

Síntese de Supervisores de Sistemas a Eventos Discretos
Temporizados e Não Temporizados

Eduard Montgomery Meira Costa

Tese de Doutorado submetida à Coordenação do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Paraíba - Campus II, como parte dos requisitos para obtenção do grau de Doutor em Ciências no domínio da Engenharia Elétrica.

Área de Concentração: Processamento da Informação

Antonio Marcus Nogueira Lima, Dr.
Orientador

Campina Grande, Paraíba, Brasil

©Eduard Montgomery Meira Costa, 23 de Novembro de 2001



C837s

Costa, Eduard Montgomery Meira.

Síntese de supervisores de sistemas eventos discretos temporizados e não temporizados / Eduard Montgomery Meira Costa. - Campina Grande, 2001.

160 f.

Tese (Doutorado em Engenharia Elétrica) - Universidade Federal da Paraíba, Centro de Ciências e Tecnologia, 2001.

"Orientação : Prof. Dr. Antônio Marcus Nogueira Lima".

Referências.

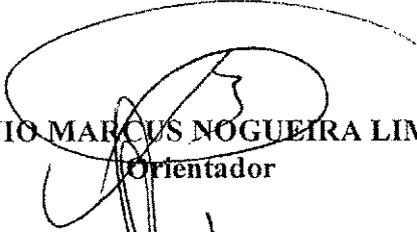
1. Sistemas a Eventos Discretos. 2. Síntese de Supervisores. 3. Teoria de Controle Supervisório. 4. Álgebra de Dióides. 5. Tese - Engenharia Elétrica. I. Lima, Antônio Marcus Nogueira. II. Universidade Federal da Paraíba - Campina Grande (PB). III. Título

CDU 621.3(043)

**SÍNTESE DE SUPERVISORES DE SISTEMAS A EVENTOS DISCRETOS
TEMPORIZADOS E NÃO TEMPORIZADOS**

EDUARD MONTGOMERY MEIRA COSTA

Tese Aprovada em 23.11.2001



PROF. ANTONIO MARCUS NOGUEIRA LIMA, Dr., UFPB
Orientador



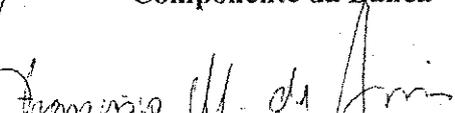
PROF. JOSÉ EDUARDO RIBEIRO CURY, Ph.D., UFSC
Componente da Banca



PROF. RAFAEL SANTOS MENDES, Dr., UNICAMP
Componente da Banca



PROF. ANGELO PERKUSICH, D.Sc., UFPB
Componente da Banca



PROF. FRANCISCO MARCOS DE ASSIS, Dr., UFPB
Componente da Banca

CAMPINA GRANDE – PB
Novembro - 2001

Síntese de Supervisores de Sistemas a Eventos Discretos
Temporizados e Não Temporizados

Eduard Montgomery Meira Costa

Tese de Doutorado apresentada em 23 de Novembro de 2001

Antonio Marcus Nogueira Lima, Dr.
Orientador

Comissão Examinadora
Angelo Perkusich, DSc.
José Eduardo Ribeiro Cury, Dr.
Rafael Santos Mendes, Dr.
Francisco Marcos de Assis, Dr.

Campina Grande, Paraíba, Brasil, 23 de Novembro de 2001

Dedicatória

Dedico esse trabalho aos curiosos e aos criativos. Aos ficcionistas e aos ocultistas. Aos que têm sede de conhecimento e que buscam solucionar os problemas do cotidiano utilizando a ciência. Aos que encontram satisfação nas descobertas do dia a dia.

À minha esposa, Ana Cristina Chaves Andrade, que reforça meus ideais a cada dia.

Aos meus filhos Ariadne Messalina B. Meira, Weena Wells Chaves Costa e Ícaro Gilgmesh B. Meira.

À meu pai, Nirso Dias Costa (*in memoriam*), o “eletricista da escola”, grande curioso e criativo, especialmente da eletricidade.

À minha mãe, Lúcia Maria Meira Costa, que me foi professora na escola e para a vida.

À Rildemar e Robenilda, amigos-irmãos que sempre estarão próximos, desde que a distância que nos separa nunca será significante.

À Gilvanildo Jeremias, que muito me acompanhou nessa trajetória Campinense.

Ao meu orientador Antonio Marcus Nogueira Lima, e aos professores Benedito Antonio Luciano e Hiran de Melo.

Aos colegas do curso, especialmente Luiz Alberto Luz de Almeida e Gláucio Brandão, e aos funcionários do DEE, especialmente à Ângela Matias, Seu Romualdo e José Roberto.

Ao destino, meu eterno conhecido.

* * *

“A imaginação é mais importante que o conhecimento.”

Albert Einstein

“A inteligência é um princípio que fala.”

Eliphas Levi

* * *

Agradecimentos

Agradeço ao professor Antonio Marcus Nogueira Lima pela sua orientação, colaboração e amizade, sem as quais não seria possível a realização deste trabalho. Ao professor José Eduardo Ribeiro Cury, pelos importantes comentários que se transformaram em grandes contribuições.

À CAPES, que proporcionou o suporte financeiro para viabilizar a realização deste trabalho.

Resumo

Esse trabalho apresenta uma formalização alternativa para a síntese de supervisores de sistemas a eventos discretos utilizando a álgebra de dióides. Tanto o sistema como a especificação de comportamento são representados por matrizes de incidência. A síntese do supervisor é realizada através de operações algébricas da álgebra de dióides. Os resultados encontrados com esta formalização são semelhantes aos resultados obtidos com o algoritmo clássico da suprema sublinguagem controlável. A complexidade computacional do algoritmo proposto é igual à complexidade do algoritmo clássico.

A formalização apresentada permite estender a formulação do problema de controle supervísório para incluir uma representação temporal, e proceder à síntese de supervisores de sistemas a eventos discretos temporizados. Essa extensão utiliza o autômato temporizado, denominado autômato $(\max,+)$, para representar sistemas a eventos discretos. De forma semelhante ao caso não-temporizado, tanto o sistema como a especificação de comportamento são representados por matrizes de incidência definidas na álgebra $(\max,+)$. A classe de sistemas a eventos discretos que pode ser tratada com o formalismo apresentado é restrita ao caso em que o tempo de vida de um evento é especificado em termos de número real que é interpretado como o tempo mínimo para a habilitação do evento. Dessa forma, a abordagem proposta determina uma formulação unificada para o tratamento desses sistemas.

Abstract

This thesis presents an alternative formulation for discrete event systems supervisor synthesis using dioid algebra. Both system model and behavior specification are interpreted as incidence matrix. The supervisor synthesis is realized through algebraic operations of the dioid algebra. The result found with this formalization are similar to the obtained result with supremal controllable sublanguage classical algorithm. The computational complexity of the proposed algorithm is equal to the classical algorithm complexity.

The presented formalization allows extend the problem formulation to include a temporal representation and proceed to the timed discrete event systems supervisor synthesis. This extension utilizes the timed automaton, called $(\max,+)$ -automaton, to represent discrete event systems. In analogous way to the untimed case, both system model and behavior specification are represented by incidence matrices defined in the $(\max,+)$ -algebra. The class of discrete event systems that can be treated with the presented formalism is restricted to the case which the event lifetime is specified in accordance to real number that is interpreted as a minimum time to event enabling. In this way, the proposed approach determines a unified formulation for the dealing of these systems.

Conteúdo

1	Introdução	6
1.1	Controle de SEDs	7
1.2	SEDs temporizados	9
1.3	SEDs temporizados e dióides	9
1.4	Objetivo	10
1.5	Organização da Tese	11
2	Teoria de Controle Supervisório	12
2.1	Conceitos	12
2.1.1	Linguagens Formais	12
2.1.2	Autômatos	16
2.1.3	Composição de Autômatos	23
2.2	A síntese do supervisor	25
2.3	Supervisores e condições de existência	27
2.4	Supervisores próprios	29
2.4.1	Algoritmo para a Construção da $\text{sup}C(L)$	34
3	Síntese de supervisores de SEDs não temporizados	39
3.1	Conceitos	39
3.1.1	Álgebra de dióides	39
3.1.2	Linguagens formais e dióides	41
3.1.3	Séries Formais	42
3.1.4	Representação matricial de um autômato	45
3.2	Síntese do Supervisor	58
3.3	Algoritmo de síntese do supervisor	84
3.4	Exemplos da utilização do algoritmo	84
3.5	Estudo da complexidade algorítmica	92
3.6	Considerações sobre a abordagem proposta	94
4	A Síntese do supervisor de SEDs temporizados	95
4.1	Descrição da representação do tempo	95

4.2	Conceitos	97
4.2.1	Álgebra $(\max, +)$	97
4.2.2	Autômatos temporizados $(\max, +)$	98
4.2.3	Autômatos $(\max, +)$ e séries formais	104
4.3	Síntese do supervisor	107
4.3.1	Matrizes de incidência temporizadas	107
4.3.2	Formulação do problema de controle	120
4.4	Algoritmo de síntese do supervisor	140
4.5	Exemplos da utilização do algoritmo	141
4.6	Complexidade algorítmica e considerações sobre o algoritmo proposto	149
5	Conclusões e perspectivas futuras	150

Lista de Figuras

1.1	Trajectoria de um SED.	7
2.1	Exemplo de um autômato.	18
2.2	Autômato com linguagem marcada $L_m(A) = \{\alpha, \alpha\alpha, \beta\alpha, \alpha\beta\alpha, \beta\alpha\alpha, \beta\beta\alpha, \dots\}$	19
2.3	(a) Gerador e geradores (b) acessível, (c) coacessível e (d) <i>trim</i>	22
2.4	(a) Autômato G_1 , (b) Autômato G_2 e (c) Composição síncrona $G = G_1 G_2$	24
2.5	(a) Autômato G_1 , (b) Autômato G_2 e (c) Composição síncrona $G_3 = G_1 G_2$	24
2.6	(a) Gerador com todos os eventos habilitados e (b) Gerador com o evento β inibido.	26
2.7	Supervisão de um SED.	27
2.8	Autômato gerador de $L(G) = \overline{(\alpha_1\beta^2 + \alpha_2)\beta^*}$	35
2.9	Especificação $L(H) = \overline{\alpha_1\beta^2} + \alpha_2\beta^*$	35
2.10	Supervisor para a sup $C(L) = L(H_3) = \alpha_2\beta^*$	36
2.11	Autômato com linguagem $L_m(G) = (\alpha\beta)^* + (\alpha\kappa\eta)^*$	36
2.12	Especificação $L(H) = (\alpha + \kappa)^* \beta (\alpha + \kappa)^* \eta$	37
2.13	Composição síncrona $H G$	37
2.14	Supervisor para sup $C(L) = L(H'_1) = \alpha\beta\alpha\kappa\eta$	38
3.1	Autômato não determinístico com mais de uma função de transição entre dois estados.	46
3.2	Autômato para exemplificar a construção da matriz de caminhos.	47
3.3	Autômato para determinação da linguagem.	49
3.4	Autômatos equivalentes.	51
3.5	Autômatos (a) G_1 , (b) G_2 e (c) G_3 - produto síncrono de G_1 e G_2	59
3.6	Autômato determinístico.	61
3.7	Autômatos (a) A e (b) A' construídos com o Algoritmo 3.2.	64
3.8	Autômato para exemplificar o Lema 3.1. e o Corolário 3.1.	75
3.9	(a) e (b) Especificações com linguagens controláveis e (c) Especificação com linguagem não controlável.	77
3.10	Autômato para exemplificar a construção da matriz de caminhos C_{uc}^n	79
3.11	Especificação não controlável.	82
3.12	Supervisor para especificação em que $ACES(\mathbf{E} \oplus \mathbf{A}_{uc}) \triangleright \mathbf{E}$	83

3.13	Autômato para o Exemplo 3.23.	85
3.14	Especificação para o autômato do Exemplo 3.23.	86
3.15	Supervisor que gera a $\text{sup } C(L)$ construído através da especificação dada no Exemplo 3.23.	87
3.16	Autômatos G_1 e G_2	89
3.17	Autômato G_3 : composição síncrona de G_1 e G_2	89
3.18	(a) Autômato $G = A_3 E_1$ e (b) Especificação de comportamento.	90
3.19	Supervisor.	92
4.1	Autômato $(\max, +)$	99
4.2	Autômato $(\max, +)$	101
4.3	(a) Autômato $(\max, +)$ e (b) Diagrama temporal de ocorrência de seus eventos α e β	102
4.4	(a) Grafo de transições de atividades, (b) Grafo de transições temporizadas e (c) Autômato $(\max, +)$	104
4.5	Autômato $(\max, +)$ determinístico.	107
4.6	Autômato $(\max, +)$ determinístico para ilustrar a representação por matriz de incidência temporizada e vetores de estados inicial e marcados.	109
4.7	Autômato para exemplo da matriz de caminhos temporizada.	110
4.8	Autômato temporizado para determinação da linguagem temporizada.	112
4.9	Autômatos $(\max, +)$ equivalentes.	114
4.10	Autômatos (a) $A_{(\max, +)_1}$, (b) $A_{(\max, +)_2}$ e (c) $A_{(\max, +)_3}$ - composição síncrona de $A_{(\max, +)_1}$ e $A_{(\max, +)_2}$	119
4.11	Autômatos $A_{(\max, +)}$, para exemplificar a construção das matrizes \mathbf{At}_c e \mathbf{At}_{uc}	121
4.12	Diagrama de um SED temporizado supervisionado.	123
4.13	Autômatos (a) $A_{(\max, +)}$ e (b) $A'_{(\max, +)}$ construídos com o Algoritmo 4.2.	127
4.14	Autômatos $(\max, +)$ para ilustrar a utilização do operador \triangleleft	131
4.15	Autômato para exemplificar a síntese do supervisor através do Lema 4.1. e do Corolário 4.1.	133
4.16	(a) e (b) Especificações controláveis e (c) Especificação não controlável.	135
4.17	Autômato utilizado como exemplo da construção da matriz de caminhos temporizada \mathbf{Ct}_{uc}^n	136
4.18	Especificação para síntese do supervisor que não satisfaz a condição de controlabilidade.	139
4.19	Supervisor para a especificação que não satisfaz a condição de controlabilidade.	140
4.20	Autômato temporizado para construir um supervisor.	142
4.21	Especificação de comportamento temporizada.	143
4.22	Supervisor que gera a $\text{sup } C(L)$ construído através da especificação de comportamento temporizada que não satisfaz a condição de controlabilidade.	144

4.23	(a) Grafo de transição temporizada, (b) Especificação de comportamento e (c) Supervisor construído pela abordagem de Brandin e Wonham.	145
4.24	(a) Autômato (max,+), (b) Especificação de comportamento e (c) Supervisor construído pela abordagem das matrizes de incidência.	146
4.25	Especificação temporizada para o exemplo das duas máquinas e um buffer. .	147
4.26	Supervisor.	149

Lista de Símbolos e Abreviações

TCS - Teoria de Controle Supervisório

SED - Sistema a Eventos Discretos

Σ - alfabeto de símbolos

Σ^* - conjunto de todas as palavras formadas pelos símbolos de Σ

s - palavra construída pela justaposição de símbolos

$\Sigma^+ = \Sigma^* - \varepsilon$

L - linguagem sobre o alfabeto Σ

\bar{L} - linguagem prefixo-fechada sobre o alfabeto Σ

A - autômato

G - autômato gerador

G_A - grafo orientado que representa o autômato

V - conjunto de vértices do grafo orientado G_A

W - conjunto de arcos do grafo orientado G_A

Q - conjunto de estados q de um autômato

Q_m - conjunto de estados marcados

δ - função de transição de um autômato

q_0 - estado inicial de um autômato

$L(G)$ - linguagem gerada por um gerador

$\bar{L}(G)$ - linguagem prefixo-fechada gerada por um gerador

$L_m(G)$ - linguagem marcada gerada por um gerador

δ^* - função de transição estendida

$\delta(\sigma, q)!$ - função de transição definida para o par (σ, q)

Ac - componente acessível do gerador

Q_{ac} - conjunto de estados da componente acessível de um gerador

$Q_{ac,m}$ - conjunto de estados marcados da componente acessível de um gerador

δ_{ac} - função δ restrita ao domínio $\Sigma \times Q_{ac}$

Σ_c - conjunto dos eventos controláveis

Σ_{uc} - conjunto dos eventos não controláveis

σ - evento

σ_{uc} - evento não controlável

σ_c - evento controlável

- $SupC(L)$ - Suprema Sublinguagem Controlável
 $\alpha, \beta, \gamma, \dots$ - símbolos de uma alfabeto Σ definindo eventos
 G_c - gerador controlável
 Σ_c - conjunto de eventos controláveis na TCS
 Σ_{uc} - conjunto de eventos não-controláveis na TCS
 Γ - conjunto associado, ou entradas de controle na TCS
 γ - entrada de controle
 S - supervisor na TCS
 H - especificação de comportamento na TCS
 Θ - função de entrada de controle ou mapa de controle
 $|s|$ - cardinalidade ou comprimento da palavra s
 $+$ - operador lógico *ou* indicando opções em linguagens formais
 S/G - sistema supervisionado ou composição síncrona do supervisor com o gerador
 X - conjunto de estados para um gerador controlado
 X_m - conjunto de estados marcados para um gerador controlado
 ξ - função de transição parcial estendida para um gerador controlado
 $K^\dagger - SupC(L)$
 \mathbb{R} - conjunto dos números reais
 \mathbb{N} - conjunto dos números naturais
 $\mathbb{B} = \{\epsilon, e\}$ - semi anel binário
 \mathbb{R}_{\max} - semi anel $(\mathbb{R} \cup \{-\infty\}, \max, +)$
 $\mathbb{B}\langle\langle\Sigma\rangle\rangle$ - semi-anel de séries formais com coeficientes em \mathbb{B} e variáveis não comutantes em Σ
 $\mathbb{R}_{\max}\langle\langle\Sigma\rangle\rangle$ - semi-anel de séries formais com variáveis não comutantes em Σ sobre \mathbb{R}_{\max}
 $(\max, +)$ - álgebra que utiliza as operações \max e $+$
 (D, \oplus, \otimes) - dióide
 \oplus - adição na álgebra de dióides
 \otimes - multiplicação na álgebra de dióides
 \oslash - dual de \otimes na álgebra de dióides ou operador de “divisão”
 \circledast - operador de interseção na álgebra de dióides para $D\langle\langle\Sigma\rangle\rangle$
 ϵ - elemento neutro na multiplicação da álgebra de dióides
 e - elemento neutro na adição da álgebra de dióides
 $\mathcal{P}(\Sigma^*)$ - conjunto de todas as linguagens formadas com símbolos do alfabeto Σ
 $T(N)$ - função de complexidade de tempo
 $O(N)$ - ordem de complexidade
 \mathbf{A} - matriz de incidência de um autômato G
 \equiv - relação de equivalência para matrizes de incidência
 \parallel - representação da composição síncrona

C^- - matriz de composição síncrona entre as partes de duas matrizes de incidência onde $\Sigma_1 \cap \Sigma_2 = \emptyset$

P - matriz de composição síncrona entre duas matrizes de incidência onde $\Sigma_1 \cap \Sigma_2 \neq \emptyset$

$L(A)$ - linguagem da matriz de incidência A

$L(A_m)$ - linguagem marcada da matriz de incidência A

π - vetor coluna com o k -ésimo elemento igual ao elemento identidade 'e' e os demais ϵ para calcular a sublinguagem de A que contém as palavras que levam a um estado k

μ - vetor coluna com o k -ésimo elemento igual ao elemento identidade 'e' e os demais ϵ para calcular uma sublinguagem marcada da matriz A que contém as palavras que levam a um estado k

ν - vetor linha onde $\nu_i = e$, e os demais elementos $\nu_k = \epsilon, k \neq i$ para determinação de uma linha coacessível da matriz A

$D_{b_{i,j}}^-$ - matriz diagonal dos elementos $b_{i,j}$

A_l^- - matriz diagonal dos auto-laços de A

$D_{b_{i,j}}^-$ - matriz diagonal dos elementos $b_{i,j}$ compostos por elementos de Σ_1 , e que não pertencem à Σ_2

A^- - matriz de incidência que contém elementos de Σ_1 , e que não pertencem à Σ_2

I - matriz identidade

A_c - matriz de incidência dos eventos controláveis do autômato G

A_{uc} - matriz de incidência dos eventos não controláveis do autômato G

E - matriz de incidência da especificação de comportamento

E^* - matriz de incidência que gera Σ^* construída a partir da expansão de E

$A^\#$ - matriz de incidência expandida de A construída através da composição síncrona de $A || E^*$

$E^\#$ - matriz de incidência expandida em forma de submatriz de $A^\#$

A^n - matriz de caminhos de comprimento n em um autômato G

C_c^n - matriz de caminhos de comprimento n , onde seus termos contém o primeiro evento controlável ou não controlável, seguido de $(n - 1)$ eventos controláveis, em um autômato G

C_{uc}^n - matriz de caminhos de comprimento n , onde seus termos contém o primeiro evento controlável ou não controlável, seguido de $(n - 1)$ eventos controláveis, em um autômato G

B_{uc}^n - matriz de caminhos de comprimento n , onde seus termos contém o primeiro evento controlável ou não controlável, pertencente à especificação de comportamento, seguido de $(n - 1)$ eventos não controláveis, do autômato G

ϵ - linguagem vazia \emptyset quando usada a álgebra de dióides

e - palavra vazia ϵ quando usada a álgebra de dióides

ACES - operador que retorna uma matriz apenas com estados acessíveis

COACES - operador que retorna 0 se a matriz não é coacessível e 1 se é

TRIM - operador que retorna 0 se a matriz não é trim e 1 se é

- S** - matriz de incidência do supervisor
- \triangleright - contém ou é igual no contexto de matrizes que têm relações entre si
- \trianglelefteq - está contido ou é igual no contexto de matrizes que têm relações entre si
- \triangleright - contém no contexto de matrizes que têm relações entre si
- \triangleleft - está contido no contexto de matrizes que têm relações entre si
- $A_{(max,+)}$ - autômato temporizado (max,+)
- x_i - instante de tempo da n -ésima ocorrência do evento etiquetado por i
- Y** - datador de um autômato $A_{(max,+)}$
- $Y_{\mathbb{B}\langle\Sigma\rangle}$ - série formal representando uma linguagem regular não temporizada
- $Y_{\mathbb{R},max}\langle\Sigma\rangle$ - série formal representando uma linguagem regular temporizada
- $(y|s)$ - coeficiente de uma palavra em uma série formal ou tempo da palavra s em um autômato $A_{(max,+)}$, igual a ϵ se $s \notin L$ e a e se $s \in L$
- θ - atraso inicial em um autômato $A_{(max,+)}$
- ϕ - atraso final em um autômato $A_{(max,+)}$
- T** - tempo de transição em um autômato $A_{(max,+)}$
- P** - função peso de um caminho p em um autômato $A_{(max,+)}$
- $\mu(\cdot)$ - matriz com valores **T** de um autômato $A_{(max,+)}$
- At** - matriz de incidência temporizada de um autômato $A_{(max,+)}$
- $L(\mathbf{At})$ - linguagem da matriz de incidência temporizada **At** igual a série formal $Y_{\mathbb{R},max}\langle\Sigma\rangle$
- At_c** - matriz de incidência temporizada dos eventos controláveis do autômato $A_{(max,+)}$
- At_{uc}** - matriz de incidência temporizada dos eventos não controláveis do autômato $A_{(max,+)}$
- Et** - matriz de incidência da especificação de comportamento temporizada
- t_σ - tempo de vida de um evento σ
- Ct⁻** - matriz de composição síncrona entre as partes de duas matrizes de incidência temporizadas onde $\Sigma_1 \cap \Sigma_2 = \emptyset$
- Pt** - matriz de composição síncrona entre duas matrizes de incidência temporizadas onde $\Sigma_1 \cap \Sigma_2 \neq \emptyset$
- Dt⁻_{bt_{i,j}}** - matriz diagonal dos elementos $bt_{i,j}$ compostos por elementos de Σ_1 , e que não pertencem à Σ_2
- At⁻_i** - matriz diagonal dos auto-laços de **At** compostos por elementos de Σ_1 , e que não pertencem à Σ_2
- At⁻** - matriz de incidência temporizada que contém elementos de Σ_1 , e que não pertencem à Σ_2
- Et*** - matriz de incidência que gera Σ^* construída a partir da expansão de **Et**
- At[#]** - matriz de incidência temporizada expandida de **At** construída através da composição síncrona de **At**||**Et***
- Et[#]** - matriz de incidência temporizada expandida em forma de submatriz de **At[#]**
- Atⁿ** - matriz de caminhos temporizada de comprimento n em um autômato $A_{(max,+)}$

\mathbf{Ct}_c^n - matriz de caminhos temporizada de comprimento n , onde seus termos contém o primeiro evento controlável ou não controlável, seguido de $(n - 1)$ eventos controláveis, em um autômato $A_{(\max,+)}$

\mathbf{Ct}_{uc}^n - matriz de caminhos temporizada de comprimento n , onde seus termos contém o primeiro evento controlável ou não controlável, seguido de $(n - 1)$ eventos controláveis, em um autômato $A_{(\max,+)}$

\mathbf{Bt}_{uc}^n - matriz de caminhos temporizada de comprimento n , onde seus termos contém o primeiro evento controlável ou não controlável, pertencente à especificação de comportamento, seguido de $(n - 1)$ eventos não controláveis, do autômato $A_{(\max,+)}$

\mathbf{St} - matriz de incidência do supervisor temporizado

Chapter 1

Introdução

Os Sistemas a Eventos Discretos (SEDs) [RW82, RW89, Sme89, CR90, Zil93, Sme93] são sistemas que estão presentes em várias aplicações do cotidiano, como por exemplo a automação de manufatura, a robótica, a supervisão de tráfego aéreo e ferroviário, a logística (canalização e armazenamento de produtos, organização e prestação de serviços), sistemas operacionais, redes de computadores, concepção de software, gerenciamento de bases de dados e otimização de processos distribuídos. O estudo desses sistemas tem crescido muito nos últimos anos devido a sua importância industrial e sua complexidade e custo [MB97, Cot99, Tor99, HML00, Pu00, ABH01, CTC01, Zha01]. Isso, justifica o esforço despendido em sua análise e projeto.

Os SEDs apresentam em comum a forma de interagir com o ambiente. Nessa interação, várias ocorrências causam mudanças na configuração interna, ou estado do sistema. Essas ocorrências são denominadas de *eventos*. Eventos podem ser exemplificados pelo início e pelo término de uma atividade em uma máquina, pela transmissão e pela recepção de dados em um sistema de comunicação, pela partida e pela chegada de trens (ou aviões) em uma estação (ou aeroporto). Dessa forma, os eventos em um SED são, por sua natureza, instantâneos, o que lhes confere um caráter de sistema de transições discretas com relação ao tempo.

A evolução dinâmica destes sistemas pode ser exemplificada pelo gráfico apresentado na Figura 1.1, onde os eventos são representados pelas letras gregas α , β , γ , δ e ε , e os estados do sistema são representados por q_0 , q_1 , q_2 , q_3 e q_4 , em que q_0 é o estado inicial do SED. O estado inicial q_0 é o estado do SED antes da ocorrência do primeiro evento. Neste exemplo, após a seqüência de eventos observa-se que o SED retorna ao estado inicial q_0 . Este processo é denominado *reinicialização*. Se uma seqüência de eventos resulta sempre o retorno do SED a um determinado estado, denomina-se este estado de estado recorrente (*home state*).

Os SEDs têm como características fundamentais:

- Ciclo de funcionamento descrito através do encadeamento de eventos;
- Ocorrência de eventos em paralelo;

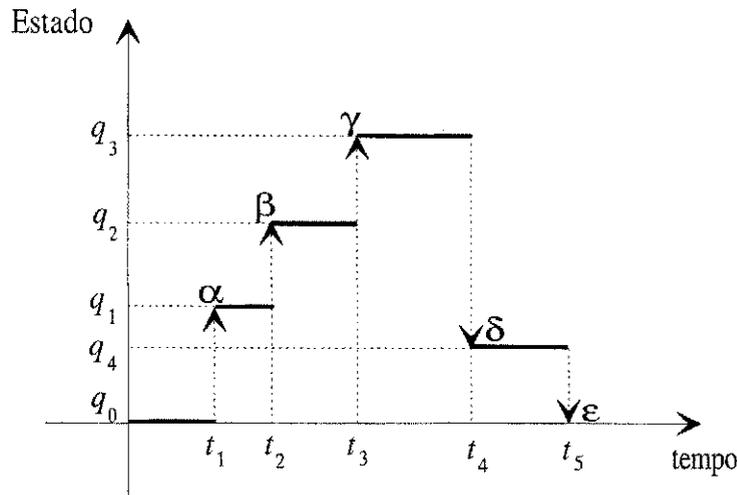


Figure 1.1: Trajetória de um SED.

- Necessidade de sincronização.

Devido à sua natureza de tempo discreto e às suas características, os SEDs não podem ser tratados adequadamente através dos modelos matemáticos convencionais baseados em equações diferenciais. Contudo, como são sistemas de alta relevância no mundo moderno, torna-se necessário encontrar soluções para problemas relacionados ao seu controle. Assim, vários paradigmas podem ser usados na modelagem dos SEDs, como as cadeias de Markov [CH90], a teoria das filas [CH90], os processos semi-markovianos generalizados [CH90], a álgebra de processos [Mil80, Hoa85], a teoria de linguagens formais e os autômatos [HU79], as redes de Petri [Rei85, Mur89, Rei92, DHP⁺93, ZD93] e a álgebra de dióides [BCOQ92, Bea96]. Cada um desses paradigmas se aplicam à problemas distintos e que necessitam de uma descrição específica. Entretanto, nenhum deles se tornou um paradigma universal para solucionar o problema de controle de SEDs, abrangendo suas mais variadas classes (SEDs cíclicos e acíclicos, temporizados e não temporizados, estocásticos e outros).

1.1 Controle de SEDs

O controle de um SED é determinado por inibições de eventos habilitados nos estados alcançados em sua evolução dinâmica. Assim, esse controle define a trajetória que representa um comportamento especificado. Geralmente, os SEDs apresentam em sua estrutura eventos que podem ser inibidos e eventos que não podem ser inibidos. Dessa forma, os eventos em um SED são classificados em duas categorias distintas, denominados eventos controláveis e eventos não controláveis. Um evento dito controlável é aquele para o qual há uma maneira para permitir ou inibir sua ocorrência. O início de atividade de uma máquina e o envio de uma mensagem podem ser classificados como eventos controláveis, pois sempre é possível não iniciar a atividade da máquina ou não enviar uma mensagem. Por outro lado, um

evento é dito não controlável quando sua ocorrência não pode ser inibida, isto é, tal evento ocorre espontaneamente. A quebra de uma esteira, o término do processamento de uma peça em uma máquina, a chegada de uma mensagem enviada são exemplos de eventos não controláveis.

Considerando a existência dos eventos controláveis e dos eventos não controláveis, o controle do SED deve garantir que em nenhum dos estados alcançados, haja possibilidades de ocorrências de eventos não controláveis que não estejam incluídos no comportamento especificado. Isso, porque um evento não controlável em um estado alcançado do SED pode sempre ocorrer, desviando a trajetória do comportamento especificado.

Entre as formalizações de controle de SEDs, encontra-se a Teoria de Controle Supervisório (TCS) [RW89, KG91, Thi96]. Esta foi desenvolvida utilizando as linguagens formais e os autômatos [HU69, HU79, Vel79] como ferramenta de modelagem do SED. Na TCS, define-se que o supervisor é o agente responsável pela avaliação dos eventos gerados pelo SED e pela determinação da ação de controle a ser aplicada para a execução de uma tarefa.

A TCS provê uma forma de controlar os SEDs, separando explicitamente o sistema a ser controlado (*open loop dynamics*), do controlador propriamente dito (*feedback control*). A partir dessa concepção são apresentadas as condições necessárias e suficientes para a existência do supervisor. A formulação genérica do problema de controle no contexto da TCS visa a determinação do autômato supervisor a partir do modelo do sistema a ser controlado e da especificação de comportamento requerida para o SED. A função do supervisor é assegurar que a linguagem gerada pela sua composição com o SED que se deseja controlar, represente a especificação de comportamento desejada, ou seja, fazer com que o SED realize uma tarefa específica. Esta teoria foi desenvolvida para construir supervisores para SEDs não temporizados. Dessa forma, sua aplicação garante condições específicas para construir supervisores onde é necessário apenas determinar a evolução do SED, não definindo instantes de tempo exatos em que os eventos devam ocorrer, nem o tempo total para a realização das tarefas. Ou seja, o supervisor define apenas o comportamento lógico do SED. Assim, embora o supervisor sintetizado garanta a realização da tarefa especificada, nada se pode dizer sobre o tempo de sua execução. Isto é, uma tarefa que poderia ser executada em um determinado tempo pode ser realizada em um tempo maior que o devido. Esta situação ilustra uma das restrições para o uso da TCS, quando se considera a aplicação a processos automáticos que exigem a determinação de controle em tempos determinados. Em muitos casos, os processos podem se tornar lentos e apresentar ociosidade parcial (parte do sistema se mantém sem utilização temporária) devido ao assincronismo de suas partes. Esse problema é tratado quando se incorpora a representação do tempo no modelo do sistema.

1.2 SEDs temporizados

O tratamento de SEDs temporizados [OW90a, Bra92, Wan92, CG96] exige a incorporação da representação do tempo em seus modelos. Várias abordagens foram formuladas para tratar desse problema. Dentre elas, em Brandin e Wonham [BW94] o problema é formalizado através da inclusão de um relógio global. Esta abordagem é muito similar ao tratamento com os SEDs não temporizados [RW89], incluindo um temporizador no modelo que é um evento denominado 'tick'. A inclusão do evento 'tick' que é sincronizado com o término de um ciclo do relógio global, resulta num aumento significativo no número de transições do modelo e, conseqüentemente, no número de estados do SED. A abordagem necessita de dois paradigmas: 1) os grafos de transições de atividades, utilizados em composições e 2) os grafos de transições temporizadas, utilizados na síntese. O sistema pode ser descrito por eventos que apresentam tempo mínimo e máximo para suas ocorrências, podendo o tempo máximo da ocorrência de um evento ser definido como infinito. Uma formalização similar é apresentada em Brave e Heymann [BH88] que introduz intervalos de tempo para as possíveis ocorrências dos eventos habilitados, relativos aos instantes de tempo em que o estado é alcançado. Nesse trabalho é demonstrado como as características temporais e lógicas podem ser separadas para tratamentos independentes. Similarmente, em Kozák [Koz91] é especificado um tempo mínimo de atraso entre as ocorrências dos eventos, que podem depender da completa história passada dos eventos. Tanto em Brave e Heymann, como em Kozák o exato atraso de tempo entre quaisquer dois eventos é desconhecido. Em Ahur e Dill [AD95] o sistema é modelado por um autômato temporizado [AD90, AD94, Bal96, Alu97, OHY⁺99] para a análise de sistemas em tempo real.

Para solucionar os problemas de sincronização e permitir a análise de desempenho para otimizar a automação de processos em SEDs temporizados, a álgebra de dióides [Gre79, GTC⁺98, GG98, ABG98, Gau95b, GAC⁺97, TGG97] se apresenta como uma teoria de alta importância.

1.3 SEDs temporizados e dióides

A álgebra de dióides, recentemente utilizada na análise de sistemas lineares [EK82, GAC⁺90, Gau92, AM00] e também para análise de sistemas clássicos de controle por Vidyasagar [Vid85] e Kucera [Kuc91], foi utilizada por Cohen *et al.* [CDQV85] para descrever o comportamento de SEDs temporizados cíclicos, cuja dinâmica é caracterizada por aspectos de sincronização. Nessa formalização, o comportamento do SED é descrito pelos tempos de ocorrência dos eventos. Isso determina a representação de sua dinâmica através de um modelo linear, onde o controle é feito através de atrasos no início de tarefas (uso de recursos), como pode ser visto em [BCOQ92, CG96, Men97, Boi99, Gau99].

Nesse contexto, são encontrados vários trabalhos para síntese e análise de SEDs tempo-

rizados, seja utilizando como ferramenta de modelagem os grafos de eventos temporizados [CGQ95, GM97, Cot99, Lea00, Lib96, Nor99] (redes de Petri temporizadas [Chr83, Fre91, CCS91], onde todos os lugares apresentam marcação binária com apenas uma transição de entrada e uma transição de saída), seja utilizando os autômatos temporizados [AD90, AD94, Bou98, PC97, BP99, TB99, TKB99, CJ99]. No caso dos autômatos temporizados, encontram-se trabalhos utilizando a álgebra de dióides na avaliação de desempenho de SEDs temporizados, como [CMQV89, Gau93, CGQ95, GM95, HUU98, BJG98], onde entre as ferramentas matemáticas usadas, encontram-se as séries formais [BR88, Gau94b, Gau94a, Gau94c, BJG95, Kli99].

Embora a utilização da álgebra de dióides seja de alta importância no estudo dos SEDs temporizados, deve-se observar que sua utilização como definida por Baccelli *et al.* [BCOQ92] não define uma separação explícita entre os eventos controláveis e não controláveis. Apenas há a possibilidade de atrasar a ocorrência de eventos em um determinado estado utilizando para isto uma entrada de controle. Já na abordagem de Cofer e Garg [CG96] a separação dos eventos em controláveis e não controláveis é considerada, podendo ser utilizada também para o modelo não temporizado. O sistema é descrito por um sistema linear, em que os vetores de estados, a especificação de comportamento e a matriz do sistema são formalizados por vetores de funções polinomiais que definem a evolução do sistema e, através de uma formulação que utiliza a álgebra de dióides, garantem as condições para a construção de supervisores. Em Gaubert [Gau93, CGQ95], o formalismo apresentado abrange em particular os sistemas lineares $(\max, +)$ e uma classe de SEDs temporizados estocásticos, podendo ser aplicada a SEDs determinísticos e não determinísticos.

A utilização da álgebra de dióides permite, então, várias formas de abordar o problema de controle de SEDs, além da descrição de suas linguagens. Seu uso em conjunto com as matrizes de incidência do autômato que modela um SED, resulta numa formalização para a síntese de supervisores para sistemas não temporizados determinísticos, que apresentam um número finito de estados [CL01a], bem como para a síntese de supervisores de SEDs temporizados [CL01b]. Logo, esta álgebra se mostra como uma ferramenta de alta importância para ser utilizada na reescritura do problema de controle de SEDs não temporizados, além de poder ser estendida para os SEDs temporizados, de forma a formalizar um tratamento unificado na síntese de supervisores de SEDs.

1.4 Objetivo

Essa Tese tem como objetivo reescrever a formulação do algoritmo de síntese de supervisores de SEDs apresentada em Ramadge e Wonham [RW87], utilizando para isto a álgebra de dióides e as matrizes de incidência.

A partir desta reescritura do algoritmo e utilizando uma representação do tempo, é apre-

sentado um algoritmo de síntese de supervisores de SEDs temporizados utilizando uma descrição similar à abordagem não temporizada, com a introdução das matrizes de incidência temporizadas e seus vetores de estados. Essa extensão utiliza o autômato temporizado, denominado autômato $(\max,+)$, para representar SEDs. De forma semelhante ao caso não-temporizado, tanto o sistema como a especificação de comportamento são representados por matrizes de incidência definidas na álgebra $(\max,+)$.

Os SEDs que essa Tese permite tratar são os SED representáveis por autômatos determinísticos finitos através de suas matrizes de incidência. Isto é, SEDs que possam ser representados por uma linguagem regular não temporizada, ou uma série formal (linguagem regular temporizada).

1.5 Organização da Tese

Para a compreensão desse trabalho, torna-se necessário, inicialmente, o conhecimento da TCS. Também, a conceituação das ferramentas utilizadas e o formalismo para a construção do supervisor. Dessa forma, essa Tese está apresentada da seguinte maneira:

Capítulo 2: É apresentada a Teoria de Controle Supervisório de Ramadge e Wonham [RW89]

Capítulo 3: É apresentada a abordagem da síntese de supervisores de SEDs não temporizados utilizando as matrizes de incidência e a álgebra de dióides;

Capítulo 4: É apresentada a abordagem da síntese de supervisores de SEDs temporizados utilizando as matrizes de incidência temporizadas e a álgebra de dióides, mais precisamente a álgebra $(\max,+)$ e as séries formais;

Capítulo 5: São apresentadas as conclusões e perspectivas futuras deste trabalho.

Em cada Capítulo são apresentados os conceitos básicos utilizados no seu respectivo desenvolvimento.

Chapter 2

Teoria de Controle Supervisório

A Teoria de Controle Supervisório (TCS) foi desenvolvida por Ramadge e Wonham [RW89], utilizando como paradigma de modelagem as linguagens formais e os autômatos [Gil62, HU69, Sal73, HU79].

Nesse Capítulo é apresentada essa teoria que é utilizada como base para o trabalho desenvolvido nessa Tese. Para o entendimento da TCS é necessário a apresentação dos conceitos das linguagens formais e dos autômatos.

2.1 Conceitos

2.1.1 Linguagens Formais

As linguagens formais são representadas por alfabetos de símbolos (letras ou dígitos), onde um determinado símbolo é uma entidade não definida formalmente. Um conjunto não vazio de símbolos, define um alfabeto, o qual, geralmente é representado pela letra grega Σ .

Exemplo 2.1 $\Sigma = \{\alpha, \beta, \gamma, \delta\}$, é um alfabeto, e os símbolos $\alpha, \beta, \gamma, \delta$ são seus elementos.

A justaposição, ou concatenação finita de símbolos, define uma palavra. Esta operação é definida por

$$cat_{\Sigma} : \Sigma \times \Sigma \rightarrow \Sigma^*$$

onde Σ^* é o conjunto de todas as palavras construídas com os símbolos de Σ .

Assim, uma palavra de dois símbolos, denotada por s , é definida por $s = cat_{\Sigma}(\alpha, \beta) = \alpha\beta$, onde $\alpha, \beta \in \Sigma$.

Com a justaposição de símbolos definindo palavras, também define-se o comprimento de uma palavra.

Definição 2.1 O comprimento de uma palavra s , representado pela cardinalidade $|s|$, é igual ao número de símbolos que a compõe.

Define-se na Teoria de Linguagens Formais a palavra nula, representada por ε , a qual é a única palavra de comprimento nulo, isto é, $|\varepsilon| = 0$. Desta forma, $\varepsilon \notin \Sigma$ pois é uma palavra, e não um símbolo do alfabeto Σ .

A operação de concatenação de símbolos é estendida para palavras como na definição a seguir.

Definição 2.2 *A concatenação de palavras é a operação*

$$cat_s : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*,$$

onde

$$\begin{aligned} cat_s(\varepsilon, s) &= cat_s(s, \varepsilon) = s, & s \in \Sigma^* \\ cat_s(s_1, s_2) &= s_1 s_2 = s, & s_1, s_2 \in \Sigma^+. \end{aligned}$$

Assim, pela Definição 2.2, dadas duas palavras u e v sobre um alfabeto Σ , com

$$\begin{aligned} u &= cat_\Sigma(\sigma_1, \dots, \sigma_k) = \sigma_1 \dots \sigma_k, \\ v &= cat_\Sigma(\sigma_{k+1}, \dots, \sigma_n) = \sigma_{k+1} \dots \sigma_n, \end{aligned}$$

sua concatenação define uma nova palavra que é

$$s = cat_s(u, v) = uv = \sigma_1 \dots \sigma_k \sigma_{k+1} \dots \sigma_n. \quad (2.1)$$

Observa-se que a palavra vazia ε é o elemento identidade na concatenação, ou seja, toda palavra s concatenada com a palavra vazia ε é sempre igual à mesma palavra. Também, observa-se que a operação de concatenação é uma operação associativa, pois

$$\begin{aligned} cat_\Sigma(cat_\Sigma(\sigma_1, \sigma_2), \sigma_3) &= cat_\Sigma(\sigma_1, cat_\Sigma(\sigma_2, \sigma_3)) & e \\ cat_s(cat_s(s_1, s_2), s_3) &= cat_s(s_1, cat_s(s_2, s_3)). \end{aligned}$$

Um conjunto de palavras é definido como:

Definição 2.3 *Dado $k \in \mathbb{N}$, denota-se por Σ^k o conjunto de todas as palavras sobre Σ cujo comprimento é igual a k .*

Exemplo 2.2 *De acordo com a Definição 2.3 e considerando $\Sigma = \{\alpha, \gamma\}$, os conjuntos de palavras de comprimento k , são definidos por:*

$$\begin{aligned} \Sigma^0 &= \{\varepsilon\}; \\ \Sigma^1 &= \{\alpha, \gamma\}; \\ \Sigma^2 &= \{\alpha\alpha, \alpha\gamma, \gamma\alpha, \gamma\gamma\}; \\ &\vdots \end{aligned}$$

Como consequência da Definição 2.3, definem-se dois conjuntos especiais, que são Σ^+ e Σ^* , dados por

$$\begin{aligned}\Sigma^+ &= \bigcup_{k=1}^{\infty} \Sigma^k = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \dots \\ \Sigma^* &= \bigcup_{k=0}^{\infty} \Sigma^k = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \dots = \Sigma^0 \cup \Sigma^+.\end{aligned}$$

Vê-se que Σ^+ é um conjunto composto de todas as palavras construídas dos elementos do alfabeto Σ , cujo comprimento é maior que zero e Σ^* é a união de Σ^+ com a palavra vazia ε .

Um conjunto de palavras formadas com os símbolos de um alfabeto Σ , é definido como uma linguagem, ou seja:

Definição 2.4 Dado um alfabeto Σ , L é uma linguagem sobre Σ se e somente se, $L \subseteq \Sigma^*$.

Com essa definição, observa-se que uma linguagem tanto pode ser $L = \Sigma^*$ como $L = \{ \}$ (linguagem vazia, a qual é diferente da linguagem Σ^0 , composta apenas da palavra vazia ε).

Define-se o prefixo de uma palavra s , como sendo uma parte inicial de comprimento arbitrário de uma palavra s . Formalmente:

Definição 2.5 O prefixo de uma palavra s sobre um alfabeto Σ é qualquer palavra $u \in \Sigma^*$ que possa ser completada com outra palavra $v \in \Sigma^*$ para formar a palavra s .

Exemplo 2.3 Da Definição 2.2, u é um prefixo de s visto que $\exists (v = \sigma_{k+1} \dots \sigma_n) \in \Sigma^*$ tal que, $uv = s$.

Assim, todos os prefixos de uma dada palavra s incluindo a palavra vazia ε , formam um conjunto, denotado por $Pre(s)$.

O conjunto que inclui todas as palavras de uma linguagem $L \subseteq \Sigma^*$ e todos os seus prefixos é definido como:

Definição 2.6 O prefixo-fechamento, ou fechamento de L , é dado por:

$$\bar{L} = \{u | \exists v \in \Sigma^* \wedge uv \in L\}. \quad (2.2)$$

Dessa definição vê-se que, para uma linguagem $L \subseteq \Sigma^*$, existe uma linguagem associada, denotada por \bar{L} , a qual é formada pelas palavras de L e por todos os seus prefixos. Disto, conclui-se que $L \subseteq \bar{L}$. Uma linguagem é dita ser prefixo-fechada se e somente se $L = \bar{L}$.

Outras operações definidas para as linguagens, são o fechamento-Kleene

$$L^* = \{\varepsilon\} \cup L \cup LL \cup LLL \cup \dots$$

que é uma operação idempotente, isto é, $(L^*)^* = L^*$; união

$$L_1 \cup L_2 = \{s | s \in L_1 \vee s \in L_2\};$$

intersecção

$$L_1 \cap L_2 = \{s \mid (s \in L_1) \wedge (s \in L_2)\}$$

e complemento

$$L^c = \{s \in \Sigma^* \mid s \notin L\}.$$

Com a definição de linguagem, pode-se estender a operação de concatenação para as linguagens como na definição a seguir.

Definição 2.7 A concatenação de linguagens é a operação

$$cat_L : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*,$$

onde

$$cat_L(\{\varepsilon\}, L) = cat_L(L, \{\varepsilon\}) = L, L \subset \Sigma^*$$

$$cat_L(L_1, L_2) = \{s_1 s_2 \mid s_1 \in L_1 \wedge s_2 \in L_2\}.$$

Exemplo 2.4 Dadas as linguagens $L_1 = \{\alpha, \alpha\beta\}$ e $L_2 = \{\kappa, \beta\eta\}$, tem-se

$$cat_L(\{\varepsilon\}, L_1) = cat_L(L_1, \{\varepsilon\}) = \{\alpha, \alpha\beta\}$$

$$cat_L(\{\varepsilon\}, L_2) = cat_L(L_2, \{\varepsilon\}) = \{\kappa, \beta\eta\}$$

$$cat_L(L_1, L_2) = \{\alpha\kappa, \alpha\beta\eta, \alpha\beta\kappa, \alpha\beta\beta\eta\}$$

$$cat_L(L_2, L_1) = \{\kappa\alpha, \kappa\alpha\beta, \beta\eta\alpha, \beta\eta\alpha\beta\}.$$

Expressões Regulares

No estudo das linguagens formais, algumas linguagens, podem ser representadas através de uma expressão regular, utilizando a álgebra convencional e os símbolos do alfabeto. Assim, linguagens complexas podem ser representadas em termos de expressões simples, como apresentado por Hopcroft e Ullman [HU79]:

- σ^* - representa a repetição do símbolo σ , por um número arbitrário de vezes;
- s^* - representa a repetição da palavra s , por um número arbitrário de vezes;
- $+$ - símbolo empregado como o operador lógico *ou*, indicando uma opção entre duas ou mais possibilidades.

As expressões regulares são definidas recursivamente da seguinte maneira:

1. \emptyset é uma expressão regular denotando o conjunto vazio, ε é uma expressão regular denotando o conjunto $\{\varepsilon\}$ e σ é uma expressão regular denotando o conjunto $\{\sigma\}$ para todo $\sigma \in \Sigma$;
2. Se s_1 e s_2 são expressões regulares, então $s_1 s_2$, s_1^* , s_2^* e $(s_1 + s_2)^*$ são expressões regulares;

3. Toda expressão regular é construída através da aplicação das regras 1 e 2 um número finito de vezes.

As expressões regulares provêem uma representação compacta para linguagens complexas formadas por um número infinito de palavras. A palavra vazia ε e a linguagem vazia \emptyset , também são consideradas nas expressões regulares para as quais têm-se as seguintes propriedades [Won99]:

$$\varepsilon s = s\varepsilon = s,$$

$$\varepsilon^* = \varepsilon,$$

e

$$\emptyset + L = L,$$

$$\emptyset L = L\emptyset = \emptyset,$$

$$\emptyset^* = \varepsilon.$$

Exemplo 2.5 Para $\Sigma = \{\alpha, \beta\}$ e uma linguagem prefixo-fechada L na qual os símbolos aparecem alternadamente, com todas as palavras iniciando sempre por α , tem-se:

$$L = \{\varepsilon, \alpha, \alpha\beta, \alpha\beta\alpha, \alpha\beta\alpha\beta, \dots\} = (\alpha\beta)^*(\varepsilon + \alpha). \quad (2.3)$$

O termo apresentado depois do segundo símbolo de igualdade pode ser lido da seguinte maneira: $\alpha\beta$ pode não ocorrer, e ocorrer ε ou α , ou $\alpha\beta$ pode ocorrer um número arbitrário de vezes e, logo após, α ocorre, ou ε (nada). Observe que L contém uma parte das palavras contidas em $\Sigma^* = \{\alpha^*\beta^*\}^*$, isto é, L é um subconjunto próprio da linguagem Σ^* .

2.1.2 Autômatos

Os autômatos são modelos matemáticos de máquinas de estados que têm entradas e saídas discretas e que reconhecem um conjunto de palavras sobre um dado alfabeto Σ . Os símbolos são lidos sequencialmente e dessa forma, um autômato pode ser visto como uma entidade de controle que tem uma variável interna que representa seu estado. Assim, cada símbolo lido resulta na atualização desta variável de acordo com uma função de transição que associa um novo estado a cada par (*evento, estado*). O conjunto de todos os estados do autômato é representado por Q . O estado inicial de um autômato é designado por q_0 . Os estados marcados, representados pelo conjunto $Q_m \subseteq Q$, são os estados que o autômato atinge ao processar palavras reconhecidas.

Os autômatos podem ser finitos ou infinitos, além de determinísticos ou não determinísticos e, seu estado atual é necessário na determinação dos estados futuros. Um autômato determinístico finito, consiste de um conjunto finito de estados Q e um conjunto finito de transições, denotado por δ . Essas transições são etiquetadas com símbolos do alfabeto Σ .

Um autômato é dito determinístico se, de cada estado não houver transições etiquetadas com os mesmos símbolos e direcionadas para mais de um estado e apresentar apenas um único estado inicial.

Definição 2.8 Um autômato finito determinístico, ou simplesmente um autômato é uma quintupla

$$A = (Q, \Sigma, \delta, q_0, Q_m)$$

onde:

- $Q = \{q_1, \dots, q_n\}$ é um conjunto finito de estados;
- $\Sigma = \{\sigma_1, \dots, \sigma_m\}$ é o alfabeto ou conjunto de símbolos;
- $\delta : \Sigma \times Q \rightarrow Q$ é a função de transição de estados, onde

$$\begin{aligned} \delta(\varepsilon, q) &= q & e \\ \delta(\sigma, q) &= q', \quad \text{para } q, q' \in Q \wedge \sigma \in \Sigma; \end{aligned} \tag{2.4}$$

- $q_0 \in Q$ é o estado inicial;
- $Q_m \subseteq Q$ é o conjunto de estados marcados.

Observando a função de transição de estados, vê-se que q' só será um estado do autômato A , se σ for uma transição definida no estado q .

Os autômatos são representados graficamente por um grafo direcionado, onde seus vértices são os estados e os arcos são as funções de transição. Os estados marcados são vértices representados por círculos concêntricos e o estado inicial indicado por uma seta que não é saída de nenhum vértice. Formalmente:

Definição 2.9 Um autômato $A = (Q, \Sigma, \delta, q_0, Q_m)$ é representado graficamente pelo grafo orientado $G_A = \{V, W\}$, onde

$$\begin{aligned} V &= Q & e \\ W &= \{(q, q', \sigma) \mid q, q' \in Q \wedge \sigma \in \Sigma \wedge \delta(\sigma, q) = q'\}. \end{aligned}$$

Exemplo 2.6 Um exemplo de autômato é apresentado na Figura 2.1, onde:

- $\Sigma = \{\alpha, \beta, \gamma\}$;
- $Q = \{0, 1, 2\}$;
- $\delta(\alpha, 0) = 1, \delta(\alpha, 1) = 2, \delta(\alpha, 2) = 2, \delta(\beta, 0) = 2, \delta(\beta, 1) = 0, \delta(\beta, 2) = 1, \delta(\gamma, 1) = 1, \delta(\gamma, 2) = 0$;
- $q_0 = 0$ & e
- $Q_m = \{1, 2\}$.

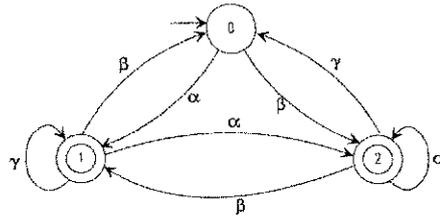


Figure 2.1: Exemplo de um autômato.

Observa-se que o autômato é construído através da função de transição, onde $\delta(x, y)$ é a posição de término, ou seja, o estado para onde o arco direcionado aponta, x é a transição que muda seu estado e y é o estado, ou vértice de origem do arco: em outras palavras, será o arco que indica uma transição x que vai do estado y para o estado $\delta(x, y)$.

Estendendo a função de transição, permite-se descrever o processamento de palavras em um autômato.

Definição 2.10 Para um autômato $A = (Q, \Sigma, \delta, q_0, Q_m)$ a função de transição estendida δ^* é definida por

$$\delta^* : \Sigma^* \rightarrow Q,$$

de tal forma que:

$$\begin{aligned} \delta^*(\varepsilon, q) &= q & e \\ \delta^*(s\sigma, q) &= \delta(\sigma, \delta^*(s, q)) & \text{para } q \in Q \wedge s \in \Sigma^* \end{aligned} \quad (2.5)$$

Geralmente, utiliza-se δ ao invés de δ^* , desde que

$$\delta^*(\sigma, q) = \delta(\sigma, \delta^*(s, q)) = \delta(\sigma, q)$$

para o caso em que $s = \varepsilon$.

Conhecidas a função de transição e o estado atual do autômato, é possível determinar seu estado após o processamento de um símbolo qualquer. Assim, processando uma palavra a partir de um determinado estado, confirma-se se o estado alcançado pertence ou não ao conjunto de estados marcados Q_m .

Definição 2.11 Dado um autômato $A = (Q, \Sigma, \delta, q_0, Q_m)$, a linguagem marcada ou reconhecida de A , é definida por:

$$L_m(A) = \{s | s \in \Sigma^* \wedge \delta(s, q_0) \in Q_m\}. \quad (2.6)$$

A linguagem reconhecida por um autômato pode ser encontrada, seguindo os arcos orientados de seu grafo, iniciando do estado inicial até os estados marcados.

Exemplo 2.7 Na Figura 2.2, é apresentado um autômato, cuja linguagem marcada é

$$L_m(A) = \{\alpha, \alpha\alpha, \beta\alpha, \alpha\beta\alpha, \beta\alpha\alpha, \beta\beta\alpha, \dots\} = (\alpha(\alpha^* + \beta^*\alpha))^*,$$

e consiste de todas as palavras formadas por α e β , sempre seguidas do evento α .

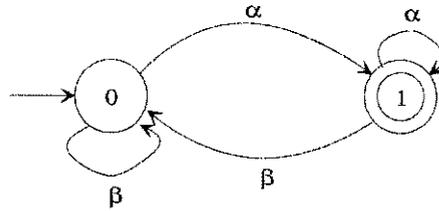


Figure 2.2: Autômato com linguagem marcada $L_m(A) = \{\alpha, \alpha\alpha, \beta\alpha, \alpha\beta\alpha, \beta\alpha\alpha, \beta\beta\alpha, \dots\}$.

Geradores

Um gerador é um autômato no qual a função de transição é definida para um subconjunto próprio de Σ^* , isto é, a função de transição permite uma representação de uma parte das palavras contidas em Σ^* . Os geradores são mais compatíveis com a representação dos SEDs.

Definição 2.12 *Um gerador é uma quintupla*

$$G = (Q, \Sigma, \delta, q_0, Q_m)$$

onde os elementos Q , Σ , q_0 , Q_m e δ têm a mesma definição do autômato (Definição 2.8), com

$$\delta : Q \times \Sigma \rightarrow Q$$

definida como a função, geralmente parcial, de transição de estados. Isto é, para cada $q \in Q$, δ é definida apenas para um subconjunto de elementos $\sigma \in \Sigma$.

A diferença existente entre os autômatos e os geradores é o fato de que, no caso dos autômatos, a função de transição não pode ser parcial (definida apenas para um subconjunto de eventos para cada estado do gerador).

Igualmente aos autômatos, tem-se a função de transição estendida, sua linguagem e sua linguagem marcada, como mostram as seguintes definições:

Definição 2.13 *Dado um gerador $G = (Q, \Sigma, \delta, q_0, Q_m)$, associa-se a cada estado $q \in Q$ o conjunto de eventos definidos $\Sigma(q)$, dado por:*

$$\Sigma(q) = \{\sigma | \sigma \in \Sigma \wedge \delta(\sigma, q)!\} \quad (2.7)$$

com $\delta(\sigma, q)!$ identificando que δ é definida para o par (σ, q) .

Definição 2.14 *Seja um gerador $G = (Q, \Sigma, \delta, q_0, Q_m)$. Sua função de transição estendida, denotada por δ^* , é uma função*

$$\delta^* : \Sigma^* \times Q \rightarrow Q$$

tal que:

$$\begin{aligned} \delta^*(\epsilon, q) &= q \\ \delta^*(s\sigma, q) &= \delta(\sigma, \delta^*(s, q)), \quad \text{para } q \in Q \text{ e } s \in \Sigma^* \text{ sempre que } q' = \delta^*(s, q) \\ &\text{e } \delta^*(\sigma, q') \text{ estiverem ambos definidos.} \end{aligned} \quad (2.8)$$

Definição 2.15 *Seja um gerador $G = (Q, \Sigma, \delta, q_0, Q_m)$. Sua linguagem gerada $L(G)$ é:*

$$L(G) = \{s | s \in \Sigma^* \wedge \delta(s, q_0)!\}. \quad (2.9)$$

A função de transição estendida pode ser denotada por δ , ao invés de δ^* , igualmente aos autômatos.

Também é observado que a linguagem gerada $L(G)$ pelo gerador é *prefixo-fechada*, isto é,

$$\forall G = (Q, \Sigma, \delta, q_0, Q_m), \quad (2.10)$$

$$L(G) = \overline{L(G)}$$

Analogamente aos autômatos, a linguagem marcada do gerador é definida como a seguir:

Definição 2.16 *Dado um gerador $G = (Q, \Sigma, \delta, q_0, Q_m)$, a linguagem marcada de G , denotada por $L_m(G)$, é:*

$$L_m(G) = \{s | s \in \Sigma^* \wedge \delta(s, q_0) \in Q_m\}. \quad (2.11)$$

O comportamento de um SED é caracterizado pela ocorrência de eventos, isto é, um SED gera palavras de comprimento crescente, à medida que evolui. De acordo com o alfabeto gerado pelo SED, podem-se encontrar seqüências de símbolos, ou palavras que não representam seqüências de eventos fisicamente possíveis. Assim, a linguagem gerada pelo sistema é um subconjunto próprio de Σ^* e inclui, para cada palavra, todos os seus prefixos. A linguagem *prefixo-fechada* representa o comportamento lógico de um SED, em que não ocorrem eventos simultâneos.

Definição 2.17 *A linguagem prefixo-fechada que representa o comportamento lógico de um SED é denominada de linguagem gerada do sistema.*

e,

Definição 2.18 *A linguagem que representa o conjunto de todas as tarefas que um SED é capaz de executar, é denominada linguagem marcada do sistema.*

Considerando que L^{sed} seja a linguagem gerada por um sistema e L_m^{sed} sua linguagem marcada, de acordo com as Definições 2.17 e 2.18, tem-se que a linguagem marcada L_m^{sed} contém as palavras geradas pelo SED, que também gera todos os seus prefixos. Ou seja, um SED produz as palavras contidas em $\overline{L_m^{sed}}$. Desta forma, a linguagem marcada do SED não é necessariamente *prefixo-fechada*. Assim, vê-se que

$$L_m^{sed} \subseteq \overline{L_m^{sed}} \subseteq L^{sed} = \overline{L^{sed}}.$$

Dessa forma, um gerador pode representar um SED, devido à sua função de transição restrita, que é definida apenas para alguns pares (*evento, estado*) do conjunto $\Sigma \times Q$. Isto

é, um SED com linguagem gerada L^{sed} e linguagem marcada L_m^{sed} , é representado por um gerador, de tal forma que

$$L(G) = L^{sed}$$

e

$$L_m(G) = L_m^{sed}$$

Devido à possibilidade dos geradores poderem representar os SEDs, algumas definições são necessárias para sua análise. Essas definições são de fundamental importância para o conceito de estrutura.

Definição 2.19 A componente acessível de um gerador $G = (Q, \Sigma, \delta, q_0, Q_m)$, denotada por $Ac(G)$ é:

$$\begin{aligned} Ac(G) &= (Q_{ac}, \Sigma, \delta_{ac}, q_0, Q_{ac,m}), \quad \text{onde} \\ Q_{ac} &= \{q | q \in Q \wedge \exists s \in \Sigma^* \wedge \delta(s, q_0) = q\}; \\ Q_{ac,m} &= Q_{ac} \cap Q_m; \\ \delta_{ac} &= \delta | (\Sigma \times Q_{ac}). \end{aligned}$$

Aqui, δ_{ac} é a função δ restrita ao domínio $\Sigma \times Q_{ac}$. Assim, um gerador G é dito acessível, na condição única de $G = Ac(G)$. A componente acessível de um gerador contém apenas os estados que são alcançados a partir do estado inicial, embora que nenhum seja marcado.

Definição 2.20 A componente coacessível de um gerador $G = (Q, \Sigma, \delta, q_0, Q_m)$, denotada por $Co(G)$ é:

$$\begin{aligned} Co(G) &= (Q_{co}, \Sigma, \delta_{co}, q_0, Q_{co,m}), \quad \text{onde} \\ Q_{co} &= \{q | q \in Q \wedge \exists s \in \Sigma^* \wedge \delta(s, q) = q_m \wedge q_m \in Q_m\}; \\ Q_{co,m} &= Q_{co} \cap Q_m; \\ \delta_{co} &= \delta | (\Sigma \times Q_{co}). \end{aligned}$$

A componente coacessível de um gerador apresenta apenas os estados em que, a partir deles, é possível atingir um estado marcado.

Definição 2.21 Um gerador $G = (Q, \Sigma, \delta, q_0, Q_m)$ é coacessível se e somente se, toda palavra de $L(G)$ for um prefixo de uma palavra de $L_m(G)$, isto é:

$$L(G) \subseteq \overline{L_m(G)}. \quad (2.12)$$

Nessa definição vê-se que em um gerador coacessível, existe pelo menos uma seqüência de eventos que o leva a um estado marcado.

Os geradores que são, ao mesmo tempo, acessíveis e coacessíveis, são denominados ajustados ou *trim*.

Exemplo 2.8 Na Figura 2.3(a), é apresentado um gerador G . Este gerador não é acessível devido ao fato de que o estado 6 não é alcançado em nenhuma seqüência a partir do estado 0. Na Figura 2.3(b), encontra-se o mesmo gerador sem o estado 6 e seus dois arcos retirados, o que o torna um gerador acessível, em que todos os estados são alcançados. Por outro lado, para encontrar o gerador coacessível, é necessário identificar os estados de G que não são coacessíveis para o estado marcado 2. Estes estados são: 3, 4 e 5. Retirando estes estados e as transições a eles ligadas, o gerador torna-se coacessível. Note que o estado 6 não é retirado, desde que dele o estado marcado 2 é alcançado. Este gerador coacessível é visto na Figura 2.3(c). Finalmente, o gerador trim, está apresentado na Figura 2.3(d), onde é visto que, independentemente da ordem em que as operações de acessibilidade e de coacessibilidade são tomadas, o resultado final não é afetado. Deve-se observar que os geradores das Figuras 2.3(b), 2.3(c) e 2.3(d) são, respectivamente, a componente acessível, a componente coacessível e a componente trim do gerador da Figura 2.3(a).

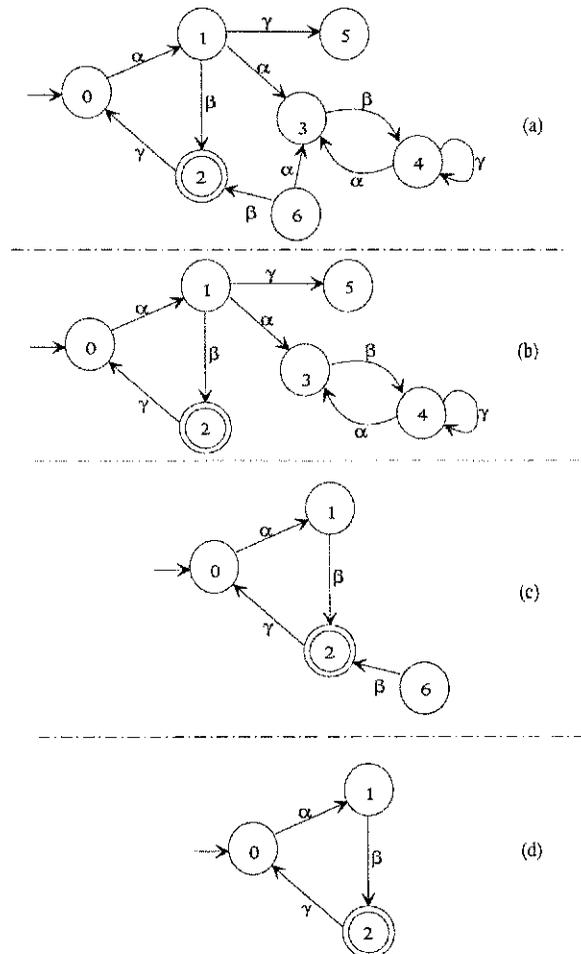


Figure 2.3: (a) Gerador e geradores (b) acessível, (c) coacessível e (d) trim.

2.1.3 Composição de Autômatos

Uma das alternativas de modelagem de SEDs requer a decomposição dos sistemas em sub-sistemas e, para cada subsistema é definido um autômato que representa seu comportamento dinâmico. Entretanto, para estudar o sistema como um todo é necessário remontar a decomposição feita para simplificar o estudo. Neste caso, é necessário compor os autômatos que representam os sub-sistemas para obter o autômato do sistema. A composição de autômatos, constrói um autômato que gera uma linguagem igual a interseção das linguagens dos autômatos envolvidos na composição.

Definição 2.22 *Sejam*

$$G_1 = (Q_1, \Sigma_1, \delta_1, q_{10}, Q_{m_1})$$

e

$$G_2 = (Q_2, \Sigma_2, \delta_2, q_{20}, Q_{m_2})$$

dois autômatos. O autômato G_3 resultante da composição síncrona é definido por

$$G_3 := (Q_3, \Sigma_3, \delta, q_{30}, Q_{m_3})$$

com $Q_3 = Q_1 \times Q_2$, $\Sigma_3 = \Sigma_1 \cup \Sigma_2$, $q_{30} = (q_{10}, q_{20})$, $Q_{m_3} = Q_{m_1} \times Q_{m_2}$ e

$$\delta_3(\sigma, (q_{1_i}, q_{2_i})) = \begin{cases} (\delta_1(\sigma, q_{1_i}), \delta_2(\sigma, q_{2_i})) = (q_{1_{i'}} , q_{2_{i'}}) & \text{se } \exists \sigma, \delta_1(\sigma, q_{1_i}) = q_{1_{i'}} \text{ e } \delta_2(\sigma, q_{2_i}) = q_{2_{i'}} \\ (\delta_1(\sigma, q_{1_i}), q_{2_i}) = (q_{1_{i'}} , q_{2_i}) & \text{se } \exists \sigma, \delta_1(\sigma, q_{1_i}) = q_{1_{i'}} \text{ apenas em } G_1 \\ (q_{1_i}, \delta_2(\sigma, q_{2_i})) = (q_{1_i}, q_{2_{i'}}) & \text{se } \exists \sigma, \delta_2(\sigma, q_{2_i}) = q_{2_{i'}} \text{ apenas em } G_2 \\ \text{indefinido} & \text{caso contrário.} \end{cases} \quad (2.13)$$

Exemplo 2.9 A composição síncrona dos autômatos G_1 e G_2 , apresentados da Figura 2.4(a) e 2.4(b), respectivamente, onde os elementos de Σ_1 são os mesmos elementos de Σ_2 , está apresentado na Figura 2.4(c). Pode-se ver que nos dois autômatos G_1 e G_2 , só se encontra um único arco (α) que sai do estado x para o estado x , em G_1 e que, equivalentemente, sai do estado 0 para o estado 1 no autômato G_2 . Por outro lado, quando o autômato G_2 encontra-se no estado 1, só há um arco saindo do mesmo, que é equivalente no autômato G_1 , que também é o arco α . Também, o estado $(x, 0)$ do autômato construído pela composição síncrona de G_1 e G_2 , não é marcado porque o estado inicial do autômato G_2 , não é. Porém, o estado $(x, 1)$ do autômato construído pela composição síncrona de G_1 e G_2 é marcado, pois em ambos os autômatos, a função de transição α , leva de um estado marcado para outro estado marcado. Deve-se observar que a Figura 2.4(c) não apresenta os outros estados da composição, desde que a função de transição é indefinida (ver equação (2.13)). Naturalmente, a última condição da equação (2.13) pode ser vista como a aplicação dos operadores de acessibilidade e coacessibilidade sobre o autômato G_3 com todos os estados $q_{G_3} = (q_{G_1}, q_{G_2})$.

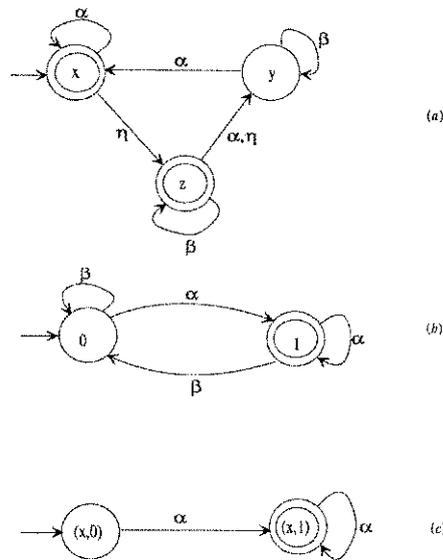


Figure 2.4: (a) Autômato G_1 , (b) Autômato G_2 e (c) Composição síncrona $G = G_1 || G_2$.

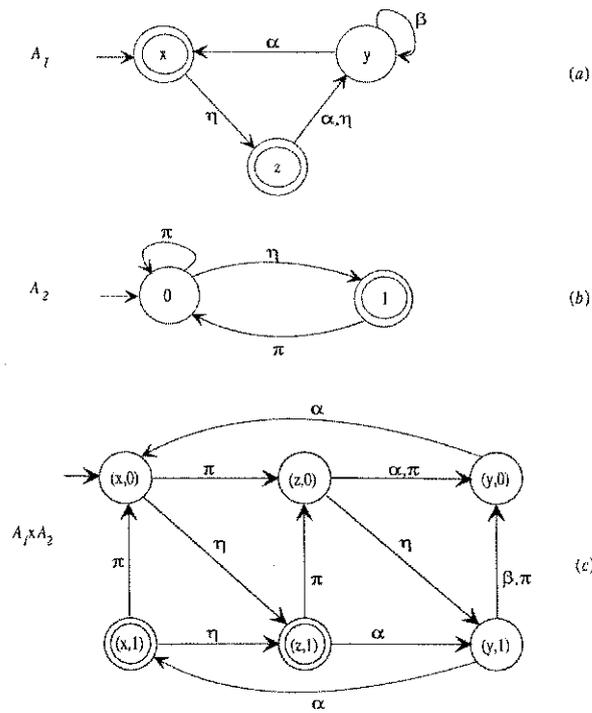


Figure 2.5: (a) Autômato G_1 , (b) Autômato G_2 e (c) Composição síncrona $G_3 = G_1 || G_2$.

Exemplo 2.10 A composição síncrona entre os autômatos G_1 e G_2 , apresentados da Figura 2.5(a) e 2.5(b), respectivamente, onde somente o elemento $\eta \in \Sigma_1 \cap \Sigma_2$, está apresentado na Figura 2.5(c).

A composição síncrona de autômatos é utilizada na TCS para definir a linguagem do SED acoplado em malha fechada ao supervisor, geralmente denotada por \mathcal{S}/G , como será vista adiante.

2.2 A síntese do supervisor

Na TCS, os SEDs são representados por geradores, onde sua linguagem gerada é $L(G)$ e a linguagem marcada é $L_m(G)$. Nela, considera-se a existência de um agente externo para assegurar que o sistema realize uma tarefa determinada. Nesta teoria, o alfabeto de eventos Σ do gerador é dividido em duas categorias distintas: os eventos controláveis e os eventos não controláveis, representados pelos conjuntos Σ_c e Σ_{uc} , respectivamente. Assim, tem-se que o alfabeto Σ é a união de Σ_c e Σ_{uc} , ou seja

$$\Sigma = \Sigma_c \cup \Sigma_{uc}.$$

Estes dois alfabetos definidos dentro do alfabeto Σ , não têm eventos comuns, isto é, sua interseção é o conjunto vazio, ou

$$\Sigma_c \cap \Sigma_{uc} = \emptyset.$$

Os eventos controláveis são aqueles que podem ser habilitados ou desabilitados em qualquer momento, enquanto que os eventos não controláveis não sofrem ação de controle, como é o caso da quebra de uma máquina, ou término de processamento de uma peça. Um evento controlável pode ser exemplificado pelo início da operação da máquina.

Os eventos controláveis são habilitados ou desabilitados pela entrada de controle que é definida como a seguir:

Definição 2.23 Dado um gerador $G = (\Sigma, Q, \delta, q_0, Q_m)$, cujo alfabeto é particionado em $\Sigma = \Sigma_c \cup \Sigma_{uc}$, o conjunto de entradas de controle associado a G é definido por:

$$\Gamma = \{\gamma \mid \Sigma_{uc} \subseteq \gamma \subseteq \Sigma\}.$$

As entradas de controle definem quais os eventos que devem estar habilitados num determinado estado. Assim, em um estado q do gerador G , onde há eventos controláveis e eventos não controláveis habilitados, uma entrada de controle γ aplicada nesse estado define quais os únicos eventos que podem ocorrer. Como os eventos não controláveis não sofrem ação de controle, todos os eventos não controláveis definidos neste estado estão, por definição, sempre habilitados. Apenas os eventos controláveis definidos em γ podem ocorrer (os demais eventos são inibidos).

Um gerador controlado é definido do seguinte modo:

Definição 2.24 Dado $\Gamma \subseteq 2^\Sigma$ como sendo o conjunto de entradas de controle, um gerador controlado G_c é um par (G, Γ) onde G é um gerador com alfabeto Σ , particionado em eventos controláveis Σ_c e eventos não controláveis Σ_{uc} , equipado com um conjunto de entradas de controle Γ .

Denomina-se *planta*, o modelo do sistema a ser controlado, semelhantemente à teoria clássica de controle. O comportamento do sistema na ausência de qualquer ação de controle é definida como *linguagem da planta*, a qual representa o comportamento do sistema.

Exemplo 2.11 Para o gerador apresentado na Figura 2.6(a), a linguagem é

$$L(G) = ((\alpha\beta + \eta)\nu + \epsilon)^*.$$

Considerando que $\Sigma_{uc} = \{\alpha, \eta\}$ e $\Sigma_c = \{\beta, \nu\}$, com a permanente desabilitação de β definida pela aplicação da entrada de controle $\gamma = \{\alpha, \eta, \nu\}$, a linguagem torna-se

$$L(G) = \alpha + (\eta\nu + \epsilon)^*,$$

que é representada pelo gerador visto na Figura 2.6(b).

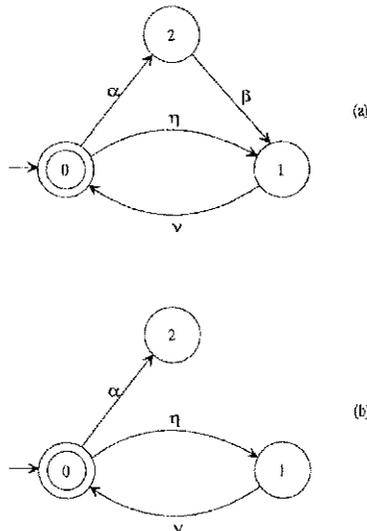


Figure 2.6: (a) Gerador com todos os eventos habilitados e (b) Gerador com o evento β inibido.

Este mecanismo de controle define um chaveamento nas entradas de controle. Dessa forma, determina que a seqüência definida na especificação de comportamento, se possível, seja seguida. Isso remete às definições de supervisor e suas condições de existência.

2.3 Supervisores e condições de existência

Fundamentada a idéia de geradores controlados, interessa chavear a entrada de controle em resposta à cadeia de eventos previamente gerada pelo gerador G . Este chaveamento é feito pelo supervisor, que é o agente externo que determina a ação de controle.

Um supervisor é definido formalmente como:

Definição 2.25 Um supervisor para um gerador controlado $G_c = (G, \Gamma)$ é um par $\mathcal{S} = (S, \Theta)$, composto de um gerador $S = (\Sigma, X, \xi, x_0, X_m)$ e de um mapa de controle Θ , em que:

- Σ é o mesmo alfabeto de G ;
- X é um conjunto de estados;
- $\xi : \Sigma^* \times X \rightarrow X$ é uma função de transição parcial estendida;
- $x_0 \in X$ é o estado inicial;
- $X_m \subseteq X$ é o conjunto de estados marcados;
- $\Theta : X \rightarrow \Gamma$ é uma função que associa a cada estado $x \in X$ uma entrada de controle $\gamma \in \Gamma$.

Na Figura 2.7, é mostrado como é representado o sistema composto pelo gerador controlado G_c supervisionado pelo supervisor \mathcal{S} , em malha fechada. Pode-se ver que o gerador controlado recebe a ação de controle e gera os eventos que são observados pelo supervisor. Essa composição determina que o gerador siga a linguagem do supervisor.

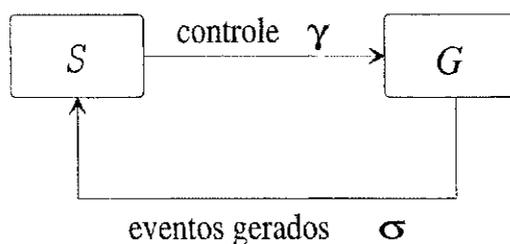


Figure 2.7: Supervisão de um SED.

De acordo com a definição de supervisor, tem-se que a ação de controle modifica a linguagem gerada do sistema sob supervisão, pois pode inibir seqüências de eventos que antes podiam ocorrer, como visto anteriormente. Logo, sua linguagem, pode ser assim definida:

Definição 2.26 Dados um gerador controlado G_c e um supervisor \mathcal{S} , a linguagem gerada pelo sistema supervisionado, denotada por $L(\mathcal{S}/G)$, é tal que

$$e \in L(\mathcal{S}/G) \quad e$$

$$s\sigma \in L(\mathcal{S}/G) \text{ se e somente se } s \in L(\mathcal{S}/G) \wedge s\sigma \in L(G) \wedge \sigma \in \Theta(\xi(s, x_0)),$$

onde Θ é o mapa de controle que associa a cada estado $x \in X$ uma entrada de controle $\gamma \in \Gamma$, que é aplicada a G .

Observação: A composição síncrona entre o supervisor e o gerador define a linguagem do sistema supervisionado. Esta linguagem representa a trajetória que o sistema deve seguir. Deve-se observar que os eventos habilitados em cada estado do supervisor são os únicos eventos que podem ocorrer no estado correspondente do gerador. Os outros eventos possíveis de ocorrerem no respectivo estado do gerador são inibidos, o que é determinado pela entrada de controle do supervisor. Esta condição pode ser vista na composição síncrona do supervisor com o gerador, observando os pares $q_{S/G} = (q_S, q_G)$, onde $q_{S/G}$ é o estado q da composição síncrona S/G , q_S é o estado q do supervisor S e q_G é o estado q do gerador G . Isto é visto na Definição 2.26, em que somente os eventos que estão habilitados em cada estado, são os que ocorrem sob supervisão.

Dada uma palavra s que pertença a $L(S/G)$, a mesma também pertence a $L(G)$, o que determina que a linguagem do sistema supervisionado satisfaz

$$L(S/G) \subseteq L(G). \quad (2.14)$$

Assim, $L(S/G)$ é uma linguagem prefixo-fechada, visto que uma palavra $s\sigma \in L(S/G)$ somente se $s \in L(S/G)$.

Define-se a linguagem controlada do sistema supervisionado por

$$L_c(S/G) = L(S/G) \cap L_m(G), \quad (2.15)$$

ou seja, $L_c(S/G)$ é a parte da linguagem marcada original sob ação de controle que representa as tarefas que são completadas sob supervisão. Logo, isto implica nas condições:

$$\begin{aligned} L_c(S/G) &\subseteq L(S/G) && \text{e} \\ L_c(S/G) &\subseteq L_m(G) \end{aligned}$$

Dessa forma, a linguagem marcada do sistema é

$$L_m(S/G) = L_c(S/G) \cap L_m(S) \quad (2.16)$$

Assim, vê-se que

$$L_m(S/G) \subseteq L_c(S/G) \subseteq L(S/G) \subseteq L(G),$$

ou seja, a linguagem $L(S/G)$, que é gerada pelo sistema composto pelo supervisor e pelo gerador controlado pode ser interpretada como o conjunto de todas as possíveis seqüências finitas de eventos que têm possibilidades de ocorrer no sistema.

2.4 Supervisores próprios

É preciso garantir que os eventos no supervisor \mathcal{S} só devem ocorrer, quando eles também ocorrerem em G_c e estiverem habilitados por Θ .

Definição 2.27 *Um supervisor \mathcal{S} é dito ser completo, em relação a um gerador G_c , quando o seguinte é verdadeiro: para todo $s \in \Sigma^*$ e $\sigma \in \Sigma$ as três condições*

$$\begin{aligned} s &\in L(\mathcal{S}/G), \\ s\sigma &\in L(G) \quad e \\ \sigma &\in \Theta(\xi(s, x_0)), \end{aligned}$$

juntas implicam em

$$\xi(s, x_0)!, \text{ isto é, } \xi(s, x_0) \text{ é definido.}$$

Esta é a condição necessária para considerar o supervisor \mathcal{S} como completo em relação a um gerador controlado G_c . Logo, se s é uma palavra que pode ocorrer no sistema supervisionado e o evento σ é uma continuação fisicamente possível desta palavra, se σ está habilitado, então a palavra $s\sigma$ deve estar definida na função de transição do supervisor.

Torna-se necessário estabelecer duas restrições a serem satisfeitas pelas linguagens $L(\mathcal{S}/G)$, $L_c(\mathcal{S}/G)$ e $L_m(\mathcal{S}/G)$, para controlar um SED de maneira satisfatória.

Definição 2.28 *Um supervisor \mathcal{S} é dito não bloqueável se e somente se*

$$\overline{L_c(\mathcal{S}/G)} = L(\mathcal{S}/G) \quad (2.17)$$

Definição 2.29 *Um supervisor \mathcal{S} é dito não rejeitável se e somente se*

$$\overline{L_m(\mathcal{S}/G)} = \overline{L_c(\mathcal{S}/G)}. \quad (2.18)$$

Dessas duas definições, vê-se que um supervisor é bloqueável se existir pelo menos uma palavra fisicamente possível em $L(\mathcal{S}/G)$ que não é prefixo de qualquer palavra em $L_c(\mathcal{S}/G)$, isto é, o sistema nunca pode completar uma tarefa especificada. Por outro lado, um supervisor é rejeitável caso exista pelo menos uma palavra em $\overline{L_c(\mathcal{S}/G)}$ representando uma palavra completada, contudo que não pertence à linguagem marcada $L_m(\mathcal{S}/G)$. Neste último caso, é possível atingir um estado em que nenhuma tarefa seja completada.

Para que a síntese do supervisor seja completa, o supervisor não deve apresentar nenhum desses problemas. Isto leva a definição de supervisor próprio.

Definição 2.30 *Um supervisor completo, isto é, não bloqueável e não rejeitável é dito supervisor próprio se*

$$\overline{L_m(\mathcal{S}/G)} = \overline{L_c(\mathcal{S}/G)} = L(\mathcal{S}/G). \quad (2.19)$$

Visto isso, chega-se ao problema principal da TCS, que está definido em determinar mudanças no comportamento de um SED G . As linguagens apresentadas anteriormente permitem a formulação de problemas abstratos de síntese de supervisores. De um modo geral, um problema desse tipo supõe que se represente por linguagens o comportamento fisicamente possível do sistema e o comportamento desejado sob supervisão. O objetivo é construir um supervisor para a *planta* tal que o comportamento do sistema em malha fechada se limite ao comportamento desejado. Para tanto, definem-se os seguintes conceitos:

Definição 2.31 *Sejam duas linguagens $K, L \subseteq \Sigma^*$. K é dita fechada em relação a L , ou L -fechada, se e somente se*

$$K = \overline{K} \cap L \quad (2.20)$$

Definição 2.32 *Dadas duas linguagens $K, L \subseteq \Sigma^*$ e um alfabeto $\Sigma = \Sigma_c \cup \Sigma_{uc}$, diz-se que K é L -controlável se e somente se*

$$\overline{K} \Sigma_{uc} \cap L \subseteq \overline{K}. \quad (2.21)$$

Essas definições são os conceitos conhecidos por *fechamento* e *controlabilidade*, respectivamente. A linguagem K é definida como a linguagem da especificação de comportamento, ou o que se deseja que o SED realize. A linguagem L é a linguagem gerada pelo SED.

Resta agora determinar as condições de existência do supervisor para a realização de uma tarefa definida. É dada, então, a seguinte proposição:

Proposição 2.1 *Seja S um supervisor completo para G_c . Então $L(S/G)$ é prefixo-fechada e $L(G)$ -controlável.*

Demonstração:

A demonstração desse Teorema é apresentada em Ziller [Zil93]. ♦

Também é necessário estabelecer as condições de existência de supervisores para os problemas formulados em termos de linguagens geradas e marcadas. Para isto, apresentam-se os teoremas a seguir:

Teorema 2.1 *Dados um gerador G tal que $L(G)$ represente seu comportamento fisicamente possível e uma linguagem especificada $K \subseteq L(G)$, existe um supervisor completo S tal que $L(S/G) = K$ se e somente se K for prefixo-fechada e $L(G)$ -controlável.*

Demonstração:

A demonstração desse Teorema é apresentada em Ziller [Zil93]. ♦

Teorema 2.2 *Dados um gerador G tal que $L_m(G)$ represente as tarefas que podem ser completadas pelo sistema na ausência de qualquer ação de controle e uma linguagem especificada $K \subseteq L_m(G)$ então*

1. Existe um supervisor não bloqueável \mathcal{S} tal que $L_c(\mathcal{S}/G) = K$ se e somente se K for $L(G)$ -fechada e $L(G)$ -controlável;
2. O supervisor \mathcal{S} será próprio somente se o gerador $S = (\Sigma, X, \xi, x_0, X_m)$ for tal que $X_m = X$.

Demonstração:

A demonstração desse Teorema é apresentada em Ziller [Zil93]. ◆

Para problemas formulados em termos de linguagens marcadas, se K satisfizer as condições do Teorema 2.2, então o supervisor será tal que $L(\mathcal{S}/G) = \overline{K}$, visto que nesse caso,

$$L_c(\mathcal{S}/G) = \overline{K} \cap L_m(G) = K. \quad (2.22)$$

Estes resultados só podem ser empregados quando a linguagem especificada K satisfaz as condições exigidas. Quando a linguagem K não satisfaz as condições exigidas à existência do supervisor, ou seja, a linguagem especificada K não é nem $L_m(G)$ -fechada, nem $L(G)$ -controlável, é necessário encontrar uma sublinguagem $K^\dagger \subseteq K$, que satisfaça todas essas condições. Esta sublinguagem é conhecida por *Suprema Sublinguagem Controlável* K^\dagger , ou $\text{sup}C(L)$, que soluciona o problema do supervisor, restritivamente.

Sendo assim, se K^\dagger solucionar de forma satisfatória o dado problema, isto é, se $K \supseteq L_A$, onde L_A é uma linguagem que representa o comportamento mais restrito que pode ser tolerado, K^\dagger pode ser utilizada em substituição à linguagem anteriormente especificada. A solução para este problema é um supervisor que implementa K^\dagger . Para o caso dos geradores de estado finito, K^\dagger é sempre computável.

Determina-se, então, o seguinte: seja $K \subseteq \Sigma^*$ uma linguagem, e seja $C(K)$ a família das linguagens controláveis de K . Então, $C(K)$ é sempre não vazia pois a linguagem vazia é controlável.

Um importante resultado em relação à controlabilidade das linguagens é que a família $C(K)$ é fechada em relação à união de linguagens, ou seja, existe uma única linguagem controlável máxima K^\dagger tal que $K^\dagger \subseteq K$. Assim, nota-se que K^\dagger pode ser a linguagem vazia.

Este problema pode ser enunciado da seguinte maneira: *Dados um gerador G , uma linguagem-alvo marcada $E \subseteq \Sigma^*$ e uma linguagem mínima admissível $L_A \subseteq E$, encontrar um supervisor próprio \mathcal{S} tal que*

$$L_A \subseteq L_c(\mathcal{S}/G) \subseteq E. \quad (2.23)$$

Quando E é $L_m(G)$ -fechada e $L(G)$ -controlável, a existência de um supervisor tal que $L_c(\mathcal{S}/G) = E$ é garantida, o que significa que o problema tem uma solução não restritiva, como visto anteriormente. Nos casos em que E não satisfaz estas condições, é possível obter uma solução minimamente restritiva, como é visto a seguir.

Considere que o conjunto de todas as sublinguagens $L(G)$ -controláveis de E , é dado por:

$$C(E) = \{K' \mid K' \subseteq E \wedge K' \Sigma_{uc} \cap L(G) = K'\} \quad (2.24)$$

e que o conjunto de todas as sublinguagens $L_m(G)$ -fechadas de E , é dado por:

$$F(E) = \{K' | K' \subseteq E \wedge \overline{K'} \cap L_m(G) = K'\}. \quad (2.25)$$

Para esclarecer a terminologia empregada adiante, considere a seguinte definição:

Definição 2.33 *Dados um conjunto W qualquer e uma operação Z sobre seus elementos, W é dito fechado sob a operação Z se e somente se para todo par (w_1, w_2)*

$$w_1, w_2 \in W \Rightarrow w_1 Z w_2 \in W.$$

Teorema 2.3 *Seja*

$$H(K) = \{K' | K' \subseteq K \wedge [\forall K'_1, K'_2 \in H(K)] K'_1 \cup K'_2 \in H(K)\}$$

um conjunto não vazio de subconjuntos de K , fechado sob a operação de união. Então existe em $H(K)$ o elemento supremo

$$\sup H(K) = \bigcup \{K : K \in H(K)\}.$$

Demonstração:

A demonstração desse Teorema é apresentada em Ziller [Zil93]. ♦

Considerando que dadas duas linguagens quaisquer $L_1, L_2 \subseteq \Sigma^*$, as operações de união de conjuntos e de prefixo-fechamento comutam, isto é, $\overline{L_1 \cup L_2} = \overline{L_1} \cup \overline{L_2}$. Estendendo este raciocínio a um número arbitrário de linguagens sobre Σ^* , então, a seguinte proposição é apresentada:

Proposição 2.2 *$C(E)$ e $F(E)$ são não vazios e fechados sob a operação de união.*

Demonstração:

A demonstração desse Teorema é apresentada em Ziller [Zil93]. ♦

Corolário 2.1 *Existem em $C(E)$ e $F(E)$, respectivamente, os elementos supremos*

$$\begin{aligned} \sup C(E) &= \bigcup \{K | K \in C(E)\} & e \\ \sup F(E) &= \bigcup \{K | K \in F(E)\}. \end{aligned}$$

Demonstração:

Demonstrado pelo Teorema 2.3. ♦

Os elementos $\sup C(E)$ e $\sup F(E)$ são denominados, respectivamente, de *Máxima Sublinguagem $L(G)$ -controlável* e *Máxima Sublinguagem $L_m(G)$ -fechada* de E .

Além do mais, a seguinte proposição é também válida:

Proposição 2.3 *O conjunto $C(E) \cap F(E)$ é não vazio e fechado sob a operação de união.*

Demonstração:

A demonstração desse Teorema é apresentada em Ziller [Zil93]. \blacklozenge

Considerando o conjunto $CF(E) = C(E) \cap F(E)$ como sendo o conjunto de todas as linguagens $L(G)$ -controlável e $L_m(G)$ -fechada de E , então,

Corolário 2.2 Existe em $C(E) \cap F(E)$ o elemento supremo

$$\sup CF(E) = \bigcup \{K \mid K \in C(E) \cap F(E)\}.$$

Demonstração:

Demonstrado pelo Teorema 2.3. \blacklozenge

Denomina-se o elemento $\sup CF(E)$ de *Máxima Sublinguagem $L(G)$ -controlável e $L_m(G)$ -fechada* de E . Assim, tem-se o seguinte:

Teorema 2.4 O problema de controle supervisório tem solução se e somente se

$$\sup CF(E) \supseteq L_A, \quad (2.26)$$

sendo $\sup CF(E)$ a solução minimamente restritiva.

Demonstração:

Demonstrado pelo Corolário 2.2. \blacklozenge

Considerando o caso particular onde todos os estados do gerador são marcados, tem-se que:

$$L_m(G) = \overline{L_m(G)} = L(G) \quad (2.27)$$

e, conseqüentemente, pela equação 2.14 e pela Definição 2.26, tem-se

$$L_c(S/G) = L(S/G). \quad (2.28)$$

Além disso, tem-se que toda linguagem *prefixo-fechada* $E \subseteq L_m(G)$ é $L_m(G)$ -fechada. Logo, pode-se garantir que, $\forall E, E \subseteq L_m(G)$:

$$\begin{aligned} E &= \sup F(E) \quad \text{e} \\ \sup CF(E) &= \sup C(E). \end{aligned}$$

Com isto, a solução para este problema é conseqüência imediata do Teorema 2.2, dado que $L(S/G) \subseteq K$. Assim, o problema de controle supervisório tem a seguinte versão para linguagens geradas: *Dadas uma planta G tal que $L_m(G) = L(G)$, uma linguagem prefixo fechada $E \subseteq \Sigma^*$ e uma linguagem mínima admissível $L_A \subseteq E$, encontrar um supervisor próprio S tal que*

$$L_A \subseteq L(S/G) \subseteq E. \quad (2.29)$$

O Teorema a seguir fornece a solução para o problema de controle supervisório para uma linguagem de especificação $E \subseteq \Sigma^*$:

Teorema 2.5 *O problema de controle supervisório para linguagens geradas tem solução, se e somente se*

$$\sup C(E) \supseteq L_A, \quad (2.30)$$

sendo $\sup C(E)$ a solução minimamente restritiva.

Por fim, é ressaltado que, caso todos os estados do gerador sejam marcados, tem-se a garantia de que qualquer supervisor obtido será não bloqueável.

2.4.1 Algoritmo para a Construção da $\sup C(L)$

Para o caso em que um gerador de estado finito G é descrito por L (comportamento do SED) e K (comportamento desejado, representado pelo gerador H), existe o seguinte algoritmo para a computação de K^\dagger :

Algoritmo 2.1 Algoritmo para a Construção de K^\dagger ([RW87])

- Dados o gerador G trim e o gerador H , faça:

1. Construir a matriz de transições \mathbf{A} do gerador H , onde

$$\mathbf{A} = [a_{i,j}], a_{i,j} = \begin{cases} \sigma, & \text{se } \exists \sigma \text{ do estado } i \text{ para o estado } j; \\ - & \text{caso contrário;} \end{cases}$$

2. Incluir ao lado direito da matriz de transições \mathbf{A} o vetor coluna que representa

$$\Sigma(H(x)) \cap \Sigma_{uc},$$

em que $H(x)$ representa os eventos habilitados no estado x do gerador H ;

3. Inclua ao lado direito da tabela o vetor coluna $\Sigma(x)$, representando os eventos habilitados no estado x do gerador G ;
4. Para cada estado x_i , em \mathbf{A} que não satisfaz $\Sigma(H(x_i)) \cap \Sigma_{uc} \subset \Sigma(x_i)$, remover a linha da tabela e a coluna da matriz \mathbf{A} referente ao estado x_i ;
5. Encontrar a componente coacessível do gerador resultante;
6. Itere este processo até que todos os estados x_i restantes satisfaçam

$$\Sigma(H(x_i)) \cap \Sigma_{uc} \subset \Sigma(x_i).$$

Exemplo 2.12 *Sejam $\Sigma = \{\alpha_1, \alpha_2, \beta\}$, $\Sigma_{uc} = \{\beta\}$, e em notação de expressões regulares*

$$\begin{aligned} L(G) &= \overline{(\alpha_1\beta^2 + \alpha_2)}\beta^*, \\ L(H) &= \overline{\alpha_1\beta^2 + \alpha_2}\beta^*, \end{aligned}$$

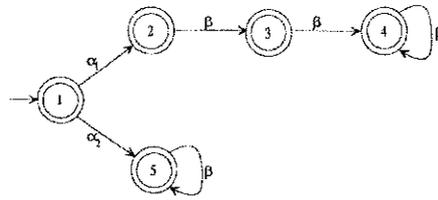


Figure 2.8: Autômato gerador de $L(G) = \overline{(\alpha_1 \beta^2 + \alpha_2)} \beta^*$.

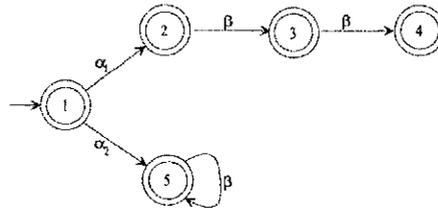


Figure 2.9: Especificação $L(H) = \overline{\alpha_1 \beta^2} + \alpha_2 \beta^*$.

que são a linguagem do gerador trim visto na Figura 2.8 e a especificação de comportamento H desejada para este gerador. Utilizando o algoritmo da $\text{sup } C(L)$ apresentado por Ramadge e Wonham [RW87], inclui-se separadamente à matriz de transição do gerador da especificação de comportamento, visto na Figura 2.9, duas colunas: uma listando $\Sigma(H(x)) \cap \Sigma_{uc}$, outra listando $\Sigma(x)$. Isto define a seguinte tabela:

	1	2	3	4	5	$\Sigma(H(x)) \cap \Sigma_{uc}$	$\Sigma(x)$
$H_0 :$		α_1			α_2		$\alpha_1 \alpha_2$
			β			β	β
				β		β	β
						β	
					β	β	β

Desde que $\Sigma(H(4)) \cap \Sigma_{uc} \not\subseteq \Sigma(4)$, remove-se o estado 4 da tabela e encontra-se que o gerador resultante é trim. O resultado é um gerador para a linguagem $L(H_1)$, apresentado na tabela a seguir:

	1	2	3	5	$\Sigma(H(x)) \cap \Sigma_{uc}$	$\Sigma(x)$
$H_1 :$		α_1		α_2		$\alpha_1 \alpha_2$
			β		β	β
					β	
				β	β	β

onde sua linguagem é $L(H_1) = \overline{\alpha_1 \beta} + \alpha_2 \beta^*$. Iterando esse procedimento é produzida a seguinte

seqüência de tabelas:

	1	2	5	$\Sigma(H(x)) \cap \Sigma_{uc}$	$\Sigma(x)$
H_2 :	1	α_1	α_2		$\alpha_1\alpha_2$
	2			β	
	5		β	β	β

 $L(H_2) = \overline{\alpha_1} + \alpha_2\beta^*$

	1	5	$\Sigma(H(x)) \cap \Sigma_{uc}$	$\Sigma(x)$
H_3 :	1	α_2		$\alpha_1\alpha_2$
	5	β	β	β

 $L(H_3) = \alpha_2\beta^*$

em que, $L(H_3)$ é a $sup C(L)$, cujo supervisor está apresentado na Figura 2.10.

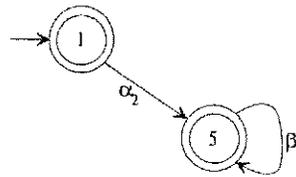


Figure 2.10: Supervisor para a $sup C(L) = L(H_3) = \alpha_2\beta^*$.

Exemplo 2.13 Considere o autômato apresentado na Figura 2.11 e a especificação de comportamento apresentada na Figura 2.12. A linguagem marcada do autômato é

$$L_m(G) = (\alpha\beta)^* + (\alpha\kappa\eta)^*$$

e a linguagem marcada da especificação de comportamento é

$$L(H) = (\alpha + \kappa)^* \beta (\alpha + \kappa)^* \eta.$$

Pode-se ver que $L(H) \not\subseteq L_m(G)$, pois α^* não existe em $L_m(G)$. Dessa forma, para calcular

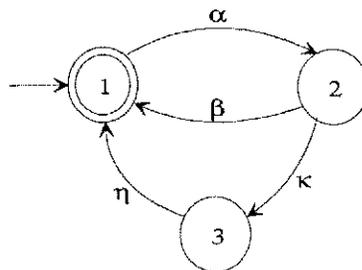


Figure 2.11: Autômato com linguagem $L_m(G) = (\alpha\beta)^* + (\alpha\kappa\eta)^*$.

a $sup C(L)$, constrói-se a composição síncrona $H||G$, a qual é vista na Figura 2.13. Com a linguagem $L(H') = L(H/G)$ dessa composição, utiliza-se o algoritmo para construção da $sup C(L)$, em que inclui-se separadamente à matriz de transição do gerador da especificação

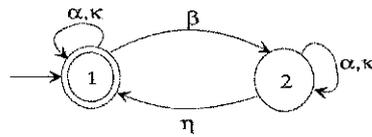


Figure 2.12: Especificação $L(H) = (\alpha + \kappa)^* \beta (\alpha + \kappa)^* \eta$.

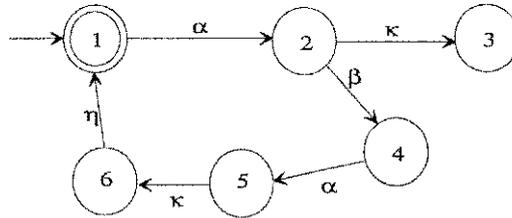


Figure 2.13: Composição síncrona $H||G$.

de comportamento e as duas colunas listando $\Sigma(H'(x)) \cap \Sigma_{uc}$ e $\Sigma(x)$. Isto define a seguinte tabela:

	1	2	3	4	5	6	$\Sigma(H'(x)) \cap \Sigma_{uc}$	$\Sigma(x)$
H'_0 : 1		α						α
2			κ	β				κ, β
3							η	
4					α			α
5						κ		κ
6	η						η	η

Desde que $\Sigma(H'(3)) \cap \Sigma_{uc} \not\subset \Sigma(3)$, remove-se o estado 3 da tabela e encontra-se que o gerador resultante é trim. O resultado é um gerador para a linguagem $L(H'_1)$, apresentado na tabela a seguir:

	1	2	4	5	6	$\Sigma(H'(x)) \cap \Sigma_{uc}$	$\Sigma(x)$
H'_1 : 1		α					α
2			β				κ, β
4				α			α
5					κ		κ
6	η					η	η

onde sua linguagem é $L(H'_1) = \alpha\beta\alpha\kappa\eta$. Como todas as outras linhas satisfazem

$$\Sigma(H'(x)) \cap \Sigma_{uc} \subset \Sigma(x),$$

e H'_1 é coacessível, então, $L(H'_1)$ é a $\text{sup}C(L)$ que é a linguagem do supervisor visto na Figura 2.14.

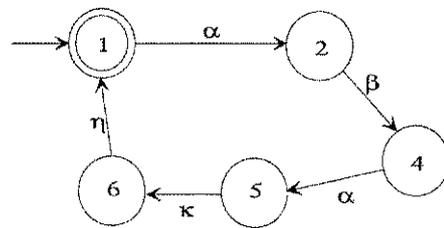


Figure 2.14: Supervisor para $\text{sup } C(L) = L(H_1) = \alpha\beta\alpha\kappa\eta$.

Chapter 3

Síntese de supervisores de SEDs não temporizados

Como visto para a TCS, a síntese do supervisor é feita utilizando um modelo para um SED e uma especificação de comportamento. A partir desta especificação, constrói-se um supervisor que, acoplado ao sistema, realiza-a completa ou restritivamente.

Esta formulação pode ser reescrita por meio da álgebra de dióides e das matrizes de incidência de um gerador que modela um SED. Nesse Capítulo é apresentada essa formulação alternativa para a síntese de supervisores de SEDs, baseada na álgebra de dióides aplicada às linguagens formais. Nessa formulação, o modelo do sistema, a especificação de comportamento e o supervisor sintetizado são representados por matrizes de incidência. Dessa forma, essa abordagem será aplicada à mesma classe de SEDs que trata a TCS, considerando que os sistemas apresentam um número finito de estados.

Para essa formulação, é necessário apresentar alguns conceitos que serão utilizados. Daqui por diante, é utilizado a denominação de autômato para definir gerador, denotado por G .

3.1 Conceitos

3.1.1 Álgebra de dióides

A álgebra de dióides é definida sobre um conjunto D dotado das operações \oplus (adição) e \otimes (multiplicação), caracterizada pela propriedade da idempotência, em que a soma de dois elementos iguais é o próprio elemento [ABG98, Gau96, GG98, GAC⁺90, GTC⁺98, TGG97]. A álgebra de dióides é particularizada de acordo com a definição do conjunto D , que pode ser um conjunto numérico, de símbolos ou de matrizes (numéricas ou de símbolos).

Um dióide é definido como a seguir:

Definição 3.1 *Um conjunto D dotado com duas operações internas \oplus (soma ou adição) e \otimes (produto ou multiplicação) é chamado um dióide, denotado por (D, \oplus, \otimes) , se os seguintes*

axiomas são verificados:

Axioma 1: *Comutatividade de \oplus , $\forall a, b \in D$*

$$a \oplus b = b \oplus a;$$

Axioma 2: *Associatividade de \oplus , $\forall a, b, c \in D$*

$$(a \oplus b) \oplus c = a \oplus (b \oplus c);$$

Axioma 3: *Associatividade de \otimes , $\forall a, b, c \in D$*

$$(a \otimes b) \otimes c = a \otimes (b \otimes c);$$

Axioma 4: *Distributividade de \otimes sobre \oplus , $\forall a, b, c \in D$*

$$\text{à direita: } (a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c);$$

$$\text{à esquerda: } c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b);$$

Axioma 5: *Elemento nulo de \oplus , denotado por ϵ , $\forall a \in D$*

$$a \oplus \epsilon = \epsilon \oplus a = a;$$

Axioma 6: *Absorção pelo elemento nulo de \oplus por \otimes , $\forall a \in D$*

$$a \otimes \epsilon = \epsilon \otimes a = \epsilon;$$

Axioma 7: *Elemento identidade de \otimes , denotado por e , $\forall a \in D$*

$$a \otimes e = e \otimes a = a;$$

Axioma 8: *Idempotência em \oplus , $\forall a \in D$*

$$a \oplus a = a.$$

Observação: Chama-se de dióide comutativo o dióide que satisfaz a comutatividade da multiplicação.

Definição 3.2 *Seja (D, \oplus, \otimes) um dióide. Se a multiplicação \otimes é comutativa, isto é, para $\forall a, b \in D$*

$$a \otimes b = b \otimes a,$$

o dióide (D, \oplus, \otimes) é dito ser comutativo.

Observação: Na álgebra de dióides, geralmente, o sinal da multiplicação \otimes é, de modo geral, omitido. Desse modo, sempre que não houver ambigüidade, utilizar-se-á a seguinte simplificação:

$$\boxed{ab = a \otimes b.}$$

3.1.2 Linguagens formais e dióides

A especificação do conjunto D e das operações \oplus e \otimes particularizam a álgebra de dióides para uma dada aplicação. Para aplicar a álgebra de dióides às linguagens formais, define-se que o conjunto D é $\mathcal{P}(\Sigma^*)$, onde $\mathcal{P}(\Sigma^*)$ é o conjunto de todas as linguagens formadas com símbolos do alfabeto Σ [Gil62]. Assim, um elemento de $\mathcal{P}(\Sigma^*)$ é uma linguagem, e as operações \oplus e \otimes são definidas como segue:

Definição 3.3 *Seja (D, \oplus, \otimes) um dióide. Sendo $D = \mathcal{P}(\Sigma^*)$, os operadores \oplus e \otimes , são definidos como as operações de união e concatenação, respectivamente, de forma que dados dois elementos $L_1, L_2 \in D$,*

$$\begin{aligned} L_1 \otimes L_2 &= \{s_1 s_2 \mid s_1 \in L_1, s_2 \in L_2\} & e \\ L_1 \oplus L_2 &= \{s \mid s \in L_1 \vee s \in L_2\}. \end{aligned}$$

O elemento nulo ‘ ϵ ’ denota a linguagem vazia \emptyset e o elemento identidade ‘ e ’ denota a linguagem $\Sigma^0 = \{\epsilon\}$.

Assim, de acordo com essa definição, duas linguagens L_1 e L_2 formam uma nova linguagem L_3 , através de seu produto na álgebra de dióides, isto é $L_3 = L_1 \otimes L_2$, ou através de sua soma, se $L_1 \neq L_2$.

Exemplo 3.1 *Dadas as linguagens $L_1 = \{a, b\}$ e $L_2 = \{bbc\}$, tem-se que*

$$\begin{aligned} L_3 &= L_1 \oplus L_2 = \{a, b, bbc\} & e \\ L_4 &= L_1 \otimes L_2 = \{abbc, bbcc\}. \end{aligned}$$

Como definido para as linguagens formais [HU69, HU79, Won99], tem-se como consequência da Definição 3.3 que a concatenação de uma linguagem L com ‘ e ’, é

$$L \otimes e = e \otimes L = L$$

e a concatenação de uma linguagem L com ϵ é

$$L \otimes \epsilon = \epsilon \otimes L = \epsilon.$$

Exemplo 3.2 *Seja $L = \{\beta\lambda\}$. Então,*

$$L \otimes e = e \otimes L = \{\beta\lambda e\} = \{e\beta\lambda\} = \{\beta\lambda\} = L$$

e,

$$L \otimes \epsilon = \epsilon \otimes L = \epsilon.$$

A operação \oplus aplicada à linguagens regulares permite representar uma linguagem através de expressões regulares. Esta representação é definida pela substituição da operação $+$ pelo operador \oplus para definir a possibilidade de escolha entre dois ou mais caminhos.

Exemplo 3.3 A linguagem $L_1 = \alpha\beta^*$ somada com a linguagem $L_2 = \alpha$, é

$$L_3 = L_1 \oplus L_2 = \alpha\beta^* \oplus \alpha = \alpha\beta^* + \alpha = \alpha(\beta^* + e)$$

que implica na união de L_1 e L_2 .

Todos os axiomas da Definição 3.1 são válidos quando $D = \mathcal{P}(\Sigma^*)$ [Cos00].

3.1.3 Séries Formais

As séries formais provêm um formalismo para tratar as linguagens formais, sejam elas temporizadas ou não temporizadas. Uma série formal é definida como [BR88, Kli99]:

Definição 3.4 Uma série formal Y sobre um alfabeto Σ com coeficientes em um dióide D é um mapa

$$Y : \Sigma^* \rightarrow D.$$

Para toda palavra $s \in \Sigma^*$, sua imagem $Y(s)$ é denotada por $(y|s)$, com $(y|s) \in D$, representando o coeficiente da palavra s em Y .

Exemplo 3.4 Dado o alfabeto $\Sigma = \{\alpha\}$, tem-se

$$\Sigma^* = \{\varepsilon, \alpha, \alpha\alpha, \alpha\alpha\alpha, \dots\}.$$

Considerando $D = \mathbb{R} \cup \{-\infty\}$, como um exemplo específico pode-se definir

$$\begin{aligned} (y|\varepsilon) &= 1, \\ (y|\alpha) &= 2, \\ (y|\alpha\alpha) &= 7, \\ (y|\alpha\alpha\alpha) &= 4, \\ &\vdots \end{aligned}$$

como os coeficientes das respectivas palavras de Σ^* .

É usual denotar o mapa Y pela série

$$Y = \bigoplus (y|s) s. \quad (3.1)$$

No caso do Exemplo 3.4, o mapa definido pode, portanto, ser reescrito como:

$$Y = 1\varepsilon \oplus 2\alpha \oplus 7\alpha\alpha \oplus 4\alpha\alpha\alpha \oplus \dots$$

Denota-se $D\langle\langle\Sigma\rangle\rangle$ o conjunto das séries formais sobre Σ com coeficientes em D . Para

$$\begin{aligned} Y_1 &: \Sigma^* \rightarrow D, \\ Y_2 &: \Sigma^* \rightarrow D \end{aligned}$$

e para toda palavra $s \in \Sigma^*$, esse conjunto é munido das seguintes operações:

$$(y_1 \oplus y_2|s) = (y_1|s) \oplus (y_2|s) \quad (3.2)$$

$$(y_1 \otimes y_2|s) = \bigoplus_{uv=s} (y_1|u) \otimes (y_2|v), \quad (3.3)$$

onde a soma em $(y_1 \otimes y_2|s)$ é finita. Essas operações são denominadas respectivamente de soma e produto de *Cauchy*.

Exemplo 3.5 Considerando $D = \{\epsilon, e\}$, $\Sigma = \{\alpha, \beta\}$ e

$$Y_1 : \Sigma^* \rightarrow D$$

$$Y_2 : \Sigma^* \rightarrow D$$

tem-se que, se

$$\begin{aligned} (y_1|\alpha) &= e, & (y_1|\alpha\beta) &= e, \\ (y_1|\beta) &= e, & (y_1|\beta\alpha) &= \epsilon, \\ (y_2|\alpha) &= \epsilon, & (y_2|\alpha\beta) &= \epsilon, \\ (y_2|\beta) &= e, & (y_2|\beta\alpha) &= e, \end{aligned}$$

então, verifica-se

$$\begin{aligned} (y_1 \oplus y_2|\alpha) &= (y_1|\alpha) \oplus (y_2|\alpha) = e \oplus \epsilon = e, \\ (y_1 \oplus y_2|\beta) &= (y_1|\beta) \oplus (y_2|\beta) = e \oplus e = e, \\ (y_1 \oplus y_2|\alpha\beta) &= (y_1|\alpha\beta) \oplus (y_2|\alpha\beta) = e \oplus \epsilon = e, \\ & \vdots \end{aligned}$$

e,

$$\begin{aligned} (y_1 \otimes y_2|\alpha\beta) &= (y_1|\alpha) \otimes (y_2|\beta) = e \otimes e = e, \\ (y_1 \otimes y_2|\beta\alpha) &= (y_1|\beta) \otimes (y_2|\alpha) = e \otimes \epsilon = \epsilon. \end{aligned}$$

Observe que a soma de Cauchy representa a união das linguagens e o produto representa a concatenação de linguagens, de acordo com a Definição 3.3. A soma garante que uma palavra pertencente apenas a uma das linguagens, sempre existe nela. O produto garante que só existe uma linguagem se existir o sufixo em uma linguagem e o prefixo na outra linguagem, que devem ser concatenados (a inexistência de um deles é definida pelo valor do coeficiente $(y|s) = \epsilon$).

A equação (3.2) define que para uma mesma palavra $s \in \Sigma^*$ que apresenta coeficientes diferentes, o coeficiente da soma é igual a soma dos coeficientes. Em termos práticos, uma mesma palavra que tem coeficientes diferentes pode ser vista como uma única palavra com um único coeficiente. Similarmente, a equação (3.3) determina que o coeficiente de uma

palavra construída pela concatenação de duas outras palavras, é determinado pela soma da multiplicação dos coeficientes das palavras concatenadas. Isto é, o coeficiente de uma palavra pode ser dividido em partes com as subpalavras que a compõe.

Também define-se para as séries formais a operação *estrela*:

Definição 3.5 A operação estrela para uma série $Y \in D\langle\langle\Sigma\rangle\rangle$ é definida como

$$Y^* = e \oplus Y \oplus Y^2 \oplus Y^3 \oplus \dots$$

onde 'e' é o elemento identidade.

As séries formais permitem descrever linguagens através da equação (3.3). Uma linguagem não temporizada pode ser descrita por meio de uma série formal definindo $D = \mathbb{B} = \{\epsilon, e\}$, que denota o semi-anel binário. Assim, tem-se:

Definição 3.6 Uma linguagem regular $L = \{s, s', \dots\} \subseteq \Sigma^*$, é representada pela série formal

$$Y_L = \bigoplus_{s \in \Sigma^*} (y|s) s \quad (3.4)$$

onde $(y|s) s \in \mathbb{B}\langle\langle\Sigma\rangle\rangle$, e $\mathbb{B}\langle\langle\Sigma\rangle\rangle$ é o semi-anel das séries formais com coeficientes em \mathbb{B} e variáveis não comutativas em Σ , tal que

$$(y|s) = e, \text{ se } s \in L, \quad (3.5)$$

$$(y|s) = \epsilon, \text{ se } s \notin L. \quad (3.6)$$

Exemplo 3.6 Dado o alfabeto $\Sigma = \{\alpha, \beta\}$, a linguagem $L = \{\epsilon, \alpha, \alpha\beta, \beta\alpha, \alpha\alpha, \beta\beta, \beta\alpha\beta\}$ é representada pela série formal

$$Y_L = e\epsilon \oplus e\alpha \oplus e\alpha\beta \oplus e\beta\alpha \oplus e\alpha\alpha \oplus e\beta\beta \oplus e\beta\alpha\beta \oplus \underbrace{\epsilon\alpha\alpha\alpha \oplus \dots \oplus \epsilon\beta\beta\beta \oplus \dots}_{\Sigma^* - L} \quad (3.7)$$

onde $(y|s) = e, \forall s \in L$, e $(y|s) = \epsilon, \forall s \in \Sigma^* - L$. A série formal da equação (3.7) pode ser escrita como

$$Y_L = e\epsilon \oplus e\alpha \oplus e\alpha\beta \oplus e\beta\alpha \oplus e\alpha\alpha \oplus e\beta\beta \oplus e\beta\alpha\beta \quad \text{ou}$$

$$Y_L = \epsilon \oplus \alpha \oplus \alpha\beta \oplus \beta\alpha \oplus \alpha\alpha \oplus \beta\beta \oplus \beta\alpha\beta$$

desde que $\epsilon \otimes L = \epsilon$, $e \otimes L = L$, $\forall L \subseteq \Sigma^*$.

Deve-se observar que uma linguagem L é descrita por uma série formal Y_L . Assim, as operações de união e concatenação de linguagens (ver Definição (3.3)) se apresentam como a soma e a multiplicação de suas respectivas séries formais.

Quando se considera $D = \mathbb{R}_{\max}\langle\langle\Sigma\rangle\rangle$, as séries formais permitem descrever uma linguagem temporizada. Essas linguagens, bem como os autômatos temporizados serão apresentados no Capítulo 4.

3.1.4 Representação matricial de um autômato

Os autômatos finitos podem ser representados por uma matriz denominada matriz de incidência, denotada por \mathbf{A} . Os elementos dessa matriz são símbolos de um alfabeto Σ . Sua definição é a seguinte:

Definição 3.7 *A um autômato $G = (\Sigma, Q, \delta, q_0, Q_m)$ com $|Q| = N$, sua matriz de incidência \mathbf{A} é definida como*

$$\mathbf{A} = [a_{i,j}], a_{i,j} = \begin{cases} \sigma & \text{se } \exists \sigma \text{ do estado } i \text{ para o estado } j; \\ \epsilon & \text{caso contrário,} \end{cases}$$

em que $\sigma \in \Sigma^*$, é uma expressão regular do tipo $\sigma = \sigma^1 + \sigma^2 + \dots + \sigma^n$, $\sigma^1, \sigma^2, \dots, \sigma^n \in \Sigma$, tal que qualquer um evento $\sigma^k \in \Sigma$ ocorrendo, provoca no autômato G a mudança do estado i para o estado j .

Observa-se nessa definição que a linha i define o estado do autômato, e a coluna j determina o próximo estado (linha) do autômato, se $a_{i,j} \neq \epsilon$. Quando em um autômato encontra-se mais de um evento que determina a mudança de um estado i para o estado j , o elemento $a_{i,j}$ é uma expressão regular. O estado inicial e os estados marcados da matriz de incidência são representados, respectivamente, por um vetor linha e um vetor coluna definidos como a seguir.

Definição 3.8 *O estado inicial de um autômato G representado por uma matriz de incidência \mathbf{A} , é definido pelo vetor linha $\theta_{1 \times N}(\mathbf{A})$,*

$$\theta(\mathbf{A}) = [e \ \epsilon \ \dots \ \epsilon],$$

isto é, o primeiro elemento é o elemento identidade 'e' e os demais elementos são dados pelo elemento nulo ϵ . Os estados marcados são definidos pelo vetor coluna $\phi_{N \times 1}(\mathbf{A})$,

$$\phi(\mathbf{A}) = \begin{cases} e & \text{se a linha } i \text{ é marcada;} \\ \epsilon & \text{caso contrário.} \end{cases}$$

Observação: Desde que o vetor de estado inicial só apresenta o primeiro elemento diferente de ϵ , definindo que a linha 1 da matriz é sempre o estado inicial, a representação gráfica de uma matriz de incidência é feita apenas com os elementos do vetor ϕ , externamente, ao seu lado direito. Deve-se observar que a Definição 3.8 indica a linha 1 como o estado inicial para esta abordagem por uma questão de conveniência. Dessa forma, o vetor θ genérico não necessariamente apresenta o estado 1 como o estado inicial. Representa-se por $\theta_j(\mathbf{A})$, o elemento da j -ésima coluna do vetor $\theta(\mathbf{A})$, e por $\phi_i(\mathbf{A})$ o elemento da i -ésima linha do vetor $\phi(\mathbf{A})$.

Para qualquer autômato finito, seja ele determinístico ou não determinístico, é sempre possível definir sua matriz de incidência. Quando o autômato é não determinístico, um mesmo evento $\sigma \in \Sigma$ se apresenta em mais de uma coluna j , para uma mesma linha i .

Exemplo 3.7 A matriz de incidência \mathbf{A} associada ao autômato da Figura 3.1, é dada por

$$\mathbf{A} = \begin{bmatrix} \epsilon & \alpha + \lambda + \beta & \beta \\ \eta & \epsilon & \epsilon \\ \epsilon & \beta & \alpha + \lambda \end{bmatrix} \begin{matrix} \epsilon \\ \epsilon \\ e \end{matrix}$$

em que a linha 3 representa um estado marcado. Observe que os elementos do vetor $\phi(\mathbf{A})$ encontram-se externamente à matriz, ao seu lado direito, onde a presença do elemento 'e' indica um estado (linha) marcado. O vetor de estado inicial $\theta(\mathbf{A})$ não é apresentado, desde que nessa abordagem, considera-se que a primeira linha da matriz sempre representa o estado inicial. Dessa forma, sempre $\theta_1(\mathbf{A}) = e$, e os demais elementos $\theta_j(\mathbf{A}) = \epsilon, j = 2, \dots, N$, não sendo necessário sua apresentação.

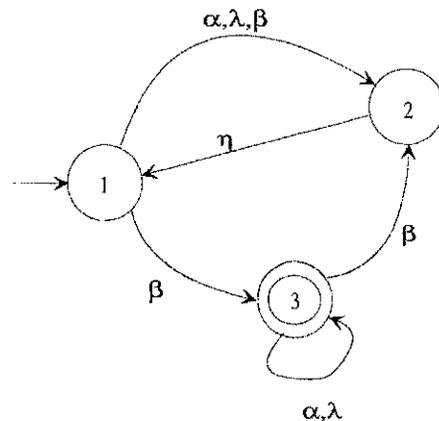


Figure 3.1: Autômato não determinístico com mais de uma função de transição entre dois estados.

Linguagem associada a uma matriz de incidência

A matriz de incidência é uma outra representação de um autômato. Isto significa que a matriz de incidência \mathbf{A} do autômato G apresenta uma linguagem $L(\mathbf{A}) = L(G)$ e uma linguagem $L_m(\mathbf{A}) = L_m(G)$.

Para construir uma linguagem a partir de uma matriz de incidência \mathbf{A} , é necessário a seguinte definição:

Definição 3.9 Seja \mathbf{A} uma matriz de incidência, cujos elementos $a_{i,j}$ definem caminhos de comprimento 1, que mudam o estado do autômato que ela representa, do estado i para o estado j . Assim, define-se a matriz

$$\mathbf{A}^n = \mathbf{A} \otimes \mathbf{A} \otimes \dots \otimes \mathbf{A},$$

como a matriz de caminhos, onde cada elemento $a_{i,j}^n$, representa um ou mais caminhos de comprimento n , que levam o autômato que ela representa, do estado i para o estado j . Os vetores de estados da matriz de caminhos \mathbf{A}^n são os mesmos de \mathbf{A} .

Observa-se que na matriz de caminhos, quando não há um caminho com n eventos que muda o estado do autômato de i para o estado j , o elemento $a_{i,j}^n = \epsilon$.

Exemplo 3.8 Do autômato G mostrado na Figura 3.2, cuja matriz de incidência é

$$\mathbf{A} = \begin{bmatrix} \epsilon & \alpha & \epsilon \\ \beta & \epsilon & \mu \\ \epsilon & \beta & \alpha \end{bmatrix} \begin{matrix} \epsilon \\ \epsilon \\ e \end{matrix}$$

A matriz de caminhos \mathbf{A}^2 desse autômato é

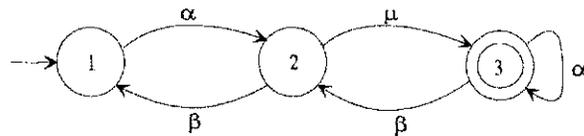


Figure 3.2: Autômato para exemplificar a construção da matriz de caminhos.

$$\mathbf{A}^2 = \mathbf{A} \otimes \mathbf{A} = \begin{bmatrix} \alpha\beta & \epsilon & \alpha\mu \\ \epsilon & \mu\beta + \beta\alpha & \mu\alpha \\ \beta\beta & \alpha\beta & \alpha\alpha + \beta\mu \end{bmatrix} \begin{matrix} \epsilon \\ \epsilon \\ e \end{matrix}$$

e sua matriz de caminhos \mathbf{A}^3 é

$$\mathbf{A}^3 = \mathbf{A} \otimes \mathbf{A} \otimes \mathbf{A} = \begin{bmatrix} \epsilon & \alpha\beta\alpha + \alpha\mu\beta & \alpha\mu\alpha \\ \mu\beta\beta + \beta\alpha\beta & \alpha\mu\beta & \mu\alpha\alpha \\ \alpha\beta\beta & \beta\beta\alpha + \alpha\alpha\beta + \beta\mu\beta & \alpha\beta\mu + \alpha\alpha\alpha + \beta\mu\alpha \end{bmatrix} \begin{matrix} \epsilon \\ \epsilon \\ e \end{matrix}$$

Nessas matrizes, cada seqüência em um elemento (i, j) representa uma palavra de comprimento 2 e 3, respectivamente. Essas seqüências definem as funções de transição estendidas do autômato G , que representam as mudanças dos estados i para os estados j . Em \mathbf{A}^2 , vê-se que do estado inicial para o estado inicial, existe a palavra $\alpha\beta$. Uma palavra reconhecida pelo autômato é vista em $a_{1,3}^2 = \alpha\mu$, e em $a_{1,3}^3 = \alpha\mu\alpha$ (linha 1, coluna 3), pois estas palavras determinam a mudança do estado inicial para o estado marcado 3 ($\phi_3(\mathbf{A}) = e$).

Com essa definição e utilizando a álgebra de dióides, tem-se que a linguagem da matriz de incidência $L(\mathbf{A})$ é o conjunto de todas as palavras formadas por seqüências iniciadas da linha 1, que levam a qualquer linha da matriz \mathbf{A} . Isto é,

Definição 3.10 Para uma dada matriz de incidência \mathbf{A} , sua linguagem é definida por

$$L(\mathbf{A}) = \bigoplus_i (\theta(\mathbf{A}) \otimes \mathbf{A}^i) = \bigoplus_i \bigoplus_{j=1}^n (\theta_1(\mathbf{A}) \otimes a_{1,j}^i), \quad (3.8)$$

onde $a_{1,j}^i$ é o elemento da linha 1, coluna j da matriz de caminhos \mathbf{A}^i .

Deve-se observar que essa definição determina a construção da linguagem da matriz \mathbf{A} utilizando apenas os elementos da linha 1 da matriz de caminhos \mathbf{A}^i , desprezando os elementos das outras linhas. Isto porque a linguagem é o conjunto formado por todas as seqüências de símbolos que partem do estado inicial (linha 1). As outras linhas $k = 2, 3, \dots$ de \mathbf{A}^i contêm informações de caminhos que iniciam dos respectivos estados k , levando ao estado j . A formalização da linguagem $L(\mathbf{A})$ é determinada como sendo a série formal sobre o semi-anel $D = \mathbb{B}\langle\langle\Sigma\rangle\rangle$. Também, deve-se observar que a linguagem $L(\mathbf{A})$ é prefixo-fechada, isto é, $L(\mathbf{A}) = \overline{L(\mathbf{A})}$, pois todas as palavras geradas em \mathbf{A}^i sempre têm como prefixos as palavras de \mathbf{A}^{i-1} . Assim, a linguagem da matriz de incidência \mathbf{A} representa a execução da matriz, que é igual a execução do autômato G representado por ela. Logo, $L(\mathbf{A}) = L(G)$.

Também, através dessa definição, pode-se determinar qual estado é alcançado quando ocorre a partir do estado 1 uma seqüência s de eventos de comprimento $|s|$.

Definição 3.11 *Seja \mathbf{A} uma matriz de incidência. A sublinguagem de \mathbf{A} , que contém todas as palavras que mudam o estado do autômato representado pela matriz \mathbf{A} para o estado k é definida como*

$$L(\mathbf{A})^k = \bigoplus_i (\theta(\mathbf{A}) \otimes \mathbf{A}^i \otimes \pi) = \bigoplus_i (\theta_1(\mathbf{A}) \otimes a_{1,k}^i),$$

onde π é um vetor coluna com o k -ésimo elemento igual ao elemento identidade 'e' e os demais iguais a ϵ , e $a_{1,k}^i$ é o elemento da linha 1, coluna k da matriz de caminhos \mathbf{A}^i .

Assim, $L(\mathbf{A})^k$ contém todas as palavras que levam da linha 1 à linha k .

O conjunto de todas as palavras de comprimento $|s| = i$ é determinado por

$$L(\mathbf{A})_i = \theta(\mathbf{A}) \otimes \mathbf{A}^i = \bigoplus_{j=1}^n (\theta_1(\mathbf{A}) \otimes a_{1,j}^i)$$

e, o conjunto de todas as palavras de comprimento $|s| = i$ que levam da linha 1 para a linha k é determinado por

$$L(\mathbf{A})_i^k = \theta(\mathbf{A}) \otimes \mathbf{A}^i \otimes \pi = \theta_1(\mathbf{A}) \otimes a_{1,k}^i.$$

Por outro lado, a linguagem marcada da matriz de incidência \mathbf{A} é definida como sendo:

Definição 3.12 *Para uma matriz de incidência \mathbf{A} , sua linguagem marcada é definida por*

$$L_m(\mathbf{A}) = \bigoplus_i (\theta(\mathbf{A}) \otimes \mathbf{A}^i \otimes \phi(\mathbf{A})) = \bigoplus_i \bigoplus_{j=1}^n (\theta_1(\mathbf{A}) \otimes a_{1,j}^i \otimes \phi_j(\mathbf{A})), \quad (3.9)$$

onde $a_{1,j}^i$ é o elemento da linha 1, coluna marcada j da matriz de caminhos \mathbf{A}^i .

Observe que essa definição é semelhante à Definição 3.10. Contudo, as palavras consideradas são as palavras que se encontram em uma coluna que leva a uma linha marcada j , denominada coluna marcada.

Semelhantemente à Definição 3.11, pode-se determinar se um estado marcado é alcançado quando ocorre, a partir do estado 1, uma seqüência s de eventos de comprimento $|s|$.

Definição 3.13 *Seja \mathbf{A} uma matriz de incidência. A sublinguagem marcada de \mathbf{A} , que contém todas as palavras que mudam o estado do autômato representado pela matriz \mathbf{A} para o estado marcado k é definida como*

$$L_m(\mathbf{A})^k = \bigoplus_i (\theta(\mathbf{A}) \otimes \mathbf{A}^i \otimes \vartheta) = \bigoplus_i (\theta_1(\mathbf{A}) \otimes a_{1,k}^i \otimes \phi_k(\mathbf{A})),$$

onde ϑ é um vetor coluna com o k -ésimo elemento igual ao elemento identidade 'e' e os demais são iguais a ϵ , e $a_{1,k}^i$ é o elemento da linha 1, coluna marcada k da matriz de caminhos \mathbf{A}^i .

Dessa forma, $L_m(\mathbf{A})^k$ contém todas as palavras reconhecidas que levam da linha 1 à linha marcada k .

O conjunto de todas as palavras reconhecidas de comprimento $|s| = i$ é determinado por

$$L_m(\mathbf{A})_i = \theta(\mathbf{A}) \otimes \mathbf{A}^i \otimes \phi(\mathbf{A}) = \bigoplus_{j=1}^n (\theta_1(\mathbf{A}) \otimes a_{1,j}^i \otimes \phi_j(\mathbf{A}))$$

e, o conjunto de todas as palavras de comprimento $|s| = i$ que levam da linha 1 à linha marcada k é determinado por

$$L_m(\mathbf{A})_i^k = \theta(\mathbf{A}) \otimes \mathbf{A}^i \otimes \vartheta = \theta_1(\mathbf{A}) \otimes a_{1,k}^i \otimes \phi_k(\mathbf{A}).$$

Exemplo 3.9 *Seja o autômato apresentado na Figura 3.3. Sua matriz de incidência \mathbf{A} é*

$$\mathbf{A} = \begin{bmatrix} \epsilon & \alpha \\ \beta & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \end{matrix}.$$

Para determinar sua linguagem, calcula-se

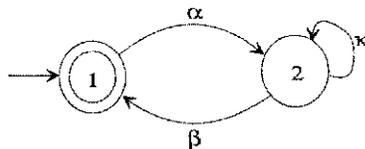


Figure 3.3: Autômato para determinação da linguagem.

$$\mathbf{A}^2 = \begin{bmatrix} \alpha\beta & \alpha\kappa \\ \kappa\beta & \beta\alpha + \kappa\kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \end{matrix}, \mathbf{A}^3 = \begin{bmatrix} \alpha\kappa\beta & \alpha\beta\alpha + \alpha\kappa\kappa \\ \beta\alpha\beta + \kappa\kappa\beta & \kappa\beta\alpha + \beta\alpha\kappa + \kappa\kappa\kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \end{matrix}, \dots$$

cujos elementos da primeira linha definem as palavras de comprimento 2, 3, ..., da linguagem de \mathbf{A} . Logo,

$$L(\mathbf{A}) = \overline{L(\mathbf{A})} = \{\epsilon, \alpha, \alpha\beta, \alpha\kappa, \alpha\kappa\beta, \alpha\beta\alpha, \alpha\kappa\kappa, \dots\}.$$

Observe que a palavra vazia ϵ está presente em $L(\mathbf{A})$, e que as palavras de comprimento i , encontram-se na linha 1 das matrizes \mathbf{A}^i . A linguagem marcada desta matriz é determinada pelas palavras da linha 1, coluna 1 de \mathbf{A}^i , desde que $\phi_1(\mathbf{A}) = e$:

$$L_m(\mathbf{A}) = \{\epsilon, \alpha\beta, \alpha\kappa\beta, \dots\}.$$

Observe também que a linguagem da matriz de incidência \mathbf{A} é a mesma linguagem do autômato G , encontrada seguindo seus arcos. Igualmente, a linguagem marcada.

Acessibilidade e coaccessibilidade dos autômatos representados por matrizes de incidência

Para um autômato G representado por uma matriz de incidência \mathbf{A} , a acessibilidade e coaccessibilidade de uma linha de \mathbf{A} são definidas como a seguir:

Definição 3.14 Uma linha j de uma matriz de incidência \mathbf{A} é acessível se para algum $i \in \mathbb{N}^*$,

$$\theta(\mathbf{A}) \otimes \mathbf{A}^i \otimes \pi \neq \epsilon,$$

em que π é um vetor coluna onde $\pi_j = e$, e os demais elementos $\pi_k = \epsilon, k \neq j$.

Assim, uma linha j é acessível se, partindo da linha 1, existe pelo menos uma seqüência $s \neq \epsilon$ que leva à linha j .

Definição 3.15 Uma linha i de uma matriz de incidência \mathbf{A} é coaccessível se para algum $k \in \mathbb{N}^*$,

$$\nu \otimes \mathbf{A}^k \otimes \phi(\mathbf{A}) \neq \epsilon,$$

em que ν é um vetor linha onde $\nu_i = e$, e os demais elementos $\nu_k = \epsilon, k \neq i$.

Dessa forma, uma linha i é coaccessível se, partindo da linha i , existe pelo menos uma seqüência $s \neq \epsilon$ que leva à linha marcada j .

Dessa maneira, uma matriz de incidência \mathbf{A} é acessível se toda linha j é acessível. Por outro lado, uma matriz de incidência \mathbf{A} é coaccessível se toda linha i é coaccessível. Uma matriz de incidência \mathbf{A} é *trim* se todas as linhas forem acessíveis e coaccessíveis ao mesmo tempo.

Exemplo 3.10 No Exemplo 3.7, todos os estados são acessíveis, pois

$$\begin{aligned} \theta(\mathbf{A}) \otimes \mathbf{A}^1 \otimes \pi &= \alpha + \lambda + \beta \neq \epsilon, \quad \text{para } \pi = \begin{bmatrix} \epsilon & e & \epsilon \end{bmatrix}^T, \\ \theta(\mathbf{A}) \otimes \mathbf{A}^2 \otimes \pi &= \beta\beta \neq \epsilon, \quad \text{para } \pi = \begin{bmatrix} \epsilon & e & \epsilon \end{bmatrix}^T, \end{aligned}$$

que levam do estado 1 ao estado 2,

$$\theta(\mathbf{A}) \otimes \mathbf{A}^1 \otimes \pi = \beta \neq \epsilon, \quad \text{para } \pi = \begin{bmatrix} \epsilon & \epsilon & e \end{bmatrix}^T,$$

que leva do estado 1 ao estado 3 e

$$\theta(\mathbf{A}) \otimes \mathbf{A}^2 \otimes \pi = \alpha\eta + \lambda\eta + \beta\eta \neq \epsilon, \quad \text{para } \pi = \begin{bmatrix} e & \epsilon & \epsilon \end{bmatrix}^T,$$

que leva do estado 1 ao estado 1. Todos os estados são coaccessíveis, pois para

$$\phi(\mathbf{A}) = \begin{bmatrix} \epsilon & \epsilon & e \end{bmatrix}^T,$$

$$\begin{aligned} \nu \otimes \mathbf{A}^1 \otimes \phi(\mathbf{A}) &= \beta \neq \epsilon, \quad \text{para } \nu = \begin{bmatrix} e & \epsilon & \epsilon \end{bmatrix} \\ \nu \otimes \mathbf{A}^2 \otimes \phi(\mathbf{A}) &= \eta\beta \neq \epsilon, \quad \text{para } \nu = \begin{bmatrix} \epsilon & e & \epsilon \end{bmatrix} \\ \nu \otimes \mathbf{A}^1 \otimes \phi(\mathbf{A}) &= \alpha + \lambda \neq \epsilon, \quad \text{para } \nu = \begin{bmatrix} \epsilon & \epsilon & e \end{bmatrix}. \end{aligned}$$

Logo, a matriz \mathbf{A} representa um autômato trim.

Equivalência entre autômatos representados por matrizes de incidência

As relações de equivalência entre autômatos [Gil62, HU69, HU79], também são válidas para as matrizes de incidência que os representam. Assim, define-se:

Definição 3.16 Duas matrizes de incidência \mathbf{A}_1 e \mathbf{A}_2 , tal que $\forall a_{1,i,j}, a_{2,i,j}, a_{1,i,j} \in \Sigma$ e $a_{2,i,j} \in \Sigma$, são ditas serem equivalentes, denotado por

$$\mathbf{A}_1 \equiv \mathbf{A}_2$$

se para quaisquer duas seqüências s_1 de \mathbf{A}_1 e s_2 de \mathbf{A}_2 , a ocorrência de um evento σ é possível tanto em \mathbf{A}_1 como em \mathbf{A}_2 , e levam às linhas i_1 de \mathbf{A}_1 e i_2 de \mathbf{A}_2 , tal que $\phi_{i_1}(\mathbf{A}_1) = e$ se e somente se $\phi_{i_2}(\mathbf{A}_2) = e$.

Exemplo 3.11 As matrizes de incidência

$$\mathbf{A}_1 = \begin{bmatrix} \epsilon & \alpha & \epsilon \\ \lambda & \epsilon & \beta \\ \epsilon & \beta & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \end{matrix}$$

e

$$\mathbf{A}_2 = \begin{bmatrix} \epsilon & \alpha \\ \lambda & \beta \end{bmatrix} \begin{matrix} e \\ \epsilon \end{matrix}$$

são equivalentes, pois

$$L_m(\mathbf{A}_1) = L_m(\mathbf{A}_2) = \alpha(\epsilon + \beta^*)\lambda.$$

Isto é, ambas reconhecem a mesma linguagem. Os autômatos que elas representam estão apresentados na Figura 3.4.

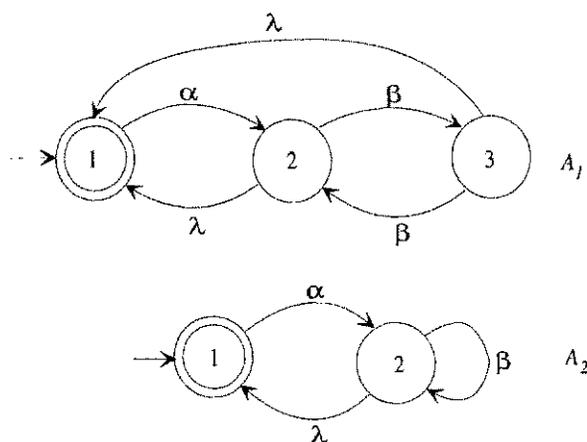


Figure 3.4: Autômatos equivalentes.

Composição de matrizes de incidência

Como as matrizes de incidência são utilizadas nessa Tese para modelar SEDs, é necessário definir sua composição síncrona (ver subseção 2.1.3). Para isto, a seguinte definição do operador de interseção é necessária:

Definição 3.17 *Dado um alfabeto Σ , considere-se o dióide da Definição 3.3. O operador \otimes define a interseção de elementos, como*

$$a \otimes b = \epsilon \oplus \left(\bigoplus_{i=1}^k \sigma^i \right), \text{ se } \sigma^i \subset a \wedge \sigma^i \subset b \quad (3.10)$$

$\forall a, b \in D$, onde

$$\begin{aligned} \sigma \otimes \epsilon &= \epsilon \\ \sigma^1 \otimes \sigma^2 &= \epsilon \\ \sigma \otimes \sigma &= \sigma, \end{aligned} \quad (3.11)$$

$\forall \sigma, \sigma^1, \sigma^2 \in \Sigma$. Para matrizes de incidência, esse operador é definido como

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B}, c_{i,j} = a_{i,j} \otimes b_{i,j},$$

para $a_{i,j} \otimes b_{i,j}$ definido de acordo com a equação (3.10) e

$$\theta(\mathbf{C}) = \theta(\mathbf{A}) \otimes \theta(\mathbf{B})$$

e

$$\phi(\mathbf{C}) = \phi(\mathbf{A}) \otimes \phi(\mathbf{B}).$$

Deve-se observar que, tornando $\sigma = e$, isto é, a palavra vazia representada por 'e', a interseção satisfaz as condição da Definição 3.17, ou seja,

$$\begin{aligned} e \otimes \epsilon &= \epsilon, \\ e \otimes \sigma' &= \epsilon, \forall \sigma' \in \Sigma \wedge \sigma' \neq e, \\ e \otimes e &= e. \end{aligned}$$

Com a definição deste operador, o produto síncrono entre duas matrizes de incidência é definido como:

Definição 3.18 *A composição síncrona de um matriz de incidência $\mathbf{A}_{m \times m}$ construída com símbolos do alfabeto Σ_1 com uma matriz de incidência $\mathbf{B}_{n \times n}$ construída com símbolos do alfabeto Σ_2 é definida por:*

$$\mathbf{P}_{mn \times mn} = \mathbf{A} \parallel \mathbf{B}, p_{k,l} = a_{i_A, j_A} \otimes b_{i_B, j_B},$$

onde $k = (i_A, i_B) = i_A + m(i_B - 1)$ e $l = (j_A, j_B) = j_A + m(j_B - 1)$, de tal forma que

$$P_{mn \times mn} = \begin{array}{|c|c|c|c|} \hline a_{1,1} \otimes [b_{1,1} \ b_{1,2} \dots \ b_{1,n}] & a_{1,2} \otimes [b_{1,1} \ b_{1,2} \dots \ b_{1,n}] & \dots & a_{1,m} \otimes [b_{1,1} \ b_{1,2} \dots \ b_{1,n}] \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline a_{m,1} \otimes [b_{1,1} \ b_{1,2} \dots \ b_{1,n}] & a_{m,2} \otimes [b_{1,1} \ b_{1,2} \dots \ b_{1,n}] & \dots & a_{m,m} \otimes [b_{1,1} \ b_{1,2} \dots \ b_{1,n}] \\ \hline a_{1,1} \otimes [b_{2,1} \ b_{2,2} \dots \ b_{2,n}] & a_{1,2} \otimes [b_{2,1} \ b_{2,2} \dots \ b_{2,n}] & \dots & a_{1,m} \otimes [b_{2,1} \ b_{2,2} \dots \ b_{2,n}] \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline a_{m,1} \otimes [b_{2,1} \ b_{2,2} \dots \ b_{2,n}] & a_{m,2} \otimes [b_{2,1} \ b_{2,2} \dots \ b_{2,n}] & \dots & a_{m,m} \otimes [b_{2,1} \ b_{2,2} \dots \ b_{2,n}] \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline a_{1,1} \otimes [b_{n,1} \ b_{n,2} \dots \ b_{n,n}] & a_{1,2} \otimes [b_{n,1} \ b_{n,2} \dots \ b_{n,n}] & \dots & a_{1,m} \otimes [b_{n,1} \ b_{n,2} \dots \ b_{n,n}] \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline a_{m,1} \otimes [b_{n,1} \ b_{n,2} \dots \ b_{n,n}] & a_{m,2} \otimes [b_{n,1} \ b_{n,2} \dots \ b_{n,n}] & \dots & a_{m,m} \otimes [b_{n,1} \ b_{n,2} \dots \ b_{n,n}] \\ \hline \end{array}$$

se $\Sigma_1 = \Sigma_2$. O vetor de estado inicial da composição é $\theta_{1,m,n}(\mathbf{P})$, onde o primeiro elemento é $\theta_1(\mathbf{P}) = e$, e os demais são iguais a ϵ ($\theta_l(\mathbf{P}) = \theta_{j_A}(\mathbf{A}) \otimes \theta_{j_B}(\mathbf{B})$), com $l = (j_A, j_B) = j_A + m(j_B - 1)$. O vetor de estados marcados é $\phi_{mn,1}(\mathbf{P})$, onde seus elementos são definidos por $\phi_k(\mathbf{P}) = \phi_{i_A}(\mathbf{A}) \otimes \phi_{i_B}(\mathbf{B})$, com $k = (i_A, i_B) = i_A + m(i_B - 1)$. Se $\exists \sigma_A \notin \Sigma_2$, ou $\exists \sigma_B \notin \Sigma_1$, tal que $\Sigma_1 \neq \Sigma_2$ e $\Sigma_1 \cap \Sigma_2 \neq \emptyset$, então,

$$P_{mn \times mn} = (\mathbf{A} \parallel \mathbf{B}) \oplus \mathbf{C}^{\neg},$$

onde

$$C_{mn \times mn}^{\neg} = \begin{array}{|c|c|c|c|c|} \hline \mathbf{A}^{-\mathbf{B}} \oplus \mathbf{D}_{b_{1,1}}^{-\mathbf{A}} & \mathbf{D}_{b_{1,2}}^{-\mathbf{A}} \oplus \mathbf{A}_l^{-\mathbf{B}} & \mathbf{D}_{b_{1,3}}^{-\mathbf{A}} \oplus \mathbf{A}_l^{-\mathbf{B}} & \dots & \mathbf{D}_{b_{1,n}}^{-\mathbf{A}} \oplus \mathbf{A}_l^{-\mathbf{B}} \\ \hline \mathbf{D}_{b_{2,1}}^{-\mathbf{A}} \oplus \mathbf{A}_l^{-\mathbf{B}} & \mathbf{A}^{-\mathbf{B}} \oplus \mathbf{D}_{b_{2,2}}^{-\mathbf{A}} & \mathbf{D}_{b_{2,3}}^{-\mathbf{A}} \oplus \mathbf{A}_l^{-\mathbf{B}} & \dots & \mathbf{D}_{b_{2,n}}^{-\mathbf{A}} \oplus \mathbf{A}_l^{-\mathbf{B}} \\ \hline \mathbf{D}_{b_{3,1}}^{-\mathbf{A}} \oplus \mathbf{A}_l^{-\mathbf{B}} & \mathbf{D}_{b_{3,2}}^{-\mathbf{A}} \oplus \mathbf{A}_l^{-\mathbf{B}} & \ddots & \ddots & \vdots \\ \hline \vdots & \vdots & \ddots & \mathbf{A}^{-\mathbf{B}} \oplus \mathbf{D}_{b_{n-1,n-1}}^{-\mathbf{A}} & \mathbf{D}_{b_{n-1,n}}^{-\mathbf{A}} \oplus \mathbf{A}_l^{-\mathbf{B}} \\ \hline \mathbf{D}_{b_{n,1}}^{-\mathbf{A}} \oplus \mathbf{A}_l^{-\mathbf{B}} & \mathbf{D}_{b_{n,2}}^{-\mathbf{A}} \oplus \mathbf{A}_l^{-\mathbf{B}} & \dots & \mathbf{D}_{b_{n,n-1}}^{-\mathbf{A}} \oplus \mathbf{A}_l^{-\mathbf{B}} & \mathbf{A}^{-\mathbf{B}} \oplus \mathbf{D}_{b_{n,n}}^{-\mathbf{A}} \\ \hline \end{array}$$

em que $\mathbf{A}^{-\mathbf{B}}$ é a matriz dos elementos de Σ_1 que não pertencem a Σ_2 , $\mathbf{A}_l^{-\mathbf{B}}$ é a matriz com a diagonal principal definida pelos auto-laços de \mathbf{A} que não pertencem a Σ_2 e $\mathbf{D}_{b_{i,j}}^{-\mathbf{A}}$ é uma matriz diagonal dos elementos de $\mathbf{B}^{-\mathbf{A}}$.

Esta definição decorre diretamente da Definição 2.22 do produto síncrono entre dois autômatos. Deve-se ver que:

1. A composição síncrona de dois autômatos gera um outro autômato com número de estados igual à multiplicação dos números de estados dos dois autômatos, embora possa haver estados não alcançáveis no autômato construído por esta composição. Assim, considerando m estados no autômato G_1 (que pode ser representado pela matriz de incidência \mathbf{A}) e n estados no autômato G_2 (que pode ser representado pela matriz de incidência \mathbf{B}), os estados do autômato G_3 (resultante da composição síncrona de

G_1 e G_2) de acordo com a Definição 2.22 são $q_{31} = (q_{11}, q_{21}), q_{32} = (q_{12}, q_{21}), \dots, q_{3m} = (q_{1m}, q_{21}), q_{3m+1} = (q_{11}, q_{22}), \dots, q_{32m} = (q_{1m}, q_{22}), \dots, q_{3(n-1) \times m} = (q_{1m}, q_{2n-1}), \dots, q_{3m \times n} = (q_{1m}, q_{2n})$.

2. Considerando primeiramente que $\Sigma_1 = \Sigma_2$, então a equação (2.13) é definida apenas pela primeira condição. Dessa forma, G_3 em uma representação de matriz, tem uma dimensão $m \times n$, onde se pode observar que os primeiros m estados de G_3 são os estados de G_1 em conjunto com o primeiro estado de G_2 .
3. Os eventos referentes a estas primeiras m linhas/colunas da matriz de G_3 são os elementos dos m estados de G_1 intersectados com os elementos do estado q_{21} de G_2 . Esta operação é semelhante a fazer uma variação de i de 1 até m para os estados de G_1 , mantendo j fixo em 1 no autômato G_2 e calcular $\sigma_{3k} = \sigma_{1i} \cap \sigma_{21}$ para determinar o k -ésimo σ_3 nos m primeiros estados de G_3 , de acordo com a primeira condição da equação (2.13). Esta mesma idéia é estendida aos demais elementos do autômato G_3 , de forma que se encontra

$$\sigma_{3k,i} = \sigma_{1i_A, j_A} \cap \sigma_{2i_B, j_B},$$

em que para cada valor de $j_B = 1$ até n , j_A varia de 1 até m , e para cada $i_B = 1$ até n , i_A varia de 1 até m . Observe que a operação de interseção pode ser substituída pelo operador \otimes . Isto determina a construção da matriz $\mathbf{P} = \mathbf{A} || \mathbf{B}$, que se apresenta com a estrutura apresentada na Definição 3.18.

4. Por outro lado, se existe algum elemento que pertence à G_1 , mas que não pertence à G_2 e vice-versa, isto é, $\Sigma'_1 \cap \Sigma'_2 = \emptyset$, em que $\Sigma'_1 \subset \Sigma_1$ e $\Sigma'_2 \subset \Sigma_2$, tem-se que a equação (2.13) é definida por todas as condições apresentadas. Os elementos que satisfazem $\Sigma''_1 \cap \Sigma''_2 \neq \emptyset$, em que $\Sigma''_1 \subset \Sigma_1$ e $\Sigma''_2 \subset \Sigma_2$, com $\Sigma''_1 \cup \Sigma'_1 = \Sigma_1$ e $\Sigma''_2 \cup \Sigma'_2 = \Sigma_2$, são definidos na composição síncrona por meio da primeira condição de (2.13), como visto anteriormente. Os elementos que só pertencem a um único alfabeto (conseqüentemente à apenas um dos autômatos), podem compor uma matriz de incidência de uma composição paralela (onde a linguagem de um autômato é independente da linguagem do outro) a qual é construída de acordo com as duas últimas condições da equação (2.13). Dessa forma, a definição de δ_3 determina as mudanças de estados de G_3 como um símbolo σ que muda o estado do autômato G_1 , de q_{1i} para o estado $q_{1i'}$, sempre mantendo o estado do autômato G_2 fixo e vice-versa. Assim, tem-se uma representação de matriz, com uma dimensão $m \times n$. Procedendo igualmente, isto é, dividindo esta matriz em submatrizes de dimensão $m \times m$, tem-se que:

- Na primeira submatriz, definida pelos primeiros m estados do autômato G_3 , encontra-se a representação da matriz de incidência \mathbf{A} do autômato G_1 , desde que mantendo fixo o estado q_{21} do autômato G_2 , a definição de δ_3 é determinada

pelas mudanças de estado do autômato G_1 . Contudo, se o autômato G_2 contiver auto-laços nesse estado (q_{21}), então esses auto-laços aparecerão nessa submatriz, em cada um dos estados de G_3 . Como os auto-laços são representados por elementos (i, i) na matriz de incidência, então, o resultado dessa primeira submatriz é $\mathbf{A} \oplus \mathbf{D}_{b_{1,1}}^{-\mathbf{A}}$, em que $\mathbf{D}_{b_{1,1}}^{-\mathbf{A}}$ é uma matriz em que os elementos da diagonal principal são dados pelo elemento $b_{1,1}$ e que não existem em \mathbf{A} , e os demais elementos são ϵ ;

- A segunda submatriz, definida como as m colunas subseqüentes (de $m + 1$ até $2m$) e mesmas linhas (de 1 até m), apresenta os elementos que definem as mudanças de estado em G_3 . Assim, δ_3 é definida como as funções de transição que mudam apenas o estado do autômato G_1 , mantendo fixo o estado q_{22} do autômato G_2 . Nesse caso, a função do estado $(1, 1)$ de G_3 para o estado $(1, 2)$ de G_3 é definida por δ_2 que leva G_2 do estado 1 para o estado 2, pois o estado q_{11} está mantido fixo. Para a função do estado $(2, 1)$ de G_3 que muda para o estado $(2, 2)$, tem-se igualmente a mesma função δ_2 , pois o estado q_{12} está mantido fixo e assim por diante nesta diagonal. Contudo, se houver algum auto-laço definido em G_1 , ele deve ser também definido. Nesse caso, um auto-laço em um estado q_{1i} de G_1 deve aparecer em cada mudança de estado de G_3 em que esse estado k de G_1 se mantém fixo: (k, i) para (k, j) . Os demais elementos dessa submatriz são iguais a ϵ , desde que δ_3 não é definido quando há mudança de estados para os dois autômatos, ao mesmo tempo: $(1, 1)$ para $(2, 2)$, $(2, 1)$ para $(1, 2)$, e assim por diante. Dessa forma, encontra-se que essa submatriz pode ser representada por $\mathbf{D}_{b_{1,2}}^{-\mathbf{A}} \oplus \mathbf{A}_l^{-\mathbf{B}}$, em que $\mathbf{D}_{b_{1,2}}^{-\mathbf{A}}$ é uma matriz cujos elementos da diagonal principal são dados pelo elemento $b_{1,2}$ e que não existem em \mathbf{A} , e os demais elementos são iguais a ϵ , e $\mathbf{A}_l^{-\mathbf{B}}$ é uma matriz em que seus elementos (i, i) são os auto-laços de \mathbf{A} definidos em suas respectivas posições (i, i) e que não pertencem a \mathbf{B} , e os demais elementos são iguais a ϵ ;
- Considerando agora a terceira submatriz, definida como as m linhas subseqüentes (de $m + 1$ até $2m$) e as colunas de 1 até m , tem-se δ_3 definida igualmente à submatriz anterior. Assim, esta submatriz pode ser representada por $\mathbf{D}_{b_{2,1}}^{-\mathbf{A}} \oplus \mathbf{A}_l^{-\mathbf{B}}$, em que $\mathbf{D}_{b_{2,1}}^{-\mathbf{A}}$ é uma matriz em que os elementos da diagonal principal são dados pelo elemento $b_{2,1}$ e que não existem em \mathbf{A} , e os demais elementos são iguais a ϵ ;
- Na quarta submatriz, sua avaliação define a mesma condição da primeira submatriz: $\mathbf{A} \oplus \mathbf{D}_{b_{2,2}}^{-\mathbf{A}}$, em que $\mathbf{D}_{b_{2,2}}^{-\mathbf{A}}$ é uma matriz em que os elementos da diagonal principal são dados pelo elemento $b_{2,2}$ e que não existem em \mathbf{A} , e os demais elementos são iguais a ϵ .
- As outras submatrizes são semelhantes à estas, que em suas respectivas posições determinam a matriz denotada por \mathbf{C}^- .

Como a composição síncrona de duas matrizes de incidência \mathbf{A} e \mathbf{B} é igual à composição síncrona dos respectivos autômatos $G_{\mathbf{A}}$ e $G_{\mathbf{B}}$ que elas representam, então, a linguagem da composição síncrona $L(\mathbf{P})$ satisfaz

$$L(\mathbf{P}) = L(\mathbf{A}||\mathbf{B}) = L(SINC(G_{\mathbf{A}}, G_{\mathbf{B}})),$$

onde $SINC(G_{\mathbf{A}}, G_{\mathbf{B}})$ representa o operador de produto síncrono de dois autômatos definido em Wonham [Won99].

Com a Definição 3.18, o seguinte algoritmo é apresentado para a construção do produto síncrono de duas matrizes de incidência, em que $\Sigma_1 \cap \Sigma_2 \neq \emptyset$:

Algoritmo 3.1 Construção do produto síncrono $\mathbf{P} = \mathbf{A}||\mathbf{B}$

1. Faça $\theta_1(\mathbf{C}) = e$
2. Faça $j_{\mathbf{A}} = j_{\mathbf{B}} = i_{\mathbf{A}} = i_{\mathbf{B}} = 1$.
3. Enquanto $i_{\mathbf{B}} \leq n$, faça:
 - a) Se $b_{i_{\mathbf{B}}, j_{\mathbf{B}}} = \epsilon$, faça $j_{\mathbf{B}} = j_{\mathbf{B}} + 1$
 - b) Se $j_{\mathbf{B}} > n$, faça $i_{\mathbf{B}} = i_{\mathbf{B}} + 1$ e $j_{\mathbf{B}} = 0$
 - c) Enquanto $i_{\mathbf{A}} \leq m$, faça:
 - i. Se $a_{i_{\mathbf{A}}, j_{\mathbf{A}}} = \epsilon$, faça $j_{\mathbf{A}} = j_{\mathbf{A}} + 1$
 - ii. Se $j_{\mathbf{A}} > m$, faça $i_{\mathbf{A}} = i_{\mathbf{A}} + 1$ e $j_{\mathbf{A}} = 0$
 - iii.. Faça $k = i_{\mathbf{A}} + m(i_{\mathbf{B}} - 1)$
 - iv. Faça $l = j_{\mathbf{A}} + m(j_{\mathbf{B}} - 1)$
 - v. Faça $p_{k,l} = a_{i_{\mathbf{A}}, j_{\mathbf{A}}} \otimes b_{i_{\mathbf{B}}, j_{\mathbf{B}}}$
 - vi. Se $\phi_{i_{\mathbf{A}}}(\mathbf{A}) = e$ e $\phi_{i_{\mathbf{B}}}(\mathbf{B}) = e$, faça $\phi_k(\mathbf{C}) = e$
 - vii. Faça $j_{\mathbf{A}} = j_{\mathbf{A}} + 1$
 - viii. Se $j_{\mathbf{A}} > m$, faça $i_{\mathbf{A}} = i_{\mathbf{A}} + 1$ e $j_{\mathbf{A}} = 0$
 - d) Faça $j_{\mathbf{B}} = j_{\mathbf{B}} + 1$
 - e) Se $j_{\mathbf{B}} > n$, faça $i_{\mathbf{B}} = i_{\mathbf{B}} + 1$ e $j_{\mathbf{B}} = 0$
4. Se $\exists \sigma_{\mathbf{A}} \notin \Sigma_2$, ou $\exists \sigma_{\mathbf{B}} \notin \Sigma_1$, componha a matriz \mathbf{C}^{\neg} , e faça $p_{k,l} = p_{k,l} \oplus c_{k,l}^{\neg}$.

Observação: Esse algoritmo é similar ao algoritmo da construção do produto síncrono de autômatos [RW89]. Dessa forma, sua ordem de complexidade é $O(nm)$. Considerando que $n = m = N$, sua ordem de complexidade é $O(N^2)$.

Exemplo 3.12 Dadas as matrizes de incidência

$$\mathbf{A} = \overbrace{\begin{bmatrix} \epsilon & \alpha \\ \beta & \epsilon \end{bmatrix}}^{m=2} \begin{matrix} e \\ \epsilon \end{matrix}$$

e

$$\mathbf{B} = \overbrace{\begin{bmatrix} \epsilon & \alpha & \epsilon \\ \beta & \epsilon & \eta \\ \lambda & \beta & \epsilon \end{bmatrix}}^{n=3} \begin{matrix} e \\ e \\ \epsilon \end{matrix}$$

que representam os autômatos da Figura 3.5(a) e 3.5(b), seu produto síncrono determina a matriz \mathbf{P} , em que seus elementos $p_{k,l} = p_{(i_A, i_B), (j_A, j_B)}$, $k = i_A + m(i_B - 1)$ e $l = j_A + m(j_B - 1)$, nos elementos $a_{i_A, j_A} = b_{i_B, j_B} \neq \epsilon$, são dados por:

- $k = 1 + 2 \times 0 = 1$, $l = 2 + 2 \times 1 = 4$, $p_{1,4} = p_{(1,1), (2,2)} = a_{1,2} \otimes b_{1,2} = \alpha$
- $k = 2 + 2 \times 1 = 4$, $l = 1 + 2 \times 0 = 1$, $p_{4,1} = p_{(2,2), (1,1)} = a_{2,1} \otimes b_{2,1} = \beta$
- $k = 2 + 2 \times 2 = 6$, $l = 1 + 2 \times 1 = 3$, $p_{6,3} = p_{(2,3), (1,2)} = a_{2,1} \otimes b_{3,2} = \beta$

$$\mathbf{A} \parallel \mathbf{B} = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \alpha & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \beta & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \beta & \epsilon & \epsilon & \epsilon \end{bmatrix} \begin{matrix} e \\ e \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \end{matrix}$$

Como $\lambda, \eta \in \mathbf{B}$, mas $\lambda, \eta \notin \mathbf{A}$, então

$$\mathbf{C}^{-1} = \begin{bmatrix} \overbrace{\begin{bmatrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{bmatrix}}^{\mathbf{A}^{-B} \oplus \mathbf{D}_{b_{1,1}}^{-A}} & \overbrace{\begin{bmatrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{bmatrix}}^{\mathbf{D}_{b_{1,2}}^{-A} \oplus \mathbf{A}_l^{-B}} & \overbrace{\begin{bmatrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{bmatrix}}^{\mathbf{D}_{b_{1,3}}^{-A} \oplus \mathbf{A}_l^{-B}} \\ \overbrace{\begin{bmatrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{bmatrix}}^{\mathbf{D}_{b_{2,1}}^{-A} \oplus \mathbf{A}_l^{-B}} & \overbrace{\begin{bmatrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{bmatrix}}^{\mathbf{A}^{-B} \oplus \mathbf{D}_{b_{2,2}}^{-A}} & \overbrace{\begin{bmatrix} \eta & \epsilon \\ \epsilon & \eta \end{bmatrix}}^{\mathbf{D}_{b_{2,2}}^{-A} \oplus \mathbf{A}_l^{-B}} \\ \overbrace{\begin{bmatrix} \lambda & \epsilon \\ \epsilon & \lambda \end{bmatrix}}^{\mathbf{D}_{b_{3,1}}^{-A} \oplus \mathbf{A}_l^{-B}} & \overbrace{\begin{bmatrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{bmatrix}}^{\mathbf{D}_{b_{3,2}}^{-A} \oplus \mathbf{A}_l^{-B}} & \overbrace{\begin{bmatrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{bmatrix}}^{\mathbf{A}^{-B} \oplus \mathbf{D}_{b_{1,1}}^{-A}} \end{bmatrix} \begin{matrix} e \\ e \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \end{matrix}$$

e

$$\mathbf{P} = (\mathbf{A} \parallel \mathbf{B}) \oplus \mathbf{C}^{-1} = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \alpha & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \eta & \epsilon \\ \beta & \epsilon & \epsilon & \epsilon & \epsilon & \eta \\ \lambda & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \lambda & \beta & \epsilon & \epsilon & \epsilon \end{bmatrix} \begin{matrix} e \\ e \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \end{matrix}$$

que representa o autômato G_3 composto por G_1 e G_2 através do produto síncrono, e que está mostrado na Figura 3.5(c). Veja que os estados marcados são definidos pelas linha $k = 1$ que representa o par $(i_{\mathbf{A}}, i_{\mathbf{B}}) = (1, 1)$ e $k = 2$ que representa o par $(i_{\mathbf{A}}, i_{\mathbf{B}}) = (2, 1)$, pois $\phi_1(\mathbf{P}) = \phi_{1_{\mathbf{A}}}(\mathbf{A}) \otimes \phi_{1_{\mathbf{B}}}(\mathbf{B}) = e$, e $\phi_2(\mathbf{P}) = \phi_{2_{\mathbf{A}}}(\mathbf{A}) \otimes \phi_{1_{\mathbf{B}}}(\mathbf{B}) = e$. O vetor de estado inicial é

$$\theta(\mathbf{P}) = \left[\theta_1(\mathbf{A}) \otimes \theta_1(\mathbf{B}) \quad \epsilon \quad \epsilon \quad \epsilon \quad \epsilon \quad \epsilon \right] = \left[e \quad \epsilon \quad \epsilon \quad \epsilon \quad \epsilon \quad \epsilon \right].$$

Deve-se observar que a linha 2 da matriz \mathbf{P} contém todos os seus elementos $p_{2,j} = \epsilon$, mas esta linha é alcançada pela seqüência $s = \theta(\mathbf{P}) \otimes \mathbf{P}^3 \otimes \pi = p_{1,4}p_{4,6}p_{6,2} = \alpha\eta\lambda$, para

$$\pi = \left[\epsilon \quad e \quad \epsilon \quad \epsilon \quad \epsilon \quad \epsilon \right]^T,$$

o que implica que esta linha representa um estado bloqueado na composição de G_1 com G_2 .

3.2 Síntese do Supervisor

Nessa Seção, é apresentada a síntese de supervisores de SEDs através da álgebra de dióides e das matrizes de incidência. Para isto, torna-se necessário compreender como é descrita a dinâmica do SED modelado por uma matriz de incidência.

Assim, dada uma matriz de incidência \mathbf{A} que é um modelo de um SED, os estados q são representados pelas linhas i . A linha 1 dessa matriz representa o estado inicial do SED. Os eventos do SED são representados por símbolos de um alfabeto Σ , em que $\Sigma = \Sigma_c \cup \Sigma_{uc}$ e $\Sigma_c \cap \Sigma_{uc} = \emptyset$. Então, um evento σ é dito estar habilitado em um estado q (linha i), se

$$\exists j | a_{i,j} = \sigma,$$

isto é, para $a_{i,j} \neq \epsilon$, há um evento σ habilitado na linha i , cuja ocorrência leva a matriz à linha j . Dessa forma, o conjunto de eventos possíveis de ocorrerem no estado q (linha i) é representado por

$$\Sigma(i) \subseteq \Sigma.$$

Um SED modelado por uma matriz de incidência gera, então, seqüências de eventos iniciadas no estado inicial (linha 1), dadas por

$$s = a_{1,k_1} a_{k_1,k_2} \dots a_{k_{n-1},k_n} \quad (3.12)$$

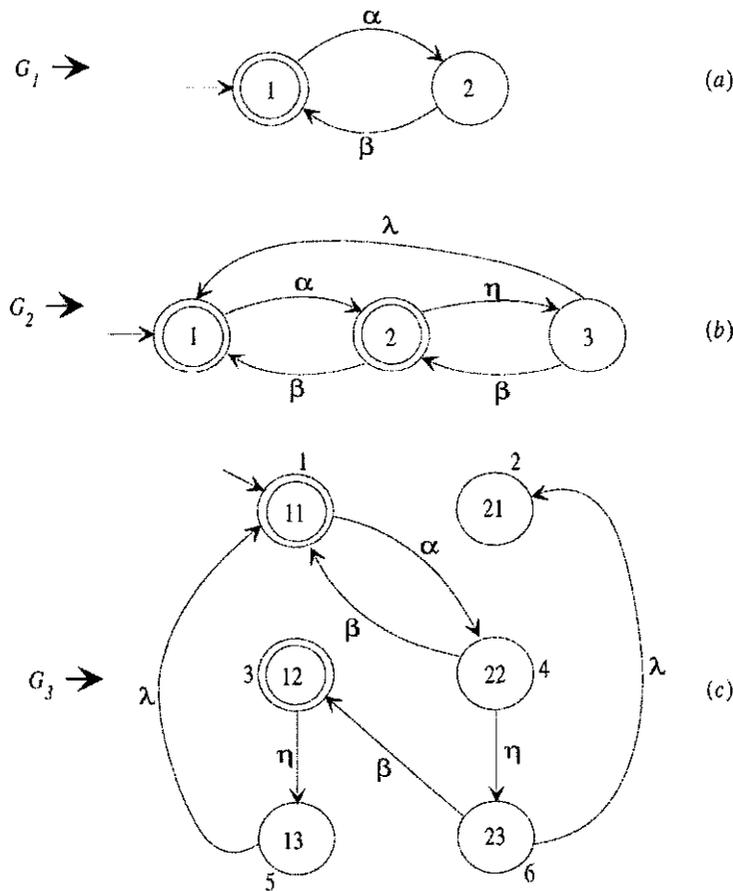


Figure 3.5: Autômatos (a) G_1 , (b) G_2 e (c) G_3 - produto síncrono de G_1 e G_2 .

construídas de acordo com a Definição 3.10. Sendo assim, a linguagem do SED modelado pela matriz de incidência é o conjunto de todas as seqüências possíveis de ocorrerem a partir de seu estado inicial. Isto é,

$$L = \{s_1, s_2, \dots\}$$

onde $s_i, i = 1, 2, \dots$ é dado de acordo com (3.12).

Por outro lado, dada a representação dos estados marcados do SED na matriz de incidência como sendo o vetor $\phi(\mathbf{A})$, a linguagem reconhecida (ou marcada) é o conjunto de todas as seqüências possíveis de ocorrerem a partir da linha 1, e que levam a uma linha marcada (ver Definição 3.12). Isto é,

$$L_m(\mathbf{A}) = \{s_1, s_2, \dots\} \tag{3.13}$$

com $s_i, i = 1, 2, \dots$ é dado de acordo com (3.12) e $\phi_{k_n}(\mathbf{A}) = e$.

Então, um SED modelado por uma matriz de incidência \mathbf{A} gera os eventos $\Sigma(i) \subseteq \Sigma$, iniciando da linha $i = 1$.

Dessa forma, tendo-se em vista como se dá a evolução dinâmica do SED modelado por uma matriz de incidência, para sintetizar um supervisor, considera-se igualmente à Ramadge e Wonham [RW89], que o alfabeto de eventos Σ é particionado em eventos controláveis Σ_c e

eventos não controláveis Σ_{uc} satisfazendo a

$$\Sigma = \Sigma_c \cup \Sigma_{uc}$$

e

$$\Sigma_c \cap \Sigma_{uc} = \emptyset.$$

De acordo com a Definição 3.7, da matriz de incidência de um autômato, e com a partição do alfabeto de eventos Σ , definem-se duas outras matrizes de incidência: \mathbf{A}_c e \mathbf{A}_{uc} , que contêm os eventos controláveis e não controláveis do autômato G .

Definição 3.19 Dado um autômato G que representa um SED, construído com eventos de um alfabeto $\Sigma = \Sigma_c \cup \Sigma_{uc}$, definem-se as matrizes de incidência \mathbf{A}_c e \mathbf{A}_{uc} , com dimensões iguais a do autômato G , isto é, $N \times N$, dadas por

$$\mathbf{A}_c = [(a_c)_{i,j}], (a_c)_{i,j} = \begin{cases} \sigma_c & \text{se } \exists \sigma_c \text{ do estado } i \text{ para o estado } j; \\ \epsilon & \text{caso contrário} \end{cases}$$

e

$$\mathbf{A}_{uc} = [(a_{uc})_{i,j}], (a_{uc})_{i,j} = \begin{cases} \sigma_{uc} & \text{se } \exists \sigma_{uc} \text{ do estado } i \text{ para o estado } j; \\ \epsilon & \text{caso contrário,} \end{cases}$$

onde $\sigma_c \in \Sigma_c$ e $\sigma_{uc} \in \Sigma_{uc}$. Os vetores de estado inicial e estados marcados dessas duas matrizes são os mesmos da matriz \mathbf{A} .

Exemplo 3.13 Considerando que o autômato visto na Figura 3.6, tem $\Sigma_c = \{\alpha, \eta\}$ e $\Sigma_{uc} = \{\beta\}$, então

$$\mathbf{A}_c = \begin{bmatrix} \epsilon & \alpha & \epsilon \\ \eta & \epsilon & \epsilon \\ \epsilon & \epsilon & \alpha \end{bmatrix} \begin{matrix} \epsilon \\ \epsilon \\ e \end{matrix}$$

e

$$\mathbf{A}_{uc} = \begin{bmatrix} \epsilon & \epsilon & \beta \\ \epsilon & \epsilon & \epsilon \\ \epsilon & \beta & \epsilon \end{bmatrix} \begin{matrix} \epsilon \\ \epsilon \\ e \end{matrix}$$

onde a linha 3 define um estado marcado, e a linha 1 define o estado inicial em ambas as matrizes.

Observação: Pode-se ver que esta partição define duas matrizes em que, a matriz de incidência \mathbf{A} é reconstruída através da sua soma, isto é,

$$\boxed{\mathbf{A} = \mathbf{A}_c \oplus \mathbf{A}_{uc}.}$$

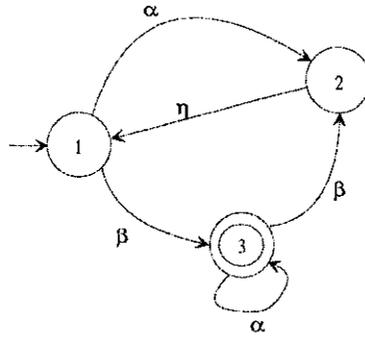


Figure 3.6: Autômato determinístico.

Exemplo 3.14 Dadas as matrizes de incidência \mathbf{A}_c e \mathbf{A}_{uc} do Exemplo 3.13, sua soma é

$$\mathbf{A}_c \oplus \mathbf{A}_{uc} = \begin{bmatrix} \epsilon & \alpha & \epsilon \\ \eta & \epsilon & \epsilon \\ \epsilon & \epsilon & \alpha \end{bmatrix} \epsilon \oplus \begin{bmatrix} \epsilon & \epsilon & \beta \\ \epsilon & \epsilon & \epsilon \\ \epsilon & \beta & \epsilon \end{bmatrix} \epsilon = \begin{bmatrix} \epsilon & \alpha & \beta \\ \eta & \epsilon & \epsilon \\ \epsilon & \beta & \alpha \end{bmatrix} \epsilon = \mathbf{A}$$

onde \mathbf{A} é a matriz de incidência do autômato visto na Figura 3.6, em que $\Sigma_c = \{\alpha, \eta\}$ e $\Sigma_{uc} = \{\beta\}$.

Deve-se observar que em uma matriz de incidência gerada através da soma de duas outras matrizes, os estados marcados são os estados (linhas) que são marcados nas duas matrizes.

Tendo definido a partição da matriz de incidência do autômato que modela o SED, torna-se necessário representar a especificação de comportamento por um autômato G_E e sua matriz de incidência \mathbf{E} , de forma semelhante a \mathbf{A} . Deve-se ver que uma especificação de comportamento é a tarefa que se deseja que o sistema realize.

Definição 3.20 Dada uma especificação de comportamento G_E para um autômato G que é um modelo de um SED, esta tem sua matriz de incidência definida por

$$\mathbf{E} = [e_{i,j}], e_{i,j} = \begin{cases} \sigma & \text{se } \exists \sigma \text{ do estado } i \text{ para o estado } j \text{ em } G_E; \\ \epsilon & \text{caso contrário} \end{cases}$$

em que $\sigma \in \Sigma^*$, é uma expressão regular do tipo $\sigma = \sigma^1 + \sigma^2 + \dots + \sigma^n$, $\sigma^1, \sigma^2, \dots, \sigma^n \in \Sigma$, tal que qualquer um evento $\sigma^k \in \Sigma$ ocorrendo, provoca no autômato da especificação de comportamento G_E , a mudança do estado i para o estado j . O estado inicial e os estados marcados são definidos igualmente à Definição 3.8.

A partir daqui, toda referência à especificação de comportamento G_E , será feita utilizando-se a representação matricial \mathbf{E} .

Observação: No contexto desse trabalho, deve-se considerar as seguintes denominações:

1. Um sub-autômato G' apresenta uma estrutura semelhante, mas apenas com parte dos estados ou com parte das transições de G ;

2. Uma sub-linguagem L' apresenta parte das seqüências da linguagem L . Não necessariamente $L' \subset L$ indica que o autômato de L' é um sub-autômato de L ;
3. Uma submatriz \mathbf{A}' é formada pelas primeiras m linhas e as primeiras m colunas de \mathbf{A} e mantém uma correlação elemento a elemento.

Para reescrever o problema de controle de Ramadge e Wonham utilizando as matrizes de incidência, é necessário que a especificação de comportamento sempre seja uma submatriz da matriz de incidência \mathbf{A} . Se \mathbf{E} não é uma submatriz de \mathbf{A} , torna-se necessário transformar as matrizes \mathbf{A} e \mathbf{E} em outras matrizes, denotadas por $\mathbf{A}^\#$ e $\mathbf{E}^\#$, respectivamente, tal que $\mathbf{E}^\#$ seja uma submatriz de $\mathbf{A}^\#$, e satisfaçam

$$\begin{aligned} L(\mathbf{A}^\#) &= L(\mathbf{A}) & \text{e} & & L_m(\mathbf{A}^\#) &= L_m(\mathbf{A}), \\ L(\mathbf{E}^\#) &= L(\mathbf{E}) & \text{e} & & L_m(\mathbf{E}^\#) &= L_m(\mathbf{E}). \end{aligned}$$

Assim, considerando que a especificação de comportamento \mathbf{E} seja apresentada diretamente como uma submatriz de \mathbf{A} , ou através da transformação em $\mathbf{E}^\#$, sempre a relação

$$L(\mathbf{E}) \subseteq L(\mathbf{A})$$

é verificada. Esses dois casos representam uma especificação de comportamento genérica, válida para as situações em que $L(\mathbf{E}) \subseteq L(\mathbf{A})$ (ver Exemplo 2.12) e $L(\mathbf{E}) \not\subseteq L(\mathbf{A})$ (ver Exemplo 2.13).

A situação de uma especificação de comportamento \mathbf{E} apresentada não ser submatriz de \mathbf{A} , necessita do seguinte procedimento:

1. Constrói-se a composição síncrona de \mathbf{A} com uma matriz qualquer que gera Σ^* , para determinar uma nova matriz $\mathbf{A}^\#$ que tem $L(\mathbf{A}^\#) = L(\mathbf{A})$ e $L_m(\mathbf{A}^\#) = L_m(\mathbf{A})$. Transformando a especificação de comportamento \mathbf{E} em uma matriz de incidência \mathbf{E}^* que gera Σ^* , através da inclusão de um estado proibido (linha i_e /coluna j_e), denominado de estado de *erro*, utiliza-se \mathbf{E}^* para construir a matriz $\mathbf{A}^\# = \mathbf{A} \parallel \mathbf{E}^*$;
2. Da matriz $\mathbf{A}^\#$, substituindo todos os elementos das linhas/colunas que formam os pares (i_A, i_e) , (j_A, j_e) pelo símbolo ϵ , encontra-se a matriz $\mathbf{E}^\#$ tal que $L(\mathbf{E}^\#) \subset L(\mathbf{A}^\#) = L(\mathbf{A})$, com $L(\mathbf{E}) = L(\mathbf{E}^\#)$ e $\mathbf{E}^\#$ sendo uma submatriz de $\mathbf{A}^\#$;
3. O vetor $\phi(\mathbf{E}^\#)$ é determinado pelos elementos resgatados do vetor original $\phi(\mathbf{E})$, cujos elementos satisfazem o par (i_A, i_E) , com $\phi_{i_E}(\mathbf{E}) = e$, o que garante $L_m(\mathbf{E}^\#) = L_m(\mathbf{E})$.

Dessa forma, a matriz $\mathbf{E}^\#$ se apresenta como uma submatriz de $\mathbf{A}^\#$, e conseqüentemente,

$$L(\mathbf{E}^\#) \subset L(\mathbf{A}^\#) = L(\mathbf{A}),$$

e

$$\begin{aligned} L(\mathbf{E}^\#) &= L(\mathbf{E}), \\ L_m(\mathbf{E}^\#) &= L_m(\mathbf{E}). \end{aligned}$$

Para os casos em que há a necessidade dessa transformação, é apresentado a seguir o algoritmo que transforma uma matriz de incidência qualquer para que sua linguagem reconhecida seja Σ^* .

Algoritmo 3.2 Transformação de matrizes de incidência para reconhecimento de Σ^*

1. Marcar todas as linhas de \mathbf{F} (tornar $\forall i, \phi_i(\mathbf{F}) = e$);
2. Para cada linha i de \mathbf{F} , incluir os auto-laços

$$f_{i,i} = \Sigma - \{\sigma^1, \sigma^2, \dots, \sigma^n\}$$

tal que $\sigma^k \notin f_{i,j}, k = 1, 2, \dots, n$;

3. Incluir no final da matriz \mathbf{F} uma coluna j_e e uma linha marcada de 'erro' i_e representando uma linha proibida, e fazer:

i) Para $i < i_e$, faça:

- a) $\forall j, j < j_e, f_{i,j_e} = \sigma$, se $\sigma \in f_{j,i} \vee \sigma \in f_{j,j}$ e $f_{i,k} \neq \sigma, \forall k, k < j_e$;
- b) $\forall j, j < j_e, f_{i,j_e} = \sigma'$ se $f_{i,j} \neq \sigma', \forall \sigma' \in \Sigma$;

ii) Faça $f_{i_e, j_e} = \Sigma$.

Teorema 3.1 Uma matriz de incidência \mathbf{F}^* construída de \mathbf{F} , com $L(\mathbf{F}) \subset \Sigma^*$, gera a linguagem $L(\mathbf{F}^*) = \Sigma^*$.

Demonstração:

Considere uma matriz de incidência \mathbf{F} com linguagem $L(\mathbf{F}) \subset \Sigma^*$. A marcação de todos os estados garante $L(\mathbf{F}) = L_m(\mathbf{F})$. A inclusão dos auto-laços

$$f_{i,i} = \Sigma - \{\sigma^1, \sigma^2, \dots, \sigma^n\}$$

define que a matriz apresente a linguagem

$$L(\mathbf{F}') = L_m(\mathbf{F}') = \text{cat}_L(L_m(\mathbf{F}), (\Sigma - \{\sigma^1, \sigma^2, \dots, \sigma^n\})^*) \supset L_m(\mathbf{F}),$$

Ou seja, a linguagem da matriz \mathbf{F} se apresenta com todas as suas palavras concatenadas com todas as palavras da linguagem $(\Sigma - \{\sigma^1, \sigma^2, \dots, \sigma^n\})^*$. Por outro lado, para cada símbolo $\sigma^k, k = 1, 2, \dots, n$, não definido na linha i de \mathbf{F} , sua introdução no elemento f_{i,j_e} determina o reconhecimento da linguagem

$$L(\mathbf{F}') = L_m(\mathbf{F}') = \text{cat}_L(\text{cat}_L(L_m(\mathbf{F}), (\Sigma - \{\sigma^1, \sigma^2, \dots, \sigma^n\})^*), \sigma^k).$$

Isto é, todas as palavras da linguagem são iniciadas por qualquer símbolo, apresentando todas as seqüências de símbolos repetidos, com cardinalidade 2, que define que para qualquer seqüência, a linha i_e é sempre alcançável. Considerando a introdução do auto-laço Σ^* em f_{i_e, j_e} , a matriz se apresenta com a linguagem

$$L(\mathbf{F}') = L_m(\mathbf{F}') = \text{cat}_L(\text{cat}_L((L_m(\mathbf{F}) \cup (\Sigma - \{\sigma^1, \sigma^2, \dots, \sigma^n\})^*), (\sigma^1 + \sigma^2 + \dots + \sigma^n)), \Sigma^*).$$

Como a linha i_e é alcançável por qualquer seqüência, e nela há o auto-laço Σ^* , então,

$$L(\mathbf{F}') = L_m(\mathbf{F}') = \Sigma^* = L(\mathbf{F}^*) = L_m(\mathbf{F}^*).$$

◆

Exemplo 3.15 Dada a matriz de incidência sobre o alfabeto $\Sigma = \{\alpha, \beta, \eta, \lambda, \kappa, \mu\}$

$$\mathbf{F} = \begin{bmatrix} \epsilon & \alpha + \beta \\ \kappa & \eta + \lambda \end{bmatrix} \begin{matrix} e \\ e \end{matrix}$$

a utilização do algoritmo 3.2 constrói a matriz

$$\mathbf{F}^* = \begin{bmatrix} \mu & \alpha + \beta & \kappa + \eta + \lambda \\ \kappa & \mu + \eta + \lambda & \alpha + \beta \\ \epsilon & \epsilon & \alpha + \beta + \kappa + \mu + \eta + \lambda \end{bmatrix} \begin{matrix} e \\ e \\ e \end{matrix}$$

que reconhece Σ^* . Os autômatos que elas representam são vistos na Figura 3.7(a), 3.7(b), respectivamente.

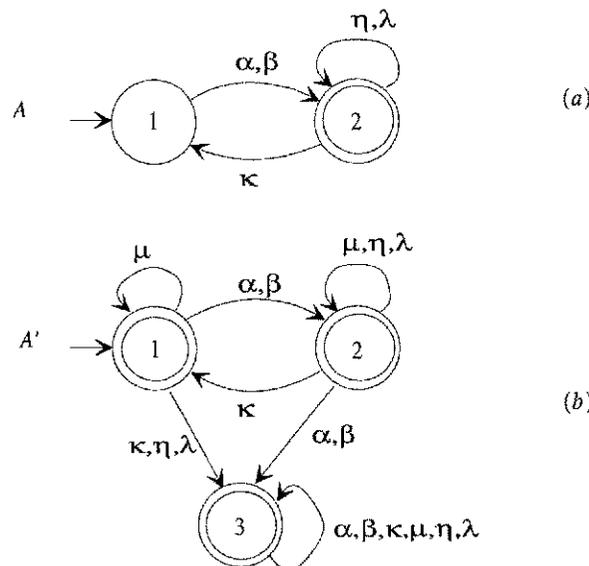


Figure 3.7: Autômatos (a) A e (b) A' construídos com o Algoritmo 3.2.

Com a transformação de uma matriz de incidência \mathbf{E} qualquer em uma matriz que gera Σ^* , através da inclusão do estado proibido, constróem-se $\mathbf{A}^\#$ e $\mathbf{E}^\#$, que satisfazem as condições $L(\mathbf{E}^\#) \subset L(\mathbf{A}^\#)$, $L(\mathbf{A}^\#) = L(\mathbf{A})$, $L(\mathbf{E}^\#) = L(\mathbf{E})$ e $L_m(\mathbf{E}^\#) = L_m(\mathbf{E})$. Essa transformação é feita através do seguinte algoritmo, em que se consideram as dimensões de \mathbf{E} e \mathbf{A} iguais a n e m , respectivamente.

Algoritmo 3.3 Transformação da matriz de incidência \mathbf{E} em $\mathbf{E}^\#$

1. Transforme \mathbf{E} em \mathbf{E}^* ;
2. Construa $\mathbf{A}^\# = \mathbf{A} \parallel \mathbf{E}^*$;
3. Para i e $j = 1$ até $n \times m$, faça:
 - a) Se $a_{i,j}^\# \neq \epsilon$, faça $e_{i,j}^\# = a_{i,j}^\#$.
4. Faça $k = 1$ até $n \times m$:
 - a) Se $\phi_k(\mathbf{A}^\#) \neq \epsilon$, faça $i_{\mathbf{E}} = ((k - (k \bmod m)) / m) + 1$, e $\phi_k(\mathbf{E}^\#) = \phi_k(\mathbf{E}^\#) \oplus \phi_{i_{\mathbf{E}}}(\mathbf{E})$.

Neste algoritmo, mod é o operador de módulo que retorna o resto da divisão k/m . O passo 4 resgata os estados marcados da especificação de comportamento \mathbf{E} , para $\mathbf{E}^\#$, de forma a garantir que $L_m(\mathbf{E}^\#) = L_m(\mathbf{E})$.

Teorema 3.2 Dadas as matrizes de incidência \mathbf{A} e \mathbf{E} , a transformação de \mathbf{A} em $\mathbf{A}^\#$ e \mathbf{E} em $\mathbf{E}^\#$ determina que $\mathbf{E}^\#$ é submatriz de $\mathbf{A}^\#$, e $L(\mathbf{E}^\#) \subset L(\mathbf{A}^\#)$, $L(\mathbf{E}^\#) = L(\mathbf{E})$ e $L_m(\mathbf{E}^\#) = L_m(\mathbf{E})$.

Demonstração:

Considere $L(\mathbf{E}) \subseteq \Sigma^*$. A transformação de \mathbf{E} em \mathbf{E}^* garante que $L(\mathbf{E}^*) = \Sigma^*$, pelo Teorema 3.1. A composição síncrona $\mathbf{A}^\# = \mathbf{A} \parallel \Sigma^* = \mathbf{A} \parallel \mathbf{E}^*$, garante que $L(\mathbf{A}) = L(\mathbf{A}^\#)$.

Sendo $e_{i,j}^\# = a_{i,j}^\#$, $\forall i, j$, $i, j \leq ((n-1) \times m)$, então $\mathbf{E}^\#$ é submatriz de $\mathbf{A}^\#$, e conseqüentemente, $L(\mathbf{E}^\#) \subset L(\mathbf{A}^\#)$.

Por outro lado, como todos os elementos $e_{i,j}^\# = \epsilon$, $\forall i, j$, $i, j > ((n-1) \times m)$, então as linhas/colunas de erro não se encontram em $\mathbf{E}^\#$, conseqüentemente, $L(\mathbf{E}^\#) = L(\mathbf{E})$.

Desde que $\phi_k(\mathbf{E}^\#) = \phi_k(\mathbf{E}^\#) \oplus \phi_{i_{\mathbf{E}}}(\mathbf{E})$, $\forall i_{\mathbf{E}} = ((k - (k \bmod m)) / m) + 1$, então $L_m(\mathbf{E}^\#) = L_m(\mathbf{E})$. \blacklozenge

O seguinte exemplo ilustra a transformação de \mathbf{A} em $\mathbf{A}^\#$ e \mathbf{E} em $\mathbf{E}^\#$, tal que $L(\mathbf{E}^\#) \subset L(\mathbf{A}^\#)$, $L(\mathbf{E}^\#) = L(\mathbf{E})$ e $L_m(\mathbf{E}^\#) = L_m(\mathbf{E})$, quando \mathbf{E} não é submatriz de \mathbf{A} .

Exemplo 3.16 Considere as matrizes

$$\mathbf{A} = \begin{bmatrix} \epsilon & \alpha_1 & \alpha_2 & \epsilon \\ \beta_1 & \epsilon & \epsilon & \alpha_2 \\ \beta_2 & \epsilon & \epsilon & \alpha_1 \\ \epsilon & \beta_2 & \beta_1 & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \\ \epsilon \end{matrix}$$

e

$$\mathbf{E} = \begin{bmatrix} \alpha_1 + \beta_2 & \beta_1 \\ \alpha_2 & \alpha_1 + \beta_2 \end{bmatrix} \begin{matrix} e \\ \epsilon \end{matrix}$$

Para construir a matriz $\mathbf{E}^\#$, tal que $L(\mathbf{E}^\#) \subset L(\mathbf{A}^\#) = L(\mathbf{A})$, constrói-se a matriz \mathbf{E}^* que gera Σ^* , a qual é dada por

$$\mathbf{E}^* = \begin{bmatrix} \alpha_1 + \beta_2 & \beta_1 & \alpha_2 \\ \alpha_2 & \alpha_1 + \beta_2 & \beta_1 \\ \epsilon & \epsilon & \alpha_1 + \beta_1 + \alpha_2 + \beta_2 \end{bmatrix} \begin{matrix} e \\ e \\ e \end{matrix}$$

onde a linha/coluna de erro é a terceira. Construindo o produto síncrono $\mathbf{A} \parallel \mathbf{E}^*$, tem-se

$$\mathbf{A}^\# = \begin{bmatrix} \epsilon & \alpha_1 & \epsilon & \alpha_2 & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \alpha_2 \\ \beta_2 & \epsilon & \epsilon & \alpha_1 & \epsilon \\ \epsilon & \beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & \beta_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \alpha_2 & \epsilon & \epsilon & \alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \alpha_2 & \epsilon & \epsilon & \epsilon & \epsilon & \beta_1 & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta_2 & \epsilon & \epsilon & \alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & \beta_1 & \epsilon \\ \epsilon & \alpha_1 & \alpha_2 & \epsilon \\ \epsilon & \beta_1 & \epsilon & \epsilon & \alpha_2 \\ \epsilon & \beta_2 & \epsilon & \epsilon & \alpha_1 \\ \epsilon & \beta_2 & \beta_1 & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \\ \epsilon \\ e \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \\ e \\ \epsilon \\ \epsilon \\ \epsilon \end{matrix}$$

em que $L(\mathbf{A}^\#) = L(\mathbf{A})$ e $L_m(\mathbf{A}^\#) = L_m(\mathbf{A})$, e as linhas/colunas formadas por uma linha/coluna de \mathbf{A} com a linha/coluna de erro de \mathbf{E}^* são as quatro últimas. Fazendo $e_{i,j}^\# = a_{i,j}^\#$, nos elementos $\forall i, j$, tal que $i, j \leq ((n-1) \times m)$ (similar a fazer $\mathbf{E}^\# = \mathbf{A}^\#$ e tornar

um evento é definido do estado i para o estado j , $s_{i,j} = \bigoplus_k \sigma^k$, tal que qualquer um evento $\sigma^k \in \Sigma$ ocorrendo, provoca no autômato da composição síncrona $\mathbf{S}||\mathbf{A}$, a mudança do estado i para o estado j .

Com essa definição, um supervisor deve gerar $L(\mathbf{S}) \subseteq L(\mathbf{A})$, a partir de uma especificação de comportamento.

Define-se para o supervisor, o conjunto de entradas de controle, como a seguir:

Definição 3.22 Para um supervisor definido por uma matriz de incidência \mathbf{S} , define-se o conjunto de entradas de controle como

$$\Gamma = \{\gamma_1, \gamma_2, \dots\}, \quad (3.14)$$

onde $\Sigma_{uc} \subseteq \gamma_i \subseteq \Sigma$, é definido como um conjunto de eventos que contém todos os eventos não controláveis e os eventos possíveis de ocorrerem na linha $i = 1, 2, \dots$ da matriz do supervisor.

Observação: A dinâmica do SED supervisionado é determinada como segue: Estando na linha 1 (tanto a matriz do SED como do supervisor), o supervisor define quais eventos podem ocorrer através de γ_1 , dentre os eventos habilitados no SED ($a_{i,j} = \sigma \in \Sigma(1) \subseteq \Sigma$). A ocorrência de um evento $a_{i,j} = \sigma \in \Sigma(1) \subseteq \Sigma$ é observado no supervisor. Este, por sua vez, avalia em sua linha j , quais eventos podem ocorrer. Isto é, avalia γ_j , e determina a ação de controle a ser aplicada no SED. Essa ação de controle inibe eventos controláveis que levam à ocorrência de eventos não desejados para que a matriz de incidência do SED siga a especificação de comportamento requerida.

Para sintetizar um supervisor \mathbf{S} para um SED modelado por uma matriz de incidência \mathbf{A} , sendo definida uma especificação de comportamento \mathbf{E} , as seguintes definições são necessárias:

Definição 3.23 O operador *ACES* é definido como

$$ACES(\mathbf{A}) = \mathbf{B}, b_{i,j} = \begin{cases} a_{i,j} & \text{se } i \text{ é acessível} \\ \epsilon & \text{caso contrário} \end{cases}$$

onde \mathbf{A} e \mathbf{B} são matrizes de incidência.

A operação $ACES(\mathbf{A})$ elimina os elementos $a_{i,j}$ de uma linha i não acessível. O seguinte algoritmo constrói a operação *ACES*, em que \mathbf{F} representa a matriz que está sendo avaliada:

Algoritmo 3.4 Operador *ACES*

1. Crie um vetor $vac_{1 \times N}$ de estados acessíveis.

a) Faça i e $j = 1$ até N

i. Para $i = 1$, se $\mathbf{F}(i, j) \neq \epsilon$, faça $vac(j) = 1$.

ii. Para $i > 1$, se $\mathbf{F}(i, j) \neq \epsilon$ e $\text{vac}(i) = 1$, então faça $\text{vac}(j) = 1$.

b) Faça $i = N - 1$ até 1 e $j = 1$ até N

i. Se $\mathbf{F}(i, j) \neq \epsilon$ e $\text{vac}(i) = 1$, então faça $\text{vac}(j) = 1$.

2. Faça para $i = 1$ até N

a) Se $\text{vac}(i) = 0$, faça $\mathbf{F}(i, j) = \epsilon$, para $j = 1$ até N .

A execução desse algoritmo é vista a seguir:

Considere que $\forall j, \text{vac}(j) = 0$. Considere $i = 1$. Se para algum $j, \mathbf{F}(1, j) \neq \epsilon$, então $\text{vac}(j) = 1$. Isso determina quais linhas são acessíveis diretamente da linha 1.

Para $i > 1$, se $\text{vac}(i) = 1$ e para algum $j, \mathbf{F}(i, j) \neq \epsilon$, então $\text{vac}(j) = 1$. Isso determina as linhas acessíveis a partir das linhas acessíveis anteriores a linha i .

Fazendo $i = N - 1$ até 1, se para algum $j, \mathbf{F}(i, j) \neq \epsilon$ e $\text{vac}(i) = 1$, então $\text{vac}(j) = 1$. Isso determina as linhas acessíveis diretamente da linha i , se ela é acessível.

As linhas i onde $\text{vac}(i) = 0$ são eliminadas com a substituição de todos os elementos por ϵ . ◆

Definição 3.24 O operador *COACES* é definido como

$$\text{COACES}(\mathbf{A}) = \mathbf{B}, b_{i,j} = \begin{cases} a_{i,j} & \text{se } \exists s | s = a_{i,j_1} a_{j_1,j_2} \dots a_{j_{n-1},j_n}, a_{i,j_1}, a_{j_1,j_2}, \dots, a_{j_{n-1},j_n} \neq \epsilon \\ & \text{e } \phi_{j_n}(\mathbf{A}) = e; \\ \epsilon & \text{caso contrário.} \end{cases}$$

A operação *COACES*(\mathbf{A}) elimina os elementos que levam para linhas não coacessíveis, tornando-as não acessíveis. A operação *COACES* é realizada pelo seguinte algoritmo, em que \mathbf{F} representa a matriz que está sendo avaliada:

Algoritmo 3.5 Operador *COACES*

1. Crie um vetor $\text{vco}_{N \times 1}$ de estados coacessíveis.

a) Faça j e $i = 1$ até N

i. Para $i = 1$, se $\mathbf{F}(i, j) \neq \epsilon$ e $\phi_j(\mathbf{F}) = e$, faça $\text{vco}(i) = 1$.

ii. Para $i > 1$, se $\mathbf{F}(i, j) \neq \epsilon$ e ($\phi_j(\mathbf{F}) = e$ ou $\text{vco}(j) = 1$), então faça $\text{vco}(i) = 1$.

b) Faça $j = 1$ até N e $i = N - 1$ até 1

i. Se $\mathbf{F}(i, j) \neq \epsilon$ e ($\phi_j(\mathbf{F}) = e$ ou $\text{vco}(j) = 1$), então faça $\text{vco}(i) = 1$.

2. Faça para $j = 1$ até N

a) Se $\text{vco}(j) = 0$, faça $\mathbf{F}(i, j) = \epsilon$, para $i = 1$ até N .

A execução deste algoritmo é vista a seguir:

Considere que $\forall i, vco(i) = 0$. Considere $i = 1$. Se para algum $j, \mathbf{F}(1, j) \neq \epsilon$, e $\phi_j(\mathbf{F}) = e$, então $vco(1) = 1$. Isso determina se a linha 1 é coaccessível.

Para $i > 1$, se $\mathbf{F}(i, j) \neq \epsilon$ e $\phi_j(\mathbf{F}) = e$, então $vco(i) = 1$. Contudo, se $\phi_j(\mathbf{F}) = \epsilon$, mas a linha j é coaccessível, então $vco(i) = 1$. Isso determina as linhas coaccessíveis a partir das linhas coaccessíveis anteriores a linha i .

Fazendo $i = N - 1$ até 1, se para algum $j, \mathbf{F}(i, j) \neq \epsilon$ e $\phi_j(\mathbf{F}) = e$, então $vco(i) = 1$. Por outro lado, se $\phi_j(\mathbf{F}) = \epsilon$, mas a linha j é coaccessível, então $vco(i) = 1$. Isso determina as linhas coaccessíveis diretamente da linha i , se ela é coaccessível.

As colunas j onde $vco(j) = 0$ são eliminadas com a substituição de todos os elementos por ϵ . ♦

Exemplo 3.17 Dada a matriz de incidência

$$\mathbf{A} = \begin{bmatrix} \alpha & \beta & \epsilon \\ \alpha & \beta & \epsilon \\ \alpha & \epsilon & \beta \end{bmatrix} \begin{matrix} \epsilon \\ e \\ e \end{matrix}$$

tem-se que

$$ACES(\mathbf{A}) = ACES \left(\begin{bmatrix} \alpha & \beta & \epsilon \\ \alpha & \beta & \epsilon \\ \alpha & \epsilon & \beta \end{bmatrix} \begin{matrix} \epsilon \\ e \\ e \end{matrix} \right) = \begin{bmatrix} \alpha & \beta & \epsilon \\ \alpha & \beta & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix} \begin{matrix} \epsilon \\ e \\ \epsilon \end{matrix}$$

e

$$COACES(\mathbf{A}) = COACES \left(\begin{bmatrix} \alpha & \beta & \epsilon \\ \alpha & \beta & \epsilon \\ \alpha & \epsilon & \beta \end{bmatrix} \begin{matrix} \epsilon \\ e \\ e \end{matrix} \right) = \begin{bmatrix} \alpha & \beta & \epsilon \\ \alpha & \beta & \epsilon \\ \alpha & \epsilon & \beta \end{bmatrix} \begin{matrix} \epsilon \\ e \\ \epsilon \end{matrix} = \mathbf{A}.$$

Contudo, para a matriz de incidência

$$\mathbf{A} = \begin{bmatrix} \alpha & \beta & \beta \\ \alpha & \beta & \epsilon \\ \epsilon & \epsilon & \alpha \end{bmatrix} \begin{matrix} \epsilon \\ e \\ \epsilon \end{matrix}$$

tem-se que

$$ACES(\mathbf{A}) = ACES \left(\begin{bmatrix} \alpha & \beta & \beta \\ \alpha & \beta & \epsilon \\ \epsilon & \epsilon & \alpha \end{bmatrix} \begin{matrix} \epsilon \\ e \\ \epsilon \end{matrix} \right) = \begin{bmatrix} \alpha & \beta & \beta \\ \alpha & \beta & \epsilon \\ \epsilon & \epsilon & \alpha \end{bmatrix} \begin{matrix} \epsilon \\ e \\ \epsilon \end{matrix} = \mathbf{A}$$

e

$$COACES(\mathbf{A}) = COACES \left(\begin{bmatrix} \alpha & \beta & \beta \\ \alpha & \beta & \epsilon \\ \epsilon & \epsilon & \alpha \end{bmatrix} \begin{matrix} \epsilon \\ e \\ \epsilon \end{matrix} \right) = \begin{bmatrix} \alpha & \beta & \epsilon \\ \alpha & \beta & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix} \begin{matrix} \epsilon \\ e \\ \epsilon \end{matrix}.$$

Com as Definições 3.23 e 3.24, também define-se o operador *TRIM*, como a seguir.

Definição 3.25 O operador *TRIM* é definido por

$$TRIM(\mathbf{A}) = ACES(COACES(\mathbf{A})) = \mathbf{B}.$$

O operador *TRIM* aplicado a uma matriz de incidência \mathbf{A} devolve uma matriz \mathbf{B} que é acessível e coacessível.

Finalmente, define-se um operador que compara duas matrizes de incidência (uma submatriz da outra), onde através desse operador, é possível determinar se a linguagem de uma matriz está contida na linguagem da outra.

Definição 3.26 Dadas duas matrizes $\mathbf{A} = [a_{i,j}]$ e $\mathbf{B} = [b_{i,j}]$, define-se o operador \leq , por

$$\mathbf{A} \leq \mathbf{B} \Leftrightarrow L(\mathbf{A}) \subseteq L(\mathbf{B}).$$

Igualmente, definem-se os operadores \supseteq , \triangleleft e \triangleright como

$$\mathbf{A} \supseteq \mathbf{B} \Leftrightarrow L(\mathbf{A}) \supseteq L(\mathbf{B}),$$

$$\mathbf{A} \triangleleft \mathbf{B} \Leftrightarrow L(\mathbf{A}) \subset L(\mathbf{B}) \text{ e}$$

$$\mathbf{A} \triangleright \mathbf{B} \Leftrightarrow L(\mathbf{A}) \supset L(\mathbf{B}).$$

Considerando que as matrizes se apresentam sempre com as mesmas dimensões e que essas matrizes tenham todos os elementos correlacionados (que é o caso tratado nesse trabalho, desde que sempre devem-se transformar as matrizes \mathbf{E} em $\mathbf{E}^\#$ e \mathbf{A} em $\mathbf{A}^\#$, tal que $\mathbf{E}^\#$ seja submatriz de $\mathbf{A}^\#$), para determinar se $\mathbf{A} \leq \mathbf{B}$ utiliza-se o seguinte algoritmo, em que se considera $\sigma \in \Sigma^*$ (expressões regulares do tipo $\sigma^1 + \dots + \sigma^n$):

Algoritmo 3.6 Verificação de $\mathbf{A} \leq \mathbf{B}$

1. Para cada elemento $a_{i,j}$ faça:

i. Se $a_{i,j} = \sigma$, e $\sigma \notin \mathbf{B}$, então $\mathbf{A} \not\leq \mathbf{B}$;

ii. Se $a_{i,j} = \sigma^i + \dots + \sigma^k \vee a_{i,j} = \epsilon$, para $\sigma^i, \dots, \sigma^k \subset \sigma$ e $b_{i,j} = \sigma$, então $\mathbf{A} \leq \mathbf{B}$;

2. Faça $\forall i$, se $\phi_i(\mathbf{A}) = e \vee \phi_i(\mathbf{A}) = \epsilon$, para $\phi_i(\mathbf{B}) = e$, e $\phi_i(\mathbf{A}) = \epsilon$, $\forall \phi_i(\mathbf{B}) = \epsilon$, então $\mathbf{A} \leq \mathbf{B}$.

A execução desse algoritmo determina se uma matriz \mathbf{A} está contida em uma matriz \mathbf{B} , como visto a seguir:

Considere duas matrizes \mathbf{A} e \mathbf{B} construídas com símbolos de Σ . Naturalmente, vê-se que a existência de um símbolo σ em \mathbf{A} que não existe em \mathbf{B} , define que a linguagem $L(\mathbf{A})$ contém palavras que não existem em $L(\mathbf{B})$. Logo, $L(\mathbf{A}) \supset L(\mathbf{B})$ e, conseqüentemente, $\mathbf{A} \not\leq \mathbf{B}$.

Por outro lado, considerando que para todo elemento de \mathbf{B} , $b_{i,j} \neq \epsilon$, se $a_{i,j} = b_{i,j}$, todas as palavras de \mathbf{A} também existem em \mathbf{B} . Logo, $L(\mathbf{A}) = L(\mathbf{B})$ e, conseqüentemente, $\mathbf{A} = \mathbf{B}$. Se

para qualquer elemento de \mathbf{B} , $b_{i,j} \neq \epsilon$, se $a_{i,j} \subset b_{i,j}$, apenas parte das palavras de \mathbf{B} existem em \mathbf{A} . Logo, $L(\mathbf{A}) \subset L(\mathbf{B})$ e, conseqüentemente, $\mathbf{A} \triangleleft \mathbf{B}$.

Considerando $i_{\mathbf{A}} = i_{\mathbf{B}} = i$, se $\phi_i(\mathbf{B}) = e$ e $\phi_i(\mathbf{A}) = e \vee \phi_i(\mathbf{A}) = \epsilon$, e $\forall \phi_i(\mathbf{B}) = \epsilon$, $\phi_i(\mathbf{A}) = \epsilon$, então todas ou parte das palavras reconhecidas de \mathbf{A} são reconhecidas por \mathbf{B} . Assim, $L_m(\mathbf{A}) \subseteq L_m(\mathbf{B})$ e, conseqüentemente, $\mathbf{A} \trianglelefteq \mathbf{B}$. Caso $\exists \phi_i(\mathbf{A}) = e$, para $\phi_i(\mathbf{B}) = \epsilon$, então algumas palavras de \mathbf{A} não são reconhecidas por \mathbf{B} e dessa forma, $L_m(\mathbf{A}) \not\subseteq L_m(\mathbf{B})$ e, conseqüentemente, $\mathbf{A} \not\trianglelefteq \mathbf{B}$. \blacklozenge

Deve-se observar que uma linguagem $L_1 \subset L_2$, se L_1 contiver algumas seqüências de L_2 .

Exemplo 3.18 Dadas as matrizes de incidência

$$\mathbf{A}_1 = \begin{bmatrix} \epsilon & \alpha \\ \beta & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \end{matrix} \quad \mathbf{A}_2 = \begin{bmatrix} \epsilon & \alpha + \beta + \lambda & \epsilon \\ \beta & \epsilon & \lambda \\ \mu & \epsilon & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \end{matrix}$$

para testar se $\mathbf{A}_2 \trianglelefteq \mathbf{A}_1$ considera-se a matriz \mathbf{A}_1 como tendo a mesma dimensão de \mathbf{A}_2 , com os elementos das últimas linhas/colunas incluídas para completar a igualdade das dimensões, definidos como ϵ . Assim, a matriz \mathbf{A}_1 , pode ser vista como

$$\mathbf{A}_1 = \begin{bmatrix} \epsilon & \alpha & \epsilon \\ \beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \end{matrix}$$

Utilizando o Algoritmo 3.6, encontra-se que $\mathbf{A}_1 \trianglelefteq \mathbf{A}_2$, pois não há nenhum elemento em \mathbf{A}_1 que não exista em \mathbf{A}_2 , e todos os elementos de \mathbf{A}_1 se encontram na matriz \mathbf{A}_2 , nas respectivas posições. Observe que $L(\mathbf{A}_1) = (\alpha\beta)^* \alpha$ é encontrada em $L(\mathbf{A}_2)$, cuja matriz contém os elementos de \mathbf{A}_1 . Isto é, $L(\mathbf{A}_1) \subset L(\mathbf{A}_2)$. Da mesma forma, pode-se ver que $L_m(\mathbf{A}_1) \subset L_m(\mathbf{A}_2)$.

Para uma especificação de comportamento \mathbf{E} , a seguinte condição deve ser satisfeita:

Definição 3.27 Uma especificação de comportamento \mathbf{E} é dita ser válida para a matriz de incidência \mathbf{A} se $\mathbf{E} \neq [\epsilon]$ e se $\forall i, j, \exists \sigma \subset e_{i,j} | \sigma \in \Sigma$, em que $[\epsilon]$ é a matriz nula onde todos os seus elementos são ϵ .

Assim, tendo uma especificação satisfazendo a Definição 3.27, e com a utilização dos operadores das Definições 3.17, 3.23, 3.24, 3.25 e 3.26 a controlabilidade de \mathbf{E} é definida como:

Definição 3.28 Dada uma especificação de comportamento \mathbf{E} válida, com $ACES(\mathbf{E}) = \mathbf{E}$, e a matriz de incidência \mathbf{A} que representa o autômato G , \mathbf{E} é controlável se

$$ACES(\mathbf{E} \oplus \mathbf{A}_{uc}) = \mathbf{E}. \quad (3.15)$$

Essa definição decorre da Definição 2.32. Neste caso, tem-se

$$\overline{L(\mathbf{E})\Sigma_{uc}} \cap L(\mathbf{A}) \subseteq \overline{L(\mathbf{E})},$$

que é equivalente a

$$L(\mathbf{E})\Sigma_{uc} \cap L(\mathbf{A}) \subseteq L(\mathbf{E}).$$

Considerando $[\Sigma_{uc}]$ como sendo uma matriz em que seus elementos é a soma de todos os eventos não controláveis, a concatenação de elementos σ_{uc} nas palavras de $L(\mathbf{E})$ é semelhante a fazer $\mathbf{E} \oplus [\Sigma_{uc}]$. Essa soma determina uma matriz onde cada elemento (i, j) contém todos os elementos não controláveis, o que significa que toda palavra da linguagem de \mathbf{E} é seguida pelos eventos não controláveis. Utilizando o operador de interseção \otimes e o operador *ACES*, a condição de controlabilidade é vista em termos das matrizes de incidência como:

$$ACES((\mathbf{E} \oplus [\Sigma_{uc}]) \otimes \mathbf{A}) \trianglelefteq \mathbf{E}. \quad (3.16)$$

Porém, como

$$\begin{aligned} \mathbf{E} \oplus [\Sigma_{uc}] &= (\mathbf{E}_c \oplus \mathbf{E}_{uc}) \oplus [\Sigma_{uc}] \\ &= (\mathbf{E}_c \oplus [\Sigma_{uc}]) \oplus (\mathbf{E}_{uc} \oplus [\Sigma_{uc}]) \\ &= (\mathbf{E}_c \oplus [\Sigma_{uc}]) \oplus [\Sigma_{uc}] \\ &= \mathbf{E}_c \oplus [\Sigma_{uc}] \end{aligned}$$

e

$$\begin{aligned} (\mathbf{E}_c \oplus [\Sigma_{uc}]) \otimes (\mathbf{A}_c \oplus \mathbf{A}_{uc}) &= (\mathbf{E}_c \otimes \mathbf{A}_c) \oplus (\mathbf{E}_c \otimes \mathbf{A}_{uc}) \oplus ([\Sigma_{uc}] \otimes \mathbf{A}_c) \oplus ([\Sigma_{uc}] \otimes \mathbf{A}_{uc}) \\ &= \mathbf{E}_c \oplus [\epsilon] \oplus [\epsilon] \oplus \mathbf{A}_{uc} \\ &= \mathbf{E}_c \oplus \mathbf{A}_{uc} \end{aligned}$$

e considerando que $\mathbf{A}_{uc} \supseteq \mathbf{E}_{uc}$, então $\mathbf{A}_{uc} = \mathbf{A}_{uc} \oplus \mathbf{E}_{uc}$ e

$$\begin{aligned} (\mathbf{E}_c \oplus [\Sigma_{uc}]) \otimes (\mathbf{A}_c \oplus \mathbf{A}_{uc}) &= \mathbf{E}_c \oplus \mathbf{E}_{uc} \oplus \mathbf{A}_{uc} \\ &= \mathbf{E} \oplus \mathbf{A}_{uc}. \end{aligned}$$

Dessa forma, a equação (3.16) pode ser reescrita como

$$ACES(\mathbf{E} \oplus \mathbf{A}_{uc}) \trianglelefteq \mathbf{E}.$$

Por outro lado, considerando que $\mathbf{E} \triangleleft \mathbf{A}$ e que, no máximo $\mathbf{E}_{uc} = \mathbf{A}_{uc}$, tem-se então

$$ACES(\mathbf{E} \oplus \mathbf{A}_{uc}) = ACES(\mathbf{E} \oplus \mathbf{E}_{uc}) = \mathbf{E}.$$

Logo, a condição de controlabilidade para o caso tratado nessa Tese pode ser escrita como na Definição 3.28.

Deve-se observar que a condição de controlabilidade de \mathbf{E} é feita sobre as matrizes que satisfazem $\mathbf{E} \trianglelefteq \mathbf{A}$. Dessa forma, quando \mathbf{E} não é uma submatriz de \mathbf{A} , o procedimento de transformar \mathbf{E} em $\mathbf{E}^\#$ e \mathbf{A} em $\mathbf{A}^\#$ deve ser utilizado para permitir a utilização da Definição 3.28.

Exemplo 3.19 Considere as matrizes de incidência dadas por:

$$\mathbf{A} = \begin{bmatrix} \epsilon & \alpha & \epsilon \\ \beta & \epsilon & \kappa \\ \eta & \epsilon & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \end{matrix}, \mathbf{E} = \begin{bmatrix} \epsilon & \alpha & \epsilon \\ \beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \end{matrix}$$

Sendo $\Sigma_{uc} = \{\eta\}$, tem-se que

$$\begin{aligned} ACES \left(\begin{bmatrix} \epsilon & \alpha & \epsilon \\ \beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \end{matrix} \oplus \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ \eta & \epsilon & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \end{matrix} \right) = \\ ACES \left(\begin{bmatrix} \epsilon & \alpha & \epsilon \\ \beta & \epsilon & \epsilon \\ \eta & \epsilon & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \end{matrix} \right) = \begin{bmatrix} \epsilon & \alpha & \epsilon \\ \beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \end{matrix} = \mathbf{E}. \end{aligned}$$

Isto é, a especificação de comportamento \mathbf{E} gera uma linguagem controlável.

Considerando uma especificação de comportamento \mathbf{E} válida, e a matriz de incidência \mathbf{A} que modela o SED, a partir da condição de controlabilidade (Definição 3.28), um supervisor pode ser sintetizado.

Lema 3.1 Um supervisor \mathbf{S} para uma matriz de incidência \mathbf{A} é definido pela especificação de comportamento \mathbf{E} válida se e somente se

$$ACES(\mathbf{E} \oplus \mathbf{A}_{uc}) = \mathbf{E}.$$

Demonstração:

(i) Considerando que \mathbf{E} é válida e controlável, então

$$ACES(\mathbf{E} \oplus \mathbf{A}_{uc}) = \mathbf{E}$$

é satisfeito.

(ii) Considere que \mathbf{E} contém todos os elementos de \mathbf{A}_{uc} . Então,

$$\mathbf{E} \oplus \mathbf{A}_{uc} = \mathbf{E}_c \oplus \mathbf{E}_{uc} \oplus \mathbf{A}_{uc} = \mathbf{E}_c \oplus \mathbf{A}_{uc} = \mathbf{E}_c \oplus \mathbf{E}_{uc} = \mathbf{E}. \quad (3.17)$$

Logo

$$ACES(\mathbf{E} \oplus \mathbf{A}_{uc}) = ACES(\mathbf{E}) = \mathbf{E}.$$

(iii) Agora, considere que \mathbf{E} não contém todos os elementos de \mathbf{A}_{uc} . Então,

$$\mathbf{E} \oplus \mathbf{A}_{uc} = \mathbf{E}_c \oplus \mathbf{E}_{uc} \oplus \mathbf{A}_{uc} = \mathbf{E}_c \oplus \mathbf{A}_{uc}. \quad (3.18)$$

Considerando que $\forall i, a_{i,j} = \sigma_c$ e $e_{i,j} = \epsilon$, onde $a_{j,k} = \sigma_{uc}$ e $e_{j,k} = \epsilon$, então

$$ACES(\mathbf{E}_c \oplus \mathbf{A}_{uc}) = ACES(\mathbf{E} \oplus \mathbf{A}_{uc}) = \mathbf{E}. \quad (3.19)$$

Logo, \mathbf{E} é uma solução não restritiva, podendo \mathbf{S} ser construído diretamente de \mathbf{E} .

- iv) Sendo $\mathbf{E} \not\leq \mathbf{A}$, a transformação de \mathbf{E} em $\mathbf{E}^\#$ e \mathbf{A} em $\mathbf{A}^\#$, de tal forma que $\mathbf{E}^\# \leq \mathbf{A}^\#$ garante as mesmas condições de (i), (ii) e (iii). \blacklozenge

A condição de existência do supervisor \mathbf{S} é que a matriz da especificação de comportamento \mathbf{E} represente uma submatriz de \mathbf{A} que seja controlável. Se a especificação \mathbf{E} define uma linguagem que satisfaz a condição de controlabilidade, o supervisor \mathbf{S} é a componente *trim* da própria especificação.

Corolário 3.1 Dada uma especificação de comportamento \mathbf{E} válida, e uma matriz de incidência trim \mathbf{A} que é o modelo do SED, $\mathbf{S} = \text{TRIM}(\mathbf{E})$ se e somente se

$$\text{ACES}(\mathbf{E} \oplus \mathbf{A}_{uc}) = \mathbf{E}.$$

Demonstração:

- (i) Se $\mathbf{S} = \text{TRIM}(\mathbf{E})$, \mathbf{E} é válida, o que garante diretamente os resultados da Demonstração do Lema 3.1.

- (ii) Se

$$\text{ACES}(\mathbf{E} \oplus \mathbf{A}_{uc}) = \mathbf{E},$$

\mathbf{E} é controlável, sendo uma solução não restritiva para \mathbf{S} . Considerando que \mathbf{E} contenha alguma linha não coaccessível, a operação $\text{TRIM}(\mathbf{E})$ elimina essa linha. Dessa forma, tem-se que $\mathbf{S} = \text{TRIM}(\mathbf{E})$. \blacklozenge

Se o Lema 3.1 é satisfeito, então a componente *trim* de \mathbf{E} é a matriz que representa o supervisor.

Exemplo 3.20 Para o autômato mostrado na Figura 3.8, onde $\Sigma = \{\alpha, \beta, \kappa, \mu\}$, $\Sigma_{uc} = \{\kappa\}$ e $\Sigma_c = \{\alpha, \beta, \mu\}$, a matriz de incidência \mathbf{A} é dada por

$$\mathbf{A} = \begin{bmatrix} \alpha & \beta & \mu \\ \alpha & \epsilon & \mu \\ \epsilon & \mu & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ e \end{matrix}$$

Para a especificação

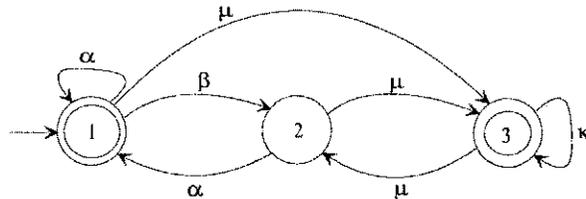


Figure 3.8: Autômato para exemplificar o Lema 3.1. e o Corolário 3.1.

$$\mathbf{E} = \begin{bmatrix} \epsilon & \beta & \epsilon \\ \epsilon & \epsilon & \mu \\ \epsilon & \epsilon & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ e \end{matrix}$$

mostrada na Figura 3.9(a), o supervisor é definido por $\mathbf{S} = \text{TRIM}(\mathbf{E}) = \mathbf{E}$, pois

$$\begin{aligned} & \text{ACES} \left(\begin{bmatrix} \epsilon & \beta & \epsilon \\ \epsilon & \epsilon & \mu \\ \epsilon & \epsilon & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ e \end{matrix} \oplus \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ e \end{matrix} \right) = \\ & \text{ACES} \left(\begin{bmatrix} \epsilon & \beta & \epsilon \\ \epsilon & \epsilon & \mu \\ \epsilon & \epsilon & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ e \end{matrix} \right) = \begin{bmatrix} \epsilon & \beta & \epsilon \\ \epsilon & \epsilon & \mu \\ \epsilon & \epsilon & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ e \end{matrix} = \mathbf{E}, \end{aligned}$$

que satisfaz o Lema 3.1. Por outro lado, considerando a especificação de comportamento

$$\mathbf{E} = \begin{bmatrix} \alpha & \beta & \mu \\ \epsilon & \epsilon & \mu \\ \epsilon & \mu & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \end{matrix}$$

mostrada na Figura 3.9(b), tem-se que

$$\begin{aligned} & \text{ACES} \left(\begin{bmatrix} \alpha & \beta & \mu \\ \epsilon & \epsilon & \mu \\ \epsilon & \mu & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \end{matrix} \oplus \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ e \end{matrix} \right) = \\ & \text{ACES} \left(\begin{bmatrix} \alpha & \beta & \mu \\ \epsilon & \epsilon & \mu \\ \epsilon & \mu & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \end{matrix} \right) = \begin{bmatrix} \alpha & \beta & \mu \\ \epsilon & \epsilon & \mu \\ \epsilon & \mu & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \end{matrix} = \mathbf{E}, \end{aligned}$$

em que \mathbf{E} é controlável. Contudo, \mathbf{E} não é trim (veja que ao sair da linha 1, a matriz não mais retorna para ela, que é a única marcada - $\phi_1(\mathbf{E}) = e$). Assim, o supervisor é dado por

$$\mathbf{S} = \text{TRIM}(\mathbf{E}) = \text{TRIM} \left(\begin{bmatrix} \alpha & \beta & \mu \\ \epsilon & \epsilon & \mu \\ \epsilon & \mu & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \end{matrix} \right) = \begin{bmatrix} \alpha & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \end{matrix}.$$

Por outro lado, considerando a especificação

$$\mathbf{E} = \begin{bmatrix} \epsilon & \beta & \mu \\ \alpha & \epsilon & \mu \\ \epsilon & \mu & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ e \end{matrix}$$

mostrada na Figura 3.9(c), tem-se que

$$\begin{aligned}
 ACES \left(\begin{array}{ccc|c} \epsilon & \beta & \mu & e \\ \alpha & \epsilon & \mu & \epsilon \\ \epsilon & \mu & \epsilon & e \end{array} \right) \oplus \left(\begin{array}{ccc|c} \epsilon & \epsilon & \epsilon & e \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \kappa & e \end{array} \right) = \\
 ACES \left(\begin{array}{ccc|c} \epsilon & \beta & \mu & e \\ \alpha & \epsilon & \mu & \epsilon \\ \epsilon & \mu & \kappa & e \end{array} \right) = \left[\begin{array}{ccc|c} \epsilon & \beta & \mu & e \\ \alpha & \epsilon & \mu & \epsilon \\ \epsilon & \mu & \kappa & e \end{array} \right] \triangleright \mathbf{E},
 \end{aligned}$$

o que não satisfaz o Lema 3.1.

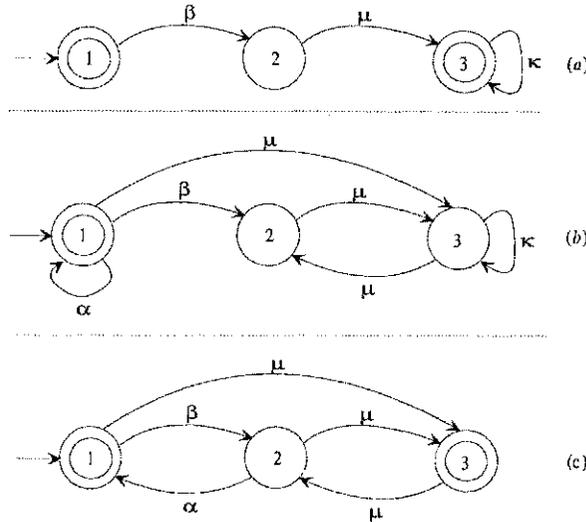


Figure 3.9: (a) e (b) Especificações com linguagens controláveis e (c) Especificação com linguagem não controlável.

Quando a especificação de comportamento não é factível por atingir estados onde eventos não controláveis que não pertencem à especificação podem ocorrer, é necessário encontrar um supervisor que realize o comportamento requerido, restritivamente. Isso quer dizer que se para uma especificação de comportamento válida, $ACES(\mathbf{E} \oplus \mathbf{A}_{uc}) \triangleright \mathbf{E}$, é necessário encontrar a suprema sublinguagem controlável ($\text{sup}C(L)$). Para isto, utiliza-se a Definição 3.9 de matriz de caminhos para encontrar os caminhos que podem ser eliminados para impedir que os estados não desejados sejam alcançados.

Considerando que os eventos da matriz de incidência pertencem ao conjunto $\Sigma = \Sigma_c \cup \Sigma_{uc}$, a matriz \mathbf{A}^n contém palavras de comprimento n , compostas de eventos controláveis e não controláveis. Então, utilizando-se dessa definição, é preciso avaliar quais os caminhos que iniciam com um evento qualquer (controlável ou não controlável), e que seja seguido apenas por eventos controláveis ou por eventos não controláveis.

Definição 3.29 *Seja $\Sigma = \Sigma_c \cup \Sigma_{uc}$, e seja um autômato G construído com eventos de Σ . A matriz de caminhos \mathbf{C}_c^n , que define as palavras que mudam o estado do autômato de i para*

o estado j , iniciados por um evento qualquer (controlável ou não controlável), e seguidos sempre de eventos controláveis, é definida por

$$\mathbf{C}_c^n = \mathbf{A} \otimes (\mathbf{A}_c)^{n-1},$$

com $\theta(\mathbf{C}_c^n) = \theta(\mathbf{A})$ e $\phi(\mathbf{C}_c^n) = \phi(\mathbf{A})$.

De acordo com essa definição, cada elemento de \mathbf{C}_c^n contém seqüências iniciadas por um evento qualquer, seguidos de $n - 1$ eventos controláveis. De forma semelhante, define-se:

Definição 3.30 Seja $\Sigma = \Sigma_c \cup \Sigma_{uc}$, e seja um autômato G construído com símbolos de Σ . A matriz de caminhos \mathbf{C}_{uc}^n , que define as palavras que mudam o estado do autômato de i para o estado j , iniciados por um evento qualquer (controlável ou não controlável), e seguidos sempre de eventos não controláveis, é definida por

$$\mathbf{C}_{uc}^n = \mathbf{A} \otimes (\mathbf{A}_{uc})^{n-1}.$$

com $\theta(\mathbf{C}_{uc}^n) = \theta(\mathbf{A})$ e $\phi(\mathbf{C}_{uc}^n) = \phi(\mathbf{A})$.

Observação: Na matriz de caminhos, um determinado evento $\sigma_{uc} \subset (a_{uc})_{i,j}$, aparece como último evento nas seqüências dos termos de $(c_{uc}^n)_{k,j}$. Estes termos de \mathbf{C}_{uc}^n são antecidos pelos elementos de $(c_{uc}^{n-1})_{k,i}$.

Exemplo 3.21 O autômato visto na Figura 3.10, com $\Sigma_c = \{\alpha, \kappa\}$ e $\Sigma_{uc} = \{\beta, \eta\}$, tem

$$\mathbf{A} = \begin{bmatrix} \epsilon & \beta & \epsilon & \epsilon & \kappa \\ \eta & \epsilon & \alpha & \epsilon & \epsilon \\ \alpha + \kappa & \epsilon & \epsilon & \eta & \epsilon \\ \eta & \beta & \epsilon & \alpha & \epsilon \\ \kappa & \epsilon & \eta & \epsilon & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \\ e \\ \epsilon \end{matrix}$$

e

$$\mathbf{A}_{uc} = \begin{bmatrix} \epsilon & \beta & \epsilon & \epsilon & \epsilon \\ \eta & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \eta & \epsilon \\ \eta & \beta & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \eta & \epsilon & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \\ e \\ \epsilon \end{matrix}$$

Calculando \mathbf{C}_{uc}^2 , tem-se

$$\mathbf{C}_{uc}^2 = \begin{bmatrix} \beta\eta & \epsilon & \kappa\eta & \epsilon & \epsilon \\ \epsilon & \eta\beta & \epsilon & \underline{\alpha\eta} & \epsilon \\ \eta\eta & \alpha\beta + \kappa\beta + \eta\beta & \epsilon & \epsilon & \epsilon \\ \beta\eta + \alpha\eta & \eta\beta + \alpha\beta & \epsilon & \epsilon & \epsilon \\ \epsilon & \kappa\beta & \epsilon & \underline{\eta\eta} & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \\ e \\ \epsilon \end{matrix}$$

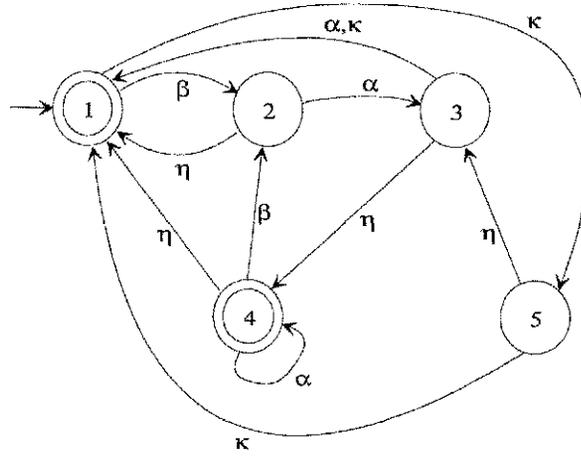


Figure 3.10: Autômato para exemplificar a construção da matriz de caminhos C_{uc}^n .

Nessa matriz, vê-se que o elemento $(a_{uc})_{3,4} = \eta$ aparece em $(c_{uc}^2)_{k,4}$, para $k = 2$ e $k = 5$. Isto é, nas linhas $k = 2$ e $k = 5$, e na coluna 3 de \mathbf{A} , encontram-se os elementos que antecedem $(a_{uc})_{3,4} = \eta$ em $(c_{uc}^2)_{k,4}$. Logo, os termos $(c_{uc}^2)_{2,4}$ e $(c_{uc}^2)_{5,4}$ contêm o elemento $(a_{uc})_{3,4} = \eta$ como sendo o último evento de suas seqüências. Calculando C_{uc}^3 , tem-se

$$C_{uc}^3 = \begin{bmatrix} \epsilon & \beta\eta\beta & \epsilon & \kappa\eta\eta & \epsilon \\ \eta\beta\eta + \alpha\eta\eta & \alpha\eta\beta & \epsilon & \epsilon & \epsilon \\ \alpha\beta\eta + \kappa\beta\eta + \eta\beta\eta & \eta\eta\beta & \epsilon & \epsilon & \epsilon \\ \eta\beta\eta + \alpha\beta\eta & \beta\eta\beta + \alpha\eta\beta & \epsilon & \epsilon & \epsilon \\ \kappa\beta\eta + \eta\eta\eta & \eta\eta\beta & \epsilon & \epsilon & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \\ e \\ \epsilon \end{matrix}$$

onde o termo $(c_{uc}^3)_{1,4}$ apresenta o termo $(c_{uc}^2)_{k,3}$, $k = 1$, antecedendo o termo $(a_{uc})_{3,4} = \eta$.

Com essas definições, procede-se à construção do supervisor quando o Lema 3.1 não é satisfeito. Para isto, utiliza-se a Definição 3.30 para definir caminhos em uma matriz de incidência que iniciem com eventos de \mathbf{E} , seguidos apenas por eventos não controláveis de \mathbf{A} .

Definição 3.31 Dada a matriz de incidência \mathbf{E} , que é uma matriz de caminhos de comprimento 1, define-se a matriz de caminhos

$$\mathbf{B}_{uc}^n = \mathbf{E} \otimes (\mathbf{A}_{uc})^{n-1}$$

em que o primeiro elemento de cada seqüência é um elemento de \mathbf{E} , e os demais são elementos pertencentes a \mathbf{A}_{uc} . O vetor de estado inicial é definido como sendo o mesmo vetor de \mathbf{E} , e o vetor de estados marcados é definido por

$$\phi_{i_{B_{uc}^n}}(\mathbf{B}_{uc}^n) = \phi_{i_{\mathbf{E}}}(\mathbf{E}) \oplus \phi_{i_{\mathbf{A}}}(\mathbf{A}).$$

Observação: A utilização da notação de uma matriz \mathbf{B} genérica citada anteriormente para algumas definições de operadores e demonstrações, não deve ser confundida com a matriz \mathbf{B}_{uc}^n , que é uma definição específica para denotar a matriz de caminhos em que os primeiros elementos de cada seqüência pertencem à \mathbf{E} , seguidos por elementos de \mathbf{A}_{uc} .

Através da matriz \mathbf{B}_{uc}^n , pode-se avaliar se existe algum evento não controlável não pertencente à especificação que possa ocorrer a partir de uma seqüência qualquer. Se existir, este evento aparece como último evento em uma determinada seqüência em \mathbf{B}_{uc}^n . A avaliação dos termos de \mathbf{B}_{uc}^n determina uma condição de teste semelhante a

$$\Sigma(H(x)) \cap \Sigma_{uc} \subset \Sigma(x).$$

Deve-se observar que, para $n = 1$, tem-se

$$\mathbf{B}_{uc}^1 = \mathbf{E} \otimes (\mathbf{A}_{uc})^0 = \mathbf{E} \otimes \mathbf{I} = \mathbf{E},$$

onde \mathbf{I} é a matriz identidade formada pelo elemento identidade e na diagonal principal e pelo elemento nulo ϵ nos demais elementos.

Igualmente à matriz de caminhos \mathbf{C}_{uc}^n , para uma matriz de caminhos \mathbf{B}_{uc}^n , um evento $\sigma_{uc} \subset (a_{uc})_{i,j}$ que não pertence à especificação de comportamento \mathbf{E} , finaliza uma seqüência nos termos de $(b_{uc}^n)_{k,j}$ e são antecidos pelos termos de $(b_{uc}^{n-1})_{k,i}$.

Assim, tem-se:

Teorema 3.3 *Dada uma especificação de comportamento \mathbf{E} válida e a matriz de incidência dos eventos não controláveis \mathbf{A}_{uc} , do autômato G trim, se*

$$ACES(\mathbf{E} \oplus \mathbf{A}_{uc}) \triangleright \mathbf{E},$$

então a sup $C(L)$ será determinada recursivamente por:

1. Para $n = 1$, $\mathbf{S}^1 = \mathbf{E}$.
2. Para $n = n + 1$, enquanto $(n \leq N) \wedge \exists \sigma_{uc} \notin \mathbf{E}$ então

$$\begin{aligned} \mathbf{B}_{uc}^n &= \mathbf{E} \otimes (\mathbf{A}_{uc})^{n-1} \\ \mathbf{S}^n &= [s_{ij}^n], s_{ij}^n = \begin{cases} s_{ij}^{n-1} & \text{se } \sigma^1 \sigma_{uc}^2 \dots \sigma_{uc}^n \in \mathbf{B}_{uc}^n \wedge \sigma_{uc}^n \in \mathbf{E}; \\ \epsilon & \text{se } \sigma_{uc}^n \notin \mathbf{E} \wedge \sigma^1 \in \Sigma_c \end{cases} \\ \mathbf{S}^n &= TRIM(\mathbf{S}^n) \end{aligned}$$

onde σ_{uc}^n é o n -ésimo evento da seqüência de um termo de \mathbf{B}_{uc}^n , que pode ser $\sigma_{uc} \notin \mathbf{E}$.

3. Se $(n > N) \wedge (\exists \sigma_{uc} \notin \mathbf{E} \text{ em } \mathbf{S}^n)$, então $\mathbf{S} = [\epsilon]$.

Demonstração:

Se

$$ACES(\mathbf{E} \oplus \mathbf{A}_{uc}) \triangleright \mathbf{E},$$

então, para

$$\mathbf{S}^1 = \mathbf{E}$$

e

$$\mathbf{B}_{uc}^n = \mathbf{E} \otimes (\mathbf{A}_{uc})^{n-1}$$

que tem seus termos dados por seqüências de eventos criadas através de multiplicações matriciais de elementos não comutativos à direita, tem-se que o elemento $(a_{uc})_{k,l} \notin \mathbf{E}$, mas que pode ocorrer em \mathbf{A} por uma seqüência em \mathbf{E} , aparece em

$$(b_{uc}^n)_{p,l} = e_{i,j} \otimes \underbrace{(a_{uc})_{j,m} \otimes (a_{uc})_{m,n} \otimes \dots \otimes (a_{uc})_{k,l}}_{n-1 \text{ eventos}}$$

como último elemento da seqüência.

(i) Se $e_{i,j}$ é controlável, então fazendo $e_{i,j} = s_{i,j} = \epsilon$, encontra-se

$$(b_{uc}^n)_{p,l} = \epsilon.$$

Isto é, todos os elementos $(a_{uc})_{j,m}, (a_{uc})_{m,n}, \dots, (a_{uc})_{k,l}$, não mais ocorrem, pois as linhas j, m, \dots, p não são mais acessíveis. Considerando que alguma linha i é não coacessível,

$$\mathbf{S}^n = COACES(\mathbf{S}^n).$$

elimina os elementos da coluna i , que torna a linha i não acessível. Assim, fazendo

$$\mathbf{S}^n = ACES(\mathbf{S}^n)$$

as linhas não acessíveis são eliminadas e \mathbf{S}^n é um supervisor coacessível. Essas duas operações equivalem a fazer

$$\mathbf{S}^n = TRIM(\mathbf{S}^n).$$

(ii) Se $e_{i,j}$ é não controlável, então para

$$\mathbf{B}_{uc}^{n+1} = \mathbf{E} \otimes (\mathbf{A}_{uc})^n, n + 1 \leq N$$

a mesma condição de (i) pode ser satisfeita.

(iii) Considerando que $e_{i,j}$ é não controlável em

$$(b_{uc}^n)_{p,l} \wedge n = N,$$

então $e_{i,j}$ não pode ser inibido e, conseqüentemente, $(a_{uc})_{k,l}$ também não pode. Contudo, \mathbf{B}_{uc}^N sendo uma matriz de caminhos com comprimento N , implica que todo caminho tem no máximo N eventos distintos, cada um ocorrendo uma única vez e mudando de um estado para outro. Disso,

$$(b_{uc}^n)_{p,l} = e_{1,j} \otimes (a_{uc})_{j,m} \otimes (a_{uc})_{m,n} \otimes \dots \otimes (a_{uc})_{k,N}$$

que determina que este caminho de comprimento máximo inicia do estado inicial, e tem N eventos não controláveis. Logo, não sendo possível inibir o primeiro evento da seqüência, o que significa

$$(n > N) \wedge (\exists \sigma_{uc} \notin \mathbf{E} \text{ em } \mathbf{S}^n).$$

Logo \mathbf{E} não é factível.

- iv) Sendo $\mathbf{E} \not\trianglelefteq \mathbf{A}$, a transformação de \mathbf{E} em $\mathbf{E}^\#$ e \mathbf{A} em $\mathbf{A}^\#$, de tal forma que $\mathbf{E}^\# \trianglelefteq \mathbf{A}^\#$ garante as mesmas condições de (i), (ii) e (iii). \blacklozenge

De acordo com esse Teorema, se o último evento de uma seqüência em \mathbf{B}_{uc}^n não pertence à especificação, a inibição do primeiro evento elimina esta seqüência. A existência de outros eventos não pertencentes à especificação, é eliminada recursivamente em \mathbf{B}_{uc}^n , para $n = 2, 3, \dots, N$, construindo o supervisor. Se algum evento não controlável que não pertence à especificação \mathbf{E} não for eliminado de acordo com o Teorema 3.3, não é possível sintetizar um supervisor para a especificação desejada. Por outro lado, se $\forall n, \mathbf{S}^n$ não for coacessível, $\mathbf{S} = [\epsilon]$.

Exemplo 3.22 Considerando que o autômato mostrado na Figura 3.8 tem

$$\mathbf{A} = \begin{bmatrix} \alpha & \beta & \mu \\ \alpha & \epsilon & \mu \\ \epsilon & \mu & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ e \end{matrix}$$

em que $\Sigma = \{\alpha, \beta, \mu, \kappa\}$ e $\Sigma_{uc} = \{\kappa\}$, é dada a especificação

$$\mathbf{E} = \begin{bmatrix} \epsilon & \beta & \epsilon \\ \alpha & \epsilon & \mu \\ \epsilon & \mu & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ e \end{matrix}$$

a qual está apresentada na Figura 3.11. Calculando $ACES(\mathbf{E} \oplus \mathbf{A}_{uc})$, encontra-se

$$ACES \left(\begin{bmatrix} \epsilon & \beta & \epsilon \\ \alpha & \epsilon & \mu \\ \epsilon & \mu & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ e \end{matrix} \oplus \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ e \end{matrix} \right) =$$

$$ACES \left(\begin{bmatrix} \epsilon & \beta & \epsilon \\ \alpha & \epsilon & \mu \\ \epsilon & \mu & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ e \end{matrix} \right) = \begin{bmatrix} \epsilon & \beta & \epsilon \\ \alpha & \epsilon & \mu \\ \epsilon & \mu & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ e \end{matrix} \triangleright \mathbf{E}.$$

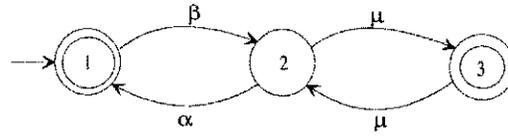


Figure 3.11: Especificação não controlável.

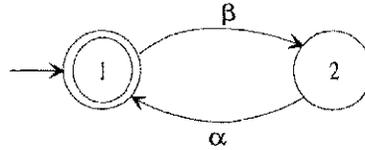


Figure 3.12: Supervisor para especificação em que $ACES(\mathbf{E} \oplus \mathbf{A}_{uc}) \triangleright \mathbf{E}$.

Então,

$$\mathbf{S}^1 = \mathbf{E} = \begin{bmatrix} \epsilon & \beta & \epsilon \\ \alpha & \epsilon & \mu \\ \epsilon & \mu & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ e \end{matrix}$$

Calculando \mathbf{B}_{uc}^2 , tem-se

$$\mathbf{B}_{uc}^2 = \mathbf{E} \otimes \mathbf{A}_{uc} = \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \mu\kappa \\ \epsilon & \epsilon & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ e \end{matrix}$$

que, utilizando o Teorema 3.3, encontra-se o termo $(b_{uc}^2)_{2,3} = \mu\kappa$, que contém o evento controlável μ seguido do evento κ , que não pertence à especificação. Como μ é controlável, então $s_{2,3}^2 = \epsilon$. Calculando $\mathbf{S}^2 = COACES(\mathbf{S}^2)$ tem-se

$$\mathbf{S}^2 = COACES(\mathbf{S}^2) = \begin{bmatrix} \alpha & \beta & \epsilon \\ \alpha & \epsilon & \epsilon \\ \epsilon & \mu & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ e \end{matrix} = \begin{bmatrix} \alpha & \beta & \epsilon \\ \alpha & \epsilon & \epsilon \\ \epsilon & \mu & \kappa \end{bmatrix} \begin{matrix} e \\ \epsilon \\ e \end{matrix}$$

Calculando $\mathbf{S} = ACES(\mathbf{S}^2)$, encontra-se o supervisor determinado pela matriz de incidência

$$\mathbf{S} = ACES(\mathbf{S}^2) = \begin{bmatrix} \epsilon & \beta & \epsilon \\ \alpha & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \end{matrix}$$

o qual está mostrado na Figura 3.12.

Observa-se no Exemplo 3.22 que, se o estado 1 da especificação não for marcado, o supervisor não é possível, pois não é coacessível.

3.3 Algoritmo de síntese do supervisor

A partir do que foi apresentado, o seguinte algoritmo de síntese do supervisor é construído:

Algoritmo 3.7 Construção da $\text{sup } C(L)$

1. Se $\mathbf{E} \not\leq \mathbf{A}$, Construa $\mathbf{A}^\#$ e $\mathbf{E}^\#$ a partir de \mathbf{A} e \mathbf{E} , e faça $\mathbf{E} = \mathbf{E}^\#$ e $\mathbf{A} = \mathbf{A}^\#$.
2. Faça $\mathbf{D} = ACES(\mathbf{E} \oplus \mathbf{A}_{uc})$.
3. Se $\mathbf{D} = \mathbf{E}$, faça $\mathbf{S} = TRIM(\mathbf{E})$ e pare.
4. Se $\mathbf{D} \triangleright \mathbf{E}$, faça $n = 1$:
 - a) $\mathbf{S}^n = \mathbf{E}$, $xdif(k, n-1) = i$ e $ydif(k, n-1) = j$ (onde $\mathbf{A}_{uc} \neq \mathbf{E}_{uc}$), para $k = 1, \dots, M$ (M número de elementos diferentes entre \mathbf{A}_{uc} e \mathbf{E}_{uc}).
 - b) Calcule \mathbf{B}_{uc}^n .
 - c) Faça para $k = 1$ até M :
 - i. Procure os elementos em $\mathbf{B}_{uc}^n(i, ydif(k, n-1))$, onde $\sigma_{uc}^n \notin \mathbf{E}$ (σ_{uc}^n sendo o último elemento da seqüência).
 - (1) Se $\sigma_{uc}^n \notin \mathbf{E}$ e $\sigma^1 \in \Sigma_c$, faça $\mathbf{S}^n(i, xdif(k, n-1)) = \epsilon$ e calcule $COACES(\mathbf{S}^n)$ e $ACES(\mathbf{S}^n)$;
 - (2) Se $\sigma_{uc}^n \notin \mathbf{E}$ e $\sigma^1 \in \Sigma_{uc}$, faça $xdif(k, n) = i$ e $ydif(k, n) = ydif(k, n-1)$.
 - d) Se $\forall k = 1$ até M , $\mathbf{S}^n(xdif(k, n-1), ydif(k, n-1)) = \epsilon$ pare. Caso contrário, faça $n = n + 1$ e retorne ao passo 4.b.
 - i) Se $n > N$ pare (\mathbf{E} não é factível).

Nesse algoritmo, as matrizes $xdif$ e $ydif$ guardam os valores de i e j de \mathbf{E} , respectivamente, para os eventos σ_{uc} acessíveis, mas que não devem ocorrer em \mathbf{E} . Sempre que uma seqüência em \mathbf{B}_{uc}^n tem o último evento σ_{uc} acessível, porém não pertencente a \mathbf{E} e o primeiro evento não controlável, estas matrizes são atualizadas com os valores de i e j de \mathbf{B}_{uc}^{n-1} . Enquanto $n \leq N$ (dimensão de \mathbf{A}), o algoritmo é repetido, buscando a $\text{sup } C(L)$.

3.4 Exemplos da utilização do algoritmo

Na apresentação desse Capítulo, foram apresentados alguns exemplos simples para demonstrar a abordagem proposta. Nessa seção são mostrados outros exemplos mostrando a utilização do algoritmo proposto.

Exemplo 3.23 No autômato visto na Figura 3.13, com $\Sigma = \{\alpha, \beta, \kappa, \eta, \lambda, \mu\}$ e $\Sigma_{uc} = \{\alpha, \lambda\}$,

as matrizes de incidência A e A_{uc} são dadas por

$$A = \begin{bmatrix} \epsilon & \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \lambda & \beta & \kappa & \mu & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \kappa + \mu & \epsilon & \epsilon & \eta & \epsilon \\ \epsilon & \epsilon & \epsilon & \alpha & \beta & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \mu & \lambda & \epsilon & \alpha & \eta & \epsilon \\ \alpha & \mu & \epsilon & \eta & \kappa & \beta + \lambda & \epsilon & \epsilon \\ \epsilon & \mu & \eta + \lambda & \epsilon & \epsilon & \epsilon & \alpha + \beta & \epsilon \end{bmatrix} \begin{matrix} \epsilon \\ e \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \\ e \end{matrix}$$

e

$$A_{uc} = \begin{bmatrix} \epsilon & \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \lambda & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \alpha & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \lambda & \epsilon & \alpha & \epsilon & \epsilon \\ \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \lambda & \epsilon & \epsilon \\ \epsilon & \epsilon & \lambda & \epsilon & \epsilon & \epsilon & \alpha & \epsilon \end{bmatrix} \begin{matrix} \epsilon \\ e \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \\ e \end{matrix}$$

Definindo a especificação de comportamento pela matriz de incidência

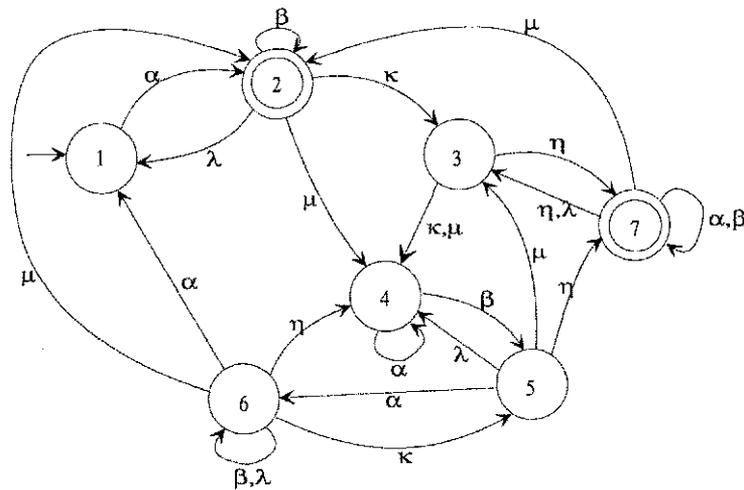


Figure 3.13: Autômato para o Exemplo 3.23.

$$E = \begin{bmatrix} \epsilon & \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \lambda & \epsilon & \kappa & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \mu & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \alpha & \beta & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \alpha & \eta & \epsilon \\ \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \lambda & \epsilon & \epsilon \\ \epsilon & \epsilon & \lambda & \epsilon & \epsilon & \epsilon & \alpha & \epsilon \end{bmatrix} \begin{matrix} \epsilon \\ e \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \\ e \end{matrix}$$

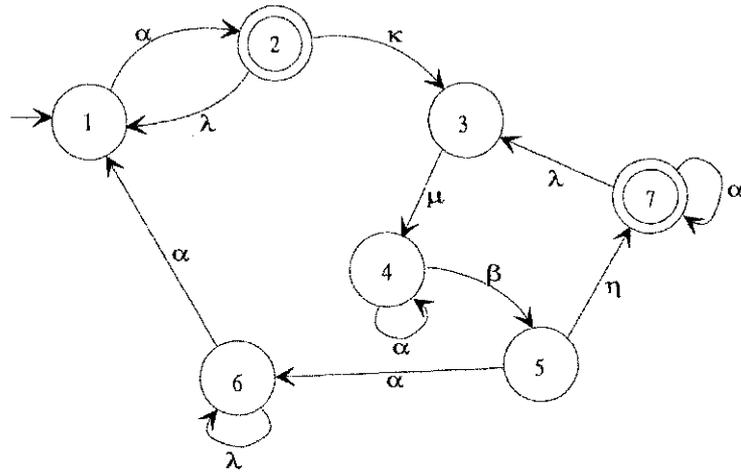


Figure 3.14: Especificação para o autômato do Exemplo 3.23.

a qual está mostrada graficamente na Figura 3.14, vê-se que \mathbf{E} é válida, e que $\mathbf{E} \triangleleft \mathbf{A}$, mas a condição de controlabilidade falha, desde que $ACES(\mathbf{E} \oplus \mathbf{A}_{uc}) \triangleright \mathbf{E}$. O termo que não pertence à \mathbf{E} é $(a_{uc})_{5,4} = \lambda$. De acordo com o Teorema 3.3, faz-se $\mathbf{S}^1 = \mathbf{E}$ e calcula-se $\mathbf{B}_{uc}^2 = \mathbf{E} \otimes \mathbf{A}_{uc}$ para avaliar quais os eventos que devem ser desabilitados para encontrar uma linguagem restrita para o supervisor.

$$\mathbf{B}_{uc}^2 = \mathbf{E} \otimes \mathbf{A}_{uc} = \begin{bmatrix} \alpha\lambda & \epsilon \\ \epsilon & \lambda\alpha & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \mu\alpha & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \eta\alpha & \epsilon & \alpha\alpha + \beta\lambda & \epsilon & \beta\alpha & \epsilon & \epsilon \\ \alpha\alpha & \epsilon & \eta\lambda & \epsilon & \epsilon & \alpha\lambda & \eta\alpha & \epsilon \\ \lambda\alpha & \alpha\alpha & \epsilon & \epsilon & \epsilon & \lambda\lambda & \epsilon & \epsilon \\ \epsilon & \epsilon & \alpha\lambda & \epsilon & \epsilon & \epsilon & \alpha\alpha & \epsilon \end{bmatrix} \begin{matrix} \epsilon \\ e \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \\ e \end{matrix}$$

Essa matriz de caminhos define as palavras de comprimento 2, que terminam com um evento não controlável e que levam o autômato do estado i para o estado j iniciando com eventos da especificação. Como o interesse é inibir o evento que antecede o evento que torna a condição de controlabilidade falsa, vê-se que o termo $(b_{uc}^2)_{4,4}$ contém a seqüência $\beta\lambda$, que leva o autômato através do evento controlável β do estado 4 para o estado 5, e depois retorna para o estado 4 pelo evento não controlável λ , que não faz parte da especificação \mathbf{E} . Tornando o elemento $s_{4,5}^1 = \epsilon$ determina-se \mathbf{S}^2 . Calculando $COACES(\mathbf{S}^2)$, encontra-se a matriz

$$\mathbf{S}^2 = COACES \left(\begin{pmatrix} \begin{bmatrix} \epsilon & \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \lambda & \epsilon & \kappa & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \mu & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \alpha & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \alpha & \eta & \epsilon \\ \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \lambda & \epsilon & \epsilon \\ \epsilon & \epsilon & \lambda & \epsilon & \epsilon & \epsilon & \alpha & \epsilon \end{bmatrix} & \begin{matrix} \epsilon \\ e \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \\ e \end{matrix} \end{pmatrix} \right) = \begin{bmatrix} \epsilon & \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \lambda & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \alpha & \eta & \epsilon \\ \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \lambda & \epsilon & \epsilon \\ \epsilon & \epsilon & \lambda & \epsilon & \epsilon & \epsilon & \alpha & \epsilon \end{bmatrix} \begin{matrix} \epsilon \\ e \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \\ e \end{matrix}$$

Calculando $ACES(S^2)$, encontra-se

$$S = ACES(S^2) = \begin{bmatrix} \epsilon & \alpha \\ \lambda & \epsilon \end{bmatrix} \begin{matrix} e \\ e \end{matrix} \quad (3.20)$$

e está mostrado na Figura 3.15. Observe que a matriz S se apresenta com uma dimensão 2, contudo sua dimensão é a mesma de S^2 , em que os elementos de todas as linhas e colunas $k = 3, \dots, 7$ são iguais a ϵ .

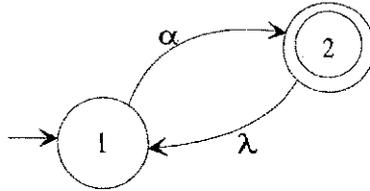


Figure 3.15: Supervisor que gera a $sup C(L)$ construído através da especificação dada no Exemplo 3.23.

Exemplo 3.24 Para o autômato da Figura 2.8, as matrizes de incidência A e A_{uc} são dadas por

$$A = \begin{bmatrix} \epsilon & \alpha_1 & \epsilon & \epsilon & \alpha_2 \\ \epsilon & \epsilon & \beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \beta & \epsilon \\ \epsilon & \epsilon & \epsilon & \beta & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta \end{bmatrix} \begin{matrix} e \\ e \\ e \\ e \\ e \end{matrix}$$

e

$$A_{uc} = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \beta & \epsilon \\ \epsilon & \epsilon & \epsilon & \beta & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta \end{bmatrix} \begin{matrix} e \\ e \\ e \\ e \\ e \end{matrix}$$

A especificação de comportamento E é vista na Figura 2.9, e tem sua matriz de incidência dada por

$$E = \begin{bmatrix} \epsilon & \alpha_1 & \epsilon & \epsilon & \alpha_2 \\ \epsilon & \epsilon & \beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \beta & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta \end{bmatrix} \begin{matrix} e \\ e \\ e \\ e \\ e \end{matrix}$$

que é uma especificação válida e $E \triangleleft A$. Porém, a condição de controlabilidade falha, desde

que $ACES(\mathbf{E} \oplus \mathbf{A}_{uc}) \triangleright \mathbf{E}$, onde o termo $(a_{uc})_{4,4} \neq e_{4,4}$. Assim, faz-se $\mathbf{S}^1 = \mathbf{E}$ e calcula-se

$$\mathbf{B}_{uc}^2 = \mathbf{E} \otimes \mathbf{A}_{uc} = \begin{bmatrix} \epsilon & \epsilon & \alpha_1\beta & \epsilon & \alpha_2\beta \\ \epsilon & \epsilon & \epsilon & \beta\beta & \epsilon \\ \epsilon & \epsilon & \epsilon & \beta\beta & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta\beta \end{bmatrix} \begin{matrix} e \\ e \\ e \\ e \\ e \end{matrix}$$

Nessa matriz, o termo $(b_{uc}^2)_{3,4}$ é $\beta\beta$, ou seja, inicia por um evento não controlável. Assim, $\mathbf{S}^2 = COACES(\mathbf{S}^2) = ACES(\mathbf{S}^2) = \mathbf{S}^1$ e calcula-se \mathbf{B}_{uc}^3 para ver se é conclusivo:

$$\mathbf{B}_{uc}^3 = \mathbf{E} \otimes (\mathbf{A}_{uc})^2 = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \alpha_1\beta\beta & \alpha_2\beta\beta \\ \epsilon & \epsilon & \epsilon & \beta\beta\beta & \epsilon \\ \epsilon & \epsilon & \epsilon & \beta\beta\beta & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta\beta\beta \end{bmatrix} \begin{matrix} e \\ e \\ e \\ e \\ e \end{matrix}$$

Observando $(b_{uc}^3)_{2,4}$, vê-se que esse termo define um caminho que leva do estado 2 para o estado 4, nele permanecendo pela ocorrência do evento não controlável β . O termo $(b_{uc}^3)_{3,4}$ também define um caminho iniciando do estado 3 para o estado 4. Portanto, como em ambos os termos $\sigma^1 = \beta$, \mathbf{B}_{uc}^3 ainda não é conclusiva e $\mathbf{S}^3 = \mathbf{S}^2$. Assim, calculando \mathbf{B}_{uc}^4 , tem-se

$$\mathbf{B}_{uc}^4 = \mathbf{E} \otimes (\mathbf{A}_{uc})^3 = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \alpha_1\beta\beta\beta & \alpha_2\beta\beta\beta \\ \epsilon & \epsilon & \epsilon & \beta\beta\beta\beta & \epsilon \\ \epsilon & \epsilon & \epsilon & \beta\beta\beta\beta & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta\beta\beta\beta \end{bmatrix} \begin{matrix} e \\ e \\ e \\ e \\ e \end{matrix}$$

que tem os termos $(b_{uc}^4)_{2,4}$ e $(b_{uc}^4)_{3,4}$, com caminhos que levam para o estado 4, mas com o primeiro evento da seqüência sendo β , e o termo $(b_{uc}^4)_{1,4}$, iniciando com o evento controlável α_1 , seguido da ocorrência do evento não controlável β por três vezes, onde $(b_{uc}^4)_{1,4}$ contém $\sigma^4 = \beta \neq e_{4,4}$, isto é $\sigma^4 \notin \mathbf{E}$. Com isso, o supervisor é definido pela inibição de α_1 no termo $s_{1,2}$, que elimina a seqüência que leva à ocorrência do evento não controlável β no estado 4:

$$\mathbf{S} = COACES(\mathbf{S}^4) = ACES(\mathbf{S}^4) = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \alpha_2 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \\ \epsilon \\ e \end{matrix}$$

o qual é o mesmo apresentado na Figura 2.10.

Exemplo 3.25 Considere o exemplo clássico de Ramadge e Wonham, onde existe um sistema com duas máquinas e um buffer. A máquina 1 é representada pelo autômato G_1 e

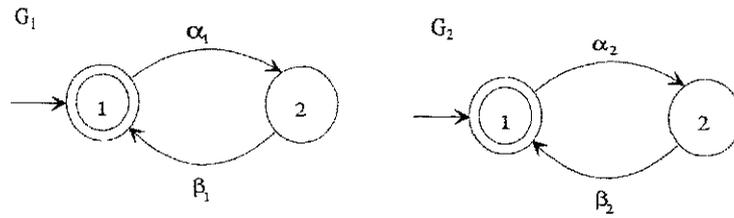


Figure 3.16: Autômatos G_1 e G_2 .

a máquina 2 é representada pelo autômato G_2 , apresentados na Figura 3.16, onde $\Sigma_c = \{\alpha_1, \alpha_2\}$ e $\Sigma_{uc} = \{\beta_1, \beta_2\}$. As matrizes de incidência desses autômatos são

$$\mathbf{A}_1 = \begin{bmatrix} \epsilon & \alpha_1 \\ \beta_1 & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \end{matrix} \quad \mathbf{A}_2 = \begin{bmatrix} \epsilon & \alpha_2 \\ \beta_2 & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \end{matrix}.$$

A composição paralela dessas duas matrizes gera a matriz

$$\mathbf{A}_3 = \begin{bmatrix} \epsilon & \alpha_1 & \alpha_2 & \epsilon \\ \beta_1 & \epsilon & \epsilon & \alpha_2 \\ \beta_2 & \epsilon & \epsilon & \alpha_1 \\ \epsilon & \beta_2 & \beta_1 & \epsilon \end{bmatrix} \begin{matrix} e \\ \epsilon \\ \epsilon \\ \epsilon \end{matrix},$$

que é a representação do autômato da Figura 3.17. Considerando a especificação

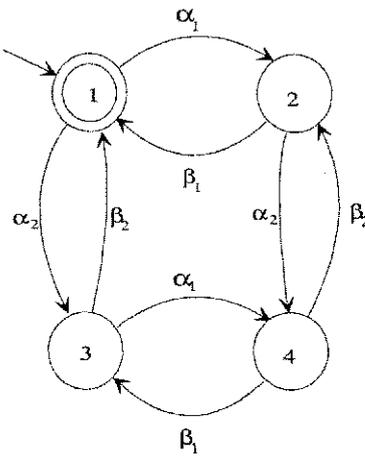


Figure 3.17: Autômato G_3 : composição síncrona de G_1 e G_2 .

$$\mathbf{E}_1 = \begin{bmatrix} \alpha_1 + \beta_2 & \beta_1 \\ \alpha_2 & \alpha_1 + \beta_2 \end{bmatrix} \begin{matrix} e \\ \epsilon \end{matrix},$$

que significa que a máquina 2 só deve iniciar o processamento se a máquina 1 tiver finalizado um item. Como $\mathbf{E}_1 \not\leq \mathbf{A}$, constróem-se as matrizes $\mathbf{A}^\#$ e $\mathbf{E}^\#$, as quais são as matrizes do Exemplo 3.16, onde os autômatos são vistos na Figura 3.18(a) e 3.18(b), respectivamente. Calculando ACES ($\mathbf{E}^\# \oplus \mathbf{A}_{uc}^\#$), tem-se

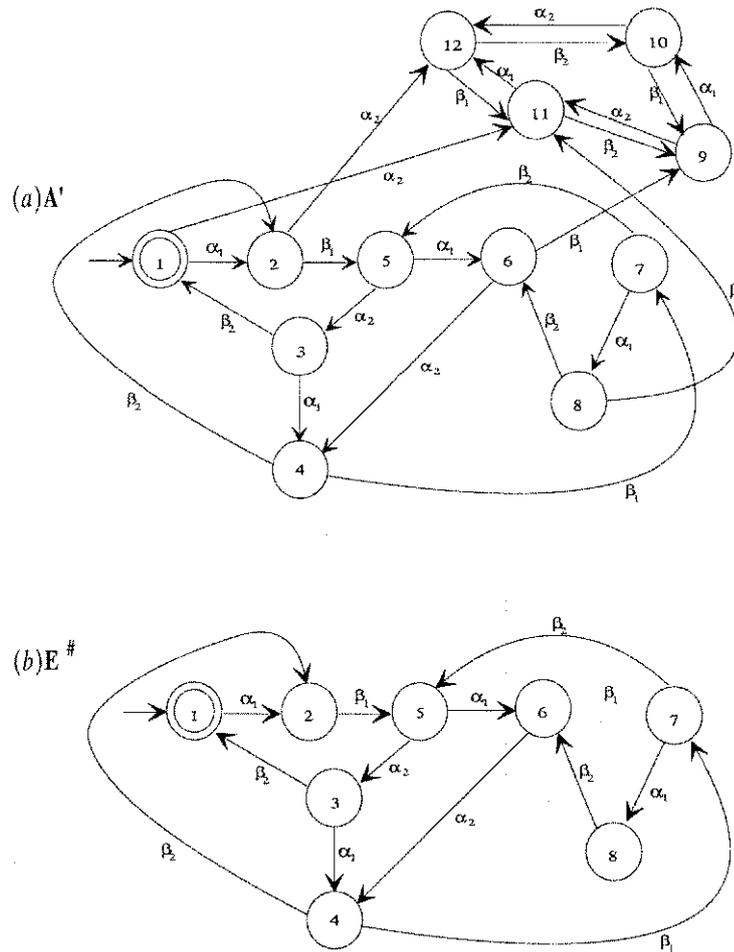


Figure 3.18: (a) Autômato $G = A_3 || E_1$ e (b) Especificação de comportamento.

$$ACES \left(\mathbf{E}^\# \oplus \mathbf{A}_{uc}^\# \right) = \begin{bmatrix}
 \epsilon & \alpha_1 & \epsilon \\
 \epsilon & \epsilon & \epsilon & \epsilon & \beta_1 & \epsilon \\
 \beta_2 & \epsilon & \epsilon & \alpha_1 & \epsilon \\
 \epsilon & \beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & \beta_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\
 \epsilon & \epsilon & \alpha_2 & \epsilon & \epsilon & \alpha_1 & \epsilon \\
 \epsilon & \epsilon & \epsilon & \alpha_2 & \epsilon & \epsilon & \epsilon & \epsilon & \beta_1 & \epsilon & \epsilon & \epsilon & \epsilon \\
 \epsilon & \epsilon & \epsilon & \epsilon & \beta_2 & \epsilon & \epsilon & \alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\
 \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & \beta_1 & \epsilon & \epsilon \\
 \epsilon & \epsilon \\
 \epsilon & \epsilon \\
 \epsilon & \beta_2 & \epsilon & \epsilon & \epsilon \\
 \epsilon & \epsilon
 \end{bmatrix} \triangleright \mathbf{E}^\#.$$

Nessa matriz, os elementos (6, 9), (8, 11) e (11, 9) não pertencem à $\mathbf{E}^\#$. Assim, faz-se $\mathbf{S}^1 =$

$E^\#$ e calcula-se

$$B_{uc}^2 = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \alpha_1\beta_1 & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \alpha_1\beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & \alpha_1\beta_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta_2\beta_1 + \beta_1\beta_2 & \epsilon \\ \alpha_2\beta_2 & \epsilon & \underline{\alpha_1\beta_1} & \epsilon & \epsilon & \epsilon \\ \epsilon & \alpha_2\beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & \alpha_2\beta_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \alpha_1\beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & \underline{\alpha_1\beta_1} & \epsilon \\ \epsilon & \underline{\beta_2\beta_1} & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \epsilon \end{bmatrix}$$

onde encontram-se os elementos $(b_{uc}^2)_{5,9}$, $(b_{uc}^2)_{7,11}$ e $(b_{uc}^2)_{8,9}$ que são os termos resultantes da multiplicação $e_{5,6}^\# \otimes (a_{uc}^\#)_{6,9}$, $e_{7,8}^\# \otimes (a_{uc}^\#)_{8,11}$ e $e_{8,6}^\# \otimes (a_{uc}^\#)_{6,9}$, respectivamente. Assim, a inibição dos eventos que antecedem β_1 em $s_{5,6}^1$ e $s_{7,8}^1$ já tornam a linha 8 não acessível, não necessitando avaliar o termo $(b_{uc}^2)_{8,9}$. Isso define a matriz S^2 , dada por

$$S^2 = \begin{bmatrix} \epsilon & \alpha_1 & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta_1 & \epsilon \\ \beta_2 & \epsilon & \epsilon & \alpha_1 & \epsilon \\ \epsilon & \beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & \beta_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \alpha_2 & \epsilon \\ \epsilon & \epsilon & \epsilon & \alpha_2 & \epsilon & \epsilon & \epsilon & \epsilon & \beta_1 & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta_2 & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & \beta_1 & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \beta_2 & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon \end{bmatrix} e$$

Fazendo $S^2 = COACES(S^2)$, para eliminar os estados não coacessíveis e $S^2 = ACES(S^2)$, para eliminar os estados não acessíveis, encontra-se

$$S^2 = \begin{bmatrix} \epsilon & \alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta_1 & \epsilon & \epsilon \\ \beta_2 & \epsilon & \epsilon & \alpha_1 & \epsilon & \epsilon & \epsilon \\ \epsilon & \beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & \beta_1 \\ \epsilon & \epsilon & \alpha_2 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta_2 & \epsilon & \epsilon \end{bmatrix} e$$

que é o supervisor apresentado na Figura 3.19.

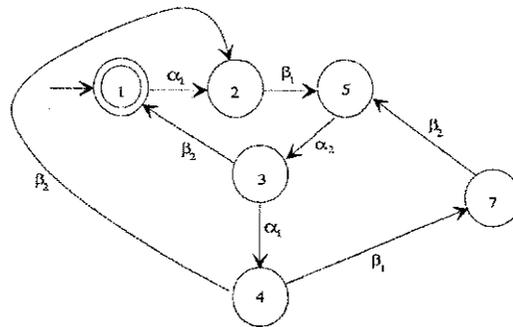


Figure 3.19: Supervisor.

Com estes exemplos, vê-se que as condições aqui formuladas, determinam uma fundamentação para a construção de um supervisor S . Também, pôde-se ver que o algoritmo proposto garante a construção de um supervisor, semelhante ao algoritmo de Ramadge e Wonham [RW87]. O valor máximo de iterações no algoritmo proposto é igual a $N - 2$. Quando este valor é alcançado e o primeiro evento ainda é não controlável, não é possível construir um supervisor para a especificação de comportamento desejada.

3.5 Estudo da complexidade algorítmica

Para o estudo da complexidade do algoritmo de síntese proposto, considerar-se-á N como o número de estados do autômato, ou a dimensão de sua matriz de incidência A , e M como o número de eventos não controláveis de A_{uc} que não pertencem à E_{uc} .

O estudo da complexidade algorítmica é feito pela avaliação do número de operações requerido a cada passo de um determinado procedimento [Kro87, CM94]. Para o caso do algoritmo aqui proposto, tem-se o seguinte:

Passo 1. Nesse passo cada elemento (i, j) da matriz E é comparado com o respectivo elemento (i, j) da matriz A . Logo, são necessárias N^2 operações. Se $E \not\leq A$, este passo tem uma ordem de complexidade $O(N^2)$, para a construção de $A^\#$ e $E^\#$, considerando que A e E^* apresentam-se com a mesma dimensão N .

Observação: A partir daqui, será considerado o pior caso, em que $E \not\leq A$. Dessa forma, deve-se entender a notação E como a representação da matriz $E^\#$ e a notação A como a representação da matriz $A^\#$.

Passo 2. Para determinação de $D = ACES(E \oplus A_{uc})$, são envolvidas $N \times N = N^2$ operações de soma na álgebra de dióides, somado com X operações para o cálculo de $ACES$ (calculado adiante). Ao todo, $N^2 + X$ operações.

Passo 3. Este passo envolve N^2 operações.

Observação: Os Passos 1, 2 e 3 tem sua função de complexidade de tempo dada por

$$T_1(N) = N^2 + N^2 + X + N^2 = 3N^2 + X.$$

Este passo representa a execução do algoritmo sobre o Lema 3.1.

Passo 4.a. Este passo envolve N^2 operações de soma na álgebra de dióides, somado com N^2 operações de comparação para definir $xdif(k, n-1)$ e $ydif(k, n-1)$, resultando em $2N^2$ operações.

Observação: Os Passos 4 e 4.a. não se repetem nesse algoritmo. Logo, para essa parte, tem-se

$$T_2(N) = 2N^2.$$

Passo 4.b. Este passo envolve N^2 operações de multiplicação na álgebra de dióides. Assim, para cada multiplicação, tem-se um máximo de operações dadas pela soma das concatenações das palavras de uma linha por uma coluna. Logo, para cada elemento resultante da multiplicação de uma linha por uma coluna, há N operações de concatenação entre os elementos das matrizes e $(N-1)$ operações de concatenações do símbolo $+$, resultando em um total de $(2N-1)$ operações. Assim, a multiplicação na álgebra de dióides requer $(2N-1)^2 = 4N^2 - 2N + 1$ operações.

Passo 4.c.i.1. Para cada elemento comparado satisfazendo o **Passo 4.c.i.**, este passo requer Y operações sobre o operador *COACES* e X operações sobre o operador *ACES*. Logo, $Y + X$ operações ao todo, para o pior caso;

Passo 4.c.i.2. Para cada elemento comparado, se o **Passo 4.c.i.** não é satisfeito, este passo envolve 1 operação de comparação.

Observação: Os Passos 4.c.i., 4.c.i.1. e 4.c.i.2. repetem-se por $M \times N$ vezes. Considerando o pior caso, onde hajam N elementos diferentes em N colunas, então $M = N$ e esses passos do algoritmo envolvem $(Y + X + 1)N^2$ operações.

Passo 4.d. Este passo requer $k = N$ (no pior caso) operações de comparação. Logo, N operações.

Passo 4.d.i. Este passo requer 1 operação de comparação.

Observação: O algoritmo repete-se, a partir do **Passo 4.b.** até o **Passo 4.d.i.**, por $N-2$ vezes, considerando o pior caso. Logo, a função de complexidade de tempo é dada por

$$T_3(N) = (N-2) \times ((4N^2 - 2N + 1) + (Y + X) + 1 + (Y + X + 1)N^2 + N + 1)$$

$$T_3(N) = (N-2) \times ((5 + Y + X)N^2 - N + Y + X + 3)$$

$$T_3(N) = (5 + Y + X)N^3 - (Y + X + 11)N^2 + (X + Y + 5)N - 2(X + Y + 3).$$

Este algoritmo tem, então,

$$\begin{aligned} T(N) &= T_1(N) + T_2(N) + T_3(N) \\ T(N) &= (5 + Y + X)N^3 - (Y + X + 6)N^2 + (X + Y + 5)N - X - 2(Y + 3). \end{aligned} \quad (3.21)$$

Nesse cálculo de $T(N)$ é considerado que a especificação não é uma submatriz do modelo do SED.

Para o cálculo de X , faz-se o estudo da complexidade do algoritmo do operador *ACES*. Este algoritmo envolve apenas operações de comparação e definição de valores, não apresentando custo computacional, apenas por avaliar e definir os valores das N colunas do vetor *vac*. Assim, considerando o pior caso, sua complexidade é de ordem $O(N)$. Logo, $X = N$.

O algoritmo do operador *COACES* é semelhante ao algoritmo do operador *ACES*, apresentando a mesma complexidade, isto é, $Y = X$.

Com isto, substituindo-se os valores de X e Y na equação (3.21), encontra-se

$$T(N) = 2N^4 + 3N^3 - 4N^2 + 2N - 6. \quad (3.22)$$

Dessa forma, de acordo com a equação (3.22), a complexidade do algoritmo apresentado é de ordem

$$O(N^4).$$

3.6 Considerações sobre a abordagem proposta

Deve-se observar que a complexidade $O(N^4)$ do algoritmo proposto é igual à complexidade do algoritmo de Ramadge e Wonham [RW89]. Esta complexidade refere-se à não satisfação do Lema 3.1, isto é, a condição $ACES(\mathbf{E} \oplus \mathbf{A}_{uc}) \triangleright \mathbf{E}$ ser satisfeita, e também que seja necessário repetir o procedimento de cálculo a partir do **Passo 4.b.** até o **Passo 4.d.i.**, por $N - 2$ vezes.

Para o algoritmo proposto, considerando que a especificação de comportamento \mathbf{E} é controlável e se apresenta como uma submatriz de \mathbf{A} , a complexidade do algoritmo é reduzida para $O(N^2)$.

Deve-se observar que o algoritmo proposto não apresenta cálculo de ponto fixo. Isto pode ser visto no Exemplo 3.24, onde o supervisor não é modificado até que se encontre em \mathbf{B}_{uc}^n uma seqüência iniciada por um evento controlável que elimine todo o caminho que leva ao estado proibido.

A transformação da matriz \mathbf{E} em $\mathbf{E}^\#$ que se apresenta como uma submatriz de $\mathbf{A}^\#$ (construída de \mathbf{A}), assegura que o operador de continência \triangleleft compare as linguagens das matrizes diretamente sobre os seus elementos. Os operadores *COACES* e *ACES* garantem que a solução do problema seja uma matriz *trim*. Caso contrário, o supervisor é uma matriz nula.

Chapter 4

A Síntese do supervisor de SEDs temporizados

Quando não se considera o tempo nos SEDs, a síntese de supervisores é realizada levando-se em conta seu comportamento lógico: um evento ou é habilitado, ou é inibido. Assim, não são permitidas inibições de eventos por tempo determinado, nem condições para operação em tempo real. A introdução de uma representação do tempo permite um aperfeiçoamento na modelagem e um refinamento no estudo de controle desses sistemas, possibilitando solucionar esses problemas. Para isto, a abordagem da síntese de supervisores para SEDs temporizados necessita de sua descrição através de um paradigma que relacione o tempo às habilitações dos eventos.

Neste Capítulo será apresentado uma abordagem de síntese de supervisores de SEDs temporizados. Esta abordagem utiliza o mesmo formalismo apresentado no Capítulo 3.

Para a compreensão deste Capítulo, será apresentado a descrição da representação do tempo nos SEDs e o posicionamento desta abordagem em relação à trabalhos desenvolvidos neste intuito. Também, serão apresentados os conceitos necessários para a formalização dessa abordagem.

4.1 Descrição da representação do tempo

A representação da evolução dinâmica dos SEDs através das linguagens formais provê uma descrição similar em termos de estados alcançados. Assim, para uma seqüência de eventos $s = \alpha\beta\kappa$, há também uma seqüência de estados $p = q_i q_{i+1} q_{i+2} q_{i+3}$, que são alcançados. Considerando que existe um intervalo de tempo entre cada ocorrência consecutiva de eventos, cada estado é alcançado em um tempo determinado. A cada instante de tempo novos eventos podem se tornar habilitados e sua ocorrência, que é instantânea, muda o estado do SED.

Para a descrição da evolução dinâmica do sistema, torna-se necessário incluir um mecanismo de temporização. Neste mecanismo se considera a inclusão do *relógio global*. Este

relógio é definido como um contador de tempo que está constantemente sendo incrementado em uma unidade de tempo. Também, este mecanismo associa a cada evento um intervalo de tempo mínimo entre duas ocorrências sucessivas, denominado *tempo de vida* do evento.

Cada evento tem um contador específico, que é inicializado com seu tempo de vida. Em um estado, os eventos definidos na função de transição têm seus contadores decrementados em uma unidade de tempo, para cada unidade de tempo incrementada no relógio global. Quando o contador de um evento é zerado, o evento torna-se habilitado. Este evento permanece habilitado até sua ocorrência, ou até a ocorrência de um outro evento também habilitado. A ocorrência de um evento habilitado determina uma mudança de estado no SED, e também, a reinicialização do seu contador.

Com a representação do tempo nos SEDs, vários trabalhos foram desenvolvidos no intuito de determinar formas de controle para estes sistemas. Dentre estes, encontra-se o formalismo de Brandin e Wonham [BW94, BWB92], que estende a TCS para tratar de SEDs que incluem uma representação temporizada. Neste trabalho é necessário definir os eventos forçados, proibitivos, prospectivos, remotos, ativos, elegíveis, entre outros, além da inclusão do evento denominado '*tick*' que sempre ocorre sob o término de um ciclo do relógio global. Brandin e Wonham utilizam os grafos de transições de atividades (GTA) [OW90b] para modelar os SEDs, bem como para construir composições síncronas. Entretanto, para a síntese do supervisor utilizam os grafo de transições temporizadas (GTT) [Law97]. Embora esta abordagem se apresente como uma ferramenta que soluciona o problema de controle de SEDs temporizados, a inclusão do evento '*tick*' resulta num aumento considerável no número de estados e transições do sistema, além de ser limitada à estudos de sistemas que não apresentam sincronismo. Por outro lado, quando se modela um SED, considera-se a situação de que sempre que um evento está habilitado, ele pode ou não ocorrer. Na abordagem de Brandin e Wonham, a modelagem utiliza eventos com tempo limite superior no qual sua ocorrência é obrigatória, (*hard deadline*), o que define uma aparente prévia introdução de controle no modelo do SED.

Quando se considera a sincronização de eventos, a álgebra de dióides [Bea96] se apresenta como ferramenta matemática para a descrição da evolução dinâmica do SED temporizado. Esta álgebra permite uma descrição da evolução de SEDs temporizados cíclicos, em termos dos tempos de ocorrência dos eventos, por meio de um sistema de equações linear [CDQV85, BCOQ92]. Nesta forma de abordar o problema de controle, há vários trabalhos desenvolvidos, utilizando como paradigma de modelagem os autômatos temporizados [AD94] ou os grafos de eventos temporizados (redes de Petri temporizadas [Chr83] que apresentam para cada lugar uma marcação binária, apenas uma transição de entrada e uma transição de saída).

Na abordagem do controle de SEDs temporizados, utilizando a álgebra de dióides [Gau92, Lib96, CHBF99, Boi99, Cot99], o supervisor avalia a saída do sistema e determina a ação de controle para inibir temporariamente os eventos [CG94, CG95, CG96].

Também, as séries formais [BR88, Gau94b, Gau94a, Gau96] são muito utilizadas para esta classe de SEDs temporizados na descrição da linguagem reconhecida de SEDs modelados por autômatos $(\max, +)$ [Gau93, Gau95a]. Estes autômatos apresentam os tempos de vida definidos em seus arcos, representando o menor tempo em que o evento pode ocorrer. Assim, o autômato $(\max, +)$ se assemelha aos GTAs [OW90b], considerando que a representação do tempo é definida diretamente no arco e restrita aos casos em que os limites superiores dos tempos de vida dos eventos são considerados iguais a infinito. Dessa forma, com o autômato $(\max, +)$ pode-se descrever a abordagem de Brandin e Wonham [BW94, BWB92], quando se consideram todos os eventos definidos como eventos remotos. Essa restrição determina um modelo que se apresenta com total ausência de controle. Para formalizar esta descrição e tornar seu tratamento semelhante à abordagem não temporizada, a definição dos eventos forçados não é obrigatória. O supervisor apenas deve executar uma ação de controle para o sistema realizar uma tarefa especificada num mínimo tempo definido pela especificação de comportamento.

Dessa forma, devido à semelhança entre os autômatos $(\max, +)$ e os GTAs, pode-se utilizar os autômatos $(\max, +)$ para solucionar o problema de controle da classe de SEDs temporizados descrita anteriormente. Sua utilização elimina o problema do aumento no número de estados do sistema. Assim, a aplicação dos autômatos $(\max, +)$ para esse caso específico, como será vista adiante, determina uma semelhança com a abordagem não temporizada apresentada no Capítulo 3. Desse modo, o paradigma utilizado neste trabalho para a modelagem dos SEDs temporizados é a matriz de incidência temporizada do autômato $(\max, +)$.

Para a compreensão dessa abordagem, são apresentados na próxima seção, os conceitos necessários para o desenvolvimento de um algoritmo, semelhante ao algoritmo de síntese para SEDs não temporizados apresentado no Capítulo 3, que soluciona o problema de controle dos SEDs temporizados, com a restrição de que os tempos de vida dos eventos são definidos como os tempos mínimos para suas ocorrências.

4.2 Conceitos

4.2.1 Álgebra $(\max, +)$

A álgebra $(\max, +)$ é um caso particular da álgebra de dióides na qual $D = \mathbb{R}_{\max}$, onde \mathbb{R}_{\max} denota o conjunto $\mathbb{R} \cup \{-\infty\}$. Para um dióide (D, \oplus, \otimes) , com $D = \mathbb{R}_{\max}$, as operações \oplus e \otimes são: \max (máximo) e $+$ (soma), respectivamente. O elemento nulo é definido como $\epsilon = -\infty$ e o elemento identidade é definido por $e = 0$.

Nesta álgebra, $-\infty$ é utilizado como o elemento nulo da adição, pois satisfaz:

$$\forall a \in \mathbb{R}, a \oplus \epsilon = a \oplus (-\infty) = \max\{a, -\infty\} = a = \max\{-\infty, a\} = (-\infty) \oplus a = \epsilon \oplus a. \quad (4.1)$$

Por outro lado, utiliza-se $e = 0$, como sendo o elemento identidade da multiplicação, pois

satisfaz

$$\forall a \in \mathbb{R}, a \otimes e = a + 0 = a = 0 + a = e \otimes a. \quad (4.2)$$

Essa estrutura algébrica, é um dióide comutativo, pois a propriedade da comutatividade da multiplicação \otimes é satisfeita.

No contexto da álgebra $(\max, +)$, o conjunto D pode ser definido sobre matrizes quadradas de dimensão n , isto é,

$$D = \mathbb{R}_{\max}^{n \times n}. \quad (4.3)$$

As propriedades do dióide $(\max, +)$ são igualmente satisfeitas, considerando que para duas matrizes $\mathbf{A}, \mathbf{B} \in \mathbb{R}_{\max}^{n \times n}$,

$$(\mathbf{A} \oplus \mathbf{B})_{i,j} = \mathbf{A}_{i,j} \oplus \mathbf{B}_{i,j} \quad (4.4)$$

$$(\mathbf{A} \otimes \mathbf{B})_{i,j} = \bigoplus_{k=1}^n (\mathbf{A}_{i,k} \otimes \mathbf{B}_{k,j}). \quad (4.5)$$

4.2.2 Autômatos temporizados $(\max, +)$

Um autômato temporizado é um autômato que inclui uma representação explícita do tempo na sua estrutura de transição. Em um autômato temporizado, um intervalo de tempo está relacionado com cada função de transição, denominado tempo de vida do evento. Este tempo de vida expressa o menor tempo que deve transcorrer para que o evento se torne habilitado e possa ocorrer. Um autômato temporizado reconhece uma linguagem temporizada [JM94, EAM97, Asa98, CJ99, ATP00], que é uma linguagem formalizada sobre pares (*evento*, *tempo*).

Dentre as estruturas de autômatos temporizados, como os GTAs e os GTTs, encontra-se a formalização do autômato $(\max, +)$ [Gau93, Gau95a]:

Definição 4.1 *Um autômato finito $A_{(\max, +)}$ sobre um alfabeto Σ é uma quádrupla*

$$A_{(\max, +)} = (Q, \theta, T, \phi)$$

onde Q é um conjunto finito de estados e θ, T e ϕ são mapas

$$\begin{aligned} \theta &: Q \rightarrow \mathbb{R}_{\max}, \\ T &: Q \times \Sigma \times Q \rightarrow \mathbb{R}_{\max}, \\ \phi &: Q \rightarrow \mathbb{R}_{\max} \end{aligned} \quad (4.6)$$

denominados *atraso inicial*, *tempo de transição* e *atraso final*, respectivamente.

Um autômato $A_{(\max, +)}$ é representado graficamente por um multigrafo valorado, definido por vértices, formados pelo conjunto de estados Q e por três tipos de arcos

1. Os arcos internos, $i \xrightarrow{\sigma} j$, $\forall i, j \in Q$ e $\sigma \in \Sigma$ tais que $T_{i,\sigma,j} \neq \epsilon$. O arco $i \xrightarrow{\sigma} j$, é valorado pelo escalar $T_{i,\sigma,j}$;

2. Os arcos de entrada $\rightarrow i$, valorados por θ_i , $\forall i \in Q$ tal que $\theta_i \neq \epsilon$;
3. Os arcos de saída $i \rightarrow$, valorados por ϕ_i , $\forall i \in Q$ tal que $\phi_i \neq \epsilon$.

Deve-se observar na representação gráfica do autômato $(\max, +)$ que apenas são apresentados os arcos de entrada e de saída em que $\theta_i \neq \epsilon$ e $\phi_i \neq \epsilon$. Ou seja, os atrasos iniciais e finais finitos são os que são representados graficamente, o que determina os estados inicial e marcados do autômato $(\max, +)$.

Exemplo 4.1 Seja $\Sigma = \{\alpha, \beta\}$. O autômato com conjunto de estados $Q = \{0, 1, 2\}$, tempos de transições

$$\begin{aligned} T_{0,\alpha,1} &= 1, \\ T_{0,\alpha,2} &= 3, \\ T_{1,\alpha,2} &= 4, \\ T_{2,\beta,2} &= 1, \\ T_{2,\beta,1} &= 5, \\ T_{2,\beta,0} &= 7, \\ T_{1,\beta,1} &= 1, \\ T_{1,\beta,0} &= 2, \end{aligned}$$

atrasos final e inicial $\phi_0 = 2$ e $\theta_0 = \epsilon = 0$, respectivamente (os outros valores de ϕ , T e θ são ϵ) está representado na Figura 4.1. Este autômato é um modelo de um sistema de

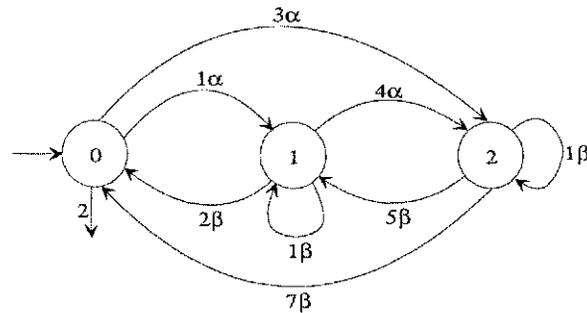


Figure 4.1: Autômato $(\max, +)$.

armazenamento de recursos não determinístico, onde os eventos α e β , representam entradas e saídas de peças. Os arcos valorados com zero (ou ϵ) podem ser omitidos, isto é, o arco de entrada $\rightarrow 0$, não valorado, define $\theta_0 = \epsilon$.

Comparando o autômato $(\max, +)$ com o autômato não temporizado (Definição 2.8), observa-se que as funções de transição dos autômatos $(\max, +)$ são incluídas no mapa T . O estado inicial é definido pelo estado $q \in Q$, em que $\theta_q \neq \epsilon$. Os estados marcados são os estados $q \in Q$, em que $\phi_q \neq \epsilon$. Dessa forma, a semântica do autômato $(\max, +)$ é a seguinte:

1. Há um relógio global que está sempre sendo incrementado;

2. O tempo de vida de um evento, denotado por $T_{q,\sigma,q'} \in T$, é o tempo mínimo necessário para sua habilitação;
3. Para iniciar a execução do autômato (alcançar o estado inicial) é transcorrido um tempo θ_{q_0} no relógio global;
4. Estando no estado inicial (ou em qualquer outro), os contadores dos eventos definidos neste estado vão sendo decrementados;
5. Quando um contador de um dos eventos definidos no estado é zerado, o evento torna-se habilitado, podendo ocorrer a qualquer instante;
6. Se com a incrementação do relógio global, for zerado o contador de um outro evento definido neste estado, ele também torna-se habilitado;
7. A ocorrência de um evento habilitado reinicializa seu contador e muda o estado do autômato, desabilitando os outros eventos;
8. Ao atingir um estado marcado, ao transcorrer o tempo de atraso final, o autômato (max,+) reconhece este estado, reconhecendo assim a palavra que o levou do estado inicial até este estado;
9. Eventos iguais com diferentes tempos de vida definem não determinismo no autômato.

Exemplo 4.2 A evolução do autômato $A_{(\max,+)}$ da Figura 4.2 é descrita como a seguir:

- i. Iniciando o relógio global, após decorridas 2 unidades de tempo, o autômato alcança o estado inicial.
- ii. Após uma unidade de tempo o autômato pode reconhecer a palavra 3ϵ .
- iii. Após mais uma unidade de tempo, o evento α torna-se habilitado.
- iv. Estando o evento α habilitado, ele permanece habilitado indefinidamente até sua ocorrência que muda o estado do autômato para o estado 2, reinicializando seu contador.
- v. No estado 2, após 3 unidades de tempo, o contador do evento β é zerado, habilitando-o.
- vi. A ocorrência do evento β no estado 2, leva o autômato ao estado 1.
- vii. Novamente no estado 1, a passagem de mais uma unidade de tempo permite o reconhecimento da palavra $8\alpha\beta$.
- viii. Considerando que β não ocorre no estado 2, a passagem de mais uma unidade de tempo habilita κ . Dessa forma, tanto o evento β , como o evento κ permanecem habilitados indefinidamente, até a ocorrência de um deles.

- ix. Se ocorre κ , o autômato alcança o estado 3.
- x. Estando no estado 3, a passagem de 2 unidades de tempo habilita β , que permanece habilitado até sua ocorrência.
- xi. Se ocorre β no estado 3, o autômato alcança o estado 1.
- xii. Novamente no estado 1, a passagem de mais uma unidade de tempo permite o reconhecimento da palavra $11\alpha\kappa\beta$.

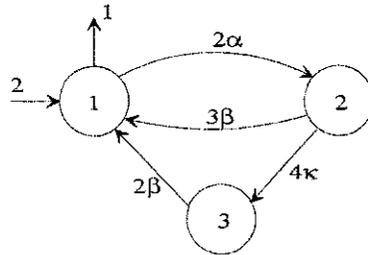


Figure 4.2: Autômato (max,+).

O reconhecimento de um caminho em um autômato (max,+) é descrito como a seguir:

Definição 4.2 Seja um autômato $A_{(\max,+)}$ e Q^n o conjunto de todas as seqüências de estados de comprimento n , definido por

$$Q^n = \{p | p = (q_0, \dots, q_n) \wedge q_0, \dots, q_n \in Q\}. \quad (4.7)$$

Define-se que uma palavra

$$s = \sigma_1 \cdots \sigma_n$$

é reconhecida no caminho

$$p = (q_0, \dots, q_n) \in Q^n,$$

se

$$P(p, s) := \theta_{q_0} + T_{q_0, \sigma_1, q_1} + \cdots + T_{q_{n-1}, \sigma_n, q_n} + \phi_{q_n} \neq \epsilon, \quad (4.8)$$

onde P é a função peso do caminho p .

De acordo com esta definição, pode-se observar que o reconhecimento de uma palavra se dá quando um estado marcado é atingido e o tempo de atraso final é transcorrido (ver Exemplo 4.2). O reconhecimento de um caminho pode também ser descrito por meio das seqüências de eventos através dos datadores. Estes são definidos a seguir:

Definição 4.3 Um datador é um mapa

$$Y : \Sigma^* \rightarrow \mathbb{R}_{\max} \quad (4.9)$$

onde Y é o tempo que um autômato $A_{(\max,+)}$ leva para percorrer uma seqüência

$$s = \sigma_1 \sigma_2 \dots \sigma_n \in \Sigma^*.$$

Dessa definição, vê-se que a função datadora define os tempos das palavras da linguagem Σ^* . Assim, o autômato $A_{(\max,+)}$ leva um tempo Y para sair de um estado q para um outro estado q' , seguindo uma seqüência s , onde $(y|s)$ denota o valor de Y na palavra s [BR88].

Definição 4.4 Um datador Y é dito reconhecível se existir um autômato $A_{(\max,+)}$ tal que

$$(y|s) \neq \epsilon. \tag{4.10}$$

Exemplo 4.3 No Exemplo 4.2, vê-se que $s = \alpha\beta$ é uma palavra reconhecida, pois

$$(y|\alpha\beta) = P(p, s) = (\theta_1 + T_{1,\alpha,2} + T_{2,\beta,1} + \phi_0) = 5 \neq \epsilon$$

com $p = (1, 2, 1)$.

Com as definições anteriores, a evolução dinâmica do autômato $(\max,+)$ pode ser representada por um vetor x de tempos das ocorrências dos eventos, igualmente ao autômato temporizado descrito em Cassandras e Lafortune [CL99]. Isto é, o i -ésimo elemento de x é um mapa

$$x_i : \mathbb{N} \rightarrow \mathbb{R}_{\max}, \tag{4.11}$$

que é interpretado como o instante de tempo da n -ésima ocorrência do evento etiquetado por i .

Exemplo 4.4 Na Figura 4.3(a), é apresentada uma parte do diagrama de ocorrências dos eventos do autômato $(\max,+)$ apresentado na Figura 4.3(b), onde $\Sigma = \{\alpha, \beta\}$. Neste dia-

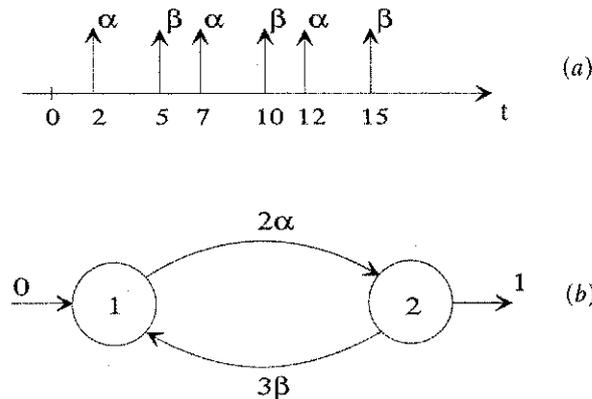


Figure 4.3: (a) Autômato $(\max,+)$ e (b) Diagrama temporal de ocorrência de seus eventos α e β .

grama, vê-se que os menores tempos de ocorrência do evento α são 2, 7 e 12, enquanto que os menores tempos de ocorrência do evento β são 5, 10 e 15. Dessa forma, encontram-se as seqüências

$$\begin{aligned} x_\alpha(1) &= 2, x_\alpha(2) = 7, x_\alpha(3) = 12 \\ x_\beta(1) &= 5, x_\beta(2) = 10, x_\beta(3) = 15. \end{aligned}$$

Logo, o vetor x é definido por

$$x(n) = \begin{bmatrix} x_\alpha(n) \\ x_\beta(n) \end{bmatrix}$$

o qual contém as informações a respeito dos tempos de ocorrência de α e β , dados por

$$\begin{array}{ccc} \begin{bmatrix} 2 \\ 5 \end{bmatrix} & \rightarrow & \begin{bmatrix} 7 \\ 10 \end{bmatrix} & \rightarrow & \begin{bmatrix} 12 \\ 15 \end{bmatrix} \\ x(1) & & x(2) & & x(3) \end{array}$$

A evolução dinâmica do autômato $(\max,+)$ pode ser descrita por meio de um sistema linear através da matriz de tempos de transição A , e pelo vetor x , utilizando a álgebra de dióides [CDQV85, BCOQ92]. Também, as séries formais podem ser utilizadas para a descrição de sua evolução dinâmica e determinação de seus estados reconhecidos.

Deve-se observar que o autômato $(\max,+)$ tem uma formalização similar ao GTA [OW90b], considerando que a representação do tempo é definida diretamente no arco e os tempos de vida são definidos como os tempos mínimos para habilitação dos eventos, isto é, o limite superior é sempre infinito. Dessa forma, os SEDs temporizados descritos em Brandin e Wonham [BW94], considerando apenas os eventos remotos, apresentam a mesma descrição da evolução dinâmica.

Exemplo 4.5 Os autômatos apresentados na Figura 4.4 reconhecem as mesmas linguagens temporizadas. O autômato da Figura 4.4(a) é um grafo de transições de atividades, em que os eventos são definidos por $(\sigma_{q,q'}, t_\sigma, \infty) \in \Sigma$, onde o índice q, q' do evento σ define o estado q onde o evento está habilitado e o estado q' que é alcançado após sua ocorrência, respectivamente, como descrito em [BW94]. Assim, especificamente para este exemplo, tem-se

$$\begin{array}{l} (\alpha_{1,2}, 2, \infty) \\ (\beta_{2,3}, 2, \infty) \\ (\beta_{3,4}, 2, \infty) \\ (\alpha_{4,1}, 3, \infty) \\ (\kappa_{4,5}, 1, \infty) \\ (\lambda_{5,1}, 2, \infty) \end{array}$$

O autômato da Figura 4.4(b) é um grafo de transições temporizadas, que é uma representação da semântica do grafo de transições de atividades da Figura 4.4(a), construído de acordo com Brandin e Wonham [BW94]. O autômato da Figura 4.4(c) é um autômato $(\max,+)$. A semântica desses autômatos é a mesma. Para qualquer um deles, a evolução dinâmica é descrita semelhantemente ao Exemplo 4.2. No caso do grafo de transições temporizadas (Figura 4.4(b)), cada arco que apresenta um evento 'tick' (representado por t), determina a passagem de uma unidade de tempo no relógio global. Observa-se que o número de estados

do autômato $(\max, +)$ é bem menor que o número de estados do grafo de transição temporizada, e apresenta uma forma visual dos tempos direta, em relação ao grafo de transições de atividades.

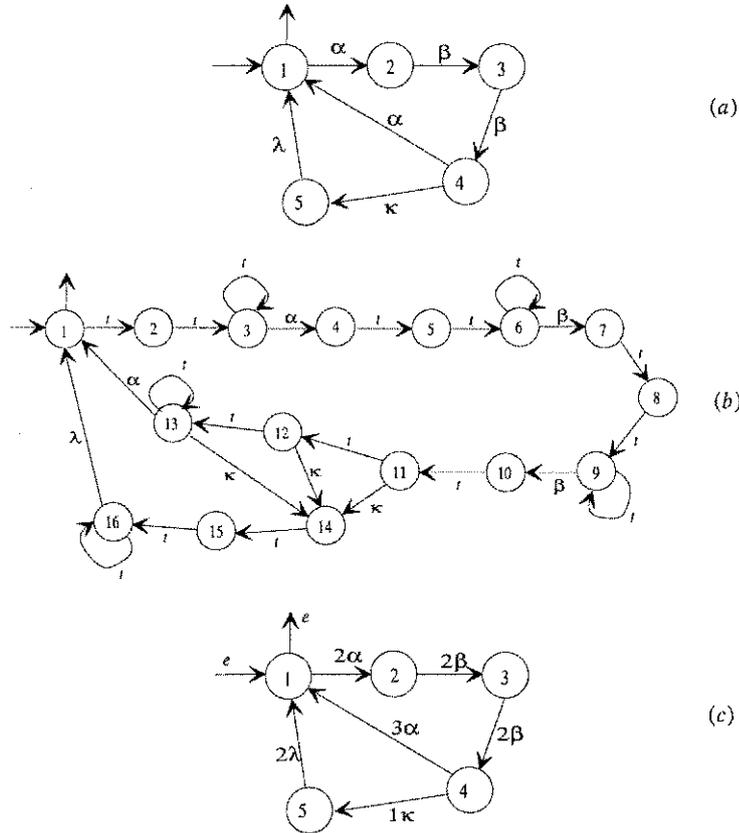


Figure 4.4: (a) Grafo de transições de atividades, (b) Grafo de transições temporizadas e (c) Autômato $(\max, +)$.

4.2.3 Autômatos $(\max, +)$ e séries formais

Uma das formas de utilizar a álgebra de dióides para definir a evolução dos autômatos $(\max, +)$, é usando as séries formais para descrever suas linguagens temporizadas. Assim, semelhantemente às séries formais binárias, definindo $D = \mathbb{R}_{\max} \langle \langle \Sigma \rangle \rangle$, representa-se uma linguagem temporizada por uma série formal. Deve-se observar que uma linguagem temporizada [AH92, EAM97, Tri98, Fri98, Asa98] é uma linguagem que apresenta em cada palavra $s \in L$ um valor numérico t_s associado, que representa um intervalo de tempo decorrido para a palavra ser reconhecida por um autômato temporizado [AD90, ACD93, AD94, AD95, Bal96, Alu97, OHY⁺99, ATP00].

Definição 4.5 Uma linguagem temporizada $L = \{t_s s, t_{s'} s', \dots\}$, com $\{s, s', \dots\} \in \Sigma^*$ e $t_s, t_{s'}, \dots \in \mathbb{R}_{\max}$, pode ser representada por uma série formal

$$Y_L = \bigoplus_{s \in \Sigma^*} (y | s) s \quad (4.12)$$

em que $(y|s) s \in \mathbb{R}_{\max} \langle\langle \Sigma \rangle\rangle$, com $(y|s) \in \mathbb{R}_{\max}$ denotando o datador da palavra s e $\mathbb{R}_{\max} \langle\langle \Sigma \rangle\rangle$ é o semi-anel das séries formais com coeficientes em \mathbb{R}_{\max} e variáveis não comutativas em Σ , tal que

$$(y|s) \neq \epsilon, \text{ se } s \in L, \quad (4.13)$$

$$(y|s) = \epsilon, \text{ se } s \notin L. \quad (4.14)$$

Exemplo 4.6 Dado o alfabeto $\Sigma = \{\alpha, \beta\}$, a linguagem $L = \{3\epsilon, 4\alpha, 2\alpha\beta, 3\beta\alpha, 5\alpha\alpha, 2\beta\beta, \beta\alpha\beta\}$ é representada pela série formal

$$Y_L = 3\epsilon \oplus 4\alpha \oplus 2\alpha\beta \oplus 3\beta\alpha \oplus 5\alpha\alpha \oplus 2\beta\beta \oplus e\beta\alpha\beta \underbrace{\oplus \epsilon\alpha\alpha\alpha \oplus \dots \oplus \epsilon\beta\beta\beta \oplus \dots}_{\text{seqüências de eventos não pertencentes à } L}.$$

ou

$$Y_L = 3\epsilon \oplus 4\alpha \oplus 2\alpha\beta \oplus 3\beta\alpha \oplus 5\alpha\alpha \oplus 2\beta\beta \oplus e\beta\alpha\beta$$

$$Y_L = 3 \oplus 4\alpha \oplus 2\alpha\beta \oplus 3\beta\alpha \oplus 5\alpha\alpha \oplus 2\beta\beta \oplus \beta\alpha\beta$$

desde que $\epsilon \otimes L = \epsilon$, $\forall L \subseteq \Sigma^*$. Esta série representa a linguagem reconhecida por um autômato $(\max, +)$. Neste contexto, $(y|s)$ denota o coeficiente da palavra s que é igual ao zero ' ϵ ' se s não é reconhecida por $A_{(\max, +)}$.

Com essa formalização, é possível avaliar a evolução dinâmica dos autômatos $(\max, +)$, como citado anteriormente.

Definição 4.6 O mapa

$$\mu : \Sigma \rightarrow \mathbb{R}_{\max}^{|Q| \times |Q|}, \quad (4.15)$$

define a aplicação dos valores $T_{q, \sigma, q'}$ dos símbolos $\sigma \in \Sigma$ sobre $\mathbb{R}_{\max}^{|Q| \times |Q|}$, onde $\mathbb{R}_{\max}^{|Q| \times |Q|}$ é o dióide $(\mathbb{R} \cup \{-\infty\}, \max, +)$ sobre matrizes quadradas de dimensão $|Q|$.

Com a aplicação deste mapa, constrói-se a matriz

$$\mu(\sigma)_{qq'} := T_{q, \sigma, q'}. \quad (4.16)$$

Identificando θ com um vetor linha contendo os arcos de entrada e os demais elementos iguais a ϵ , e ϕ com um vetor coluna contendo os arcos de saída e os demais elementos iguais a ϵ , tem-se que:

Definição 4.7 Dado um autômato $A_{(\max, +)}$, com matriz

$$\mu(\sigma)_{qq'} := T_{q, \sigma, q'}, \quad (4.17)$$

e vetores linha θ e coluna ϕ , o datador da palavra $s = \sigma_1 \dots \sigma_n$ é descrito por

$$(y|s) = (A_{(\max, +)}|s) = \theta \mu(\sigma_1) \dots \mu(\sigma_n) \phi = \theta \mu(s) \phi \quad (4.18)$$

em que $\mu(s)$ é a matriz que contém os tempos de complementação das palavras formadas por $\sigma_1 \dots \sigma_n \in \Sigma$.

Denomina-se de representação linear do datador a condição

$$(A_{(\max,+)}|s) = \theta\mu(s)\phi. \quad (4.19)$$

Também, a função datadora Y pode ser escrita como uma série formal sobre o alfabeto Σ com coeficientes em \mathbb{R}_{\max} . Isto é,

Definição 4.8 A linguagem temporizada de um autômato $A_{(\max,+)}$ é definida por uma série formal como

$$L(A_{(\max,+)}) = \bigoplus_{s \in \Sigma^*} (y|s)s, \quad (4.20)$$

em que $(y|s)s \in \mathbb{R}_{\max} \langle\langle \Sigma \rangle\rangle$, com $(y|s) \in \mathbb{R}_{\max}$ denotando o datador da palavra s e $\mathbb{R}_{\max} \langle\langle \Sigma \rangle\rangle$ é o semi-anel das séries formais com coeficientes em \mathbb{R}_{\max} e variáveis não comutativas em Σ , tal que

$$(y|s) \neq \epsilon, \text{ se } s \in L, \quad (4.21)$$

$$(y|s) = \epsilon, \text{ se } s \notin L. \quad (4.22)$$

Assim, equipando a série formal com as operações de soma e produto de *Cauchy* (equações (3.2) e (3.3)) e com a operação estrela (Definição 3.5), e identificando o mapa datador Y com a série formal Y_L , define-se:

Definição 4.9 A série formal Y_L é reconhecível se existir um autômato finito $A_{(\max,+)}$, representado pela tripla (θ, μ, ϕ) , com $\theta \in \mathbb{R}_{\max}^{1 \times |Q|}$, $\phi \in \mathbb{R}_{\max}^{|Q| \times 1}$, $\mu : \Sigma^* \rightarrow \mathbb{R}_{\max}^{|Q| \times |Q|}$ e Q finito, tal que

$$Y_L = \bigoplus_{s \in \Sigma^*} \theta\mu(s)\phi s = \bigoplus_{s \in \Sigma^*} (y|s)s, \quad (4.23)$$

em que $\theta\mu(s)\phi = (A_{(\max,+)}|s)$ é o datador da palavra s .

Exemplo 4.7 Considerando o autômato da Figura 4.5, tem-se

$$\theta = \begin{bmatrix} e & \epsilon & \epsilon \end{bmatrix} \quad \phi = \begin{bmatrix} 2 \\ \epsilon \\ \epsilon \end{bmatrix}$$

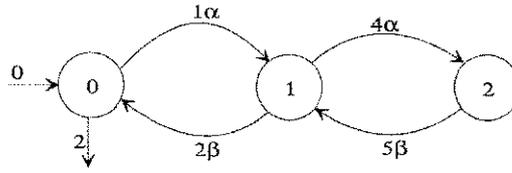
$$\mu(\alpha) = \begin{bmatrix} \epsilon & 1 & \epsilon \\ \epsilon & \epsilon & 4 \\ \epsilon & \epsilon & \epsilon \end{bmatrix} \quad \mu(\beta) = \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ 2 & \epsilon & \epsilon \\ \epsilon & 5 & \epsilon \end{bmatrix}.$$

Calculando $(y|s)$ em um caso particular, onde $s = \alpha\beta$:

$$(y|\alpha\beta) = \theta\mu(\alpha\beta)\phi = \theta\mu(\alpha)\mu(\beta)\phi = 5.$$

Mais geralmente, a série formal Y_L identificada com a linguagem temporizada do autômato admite a expressão racional

$$L(A_{(\max,+)}) = Y_L = 2(3\alpha(9\alpha\beta)^*\beta)^* = 2 \oplus 5\alpha\beta \oplus 8\alpha\beta\alpha\beta \oplus 14\alpha\alpha\beta\beta \oplus \dots$$

Figure 4.5: Autômato $(\max, +)$ determinístico.

isto é,

$$L(A_{(\max,+)}) = Y_L = (y | (\alpha\beta)^0) (\alpha\beta)^0 \oplus (y | (\alpha\beta)^1) (\alpha\beta)^1 \oplus (y | (\alpha\beta)^2) (\alpha\beta)^2 \oplus \dots$$

$$L(A_{(\max,+)}) = Y_L = \bigoplus_{s \in \Sigma^*} (y | s) s.$$

Assim, $L(A_{(\max,+)}) = Y_L$ apresenta todas as palavras formadas por α e β , iniciadas por α , e reconhecidas pelo autômato da Figura 4.5, com seus respectivos tempos de complementação.

4.3 Síntese do supervisor

Com a formalização apresentada no Capítulo 3, e utilizando os conceitos apresentados na seção anterior, aqui será apresentado um formalismo de síntese do supervisor para um SED temporizado. Especificamente, o problema aqui tratado, refere-se aos sistemas que se apresentam com tempos de vida de eventos definidos como os mínimos tempos para suas habilitações. Em outros termos, é uma aplicação a um caso restrito da abordagem de Brandin e Wonham [BW94], em que todos os eventos são definidos como eventos remotos (limite superior de tempo de ocorrência igual a infinito). Assim, considerar-se-á que, igualmente à TCS, o alfabeto de eventos Σ é particionado em eventos controláveis Σ_c e eventos não controláveis Σ_{uc} , satisfazendo

$$\Sigma = \Sigma_c \cup \Sigma_{uc}$$

e

$$\Sigma_c \cap \Sigma_{uc} = \emptyset.$$

Como essa abordagem é um formalismo unificado para a síntese de supervisores de SEDs não temporizados e temporizados, é necessário apresentar as definições das matrizes de incidência temporizadas, para contextualizar esse Capítulo com a abordagem apresentada no Capítulo 3.

4.3.1 Matrizes de incidência temporizadas

O formalismo para sintetizar um supervisor para um SED temporizado, requer a definição de uma matriz de incidência que apresente não só os eventos, como é o caso da Definição 3.7, mas também seus tempos de vida. Assim, tem-se:

Definição 4.10 Seja $A_{(\max,+)}$ um autômato temporizado. Defina-se a matriz de incidência temporizada, denotada por \mathbf{At} , como

$$\mathbf{At} = [at_{i,j}]; at_{i,j} = \begin{cases} t_\sigma \sigma & \text{se } \exists \sigma \text{ do estado } i \text{ para o estado } j; \\ \epsilon & \text{caso contrário,} \end{cases}$$

em que t_σ é o tempo de vida do evento σ que leva o autômato $A_{(\max,+)}$ do estado i para o estado j . Se mais de um evento é definido do estado i para o estado j , $at_{i,j} = \bigoplus_k t_{\sigma^k} \sigma^k$, para $\sigma^k \in \Sigma$, $k = 1, 2, \dots$, de modo que qualquer evento σ^k ocorrendo, provoca no autômato $A_{(\max,+)}$ a mudança do estado i para o estado j . O estado inicial é definido como sendo o estado 1, representado pelo vetor linha

$$\theta(\mathbf{At}) = [t_{in} \ \epsilon \ \dots \ \epsilon],$$

com t_{in} representando o atraso inicial, e os estados marcados são representados pelo vetor coluna

$$\phi(\mathbf{At}) = [t_{m_1} \ t_{m_2} \ \dots \ t_{m_n}]^T,$$

com t_{m_i} representando os atrasos finais.

Pode-se ver que a representação para um autômato temporizado exige a descrição do vetor θ , pois é nele que está descrito o atraso inicial. Os atrasos finais são os valores t_{m_i} , em que se um estado k não é marcado, $t_{m_k} = \epsilon$.

Por conveniência, daqui por diante, quando for feita referência a um elemento do vetor θ , será utilizada a notação θ_j , referindo-se à sua j -ésima coluna. Similarmente, toda referência a um elemento do vetor ϕ será feita utilizando a notação ϕ_i , referindo-se à sua i -ésima linha.

Exemplo 4.8 O autômato $A_{(\max,+)}$ apresentado na Figura 4.6, tem sua matriz de incidência temporizada construída de acordo com a Definição 4.10, a qual é dada por

$$\mathbf{At} = \begin{bmatrix} \epsilon & 3\alpha & \epsilon \\ 4\beta & \epsilon & 3\alpha \\ 2\kappa & \epsilon & \epsilon \end{bmatrix},$$

com o estado inicial definido por

$$\theta(\mathbf{At}) = [2 \ \epsilon \ \epsilon]$$

e os estados marcados definidos pelo vetor coluna

$$\phi(\mathbf{At}) = \begin{bmatrix} 2 \\ 3 \\ \epsilon \end{bmatrix}.$$

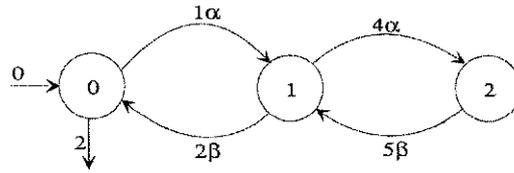


Figure 4.6: Autômato $(\max, +)$ determinístico para ilustrar a representação por matriz de incidência temporizada e vetores de estados inicial e marcados.

Deve-se observar que a matriz de incidência temporizada pode ser construída por

$$\mathbf{At} = \bigoplus_{i=1}^n \mu(\sigma_i) \otimes \sigma_i$$

com $\mu(\sigma_i)$ dada pela Definição 4.6, e σ_i sendo o i -ésimo evento do alfabeto Σ . Sendo assim, para a representação de um autômato $A_{(\max,+)}$ por uma matriz de incidência temporizada e pelos seus vetores de estados, a descrição em termos das séries formais é válida, podendo-se através dessa representação, determinar a linguagem temporizada reconhecida pelo autômato $A_{(\max,+)}$. No caso da representação matricial, define-se:

Definição 4.11 *A um autômato $A_{(\max,+)}$ que tem sua linguagem reconhecida temporizada representada pela série formal*

$$Y_L = \bigoplus_{s \in \Sigma^*} (y|s) s, \tag{4.24}$$

para a representação matricial de $A_{(\max,+)}$ dada por \mathbf{At} , $\theta(\mathbf{At})$ e $\phi(\mathbf{At})$, define-se sua linguagem marcada temporizada como sendo

$$L_m(\mathbf{At}) = L(A_{(\max,+)}) = Y_L. \tag{4.25}$$

Semelhantemente à construção das linguagens das matrizes de incidência não temporizadas, podem-se definir as linguagens da matriz de incidência temporizada, de forma a tornar essa abordagem unificada. Para isto, define-se:

Definição 4.12 *Seja \mathbf{At} uma matriz de incidência, cujos elementos $at_{i,j}$ definem caminhos de comprimento 1, que mudam o estado do autômato que ela representa, do estado i para o estado j , com um tempo de vida t_σ . Assim, a matriz*

$$\mathbf{At}^n = \mathbf{At} \otimes \mathbf{At} \otimes \dots \otimes \mathbf{At}, \tag{4.26}$$

é uma matriz de caminhos, onde cada elemento $at_{i,j}^n$ representa um ou mais caminhos de comprimento n , formado de eventos controláveis ou não controláveis, que levam o autômato que ela representa, do estado i para o estado j , com um tempo total $t_s = t_{\sigma^1} + \dots + t_{\sigma^n}$, $s = \sigma^1 \dots \sigma^n$. Os vetores de estado inicial e de estados marcados da matriz de caminhos \mathbf{At}^n são os mesmos de \mathbf{At} .

Observa-se que, igualmente à matriz de caminhos não temporizada, na matriz de caminhos temporizada, quando não há um caminho com n eventos que muda o estado do autômato do estado i para o estado j , tem-se $at_{i,j}^n = \epsilon$. A matriz \mathbf{At}^n contém palavras s de comprimento n , compostas de eventos controláveis e não controláveis, que são percorridas em um tempo $t_s = t_{\sigma^1} + \dots + t_{\sigma^n}$.

Exemplo 4.9 Do autômato $A_{(\max,+)}$ mostrado na Figura 4.7, sua representação matricial é dada por

$$\mathbf{At} = \begin{bmatrix} \epsilon & 4\alpha & \epsilon \\ 5\beta & \epsilon & 2\mu \\ \epsilon & 2\beta & 2\alpha \end{bmatrix}, \quad \theta(\mathbf{At}) = \begin{bmatrix} 1 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{At}) = \begin{bmatrix} 3 \\ \epsilon \\ 2 \end{bmatrix}.$$

A matriz de caminhos \mathbf{At}^2 desse autômato é

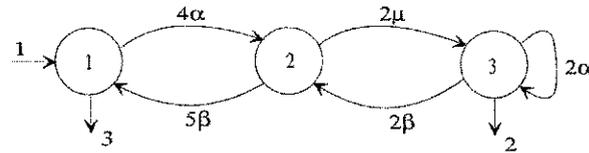


Figure 4.7: Autômato para exemplo da matriz de caminhos temporizada.

$$\mathbf{At}^2 = \mathbf{At} \otimes \mathbf{At} = \begin{bmatrix} 9\alpha\beta & \epsilon & 6\alpha\mu \\ \epsilon & 4\mu\beta + 9\beta\alpha & 4\mu\alpha \\ 7\beta\beta & 4\alpha\beta & 4\alpha\alpha + 4\beta\mu \end{bmatrix}$$

com $\theta(\mathbf{At}^2) = \theta(\mathbf{At})$ e $\phi(\mathbf{At}^2) = \phi(\mathbf{At})$. Sua matriz de caminhos \mathbf{At}^3 é

$$\mathbf{At}^3 = \begin{bmatrix} \epsilon & 13\alpha\beta\alpha + 8\alpha\mu\beta & 8\alpha\mu\alpha \\ 9\mu\beta\beta + 14\beta\alpha\beta & 6\alpha\mu\beta & 6\mu\alpha\alpha \\ 9\alpha\beta\beta & 11\beta\beta\alpha + 6\alpha\alpha\beta + 6\beta\mu\beta & 6\alpha\beta\mu + 6\alpha\alpha\alpha + 6\beta\mu\alpha \end{bmatrix}$$

com $\theta(\mathbf{At}^3) = \theta(\mathbf{At})$ e $\phi(\mathbf{At}^3) = \phi(\mathbf{At})$. Nessas matrizes de caminhos, cada seqüência representa uma palavra de comprimento 2 e 3, respectivamente, que muda o estado do autômato $A_{(\max,+)}$, do estado i para o estado j . Em \mathbf{At}^2 , vê-se que do estado inicial para o estado inicial, existe a palavra $\alpha\beta$, com tempo $t_{\alpha\beta} = 9$, e que é uma palavra reconhecida, pois

$$(y|\alpha\beta) = \theta(\mathbf{At}) \otimes \mu(\alpha) \otimes \mu(\beta) \otimes \phi(\mathbf{At}) = 13 \neq \epsilon,$$

de acordo com a Definição 4.7.

Com essa definição e utilizando a álgebra de dióides, tem-se que a linguagem da matriz de incidência temporizada $L(\mathbf{At})$ é o conjunto de todas as palavras formadas por seqüências iniciadas da linha 1, que levam a qualquer linha da matriz \mathbf{At} , com um tempo igual à soma de todos os tempos de vida dos eventos da seqüência com o atraso inicial. Isto é,

Definição 4.13 Para uma dada matriz de incidência temporizada \mathbf{At} , sua linguagem é definida como

$$L(\mathbf{At}) = \bigoplus_i (\theta(\mathbf{At}) \otimes \mathbf{At}^i) = \bigoplus_i \bigoplus_{j=1}^n (\theta_1(\mathbf{At}) \otimes at_{1,j}^i), \quad (4.27)$$

onde $\theta_1(\mathbf{At})$ é o elemento da primeira coluna do vetor de estado inicial $\theta(\mathbf{At})$, $at_{1,j}^i$ é o elemento da linha 1, coluna j da matriz de caminhos temporizada \mathbf{At}^i .

Com essa definição, pode-se determinar qual estado é alcançado quando ocorre a partir do estado 1 uma seqüência s de eventos de comprimento $|s|$ com seu respectivo tempo, como a seguir

Definição 4.14 Seja \mathbf{At} uma matriz de incidência temporizada. A sublinguagem de \mathbf{At} , que contém todas as palavras que mudam o estado do autômato representado pela matriz \mathbf{At} para o estado k é definida como

$$L(\mathbf{At})^k = \bigoplus_i (\theta(\mathbf{At}) \otimes \mathbf{At}^i \otimes \pi) = \bigoplus_i (\theta_1(\mathbf{At}) \otimes at_{1,k}^i), \quad (4.28)$$

onde π é um vetor coluna com o k -ésimo elemento igual ao elemento identidade 'e' e os demais ϵ , e $\theta_1(\mathbf{At})$ é o elemento da primeira coluna do vetor de estado inicial $\theta(\mathbf{At})$, $at_{1,k}^i$ é o elemento da linha 1, coluna k da matriz de caminhos temporizada \mathbf{At}^i .

O conjunto de todas as palavras de comprimento $|s| = i$ é determinado por

$$L(\mathbf{At})_i = \theta(\mathbf{At}) \otimes \mathbf{At}^i = \bigoplus_{j=1}^n (\theta_1(\mathbf{At}) \otimes at_{1,j}^i) \quad (4.29)$$

e, o conjunto de todas as palavras de comprimento $|s| = i$ que levam do estado 1 para o estado k é determinado por

$$L(\mathbf{At})_i^k = \theta(\mathbf{At}) \otimes \mathbf{At}^i \otimes \pi = \theta_1(\mathbf{At}) \otimes at_{1,k}^i. \quad (4.30)$$

Por outro lado, além do formalismo das séries formais, a linguagem marcada da matriz de incidência \mathbf{At} também pode ser definida como sendo:

Definição 4.15 Para uma matriz de incidência \mathbf{At} , sua linguagem marcada é definida como

$$L_m(\mathbf{At}) = \bigoplus_i (\theta(\mathbf{At}) \otimes \mathbf{At}^i \otimes \phi(\mathbf{At})) = \bigoplus_i \bigoplus_{j=1}^n (\theta_1(\mathbf{At}) \otimes at_{1,j}^i \otimes \phi_j(\mathbf{At})), \quad (4.31)$$

onde $\theta_1(\mathbf{At})$ é o elemento da primeira coluna do vetor de estado inicial $\theta(\mathbf{At})$, $at_{1,j}^i$ é o elemento da linha 1, coluna marcada j da matriz de caminhos temporizada \mathbf{At}^i e $\phi_j(\mathbf{At})$ é o elemento da j -ésima linha do vetor de estados marcados $\phi(\mathbf{At})$.

Com essa definição, pode-se determinar se um estado marcado é alcançado quando ocorre, a partir do estado 1, uma seqüência s de eventos de comprimento $|s|$, como a seguir:

Definição 4.16 *Seja \mathbf{At} uma matriz de incidência temporizada. A sublinguagem de \mathbf{At} , que contém todas as palavras que mudam o estado do autômato representado pela matriz \mathbf{At} para o estado marcado k é definida como*

$$L(\mathbf{At})_m^k = \bigoplus_i (\theta(\mathbf{At}) \otimes \mathbf{At}^i \otimes \vartheta) = \bigoplus_i (\theta_1(\mathbf{At}) \otimes at_{1,k}^i \otimes \phi_k(\mathbf{At})), \quad (4.32)$$

onde ϑ é um vetor coluna tal que $\vartheta_k = \phi_k(\mathbf{At})$ e os demais elementos são iguais a ϵ ; $\theta_1(\mathbf{At})$ é o elemento da primeira coluna do vetor de estado inicial $\theta(\mathbf{At})$, $at_{1,k}^i$ é o elemento da linha 1, coluna marcada k da matriz de caminhos temporizada \mathbf{At}^i e $\phi_k(\mathbf{At})$ é o elemento da k -ésima linha do vetor de estados marcados $\phi(\mathbf{At})$.

Dessa forma, $L(\mathbf{At})_m^k$ contém todas as palavras reconhecidas que levam da linha 1 à linha marcada k .

O conjunto de todas as palavras reconhecidas de comprimento $|s| = i$ é determinado por

$$L_m(\mathbf{At})_i = \theta(\mathbf{At}) \otimes \mathbf{At}^i \otimes \phi(\mathbf{At}) = \bigoplus_{j=1}^n (\theta_1(\mathbf{At}) \otimes at_{1,j}^i \otimes \phi_j(\mathbf{At})) \quad (4.33)$$

e, o conjunto de todas as palavras de comprimento $|s| = i$ que levam do estado 1 para o estado marcado k é determinado por

$$L_m(\mathbf{At})_i^k = \theta(\mathbf{At}) \otimes \mathbf{At}^i \otimes \vartheta = \theta_1(\mathbf{At}) \otimes at_{1,k}^i \otimes \phi_k(\mathbf{At}). \quad (4.34)$$

Exemplo 4.10 *Seja o autômato apresentado na Figura 4.8. Sua representação matricial é*

$$\mathbf{At} = \begin{bmatrix} \epsilon & 5\alpha \\ 3\beta & \kappa \end{bmatrix}, \theta(\mathbf{At}) = \begin{bmatrix} 1 & \epsilon \end{bmatrix}, \phi(\mathbf{At}) = \begin{bmatrix} 3 \\ e \end{bmatrix}.$$

Para determinar sua linguagem, calcula-se

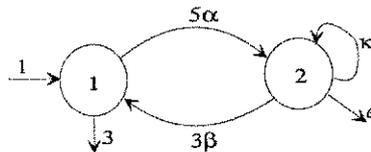


Figure 4.8: Autômato temporizado para determinação da linguagem temporizada.

$$\mathbf{At}^2 = \begin{bmatrix} 8\alpha\beta & 5\alpha\kappa \\ 3\kappa\beta & 8\beta\alpha + \kappa\kappa \end{bmatrix}$$

$$\mathbf{At}^3 = \begin{bmatrix} 8\alpha\kappa\beta & 13\alpha\beta\alpha + 5\alpha\kappa\kappa \\ 11\beta\alpha\beta + 3\kappa\kappa\beta & 8\kappa\beta\alpha + 8\beta\alpha\kappa + \kappa\kappa\kappa \end{bmatrix}$$

⋮

cujos elementos da primeira linha definem as palavras de comprimento 2, 3, ..., da linguagem de \mathbf{At} . Logo, multiplicando essas matrizes por $\theta_1(\mathbf{At})$, encontra-se

$$L(\mathbf{At}) = \{1\varepsilon, 6\alpha, 9\alpha\beta, 6\alpha\kappa, 9\alpha\kappa\beta, 14\alpha\beta\alpha, 6\alpha\kappa\kappa, \dots\}.$$

Observe que a palavra vazia ε está presente em $L(\mathbf{At})$, a qual tem um tempo de vida igual a 1 (tempo de atraso inicial do autômato), e que as palavras de comprimento i , encontram-se na linha 1 das matrizes \mathbf{At}^i . A linguagem marcada desta matriz é determinada pelas palavras da linha 1 de \mathbf{At}^i , multiplicadas por $\theta_1(\mathbf{At})$ e $\phi_k(\mathbf{At})$:

$$L_m(\mathbf{At}) = \{4\varepsilon, 6\alpha, 12\alpha\beta, 6\alpha\kappa, 12\alpha\kappa\beta, 14\alpha\beta\alpha, 6\alpha\kappa\kappa, \dots\},$$

que é a mesma linguagem encontrada utilizando as séries formais, ou seja,

$$L_m(\mathbf{At}) = Y_L = 4\varepsilon \oplus 6\alpha \oplus 12\alpha\beta \oplus 6\alpha\kappa \oplus 12\alpha\kappa\beta \oplus 14\alpha\beta\alpha \oplus 6\alpha\kappa\kappa, \dots$$

Com essas definições, quando forem feitas referências às linguagens da representação matricial do autômato $A_{(\max,+)}$, utilizar-se-á $L(\mathbf{At})$ e $L_m(\mathbf{At})$.

A equivalência entre autômatos representados por matrizes de incidência temporizadas é definida semelhantemente a equivalência de matrizes de incidência não temporizadas. Contudo, inclui-se nessa situação, a representação temporal.

Definição 4.17 Duas matrizes de incidência temporizadas \mathbf{At}_1 e \mathbf{At}_2 , tal que $\forall at_{1,i,j}, at_{2,i,j}$, $at_{1,i,j} \subset \Sigma$ e $at_{2,i,j} \subset \Sigma$, são ditas equivalentes, representado por

$$\mathbf{At}_1 \equiv \mathbf{At}_2 \quad (4.35)$$

se para quaisquer duas seqüências temporizadas s_1 de \mathbf{At}_1 e s_2 de \mathbf{At}_2 , em que $(y|s_1) = (y|s_2)$, então $\exists \sigma|\sigma$ é possível tanto em \mathbf{At}_1 como em \mathbf{At}_2 , tal que $\phi_{i_1}(\mathbf{At}_1) = \phi_{i_2}(\mathbf{At}_2)$ e $(y|s_1\sigma) = (y|s_2\sigma)$.

Exemplo 4.11 As matrizes de incidência temporizadas

$$\mathbf{At}_1 = \begin{bmatrix} \varepsilon & 2\alpha & \varepsilon \\ 2\lambda & \varepsilon & 3\beta \\ \varepsilon & 3\beta & \varepsilon \end{bmatrix}, \quad \theta(\mathbf{At}_1) = \begin{bmatrix} 2 & \varepsilon & \varepsilon \end{bmatrix}, \quad \phi(\mathbf{At}_1) = \begin{bmatrix} \varepsilon \\ 3 \\ 3 \end{bmatrix}$$

$$\mathbf{At}_2 = \begin{bmatrix} \varepsilon & 2\alpha \\ 2\lambda & 3\beta \end{bmatrix}, \quad \theta(\mathbf{At}_2) = \begin{bmatrix} 2 & \varepsilon \end{bmatrix}, \quad \phi(\mathbf{At}_2) = \begin{bmatrix} \varepsilon \\ 3 \end{bmatrix}$$

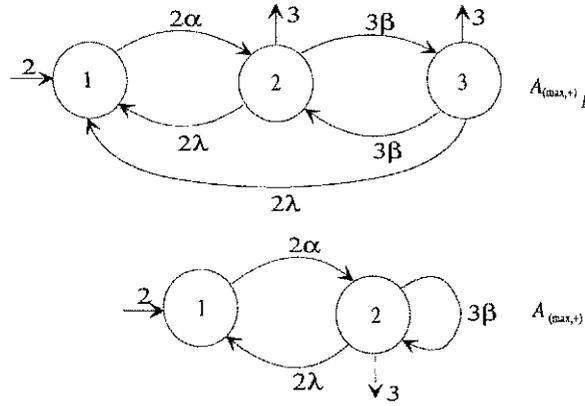
são equivalentes, pois satisfaz a Definição 4.17. Ambas apresentam a mesma linguagem temporizada

$$L(\mathbf{At}_1) = L(\mathbf{At}_2) = \{2\varepsilon, 4\alpha, 6\alpha\lambda, 8\alpha\lambda\alpha, 7\alpha\beta, 10\alpha\beta\beta, \dots\},$$

e reconhecem a mesma linguagem marcada

$$L_m(\mathbf{At}_1) = L_m(\mathbf{At}_2) = \{7\alpha, 11\alpha\lambda\alpha, 7\alpha\beta, 10\alpha\beta\beta, \dots\},$$

as quais podem ser encontradas utilizando as Definições 4.13 e 4.15. Os autômatos que elas representam estão apresentados na Figura 4.9.


 Figure 4.9: Autômatos $(\max, +)$ equivalentes.

Similarmente à composição síncrona de autômatos utilizada na TCS, considerando os autômato $(\max, +)$, nessa abordagem é formalizada semelhantemente à Definição 3.18, considerando que os tempos de vida dos eventos satisfazem às operações de soma e produto de *Cauchy* (equações (3.2) e (3.3)) para $D = \mathbb{R}_{\max}$. Para isto, as seguintes definições são necessárias:

Definição 4.18 Dado um dióide $D = \mathbb{R}_{\max} \langle \langle \Sigma \rangle \rangle$, o operador \otimes define a interseção de elementos, como

$$at \otimes bt = \bigoplus_{i=1}^k (t_{\sigma^i} \oplus t'_{\sigma^i}) \sigma^i, \text{ se } t_{\sigma^i} \sigma^i \subset at \wedge t'_{\sigma^i} \sigma^i \subset bt. \quad (4.36)$$

$\forall at, bt \in D$, onde

$$\begin{aligned} t_{\sigma} \sigma \otimes \epsilon &= \epsilon \\ t_{\sigma^1} \sigma^1 \otimes t'_{\sigma^2} \sigma^2 &= \epsilon, \sigma^1 \neq \sigma^2 \\ t_{\sigma} \sigma \otimes t'_{\sigma} \sigma &= (t_{\sigma} \oplus t'_{\sigma}) \sigma, \end{aligned} \quad (4.37)$$

$\forall \sigma, \sigma^1, \sigma^2 \in \Sigma$ e $\forall t_{\sigma}, t_{\sigma^1}, t'_{\sigma}, t'_{\sigma^2} \in \mathbb{R}_{\max}$. Para matrizes de incidência temporizadas, esse operador é definido como

$$\mathbf{C} = \mathbf{A}t \otimes \mathbf{B}t, c_{i,j} = at_{i,j} \otimes bt_{i,j}, \quad (4.38)$$

para $at_{i,j} \otimes bt_{i,j}$ definido de acordo com 4.36, e

$$\theta(\mathbf{C}) = \theta(\mathbf{A}t) \otimes \theta(\mathbf{B}t) \text{ e } \phi(\mathbf{C}) = \phi(\mathbf{A}t) \otimes \phi(\mathbf{B}t).$$

Observa-se que, da mesma forma que a Definição 3.17, tornando $\sigma = \sigma' = e$ (palavra vazia), a interseção satisfaz as condição da Definição 4.18, ou seja,

$$\begin{aligned} te \otimes \epsilon &= t \otimes \epsilon = \epsilon, \quad e \\ te \otimes t'e &= (t \oplus t')e = (t \oplus t'), \end{aligned}$$

desde que a segunda equação só é válida para $\sigma^1 \neq \sigma^2$. Dessa forma, para o caso dos vetores de estados θ e ϕ , a operação de interseção \otimes só retorna um valor diferente de ϵ

quando todos os elementos operados são diferentes de ϵ . No caso do vetor de estado inicial, esse operador torna-se igual ao operador \oplus , desde que apenas o primeiro elemento dos dois vetores é diferente de ϵ , o que satisfaz

$$\begin{aligned} \theta_1(\mathbf{C}) &= \theta_1(\mathbf{At}) \oplus \theta_1(\mathbf{Bt}) & e \\ \theta_k(\mathbf{C}) &= \epsilon & 1 < k \leq n. \end{aligned}$$

Para o caso específico de controle de SEDs, essa definição deve ser restringida, devido ao fato de não se poder aplicar controle a eventos não controláveis, impondo atrasos. Assim, tem-se:

Definição 4.19 Considerando que \mathbf{At} representa uma especificação de comportamento (tempos dos eventos variáveis) para um SED temporizado modelado por uma matriz de incidência \mathbf{Bt} (tempos dos eventos não controláveis constantes), em que $at_{i,j}$ e $bt_{i,j}$ podem ser expressões regulares do tipo $t_{\sigma^1}\sigma^1 + \dots + t_{\sigma^k}\sigma^k$ e $t'_{\sigma^1}\sigma^1 + \dots + t'_{\sigma^k}\sigma^k$, respectivamente, então o operador de interseção \otimes é definido de acordo com a equação (4.38), em que $at_{i,j} \otimes bt_{i,j}$ é definido como

$$at_{i,j} \otimes bt_{i,j} = \begin{cases} \bigoplus_{i=1}^k (t_{\sigma^i} \oplus t'_{\sigma^i}) \sigma^i & \text{se } (t_{\sigma^i} \sigma^i \subset a) \wedge (t'_{\sigma^i} \sigma^i \subset b) \wedge (\sigma^i \in \Sigma_c) \\ \bigoplus_{i=1}^k t'_{\sigma^i} \sigma^i & \text{se } (t_{\sigma^i} \sigma^i \subset a) \wedge (t'_{\sigma^i} \sigma^i \subset b) \wedge (\sigma^i \in \Sigma_{uc}). \end{cases} \quad (4.39)$$

Com a Definição 4.19, o operador \otimes pode ser utilizado para os procedimentos da síntese do supervisor, como será visto adiante. Isso, porque quando se considera $\Sigma = \Sigma_c \cup \Sigma_{uc}$, tanto na composição síncrona de dois autômatos quanto em outras operações, o tempo de um evento controlável pode ser modificado (apenas ampliado) através da interferência de um supervisor. Porém, no modelo do SED, em hipótese alguma, pode haver variação no tempo de vida de um evento não controlável. Dessa maneira, o produto síncrono de duas matrizes de incidência temporizadas é definido como a seguir:

Definição 4.20 Dadas duas matrizes de incidência temporizadas $\mathbf{At}_{m \times m}$ e $\mathbf{Bt}_{n \times n}$, as quais representam o autômato $A_{(\max,+)_1}$ construído com símbolos do alfabeto Σ_1 e o autômato $A_{(\max,+)_2}$ construído com símbolos do alfabeto Σ_2 , respectivamente, compõe-se uma nova matriz de incidência $\mathbf{Pt}_{m \times m \times n}$, isto é, com dimensão $m \times n$, construída com símbolos dos alfabetos Σ_1 e Σ_2 , através da composição paralela definida como:

$$\mathbf{Pt}_{m \times m \times n} = \mathbf{At} \parallel \mathbf{Bt}, pt_{k,l} = at_{i_{\mathbf{At}}, j_{\mathbf{At}}} \otimes bt_{i_{\mathbf{Bt}}, j_{\mathbf{Bt}}},$$

onde $k = (i_{\text{At}}, i_{\text{Bt}}) = i_{\text{At}} + m(i_{\text{Bt}} - 1)$ e $l = (j_{\text{At}}, j_{\text{Bt}}) = j_{\text{At}} + m(j_{\text{Bt}} - 1)$, de tal forma que

$$\mathbf{Pt}_{m,n \times m,n} = \begin{array}{|c|c|c|c|} \hline at_{1,1} \otimes [bt_{1,1} \ bt_{1,2} \ \dots \ bt_{1,n}] & at_{1,2} \otimes [bt_{1,1} \ bt_{1,2} \ \dots \ bt_{1,n}] & \dots & at_{1,m} \otimes [bt_{1,1} \ bt_{1,2} \ \dots \ bt_{1,n}] \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline at_{m,1} \otimes [bt_{1,1} \ bt_{1,2} \ \dots \ bt_{1,n}] & at_{m,2} \otimes [bt_{1,1} \ bt_{1,2} \ \dots \ bt_{1,n}] & \dots & at_{m,m} \otimes [bt_{1,1} \ bt_{1,2} \ \dots \ bt_{1,n}] \\ \hline at_{1,1} \otimes [bt_{2,1} \ bt_{2,2} \ \dots \ bt_{2,n}] & at_{1,2} \otimes [bt_{2,1} \ bt_{2,2} \ \dots \ bt_{2,n}] & \dots & at_{1,m} \otimes [bt_{2,1} \ bt_{2,2} \ \dots \ bt_{2,n}] \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline at_{m,1} \otimes [bt_{2,1} \ bt_{2,2} \ \dots \ bt_{2,n}] & at_{m,2} \otimes [bt_{2,1} \ bt_{2,2} \ \dots \ bt_{2,n}] & \dots & at_{m,m} \otimes [bt_{2,1} \ bt_{2,2} \ \dots \ bt_{2,n}] \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline at_{1,1} \otimes [bt_{n,1} \ bt_{n,2} \ \dots \ bt_{n,n}] & at_{1,2} \otimes [bt_{n,1} \ bt_{n,2} \ \dots \ bt_{n,n}] & \dots & at_{1,m} \otimes [bt_{n,1} \ bt_{n,2} \ \dots \ bt_{n,n}] \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline at_{m,1} \otimes [bt_{n,1} \ bt_{n,2} \ \dots \ bt_{n,n}] & at_{m,2} \otimes [bt_{n,1} \ bt_{n,2} \ \dots \ bt_{n,n}] & \dots & at_{m,m} \otimes [bt_{n,1} \ bt_{n,2} \ \dots \ bt_{n,n}] \\ \hline \end{array}$$

se $\Sigma_1 = \Sigma_2$. Os estados marcados da composição são definidos por um vetor coluna $\phi_{m,n,1}(\mathbf{Pt})$, em que seu k -ésimo elemento é

$$\begin{aligned} \phi_k(\mathbf{Pt}) &= \phi_{i_{\text{At}}}(\mathbf{At}) \otimes \phi_{i_{\text{Bt}}}(\mathbf{Bt}) \\ &= \begin{cases} \phi_{i_{\text{At}}}(\mathbf{At}) \oplus \phi_{i_{\text{Bt}}}(\mathbf{Bt}) & \text{se } \phi_{i_{\text{At}}}(\mathbf{At}) \neq \epsilon \wedge \phi_{i_{\text{Bt}}}(\mathbf{Bt}) \neq \epsilon \\ \epsilon & \text{se } \phi_{i_{\text{At}}}(\mathbf{At}) = \epsilon \vee \phi_{i_{\text{Bt}}}(\mathbf{Bt}) = \epsilon. \end{cases} \end{aligned}$$

com $k = (i_{\text{At}}, i_{\text{Bt}}) = i_{\text{At}} + m(i_{\text{Bt}} - 1)$. O vetor de estado inicial é definido por

$$\begin{aligned} \theta_{1,m \times n}(\mathbf{Pt}) &= \begin{bmatrix} \theta_1(\mathbf{At}) \otimes \theta_1(\mathbf{Bt}) & \epsilon & \dots & \epsilon \\ \theta_1(\mathbf{At}) \oplus \theta_1(\mathbf{Bt}) & \epsilon & \dots & \epsilon \end{bmatrix} \\ &= \begin{bmatrix} \theta_1(\mathbf{At}) \otimes \theta_1(\mathbf{Bt}) & \epsilon & \dots & \epsilon \\ \theta_1(\mathbf{At}) \oplus \theta_1(\mathbf{Bt}) & \epsilon & \dots & \epsilon \end{bmatrix} \end{aligned}$$

Se $\exists \sigma_{\text{At}} \notin \Sigma_2$, ou $\exists \sigma_{\text{Bt}} \notin \Sigma_1$, de tal forma que $\Sigma_1 \cap \Sigma_2 \neq \emptyset$, então,

$$\mathbf{Pt}_{m,n \times m,n} = (\mathbf{At} \parallel \mathbf{Bt}) \oplus \mathbf{Ct}^\top,$$

onde

$$\mathbf{Ct}^\top = \begin{array}{|c|c|c|c|c|} \hline \mathbf{At}^{-\text{Bt}} \oplus \mathbf{Dt}_{bt_{1,1}}^{-\text{At}} & \mathbf{Dt}_{bt_{1,2}}^{-\text{At}} \oplus \mathbf{At}_l^{-\text{Bt}} & \dots & \dots & \mathbf{Dt}_{bt_{1,n}}^{-\text{At}} \oplus \mathbf{At}_l^{-\text{Bt}} \\ \hline \mathbf{Dt}_{bt_{2,1}}^{-\text{At}} \oplus \mathbf{At}_l^{-\text{Bt}} & \mathbf{At}^{-\text{Bt}} \oplus \mathbf{Dt}_{bt_{2,2}}^{-\text{At}} & \ddots & \dots & \mathbf{Dt}_{bt_{2,n}}^{-\text{At}} \oplus \mathbf{At}_l^{-\text{Bt}} \\ \hline \mathbf{Dt}_{bt_{3,1}}^{-\text{At}} \oplus \mathbf{At}_l^{-\text{Bt}} & \mathbf{Dt}_{bt_{3,2}}^{-\text{At}} \oplus \mathbf{At}_l^{-\text{Bt}} & \ddots & \ddots & \vdots \\ \hline \vdots & \vdots & \ddots & \mathbf{At}^{-\text{Bt}} \oplus \mathbf{Dt}_{bt_{n-1,n-1}}^{-\text{At}} & \mathbf{Dt}_{bt_{n-1,n}}^{-\text{At}} \oplus \mathbf{At}_l^{-\text{Bt}} \\ \hline \mathbf{Dt}_{bt_{n,1}}^{-\text{At}} \oplus \mathbf{At}_l^{-\text{Bt}} & \mathbf{Dt}_{bt_{n,2}}^{-\text{At}} \oplus \mathbf{At}_l^{-\text{Bt}} & \dots & \mathbf{Dt}_{bt_{n,n-1}}^{-\text{At}} \oplus \mathbf{At}_l^{-\text{Bt}} & \mathbf{At}^{-\text{Bt}} \oplus \mathbf{Dt}_{bt_{n,n}}^{-\text{At}} \\ \hline \end{array}$$

em que \mathbf{Ct}^\top tem dimensão $m \times n$, $\mathbf{At}^{-\text{Bt}}$ é a matriz dos elementos de Σ_1 que não pertencem a Σ_2 , $\mathbf{Dt}_{bt_{i,j}}^{-\text{At}}$ é uma matriz diagonal temporizada de dimensão $m \times m$ formada pelos elementos $bt_{i,j}$ que não são definidos em \mathbf{At} na diagonal principal e ϵ nos demais elementos e $\mathbf{At}_l^{-\text{Bt}}$ é a matriz dos auto-laços de $\mathbf{At}^{-\text{Bt}}$.

Deve-se observar que a Definição 4.20 é construída igualmente à Definição 3.18, considerando os tempos de vida dos eventos, como pode-se ver em Brandin e Wonham [BW94].

Também deve-se ver que a definição da matriz $\mathbf{P} = \mathbf{A}||\mathbf{B}$ é geralmente utilizada para construir a composição síncrona do supervisor com o gerador. A matriz \mathbf{Ct}^\neg , quando todos os elementos de Σ_1 são diferentes de Σ_2 , pode ser utilizada para a construção do gerador do sistema, a partir da composição dos subsistemas que o compõem. Assim, com a Definição 4.20, o seguinte algoritmo é apresentado para a construção do produto síncrono de duas matrizes de incidência temporizadas, em que $\Sigma_1 \cap \Sigma_2 \neq \emptyset$:

Algoritmo 4.1 *Construção do produto síncrono $\mathbf{Pt} = \mathbf{At}||\mathbf{Bt}$*

1. Faça $j_{\mathbf{At}} = j_{\mathbf{Bt}} = i_{\mathbf{At}} = i_{\mathbf{Bt}} = 1$.
2. Faça $\theta_1(\mathbf{Pt}) = \theta_1(\mathbf{At}) \oplus \theta_1(\mathbf{Bt})$
3. Enquanto $i_{\mathbf{Bt}} \leq n$, faça:
 - a) Se $bt_{i_{\mathbf{Bt}}, j_{\mathbf{Bt}}} = \epsilon$, faça $j_{\mathbf{Bt}} = j_{\mathbf{Bt}} + 1$
 - b) Se $j_{\mathbf{Bt}} > n$, faça $i_{\mathbf{Bt}} = i_{\mathbf{Bt}} + 1$ e $j_{\mathbf{Bt}} = 0$
 - c) Enquanto $i_{\mathbf{At}} \leq m$, faça:
 - i. Se $at_{i_{\mathbf{At}}, j_{\mathbf{At}}} = \epsilon$, faça $j_{\mathbf{At}} = j_{\mathbf{At}} + 1$
 - ii. Se $j_{\mathbf{At}} > m$, faça $i_{\mathbf{At}} = i_{\mathbf{At}} + 1$ e $j_{\mathbf{At}} = 0$
 - iii. Faça $k = i_{\mathbf{At}} + m(i_{\mathbf{Bt}} - 1)$
 - iv. Faça $l = j_{\mathbf{At}} + m(j_{\mathbf{Bt}} - 1)$
 - v. Faça $pt_{k,l} = at_{i_{\mathbf{At}}, j_{\mathbf{At}}} \otimes bt_{i_{\mathbf{Bt}}, j_{\mathbf{Bt}}}$
 - vi. $\phi_k(\mathbf{Pt}) = \phi_{i_{\mathbf{At}}}(\mathbf{At}) \otimes \phi_{i_{\mathbf{Bt}}}(\mathbf{Bt})$.
 - vii. Faça $j_{\mathbf{At}} = j_{\mathbf{At}} + 1$
 - viii. Se $j_{\mathbf{At}} > m$, faça $i_{\mathbf{At}} = i_{\mathbf{At}} + 1$ e $j_{\mathbf{At}} = 0$
 - d) Faça $j_{\mathbf{Bt}} = j_{\mathbf{Bt}} + 1$
 - e) Se $j_{\mathbf{Bt}} > n$, faça $i_{\mathbf{Bt}} = i_{\mathbf{Bt}} + 1$ e $j_{\mathbf{Bt}} = 0$
4. Se $\exists \sigma_{\mathbf{At}} \notin \Sigma_2$, ou $\exists \sigma_{\mathbf{Bt}} \notin \Sigma_1$, componha a matriz \mathbf{Ct}^\neg , e faça $pt_{k,l} = pt_{k,l} \oplus ct_{k,l}^\neg$.

Observação: Esse algoritmo utiliza um procedimento semelhante ao algoritmo da construção do produto síncrono de autômatos [RW89], e conseqüentemente, ao Algoritmo 3.1. Sua diferença está na introdução dos tempos de vida dos eventos. Contudo, esta diferença não causa aumento na complexidade do algoritmo. Dessa forma, sua ordem de complexidade é $O(nm)$. Considerando que $n = m = N$, sua ordem de complexidade é $O(N^2)$.

Exemplo 4.12 Dadas

$$\mathbf{At} = \begin{bmatrix} \epsilon & 2\alpha \\ 3\beta & \epsilon \end{bmatrix}, \quad \theta(\mathbf{At}) = \begin{bmatrix} \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{At}) = \begin{bmatrix} 3 \\ \epsilon \end{bmatrix}$$

e

$$\mathbf{Bt} = \begin{bmatrix} \epsilon & 2\alpha & \epsilon \\ 2\beta & \epsilon & 3\eta \\ 2\lambda & 4\beta & \epsilon \end{bmatrix}, \quad \theta(\mathbf{Bt}) = \begin{bmatrix} 2 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{Bt}) = \begin{bmatrix} 2 \\ 1 \\ \epsilon \end{bmatrix}$$

que representam os autômatos da Figura 4.10(a) e 4.10(b), seu produto síncrono determina a matriz \mathbf{Pt} , em que seus elementos $pt_{k,l} = pt_{(i_{\mathbf{At}}, i_{\mathbf{Bt}}), (j_{\mathbf{At}}, j_{\mathbf{Bt}})}$; $k = i_{\mathbf{At}} + m(i_{\mathbf{Bt}} - 1)$ e $l = j_{\mathbf{At}} + m(j_{\mathbf{Bt}} - 1)$, nos elementos $a_{i_{\mathbf{At}}, j_{\mathbf{At}}} = b_{i_{\mathbf{Bt}}, j_{\mathbf{Bt}}} \neq \epsilon$, são dados por:

- $k = 1 + 2 \times 0 = 1$, $l = 2 + 2 \times 1 = 4$, $pt_{1,4} = pt_{(1,1), (2,2)} = at_{1,2} \otimes bt_{1,2} = 2\alpha$
- $k = 2 + 2 \times 1 = 4$, $l = 1 + 2 \times 0 = 1$, $pt_{4,1} = pt_{(2,2), (1,1)} = at_{2,1} \otimes bt_{2,1} = 3\beta$
- $k = 2 + 2 \times 2 = 6$, $l = 1 + 2 \times 1 = 3$, $pt_{6,3} = pt_{(2,3), (1,2)} = at_{2,1} \otimes bt_{3,2} = 4\beta$

$$\mathbf{Pt} = \begin{bmatrix} \epsilon & \epsilon & \epsilon & 2\alpha & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ 3\beta & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 4\beta & \epsilon & \epsilon & \epsilon \end{bmatrix}$$

com

$$\theta(\mathbf{Pt}) = \begin{bmatrix} 2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{Pt}) = \begin{bmatrix} 3 \\ \epsilon \\ 3 \\ \epsilon \\ \epsilon \\ \epsilon \end{bmatrix} \quad (4.40)$$

Como $\lambda, \eta \in \mathbf{Bt}$, mas $\lambda, \eta \notin \mathbf{At}$, então

$$\mathbf{Ct}^- = \begin{bmatrix} \overbrace{\mathbf{At}^{-\mathbf{Bt}} \oplus \mathbf{Dt}_{bt_{1,1}}^{-\mathbf{At}}} & \overbrace{\mathbf{Dt}_{bt_{1,2}}^{-\mathbf{At}} \oplus \mathbf{At}_l^{-\mathbf{Bt}}} & \overbrace{\mathbf{Dt}_{bt_{1,3}}^{-\mathbf{At}} \oplus \mathbf{At}_l^{-\mathbf{Bt}}} \\ \begin{bmatrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{bmatrix} & \begin{bmatrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{bmatrix} & \begin{bmatrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{bmatrix} \\ \overbrace{\mathbf{Dt}_{bt_{2,1}}^{-\mathbf{At}} \oplus \mathbf{At}_l^{-\mathbf{Bt}}} & \overbrace{\mathbf{At}^{-\mathbf{Bt}} \oplus \mathbf{Dt}_{bt_{2,2}}^{-\mathbf{At}}} & \overbrace{\mathbf{Dt}_{bt_{2,2}}^{-\mathbf{At}} \oplus \mathbf{At}_l^{-\mathbf{Bt}}} \\ \begin{bmatrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{bmatrix} & \begin{bmatrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{bmatrix} & \begin{bmatrix} 3\eta & \epsilon \\ \epsilon & 3\eta \end{bmatrix} \\ \overbrace{\mathbf{Dt}_{bt_{3,1}}^{-\mathbf{At}} \oplus \mathbf{At}_l^{-\mathbf{Bt}}} & \overbrace{\mathbf{Dt}_{bt_{3,2}}^{-\mathbf{At}} \oplus \mathbf{At}_l^{-\mathbf{Bt}}} & \overbrace{\mathbf{At}^{-\mathbf{Bt}} \oplus \mathbf{Dt}_{bt_{1,1}}^{-\mathbf{At}}} \\ \begin{bmatrix} 2\lambda & \epsilon \\ \epsilon & 2\lambda \end{bmatrix} & \begin{bmatrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{bmatrix} & \begin{bmatrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{bmatrix} \end{bmatrix}$$

e

$$Pt = Pt \oplus Ct^T = \begin{bmatrix} \epsilon & \epsilon & \epsilon & 2\alpha & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 3\eta & \epsilon \\ 3\beta & \epsilon & \epsilon & \epsilon & \epsilon & 3\eta \\ 2\lambda & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 2\lambda & 4\beta & \epsilon & \epsilon & \epsilon \end{bmatrix}$$

com $\theta(Pt)$ e $\phi(Pt)$ dados por 4.40. Assim, Pt em conjunto com $\theta(Pt)$ e $\phi(Pt)$ representam o autômato $A_{(max,+)_3}$ composto por $A_{(max,+)_1}$ e $A_{(max,+)_2}$ através do produto síncrono visto na Figura 4.10(c). Deve-se observar que a linha 2 contém todos os seus

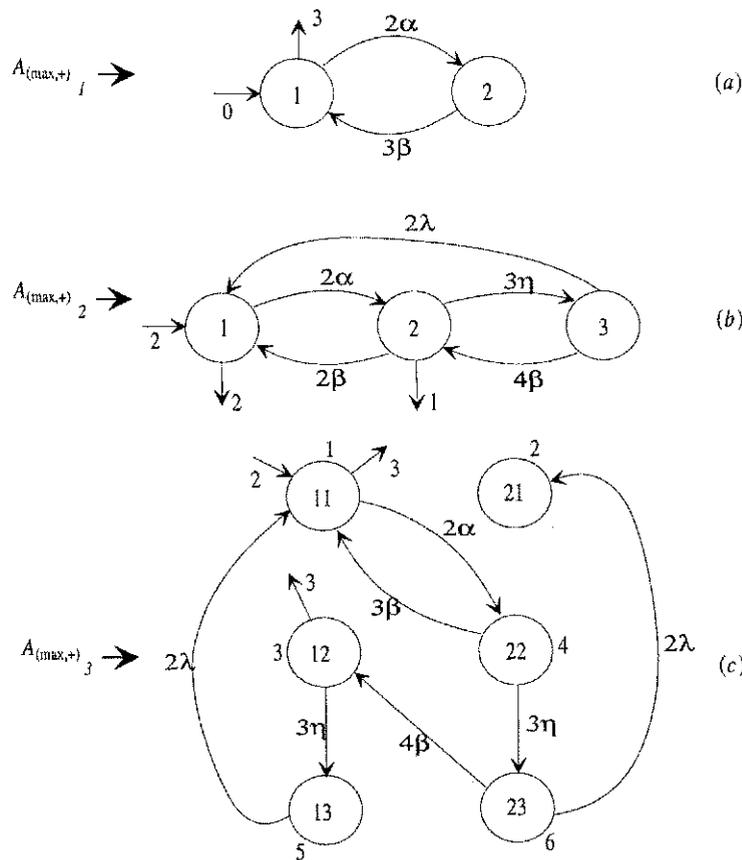


Figure 4.10: Autômatos (a) $A_{(max,+)_1}$, (b) $A_{(max,+)_2}$ e (c) $A_{(max,+)_3}$ - composição síncrona de $A_{(max,+)_1}$ e $A_{(max,+)_2}$.

elementos $pt_{2,j} = \epsilon$, mas esta linha é alcançada pela seqüência $s = \theta(Pt) \otimes Pt^3 \otimes \pi = pt_{1,4}pt_{4,6}pt_{6,2} = 9\alpha\eta\lambda$, para

$$\pi = \left[\epsilon \ e \ \epsilon \ \epsilon \ \epsilon \ \epsilon \right]^T,$$

o que implica que esta linha representa um estado bloqueado na composição de $A_{(max,+)_1}$ com $A_{(max,+)_2}$.

Observação: A Definição 4.20 constrói a composição paralela de duas matrizes de incidência temporizadas \mathbf{At} e \mathbf{Bt} , quando $\Sigma_{\mathbf{At}} \cap \Sigma_{\mathbf{Bt}} = \emptyset$, através da construção da matriz \mathbf{Ct}^\top . Por outro lado, considerando todos os tempos de vida dos eventos $t_\sigma = e$, a Definição 4.20 abrange também a Definição 3.18. Também, deve-se observar que a Definição 4.20 utiliza o operador \oplus de acordo com a Definição 4.18. Adiante, quando for definido a divisão do alfabeto de eventos Σ de um SED em eventos controláveis Σ_c e eventos não controláveis Σ_{uc} , tanto o produto síncrono como quaisquer outras operações que utilizem o operador \oplus , considerá-lo-ão de acordo com a Definição 4.19.

4.3.2 Formulação do problema de controle

Tendo sido formalizada a conceituação das matrizes de incidência temporizadas, necessita-se agora contextualizá-las com a TCS. Primeiramente, deve-se considerar que a inclusão da temporização determina que a diferença da dinâmica do SED se apresenta apenas na inclusão do tempo de vida t_σ de cada evento σ . Assim, o problema de controle para o caso temporizado pode ser enunciado da seguinte maneira: *Dados um autômato temporizado $A_{(\max,+)}$, uma linguagem-alvo temporizada reconhecível E e uma linguagem temporizada mínima admissível $L_A \subseteq E$, encontrar um supervisor próprio S tal que*

$$L_A \subseteq L_c(S/G) \subseteq E. \quad (4.41)$$

Com o problema de controle enunciado e considerando a partição do alfabeto Σ , como citado anteriormente, e formalizando este problema em termos das matrizes de incidência, definem-se as matrizes de incidência temporizadas dos eventos controláveis e dos eventos não controláveis.

Definição 4.21 *Dado um alfabeto de eventos Σ , e considerando $\Sigma = \Sigma_c \cup \Sigma_{uc}$, definem-se as matrizes de incidência temporizadas dos eventos controláveis \mathbf{At}_c e dos eventos não controláveis \mathbf{At}_{uc} por*

$$\mathbf{At}_c = \left[(at_c)_{i,j} \right]; (at_c)_{i,j} = \begin{cases} t_\sigma \sigma_c & \text{se } \exists \sigma_c \text{ do estado } i \text{ para o estado } j; \\ \epsilon & \text{caso contrário,} \end{cases}$$

e

$$\mathbf{At}_{uc} = \left[(at_{uc})_{i,j} \right]; (at_{uc})_{i,j} = \begin{cases} t_\sigma \sigma_{uc} & \text{se } \exists \sigma_{uc} \text{ do estado } i \text{ para o estado } j; \\ \epsilon & \text{caso contrário,} \end{cases}$$

em que t_σ é o tempo de vida do evento que leva o autômato $A_{(\max,+)}$ do estado i para o estado j . Se mais de um evento é definido do estado i para o estado j , $(at_c)_{i,j} = \bigoplus_k t_{\sigma^k} \sigma^k$ e $(at_{uc})_{i,j} = \bigoplus_l t_{\sigma^l} \sigma^l$. Os vetores de estado inicial e de estados marcados são os mesmos da matriz \mathbf{At} .

Dessa definição, observa-se como no caso da matriz de incidência não temporizada, que

$$\mathbf{At} = \mathbf{At}_c \oplus \mathbf{At}_{uc},$$

e que sendo $\theta(\mathbf{At}_c) = \theta(\mathbf{At}_{uc})$, então $\theta(\mathbf{At}_c) \oplus \theta(\mathbf{At}_{uc}) = \theta(\mathbf{At})$ e, $\phi(\mathbf{At}_c) = \phi(\mathbf{At}_{uc})$, então $\phi(\mathbf{At}_c) \oplus \phi(\mathbf{At}_{uc}) = \phi(\mathbf{At})$.

Exemplo 4.13 Considerando que o autômato $(\max, +)$ da Figura 4.11 tem $\Sigma_c = \{\alpha, \beta\}$ e $\Sigma_{uc} = \{\eta, \kappa\}$, então

$$\mathbf{At}_c = \begin{bmatrix} \epsilon & 2\alpha & \epsilon \\ \epsilon & \epsilon & \epsilon \\ \epsilon & 4\beta & \epsilon \end{bmatrix}$$

e

$$\mathbf{At}_{uc} = \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ 2\kappa & \epsilon & 3\eta \\ 2\kappa & \epsilon & \epsilon \end{bmatrix}.$$

Os vetores de estado inicial e estados marcados são

$$\theta(\mathbf{At}_c) = \theta(\mathbf{At}_{uc}) = \begin{bmatrix} 2 & \epsilon & \epsilon \end{bmatrix} \quad e \quad \phi(\mathbf{At}_c) = \phi(\mathbf{At}_{uc}) = \begin{bmatrix} 2 \\ 1 \\ \epsilon \end{bmatrix}.$$

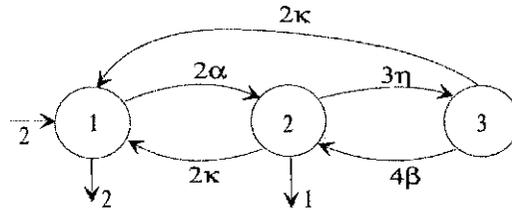


Figure 4.11: Autômatos $A_{(\max,+)}$, para exemplificar a construção das matrizes \mathbf{At}_c e \mathbf{At}_{uc} .

Através desta partição da matriz de incidência temporizada, o procedimento para a síntese do supervisor é semelhante ao procedimento apresentado no Capítulo 3, para matrizes de incidência não temporizadas. Assim, necessita-se definir a especificação de comportamento e o supervisor para os SEDs temporizados. Neste caso, a especificação de comportamento é uma tarefa que se deseja que o sistema realize, a qual inclui tanto o comportamento lógico (similar ao caso não temporizado), quanto os tempos mínimos para os inícios das atividades (que devem sempre ser iguais ou maiores que os do sistema, definindo atrasos), determinando o comportamento lógico e temporizado que o sistema deve apresentar sob supervisão.

Definição 4.22 Uma especificação de comportamento temporizada para um autômato temporizado $A_{(\max,+)}$ que é um modelo de um SED temporizado, denotada por \mathbf{Et} , é definida por

$$\mathbf{Et} = [et_{i,j}]; et_{i,j} = \begin{cases} t_\sigma \sigma & \text{se } \exists \sigma \text{ do estado } i \text{ para o estado } j; \\ \epsilon & \text{caso contrário,} \end{cases}$$

em que t_σ é o tempo de vida do evento σ que leva o autômato da especificação do estado i para o estado j . Se mais de um evento é definido do estado i para o estado j , $et_{i,j} = \bigoplus_l t_{\sigma^l} \sigma^l$, para $\sigma^l \in \Sigma$, $l = 1, 2, \dots$, de modo que qualquer evento σ^l ocorrendo, provoca no autômato $A_{(\max,+)}$ a mudança do estado i para o estado j . O estado inicial é definido como sendo o estado 1 e é representado pelo vetor linha

$$\theta(\mathbf{Et}) = \left[t_i \quad \epsilon \quad \dots \quad \epsilon \right]$$

com t_i representando o atraso inicial, e os estados marcados são representados pelo vetor coluna

$$\phi(\mathbf{Et}) = \left[t_{m_1} \quad t_{m_2} \quad \dots \quad t_{m_n} \right]^T,$$

com t_{m_i} representando os atrasos finais. Se um estado k não é marcado, $t_{m_k} = \epsilon$.

O supervisor é definido como a seguir:

Definição 4.23 Um supervisor construído através de uma especificação de comportamento \mathbf{Et} para um SED temporizado modelado por uma matriz de incidência temporizada \mathbf{At} é definido como sendo a matriz

$$\mathbf{St} = [st_{i,j}], st_{i,j} = \begin{cases} t_\sigma \sigma \subset et_{i,j} & \text{se } t_\sigma \sigma \subset et_{i,j} \text{ pode ocorrer em } \mathbf{At}; \\ \epsilon & \text{caso contrário} \end{cases}$$

em que $st_{i,j} = \epsilon$ para $at_{i,j} \neq \epsilon$, implica em dizer que existe um controle para inibir o evento que se encontra em $at_{i,j}$, para $at_{i,j} = \sigma_c$, ou que o estado (linha) j não é acessível. Se mais de um evento é definido do estado i para o estado j , $st_{i,j} = \bigoplus_l t_{\sigma^l} \sigma^l$, para $\sigma^l \in \Sigma$, $l = 1, 2, \dots$, de modo que qualquer evento σ^l ocorrendo, provoca no autômato da composição síncrona $\mathbf{St} \parallel \mathbf{At}$ a mudança do estado i para o estado j . O estado inicial é definido como sendo o estado 1 e é representado pelo vetor linha

$$\theta(\mathbf{St}) = \left[t_i \quad \epsilon \quad \dots \quad \epsilon \right]$$

com t_i representando o atraso inicial, e os estados marcados são representados pelo vetor coluna

$$\phi(\mathbf{St}) = \left[t_{m_1} \quad t_{m_2} \quad \dots \quad t_{m_n} \right]^T,$$

com t_{m_i} representando os atrasos finais. Se um estado k não é marcado, $t_{m_k} = \epsilon$.

Na abordagem não temporizada, o supervisor é o agente que recebe os eventos σ gerados pelo SED, e determina a ação de controle γ a ser aplicada de forma ao SED realizar a tarefa desejada, definida pela especificação de comportamento.

No caso dessa abordagem temporizada, o supervisor é o agente que recebe os eventos σ gerados pelo SED e seus respectivos tempos de vida, e determina a ação de controle

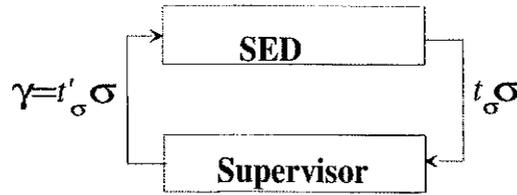


Figure 4.12: Diagrama de um SED temporizado supervisionado.

definida como uma inibição total ou temporária nos eventos controláveis. Este formalismo é apresentado na Figura 4.12.

Também para o supervisor temporizado, define-se o conjunto de entradas de controle semelhantemente à Definição 3.22, isto é,

Definição 4.24 O conjunto de entradas de controle do supervisor temporizado é definido como

$$\Gamma = \{\gamma_1, \gamma_2, \dots\},$$

onde

$$\gamma_i = (t'_{\sigma_k} \sigma_k), \forall \sigma_k \in \Sigma(i), k = 1, 2, \dots$$

com

$$(t'_{\sigma_k} = t_{\sigma_k}), \forall \sigma_k \in \Sigma_{uc}$$

e

$$(t'_{\sigma_k} \geq t_{\sigma_k}), \forall \sigma_k \in \Sigma_c$$

em que $(t'_{\sigma_k} \geq t_{\sigma_k})$ significa que o evento σ_k é inibido por um tempo $t = t'_{\sigma_k} - t_{\sigma_k}$.

De acordo com a Definição 4.24, as entradas de controle do supervisor temporizado são

$$\gamma = t'_\sigma \sigma$$

que determinam quais eventos estão habilitados em um determinado estado, e seu tempo de vida correspondente $(t'_\sigma \geq t_\sigma)$. Dessa forma, o supervisor inibe temporariamente o evento σ , por um tempo t definido por $t = t'_\sigma - t_\sigma$ (em notação da álgebra de dióides: $t = t'_\sigma \oslash t_\sigma$, em que \oslash é o operador dual de \otimes). Se a inibição é total (o evento deve ser permanentemente inibido) $t'_\sigma = \epsilon$, como visto anteriormente, na explicação da Figura 4.12. Dessa maneira, a evolução do sistema temporizado supervisionado é definida considerando a construção da linguagem reconhecida por meio das séries formais ou pela Definição 4.15.

Com essas definições, a contextualização com a TCS é determinada nessa abordagem como um problema de estados proibidos restrito ao controle de SEDs temporizados, onde os tempos de vida dos eventos são definidos como o tempo mínimo para suas ocorrências. Assim, o mesmo procedimento apresentado na abordagem não temporizada é utilizado, onde devem ser consideradas as seguintes denominações:

1. Um sub-autômato $A'_{(\max,+)}$ apresenta uma estrutura semelhante, mas apenas com parte dos estados ou com parte das transições de $A_{(\max,+)}$, em que o tempo de vida de cada transição de $A'_{(\max,+)}$ deve ser maior ou igual ao tempo de vida da respectiva transição em $A_{(\max,+)}$;
2. Uma sub-linguagem temporizada L' apresenta parte das seqüências da linguagem L , em que cada evento de L' apresenta um tempo de vida maior ou igual ao respectivo tempo de vida do mesmo evento em L . Não necessariamente $L' \subset L$ indica que o autômato de L' é um sub-autômato de L ;
3. Uma submatriz \mathbf{At}' é formada pelas primeiras m linhas e as primeiras m colunas de \mathbf{At} e mantém uma correlação elemento a elemento, apresentando para cada evento de \mathbf{At}' um tempo de vida maior ou igual ao respectivo evento de \mathbf{At} .

Dessa forma, para realizar a síntese de supervisores de SEDs temporizados utilizando as matrizes de incidência temporizadas, é necessário que a especificação de comportamento sempre seja uma submatriz da matriz de incidência \mathbf{At} . Se \mathbf{Et} não é uma submatriz de \mathbf{At} , torna-se necessário transformar \mathbf{At} em $\mathbf{At}^\#$, e construir uma nova matriz $\mathbf{Et}^\#$ que se apresente como uma submatriz de $\mathbf{At}^\#$, tal que

$$\begin{aligned} L(\mathbf{At}^\#) &= L(\mathbf{At}), \\ L_m(\mathbf{At}^\#) &= L_m(\mathbf{At}), \end{aligned}$$

e

$$\begin{aligned} L(\mathbf{Et}^\#) &= L(\mathbf{Et}), \\ L_m(\mathbf{Et}^\#) &= L_m(\mathbf{Et}). \end{aligned}$$

Assim, considerando que a especificação de comportamento \mathbf{Et} seja uma submatriz de \mathbf{At} , ou seja transformada em $\mathbf{Et}^\#$, sempre a relação

$$L(\mathbf{Et}) \subseteq L(\mathbf{At})$$

é verificada. Esses dois casos representam uma especificação de comportamento genérica.

Quando uma especificação de comportamento temporizada \mathbf{Et} não é submatriz de \mathbf{At} , necessita-se do seguinte procedimento:

1. Constrói-se a composição síncrona de \mathbf{At} com uma matriz qualquer que gera Σ^* , para determinar uma nova matriz $\mathbf{At}^\#$ que gera $L(\mathbf{At}^\#) = L(\mathbf{At})$. Transformando a especificação de comportamento \mathbf{Et} em uma matriz de incidência \mathbf{Et}^* que gera Σ^* , através da inclusão de um estado proibido (linha i_e /coluna j_e), denominado de estado de *erro*, utiliza-se \mathbf{Et}^* para construir a matriz $\mathbf{At}^\# = \mathbf{At} \parallel \mathbf{Et}^*$;
2. Da matriz $\mathbf{At}^\#$, substituindo todos os elementos das linhas/colunas que formam os pares $(i_{\mathbf{At}}, i_e)$, $(j_{\mathbf{At}}, j_e)$ pelo símbolo ϵ , encontra-se a matriz $\mathbf{Et}^\#$ tal que $L(\mathbf{Et}^\#) \subseteq L(\mathbf{At}^\#) = L(\mathbf{At})$ e $\mathbf{Et}^\#$ sendo uma submatriz de $\mathbf{At}^\#$;

3. Os tempos de vida dos eventos de \mathbf{Et} , são resgatados para $\mathbf{Et}^\#$ (que estão iguais aos tempos de vida dos eventos de \mathbf{At}) por $t_\sigma \oplus t'_\sigma$, onde t_σ é o tempo de vida do evento $\sigma \subset et_{i,j}$ e t'_σ é o tempo de vida do evento $et_{k,l}^\#$, com k sendo o par $(i_{\mathbf{At}}, i_{\mathbf{Et}})$ em que $i_{\mathbf{Et}} = i$ e l sendo o par $(j_{\mathbf{At}}, j_{\mathbf{Et}})$ em que $j_{\mathbf{Et}} = j$;
4. O vetor $\theta(\mathbf{Et}^\#)$ é construído através do resgate dos elementos do vetor de estado inicial $\theta(\mathbf{Et})$, como $\theta_1(\mathbf{Et}^\#) = \theta_1(\mathbf{Et}^\#) \oplus \theta_1(\mathbf{Et})$, tal que $L(\mathbf{Et}^\#) \subset L(\mathbf{At}^\#)$ e $L(\mathbf{Et}) = L(\mathbf{Et}^\#)$;
5. O vetor $\phi(\mathbf{Et}^\#)$ é determinado pelos elementos resgatados do vetor original $\phi(\mathbf{Et})$, como $\phi_k(\mathbf{Et}^\#) = \phi_k(\mathbf{Et}^\#) \oplus \phi_i(\mathbf{Et})$ em que k é definido como o par $(i_{\mathbf{At}}, i_{\mathbf{Et}})$ e $i = i_{\mathbf{Et}}$, com $\phi_i(\mathbf{Et}) \neq \epsilon$, tal que $L_m(\mathbf{Et}^\#) = L_m(\mathbf{Et})$.

Com esse procedimento, a matriz $\mathbf{Et}^\#$ se apresenta como uma submatriz de $\mathbf{At}^\#$, e conseqüentemente,

$$L(\mathbf{Et}^\#) \subset L(\mathbf{At}^\#) = L(\mathbf{At}),$$

e

$$\begin{aligned} L(\mathbf{Et}^\#) &= L(\mathbf{Et}) \\ L_m(\mathbf{Et}^\#) &= L_m(\mathbf{Et}). \end{aligned}$$

Observação: Deve-se observar que no contexto das matrizes de incidência temporizadas, a matriz que gera Σ^* deve ser considerada com todos os tempos de vida dos eventos $t_\sigma = e$. Isto porque a composição síncrona $\mathbf{At} \parallel \Sigma^*$ deve ser uma matriz $\mathbf{At}^\#$ que deve apresentar a mesma linguagem da matriz \mathbf{At} . Pela própria definição da composição síncrona, os tempos de vida da matriz resultante da composição é $t_\sigma \oplus t'_\sigma$, quando $\sigma \in \Sigma_c$. Assim, considerando $\sigma \subset at_{i,j}$, para $t'_\sigma = e$, a matriz resultante $\mathbf{At} \parallel \Sigma^*$ tem sempre os tempos de vida satisfazendo $t_\sigma \oplus e = t_\sigma$, e dessa forma, garante-se que $L(\mathbf{At}^\#) = L(\mathbf{At})$.

Para os casos em que há a necessidade da transformação de \mathbf{Et} em $\mathbf{Et}^\#$, é apresentado a seguir o algoritmo que transforma uma matriz de incidência temporizada qualquer para que sua linguagem reconhecida seja Σ^* .

Algoritmo 4.2 Transformação de matrizes de incidência temporizadas \mathbf{Ft} para reconhecimento de Σ^*

1. Tornar os elementos $\theta_1(\mathbf{Ft}) = e$, $\phi_i(\mathbf{Ft}) = e$, sendo o índice o elemento da coluna/linha do vetor respectivo, e $\forall ft_{i,j} = t_{\sigma^1} \sigma^1 + \dots + t_{\sigma^n} \sigma^n$, $t_{\sigma^k} = e$, $k = 1, \dots, n$;
2. Para cada linha i de \mathbf{Ft} , incluir os auto-laços

$$ft_{i,i} = \Sigma - \{\sigma^1, \sigma^2, \dots, \sigma^n\}$$

tal que $\sigma^k \notin ft_{i,j}$, $k = 1, 2, \dots, n$;

3. Incluir nos vetores $\theta(\mathbf{Ft})$ e $\phi(\mathbf{Ft})$, o elemento $\theta_{j+1}(\mathbf{Ft}) = \epsilon$ e $\phi_{j+1}(\mathbf{Ft}) = e$.
4. Incluir no final da matriz \mathbf{Ft} uma coluna j_e e uma linha marcada de 'erro' i_e representando uma linha proibida, e fazer:
 - i) Para $i < i_e$, faça:
 - a) $\forall j, j < j_e, ft_{i,j_e} = \sigma$, se $\sigma \subset ft_{j,i} \vee \sigma \subset ft_{j,j}$ e $ft_{i,k} \neq \sigma, \forall k, k < j_e$;
 - b) $\forall j, j < j_e, ft_{i,j_e} = \sigma'$ se $ft_{i,j} \neq \sigma'$
 - ii) Faça $ft_{i_e,j_e} = \Sigma$.

Teorema 4.1 Uma matriz de incidência temporizada \mathbf{Ft}^* construída de \mathbf{Ft} , com $L(\mathbf{Ft}) \subset \Sigma^*$, gera a linguagem $L(\mathbf{Ft}^*) = \Sigma^*$.

Demonstração:

Este Teorema é demonstrado similarmente ao Teorema 3.1 utilizando as matrizes de incidência temporizadas. \blacklozenge

Exemplo 4.14 Dada a matriz de incidência temporizada sobre o alfabeto $\Sigma = \{\alpha, \beta, \eta, \lambda, \kappa, \mu\}$

$$\mathbf{At} = \begin{bmatrix} \epsilon & 2\alpha + 3\beta \\ \kappa & 5\eta + \lambda \end{bmatrix}, \quad \theta(\mathbf{At}) = \begin{bmatrix} 2 & \epsilon \end{bmatrix}, \quad \phi(\mathbf{At}) = \begin{bmatrix} \epsilon \\ 3 \end{bmatrix}$$

a utilização do algoritmo 4.2 constrói a matriz

$$\mathbf{At}^* = \begin{bmatrix} \mu & \alpha + \beta & \kappa + \eta + \lambda \\ \kappa & \mu + \eta + \lambda & \alpha + \beta \\ \epsilon & \epsilon & \alpha + \beta + \kappa + \mu + \eta + \lambda \end{bmatrix}, \quad \theta(\mathbf{At}^*) = \begin{bmatrix} e & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{At}^*) = \begin{bmatrix} e \\ e \\ e \end{bmatrix}$$

que reconhece Σ^* . Os autômatos que elas representam são vistos na Figura 4.13(a), 4.13(b), respectivamente.

Com a transformação de uma matriz de incidência \mathbf{Et} qualquer em uma matriz que gera Σ^* , a construção de $\mathbf{At}^\#$ e $\mathbf{Et}^\#$, que satisfazem as condições $L(\mathbf{Et}^\#) \subset L(\mathbf{At}^\#)$, $L(\mathbf{At}^\#) = L(\mathbf{At})$, $L(\mathbf{Et}^\#) = L(\mathbf{Et})$ e $L_m(\mathbf{Et}^\#) = L_m(\mathbf{Et})$ é feita utilizando o seguinte algoritmo, em que se consideram as dimensões de \mathbf{Et} e \mathbf{At} iguais a n e m , respectivamente.

Algoritmo 4.3 Transformação da matriz de incidência \mathbf{Et} em $\mathbf{Et}^\#$

1. Transforme \mathbf{Et} em \mathbf{Et}^* ;
2. Construa $\mathbf{At}^\# = \mathbf{At} \parallel \mathbf{Et}^*$;
3. Para i e $j = 1$ até $n \times m$, faça:

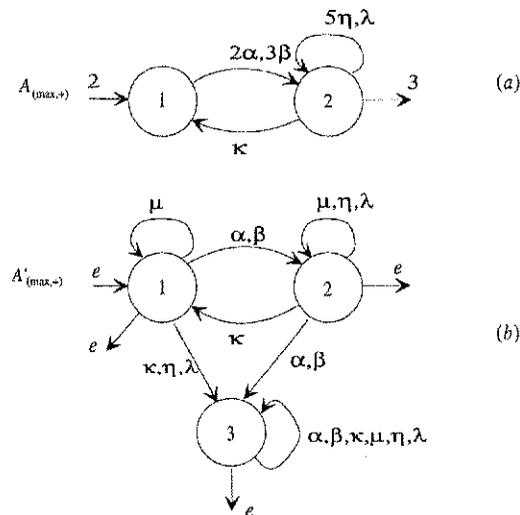


Figure 4.13: Autômatos (a) $A_{(max,+)}$ e (b) $A'_{(max,+)}$ construídos com o Algoritmo 4.2.

a) Se $at_{i,j}^{\#} \neq \epsilon$, faça $et_{i,j}^{\#} = at_{i,j}^{\#}$.

4. Faça $k = 1$ até $n \times m$:

a) Se $\phi_k(\mathbf{At}^{\#}) \neq \epsilon$, faça $i_{\mathbf{Et}} = ((k - (k \bmod m)) / m) + 1$, e $\phi_k(\mathbf{Et}^{\#}) = \phi_k(\mathbf{Et}^{\#}) \oplus \phi_{i_{\mathbf{E}}}(\mathbf{Et})$;

5. Faça $\theta_1(\mathbf{Et}^{\#}) = \theta_1(\mathbf{Et}^{\#}) \oplus \theta_1(\mathbf{Et})$.

Neste algoritmo, mod é o operador de módulo que retorna o resto da divisão k/m . O **passo 4** resgata os atrasos finais dos estados marcados e o **passo 5** resgata o atraso inicial da especificação de comportamento \mathbf{Et} , para $\mathbf{Et}^{\#}$, de forma a garantir que $L(\mathbf{Et}^{\#}) = L(\mathbf{Et})$ e $L_m(\mathbf{Et}^{\#}) = L_m(\mathbf{Et})$.

Teorema 4.2 Dadas as matrizes de incidência \mathbf{At} e \mathbf{Et} , a transformação de \mathbf{At} em $\mathbf{At}^{\#}$ e \mathbf{Et} em $\mathbf{Et}^{\#}$ determina que $\mathbf{Et}^{\#}$ é submatriz de $\mathbf{At}^{\#}$, e $L(\mathbf{Et}^{\#}) \subset L(\mathbf{At}^{\#})$, $L(\mathbf{Et}^{\#}) = L(\mathbf{Et})$ e $L_m(\mathbf{Et}^{\#}) = L_m(\mathbf{Et})$.

Demonstração:

Este Teorema é demonstrado igualmente ao Teorema 3.2, utilizando as matrizes de incidência temporizadas. ♦

Observação: Deve-se observar que o resgate dos tempos de vida dos eventos e dos atrasos inicial e finais eliminam a possibilidade de se introduzir uma especificação que defina uma redução no tempo de vida de um evento do sistema.

O seguinte exemplo ilustra a transformação de \mathbf{At} em $\mathbf{At}^{\#}$ e \mathbf{Et} em $\mathbf{Et}^{\#}$, tal que $L(\mathbf{Et}^{\#}) \subset L(\mathbf{At}^{\#})$, $L(\mathbf{Et}^{\#}) = L(\mathbf{Et})$ e $L_m(\mathbf{Et}^{\#}) = L_m(\mathbf{Et})$, quando \mathbf{Et} não é submatriz de \mathbf{At} .

Exemplo 4.15 Considere a matriz

$$\mathbf{At} = \begin{bmatrix} \epsilon & 2\alpha_1 & 3\alpha_2 & \epsilon \\ 2\beta_1 & \epsilon & \epsilon & 3\alpha_2 \\ 4\beta_2 & \epsilon & \epsilon & 2\alpha_1 \\ \epsilon & 4\beta_2 & 2\beta_1 & \epsilon \end{bmatrix}, \quad \theta(\mathbf{At}) = [3 \ \epsilon \ \epsilon \ \epsilon], \quad \phi(\mathbf{At}) = \begin{bmatrix} e \\ \epsilon \\ \epsilon \\ 3 \end{bmatrix}$$

e a especificação

$$\mathbf{Et} = \begin{bmatrix} \alpha_1 + \beta_2 & 5\beta_1 \\ 4\alpha_2 & \alpha_1 + \beta_2 \end{bmatrix}, \quad \theta(\mathbf{Et}) = [4 \ \epsilon], \quad \phi(\mathbf{Et}) = \begin{bmatrix} 2 \\ \epsilon \end{bmatrix}$$

onde $\Sigma_{uc} = \{\beta_1, \beta_2\}$. Para construir a matriz $\mathbf{Et}^\#$, tal que $L(\mathbf{Et}^\#) \subset L(\mathbf{At}^\#) = L(\mathbf{At})$, constrói-se a matriz \mathbf{Et}^* que gera Σ^* , a qual é dada por

$$\mathbf{Et}^* = \begin{bmatrix} \alpha_1 + \beta_2 & \beta_1 & \alpha_2 \\ \alpha_2 & \alpha_1 + \beta_2 & \beta_1 \\ \epsilon & \epsilon & \alpha_1 + \beta_1 + \alpha_2 + \beta_2 \end{bmatrix}, \quad \theta(\mathbf{Et}^*) = [e \ \epsilon \ \epsilon], \quad \phi(\mathbf{Et}^*) = \begin{bmatrix} e \\ e \\ e \end{bmatrix}$$

onde a linha/coluna de erro é a terceira. Construindo o produto síncrono $\mathbf{At} \parallel \mathbf{Et}^*$, tem-se

$$\mathbf{At}^\# = \begin{bmatrix} \epsilon & 2\alpha_1 & \epsilon & 3\alpha_2 & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 3\alpha_2 \\ 4\beta_2 & \epsilon & \epsilon & 2\alpha_1 & \epsilon \\ \epsilon & 4\beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 3\alpha_2 & \epsilon & \epsilon & 2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 3\alpha_2 & \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & \epsilon & \epsilon & 2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 & \epsilon \\ \epsilon & 2\alpha_1 & 3\alpha_2 & \epsilon \\ \epsilon & 2\beta_1 & \epsilon & \epsilon & 3\alpha_2 \\ \epsilon & 4\beta_2 & \epsilon & \epsilon & 2\alpha_1 \\ \epsilon & 4\beta_2 & 2\beta_1 & \epsilon \end{bmatrix}$$

$$\theta(\mathbf{At}^\#) = [3 \ \epsilon \ \epsilon],$$

$$\phi(\mathbf{At}^\#) = [e \ \epsilon \ \epsilon \ 3 \ e \ \epsilon \ \epsilon \ 3 \ e \ \epsilon \ \epsilon \ 3]^T$$

em que as linhas/colunas formadas por uma linha/coluna de \mathbf{At} com a linha/coluna de erro de \mathbf{Et}^* são as quatro últimas. Fazendo $et_{i,j}^\# = at_{i,j}^\#$, nos elementos $\forall i, j$, tal que $i, j \leq ((n-1) \times m)$ (igual a fazer $\mathbf{Et}^\# = \mathbf{At}^\#$ e tornar todos os elementos que satisfazem os pares

$(i_{At}, i_e), (j_{At}, j_e)$ iguais a ϵ) e resgatando os tempos de vida dos eventos de \mathbf{Et} , encontra-se

$$\mathbf{Et}^\# = \begin{bmatrix} \epsilon & 2\alpha_1 & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 & \epsilon \\ 4\beta_2 & \epsilon & \epsilon & 2\alpha_1 & \epsilon \\ \epsilon & 4\beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 4\alpha_2 & \epsilon & \epsilon & 2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 4\alpha_2 & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & \epsilon & \epsilon & 2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \epsilon \\ \epsilon & \epsilon \end{bmatrix}$$

$$\theta(\mathbf{Et}^\#) = \begin{bmatrix} 4 & \epsilon \end{bmatrix},$$

$$\phi(\mathbf{Et}^\#) = \begin{bmatrix} 2 & \epsilon \end{bmatrix}^T$$

Observe que apenas $\phi_1(\mathbf{Et}^\#) \neq \epsilon$, desde que satisfaz $\phi_1(\mathbf{Et}^\#) = \phi_1(\mathbf{At}) \oplus \phi_1(\mathbf{Et})$. A matriz $\mathbf{Et}^\#$ é uma submatriz da matriz $\mathbf{At}^\#$, com linguagem $L(\mathbf{Et}^\#) = L(\mathbf{Et})$ e linguagem marcada $L_m(\mathbf{Et}^\#) = L_m(\mathbf{Et})$, tal que $L(\mathbf{Et}^\#) \subset L(\mathbf{At}^\#)$.

Observação: A partir daqui, sempre que se citar a especificação de comportamento \mathbf{Et} estará sendo considerada a sua transformação em $\mathbf{Et}^\#$, para os casos em que \mathbf{Et} não é uma submatriz de \mathbf{At} . Da mesma forma, \mathbf{At} implica em $\mathbf{At}^\#$.

Semelhantemente à Definição 3.26, para sintetizar um supervisor \mathbf{St} através de uma especificação de comportamento \mathbf{Et} , para um SED modelado por uma matriz de incidência temporizada \mathbf{At} , define-se:

Definição 4.25 Dadas duas matrizes temporizadas $\mathbf{At} = [at_{i,j}]$ e $\mathbf{Bt} = [bt_{i,j}]$, tal que \mathbf{At} e \mathbf{Bt} são relacionadas como matriz e submatriz, define-se o operador \trianglelefteq por

$$\mathbf{At} \trianglelefteq \mathbf{Bt} \Leftrightarrow L(\mathbf{At}) \subseteq L(\mathbf{Bt}).$$

Igualmente, definem-se os operadores \trianglerighteq , \triangleleft e \triangleright como

$$\mathbf{At} \trianglerighteq \mathbf{Bt} \Leftrightarrow L(\mathbf{At}) \supseteq L(\mathbf{Bt}),$$

$$\mathbf{At} \triangleleft \mathbf{Bt} \Leftrightarrow L(\mathbf{At}) \subset L(\mathbf{Bt}),$$

$$\mathbf{At} \triangleright \mathbf{Bt} \Leftrightarrow L(\mathbf{At}) \supset L(\mathbf{Bt}).$$

Assim, com a Definição 4.25, considerando um elemento genérico $\sigma = \sigma^1 + \dots + \sigma^n$, para determinar se $\mathbf{At} \trianglelefteq \mathbf{Bt}$ utiliza-se o seguinte algoritmo:

Algoritmo 4.4 Verificação de $\mathbf{At} \trianglelefteq \mathbf{Bt}$

1. Para cada elemento $at_{i,j}$ faça:
 - i. Se $\sigma \notin \mathbf{Bt}$, e $at_{i,j} = \sigma$, então $\mathbf{At} \not\trianglelefteq \mathbf{Bt}$;
 - ii. Se $(at_{i,j} = t'_{\sigma^h} \sigma^h + \dots + t'_{\sigma^k} \sigma^k, t'_{\sigma^i} \geq t_{\sigma^i}) \vee (at_{i,j} = \epsilon)$, para $\sigma^i, \dots, \sigma^k \subset \sigma$ e $bt_{i,j} = t_{\sigma} \sigma$, então $\mathbf{At} \trianglelefteq \mathbf{Bt}$;
2. Se $\theta_1(\mathbf{At}) \geq \theta_1(\mathbf{Bt})$, então $\mathbf{At} \trianglelefteq \mathbf{Bt}$.
3. Faça $\forall i$, se $\phi_i(\mathbf{At}) \geq \phi_i(\mathbf{Bt}) \vee \phi_i(\mathbf{At}) = \epsilon$, para $\phi_i(\mathbf{Bt}) \neq \epsilon$, e $\phi_i(\mathbf{At}) = \epsilon, \forall \phi_i(\mathbf{Bt}) = \epsilon$, então $\mathbf{At} \trianglelefteq \mathbf{Bt}$.

Observação: Esse algoritmo apresenta uma execução semelhante ao Algoritmo 3.6. ◆

A condição de $\mathbf{At} \trianglelefteq \mathbf{Bt}$, implica em que toda seqüência de \mathbf{At} , que é formada por eventos com seus respectivos tempos de vida (a soma desses tempos define o tempo da seqüência) deve ser executada em um tempo maior ou igual a execução da mesma seqüência em \mathbf{Bt} . Em termos de SEDs, isto implica em dizer que a linguagem que contém uma outra, é executada em um tempo menor [CG94, CG96, AD90, AD94, Alu97, ATP00]. Então, $L(\mathbf{At}) \subseteq L(\mathbf{Bt})$ significa que todas as seqüências da linguagem de \mathbf{At} são encontradas na linguagem de \mathbf{Bt} , e estas seqüências satisfazem

$$(y|s_{\mathbf{At}}) \geq (y|s_{\mathbf{Bt}}),$$

onde cada evento tem um tempo de vida satisfazendo

$$t_{\sigma^i_{\mathbf{At}}} \geq t_{\sigma^i_{\mathbf{Bt}}}.$$

Isto garante o reconhecimento das mesmas seqüências em duas matrizes de incidência temporizadas, de acordo com as Definições 4.2, 4.7, 4.4 e 4.15.

Exemplo 4.16 Considere o autômato mostrado na Figura 4.14(a), que tem sua representação matricial dada por

$$\mathbf{At}_1 = \begin{bmatrix} 2\beta & 6\alpha & \epsilon \\ 4\beta & 2\alpha & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \theta(\mathbf{At}_1) = \begin{bmatrix} 4 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{At}_1) = \begin{bmatrix} 5 \\ \epsilon \\ \epsilon \end{bmatrix}.$$

Para o autômato mostrado na Figura 4.14(b), que tem sua representação matricial dada por

$$\mathbf{At}_2 = \begin{bmatrix} 2\beta & 3\alpha & \epsilon \\ 2\beta & 2\alpha & 5\kappa \\ \epsilon & 4\beta & \epsilon \end{bmatrix}, \quad \theta(\mathbf{At}_2) = \begin{bmatrix} 2 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{At}_2) = \begin{bmatrix} 5 \\ \epsilon \\ e \end{bmatrix},$$

segundo o Algoritmo 4.4, verifica-se que $\mathbf{At}_1 \triangleleft \mathbf{At}_2$.

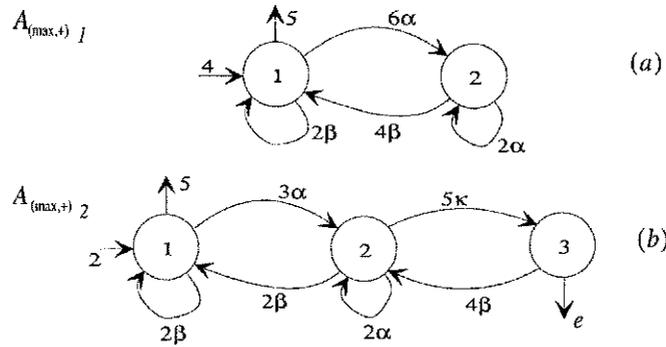


Figure 4.14: Autômatos $(\max, +)$ para ilustrar a utilização do operador \triangleleft .

Com relação aos operadores *ACES*, *COACES* e *TRIM*, as Definições 3.23, 3.24 e 3.25, satisfazem também as matrizes de incidência temporizadas. Assim, com as Definições 3.23, 3.24, 3.25, 4.19 e 4.25, define-se que uma especificação de comportamento deve satisfazer a seguinte condição:

Definição 4.26 *Uma especificação de comportamento \mathbf{Et} é válida para a matriz de incidência \mathbf{At} se $\mathbf{Et} \neq [\epsilon]$ e*

$$\begin{aligned} & \forall et_{i,j}, t_\sigma \geq t'_\sigma, \text{ com } t'_\sigma \text{ sendo o tempo de vida de } \sigma \subset at_{i,j} \\ & \theta_1(\mathbf{Et}) \geq \theta_1(\mathbf{At}) \quad e \\ & \phi_i(\mathbf{Et}) \geq \phi_i(\mathbf{At}) \vee \phi_i(\mathbf{Et}) = \epsilon, \forall \phi_i(\mathbf{At}) \neq \epsilon, \forall i = 1 \text{ até } N, \end{aligned}$$

em que $[\epsilon]$ é a matriz nula, onde todos os seus elementos são ϵ e se $\forall i, j, \exists \sigma \subset et_{i,j} | \sigma \in \Sigma$.

Assim, tendo uma especificação de comportamento satisfazendo a Definição 4.26, define-se a condição de controlabilidade para SEDs modelados por matrizes de incidência temporizadas.

Definição 4.27 *Dada uma especificação de comportamento \mathbf{Et} válida, com $ACES(\mathbf{Et}) = \mathbf{Et}$, e a matriz de incidência temporizada \mathbf{At} que representa o autômato $A_{(\max,+)}$, a condição de controlabilidade para \mathbf{Et} é definida por*

$$ACES(\mathbf{Et} \oplus \mathbf{At}_{uc}) = \mathbf{Et}. \tag{4.42}$$

Essa definição é uma extensão da Definição 2.32 para o caso temporizado. Deve-se ver que o operador \leq avalia os vetores de estado inicial θ e de estados marcados ϕ (ver Algoritmo 4.4).

Exemplo 4.17 *Considerando as matrizes de incidência:*

$$\mathbf{At} = \begin{bmatrix} \epsilon & 3\alpha & \epsilon \\ 3\beta & \epsilon & 2\kappa \\ 3\lambda & \epsilon & \epsilon \end{bmatrix}, \quad \theta(\mathbf{At}) = \begin{bmatrix} 2 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{At}) = \begin{bmatrix} 2 \\ \epsilon \\ \epsilon \end{bmatrix} \quad e$$

$$\mathbf{Et} = \begin{bmatrix} \epsilon & 3\alpha & \epsilon \\ 4\beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \theta(\mathbf{Et}) = [2 \ \epsilon \ \epsilon], \quad \phi(\mathbf{Et}) = \begin{bmatrix} 3 \\ \epsilon \\ \epsilon \end{bmatrix}$$

e sendo $\Sigma_{uc} = \{\lambda\}$, então

$$\begin{aligned} ACES \left(\begin{bmatrix} \epsilon & 3\alpha & \epsilon \\ 4\beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix} \oplus \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ 3\lambda & \epsilon & \epsilon \end{bmatrix} \right) = \\ ACES \left(\begin{bmatrix} \epsilon & 3\alpha & \epsilon \\ 4\beta & \epsilon & \epsilon \\ 3\lambda & \epsilon & \epsilon \end{bmatrix} \right) = \begin{bmatrix} \epsilon & 3\alpha & \epsilon \\ 4\beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix} = \mathbf{Et}, \end{aligned}$$

e os vetores de estados $\theta(\mathbf{Et}) = \theta(\mathbf{At})$ e $\phi_1(\mathbf{Et}) > \phi_1(\mathbf{At})$ (ϕ_2 e ϕ_3 são iguais) garantem que \mathbf{Et} é uma especificação que gera uma linguagem controlável.

Considerando uma especificação de comportamento \mathbf{Et} , e a matriz de incidência temporizada \mathbf{At} que modela o SED, a partir da condição de controlabilidade (Definição 4.27), um supervisor pode ser sintetizado.

Lema 4.1 *Um supervisor \mathbf{St} para uma matriz de incidência \mathbf{At} é definido pela especificação de comportamento \mathbf{Et} se e somente se*

$$ACES(\mathbf{Et} \oplus \mathbf{At}_{uc}) = \mathbf{Et}.$$

Demonstração:

Igual à Demonstração do Lema 3.1, trocando as matrizes de incidência \mathbf{A} e \mathbf{E} pelas matrizes de incidência temporizadas \mathbf{At} e \mathbf{Et} . ♦

Assim, a condição de existência do supervisor \mathbf{St} é que a especificação de comportamento \mathbf{Et} contenha uma linguagem $L(\mathbf{Et}) \subseteq L(\mathbf{At})$, a qual deve ser controlável. Logo, se a especificação \mathbf{Et} define uma linguagem que satisfaz a condição de controlabilidade, o supervisor é a componente *trim* da própria especificação.

Corolário 4.1 *Dada uma especificação de comportamento \mathbf{Et} válida e uma matriz de incidência temporizada trim \mathbf{At} que é o modelo de um SED, $\mathbf{St} = TRIM(\mathbf{Et})$ se e somente se*

$$ACES(\mathbf{Et} \oplus \mathbf{At}_{uc}) = \mathbf{Et}.$$

Demonstração:

Igual à Demonstração do Corolário 3.1, trocando as matrizes de incidência \mathbf{A} e \mathbf{E} pelas matrizes de incidência temporizadas \mathbf{At} e \mathbf{Et} . ♦

Assim, se o Lema 4.1 é satisfeito, então a componente *trim* de \mathbf{Et} define o supervisor.

Exemplo 4.18 Para o autômato mostrado na Figura 4.15, onde $\Sigma = \{\alpha, \beta, \kappa, \mu\}$, $\Sigma_{uc} = \{\kappa\}$ e $\Sigma_c = \{\alpha, \beta, \mu\}$, sua representação matricial é dada por

$$\mathbf{At} = \begin{bmatrix} 2\alpha & 3\beta & 3\mu \\ \alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{bmatrix}, \quad \theta(\mathbf{At}) = [2 \ \epsilon \ \epsilon], \quad \phi(\mathbf{At}) = \begin{bmatrix} 3 \\ \epsilon \\ 1 \end{bmatrix}$$

Para a especificação

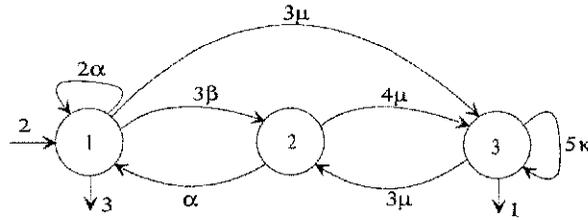


Figure 4.15: Autômato para exemplificar a síntese do supervisor através do Lema 4.1. e do Corolário 4.1.

$$\mathbf{Et} = \begin{bmatrix} \epsilon & 4\beta & \epsilon \\ \epsilon & \epsilon & 5\mu \\ \epsilon & \epsilon & 5\kappa \end{bmatrix}, \quad \theta(\mathbf{Et}) = [3 \ \epsilon \ \epsilon], \quad \phi(\mathbf{Et}) = \begin{bmatrix} \epsilon \\ \epsilon \\ 2 \end{bmatrix}$$

mostrada na Figura 4.16(a), o supervisor é definido por $\mathbf{St} = \text{TRIM}(\mathbf{Et}) = \mathbf{Et}$, pois

$$\begin{aligned} & \text{ACES} \left(\left(\begin{bmatrix} \epsilon & 4\beta & \epsilon \\ \epsilon & \epsilon & 5\mu \\ \epsilon & \epsilon & 5\kappa \end{bmatrix} \oplus \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 5\kappa \end{bmatrix} \right) \right) = \\ & \text{ACES} \left(\begin{bmatrix} \epsilon & 4\beta & \epsilon \\ \epsilon & \epsilon & 5\mu \\ \epsilon & \epsilon & 5\kappa \end{bmatrix} \right) = \begin{bmatrix} \epsilon & 4\beta & \epsilon \\ \epsilon & \epsilon & 5\mu \\ \epsilon & \epsilon & 5\kappa \end{bmatrix} = \mathbf{Et}, \end{aligned}$$

e os vetores de estados são $\theta_1(\mathbf{Et}) > \theta_1(\mathbf{At})$ (θ_2 e θ_3 são iguais) e $\phi_1(\mathbf{Et}) = \epsilon$, $\phi_2(\mathbf{Et}) = \phi_2(\mathbf{At})$ e $\phi_3(\mathbf{Et}) > \phi_3(\mathbf{At})$, que garante que \mathbf{Et} é uma especificação válida, e satisfaz o Lema 4.1. Entretanto, considerando a especificação de comportamento

$$\mathbf{Et} = \begin{bmatrix} 3\alpha & 3\beta & 4\mu \\ \epsilon & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{bmatrix}, \quad \theta(\mathbf{Et}) = [4 \ \epsilon \ \epsilon], \quad \phi(\mathbf{Et}) = \begin{bmatrix} 5 \\ \epsilon \\ \epsilon \end{bmatrix}$$

mostrada na Figura 4.16(b), tem-se que

$$\begin{aligned} & \text{ACES} \left(\left(\begin{bmatrix} 3\alpha & 3\beta & 4\mu \\ \epsilon & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{bmatrix} \oplus \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 5\kappa \end{bmatrix} \right) \right) = \\ & \text{ACES} \left(\begin{bmatrix} 3\alpha & 3\beta & 4\mu \\ \epsilon & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{bmatrix} \right) = \begin{bmatrix} 3\alpha & 3\beta & 4\mu \\ \epsilon & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{bmatrix} = \mathbf{Et}, \end{aligned}$$

em que os vetores de estados são $\theta_1(\mathbf{Et}) > \theta_1(\mathbf{At})$ (θ_2 e θ_3 são iguais) e $\phi_1(\mathbf{Et}) = 5$, $\phi_2(\mathbf{Et}) > \phi_2(\mathbf{At})$ e $\phi_3(\mathbf{Et}) = \epsilon$, que garante que \mathbf{Et} é controlável. Contudo, \mathbf{Et} não é trim (veja que ao sair da linha 1, a matriz não mais retorna para ela, que é a única marcada - $\phi_1(\mathbf{Et}) = 5$). Assim, o supervisor é dado por

$$\mathbf{St} = \text{TRIM}(\mathbf{Et}) = \text{TRIM} \left(\begin{bmatrix} 3\alpha & 3\beta & 4\mu \\ \epsilon & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{bmatrix} \right)$$

$$\mathbf{St} = \begin{bmatrix} 3\alpha & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \theta(\mathbf{St}) = [4 \ \epsilon \ \epsilon], \quad \phi(\mathbf{St}) = \begin{bmatrix} 5 \\ \epsilon \\ \epsilon \end{bmatrix}.$$

Por outro lado, para a especificação

$$\mathbf{Et} = \begin{bmatrix} \epsilon & 3\beta & 4\mu \\ 2\alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & \epsilon \end{bmatrix}, \quad \theta(\mathbf{Et}) = [3 \ \epsilon \ \epsilon], \quad \phi(\mathbf{Et}) = \begin{bmatrix} 3 \\ \epsilon \\ 2 \end{bmatrix}$$

mostrada na Figura 4.16(c), tem-se que

$$\text{ACES} \left(\begin{bmatrix} \epsilon & 3\beta & 4\mu \\ 2\alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & \epsilon \end{bmatrix} \oplus \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 5\kappa \end{bmatrix} \right) =$$

$$\text{ACES} \left(\begin{bmatrix} \epsilon & 3\beta & 4\mu \\ 2\alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{bmatrix} \right) = \begin{bmatrix} \epsilon & 3\beta & 4\mu \\ 2\alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{bmatrix} \triangleright \mathbf{Et},$$

o que não satisfaz o Lema 4.1.

Da mesma forma que para a síntese de supervisores para SEDs não temporizados, quando a especificação de comportamento não é factível por atingir estados onde eventos não controláveis podem ocorrer, é necessário encontrar um supervisor que realize o comportamento especificado restritivamente. Isto que dizer que para uma especificação de comportamento válida, se

$$\text{ACES}(\mathbf{Et} \oplus \mathbf{At}_{uc}) \triangleright \mathbf{Et},$$

então, é necessário encontrar a suprema sublinguagem controlável. A solução desse problema pode ser encontrada utilizando a Definição 4.12 da matriz de caminhos temporizada.

Considerando que os eventos da matriz de incidência temporizada pertencem ao conjunto $\Sigma = \Sigma_c \cup \Sigma_{uc}$, a matriz \mathbf{At}^n contém palavras de comprimento n , compostas de eventos controláveis e não controláveis. Então, utilizando-se dessa definição, é preciso avaliar quais os caminhos que iniciam com um evento qualquer (controlável ou não controlável), e que seja seguido apenas por eventos controláveis ou por eventos não controláveis.

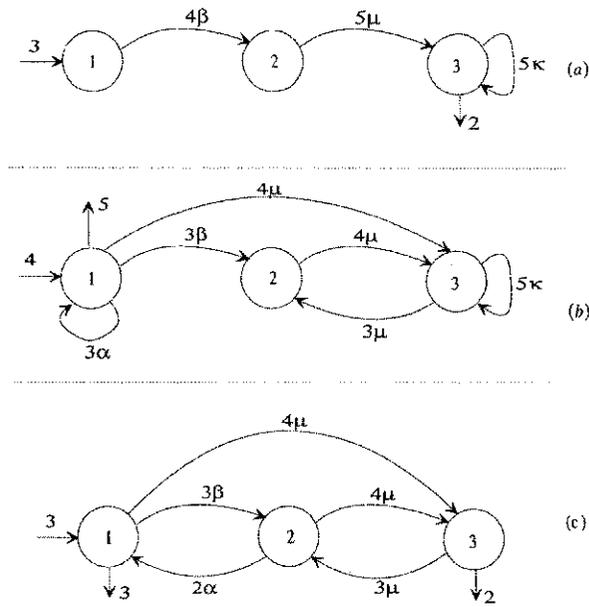


Figure 4.16: (a) e (b) Especificações controláveis e (c) Especificação não controlável.

Definição 4.28 Seja $\Sigma = \Sigma_c \cup \Sigma_{uc}$, e seja um autômato $A_{(\max,+)}$ construído com símbolos de Σ . A matriz de caminhos \mathbf{Ct}_c^n , que define as palavras que mudam o estado do autômato do estado i para o estado j , iniciados por um evento qualquer (controlável ou não controlável), e seguidos sempre de eventos controláveis, é definida por

$$\mathbf{Ct}_c^n = \mathbf{At} \otimes (\mathbf{At}_c)^{n-1}.$$

O vetor de estados marcados $\phi(\mathbf{Ct}_c^n)$ e o vetor de estado inicial $\theta(\mathbf{Ct}_c^n)$ são os mesmos de \mathbf{At} .

De acordo com essa definição, cada elemento de \mathbf{Ct}_c^n contém seqüências iniciadas por um evento qualquer, seguidos de $n - 1$ eventos controláveis. De forma semelhante, define-se:

Definição 4.29 Seja $\Sigma = \Sigma_c \cup \Sigma_{uc}$, e seja um autômato $A_{(\max,+)}$ construído com símbolos de Σ . A matriz de caminhos \mathbf{Ct}_{uc}^n , que define as palavras que mudam o estado do autômato de i para o estado j , iniciados por um evento qualquer (controlável ou não controlável), e seguidos sempre de eventos não controláveis, é definida por

$$\mathbf{Ct}_{uc}^n = \mathbf{At} \otimes (\mathbf{At}_{uc})^{n-1}.$$

O vetor de estados marcados $\phi(\mathbf{Ct}_{uc}^n)$ e o vetor de estado inicial $\theta(\mathbf{Ct}_{uc}^n)$ são os mesmos de \mathbf{At} .

Igualmente à matriz de caminhos não temporizada, um determinado evento σ_{uc} de $(at_{uc})_{i,j}$, aparece como último evento nas seqüências dos termos de $(ct_{uc}^n)_{k,j}$. Estes termos de \mathbf{Ct}_{uc}^n são antecedidos pelos elementos de $(ct_{uc}^{n-1})_{k,i}$.

Exemplo 4.19 O autômato visto na Figura 4.17, com $\Sigma_c = \{\alpha, \kappa\}$ e $\Sigma_{uc} = \{\beta, \eta\}$, tem

$$\mathbf{At} = \begin{bmatrix} \epsilon & 2\beta & \epsilon & \epsilon & \epsilon \\ 5\eta & \epsilon & 3\alpha & \epsilon & \epsilon \\ 2\alpha + 3\kappa & \epsilon & \epsilon & 3\eta & \epsilon \\ 1\eta & 2\beta & \epsilon & \epsilon\alpha & 3\kappa \\ \epsilon & \epsilon & 2\eta & \epsilon & \epsilon \end{bmatrix}$$

$$\theta(\mathbf{At}) = \begin{bmatrix} 2 & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{At}) = \begin{bmatrix} 2 \\ \epsilon \\ 4 \\ 3 \\ \epsilon \end{bmatrix}$$

e

$$\mathbf{At}_{uc} = \begin{bmatrix} \epsilon & 2\beta & \epsilon & \epsilon & \epsilon \\ 5\eta & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 3\eta & \epsilon \\ 1\eta & 2\beta & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 2\eta & \epsilon & \epsilon \end{bmatrix}$$

com $\theta(\mathbf{At}_{uc}) = \theta(\mathbf{At})$ e $\phi(\mathbf{At}_{uc}) = \phi(\mathbf{At})$. Calculando \mathbf{Ct}_{uc}^2 , tem-se

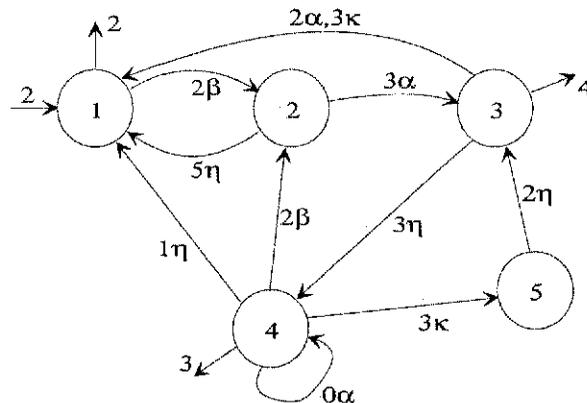


Figure 4.17: Autômato utilizado como exemplo da construção da matriz de caminhos temporizada \mathbf{Ct}_{uc}^n .

$$\mathbf{Ct}_{uc}^2 = \begin{bmatrix} 7\beta\eta & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 7\eta\beta & \epsilon & 6\alpha\eta & \epsilon \\ 4\eta\eta & 4\alpha\beta + 5\kappa\beta + 5\eta\beta & \epsilon & \epsilon & \epsilon \\ 7\beta\eta + 1\alpha\eta & 3\eta\beta + 2\alpha\beta & 5\kappa\eta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 5\eta\eta & \epsilon \end{bmatrix}$$

com $\theta(\mathbf{Ct}_{uc}^2) = \theta(\mathbf{At})$ e $\phi(\mathbf{Ct}_{uc}^2) = \phi(\mathbf{At})$. Nela, vê-se que o elemento $(at_{uc})_{3,4} = \eta$ aparece em $(ct_{uc}^2)_{k,A}$, para $k = 2$ e $k = 5$. Isto é, nas linhas $k = 2$ e $k = 5$, e na coluna 3 de \mathbf{At} ,

encontram-se os elementos que antecedem $(at_{uc})_{3,4} = \eta$ em $(ct_{uc}^2)_{k,4}$. Logo, os termos $(at_{uc}^2)_{2,4}$ e $(at_{uc}^2)_{5,4}$ contêm o elemento $(at_{uc})_{3,4} = \eta$ como sendo o último evento de suas seqüências. Calculando \mathbf{Ct}_{uc}^3 , tem-se

$$\mathbf{Ct}_{uc}^3 = \begin{bmatrix} \epsilon & 9\beta\eta\beta & \epsilon & \epsilon & \epsilon \\ 12\eta\beta\eta + 7\alpha\eta\eta & 8\alpha\eta\beta & \epsilon & \epsilon & \epsilon \\ 9\alpha\beta\eta + 10\kappa\beta\eta + 10\eta\beta\eta & 6\eta\eta\beta & \epsilon & \epsilon & \epsilon \\ 8\eta\beta\eta + 7\alpha\beta\eta & 9\beta\eta\beta + 3\alpha\eta\beta & \epsilon & 8\kappa\eta\eta & \epsilon \\ 6\eta\eta\eta & 7\eta\eta\beta & \epsilon & \epsilon & \epsilon \end{bmatrix}$$

em que $\theta(\mathbf{Ct}_{uc}^2) = \theta(\mathbf{At})$ e $\phi(\mathbf{Ct}_{uc}^2) = \phi(\mathbf{At})$. Nessa matriz, o termo $(at_{uc}^3)_{4,4}$ apresenta o termo $(ct_{uc}^2)_{k,3}$, $k = 4$, antecedendo o termo $(at_{uc})_{3,4} = \eta$.

Com essas definições, procede-se à construção do supervisor quando o Lema 4.1 não é satisfeito. Assim, utilizando a mesma idéia da Definição 3.30 para definir caminhos em um dado autômato que iniciam com eventos da especificação de comportamento \mathbf{Et} , seguidos apenas por eventos não controláveis de \mathbf{At} , tem-se:

Definição 4.30 Dada a matriz de incidência \mathbf{Et} , que é uma matriz de caminhos de comprimento 1, define-se a matriz de caminhos temporizada

$$\mathbf{Bt}_{uc}^n = \mathbf{Et} \otimes (\mathbf{At}_{uc})^{n-1}$$

como sendo uma matriz de caminhos, em que o primeiro elemento de cada seqüência é um elemento de \mathbf{Et} , e os demais são elementos pertencentes a \mathbf{At}_{uc} . O vetor de estados marcados ϕ é definido por

$$\phi_i(\mathbf{Bt}_{uc}^n) = \begin{cases} \phi_i(\mathbf{Et}) \oplus \phi_i(\mathbf{At}), & \text{se } \phi_i(\mathbf{At}) \neq \epsilon \\ \epsilon & \text{caso contrário,} \end{cases} \quad (4.43)$$

e o vetor de estado inicial é $\theta(\mathbf{Bt}_{uc}^n) = \theta(\mathbf{At}) = \theta(\mathbf{Et})$.

Observação: A utilização da notação de uma matriz \mathbf{Bt} genérica citada anteriormente para algumas definições de operadores não deve ser confundida com a matriz \mathbf{Bt}_{uc}^n , que é uma definição específica para denotar a matriz de caminhos em que os primeiros elementos de cada caminho pertencem à \mathbf{Et} , seguidos por elementos de \mathbf{At}_{uc} .

Na matriz \mathbf{Bt}_{uc}^n pode-se avaliar se existe algum evento não controlável do autômato, em que sua ocorrência seja possível a partir de uma seqüência da especificação. Se existir, este evento aparece como último evento em uma determinada seqüência em \mathbf{Bt}_{uc}^n . A avaliação dos termos de \mathbf{Bt}_{uc}^n determina uma condição de teste semelhante a

$$\Sigma(H(x)) \cap \Sigma_{uc} \subset \Sigma(x),$$

contudo, utilizada para o caso temporizado específico dessa Tese. Também, deve-se observar que, para $n = 1$, tem-se

$$\mathbf{Bt}_{uc}^1 = \mathbf{Et} \otimes (\mathbf{At}_{uc})^0 = \mathbf{Et} \otimes \mathbf{I} = \mathbf{Et},$$

onde \mathbf{I} é a matriz identidade formada por e na diagonal principal e ϵ nos demais elementos.

Igualmente à matriz de caminhos \mathbf{Ct}_{uc}^n , para uma matriz de caminhos \mathbf{Bt}_{uc}^n , um evento $\sigma_{uc} \in (at_{uc})_{i,j}$ que não pertence à \mathbf{Et} , pode aparecer finalizando uma seqüência nos termos de $(bt_{uc}^n)_{k,j}$. Estes termos são antecedidos pelos termos de $(bt_{uc}^{n-1})_{k,i}$.

Teorema 4.3 *Dada uma especificação de comportamento \mathbf{Et} válida e a matriz de incidência temporizada dos eventos não controláveis \mathbf{At}_{uc} , do autômato $A_{(\max,+)}$ trim, se*

$$ACES(\mathbf{Et} \oplus \mathbf{At}_{uc}) \triangleright \mathbf{Et},$$

então a sup $C(L)$ será determinada recursivamente por:

1. Para $n = 1$, $\mathbf{St}^1 = \mathbf{Et}$.
2. Para $n = n + 1$, enquanto $(n \leq N) \wedge \exists \sigma_{uc} \notin \mathbf{Et}$ então

$$\begin{aligned} \mathbf{Bt}_{uc}^n &= \mathbf{Et} \otimes (\mathbf{At}_{uc})^{n-1} \\ \mathbf{St}^n &= [st_{ij}^n], \quad st_{ij}^n = \begin{cases} st_{ij}^{n-1} & \text{se } \sigma^1 \sigma_{uc}^2 \dots \sigma_{uc}^n \in \mathbf{Bt}_{uc}^n \wedge \sigma_{uc}^n \in \mathbf{Et}; \\ \epsilon & \text{se } \sigma_{uc}^n \notin \mathbf{Et} \wedge \sigma^1 \in \Sigma_c \end{cases} \\ \mathbf{St}^n &= TRIM(\mathbf{St}^n) \end{aligned}$$

onde σ_{uc}^n é o n -ésimo evento da seqüência de um termo de \mathbf{Bt}_{uc}^n , que pode ser $\sigma_{uc} \notin \mathbf{Et}$.

3. Se $(n > N) \wedge (\exists \sigma_{uc} \notin \mathbf{Et} \text{ em } \mathbf{St}^n)$, então $\mathbf{St} = [\epsilon]$.

Demonstração:

Igual à demonstração do Teorema 3.3, utilizando as matrizes de incidência temporizadas.



Vê-se nesse Teorema que se o último evento de uma seqüência em \mathbf{Bt}_{uc}^n não pertence à especificação, a inibição do primeiro evento elimina esta seqüência. A existência de outros eventos não pertencentes à especificação, é eliminada recursivamente em \mathbf{Bt}_{uc}^n , para $n = 2, 3, \dots, N$, construindo o supervisor. Se algum evento não controlável que não pertence à especificação \mathbf{Et} não for eliminado de acordo com o Teorema 4.3, não é possível sintetizar um supervisor para a especificação desejada. De forma similar ao caso não temporizado, se $\forall n$, \mathbf{St}^n não for coacessível, $\mathbf{S} = [\epsilon]$.

Exemplo 4.20 *Considerando que o autômato mostrado na Figura 4.15 tem*

$$\mathbf{At} = \begin{bmatrix} 2\alpha & 3\beta & 3\mu \\ \alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{bmatrix}, \quad \theta(\mathbf{At}) = \begin{bmatrix} 2 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{At}) = \begin{bmatrix} 3 \\ \epsilon \\ 1 \end{bmatrix}$$

em que $\Sigma = \{\alpha, \beta, \mu, \kappa\}$ e $\Sigma_{uc} = \{\kappa\}$, é dada a especificação

$$\mathbf{Et} = \begin{bmatrix} \epsilon & 4\beta & 3\mu \\ \alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & \epsilon \end{bmatrix}, \quad \theta(\mathbf{Et}) = [4 \ \epsilon \ \epsilon], \quad \phi(\mathbf{Et}) = \begin{bmatrix} 5 \\ \epsilon \\ \epsilon \end{bmatrix}$$

a qual está apresentada na Figura 4.18. Calculando $ACES(\mathbf{Et} \oplus \mathbf{At}_{uc})$, encontra-se

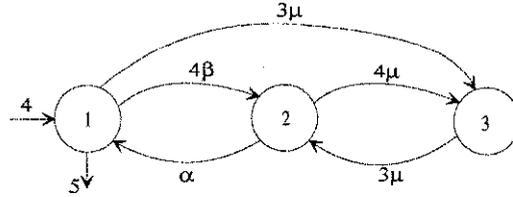


Figure 4.18: Especificação para síntese do supervisor que não satisfaz a condição de controlabilidade.

$$ACES \left(\left(\begin{bmatrix} \epsilon & 4\beta & 3\mu \\ \alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & \epsilon \end{bmatrix} \oplus \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 5\kappa \end{bmatrix} \right) \right) =$$

$$ACES \left(\begin{bmatrix} \epsilon & 4\beta & 3\mu \\ \alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{bmatrix} \right) = \begin{bmatrix} \epsilon & 4\beta & 3\mu \\ \alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{bmatrix} \triangleright \mathbf{Et}$$

Então,

$$\mathbf{St}^1 = \mathbf{Et} = \begin{bmatrix} \epsilon & 4\beta & 3\mu \\ \alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & \epsilon \end{bmatrix}.$$

Calculando \mathbf{Bt}_{uc}^2 , tem-se

$$\mathbf{Bt}_{uc}^2 = \mathbf{Et} \otimes \mathbf{At}_{uc} = \begin{bmatrix} \epsilon & \epsilon & 8\mu\kappa \\ \epsilon & \epsilon & 9\mu\kappa \\ \epsilon & \epsilon & \epsilon \end{bmatrix}$$

que, utilizando o Teorema 4.3, encontram-se os termos $(bt_{uc}^2)_{1,3} = 8\mu\kappa$ e $(bt_{uc}^2)_{2,3} = 9\mu\kappa$, que contêm o evento controlável μ seguido do evento não controlável κ , que não pertence à especificação. Como μ é controlável, então fazendo $st_{1,3}^2 = st_{2,3}^2 = \epsilon$, elimina-se o estado 3. Logo, tornando-se o estado 3 não acessível, $st_{3,2}^2 = st_{3,3}^2 = \epsilon$. Assim, o supervisor é determinado por

$$\mathbf{St} = \text{TRIM}(\mathbf{St}^2) = \begin{bmatrix} \epsilon & 4\beta & \epsilon \\ \alpha & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \theta(\mathbf{St}) = [4 \ \epsilon \ \epsilon], \quad \phi(\mathbf{St}) = \begin{bmatrix} 5 \\ \epsilon \\ \epsilon \end{bmatrix}$$

o qual está mostrado na Figura 4.19. Este supervisor determina que o SED reconheça a linguagem temporizada

$$L_m(\mathbf{St}|\mathbf{At}) = (y|(\beta\alpha)^0)(\beta\alpha)^0 \oplus (y|(\beta\alpha)^1)(\beta\alpha)^1 \oplus \dots$$

$$L_m(\mathbf{St}|\mathbf{At}) = 9 + 14\beta\alpha + 19\beta\alpha + \dots$$

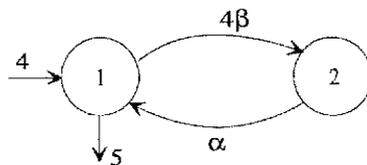


Figure 4.19: Supervisor para a especificação que não satisfaz a condição de controlabilidade.

Observa-se no Exemplo 4.20 que, se o estado 1 da especificação não for marcado, $\mathbf{S} = [\epsilon]$.

4.4 Algoritmo de síntese do supervisor

A partir do que foi apresentado, o seguinte algoritmo de síntese do supervisor é construído:

Algoritmo 4.5 Construção da $\text{sup}C(L)$

1. Se $\mathbf{Et} \not\leq \mathbf{At}$, Construa $\mathbf{At}^\#$ e $\mathbf{Et}^\#$ a partir de \mathbf{At} e \mathbf{Et} , e faça $\mathbf{Et} = \mathbf{Et}^\#$ e $\mathbf{At} = \mathbf{At}^\#$.
2. Faça $\mathbf{Dt} = \text{ACES}(\mathbf{Et} \oplus \mathbf{At}_{uc})$.
3. Se $\mathbf{Dt} = \mathbf{Et}$, faça $\mathbf{St} = \text{TRIM}(\mathbf{Et})$ e pare.
4. Se $\mathbf{Dt} \triangleright \mathbf{Et}$, faça $n = 1$:
 - a) $\mathbf{St}^n = \mathbf{Et}$, $x\text{dif}(k, n-1) = i$ e $y\text{dif}(k, n-1) = j$ (onde $\mathbf{At}_{uc} \neq \mathbf{Et}_{uc}$), para $k = 1, \dots, M$ (M número de elementos diferentes entre \mathbf{At}_{uc} e \mathbf{Et}_{uc}).
 - b) Calcule \mathbf{Bt}_{uc}^n .
 - c) Faça para $k = 1$ até M
 - i. Procure os elementos em $\mathbf{Bt}_{uc}^n(i, y\text{dif}(k, n-1))$, onde $\sigma_{uc}^n \notin \mathbf{Et}$ (σ_{uc}^n sendo o último elemento da seqüência).
 - (1) Se $\sigma_{uc}^n \notin \mathbf{Et}$ e $\sigma^1 \in \Sigma_c$, faça $\mathbf{St}^n(i, x\text{dif}(k, n-1)) = \epsilon$ e calcule $\text{COACES}(\mathbf{St}^n)$ e $\text{ACES}(\mathbf{St}^n)$;
 - (2) Se $\sigma_{uc}^n \notin \mathbf{Et}$ e $\sigma^1 \in \Sigma_{uc}$, faça $x\text{dif}(k, n) = i$ e $y\text{dif}(k, n) = y\text{dif}(k, n-1)$.
 - d) Se $\forall k = 1$ até M , $\mathbf{St}^n(x\text{dif}(k, n-1), y\text{dif}(k, n-1)) = \epsilon$ pare. Caso contrário, faça $n = n + 1$ e retorne ao passo 4.b.

i) Se $n > N$ pare (Et não é factível).

Os algoritmos dos operadores *ACES* e *COACES* utilizados para essa abordagem temporizada, são os mesmos do caso não temporizado. As mesmas considerações feitas para o algoritmo não temporizado são válidas para o caso temporizado, aqui estudado. Observa-se que esse algoritmo é semelhante ao Algoritmo 3.7, apresentando apenas a inclusão da representação temporal e utilizando os operadores de composição, expansão, interseção e continência definidos para este caso. Dessa forma, é uma alternativa ao algoritmo de Brandin e Wonham [BW94], na construção de supervisores de SEDs temporizados quando se consideram todos os eventos pertencentes ao conjunto de eventos remotos. Por outro lado, deve-se observar que a transformação dos tempos de vida de todos os eventos em $t_\sigma = e$, bem como o atraso inicial e todos os atrasos finais, o algoritmo soluciona o caso não temporizado apresentado no Capítulo 3.

4.5 Exemplos da utilização do algoritmo

Exemplo 4.21 No autômato visto na Figura 4.20, com $\Sigma = \{\alpha, \beta, \kappa, \eta, \lambda, \mu\}$ e $\Sigma_{uc} = \{\alpha, \kappa, \lambda\}$, a representação matricial é dada por

$$\mathbf{At} = \begin{bmatrix} \epsilon & 2\alpha & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ 3\lambda & 2\beta & 1\kappa & 2\mu & \epsilon & \epsilon & \epsilon \\ \epsilon & 4\beta & \epsilon & 3\kappa + 2\mu & \epsilon & \epsilon & 5\eta \\ 2\eta & \epsilon & \epsilon & \epsilon & 4\beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 1\lambda & \epsilon & 2\alpha & 2\eta \\ 3\alpha & \epsilon & \epsilon & 3\eta & 3\kappa & 3\beta + 2\lambda & \epsilon \\ \epsilon & e\mu & 5\eta + 3\lambda & \epsilon & \epsilon & \epsilon & 4\alpha + e\beta \end{bmatrix}$$

$$\theta(\mathbf{At}) = \begin{bmatrix} 1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}, \phi(\mathbf{At}) = \begin{bmatrix} \epsilon \\ 2 \\ 1 \\ \epsilon \\ \epsilon \\ \epsilon \\ 3 \end{bmatrix}$$

Definindo a especificação de comportamento por

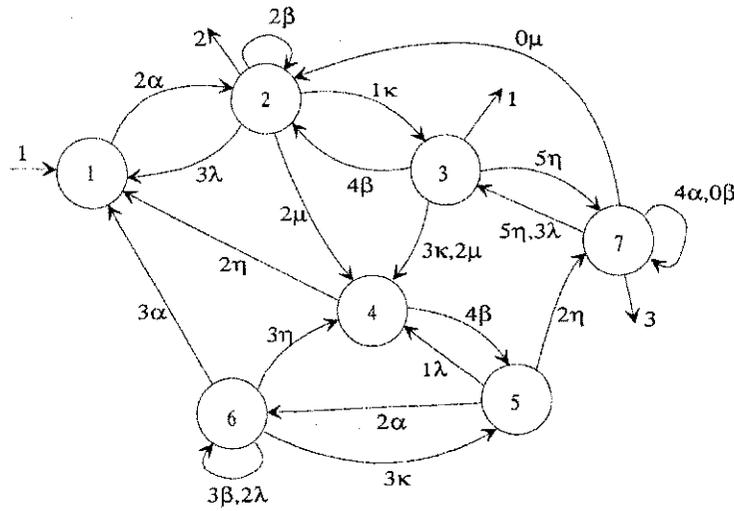


Figure 4.20: Autômato temporizado para construir um supervisor.

$$\mathbf{Et} = \begin{bmatrix} \epsilon & 2\alpha & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ 3\lambda & \epsilon & 1\kappa & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 3\kappa & \epsilon & \epsilon & \epsilon \\ 2\eta & \epsilon & \epsilon & \epsilon & 4\beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 2\alpha & 2\eta \\ 3\alpha & \epsilon & \epsilon & \epsilon & \epsilon & 2\lambda & \epsilon \\ \epsilon & \epsilon & 3\lambda & \epsilon & \epsilon & \epsilon & 4\alpha \end{bmatrix}$$

$$\theta(\mathbf{Et}) = \begin{bmatrix} 2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}, \phi(\mathbf{Et}) = \begin{bmatrix} \epsilon \\ 2 \\ 1 \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \end{bmatrix}$$

a qual está mostrada graficamente na Figura 4.21, observa-se que $ACES(\mathbf{Et} \oplus \mathbf{At}_{uc}) \triangleright \mathbf{Et}$. Assim, os termos que não pertencem à \mathbf{Et} são $(at_{uc})_{5,4} = \lambda$ e $(at_{uc})_{6,5} = \kappa$. De acordo com o Teorema 4.3, faz-se $\mathbf{St}^1 = \mathbf{Et}$ e calcula-se $\mathbf{Bt}_{uc}^2 = \mathbf{Et} \otimes \mathbf{At}_{uc}$ para avaliar quais os eventos que devem ser desabilitados para encontrar uma linguagem restrita para o supervisor.

$$\mathbf{Bt}_{uc}^2 = \begin{bmatrix} 5\alpha\lambda & \epsilon & 3\alpha\kappa & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 5\lambda\alpha & \epsilon & 4\kappa\kappa & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 4\eta\alpha & \epsilon & \underline{5\beta\lambda} & \epsilon & 6\beta\alpha & \epsilon \\ 5\alpha\alpha & \epsilon & 5\eta\lambda & \epsilon & \underline{5\alpha\kappa} & 4\alpha\lambda & 6\eta\alpha \\ 5\lambda\alpha & 5\alpha\alpha & \epsilon & \epsilon & 5\lambda\kappa & 4\lambda\lambda & \epsilon \\ \epsilon & \epsilon & 7\alpha\lambda & 6\lambda\kappa & \epsilon & \epsilon & 8\alpha\alpha \end{bmatrix}$$

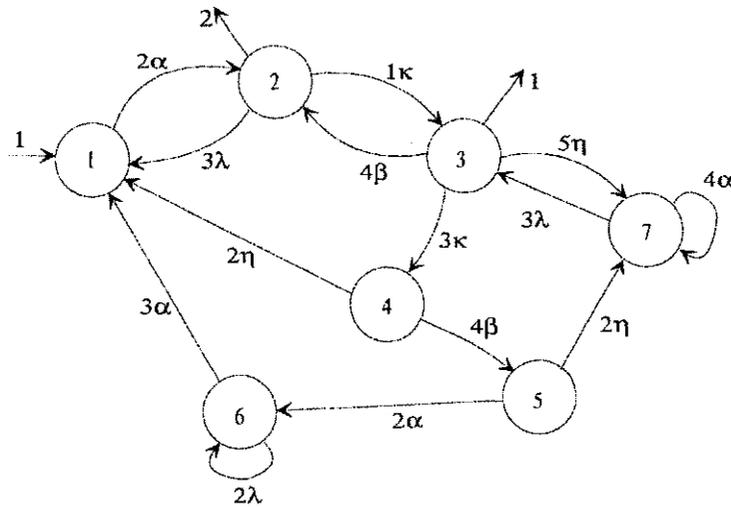


Figure 4.21: Especificação de comportamento temporizada.

com $\theta(\mathbf{Bt}_{uc}^2) = \theta(\mathbf{Et})$ e $\phi(\mathbf{Bt}_{uc}^2) = \phi(\mathbf{Et})$. Pode-se observar nessa matriz de caminhos que o termo $(bt_{uc}^2)_{4,4} = 5\beta\lambda$. Esse termo indica uma palavra que leva o autômato através do evento controlável β do estado 4 para o estado 5, e depois retorna para o estado 4 pelo evento não controlável λ , o qual não faz parte da especificação \mathbf{Et} . Tornando $st_{4,5} = \epsilon$, calcula-se

$$St^2 = COACES(St^2) = \begin{bmatrix} \epsilon & 2\alpha & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ 3\lambda & \epsilon & 1\kappa & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 3\kappa & \epsilon & \epsilon & \epsilon \\ 2\eta & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 2\alpha & 2\eta \\ 3\alpha & \epsilon & \epsilon & \epsilon & \epsilon & 2\lambda & \epsilon \\ \epsilon & \epsilon & 3\lambda & \epsilon & \epsilon & \epsilon & 4\alpha \end{bmatrix}$$

$$\theta(\mathbf{Et}) = \begin{bmatrix} 2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}, \phi(\mathbf{Et}) = \begin{bmatrix} \epsilon \\ 2 \\ 1 \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \end{bmatrix}$$

e

$$\text{St}^2 = ACES(\text{St}^2) = \begin{bmatrix} \epsilon & 2\alpha & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ 3\lambda & \epsilon & 1\kappa & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 3\kappa & \epsilon & \epsilon & \epsilon \\ 2\eta & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix} \tag{4.44}$$

$$\theta(\mathbf{Et}) = \begin{bmatrix} 2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}, \phi(\mathbf{Et}) = \begin{bmatrix} \epsilon \\ 2 \\ 1 \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \end{bmatrix}$$

que é o supervisor St , o qual está mostrado na Figura 4.22.

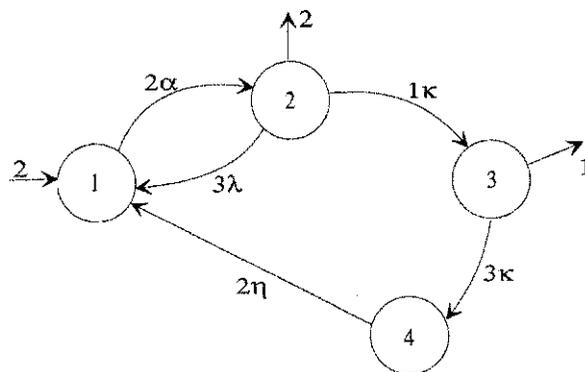


Figure 4.22: Supervisor que gera a $\text{sup}C(L)$ construído através da especificação de comportamento temporizada que não satisfaz a condição de controlabilidade.

Exemplo 4.22 Considere o GTT apresentado na Figura 4.23(a) e considere $\Sigma_{uc} = \{\kappa\}$, com todos os eventos definidos como eventos remotos. Para a especificação de comportamento mostrada na Figura 4.23(b), utilizando a abordagem de Brandin e Wonham [BW94], a especificação não é factível. Fazendo a composição síncrona dos GTAs, cujas evoluções dinâmicas são representadas pelas respectivas estruturas dos GTTs (Figuras 4.23(a) e 4.23(b)), encontra-se o GTT apresentado na Figura 4.23(c). Utilizando o procedimento de síntese de [BW94], encontra-se o supervisor visto na Figura 4.23(d). Considerando a abordagem aqui proposta, os GTTs das Figuras 4.23(a) e 4.23(b), são representados pelos autômatos (max, +) vistos

nas Figuras 4.24(a) e 4.24(b), que têm suas representações matriciais dadas por

$$\mathbf{At} = \begin{bmatrix} \epsilon & 2\alpha & 1\beta & \epsilon \\ 1\alpha & \epsilon & \epsilon & \epsilon \\ \epsilon & 2\kappa & \epsilon & 2\alpha \\ e\alpha & \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \theta(\mathbf{At}) = [e \ \epsilon \ \epsilon \ \epsilon], \quad \phi(\mathbf{At}) = \begin{bmatrix} e \\ \epsilon \\ \epsilon \\ \epsilon \end{bmatrix}$$

$$\mathbf{Et} = \begin{bmatrix} \epsilon & 3\alpha & 2\beta & \epsilon \\ 2\alpha & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & e\alpha \\ e\alpha & \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \theta(\mathbf{Et}) = [e \ \epsilon \ \epsilon \ \epsilon], \quad \phi(\mathbf{Et}) = \begin{bmatrix} e \\ \epsilon \\ \epsilon \\ \epsilon \end{bmatrix}$$

Essa especificação \mathbf{Et} é uma submatriz de \mathbf{At} . Porém, para ser válida, o tempo de vida do

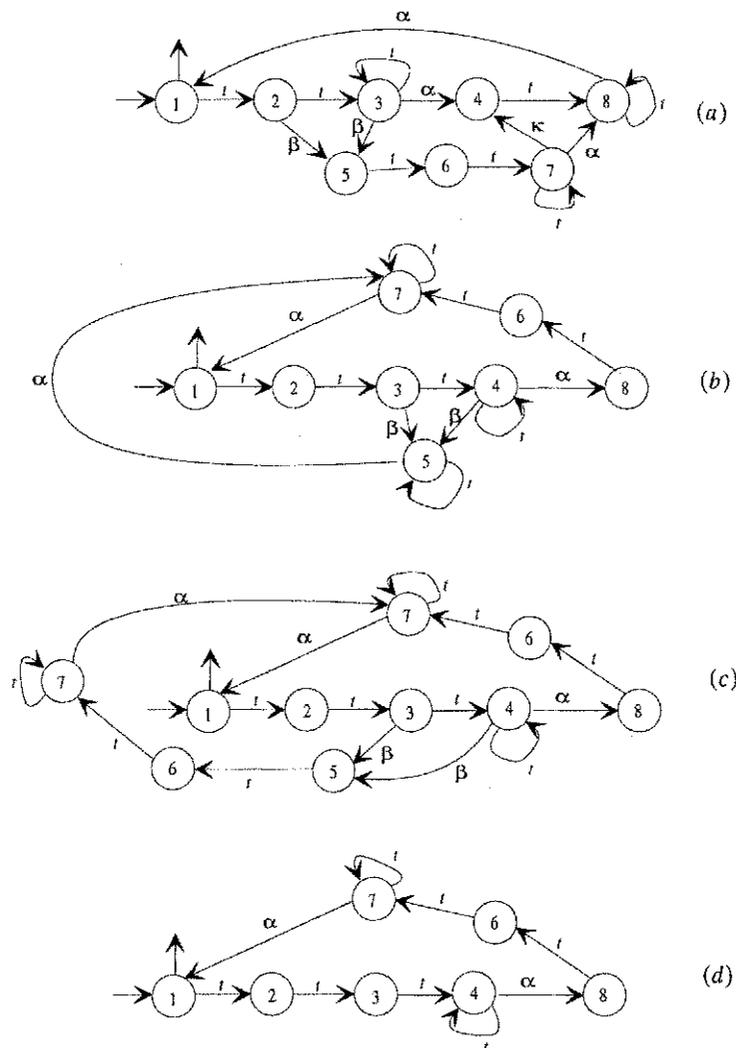


Figure 4.23: (a) Grafo de transição temporizada, (b) Especificação de comportamento e (c) Supervisor construído pela abordagem de Brandin e Wonham.

evento α no elemento $et_{3,4}$ deve ser $t_\alpha = e \oplus 2 = 2$. Dessa forma, \mathbf{Et} se torna válida (vista

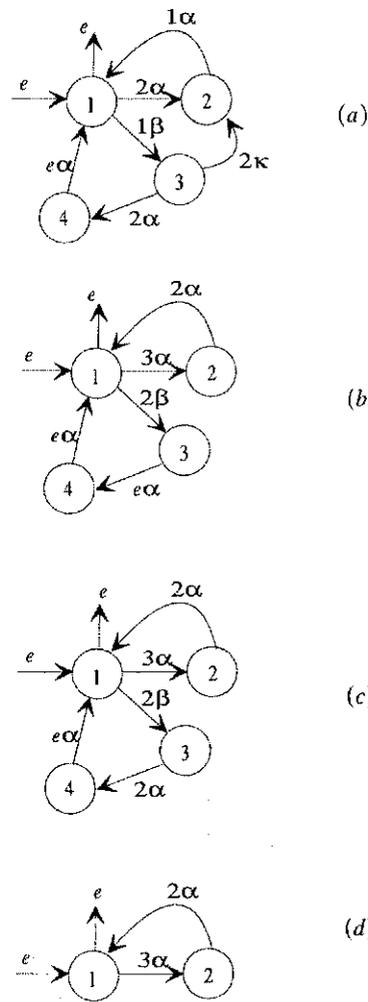


Figure 4.24: (a) Autômato (max,+), (b) Especificação de comportamento e (c) Supervisor construído pela abordagem das matrizes de incidência.

na Figura 4.24(c)). Assim, tem-se

$$\begin{aligned}
 & ACES \left(\left[\begin{array}{cccc} \epsilon & 3\alpha & 2\beta & \epsilon \\ 2\alpha & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 2\alpha \\ e\alpha & \epsilon & \epsilon & \epsilon \end{array} \right] \oplus \left[\begin{array}{cccc} \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 2\kappa & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \end{array} \right] \right) \\
 & ACES \left(\left[\begin{array}{cccc} \epsilon & 3\alpha & 2\beta & \epsilon \\ 2\alpha & \epsilon & \epsilon & \epsilon \\ \epsilon & 2\kappa & \epsilon & 2\alpha \\ e\alpha & \epsilon & \epsilon & \epsilon \end{array} \right] \right) = \left[\begin{array}{cccc} \epsilon & 3\alpha & 2\beta & \epsilon \\ 2\alpha & \epsilon & \epsilon & \epsilon \\ \epsilon & 2\kappa & \epsilon & 2\alpha \\ e\alpha & \epsilon & \epsilon & \epsilon \end{array} \right] \triangleright \mathbf{E}.
 \end{aligned}$$

Então, faz-se

$$\mathbf{St}^1 = \mathbf{Et} = \left[\begin{array}{cccc} \epsilon & 3\alpha & 2\beta & \epsilon \\ 2\alpha & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 2\alpha \\ e\alpha & \epsilon & \epsilon & \epsilon \end{array} \right]$$

e calcula-se

$$\mathbf{Bt}_{uc}^2 = \begin{bmatrix} \epsilon & 4\beta\kappa & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}$$

com $\theta(\mathbf{Bt}_{uc}^2) = \theta(\mathbf{Et})$ e $\phi(\mathbf{Bt}_{uc}^2) = \phi(\mathbf{Et})$ (ver equação 4.43). Assim, para eliminar o evento κ que não se encontra em \mathbf{Et} , deve-se inibir o evento β nos estados 1. Com isso, encontra-se

$$\mathbf{St} = \text{COACES}(\mathbf{St}^2) = \text{COACES} \left(\begin{bmatrix} \epsilon & 3\alpha & 2\beta & \epsilon \\ 2\alpha & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 2\alpha \\ e\alpha & \epsilon & \epsilon & \epsilon \end{bmatrix} \right) = \begin{bmatrix} \epsilon & 3\alpha & \epsilon & \epsilon \\ 2\alpha & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 2\alpha \\ e\alpha & \epsilon & \epsilon & \epsilon \end{bmatrix}$$

e

$$\mathbf{St} = \text{ACES}(\mathbf{St}^2) = \text{ACES} \left(\begin{bmatrix} \epsilon & 3\alpha & \epsilon & \epsilon \\ 2\alpha & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 2\alpha \\ e\alpha & \epsilon & \epsilon & \epsilon \end{bmatrix} \right) = \begin{bmatrix} \epsilon & 3\alpha & \epsilon & \epsilon \\ 2\alpha & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}$$

com

$$\theta(\mathbf{St}) = \begin{bmatrix} e & \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{St}) = \begin{bmatrix} e \\ \epsilon \\ \epsilon \\ \epsilon \end{bmatrix}$$

que é o supervisor para a especificação dada, mostrado na Figura 4.24(d).

Exemplo 4.23 Considere o Exemplo 4.15, que é uma extensão do exemplo clássico do sistema das duas máquinas e um buffer. Sendo $\Sigma = \{\alpha_1, \alpha_2, \beta_1, \beta_2\}$ e $\Sigma_{uc} = \{\beta_1, \beta_2\}$, para a especificação expandida $\mathbf{Et}^\#$ em forma de submatriz de $\mathbf{At}^\#$ (vista na Figura 4.25), encontra-

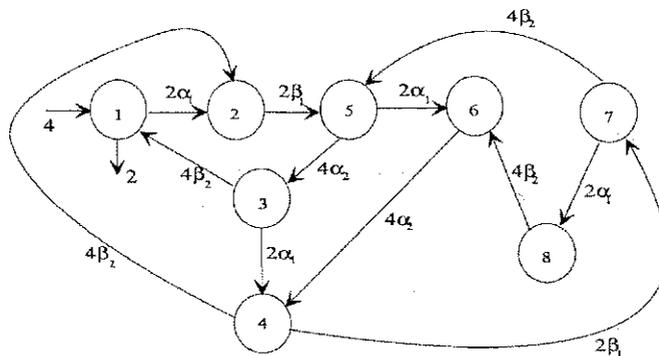


Figure 4.25: Especificação temporizada para o exemplo das duas máquinas e um buffer.

se $\text{ACES}((\mathbf{Et}^\# \oplus [\Sigma_{uc}]) \otimes \mathbf{At}^\#) \triangleright \mathbf{Et}^\#$, em que os elementos $et_{8,9}^\#$ e $et_{8,11}^\#$ não pertencem

que é o supervisor apresentado na Figura 4.26. Este supervisor garante que os eventos sejam atrasados para só ocorrerem após os tempos determinados, definindo que a linguagem do sistema supervisionado seja sua própria linguagem temporizada, isto é $L(\text{St}||\text{At}^\#) = L(\text{St})$.

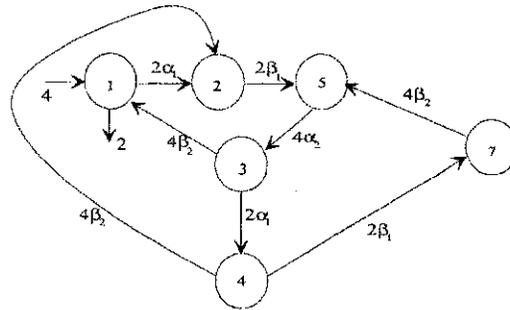


Figure 4.26: Supervisor.

4.6 Complexidade algorítmica e considerações sobre o algoritmo proposto

Considerando que o algoritmo proposto para a síntese de supervisores de SEDs temporizados apresenta um procedimento semelhante ao algoritmo da síntese de SEDs não temporizados, o estudo de complexidade mostra-se igual. Isto é, $O(N^4)$, mesmo com a utilização dos operadores definidos especificamente para este caso. A introdução dessas redefinições dos operadores para o caso temporizado, apenas aumenta algumas operações para o cálculo da função de complexidade de tempo, mas não interfere na ordem de complexidade assintótica.

Deve-se observar que esse algoritmo proposto para a síntese do supervisor de SEDs temporizados, é uma extensão do algoritmo proposto no Capítulo 3, para a síntese de supervisores não temporizados, utilizando os tempos de vida dos eventos como $t_\sigma \in \mathbb{R}_{\max}$, enquanto o caso não temporizado utiliza $t_\sigma \in \mathbb{B}$.

O supervisor construído com este algoritmo proposto satisfaz as condições de Brandin e Wonham [BW94], considerando que os limites superiores dos intervalos de tempos de vida dos eventos são sempre iguais a infinito. Também, a utilização das séries formais garante que o supervisor construído determine que o SED reconheça uma linguagem que pode ser representada por uma equação em termos dos seus datadores, como em Gaubert [Gau93, Gau95a].

Para o caso de especificações de comportamento definidas com tempos de vida de eventos ou atrasos inicial ou finais que se apresentam como accleração da execução do sistema, o algoritmo também prevê sua transformação em uma especificação válida, tornando esses valores iguais aos dos tempos de vida dos eventos do sistema.

Chapter 5

Conclusões e perspectivas futuras

A abordagem proposta nesse trabalho é uma alternativa para a construção de supervisores para SEDs temporizados e não temporizados de uma forma unificada baseada em um formalismo matricial. Nessa abordagem o modelo de SEDs temporizados e não temporizados e sua respectiva especificação de comportamento são definidos através de matrizes de incidência. Os elementos da matriz de incidência são símbolos de um alfabeto que representam os eventos do sistema. De maneira semelhante à TCS, os eventos do sistema são classificados como eventos controláveis e eventos não controláveis.

Foi demonstrado que a síntese do supervisor nessa abordagem pode ser realizada através de operações matriciais efetuadas com as matrizes que representam o SED e sua especificação de comportamento. Todas as operações matriciais nessa abordagem são definidas na álgebra de dióides.

Desenvolveu-se um procedimento que inclui os testes de validade da especificação de comportamento, controlabilidade, acessibilidade e coacessibilidade, além da verificação de submatrizes do modelo do SED, síntese e construção da composição síncrona do sistema supervisionado. A formulação desse procedimento exigiu a definição de novos operadores para a álgebra de dióides que se aplicam especificamente neste contexto. Assim, o formalismo apresentado determina também uma formulação única para a solução dos problemas especificados, utilizando para isto dos algoritmos desenvolvidos para realizar todas as operações na forma matricial. Com essa formalização, os algoritmos podem ser implementados através de *softwares* especificamente desenvolvidos para trabalhar com matrizes. Além do mais, a complexidade computacional do procedimento desenvolvido apresentou uma ordem de $O(N^4)$, tanto para o caso não temporizado, como para o caso temporizado, onde N é a dimensão das matrizes de incidência, ou o número de estados dos autômatos.

Considerando a ordem de complexidade do procedimento, o algoritmo mostra-se como uma importante alternativa na síntese de supervisores de SEDs temporizados, quando se consideram todos os tempos de vida dos eventos definidos como os tempos mínimos para suas habilitações. Por outro lado, considerando que a especificação de comportamento seja

definida como uma submatriz do modelo do sistema, a ordem de complexidade do procedimento desenvolvido é reduzida para $O(N^2)$, que se compara à complexidade do algoritmo clássico da construção do supervisor para SEDs não temporizados de Ramadge e Wonham [RW87].

A apresentação dos exemplos ilustra a eficácia da abordagem. Assim, ficou comprovado que a formalização é garantida para SEDs modelados por matrizes de incidência.

O procedimento proposto se aplica a SEDs determinísticos não temporizados ou temporizados que apresentam um número finito de estados. Também, se aplica a SEDs cíclicos ou não cíclicos. Para o caso dos SEDs temporizados, deve-se levar em consideração que os tempos de vida dos eventos são definidos como os tempos mínimos para suas ocorrências (limites de tempos superiores iguais a infinito), definidos em Brandin e Wohnam [BW94] como eventos *remotos*.

A formalização alternativa apresentada nesse trabalho para a síntese de supervisores de SEDs temporizados e não temporizados, define uma perspectiva de síntese de supervisores de sistemas a eventos discretos com lógica temporizada. Contudo, como não sendo um trabalho de todo auto-contido, necessita de sua ampliação em vários aspectos, como os pontos citados a seguir:

- Definir as condições para incluir a representação dos eventos *prospectivos* [BW94], que apresentam os tempos de vida dos eventos definidos por intervalos de tempo com limite superior finito;
- Construir supervisores para SEDs temporizados, considerando os eventos *prospectivos*;
- Utilizar essa abordagem para avaliação de desempenho;
- Definir as condições para aplicação dessa abordagem aos SEDs não determinísticos, desde que a teoria dos dióides não determina nenhuma restrição para este caso;
- Determinar as condições para construir supervisores para SEDs com observação parcial de eventos tanto temporizados, como não temporizados;
- Utilizar esta abordagem para síntese modular;
- Desenvolver o formalismo para controle hierárquico;
- Aplicar a estrutura da formalização proposta aos grafos de eventos temporizados.

Bibliography

- [ABG98] M. Akian, R. Bapat, and S. Gaubert. Asymptotics of the perron eigenvalue and eigenvector using max-algebra. Technical report, Institut National de Recherche en Informatique et en Automatique, 1998.
- [ABH01] C. Jard A. Benveniste, E. Fabre and S. Haar. Diagnosis of asynchronous discrete event systems, a net unfolding approach. Technical Report 4181, Institut National de Recherche en Informatique et en Automatique - INRIA, Mai 2001.
- [ACD93] R. Alur, C. Courcoubetis, and D. Dill. Model checking in dense real-time. *Information and Computation*, 104(1):2–24, 1993.
- [AD90] R. Alur and D. Dill. Automata for modeling real-time systems. *Proc. 17th International Colloquim on Automata, Languages and Programming, Lectures Notes on Computer Science, New York: Springer Verlag*, 443:322–335, 1990.
- [AD94] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, (126):183–235, 1994.
- [AD95] R. Alur and D. Dill. Automata-theoretic verification of real-time systems. Technical report, Computing Science Research (Bell Labs) and Computer Science Department (Stanford University), 1995.
- [AH92] R. Alur and T.A. Henzinger. Back to the future: Towards a theory of timed regular languages. *IEEE 0-8186-2900-2/92*, pages 177–186, 1992.
- [Alu97] R. Alur. Timed automata. *Proc. NATO-ASI Summer School, Antalya, Turkey*, 1997.
- [AM00] S.M. Aji and R.J. McEliece. The generalized distributive law. *IEEE Transactions on Information Theory*, 46(2):325–343, March 2000.
- [Asa98] E. Asarin. Equations on timed languages. *Proc. of Hybrid'98*, 1998.
- [ATP00] R. Alur, S. La Torre, and G.J. Pappas. *Optimal Paths in Weighted Timed Automata*. University of Pennsylvania, Bell Labs and Università degli Studi di Salerno, 2000.

- [Bal96] F. Balarin. Approximate reachability analysis of timed automata. *IEEE - 1052-8725/96*, pages 52–61, 1996.
- [BCOQ92] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. *Synchronisation and Linearity. An Algebra for Discrete Event Systems*. John Wiley Sons, 1992.
- [Bea96] J.A. Beachy. *Abstract Algebra II*. Waveland Press, Inc., 1996.
- [BH88] Y. Brave and M. Heymann. Formulation and control of real time discrete event processes. *Proc. 27th. IEEE Conf. Decision and Control, New York*, pages 1131–1132, 1988.
- [BJG95] A. Benveniste, C. Jard, and S. Gaubert. Monotone rational series and max-plus algebraic models of real-time systems. *Proceedings of 4th Workshop on Discrete Event Systems, WODES95, Cagliari, Italy, 1995*.
- [BJG98] A. Benveniste, C. Jard, and S. Gaubert. Algebraic techniques for timed systems. 1998. Esprit LTR-SYRF project.
- [Boi99] J.L. Boimond. Sur l'étude des systèmes à événements discrets dans l'algèbre des diodes: Identification, commande des graphes d'événements temporisés, représentation des graphes d'événements temporisés à paramètres variables. Technical report, l'Université d'Angers, 1999.
- [Bou98] P. Bouyer. Automates temporisés et modularité. Mémoire de DEA, 1998.
- [BP99] P. Bouyer and A. Petit. Decomposition of timed automata. *Proc. 26th Int. Coll. Automata, Languages, and Programming*, 1644 of Lectures Notes in Computer Science:210–219, 1999.
- [BR88] J. Berstel and C. Reutenauer. *Rational Series and their Languages*. Springer, 1988.
- [Bra92] J.C. Bradfield. *Verifying Temporal Properties of Systems*. Progress in Theoretical Computer Science. Birkhäuser, 1992.
- [BW94] B.A. Brandin and W.M. Wonham. Supervisory control of timed discrete-event systems. *IEEE Transactions on Automatic Control*, 39(2):329–342, 1994.
- [BWB92] B.A. Brandin, W.M. Wonham, and B. Benhabib. Manufacturing cell supervisory control - a timed discrete event system approach. In *Proceedings of the 1992 IEEE International Conf. On Robotics and Automation*, pages 931–936, May 1992.
- [CCS91] P. Cofrancesco, A. Cristoforetti, and R. Scattolini. Petri nets based approach to software development for real-time control. *IEE Proceedings-D Control Theory and Applications*, 138(5):474–478, 1991.

- [CDQV85] G. Cohen, D. Dubois, J.P. Quadrat, and M. Viot. A linear system theoretic view of discrete event process and its use for performance evaluation in manufacturing. *IEEE Transactions on Automatic Control*, 30(3):210–220, 1985.
- [CG94] D.D. Cofer and V.K. Garg. Supervisory control of timed event graphs. *IEEE - 0-7803-2129-4/94*, 1994.
- [CG95] D.D. Cofer and V.K. Garg. Control of event separation times in discrete event systems. *Research supported in part by NSF grant CCR-9110605, a General Motors Fellowship, and an IBM grant*, 1995.
- [CG96] D.D. Cofer and V.K. Garg. Supervisory control of real time discrete-event systems using lattice theory. *IEEE Transactions on Automatic Control*, 41(2):199–209, 1996.
- [CGQ95] G. Cohen, S. Gaubert, and J.P. Quadrat. Algebraic system analysis of timed petri nets. In Cambridge University Press J. Gunawardena Ed., editor, *Idempotency - Collection of the Isaac Newton Institute*, 1995.
- [CH90] X.R. Cao and Y.C. Ho. Models of discrete event dynamic systems. *IEEE Control System Magazine*, 10(4):69–76, 1990.
- [CHBF99] B. Cottenceau, L. Hardouin, J.L. Boimond, and J.L. Ferrier. Synthesis of greatest linear feedback for timed-event graphs in dioid. *IEEE Transactions of Automatic Control*, 44(6):1258–1262, June 1999.
- [Chr83] P. Chrétienne. *Les Réseaux de Petri Temporisés*. Thèse d'état. Université Paris VI, Paris, 1983.
- [CJ99] H. Comon and Y. Jurski. Timed automata and the theory of real numbers. *Proc. 10th Int. Conf. Concurrency Theory*, 1664 of Lectures in Computer Science:242–257, 1999.
- [CL99] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.
- [CL01a] E.M.M. Costa and A.M.N. Lima. Síntese de supervisores de sistemas a eventos discretos utilizando dióides. *Anais do V Simpósio Brasileiro de Automação Inteligente (SBAI)*, 2001.
- [CL01b] E.M.M. Costa and A.M.N. Lima. Síntese de supervisores de sistemas a eventos discretos temporizados. *Anais do V Simpósio Brasileiro de Automação Inteligente (SBAI)*, 2001.

- [CM94] R.E. Campello and R. Maculan. *Algoritmos e Heurísticas, Desenvolvimento e Avaliação de Performance*. Editora da Universidade Federal Fluminense - RJ, 1994.
- [CMQV89] G. Cohen, P. Moller, J.P. Quadrat, and M. Viot. Algebraic tools for the performance evaluation of discrete event systems. *Proceedings of the IEEE*, 77(1):39-56, january 1989.
- [Cos00] E.M.M. Costa. *A Abordagem da Síntese de Supervisores de Sistemas a Eventos Discretos Através dos Dióides*. Relatório de Projeto e Pesquisa. Universidade Federal da Paraíba - Campus II, Campina Grande, PB, Brasil, Agosto, 2000.
- [Cot99] B. Cottenceau. *Contribution à la Commande de Systèmes à Événements Discrets: Synthèse de Correcteurs pour les Graphes d'Événements Temporisés dans les Dioïdes*. PhD thesis, l'Université d'Angers, 1999.
- [CR90] C.G. Cassandras and P.J.G. Ramadge. Toward a control theory for discrete event systems. *IEEE Control Systems Magazine*, pages 66-68, June 1990.
- [CTC01] J.E.R. Cury, C.R.C. Torrico, and A.E.C. Da Cunha. A new approach for supervisory control of discrete event systems. 2001.
- [DHP+93] F. DiCesare, G. Harhalakis, J.M. Proth, M. Silva, and F.B. Vernadat. *Practice of Petri Nets in Manufacturing*. Chapman and Hall, 1993.
- [EAM97] P. Caspi E. Asarin and O. Maler. A kleene theorem for timed automata. *Proc. of LICS'97*, pages 160-171, 1997.
- [EK82] E. Emre and P.P. Khargonekar. Regulation of split linear systems over rings: Coefficient assignment and observers. *IEEE Transactions on Automatic Control*, AC-27(1):104-113, 1982.
- [Fre91] P. Freedman. Time, petri nets, and robotics. *IEEE Transactions on Automatic Control*, 7(4):417-433, 1991.
- [Fri98] L. Fribourg. *A Closed-Form Evaluation for Extended Timed Automata*. Research Report LSV-98-2, Laboratoire Spécification et Vérification, Ecole Normale Supérieure de Cachan, France, 1998.
- [GAC+90] S. Gaubert, M. Akian, G. Cohen, R. Nikoukhah, and J.P. Quadrat. Linear systems in $(\max,+)$ algebra. In *Proc. Of the 29th Conference on Decision and Control*, 1990.

- [GAC⁺97] S. Gaubert, M. Akian, G. Cohen, J.P. Quadrat, and M. Viot. *Methods and Applications of (max,+) Linear Algebra*. Institut National de Recherche en Informatique et en Automatique, No. 3088, Janvier, 1997.
- [Gau92] S. Gaubert. *Théorie des Systèmes Linéaires dans les Dioïdes*. PhD thesis, École Nationale Supérieure des Mines de Paris, 1992.
- [Gau93] S. Gaubert. Performance evaluation of timed automata. Technical report, Institut National de Recherche en Informatique et en Automatique, 1993.
- [Gau94a] S. Gaubert. On rational series in one variable over certain dioids. Technical report, Institut National de Recherche en Informatique et en Automatique, 1994.
- [Gau94b] S. Gaubert. Rational series over dioids and discrete event systems. *Proc. of the 11th Int. Conf. on Analysis and Optimization of Systems: Discrete Event Systems, Sophia Antipolis, Lectures Notes in Control and Information Sciences 199, Springer, 1994*.
- [Gau94c] S. Gaubert. Rational series over dioids and discrete event systems. *Proceedings of the 11th Int. Conf. on Analysis and Optimization of Systems: Discrete Event Systems, Sophia Antipolis, Lectures Notes in Control and Information Sciences, Springer, (199), June 1994*.
- [Gau95a] S. Gaubert. Performance evaluation of (max,+) automata. *IEEE Transactions on Automatic Control*, 40(12):2014–2025, December 1995.
- [Gau95b] S. Gaubert. Resource optimization and (min,+) spectral theory. *IEEE Transactions on Automatic Control*, 40(11):1931–1934, November 1995.
- [Gau96] S. Gaubert. On the burnside problem for semigroups of matrices in the (max,+) algebra. *Semigroup Forum, Springer-Verlag New York Inc.*, 52:271–292, 1996.
- [Gau99] S. Gaubert. Introduction aux systèmes dynamiques à Événements discrets. Notes de Cours, 1999.
- [GG98] S. Gaubert and J. Gunawardena. The duality theorem for min-max function. *C.R.A.S.*, (t. 326, Serie I):43–48, 1998.
- [Gil62] A. Gill. *Introduction to the Theory of Finite-State Machines*. McGraw-Hill Electronic Sciences Series. McGRAW-HILL Book Company, 1962.
- [GM95] S. Gaubert and J. Mairesse. Task resource models and (max,+) automata. In J. Gunawardena Ed., editor, *Idempotency - Collection of the Isaac Newton Institute*, 1995.

- [GM97] S. Gaubert and J. Mairesse. Modeling and analysis of timed petri nets using heaps of pieces. *IEEE Transactions on Automatic Control*, 44(4):683–697, April 1997.
- [Gre79] R.C. Green. *Minimax Algebra*. Lectures Notes in Economics and Mathematical Systems. Springer-Verlag, 1979.
- [GTC⁺98] S. Gaubert, J.C. Terrasson, G. Cohen, M. Mc Gettrick, and J.P. Quadrat. Numerical computation of spectral elements in max-plus algebra. Technical report, European Community Framework, 1998.
- [HML00] O. Boivineau H. Marchand and S. Lafortune. Optimal control of discrete event systems under partial observation. Technical Report 4046, Institut National de Recherche en Informatique et en Automatique - INRIA, Novembre 2000.
- [Hoa85] C.A.R. Hoare. *Communicating Sequential Process*. Prentice-Hall, London,UK, 1985.
- [HU69] J.E. Hopcroft and J.D. Ullman. *Formal Languages and Their Relation to Automata*. Addison-Wesley Publishing Company, 1969.
- [HU79] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, USA, 1979.
- [Huu98] R. Huuck. *Transformation Von Echtzeit-Bedingungs-/Ereignissystemen in Echtzeitautomaten: Ein Weg Zur Automatischen Verifikation*. PhD thesis, Universität zu Kiel, Juni 1998.
- [JM94] F. Jahanian and A.K. Mok. Modechart: A specification language for real-time systems. *IEEE Transactions of Software Engineering*, 20(12):933–947, December 1994.
- [KG91] R. Kumar and V.K. Garg. Optimal control of discrete event dynamical systems using network flow techniques. *Proceedings of 1991 Annual Allerton Conference*, pages 705–714, October 1991.
- [Kli99] I. Klimann. *Langages, Séries et Contrôle de Trajectoires*. PhD thesis, l'Université Denis Diderot - Paris 7, Juin 1999.
- [Koz91] P. Kozák. *Supervisory control of discrete event processes: a real-time extensions*. Inst. of Information Theory and Automation, Czechoslovak Acad. of Sciences, Prague, March, 1991.
- [Kro87] L. Kronsjö. *Algorithms, their complexity and efficiency*. Jonh Willey and Sons, 2nd edition, 1987.

- [Kuc91] V. Kucera. *Analysis and Design of Discrete Linear Control Systems*. Academia, 1991.
- [Law97] M.S. Lawford. *Model Reduction of Discrete Real-Time Systems*. PhD thesis, University of Toronto, 1997.
- [Lea00] G.V. Leandro. *Síntese de Controladores para Transitórios em Escalonamento de Job Shop Cíclico*. Tese de Doutorado. UNICAMP, Campinas, SP, Brasil, Agosto 2000.
- [Lib96] L. Libeaut. *Sur l'utilisation des Dioïdes pour la Commande des Systèmes a Événements Discrets*. PhD thesis, École Doctorale Sciences pour L'Ingénieur de Nantes, 1996.
- [MB97] H. Marchand and M. Le Borgne. Partial order control and optimal control of discrete event systems modeled as polynomial dynamical systems over galois fields. Technical Report 3273, Institut National De Recherche en Informatique et en Automatique - INRIA, Octobre 1997.
- [Men97] E. Menguy. *Contribution à la Commande des Systèmes Linéaires dans les Dioïdes*. Thèse de Doctorat, École Doctorale Sciences Pour L'Ingénieur de Nantes, 1997.
- [Mil80] R. Milner. *A Calculus of Communicating Systems*. Springer Verlag, New York, USA, 1980.
- [Mur89] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.
- [Nor99] A.B. Noronha. *Controle de Transitórios em Escalonamento de Job Shop Cíclico*. Tese de Doutorado. UNICAMP, Campinas, SP, Brasil, Outubro 1999.
- [OHY+99] K. Okano, S. Hattori, A. Yamamoto, T. Higashino, and K. Tanigushi. Specification of real-time systems using a timed automata model with shared variables and verification of partial-deadlock freeness. *IEEE - 0-7695-0358-5/99*, pages 576–581, 1999.
- [OW90a] J.S. Ostroff and W.M. Wonham. A framework for real-time discrete event control. *IEEE Transactions on Automatic Control*, 35(4):386–397, 1990.
- [OW90b] J.S. Ostroff and W.M. Wonham. A framework for real-time discrete event control. *IEEE Transactions on Automatic Control*, 35(4):386–397, April 1990.
- [PC97] Y. Park and E.K.P. Chong. On d-inversion in interruptive timed discrete-event systems. *IEEE Transactions on Automatic Control*, 42(11):1550–1554, 1997.

- [Pu00] K.Q. Pu. Modeling and control of discrete event systems with hierarchical abstraction. Master's thesis, University of Toronto, 2000.
- [Rei85] W. Reisig. *Petri Nets: an Introduction*. Springer-Verlag Berlin Heidelberg, 1985.
- [Rei92] W. Reisig. *A Primer in Petri Net Design*. Springer-Verlag, 1992.
- [RW82] P.J.G. Ramadge and W.M. Wonham. Supervision of discrete event processes. *Proceedings of 21st Conference on Decision and Control*, pages 1228–1229, 1982.
- [RW87] P.J.G. Ramadge and W.M. Wonham. On the supremal controllable sublanguage of a given language. *SIAM Journal of Control and Optimization*, 25(3):637–659, May 1987.
- [RW89] P.J.G. Ramadge and W.M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, 1989.
- [Sal73] A. Salomaa. *Formal Languages*. ACM Monograph Series. Academic Press, Inc. New York, 1973.
- [Sme89] R. Smedinga. *Control of Discrete Events*. PhD thesis, Rijksuniversiteit Groningen, 20 januari 1989.
- [Sme93] R. Smedinga. Discrete event systems. Technical report, University of Groningen, October 1993.
- [TB99] S. Tasiran and R.K. Brayton. On iterative verification with timed automata. Technical report, Departament of Electrical Engineering and Computer Sciences, University of California, Berkeley, 1999.
- [TGG97] J.C. Terrasson, S. Gaubert, and J. Gunawardena. Dynamics of min-max functions. Technical report, HP Laboratories Technical Report, 1997.
- [Thi96] J.G. Thistle. Supervisory control of discrete event systems. *Mathl. Comput. Modeling*, 23(11/12):25–53, 1996.
- [TKB99] S. Tasiran, Y. Kukimoto, and R.K. Brayton. Computing delay with coupling using timed automata. Technical report, Departament of Electrical Engineering and Computer Sciences, University of California, Berkeley, 1999.
- [Tor99] C.R.C. Torrico. Implementação de controle supervisorio de sistemas a eventos discretos aplicado a processos de manufatura. Technical report, Universidade de Santa Catarina, Santa Catarina, Brasil, 15 de Março 1999.
- [Tri98] S. Tripakis. *L'Analyse Formelle Des Systèmes Temporisés En Pratique*. PhD thesis, L'Université Joseph Fourier, Décembre 1998.

- [Vel79] P.A.S. Veloso. *Máquinas e Linguagens. Uma Introdução à Teoria de Autômatos*. Escola de Computação Instituto de Matemática e Estatística da USP - São Paulo, 1979.
- [Vid85] M. Vidyasagar. *Control Systems Synthesis - a Factorization Approach*. MIT Press, 1985.
- [Wan92] F-Y. Wang. Supervisory control for concurrent discrete event dynamic systems based on petri nets (extended abstract). *Proc. of the 31st Conference on Decision and Control*, pages 1196–1197, 1992.
- [Won99] W.M. Wonham. Sed notes. Course notes, 1999.
- [ZD93] M.C. Zhou and F. DiCesare. *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Kluwer Academic Publishers, 1993.
- [Zha01] Z. Zhang. Smart TCT: An efficient algorithm for supervisory control design. Master's thesis, University of Toronto, June 2001.
- [Zil93] R.M. Ziller. *A Abordagem Ramadge-Wonham no Controle de Sistemas a Eventos Discretos: Contribuições à Teoria*. Dissertação de Mestrado. Universidade Federal de Santa Catarina, Santa Catarina, Brasil, 1993.