se
      AUX2:=AUX2.H
    até(AUX2=VAZIA)
    fim repita
  até(AUX1=VAZIA)
  fim repita
  ÚltimaTabela:=SUCESSO
fim ÚltimaTabela
```

4.2.1.3. Inclusão da Origem

Como é utilizado o vetor C1 para auxiliar na rotulação dos grupos de origem de cada um dos estados nas Tabelas Pk, este procedimento inclui cada estado no vetor C1 quando esse estado é gerado pela Tabela Pk.

1)Parâmetro de Entrada :

Conjunto de estados da Máquina M representado por S1.
Identificação do estado da Tabela Pk representado por IDENTIFICADOR.
Identificação do rótulo do estado da Tabela Pk referenciado por identificador e representado por ID_G.

2)Parâmetro de Saída :

Vetor de auxílio para rotulação dos estados da Tabela Pk, representado por C1.

3) Descrição de Variáveis Locais :

I : contador de laço.
N : número de estados da Máquina M.

4)Procedimentos Utilizados :

Nenhum.

5)Procedimentos que o utilizam :

GeraçãoPk.

6)Descrição do Algoritmo em Pseudocódigo :

```
Procedimento IncluiOrigem(IDENTIFICADOR,ID_G,S1,C1)
início
  para I:=1 até N faça
    se (S[I]=IDENTIFICADOR)
      então
        C1[I].origem:=ID_G
        C1[I].estado:=IDENTIFICADOR
    fim se
  fim para
fim IncluiOrigem
```

4.2.1.4. Obtenção do Índice

Este procedimento obtém o grupo de origem de um determinado estado no vetor S1 que seja rotulado pela variável IDENTIFICADOR na Tabela Pk corrente.

1)Parâmetro de Entrada :

Conjunto de estados da Máquina M representado por S1.
Vetor de auxílio para rotulação dos estados da Tabela Pk, representado por C1.
Indentificação do estado da Tabela Pk representado por IDENTIFICADOR.

2)Parâmetro de Saída :

Rótulo do estado da Tabela Pk representado por INDICE.

3) Descrição de Variáveis Locais :

I : contador de laço.
N : número de estados da Máquina M.

4)Procedimentos Utilizados :

Nenhum.

5)Procedimentos que o utilizam :

a)GeraçãoPk. b)GeraçãoP1.

6)Descrição do Algoritmo em Pseudocódigo :

```
Procedimento ObterÍndice(IDENTIFICADOR,INDICE,C1,G1)
início
  para I:=1 até N faça
    se(S[I]=IDENTIFICADOR)
      então
        INDICE:=C1[I].origem
    fim se
  fim para
fim ObterÍndice
```

4.2.1.5. Identificação de Linhas Semelhantes

Neste procedimento identifica-se se dois estados são disjuntos ou associados de acordo com as suas respectivas linhas na subtabela Z_v da Tabela de Transição da Máquina M.

1)Parâmetro de Entrada :

Subtabela Z_v da Máquina M, representada por FZ1.
Dois estados da subtabela Z_v da Máquina M, representados por I e J.

2)Parâmetro de Saída :

Nenhum.

3) Descrição de Variáveis Locais :

L : contador de laço.
N : número de estados da Máquina M.
SUCESSO : variável de caracterização dos estados da subtabela Z_v da Máquina M.

4)Procedimentos Utilizados :

Nenhum.

5)Procedimentos que o utilizam :

GeraçãoP1.

6)Descrição do Algoritmo em Pseudocódigo :

```
Procedimento LinhasSemelhantes(I,J)
início
 SUCESSO:=verdadeiro
 para L:=1 até P faça
    se(FZ1[I,L] ≠ FZ1[J,L]
       então
          SUCESSO:=falso
    fim se
 fim para
 LinhasSemelhantes:=SUCESSO
fim LinhasSemelhantes
```

4.2.1.6. Identificação de Listas Semelhantes

Este procedimento identifica se dois estados são disjuntos ou associados de acordo com as suas respectivas linhas da tabela Pk corrente da Máquina M.

1)Parâmetro de Entrada :

Dois estados da Tabela Pk da Máquina M, representados por CAB1 e CAB2.

2)Parâmetro de Saída :

Nenhum.

3) Descrição de Variáveis Locais :

I : contador de laço.
N : número de estados da Máquina M.
SUCESSO : variável de caracterização dos estados da Tabela Pk da Máquina M.

4)Procedimentos Utilizados :

Nenhum.

5)Procedimentos que o utilizam :

GeraçãoPk.

6)Descrição do Algoritmo em Pseudocódigo :

```
Procedimento ListasSemelhantes(CAB1,CAB2)
início
  SUCESSO:=verdadeiro
  para I:=1 até P faça
    se(CAB1.E[I].dado ≠ CAB2.E[I].dado)
      então
        SUCESSO:=falso
    fim se
  fim para
  ListasSemelhantes:=SUCESSO
fim ListasSemelhantes
```

4.2.1.7. Geração de uma Seqüência Minimal de Diagnose.

Este algoritmo gera uma seqüência minimal de diagnose para um par de estados S_i e S_j da Máquina M, de acordo com o procedimento apresentado anteriormente neste trabalho.

1)Parâmetro de Entrada :

Dois estados S_i e S_j da Máquina M, representados por EST1 e EST2. Tabelas Pk representadas por TAB.

2)Parâmetro de Saída :

Seqüência Minimal de Diagnose de EST1 e EST2, representada por SMD.

3) Descrição de Variáveis Locais :

SUCESSO : variável auxiliar.
AUX1,
GAUX1 : variáveis auxiliares de acesso às Tabelas Pk.
GAUX2,
GAUX2a,
GAUX1a : variáveis auxiliares de acesso aos estados de cada grupo da Tabela Pk.
L,K : índices auxiliares.
ESTA1,
ESTA2 : variáveis auxiliares.
I : contador de laço.
P : número de símbolos de entrada da Máquina M.
VAZIA : variável dinâmica ainda não inicializada.

4)Procedimentos Utilizados :

Buscal.

5)Procedimentos que o utilizam :

Nenhum.

6)Descrição do Algoritmo em Pseudocódigo :

```
Algoritmo GSMD(TAB,EST1,EST2,SMD)
início
  Buscal(TAB,EST1,EST2,L)
  K:=1
  ESTA1:=EST1
  ESTA2:=EST2
  repita
    se(L-K)=0
      então
        IND1:=índice de ESTA1
        IND2:=índice de ESTA2
        SUCESSO:=falso
        I:=1
        enquanto(não SUCESSO ou I≤P) faça
          se (FZ[IND1,I] ≠ FZ[IND2,I])
            então
              SMD:=SMD+A[I]
              SUCESSO:=verdadeiro
          fim se
          I:=I+1
        fim enquanto
```

```
        senão
          AUX+1:=TAB
          SUCESSO:=falso
          enquanto(AUX1 ≠ vazia ou não SUCESSO) faça
            se(AUX1.índice=L-K)
              então
                SUCESSO:=verdadeiro
              senão
                AUX1:=AUX1.prox
            fim se
          fim enquanto
          GAUX1:=AUX1.P
          GAUX2:=GAUX1.H
          enquanto (GAUX1 ≠ VAZIA) faça
            enquanto (GAUX2 ≠ VAZIA) faça
              se(ESTA1=GAUX2.estado)
                então
                  GAUX1a:=GAUX2
              fim se
              se(ESTA2=GAUX2.estado)
                então
                  GAUX2a:=GAUX2
              fim se
            fim enquanto
            GAUX1:=GAUX1.V
          fim enquanto
          I:=1
          SUCESSO:=falso
          enquanto(não SUCESSO ou I≤P)  faça
            se(GAUX1a.E[I].origem ≠ GAUX2a.E[I].origem)
              então
                SMD:=SMD+A[I]
                SUCESSO:=verdadeiro
                K:=K+1
                ESTA1:=GAUX1a.E[I].dado
                ESTA2:=GAUX2a.E[I].dado
            fim se
            I:=I+1
          fim enquanto
        fim se
      até(L-K=0)
      fim repita
    fim GSMD
```

4.2.1.8. Geração do Índice da Seqüência Minimal de Diagnose

Este algoritmo faz uma verificação entre as Tabelas Pk para calcular o índice t, onde na Tabela Pt os estados s_i e s_j são linhas disjuntas e na Tabela P(t-1) os estados são linhas associadas.

1)Parâmetro de Entrada :

Dois estados da Máquina M, representados por EST1 e EST2.
Tabelas Pk representadas por TAB.

2)Parâmetro de Saída :

Índice L, representado por INDL.

3) Descrição de Variáveis Locais :

ACHOU : variável auxiliar.
AUX1,
AUX2 : variáveis auxiliares de acesso às Tabelas Pk.
GAUX1,
GAUX2 : variáveis auxiliares de acesso aos grupos das Tabelas Pk.
L,K : índices auxiliares.
GID1,
GID2 : variáveis auxiliares.
LOCA1,
LOCA2 : variáveis auxiliares de estrutura LOCAL.
ES1,
ES2 : varíaveis auxiliares de acesso aos estados dos grupos das Tabelas Pk.
EST1,
EST2 : variáveis auxiliares.
VAZIA : variável dinâmica ainda não inicializada.

4)Procedimentos Utilizados :

Nenhum.

5)Procedimentos que o utilizam :

GSMD.

6)Descrição do Algoritmo em Pseudocódigo :

```
Procedimento Buscal(TAB,EST1,EST2,INDL)
início
  ACHOU:=falso
  AUX1:=TAB    INDL:=0
  AUX2:=TAB.prox
  enquanto(AUX1.prox ≠ VAZIA ou não ACHOU) faça
    enquanto(AUX2 ≠ VAZIA ou não ACHOU) faça
      GAUX1:=AUX1.P
      GAUX2:=AUX2.P
      enquanto(GAUX1 ≠ VAZIA) faça
        GID1:=GAUX1.grupo
        ES1:=GAUX1.H1
        enquanto(ES1 ≠ VAZIA) faça
          se(ES1.estado=EST1)
            então
              LOCA1.e1:=GID1
          fim se
          se(ES1.estado=EST2)
            então
              LOCA1.e2:=GID1
          fim se
          ES1:=ES1.H
        fim enquanto
        GAUX1:=GAUX1.V
      fim enquanto
      enquanto(GAUX2 ≠ VAZIA) faça
        GID2:=GAUX2.grupo
        ES2:=GAUX2.H1
        enquanto(ES2 ≠ VAZIA) faça
          se(ES2.estado=EST1)
            então
              LOCA2.e1:=GID2
          fim se
          se(ES2.estado=EST2)
            então
              LOCA2.e2:=GID2
          fim se
          ES2:=ES2.H
        fim enquanto
        g2:=g2.V
      fim enquanto
      se(LOCA1.e1 ≠ LOCA1.e2)
        então
          INDL:=1
          ACHOU:=verdadeiro
      fim se
      se(LOCA1.e1=LOCA1.e2 e LOCA2.e1=LOCA2.e2)
        então
          INDL:=AUX2.indice
          ACHOU:=verdadeiro
      fim se
      AUX2:=AUX2.prox
    fim enquanto
    AUX1:=AUX1.prox
  fim enquanto
fim Buscal
```

4.2.2. Identificação de uma Máquina Completamente Especificada

Neste algoritmo é feita uma varredura nas subtabela Z_v e S_{v+1} da Máquina M para verificar se todas as entradas dessas subtabelas estão preenchidas, caracterizando a Máquina como Completamente Especificada ou não.

1)Parâmetro de Entrada :

Subtabela Z_v da Máquina M, representada por FZ1.
Subtabela S_{v+1} da Máquina M, representada por FS1.

2)Parâmetro de Saída :

Nenhum.

3) Descrição de Variáveis Locais :

I,J : contadores de laço.
N : número de estados da Máquina M.
SUCESSO : variável de identificação da propriedade da Máquina M.

4)Procedimentos Utilizados :

Nenhum.

5)Procedimentos que o utilizam :

Nenhum.

6)Descrição do Algoritmo em Pseudocódigo :

```
Algoritmo Especificada(FZ1,FS1)
início
  SUCESSO:=verdadeiro
  para I:=1 até N faça
    para J:=1 até P faça
      se(FZ1[I,J]=' ' ou FS1[I,J]=' ')
        então
          SUCESSO:=falso
      fim se
    fim para
  fim para
  Especificada:=SUCESSO
fim Especificada
```

4.2.3. Identificação de uma Máquina Minimal

O algoritmo de identificação de minimalidade da Máquina M parte do princípio de que se a última Tabela Pk gerada apresentar o número de grupos idêntico ao número de estados da Máquina M, a propriedade é verificada.

1)Parâmetro de Entrada :

Subtabela Z_V da Máquina M, representada por FZ1.

2)Parâmetro de Saída :

Nenhum.

3) Descrição de Variáveis Locais :

SUCESSO : variável de identificação da propriedade da Máquina M.
AUX,
AUX1 : variáveis auxiliares de acesso às Tabelas Pk.
AUX2,
AUX3 : variáveis auxiliares de acesso aos grupos das Tabelas Pk.
VAZIA : variável dinâmica ainda não inicializada.

4)Procedimentos Utilizados :

GeraçãoPk.

5)Procedimentos que o utilizam :

Nenhum.

6)Descrição do Algoritmo em Pseudocódigo :

```
Algoritmo Minimal(F1)
ínicio
  SUCESSO:=verdadeiro
  Tab:=VAZIA
  GeraçãoPk(FZ1,Tab)
  AUX:=Tab
  repita
    AUX1:=AUX
    AUX:=AUX.prox
  até (AUX=VAZIA)
  fim repita
  AUX2:=AUX1.P
  repita
    AUX3:=AUX2
    AUX2:=AUX.V
  até (AUX2=VAZIA)
  fim repita
  se(AUX3.grupo ≠ N)
     então
        SUCESSO:=falso
  fim se
  Minimal:=SUCESSO
fim Minimal
```

4.2.4. Identificação de uma Máquina Determinística

Neste algoritmo é verificada a propriedade de determinismo através da comparacão do número de linhas da subtabela S_{v+1} com o número de estados da Máquina M e o número de colunas da subtabela S_{v+1} com o número de símbolos de entrada da Máquina M.

1)Parâmetro de Entrada :

Número de estados da Máquina M, representado por N.
Número de símbolos de entrada da Máquina M, representado por P.
Número de linhas da subtabela S_{v+1} da Máquina M, representado por R.
Número de colunas da subtabela S_{v+1} da Máquina M, representado por S.

2)Parâmetro de Saída :

Nenhum.

3) Descrição de Variáveis Locais :

SUCESSO : variável de identificação da propriedade da Máquina M.

4)Procedimentos Utilizados :

Nenhum.

5)Procedimentos que o utilizam :

Nenhum.

6)Descrição do Algoritmo em Pseudocódigo :

```
Algoritmo Determinística(R,S,N,P)
início
  SUCESSO:=verdadeiro
  se(R ≠ N)
    então
      SUCESSO:=falso
  fim se
  se(S ≠ P)
    então
      SUCESSO:=falso
  fim se
  Determinística:=SUCESSO
fim Determinística
```

4.2.5. Identificação de uma Máquina de Mealy

Este algoritmo verifica se a Máquina M é de Mealy através da comparacão do número de linhas da subtabela Z_v com o número de estados da Máquina M e do número de colunas da subtabela Z_v com o número de símbolos de entrada da Máquina M.

1)Parâmetro de Entrada :

Número de estados da Máquina M, representado por N.
Número de símbolos de entrada da Máquina M, representado por P.
Número de linhas da subtabela Z_N da Máquina M, representado por R1.
Número de colunas da subtabela Z_N da Máquina M, representado por S1.

2)Parâmetro de Saída :
Nenhum.

3) Descrição de Variáveis Locais :

SUCESSO : variável de identificação da propriedade da Máquina M.

4)Procedimentos Utilizados :

Nenhum.

5)Procedimentos que o utilizam :

Nenhum.

6)Descrição do Algoritmo em Pseudocódigo :

```
Algoritmo Mealy(R1,S1,N,P)
início
  SUCESSO:=verdadeiro
  se(R1 ≠ N)
    então
      SUCESSO:=falso
  fim se
  se(S1 ≠ P)
    então
      SUCESSO:=falso
  fim se
  Mealy:=SUCESSO
fim Mealy
```

5. Conclusão

Para um sistema sob a forma de uma Máquina M, foram apresentados neste trabalho as definições das propriedades de Especificação Completa, Minimalidade, Determinismo, e de Máquina de Mealy e a implementação dos algoritmos de verificação dessas propriedades, além dos algoritmos de geração de Tabelas Pk e de Seqüências Minimais de Diagnose. Estas implementações são necessárias na geração das seqüências básicas de M [NAKA94a] e na aplicação de critérios de geração de seqüências de teste baseados em Máquinas de Estado Finito [NAKA94b].

Uma descrição detalhada desses algoritmos e de suas estruturas de dados, que também são utilizadas em [NAKA94a] e [NAKA94b], foi apresentada na Seção 4.

Bibliografia

[BOYD91] BOYD, S.C. and URAL, H., " On the Complexity of Generating Optimal Test Sequences ", in IEEE Trans. on Software Engineering, Vol.17, No.9, September 1991, pp.976-978 .

[FUJI91] FUJIWARA, S. et alli.," Test Selection Based on Finite-State Models " , in IEEE Trans. on Software Engineering, Vol.17, No.6, June 1991, pp.591-603 .

[GILL62] GILL, A., Introduction to the Theory of Finite-State Machines, New York : McGraw-Hill, 1962 .

[KOHA78] KOHAVI, Z., Switching and Finite Automata Theory, New York : McGraw-Hill, 1978.

[NAKA94a] NAKAZATO, K.K., MALDONADO, J.C., FABBRI, S.C.P.F. e MASIERO, P.C. "Seqüências Básicas de Máquinas de Estado Finito : Aspectos Teóricos e de Implementação.", Relatório Técnico do Instituto de Ciências Matemáticas de São Carlos - Universidade de São Paulo, 1994 (em elaboração).

[NAKA94b] NAKAZATO, K.K., MALDONADO, J.C., FABBRI, S.C.P.F. e MASIERO, P.C. "Critérios de Geração de Seqüências de Teste baseados em Máquinas de Estado Finito" , Relatório Técnico do Instituto de Ciências Matemáticas de São Carlos - Universidade de São Paulo, 1994 (em elaboração).