Instituto de Ciências Matemáticas de São Carlos

ISSN - 0109-2569

Seqüências básicas de máquinas de estado finito: aspectos teóricos e de implementação

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

Nº 28

RELATÓRIOS TÉCNICOS DO ICMSC

SÃO CARLOS
Out. / 1994

SYSNO 872271
DATA / /
ICMC - SBAB

Seqüências Básicas de Máquinas de Estado Finito : Aspectos Teóricos e de Implementação

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. Conceitos e Seqüências Básicas de MEF 5
    3.1 Conceitos 6
    3.2 Seqüências Básicas de MEF 13
4. Comparação entre Seqüências Básicas de MEF 23
5. Aspectos de Implementação 27
    5.1. Principais Estruturas de Dados utilizadas nos Algoritmos 28
        5.1.1. Árvore de Sincronização 28
        5.1.2. Seqüência de Sincronização 29
        5.1.3. Árvore de Distinção 30
        5.1.4. Seqüência Distinguível 31
        5.1.5. Estruturas para Geração de Seqüências Únicas de Entrada/Saída 31
            5.1.5.1. Conjunto de Rótulos Distintos 31
            5.1.5.2. Conjunto de Arestas 31
            5.1.5.3. Vetor de seqüências de comprimento (L-1) 31
            5.1.5.4. Vetor de estados finais para seqüências de comprimento 32
            5.1.5.5. Vetor de duplas de estados para seqüências de comprimento 32
            5.1.5.6. Vetor de seqüências de comprimento L 32
            5.1.5.7. Vetor de estados finais para seqüências de comprimento L 32
            5.1.5.8. Vetor de duplas de estados para seqüências de comprimento L 33
        5.1.6. Conjunto de Caracterização 33
    5.2. Análise dos Algoritmos 34
        5.2.1. Geração de Seqüências de Sincronização 34
        5.2.2. Geração de Seqüências Distinguíveis 35
        5.2.3. Geração de Seqüências Únicas de Entrada/Saída 36
        5.2.4. Geração do Conjunto de Caracterização 36
6. Conclusão 37
Bibliografia 38

Lista de Tabelas

Tabela 1 : Forma Geral de uma Tabela de Transição 3
Tabela 2 : Tabela de Transição da Máquina M1 4
Tabela 3 : Tabela de Transição da Máquina M2 5
Tabela 4 : Conjunto de a-Antecessores do Estado 1. 7
Tabela 5 : Tabela de Transição da Máquina M3 8
Tabela 6: I1 de I1 9
Tabela 7 : I2 de I2 9
Tabela 8: Tabela de Transição da Máquina M4 10
Tabela 9 : Tabela de Transição da Máquina M5 12
Tabela 10 : Tabela de Transição da Máquina M6 15
Tabela 11 : Tabela de Seqüências de Sincronização da Máquina M6 16
Tabela 11.1 : Aplicação de SS(1) à Máquina M6. 17
Tabela 11.2 : Aplicação de SS(2) à Máquina M6. 17
Tabela 11.3 : Aplicação de SS(3) à Máquina M6. 17
Tabela 11.4 : Aplicação de SS(4) à Máquina M6. 18
Tabela 11.5 : Aplicação de SS(5) à Máquina M6. 18
Tabela 11.6 : Aplicação de SS(6) à Máquina M6. 18
Tabela 12 : Seqüências Distinguíveis da Máquina M6. 19
Tabela 12.1 : Aplicação de DS1 à Máquina M6. 20
Tabela 12.2 : Aplicação de DS2 à Máquina M6. 20
Tabela 12.3 : Aplicação de DS3 à Máquina M6. 20
Tabela 12.4 : Aplicação de DS4 à Máquina M6. 21
Tabela 13 : Seqüências Únicas de Entrada/Saída da Máquina M6. 21
Tabela 14 : Aplicação do Conjunto de Caracterização na Máquina M6. 22
Tabela 15 : Tabela de Transição da Máquina M7 25
Tabela 16 : Tabela de Transição da Máquina M8 27

Apêndice A

A1. Geração da Árvore de Sincronização 40
A2. Inclusão de um Estado numa Incerteza da Árvore de Sincronização 42
A3. Busca de um Nó na Árvore de Sincronização 44
A4. Geração da Seqüência de Sincronização 45
A5. Geração da Árvore de Distinção 47
A6. Inclusão de um Estado numa Incerteza da Árvore de Distinção 49
A7. Busca de um Nó na Árvore de Distincão 52
A8. Identificação de um Nó Simples na Árvore de Distinção 54
A9. Identificação de um Nó Múltiplo na Árvore de Distinção 55
A10. Geração da Seqüência Distinguível 56
A11. Geração das Sequências Únicas de Entrada/Saída 58
A12. Geração do Conjunto de Caracterização 60
A13. Inclusão de uma Sequência Minimal de Diagnose no Conjunto de Caracterização 61
A14. Semelhança entre Sequências do Conjunto de Caracterização e a Sequência Minimal de Diagnose 62

Lista de Figuras

Figura 1 : Máquina M1 4
Figura 2 : Máquina M2 5
Figura 3: Máquina M3 7
Figura 4: Máquina M4 9
Figura 5: Árvore Sucessora da Máquina M2 11
Figura 6 : Máquina M5 12
Figura 7 : Árvore de Sincronização da Máquina M5. 12
Figura 8 : Árvore de Distinção da Máquina M2. 13
Figura 9 : Máquina M6 14
Figura 10 : Árvore de Sincronização da Máquina M6. 15
Figura 11 : Árvore de Distinção da Máquina M6. 19
Figura 12 : Máquina M7 25
Figura 13 : Árvore de Sincronização da Máquina M7. 25
Figura 14 : Árvore de Distinção da Máquina M7. 26
Figura 15 : Máquina M8 26
Figura 16 : Árvore de Distinção da Máquina M8. 27
Figura 17 : Árvore de Sincronização da Máquina M8. 27

Seqüências Básicas de Máquinas de Estado Finito : Aspectos Teóricos e de Implementação

Sumário

Neste trabalho são apresentados os conceitos fundamentais para a geração de seqüências básicas de Máquinas de Estado Finito (MEF). Essas seqüências consistem basicamente de seqüências especiais de símbolos de entrada para a especificação e implementação de uma ferramenta de geração de seqüências de teste baseada em MEF. Os algoritmos para a geração dessas seqüências são também apresentados.

1. Introdução

Inicialmente são introduzidos os conceitos essenciais para a identificação e compreensão de seqüências especiais de símbolos de entrada, as chamadas seqüências básicas, que são utilizadas no estabelecimento de critérios de geração de seqüências de teste baseados em Máquinas de Estado Finito (MEF), assim como os algoritmos para a geração dessas seqüências. Os conceitos, terminologia e algoritmos apresentados a seguir foram extraídos essencialmente de [GILL62].

As seqüências são, segundo GILL [GILL62], soluções para o problema de identificação de estados, que consiste em:

i) identificar o estado inicial de uma MEF, isto é, o estado no qual a Máquina se encontra quando é apresentada para as atividades de teste.

ii) identificar o estado final de uma MEF, isto é, o estado no qual a Máquina se encontra quando as atividades de teste são completadas.

Essas soluções, de acordo com GILL [GILL62], tornam possível fazer prognósticos a respeito do comportamento da Máquina, que é descrita através de Tabelas de Transição.

Este trabalho está organizado em cinco seções. Nesta Seção foram apresentados os objetivos e principais tópicos abordados. A Seção 2

apresenta uma descrição sucinta da técnica de modelagem MEF. Na Seção 3 são introduzidos conceitos necessários à definição de seqüências básicas para MEF. A Seção 4 apresenta algumas comparações entre seqüências básicas. Na Seção 5 são apresentadas as condições e propriedades que as MEF devem satisfazer para a efetiva geração das seqüências básicas, além dos algoritmos de geração e a análise dos mesmos .

2. Máquinas de Estado Finito

Definição 1 : 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 quando se referir a um Grafo Direcionado de uma MEF será usado 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						

Exemplo 1 : Seja M1 uma MEF, representada na Figura 1 e com seu comportamento descrito na Tabela 2 e a MEF M2 representada na Figura 2 com seu comportamento descrito na Tabela 3.

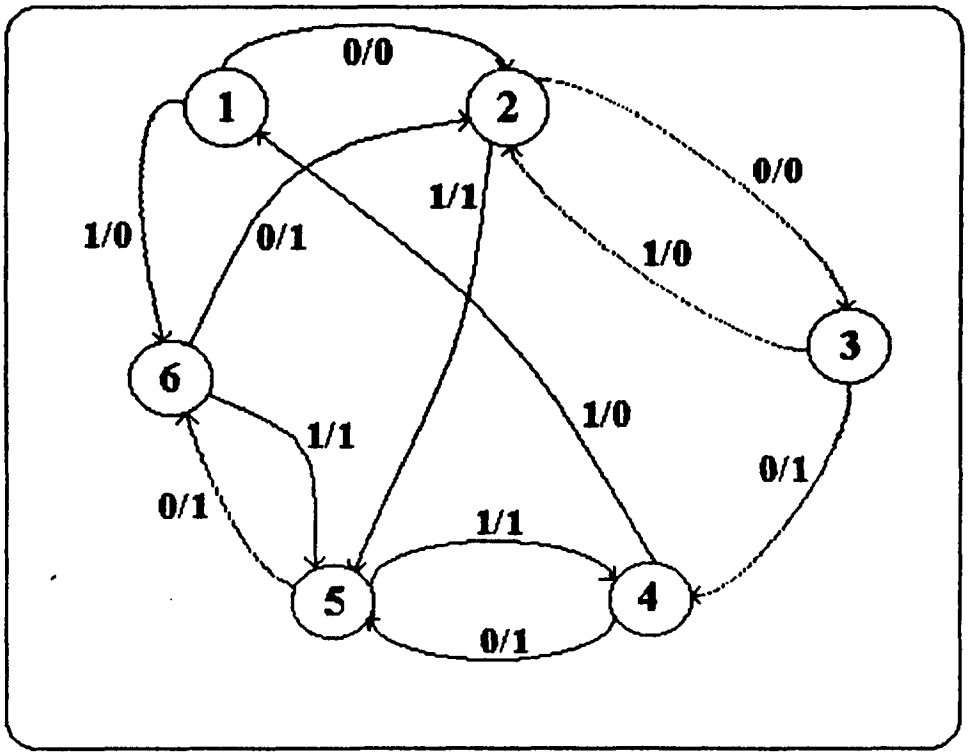


Figura 1 : Máquina M1.

Tabela 2 : Tabela de Transição da Máquina M1.

S \ X	0	1	0	1
1	0	0	2	6
2	0	1	3	5
3	1	0	4	2
4	1	0	5	1
5	1	1	6	4
6	1	1	2	5

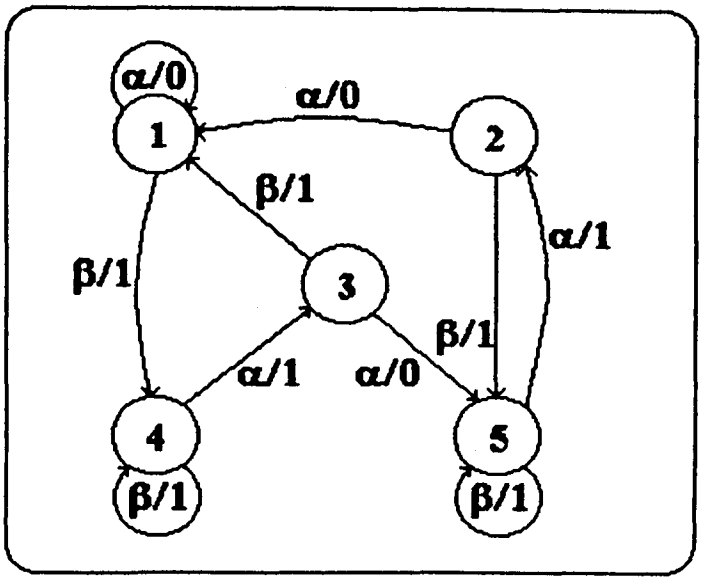


Figura 2 : Máquina M2.

Tabela 3 : Tabela de Transição da Máquina M2.

Sv \ X	α	β	α	β
1	0	1	1	4
2	0	1	1	5
3	0	1	5	1
4	1	1	3	4
5	1	1	2	5

Após a definição de MEF e de suas formas de representação, são identificadas , a seguir, as seqüências básicas de MEF que satisfazem certas propriedades.

3. Conceitos e Seqüências Básicas de MEF

Nesta seção serão introduzidos os conceitos de incerteza, tipos de incerteza, conjunto admissível, grupo A, árvore sucessora, árvore de sincronização e árvore de distinção, necessários à definição das seguintes seqüências básicas de MEF :

- seqüência de identificação .
- seqüência de sincronização (SS) .
- seqüência distinguível (DS) .

- seqüência única de entrada/saída (UIO) .
- conjunto de caracterização .

Neste trabalho supõe-se que todas as MEFs consideradas satisfaçam as seguintes propriedades :

- Minimalidade;
- Especificação Completa ;
- Máquina de Mealy;
- Determinismo;
- Conectividade Forte.

Todas essas propriedades têm sua definição e algoritmos de identificação apresentadas em [NAKA94a], exceto a última propriedade, que é definida na Sub-Seção 3.2.

3.1 Conceitos

O conceito de **Incerteza** reflete as situações em que não se pode identificar em qual estado a MEF se encontrava, observando somente o símbolo de saída gerado.

Definição 2 : Se a MEF M no estado s_i for estimulada por um símbolo de entrada x_i, levando a Máquina M ao estado s_j, o estado s_i é x_i-**Antecessor** do estado s_j.

Exemplo 2 : Na MEF M2 da Figura 2 com sua Tabela de Transição na Tabela 3, se a Máquina M2 estiver no estado 2 e for aplicado o símbolo de entrada α levando a Máquina M2 ao estado 1, então o estado 2 é α -Antecessor do estado 1.

Definição 3 : Uma **Incerteza** I de uma MEF M em relação a um símbolo de entrada x_i é um conjunto de estados de M onde os x_i-antecessores dos estados de I produzem os mesmos símbolos de saída quando estimulados por x_i.

Exemplo 3 : Na MEF M2 representada pela Figura 2, com sua Tabela de Transição dada na Tabela 3, o conjunto I={1,1,5} é uma incerteza em relação ao símbolo de entrada α pois os estados 1,2 e 3 são α -antecessores dos estados de I. Dessa forma tem-se a Tabela 4 :

Tabela 4 : Conjunto de α-Antecessores do Estado 1.

S_V \ X_V	α*	α**
1	0	1
2	0	1
3	0	5

onde a coluna α^* representa o símbolo de saída gerado e α^{**} representa o próximo estado de M2; assim, os elementos do conjunto {1,2,3} de α-antecessores de I, quando estimulados pelo símbolo de entrada α, geram o mesmo símbolo de saída 0.

Definição 4 : Uma incerteza que contém um único estado é chamada de **Incerteza Simples**.

Exemplo 4 : Seja M3 uma MEF representada pela Figura 3, com sua Tabela de Transição dada na Tabela 5.

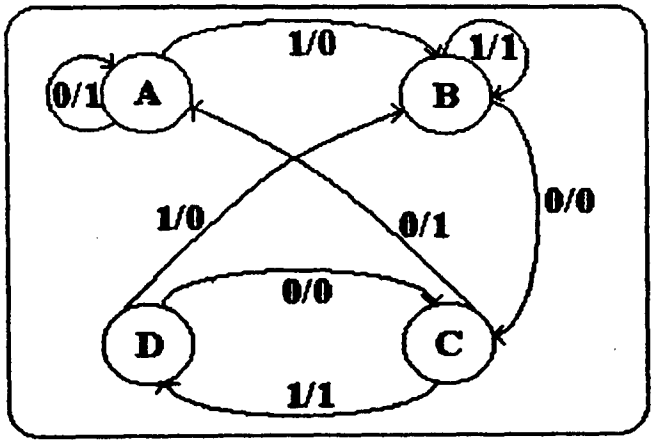


Figura 3: Máquina M3.

Tabela 5 : Tabela de Transição da Máquina M3.

S_r \ X_v	0	1	0	1
A	1	0	A	B
B	0	1	C	B
C	1	1	A	D
D	0	0	C	B

O conjunto {A} é uma incerteza simples em relação ao símbolo de entrada 0, pois o conjunto {A,C} é um conjunto de 0-antecessores do conjunto {A}.

Definição 5 : Uma incerteza que contém dois ou mais estados idênticos é chamada de **Incerteza Múltipla**.

Exemplo 5 : Na MEF M2 da Figura 2, o conjunto I={1, 1, 5} apresentado no Exemplo 1 é uma Incerteza Múltipla em relação ao símbolo de entrada α, pois o estado 1 ocorre duas vezes em I.

Definição 6 : Um **Conjunto Admissível** da MEF M é um conjunto de estados $\{S_{i_1}, S_{i_2}, \ldots, S_{i_m}\}$ de M, representado por A(M), onde os estados são candidatos a estado inicial de M e é representado por A(M).

Exemplo 6 : Na MEF M3 da Figura 3, tem-se A(M3) = {A, B, C, D}, onde os estados de A(M3) são candidatos a estado inicial da Máquina M3.

Definição 7 : Para uma Máquina M cujo conjunto admissível tem tamanho m, um **Grupo A** é um conjunto de incertezas onde m é o número total de estados de todas as incertezas incluídas no grupo A .

Exemplo 7 : Na MEF M2 da Figura 2 com sua Tabela de Transição na Tabela 3, o conjunto de incertezas GA={{1,1,5},{3,2}} é um grupo A em relação ao símbolo de entrada α, pois pode-se estabelecer os

conjuntos de α-antecessores de cada uma das incertezas de GA, usando a mesma notação do Exemplo 3 da seguinte forma :

Tabela 6: αI1 de I1.

S_σ \ X_σ	α*	α**
1	0	1
2	0	1
3	0	5

Tabela 7 : αI2 de I2.

S_σ \ X_σ	α*	α**
4	1	3
5	1	2

O conjunto I1={1,1,5} é uma incerteza pois para o conjunto αI1 = {1,2,3} de α-antecessores de I1, cada estado de αI1 quando estimulado pelo símbolo de entrada α, gera o símbolo de saída 0 e tem como próximo estado um estado de I1. Da mesma forma o conjunto I2={3,2} é uma incerteza pois para o conjunto αI2={4,5} de α-antecessores de I2, cada estado de αI2 quando estimulado pelo símbolo de entrada α, gera o símbolo de saída 1 e tem como próximo estado um estado de I2.

Definição 8 : Um grupo A onde todas as incertezas são simples é chamado de **Grupo A Simples** .

Exemplo 8 : Seja M4 uma MEF representada pela Figura 4 com sua Tabela de Transição dada na Tabela 8. O conjunto GA={{A},{B}} em relação ao símbolo de entrada 0 é um grupo A simples, pois as incertezas de GA são incertezas simples em relação ao símbolo de entrada 0.

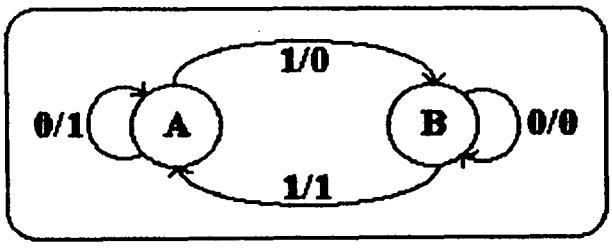


Figura 4: Máquina M4.

Tabela 8: Tabela de Transição da Máquina M4.

S_Y \ X_Y	0	1	0	1
A	1	0	A	B
B	0	1	B	A

Definição 9 : Uma **Árvore Sucessora** é uma estrutura definida para uma MEF M e um conjunto admissível A(M). Essa estrutura é composta de ramos arranjados em níveis sucessivos onde o primeiro Nível é representado por zero, o segundo Nível por um e assim por diante. Todo ramo da árvore sucessora, exceto o ramo inicial, representa um símbolo de entrada a_i, com $a_i \in X$, e cada ramo está associado com um nó, sendo este nó um grupo A. O ramo inicial está associado com um nó formado por A(M). Cada nó N_j no Nível k , gera p nós no Nível (k+1) da seguinte forma : Seja $N_j = \{I_1, I_2, \ldots, I_r\}$ um nó associado ao ramo que representa a_i, no Nível k. Serão gerados p nós em relação a cada $a_i \in X$, denotados por $N'_{j_{a_i}} = \{A_{1_{a_i}}, A_{2_{a_i}}, \ldots, A_{r_{a_i}}\}$, que estarão no Nível (k+1) e serão chamados de filhos de N_j. Cada $A_{t_{a_i}}$ de $N'_{j_{a_i}}$ será um grupo A gerado pela incerteza It em relação ao símbolo de entrada a_i , para todo $I_t \in N_j$.

Exemplo 9 : A Figura 5 representa a árvore sucessora da MEF M2 até o Nível 2, uma vez que uma árvore sucessora é infinita segundo a definição apresentada.

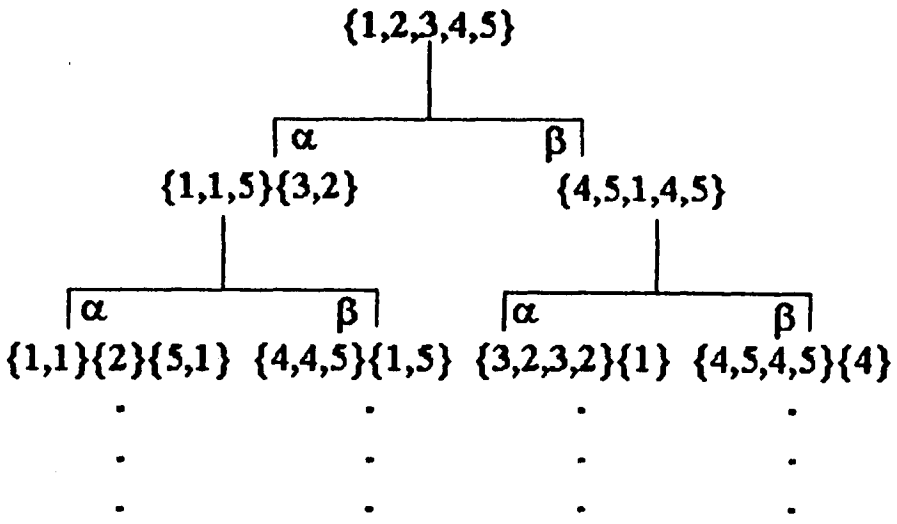


Figura 5: Árvore Sucessora da Máquina M2.

Definição 10 : A seqüência de t ramos tal que o k-ésimo ramo está no k-ésimo Nível (k=1,2,...,t) e tal que o (k+1)-ésimo ramo é gerado pelo k-ésimo ramo (k=1,2,...,t-1) é chamado de **Caminho da Árvore Sucessora**.

Exemplo 10 : Em relação a árvore sucessora da Figura 5, a seqüência de símbolos de entrada αβ, associada com o ramo r1 formado pelo grupo A GA1 = {{1,1,5},{3,2}} e representando o símbolo de entrada α no Nível 1 e com o ramo r2 formado pelo grupo A GA2 = {{4,4,5},{1,5}} gerado por r1 e representando o símbolo de entrada β no Nível 2, é considerado um caminho dessa árvore sucessora.

Definição 11 : Uma **Árvore de Sincronização** de uma MEF M é uma árvore sucessora construída ignorando-se as saídas e associando cada nó no j-ésimo Nível com a incerteza que corresponde aos estados finais que resultam da aplicação dos j-primeiros símbolos de entrada. Um nó no j-ésimo Nível da árvore de sincronização torna-se terminal quando este nó já ocorreu anteriormente na árvore de sincronização num Nível anterior.

Exemplo 11 : Seja M5 uma MEF representada pela Figura 6 com sua Tabela de Transição na Tabela 9. A árvore de sincronização da Máquina M5 é apresentada na Figura 7 e tem A(M5) = {A,B,C,D}.

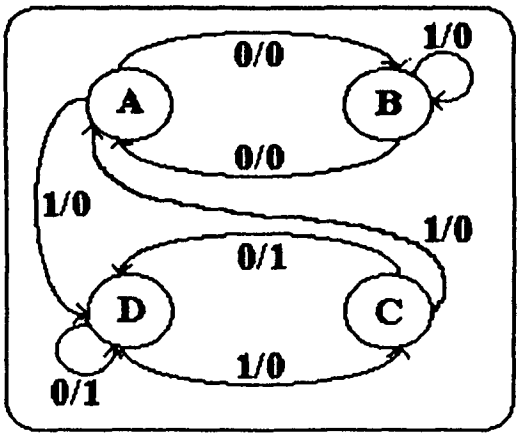


Figura 6 : Máquina M5.

Tabela 9 : Tabela de Transição da Máquina M5.

Sv \ Xv	0	1	0	1
A	0	0	B	D
B	0	0	A	B
C	1	0	D	A
D	1	0	D	C

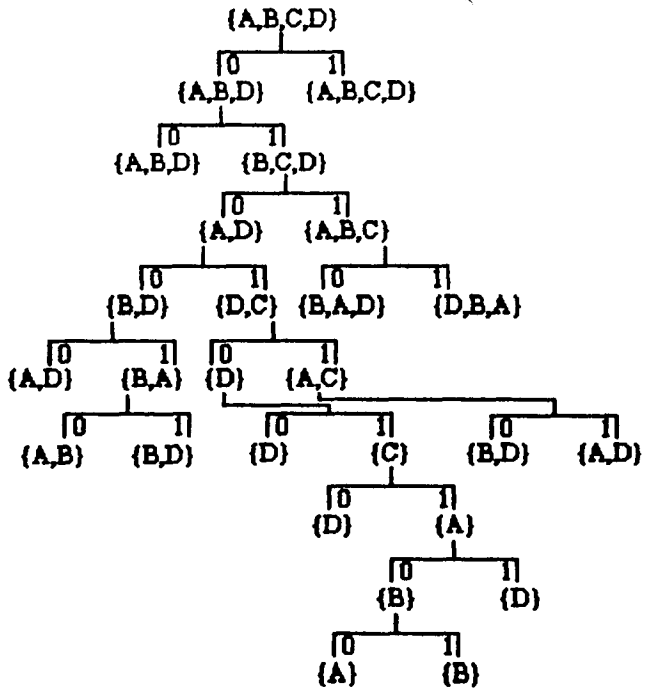


Figura 7 : Árvore de Sincronização da Máquina M5.

Definição 12 : Uma **Árvore de Distinção** de uma MEF M é uma árvore sucessora onde um nó no k-ésimo Nível torna-se terminal quando :

(1) O grupo A associado ao nó apresenta uma incerteza múltipla.

(2) O grupo A associado ao nó já ocorreu na árvore de distinção num Nível anterior a k ou no próprio Nível k.

(3) O grupo A associado ao nó é um grupo A simples.

Exemplo 12 : Seja M5 uma MEF representada pela Figura 6, com sua Tabela de Transição dada na Tabela 9. A árvore de distinção de M5 é apresentada na Figura 8 e tem A(M5) = {A,B,C,D}.

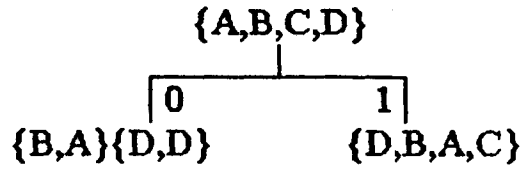


Figura 8 : Árvore de Distinção da Máquina M2.

Em seguida serão apresentadas as definições das seqüências básicas de uma MEF, utilizando os conceitos apresentados nesta seção.

3.2 Seqüências Básicas de MEF

As denominadas **seqüências básicas**, isto é, a seqüência de identificação, a seqüência de sincronização, a seqüência distinguível e a seqüência única de entrada/saída, definidas neste trabalho, são pertinentes ao estabelecimento de critérios de geração de seqüências de teste baseados em MEF.

Definição 13 : Uma **Seqüência de Identificação** de uma MEF M é uma seqüência de símbolos de entrada que tem como objetivo identificar, de maneira única, cada um dos estados da Máquina M.

Exemplo 13 : Uma Seqüência Distinguível (DS) (definição 15) de uma MEF M é uma Seqüência de Identificação uma vez quando se aplica a mesma em cada um dos estados da Máquina M, as seqüências de saída obtidas serão diferentes para cada um dos estados, possibilitando assim identificar de forma única cada um dos estados da Máquina M.

Definição 14 : Uma **Seqüência de Sincronização** (SS) de uma MEF M é uma seqüência de símbolos de entrada que leva a Máquina M a um estado final especificado, independente da saída gerada ou do estado inicial [KOHA78].

Uma Seqüência de Sincronização é descrita por um caminho na Árvore de Sincronização do nó inicial até um nó formado por um grupo A simples. Assim, esta seqüência leva a Máquina M de qualquer estado para o estado que forma o grupo A simples.

Exemplo 14 : Seja M6 uma MEF representada pela Figura 9, com sua Tabela de Transição dada na Tabela 10 e sua Árvore de Sincronização na Figura 10.

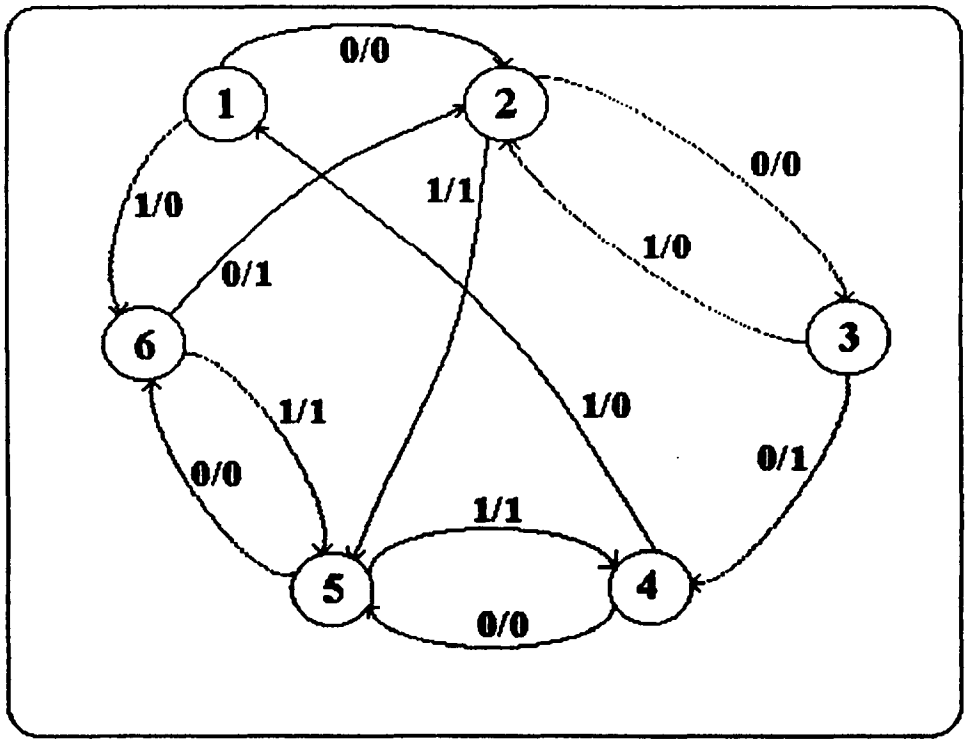


Figura 9 : Máquina M6.

Tabela 10 : Tabela de Transição da Máquina M6.

S \ E	0	1	0	1
1	0	0	2	6
2	0	1	3	5
3	1	0	4	2
4	0	0	5	1
5	0	1	6	4
6	1	1	2	5

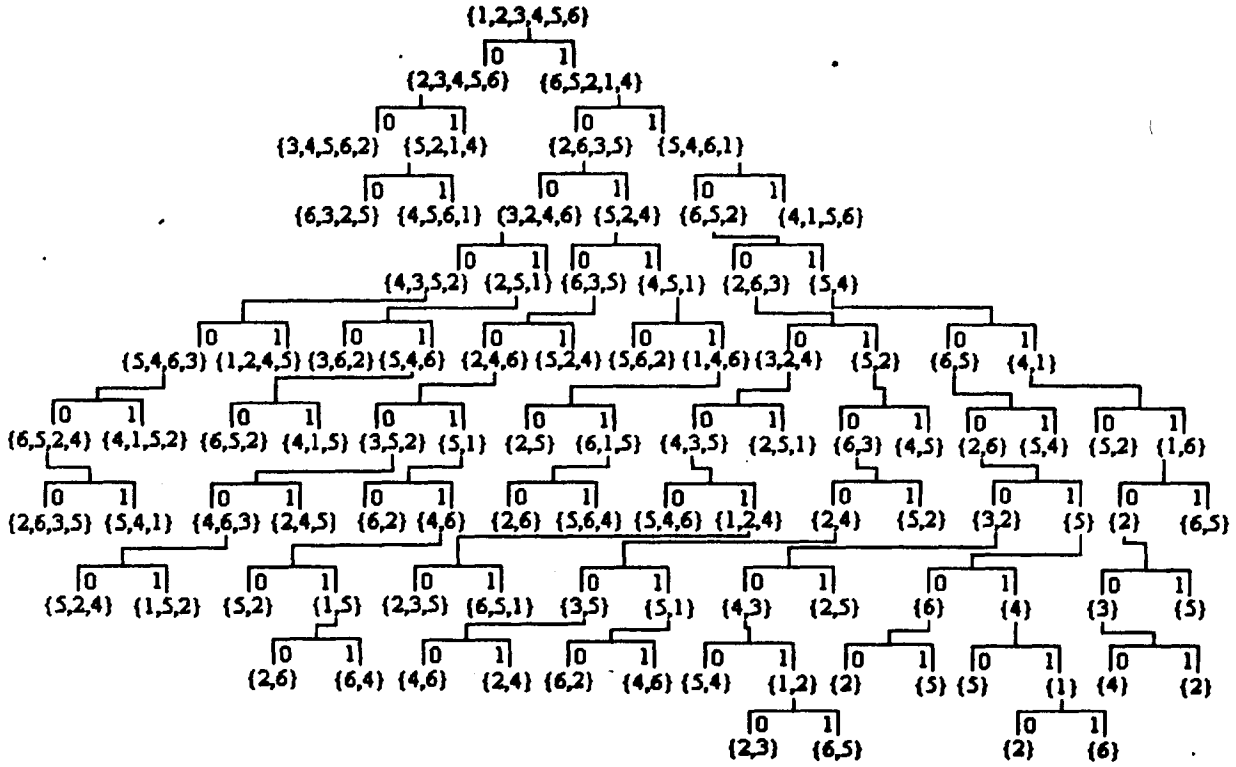


Figura 10 : Árvore de Sincronização da Máquina M6.

De acordo com o algoritmo apresentado no Apêndice A, pode-se estabelecer a Tabela 11 com as seqüências mais curtas de sincronização de todos os estados da Máquina M6 uma vez que a árvore de sincronização dessa Máquina permite a identificação de mais de uma seqüência de sincronização para alguns estados .

Tabela 11 : Tabela de Seqüências de Sincronização da Máquina M6.

SS(1)	110100111
SS(2)	1101110
SS(3)	11011100
SS(4)	11010011
SS(5)	1101001
SS(6)	11010010

Tabela 11.1 : Aplicação de SS(1) à Máquina M6.

Sv \ SS(1)	1	1	0	1	0	0	1	1	1
1	6	5	6	5	6	2	5	4	1
2	5	4	5	4	5	6	5	4	1
3	2	5	6	5	6	2	5	4	1
4	1	6	2	5	6	2	5	4	1
5	4	1	2	5	6	2	5	4	1
6	5	4	5	4	5	6	5	4	1

Tabela 11.2 : Aplicação de SS(2) à Máquina M6.

Sv \ SS(2)	1	1	0	1	1	1	0
1	6	5	6	5	4	1	2
2	5	4	5	4	1	6	2
3	2	5	6	5	4	1	2
4	1	6	2	5	4	1	2
5	4	1	2	5	4	1	2
6	5	4	5	4	1	6	2

Tabela 11.3 : Aplicação de SS(3) à Máquina M6.

Sv \ SS(3)	1	1	0	1	1	1	0	0
1	6	5	6	5	4	1	2	3
2	5	4	5	4	1	6	2	3
3	2	5	6	5	4	1	2	3
4	1	6	2	5	4	1	2	3
5	4	1	2	5	4	1	2	3
6	5	4	5	4	1	6	2	3

Tabela 11.4 : Aplicação de SS(4) à Máquina M6.

S_T \ SS(4)	1	1	0	1	0	0	1	1
1	6	5	6	5	6	2	5	4
2	5	4	5	4	5	6	5	4
3	2	5	6	5	6	2	5	4
4	1	6	2	5	6	2	5	4
5	4	1	2	5	6	2	5	4
6	5	4	5	4	5	6	5	4

Tabela 11.5 : Aplicação de SS(5) à Máquina M6.

S_T \ SS(5)	1	1	0	1	0	0	1
1	6	5	6	5	6	2	5
2	5	4	5	4	5	6	5
3	2	5	6	5	6	2	5
4	1	6	2	5	6	2	5
5	4	1	2	5	6	2	5
6	5	4	5	4	5	6	5

Tabela 11.6 : Aplicação de SS(6) à Máquina M6.

S_T \ SS(6)	1	1	0	1	0	0	1	1	0
1	6	5	6	5	6	2	5	4	6
2	5	4	5	4	5	6	5	4	6
3	2	5	6	5	6	2	5	4	6
4	1	6	2	5	6	2	5	4	6
5	4	1	2	5	6	2	5	4	6
6	5	4	5	4	5	6	5	4	6

Os comportamentos característicos das seqüências de sincronização de todos os estados da Máquina M6 são apresentados nas Tabelas 11.1, 11.2, 11.3, 11.4, 11.5 e 11.6; assim, aplicando $SS(S_i)$ em qualquer um dos estados da Máquina M6, sempre se chegará ao estado final S_i.

Definição 15 : Uma **Seqüência Distinguível** (DS) de uma MEF M é uma seqüência de identificação que produz seqüências de saída diferentes para cada um dos estados da Máquina M.

Uma Seqüência Distinguível é descrita por um caminho da Árvore de Distinção e que tem como origem o nó inicial e como destino um nó formado por um grupo A simples.

Exemplo 15 : A árvore de distinção da MEF M6 é apresentada pela Figura 11 . De acordo com o algoritmo apresentado no Apêndice A, é possível identificar várias seqüências distinguíveis da Máquina M6, isto é, caminhos do nó inicial até um nó simples. Dessa forma pode-se estabelecer a Tabela 12.

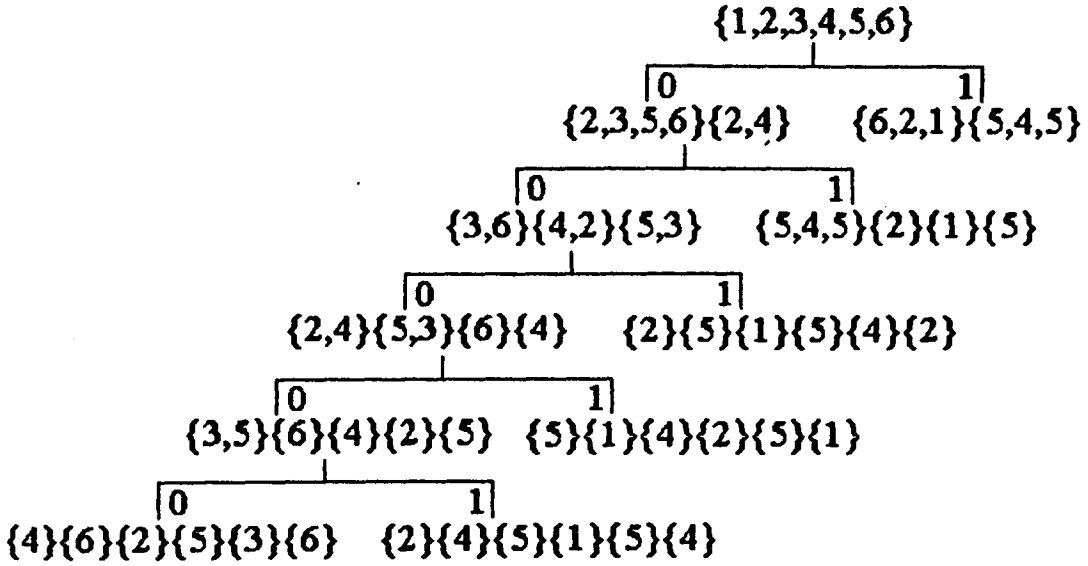


Figura 11 : Árvore de Distinção da Máquina M6.

Tabela 12 : Seqüências Distinguíveis da Máquina M6.

DS1(A1)	001
DS2(A1)	0001
DS3(A1)	00000
DS4(A1)	00001

Tabela 12.1 : Aplicação de DS1 à Máquina M6.

Sv \ DS1	0	0	1
1	0	0	0
2	0	1	0
3	1	0	1
4	0	0	1
5	0	1	1
6	1	0	0

Tabela 12.2 : Aplicação de DS2 à Máquina M6.

Sv \ DS2	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	1	0	0	1
4	0	0	1	1
5	0	1	0	0
6	1	0	0	0

Tabela 12.3 : Aplicação de DS3 à Máquina M6.

Sv \ DS3	0	0	0	0	0
1	0	0	1	0	0
2	0	1	0	0	1
3	1	0	0	1	0
4	0	0	1	0	1
5	0	1	0	1	0
6	1	0	1	0	0

Tabela 12.4 : Aplicação de DS4 à Máquina M6.

Sr \ DS4	0	0	0	0	1
1	0	0	1	0	1
2	0	1	0	0	1
3	1	0	0	1	1
4	0	0	1	0	0
5	0	1	0	1	0
6	1	0	1	0	1

Nas Tabelas 12.1, 12.2, 12.3 e 12.4, pode-se observar o comportamento característico das seqüências distinguíveis através das respectivas seqüências de saída geradas pela Máquina M6. Assim aplicando uma DSi (i=1,2,3,4) sobre todos os estados da Máquina M6, serão produzidas seqüências de saída diferentes para cada um dos estados da Máquina M6.

Definição 16 : Uma **Seqüência Única de Entrada/Saída** (UIO) de uma MEF M é uma seqüência de identificação que relaciona um símbolo de entrada e um símbolo de saída e que é única para cada um dos estados de M. Então, se $UIO(S_i) = (x_{j_1} / z_{j_1})(x_{j_2} / z_{j_2})...(x_{j_c} / z_{j_c})$ entre todos os estados de M, somente S_i apresentará o comportamento descrito por $UIO(S_i)$ [SABN88].

Exemplo 16 : Para a MEF M6, de acordo com o algoritmo de geração de seqüências únicas de entrada/saída (UIO) descrito no Apêndice A, pode-se estabelecer a Tabela 13 .

Tabela 13 : Seqüências Únicas de Entrada/Saída da Máquina M6.

UIO(1)	1\0.0\1
UIO(2)	0\0.1\0
UIO(3)	0\1.1\0
UIO(4)	1\0.1\0
UIO(5)	1\1.1\0
UIO(6)	0\1.1\1

Na Tabela 13 observa-se os comportamentos característicos das seqüências UIO de cada estado da Máquina M6 uma vez que para cada estado S_i, com i=1,2,3,4,5,6, somente S_i apresenta a seqüência descrita em UIO(S_i) identificando de maneira única cada estado da Máquina M6.

Definição 17 : Um **Conjunto de Caracterização** (W) de uma MEF M é um conjunto de seqüências de identificação $\{seq_1, seq_2, \ldots, seq_d\}$ tal que os símbolos de saída observados após a aplicação dessas seqüências (numa ordem fixada) sobre cada um dos estados são diferentes.

O Conjunto de Caracterização pode ser obtido através da geração de Tabelas Pk [NAKAa94], mas não é necessário que a MEF seja fortemente conectada , propriedade esta apresentada na definição 18, a seguir.

Exemplo 17 : Para a MEF M6, com o uso do algoritmo de geração do conjunto de caracterização (W) apresentado na próxima seção, pode-se estabelecer o seguinte conjunto W = {{0},{1},{11},{10}} e construir a Tabela 14.

Tabela 14 : Aplicação do Conjunto de Caracterização na Máquina M6.

S_r \ W	0	1	11	10
1	0	0	01	01
2	0	1	11	10
3	1	0	01	00
4	0	0	00	00
5	0	1	10	10
6	1	1	11	10

Na Tabela 14 pode-se observar que as seqüências de W quando aplicadas a cada estado da Máquina M6, geram seqüências de saída diferentes para cada estado da Máquina M6, confirmando o comportamento do conjunto W em relação às seqüências de saída geradas pela Máquina M6 quando se aplica o conjunto de caracterização em cada estado da Máquina M6.

Definição 18 : Uma MEF M é **Fortemente Conectada** se para todo par de estados (s_i, s_j) da Máquina M, existir uma seqüência de símbolos de entrada tal que essa seqüência leve a Máquina do estado s_i para o estado s_j.

Exemplo 18 : A MEF M6 da Figura 9 com sua Tabela de Transição na Tabela 10 e sua Árvore de Distinção na Figura 11 é uma Máquina Fortemente Conectada, pois existe pelo menos um caminho na Árvore de Distinção que tenha como origem um estado s_i e destino um estado s_j, para todo par de estados (s_i, s_j) de M6.

Apresentam-se a seguir alguns exemplos de MEF onde a identificação de uma seqüência básica não implica na identificação obrigatória de outra seqüência básica em relação à mesma MEF .

4. Comparação entre Seqüências Básicas de MEF

Após a definição das seqüências básicas feita anteriormente, pode-se estabelecer alguns lemas.

Lema 1 : Seja M uma MEF. Se M possui uma **Seqüência Distinguível**, então a Máquina M possui **Seqüências Únicas de Entrada/Saída** para cada estado de M.

Demonstração :
(==>)
Como por hipótese a Máquina M possui uma seqüência distinguível, denomina-se a seqüência de DS(M)=$a_{i_1} a_{i_2} \cdots a_{i_r}$.
Por definição, ao aplicar-se DS(M) sobre cada um dos estados de M, todas as seqüências de saída serão diferentes entre si. Dessa forma, as seqüências que relacionam os símbolos de entrada e saída para cada estado também serão diferentes, pois os símbolos de saída dessas

seqüências são diferentes. Como essas seqüências de entrada e saída estão de acordo com a definição de seqüências únicas de entrada/saída e são apresentadas para cada estado de M então M possui seqüências únicas de entrada/saída para cada estado de M.

Lema 2 : Seja M uma MEF. Se a Máquina M possui uma **Seqüência Distingüível** então M possui um **Conjunto de Caracterização** formado apenas por essa seqüência distinguível .

Demonstração :
(==>)
Por hipótese M possui uma seqüência distinguível que será denominada de DS(M)=$a_{i_1} a_{i_2} \cdots a_{i_r}$.
De acordo com a definição de seqüência distinguível, quando aplica-se DS(M) sobre cada um dos estados da Máquina M, todas as seqüências de saída geradas por DS(M) serão diferentes para cada um dos estados de M; assim essas seqüências de saída se comportam da mesma forma que as seqüências de saída produzidas por M, quando aplica-se o conjunto de caracterização sobre cada um dos estados de M, então M possui um conjunto de caracterização formado pela própria DS(M).

Lema 3 : Se uma MEF M possui uma **Seqüência Distingüível** então M não possui necessariamente uma **Seqüência de Sincronização**.

Demonstração :
Usa-se como contra-exemplo o Exemplo 19.

Exemplo 19 : A MEF M7 da Figura 12, com sua Tabela de Transição dada na Tabela 15, possui Seqüência de Sincronização de acordo com a Árvore de Sincronização da Máquina M7 na Figura 13, uma vez que existe pelo menos um nó formado por um grupo A simples nesta árvore, mas a Máquina M7 não possui Seqüência Distinguível de acordo com a Árvore de Distinção da Máquina M7 na Figura 14, uma vez que não existe nenhum nó formado por um grupo A simples nessa Árvore de Distinção.

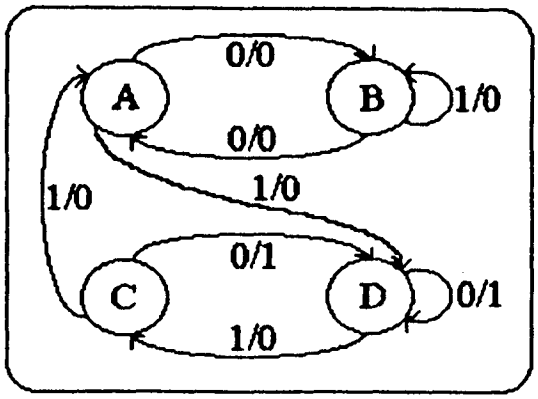


Figura 12 : Máquina M7.

Tabela 15 : Tabela de Transição da Máquina M7 .

x_v \ S_v	0	1	0	1
A	0	0	B	D
B	0	0	A	B
C	1	0	D	A
D	1	0	D	C

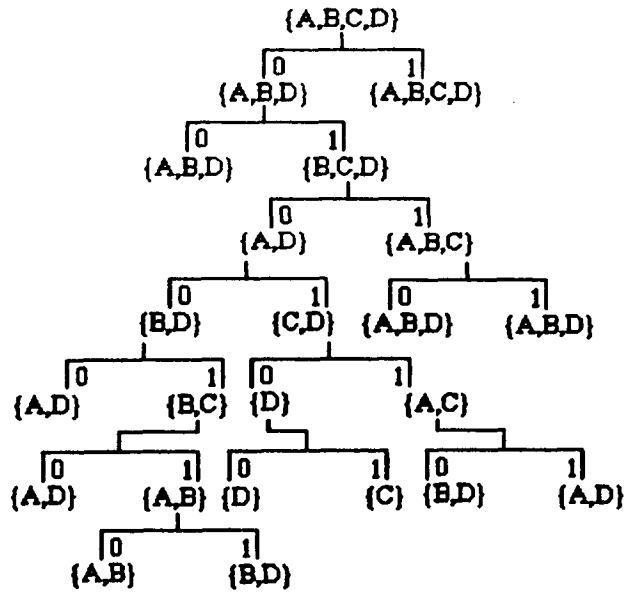


Figura 13 : Árvore de Sincronização da Máquina M7.

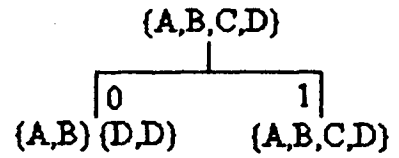


Figura 14 : Árvore de Distinção da Máquina M7.

Lema 4 : Se uma MEF M possui uma **Seqüência de Sincronização** então M não possui necessariamente uma **Seqüência Distingüível**.

Demonstração :
Usa-se como contra-exemplo o Exemplo 20.

Exemplo 20 : A MEF M8 da Figura 15 com sua Tabela de Transição na Tabela 16 não possui Seqüência de Sincronização uma vez que de acordo com a Árvore de Sincronização da Máquina M8 apresentada na Figura 17, não é possível identificar uma Seqüência de Sincronização pois não existe nenhum nó formado por um grupo A simples nessa Árvore de Sincronização, mas a Máquina M8 possui Seqüência Distinguível uma vez que de acordo com a Árvore de Distinção da Máquina M8 da Figura 16, é possível identificar uma Seqüência Distinguível pois existe pelo menos um nó formado por um grupo A simples.

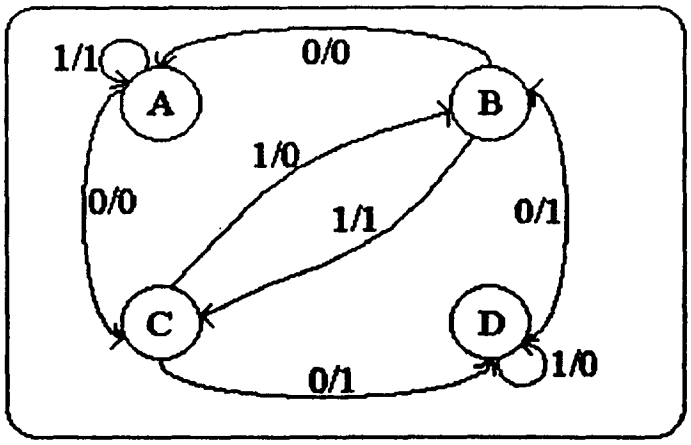


Figura 15 : Máquina M8.

Tabela 16 : Tabela de Transição da Máquina M8.

X_T \ S_T	0	1	0	1
A	0	1	C	A
B	0	1	A	C
C	1	0	D	B
D	1	0	B	D

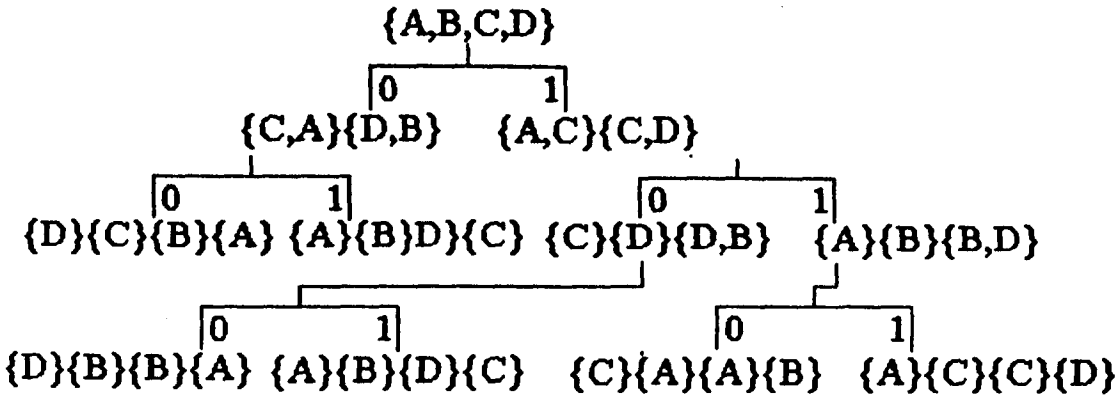


Figura 16 : Árvore de Distinção da Máquina M8.

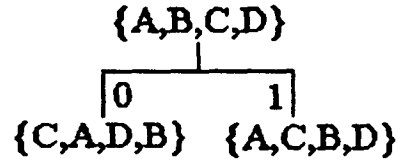


Figura 17 : Árvore de Sincronização da Máquina M8.

5. Aspectos de Implementação

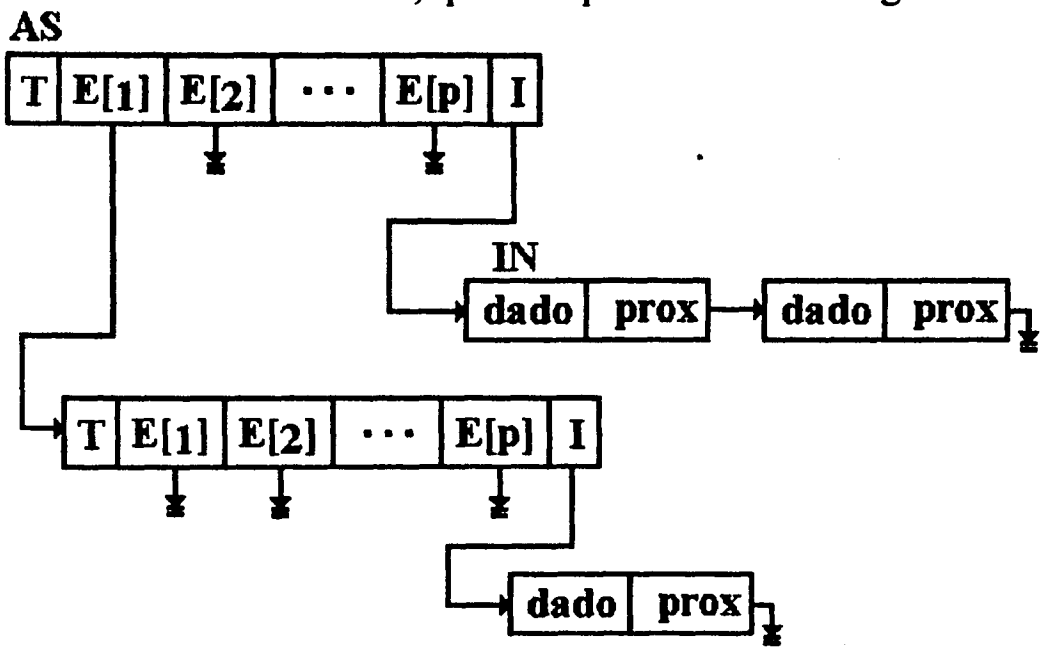
Nesta Seção são apresentadas as principais estruturas de dados utilizadas nos algoritmos do Apêndice A, que geram as seqüências básicas definidas anteriormente, além da análise desses algoritmos.

5.1. Principais Estruturas de Dados utilizadas nos Algoritmos

A seguir é apresentado para cada estrutura de dados a sua representação gráfica, acompanhada da descrição de cada campo da estrutura.

5.1.1. Árvore de Sincronização

A Árvore de Sincronização será representada por uma estrutura dinâmica denominada AS, que é esquematizada da seguinte forma :



É definida a seguir a notação para cada um dos campos da estrutura AS e das estruturas incluídas em AS :

A) Estrutura AS :

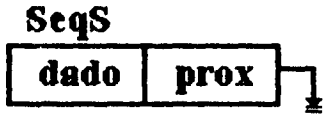
T = rótulo de identificação de um nó terminal.
E[i] = ponteiro para o nó associado ao símbolo de entrada X[i].
I = ponteiro para a incerteza associado ao nó corrente.

B) Estrutura IN :

dado = identificação de estado da Máquina M.
prox = ponteiro para o próximo estado da lista de estados que forma a incerteza do nó corrente.

5.1.2. Seqüência de Sincronização

A Seqüência de Sincronização será representada por uma estrutura dinâmica denominada SeqS, esquematizada da seguinte forma :

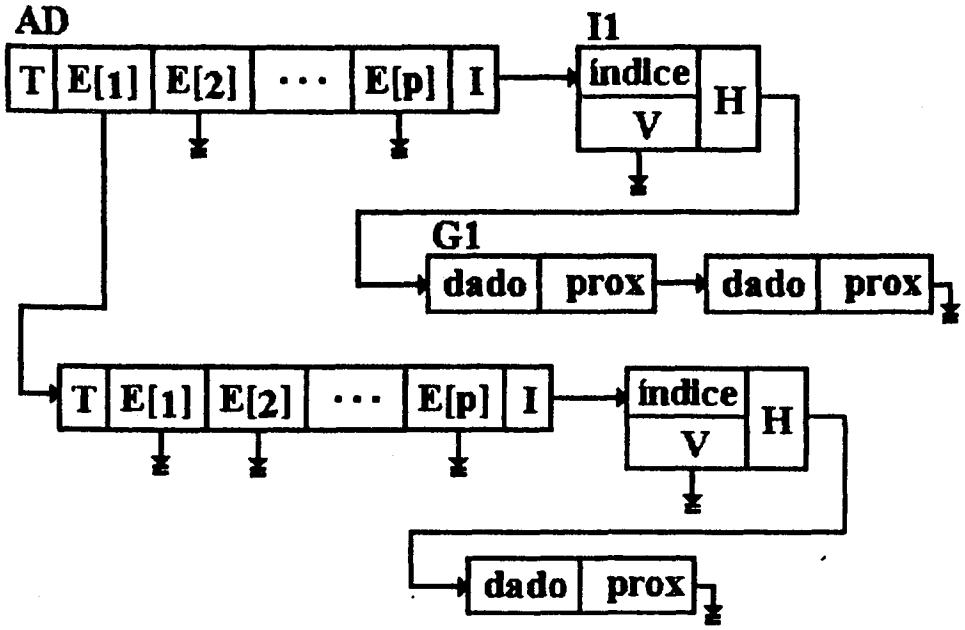


A estrutura SeqS apresenta seus campos com a seguinte notação :

dado = símbolo de entrada da Máquina M.
prox = ponteiro para o próximo símbolo de entrada da Seqüência de Sincronização.

5.1.3. Árvore de Distinção

A Árvore de Distinção será representada por uma estrutura denominada AD que é esquematizada da seguinte forma :



É definida a seguir a notação para cada um dos campos da estrutura AD e das estruturas incluídas em AD :

A) Estrutura AD :

T = rótulo de identificação de um nó terminal.

E[i] = ponteiro para o nó associado ao símbolo de entrada A[i].

I = ponteiro para a lista de incertezas que forma o grupo A associado.

B) Estrutura I1 :

índice = índice do símbolo de saída que caracteriza a incerteza.

H = ponteiro para a lista de estados que forma a incerteza em relação ao símbolo de saída indicado em índice.

V = ponteiro para a próxima incerteza.

C) Estrutura S1 :

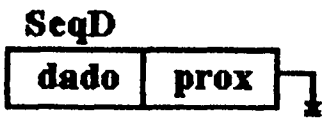
dado = identificação do estado que forma a incerteza corrente.

prox = ponteiro para o próximo estado da incerteza corrente.

Instituto de Ciências Biblioteca

5.1.4. Seqüência Distinguível

A Seqüência Distinguível será representada por uma estrutura dinâmica denominada SeqD, esquematizada da seguinte forma :



A estrutura SeqD apresenta seus campos com a seguinte notação :

dado = símbolo de entrada da Máquina M.
prox = ponteiro para o próximo símbolo de entrada da Seqüência Distinguível.

5.1.5. Estruturas para Geração de Seqüências Únicas de Entrada/Saída

5.1.5.1. Conjunto de Rótulos Distintos

O conjunto de rótulos distintos de cada uma das arestas da MEF M será denotado por $ED = \{L_{a_1}, L_{a_2}, \ldots, L_{a_i}\}$.

5.1.5.2. Conjunto de Arestas

O conjunto de arestas com o rótulo L_{a_i}, para cada $L_{a_i} \in ED$ será denotado por $Set(L_{a_i})$.

5.1.5.3. Vetor de seqüências de comprimento (L-1)

OP1 =	OP1[1]	OP1[2]	. . .	OP1[ond]

onde L é o comprimento das seqüências de entrada/saída sendo examinadas e **ond** é o número de seqüências distintas de comprimento (L-1).

5.1.5.4. Vetor de estados finais para seqüências de comprimento (L-1)

Oend=	Oend1[1]	Oend1[2]	. . .	Oend1[ond]

Cada elemento Oend1[i] representa o estado final da Máquina M em relação à seqüência OP1[i].

5.1.5.5. Vetor de duplas de estados para seqüências de comprimento (L-1)

Oled1 =	Oled1[1]	Oled1[2]	. . .	Oled1[ond]

Este vetor armazena em cada Oled1[i] uma dupla de estados que representa o estado inicial e o estado final da Máquina M em relação à seqüência de OP1[i].

5.1.5.6. Vetor de seqüências de comprimento L

P1 =	P1[1]	P1[2]	. . .	P1[nd]

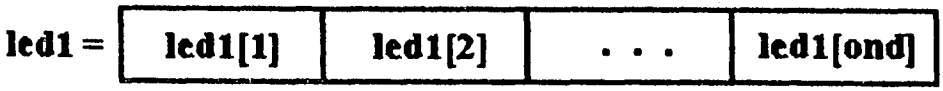
onde **nd** é o número de seqüências de comprimento L.

5.1.5.7. Vetor de estados finais para seqüências de comprimento L

end1 =	end1[1]	end1[2]	. . .	end1[ond]

Cada elemento end1[i] representa o estado final da Máquina M em relação à seqüência P1[i].

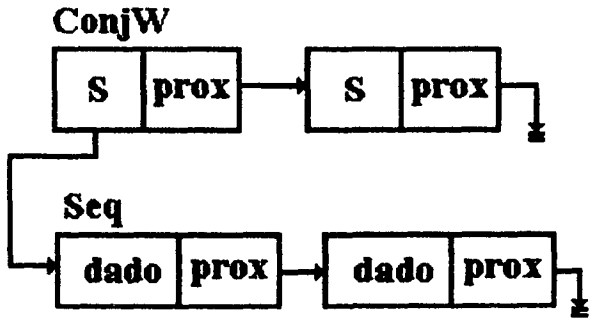
5.1.5.8. Vetor de duplas de estados para seqüências de comprimento L



Este vetor armazena em cada elemento led1[i] uma dupla de estados que contêm o estado inicial e o estado final da Máquina M em relação à seqüência representada por P1[i].

5.1.6. Conjunto de Caracterização

O Conjunto de Caracterização é representado por uma estrutura dinâmica denominada ConjW, esquematizada da seguinte forma:



É definida a seguir a notação para a estrutura ConjW e as estruturas incluídas em ConjW:

A) Estrutura ConjW :

S = ponteiro para uma lista que representa uma seqüência de símbolos de entrada da Máquina M.
prox = ponteiro para a próxima seqüência de símbolos de entrada.

B) Estrutura Seq :

dado = símbolo de entrada da Máquina M.
prox = ponteiro para o próximo símbolo de entrada da Máquina M.

5.2. Análise dos Algoritmos

Em relação aos algoritmos do Apêndice, é apresentada a seguir uma análise dos mesmos, baseada em diversos autores.

5.2.1. Geração de Seqüências de Sincronização

De acordo com KOHAVI [KOHA78], se uma máquina apresenta uma Seqüência de Sincronização, isto é, é possível que uma determinada Árvore de Sincronização seja gerada, então tal seqüência tem comprimento h1 tal que

$$h1 = \frac{n.(n+1).(n-1)}{6} \qquad \text{(a)}$$

onde n= número de estados de M.

O número máximo de nós de uma Árvore Sucessora e denominado de nosuc, é calculado da seguinte forma :

$$nosuc = \frac{p^{(h+1)} - 1}{(p-1)}$$

onde

p= número de símbolos de entrada de M
h= altura da Árvore Sucessora .

Como a Árvore de Sincronização é uma Árvore Sucessora, pode-se calcular o número máximo de nós da Árvore de Sincronização denominada nosin, da seguinte forma :

$$nosin = \frac{p^{(h1+1)} - 1}{(p-1)} \qquad \text{(b)}$$

De acordo com (a) e (b), a complexidade do algoritmo de geração da Árvore de Sincronização quando ela existe, é de $O(p^4)$. Como o algoritmo de geração de Seqüências de Sincronização percorre no máximo uma vez toda a Árvore de Sincronização, a complexidade desse algoritmo também e de $O(p^4)$.

5.2.2. Geração de Seqüências Distinguíveis

Segundo KOHAVI [KOHA78], se uma máquina apresenta uma Seqüência Distinguível, isto é, é possível que uma determinada Árvore de Distinção possa ser gerada, então tal seqüência tem comprimento h2, tal que :

$$h2 = (n-1).n^n \qquad \text{(c)}$$

onde n=número de estados de M.

Sendo a Árvore de Distinção uma Árvore Sucessora, o número máximo de nós da Árvore de Distinção, denominado nodis, pode ser calculado da seguinte forma :

$$nodis = \frac{p^{(h2+1)} - 1}{(p-1)} \qquad \text{(d)}$$

De acordo com (c) e (d), a complexidade do algoritmo de geração da Árvore de Distinção é de $O(p^{(n+2)})$, quando ela existe. Pelo fato do algoritmo de geração de Seqüências Distinguíveis percorrer no máximo uma vez toda a Árvore de Distinção, a complexidade desse algoritmo também de $O(p^{(n+2)})$.

5.2.3. Geração de Seqüências Únicas de Entrada/Saída

De acordo com SABNANI [SABN88], a complexidade do algoritmo de Seqüências Únicas de entrada/saída apresentado em Apêndice é de $O(n^2 \cdot dmax^{(2.n^2+2)})$, onde dmax é o maior dos números de arestas saindo de cada estado e n é o número de estados de M. O comprimento máximo das Seqüências Únicas geradas por este algoritmo é de 2^{n^2}.

5.2.4. Geração do Conjunto de Caracterização

Segundo CHOW [CHOW78], a complexidade do algoritmo de geração do Conjunto de Caracterização é de $O(n^2 . p)$, onde p é o número de símbolos de entrada de M e m é o número de estados de M, pois de acordo com GILL [GILL62], para uma máquina minimal, existem no máximo (n-1) Tabelas Pk [NAKA94a], e o esforço para construir uma Tabela Pk é proporcional a n.p, isto é, o número de elementos da Tabela Pk, e como a construção do Conjunto de Caracterização é baseada em Tabelas Pk, a complexidade do algoritmo de geração do Conjunto de Caracterização é de $O(n^2 . p)$.

6. Conclusão

Neste trabalho foram apresentados os principais conceitos e terminologia pertinentes à geração de seqüências básicas de MEF. Tratam-se de seqüências básicas as seqüências de identificação e a seqüência de sincronização. As seqüências de identificação são aquelas que têm como objetivo identificar de maneira única cada um dos estados da MEF e a seqüência de sincronização visa identificar o estado final da MEF independentemente da saída gerada ou do estado inicial. As seqüências básicas são utilizadas nos seguintes critérios de geração de seqüências de teste : DS [GONE70], UIO [SABN88], W [CHOW78] e Wp [FUJI91], que exigem que a MEF satisfaça a certas propriedades como por exemplo minimalidade e especificação completa. Nakazato et all descrevem as propriedades e os respectivos algoritmos de identificação implementados [NAKA94a]. Os critérios de geração de seqüências de teste com a descrição de seus procedimentos, além da implementação de seus algoritmos, são apresentados em [NAKA94b].

Descrevem-se ainda neste trabalho os algoritmos para a implementação dessas seqüências básicas e sintetizam-se análises comparativas entre essas seqüências, procurando fornecer um embasamento teórico maior para a escolha dos critérios de geração de seqüências de teste, alvo desta linha de pesquisa.

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 .

[CHOW78] CHOW, T. S. "Testing Software Design modeled by Finite State Machines" in IEEE Trans. on Software Engineering, vol.SE-4, March 1978, pp 178-187.

[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 .

[GONE70] GÖNENÇ, G. "A Method for Design of Fault-Detection Experiments" in IEEE Trans. on Computers, vol.C-19,No.6,1970,pp 551-558.

[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. "Propriedades de Máquinas de Estado Finito Relevantes para Critérios de Geração de Seqüências de Teste.", 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).

[SABN88] SABNANI, K.K., and DAHBURA, A., "A Protocol Test Generation Procedure", Comput. Networks and ISDN Systems, Vol. 15, No.4, 1988, pp.285-297.

Apêndice A

A1. Geração da Árvore de Sincronização

Este algoritmo gera a Árvore de Sincronização através do uso de recursão e de algumas estruturas apresentadas anteriormente neste trabalho, além de outras estruturas definidas em [Naka94a]. Dentro deste algoritmo deve-se destacar os seguintes aspectos:

1) Parâmetros 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 :

Árvore de Sincronização representada por S.

3) Descrição de Variáveis Locais :

S1 : nó gerado por S.
RAIZ : raiz da Árvore de Sincronização.
NOS : variável auxiliar para os nós gerados por S.
IN1 : incerteza do nó corrente a ser gerado.
AUX,
AUX1,
AUX2,
CAB : variáveis auxiliares para as listas de incertezas de S e S1.
I : contador de laço.
TERMINA_NIVEL : variável auxiliar.
VAZIA : variável dinâmica não inicializada.

4) Procedimentos Utilizados :

a) IncluaAS b)BuscaAS

5) Procedimentos que o utilizam :

5) Procedimentos que o utilizam :

Nenhum.

6) Descrição do Algoritmo em Pseudocódigo :

```
Procedimento ConstroiAS(FZ1,FS1,S)
início
 se S=VAZIA
    então
      /* Geração do nó raiz da Árvore de Sincronização  */
      Aloque espaço para S
      RAIZ:=S
      para I := 1 até P faça
        S.E[i] := VAZIA
      fim para
      Aloque espaço para IN1
      S.I1:=IN1
      IN1.dado:=S[1]     IN1.prox:=VAZIA
      para I :=2 até N faça
        AUX:=IN1
        Aloque espaço para IN1
        IN1.dado:=S[I]   IN1.prox:=VAZIA
        AUX.prox:=IN1
      fim para
 fim se
 para I:=1 até P faça
    /*    Geração dos p:=filhos do nó S     */
    Aloque espaço para S1
    S.E[i]:=S1
    Aloque espaço para IN1
    S1.I:=IN1
    AUX1:=S.I
    CAB:=S1.I    CAB.dado:=-1
    repita
      IncluaAS(CAB,FS[AUX1.dado,I])
      AUX1:=AUX1.prox
    até(AUX1:=VAZIA)
    fim repita
    NOS:=S.E[I]
    NOS.T:=falso
    se BuscaAS(RAIZ,NOS)
       então NOS.T:=verdadeiro
    fim se
 fim para
 /* Geração do próximo nível da Ávore de Sincronização  */
 para I:=1 até P faça
    NOS:=S.E[i]
    se NOS.T:=falso
       então  ConstroiAS(NOS)
    fim se
 fim para
fim ConstroiAS
```

A2. Inclusão de um Estado numa Incerteza da Árvore de Sincronização

Este procedimento inclui um estado numa incerteza que forma um grupo A associado a um nó filho da Árvore de Sincronização da Máquina M.

1) Parâmetros de Entrada :

Grupo A de um nó filho da Árvore de Sincronização representado por CABEÇA.
Próximo estado da Máquina M representado por PROX_ESTADO.

2) Parâmetro de Saída :

Grupo A de um nó filho da Árvore de Sincronização representado por CABECA.

3) Descrição de Variáveis Locais :

I2 : incerteza que forma um grupo A associado a um nó da Árvore de Sincronização.
AUX1,
AUX2 : variáveis auxiliares para as listas de estados das incertezas de S e S1.
I : contador de laço.
PARADA : variável auxiliar.
VAZIA : variável dinâmica não inicializada.

4) Procedimentos Utilizados :

Nenhum.

5) Procedimentos que o utilizam :

Nenhum.

6) Descrição do Algoritmo em Pseudocódigo :

```
Procedimento IncluaAS(CABECA,PROX_ESTADO)
início
  se CABECA.dado ≠ -1
    então
      AUX1:=CABECA
      PARADA:=falso
      enquanto(AUX1 ≠ VAZIA ou não PARADA) faça
        se AUX1.dado := PROX_ESTADO
           então PARADA:=verdadeiro
        fim se
        AUX2:=AUX1
        AUX1:=AUX1.prox
      fim enquanto
      se não PARADA e AUX1:=VAZIA
        então
          Aloque espaço para IN1
          IN1.dado:=PROX_ESTADO
          IN1.prox:=VAZIA
          AUX2.prox:=IN1
      fim se
    senão
      CABECA.dado:=PROX_ESTADO
      CABECA.prox:=VAZIA
  fim se
fim IncluaAS
```

A3. Busca de um Nó na Árvore de Sincronização

Este procedimento faz uma busca na Árvore de Sincronização com o objetivo de identificar se o nó filho tem um nó semelhantes que foi incorporado anteriormente a esta árvore.

1) Parâmetros de Entrada :

Raiz da Árvore de Sincronização representado por RAIZ.
Nó filho da Árvore de Sincronização representado por NOS.

2) Parâmetro de Saída :

Nenhum.

3) Descrição de Variáveis Locais :

AUX : variável auxiliar para percorrer os nós da Árvore de Sincronização.
AUX1,
AJUDA1 : variáveis auxiliares para as listas de incertezas.
AUX2,
AJUDA2 : variáveis auxiliares para a lista de estados de uma incerteza.
I : contador de laço.
PARADA : variável auxiliar.
VAZIA : variável dinâmica não inicializada.

4) Procedimentos Utilizados :

Nenhum.

5) Procedimentos que o utilizam :

Nenhum.

6) Descrição do Algoritmo em Pseudocódigo :

```
Procedimento BuscaAS(RAIZ,NOS)
início
  PARADA:=falso
  se RAIZ ≠ NOS
     AUX:=RAIZ
     AUX1:=AUX.I
     AJUDA1:=NOS.I
     repita
       se AUX1.dado ≠ AJUDA1.dado
          então PARADA:=verdadeiro
       fim se
       AUX1:=AUX1.prox
       AJUDA1:=AJUDA1.prox
     até PARADA:=verdadeiro ou AUX1:=VAZIA ou AJUDA1:=vazia
     fim repita
  fim se
  I:=1
  se não PARADA
     então
        enquanto não PARADA ou NOS.E[I] ≠ VAZIA
           se BuscaAS(NOS.E[i],NOS)
              então PARADA:=verdadeiro
           fim se
           I:=I+1
         fim enquanto
  fim se
  BuscaAS:=PARADA
fim BuscaAS
```

A4. Geração da Seqüência de Sincronização

Este algoritmo gera uma Seqüência de Sincronização através da busca de um caminho na Árvore de Sincronização que tenha como início o nó raiz e como fim um nó que apresente uma incerteza simples.

1) Parâmetros de Entrada :

Nó da Árvore de Sincronização representado por NOS.
Seqüência de Sincronização representada por SS1.

2) Parâmetro de Saída :

Seqüência de Sincronização representada por SS1.
Variável de parada para a recursão representada por CONTINUA.

3) Descrição de Variáveis Locais :

AUX1 : variável auxiliar para percorrer a lista de estados da incerteza.
I : contador de laço.
VAZIA : variável dinâmica não inicializada.

4) Procedimentos Utilizados :

Nenhum.

5) Procedimentos que o utilizam :

Nenhum.

6) Descrição do Algoritmo em Pseudocódigo :

```
Procedimento GeraçãoSS(NOS,SS1,CONTINUA)
início
  CONTINUA:=falso
  I:=1
  enquanto não CONTINUA ou I≤P faça
    AUX1:=NOS.E[I].IN
    se AUX1.prox=VAZIA
      então CONTINUA:=verdadeiro
    fim se
    se CONTINUA
      então
        AUX1:=SS1
        Aloque espaço para SS1
        SS1.dado:=A[I]
        SS1.prox:=VAZIA
        AUX1.prox:=SS1
    fim se
    I:=I+1
  fim enquanto
  se não CONTINUA
    então
      I:=1
      enquanto não CONTINUA ou I≤P faça
        GeraçãoSS(NOS.E[I],SS1,CONTINUA)
        I:=I+1
      fim enquanto
  fim se
fim GeraçãoSS
```

A5. Geração da Árvore de Distinção

Este algoritmo gera a Árvore de Distinção da Máquina M através do uso de recursão e de algumas estruturas apresentadas anteriormente neste trabalho, além de outras estruturas definidas em [Naka94a]. Dentro deste algoritmo deve-se destacar os seguintes aspectos:

1) Parâmetros 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 :

Árvore de Distincão representada por D.

3) Descrição de Variáveis Locais :

D1 : nó gerado por D.
RAIZ : raiz da Árvore de Distinção.
NO : variável auxiliar para os nós gerados por D.
I2 : incerteza do nó corrente a ser gerado.
G2 : lista de estados de uma incerteza I2.
AUX : variável auxiliar para lista de estados G2.
AUX1,
AUX2,
CAB : variáveis auxiliares para as listas de incertezas de D e D1.
I : contador de laço.
TERMINA_NÍVEL : variável auxiliar.
VAZIA : variável dinâmica não inicializada.

4) Procedimentos Utilizados :

a) IncluaAD b) Múltipla
c) Simples d) BuscaAD

5) Procedimentos que o utilizam :

5) Procedimentos que o utilizam :

Nenhum.

6) Descrição do Algoritmo em Pseudocódigo :

```
Procedimento ConstroiAD(FZ1,FS1,D)
início
  se D:=VAZIA
    então
      /* Geração do nó raiz da Árvore de Distinção  */
      Aloque espaço para D
      RAIZ:=D
      para I:=1 até P faça
        D.E[I]:=VAZIA
      fim para
      Aloque espaço para I2
      D.I1:=I2
      Aloque espaço para G2
      I2.H:=G2
      I2.V:=VAZIA
      I2.índice:=0
      G2.dado:=S[1]
      G2.prox:=VAZIA
      para I:=2 até N faça
        AUX:=G2
        Aloque espaço para G2
        G2.dado:=S[I]
        G2.prox:=VAZIA
        AUX.prox:=G2
      fim para
  fim se
  para I:=1 até P faça
    Aloque espaço para D1
    D.E[I]:=D1
    Aloque espaço para I2
    D1.I:=I2
    I2.H:=VAZIA
    I2.V:=VAZIA
    I2.índice:=0
    AUX1:=D.I
    CAB:=D1.I
    repita
      AUX2:=AUX1.H
      repita
        Inclua(CAB,FZ1[AUX2.dado,A[I]],FS1[AUX2.dado,A[I])
        AUX2:=AUX2.prox
      até AUX2:=VAZIA
      AUX1:=AUX1.prox
    até AUX1:=VAZIA
    TERMINA_NIVEL:=falso
    NO:=D.E[I]
    NO.T:=falso
```

```
    se Simples(NO)
      então TERMINA_NIVEL:=verdadeiro
    fim se
    se Busca(RAIZ,NO)
      então NO.T:=verdadeiro
    fim se
    se TERMINA_NIVEL:=verdadeiro
       então
         para J:=1 até P faça
           NO:=D.E[J]
           NO.T:=verdadeiro
         fim para
    fim se
  fim para
  /*  Geração do próximo nível da Árvore de Distinção  */
  para I:=1 até P faça
    NO:=D.E[I]
    se NO.T := falso
      então ConstroiAD(NO)
    fim se
  fim para
fim ConstroiAD
```

A6. Inclusão de um Estado numa Incerteza da Árvore de Distinção

Este procedimento inclui um estado numa incerteza que forma um grupo A associado a um nó filho da Árvore de Distincão da Máquina M.

1) Parâmetros de Entrada :

Grupo A de um nó filho da Árvore de Distincão representado por CABEÇA.
Símbolo de saída da Máquina M representado por SAIDA.
Próximo estado da Máquina M representado por PROX_ESTADO.

2) Parâmetro de Saída :

Grupo A de um nó filho da Árvore de Distincão representado por CABEÇA.

3) Descrição de Variáveis Locais :

I2 : incerteza que forma um grupo A associado a um nó da Árvore de Distinção.
G2 : estado que faz parte de uma incerteza.
AUX,
AUXA,
AUX1,
AUX1A: variáveis auxiliares para as listas de incertezas.
I : contador de laço.
FIM : variável auxiliar.
VAZIA : variável dinâmica não inicializada.

4) Procedimentos Utilizados :

Nenhum.

5) Procedimentos que o utilizam :

Nenhum.

6) Descrição do Algoritmo em Pseudocódigo :

```
Procedimento IncluaAD(CABECA,SAIDA,PROX_ESTADO)
início
  AUX:=CABEÇA
  FIM:=falso
  repita
    se AUX.indice:=SAIDA
      /* Inclusão de um estado numa incerteza já existente */
      então
        FIM:=verdadeiro
        AUX1:=AUX.H
        /* Busca do fim da incerteza */
        enquanto AUX1 ≠ VAZIA
          AUX1A:=AUX1
          AUX1:=AUX1.prox
        fim enquanto
        Aloque espaço para G2
        AUX1A.prox:=G2
        G2.prox:=VAZIA
        G2.dado:=PROX_ESTADO
    fim se
    AUX:=AUX.V
  até AUX=VAZIA
  fim repita
  se FIM =falso
    então
      /* Inclusão de uma nova incerteza */
      AUX:=CABEÇA
      enquanto AUX ≠ VAZIA faça
        AUXA:=AUX
        AUX:=AUX.V
      fim enquanto
      Aloque espaço para I2
      AUXA.V:=I2
      I2.indice:=SAIDA
      I2.V:=VAZIA
      Aloque espaço para G2
      G2.dado:=PROX_ESTADO
      G2.prox:=VAZIA
      I2.H:=G2
  fim se
fim IncluaAD
```

A7. Busca de um Nó na Árvore de Distincão

Este procedimento faz uma busca na Árvore de Distincão com o objetivo de identificar se o nó filho tem um nó semelhante que foi incluído anteriormente a esta Árvore.

1) Parâmetros de Entrada :

Raiz da Árvore de Distinção representado por RAIZ.
Nó filho da Árvore de Distinção representado por NO.

2) Parâmetro de Saída :

Nenhum.

3) Descrição de Variáveis Locais :

AUX : variável auxiliar par percorrer os nós da Árvore de Distinção.
AUX1,
AJUDA1 : variáveis auxiliares para as listas de incertezas.
AUX2,
AJUDA2 : variáveis auxiliares para a lista de estados de uma incerteza.
I : contador de laço.
PARADA : variável auxiliar.
VAZIA : variável dinâmica não inicializada.

4) Procedimentos Utilizados :

Nenhum.

5) Procedimentos que o utilizam :

Nenhum.

6) Descrição do Algoritmo em Pseudocódigo :

```
Procedimento BuscaAD(RAIZ,NO)
início
  PARADA:=falso
  se RAIZ ≠ NO
    então
      AUX:=RAIZ
      AUX1:=AUX.I
      AJUDA1:=NO.I
      repita
        AUX2:=AUX1.H
        AJUDA2:=AJUDA1.H
        se AUX1.indice=AJUDA1.indice
          então
            repita
              se AUX2.dado ≠ AJUDA2.dado
                então PARADA:=verdadeiro
              fim se
              AUX2:=AUX2.prox
              AJUDA2:=AJUDA2.prox
            até PARADA=verdadeiro ou AUX2=VAZIA ou
                AJUDA2=VAZIA
            fim repita
        fim se
        AUX1:=AUX1.V
        AJUDA1:=AJUDA1.V
      até PARADA=verdadeiro ou AUX1=VAZIA ou
          AJUDA1=VAZIA
      fim repita
      I:=1
      se não PARADA
        então
          enquanto não PARADA ou NO.E[I] ≠ VAZIA
            se Busca(NO.E[I],NO)
              então PARADA:=verdadeiro
            fim se
            I:=I+1
          fim enquanto
      fim se
  fim se
  BuscaAD:=PARADA
fim BuscaAD
```

A8. Identificação de um Nó Simples na Árvore de Distinção

Este procedimento verifica se um nó da Árvore de Distincão é simples, isto é, se este nó tem um grupo A simples associado a ele.

1) Parâmetros de Entrada :

Um nó da Árvore de Distinção representado por NO.

2) Parâmetro de Saída :

Nenhum.

3) Descrição de Variáveis Locais :

AUX1 : variável auxiliar para as listas de incertezas.
AUX2 : variável auxiliar para a lista de estados.
CONT1,
CONT2 : contadores auxiliares.
SUCESSO : variável auxiliar.
VAZIA : variável dinâmica não inicializada.

4) Procedimentos Utilizados :

Nenhum.

5) Procedimentos que o utilizam :

Nenhum.

6) Descrição do Algoritmo em Pseudocódigo :

```
Procedimento Simples(NO)
início
  SUCESSO:=falso
  AUX1:=NO.I
  CONT1:=0
  CONT2:=0
  enquanto AUX1 ≠ VAZIA faça
    AUX2:=AUX1.H
    se AUX2.prox=VAZIA
      então CONT1:=CONT1+1
    fim se
    CONT2:=CONT2+1
    AUX1:=AUX1.V
  fim enquanto
  se CONT1=CONT2
    então SUCESSO:=verdadeiro
  fim se
  Simples:=SUCESSO
fim Simples
```

A9. Identificação de um Nó Múltiplo na Árvore de Distinção

Este procedimento verifica se um nó da Árvore de Distincão é múltiplo, isto é, se o grupo A associado a este nó apresenta uma incerteza múltipla.

1) Parâmetros de Entrada :

Um nó da Árvore de Distinção representado por NO.

2) Parâmetro de Saída :

Nenhum.

3) Descrição de Variáveis Locais :

AUX1 : variável auxiliar para as listas de incertezas.
AUX2,
AUX2A : variáveis auxiliares para a lista de estados.
SUCESSO : variável auxiliar.
VAZIA : variável dinâmica não inicializada.

4) Procedimentos Utilizados :

Nenhum.

5) Procedimentos que o utilizam :

Nenhum.

6) Descrição do Algoritmo em Pseudocódigo :

```
Procedimento Multipla(NO)
início
  AUX1:=NO.I
  repita
    AUX2:=AUX1.H
    SUCESSO:=falso
    enquanto não SUCESSO ou AUX1 ≠ VAZIA faça
      AUX2A:=AUX2
      AUX2:=AUX2.prox
      se AUX2=AUX2A
        então
          SUCESSO:=verdadeiro
      fim se
    fim enquanto
    AUX1:=AUX1.V
  até AUX1=VAZIA ou SUCESSO
  fim repita
  Multipla:=SUCESSO
fim Multipla
```

A10. Geração da Seqüência Distinguível

Este procedimento gera uma Seqüência Distinguível através da busca de um caminho na Árvore de Distinção que tenha como origem o nó raiz e como destino um nó que apresente um grupo A simples.

1) Parâmetros de Entrada :

Nó da Árvore de Distinção representado por NO.
Seqüência Distinguível representada por SD.

2) Parâmetro de Saída :

Seqüência Distinguível representada por SD.
Variável de parada para a recursão representada por CONTINUA.

3) Descrição de Variáveis Locais :

I : contador de laço.
VAZIA : variável dinâmica não inicializada.

4) Procedimentos Utilizados :

Nenhum.

5) Procedimentos que o utilizam :

Nenhum.

6) Descrição do Algoritmo em Pseudocódigo :

```
Procedimento GeraçãoSD(NO,SD,CONTINUA)
início
  CONTINUA:=falso
  I:=1
  enquanto não CONTINUA ou I≤P faça
     CONTINUA:=Simples(NO.E[I])
     se CONTINUA
        então
           AUX1:=SD
           Aloque espaço para SD
           DS.dado:=A[I]
           DS.prox:=VAZIA
           AUX1.prox:=DS
     fim se
     I:=I+1
  fim enquanto
  se não CONTINUA
     então
        I:=1
        enquanto não continua ou I≤P faça
           GeraçãoSD(NO.E[I],SD,CONTINUA)
           I:=I+1
        fim enquanto
  fim se
fim GeraçãoSD
```

A11. Geração das Sequências Únicas de Entrada/Saída

Este algoritmo gera, com um limite de comprimento de 2^{n^2}, as Seqüências Únicas de entrada/saída para todo estado da Máquina M.

1) Parâmetros de Entrada :

Tabela de Transição da Máquina M com as Subtabelas Z_v e S_{v+1}, representadas por FZ1 e FS1.
Conjunto ED de rótulos da Máquina M.

2) Parâmetro de Saída :

Sequência Única de Entrada/Saída do estado s da Máquina M, representado pelo conjunto UIO[s].

3) Descrição de Variáveis Locais :

s : estado corrente.
os : conjunto de arestas de saída do estado s.
Label(oi) : rótulo da aresta oi.
Tail(oi) : estado final da aresta oi.
Opattern : representa OP1.
Oend : representa Oend1.
Oled : representa Oled1.
Pattern : representa P1.
End : representa end1.
Led : representa led1.
k : número de arestas de saída do estado s.

4) Procedimentos Utilizados :

Nenhum.

5) Procedimentos que o utilizam :

Nenhum.

6) Descrição do Algoritmo em Pseudocódigo :

```
GeraçãoUIO(UIO[s])
início
  UIO[S]:=0
  L:=1
  ond:=k;
  para cada oi ∈OS faça
    Opattern[i]:=Label(oi)
    Oend[i]:=Tail(oi)
    Oled[i]:=Set(Label(oi)) - {oi}
    se Oled[i]:=vazio
       então
          UIO[s]:=Label(oi)
    fim se
  fim para
  enquanto (UIO[s]:=0 e l<2^n^2 ) faça
    l:=l+1
    nd:=0
    para i:=1 até ond faça
       Calcule o conjunto Oout de arestas saindo do estado Oend[i]
       para cada aresta O_out ∈ Oout faça
          nd:=nd+1
          Pattern[nd]:=Opattern[i]@Label( O_out )
          End[nd]:=Tail(O_out)
          Led[nd]:=vazio
          para cada dupla(h,t)∈Oled[i] faça
             para cada aresta o de saída do estado t faça
                se Label(o):=Label( O_out )
                   então
                      Led[nd]:=Led[nd]∪(h,Tail(o))
                fim se
                se Led[nd]:=vazia
                   então
                      UIO[s]:=Pattern[nd]
                fim se
             fim para
          fim para
       fim para
       ond:=nd
       para i:=1 até nd faça
          Opattern[i]:=Pattern[i]
          Oend[i]:=End[i]
          Oled[i]:=Led[i]
       fim para
  fim enquanto
fim GeraçãoUIO
```

A12. Geração do Conjunto de Caracterização

Este algoritmo faz uso dos algoritmos de Geração das Tabelas Pk e da Sequência Minimal de Diagnose apresentados em [Naka94a].

1) Parâmetro de Entrada :

Conjunto de estados da Máquina M, representado por S1.

2) Parâmetro de Saída :

Conjunto de Caracterização, representado por CW.

3) Descrição de Variáveis Locais :

TAB : lista de Tabelas Pk geradas.
SMD : sequência minimal de diagnose de S_i e S_j de M do tipo SeqS.
N : número de estados da Máquina M.
I,J : contadores de laço.

4) Procedimentos Utilizados :

a)GeraçãoPk. b)GSMD. c)IncluaW.

5) Procedimentos que o utilizam :

Nenhum.

6) Descrição do Algoritmo em Pseudocódigo :

```
GeraçãoW(S1,CW)
início
  GeraçãoPk(TAB)
  para I:=1 até N-1 faça
    para J:=I+1 até N faça
      GSMD(S1[I],S1[J])
      IncluaW(CW,SMD)
    fim para
  fim para
fim GeraçãoW
```

A13. Inclusão de uma Sequência Minimal de Diagnose no Conjunto de Caracterização

Este procedimento verifica se a Sequência Minimal de Diagnose SMD não apresenta nenhuma sequência semelhante no Conjunto de Caracterização e caso isso ocorra, o procedimento inclui SMD no Conjunto de Caracterização.

1) Parâmetro de Entrada :

Conjunto de Caracterização representado por CABECAW.
Sequência Minimal de Diagnose representada por SMD.

2) Parâmetro de Saída :

Conjunto de Caracterização, representado por CABECAW.

3) Descrição de Variáveis Locais :

AUX,
AUXA : variáveis auxiliares para acesso às sequências do conjunto de caracterização.
ACHOU : variável auxiliar.
VAZIA : variavel dinâmica não inicializada.

4) Procedimentos Utilizados :

Semelhantes.

5) Procedimentos que o utilizam :

GeraçãoW.

6) Descrição do Algoritmo em Pseudocódigo :

```
Procedimento IncluaW(CABECAW,SMD)
início
  AUX:=CABECAW
  ACHOU:=FALSO
  enquanto AUX ≠ VAZIA faça
    AUXA:=AUX
    se Semelhantes(AUX.S,SMD)
      então
        ACHOU:=verdadeiro
    fim se
    AUX:=AUX.prox
  fim enquanto
  se não ACHOU
    então
      Aloque espaço para AUX
      AUX.S:=SMD
      AUX.prox:=VAZIA
      AUXA.prox:=AUX
  fim se
fim IncluaW
```

A14. Semelhança entre Sequências do Conjunto de Caracterização e a Sequência Minimal de Diagnose

Este procedimento verifica se existe alguma sequência dentro do Conjunto de Caracterização que seja semelhante a SMD.

1) Parâmetro de Entrada :

Uma Sequência do Conjunto de Caracterização, representada por SEQW.
Uma Sequência Minimal de Diagnose, representada por SMD.

2) Parâmetro de Saída :

Nenhum.

3) Descrição de Variáveis Locais :

AUX1,
AUX2 : variáveis auxiliares de acesso aos símbolos de entrada de smd e da sequência do conjunto de caracterização.
SUCESSO : variável auxiliar.

4) Procedimentos Utilizados :

Nenhum.

5) Procedimentos que o utilizam :

IncluaW.

6) Descrição do Algoritmo em Pseudocódigo :

```
Procedimento Semelhantes(SEQW,SMD)
início
  AUX1:=SEQW
  AUX2:=SMD
  SUCESSO:=verdadeiro
  enquanto AUX1 ≠ VAZIA ou  AUX2 ≠ VAZIA ou SUCESSO faça
    se AUX1.dado ≠ AUX2.dado
      então
        SUCESSO:=falso
    fim se
    AUX1:=AUX1.prox
    AUX2:=AUX2.prox
  fim enquanto
  Semelhantes:=SUCESSO
fim Semelhantes
```