

RELATÓRIO - BOLSA DCR
SÍNTESE DE SUPERVISORES DE SISTEMAS A
EVENTOS DISCRETOS TEMPORIZADOS

Eduard Montgomery Meira Costa
Universidade Federal da Bahia
Escola Politécnica
Departamento de Engenharia Elétrica
Processo 301265/01-1

Agosto de 2003

Conteúdo

I	Desenvolvimento Teórico:	
	Autômatos com Temporização Variável e Síntese de Supervisores	
	Temporalmente Adaptativos para Sistemas a Eventos Discretos	1
1	Introdução	3
1.1	Autômatos e Modelos de SEDs	5
2	Preliminares	7
2.1	Álgebra de Dióides	7
2.1.1	Álgebra $(\max, +)$	7
2.2	Séries Formais	8
2.3	Representação Matricial de Autômatos	10
2.3.1	Linguagem associada a uma matriz de incidência	11
2.3.2	Matrizes de incidência temporizadas	13
3	Autômatos com Temporização Variável	19
3.1	Representação Matricial	22
3.1.1	Linguagens	23
3.2	Exemplos	25
4	Aplicação dos ATVs aos SEDTs	29
4.1	Acessibilidade e Coaccessibilidade	29
4.2	Composição Síncrona	30
4.3	Síntese do supervisor	32
4.4	Supervisor Temporalmente Adaptativo	42
4.5	Exemplo	43
5	Conclusões	47
II	Produção Científica	53
III	Formação de Recursos Humanos	59

Lista de Figuras

2.1	Autômato não determinístico com mais de uma função de transição entre dois estados.	11
2.2	Autômato para exemplificar a construção da matriz de caminhos.	12
2.3	Autômato para determinação da linguagem.	13
2.4	Autômato $(\max, +)$ determinístico para ilustrar a representação por matriz de incidência temporizada e vetores de estados inicial e marcados.	15
2.5	Autômato para exemplo da matriz de caminhos temporizada.	16
2.6	Autômato temporizado para determinação da linguagem temporizada.	17
3.1	Exemplo de um <i>ATV</i> .	20
3.2	Autômato com Temporização Variável	21
3.3	Autômato para exemplo da matriz de caminhos	24
3.4	(a) Autômato $(\max, +)$ infinito, (b) particionamento do autômato $(\max, +)$ e (c) <i>ATV</i> como modelo compacto do autômato $(\max, +)$.	27
3.5	Sistema.	28
3.6	(a) Sistema modelado por um <i>ATV</i> e (b) sistema modelado por um autômato $(\max, +)$.	28
4.1	Diagrama de um SED temporizado supervisionado	33
4.2	Autômatos (a) <i>ATV</i> e (b) <i>ATV'</i> construído com o Algoritmo 1.	35
4.3	Autômato para exemplificar a síntese do supervisor	39
4.4	Especificação controlável	39
4.5	(a) Especificação não controlável e (b) Supervisor	41
4.6	Gráfico das funções temporais do evento β que aponta do estado 1 para o estado 2. O máximo entre as duas funções para a composição síncrona do supervisor com o <i>ATV</i> é definido pela função temporal da especificação.	41
4.7	Furadeira.	44
4.8	Modelo em autômato $(\max, +)$ da furadeira.	44
4.9	Modelo em <i>ATV</i> da especificação de comportamento.	45

Parte I

Desenvolvimento Teórico: Autômatos com Temporização Variável e Síntese de Supervisores Temporalmente Adaptativos para Sistemas a Eventos Discretos

Capítulo 1

Introdução

Os Sistemas a Eventos Discretos (SEDs) [1, 2, 3, 4, 5, 6] 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.

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.

O estudo dos SEDs levou à fundamentação da teoria denominada Teoria de Controle Supervisório (TCS) [7, 2], que é uma forma elegante de controlar estes sistemas. A TCS foi desenvolvida utilizando como paradigma de modelagem dos SEDs as Linguagens Formais e os Autômatos [8], mas é totalmente independente do paradigma de modelagem. Esta teoria define através do modelo do SED e de uma especificação de comportamento uma entidade de controle denominada supervisor, que avalia os eventos, ou transições de estado, gerados pelo sistema e gera uma ação de controle a ser aplicada ao sistema em malha fechada, de forma ao SED seguir ao máximo a especificação de comportamento dada.

Com o avanço dos estudos dos SEDs, tem-se formalizado ferramentas para o controle destes sistemas considerando o tempo. Dentre estes estudos, encontra-se a abordagem de Brandin e Wonham [9], em que a temporização do SED é feita através da inclusão de um evento denominado '*tick*' que ocorre sob a dependência da descida de um relógio global. Devido à inclusão deste evento, o sistema apresenta um aumento considerável no número de estados. A síntese do supervisor é feita semelhantemente à abordagem de Ramadge e Wonham [10]. Esta abordagem é limitada a estudos de sistemas que não apresentam concorrência e sincronismo, necessitando de uma especificação dada em termos de uma linguagem incluindo o evento '*tick*'.

Quando o sistema apresenta a necessidade de sincronismo, a álgebra de dióides [11] se apresenta como uma ferramenta matemática fundamental para a descrição da evolução dinâmica do SED, em termos dos tempos de ocorrência dos eventos, permitindo assim, uma descrição linear [12, 13] semelhante à Teoria de Controle Clássica. Nesta forma de abordar o problema de controle de SEDs temporizados, há vários trabalhos desenvolvidos, utilizando como paradigma de modelagem os autômatos temporizados [14] ou os grafos de eventos temporizados, que são redes de Petri temporizadas [15] que apresentam para cada lugar apenas uma transição de entrada e uma transição

de saída. Atualmente, a aplicação da teoria dos dióides aos SEDs é freqüentemente feita tendo como paradigma de modelagem os grafos de eventos temporizados. Contudo, toda a formulação matemática da álgebra de dióides também é válida para os autômatos temporizados [16], embora neste caso, não seja possível modelar concorrência.

Na abordagem do controle de SEDs temporizados, utilizando a álgebra de dióides [17, 18, 19, 20, 21], a especificação de comportamento é fundamentada por um vetor de funções de atraso para os eventos, definindo um supervisor que avalia a saída do sistema e aplica inibições temporárias aos eventos, determinando os tempos de suas ocorrências [22, 23, 16]. Também na visão dos dióides, as séries formais [24, 25, 26] são utilizadas na avaliação de desempenho e descrição da linguagem reconhecida de SEDs modelados por autômatos $(\max, +)$ [27, 28], que é uma classe dos autômatos temporizados. Assim, os autômatos $(\max, +)$ se apresentam, em conjunto com as séries formais, como um importante paradigma para o estudo da síntese de supervisores para SEDs temporizados, em que as séries formais podem descrever a linguagem reconhecida do autômato, seja ela temporizada ou não temporizada, além de poder ser utilizada na avaliação de desempenho dessa classe de SEDs. Os autômatos $(\max, +)$ podem ser utilizados como paradigma de modelagem para os SEDs temporizados, inclusive se este for não determinístico.

Dentre estes paradigmas, tanto os grafos de eventos temporizados como os autômatos temporizados modelam SEDs, para os quais o conceito de tempo nos eventos é definido por um valor constante, ou através de um intervalo de tempo. Assim, até o presente momento, não existe um formalismo para o estudo de SEDs que apresentam eventos com tempos de vida variáveis. Nesse contexto, o conceito de funções temporais aplicado aos autômatos $(\max, +)$ é uma nova alternativa na formalização de uma ferramenta para a modelagem e controle de SEDs temporizados, em que a álgebra de dióides pode ser avaliada para sua descrição. A descrição de um SED temporizado através da álgebra de dióides permite descrever sua dinâmica por meio de um sistema de equações linear semelhante ao da Teoria de Controle Clássica, o que implica na determinação de avaliação de desempenho e, conseqüentemente, busca de controle ótimo. As séries formais, por outro lado, permitem descrever as linguagens temporizadas do sistema utilizando as matrizes de incidência do modelo, seja ele um grafo de eventos temporizados ou um autômato $(\max, +)$.

A avaliação de uma ferramenta utilizando a álgebra de dióides para a síntese de supervisores de SEDs temporizados, unindo a fundamentação lógica de Ramadge e Wonham [2] com o modelo definido por um autômato $(\max, +)$, foi baseada em [29], em que esta ferramenta mostra-se como alternativa para o algoritmo da construção da suprema sub-linguagem controlável ($SupC(L)$) [10] de SEDs não temporizados [30] e temporizados [31], como visto em [32]. Este formalismo é aplicado a SEDTs determinísticos e com número finito de estados. Sendo assim, a aplicação da álgebra de dióides revela-se como fundamental no estudo do controle de SEDs temporizados.

Na maioria dos casos, os SEDs temporizados estudados até o presente momento restringem o modelo pela temporização constante, ou com limites inferior/superior de tempo sobre a ocorrência de um evento, em que para um dado estado q , um evento ocorre levando um determinado tempo t constante para levar o sistema ao estado q' . Sempre que o sistema volta a este estado q , o mesmo evento pode ocorrer, gastando o mesmo tempo t para levar o sistema novamente ao estado q' . Por outro lado, em [33], é proposta uma abordagem para a qual há variação nas entradas do sistema de forma que os tempos de vida dos eventos não são constantes, porém são avaliados a intervalos constantes. Considerando que os intervalos não sejam constantes, isto é, o tempo de vida de um determinado evento dependa do tempo de vida de um outro evento anterior, alguns tipos de sistema não são contemplados. Esses tipos de sistemas podem ser caracterizados por um determinado processo para o qual, em um determinado estado q , um evento ocorre segundo uma variação temporal descrita por uma função $f(t)$ contínua. Assim, sempre que o sistema se encontra neste estado q , o tempo de vida deste evento pode ser diferente do tempo de vida da ocorrência

anterior, sendo dado pelo valor da função neste instante de tempo imediato, o qual depende dos acontecimentos anteriores à sua ocorrência.

Um paradigma que representa estes tipos de sistemas é denominado de Autômato com TempORIZAÇÃO Variável (ATV), e foi introduzido em [34]. Este modelo mostrou-se de fundamental importância no estudo dos SEDs em conjunto com a abordagem de [35] que utiliza a álgebra de dióides e as matrizes de incidência na formulação do controle destes sistemas, como visto em [36, 37], desde que na maioria dos casos, o modelo do sistema é construído sem considerar tais variações, embora elas aconteçam na prática.

Quando se consideram variações temporais nos tempos de vida dos eventos, para os casos tratados com os tempos de vida dos eventos constantes, na prática implica conceber uma especificação para cada variação, ou mesmo uma nova formulação do modelo incluindo novos estados e transições. Dessa forma, a utilização do ATV define uma nova formulação para os modelos, de tal forma que um controlador para tais sistemas deve apresentar uma metodologia de adaptação temporal para as variações que sejam inerentes ao sistema. Em outros termos, o supervisor deve ser adaptativo com relação às variações dos tempos de vida dos eventos. Esta formulação é tratada aqui.

1.1 Autômatos e Modelos de SEDs

Autômatos [8] são modelos de máquinas de estado que podem representar algumas classes de sistemas. Dentre esses sistemas, encontram-se os Sistemas a Eventos Discretos (SEDs) [1], que são sistemas cuja evolução dinâmica é descrita pela ocorrência de eventos que alteram o estado do sistema.

Os autômatos permitem estruturar modelos de sistemas em que é necessário estudar seu funcionamento lógico através de sua evolução dinâmica descrita na forma de linguagens formais [38]. Exemplo disto é a formalização do problema de controle de SEDs [2].

Quando se necessita expressar uma representação temporal em um modelo de um sistema, um paradigma que pode ser utilizado é o autômato temporizado [39, 14, 40]. Esse autômato permite incluir representações de tempo em sua estrutura, tal que os símbolos ocorrem de acordo com instantes de tempo específicos, determinados em um ou mais relógio global. No caso do autômato temporizado, a linguagem é representada por um conjunto de pares (tempo, símbolo), sendo denominada de linguagem temporizada [41, 42, 43].

Outro formalismo utilizado para a representação temporal nos autômatos são os grafos de transições de atividades (GTAs) [44], que apresentam uma estrutura formal semelhante aos autômatos não temporizados. Porém, os GTAs incluem as definições dos tempos máximo e mínimo associados aos símbolos, em que um símbolo é definido como uma tripla $(t_{\min}, \text{evento}, t_{\max})$, em que t_{\min} é o menor tempo em que um símbolo torna-se habilitado e t_{\max} é o maior tempo em que o símbolo pode ocorrer. A evolução dinâmica dos GTAs pode ser representada através dos grafos de transições temporizadas (GTTs) [45], que incluem o símbolo t (*tick*) que é sincronizado a um relógio global. Esta representação gráfica apresenta um aumento de estados que ocorre devido à inclusão deste símbolo. Este tipo de autômato é utilizado por Brandin e Wonham [9] para controle de SEDs temporizados. Os GTAs em conjunto com os GTTs permitem representar sistemas que apresentam transições de estados com limites máximo e mínimo de ocorrência. A linguagem associada é uma linguagem formal que inclui o símbolo t .

Para a representação de sistemas em que os tempos são definidos como apenas o limite mínimo em que um símbolo pode ocorrer, o que representa um modelo livre de controle, encontra-se o autômato $(\max, +)$ [28], que é similar ao GTA, mas apresentando as definições dos tempos associados aos símbolos apresentados diretamente nos arcos (representação gráfica das funções de transição).

Este tipo de autômato temporizado é utilizado em conjunto com sua representação matricial [46] para tratar o problema de controle de SEDs temporizados [31, 32].

Dentre os vários formalismos de autômatos temporizados, apenas o ATV [34] se apresenta para representar diretamente modelos de sistemas em que o tempo associado ao símbolo varie de acordo com o último tempo decorrido em uma transição passada. Os formalismos citados anteriormente apresentam tempos constantes (como o autômato $(\max, +)$) ou definidos por intervalos de tempo (como os GTAs). Assim, para sistemas em que, a cada momento que ocorre um retorno a um estado anterior, o tempo associado ao símbolo neste estado recorrente adquire um novo valor dependente do tempo decorrido na ocorrência da transição que levou o autômato a este estado, estes modelos não satisfazem tais condições, mas o ATV pode ser utilizado diretamente para sua aplicação à síntese de supervisores de SED temporizados com variações nos tempos de vida [36, 37, 47].

Neste relatório é apresentada a utilização do Autômato com Temporização Variável [34] e sua aplicação à síntese de supervisores de SEDs temporizados, onde é introduzido o conceito de supervisores temporalmente adaptativos.

Capítulo 2

Preliminares

Neste Capítulo são apresentados alguns conceitos que serão utilizados no desenvolvimento deste Relatório. São eles: a álgebra de dióides, as séries formais e a representação matricial dos autômatos.

2.1 Álgebra de Dióides

Definição 2.1 *Um dióide é um conjunto D dotado de duas operações: \oplus (soma) e \otimes (multiplicação), que satisfaz os seguintes axiomas:*

Axioma 1: *Comutatividade de \oplus : $a \oplus b = b \oplus a$*

Axioma 2: *Associatividade de \oplus : $(a \oplus b) \oplus c = a \oplus (b \oplus c)$*

Axioma 3: *Associatividade de \otimes : $(a \otimes b) \otimes c = a \otimes (b \otimes c)$*

Axioma 4: *Distributividade de \otimes sobre \oplus :*

$$\begin{aligned}(a \oplus b) \otimes c &= (a \otimes c) \oplus (b \otimes c) \\ c \otimes (a \oplus b) &= (c \otimes a) \oplus (c \otimes b)\end{aligned}$$

Axioma 5: *Elemento nulo em \oplus : $a \oplus \epsilon = a$, $\forall a \in D$ e algum $\epsilon \in D$*

Axioma 6: *Absorção pelo elemento nulo em \otimes : $a \otimes \epsilon = \epsilon$*

Axioma 7: *Elemento identidade em \otimes : $a \otimes e = a$*

Axioma 8: *Idempotência em \oplus : $a \oplus a = a$.*

Um dióide é dito ser comutativo se \otimes é comutativo. Quando se considera $D = \mathbb{R}_{\max} = \mathbb{R} \cup -\infty$, $\epsilon = -\infty$, $e = 0$, \oplus é a operação usual \max e \otimes é a operação usual $+$. Nesse caso, D é comutativo e denomina-se a álgebra de “álgebra $(\max, +)$ ”.

2.1.1 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 $e = 0$, pois satisfazem:

$$\begin{aligned} \forall a \in \mathbb{R}, \quad a \oplus \epsilon &= a \oplus (-\infty) = \max\{a, -\infty\} = a = \\ &= \max\{-\infty, a\} = (-\infty) \oplus a = \epsilon \oplus a \end{aligned} \quad (2.1)$$

e

$$\forall a \in \mathbb{R}, \quad a \otimes e = a + 0 = a = 0 + a = e \otimes a. \quad (2.2)$$

Assim, para quaisquer dois elementos $a, b \in D$, com $D = \mathbb{R}_{\max}$, tem-se então que

$$a \otimes b = a + b = b + a = b \otimes a \quad (2.3)$$

de onde se vê que a álgebra $(\max, +)$ é um dióide comutativo.

No contexto da álgebra $(\max, +)$, o conjunto D pode ser definido sobre matrizes quadradas de dimensão n , isto é,

$$D = \mathbb{R}_{\max}^{n \times n}. \quad (2.4)$$

As propriedades do dióide $(\max, +)$ são igualmente satisfeitas, considerando que para duas matrizes $\mathbf{A}, \mathbf{B} \in \mathbb{R}_{\max}^{n \times n}$,

$$(\mathbf{A} \oplus \mathbf{B})_{i,j} = \mathbf{A}_{i,j} \oplus \mathbf{B}_{i,j} \quad (2.5)$$

$$(\mathbf{A} \otimes \mathbf{B})_{i,j} = \bigoplus_{i=1}^n (\mathbf{A}_{i,j} \otimes \mathbf{B}_{i,j}). \quad (2.6)$$

2.2 Séries Formais

As séries formais provêem um formalismo para tratar as linguagens formais, sejam elas temporizadas ou não temporizadas. Uma série formal é definida como [24, 48]:

Definição 2.2 *Uma série formal Y sobre um alfabeto Σ com coeficientes em um dióide D é um mapa*

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

onde 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 .

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

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

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

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

$$(y_1 \oplus y_2|s) = (y_1|s) \oplus (y_2|s) \quad (2.7)$$

$$(y_1 \otimes y_2|s) = \bigoplus_{uv=s} (y_1|u) \otimes (y_2|v), \quad (2.8)$$

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

As séries formais permitem descrever linguagens através da equação (2.8). 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*, como a seguir.

Definição 2.3 Uma linguagem regular $L = \{s, s', \dots\} \subseteq \Sigma^*$, é representada pela série formal

$$Y_L = \bigoplus_{s \in \Sigma^*} (y|s) s \quad (2.9)$$

onde $(y|s) \in \mathbb{B}$, e $s \in \Sigma^*$, 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 .

Exemplo 2.1 Dado o alfabeto $\Sigma = \{\alpha, \beta\}$, a linguagem $L = \{\varepsilon, \alpha, \alpha\beta, \beta\alpha, \alpha\alpha, \beta\beta, \beta\alpha\beta\}$ é representada pela série formal

$$Y_L = e\varepsilon \oplus e\alpha \oplus e\alpha\beta \oplus e\beta\alpha \oplus e\alpha\alpha \oplus e\beta\beta \oplus e\beta\alpha\beta \oplus \underbrace{\oplus e\alpha\alpha\alpha \oplus \dots \oplus e\beta\beta\beta \oplus \dots}_{\Sigma^* - L} \quad (2.10)$$

onde $(y|s) = e, \forall s \in L$, e $(y|s) = \epsilon, \forall s \in \Sigma^* - L$. A série formal da equação (2.10) pode ser escrita como

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

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

Deve-se observar que para uma linguagem L descrita por uma série formal Y_L , as operações de união e concatenação de linguagens 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 [41, 42, 43] que é uma linguagem que apresenta em cada palavra $s \in L$ um valor numérico t_s associado. Este valor numérico representa um intervalo de tempo decorrido para a palavra ser reconhecida por um autômato temporizado [39, 14, 40].

Definição 2.4 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

$$Y_L = \bigoplus_{s \in \Sigma^*} (y|s) s \quad (2.11)$$

onde $(y|s) \in \mathbb{R}_{\max}$, $s \in \Sigma$ 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 .

Exemplo 2.2 Dado o alfabeto $\Sigma = \{\alpha, \beta\}$, a linguagem $L = \{3\varepsilon, 4\alpha, 2\alpha\beta, 3\beta\alpha, 5\alpha\alpha, 2\beta\beta, \beta\alpha\beta\}$ é representada pela série formal

$$Y_L = 3\varepsilon \oplus 4\alpha \oplus 2\alpha\beta \oplus 3\beta\alpha \oplus 5\alpha\alpha \oplus 2\beta\beta \oplus e\beta\alpha\beta \oplus \underbrace{\oplus e\alpha\alpha\alpha \oplus \dots \oplus e\beta\beta\beta \oplus \dots}_{s \notin L}.$$

ou

$$Y_L = 3\varepsilon \oplus 4\alpha \oplus 2\alpha\beta \oplus 3\beta\alpha \oplus 5\alpha\alpha \oplus 2\beta\beta \oplus e\beta\alpha\beta \\ Y_L = 3 \oplus 4\alpha \oplus 2\alpha\beta \oplus 3\beta\alpha \oplus 5\alpha\alpha \oplus 2\beta\beta \oplus \beta\alpha\beta$$

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.

2.3 Representação Matricial de Autômatos

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 2.5 *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 2.6 *O estado inicial de um autômato G representado por uma matriz de incidência \mathbf{A} , é definido pelo vetor linha $\theta_{1 \times N}(\mathbf{A})$,*

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

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 2.6 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 2.3 *A matriz de incidência \mathbf{A} associada ao autômato da Figura 2.1, 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.

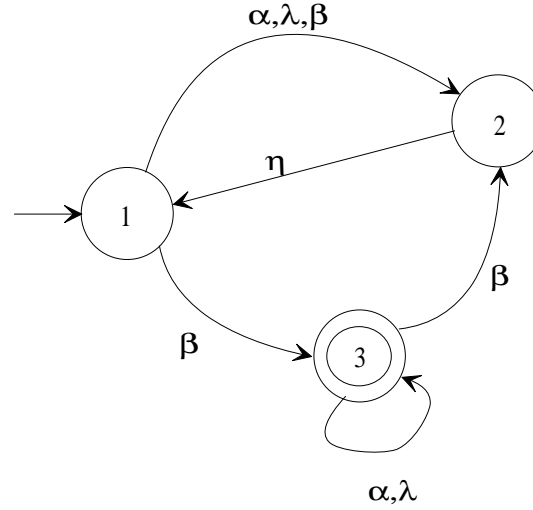


Figura 2.1: Autômato não determinístico com mais de uma função de transição entre dois estados.

2.3.1 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 2.7 *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 2.4 *Do autômato G mostrado na Figura 2.2, cuja matriz de incidência é*

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

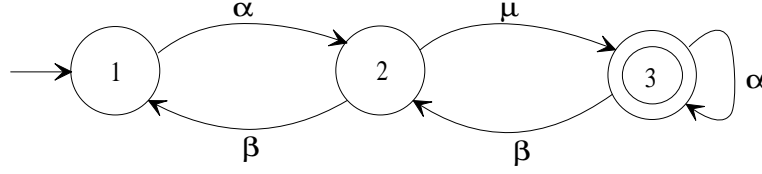


Figura 2.2: Autômato para exemplificar a construção da matriz de caminhos.

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

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

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

$$L(\mathbf{A}) = \bigoplus_i (\theta(\mathbf{A}) \otimes \mathbf{A}^i) = \bigoplus_i \bigoplus_{j=1}^n (\theta_1(\mathbf{A}) \otimes a_{1,j}^i), \quad (2.12)$$

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, $L(\mathbf{A}) = L(G)$.

A linguagem marcada da matriz de incidência \mathbf{A} é definida como sendo:

Definição 2.9 Para uma matriz de incidência \mathbf{A} , sua linguagem marcada é definida por

$$L_m(\mathbf{A}) = \bigoplus_i (\theta(\mathbf{A}) \otimes \mathbf{A}^i \otimes \phi(\mathbf{A})) = \bigoplus_i \bigoplus_{j=1}^n (\theta_1(\mathbf{A}) \otimes a_{1,j}^i \otimes \phi_j(\mathbf{A})), \quad (2.13)$$

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 2.8. Contudo, as palavras consideradas são as palavras que se encontram em uma coluna que leva a uma linha marcada j , denominada coluna marcada.

Exemplo 2.5 Seja o autômato apresentado na Figura 2.3. 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

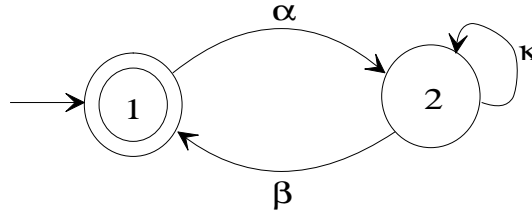


Figura 2.3: Autômato para determinação da linguagem.

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

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.

2.3.2 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 2.5, mas também seus tempos de vida. Assim, tem-se:

Definição 2.10 Seja $A_{(\max,+)}$ um autômato temporizado. Define-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

$$\theta(\mathbf{At}) = [t_{in} \quad \epsilon \quad \dots \quad \epsilon],$$

com t_{in} representando o atraso inicial, e os estados marcados são representados pelo vetor coluna

$$\phi(\mathbf{At}) = [t_{m_1} \quad t_{m_2} \quad \dots \quad t_{m_n}]^T,$$

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 2.6 O autômato $A_{(\max,+)}$ apresentado na Figura 2.4, tem sua matriz de incidência temporizada construída de acordo com a Definição 2.10, 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

$$\theta(\mathbf{At}) = [2 \quad \epsilon \quad \epsilon]$$

e os estados marcados definidos pelo vetor coluna

$$\phi(\mathbf{At}) = \begin{bmatrix} 2 \\ 3 \\ \epsilon \end{bmatrix}.$$

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 $\mu(\sigma_i)$ sendo o mapa

$$\mu : \Sigma \rightarrow \mathbb{R}_{\max}^{|Q| \times |Q|}, \quad (2.14)$$

que define a aplicação dos tempos de vida t_σ dos símbolos $\sigma \in \Sigma$ sobre $\mathbb{R}_{\max}^{|Q| \times |Q|}$, onde $\mathbb{R}_{\max}^{|Q| \times |Q|}$ é o díóide $(\mathbb{R} \cup \{-\infty\}, \max, +)$ sobre matrizes quadradas de dimensão $|Q|$, 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:

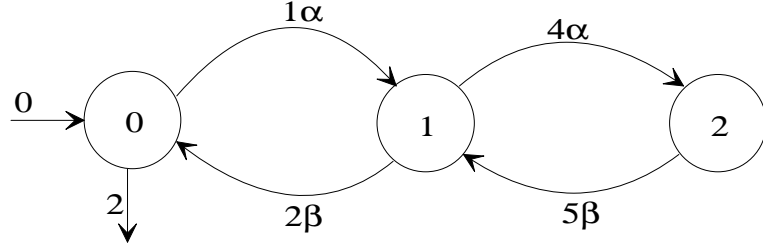


Figura 2.4: Autômato $(\max, +)$ determinístico para ilustrar a representação por matriz de incidência temporizada e vetores de estados inicial e marcados.

Definição 2.11 *A um autômato $A_{(\max, +)}$ que tem sua linguagem reconhecida temporizada representada pela série formal*

$$Y_L = \bigoplus_{s \in \Sigma^*} (y|s) s, \quad (2.15)$$

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

$$L_m(\mathbf{At}) = L(A_{(\max, +)}) = Y_L. \quad (2.16)$$

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 2.12 *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*

$$\mathbf{At}^n = \mathbf{At} \otimes \mathbf{At} \otimes \cdots \otimes \mathbf{At}, \quad (2.17)$$

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 2.7 *Do autômato $A_{(\max, +)}$ mostrado na Figura 2.5, sua representação matricial é dada por*

$$\mathbf{At} = \begin{bmatrix} \epsilon & 4\alpha & \epsilon \\ 5\beta & \epsilon & 2\mu \\ \epsilon & 2\beta & 2\alpha \end{bmatrix}, \quad \theta(\mathbf{At}) = [1 \quad \epsilon \quad \epsilon], \quad \phi(\mathbf{At}) = \begin{bmatrix} 3 \\ \epsilon \\ 2 \end{bmatrix}.$$

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

$$\mathbf{At}^2 = \mathbf{At} \otimes \mathbf{At} = \begin{bmatrix} 9\alpha\beta & \epsilon & 6\alpha\mu \\ \epsilon & 4\mu\beta + 9\beta\alpha & 4\mu\alpha \\ 7\beta\beta & 4\alpha\beta & 4\alpha\alpha + 4\beta\mu \end{bmatrix}$$

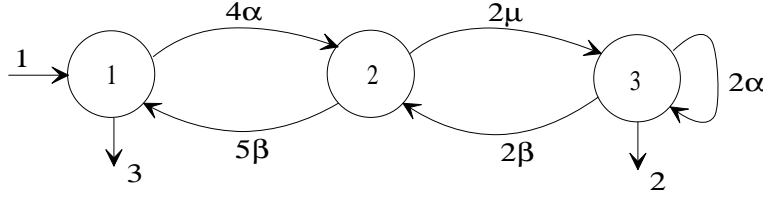


Figura 2.5: Autômato para exemplo da matriz de caminhos temporizada.

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

$$(y|\alpha\beta) = \theta(\mathbf{At}) \otimes \mu(\alpha) \otimes \mu(\beta) \otimes \phi(\mathbf{At}) = 13 \neq \epsilon.$$

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 2.13 Para uma dada matriz de incidência temporizada \mathbf{At} , sua linguagem é definida como

$$L(\mathbf{At}) = \bigoplus_i (\theta(\mathbf{At}) \otimes \mathbf{At}^i) = \bigoplus_i \bigoplus_{j=1}^n (\theta_1(\mathbf{At}) \otimes at_{1,j}^i), \quad (2.18)$$

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 .

Além do formalismo das séries formais, a linguagem marcada da matriz de incidência \mathbf{At} também pode ser definida como sendo:

Definição 2.14 Para uma matriz de incidência \mathbf{At} , sua linguagem marcada é definida como

$$L_m(\mathbf{At}) = \bigoplus_i (\theta(\mathbf{At}) \otimes \mathbf{At}^i \otimes \phi(\mathbf{At})) = \bigoplus_i \bigoplus_{j=1}^n (\theta_1(\mathbf{At}) \otimes at_{1,j}^i \otimes \phi_j(\mathbf{At})), \quad (2.19)$$

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 $\phi(\mathbf{At})$.

Exemplo 2.8 Seja o autômato apresentado na Figura 2.6. Sua representação matricial é

$$\mathbf{At} = \begin{bmatrix} \epsilon & 5\alpha \\ 3\beta & \kappa \end{bmatrix}, \theta(\mathbf{At}) = \begin{bmatrix} 1 & \epsilon \end{bmatrix}, \phi(\mathbf{At}) = \begin{bmatrix} 3 \\ e \end{bmatrix}.$$

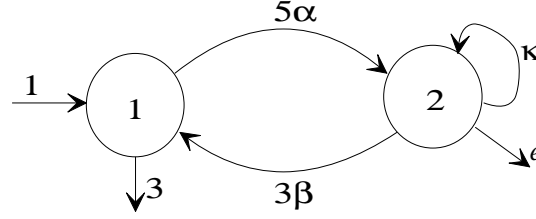


Figura 2.6: Autômato temporizado para determinação da linguagem temporizada.

Para determinar sua linguagem, calcula-se

$$\mathbf{At}^2 = \begin{bmatrix} 8\alpha\beta & 5\alpha\kappa \\ 3\kappa\beta & 8\beta\alpha + \kappa\kappa \end{bmatrix}$$

$$\mathbf{At}^3 = \begin{bmatrix} 8\alpha\kappa\beta & 13\alpha\beta\alpha + 5\alpha\kappa\kappa \\ 11\beta\alpha\beta + 3\kappa\kappa\beta & 8\kappa\beta\alpha + 8\beta\alpha\kappa + \kappa\kappa\kappa \end{bmatrix}$$

$$\vdots$$

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 $L_m(\mathbf{At})$.

Capítulo 3

Autômatos com Temporização Variável

Um autômato temporizado inclui uma representação do tempo na sua estrutura de transição, em que um intervalo de tempo está relacionado com cada função de transição, denominado tempo de vida do evento. O tempo de vida expressa o menor tempo para que o evento se torne habilitado. Um autômato temporizado reconhece uma linguagem temporizada [49], que é uma linguagem formalizada sobre pares (*evento*, *tempo*). Dentre as estruturas de autômatos temporizados, encontra-se a formalização do autômato $(\max, +)$ [28] que apresenta além de uma função de transição, um tempo de vida associado ao arco, bem como tempos de atrasos iniciais e atrasos finais. Este autômato pode ser utilizado para modelagem de sistemas em que o tempo de vida do evento seja definido como o instante de tempo em que o evento se torna habilitado, permanecendo habilitado até sua ocorrência [32]. No entanto, para casos em que em um determinado estado alcançado o tempo de vida do evento varia de acordo com o último tempo decorrido para o autômato chegar a este estado é necessário a definição do autômato com temporização variável, a qual é apresentada a seguir:

Definição 3.1 *Um autômato finito ATV sobre um alfabeto Σ é uma sextupla*

$$ATV = (Q, \Sigma, q_0, t_0, t_i, t_f),$$

em que:

- $Q = \{q_0, \dots, q_n\}$ é um conjunto finito de estados;
- $\Sigma = \{\sigma_1, \dots, \sigma_m\}$ é um alfabeto;
- q_0 é o estado inicial;
- $t_0 : q_0 \rightarrow R_{\max}$ é o atraso inicial;
- $t_i : t_{i-1} \times Q \times \Sigma \times Q \rightarrow R_{\max}$ é a função de tempo de transição dependente do último tempo decorrido e
- $t_f : t_{i-1} \times Q \rightarrow R_{\max}$ é a função de atraso final dependente do último tempo decorrido.

Um ATV é representado graficamente por vértices formados pelo conjunto de estados Q e pelos arcos a seguir:

1. Internos, $q_j \xrightarrow{\sigma} q_{j+1}$, $\forall q_j, q_{j+1} \in Q$ e $\sigma \in \Sigma$ tais que $t_i \neq \epsilon$. O arco $q_j \xrightarrow{\sigma} q_{j+1}$, é valorado por

$$t_i = f(t_{i-1}) \tag{3.1}$$

tal que o tempo de transição t_i do arco que aponta de um estado q_a para o estado q_b é uma função f do tempo decorrido anteriormente t_{i-1} pela ocorrência de um determinado evento, e que levou o autômato do estado q_c para o estado q_a ;

2. O arco de entrada $\rightarrow q_0$, valorado por $t_0 \neq \epsilon$;
3. Os arcos de saída $q_j \rightarrow$, valorados por t_f , $\forall q_f \in Q$ tais que

$$t_f = f(t_{i-1}) \neq \epsilon, \quad (3.2)$$

com t_f definido de forma igual à t_i .

Exemplo 3.1 Na Figura 3.1 é visto um exemplo de um *ATV*. Observe que cada arco tem uma função temporal que determina o tempo de vida do evento dependente do tempo decorrido anteriormente, seja ele o tempo de atraso inicial (no estado inicial), seja pelo tempo de vida do último evento ocorrido no *ATV*. As funções de tempo de transição são $f_1 = t^2 - t$ e $f_2 = t$, e o tempo de atraso final é definido pela função $t_f = \sqrt{t}$. Assim, decorrido o tempo de atraso inicial $t_0 = 3$, o evento α tem um tempo de vida dado por $f_1 = 3^2 - 3 = 6$. Com a ocorrência do evento α , o estado 2 é alcançado, e o tempo de vida do evento β é $f_2 = 6$. Com sua ocorrência, o estado inicial é novamente alcançado e o novo tempo de vida do evento α é dado por $f_1 = 6^2 - 6 = 30$, e assim por diante. Os tempos de atraso final para uma palavra ser reconhecida (quando o estado 2 é alcançado) são dependentes do tempo decorrido no evento α , isto é, ao ocorrer a primeira vez o evento α , $t_f = \sqrt{6}$. Na segunda vez que o evento α ocorre, $t_f = \sqrt{30}$ e assim por diante.

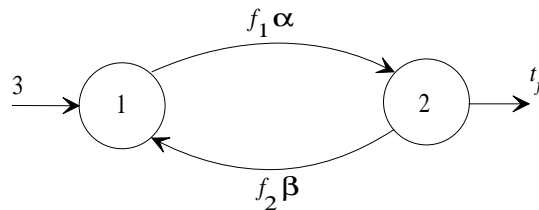


Figura 3.1: Exemplo de um *ATV*.

Observa-se que o autômato $(\max, +)$ é um caso restrito do *ATV*, em que se consideram todas as funções de tempos de vida dos eventos definidas como constantes.

Com a definição do *ATV*, sua semântica é definida de maneira similar ao autômato $(\max, +)$, considerando as variações nos tempos das transições, como a seguir:

1. Há um relógio global sendo incrementado;
2. O tempo de vida t_i de um evento σ , é uma função f do último tempo decorrido t_{i-1} de um evento σ' , que levou o *ATV* para o estado em que o evento σ é definido, e seu valor calculado é o tempo mínimo necessário para sua habilitação;
3. Para iniciar a execução do autômato (alcançar o estado inicial) é transcorrido um tempo t_0 no relógio global;
4. Estando no estado inicial (ou em qualquer outro), os contadores dos eventos definidos neste estado (valores calculados de cada função de tempo de vida de evento - $t_i(\sigma)$) vão sendo decrementados;
5. Quando um contador de um dos eventos definidos no estado é zerado, o evento torna-se habilitado e pode 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 *ATV* reconhece este estado, reconhecendo assim a palavra que o levou do estado inicial até este estado;
9. Eventos iguais com diferentes funções de tempos de vida definem não determinismo no *ATV*.

Um exemplo da execução de um *ATV* é apresentado a seguir.

Exemplo 3.2 Seja $\Sigma = \{\alpha, \beta\}$. O autômato com conjunto de estados $Q = \{1, 2, 3\}$, atraso inicial $t_0 = 2$, funções de tempos de transições $f_1(t) = t^2 - t$, $f_2(t) = 3t$, $f_3(t) = 4$, $f_4(t) = 1 + 2t$ e função de atraso final $f_5(t_{i-1}) = 5 - \frac{2}{t}$ (os outros valores de $f(t_i)$ e $f(t_f)$ são ϵ) está apresentado na Figura 3.2. O valor de t é interpretado como t_{i-1} , isto é, o último tempo decorrido para atingir um dado estado. Assim, o tempo de cada transição em um determinado instante de tempo é sempre calculado em relação ao tempo decorrido para atingir o estado em que o autômato se encontra. Assim, o valor do primeiro tempo de transição no arco $q_1 \xrightarrow{\alpha} q_2$ é $t_1 = f_1(2) = 2$, desde que $t_0 = 2$. t_1 é o tempo mínimo para a ocorrência do evento α que muda o estado do autômato de q_1 para q_2 . O tempo final para reconhecer a palavra vazia ϵ é $t_f = f_5(2) = 4$, tal que seu reconhecimento tem um tempo total $t = 6$. Na ocorrência de α , o autômato vai para o estado 2, onde o tempo de transição para α é $t_2 = f_2(2) = 6$ e para β é $t_2 = f_4(2) = 5$. Se ocorre α , o autômato vai para o estado 3, onde o tempo de transição para β é $t_3(5) = 4$. Ainda no estado 2, se ocorre β , o autômato vai para o estado 1, onde o novo tempo de transição para α é $t_3(6) = 30$ e para reconhecer a palavra $\alpha\beta$, o atraso final é $t_f = 4,6$, tal que seu reconhecimento tem um tempo total $t = 13,6$.

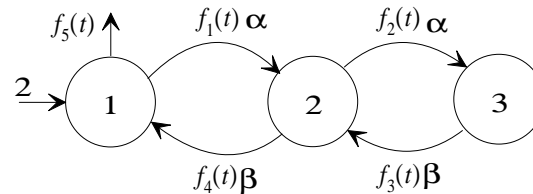


Figura 3.2: Autômato com Temporização Variável

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. Este reconhecimento pode ser descrito por meio das seqüências de eventos através dos datadores definidos sobre o conjunto $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$, do dióide $(\max, +)$ [11].

Com a definição do dióide $(\max, +)$, define-se um datador para um *ATV* como a seguir:

Definição 3.2 Um datador é um mapa

$$Y : \Sigma^* \rightarrow \mathbb{R}_{\max},$$

onde Y é o tempo que um autômato *ATV* leva para percorrer uma seqüência

$$s = \sigma_1 \sigma_2 \dots \sigma_n \in \Sigma^*.$$

O datador define os tempos das palavras de Σ^* . Assim, o autômato ATV leva um tempo Y para percorrer uma seqüência s , partindo de um estado q para um outro estado q' , onde $(y|s)$ denota o valor de Y na palavra s [24].

Definição 3.3 Um datador Y é dito reconhecível se existir um ATV tal que $(y|s) \neq \epsilon$.

Exemplo 3.3 No Exemplo 3.2, vê-se que $s = \alpha\beta$ é uma palavra reconhecida, pois

$$(y|\alpha\beta) = (t_0 + f_1(t_0) + f_4(t_1) + t_f) = 13,67 \neq \epsilon.$$

3.1 Representação Matricial

Dentre as representações dos autômatos, a matriz de incidência é uma importante definição. Esta representação é utilizada para formalismos de construção de linguagens em autômatos não temporizados [46], bem como para autômatos temporizados $(\max, +)$ [32]. No caso dos $ATVs$, esse formalismo é semelhante ao formalismo dos autômatos $(\max, +)$. A definição da matriz de incidência de um ATV é apresentada a seguir:

Definição 3.4 Seja ATV um autômato com temporização variável. Define-se a matriz de incidência temporizada, denotada por \mathbf{Av} , como

$$\mathbf{Av} = [av_{i,j}]; av_{i,j} = \begin{cases} f(t_\sigma) \sigma & \text{se } \exists \sigma \text{ do estado } i \text{ para o estado } j; \\ \epsilon & \text{caso contrário,} \end{cases}$$

em que $f(t_\sigma)$ é a função do tempo de vida do evento σ que leva o autômato do estado i para o estado j . Se mais de um evento é definido do estado i para o estado j , então

$$av_{i,j} = \bigoplus_k f(t_{\sigma_k}) \sigma^k.$$

O estado inicial é definido como sendo o estado 1, representado pelo vetor linha

$$\theta(\mathbf{Av}) = [t_0 \quad \epsilon \quad \dots \quad \epsilon],$$

e os estados marcados são representados pelo vetor coluna

$$\phi(\mathbf{Av}) = [f(t_{f_1}) \quad f(t_{f_2}) \quad \dots \quad f(t_{f_n})]^T.$$

Nessa representação, se um estado k não é marcado, $f(t_{f_k}) = \epsilon$. A notação θ_j , refere-se a j -ésima coluna do vetor θ , e ϕ_i refere-se a i -ésima linha do vetor ϕ .

Exemplo 3.4 O ATV apresentado na Figura 3.2, tem sua matriz de incidência dada por

$$\mathbf{Av} = \begin{bmatrix} \epsilon & (t^2 - t)\alpha & \epsilon \\ (1 + 2t)\beta & \epsilon & (3t)\alpha \\ \epsilon & 4\beta & \epsilon \end{bmatrix}$$

$$\theta(\mathbf{Av}) = [2 \quad \epsilon \quad \epsilon], \quad \phi(\mathbf{Av}) = \begin{bmatrix} 5 - 2/t \\ \epsilon \\ \epsilon \end{bmatrix}$$

3.1.1 Linguagens

Há duas formas de construir as linguagens dos *ATV*s: utilizando das séries formais e utilizando as matrizes de incidência. Ambas, são formalizadas através da álgebra de dióides.

A seguir são apresentados os conceitos básicos das séries formais, para proceder ao estudo das linguagens dos *ATV*s.

Descrição das Linguagens dos *ATV*s por Séries Formais

A utilização das séries formais para definir a evolução dinâmica dos *ATV* é similar aos autômatos $(\max, +)$.

Definição 3.5 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

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

onde $(y|s) \in \mathbb{R}_{\max}$, $s \in \Sigma$ 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 .

Exemplo 3.5 Dado o alfabeto $\Sigma = \{\alpha, \beta\}$, a linguagem $L = \{3\epsilon, 4\alpha, 6, 3\alpha\beta, 3\beta\alpha\}$ é representada pela série formal

$$Y_L = 3\epsilon \oplus 4\alpha \oplus 6, 3\alpha\beta \oplus 3\beta\alpha = 3 \oplus 4\alpha \oplus 6, 3\alpha\beta \oplus 3\beta\alpha$$

desde que $\epsilon \otimes L = \epsilon$, $\forall L \subseteq \Sigma^*$. Esta série representa a linguagem reconhecida por um *ATV*. Neste contexto, $(y|s)$ denota o coeficiente da palavra s que é igual ao zero ' ϵ ' se s não é reconhecida pelo *ATV*.

Com essa formalização, tem-se que a função datadora Y pode ser escrita através de uma série formal sobre o alfabeto Σ com coeficientes em \mathbb{R}_{\max} . Isto é,

Definição 3.6 A linguagem temporizada L de um *ATV* é definida por uma série formal como

$$L(ATV) = \bigoplus_{s \in \Sigma^*} (y|s) s,$$

em que $(y|s) \in \mathbb{R}_{\max}$ denota o datador da palavra s .

Exemplo 3.6 Considerando o autômato do Exemplo 3.2, a série formal Y_L identificada com a linguagem temporizada do autômato $L(ATV)$ admite a expressão racional

$$Y_L = 6 \oplus 13, 67\alpha\beta \oplus 74, 95\alpha\beta\alpha\beta \oplus 27, 78\alpha\alpha\beta\beta \oplus \dots$$

Construção das Linguagens dos *ATV* através das Matrizes de Incidência

A construção das linguagens das matrizes de incidência temporizadas utiliza a Definição 2.12 da matriz de caminhos temporizada, sendo ela modificada como a seguir:

Definição 3.7 Seja \mathbf{Av} uma matriz de incidência. A matriz

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

é uma matriz de caminhos, onde cada elemento $av_{i,j}^n$ representa um ou mais caminhos de comprimento n , formados de eventos controláveis ou não controláveis, que levam o autômato que ela representa, do estado i para o estado j , com um tempo total

$$f(t_s) = f(t_{\sigma^1}) \otimes \dots \otimes f(t_{\sigma^n}), s = \sigma^1 \dots \sigma^n.$$

Os vetores de estado inicial e de estados marcados da matriz de caminhos \mathbf{Av}^n são os mesmos de \mathbf{Av} .

Se não há um caminho com n eventos que muda o estado do autômato do estado i para o estado j , $av_{i,j}^n = \epsilon$. A matriz \mathbf{Av}^n contém palavras s de comprimento n , compostas de eventos controláveis e não controláveis, que são percorridas em um tempo

$$f(t_s) = f(t_{\sigma^1}) \otimes \dots \otimes f(t_{\sigma^n}).$$

Exemplo 3.7 Do autômato ATV mostrado na Figura 3.3, sua representação matricial é dada por

$$\mathbf{Av} = \begin{bmatrix} \epsilon & 2t\alpha \\ (2-t)\beta & \epsilon \end{bmatrix}, \quad \begin{aligned} \theta(\mathbf{Av}) &= \begin{bmatrix} 2 & \epsilon \end{bmatrix}, \\ \phi(\mathbf{Av}) &= \begin{bmatrix} 3t \\ \epsilon \end{bmatrix}. \end{aligned}$$

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

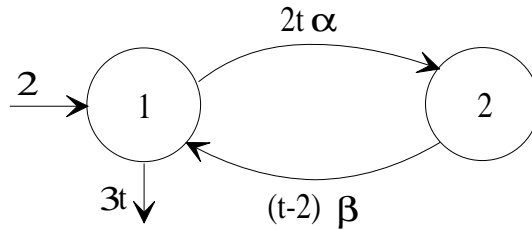


Figura 3.3: Autômato para exemplo da matriz de caminhos

$$\mathbf{Av}^2 = \mathbf{Av} \otimes \mathbf{Av} = \begin{bmatrix} ((2t_1) + (t_2 - 2))\alpha\beta & \epsilon \\ \epsilon & ((t_2 - 2) + (2t_3))\beta\alpha \end{bmatrix}$$

com $\theta(\mathbf{Av}^2) = \theta(\mathbf{Av})$ e $\phi(\mathbf{Av}^2) = \phi(\mathbf{Av})$. Nessa matriz de caminhos, cada seqüência representa uma palavra de comprimento 2, que muda o estado do autômato ATV, do estado i para o estado j . Observe que cada termo (i, j) de \mathbf{Av}^2 não contém uma equação apenas em t . Isto porque cada tempo t_k é calculado separadamente, de acordo com a evolução do ATV.

Com essa definição, tem-se que a linguagem da matriz do ATV é definida como a seguir:

Definição 3.8 Para uma matriz de incidência \mathbf{Av} , sua linguagem é definida como

$$L(\mathbf{Av}) = \bigoplus_i (\theta(\mathbf{Av}) \otimes \mathbf{Av}^i) = \bigoplus_i \bigoplus_{j=1}^n (\theta_1(\mathbf{Av}) \otimes av_{1,j}^i),$$

onde $\theta_1(\mathbf{Av})$ é o elemento da primeira coluna do vetor de estado inicial $\theta(\mathbf{Av})$, $av_{1,j}^i$ é o elemento da linha 1, coluna j da matriz de caminhos temporizada \mathbf{Av}^i .

A linguagem marcada é definida como:

Definição 3.9 Para uma matriz de incidência \mathbf{Av} , sua linguagem marcada é definida como

$$\begin{aligned} L_m(\mathbf{Av}) &= \bigoplus_i (\theta(\mathbf{Av}) \otimes \mathbf{Av}^i \otimes \phi(\mathbf{Av})) \\ &= \bigoplus_i \bigoplus_{j=1}^n (\theta_1(\mathbf{Av}) \otimes av_{1,j}^i \otimes \phi_j(\mathbf{Av})), \end{aligned}$$

onde $\theta_1(\mathbf{Av})$ é o elemento da primeira coluna do vetor de estado inicial $\theta(\mathbf{Av})$, $av_{1,j}^i$ é o elemento da linha 1, coluna marcada j da matriz de caminhos temporizada \mathbf{Av}^i e $\phi_j(\mathbf{Av})$ é o elemento da j -ésima linha do vetor de estados marcados $\phi(\mathbf{Av})$.

Exemplo 3.8 Seja o autômato apresentado na Figura 3.3. Para determinar sua linguagem, utiliza-se a matriz de incidência, a matriz de caminhos \mathbf{Av}^2 encontrada no Exemplo 3.7, e calcula-se

$$\mathbf{Av}^3 = \begin{bmatrix} \epsilon & ((2t_1) + (t_2 - 2) + (2t_3)) \alpha \beta \alpha \\ ((t_2 - 2) + (2t_3) + (t_4 - 2)) \beta \alpha \beta & \epsilon \\ \vdots & \vdots \end{bmatrix}$$

em que, para cada uma dessas matrizes citadas, os elementos da primeira linha definem as palavras de comprimento 1, 2, 3, ..., da linguagem de \mathbf{Av} . Calculando-se os tempos t_i , tem-se $t_1 = 2$, $t_2 = 4$, $t_3 = 4$, $t_4 = 4, \dots$. Substituindo esses valores e multiplicando essas matrizes por $\theta_1(\mathbf{Av})$, encontra-se

$$L(\mathbf{Av}) = \{2\epsilon, 6\alpha, 8\alpha\beta, 12\alpha\beta\alpha, \dots\}.$$

Observe que a palavra vazia ϵ está presente em $L(\mathbf{Av})$, a qual tem um tempo de vida igual a 2 (tempo de atraso inicial do autômato), e que as palavras de comprimento i , encontram-se na linha 1 das matrizes \mathbf{Av}^i . A linguagem marcada desta matriz é determinada pelas palavras da linha 1 de \mathbf{Av}^i , multiplicadas por $\theta_1(\mathbf{Av})$ e $\phi_k(\mathbf{Av})$, em que seu valor é calculado em relação ao último tempo decorrido. Então, $L_m(\mathbf{Av})$ é dada por

$$L_m(\mathbf{Av}) = \{8\epsilon, 16\alpha\beta, \dots\}.$$

3.2 Exemplos

Nessa seção são apresentados exemplos do *ATV*, para mostrar algumas de suas potencialidades em termos de representação.

Exemplo 3.9 Seja o autômato $(\max, +)$ não determinístico infinito, visto na Figura 3.4(a). Este autômato apresenta vários estados iniciais que representam entradas de processos externos. Dependendo do tempo de chegada do processo externo, define-se o estado inicial do autômato. Estando em um estado inicial, o autômato pode seguir sua evolução dinâmica infinitamente, sempre aumentando os tempos de vida dos eventos. O autômato pode reconhecer uma palavra sempre após o acontecimento do evento κ , decorridas duas unidades de tempo (tempo constante). Observe que este autômato pode ser particionado em vários sub-autômatos semelhantes com os tempos de vida dos eventos diferentes e ligados em seqüência, o que pode ser visto na Figura 3.4(b). Dessa forma, descrevendo os tempos de atraso iniciais pela função $f_0 = t + 2$, e os tempos de vida dos eventos respectivos por: $\alpha - f_1 = t^2 - 2t$; $\beta - f_2 = t$; $\mu - f_3 = t$ e $\kappa - f_4 = t + 2$, pode-se representar este autômato pelo *ATV* visto na Figura 3.4(c), que é um modelo compacto do autômato infinito da Figura 3.4(a).

Exemplo 3.10 Considere um suposto sistema em que há dois tipos de peças (p_1 e p_2) para serem processadas. Cada tipo de peça necessita ter sua temperatura elevada em um valor específico, para poder ser processado (evento α). Também, cada um tipo tem um tempo específico para ser processado. Ao ser processada uma peça (evento β), esta é enviada a um buffer onde é consumida (evento κ), e o sistema retorna ao estado inicial. A primeira peça a ser processada sempre é do tipo p_1 . Se a peça que está para ser processada não atingir a temperatura ideal, ela deve ser reesquentada para poder ser reprocessada (evento η). Se após o processamento esta peça ficar defeituosa, ela é enviada a um lugar de peças defeituosas (evento μ), em que é refeita e devolvida ao buffer de entrada (evento π), onde é reprocessada. Este suposto sistema é apresentado na Figura 3.5. O ATV que modela este sistema está apresentado na Figura 3.6(a), em que as funções temporais dos arcos são definidas por: $f_1 = t^2 - 1$, $f_2 = 2t + 1$, $f_3 = 2 \vee 3$, $f_4 = \sqrt{t + 1}$, $f_5 = 2t$ e $f_6 = \sqrt{(\frac{t}{2} + 1)} / 2$. A função f_3 define uma condição aleatória, cujo valor pode ser 2 ou 3 unidades de tempo. A função de atraso final é $f_7 = t + 2$ e $t_0 = 2$. Com essas funções, observa-se que é possível diferenciar o processo no autômato (qual tipo de peça está sendo processada), pelos tempos de vida específicos dos eventos, embora os eventos sejam os mesmos para ambos tipos de peças. Nesse caso, o sistema é determinístico. Na Figura 3.6(b), encontra-se o sistema modelado por um autômato $(\max, +)$. Vê-se nesta figura que neste caso, o modelo do sistema é não determinístico e contém mais estados do que o ATV. Além do mais, existe a necessidade da diferenciação de alguns estados (caso do estado inicial) e dos tempos de vida dos eventos, para poder se reconhecer qual o processo em andamento. Observe que, como inicialmente o sistema sempre processa uma peça do tipo p_1 , no ATV isto é representado diretamente pelo valor da função f_1 calculado sobre $t_0 = 2$. No caso do autômato $(\max, +)$ há pseudo-estados iniciais para que a evolução dinâmica satisfaça os requisitos. Assim, o modelo em ATV deste sistema mostra-se bastante simplificado.

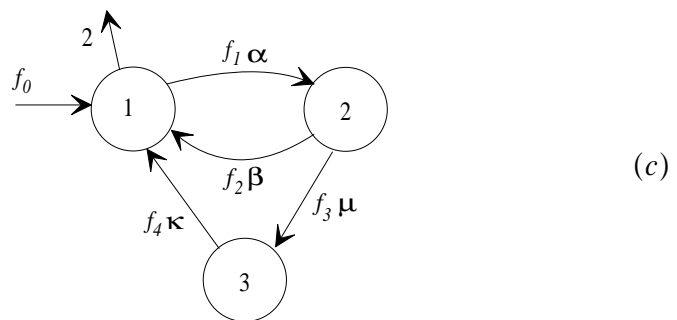
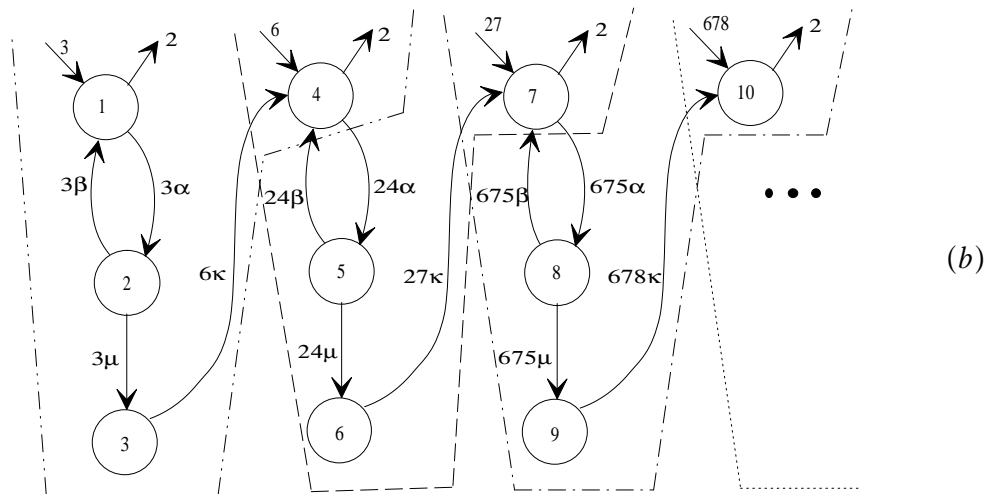
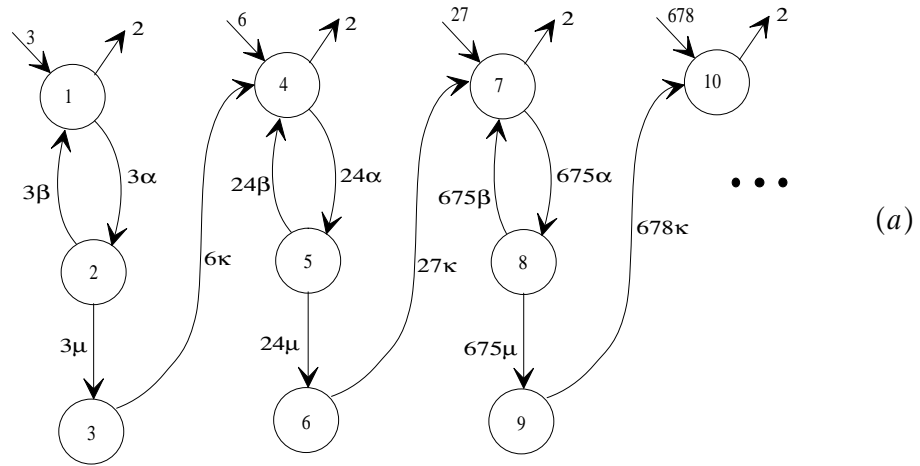


Figura 3.4: (a) Autômato $(\max,+)$ infinito, (b) particionamento do autômato $(\max,+)$ e (c) ATV como modelo compacto do autômato $(\max,+)$.

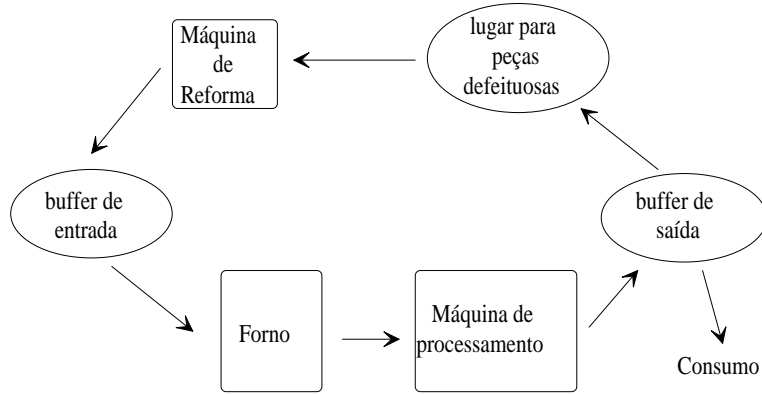
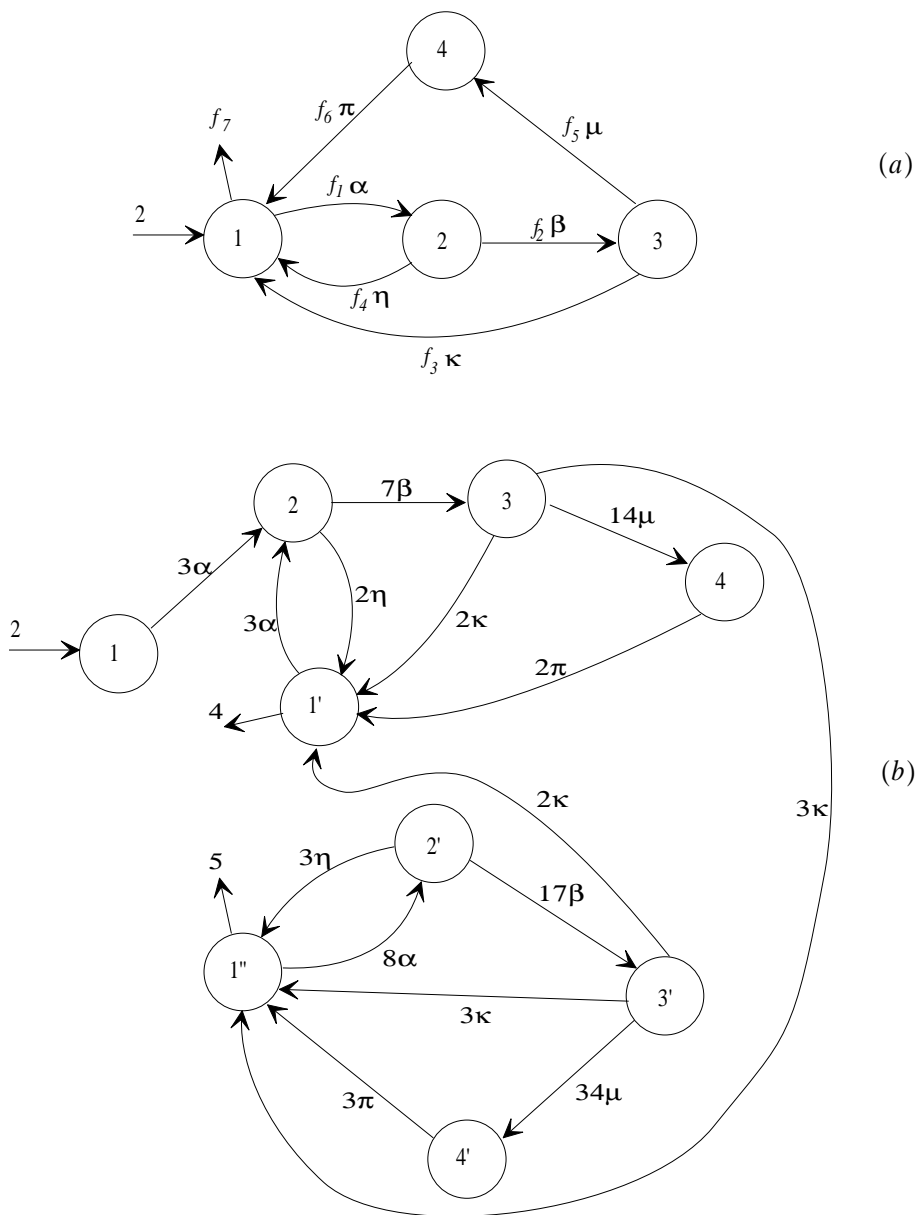


Figura 3.5: Sistema.

Figura 3.6: (a) Sistema modelado por um *ATV* e (b) sistema modelado por um autômato $(\max,+)$.

Capítulo 4

Aplicação dos ATVs aos SEDTs

Dado um *ATV* com representação matricial \mathbf{Av} , θ e ϕ , é necessário definir as operações e propriedades de suas matrizes de incidência para proceder à síntese do supervisor para SEDTs modelados por *ATVs*. Assim, tem-se a seguir as definições da acessibilidade e coacessibilidade das matrizes de incidência e a definição da composição síncrona de *ATVs*.

4.1 Acessibilidade e Coacessibilidade

A acessibilidade e coacessibilidade de uma linha de \mathbf{Av} são definidas como:

Definição 4.1 Uma linha j de uma matriz de incidência \mathbf{Av} é acessível se para algum $i \in \mathbb{N}^*$,

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

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

Definição 4.2 Uma linha i de uma matriz de incidência \mathbf{Av} é coacessível se para algum $k \in \mathbb{N}^*$,

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

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

Assim, uma linha j é acessível se, partindo da linha 1, existe pelo menos uma seqüência $s \neq \epsilon$ que leva à linha j e uma linha i é coacessível se, partindo da linha i , existe pelo menos uma seqüência $s \neq \epsilon$ que leva à linha marcada j . Dessa maneira, uma matriz \mathbf{Av} é acessível se toda linha j é acessível, e coacessível se toda linha i é coacessível. Uma matriz de incidência \mathbf{Av} é *trim* se todas as linhas forem acessíveis e coacessíveis.

Exemplo 4.1 No autômato da Figura 3.3, todos os estados são acessíveis, pois $\theta(\mathbf{Av}) \otimes \mathbf{Av}^1 \otimes \pi = 6\alpha \neq \epsilon$, para $\pi = [\epsilon \ e]^T$, que leva do estado 1 ao estado 2 e $\theta(\mathbf{Av}) \otimes \mathbf{Av}^2 \otimes \pi = 8\alpha\beta \neq \epsilon$, para $\pi = [e \ \epsilon]^T$, que leva do estado 1 ao estado 1. Todos os estados são coacessíveis, pois $\nu \otimes \mathbf{Av}^2 \otimes \phi(\mathbf{Av}) = 14\alpha\beta \neq \epsilon$, para $\nu = [e \ \epsilon]$, $\nu \otimes \mathbf{Av}^1 \otimes \phi(\mathbf{Av}) = 8\beta \neq \epsilon$, para $\nu = [\epsilon \ e]$. Logo, a matriz \mathbf{Av} representa um autômato *trim*.

4.2 Composição Síncrona

A composição síncrona de *ATV* necessita do seguinte operador de interseção:

Definição 4.3 Dado um dióide $D = \mathbb{R}_{\max} \langle \langle \Sigma \rangle \rangle$, o operador \otimes define a interseção de elementos, como

$$a \otimes b = \bigoplus_{i=1}^k (f(t_{\sigma^i}) \oplus f(t'_{\sigma^i})) \sigma^i, \text{ se } f(t_{\sigma^i}) \sigma^i \subset a \wedge f(t'_{\sigma^i}) \sigma^i \subset b. \quad (4.1)$$

$\forall a, b \in D$, onde $f(t_{\sigma}) \sigma \otimes \epsilon = \epsilon$, $f(t_{\sigma^1}) \sigma^1 \otimes f(t'_{\sigma^2}) \sigma^2 = \epsilon$ e $f(t_{\sigma}) \sigma \otimes f(t'_{\sigma}) \sigma = (f(t_{\sigma}) \sigma \oplus f(t'_{\sigma}) \sigma) \sigma$, $\forall \sigma, \sigma^1, \sigma^2 \in \Sigma$ e $\forall f(t_{\sigma}), f(t_{\sigma^1}), f(t'_{\sigma}), f(t'_{\sigma^2}) \in \mathbb{R}_{\max}$. Para matrizes de incidência temporizadas, esse operador é definido como

$$\mathbf{Cv} = \mathbf{Av} \otimes \mathbf{Bv}, cv_{i,j} = av_{i,j} \otimes bv_{i,j}, \quad (4.2)$$

para $av_{i,j} \otimes bv_{i,j}$ definido de acordo com 4.1, e

$$\theta(\mathbf{Cv}) = \theta(\mathbf{Av}) \otimes \theta(\mathbf{Bv}) \text{ e } \phi(\mathbf{Cv}) = \phi(\mathbf{Av}) \otimes \phi(\mathbf{Bv}).$$

No caso específico de controle de SEDTs, essa definição deve ser restringida, devido ao fato de não se poder aplicar controle a eventos não controláveis, impondo atrasos. Assim, tem-se:

Definição 4.4 Considerando que \mathbf{Av} representa uma especificação de comportamento (tempos de vida dos eventos variáveis) para um SED temporizado modelado por uma matriz de incidência \mathbf{Bv} (tempos dos eventos não controláveis constantes), em que $av_{i,j}$ e $bv_{i,j}$ podem ser expressões regulares do tipo $f(t_{\sigma^1}) \sigma^1 + \dots + f(t_{\sigma^k}) \sigma^k$ e $f(t'_{\sigma^1}) \sigma^1 + \dots + f(t'_{\sigma^k}) \sigma^k$, respectivamente, então o operador de interseção \otimes é definido como

$$av \otimes bv = \begin{cases} \bigoplus_{i=1}^k (f(t_{\sigma^i}) \oplus f(t'_{\sigma^i})) \sigma^i & \text{se } (f(t_{\sigma^i}) \sigma^i \subset av) \wedge (f(t'_{\sigma^i}) \sigma^i \subset bv) \\ & \wedge (\sigma^i \in \Sigma_c) \\ \bigoplus_{i=1}^k f(t'_{\sigma^i}) \sigma^i & \text{se } (f(t_{\sigma^i}) \sigma^i \subset av) \wedge (f(t'_{\sigma^i}) \sigma^i \subset bv) \\ & \wedge (\sigma^i \in \Sigma_{uc}). \end{cases} \quad (4.3)$$

A Definição 4.4, se aplica ao caso em que se considera $\Sigma = \Sigma_c \cup \Sigma_{uc}$. Assim, o produto síncrono é definido a seguir:

Definição 4.5 Dadas duas matrizes de incidência temporizadas $\mathbf{Av}_{m \times m}$ e $\mathbf{Bv}_{n \times n}$, as quais representam o ATV_1 construído com símbolos do alfabeto Σ_1 e o ATV_2 construído com símbolos do alfabeto Σ_2 , respectivamente, compõe-se uma nova matriz de incidência \mathbf{Pv} , com dimensão $m \times n$, construída com símbolos dos alfabetos Σ_1 e Σ_2 , através da composição paralela definida por

$$\mathbf{Pv} = \mathbf{Av} || \mathbf{Bv}, pv_{k,l} = av_{i_{\mathbf{Av}}, j_{\mathbf{Av}}} \otimes bv_{i_{\mathbf{Bv}}, j_{\mathbf{Bv}}},$$

onde

$$k = (i_{\mathbf{Av}}, i_{\mathbf{Bv}}) = i_{\mathbf{Av}} + m(i_{\mathbf{Bv}} - 1)$$

e

$$l = (j_{\mathbf{Av}}, j_{\mathbf{Bv}}) = j_{\mathbf{Av}} + m(j_{\mathbf{Bv}} - 1),$$

de tal forma que

$$\mathbf{Pv} = \begin{bmatrix} av_{1,1} \otimes [bv_{1,1} \ bv_{1,2} \ \dots \ bv_{1,n}] & av_{1,2} \otimes [bv_{1,1} \ bv_{1,2} \ \dots \ bv_{1,n}] & \dots & av_{1,m} \otimes [bv_{1,1} \ bv_{1,2} \ \dots \ bv_{1,n}] \\ \vdots & \vdots & \ddots & \vdots \\ av_{m,1} \otimes [bv_{1,1} \ bv_{1,2} \ \dots \ bv_{1,n}] & av_{m,2} \otimes [bv_{1,1} \ bv_{1,2} \ \dots \ bv_{1,n}] & \dots & av_{m,m} \otimes [bv_{1,1} \ bv_{1,2} \ \dots \ bv_{1,n}] \\ av_{1,1} \otimes [bv_{2,1} \ bv_{2,2} \ \dots \ bv_{2,n}] & av_{1,2} \otimes [bv_{2,1} \ bv_{2,2} \ \dots \ bv_{2,n}] & \dots & av_{1,m} \otimes [bv_{2,1} \ bv_{2,2} \ \dots \ bv_{2,n}] \\ \vdots & \vdots & \ddots & \vdots \\ av_{m,1} \otimes [bv_{2,1} \ bv_{2,2} \ \dots \ bv_{2,n}] & av_{m,2} \otimes [bv_{2,1} \ bv_{2,2} \ \dots \ bv_{2,n}] & \dots & av_{m,m} \otimes [bv_{2,1} \ bv_{2,2} \ \dots \ bv_{2,n}] \\ \vdots & \vdots & \ddots & \vdots \\ av_{1,1} \otimes [bv_{n,1} \ bv_{n,2} \ \dots \ bv_{n,n}] & av_{1,2} \otimes [bv_{n,1} \ bv_{n,2} \ \dots \ bv_{n,n}] & \dots & av_{1,m} \otimes [bv_{n,1} \ bv_{n,2} \ \dots \ bv_{n,n}] \\ \vdots & \vdots & \ddots & \vdots \\ av_{m,1} \otimes [bv_{n,1} \ bv_{n,2} \ \dots \ bv_{n,n}] & av_{m,2} \otimes [bv_{n,1} \ bv_{n,2} \ \dots \ bv_{n,n}] & \dots & av_{m,m} \otimes [bv_{n,1} \ bv_{n,2} \ \dots \ bv_{n,n}] \end{bmatrix}$$

se $\Sigma_1 = \Sigma_2$. Os estados marcados da composição são definidos por um vetor coluna $\phi_{m \times n, 1}(\mathbf{Pv})$, em que seu k -ésimo elemento é

$$\begin{aligned} \phi_k(\mathbf{Pv}) &= \phi_{i_{\mathbf{Av}}}(\mathbf{Av}) \otimes \phi_{i_{\mathbf{Bv}}}(\mathbf{Bv}) \\ &= \begin{cases} \phi_{i_{\mathbf{Av}}}(\mathbf{Av}) \oplus \phi_{i_{\mathbf{Bv}}}(\mathbf{Bv}) & \text{se } \phi_{i_{\mathbf{Av}}}(\mathbf{Av}) \neq \epsilon \wedge \phi_{i_{\mathbf{Bv}}}(\mathbf{Bv}) \neq \epsilon \\ \epsilon & \text{se } \phi_{i_{\mathbf{Av}}}(\mathbf{Av}) = \epsilon \vee \phi_{i_{\mathbf{Bv}}}(\mathbf{Bv}) = \epsilon. \end{cases} \end{aligned}$$

com

$$k = (i_{\mathbf{Av}}, i_{\mathbf{Bv}}) = i_{\mathbf{Av}} + m(i_{\mathbf{Bv}} - 1).$$

O vetor de estado inicial é definido por

$$\begin{aligned} \theta_{1, m \times n}(\mathbf{Pv}) &= \begin{bmatrix} \theta_1(\mathbf{Av}) \otimes \theta_1(\mathbf{Bv}) & \epsilon & \dots & \epsilon \\ \theta_1(\mathbf{Av}) \oplus \theta_1(\mathbf{Bv}) & \epsilon & \dots & \epsilon \end{bmatrix} \\ &= \begin{bmatrix} \theta_1(\mathbf{Av}) \otimes \theta_1(\mathbf{Bv}) & \epsilon & \dots & \epsilon \\ \theta_1(\mathbf{Av}) \oplus \theta_1(\mathbf{Bv}) & \epsilon & \dots & \epsilon \end{bmatrix} \end{aligned}$$

Se $\exists \sigma_{\mathbf{Av}} \notin \Sigma_2$, ou $\exists \sigma_{\mathbf{Bv}} \notin \Sigma_1$, de tal forma que $\Sigma_1 \cap \Sigma_2 \neq \emptyset$, então,

$$\mathbf{Pv} = (\mathbf{Av} || \mathbf{Bv}) \oplus \mathbf{Cv}^\neg,$$

onde

$$\mathbf{Cv}^\neg = \begin{bmatrix} \mathbf{Av}^{-\mathbf{Bv}} \oplus \mathbf{D}_{bv_{1,1}}^{-\mathbf{Av}} & \mathbf{D}_{bv_{1,2}}^{-\mathbf{Av}} & \dots & \mathbf{D}_{bv_{1,n}}^{-\mathbf{Av}} \\ \mathbf{D}_{bv_{2,1}}^{-\mathbf{Av}} & \mathbf{Av}^{-\mathbf{Bv}} \oplus \mathbf{D}_{bv_{2,2}}^{-\mathbf{Av}} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \mathbf{D}_{bv_{n-1,n}}^{-\mathbf{Av}} \\ \mathbf{D}_{bv_{n,1}}^{-\mathbf{Av}} & \mathbf{D}_{bv_{n,2}}^{-\mathbf{Av}} & \dots & \mathbf{Av}^{-\mathbf{Bv}} \oplus \mathbf{D}_{bv_{n,n}}^{-\mathbf{Av}} \end{bmatrix}$$

em que \mathbf{Cv}^\neg tem dimensão $m \times n$, $\mathbf{Av}^{-\mathbf{Bv}}$ é a matriz dos elementos de Σ_1 que não pertencem a Σ_2 e $\mathbf{D}_{bv_{i,j}}^{-\mathbf{Av}}$ é uma matriz diagonal temporizada de dimensão $m \times m$ formada pelos elementos $bv_{i,j}$ que não são definidos em \mathbf{Av} na diagonal principal e ϵ nos demais elementos.

Esta definição é similar à definição da construção do produto síncrono de matrizes de incidência temporizadas de autômatos ($max, +$). Sua diferença reside na utilização das funções temporais ao invés de tempos de vida de eventos constantes. Da mesma forma, deve-se observar que a matriz \mathbf{Cv}^\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.

4.3 Síntese do supervisor

Considerando $\Sigma = \Sigma_c \cup \Sigma_{uc}$, as seguintes definições são necessária para proceder a síntese de supervisores de SEDTs.

Definição 4.6 Dado um alfabeto de eventos Σ , e considerando $\Sigma = \Sigma_c \cup \Sigma_{uc}$, define-se a matriz de incidência temporizada dos eventos não controláveis \mathbf{Av}_{uc} por

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

em que $f(t_\sigma)$ é função do tempo de vida do evento que leva o ATV do estado i para o estado j . Se mais de um evento é definido do estado i para o estado j ,

$$(av_{uc})_{i,j} = \bigoplus_l f(t_{\sigma^l})\sigma^l.$$

Os vetores de estado inicial e de estados marcados são os mesmos da matriz \mathbf{Av} .

Definição 4.7 Uma especificação de comportamento para um ATV, denotada por \mathbf{Ev} , é definida por

$$\mathbf{Ev} = [ev_{i,j}]; ev_{i,j} = \begin{cases} f(t_\sigma)\sigma & \text{se } \exists \sigma \text{ do estado } i \text{ para o estado } j; \\ \epsilon & \text{caso contrário,} \end{cases}$$

em que $f(t_\sigma)$ é o tempo de vida do evento σ que leva o ATV da especificação do estado i para o estado j . Se mais de um evento é definido do estado i para o estado j ,

$$ev_{i,j} = \bigoplus_l f(t_{\sigma^l})\sigma^l.$$

O estado inicial é definido como sendo o estado 1 e é representado pelo vetor linha

$$\theta(\mathbf{Ev}) = [t_0 \quad \epsilon \quad \dots \quad \epsilon],$$

e os estados marcados são representados pelo vetor coluna

$$\phi(\mathbf{Ev}) = [f(t_{f_1}) \quad f(t_{f_2}) \quad \dots \quad f(t_{f_n})]^T.$$

Se um estado k não é marcado, $f(t_{f_k}) = \epsilon$.

Definição 4.8 Um supervisor construído através de uma especificação de comportamento \mathbf{Ev} é definido como sendo a matriz

$$\mathbf{Sv} = [st_{i,j}], st_{i,j} = \begin{cases} f(t_\sigma)\sigma \subset ev_{i,j} & \text{se } f(t_\sigma)\sigma \subset ev_{i,j} \text{ pode ocorrer em } \mathbf{Av}; \\ \epsilon & \text{caso contrário} \end{cases}$$

em que $sv_{i,j} = \epsilon$ para $av_{i,j} \neq \epsilon$, implica em dizer que existe um controle para inibir o evento que se encontra em $av_{i,j}$, para $av_{i,j} = \sigma_c$, ou que o estado (linha) j não é acessível. Se mais de um evento é definido do estado i para o estado j , $sv_{i,j} = \bigoplus_l f(t_{\sigma^l})\sigma^l$. O estado inicial é definido como sendo o estado 1 e é representado pelo vetor linha

$$\theta(\mathbf{Sv}) = [t_0 \quad \epsilon \quad \dots \quad \epsilon]$$

e os estados marcados são representados pelo vetor coluna

$$\phi(\mathbf{Sv}) = [f(t_{f_1}) \quad f(t_{f_2}) \quad \dots \quad f(t_{f_n})]^T.$$

Se um estado k não é marcado, $f(t_{f_k}) = \epsilon$.

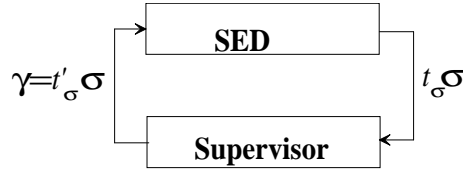


Figura 4.1: Diagrama de um SED temporizado supervisionado

Nessa abordagem o supervisor recebe os eventos σ do SEDT e seus respectivos tempos de vida, e determina a ação de controle definida como uma inibição total ou temporária nos eventos controláveis (ver Figura 4.1).

Para o supervisor, define-se o conjunto de entradas de controle como:

Definição 4.9 *O conjunto de entradas de controle do supervisor temporizado é definido como $\Gamma = \{\gamma_1, \gamma_2, \dots\}$, onde $\gamma_i = f(t'_{\sigma_k})\sigma_k, \forall \sigma_k \in \Sigma(i), k = 1, 2, \dots$, com $f(t'_{\sigma_k}) = f(t_{\sigma_k}), \forall \sigma_k \in \Sigma_{uc}$ e $(f(t'_{\sigma_k}) \geq f(t_{\sigma_k})), \forall \sigma_k \in \Sigma_c$, em que $f(t'_{\sigma_k}) \geq f(t_{\sigma_k})$ significa que o evento σ_k é inibido por um tempo $t = f(t'_{\sigma_k}) - f(t_{\sigma_k})$.*

De acordo com a Definição 4.9, as entradas de controle do supervisor temporizado são $\gamma = f(t'_\sigma)\sigma$ que determinam quais eventos estão habilitados em um estado, e seu tempo de vida correspondente $(f(t'_\sigma) \geq f(t_\sigma))$. Dessa forma, o supervisor inibe temporariamente o evento σ , por um tempo t definido por $t = f(t'_\sigma) - f(t_\sigma)$. Se a inibição é total $f(t'_\sigma) = \epsilon$. Assim, a evolução do sistema temporizado supervisionado é definida considerando a construção da linguagem reconhecida por meio das séries formais ou pela Definição 3.9 da composição síncrona $\mathbf{Sv} \parallel \mathbf{Av}$.

Para contextualizar essa abordagem com a TCS as seguintes denominações são consideradas:

1. Um sub-autômato ATV' apresenta uma estrutura semelhante, mas apenas com parte dos estados ou com parte das transições de ATV , em que as funções de tempo de vida de cada transição do ATV' devem ser tais que seus valores sejam maiores ou iguais aos das funções de tempo de vida da respectiva transição em ATV ;
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 ;
3. Uma submatriz \mathbf{Av}' é formada pelas primeiras m linhas e as primeiras m colunas de \mathbf{Av} e mantém uma correlação elemento a elemento, apresentando para cada evento de \mathbf{Av}' um tempo de vida maior ou igual ao respectivo evento de \mathbf{Av} .

Para realizar a síntese de supervisores de SEDTs, é necessário que \mathbf{Ev} seja uma submatriz de \mathbf{Av} . Se \mathbf{Ev} não é uma submatriz de \mathbf{Av} , deve-se transformar \mathbf{Av} em $\mathbf{Av}^\#$, e construir $\mathbf{Ev}^\#$ que se apresente como submatriz de $\mathbf{Av}^\#$, tal que $L(\mathbf{Av}^\#) = L(\mathbf{Av})$ e $L(\mathbf{Ev}^\#) = L(\mathbf{Ev})$. Assim, considerando que \mathbf{Ev} seja uma submatriz de \mathbf{Av} , ou seja transformada em $\mathbf{Ev}^\#$, sempre a relação $L(\mathbf{Ev}) \subseteq L(\mathbf{Av})$ é verificada. Quando \mathbf{Ev} não é submatriz de \mathbf{Av} , necessita-se do seguinte procedimento:

1. Constrói-se a composição síncrona de \mathbf{Av} com uma matriz qualquer que gera Σ^* , para determinar uma nova matriz $\mathbf{Av}^\#$ que gera $L(\mathbf{Av}^\#) = L(\mathbf{Av})$. Transformando \mathbf{Ev} em uma matriz de incidência \mathbf{Ev}^* que gera Σ^* , através da inclusão de um estado proibido (linha i_e /coluna j_e), denominado de estado de *erro*, utiliza-se \mathbf{Ev}^* para construir a matriz $\mathbf{Av}^\# = \mathbf{Av} \parallel \mathbf{Ev}^*$;

2. Da matriz $\mathbf{Av}^\#$, substituindo todos os elementos das linhas/colunas que formam os pares $(i_{\mathbf{Av}}, i_e)$, $(j_{\mathbf{Av}}, j_e)$ pelo símbolo ϵ , encontra-se a matriz $\mathbf{Ev}^\#$ tal que $L(\mathbf{Ev}^\#) \subset L(\mathbf{Av}^\#) = L(\mathbf{Av})$, com $L(\mathbf{Ev}) = L(\mathbf{Ev}^\#)$ e $\mathbf{Ev}^\#$ sendo uma submatriz de $\mathbf{Av}^\#$;
3. Os tempos de vida dos eventos de \mathbf{Ev} , são resgatados para $\mathbf{Ev}^\#$ (que estão iguais aos tempos de vida dos eventos de \mathbf{Av}) por $f(t_\sigma) \oplus f(t'_\sigma)$, onde $f(t_\sigma)$ é a função de tempo de vida do evento $\sigma \in ev_{i,j}$ e $f(t'_\sigma)$ é a função de tempo de vida do evento $ev_{k,l}^\#$, com k sendo o par $(i_{\mathbf{Av}}, i_{\mathbf{Ev}})$ em que $i_{\mathbf{Ev}} = i$ e l sendo o par $(j_{\mathbf{Av}}, j_{\mathbf{Ev}})$ em que $j_{\mathbf{Ev}} = j$;
4. O vetor $\theta(\mathbf{Ev}^\#)$ é construído através do resgate dos elementos do vetor de estado inicial $\theta(\mathbf{Ev})$, como $\theta_1(\mathbf{Ev}^\#) = \theta_1(\mathbf{Ev}^\#) \oplus \theta_1(\mathbf{Ev})$;
5. O vetor $\phi(\mathbf{Ev}^\#)$ é determinado pelos elementos resgatados do vetor original $\phi(\mathbf{Ev})$, como $\phi_k(\mathbf{Ev}^\#) = \phi_k(\mathbf{Ev}^\#) \otimes \phi_i(\mathbf{Ev})$ em que k é definido como o par $(i_{\mathbf{Av}}, i_{\mathbf{Ev}})$ e $i = i_{\mathbf{Ev}}$, com $\phi_i(\mathbf{Ev}) \neq \epsilon$.

Dessa forma, a matriz $\mathbf{Ev}^\#$ se apresenta como uma submatriz de $\mathbf{Av}^\#$, e conseqüentemente, $L(\mathbf{Ev}^\#) \subset L(\mathbf{Av}^\#) = L(\mathbf{Av})$, e $L(\mathbf{Ev}^\#) = L(\mathbf{Ev})$.

Observação: A matriz que gera Σ^* deve ser considerada com todos os tempos de vida dos eventos $t_\sigma = e$, para que $L(\mathbf{Av}) = L(\mathbf{Av} \parallel \Sigma^*)$. Também, 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.

Para os casos em que há a necessidade da transformação de \mathbf{Ev} em $\mathbf{Ev}^\#$, é apresentado a seguir o algoritmo que transforma uma matriz de incidência temporizada qualquer para que sua linguagem reconhecida seja Σ^* .

Algoritmo 1 Transformação de matrizes de incidência temporizadas \mathbf{Fv} para reconhecimento de Σ^*

1. Tornar os elementos $\theta_1(\mathbf{Fv}) = e$, $\phi_i(\mathbf{Fv}) = e$, sendo o índice o elemento da coluna/linha do vetor respectivo, e $\forall f v_{i,j} = f(t_{\sigma^1})\sigma^1 + \dots + f(t_{\sigma^n})\sigma^n$, $f(t_{\sigma^k}) = e$, $k = 1, \dots, n$;
2. Para cada linha i de \mathbf{Ft} , incluir os auto-laços

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

tal que $\sigma^k \not\subset f v_{i,j}$, $k = 1, 2, \dots, n$;

3. Incluir nos vetores $\theta(\mathbf{Fv})$ e $\phi(\mathbf{Fv})$, o elemento $\theta_{j+1}(\mathbf{Fv}) = \epsilon$ e $\phi_{j+1}(\mathbf{Fv}) = e$.
4. Incluir no final da matriz \mathbf{Fv} 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) $\forall j, j < j_e$, $f v_{i,j_e} = \sigma$, se $\sigma \subset f v_{j,i} \vee \sigma \subset f v_{j,j}$ e $f v_{i,k} \neq \sigma, \forall k, k < j_e$;
- b) $\forall j, j < j_e$, $f v_{i,j_e} = \sigma'$ se $f v_{i,j} \neq \sigma'$

ii) Faça $f v_{i_e, j_e} = \Sigma$.

Exemplo 4.2 Dada a matriz de incidência temporizada sobre o alfabeto $\Sigma = \{\alpha, \beta, \eta, \lambda, \kappa, \mu\}$, e sejam as funções

$$f_1 = 3t$$

e

$$f_2 = 2t^2$$

$$\mathbf{Av} = \begin{bmatrix} \epsilon & 3t\alpha + 3\beta \\ \kappa & 5\eta + \lambda \end{bmatrix}, \quad \theta(\mathbf{Av}) = \begin{bmatrix} 2 & \epsilon \end{bmatrix}, \quad \phi(\mathbf{Av}) = \begin{bmatrix} \epsilon \\ 2t^2 \end{bmatrix}$$

a utilização do algoritmo 1 constrói a matriz

$$\mathbf{Av}^* = \begin{bmatrix} \mu & \alpha + \beta & \kappa + \eta + \lambda \\ \kappa & \mu + \eta + \lambda & \alpha + \beta \\ \epsilon & \epsilon & \alpha + \beta + \kappa + \mu + \eta + \lambda \end{bmatrix}, \quad \theta(\mathbf{Av}) = \begin{bmatrix} e & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{Av}) = \begin{bmatrix} e \\ e \\ e \end{bmatrix}$$

que reconhece Σ^* . Os autômatos que elas representam são vistos na Figura 4.2(a), 4.2(b), respectivamente.

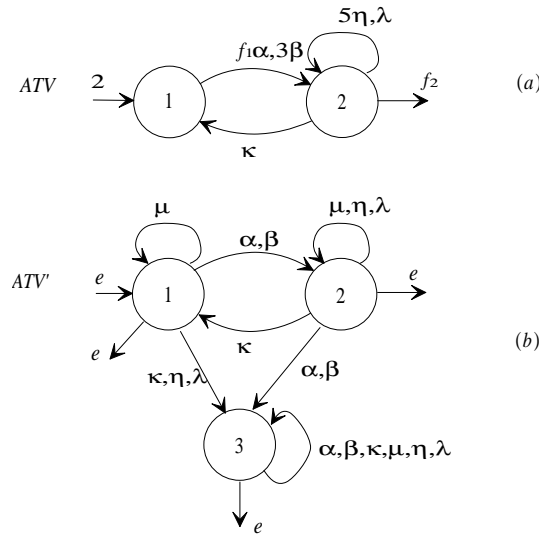


Figura 4.2: Autômatos (a) ATV e (b) ATV' construído com o Algoritmo 1.

Com a transformação de uma matriz de incidência $\mathbf{E}t$ qualquer em uma matriz que gera Σ^* , a construção de $\mathbf{Av}^\#$ e $\mathbf{Ev}^\#$, que satisfazem as condições $L(\mathbf{Ev}^\#) \subset L(\mathbf{Av}^\#)$, $L(\mathbf{Av}^\#) = L(\mathbf{Av})$, $L(\mathbf{Ev}^\#) = L(\mathbf{Ev})$ e $L_m(\mathbf{Ev}^\#) = L_m(\mathbf{Ev})$ é feita utilizando o seguinte algoritmo, em que se consideram as dimensões de \mathbf{Ev} e \mathbf{Av} iguais a n e m , respectivamente.

Algoritmo 2 Transformação da matriz de incidência \mathbf{Ev} em $\mathbf{Ev}^\#$

1. Transforme \mathbf{Ev} em \mathbf{Ev}^* ;
2. Construa $\mathbf{Av}^\# = \mathbf{Av} \parallel \mathbf{Ev}^*$;
3. Para i e $j = 1$ até $n \times m$, faça:
 - a) Se $av_{i,j}^\# \neq \epsilon$, faça $ev_{i,j}^\# = av_{i,j}^\#$.

4. Faça $k = 1$ até $n \times m$:

a) Se $\phi_k(\mathbf{Av}^\#) \neq \epsilon$, faça $i_{\mathbf{Ev}} = ((k - (k \bmod m)) / m) + 1$, e $\phi_k(\mathbf{Ev}^\#) = \phi_k(\mathbf{Ev}^\#) \otimes \phi_{i_{\mathbf{Ev}}}(\mathbf{Ev})$;

5. Faça $\theta_1(\mathbf{Ev}^\#) = \theta_1(\mathbf{Ev}^\#) \oplus \theta_1(\mathbf{Ev})$.

Neste algoritmo, \bmod é o operador de módulo que retorna o resto da divisão k/m . O **passo 4** resgata as funções de atraso final dos estados marcados e o **passo 5** resgata a função de atraso inicial da especificação de comportamento \mathbf{Ev} , para $\mathbf{Ev}^\#$, de forma a garantir que $L(\mathbf{Ev}^\#) = L(\mathbf{Ev})$ e $L_m(\mathbf{Ev}^\#) = L_m(\mathbf{Ev})$.

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

Exemplo 4.3 Considere a matriz

$$\mathbf{Av} = \begin{bmatrix} \epsilon & 2t\alpha_1 & 3\alpha_2 & \epsilon \\ 2\beta_1 & \epsilon & \epsilon & 3\alpha_2 \\ 4\beta_2 & \epsilon & \epsilon & 2t^2\alpha_1 \\ \epsilon & 4\beta_2 & (2t-1)\beta_1 & \epsilon \end{bmatrix}, \quad \theta(\mathbf{Av}) = [3 \quad \epsilon \quad \epsilon \quad \epsilon], \quad \phi(\mathbf{Av}) = \begin{bmatrix} e \\ \epsilon \\ \epsilon \\ 3t \end{bmatrix}$$

e a especificação

$$\mathbf{Ev} = \begin{bmatrix} 2t^2\alpha_1 + \beta_2 & 5\beta_1 \\ 4\alpha_2 & \alpha_1 + \beta_2 \end{bmatrix}, \quad \theta(\mathbf{Ev}) = [4 \quad \epsilon], \quad \phi(\mathbf{Ev}) = \begin{bmatrix} 2 + t^2 \\ \epsilon \end{bmatrix}$$

onde $\Sigma_{uc} = \{\beta_1, \beta_2\}$. Para construir a matriz $\mathbf{Ev}^\#$, tal que $L(\mathbf{Ev}^\#) \subset L(\mathbf{Av}^\#) = L(\mathbf{Av})$, constrói-se a matriz \mathbf{Ev}^* que gera Σ^* , a qual é dada por

$$\mathbf{Ev}^* = \begin{bmatrix} \alpha_1 + \beta_2 & \beta_1 & \alpha_2 \\ \alpha_2 & \alpha_1 + \beta_2 & \beta_1 \\ \epsilon & \epsilon & \alpha_1 + \beta_1 + \alpha_2 + \beta_2 \end{bmatrix}, \quad \theta(\mathbf{Ev}) = [e \quad \epsilon \quad \epsilon], \quad \phi(\mathbf{Ev}) = \begin{bmatrix} e \\ e \\ e \end{bmatrix}$$

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

$$\mathbf{Av}^\# = \begin{bmatrix} \epsilon & 2t\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 3\alpha_2 & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 3\alpha_2 \\ 4\beta_2 & \epsilon & \epsilon & 2t^2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 4\beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & (2t-1)\beta_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 3\alpha_2 & \epsilon & \epsilon & 2t\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 3\alpha_2 & \epsilon & \epsilon & \epsilon & 2\beta_1 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & \epsilon & \epsilon & 2t^2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & \epsilon & \epsilon & \epsilon & (2t-1)\beta_1 & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 2t\alpha_1 & 3\alpha_2 & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 & \epsilon & \epsilon & 3\alpha_2 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & \epsilon & \epsilon & 2t^2\alpha_1 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & (2t-1)\beta_1 & \epsilon \end{bmatrix}$$

$$\theta(\mathbf{Av}^\#) = [3 \quad \epsilon \quad \epsilon \quad \epsilon \quad \epsilon \quad \epsilon \quad \epsilon \quad \epsilon \quad \epsilon \quad \epsilon \quad \epsilon \quad \epsilon],$$

$$\phi(\mathbf{Av}^\#) = [e \quad \epsilon \quad \epsilon \quad 3t \quad e \quad \epsilon \quad \epsilon \quad 3t \quad e \quad \epsilon \quad \epsilon \quad 3t]^T$$

em que as linhas/colunas formadas por uma linha/coluna de \mathbf{Av} com a linha/coluna de erro de \mathbf{Ev}^* são as quatro últimas. Fazendo $ev_{i,j}^\# = av_{i,j}^\#$, nos elementos $\forall i, j$, tal que $i, j \leq ((n-1) \times m)$ (igual a fazer $\mathbf{Ev}^\# = \mathbf{Av}^\#$ e tornar todos os elementos que satisfazem os pares $(i_{\mathbf{Av}}, i_e)$, $(j_{\mathbf{Av}}, j_e)$ iguais a ϵ) e resgatando os tempos de vida dos eventos de \mathbf{Ev} , encontra-se

$$\mathbf{Ev}^\# = \begin{bmatrix} \epsilon & 2t^2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 5\beta_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ 4\beta_2 & \epsilon & \epsilon & 2t^2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 4\beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & (5 \oplus (2t-1))\beta_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 4\alpha_2 & \epsilon & \epsilon & 2t\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 4\alpha_2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & \epsilon & \epsilon & 2t^2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}$$

$$\theta(\mathbf{Ev}^\#) = [4 \ \epsilon \ \epsilon \ \epsilon \ \epsilon \ \epsilon \ \epsilon \ \epsilon \ \epsilon \ \epsilon \ \epsilon \ \epsilon],$$

$$\phi(\mathbf{Ev}^\#) = [2 + t^2 \ \epsilon \ \epsilon \ \epsilon \ \epsilon \ \epsilon \ \epsilon \ \epsilon \ \epsilon \ \epsilon \ \epsilon \ \epsilon]^T$$

Observe que apenas $\phi_1(\mathbf{Ev}^\#) \neq \epsilon$, desde que satisfaz $\phi_1(\mathbf{Ev}^\#) = \phi_1(\mathbf{Av}) \otimes \phi_1(\mathbf{Ev})$. A matriz $\mathbf{Ev}^\#$ é uma submatriz da matriz $\mathbf{Av}^\#$, com linguagem $L(\mathbf{Ev}^\#) = L(\mathbf{Ev})$ e linguagem marcada $L_m(\mathbf{Ev}^\#) = L_m(\mathbf{Ev})$, tal que $L(\mathbf{Ev}^\#) \subset L(\mathbf{Av}^\#)$.

Observação: Quando todas as funções de tempo de vida dos eventos são constantes, encontra-se a formulação da construção de \mathbf{At} em $\mathbf{At}^\#$ e \mathbf{Et} em $\mathbf{Et}^\#$. Caso todas estes valores sejam iguais ao elemento unidade 'e', encontra-se a formulação para a construção de \mathbf{A} em $\mathbf{A}^\#$ e \mathbf{E} em $\mathbf{E}^\#$.

Para a síntese do supervisor \mathbf{Sv} , as seguintes operações são necessárias:

Definição 4.10 O operador *ACES* é definido como

$$ACES(\mathbf{Av}) = \mathbf{Bv}, bv_{i,j} = \begin{cases} av_{i,j} & \text{se } i \text{ é acessível} \\ \epsilon & \text{caso contrário.} \end{cases}$$

A operação $ACES(\mathbf{Av})$ elimina os elementos $av_{i,j}$ de uma linha i não acessível.

Definição 4.11 O operador *COACES* é definido por

$$COACES(\mathbf{Av}) = \mathbf{Bv},$$

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

A operação $COACES(\mathbf{Av})$ elimina os elementos que levam para linhas não coacessíveis, tornando-as não acessíveis.

Definição 4.12 O operador *TRIM*: é definido por $TRIM(\mathbf{Av}) = ACES(COACES(\mathbf{Av})) = \mathbf{Bv}$.

O operador *TRIM* aplicado a uma matriz de incidência \mathbf{Av} devolve uma matriz \mathbf{Bv} 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 4.13 Dadas duas matrizes temporizadas $\mathbf{Av} = [av_{i,j}]$ e $\mathbf{Bv} = [bv_{i,j}]$, define-se o operador \leq por

$$\mathbf{Av} \leq \mathbf{Bv} \Rightarrow L(\mathbf{Av}) \subseteq L(\mathbf{Bv}).$$

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

$$\mathbf{Av} \geq \mathbf{Bv} \Rightarrow L(\mathbf{Av}) \supseteq L(\mathbf{Bv}),$$

$$\mathbf{Av} \triangleleft \mathbf{Bv} \Rightarrow L(\mathbf{Av}) \subset L(\mathbf{Bv}),$$

$$\mathbf{Av} \triangleright \mathbf{Bv} \Rightarrow L(\mathbf{Av}) \supset L(\mathbf{Bv}).$$

A condição de $\mathbf{Av} \leq \mathbf{Bv}$, implica em que toda seqüência de \mathbf{Av} , que é formada por eventos com seus respectivos tempos de vida deve ser executada em um tempo maior ou igual a execução da mesma seqüência em \mathbf{Bv} . Em termos de SEDTs, isto quer dizer que a linguagem que contém uma outra, é executada em um tempo menor [14]. Então, $L(\mathbf{Av}) \subseteq L(\mathbf{Bv})$ significa que todas as seqüências da linguagem de \mathbf{Av} são encontradas na linguagem de \mathbf{Bv} , e estas seqüências satisfazem $(y|_{s_{\mathbf{Av}}}) \geq (y|_{s_{\mathbf{Bv}}})$, onde cada evento tem um tempo de vida satisfazendo $t_{\sigma_{i_{\mathbf{Av}}}} \geq t_{\sigma_{i_{\mathbf{Bv}}}}$.

Com estas definições, tem-se que a especificação de comportamento \mathbf{Ev} deve satisfazer a seguinte condição:

Definição 4.14 Uma especificação de comportamento \mathbf{Ev} é válida para a matriz de incidência \mathbf{Av} se $\mathbf{Ev} \neq [\epsilon]$ e

$$\begin{aligned} & \forall ev_{i,j}, f(t_\sigma) \geq f(t'_\sigma), f(t'_\sigma) \text{ função de tempo de vida de } \sigma \subset av_{i,j} \\ & \theta_1(\mathbf{Ev}) \geq \theta_1(\mathbf{Av}) \quad \text{e} \\ & \phi_i(\mathbf{Ev}) \geq \phi_i(\mathbf{Av}) \vee \phi_i(\mathbf{Ev}) = \epsilon, \forall \phi_i(\mathbf{Av}) \neq \epsilon, \forall i = 1 \text{ até } N, \end{aligned}$$

onde $[\epsilon]$ é a matriz nula, onde todos os seus elementos são ϵ e se $\forall i, j, \sigma \subset ev_{i,j}, \sigma \in \Sigma$.

Assim, sendo \mathbf{Ev} válida, define-se a condição de controlabilidade.

Definição 4.15 Dada uma especificação de comportamento \mathbf{Ev} válida, com $ACES(\mathbf{Ev}) = \mathbf{Ev}$, e a matriz de incidência temporizada \mathbf{Av} que representa o ATV, a condição de controlabilidade para \mathbf{Ev} é definida por

$$ACES(\mathbf{Ev} \oplus \mathbf{Av}_{uc}) = \mathbf{Ev}.$$

Deve-se ver que a igualdade avalia os vetores de estado inicial θ e de estados marcados ϕ . Disso, tem-se:

Lema 4.1 Um supervisor \mathbf{Sv} para uma matriz de incidência \mathbf{Av} é definido por \mathbf{Ev} se e só se

$$ACES(\mathbf{Ev} \oplus \mathbf{Av}_{uc}) = \mathbf{Ev}.$$

Prova: A prova desse Lema é igual à apresentada em Costa [32] para \mathbf{At} e \mathbf{Et} .

Se a especificação \mathbf{Ev} define uma linguagem que satisfaz a condição de controlabilidade, o supervisor é a sua componente *trim*.

Corolário 4.1 Dada uma especificação de comportamento \mathbf{Ev} válida e uma matriz de incidência temporizada *trim* \mathbf{Av} que é o modelo de um SEDT, $\mathbf{Sv} = \text{TRIM}(\mathbf{Ev})$ se e só se

$$\text{ACES}(\mathbf{Ev} \oplus \mathbf{Av}_{uc}) = \mathbf{Ev}.$$

Prova: A prova desse Corolário é igual à apresentada em Costa [32] para \mathbf{At} e \mathbf{Et} .

Exemplo 4.4 Para o ATV mostrado na Figura 4.3, onde $\Sigma = \{\alpha, \beta, \kappa, \mu\}$, $\Sigma_{uc} = \{\kappa\}$ e $\Sigma_c = \{\alpha, \beta, \mu\}$, sua representação matricial é dada por

$$\mathbf{Av} = \begin{bmatrix} \epsilon & (t^2 - 2t)\beta & 3t\mu \\ (t - 2)\alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & (t - 1)\kappa \end{bmatrix}, \quad \theta(\mathbf{Av}) = \begin{bmatrix} 2 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{Av}) = \begin{bmatrix} 3t & \epsilon & 1 \end{bmatrix}^T$$

Considerando a especificação de comportamento

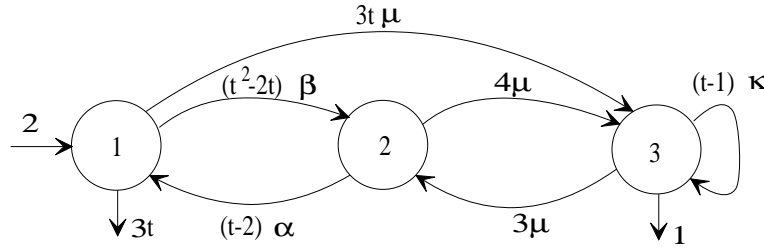


Figura 4.3: Autômato para exemplificar a síntese do supervisor

$$\mathbf{Ev} = \begin{bmatrix} \epsilon & (t^2 - 2t)\beta & (3t + 1)\mu \\ (t^2 - 2)\alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & (t - 1)\kappa \end{bmatrix}, \quad \theta(\mathbf{Ev}) = \begin{bmatrix} 4 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{Ev}) = \begin{bmatrix} 5t & \epsilon & 2 \end{bmatrix}^T$$

mostrada na Figura 4.4(b), tem-se que $\text{ACES}(\mathbf{Ev} \oplus \mathbf{Av}_{uc}) = \mathbf{Ev}$, em que os vetores de estados são $\theta_1(\mathbf{Ev}) > \theta_1(\mathbf{Av})$ (θ_2 e θ_3 são iguais) e $\phi_1(\mathbf{Ev}) = 5t$ e $\phi_3(\mathbf{Ev}) = 2$, que garante que \mathbf{Ev} é controlável. Como \mathbf{Ev} é trim, o supervisor é dado por

$$\mathbf{Sv} = \text{TRIM}(\mathbf{Ev}) = \mathbf{Ev}.$$

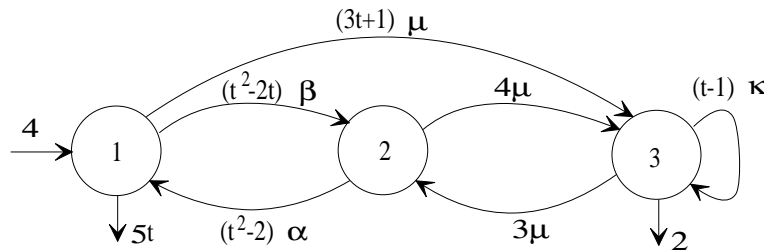


Figura 4.4: Especificação controlável

Para uma dada especificação de comportamento \mathbf{Ev} , se $ACES(\mathbf{Ev} \oplus \mathbf{Av}_{uc}) \triangleright \mathbf{Ev}$, então, é necessário encontrar a suprema sublinguagem controlável. A solução desse problema pode ser encontrada utilizando a seguinte:

Definição 4.16 Dada a matriz de incidência \mathbf{Ev} , define-se a matriz de caminhos $\mathbf{Bv}_{uc}^n = \mathbf{Ev} \otimes (\mathbf{Av}_{uc})^{n-1}$, em que o primeiro elemento de cada seqüência é um elemento de \mathbf{Ev} , e os demais são elementos pertencentes a \mathbf{Av}_{uc} . O vetor de estados marcados ϕ é definido por

$$\begin{aligned} \phi_i(\mathbf{Bv}_{uc}^n) &= \phi_i(\mathbf{Ev}) \otimes \phi_i(\mathbf{Av}) \\ &= \begin{cases} \phi_i(\mathbf{Ev}) \oplus \phi_i(\mathbf{Av}), & \text{se } \phi_i(\mathbf{Av}) \neq \epsilon \\ \epsilon & \text{caso contrário,} \end{cases} \end{aligned} \quad (4.4)$$

e o vetor de estado inicial é $\theta(\mathbf{Bv}_{uc}^n) = \theta(\mathbf{Av}) = \theta(\mathbf{Ev})$.

Nessa matriz, um determinado evento σ_{uc} de $(av_{uc})_{i,j}$, aparece como último evento nas seqüências dos termos de $(bv_{uc}^n)_{k,j}$. Estes termos de \mathbf{Bv}_{uc}^n são antecidos pelos elementos de $(bv_{uc}^{n-1})_{k,i}$. Entretanto, devido à complexidade da construção da matriz \mathbf{Bv}_{uc}^n , devido às funções de tempos de vida, e como é de interesse encontrar apenas os caminhos que geram ocorrências de eventos não pertencentes à especificação \mathbf{Ev} , pode-se utilizar a matriz de caminhos não temporizada \mathbf{B}_{uc}^n para isto, que satisfaz as mesmas condições sem a necessidade de cálculo dos tempos dos caminhos. Dessa forma, tem-se:

Teorema 4.1 Dada \mathbf{Ev} válida e a matriz \mathbf{Av}_{uc} , se $ACES(\mathbf{Ev} \oplus \mathbf{Av}_{uc}) \triangleright \mathbf{Ev}$, então a sup $C(L)$ será determinada recursivamente por:

1. Para $n = 1$, $\mathbf{Sv}^1 = \mathbf{Ev}$.
2. Para $n = n + 1$, enquanto $(n \leq N) \wedge \exists \sigma_{uc} \notin \mathbf{Ev}$ então

$$\begin{aligned} \mathbf{B}_{uc}^n &= \mathbf{E} \otimes (\mathbf{A}_{uc})^{n-1} \\ \mathbf{Sv}^n &= [sv_{ij}^n], sv_{ij}^n = \begin{cases} sv_{i,j}^{n-1} & \text{se } \sigma^1 \sigma_{uc}^2 \dots \sigma_{uc}^n \in \mathbf{B}_{uc}^n \wedge \sigma_{uc}^n \in \mathbf{Ev}; \\ \epsilon & \text{se } \sigma_{uc}^n \notin \mathbf{Ev} \wedge \sigma^1 \in \Sigma_c \end{cases} \\ \mathbf{Sv}^n &= TRIM(\mathbf{Sv}^n) \end{aligned}$$

onde σ_{uc}^n é o n -ésimo evento da seqüência de um termo de \mathbf{B}_{uc}^n , que pode ser $\sigma_{uc} \notin \mathbf{Ev}$; \mathbf{E} e \mathbf{A}_{uc} são as matrizes \mathbf{Ev} e \mathbf{Av}_{uc} com todos os tempos de vida reduzidos ao elemento identidade 'e', respectivamente.

3. Se $(n > N) \wedge (\exists \sigma_{uc} \notin \mathbf{Ev} \text{ em } \mathbf{Sv}^n)$, então $\mathbf{Sv} = [\epsilon]$.

Prova: A prova desse Teorema é igual à apresentada em Costa [32] para \mathbf{At} e \mathbf{Et} .

Exemplo 4.5 Para o ATV da Figura 4.3, seja a especificação

$$\mathbf{Ev} = \begin{bmatrix} \epsilon & (t^2 - 1)\beta & 3t\mu \\ (t - 2)\alpha & \epsilon & 5\mu \\ \epsilon & 3t\mu & \epsilon \end{bmatrix}, \quad \theta(\mathbf{Ev}) = [4 \quad \epsilon \quad \epsilon], \quad \phi(\mathbf{Ev}) = \begin{bmatrix} 5t \\ \epsilon \\ \epsilon \end{bmatrix}$$

a qual está apresentada na Figura 4.5(a). O teste de controlabilidade dá $ACES(\mathbf{Ev} \oplus \mathbf{Av}_{uc}) \triangleright$

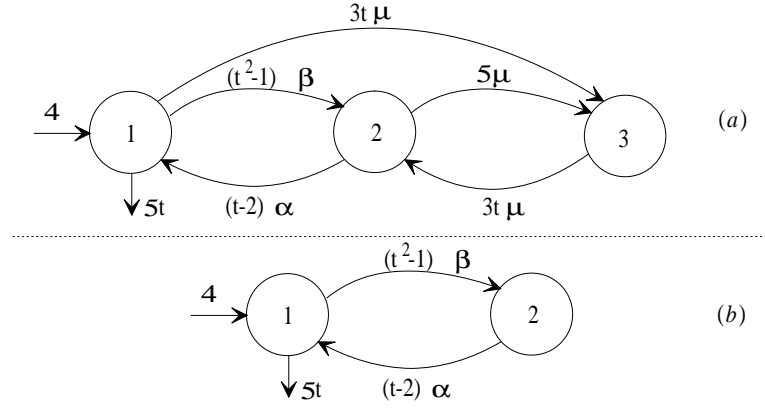
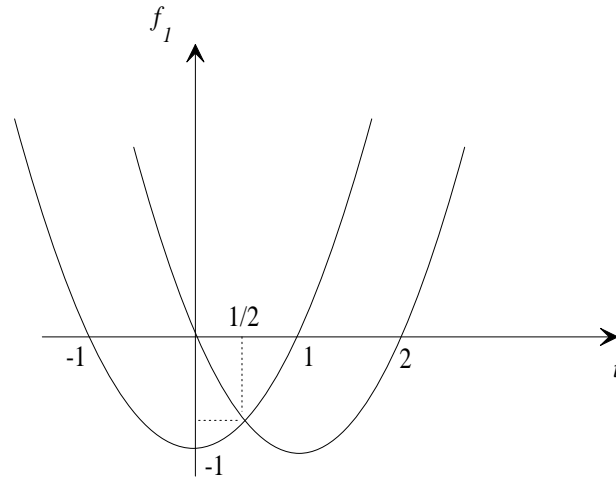


Figura 4.5: (a) Especificação não controlável e (b) Supervisor

Figura 4.6: Gráfico das funções temporais do evento β que aponta do estado 1 para o estado 2. O máximo entre as duas funções para a composição síncrona do supervisor com o *ATV* é definido pela função temporal da especificação.

Ev. Assim, construindo a composição síncrona, encontram-se os novos tempos de vida do evento β (em $ev_{1,2}$) dado por

$$f_1 = \max((t^2 - 2t), (t^2 - 1)) = \begin{cases} (t^2 - 2t), & \text{se } 0 < t \leq \frac{1}{2} \\ (t^2 - 1), & \text{se } t > \frac{1}{2}, \end{cases}$$

que é vista na Figura 4.6. Assim, fazendo $\mathbf{Sv}^1 = \mathbf{Ev}'$. Calculando \mathbf{B}_{uc}^2 , tem-se

$$\mathbf{B}_{uc}^2 = \mathbf{E}' \otimes \mathbf{A}_{uc} = \begin{bmatrix} \epsilon & \epsilon & \mu\kappa \\ \epsilon & \epsilon & \mu\kappa \\ \epsilon & \epsilon & \epsilon \end{bmatrix}$$

onde são encontrados os termos $(b_{uc}^2)_{1,3} = \mu\kappa$ e $(b_{uc}^2)_{2,3} = \mu\kappa$, que contêm o evento controlável μ seguido do evento não controlável κ , que não pertence à especificação. Como μ é controlável, então fazendo $sv_{1,3}^2 = sv_{2,3}^2 = \epsilon$, elimina-se o estado 3. Logo, tornando-se o estado 3 não acessível, $sv_{3,2}^2 = sv_{3,3}^2 = \epsilon$. Para este caso, deve-se observar que em nenhum momento da evolução síncrona do *ATV* com este supervisor, o tempo anterior ao estado 1 é menor que 4 unidades de tempo. Logo, f_1 torna-se igual à

$$f_1 = (t^2 - 1).$$

Assim, o supervisor é determinado por

$$\mathbf{Sv} = \text{TRIM}(\mathbf{Sv}^2) = \begin{bmatrix} \epsilon & (t^2 - 1)\beta & \epsilon \\ (t - 2)\alpha & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \theta(\mathbf{Sv}) = \begin{bmatrix} 4 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{Sv}) = \begin{bmatrix} 5t & \epsilon & \epsilon \end{bmatrix}^T$$

o qual está mostrado na Figura 4.5(b).

4.4 Supervisor Temporalmente Adaptativo

Para ser iniciada a discussão sobre a construção de um supervisor temporalmente adaptativo, é de importância básica contextualizar o trabalho.

Quando um SED temporizado é modelado por um autômato $(max, +)$, considera-se que os tempos de habilitação/ocorrência dos eventos têm um valor mínimo, denominado de tempo de vida do evento. Neste caso, os SEDs temporizados modelados por este paradigma apresentam um valor constante para habilitação/ocorrência dos eventos, os quais podem ser maiores, devido à estes valores serem limites inferiores. Entretanto, em nenhuma especificidade, este tempo pode ser menor para alguma execução do autômato. Por outro lado, um supervisor para um SED temporizado modelado por um autômato $(max, +)$ também mantém as condições únicas de só poderem atrasar a execução de eventos, aumentando seus tempos de vida para uma determinada execução. Considerando que em alguns SEDs temporizados seja necessário modificar os tempos de vida dos eventos, para cada execução do autômato, mesmo sendo a mesma especificação lógica, torna-se necessário definir novas especificações de comportamento (com os tempos de vida diferentes) e gerar um novo supervisor para cada uma delas. Em outros termos, é uma tarefa inviável se estas modificações são constantes para cada pequena tarefa realizada. Por exemplo, ligar e desligar uma máquina, considerando que cada vez que se desliga, é necessário ter um tempo maior para religá-la e o tempo para desligar é menor. Tal situação pode ser formulada em termos de funções nos arcos que definam os novos tempos de vida para cada execução, o que pode ser representada por um ATV. Sendo assim, define-se um supervisor temporalmente adaptativo como a seguir:

Definição 4.17 *Um supervisor temporalmente adaptativo \mathbf{S} para um SED temporizado G com linguagem $L(G)$, é um supervisor cuja dinâmica é descrita por uma linguagem temporizada $L(\mathbf{S})$, tal que,*

$$\begin{aligned} \forall s \in \Sigma^* | t_s s \in L(G), \\ \text{se } \exists t'_s s, t''_s s, \dots \in L(\mathbf{S}) \Rightarrow t'_s \geq t_s, t''_s \geq t_s, \dots \end{aligned} \quad (4.5)$$

em que t_s é o tempo de vida da palavra s em $L(G)$, e $t'_s \neq t''_s \neq \dots$ são os tempos de vida da palavra s em $L(\mathbf{S})$, e $\forall t_s s \in L(G) | t_s s = t_{\sigma^1} \sigma^1 \dots t_{\sigma^n} \sigma^n$, com $t_s = \bigoplus_{i=1}^n t_{\sigma^i}$, se $\exists t'_s s, t''_s s, \dots \in L(\mathbf{S})$, então

$$\begin{aligned} t'_s &= \bigoplus_{i=1}^n t'_{\sigma^i}, \text{ com } t'_{\sigma^i} \geq t_{\sigma^i}, \\ t''_s &= \bigoplus_{i=1}^n t''_{\sigma^i}, \text{ com } t''_{\sigma^i} \geq t_{\sigma^i}, \\ &\vdots \end{aligned} \quad (4.6)$$

em que t_{σ^i} é o tempo de vida do evento σ da palavra s em $L(G)$, e $t'_\sigma \neq t''_\sigma \neq \dots$ são os tempos de vida dos eventos da palavra s em $L(\mathbf{S})$.

De acordo com essa definição, o supervisor temporalmente adaptativo pode conter algumas das seqüências de eventos do SED temporizado, entretanto, todas estas seqüências têm seus tempos de vida maiores ou iguais ao tempo de vida da palavra s do SED temporizado. Em outros termos, para

cada seqüência do SED temporizado, o supervisor pode gerar a mesma seqüência com diferentes tempos de execução.

Naturalmente, a representação de um supervisor temporalmente adaptativo pode ser feita por um autômato $(max, +)$. Contudo, esta representação pode se tornar com um número infinito de estados, dependendo da linguagem temporizada a ser gerada. Dessa forma, este problema pode ser evitado utilizando a formalização do ATV para especificar o comportamento do sistema e, conseqüentemente, sintetizar o supervisor.

A formalização para a construção do supervisor temporalmente adaptativo para um SED temporizado segue a mesma formulação de [36, 47, 37] e apresentada anteriormente, para supervisão do SED temporizado modelado por um ATV. Assim, dado um modelo de um SED temporizado, como normalmente é definido, ou seja, sem variações temporais nos tempos de vida dos eventos (modelo em autômato $(max, +)$), a especificação da tarefa a ser realizada deve descrever com clareza o comportamento do sistema a ser controlado, considerando as variações temporais a serem respeitadas em sua execução. Ou seja, deve-se construir as funções temporais dos eventos de forma que satisfaça os requisitos do comportamento desejado.

Definida a especificação de comportamento, a síntese do supervisor prevê as condições para evitar bloqueios e situações não desejadas na execução do sistema, inclusive avaliando as condições para que a linguagem supervisionada $L(S/G)$ não defina que tempos de vida de eventos sejam reduzidos para aquém do tempo de vida do modelo do SED temporizado [36, 47, 37]. Considerando esta formulação, tem-se a seguinte proposição:

Proposição 4.1 *Dado um SED temporizado modelado por um autômato $(max, +)$ e uma especificação de comportamento definida por um ATV, um supervisor S que seja adaptativo em relação aos tempos de vida dos eventos do SED temporizado pode ser sintetizado.*

Demonstração: *Obviamente, pelas condições apresentadas no Lema 22, Corolário 23 e Teorema 26 de [36], garante-se a construção de um supervisor para o SED temporizado. Por outro lado, considerando que estas condições são satisfeitas, então também as condições (4.5) e (4.6) da Definição 4.17 são satisfeitas, o que garante que um supervisor sintetizado é temporalmente adaptativo.*

A síntese do supervisor temporalmente adaptativo segue a mesma formulação do algoritmo de síntese apresentado em [36, 37], utilizando o modelo comumente definido com tempos de vida constantes. Se para um SED temporizado modelado por um autômato $(max, +)$ e uma especificação de comportamento definida um supervisor é factível, este é garantido ser temporalmente adaptativo de acordo com a Proposição 4.1. Neste caso, desde que a especificação de comportamento é definida ou transformada em uma submatriz da matriz do modelo do SED temporizado, o supervisor será também uma submatriz do modelo, e garante que o sistema supervisionado varie os tempos de vida dos eventos em qualquer execução.

4.5 Exemplo

Nessa seção é apresentado um exemplo ilustrativo de supervisor temporalmente adaptativo, cuja síntese é baseada no algoritmo apresentado em [36, 37], o qual pode ser visto em [50, 51].

Exemplo 4.6 *Considere um sistema que tem uma furadeira e um grande número de peças a serem perfuradas, como visto na Figura 4.7. Para uma peça ser perfurada, é necessário que a peça espere 2 unidades de tempo. Quando uma peça é perfurada (evento α), o sistema retorna ao estado*

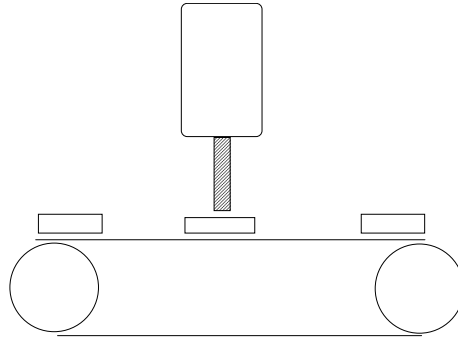


Figura 4.7: Furadeira.

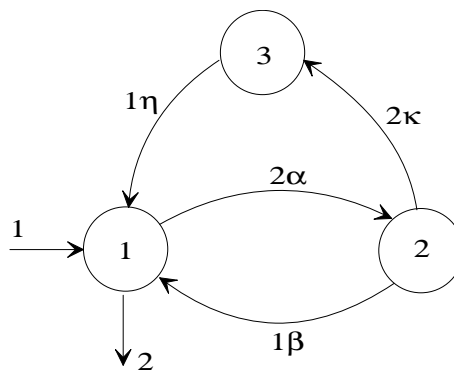
inicial (evento β), gastando 1 unidade de tempo e podendo iniciar uma nova perfuração de peça, ou finalizar o trabalho (o que gasta mais 2 unidades de tempo). Para cada peça perfurada, a temperatura da broca da furadeira é elevada de 11°C . Para uma nova peça na espera para ser perfurada, esta temperatura reduz 1°C . Sempre que a broca da furadeira atinge uma temperatura superior a 100°C , ela pode quebrar e tem de ser levada para ser trocada, o que gasta 2 unidades de tempo (evento κ). Após a broca da furadeira ser trocada, o sistema retorna ao estado inicial (evento η), gastando mais 1 unidade de tempo. Como geralmente é construído o modelo em termos de um autômato $(\max, +)$, não são consideradas estas condições de variação de temperatura, e o modelo é definido como apresentado na Figura 4.8, que tem como representação matricial:

$$\mathbf{A} = \begin{bmatrix} \epsilon & 2\alpha & \epsilon \\ 1\beta & \epsilon & 2\kappa \\ 1\eta & \epsilon & \epsilon \end{bmatrix}$$

$$\theta(\mathbf{A}) = \begin{bmatrix} 1 & \epsilon & \epsilon \end{bmatrix}$$

$$\phi(\mathbf{A}) = \begin{bmatrix} 2 \\ \epsilon \\ \epsilon \end{bmatrix}$$

Neste modelo, deve-se observar que a temperatura que leva a fragilidade da broca para rompimento

Figura 4.8: Modelo em autômato $(\max, +)$ da furadeira.

(100°C), é alcançada quando são perfuradas 9 peças. Considerando agora, que se tem acesso às informações citadas e que a temperatura inicial da broca da furadeira é 10°C , é necessário definir uma especificação de comportamento que garanta que a broca da furadeira tenha um tempo de vida superior para perfurar mais peças sem que haja sua quebra por atingir a temperatura de rompimento

(acima dos 100°C). Dessa forma, descreve-se o comportamento desejado de acordo com a seguinte matriz de incidência:

$$\mathbf{E} = \begin{bmatrix} \epsilon & f_1\alpha & \epsilon \\ f_2\beta & \epsilon & f_3\kappa \\ f_4\eta & \epsilon & \epsilon \end{bmatrix}$$

$$\theta(\mathbf{E}) = \begin{bmatrix} 1 & \epsilon & \epsilon \end{bmatrix}$$

$$\phi(\mathbf{E}) = \begin{bmatrix} 2 \\ \epsilon \\ \epsilon \end{bmatrix}$$

e cujo ATV é visto na Figura 4.9, em que as funções de tempo de vida dos eventos são dadas por

$$\begin{aligned} f_1 &= t + 2; \\ f_2 &= t; \\ f_3 &= 3t; \\ f_4 &= 1. \end{aligned}$$

Essas funções garantem que: (1) para uma peça ser perfurada, haja um tempo de espera sempre

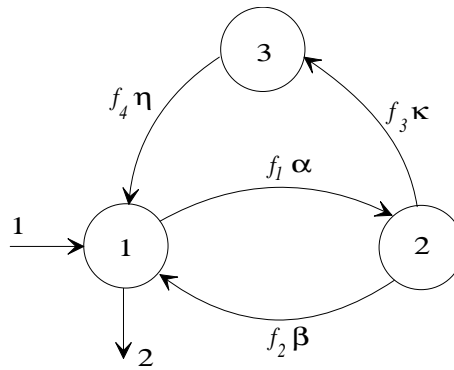


Figura 4.9: Modelo em ATV da especificação de comportamento.

maior, de forma a garantir uma queda na temperatura da broca da furadeira; (2) que o retorno ao estado inicial também ajude a garantir esta queda e, (3) que uma peça quebrada gaste um tempo três vezes maior que o último passado, de forma a esfriar a peça antes de ser trocada. O tempo de retorno às atividades é o mesmo valor constante 1, desde que com uma nova broca, o sistema reinicia seu processamento normal. Para este caso, deve-se observar que as funções são sempre maiores que os valores dos tempos de vida dos eventos originais do modelo. Assim, o supervisor sintetizado tem exatamente a mesma estrutura da especificação de comportamento dada e garante que o sistema se adapte temporalmente para que a temperatura não se eleve a ponto de atingir a temperatura de rompimento (100°C). Deve-se observar que para este supervisor, os valores dos tempos de vida dos eventos α e β , bem como as temperaturas iniciais T_i (antes de iniciar a perfuração) e finais T_f

(após a perfuração) da broca a cada perfuração são dadas de acordo com a seguinte tabela:

α	β	T_i	T_f
2s	2s	10°C	21°C
4s	4s	19°C	30°C
6s	6s	26,6°C	37,6°C
8s	8s	33,2°C	44,2°C
10s	10s	38,2°C	49,2°C
12s	12s	42,2°C	53,2°C
14s	14s	45,6°C	55,6°C
16s	16s	45,6°C	56,6°C
18s	18s	45,4°C	56,4°C

⋮

Nesta tabela, também observa-se que após atingir 16s de tempo para retorno ao estado inicial, a temperatura da broca tende a cair cada vez mais. Dessa forma, para não haver um atraso cada vez mais superior, a função f_2 pode ser definida por

$$f_2 = \begin{cases} t, & \text{se } f_1 \leq 16 \\ 16, & \text{caso contrário.} \end{cases}$$

o que garante da mesma maneira um supervisor temporalmente adaptativo e estável para o sistema, e que aumenta o tempo de vida da broca da furadeira, permitindo utilizá-la para o processamento de mais peças (embora o tempo gasto para o processamento da mesma quantidade de peças seja ampliado).

Capítulo 5

Conclusões

Esse trabalho introduz o conceito de supervisor temporalmente adaptativo, utilizando o autômato com temporização variável [34] na definição da especificação de comportamento e síntese do supervisor de sistemas a eventos discretos temporizados. A formalização de síntese do supervisor temporalmente adaptativo é baseada no procedimento algorítmico de síntese do supervisor para SEDs temporizados de [32, 35, 36, 47, 37]. As condições para a síntese do supervisor temporalmente adaptativo são garantidas pela Proposição 4.1.

Com a definição do supervisor temporalmente adaptativo, sistemas que utilizam recursos que sofrem avarias devido à situações de desgastes, elevação de temperatura, entre outras, elevam seu tempo de vida e utilização no processamento de tarefas, o que reduz custos, embora aumentem o tempo de processamento em determinadas atividades.

Assim, esse trabalho mostra que os ATVs são de grande importância na síntese de supervisores de SEDs temporizados, em que o exemplo apresentado demonstra a validade deste formalismo, onde pode-se ver que os recursos do sistema têm uma ampliação em seus tempos de atividades devido a um breve estudo feito sobre o mesmo.

Pôde-se ver que a introdução do autômato com temporização variável e sua aplicação à modelagem e controle de sistemas a eventos discretos temporizados se apresenta como um novo formalismo que permite sua utilização na definição de um controle adaptativo referente à variações temporais no sistema. Diferentemente dos casos tratados na literatura, em que a formalização de controle adaptativo utilizada o supervisor é construído como um controlador chaveado para se adaptar à cada tipo de mudança no comportamento [52, 53, 54, 55], aqui a implementação do supervisor é referente a sua adaptação em relação à mudanças nos tempos de vida dos eventos, como é o caso de variação no tempo de vida dos eventos devido à situações específicas como o desgaste de recursos no sistema.

Bibliografia

- [1] 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.
- [2] P.J.G. Ramadge and W.M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, 1989.
- [3] R. Smedinga. *Control of Discrete Events*. PhD thesis, Rijksuniversiteit Groningen, 20 januari 1989.
- [4] 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.
- [5] 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.
- [6] R. Smedinga. Discrete event systems. Technical report, University of Groningen, October 1993.
- [7] J.N. Tsitsiklis. On the control of discrete-event dynamical systems. *Mathematics of Control, Signals, and Systems*, 2:95–107, 1989.
- [8] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, USA, 1979.
- [9] B.A. Brandin and W.M. Wonham. Supervisory control of timed discrete-event systems. *IEEE Transactions on Automatic Control*, 39(2):329–342, 1994.
- [10] 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.
- [11] J.A. Beachy. *Abstract Algebra II*. Waveland Press, Inc., 1996.
- [12] 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.
- [13] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. *Synchronisation and Linearity. An Algebra for Discrete Event Systems*. John Wiley Sons, 1992.
- [14] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, (126):183–235, 1994.
- [15] P. Chrétienne. *Les Réseaux de Petri Temporisés*. Thèse d'état. Université Paris VI, Paris, 1983.

- [16] 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.
- [17] S. Gaubert. *Théorie des Systèmes Linéaires dans les Dioïdes*. PhD thesis, École Nationale Supérieure des Mines de Paris, 1992.
- [18] L. Libeaut. *Sur l'utilisation des Dioïdes pour la Commande des Systèmes à Événements Discrets*. PhD thesis, École Doctorale Sciences pour L'Ingenieur de Nantes, 1996.
- [19] 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.
- [20] J.L. Boimond. Sur l'étude des systèmes à événements discrets dans l'algèbre des dioïdes: Identification, commande des graphes d'événements temporisés, représentation des graphes d'événements temporisés à paramètres variables. Technical report, l'Université d'Angers, 1999.
- [21] 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.
- [22] D.D. Cofer and V.K. Garg. Supervisory control of timed event graphs. *IEEE - 0-7803-2129-4/94*, 1994.
- [23] 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.
- [24] J. Berstel and C. Reutenauer. *Rational Series and their Languages*. Springer, 1988.
- [25] 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.
- [26] S. Gaubert. On rational series in one variable over certain dioids. Technical report, Institut National de Recherche en Informatique et en Automatique, 1994.
- [27] S. Gaubert. Performance evaluation of timed automata. Technical report, Institut National de Recherche en Informatique et en Automatique, 1993.
- [28] S. Gaubert. Performance evaluation of $(\max,+)$ automata. *IEEE Transactions on Automatic Control*, 40(12):2014–2025, December 1995.
- [29] 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.
- [30] 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.
- [31] 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.

- [32] E.M.M. Costa. *Síntese de Supervisores de Sistemas a Eventos Discretos Temporizados e Não Temporizados*. Tese de Doutorado. Universidade Federal da Paraíba - UFPB - Campus II, Campina Grande, Paraíba, Brasil., Novembro de 2001.
- [33] E. Menguy. *Contribution à la Commande des Systèmes Linéaires dans les Dioïdes*. Thèse de Doctorat, École Doctorale Sciences Pour L'Ingénieur de Nantes, 1997.
- [34] E.M.M. Costa. Autômatos com temporização variável: Um novo formalismo para representação de sistemas temporizados. *Revista Ciência e Engenharia, Universidade Federal de Uberlândia*, Julho/Dezembro 2002.
- [35] E.M.M. Costa and A.M.N. Lima. Utilizando dióides na síntese do controlador de sistemas a eventos discretos temporizados. *Anais do Congresso Brasileiro de Automática - CBA2002*, 2002.
- [36] E.M.M. Costa. Síntese de supervisores de sistemas a eventos discretos temporizados utilizando autômatos com temporização variável. *Anais do Congresso Brasileiro de Automática - CBA2002*, 2002.
- [37] E.M.M. Costa and A.M.N. Lima. Synthesis of supervisors for time-varying discrete event systems. *Submetido à revista Controle e Automação da Sociedade Brasileira de Automática*, 2003.
- [38] A. Salomaa. *Formal Languages*. ACM Monograph Series. Academic Press, Inc. New York, 1973.
- [39] R. Alur and D. Dill. Automata for modeling real-time systems. *Proc. 17th International Colloquium on Automata, Languages and Programming, Lectures Notes on Computer Science*, New York: Springer Verlag, 443:322–335, 1990.
- [40] R. Alur. Timed automata. *Proc. NATO-ASI Summer School, Antalaya, Turkey*, 1997.
- [41] 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.
- [42] P. Caspi E. Asarin and O. Maler. A kleene theorem for timed automata. *Proc. of LICS'97*, pages 160–171, 1997.
- [43] E. Asarin. Equations on timed languages. *Proc. of Hybrid'98*, 1998.
- [44] 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.
- [45] M.S. Lawford. *Model Reduction of Discrete Real-Time Systems*. PhD thesis, University of Toronto, 1997.
- [46] A. Gill. *Introduction to the Theory of Finite-State Machines*. McGraw-Hill Electronic Sciences Series. McGRAW-HILL Book Company, 1962.
- [47] E.M.M. Costa. Autômatos com temporização variável e aplicação à síntese de supervisores de sistemas a eventos discretos temporizados. *Anais do III Seminário Nacional de Controle e Automação - SNCA2003*, 2003.

- [48] I. Klimann. *Langages, Séries et Contrôle de Trajectoires*. PhD thesis, l'Université Denis Diderot - Paris 7, Juin 1999.
- [49] 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.
- [50] E.M.M. Costa. Um formalismo de síntese de supervisores de sistemas a eventos discretos temporizados adaptativos à mudanças temporais. *Anais do VI Simpósio Brasileiro de Automação Inteligente, VI SBAI.*, 2003.
- [51] E.M.M. Costa. Utilizando autômatos com temporização variável como representação de supervisores temporalmente adaptativos. *Revista Diálogos e Ciência da Faculdade de Tecnologia e Ciências de Feira de Santana, BA.*, 2003.
- [52] G. Karsai, A. Ledecz, J. Sztipanovits, G. Peceli, G. Simon, and T. Kovacszy. An approach to self-adaptive software based on supervisory control. 2002. DARPA/ITO SEC program (F33615-99-C-3611).
- [53] J. Hespanha, D. Liberzon, A.S. Morse, B.D.O. Anderson, T.S. Brinsmead, and F. De Bruyne. Multiple model adaptive control, part 2: Switching. 2002. Research supported by AFOSR, DARPA, NSF and ONR.
- [54] J. Hespanha, D. Liberzon, and A.S. Morse. Overcoming the limitations of adaptive control by means of logic-based switching. 2001. Paper submitted to System Control Letters.
- [55] J.P. Hespanha. Tutorial on supervisory control. *Lectures Notes for the Workshop Control using Logic and Switching for the 40th Conf. on Decision and Control, Orlando, Florida.*, 2001.
- [56] G.C. Barroso. *Uma Nova Abordagem para a Síntese de Supervisores de Sistemas a Eventos Discretos*. Tese de Doutorado. Universidade Federal da Paraíba, Campus II, Campina Grande, PB, Brasil, 1996.

Parte II

Produção Científica

Durante a realização desta segunda fase do Projeto DCR, foram produzidos os seguintes materiais, os quais estão classificados como artigos e livros, publicados, aprovados e submetidos, e softwares:

Artigos:

Congressos:

Publicados:

III Seminário Nacional de Controle e Automação (III SNCA) - Junho de 2003, Salvador, BA.

1. *Autômatos com Temporização Variável e Aplicação à Síntese de Supervisores de Sistemas a Eventos Discretos Temporizados.*

Submetidos (não aprovados):

VI Simpósio Brasileiro de Automação Inteligente - VI SBAI.

1. *Síntese de Supervisores de Sistemas a Eventos Discretos com Observação Parcial de Eventos Utilizando Dióides: Aplicação às Matrizes de Incidência com Linguagens Normais.* Em co-autoria com Antonio Marcus N. Lima, Dr., UFCG.
2. *Síntese de Supervisores de Sistemas a Eventos Discretos Temporizados Através dos Dióides.* Em co-autoria com Antonio Marcus N. Lima, Dr., UFCG.
3. *Representação Matricial de Grafos de Transições Temporizadas Através dos Dióides.* Em co-autoria com Antonio Marcus N. Lima, Dr., UFCG.
4. *Síntese de Supervisores de Sistemas a Eventos Discretos com Observação Parcial de Eventos Utilizando Dióides: Um Formalismo para Matrizes de Incidência com Linguagens Observáveis.* Em co-autoria com Antonio Marcus N. Lima, Dr., UFCG.
5. *Um Formalismo de Síntese de Supervisores de Sistemas a Eventos Discretos Temporizados Adaptativos à Mudanças Temporais .*
6. *Uma Formalização de Síntese de Supervisores de Sistemas a Eventos Discretos com Decisão.* Em co-autoria com Gilberto Bispo da Silva Góes - CEFET-BA.
7. *Síntese de Supervisores de Sistemas a Eventos Discretos com Observação Total “Aparente”.*
8. *Uma Formulação de Supervisores com Observação Total “Aparente” de Sistemas a Eventos Discretos com Eventos não Observáveis Utilizando Dióides e Matrizes de Incidência.*

Revistas:

Publicados:

Revista Ciência e Engenharia da UFU

1. *Autômatos com Temporização Variável: Um Novo Formalismo para Representação de Sistemas Temporizados.*

Revista Diálogos e Ciência da FTC-FSA

1. *Utilizando Autômatos com Temporização Variável como Representação de Supervisores Temporalmente Adaptativos.*

Submetidos:

Revista Controle & Automação da SBA

1. *Synthesis of Supervisors for Time-Varying Discrete Event Systems.* Em co-autoria com Antonio Marcus N. Lima, Dr., UFCG.
2. *Um Formalismo para Síntese de Supervisores de Sistemas a Eventos Discretos Baseado em Redes de Petri Síncronas.* Em co-autoria com Gilberto Bispo da Silva Góes - CEFET-BA.

Revista Ciência e Engenharia da UFU

1. *Um Formalismo de Decisão para Supervisores de Sistemas a Eventos Discretos.* Em co-autoria com Gilberto Bispo da Silva Góes - CEFET-BA.
2. *Síntese de Supervisores Temporalmente Adaptativos para Sistemas a Eventos Discretos.*

Discrete Event Dynamic Systems: theory and applications Journal, Kluwer Academic Publishers.

1. *Supervisor synthesis of discrete event systems based on dioid algebra.* Em co-autoria com Antonio Marcus N. Lima, Dr., UFCG.

Revista Diálogos e Ciência da FTC-FSA

1. *Supervisores de Sistemas a Eventos Discretos com Decisão para Falha em Sensores Baseados em Redes de Petri Síncronas.* Em co-autoria com Gilberto Bispo da Silva Góes - CEFET-BA.

Livros:

Encaminhados para publicação:

1. *Programação em C para Windows.* Editora da Universidade Federal da Bahia - EDUFBA, Salvador, BA. Ainda sem resposta.
2. *Sistemas Dinâmicos a Eventos Discretos.* Em co-autoria com Antonio Marcus N. Lima, Dr., UFCG. Editora da Universidade Federal da Bahia - EDUFBA, Salvador, BA. Ainda sem resposta.
3. *Introdução aos Sistemas a Eventos Discretos e à Teoria de Controle Supervisório.* Editora da Universidade Federal da Bahia - EDUFBA, Salvador, BA. Ainda sem resposta.

A ser encaminhado para publicação:

1. *Introdução às Redes de Petri e à Modelagem de Sistemas*. Em co-autoria com Antonio Marcus N. Lima, Dr., UFCG. Procurando Editora.

Em preparação:

1. *Eletromagnetismo: Eletrostática e Magnetostática*.

Softwares:

Biblioteca de Dióides em C

Foi preparada uma biblioteca em linguagem C para trabalhar com dióides. Nesta biblioteca encontra-se um conjunto de funções para realizar as operações de adição (\oplus) e multiplicação (\otimes) para as definições específicas do conjunto D a seguir:

- $D = \mathcal{P}(\Sigma^*)$ - para aplicações às linguagens formais;
- $D = \mathbb{R}_{\max}$ - realizando as operações sobre o dióide $(\max, +)$;
- $D = \mathbb{R}_{\min}$ - realizando as operações sobre o dióide $(\min, +)$.

Também, encontram-se nesta biblioteca as operações sobre as séries formais e racionais, fundamentando assim, uma ferramenta para utilização com projetos que necessitem desta álgebra.

RPSED

Um outro Software que está sendo desenvolvido é denominado *RPSED*, o qual é um editor de Redes de Petri para Windows, onde o usuário pode construir supervisores utilizando a abordagem das redes de Petri com função de habilitação de transições (RPFHT) formalizada em [56].

RWDES

O software RWDES foi melhor estruturado, eliminando alguns problemas apresentados, tendo se tornado disponível em mais uma versão teste na página:

<http://www.geocities.com/emontyc>

Parte III

Formação de Recursos Humanos

Introdução

A formação de recursos humanos como parte integrante deste projeto compreende a apresentação de cursos, mini-cursos e palestras a serem apresentados ao público em instituições de ensino do Estado da Bahia, bem como a orientação de alunos de graduação e pós-graduação e preparação de materiais para utilização em acompanhamento de cursos.

Disciplinas:

Durante a realização desta segunda fase do Projeto DCR, foram ministradas as disciplinas *Redes de Petri e Modelagem de Sistemas* e *Automação de Processos Industriais* no Mestrado em Engenharia Elétrica. Esta última, em conjunto com o Professor Jês de Jesus F. Cerqueira. No segundo semestre letivo deste ano deverá ser ministrada a disciplina *Controle de Sistemas a Eventos Discretos*.

Palestras:

Durante o ano de 2002, foram apresentadas as seguintes palestras no departamento de Engenharia Elétrica da UFBA:

- Utilizando Dióides na Síntese do Supervisor de Sistemas a Eventos Discretos Temporizados;
- Síntese de Supervisores de Sistemas a Eventos Discretos com Observação Parcial de Eventos Utilizando Dióides e Matrizes de Incidência.

Também, foi apresentada a seguinte palestra na I Jornada de Computação de Feira de Santana da Faculdade de Tecnologia e Ciências:

- Simulação de Sistemas Utilizando Linguagens de Alto Nível

Orientações:

Também, como parte integrante da formação de recursos humanos, a orientação de trabalhos de pesquisa na área é de importância fundamental. No caso do Mestrado em Engenharia Elétrica, foi finalizada a orientação do aluno Gilberto Bispo da Silva Góes, cuja defesa foi aprovada, tendo como título de sua dissertação:

Uma Abordagem para a Síntese de Supervisores de Sistemas a Eventos Discretos Utilizando Redes de Petri Síncronas.

Na graduação de Engenharia Elétrica, foi finalizada a orientação do primeiro ano do aluno Osmar Franca Siebel no projeto PIBIC:

Controle de Sistemas a Eventos Discretos Temporizados.

Para este período seguinte, está sendo iniciada a continuidade do projeto com o aluno acima referido, o qual foi aprovado para seu segundo ano de bolsa.

Materiais Didáticos:

Várias apostilas foram preparadas para apresentação e utilização em cursos/disciplinas apresentadas, bem como para orientação de alunos, como a seguir:

1. Sistemas a Eventos Discretos;
2. Redes de Petri e Modelagem de Sistemas;
3. Redes de Petri e Sistemas a Eventos Discretos;
4. Álgebra de Dióides;
5. Álgebra de Dióides e Sistemas a Eventos Discretos;
6. Redes de Petri, Controle Supervisório e Controladores Lógicos Programáveis - Em co-autoria com Gilberto Bispo da Silva Góes;
7. Programação em C para Windows;
8. Programação Aplicada em C para Windows.

Todas estas apostilas encontram-se disponíveis na página:

<http://www.geocities.com/emontyc>