

Apostila
Álgebra de Dióides e Aplicações aos Sistemas a Eventos
Discretos

Eduard Montgomery Meira Costa, DSc
UFBA

©Eduard Montgomery Meira Costa, DSc
UFBA, 2002

Parte I

Álgebra de Dióides

Introdução

A álgebra de dióides [1] é uma álgebra definida sobre um conjunto, denotado por D , e que utiliza duas operações específicas: \oplus (soma) e \otimes (multiplicação), cuja principal característica é a idempotência, isto é, a soma de dois elementos iguais é igual ao próprio elemento. O conjunto D apresenta-se com uma notação genérica para definir de uma maneira ampla esta álgebra. Dependendo da particularização do conjunto D , a álgebra de dióides apresenta características especiais para o tratamento de variados tipos de problemas da engenharia e da matemática. As operações de soma e multiplicação utilizadas são formalizadas como operações específicas que podem apresentar similaridades com operações usuais das álgebras comuns.

Em geral, esta álgebra é utilizada na análise de sistemas lineares [2, 3, 1, 4], na análise de sistemas clássicos de controle [5, 6], para descrever o comportamento de sistemas a eventos discretos temporizados cíclicos cuja dinâmica é caracterizada por aspectos de sincronização [7, 8, 9, 10, 11, 12], na avaliação de desempenho de SEDs temporizados [13, 14, 15, 16, 17, 18], descrever linguagens utilizando as séries formais [19, 20, 21, 22, 23, 24, 25], entre outras.

A versatilidade desta álgebra determina uma formulação de grande importância para formalizações de análises de modelos temporizados ou não temporizados de sistemas que podem ser descritos de uma forma matricial, gráfica, por meio de linguagens formais. Especialmente para modelos de sistemas que apresentam características discretas, cuja evolução dinâmica é descrita através da ocorrência de transições de estados (como é o caso dos sistemas a eventos discretos [26, 27, 28, 29, 30, 31]), sejam elas temporizadas ou não temporizadas, a álgebra de dióides tem sido muito aplicada.

Na álgebra de dióides a operação de soma, isto é, \oplus , não contém inversa. Porém, esta situação é contornada por meio de estruturas matemáticas especiais para solucionar tal problema, garantindo sua ampla utilização no estudo de sistemas.

Sendo assim, o conhecimento do formalismo desta estrutura matemática, por ser de grande utilidade atualmente nas áreas de engenharia, é de incontestável necessidade. Aqui será apresentada a álgebra de dióides em sua formalização generalizada, bem como algumas fundamentações algébricas particulares sobre o conjunto D , que definem a álgebra $(\max, +)$, a álgebra $(\min, +)$, a álgebra de caminhos e algumas de suas aplicações.

Álgebra de dióides

Um dióide é definido como a seguir:

Definição 1 *Um conjunto D dotado com duas operações internas \oplus (soma ou adição) e \otimes (produto ou multiplicação) é chamado um dióide, denotado por (D, \oplus, \otimes) , se os seguintes axiomas são verificados:*

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$a \oplus a = a.$$

Pode-se observar que todas as propriedades da álgebra comum são definidas para a álgebra de dióides, isto é, a associatividade, a distributividade, a existência do elemento nulo e do elemento identidade. Contudo, a comutatividade para um dióide, por definição, só é válida para a soma. Neste caso, o dióide é dito não comutativo. Também, a propriedade da idempotência é a principal característica dos dióides, onde a soma de dois elementos iguais é igual ao próprio elemento. O elemento nulo desta álgebra é denotado por ϵ , tal que, sua adição com qualquer elemento de D é o próprio elemento, e sua multiplicação por qualquer elemento de D é sempre o elemento nulo. O elemento identidade é o elemento denotado por e , tal que sua multiplicação com qualquer elemento de D é o próprio elemento.

Quando um dióide satisfaz a propriedade de comutatividade da multiplicação \otimes , o dióide é dito ser comutativo.

Definição 2 Seja (D, \oplus, \otimes) um dióide. Se a multiplicação \otimes é comutativa, isto é, para $\forall a, b \in D$

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

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

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

$$ab = a \otimes b.$$

Em algumas particularizações do conjunto D , define-se a inversão da operação \otimes , como sendo a divisão, denotada por \oslash . Esta operação satisfaz a condição costumeira:

$$a \otimes b = c \Rightarrow a = c \oslash b$$

se $b \neq \epsilon$, e

$$b = c \oslash a$$

se $a \neq \epsilon$, $\forall a, b, c \in D$.

Também, define-se o dióide completo como a seguir:

Definição 3 Um dióide é dito integral se

$$a \otimes b = \epsilon \Rightarrow a = \epsilon \text{ ou } b = \epsilon, \forall a, b \in D.$$

Quando $a \neq \epsilon$, $b \neq \epsilon$ e $ab = \epsilon$, então a e b são chamados de divisores de zero. Assim, num dióide completo não são admitidos divisores de zero.

Quando se define o dióide como um conjunto de matrizes, isto é, $D^{n \times n}$, as operações \oplus e \otimes têm as mesmas características da Definição 1, mas satisfazendo o seguinte: $\forall \mathbf{A}, \mathbf{B} \in D^{n \times n}$,

$$(\mathbf{A} \oplus \mathbf{B})_{i,j} = \mathbf{A}_{i,j} \oplus \mathbf{B}_{i,j} \quad (0.1)$$

$$(\mathbf{A} \otimes \mathbf{B})_{i,j} = \bigoplus_{k=1}^n (\mathbf{A}_{i,k} \otimes \mathbf{B}_{k,j}). \quad (0.2)$$

Ou seja, as operações de soma e multiplicação sobre as matrizes apresentam as mesmas formalizações da soma e da multiplicação da álgebra usual.

Exemplo 1 Considere $a, b \in D$, tal que $a \otimes b = c$ e $b \otimes a = d$, e sejam as matrizes

$$\mathbf{A} = \begin{bmatrix} a & \epsilon \\ \epsilon & b \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} a & b \\ a & \epsilon \end{bmatrix}$$

Então, tem-se

$$\mathbf{A} \oplus \mathbf{B} = \begin{bmatrix} a & \epsilon \\ \epsilon & b \end{bmatrix} \oplus \begin{bmatrix} a & b \\ a & \epsilon \end{bmatrix} = \begin{bmatrix} a \oplus a & b \oplus \epsilon \\ \epsilon \oplus a & b \oplus \epsilon \end{bmatrix} = \begin{bmatrix} a & b \\ a & b \end{bmatrix}$$

e

$$\begin{aligned} \mathbf{A} \otimes \mathbf{B} &= \begin{bmatrix} a & \epsilon \\ \epsilon & b \end{bmatrix} \otimes \begin{bmatrix} a & b \\ a & \epsilon \end{bmatrix} = \begin{bmatrix} a \otimes a \oplus \epsilon \otimes a & a \otimes b \oplus \epsilon \otimes \epsilon \\ \epsilon \otimes a \oplus b \otimes a & \epsilon \otimes b \oplus b \otimes \epsilon \end{bmatrix} \\ &= \begin{bmatrix} a \oplus \epsilon & c \oplus \epsilon \\ \epsilon \oplus d & \epsilon \oplus \epsilon \end{bmatrix} \\ &= \begin{bmatrix} a & c \\ d & \epsilon \end{bmatrix} \end{aligned}$$

O problema da inversão de \oplus

Nos dióides, a operação \oplus não tem inversa. Porém, este problema é contornado pela relação de ordem induzida pela propriedade da idempotência, denotada por \preceq , definida como:

$$a \preceq b \Leftrightarrow a \oplus b = b.$$

A relação de ordem \preceq é dita ser total se

$$\forall (a, b) \in D^2, a \oplus b = a \text{ ou } b.$$

Isto é, para todo par de elementos no dióide, a soma é sempre verificada para o primeiro, ou para o segundo elemento do par.

Esta relação de ordem é compatível com as operações \oplus e \otimes . Ou seja,

$$a \preceq b \Rightarrow a \oplus c \preceq b \oplus c, \forall c \in D,$$

$$a \preceq b \Rightarrow a \otimes c \preceq b \otimes c, \forall c \in D.$$

Deve-se observar que

$$a \oplus b = \epsilon \Rightarrow a = \epsilon \text{ e } b = \epsilon, \forall a, b \in D.$$

O conjunto (D, \preceq) apresenta um limite superior, denotado por \vee , tal que

$$a \vee b = a \oplus b.$$

Ou seja, a soma de dois elementos de um dióide D é sempre o limite superior entre os dois.

Com a formalização da relação de ordem, define-se o dióide completo como a seguir:

Definição 4 *Um dióide é dito ser completo se são verificadas as propriedades de distributividade para somas infinitas. Isto é, $\forall X \subset D$, e $\forall a \in D$,*

$$\begin{aligned} \left(\bigoplus_{x \in X} x \right) \otimes a &= \bigoplus_{x \in X} (x \otimes a), \\ a \otimes \left(\bigoplus_{x \in X} x \right) &= \bigoplus_{x \in X} (a \otimes x). \end{aligned}$$

Um dióide completo admite um elemento superior, denotado por T , definido por

$$T = \bigoplus_{a \in D} a.$$

Este elemento satisfaz as propriedades

$$T \oplus b = T, \forall b \in D$$

e

$$T \otimes \epsilon = \epsilon.$$

Também, define-se para um dióide completo um limite inferior, denotado por \wedge , por

$$a \wedge b = \vee \{c \mid c \preceq a \text{ e } c \preceq b\}. \quad (0.3)$$

O conjunto D munido das operações \vee e \wedge satisfaz a seguinte equivalência:

$$a \preceq b \Leftrightarrow a \oplus b = b \Leftrightarrow a \wedge b = a, \forall a, b \in D. \quad (0.4)$$

Isto é, para uma relação de ordem definida entre dois elementos a e b , sua soma é o elemento b , tal que existe um limite inferior entre os dois, que é a .

Todo o formalismo apresentado sobre a relação de ordem é utilizado para fundamentar a solução de problemas de equações lineares sobre os dióides.

Equações lineares nos dióides

O problema da residuação é utilizado para solucionar equações e inequações do tipo

$$x = ax \oplus b \quad (0.5)$$

e

$$x \succeq ax \oplus b \quad (0.6)$$

que aparecem como equações implícitas de grafos de eventos temporizados que modelam determinados tipos de sistemas, como é o caso dos SEDTs.

Uma outra classe de equações que são estudadas igualmente, é da forma

$$ax = b.$$

Para este caso específico, a máxima sub-solução é dada por

$$b \oslash a = \vee \{x \mid ax \preceq b\}, \quad (0.7)$$

onde $b \oslash a$ é chamado de resíduo de b por a . Essa noção é utilizada para solucionar problemas mais complexos como as equações (0.5) e (0.6), seja utilizando elementos singulares ou matrizes.

Para o caso de elementos singulares, tem-se o seguinte:

Teorema 1 *A inequação $x \succeq ax \oplus b$ num dióide completo admite uma ínfima solução, igual à a^*b , onde a^* é definido por*

$$a^* = \bigoplus_{n \in \mathbb{N}} a^n = e \oplus a \oplus a^2 \oplus \dots \quad (0.8)$$

*Em outros termos, $\bar{x} = a^*b$ soluciona a igualdade ($\bar{x} = a\bar{x} \oplus b$).*

Demonstração: Apresentada em [12]. ◆

Quando utilizando matrizes, esses resultados são similares. Entretanto, devem ser consideradas as operações de multiplicação e adição matriciais (equações (0.1) e (0.2)). Assim, tem-se:

Proposição 1 *Seja D um dióide completo. Sendo dadas duas matrizes $A \in D^{n \times p}$ e $B \in D^{n \times m}$, o resíduo de B por A , denotado por $B \oslash A$, é uma matriz de dimensão $p \times m$, obtida por*

$$(B \oslash A)_{i,j} = \bigwedge_{k=1}^n B_{k,j} \oslash A_{k,i}. \quad (0.9)$$

Utilizando este resultado e considerando a desigualdade:

$$x \succeq Ax \oplus b, \quad (0.10)$$

onde $x, b \in D^n$ e $A \in D^{n \times n}$, a solução para um dióide completo é dada da mesma forma que o Teorema 1, isto é, $x \succeq A^*b$, sendo x um vetor e

$$A^* = \bigoplus_{n \in \mathbb{N}} A^n = I \oplus A \oplus A^2 \oplus \dots \quad (0.11)$$

com I sendo a matriz identidade

$$I = \begin{bmatrix} e & \epsilon & \epsilon & \cdots & \epsilon \\ \epsilon & e & \epsilon & \cdots & \epsilon \\ \epsilon & \epsilon & \ddots & \ddots & \vdots \\ \vdots & \vdots & \epsilon & e & \epsilon \\ \epsilon & \epsilon & \cdots & \epsilon & e \end{bmatrix}$$

Para cada aplicação específica dos dióides, são estudadas características particulares, como auto-valores e auto-vetores, séries formais, séries racionais, entre outras. No decorrer desse texto, serão apresentadas as características próprias de alguns dióides particulares.

Linguagens Formais e Dióides

Para entender o contexto das linguagens formais nos dióides é necessário conceituá-las.

As linguagens formais são representadas por alfabetos não vazios de símbolos, o qual, geralmente é representado pela letra grega Σ . A concatenação finita de símbolos, define uma palavra. Esta operação é definida por

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

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

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

O comprimento de uma palavra, representado pela cardinalidade $|s|$, é igual ao número de símbolos que a compõe. Define-se na Teoria de Linguagens Formais a palavra nula, representada por ε , a qual é a única palavra de comprimento nulo, isto é, $|\varepsilon| = 0$. Desta forma, $\varepsilon \notin \Sigma$ pois é uma palavra, e não um símbolo do alfabeto Σ .

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

Definição 5 *A concatenação de palavras é a operação*

$$\text{cat}_s : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*,$$

onde

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

Assim, pela Definição 5, dadas duas palavras u e v sobre um alfabeto Σ , com

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

sua concatenação define uma nova palavra que é

$$s = \text{cat}_s(u, v) = uv = \sigma_1 \dots \sigma_k \sigma_{k+1} \dots \sigma_n. \quad (0.12)$$

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

$$\begin{aligned} \text{cat}_\Sigma(\text{cat}_\Sigma(\sigma_1, \sigma_2), \sigma_3) &= \text{cat}_\Sigma(\sigma_1, \text{cat}_\Sigma(\sigma_2, \sigma_3)) & \text{e} \\ \text{cat}_s(\text{cat}_s(s_1, s_2), s_3) &= \text{cat}_s(s_1, \text{cat}_s(s_2, s_3)). \end{aligned}$$

Um conjunto de palavras é definido como:

Definição 6 *Dado $k \in \mathbb{N}$, denota-se por Σ^k o conjunto de todas as palavras s sobre Σ cujo comprimento é igual a $|s| = k$.*

Como consequência desta definição, definem-se dois conjuntos especiais, que são Σ^+ e Σ^* , dados por

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

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

Um conjunto de palavras define uma linguagem, ou seja:

Definição 7 Dado um alfabeto Σ , L é uma linguagem sobre Σ se e somente se, $L \subseteq \Sigma^*$.

Uma linguagem tanto pode ser $L = \Sigma^*$ como $L = \{ \}$ (linguagem vazia, a qual é diferente da linguagem Σ^0 , composta apenas da palavra vazia ε).

Define-se o prefixo de uma palavra s , como sendo uma parte inicial de comprimento arbitrário de uma palavra s . Em outras palavras, um prefixo de uma palavra s é qualquer palavra $u \in \Sigma^*$ que possa ser completada com outra palavra $v \in \Sigma^*$ para formar s . Todos os prefixos de uma dada palavra s incluindo a palavra vazia ε , formam o conjunto denotado por $Pre(s)$.

O conjunto que inclui todas as palavras de uma linguagem $L \subseteq \Sigma^*$ e todos os seus prefixos é definido como prefixo-fechamento, ou fechamento de L , o qual é dado por:

$$\overline{L} = \{u | \exists v \in \Sigma^* \wedge uv \in L\}. \quad (0.13)$$

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

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

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

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

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

intersecção

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

e complemento

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

Com a definição de linguagem, estende-se a operação de concatenação para as linguagens como

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

onde

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

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

No estudo das linguagens formais, algumas linguagens, podem ser representadas através de uma expressão regular, utilizando a álgebra convencional e os símbolos do alfabeto. Assim, linguagens complexas podem ser representadas em termos de expressões simples, como apresentado por Hopcroft e Ullman [32]:

- σ^* - representa a repetição do símbolo σ , por um número arbitrário de vezes;
- s^* - representa a repetição da palavra s , por um número arbitrário de vezes;

- + - símbolo empregado como o operador lógico *ou*, indicando uma opção entre duas ou mais possibilidades.

As expressões regulares são definidas recursivamente da seguinte maneira:

1. \emptyset é uma expressão regular denotando o conjunto vazio, ε é uma expressão regular denotando o conjunto $\{\varepsilon\}$ e σ é uma expressão regular denotando o conjunto $\{\sigma\}$ para todo $\sigma \in \Sigma$;
2. Se s_1 e s_2 são expressões regulares, então s_1s_2 , s_1^* , s_2^* e $(s_1 + s_2)^*$ são expressões regulares;
3. Toda expressão regular é construída através da aplicação das regras 1 e 2 um número finito de vezes.

As expressões regulares provêm uma representação compacta para linguagens complexas formadas por um número infinito de palavras. A palavra vazia ε e a linguagem vazia \emptyset , também são consideradas nas expressões regulares para as quais têm-se as seguintes propriedades [33]:

$$\varepsilon s = s\varepsilon = s, \quad \varepsilon^* = \varepsilon,$$

e

$$\emptyset + L = L, \quad \emptyset L = L\emptyset = \emptyset, \quad \emptyset^* = \varepsilon.$$

Exemplo 2 Para $\Sigma = \{\alpha, \beta\}$ e uma linguagem prefixo-fechada L na qual os símbolos aparecem alternadamente, com todas as palavras iniciando sempre por α , tem-se:

$$L = \{\varepsilon, \alpha, \alpha\beta, \alpha\beta\alpha, \alpha\beta\alpha\beta, \dots\} = (\alpha\beta)^*(\varepsilon + \alpha). \quad (0.14)$$

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

As linguagens formais apresentam uma representação gráfica denominada de autômato, que é um grafo orientado formado por vértices (círculos) e arcos etiquetados pelos símbolos de Σ . Os autômatos são utilizados para representar máquinas de estados finitos, e apresentam em sua estrutura um estado inicial (representado por um vértice com uma seta que não é saída de nenhum lugar) e estados marcados (representados por dois círculos concêntricos). Assim, os autômatos apresentam uma linguagem associada, que representa todos os caminhos que podem ser seguidos, e uma linguagem marcada, que representa apenas os caminhos que levam do estado inicial a um estado marcado. Ambas são formalizadas através das linguagens formais. Na Figura 0.1, é apresentado um autômato, cuja linguagem é

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

e cuja linguagem marcada é

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

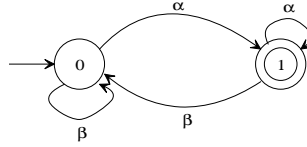


Figura 0.1: Autômato.

Na segunda parte desta Apostila, os autômatos serão apresentados em detalhes para a compreensão da Teoria de Controle Supervisório, que é a formulação clássica de controle de Sistemas a Eventos Discretos.

Representação das Linguagens Formais nos Dióides

Para aplicar a álgebra de dióides às linguagens formais, define-se que o conjunto D é $\mathcal{P}(\Sigma^*)$, onde $\mathcal{P}(\Sigma^*)$ é o conjunto de todas as linguagens formadas com símbolos do alfabeto Σ [34]. Assim, um elemento de $\mathcal{P}(\Sigma^*)$ é uma linguagem, e as operações \oplus e \otimes são definidas como segue:

Definição 8 *Seja (D, \oplus, \otimes) um dióide. Sendo $D = \mathcal{P}(\Sigma^*)$, os operadores \oplus e \otimes , são definidos como as operações de união e concatenação, respectivamente, de forma que dados dois elementos $L_1, L_2 \in D$,*

$$L_1 \otimes L_2 = \{s_1 s_2 \mid s_1 \in L_1, s_2 \in L_2\} \quad e$$

$$L_1 \oplus L_2 = \{s \mid s \in L_1 \vee s \in L_2\}.$$

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

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

Exemplo 3 *Dadas as linguagens $L_1 = \{a, b\}$ e $L_2 = \{bbc\}$, tem-se que*

$$L_3 = L_1 \oplus L_2 = \{a, b, bbc\} \quad e$$

$$L_4 = L_1 \otimes L_2 = \{abbc, bbbc\}.$$

Como definido para as linguagens formais [35, 32, 33], tem-se como consequência da Definição 8 que a concatenação de uma linguagem L com ‘ e ’, é

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

e a concatenação de uma linguagem L com ϵ é

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

Exemplo 4 Seja $L = \{\beta\lambda\}$. Então,

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

e ,

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

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

Exemplo 5 A linguagem $L_1 = \alpha\beta^*$ somada com a linguagem $L_2 = \alpha$, é

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

que implica na união de L_1 e L_2 .

Todos os axiomas da Definição 1 são válidos quando $D = \mathcal{P}(\Sigma^*)$ [36]. Também, deve-se observar que $D = \mathcal{P}(\Sigma^*)$ é um dióide não comutativo, desde que para dois símbolos quaisquer $\alpha, \beta \in \Sigma$, $\alpha\beta \neq \beta\alpha$.

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 [19, 24]:

Definição 9 Uma série formal Y sobre um alfabeto Σ com coeficientes em um dióide D é um mapa

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

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

Exemplo 6 Dado o alfabeto $\Sigma = \{\alpha\}$, tem-se

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

Considerando $D = \{\epsilon, e\}$, como um exemplo específico pode-se definir

$$\begin{aligned} (y|\varepsilon) &= e, \\ (y|\alpha) &= e, \\ (y|\alpha\alpha) &= \epsilon, \\ (y|\alpha\beta\alpha) &= e, \\ &\vdots \end{aligned}$$

como os coeficientes das respectivas palavras de Σ^* .

É usual denotar o mapa Y pela série

$$Y = \bigoplus (y|s) s. \quad (0.15)$$

No caso do Exemplo 6, o mapa definido pode, portanto, ser reescrito como:

$$Y = e\varepsilon \oplus e\alpha \oplus \underbrace{\epsilon\alpha\alpha}_{=\epsilon} \oplus e\alpha\beta\alpha \oplus \dots$$

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

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

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

$$(y_1 \oplus y_2|s) = (y_1|s) \oplus (y_2|s) \quad (0.16)$$

$$(y_1 \otimes y_2|s) = \bigoplus_{uv=s} (y_1|u) \otimes (y_2|v), \quad (0.17)$$

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

Exemplo 7 Considerando $D = \{\epsilon, e\}$, $\Sigma = \{\alpha, \beta\}$ e

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

tem-se que, se

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

então, verifica-se

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

e,

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

Observe que a soma de Cauchy representa a união das linguagens e o produto representa a concatenação de linguagens, de acordo com a Definição 8. A soma garante que uma palavra pertencente apenas a uma das linguagens, sempre existe nela. O produto garante que só existe uma linguagem se existir o sufixo em uma linguagem e o prefixo na outra linguagem, que devem ser concatenados (a inexistência de um deles é definida pelo valor do coeficiente $(y|s) = \epsilon$).

A equação (0.16) define que para uma mesma palavra $s \in \Sigma^*$ que apresenta coeficientes diferentes, o coeficiente da soma é igual a soma dos coeficientes. Em termos práticos, uma mesma palavra que tem coeficientes diferentes pode ser vista como uma única palavra com um único coeficiente. Similarmente, a equação (0.17) determina que o coeficiente de uma palavra construída pela concatenação de duas outras palavras, é determinado pela soma da multiplicação dos coeficientes das palavras concatenadas. Isto é, o coeficiente de uma palavra pode ser dividido em partes com as subpalavras que a compõe.

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

Definição 10 A operação estrela para uma série $Y \in D \langle \langle \Sigma \rangle \rangle$ é definida como

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

onde ' e ' é o elemento identidade.

As séries formais permitem descrever linguagens através da equação (0.17). Uma linguagem não temporizada pode ser descrita por meio de uma série formal definindo

$$D = \mathbb{B} = \{\epsilon, e\},$$

que denota o semi-anel *binário*. Assim, tem-se:

Definição 11 Uma linguagem regular $L = \{s, s', \dots\} \subseteq \Sigma^*$, é representada pela série formal

$$Y_L = \bigoplus_{s \in \Sigma^*} (y|s) s \quad (0.18)$$

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

$$(y|s) = e, \text{ se } s \in L, \quad (0.19)$$

$$(y|s) = \epsilon, \text{ se } s \notin L. \quad (0.20)$$

Exemplo 8 Dado o alfabeto $\Sigma = \{\alpha, \beta\}$, a linguagem $L = \{\epsilon, \alpha, \alpha\beta, \beta\alpha, \alpha\alpha, \beta\beta, \beta\alpha\beta\}$ é representada pela série formal

$$Y_L = e\epsilon \oplus e\alpha \oplus e\alpha\beta \oplus e\beta\alpha \oplus e\alpha\alpha \oplus e\beta\beta \oplus e\beta\alpha\beta \underbrace{\oplus e\alpha\alpha\alpha \oplus \dots \oplus e\beta\beta\beta \oplus \dots}_{\Sigma^* - L} \quad (0.21)$$

onde $(y|s) = e, \forall s \in L$, $e (y|s) = \epsilon, \forall s \in \Sigma^* - L$. A série formal da equação (0.21) 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 uma linguagem L é descrita por uma série formal Y_L . Assim, as operações de união e concatenação de linguagens (ver Definição (8)) 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. Isto será visto adiante.

Álgebra $(\max, +)$

A álgebra $(\max, +)$ é um caso particular da álgebra de dióides na qual $D = \mathbb{R}_{\max}$, onde \mathbb{R}_{\max} denota o conjunto $\mathbb{R} \cup \{-\infty\}$. Para um dióide (D, \oplus, \otimes) , com $D = \mathbb{R}_{\max}$, as operações \oplus e \otimes são: \max (máximo) e $+$ (soma), respectivamente. O elemento nulo é definido como $\epsilon = -\infty$ e o elemento identidade é definido por $e = 0$.

Nesta álgebra, $-\infty$ é utilizado como o elemento nulo da adição, pois satisfaz:

$$\forall a \in \mathbb{R}, a \oplus \epsilon = a \oplus (-\infty) = \max\{a, -\infty\} = a = \max\{-\infty, a\} = (-\infty) \oplus a = \epsilon \oplus a. \quad (0.22)$$

Por outro lado, utiliza-se $e = 0$, como sendo o elemento identidade da multiplicação, pois satisfaz

$$\forall a \in \mathbb{R}, a \otimes e = a + 0 = a = 0 + a = e \otimes a. \quad (0.23)$$

Essa estrutura algébrica, é um dióide comutativo, pois a propriedade da comutatividade da multiplicação \otimes é satisfeita. Por exemplo,

$$1 \otimes 2 = 1 + 2 = 3 = 2 + 1 = 2 \otimes 1.$$

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

$$D = \mathbb{R}_{\max}^{n \times n}. \quad (0.24)$$

As propriedades do dióide $(\max, +)$ são igualmente satisfeitas, quando usando matrizes, isto é, dadas duas matrizes $\mathbf{A}, \mathbf{B} \in \mathbb{R}_{\max}^{n \times n}$, as equações (0.1) e (0.2) são verificadas.

Também, vê-se o problema da inversão da operação \oplus , desde que esta é a operação usual \max . As condições apresentadas para a solução de equações lineares são utilizadas para encontrar os resíduos.

No caso da operação \otimes , que é usualmente a operação $+$, a solução para a equação (0.7) pode ser entendida como o valor que somado (usual $+$, que nos dióides é \otimes) com o elemento a , tem solução dada por b . Em outros termos,

$$b \oslash a = \vee \{x | ax \preceq b\} \Rightarrow a + x \preceq b \Rightarrow x \preceq b - a.$$

Estes valores devem ser utilizados na equação (0.9) para encontrar o resíduo para sistemas matriciais.

Exemplo 9 Considere as matrizes com coeficientes em \mathbb{R}_{\max} dadas por

$$\mathbf{A} = \begin{bmatrix} 2 & 2 \\ 1 & e \end{bmatrix} \quad e \quad \mathbf{B} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}.$$

Então, tem-se que, de acordo com a equação (0.9), a máxima solução é:

$$\mathbf{B} \oslash \mathbf{A} = \begin{bmatrix} (3 \oslash 2) \wedge (1 \oslash 1) \\ (3 \oslash 2) \wedge (1 \oslash e) \end{bmatrix} = \begin{bmatrix} 1 \wedge e \\ 1 \wedge 1 \end{bmatrix} = \begin{bmatrix} e \\ 1 \end{bmatrix}.$$

Pode-se verificar que $\mathbf{B} \oslash \mathbf{A}$ satisfaz

$$\mathbf{A}(\mathbf{B} \oslash \mathbf{A}) = \begin{bmatrix} 2 \otimes e \oplus 2 \otimes 1 \\ 1 \otimes e \oplus e \otimes 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix},$$

o que determina que efetivamente ela é a solução que satisfaz a igualdade. Assim, a equação $\mathbf{A}\mathbf{X} = \mathbf{B}$ admite uma solução que é $\mathbf{X} = \mathbf{B} \oslash \mathbf{A}$.

A solução para $x \succeq Ax \oplus b$ no caso de $D = \mathbb{R}_{\max}$, pode ser vista no exemplo a seguir:

Exemplo 10 Seja $\mathbf{A} \in \mathbb{R}_{\max}^{2 \times 2}$ definida por

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

e $b \in \mathbb{R}_{\max}^2$ dado por

$$b = \begin{bmatrix} \epsilon \\ e \end{bmatrix}.$$

Calculando \mathbf{A}^* , tem-se:

$$\mathbf{A}^* = \begin{bmatrix} e & \epsilon \\ \epsilon & e \end{bmatrix} \oplus \begin{bmatrix} \epsilon & e \\ e & e \end{bmatrix} \oplus \begin{bmatrix} e & e \\ e & e \end{bmatrix} \oplus \dots = \begin{bmatrix} e & e \\ e & e \end{bmatrix}.$$

Calculando \mathbf{A}^*b , tem-se

$$\mathbf{A}^*b = \begin{bmatrix} e & e \\ e & e \end{bmatrix} \otimes \begin{bmatrix} \epsilon \\ e \end{bmatrix} = \begin{bmatrix} e \\ e \end{bmatrix},$$

que é o valor de x que satisfaz a igualdade

$$x = \mathbf{A}x \oplus b = \begin{bmatrix} \epsilon & e \\ e & e \end{bmatrix} \otimes \begin{bmatrix} e \\ e \end{bmatrix} \oplus \begin{bmatrix} \epsilon \\ e \end{bmatrix} = \begin{bmatrix} e \\ e \end{bmatrix}.$$

A álgebra $(\max, +)$ é muito utilizada para descrição de reconhecimento de linguagens temporizadas em autômatos temporizados e redes de Petri temporizadas. Dessa forma, torna-se necessário compreender esses formalismos para apresentar sua utilização.

Autômatos temporizados (max,+)

Um autômato temporizado é um autômato que inclui uma representação explícita do tempo na sua estrutura de transição. Em um autômato temporizado, um intervalo de tempo está relacionado com cada função de transição, denominado tempo de vida do evento. Este tempo de vida expressa o menor tempo que deve transcorrer para que o evento se torne habilitado e possa ocorrer. Um autômato temporizado reconhece uma linguagem temporizada [37, 38, 39, 40, 41], que é uma linguagem formalizada sobre pares (*evento*, *tempo*).

Dentre as estruturas de autômatos temporizados, como os GTAs e os GTTs, encontra-se a formalização do autômato (max, +) [14, 42]:

Definição 12 *Um autômato finito $A_{(\max,+)}$ sobre um alfabeto Σ é uma quádrupla $A_{(\max,+)} = (Q, \theta, T, \phi)$, onde Q é um conjunto finito de estados e θ , T e ϕ são mapas*

$$\theta : Q \rightarrow \mathbb{R}_{\max}, \quad (0.25)$$

$$T : Q \times \Sigma \times Q \rightarrow \mathbb{R}_{\max}, \quad (0.26)$$

$$\phi : Q \rightarrow \mathbb{R}_{\max} \quad (0.27)$$

denominados atraso inicial, tempo de transição e atraso final, respectivamente.

Um autômato $A_{(\max,+)}$ é representado graficamente por um multigrafo valorado, definido por vértices, formados pelo conjunto de estados Q e por três tipos de arcos

1. Os arcos internos, $i \xrightarrow{\sigma} j$, $\forall i, j \in Q$ e $\sigma \in \Sigma$ tais que $T_{i,\sigma,j} \neq \epsilon$. O arco $i \xrightarrow{\sigma} j$, é valorado pelo escalar $T_{i,\sigma,j}$;
2. Os arcos de entrada $\rightarrow i$, valorados por θ_i , $\forall i \in Q$ tal que $\theta_i \neq \epsilon$;
3. Os arcos de saída $i \rightarrow$, valorados por ϕ_i , $\forall i \in Q$ tal que $\phi_i \neq \epsilon$.

Deve-se observar na representação gráfica do autômato (max, +) que apenas são apresentados os arcos de entrada e de saída em que $\theta_i \neq \epsilon$ e $\phi_i \neq \epsilon$. Ou seja, os atrasos iniciais e finais finitos são os que são representados graficamente, o que determina os estados inicial e marcados do autômato (max,+).

Exemplo 11 *Seja $\Sigma = \{\alpha, \beta\}$. O autômato com conjunto de estados $Q = \{0, 1, 2\}$, tempos de transições $T_{0,\alpha,1} = 1$, $T_{0,\alpha,2} = 3$, $T_{1,\alpha,2} = 4$, $T_{2,\beta,2} = 1$, $T_{2,\beta,1} = 5$, $T_{2,\beta,0} = 7$, $T_{1,\beta,1} = 1$ e $T_{1,\beta,0} = 2$, atrasos final e inicial $\phi_0 = 2$ e $\theta_0 = e = 0$, respectivamente (os outros valores de ϕ , T e θ são iguais a ϵ) está representado na Figura 0.2. Os arcos valorados com zero (ou 'e') podem ser omitidos, isto é, o arco de entrada $\rightarrow 0$, não valorado, define $\theta_0 = e$.*

Comparando o autômato (max,+) com o autômato não temporizado, observa-se que as funções de transição dos autômatos (max,+) são incluídas no mapa T . O estado inicial é definido pelo estado $q \in Q$, em que $\theta_q \neq \epsilon$. Os estados marcados são os estados $q \in Q$, em que $\phi_q \neq \epsilon$.

A semântica do autômato (max,+) é a seguinte:

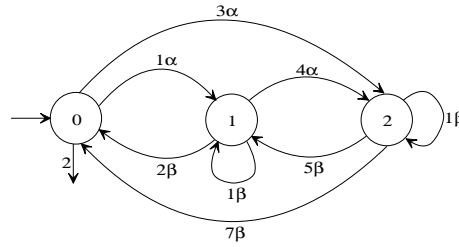


Figura 0.2: Autômato $(\max, +)$.

1. Há um relógio global que está sempre sendo incrementado;
2. O tempo de vida de um evento, denotado por $T_{q,\sigma,q'} \in T$, é o tempo mínimo necessário para sua habilitação;
3. Para iniciar a execução do autômato (alcançar o estado inicial) é transcorrido um tempo θ_{q_0} no relógio global;
4. Estando no estado inicial (ou em qualquer outro), os contadores dos eventos definidos neste estado vão sendo decrementados;
5. Quando um contador de um dos eventos definidos no estado é zerado, o evento torna-se habilitado, podendo ocorrer a qualquer instante;
6. Se com a incrementação do relógio global, for zerado o contador de um outro evento definido neste estado, ele também torna-se habilitado;
7. A ocorrência de um evento habilitado reinicializa seu contador e muda o estado do autômato, desabilitando os outros eventos;
8. Ao atingir um estado marcado, ao transcorrer o tempo de atraso final, o autômato $(\max, +)$ reconhece este estado, reconhecendo assim a palavra que o levou do estado inicial até ele;
9. Eventos iguais com diferentes tempos de vida definem não determinismo no autômato $(\max, +)$.

Exemplo 12 A evolução do autômato $A_{(\max, +)}$ da Figura 0.3 é descrita como a seguir:

- i. Iniciando o relógio global, após decorridas 2 unidades de tempo, o autômato alcança o estado inicial.
- ii. Após uma unidade de tempo o autômato pode reconhecer a palavra 3ε .
- iii. Após mais uma unidade de tempo, o evento α torna-se habilitado.
- iv. Estando o evento α habilitado, ele permanece habilitado indefinidamente até sua ocorrência que muda o estado do autômato para o estado 2, reiniciando seu contador.

- v. No estado 2, após 3 unidades de tempo, o contador do evento β é zerado, habilitando-o.
- vi. A ocorrência do evento β no estado 2, leva o autômato ao estado 1.
- vii. Novamente no estado 1, a passagem de mais uma unidade de tempo permite o reconhecimento da palavra $8\alpha\beta$.
- viii. Considerando que β não ocorre no estado 2, a passagem de mais uma unidade de tempo habilita κ . Dessa forma, tanto o evento β , como o evento κ permanecem habilitados indefinidamente, até a ocorrência de um deles.
- ix. Se ocorre κ , o autômato alcança o estado 3.
- x. Estando no estado 3, a passagem de 2 unidades de tempo habilita β , que permanece habilitado até sua ocorrência.
- xi. Se ocorre β no estado 3, o autômato alcança o estado 1.
- xii. Novamente no estado 1, a passagem de mais uma unidade de tempo permite o reconhecimento da palavra $11\alpha\kappa\beta$.

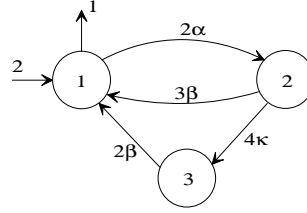


Figura 0.3: Autômato $(\max,+)$.

O reconhecimento de um caminho em um autômato $(\max,+)$ é descrito como a seguir:

Definição 13 *Seja um autômato $A_{(\max,+)}$ e Q^n o conjunto de todas as seqüências de estados de comprimento n , definido por*

$$Q^n = \{p | p = (q_0, \dots, q_n) \wedge q_0, \dots, q_n \in Q\}. \quad (0.28)$$

Define-se que uma palavra $s = \sigma_1 \dots \sigma_n$ é reconhecida no caminho

$$p = (q_0, \dots, q_n) \in Q^n,$$

se

$$P(p, s) := \theta_{q_0} + T_{q_0, \sigma_1, q_1} + \dots + T_{q_{n-1}, \sigma_n, q_n} + \phi_{q_n} \neq \epsilon, \quad (0.29)$$

onde P é a função peso do caminho p .

De acordo com esta definição, pode-se observar que o reconhecimento de uma palavra se dá quando um estado marcado é atingido e o tempo de atraso final é transcorrido (ver Exemplo 12). O reconhecimento de um caminho pode também ser descrito por meio das seqüências de eventos através dos datadores. Estes são definidos a seguir:

Definição 14 *Um datador é um mapa*

$$Y : \Sigma^* \rightarrow \mathbb{R}_{\max} \quad (0.30)$$

onde Y é o tempo que um autômato $A_{(\max,+)}$ leva para percorrer uma seqüência

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

Dessa definição, vê-se que a função datadora define os tempos das palavras da linguagem Σ^* . Assim, o autômato $A_{(\max,+)}$ leva um tempo Y para sair de um estado q para um outro estado q' , seguindo uma seqüência s , onde $(y|s)$ denota o valor de Y na palavra s [19].

Definição 15 *Um datador Y é dito reconhecível se existir um autômato $A_{(\max,+)}$ tal que*

$$(y|s) \neq \epsilon. \quad (0.31)$$

Exemplo 13 *No Exemplo 12, vê-se que $s = \alpha\beta$ é uma palavra reconhecida, pois*

$$(y|\alpha\beta) = P(p, s) = (\theta_1 + T_{1,\alpha,2} + T_{2,\beta,1} + \phi_0) = 5 \neq \epsilon$$

com $p = (1, 2, 1)$.

Com as definições anteriores, a evolução dinâmica do autômato $(\max,+)$ pode ser representada por um vetor x de tempos das ocorrências dos eventos, igualmente ao autômato temporizado descrito em Cassandras e Lafortune [43]. Isto é, o i -ésimo elemento de x é um mapa

$$x_i : \mathbb{N} \rightarrow \mathbb{R}_{\max}, \quad (0.32)$$

que é interpretado como o instante de tempo da n -ésima ocorrência do evento etiquetado por i .

Exemplo 14 *Na Figura 0.4(a), é apresentado uma parte do diagrama de ocorrências dos eventos do autômato $(\max,+)$ apresentado na Figura 0.4(b), onde $\Sigma = \{\alpha, \beta\}$. Neste diagrama, vê-se que os menores tempos de ocorrência do evento α são 2, 7 e 12, enquanto que os menores tempos de ocorrência do evento β são 5, 10 e 15. Dessa forma, encontram-se as seqüências*

$$\begin{aligned} x_\alpha(1) &= 2, x_\alpha(2) = 7, x_\alpha(3) = 12 \\ x_\beta(1) &= 5, x_\beta(2) = 10, x_\beta(3) = 15. \end{aligned}$$

Logo, o vetor x é definido por

$$x(n) = \begin{bmatrix} x_\alpha(n) \\ x_\beta(n) \end{bmatrix}$$

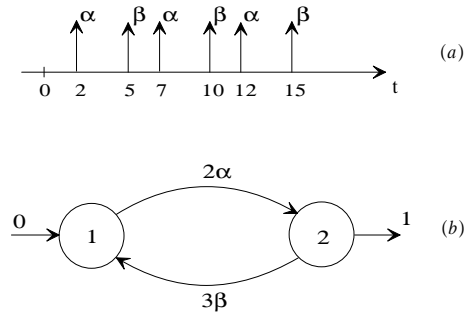


Figura 0.4: (a) Diagrama temporal de ocorrência dos eventos α e β do (b) Autômato $(\max, +)$.

o qual contém as informações a respeito dos tempos de ocorrência de α e β , dados por

$$\begin{array}{ccc} \begin{bmatrix} 2 \\ 5 \end{bmatrix} & \rightarrow & \begin{bmatrix} 7 \\ 10 \end{bmatrix} \rightarrow \begin{bmatrix} 12 \\ 15 \end{bmatrix} \\ x(1) & & x(2) \quad x(3) \end{array}$$

A evolução dinâmica do autômato $(\max, +)$ pode ser descrita por meio de um sistema linear através da matriz de tempos de transição \mathbf{A} , e pelo vetor x , utilizando a álgebra de dióides [7, 8]. Também, as séries formais podem ser utilizadas para a descrição de sua evolução dinâmica e determinação de seus estados reconhecidos.

Deve-se observar que o autômato $(\max, +)$ tem uma formalização similar ao GTA [44], considerando que a representação do tempo é definida diretamente no arco e os tempos de vida são definidos como os tempos mínimos para habilitação dos eventos, isto é, o limite superior é sempre infinito. Dessa forma, os SEDs temporizados descritos em Brandin e Wonham [45], considerando apenas os eventos remotos, apresentam a mesma descrição da evolução dinâmica.

Exemplo 15 Os autômatos apresentados na Figura 0.5 reconhecem as mesmas linguagens temporizadas. O autômato da Figura 0.5(a) é um grafo de transições de atividades, em que os eventos são definidos por $(\sigma_{q,q'}, t_\sigma, \infty) \in \Sigma$, onde o índice q, q' do evento σ define o estado q onde o evento está habilitado e o estado q' que é alcançado após sua ocorrência, respectivamente, como descrito em [45]. Assim, especificamente para este exemplo, tem-se $(\alpha_{1,2}, 2, \infty)$, $(\beta_{2,3}, 2, \infty)$, $(\beta_{3,4}, 2, \infty)$, $(\alpha_{4,1}, 3, \infty)$, $(\kappa_{4,5}, 1, \infty)$ e $(\lambda_{5,1}, 2, \infty)$. O autômato da Figura 0.5(b) é um grafo de transições temporizadas, que é uma representação da semântica do grafo de transições de atividades da Figura 0.5(a), construído de acordo com Brandin e Wonham [45]. O autômato da Figura 0.5(c) é um autômato $(\max, +)$. A semântica desses autômatos é a mesma. Para qualquer um deles, a evolução dinâmica é descrita semelhantemente ao Exemplo 12. No caso do grafo de transições temporizadas (Figura 0.5(b)), cada arco que apresenta um evento ‘tick’ (representado por t), determina a passagem de uma unidade de tempo no relógio global. Observa-se que o número de estados do autômato $(\max, +)$ é bem menor que o número de estados do grafo de transição temporizada, e apresenta uma forma visual dos tempos direta, em relação ao grafo de transições de atividades.

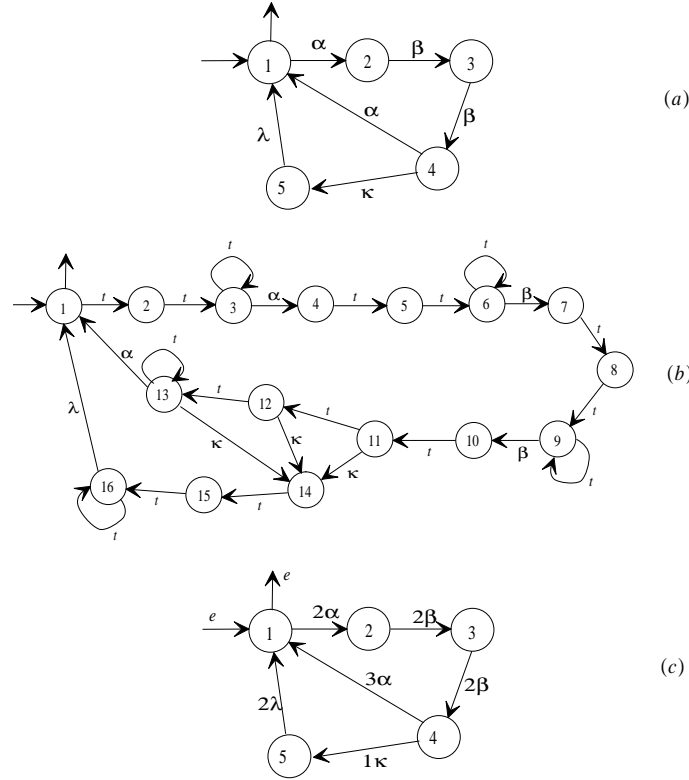


Figura 0.5: (a) Grafo de transições de atividades, (b) Grafo de transições temporizadas e (c) Autômato $(\max, +)$.

Observação: Quando se considera todos os tempos de vida dos eventos de um autômato $(\max, +)$ iguais a 'e', isto é, $\forall \sigma \in \Sigma, t_\sigma = e$, tem-se a representação do autômato de estados finitos (ver Figura 0.1).

Autômatos $(\max, +)$ e séries formais

Uma das formas de utilizar a álgebra de dióides para definir a evolução dos autômatos $(\max, +)$, é usando as séries formais para descrever suas linguagens temporizadas. Assim, semelhantemente às séries formais binárias, definindo $D = \mathbb{R}_{\max} \langle \langle \Sigma \rangle \rangle$, representa-se uma linguagem temporizada por uma série formal. Deve-se observar que uma linguagem temporizada [46, 38, 47, 48, 39] é uma linguagem que apresenta em cada palavra $s \in L$ um valor numérico t_s associado, que representa um intervalo de tempo decorrido para a palavra ser reconhecida por um autômato temporizado [49, 50, 51, 52, 53, 54, 55, 41].

Definição 16 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

$$Y_L = \bigoplus_{s \in \Sigma^*} (y|s) s \quad (0.33)$$

em que $(y|s) s \in \mathbb{R}_{\max} \langle \langle \Sigma \rangle \rangle$, com $(y|s) \in \mathbb{R}_{\max}$ denotando o datador da palavra s e $\mathbb{R}_{\max} \langle \langle \Sigma \rangle \rangle$ é o semi-anel das séries formais com coeficientes em \mathbb{R}_{\max} e variáveis não comutativas em Σ , tal

que

$$(y|s) \neq \epsilon, \text{ se } s \in L, \quad (0.34)$$

$$(y|s) = \epsilon, \text{ se } s \notin L. \quad (0.35)$$

Exemplo 16 Dado o alfabeto $\Sigma = \{\alpha, \beta\}$, a linguagem

$$L = \{3\epsilon, 4\alpha, 2\alpha\beta, 3\beta\alpha, 5\alpha\alpha, 2\beta\beta, \beta\alpha\beta\}$$

é representada pela série formal

$$Y_L = 3\epsilon \oplus 4\alpha \oplus 2\alpha\beta \oplus 3\beta\alpha \oplus 5\alpha\alpha \oplus 2\beta\beta \oplus e\beta\alpha\beta \underbrace{\oplus \epsilon\alpha\alpha\alpha \oplus \dots \oplus \epsilon\beta\beta\beta \oplus \dots}_{\text{seqüências de eventos não pertencentes à } L}.$$

ou

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

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

desde que $\epsilon \otimes L = \epsilon$, $\forall L \subseteq \Sigma^*$. Esta série representa a linguagem reconhecida por um autômato $(\max, +)$. Neste contexto, $(y|s)$ denota o coeficiente da palavra s que é igual ao zero ‘ ϵ ’ se s não é reconhecida por $A_{(\max, +)}$.

Com essa formalização, é possível avaliar a evolução dinâmica dos autômatos $(\max, +)$, como citado anteriormente.

Definição 17 O mapa

$$\mu : \Sigma \rightarrow \mathbb{R}_{\max}^{|Q| \times |Q|}, \quad (0.36)$$

define a aplicação dos valores $T_{q, \sigma, q'}$ dos símbolos $\sigma \in \Sigma$ sobre $\mathbb{R}_{\max}^{|Q| \times |Q|}$, onde $\mathbb{R}_{\max}^{|Q| \times |Q|}$ é o dióide $(\mathbb{R} \cup \{-\infty\}, \max, +)$ sobre matrizes quadradas de dimensão $|Q|$.

Com a aplicação deste mapa, constrói-se a matriz

$$\mu(\sigma)_{qq'} := T_{q, \sigma, q'}. \quad (0.37)$$

Identificando θ com um vetor linha contendo os arcos de entrada e os demais elementos iguais a ϵ , e ϕ com um vetor coluna contendo os arcos de saída e os demais elementos iguais a ϵ , tem-se que:

Definição 18 Dado um autômato $A_{(\max, +)}$, com matriz

$$\mu(\sigma)_{qq'} := T_{q, \sigma, q'}, \quad (0.38)$$

e vetores linha θ e coluna ϕ , o datador da palavra $s = \sigma_1 \dots \sigma_n$ é descrito por

$$(y|s) = (A_{(\max, +)}|s) = \theta \mu(\sigma_1) \dots \mu(\sigma_n) \phi = \theta \mu(s) \phi \quad (0.39)$$

em que $\mu(s)$ é a matriz que contém os tempos de complementação das palavras formadas por $\sigma_1 \dots \sigma_n \in \Sigma$.

Denomina-se de representação linear do datador a condição

$$(A_{(\max,+)}|s) = \theta\mu(s)\phi. \quad (0.40)$$

Também, a função datadora Y pode ser escrita como uma série formal sobre o alfabeto Σ com coeficientes em \mathbb{R}_{\max} . Isto é,

Definição 19 *A linguagem temporizada de um autômato $A_{(\max,+)}$ é definida por uma série formal como*

$$L(A_{(\max,+)}) = \bigoplus_{s \in \Sigma^*} (y|s) s, \quad (0.41)$$

em que $(y|s) s \in \mathbb{R}_{\max} \langle \langle \Sigma \rangle \rangle$, com $(y|s) \in \mathbb{R}_{\max}$ denotando o datador da palavra s e $\mathbb{R}_{\max} \langle \langle \Sigma \rangle \rangle$ é o semi-anel das séries formais com coeficientes em \mathbb{R}_{\max} e variáveis não comutativas em Σ , tal que

$$(y|s) \neq \epsilon, \text{ se } s \in L, \quad (0.42)$$

$$(y|s) = \epsilon, \text{ se } s \notin L. \quad (0.43)$$

Assim, equipando a série formal com as operações de soma e produto de *Cauchy* (equações (0.16) e (0.17)) e com a operação estrela (Definição 10), e identificando o mapa datador Y com a série formal Y_L , define-se:

Definição 20 *A série formal Y_L é reconhecível se existir um autômato finito $A_{(\max,+)}$, representado pela tripla (θ, μ, ϕ) , com $\theta \in \mathbb{R}_{\max}^{1 \times |Q|}$, $\phi \in \mathbb{R}_{\max}^{|Q| \times 1}$, $\mu : \Sigma^* \rightarrow \mathbb{R}_{\max}^{|Q| \times |Q|}$ e Q finito, tal que*

$$Y_L = \bigoplus_{s \in \Sigma^*} \theta\mu(s)\phi s = \bigoplus_{s \in \Sigma^*} (y|s) s, \quad (0.44)$$

em que $\theta\mu(s)\phi = (A_{(\max,+)}|s)$ é o datador da palavra s .

Exemplo 17 *Considerando o autômato da Figura 0.6, tem-se*

$$\theta = \begin{bmatrix} e & \epsilon & \epsilon \end{bmatrix} \quad \phi = \begin{bmatrix} 2 \\ \epsilon \\ \epsilon \end{bmatrix}$$

$$\mu(\alpha) = \begin{bmatrix} \epsilon & 1 & \epsilon \\ \epsilon & \epsilon & 4 \\ \epsilon & \epsilon & \epsilon \end{bmatrix} \quad \mu(\beta) = \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ 2 & \epsilon & \epsilon \\ \epsilon & 5 & \epsilon \end{bmatrix}.$$

Calculando $(y|s)$ em um caso particular, onde $s = \alpha\beta$:

$$(y|\alpha\beta) = \theta\mu(\alpha\beta)\phi = \theta\mu(\alpha)\mu(\beta)\phi = 5.$$

Mais geralmente, a série formal Y_L identificada com a linguagem temporizada do autômato admite a expressão racional

$$L(A_{(\max,+)}) = Y_L = 2(3\alpha(9\alpha\beta)^*\beta)^* = 2 \oplus 5\alpha\beta \oplus 8\alpha\beta\alpha\beta \oplus 14\alpha\alpha\beta\beta \oplus \dots$$

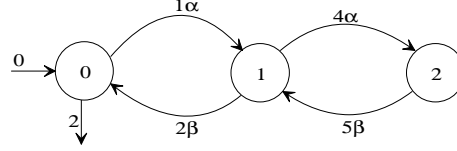


Figura 0.6: Autômato $(\max, +)$ determinístico.

isto é,

$$L(A_{(\max,+)}) = Y_L = (y|(\alpha\beta)^0)(\alpha\beta)^0 \oplus (y|(\alpha\beta)^1)(\alpha\beta)^1 \oplus (y|(\alpha\beta)^2)(\alpha\beta)^2 \oplus \dots$$

$$L(A_{(\max,+)}) = Y_L = \bigoplus_{s \in \Sigma^*} (y|s) s.$$

Assim, $L(A_{(\max,+)}) = Y_L$ apresenta todas as palavras formadas por α e β , iniciadas por α , e reconhecidas pelo autômato da Figura 0.6, com seus respectivos tempos de complementação.

Redes de Petri Temporizadas e Álgebra de Dióides

Assim como os autômatos temporizados para modelar sistemas, existem também, as redes de Petri temporizadas (RPT) [56]. Entretanto, para compreender as RPTs e aplicar a teoria dos dióides em sua descrição dinâmica, é necessário entender o que são as redes de Petri. Aqui é apresentada uma conceituação básica destas para formalizar o que é de interesse.

Redes de Petri

As redes de Petri [57, 58, 59, 60] são grafos direcionados, bi-partidos e ponderados, que apresentam uma marcação inicial. O conceito de grafo bi-partido define que uma rede de Petri tem dois tipos de nós, os quais são denominados lugares e transições. O termo grafo direcionado determina a existência de arcos, os quais são direcionados sempre de um lugar para uma transição e de uma transição sempre para um lugar. No primeiro caso, denominam-se lugar de entrada da transição e, transição de saída do lugar, e vice-versa no segundo caso. Os arcos em uma rede de Petri apresentam uma ponderação associada, onde um arco com peso k representa k arcos paralelos ligando dois nós. A marcação inicial de uma rede de Petri representa o estado em que a rede se encontra inicialmente.

Formalmente, uma rede de Petri é definida como a seguir:

Definição 21 Uma rede de Petri é uma sextupla $RP = (P, T, A_r, K, W, M_0)$, onde:

- $P = \{p_1, p_2, \dots, p_m\}$ é um conjunto finito de lugares;
- $T = \{t_1, t_2, \dots, t_n\}$ é um conjunto finito de transições;
- $A_r \subseteq (P \times T) \cup (T \times P)$ é um conjunto de arcos;
- $K : P \rightarrow \mathbb{N} \cup \{\infty\}$ é a função de capacidade;

- $W : A_r \rightarrow \mathbb{N}^+$ é a função de ponderação;
- $M_0 : P \rightarrow \mathbb{N}$ é a função de marcação inicial, que satisfaz $\forall p \in P : M_0(p) \leq K(p)$.

Deve-se observar que, pela definição de rede de Petri, as seguintes condições são satisfeitas:

$$P \cap T = \emptyset \quad (0.45)$$

e

$$P \cup T \neq \emptyset. \quad (0.46)$$

Assim, pela condição (0.45), os lugares e as transições são nós distintos, o que formaliza o termo bi-partido. Por outro lado, a condição (0.46) refere-se a que em uma rede de Petri existe pelo menos um lugar ou uma transição.

Uma rede de Petri é formada, então, por uma estrutura, formalizada pelos lugares e transições conectados pelos arcos, e uma marcação inicial.

Uma estrutura de rede de Petri é a rede sem a marcação inicial, a qual é denotada por $E_{rp} = (P, T, A_r, K, W)$. Dessa forma, considerando a definição da estrutura de rede de Petri, a notação de uma rede de Petri pode ser abreviada por $RP = \{E_{rp}, M_0\}$.

A representação gráfica de uma rede de Petri é a seguinte:

- Os lugares em uma rede de Petri são representados por círculos;
- A capacidade dos lugares são representadas pela igualdade de K ao seu valor, junto ao respectivo lugar. Quando este não aparece, o lugar apresenta capacidade infinita;
- As transições são representadas por barras ou retângulos;
- A marcação de uma rede de Petri atribui a cada lugar um número inteiro não negativo m , de onde se diz que o lugar com marcação m tem m fichas, as quais são representadas por pequenos círculos pretos. A marcação de uma rede de Petri é designada por um vetor

$$M = [m_1 \quad m_2 \quad m_3 \quad \dots \quad m_k]^T,$$

onde k é o número de lugares da rede, e m_i o número de fichas no lugar i ;

- Os arcos são sempre direcionados de uma transição para um lugar e de um lugar para uma transição;
- Os pesos dos arcos são representados por números juntos aos mesmos. Quando em um arco não aparece seu peso, então este arco tem, por definição, peso 1.

Exemplo 18 Na Figura 0.7, é apresentada uma rede de Petri. Nesta Figura, tem-se que os lugares são os círculos etiquetados por p_1 , p_2 , p_3 e p_4 e as transições são as barras etiquetadas por t_1 , t_2 e t_3 . Pela definição de rede de Petri, tem-se:

$$P = \{p_1, p_2, p_3, p_4\};$$

$$T = \{t_1, t_2, t_3\};$$

$$A_r = \{(p_1, t_1), (p_1, t_2), (p_2, t_1), (p_3, t_2), (p_4, t_3), (t_1, p_3), (t_2, p_2), (t_2, p_4), (t_3, p_2)\};$$

$$K(p_2) = 4; K(p_3) = 5$$

$$W(p_1, t_1) = 2; W(p_1, t_2) = 3; W(p_2, t_1) = 1; W(p_3, t_2) = 1; W(p_4, t_3) = 2;$$

$$W(t_1, p_3) = 1; W(t_2, p_2) = 2; W(t_2, p_4) = 3; W(t_3, p_2) = 1;$$

$$M_0 = \begin{bmatrix} 3 & 0 & 1 & 1 \end{bmatrix}^T.$$

Observe que os arcos são definidos por pares (p_i, t_j) ou (t_j, p_i) , que determinam respectivamente um arco saindo do lugar p_i para a transição t_j , e saindo da transição t_j para o lugar p_i .

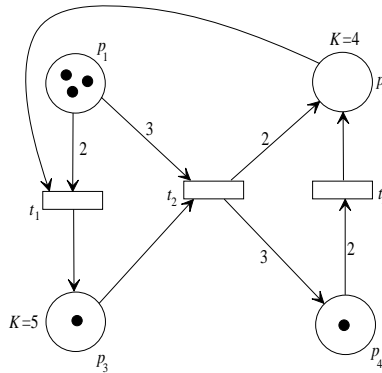


Figura 0.7: Exemplo de rede de Petri.

As redes de Petri possuem uma evolução dinâmica determinada por disparos de transições habilitadas em determinadas marcações e que modificam as marcações dos lugares. Assim, para um lugar p_i que contém um número n_i de fichas e habilita uma transição t , se esta transição dispara, há uma redução de fichas no número de fichas do lugar p_i para $n_i - w(p_i, t)$, com $w(p_i, t)$ sendo o peso do arco que liga p_i a t . Se há um lugar p_j com n_j fichas, sendo lugar de saída da transição t , então uma quantidade de fichas em p_j é aumentada para $n_j + w(t, p_j)$, em que $w(t, p_j)$ é o peso do arco ligando t a p_j . A evolução dinâmica de uma rede de Petri é formalizada através de algumas regras, as quais são apresentadas a seguir.

Definição 22 A mudança da marcação da rede de Petri segue as seguintes regras de disparo das transições:

1. Uma transição t é dita estar habilitada (pronta para disparar) em uma marcação M se e somente se

$$\forall p \in P \text{ que é entrada de } t : W(p, t) \leq M(p)$$

$$\forall p \in P \text{ que é saída de } t : M(p) \leq K(p) - W(t, p);$$

2. Uma transição habilitada pode ou não disparar;
3. O disparo de uma transição $t \in T$, habilitada na marcação M , é instantânea e resulta em uma nova marcação M' da rede dada pela equação:

$$M'(p) = M(p) - W(p, t) + W(t, p), \forall p \in P; \quad (0.47)$$

4. A ocorrência do disparo de t , que modifica a marcação M da rede para uma nova marcação M' , é denotada por $M[t\rangle M'$.

Exemplo 19 Na Figura 0.8, é visto um exemplo de uma rede de Petri com marcação inicial

$$M_0 = [2 \quad 0 \quad 0]^T,$$

e sua nova marcação, após o disparo da transição t . Observa-se que após o disparo de t , a nova marcação é

$$M_1 = [1 \quad 2 \quad 1]^T,$$

e que esta marcação está de acordo com a regra 3 de disparo das transições.

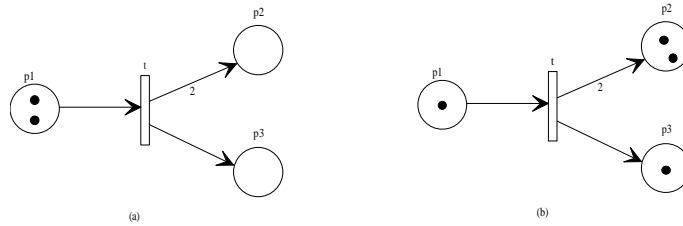


Figura 0.8: Exemplo de uma rede de Petri antes e após o disparo de sua transição.

Observação: Utiliza-se também a notação $M_k[t\rangle$ significando que t está habilitada na marcação M_k .

Define-se como grafo de eventos a rede de Petri que apresenta em cada lugar apenas um arco de saída e um arco de entrada.

As redes de Petri, apresentam várias vantagens sobre os outros modelos quando modelando sistemas. Dentre estas vantagens, podem ser citadas:

1. a facilidade para modelar as características de um sistema, ou seja, concorrência, sincronismo e assincronismo, conflito, exclusão mútua, relações de precedência, não determinismo e bloqueio;
2. excelente visualização de dependência de sistemas; focalização na informação local;
3. abordagens de modelagem do tipo refinamento e do tipo composição modular;
4. possibilidade de gerar código de controle supervisorio diretamente de sua representação gráfica;
5. possibilidade de testar propriedades indesejáveis para os sistemas, tais como bloqueio e reinicialização;

6. análise de desempenho sem simulação é possível para muitos sistemas, desde que taxas de produção, utilização de recursos, segurança e desempenho podem ser avaliadas;
7. simulação dos eventos discretos diretamente a partir do modelo e informação do estado atual do sistema que permite monitoração em tempo real [61, 62, 63, 43].

As redes de Petri possuem várias propriedades estruturais e comportamentais, além de métodos específicos de análise. Porém, as definições básicas apresentadas são as necessárias para compreender as RPTs.

Redes de Petri Temporizadas

As RPTs são redes de Petri que incluem em sua estrutura uma representação temporal. Em uma RPT os temporizadores estão associados à transições, similarmente aos autômatos temporizados. A utilização das RPTs tem sido ampla na modelagem de sistemas, como os sistemas a eventos discretos, quando considerando sua temporização. Entretanto, para utilizar a álgebra $(\max, +)$ na descrição dinâmica, é mais usado o grafo de eventos temporizado (GET). Para esta classe, a álgebra $(\max, +)$ descreve sua dinâmica em termos dos tempos de ocorrência dos eventos como um sistema de equações lineares. Dessa forma, aqui será apresentada esta classe de RPT que, por conveniência, será tratada como RPT, ao invés de GET. Deve-se observar que há outras classes de RPTs, mas aqui é tratada apenas esta classe específica.

Definição 23 *Uma rede de Petri temporizada é uma dupla*

$$RPT = (RP, \theta),$$

na qual RP é uma rede de Petri marcada e θ é uma função de tempo que associa a cada transição um valor $\theta(t_j) = d_j$ que representa o tempo associado à transição t_j .

Assim, em uma RPT, está associado a cada transição t uma duração mínima de disparo $\theta(t)$. Com a habilitação da transição, fichas são retiradas dos lugares de entrada e após decorrido o tempo $\theta(t)$, fichas são colocadas nos lugares de saída.

Exemplo 20 *Na Figura 0.9(a) é apresentada uma RPT antes do disparo da transição t . Quando a transição t torna-se habilitada, as fichas são retiradas dos lugares de entrada (Figura 0.9(b)) e, após o disparo de t , transcorridos três unidades de tempo (tempo associado à transição t), as fichas são colocadas nos lugares de saída (Figura 0.9(c)).*

Geralmente, as RPTs são representadas por outra forma, em que esta forma é construída através de uma simples transformação das transições. Este tipo de transformação é visto no exemplo a seguir:

Exemplo 21 *Na Figura 0.10 é apresentada uma transição de uma RPT, com tempo associado $\theta(t) = \mathbf{t}$ e sua transformação em uma RPT com duas transições sem temporização ($\theta(t_1) =$*

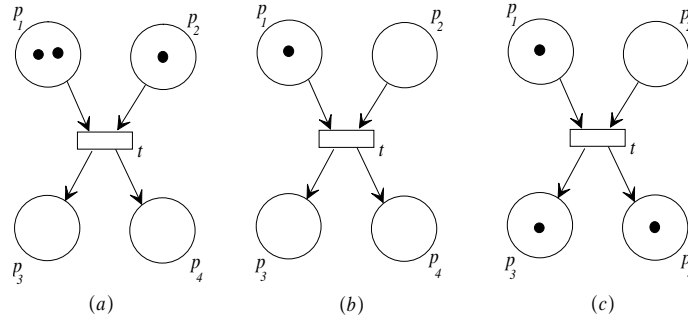


Figura 0.9: Exemplo de uma rede de Petri temporizada antes e após o disparo de sua transição.

$\theta(t_2) = 0$) e com o valor da temporização passado ao lugar introduzido como $\theta(p_1) = t$. A evolução de uma RPT que apresenta temporização em p é similar à evolução de uma RPT semelhante (transformada) com temporização em t . Isto porque o lugar introduzido simula o desaparecimento da ficha quando t está habilitada e sua guarda pelo tempo $\theta(t)$ (para a rede com temporização em t), e t_2 simula o término do tempo de guarda da ficha pela transição t .

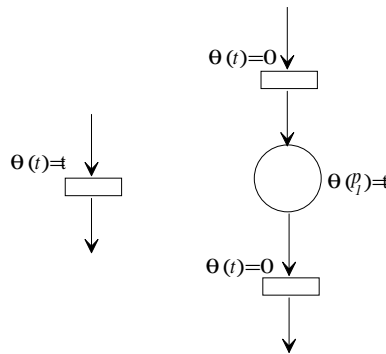


Figura 0.10: Exemplo de transformação de transições temporizadas.

A representação usual de uma RPT é formalizada como uma transformação, tal que a rede se apresente com o tempo representado internamente aos lugares como um número de barras verticais (as fichas também são colocadas internamente aos lugares). Esta transformação modifica a estrutura da RPT onde representa-se por uma barra horizontal (com etiquetas x_i) os datadores ou tempos de ocorrência das transições e os lugares por círculos com barras verticais e fichas.

Exemplo 22 Na Figura 0.11 é apresentada a representação usualmente utilizada para RPT, onde são colocados $\theta(t)$ barras verticais definindo o tempo em que a ficha permanece guardada.

Devido às temporizações nas transições, a dinâmica das RPTs necessita de uma descrição específica, onde duas estruturas concorrentes com uma única transição final (que só se torna habilitada quando as fichas das duas estruturas chegam aos respectivos lugares de entrada da transição) representa situações que exigem sincronização. Esta situação determina um formalismo algébrico para descrever a sua dinâmica.

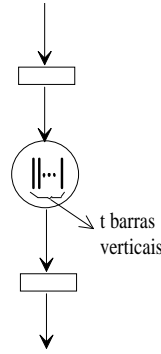


Figura 0.11: Representação utilizada para GETs.

Descrição da Dinâmica via Dióides ($\max, +$)

Situações como a de concorrência entre sub-estruturas de uma RPT, em conjunto com transições que exigem sincronização para que a transição se torne habilitada, impõem um tratamento específico para descrever a evolução dinâmica da RPT.

Uma RPT que apresente condições como esta descrita anteriormente, exige que a transição final das estruturas concorrentes, só se torne habilitada no máximo tempo decorrido entre as duas estruturas. Em geral, situações de sincronização são encontradas na evolução da RPT, tal que o n -ésimo disparo de uma transição t_i é uma função de disparos anteriores ($n - i$) de outras transições. Por exemplo, na Figura 0.12 é vista uma RPT que apresenta esta situação. Nela,

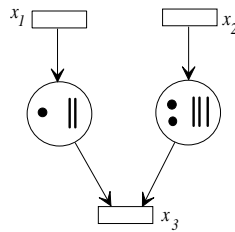


Figura 0.12: RPT.

pode-se ver que o lugar entre as transições x_2 e x_3 é temporizado em $d = 3$ unidades de tempo, e o lugar entre as transições x_1 e x_3 , em $d = 2$ unidades de tempo. O tempo necessário para o n -ésimo disparo da transição x_3 é dado por seu datador (representado por x_3), o qual é maior ou igual ao tempo máximo decorrido entre o datador x_1 (anterior a x_3 em um disparo, que é definido pela quantidade de fichas no lugar) somado com o tempo de liberação da ficha que é 2 unidades de tempo e o datador x_2 (também anterior a x_3 , só que em 2 disparos, que é definido pela quantidade de fichas no lugar) somado com o tempo de liberação da ficha que é 3 unidades de tempo. Em outros termos

$$x_3(n) \geq \max(x_1(n-1) + 2, x_2(n-2) + 3),$$

que implica em dizer que o n -ésimo tempo x_3 é o máximo entre o $(n-1)$ -ésimo tempo x_1 e o tempo de liberação da ficha e o $(n-2)$ -ésimo tempo x_2 e o tempo de liberação da ficha, nos

respectivos lugares.

Com essa representação, a função \max e a função $+$, apresentam-se na descrição da evolução dinâmica da RPT. Esta formulação pode ser escrita como um dióide $(\max, +)$, que gera uma equação matricial nesta álgebra.

Exemplo 23 Na Figura 0.13 encontra-se uma RPT, onde os círculos pretos dentro dos lugares representam a marcação inicial, e as barras representam os tempos associados para liberação das fichas nos lugares. Os datadores das transições satisfazem as seguintes desigualdades:

$$\begin{aligned} x_1(n) &\geq \max(u_1(n) + 3, x_2(n-1)) \\ x_2(n) &\geq \max(1 + u_2(n-1), 1 + x_1(n)) \\ x_3(n) &\geq \max(2 + x_3(n-1), x_1(n), x_2(n) + 1) \\ y(n) &\geq \max(x_3(n) + 3, x_2(n-1)). \end{aligned} \quad (0.48)$$

No dióide \mathbb{R}_{\max} , escreve-se (0.48) como

$$\begin{aligned} x_1(n) &\geq 3u_1(n) \oplus x_2(n-1) \\ x_2(n) &\geq 1u_2(n-1) \oplus 1x_1(n) \\ x_3(n) &\geq 2x_3(n-1) \oplus x_1(n) \oplus 1x_2(n) \\ y(n) &\geq 3x_3(n) \oplus x_2(n-1) \end{aligned} \quad (0.49)$$

que matricialmente pode ser escrita como

$$\begin{aligned} \begin{bmatrix} x_1(n) \\ x_2(n) \\ x_3(n) \end{bmatrix} &\geq \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ 1 & \epsilon & \epsilon \\ e & 1 & \epsilon \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \\ x_3(n) \end{bmatrix} \oplus \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 2 \end{bmatrix} \begin{bmatrix} x_1(n-1) \\ x_2(n-1) \\ x_3(n-1) \end{bmatrix} \oplus \begin{bmatrix} 3 & \epsilon \\ \epsilon & 1 \\ \epsilon & \epsilon \end{bmatrix} \begin{bmatrix} u_1(n) \\ u_2(n-1) \end{bmatrix} \\ y(n) &\geq \begin{bmatrix} \epsilon & \epsilon & 3 \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \\ x_3(n) \end{bmatrix} \oplus \begin{bmatrix} \epsilon & e & \epsilon \end{bmatrix} \begin{bmatrix} x_1(n-1) \\ x_2(n-1) \\ x_3(n-1) \end{bmatrix} \end{aligned}$$

que é um sistema da forma

$$\mathbf{x}(n) \geq \mathbf{A}_0 \mathbf{x}(n) \oplus \mathbf{A}_1 \mathbf{x}(n-1) \oplus \mathbf{B}_0 \mathbf{u}(n) \oplus \mathbf{B}_1 \mathbf{u}(n-1) \quad (0.50)$$

$$\mathbf{y}(n) \geq \mathbf{C}_0 \mathbf{x}(n) \oplus \mathbf{C}_1 \mathbf{x}(n-1). \quad (0.51)$$

Deve-se observar que o operador \geq é utilizado aqui, desde que na álgebra $(\max, +)$ ele coincide com \succeq , isto é, $\forall a, b \in \mathbb{R}_{\max}, a \succeq b \Rightarrow a \geq b$. Por exemplo, $3 \succ 2 \Rightarrow 3 > 2$, desde que $\max(3, 2) = 3$.

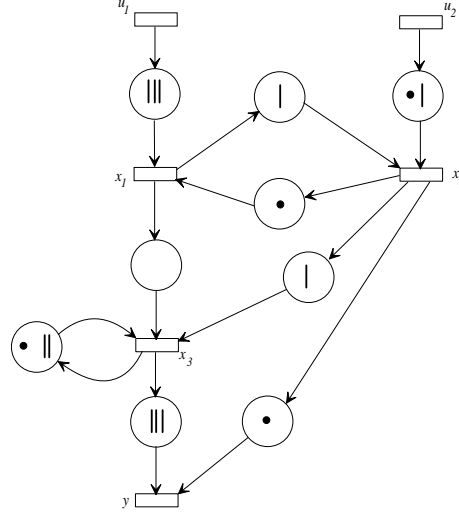


Figura 0.13: RPT para formulação da equação dinâmica na álgebra $(\max, +)$.

Álgebra $(\min, +)$

A álgebra $(\min, +)$ é o caso dual da álgebra $(\max, +)$, na qual $D = \mathbb{R}_{\min}$, onde \mathbb{R}_{\min} denota o conjunto $\mathbb{R} \cup \{+\infty\}$. Para um dióide (D, \oplus, \otimes) , com $D = \mathbb{R}_{\min}$, as operações \oplus e \otimes são: \min (mínimo) e $+$ (soma), respectivamente. O elemento nulo é definido como $\epsilon = +\infty$ e o elemento identidade é definido por $e = 0$.

Nesta álgebra, $+\infty$ é utilizado como o elemento nulo da adição, pois satisfaz:

$$\forall a \in \mathbb{R}, a \oplus \epsilon = a \oplus (+\infty) = \min \{a, +\infty\} = a = \min \{+\infty, a\} = (+\infty) \oplus a = \epsilon \oplus a. \quad (0.52)$$

Por outro lado, utiliza-se $e = 0$, como sendo o elemento identidade da multiplicação, pois satisfaz

$$\forall a \in \mathbb{R}, a \otimes e = a + 0 = a = 0 + a = e \otimes a. \quad (0.53)$$

Essa estrutura algébrica, é um dióide comutativo, pois a propriedade da comutatividade da multiplicação \otimes é satisfeita.

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

$$D = \mathbb{R}_{\min}^{n \times n}. \quad (0.54)$$

As propriedades do dióide $(\min, +)$ são igualmente satisfeitas, quando se consideram matrizes em $\mathbb{R}_{\min}^{n \times n}$.

A álgebra $(\min, +)$ pode ser utilizada de forma similar à álgebra $(\max, +)$, tanto para os autômatos temporizados como para as RPTs. A diferença é na formalização da descrição da evolução dinâmica. Enquanto na álgebra $(\max, +)$ utilizam-se os datadores $x(n)$ para descrever a evolução dinâmica, na álgebra $(\min, +)$ utilizam-se os contadores. Os contadores $x(t)$ descrevem o número de vezes que uma transição ocorre. Dessa forma, a evolução dinâmica do sistema

(RPT) descrita na álgebra $(\max, +)$ no Exemplo 23 pode ser reescrita na álgebra $(\min, +)$ como

$$\begin{aligned} x_1(t) &\leq \min(u_1(t-3), x_2(t) + 1) \\ x_2(t) &\leq \min(1 + u_2(t-1), x_1(t-1)) \\ x_3(t) &\leq \min(x_1(t), x_2(t-1), 1 + x_3(t-2)) \\ y(t) &\leq \min(1 + x_2(t), x_3(t-3)), \end{aligned} \quad (0.55)$$

que no dióide \mathbb{R}_{\min} , escreve-se

$$\begin{aligned} x_1(t) &\succeq u_1(t-3) \oplus 1x_2(t) \\ x_2(t) &\succeq 1u_2(t-1) \oplus x_1(t-1) \\ x_3(t) &\succeq x_1(t) \oplus x_2(t-1) \oplus 1x_3(t-2) \\ y(t) &\succeq 1x_2(t) \oplus x_3(t-3). \end{aligned} \quad (0.56)$$

desde que $a \oplus b = \min(a, b)$ e $a \otimes b = a + b$ e $a \succeq b$ se e somente se $a \oplus b = a$ se e somente se $a \leq b$. Então, pode-se escrever matricialmente como

$$\begin{aligned} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} &\succeq \begin{bmatrix} \epsilon & 1 & \epsilon \\ \epsilon & \epsilon & \epsilon \\ 1 & \epsilon & \epsilon \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} \oplus \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ e & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix} \begin{bmatrix} x_1(t-1) \\ x_2(t-1) \\ x_3(t-1) \end{bmatrix} \oplus \\ &\oplus \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 1 \end{bmatrix} \begin{bmatrix} x_1(t-2) \\ x_2(t-2) \\ x_3(t-2) \end{bmatrix} \oplus \begin{bmatrix} e & \epsilon \\ \epsilon & 1 \\ \epsilon & \epsilon \end{bmatrix} \begin{bmatrix} u_1(t-3) \\ u_2(t-1) \end{bmatrix} \\ y(n) &\geq \begin{bmatrix} \epsilon & 1 & \epsilon \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} \oplus \begin{bmatrix} \epsilon & \epsilon & e \end{bmatrix} \begin{bmatrix} x_1(t-3) \\ x_2(t-3) \\ x_3(t-3) \end{bmatrix} \end{aligned}$$

que é um sistema da forma

$$\begin{aligned} \mathbf{x}(t) &\geq \mathbf{A}_0\mathbf{x}(t) \oplus \mathbf{A}_1\mathbf{x}(t-1) \oplus \mathbf{A}_2\mathbf{x}(t-2) \oplus \mathbf{B}_1\mathbf{u}(t-1) \oplus \mathbf{B}_3\mathbf{u}(t-3) \\ \mathbf{y}(n) &\geq \mathbf{C}_0\mathbf{x}(t) \oplus \mathbf{C}_3\mathbf{x}(n-3). \end{aligned}$$

Considerações sobre as Matrizes

Como um sistema modelado por um autômato temporizado ou por uma RPT apresentam matrizes específicas quando trabalhando com as álgebras $(\max, +)$ ou $(\min, +)$, características próprias das matrizes podem ser calculadas para análises dos sistemas que estão modelados por estes paradigmas. Essas características são os valores próprios (auto-valores e auto-vetores) e a periodicidade da matriz. Entretanto, a formalização utilizada para esses cálculos são baseados na álgebra específica para o qual o modelo dinâmico foi construído (álgebra $(\max, +)$ ou álgebra $(\min, +)$). Com base nessas características das matrizes, podem-se calcular taxas de produção (em sistemas de manufatura), mais geralmente, fazer análises de desempenho do sistema; analisar e sintetizar controladores (que realimentem o sistema por meio de atrasos introduzidos em

sua entrada de acordo com a saída) que realizem especificações de comportamento definidas, entre outros. Isto será discutido na segunda parte desta Apostila.

Parte II

Aplicação da Álgebra de Dióides aos Sistemas a Eventos Discretos

Introdução

Os Sistemas a Eventos Discretos (SEDs) [26, 28, 64, 65, 30, 66] são sistemas que estão presentes em várias aplicações do cotidiano, como a automação de manufatura, a robótica, a supervisão de tráfego aéreo e ferroviário e as redes de computadores. Seu estudo tem crescido muito nos últimos anos devido a sua importância industrial e sua complexidade e custo [67, 68, 69, 70, 71, 72, 73, 74], o que justifica o esforço despendido em sua análise e projeto.

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

A evolução dinâmica característica destes sistemas pode ser vista na Figura 0.14, onde os eventos são representados pelas letras gregas α , β , γ , δ e ε , e os estados do sistema são representados por q_0 , q_1 , q_2 , q_3 e q_4 , em que q_0 é seu estado inicial, que é o estado do SED antes da ocorrência do primeiro evento. Neste exemplo, após a seqüência de eventos observa-se que o SED retorna ao estado inicial q_0 . Este processo é denominado *reinicialização*. Se uma seqüência de eventos resulta sempre o retorno do SED a um determinado estado, denomina-se este estado de estado recorrente (*home state*).

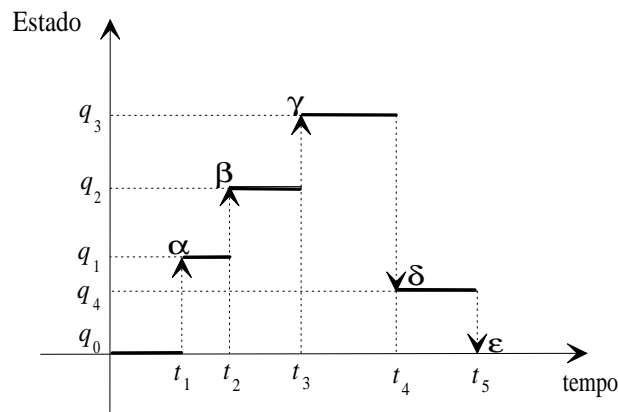


Figura 0.14: Trajetória de um SED.

Os SEDs têm como características fundamentais: 1) Ciclo de funcionamento descrito através do encadeamento de eventos; 2) Ocorrência de eventos em paralelo e 3) Necessidade de sincronização.

Devido à sua natureza de tempo discreto e às suas características, os SEDs não podem ser tratados adequadamente através dos modelos matemáticos convencionais, como as equações diferenciais. Porém, como são sistemas de alta relevância no mundo moderno, torna-se necessário encontrar soluções para problemas relacionados ao seu controle. Vários paradigmas podem

ser usados na modelagem dos SEDs, como as cadeias de Markov [75], a teoria das filas [75], os processos semi-markovianos generalizados [75], a álgebra de processos [76, 77], a teoria de linguagens formais e os autômatos [32], as redes de Petri [58, 59, 60, 78, 61] e a álgebra de dióides [8, 79]. Cada um desses paradigmas se aplicam à problemas distintos e que necessitam de uma descrição específica. Entretanto, nenhum deles se tornou um paradigma universal para solucionar o problema de controle de SEDs, abrangendo suas mais variadas classes (SEDs cíclicos e acíclicos, temporizados e não temporizados, estocásticos e outros).

O controle de um SED é definido por inibições de eventos habilitados nos estados alcançados em sua evolução dinâmica. Assim, esse controle determina a trajetória que representa um comportamento especificado. Geralmente, os SEDs apresentam em sua estrutura alguns eventos que podem ser inibidos e outros que não podem ser inibidos, o que classifica-os em duas categorias distintas, denominados eventos controláveis e eventos não controláveis. Um evento controlável é aquele em que se pode permitir ou inibir sua ocorrência (o início de atividade de uma máquina e o envio de uma mensagem) e um evento é dito não controlável quando sua ocorrência não pode ser inibida, isto é, tal evento ocorre espontaneamente (a quebra de uma esteira, o término do processamento de uma peça em uma máquina, a chegada de uma mensagem enviada).

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

Entre as formalizações de controle de SEDs, encontra-se a Teoria de Controle Supervisório (TCS) [28, 80, 31], que foi desenvolvida utilizando as linguagens formais e os autômatos [35, 32, 81] como ferramenta de modelagem do SED. Na TCS, define-se que o supervisor é o agente responsável pela avaliação dos eventos gerados pelo SED e pela determinação da ação de controle a ser aplicada para a execução de uma tarefa. A formulação genérica do problema de controle no contexto da TCS visa a determinação do autômato supervisor a partir do modelo do sistema a ser controlado e da especificação de comportamento requerida para o SED. A função do supervisor é assegurar que a linguagem gerada pela sua composição com o SED que se deseja controlar, represente a especificação de comportamento desejada, ou seja, fazer com que o SED realize uma tarefa específica. Esta teoria foi desenvolvida para construir supervisores para SEDs não temporizados. Dessa forma, sua aplicação garante condições específicas para construir supervisores onde é necessário apenas determinar a evolução do SED (comportamento lógico), não definindo instantes de tempo exatos em que os eventos devam ocorrer, nem o tempo total para a realização das tarefas.

A introdução de uma representação do tempo permite um aperfeiçoamento na modelagem e um refinamento no estudo de controle desses sistemas, possibilitando solucionar esses problemas. Para isto, a abordagem da síntese de supervisores para SEDs temporizados necessita de sua descrição através de um paradigma que relacione o tempo às habilitações dos eventos.

A representação da evolução dinâmica dos SEDs através das linguagens formais provê uma descrição similar em termos de estados alcançados. Assim, para uma seqüência de eventos $s = \alpha\beta\kappa$, há também uma seqüência de estados $p = q_i q_{i+1} q_{i+2} q_{i+3}$, que são alcançados. Considerando que existe um intervalo de tempo entre cada ocorrência consecutiva de eventos, cada estado é alcançado em um tempo determinado. A cada instante de tempo novos eventos podem se tornar habilitados e sua ocorrência, que é instantânea, muda o estado do SED.

Para a descrição da evolução dinâmica do sistema, torna-se necessário incluir um mecanismo de temporização. Neste mecanismo se considera a inclusão do *relógio global*. Este relógio é definido como um contador de tempo que está constantemente sendo incrementado em uma unidade de tempo. Também, este mecanismo associa a cada evento um intervalo de tempo mínimo entre duas ocorrências sucessivas, denominado *tempo de vida* do evento.

Cada evento tem um contador específico, que é inicializado com seu tempo de vida. Em um estado, os eventos definidos na função de transição têm seus contadores decrementados em uma unidade de tempo, para cada unidade de tempo incrementada no relógio global. Quando o contador de um evento é zerado, o evento torna-se habilitado. Este evento permanece habilitado até sua ocorrência, ou até a ocorrência de um outro evento também habilitado. A ocorrência de um evento habilitado determina uma mudança de estado no SED, e também, a reinicialização do seu contador.

Com a representação do tempo nos SEDs, vários trabalhos foram desenvolvidos no intuito de determinar formas de controle para estes sistemas. Dentre estes, encontra-se o formalismo de Brandin e Wonham [45, 82], que estende a TCS para tratar de SEDs que incluem uma representação temporizada. Neste trabalho é necessário definir os eventos forçados, proibitivos, prospectivos, remotos, ativos, elegíveis, entre outros, além da inclusão do evento denominado ‘*tick*’ que sempre ocorre sob o término de um ciclo do relógio global. Brandin e Wonham utilizam os grafos de transições de atividades (GTA) [44] para modelar os SEDs, bem como para construir composições síncronas. Entretanto, para a síntese do supervisor utilizam os grafo de transições temporizadas (GTT) [83]. Embora esta abordagem se apresente como uma ferramenta que soluciona o problema de controle de SEDs temporizados, a inclusão do evento ‘*tick*’ resulta num aumento considerável no número de estados e transições do sistema, além de ser limitada à estudos de sistemas que não apresentam sincronismo. Por outro lado, quando se modela um SED, considera-se a situação de que sempre que um evento está habilitado, ele pode ou não ocorrer. Na abordagem de Brandin e Wonham, a modelagem utiliza eventos com tempo limite superior no qual sua ocorrência é obrigatória (*hard deadline*), o que define uma aparente prévia introdução de controle no modelo do SED.

Quando se considera a sincronização de eventos, a álgebra de dióides [79] se apresenta como ferramenta matemática para a descrição da evolução dinâmica do SED temporizado. Esta álgebra permite uma descrição da evolução de SEDs temporizados cíclicos, em termos dos tempos de ocorrência dos eventos, por meio de um sistema de equações linear [7, 8]. Nesta forma de

abordar o problema de controle, há vários trabalhos desenvolvidos, utilizando como paradigma de modelagem os autômatos temporizados [51] ou os grafos de eventos temporizados (redes de Petri temporizadas [56] que apresentam para cada lugar uma marcação binária, apenas uma transição de entrada e uma transição de saída).

Na abordagem do controle de SEDs temporizados, utilizando a álgebra de dióides [1, 84, 85, 11, 68], o supervisor avalia a saída do sistema e determina a ação de controle para inibir temporariamente os eventos [86, 87, 9] de forma a atrasar as entradas do sistema para que a saída seja a especificação dada.

Também, as séries formais [19, 20, 21, 88] são muito utilizadas para esta classe de SEDs temporizados na descrição da linguagem reconhecida de SEDs modelados por autômatos $(\max, +)$ [14, 42]. Estes autômatos apresentam os tempos de vida definidos em seus arcos, representando o menor tempo em que o evento pode ocorrer. Assim, o autômato $(\max, +)$ se assemelha aos GTAs [44], considerando que a representação do tempo é definida diretamente no arco e restrita aos casos em que os limites superiores dos tempos de vida dos eventos são considerados iguais a infinito. Dessa forma, com o autômato $(\max, +)$ pode-se descrever a abordagem de Brandin e Wonham [45, 82], quando se consideram todos os eventos definidos como eventos remotos. Essa restrição determina um modelo que se apresenta com total ausência de controle. Para formalizar esta descrição e tornar seu tratamento semelhante à abordagem não temporizada, a definição dos eventos forçados não é obrigatória. O supervisor apenas deve executar uma ação de controle para o sistema realizar uma tarefa especificada num mínimo tempo definido pela especificação de comportamento.

Dessa forma, devido à semelhança entre os autômatos $(\max, +)$ e os GTAs, pode-se utilizar os autômatos $(\max, +)$ para solucionar o problema de controle da classe de SEDs temporizados descrita anteriormente. Sua utilização elimina o problema do aumento no número de estados do sistema. Também, utilizando esta mesma formalização, pode-se reescrever o problema de controle de SEDs não temporizados utilizando a álgebra de dióides aplicada às linguagens ou às séries formais [25]. A seguir é apresentada a TCS para que se compreenda o formalismo de controle de SEDs, e logo após, é apresentada a formalização de controle de SEDs temporizados através desta álgebra, e a especificação que a torna aplicável ao caso não temporizado.

Teoria de Controle Supervisório

Para o tratamento dos SEDs via TCS, é necessário compreender a formalização básica da teoria de autômatos.

Como visto anteriormente, os autômatos são modelos matemáticos de máquinas de estados que têm entradas e saídas discretas e que reconhecem um conjunto de palavras sobre um dado alfabeto Σ . Os símbolos são lidos sequencialmente e dessa forma, um autômato pode ser visto como uma entidade de controle que tem uma variável interna que representa seu estado. Assim,

cada símbolo lido resulta na atualização desta variável de acordo com uma função de transição que associa um novo estado a cada par (*evento, estado*). O conjunto de todos os estados do autômato é representado por Q . O estado inicial de um autômato é designado por q_0 . Os estados marcados, representados pelo conjunto $Q_m \subseteq Q$, são os estados que o autômato atinge ao processar palavras reconhecidas.

Definição 24 *Um autômato finito determinístico, ou simplesmente um autômato é uma quintupla $A = (Q, \Sigma, \delta, q_0, Q_m)$, onde:*

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

$$\begin{aligned} \delta(\varepsilon, q) &= q & e \\ \delta(\sigma, q) &= q', \quad \text{para } q, q' \in Q \wedge \sigma \in \Sigma; \end{aligned} \quad (0.57)$$

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

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

Estendendo a função de transição, permite-se descrever o processamento de palavras em um autômato. Assim, define-se a função de transição estendida δ^* como $\delta^* : \Sigma^* \rightarrow Q$, de tal forma que:

$$\begin{aligned} \delta^*(\varepsilon, q) &= q & e \\ \delta^*(s\sigma, q) &= \delta(\sigma, \delta^*(s, q)) \quad \text{para } q \in Q \wedge s \in \Sigma^* \end{aligned} \quad (0.58)$$

Geralmente, utiliza-se δ ao invés de δ^* , desde que $\delta^*(\sigma, q) = \delta(\sigma, \delta^*(s, q)) = \delta(\sigma, q)$ para o caso em que $s = \varepsilon$.

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

Definição 25 *Dado um autômato $A = (Q, \Sigma, \delta, q_0, Q_m)$, a linguagem marcada ou reconhecida de A , é definida por:*

$$L_m(A) = \{s | s \in \Sigma^* \wedge \delta(s, q_0) \in Q_m\}. \quad (0.59)$$

A linguagem reconhecida por um autômato pode ser encontrada, seguindo os arcos orientados de seu grafo, iniciando do estado inicial até os estados marcados, como visto na Figura 0.1.

Geradores

Um gerador é um autômato no qual a função de transição é definida para um subconjunto próprio de Σ^* , isto é, a função de transição permite uma representação de uma parte das palavras contidas em Σ^* . Os geradores são mais compatíveis com a representação dos SEDs. Embora haja esta diferenciação, costuma-se utilizar a designação de autômato para definir geradores. Assim, aqui são apresentados os geradores, mas deve-se considerar que quando se falar de autômato modelando SEDs, estará tratando de geradores.

Definição 26 *Um gerador é uma quintupla $G = (Q, \Sigma, \delta, q_0, Q_m)$, onde os elementos Q , Σ , q_0 , Q_m e δ têm a mesma definição do autômato (Definição 24), com $\delta : Q \times \Sigma \rightarrow Q$ definida como a função, geralmente parcial, de transição de estados. Isto é, para cada $q \in Q$, δ é definida apenas para um subconjunto de elementos $\sigma \in \Sigma$.*

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

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

Definição 27 *Dado um gerador $G = (Q, \Sigma, \delta, q_0, Q_m)$, associa-se a cada estado $q \in Q$ o conjunto de eventos definidos $\Sigma(q)$, dado por:*

$$\Sigma(q) = \{\sigma | \sigma \in \Sigma \wedge \delta(\sigma, q)!\} \quad (0.60)$$

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

Definição 28 *Seja um gerador $G = (Q, \Sigma, \delta, q_0, Q_m)$. Sua função de transição estendida, denotada por δ^* , é uma função $\delta^* : \Sigma^* \times Q \rightarrow Q$, tal que:*

$$\begin{aligned} \delta^*(\varepsilon, q) &= q \\ \delta^*(s\sigma, q) &= \delta(\sigma, \delta^*(s, q)), \quad \text{para } q \in Q \text{ e } s \in \Sigma^* \text{ sempre que } q' = \delta^*(s, q) \\ &\quad \text{e } \delta^*(\sigma, q') \text{ estiverem ambos definidos.} \end{aligned} \quad (0.61)$$

Definição 29 *Seja um gerador $G = (Q, \Sigma, \delta, q_0, Q_m)$. Sua linguagem gerada $L(G)$ é:*

$$L(G) = \{s | s \in \Sigma^* \wedge \delta(s, q_0)!\}. \quad (0.62)$$

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

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

$$\begin{aligned} \forall G = (Q, \Sigma, \delta, q_0, Q_m), \\ L(G) &= \overline{L(G)} \end{aligned} \quad (0.63)$$

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

Definição 30 Dado um gerador $G = (Q, \Sigma, \delta, q_0, Q_m)$, a linguagem marcada de G , denotada por $L_m(G)$, é:

$$L_m(G) = \{s \mid s \in \Sigma^* \wedge \delta(s, q_0) \in Q_m\}. \quad (0.64)$$

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

Definição 31 A linguagem *prefixo-fechada* que representa o comportamento lógico de um SED é denominada de linguagem gerada do sistema.

e,

Definição 32 A linguagem que representa o conjunto de todas as tarefas que um SED é capaz de executar, é denominada linguagem marcada do sistema.

Considerando que L^{sed} seja a linguagem gerada por um sistema e L_m^{sed} sua linguagem marcada, de acordo com as Definições 31 e 32, tem-se que a linguagem marcada L_m^{sed} contém as palavras geradas pelo SED, que também gera todos os seus prefixos. Ou seja, um SED produz as palavras contidas em $\overline{L_m^{sed}}$. Desta forma, a linguagem marcada do SED não é necessariamente *prefixo-fechada*. Assim, vê-se que

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

Dessa forma, um gerador pode representar um SED, devido à sua função de transição restrita, que é definida apenas para alguns pares (*evento, estado*) do conjunto $\Sigma \times Q$. Isto é, um SED com linguagem gerada L^{sed} e linguagem marcada L_m^{sed} , é representado por um gerador, de tal forma que $L(G) = L^{sed}$ e $L_m(G) = L_m^{sed}$.

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

Definição 33 A componente acessível de um gerador $G = (Q, \Sigma, \delta, q_0, Q_m)$, denotada por $Ac(G)$ é:

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

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

Definição 34 A componente coacessível de um gerador $G = (Q, \Sigma, \delta, q_0, Q_m)$, denotada por $Co(G)$ é:

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

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

Definição 35 Um gerador $G = (Q, \Sigma, \delta, q_0, Q_m)$ é coacessível se e somente se, toda palavra de $L(G)$ for um prefixo de uma palavra de $L_m(G)$, isto é:

$$L(G) \subseteq \overline{L_m(G)}. \quad (0.65)$$

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

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

Exemplo 24 Na Figura 0.15(a), é apresentado um gerador G . Este gerador não é acessível devido ao fato de que o estado 6 não é alcançado em nenhuma seqüência a partir do estado 0. Na Figura 0.15(b), encontra-se o mesmo gerador sem o estado 6 e seus dois arcos retirados, o que o torna um gerador acessível, em que todos os estados são alcançados. Por outro lado, para encontrar o gerador coacessível, é necessário identificar os estados de G que não são coacessíveis para o estado marcado 2. Estes estados são: 3, 4 e 5. Retirando estes estados e as transições a eles ligadas, o gerador torna-se coacessível. Note que o estado 6 não é retirado, desde que dele o estado marcado 2 é alcançado. Este gerador coacessível é visto na Figura 0.15(c). Finalmente, o gerador *trim*, está apresentado na Figura 0.15(d), onde é visto que, independentemente da ordem em que as operações de acessibilidade e de coacessibilidade são tomadas, o resultado final não é afetado. Deve-se observar que os geradores das Figuras 0.15(b), 0.15(c) e 0.15(d) são, respectivamente, a componente acessível, a componente coacessível e a componente *trim* do gerador da Figura 0.15(a).

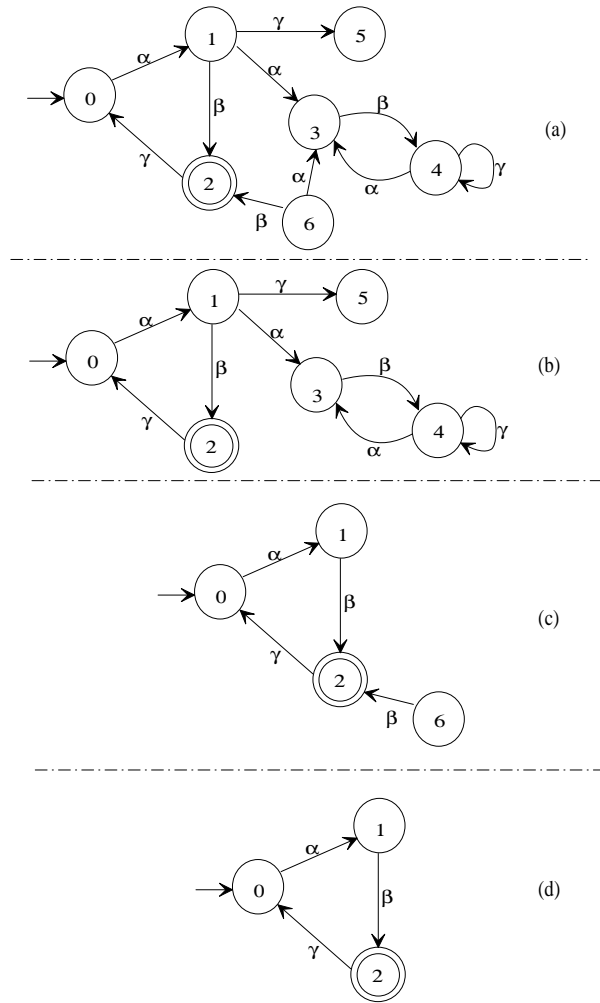


Figura 0.15: (a) Gerador e geradores (b) acessível, (c) coacessível e (d) trim.

Composição de Autômatos

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

Definição 36 Sejam $G_1 = (Q_1, \Sigma_1, \delta_1, q_{10}, Q_{m_1})$ e $G_2 = (Q_2, \Sigma_2, \delta_2, q_{20}, Q_{m_2})$ dois autômatos. O autômato G_3 resultante da composição síncrona é definido por $G_3 := (Q_3, \Sigma_3, \delta, q_{30}, Q_{m_3})$ com $Q_3 = Q_1 \times Q_2$, $\Sigma_3 = \Sigma_1 \cup \Sigma_2$, $q_{30} = (q_{10}, q_{20})$, $Q_{m_3} = Q_{m_1} \times Q_{m_2}$ e

$$\delta_3(\sigma, (q_{1_i}, q_{2_i})) = \begin{cases} (\delta_1(\sigma, q_{1_i}), \delta_2(\sigma, q_{2_i})) = (q_{1_{i'}}, q_{2_{i'}}) & \text{se } \exists \sigma, \delta_1(\sigma, q_{1_i}) = q_{1_{i'}} \text{ e } \delta_2(\sigma, q_{2_i}) = q_{2_{i'}} \\ (\delta_1(\sigma, q_{1_i}), q_{2_i}) = (q_{1_{i'}}, q_{2_i}) & \text{se } \exists \sigma, \delta_1(\sigma, q_{1_i}) = q_{1_{i'}} \text{ apenas em } G_1 \\ (q_{1_i}, \delta_2(\sigma, q_{2_i})) = (q_{1_i}, q_{2_{i'}}) & \text{se } \exists \sigma, \delta_2(\sigma, q_{2_i}) = q_{2_{i'}} \text{ apenas em } G_2 \\ \text{indefinido} & \text{caso contrário.} \end{cases} \quad (0.66)$$

Exemplo 25 A composição síncrona dos autômatos G_1 e G_2 , apresentados da Figura 0.16(a) e 0.16(b), respectivamente, onde os elementos de Σ_1 são os mesmos elementos de Σ_2 , está apresentado na Figura 0.16(c). Pode-se ver que nos dois autômatos G_1 e G_2 , só se encontra um único arco (α) que sai do estado x para o estado x , em G_1 e que, equivalentemente, sai do estado 0 para o estado 1 no autômato G_2 . Por outro lado, quando o autômato G_2 encontra-se no estado 1, só há um arco saindo do mesmo, que é equivalente no autômato G_1 , que também é o arco α . Também, o estado $(x,0)$ do autômato construído pela composição síncrona de G_1 e G_2 , não é marcado porque o estado inicial do autômato G_2 , não é. Porém, o estado $(x,1)$ do autômato construído pela composição síncrona de G_1 e G_2 é marcado, pois em ambos os autômatos, a função de transição α , leva de um estado marcado para outro estado marcado. Deve-se observar que a Figura 0.16(c) não apresenta os outros estados da composição, desde que a função de transição é indefinida (ver equação (0.66)). Naturalmente, a última condição da equação (0.66) pode ser vista como a aplicação dos operadores de acessibilidade e coacessibilidade sobre o autômato G_3 com todos os estados $q_{G_3} = (q_{G_1}, q_{G_2})$.

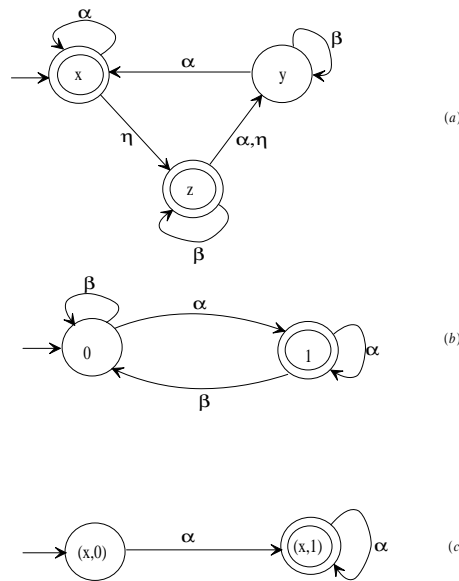


Figura 0.16: (a) Autômato G_1 , (b) Autômato G_2 e (c) Composição síncrona $G = G_1 || G_2$.

Exemplo 26 A composição síncrona entre os autômatos G_1 e G_2 , apresentados da Figura 0.17(a) e 0.17(b), respectivamente, onde somente o elemento $\eta \in \Sigma_1 \cap \Sigma_2$, está apresentado na Figura 0.17(c).

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

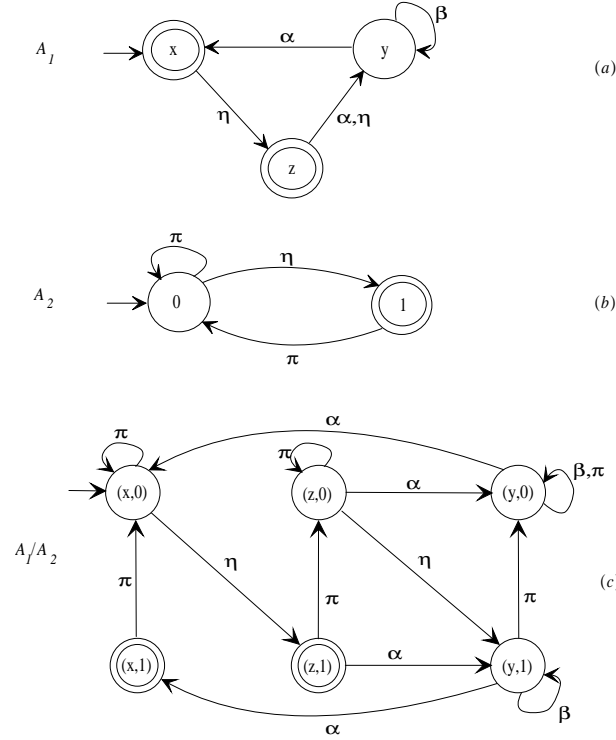


Figura 0.17: (a) Autômato G_1 , (b) Autômato G_2 e (c) Composição síncrona $G_3 = G_1 || G_2$.

A síntese do supervisor

Na TCS, os SEDs são representados por geradores, onde sua linguagem gerada é $L(G)$ e a linguagem marcada é $L_m(G)$. Nela, considera-se a existência de um agente externo para assegurar que o sistema realize uma tarefa determinada. Nesta teoria, o alfabeto de eventos Σ do gerador é dividido em duas categorias distintas: os eventos controláveis e os eventos não controláveis, representados pelos conjuntos Σ_c e Σ_{uc} , respectivamente. Assim, tem-se que o alfabeto Σ é a união de Σ_c e Σ_{uc} , ou seja $\Sigma = \Sigma_c \cup \Sigma_{uc}$. Estes dois alfabetos definidos dentro do alfabeto Σ , não têm eventos comuns, isto é, sua interseção é o conjunto vazio, ou $\Sigma_c \cap \Sigma_{uc} = \emptyset$.

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

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

Definição 37 Dado um gerador $G = (\Sigma, Q, \delta, q_0, Q_m)$, cujo alfabeto é particionado em $\Sigma = \Sigma_c \cup \Sigma_{uc}$, o conjunto de entradas de controle associado a G é definido por:

$$\Gamma = \{\gamma | \Sigma_{uc} \subseteq \gamma \subseteq \Sigma\}. \quad (0.67)$$

As entradas de controle definem quais os eventos que devem estar habilitados num determinado estado. Assim, em um estado q do gerador G , onde há eventos controláveis e eventos não

controláveis habilitados, uma entrada de controle γ aplicada nesse estado define quais os únicos eventos que podem ocorrer. Como os eventos não controláveis não sofrem ação de controle, todos os eventos não controláveis definidos neste estado estão, por definição, sempre habilitados. Apenas os eventos controláveis definidos em γ podem ocorrer (os demais eventos são inibidos).

Um gerador controlado é definido do seguinte modo:

Definição 38 Dado $\Gamma \subseteq 2^\Sigma$ como sendo o conjunto de entradas de controle, um gerador controlado G_c é um par (G, Γ) onde G é um gerador com alfabeto Σ , particionado em eventos controláveis Σ_c e eventos não controláveis Σ_{uc} , equipado com um conjunto de entradas de controle Γ .

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

Exemplo 27 Para o gerador apresentado na Figura 0.18(a), a linguagem é

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

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

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

que é representada pelo gerador visto na Figura 0.18(b).

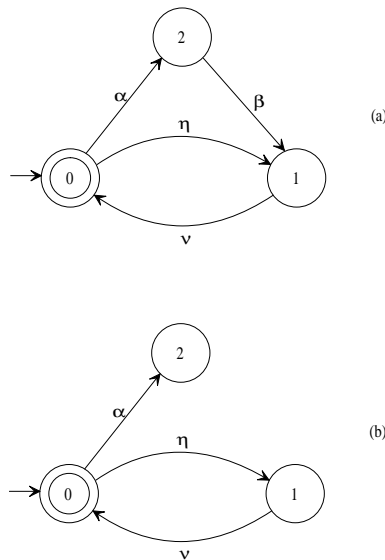


Figura 0.18: (a) Gerador com todos os eventos habilitados e (b) Gerador com o evento β inibido.

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

Supervisores e condições de existência

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

Definição 39 Um supervisor para um gerador controlado $G_c = (G, \Gamma)$ é um par $\mathcal{S} = (S, \Theta)$, composto de um gerador $S = (\Sigma, X, \xi, x_0, X_m)$ e de um mapa de controle Θ , em que:

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

Na Figura 0.19, é mostrado como é representado o sistema composto pelo gerador controlado G_c supervisionado pelo supervisor \mathcal{S} , em malha fechada. Pode-se ver que o gerador controlado recebe a ação de controle e gera os eventos que são observados pelo supervisor. Essa composição determina que o gerador siga a linguagem do supervisor.

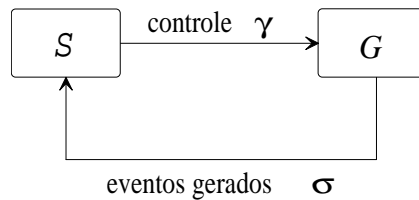


Figura 0.19: Supervisão de um SED.

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

Definição 40 Dados um gerador controlado G_c e um supervisor \mathcal{S} , a linguagem gerada pelo sistema supervisionado, denotada por $L(\mathcal{S}/G)$, é tal que

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

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

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

Deve-se observar que a composição síncrona entre o supervisor e o gerador define a linguagem do sistema supervisionado. Esta linguagem representa a trajetória que o sistema deve seguir. Os eventos habilitados em cada estado do supervisor são os únicos eventos que podem ocorrer no estado correspondente do gerador. Os outros eventos possíveis de ocorrerem no respectivo estado do gerador são inibidos, o que é determinado pela entrada de controle do supervisor. Esta condição pode ser vista na composição síncrona do supervisor com o gerador, observando os pares $q_{S/G} = (q_S, q_G)$, onde $q_{S/G}$ é o estado q da composição síncrona \mathcal{S}/G , q_S é o estado q do supervisor \mathcal{S} e q_G é o estado q do gerador G . Isto é visto na Definição 40, em que somente os eventos que estão habilitados em cada estado, são os que ocorrem sob supervisão.

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

$$L(\mathcal{S}/G) \subseteq L(G). \quad (0.68)$$

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

Define-se a linguagem controlada do sistema supervisionado por

$$L_c(\mathcal{S}/G) = L(\mathcal{S}/G) \cap L_m(G), \quad (0.69)$$

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

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

Dessa forma, a linguagem marcada do sistema é

$$L_m(\mathcal{S}/G) = L_c(\mathcal{S}/G) \cap L_m(S) \quad (0.70)$$

Assim, vê-se que

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

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

Supervisores próprios

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

Definição 41 *Um supervisor \mathcal{S} é dito ser completo, em relação a um gerador G_c , quando o seguinte é verdadeiro: para todo $s \in \Sigma^*$ e $\sigma \in \Sigma$ as três condições*

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

juntas implicam em $\xi(s, x_0)!$, isto é, $\xi(s, x_0)$ é definido.

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

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

Definição 42 *Um supervisor \mathcal{S} é dito não bloqueável se e somente se*

$$\overline{L_c(\mathcal{S}/G)} = L(\mathcal{S}/G) \quad (0.71)$$

Definição 43 *Um supervisor \mathcal{S} é dito não rejeitável se e somente se*

$$\overline{L_m(\mathcal{S}/G)} = \overline{L_c(\mathcal{S}/G)}. \quad (0.72)$$

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

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

Definição 44 *Um supervisor completo, isto é, não bloqueável e não rejeitável é dito supervisor próprio se*

$$\overline{L_m(\mathcal{S}/G)} = \overline{L_c(\mathcal{S}/G)} = L(\mathcal{S}/G). \quad (0.73)$$

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

Definição 45 *Sejam duas linguagens $K, L \subseteq \Sigma^*$. K é dita fechada em relação a L , ou L -fechada, se e somente se*

$$K = \overline{K} \cap L \quad (0.74)$$

Definição 46 Dadas duas linguagens $K, L \subseteq \Sigma^*$ e um alfabeto $\Sigma = \Sigma_c \cup \Sigma_{uc}$, diz-se que K é L -controlável se e somente se

$$\overline{K}\Sigma_{uc} \cap L \subseteq \overline{K}. \quad (0.75)$$

Para determinar as condições de existência do supervisor para a realização de uma tarefa definida, é dada a seguinte proposição:

Proposição 2 Seja \mathcal{S} um supervisor completo para G_c . Então $L(\mathcal{S}/G)$ é prefixo-fechada e $L(G)$ -controlável.

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

Teorema 2 Dados um gerador G tal que $L(G)$ represente seu comportamento fisicamente possível e uma linguagem especificada $K \subseteq L(G)$, existe um supervisor completo \mathcal{S} tal que $L(\mathcal{S}/G) = K$ se e somente se K for prefixo-fechada e $L(G)$ -controlável.

Teorema 3 Dados um gerador G tal que $L_m(G)$ represente as tarefas que podem ser completadas pelo sistema na ausência de qualquer ação de controle e uma linguagem especificada $K \subseteq L_m(G)$ então

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

Para problemas formulados em termos de linguagens marcadas, se K satisfizer as condições do Teorema 3, então o supervisor será tal que $L(\mathcal{S}/G) = \overline{K}$, visto que nesse caso,

$$L_c(\mathcal{S}/G) = \overline{K} \cap L_m(G) = K. \quad (0.76)$$

Estes resultados só podem ser empregados quando a linguagem especificada K satisfaz as condições exigidas. Quando a linguagem K não satisfaz as condições exigidas à existência do supervisor, ou seja, a linguagem especificada K não é nem $L_m(G)$ -fechada, nem $L(G)$ -controlável, é necessário encontrar uma sublinguagem $K^\uparrow \subseteq K$, que satisfaça todas essas condições. Esta sublinguagem é conhecida por *Suprema Sublinguagem Controlável* K^\uparrow , ou $\sup C(L)$, que soluciona o problema do supervisor, restritivamente. Sendo assim, se K^\uparrow solucionar de forma satisfatória o dado problema, isto é, se $K \supseteq L_A$, onde L_A é uma linguagem que representa o comportamento mais restrito que pode ser tolerado, K^\uparrow pode ser utilizada em substituição à linguagem anteriormente especificada. A solução para este problema é um supervisor que implementa K^\uparrow . Para o caso dos geradores de estado finito, K^\uparrow é sempre computável.

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

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

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

$$L_A \subseteq L_c(\mathcal{S}/G) \subseteq E. \quad (0.77)$$

Quando E é $L_m(G)$ -fechada e $L(G)$ -controlável, a existência de um supervisor tal que $L_c(\mathcal{S}/G) = E$ é garantida, o que significa que o problema tem uma solução não restritiva, como visto anteriormente. Nos casos em que E não satisfaz estas condições, é possível obter uma solução minimamente restritiva. Esta solução pode ser encontrada utilizando o algoritmo a seguir, onde se consideram um gerador de estado finito G descrito por L (comportamento do SED) e K (comportamento desejado):

Algoritmo 1 Algoritmo para a Construção de K^\uparrow ([27])

- *Dados o gerador G trim e o gerador H (composição síncrona do gerador G com a especificação desejada), faça:*

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

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

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

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

Exemplo 28 Sejam $\Sigma = \{\alpha_1, \alpha_2, \beta\}$, $\Sigma_{uc} = \{\beta\}$, e em notação de expressões regulares

$$L(G) = \overline{(\alpha_1\beta^2 + \alpha_2)}\beta^*,$$

$$L(H) = \overline{\alpha_1\beta^2} + \alpha_2\beta^*,$$

que são a linguagem do gerador trim visto na Figura 0.20 e a especificação de comportamento H desejada para este gerador. Utilizando o algoritmo da $\text{sup } C(L)$ apresentado por Ramadge

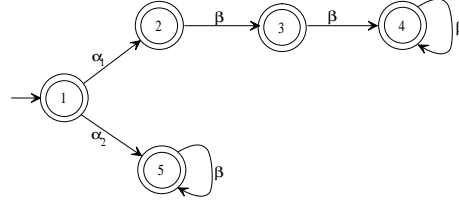


Figura 0.20: Autômato gerador de $L(G) = \overline{(\alpha_1\beta^2 + \alpha_2)}\beta^*$.

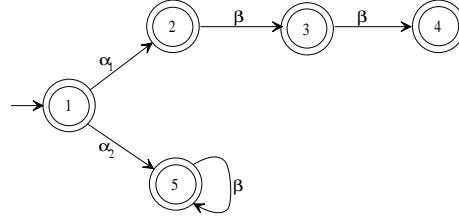


Figura 0.21: Especificação $L(H) = \overline{\alpha_1\beta^2} + \alpha_2\beta^*$.

e Wonham [27], inclui-se separadamente à matriz de transição do gerador da especificação de comportamento, visto na Figura 0.21, duas colunas: uma listando $\Sigma(H(x)) \cap \Sigma_{uc}$, outra listando $\Sigma(x)$. Isto define a seguinte tabela:

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

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

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

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

$H_2 :$		1 2 5	$\Sigma(H(x)) \cap \Sigma_{uc}$	$\Sigma(x)$	$L(H_2) = \overline{\alpha_1} + \alpha_2\beta^*$
	1	α_1 α_2		$\alpha_1\alpha_2$	
	2		β		
	5	β	β	β	

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

em que, $L(H_3)$ é a $\text{sup } C(L)$, cujo supervisor está apresentado na Figura 0.22.

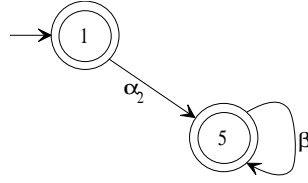


Figura 0.22: Supervisor para a $\text{sup } C(L) = L(H_3) = \alpha_2\beta^*$.

Exemplo 29 Considere o autômato apresentado na Figura 0.23 e a especificação de comportamento apresentada na Figura 0.24. A linguagem marcada do autômato é

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

e a linguagem marcada da especificação de comportamento é

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

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

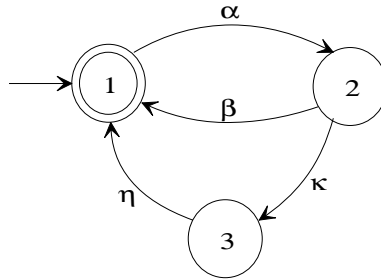


Figura 0.23: Autômato com linguagem $L_m(G) = (\alpha\beta)^* + (\alpha\kappa\eta)^*$.

a $\text{sup } C(L)$, constrói-se a composição síncrona $H||G$, a qual é vista na Figura 0.25. Com a linguagem $L(H') = L(H/G)$ dessa composição, utiliza-se o algoritmo para construção da

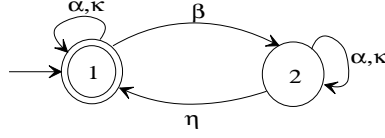


Figura 0.24: Especificação $L(H) = (\alpha + \kappa)^* \beta (\alpha + \kappa)^* \eta$.

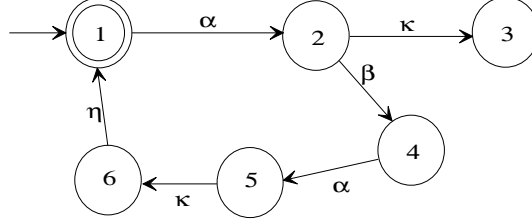


Figura 0.25: Composição síncrona $H||G$.

$\sup C(L)$, em que inclui-se separadamente à matriz de transição do gerador da especificação de comportamento e as duas colunas listando $\Sigma(H'(x)) \cap \Sigma_{uc}$ e $\Sigma(x)$. Isto define a seguinte tabela:

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

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

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

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

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

e H'_1 é coacessível, então, $L(H'_1)$ é a $\sup C(L)$ que é a linguagem do supervisor visto na Figura 0.26.

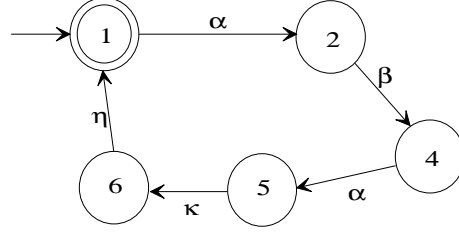


Figura 0.26: Supervisor para $\sup C(L) = L(H'_1) = \alpha\beta\alpha\kappa\eta$.

Síntese de Supervisores de SEDs Temporizados modelados por Autômatos $(\max,+)$ Utilizando Dióides

A aplicação dos autômatos $(\max,+)$ para o caso específico de SEDs temporizados que apresentam todos os tempos de vida definidos como os tempos mínimos para sua habilitação, considera igualmente à TCS que $\Sigma = \Sigma_c \cup \Sigma_{uc}$. Para este formalismo, é necessário a definição da matriz de incidência associada a um autômato $(\max,+)$.

Definição 47 *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

$$\theta(\mathbf{At}) = \begin{bmatrix} t_{in} & \epsilon & \dots & \epsilon \end{bmatrix},$$

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

$$\phi(\mathbf{At}) = \begin{bmatrix} t_{m_1} & t_{m_2} & \dots & t_{m_n} \end{bmatrix}^T,$$

com t_{m_i} representando os atrasos finais.

Nessa representação, cada termo $at_{i,j}$ define a mudança do estado designado pela linha i para o estado designado pela coluna j . Se existe mais do que um evento causando a mudança de estado entre esses vértices, o termo $at_{i,j}$ é uma expressão regular. Os vetores $\theta(\mathbf{At})$ e $\phi(\mathbf{At})$ definem as informações sobre o estado inicial e os estados marcados. Representa-se por $\theta_j(\mathbf{At})$ o elemento da j -ésima coluna do vetor $\theta(\mathbf{At})$ e por $\phi_i(\mathbf{At})$ o elemento da i -ésima linha do vetor $\phi(\mathbf{At})$. Os atrasos finais são os valores t_{m_i} , em que se um estado k não é marcado, $t_{m_k} = \epsilon$.

Exemplo 30 O autômato $A_{(\max,+)}$ apresentado na Figura 0.27, tem sua matriz de incidência temporizada construída de acordo com a Definição 47, 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

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

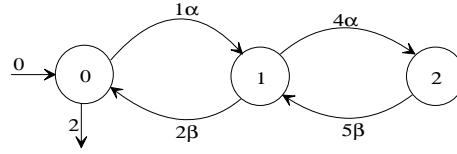


Figura 0.27: Autômato $(\max,+)$ determinístico para ilustrar a representação por matriz de incidência temporizada e vetores de estados inicial e marcados.

Deve-se observar que a matriz de incidência temporizada pode ser construída por

$$\mathbf{At} = \bigoplus_{i=1}^n \mu(\sigma_i) \otimes \sigma_i$$

com $\mu(\sigma_i)$ dada pela Definição 17, e σ_i sendo o i -ésimo evento do alfabeto Σ . Sendo assim, para a representação de um autômato $A_{(\max,+)}$ por uma matriz de incidência temporizada e pelos seus vetores de estados, a descrição em termos das séries formais é válida, podendo-se através dessa representação, determinar a linguagem temporizada reconhecida pelo autômato $A_{(\max,+)}$. No caso da representação matricial, define-se:

Definição 48 A um autômato $A_{(\max,+)}$ que tem sua linguagem reconhecida temporizada representada pela série formal

$$Y_L = \bigoplus_{s \in \Sigma^*} (y|s) s, \quad (0.78)$$

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

$$L_m(\mathbf{At}) = L(A_{(\max,+)}) = Y_L. \quad (0.79)$$

Semelhantemente à construção das linguagens das matrizes de incidência não temporizadas, podem-se definir as linguagens da matriz de incidência temporizada. Para isto, define-se:

Definição 49 *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*

$$\mathbf{At}^n = \mathbf{At} \otimes \mathbf{At} \otimes \cdots \otimes \mathbf{At}, \quad (0.80)$$

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 31 *Do autômato $A_{(\max,+)}$ mostrado na Figura 0.28, sua representação matricial é dada por*

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

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

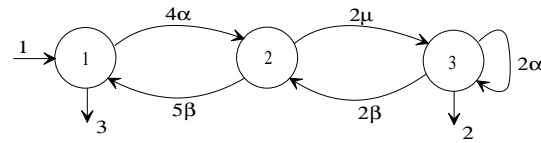


Figura 0.28: Autômato para exemplo da matriz de caminhos temporizada.

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

com $\theta(\mathbf{At}^2) = \theta(\mathbf{At})$ e $\phi(\mathbf{At}^2) = \phi(\mathbf{At})$. Sua matriz de caminhos \mathbf{At}^3 é

$$\mathbf{At}^3 = \begin{bmatrix} \epsilon & 13\alpha\beta\alpha + 8\alpha\mu\beta & 8\alpha\mu\alpha \\ 9\mu\beta\beta + 14\beta\alpha\beta & 6\alpha\mu\beta & 6\mu\alpha\alpha \\ 9\alpha\beta\beta & 11\beta\beta\alpha + 6\alpha\alpha\beta + 6\beta\mu\beta & 6\alpha\beta\mu + 6\alpha\alpha\alpha + 6\beta\mu\alpha \end{bmatrix}$$

com $\theta(\mathbf{At}^3) = \theta(\mathbf{At})$ e $\phi(\mathbf{At}^3) = \phi(\mathbf{At})$. Nessas matrizes de caminhos, cada seqüência representa uma palavra de comprimento 2 e 3, respectivamente, que muda o estado do autômato $A_{(\max,+)}$, do estado i para o estado j . Em \mathbf{At}^2 , vê-se que do estado inicial para o estado inicial, existe a palavra $\alpha\beta$, com tempo $t_{\alpha\beta} = 9$, e que é uma palavra reconhecida, pois

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

de acordo com a Definição 18.

Com essa definição e utilizando a álgebra de dióides, tem-se que a linguagem da matriz de incidência temporizada $L(\mathbf{At})$ é definida por:

Definição 50 Para uma dada matriz de incidência temporizada \mathbf{At} , sua linguagem é definida como

$$L(\mathbf{At}) = \bigoplus_i (\theta(\mathbf{At}) \otimes \mathbf{At}^i) = \bigoplus_i \bigoplus_{j=1}^n (\theta_1(\mathbf{At}) \otimes at_{1,j}^i), \quad (0.81)$$

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 .

Por outro lado, além das séries formais, a linguagem marcada da matriz de incidência \mathbf{At} também pode ser definida como sendo:

Definição 51 Para uma matriz de incidência \mathbf{At} , sua linguagem marcada é definida como

$$L_m(\mathbf{At}) = \bigoplus_i (\theta(\mathbf{At}) \otimes \mathbf{At}^i \otimes \phi(\mathbf{At})) = \bigoplus_i \bigoplus_{j=1}^n (\theta_1(\mathbf{At}) \otimes at_{1,j}^i \otimes \phi_j(\mathbf{At})), \quad (0.82)$$

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 32 Seja o autômato apresentado na Figura 0.29. Sua representação matricial é

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

Para determinar sua linguagem, calcula-se

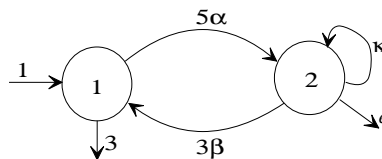


Figura 0.29: Autômato temporizado para determinação da linguagem temporizada.

$$\begin{aligned} \mathbf{At}^2 &= \begin{bmatrix} 8\alpha\beta & 5\alpha\kappa \\ 3\kappa\beta & 8\beta\alpha + \kappa\kappa \end{bmatrix} \\ \mathbf{At}^3 &= \begin{bmatrix} 8\alpha\kappa\beta & 13\alpha\beta\alpha + 5\alpha\kappa\kappa \\ 11\beta\alpha\beta + 3\kappa\kappa\beta & 8\kappa\beta\alpha + 8\beta\alpha\kappa + \kappa\kappa\kappa \end{bmatrix} \\ &\vdots \end{aligned}$$

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 $L_m(\mathbf{At})$.

Para um autômato $A_{(\max,+)}$ representado por uma matriz de incidência \mathbf{At} , a acessibilidade e coacessibilidade são definidas como a seguir:

Definição 52 Uma linha j de uma matriz de incidência \mathbf{At} é acessível se para algum $i \in \mathbb{N}^*$,

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

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

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

Definição 53 Uma linha i de uma matriz de incidência \mathbf{At} é coacessível se para algum $k \in \mathbb{N}^*$,

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

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

Dessa forma, uma linha i é coacessível se, partindo da linha i , existe pelo menos uma seqüência $s \neq \epsilon$ que leva à linha marcada j . Também, uma matriz de incidência \mathbf{At} é acessível se toda linha j é acessível. Por outro lado, uma matriz de incidência \mathbf{At} é coacessível se toda linha i é coacessível. Uma matriz de incidência \mathbf{At} é *trim* se todas as linhas forem acessíveis e coacessíveis ao mesmo tempo.

Exemplo 33 No autômato da figura 0.29, todos os estados são acessíveis, pois

$$\theta(\mathbf{At}) \otimes \mathbf{At}^1 \otimes \pi = 6\alpha \neq \epsilon, \quad \text{para } \pi = \begin{bmatrix} \epsilon & e \end{bmatrix}^T,$$

que leva do estado 1 ao estado 2 e

$$\theta(\mathbf{At}) \otimes \mathbf{At}^2 \otimes \pi = 9\alpha\beta \neq \epsilon, \quad \text{para } \pi = \begin{bmatrix} e & \epsilon \end{bmatrix}^T,$$

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

$$\phi(\mathbf{At}) = \begin{bmatrix} 3 & e \end{bmatrix}^T,$$

$$\begin{aligned} \nu \otimes \mathbf{At}^1 \otimes \phi(\mathbf{At}) &= 5\alpha \neq \epsilon, \quad \text{para } \nu = \begin{bmatrix} e & \epsilon \end{bmatrix} \\ \nu \otimes \mathbf{At}^1 \otimes \phi(\mathbf{At}) &= 6\beta \neq \epsilon, \quad \text{para } \nu = \begin{bmatrix} \epsilon & e \end{bmatrix} \end{aligned}$$

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

Similarmente à composição síncrona de autômatos utilizada na TCS, considerando os autômatos $(\max, +)$, nessa abordagem é formalizada considerando que os tempos de vida dos eventos satisfazem às operações de soma e produto de *Cauchy* (equações (0.16) e (0.17)) para $D = \mathbb{R}_{\max}$. Para isto, as seguintes definições são necessárias:

Definição 54 Dado um dióide $D = \mathbb{R}_{\max} \langle \langle \Sigma \rangle \rangle$, o operador \otimes define a interseção de elementos, como

$$at \otimes bt = \bigoplus_{i=1}^k (t_{\sigma^i} \oplus t'_{\sigma^i}) \sigma^i, \quad \text{se } t_{\sigma^i} \sigma^i \subset at \wedge t'_{\sigma^i} \sigma^i \subset bt. \quad (0.83)$$

$\forall at, bt \in D$, onde

$$\begin{aligned} t_{\sigma} \sigma \otimes \epsilon &= \epsilon \\ t_{\sigma^1} \sigma^1 \otimes t'_{\sigma^2} \sigma^2 &= \epsilon, \quad \sigma^1 \neq \sigma^2 \\ t_{\sigma} \sigma \otimes t'_{\sigma} \sigma &= (t_{\sigma} \oplus t'_{\sigma}) \sigma, \end{aligned} \quad (0.84)$$

$\forall \sigma, \sigma^1, \sigma^2 \in \Sigma$ e $\forall t_{\sigma}, t_{\sigma^1}, t'_{\sigma}, t'_{\sigma^2} \in \mathbb{R}_{\max}$. Para matrizes de incidência temporizadas, esse operador é definido como

$$\mathbf{C} = \mathbf{At} \otimes \mathbf{Bt}, \quad c_{i,j} = at_{i,j} \otimes bt_{i,j}, \quad (0.85)$$

para $at_{i,j} \otimes bt_{i,j}$ definido de acordo com 0.83, e

$$\theta(\mathbf{C}) = \theta(\mathbf{At}) \otimes \theta(\mathbf{Bt}) \quad \text{e} \quad \phi(\mathbf{C}) = \phi(\mathbf{At}) \otimes \phi(\mathbf{Bt}).$$

Observa-se que, tornando $\sigma = \sigma' = e$ (palavra vazia), a interseção satisfaz as condição da Definição 54, ou seja,

$$\begin{aligned} te \otimes \epsilon &= t \otimes \epsilon = \epsilon, \quad \text{e} \\ te \otimes t'e &= (t \oplus t')e = (t \oplus t'), \end{aligned}$$

desde que a segunda equação só é válida para $\sigma^1 \neq \sigma^2$. Dessa forma, para o caso dos vetores de estados θ e ϕ , a operação de interseção \otimes só retorna um valor diferente de ϵ quando todos

os elementos operados são diferentes de ϵ . No caso do vetor de estado inicial, esse operador torna-se igual ao operador \oplus , desde que apenas o primeiro elemento dos dois vetores é diferente de ϵ , o que satisfaz

$$\begin{aligned}\theta_1(\mathbf{C}) &= \theta_1(\mathbf{At}) \oplus \theta_1(\mathbf{Bt}) & \text{e} \\ \theta_k(\mathbf{C}) &= \epsilon & 1 < k \leq n.\end{aligned}$$

Para o caso específico de controle de SEDs, essa definição deve ser restringida, devido ao fato de não se poder aplicar controle a eventos não controláveis, impondo atrasos. Assim, tem-se:

Definição 55 *Considerando que \mathbf{At} representa uma especificação de comportamento (tempos dos eventos variáveis) para um SED temporizado modelado por uma matriz de incidência \mathbf{Bt} (tempos dos eventos não controláveis constantes), em que $at_{i,j}$ e $bt_{i,j}$ podem ser expressões regulares do tipo $t_{\sigma^1}\sigma^1 + \dots + t_{\sigma^k}\sigma^k$ e $t'_{\sigma^1}\sigma^1 + \dots + t'_{\sigma^k}\sigma^k$, respectivamente, então o operador de interseção \otimes é definido de acordo com a equação (0.85), em que $at_{i,j} \otimes bt_{i,j}$ é definido como*

$$at_{i,j} \otimes bt_{i,j} = \begin{cases} \bigoplus_{i=1}^k (t_{\sigma^i} \oplus t'_{\sigma^i}) \sigma^i & \text{se } (t_{\sigma^i} \sigma^i \subset a) \wedge (t'_{\sigma^i} \sigma^i \subset b) \wedge (\sigma^i \in \Sigma_c) \\ \bigoplus_{i=1}^k t'_{\sigma^i} \sigma^i & \text{se } (t_{\sigma^i} \sigma^i \subset a) \wedge (t'_{\sigma^i} \sigma^i \subset b) \wedge (\sigma^i \in \Sigma_{uc}). \end{cases} \quad (0.86)$$

Com a Definição 55, o operador \otimes pode ser utilizado para os procedimentos da síntese do supervisor, como será visto adiante. Isso, porque quando se considera $\Sigma = \Sigma_c \cup \Sigma_{uc}$, tanto na composição síncrona de dois autômatos quanto em outras operações, o tempo de um evento controlável pode ser modificado (apenas ampliado) através da interferência de um supervisor. Porém, no modelo do SED, em hipótese alguma, pode haver variação no tempo de vida de um evento não controlável. Dessa maneira, o produto síncrono de duas matrizes de incidência temporizadas é definido como a seguir:

Definição 56 *Dadas duas matrizes de incidência temporizadas $\mathbf{At}_{m \times m}$ e $\mathbf{Bt}_{n \times n}$, as quais representam o autômato $A_{(\max,+)_1}$ construído com símbolos do alfabeto Σ_1 e o autômato $A_{(\max,+)_2}$ construído com símbolos do alfabeto Σ_2 , respectivamente, compõe-se uma nova matriz de incidência \mathbf{Pt} , com dimensão $m \times n$, construída com símbolos dos alfabetos Σ_1 e Σ_2 , através da composição paralela definida como:*

$$\mathbf{Pt} = \mathbf{At} || \mathbf{Bt}, pt_{k,l} = at_{i_{\mathbf{At}}, j_{\mathbf{At}}} \otimes bt_{i_{\mathbf{Bt}}, j_{\mathbf{Bt}}},$$

onde $k = (i_{\mathbf{At}}, i_{\mathbf{Bt}}) = i_{\mathbf{At}} + m(i_{\mathbf{Bt}} - 1)$ e $l = (j_{\mathbf{At}}, j_{\mathbf{Bt}}) = j_{\mathbf{At}} + m(j_{\mathbf{Bt}} - 1)$, de tal forma que

$$P_{mn \times mn} = \begin{bmatrix} bt_{1,1} \otimes [at_{1,1} at_{1,2} \dots at_{1,n}] & \dots & bt_{1,m} \otimes [at_{1,1} at_{1,2} \dots at_{1,n}] \\ \vdots & \vdots & \vdots \\ bt_{1,1} \otimes [at_{n,1} at_{n,2} \dots at_{n,n}] & \dots & bt_{1,m} \otimes [at_{n,1} at_{n,2} \dots at_{n,n}] \\ bt_{2,1} \otimes [at_{1,1} at_{1,2} \dots at_{1,n}] & \dots & bt_{2,m} \otimes [at_{1,1} at_{1,2} \dots at_{1,n}] \\ \vdots & \vdots & \vdots \\ bt_{2,1} \otimes [at_{n,1} at_{n,2} \dots at_{n,n}] & \dots & bt_{2,m} \otimes [at_{n,1} at_{n,2} \dots at_{n,n}] \\ \vdots & \vdots & \vdots \\ bt_{m,1} \otimes [at_{1,1} at_{1,2} \dots at_{1,n}] & \dots & bt_{m,m} \otimes [at_{1,1} at_{1,2} \dots at_{1,n}] \\ \vdots & \vdots & \vdots \\ bt_{m,1} \otimes [at_{n,1} at_{n,2} \dots at_{n,n}] & \dots & bt_{m,m} \otimes [at_{n,1} at_{n,2} \dots at_{n,n}] \end{bmatrix}$$

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

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

com $k = (i_{\mathbf{At}}, i_{\mathbf{Bt}}) = i_{\mathbf{At}} + m(i_{\mathbf{Bt}} - 1)$. O vetor de estado inicial é definido por

$$\begin{aligned} \theta_{1,m \times n}(\mathbf{Pt}) &= \begin{bmatrix} \theta_1(\mathbf{At}) \otimes \theta_1(\mathbf{Bt}) & \epsilon & \dots & \epsilon \\ \theta_1(\mathbf{At}) \oplus \theta_1(\mathbf{Bt}) & \epsilon & \dots & \epsilon \end{bmatrix} \\ &= \begin{bmatrix} \theta_1(\mathbf{At}) \otimes \theta_1(\mathbf{Bt}) & \epsilon & \dots & \epsilon \\ \theta_1(\mathbf{At}) \oplus \theta_1(\mathbf{Bt}) & \epsilon & \dots & \epsilon \end{bmatrix}. \end{aligned}$$

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

$$\mathbf{Pt}_{m,n \times m,n} = (\mathbf{At} || \mathbf{Bt}) \oplus \mathbf{Ct}^\neg,$$

onde

$$\mathbf{Ct}^\neg = \begin{bmatrix} \mathbf{At}^{\neg \mathbf{Bt}} \oplus \mathbf{Dt}_{bt_{1,1}}^{\neg \mathbf{At}} & \mathbf{Dt}_{bt_{1,2}}^{\neg \mathbf{At}} & \dots & \mathbf{Dt}_{bt_{1,n}}^{\neg \mathbf{At}} \\ \mathbf{Dt}_{bt_{2,1}}^{\neg \mathbf{At}} & \mathbf{At}^{\neg \mathbf{Bt}} \oplus \mathbf{Dt}_{bt_{2,2}}^{\neg \mathbf{At}} & \dots & \mathbf{Dt}_{bt_{2,n}}^{\neg \mathbf{At}} \\ \mathbf{Dt}_{bt_{3,1}}^{\neg \mathbf{At}} & \mathbf{Dt}_{bt_{3,2}}^{\neg \mathbf{At}} & \dots & \vdots \\ \vdots & \vdots & \dots & \mathbf{Dt}_{bt_{n-1,n}}^{\neg \mathbf{At}} \\ \mathbf{Dt}_{bt_{n,1}}^{\neg \mathbf{At}} & \mathbf{Dt}_{bt_{n,2}}^{\neg \mathbf{At}} & \dots & \mathbf{At}^{\neg \mathbf{Bt}} \oplus \mathbf{Dt}_{bt_{n,n}}^{\neg \mathbf{At}} \end{bmatrix}$$

em que \mathbf{Ct}^\neg tem dimensão $m \times n$, $\mathbf{At}^{\neg \mathbf{Bt}}$ é a matriz dos elementos de Σ_1 que não pertencem a Σ_2 e $\mathbf{Dt}_{bt_{i,j}}^{\neg \mathbf{At}}$ é uma matriz diagonal temporizada de dimensão $m \times m$ formada pelos elementos $bt_{i,j}$ que não são definidos em \mathbf{At} , na diagonal principal, e ϵ nos demais elementos.

A definição da matriz $\mathbf{Pt} = \mathbf{At} || \mathbf{Bt}$ é geralmente utilizada para construir a composição síncrona do supervisor com o gerador. A matriz \mathbf{Ct}^\neg , quando todos os elementos de Σ_1 são diferentes de Σ_2 , pode ser utilizada para a construção do gerador do sistema, a partir da composição dos subsistemas que o compõem.

Exemplo 34 *Dadas*

$$\mathbf{At} = \begin{bmatrix} \epsilon & 2\alpha \\ 3\beta & \epsilon \end{bmatrix}, \quad \theta(\mathbf{At}) = \begin{bmatrix} e & \epsilon \end{bmatrix}, \quad \phi(\mathbf{At}) = \begin{bmatrix} 3 \\ \epsilon \end{bmatrix}$$

e

$$\mathbf{Bt} = \begin{bmatrix} \epsilon & 2\alpha & \epsilon \\ 2\beta & \epsilon & 3\eta \\ 2\lambda & 4\beta & \epsilon \end{bmatrix}, \quad \theta(\mathbf{Bt}) = \begin{bmatrix} 2 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{Bt}) = \begin{bmatrix} 2 \\ 1 \\ \epsilon \end{bmatrix}$$

que representam os autômatos da Figura 0.30(a) e 0.30(b), seu produto síncrono determina a matriz \mathbf{Pt} , em que seus elementos $pt_{k,l} = pt_{(i_{\mathbf{At}}, i_{\mathbf{Bt}}), (j_{\mathbf{At}}, j_{\mathbf{Bt}})}$, $k = i_{\mathbf{At}} + m(i_{\mathbf{Bt}} - 1)$ e $l = j_{\mathbf{At}} + m(j_{\mathbf{Bt}} - 1)$, nos elementos $a_{i_{\mathbf{At}}, j_{\mathbf{At}}} = b_{i_{\mathbf{Bt}}, j_{\mathbf{Bt}}} \neq \epsilon$, são dados por:

- $k = 1 + 2 \times 0 = 1$, $l = 2 + 2 \times 1 = 4$, $pt_{1,4} = pt_{(1,1), (2,2)} = at_{1,2} \otimes bt_{1,2} = 2\alpha$
- $k = 2 + 2 \times 1 = 4$, $l = 1 + 2 \times 0 = 1$, $pt_{4,1} = pt_{(2,2), (1,1)} = at_{2,1} \otimes bt_{2,1} = 3\beta$
- $k = 2 + 2 \times 2 = 6$, $l = 1 + 2 \times 1 = 3$, $pt_{6,3} = pt_{(2,3), (1,2)} = at_{2,1} \otimes bt_{3,2} = 4\beta$

$$\mathbf{Pt} = \begin{bmatrix} \epsilon & \epsilon & \epsilon & 2\alpha & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ 3\beta & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 4\beta & \epsilon & \epsilon & \epsilon \end{bmatrix}$$

com

$$\theta(\mathbf{Pt}) = \begin{bmatrix} 2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{Pt}) = \begin{bmatrix} 3 \\ \epsilon \\ 3 \\ \epsilon \\ \epsilon \\ \epsilon \end{bmatrix} \quad (0.87)$$

Como $\lambda, \eta \in \mathbf{Bt}$, mas $\lambda, \eta \notin \mathbf{At}$, então

$$\mathbf{Ct}^\neg = \begin{bmatrix} \overbrace{\begin{matrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{matrix}}^{\mathbf{At}^\neg \mathbf{Bt} \oplus \mathbf{Dt}_{bt_{1,1}}^\neg \mathbf{At}} & \overbrace{\begin{matrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{matrix}}^{\mathbf{Dt}_{bt_{1,2}}^\neg \mathbf{At}} & \overbrace{\begin{matrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{matrix}}^{\mathbf{Dt}_{bt_{1,3}}^\neg \mathbf{At}} \\ \overbrace{\begin{matrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{matrix}}^{\mathbf{Dt}_{bt_{2,1}}^\neg \mathbf{At}} & \overbrace{\begin{matrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{matrix}}^{\mathbf{At}^\neg \mathbf{Bt} \oplus \mathbf{Dt}_{bt_{2,2}}^\neg \mathbf{At}} & \overbrace{\begin{matrix} 3\eta & \epsilon \\ \epsilon & 3\eta \end{matrix}}^{\mathbf{Dt}_{bt_{2,2}}^\neg \mathbf{At}} \\ \overbrace{\begin{matrix} 2\lambda & \epsilon \\ \epsilon & 2\lambda \end{matrix}}^{\mathbf{Dt}_{bt_{3,1}}^\neg \mathbf{At}} & \overbrace{\begin{matrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{matrix}}^{\mathbf{Dt}_{bt_{3,2}}^\neg \mathbf{At}} & \overbrace{\begin{matrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{matrix}}^{\mathbf{At}^\neg \mathbf{Bt} \oplus \mathbf{Dt}_{bt_{1,1}}^\neg \mathbf{At}} \end{bmatrix}$$

e

$$\mathbf{Pt} = \mathbf{Pt} \oplus \mathbf{Ct}^\top = \begin{bmatrix} \epsilon & \epsilon & \epsilon & 2\alpha & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 3\eta & \epsilon \\ 3\beta & \epsilon & \epsilon & \epsilon & \epsilon & 3\eta \\ 2\lambda & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 2\lambda & 4\beta & \epsilon & \epsilon & \epsilon \end{bmatrix}$$

com $\theta(\mathbf{Pt})$ e $\phi(\mathbf{Pt})$ dados por 0.87. Assim, \mathbf{Pt} em conjunto com $\theta(\mathbf{Pt})$ e $\phi(\mathbf{Pt})$ representam o autômato $A_{(\max,+)_3}$ composto por $A_{(\max,+)_1}$ e $A_{(\max,+)_2}$ através do produto síncrono visto na Figura 0.30(c). Deve-se observar que a linha 2 contém todos os seus elementos

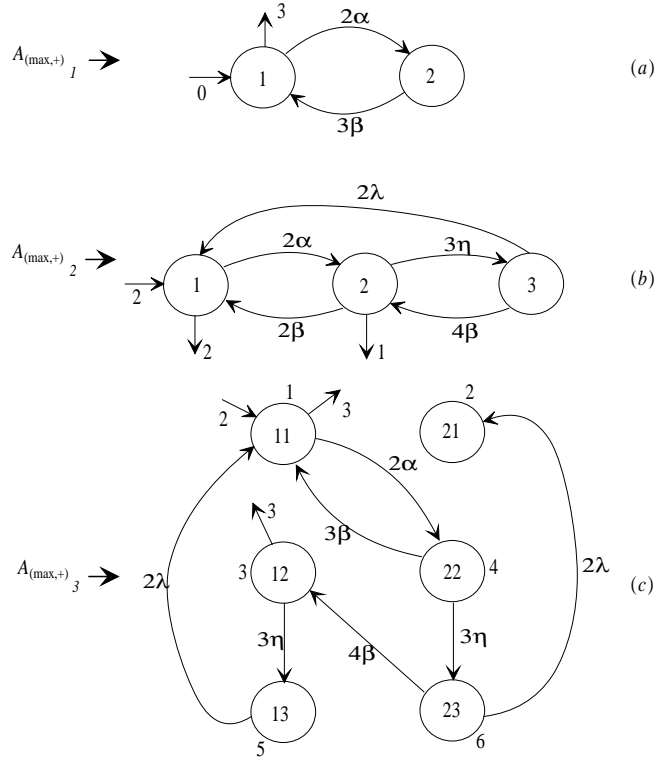


Figura 0.30: Autômatos (a) $A_{(\max,+)_1}$, (b) $A_{(\max,+)_2}$ e (c) $A_{(\max,+)_3}$ - composição síncrona de $A_{(\max,+)_1}$ e $A_{(\max,+)_2}$.

$pt_{2,j} = \epsilon$, mas esta linha é alcançada pela sequência $s = \theta(\mathbf{Pt}) \otimes \mathbf{Pt}^3 \otimes \pi = pt_{1,4}pt_{4,6}pt_{6,2} = 9\alpha\eta\lambda$, para

$$\pi = \begin{bmatrix} \epsilon & e & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}^T,$$

o que implica que esta linha representa um estado bloqueado na composição de $A_{(\max,+)_1}$ com $A_{(\max,+)_2}$.

Deve-se observar que a Definição 56 constrói a composição paralela de duas matrizes de incidência temporizadas \mathbf{At} e \mathbf{Bt} , quando $\Sigma_{\mathbf{At}} \cap \Sigma_{\mathbf{Bt}} = \emptyset$, através da construção da matriz \mathbf{Ct}^\top . Por outro lado, considerando todos os tempos de vida dos eventos $t_\sigma = e$, a Definição 56 constrói

a composição síncrona para o caso não temporizado. Também, deve-se observar que a Definição 56 utiliza o operador \otimes de acordo com a Definição 54. Contudo, quando tratando de síntese de supervisores de SEDs, deve-se considerar o operador \otimes de acordo com a Definição 55.

Com este formalismo, o problema de controle de SEDs temporizados é enunciado como a seguir: *Dados um autômato temporizado $A_{(\max,+)}$, uma linguagem-alvo temporizada reconhecível E e uma linguagem temporizada mínima admissível $L_A \subseteq E$, encontrar um supervisor próprio \mathcal{S} tal que*

$$L_A \subseteq L_c(\mathcal{S}/G) \subseteq E. \quad (0.88)$$

Para solucionar o problema, define-se:

Definição 57 *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{At}_{uc} por*

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

em que t_σ é o tempo de vida do evento que leva o autômato $A_{(\max,+)}$ do estado i para o estado j . Se mais de um evento é definido do estado i para o estado j , $(at_{uc})_{i,j} = \bigoplus_l t_{\sigma^l} \sigma^l$. Os vetores de estado inicial e de estados marcados são os mesmos da matriz \mathbf{At} .

Exemplo 35 *Considerando que o autômato $(\max,+)$ da Figura 0.31 tem $\Sigma_c = \{\alpha, \beta\}$ e $\Sigma_{uc} = \{\eta, \kappa\}$, então*

$$\mathbf{At}_{uc} = \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ 2\kappa & \epsilon & 3\eta \\ 2\kappa & \epsilon & \epsilon \end{bmatrix}.$$

Os vetores de estado inicial e estados marcados são

$$\theta(\mathbf{At}_{uc}) = \begin{bmatrix} 2 & \epsilon & \epsilon \end{bmatrix} \quad e \quad \phi(\mathbf{At}_{uc}) = \begin{bmatrix} 2 \\ 1 \\ \epsilon \end{bmatrix}.$$

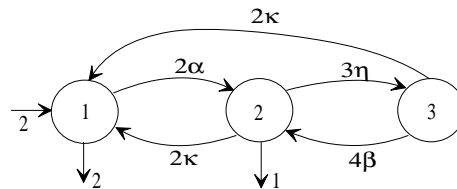


Figura 0.31: Autômatos $A_{(\max,+)}$, para exemplificar a construção da matriz \mathbf{At}_{uc} .

Com essa definição, o procedimento para a síntese do supervisor é semelhante ao procedimento apresentado para SEDs não temporizados. Assim, necessita-se definir a especificação de

comportamento e o supervisor para os SEDs temporizados. Neste caso, a especificação de comportamento é uma tarefa que se deseja que o sistema realize, a qual inclui tanto o comportamento lógico (similar ao caso não temporizado), quanto os tempos mínimos para os inícios das atividades (que devem sempre ser iguais ou maiores que os do sistema, definindo atrasos), determinando o comportamento lógico e temporizado que o sistema deve apresentar sob supervisão.

Definição 58 *Uma especificação de comportamento temporizada para um autômato temporizado $A_{(\max,+)}$ que é um modelo de um SED temporizado, denotada por \mathbf{Et} , é definida por*

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

em que t_σ é o tempo de vida do evento σ que leva o autômato da especificação do estado i para o estado j . Se mais de um evento é definido do estado i para o estado j , $et_{i,j} = \bigoplus_l t_{\sigma^l} \sigma^l$, para $\sigma^l \in \Sigma$, $l = 1, 2, \dots$, de modo que qualquer evento σ^l ocorrendo, provoca no autômato $A_{(\max,+)}$ a mudança do estado i para o estado j . O estado inicial é definido como sendo o estado 1 e é representado pelo vetor linha

$$\theta(\mathbf{Et}) = \begin{bmatrix} t_i & \epsilon & \dots & \epsilon \end{bmatrix}$$

com t_i representando o atraso inicial, e os estados marcados são representados pelo vetor coluna

$$\phi(\mathbf{Et}) = \begin{bmatrix} t_{m_1} & t_{m_2} & \dots & t_{m_n} \end{bmatrix}^T,$$

com t_{m_i} representando os atrasos finais. Se um estado k não é marcado, $t_{m_k} = \epsilon$.

O supervisor é definido como a seguir:

Definição 59 *Um supervisor construído através de uma especificação de comportamento \mathbf{Et} para um SED temporizado modelado por uma matriz de incidência temporizada \mathbf{At} é definido como sendo a matriz*

$$\mathbf{St} = [st_{i,j}], st_{i,j} = \begin{cases} t_\sigma \sigma \subset et_{i,j} & \text{se } t_\sigma \sigma \subset et_{i,j} \text{ pode ocorrer em } \mathbf{At}; \\ \epsilon & \text{caso contrário} \end{cases}$$

em que $st_{i,j} = \epsilon$ para $at_{i,j} \neq \epsilon$, implica em dizer que existe um controle para inibir o evento que se encontra em $at_{i,j}$, para $at_{i,j} = \sigma_c$, ou que o estado (linha) j não é acessível. Se mais de um evento é definido do estado i para o estado j , $st_{i,j} = \bigoplus_l t_{\sigma^l} \sigma^l$, para $\sigma^l \in \Sigma$, $l = 1, 2, \dots$, de modo que qualquer evento σ^l ocorrendo, provoca no autômato da composição síncrona $\mathbf{St}||\mathbf{At}$ a mudança do estado i para o estado j . O estado inicial é definido como sendo o estado 1 e é representado pelo vetor linha

$$\theta(\mathbf{St}) = \begin{bmatrix} t_i & \epsilon & \dots & \epsilon \end{bmatrix}$$

com t_i representando o atraso inicial, e os estados marcados são representados pelo vetor coluna

$$\phi(\mathbf{St}) = \begin{bmatrix} t_{m_1} & t_{m_2} & \dots & t_{m_n} \end{bmatrix}^T,$$

com t_{m_i} representando os atrasos finais. Se um estado k não é marcado, $t_{m_k} = \epsilon$.

Na abordagem não temporizada, o supervisor é o agente que recebe os eventos σ gerados pelo SED, e determina a ação de controle γ a ser aplicada de forma ao SED realizar a tarefa desejada, definida pela especificação de comportamento. Para o caso da abordagem temporizada, o supervisor é o agente que recebe os eventos σ gerados pelo SED e seus respectivos tempos de vida, e determina a ação de controle definida como uma inibição com tempo determinado nos eventos controláveis. Este formalismo é apresentado na Figura 0.32.

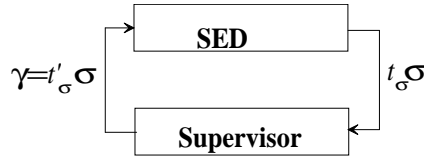


Figura 0.32: Diagrama de um SED temporizado supervisionado.

Com essas definições, a contextualização com a TCS é determinada como um problema de estados proibidos restrito ao controle de SEDs temporizados, onde os tempos de vida dos eventos são definidos como o tempo mínimo para suas ocorrências. Assim, o mesmo procedimento apresentado na abordagem não temporizada é utilizado, onde devem ser consideradas as seguintes denominações:

1. Um sub-autômato $A'_{(\max,+)}$ apresenta uma estrutura semelhante, mas apenas com parte dos estados ou com parte das transições de $A_{(\max,+)}$, em que o tempo de vida de cada transição de $A'_{(\max,+)}$ deve ser maior ou igual ao tempo de vida da respectiva transição em $A_{(\max,+)}$;
2. Uma sub-linguagem temporizada L' apresenta parte das seqüências da linguagem L , em que cada evento de L' apresenta um tempo de vida maior ou igual ao respectivo tempo de vida do mesmo evento em L . Não necessariamente $L' \subset L$ indica que o autômato de L' é um sub-autômato de L ;
3. Uma submatriz \mathbf{At}' é formada pelas primeiras m linhas e as primeiras m colunas de \mathbf{At} e mantém uma correlação elemento a elemento, apresentando para cada evento de \mathbf{At}' um tempo de vida maior ou igual ao respectivo evento de \mathbf{At} .

Dessa forma, para realizar a síntese de supervisores de SEDs temporizados utilizando as matrizes de incidência temporizadas, é necessário que a especificação de comportamento sempre seja uma submatriz da matriz de incidência \mathbf{At} . Se \mathbf{Et} não é uma submatriz de \mathbf{At} , torna-se necessário transformar \mathbf{At} em $\mathbf{At}^\#$, e construir uma nova matriz $\mathbf{Et}^\#$ que se apresente como uma submatriz de $\mathbf{At}^\#$, tal que $L(\mathbf{At}^\#) = L(\mathbf{At})$, $L_m(\mathbf{At}^\#) = L_m(\mathbf{At})$, e $L(\mathbf{Et}^\#) = L(\mathbf{Et})$, $L_m(\mathbf{Et}^\#) = L_m(\mathbf{Et})$. Assim, considerando que a especificação de comportamento \mathbf{Et} seja uma submatriz de \mathbf{At} , ou seja transformada em $\mathbf{Et}^\#$, sempre a relação $L(\mathbf{Et}) \subseteq L(\mathbf{At})$ é verificada. Esses dois casos representam uma especificação de comportamento genérica. Quando

uma especificação de comportamento temporizada \mathbf{Et} não é submatriz de \mathbf{At} , necessita-se do seguinte procedimento:

1. Constrói-se a composição síncrona de \mathbf{At} com uma matriz qualquer que gera Σ^* , para determinar uma nova matriz $\mathbf{At}^\#$ que gera $L(\mathbf{At}^\#) = L(\mathbf{At})$. Transformando a especificação de comportamento \mathbf{Et} em uma matriz de incidência \mathbf{Et}^* que gera Σ^* , através da inclusão de um estado proibido (linha i_e /coluna j_e), denominado de estado de *erro*, utiliza-se \mathbf{Et}^* para construir a matriz $\mathbf{At}^\# = \mathbf{At} || \mathbf{Et}^*$;
2. Da matriz $\mathbf{At}^\#$, substituindo todos os elementos das linhas/colunas que formam os pares $(i_{\mathbf{At}}, i_e)$, $(j_{\mathbf{At}}, j_e)$ pelo símbolo ϵ , encontra-se a matriz $\mathbf{Et}^\#$ tal que $L(\mathbf{Et}^\#) \subset L(\mathbf{At}^\#) = L(\mathbf{At})$ e $\mathbf{Et}^\#$ sendo uma submatriz de $\mathbf{At}^\#$;
3. Os tempos de vida dos eventos de \mathbf{Et} , são resgatados para $\mathbf{Et}^\#$ (que estão iguais aos tempos de vida dos eventos de \mathbf{At}) por $t_\sigma \oplus t'_\sigma$, onde t_σ é o tempo de vida do evento $\sigma \subset et_{i,j}$ e t'_σ é o tempo de vida do evento $et_{k,l}^\#$, com k sendo o par $(i_{\mathbf{At}}, i_{\mathbf{Et}})$ em que $i_{\mathbf{Et}} = i$ e l sendo o par $(j_{\mathbf{At}}, j_{\mathbf{Et}})$ em que $j_{\mathbf{Et}} = j$;
4. O vetor $\theta(\mathbf{Et}^\#)$ é construído através do resgate dos elementos do vetor de estado inicial $\theta(\mathbf{Et})$, como $\theta_1(\mathbf{Et}^\#) = \theta_1(\mathbf{Et}^\#) \oplus \theta_1(\mathbf{Et})$, tal que $L(\mathbf{Et}^\#) \subset L(\mathbf{At}^\#)$ e $L(\mathbf{Et}) = L(\mathbf{Et}^\#)$;
5. O vetor $\phi(\mathbf{Et}^\#)$ é determinado pelos elementos resgatados do vetor original $\phi(\mathbf{Et})$, como $\phi_k(\mathbf{Et}^\#) = \phi_k(\mathbf{Et}^\#) \oplus \phi_i(\mathbf{Et})$ em que k é definido como o par $(i_{\mathbf{At}}, i_{\mathbf{Et}})$ e $i = i_{\mathbf{Et}}$, com $\phi_i(\mathbf{Et}) \neq \epsilon$, tal que $L_m(\mathbf{Et}^\#) = L_m(\mathbf{Et})$.

Observação: Deve-se observar que no contexto das matrizes de incidência temporizadas, a matriz que gera Σ^* deve ser considerada com todos os tempos de vida dos eventos $t_\sigma = e$. Isto porque a composição síncrona $\mathbf{At} || \Sigma^*$ deve ser uma matriz $\mathbf{At}^\#$ que deve apresentar a mesma linguagem da matriz \mathbf{At} . Pela própria definição da composição síncrona, os tempos de vida da matriz resultante da composição é $t_\sigma \oplus t'_\sigma$, quando $\sigma \in \Sigma_e$. Assim, considerando $\sigma \subset at_{i,j}$, para $t'_\sigma = e$, a matriz resultante $\mathbf{At} || \Sigma^*$ tem sempre os tempos de vida satisfazendo $t_\sigma \oplus e = t_\sigma$, e dessa forma, garante-se que $L(\mathbf{At}^\#) = L(\mathbf{At})$.

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

Algoritmo 2 Transformação de matrizes de incidência temporizadas \mathbf{Ft} para reconhecimento de Σ^*

1. Tornar os elementos $\theta_1(\mathbf{Ft}) = e$, $\phi_i(\mathbf{Ft}) = e$, sendo o índice o elemento da coluna/linha do vetor respectivo, e $\forall ft_{i,j} = t_{\sigma^1}\sigma^1 + \dots + t_{\sigma^n}\sigma^n$, $t_{\sigma^k} = e$, $k = 1, \dots, n$;

2. Para cada linha i de \mathbf{Ft} , incluir os auto-laços

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

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

3. Incluir nos vetores $\theta(\mathbf{Ft})$ e $\phi(\mathbf{Ft})$, o elemento $\theta_{j+1}(\mathbf{Ft}) = \epsilon$ e $\phi_{j+1}(\mathbf{Ft}) = e$.

4. Incluir no final da matriz \mathbf{Ft} uma coluna j_e e uma linha marcada de ‘erro’ i_e representando uma linha proibida, e fazer:

i) Para $i < i_e$, faça:

a) $\forall j, j < j_e, ft_{i,j_e} = \sigma$, se $\sigma \in ft_{j,i} \vee \sigma \in ft_{j,j}$ e $ft_{i,k} \neq \sigma, \forall k, k < j_e;$

b) $\forall j, j < j_e, ft_{i,j_e} = \sigma'$ se $ft_{i,j} \neq \sigma'$

ii) Faça $ft_{i_e, j_e} = \Sigma$.

Exemplo 36 Dada a matriz de incidência temporizada sobre o alfabeto $\Sigma = \{\alpha, \beta, \eta, \lambda, \kappa, \mu\}$

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

a utilização do algoritmo 2 constrói a matriz

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

que reconhece Σ^* . Os autômatos que elas representam são vistos na Figura 0.33(a), 0.33(b), respectivamente.

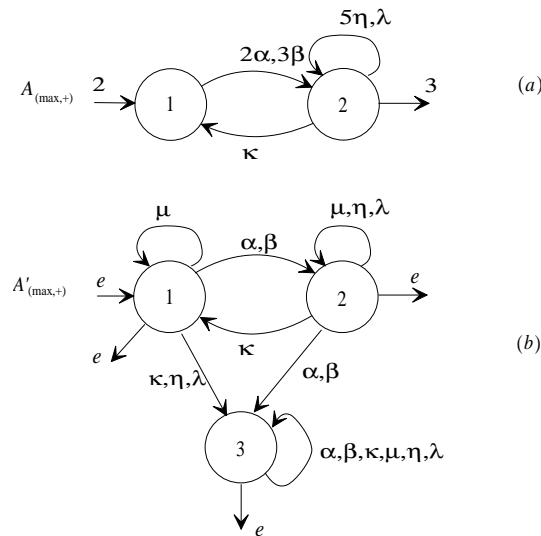


Figura 0.33: Autômatos (a) $A_{(\max,+)}$ e (b) $A'_{(\max,+)}$ construídos com o Algoritmo 2.

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

Algoritmo 3 Transformação da matriz de incidência \mathbf{Et} em $\mathbf{Et}^\#$

1. Transforme \mathbf{Et} em \mathbf{Et}^* ;
2. Construa $\mathbf{At}^\# = \mathbf{At} \parallel \mathbf{Et}^*$;
3. Para i e $j = 1$ até $n \times m$, faça:
 - a) Se $at_{i,j}^\# \neq \epsilon$, faça $et_{i,j}^\# = at_{i,j}^\#$.
4. Faça $k = 1$ até $n \times m$:
 - a) Se $\phi_k(\mathbf{At}^\#) \neq \epsilon$, faça $i_{\mathbf{Et}} = ((k - (k \bmod m)) / m) + 1$, e $\phi_k(\mathbf{Et}^\#) = \phi_k(\mathbf{Et}^\#) \oplus \phi_{i_{\mathbf{Et}}}(\mathbf{Et})$;
5. Faça $\theta_1(\mathbf{Et}^\#) = \theta_1(\mathbf{Et}^\#) \oplus \theta_1(\mathbf{Et})$.

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

Observação: Deve-se observar que o resgate dos tempos de vida dos eventos e dos atrasos inicial e finais eliminam a possibilidade de se introduzir uma especificação que defina uma redução no tempo de vida de um evento do sistema.

Exemplo 37 Considere a matriz

$$\mathbf{At} = \begin{bmatrix} \epsilon & 2\alpha_1 & 3\alpha_2 & \epsilon \\ 2\beta_1 & \epsilon & \epsilon & 3\alpha_2 \\ 4\beta_2 & \epsilon & \epsilon & 2\alpha_1 \\ \epsilon & 4\beta_2 & 2\beta_1 & \epsilon \end{bmatrix}, \quad \theta(\mathbf{At}) = \begin{bmatrix} 3 & \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{At}) = \begin{bmatrix} e \\ \epsilon \\ \epsilon \\ 3 \end{bmatrix}$$

e a especificação

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

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

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

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

$$\mathbf{At}^\# = \begin{bmatrix} \epsilon & 2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 3\alpha_2 & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 3\alpha_2 \\ 4\beta_2 & \epsilon & \epsilon & 2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 4\beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 3\alpha_2 & \epsilon & \epsilon & 2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 3\alpha_2 & \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & \epsilon & \epsilon & 2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 2\alpha_1 & 3\alpha_2 & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 & \epsilon & \epsilon & 3\alpha_2 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & \epsilon & \epsilon & 2\alpha_1 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & 2\beta_1 & \epsilon \end{bmatrix}$$

$$\theta(\mathbf{At}^\#) = \begin{bmatrix} 3 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix},$$

$$\phi(\mathbf{At}^\#) = \begin{bmatrix} e & \epsilon & \epsilon & 3 & e & \epsilon & \epsilon & 3 & e & \epsilon & \epsilon & 3 \end{bmatrix}^T$$

em que as linhas/colunas formadas por uma linha/coluna de \mathbf{At} com a linha/coluna de erro de \mathbf{Et}^* são as quatro últimas. Fazendo $et_{i,j}^\# = at_{i,j}^\#$, nos elementos $\forall i, j$, tal que $i, j \leq ((n-1) \times m)$ (igual a fazer $\mathbf{Et}^\# = \mathbf{At}^\#$ e tornar todos os elementos que satisfazem os pares $(i_{\mathbf{At}}, i_e)$, $(j_{\mathbf{At}}, j_e)$

iguais a ϵ) e resgatando os tempos de vida dos eventos de \mathbf{Et} , encontra-se

$$\mathbf{Et}^\# = \begin{bmatrix} \epsilon & 2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ 4\beta_2 & \epsilon & \epsilon & 2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 4\beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 4\alpha_2 & \epsilon & \epsilon & 2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 4\alpha_2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & \epsilon & \epsilon & 2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}$$

$$\theta(\mathbf{Et}^\#) = \begin{bmatrix} 4 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix},$$

$$\phi(\mathbf{Et}^\#) = \begin{bmatrix} 2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}^T$$

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

Observação: A partir daqui, sempre que se citar a especificação de comportamento \mathbf{Et} estará sendo considerada a sua transformação em $\mathbf{Et}^\#$, para os casos em que \mathbf{Et} não é uma submatriz de \mathbf{At} . Da mesma forma, \mathbf{At} implica em $\mathbf{At}^\#$.

As seguintes definições são também necessárias:

Definição 60 O operador ACES é definido como

$$ACES(\mathbf{At}) = \mathbf{Bt}, bt_{i,j} = \begin{cases} at_{i,j} & \text{se } i \text{ é acessível} \\ \epsilon & \text{caso contrário} \end{cases}$$

onde \mathbf{At} e \mathbf{Bt} são matrizes de incidência.

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

Definição 61 O operador COACES é definido como

$$COACES(\mathbf{At}) = \mathbf{Bt},$$

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

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

Com as Definições 60 e 61, também define-se o operador $TRIM$, como a seguir.

Definição 62 *O operador $TRIM$ é definido por*

$$TRIM(\mathbf{At}) = ACES(COACES(\mathbf{At})) = \mathbf{Bt}.$$

O operador $TRIM$ aplicado a uma matriz de incidência \mathbf{At} devolve uma matriz \mathbf{Bt} que é acessível e coacessível.

Definição 63 *Dadas duas matrizes temporizadas $\mathbf{At} = [at_{i,j}]$ e $\mathbf{Bt} = [bt_{i,j}]$, define-se o operador \leq por*

$$\mathbf{At} \leq \mathbf{Bt} \Leftrightarrow L(\mathbf{At}) \subseteq L(\mathbf{Bt}).$$

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

$$\mathbf{At} \geq \mathbf{Bt} \Leftrightarrow L(\mathbf{At}) \supseteq L(\mathbf{Bt}),$$

$$\mathbf{At} \triangleleft \mathbf{Bt} \Leftrightarrow L(\mathbf{At}) \subset L(\mathbf{Bt}),$$

$$\mathbf{At} \triangleright \mathbf{Bt} \Leftrightarrow L(\mathbf{At}) \supset L(\mathbf{Bt}).$$

Os operadores da Definição 63, são utilizados considerando um elemento genérico $\sigma = \sigma^1 + \dots + \sigma^n$. A condição de $\mathbf{At} \leq \mathbf{Bt}$, implica em que toda seqüência de \mathbf{At} , que é formada por eventos com seus respectivos tempos de vida (a soma desses tempos define o tempo da seqüência) deve ser executada em um tempo maior ou igual a execução da mesma seqüência em \mathbf{Bt} . Em termos de SEDs, isto implica em dizer que a linguagem que contém uma outra, é executada em um tempo menor [86, 9, 49, 51, 54, 41]. Então, $L(\mathbf{At}) \subseteq L(\mathbf{Bt})$ significa que todas as seqüências da linguagem de \mathbf{At} são encontradas na linguagem de \mathbf{Bt} , e estas seqüências satisfazem

$$(y|_{s_{\mathbf{At}}}) \geq (y|_{s_{\mathbf{Bt}}}),$$

onde cada evento tem um tempo de vida satisfazendo

$$t_{\sigma_i \mathbf{At}} \geq t_{\sigma_i \mathbf{Bt}}.$$

No caso destes operadores, deve-se observar que sua utilização garante a avaliação dos vetores θ e ϕ das matrizes. Isto garante o reconhecimento das mesmas seqüências em duas matrizes de incidência temporizadas, de acordo com as Definições 13, 18, 15 e 51.

Exemplo 38 *Considere o autômato mostrado na Figura 0.34(a), que tem sua representação matricial dada por*

$$\mathbf{At}_1 = \begin{bmatrix} 2\beta & 6\alpha & \epsilon \\ 4\beta & 2\alpha & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \theta(\mathbf{At}_1) = \begin{bmatrix} 4 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{At}_1) = \begin{bmatrix} 5 \\ \epsilon \\ \epsilon \end{bmatrix}.$$

Para o autômato mostrado na Figura 0.34(b), que tem sua representação matricial dada por

$$\mathbf{At}_2 = \begin{bmatrix} 2\beta & 3\alpha & \epsilon \\ 2\beta & 2\alpha & 5\kappa \\ \epsilon & 4\beta & \epsilon \end{bmatrix}, \quad \theta(\mathbf{At}_2) = \begin{bmatrix} 2 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{At}_2) = \begin{bmatrix} 5 \\ \epsilon \\ e \end{bmatrix},$$

onde verifica-se que $\mathbf{At}_1 \triangleleft \mathbf{At}_2$.

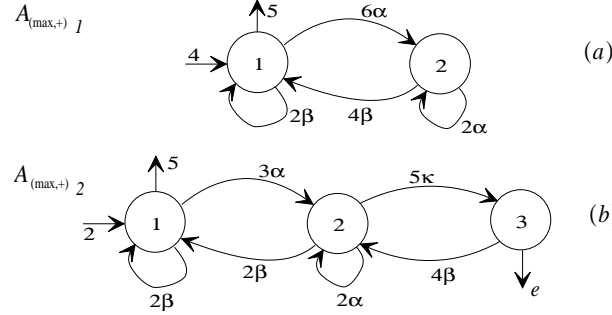


Figura 0.34: Autômatos $(\max, +)$ para ilustrar a utilização do operador \triangleleft .

Assim, tem-se que \mathbf{Et} deve satisfazer a seguinte condição:

Definição 64 Uma especificação de comportamento \mathbf{Et} é válida para a matriz de incidência \mathbf{At} se $\mathbf{Et} \neq [\epsilon]$ e

$$\begin{aligned} & \forall et_{i,j}, t_\sigma \geq t'_\sigma, \text{ com } t'_\sigma \text{ sendo o tempo de vida de } \sigma \subset at_{i,j} \\ & \theta_1(\mathbf{Et}) \geq \theta_1(\mathbf{At}) \quad e \\ & \phi_i(\mathbf{Et}) \geq \phi_i(\mathbf{At}) \vee \phi_i(\mathbf{Et}) = \epsilon, \forall \phi_i(\mathbf{At}) \neq \epsilon, \forall i = 1 \text{ até } N, \end{aligned}$$

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

Assim, tendo \mathbf{Et} válida, define-se sua condição de controlabilidade como a seguir:

Definição 65 Dada uma especificação de comportamento \mathbf{Et} válida, com $ACES(\mathbf{Et}) = \mathbf{Et}$, e a matriz de incidência temporizada \mathbf{At} que representa o autômato $A_{(\max, +)}$, a condição de controlabilidade para \mathbf{Et} é definida por

$$ACES(\mathbf{Et} \oplus \mathbf{At}_{uc}) = \mathbf{Et}. \quad (0.89)$$

Essa definição é uma extensão da Definição 46 para o caso temporizado. Deve-se ver que o operador \trianglelefteq avalia os vetores de estado inicial θ e de estados marcados ϕ .

Exemplo 39 Considerando as matrizes de incidência:

$$\begin{aligned} \mathbf{At} &= \begin{bmatrix} \epsilon & 3\alpha & \epsilon \\ 3\beta & \epsilon & 2\kappa \\ 3\lambda & \epsilon & \epsilon \end{bmatrix}, \quad \theta(\mathbf{At}) = \begin{bmatrix} 2 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{At}) = \begin{bmatrix} 2 \\ \epsilon \\ \epsilon \end{bmatrix} \quad e \\ \mathbf{Et} &= \begin{bmatrix} \epsilon & 3\alpha & \epsilon \\ 4\beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \theta(\mathbf{Et}) = \begin{bmatrix} 2 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{Et}) = \begin{bmatrix} 3 \\ \epsilon \\ \epsilon \end{bmatrix} \end{aligned}$$

e sendo $\Sigma_{uc} = \{\lambda\}$, então

$$ACES \left(\begin{pmatrix} \begin{bmatrix} \epsilon & 3\alpha & \epsilon \\ 4\beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix} \oplus \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ 3\lambda & \epsilon & \epsilon \end{bmatrix} \right) =$$

$$ACES \left(\begin{bmatrix} \epsilon & 3\alpha & \epsilon \\ 4\beta & \epsilon & \epsilon \\ 3\lambda & \epsilon & \epsilon \end{bmatrix} \right) = \begin{bmatrix} \epsilon & 3\alpha & \epsilon \\ 4\beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix} = \mathbf{Et},$$

e os vetores de estados $\theta(\mathbf{Et}) = \theta(\mathbf{At})$ e $\phi_1(\mathbf{Et}) > \phi_1(\mathbf{At})$ (ϕ_2 e ϕ_3 são iguais) garantem que \mathbf{Et} é uma especificação que gera uma linguagem controlável.

Considerando uma especificação de comportamento \mathbf{Et} , e a matriz de incidência temporizada \mathbf{At} que modela o SED, a partir da condição de controlabilidade (Definição 65), um supervisor pode ser sintetizado.

Lema 1 Um supervisor \mathbf{St} para uma matriz de incidência \mathbf{At} é definido pela especificação de comportamento \mathbf{Et} se e somente se

$$ACES(\mathbf{Et} \oplus \mathbf{At}_{uc}) = \mathbf{Et}.$$

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

Corolário 1 Dada uma especificação de comportamento \mathbf{Et} válida e uma matriz de incidência temporizada *trim* \mathbf{At} que é o modelo de um SED, $\mathbf{St} = \text{TRIM}(\mathbf{Et})$ se e somente se

$$ACES(\mathbf{Et} \oplus \mathbf{At}_{uc}) = \mathbf{Et}.$$

Exemplo 40 Para o autômato mostrado na Figura 0.35, onde $\Sigma = \{\alpha, \beta, \kappa, \mu\}$, $\Sigma_{uc} = \{\kappa\}$ e $\Sigma_c = \{\alpha, \beta, \mu\}$, sua representação matricial é dada por

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

Para a especificação

$$\mathbf{Et} = \begin{bmatrix} \epsilon & 4\beta & \epsilon \\ \epsilon & \epsilon & 5\mu \\ \epsilon & \epsilon & 5\kappa \end{bmatrix}, \quad \theta(\mathbf{Et}) = \begin{bmatrix} 3 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{Et}) = \begin{bmatrix} \epsilon \\ \epsilon \\ 2 \end{bmatrix}$$

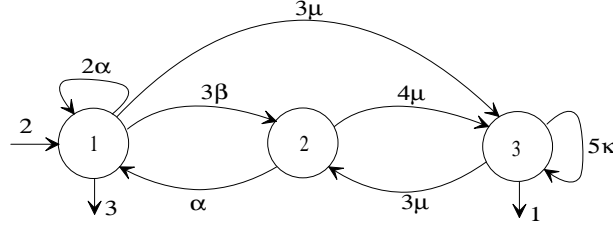


Figura 0.35: Autômato para exemplificar a síntese do supervisor através do Lema 1. e do Corolário 1.

mostrada na Figura 0.36(a), o supervisor é definido por $\mathbf{St} = \text{TRIM}(\mathbf{Et}) = \mathbf{Et}$, pois

$$\begin{aligned} & \text{ACES} \left(\left[\begin{array}{ccc} \epsilon & 4\beta & \epsilon \\ \epsilon & \epsilon & 5\mu \\ \epsilon & \epsilon & 5\kappa \end{array} \right] \oplus \left[\begin{array}{ccc} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 5\kappa \end{array} \right] \right) = \\ & \text{ACES} \left(\left[\begin{array}{ccc} \epsilon & 4\beta & \epsilon \\ \epsilon & \epsilon & 5\mu \\ \epsilon & \epsilon & 5\kappa \end{array} \right] \right) = \left[\begin{array}{ccc} \epsilon & 4\beta & \epsilon \\ \epsilon & \epsilon & 5\mu \\ \epsilon & \epsilon & 5\kappa \end{array} \right] = \mathbf{Et}, \end{aligned}$$

e os vetores de estados são $\theta_1(\mathbf{Et}) > \theta_1(\mathbf{At})$ (θ_2 e θ_3 são iguais) e $\phi_1(\mathbf{Et}) = \epsilon$, $\phi_2(\mathbf{Et}) = \phi_2(\mathbf{At})$ e $\phi_3(\mathbf{Et}) > \phi_3(\mathbf{At})$, que garante que \mathbf{Et} é uma especificação válida, e satisfaz o Lema 1. Entretanto, considerando a especificação de comportamento

$$\mathbf{Et} = \left[\begin{array}{ccc} 3\alpha & 3\beta & 4\mu \\ \epsilon & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{array} \right], \quad \theta(\mathbf{Et}) = \left[\begin{array}{ccc} 4 & \epsilon & \epsilon \end{array} \right], \quad \phi(\mathbf{Et}) = \left[\begin{array}{c} 5 \\ \epsilon \\ \epsilon \end{array} \right]$$

mostrada na Figura 0.36(b), tem-se que

$$\begin{aligned} & \text{ACES} \left(\left[\begin{array}{ccc} 3\alpha & 3\beta & 4\mu \\ \epsilon & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{array} \right] \oplus \left[\begin{array}{ccc} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 5\kappa \end{array} \right] \right) = \\ & \text{ACES} \left(\left[\begin{array}{ccc} 3\alpha & 3\beta & 4\mu \\ \epsilon & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{array} \right] \right) = \left[\begin{array}{ccc} 3\alpha & 3\beta & 4\mu \\ \epsilon & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{array} \right] = \mathbf{Et}, \end{aligned}$$

em que os vetores de estados são $\theta_1(\mathbf{Et}) > \theta_1(\mathbf{At})$ (θ_2 e θ_3 são iguais) e $\phi_1(\mathbf{Et}) = 5$, $\phi_2(\mathbf{Et}) > \phi_2(\mathbf{At})$ e $\phi_3(\mathbf{Et}) = \epsilon$, que garante que \mathbf{Et} é controlável. Contudo, \mathbf{Et} não é trim (veja que ao sair da linha 1, a matriz não mais retorna para ela, que é a única marcada - $\phi_1(\mathbf{Et}) = 5$). Assim, o supervisor é dado por

$$\mathbf{St} = \text{TRIM}(\mathbf{Et}) = \text{TRIM} \left(\left[\begin{array}{ccc} 3\alpha & 3\beta & 4\mu \\ \epsilon & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{array} \right] \right)$$

$$\mathbf{St} = \begin{bmatrix} 3\alpha & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \theta(\mathbf{St}) = \begin{bmatrix} 4 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{St}) = \begin{bmatrix} 5 \\ \epsilon \\ \epsilon \end{bmatrix}.$$

Por outro lado, para a especificação

$$\mathbf{Et} = \begin{bmatrix} \epsilon & 3\beta & 4\mu \\ 2\alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & \epsilon \end{bmatrix}, \quad \theta(\mathbf{Et}) = \begin{bmatrix} 3 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{Et}) = \begin{bmatrix} 3 \\ \epsilon \\ 2 \end{bmatrix}$$

mostrada na Figura 0.36(c), tem-se que

$$\begin{aligned} & ACES \left(\begin{bmatrix} \epsilon & 3\beta & 4\mu \\ 2\alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & \epsilon \end{bmatrix} \oplus \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 5\kappa \end{bmatrix} \right) = \\ & ACES \left(\begin{bmatrix} \epsilon & 3\beta & 4\mu \\ 2\alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{bmatrix} \right) = \begin{bmatrix} \epsilon & 3\beta & 4\mu \\ 2\alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{bmatrix} \triangleright \mathbf{Et}, \end{aligned}$$

o que não satisfaz o Lema 1.

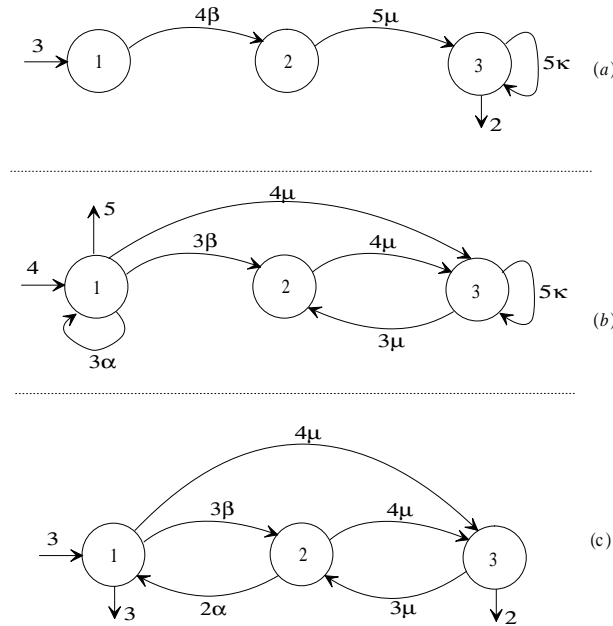


Figura 0.36: (a) e (b) Especificações controláveis e (c) Especificação não controlável.

Da mesma forma que para a síntese de supervisores para SEDs não temporizados, quando a especificação de comportamento não é factível por atingir estados onde eventos não controláveis podem ocorrer, é necessário encontrar um supervisor que realize o comportamento especificado restritivamente. Isto quer dizer que para uma especificação de comportamento válida, se

$$ACES(\mathbf{Et} \oplus \mathbf{At}_{uc}) \triangleright \mathbf{Et},$$

então, é necessário encontrar a suprema sublinguagem controlável. A solução desse problema pode ser encontrada utilizando a Definição 49 da matriz de caminhos temporizada.

Considerando que os eventos da matriz de incidência temporizada pertencem ao conjunto $\Sigma = \Sigma_c \cup \Sigma_{uc}$, a matriz \mathbf{At}^n contém palavras de comprimento n , compostas de eventos controláveis e não controláveis. Então, utilizando-se dessa definição, é preciso avaliar quais os caminhos que iniciam com um evento qualquer (controlável ou não controlável), e que seja seguido apenas por eventos não controláveis.

Definição 66 *Seja $\Sigma = \Sigma_c \cup \Sigma_{uc}$, e seja um autômato $A_{(\max,+)}$ construído com símbolos de Σ . A matriz de caminhos \mathbf{Ct}_{uc}^n , que define as palavras que mudam o estado do autômato de i para o estado j , iniciados por um evento qualquer (controlável ou não controlável), e seguidos sempre de eventos não controláveis, é definida por*

$$\mathbf{Ct}_{uc}^n = \mathbf{At} \otimes (\mathbf{At}_{uc})^{n-1}.$$

O vetor de estados marcados $\phi(\mathbf{Ct}_{uc}^n)$ e o vetor de estado inicial $\theta(\mathbf{Ct}_{uc}^n)$ são os mesmos de \mathbf{At} .

Igualmente à matriz de caminhos não temporizada, um determinado evento σ_{uc} de $(at_{uc})_{i,j}$, aparece como último evento nas seqüências dos termos de $(ct_{uc}^n)_{k,j}$. Estes termos de \mathbf{Ct}_{uc}^n são antecidos pelos elementos de $(ct_{uc}^{n-1})_{k,i}$.

Exemplo 41 *O autômato visto na Figura 0.37, com $\Sigma_c = \{\alpha, \kappa\}$ e $\Sigma_{uc} = \{\beta, \eta\}$, tem*

$$\mathbf{At} = \begin{bmatrix} \epsilon & 2\beta & \epsilon & \epsilon & \epsilon \\ 5\eta & \epsilon & 3\alpha & \epsilon & \epsilon \\ 2\alpha + 3\kappa & \epsilon & \epsilon & 3\eta & \epsilon \\ 1\eta & 2\beta & \epsilon & e\alpha & 3\kappa \\ \epsilon & \epsilon & 2\eta & \epsilon & \epsilon \end{bmatrix}$$

$$\theta(\mathbf{At}) = \begin{bmatrix} 2 & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{At}) = \begin{bmatrix} 2 \\ \epsilon \\ 4 \\ 3 \\ \epsilon \end{bmatrix}$$

e

$$\mathbf{At}_{uc} = \begin{bmatrix} \epsilon & 2\beta & \epsilon & \epsilon & \epsilon \\ 5\eta & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 3\eta & \epsilon \\ 1\eta & 2\beta & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 2\eta & \epsilon & \epsilon \end{bmatrix}$$

com $\theta(\mathbf{At}_{uc}) = \theta(\mathbf{At})$ e $\phi(\mathbf{At}_{uc}) = \phi(\mathbf{At})$. Calculando \mathbf{Ct}_{uc}^2 , tem-se

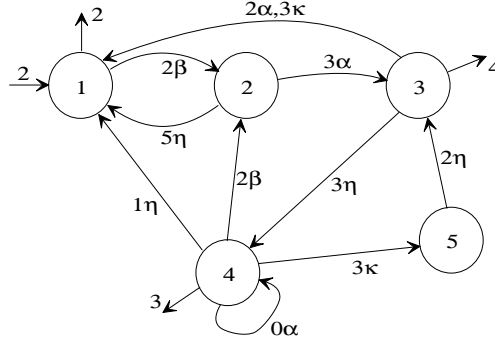


Figura 0.37: Autômato para construção da matriz de caminhos temporizada \mathbf{Ct}_{uc}^n .

$$\mathbf{Ct}_{uc}^2 = \begin{bmatrix} 7\beta\eta & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 7\eta\beta & \epsilon & 6\alpha\underline{\eta} & \epsilon \\ 4\eta\eta & 4\alpha\beta + 5\kappa\beta + 5\eta\beta & \epsilon & \epsilon & \epsilon \\ 7\beta\eta + 1\alpha\eta & 3\eta\beta + 2\alpha\beta & 5\kappa\eta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 5\eta\underline{\eta} & \epsilon \end{bmatrix}$$

com $\theta(\mathbf{Ct}_{uc}^2) = \theta(\mathbf{At})$ e $\phi(\mathbf{Ct}_{uc}^2) = \phi(\mathbf{At})$. Nela, vê-se que o elemento $(at_{uc})_{3,4} = \eta$ aparece em $(ct_{uc}^2)_{k,4}$, para $k = 2$ e $k = 5$. Isto é, nas linhas $k = 2$ e $k = 5$, e na coluna 3 de \mathbf{At} , encontram-se os elementos que antecedem $(at_{uc})_{3,4} = \eta$ em $(ct_{uc}^2)_{k,4}$. Logo, os termos $(at_{uc}^2)_{2,4}$ e $(at_{uc}^2)_{5,4}$ contêm o elemento $(at_{uc})_{3,4} = \eta$ como sendo o último evento de suas seqüências. Calculando \mathbf{Ct}_{uc}^3 , tem-se

$$\mathbf{Ct}_{uc}^3 = \begin{bmatrix} \epsilon & 9\beta\eta\beta & \epsilon & \epsilon & \epsilon \\ 12\eta\beta\eta + 7\alpha\eta\eta & 8\alpha\eta\beta & \epsilon & \epsilon & \epsilon \\ 9\alpha\beta\eta + 10\kappa\beta\eta + 10\eta\beta\eta & 6\eta\eta\beta & \epsilon & \epsilon & \epsilon \\ 8\eta\beta\eta + 7\alpha\beta\eta & 9\beta\eta\beta + 3\alpha\eta\beta & \epsilon & 8\kappa\eta\underline{\eta} & \epsilon \\ 6\eta\eta\eta & 7\eta\eta\beta & \epsilon & \epsilon & \epsilon \end{bmatrix}$$

em que $\theta(\mathbf{Ct}_{uc}^3) = \theta(\mathbf{At})$ e $\phi(\mathbf{Ct}_{uc}^3) = \phi(\mathbf{At})$. Nessa matriz, o termo $(at_{uc}^3)_{4,4}$ apresenta o termo $(ct_{uc}^2)_{k,3}$, $k = 4$, antecedendo o termo $(at_{uc})_{3,4} = \eta$.

Com isto, procede-se à construção do supervisor quando o Lema 1 não é satisfeito. A seguinte definição é de importância fundamental para solucionar o problema:

Definição 67 Dada a matriz de incidência \mathbf{Et} , que é uma matriz de caminhos de comprimento 1, define-se a matriz de caminhos temporizada

$$\mathbf{Bt}_{uc}^n = \mathbf{Et} \otimes (\mathbf{At}_{uc})^{n-1}$$

como sendo uma matriz de caminhos, em que o primeiro elemento de cada seqüência é um elemento de \mathbf{Et} , e os demais são elementos pertencentes a \mathbf{At}_{uc} . O vetor de estados marcados ϕ é definido por

$$\phi_i(\mathbf{Bt}_{uc}^n) = \begin{cases} \phi_i(\mathbf{Et}) \oplus \phi_i(\mathbf{At}), & \text{se } \phi_i(\mathbf{At}) \neq \epsilon \\ \epsilon & \text{caso contrário,} \end{cases} \quad (0.90)$$

e o vetor de estado inicial é $\theta(\mathbf{Bt}_{uc}^n) = \theta(\mathbf{At}) = \theta(\mathbf{Et})$.

Observação: A utilização da notação de uma matriz \mathbf{Bt} genérica citada anteriormente para algumas definições de operadores não deve ser confundida com a matriz \mathbf{Bt}_{uc}^n , que é uma definição específica para denotar a matriz de caminhos em que os primeiros elementos de cada caminho pertencem à \mathbf{Et} , seguidos por elementos de \mathbf{At}_{uc} .

Na matriz \mathbf{Bt}_{uc}^n pode-se avaliar se existe algum evento não controlável do autômato, em que sua ocorrência seja possível a partir de uma seqüência da especificação. Se existir, este evento aparece como último evento em uma determinada seqüência em \mathbf{Bt}_{uc}^n . A avaliação dos termos de \mathbf{Bt}_{uc}^n determina uma condição de teste semelhante a

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

contudo, utilizada para o caso temporizado específico aqui apresentado. Também, deve-se observar que, para $n = 1$, tem-se

$$\mathbf{Bt}_{uc}^1 = \mathbf{Et} \otimes (\mathbf{At}_{uc})^0 = \mathbf{Et} \otimes \mathbf{I} = \mathbf{Et},$$

onde \mathbf{I} é a matriz identidade formada por e na diagonal principal e ϵ nos demais elementos.

Igualmente à matriz de caminhos \mathbf{Ct}_{uc}^n , para uma matriz de caminhos \mathbf{Bt}_{uc}^n , um evento $\sigma_{uc} \subset (at_{uc})_{i,j}$ que não pertence à \mathbf{Et} , pode aparecer finalizando uma seqüência nos termos de $(bt_{uc}^n)_{k,j}$. Estes termos são antecidos pelos termos de $(bt_{uc}^{n-1})_{k,i}$.

Teorema 4 *Dada uma especificação de comportamento \mathbf{Et} válida e a matriz de incidência temporizada dos eventos não controláveis \mathbf{At}_{uc} , do autômato $A_{(\max,+)}$ trim, se*

$$ACES(\mathbf{Et} \oplus \mathbf{At}_{uc}) \triangleright \mathbf{Et},$$

então a $\sup C(L)$ será determinada recursivamente por:

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

$$\begin{aligned} \mathbf{Bt}_{uc}^n &= \mathbf{Et} \otimes (\mathbf{At}_{uc})^{n-1} \\ \mathbf{St}^n &= [st_{ij}^n], \quad st_{ij}^n = \begin{cases} st_{i,j}^{n-1} & \text{se } \sigma^1 \sigma_{uc}^2 \dots \sigma_{uc}^n \in \mathbf{Bt}_{uc}^n \wedge \sigma_{uc}^n \in \mathbf{Et}; \\ \epsilon & \text{se } \sigma_{uc}^n \notin \mathbf{Et} \wedge \sigma^1 \in \Sigma_c \end{cases} \\ \mathbf{St}^n &= TRIM(\mathbf{St}^n) \end{aligned}$$

onde σ_{uc}^n é o n -ésimo evento da seqüência de um termo de \mathbf{Bt}_{uc}^n , que pode ser $\sigma_{uc} \notin \mathbf{Et}$.

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

Vê-se nesse teorema que se o último evento de uma seqüência em \mathbf{Bt}_{uc}^n não pertence à especificação, a inibição do primeiro evento elimina esta seqüência. A existência de outros eventos não pertencentes à especificação, é eliminada recursivamente em \mathbf{Bt}_{uc}^n , para $n = 2, 3, \dots, N$, construindo o supervisor. Se algum evento não controlável que não pertence à especificação \mathbf{Et} não for eliminado de acordo com o Teorema 4, não é possível sintetizar um supervisor para a especificação desejada. De forma similar ao caso não temporizado, se $\forall n, \mathbf{St}^n$ não for coacessível, $\mathbf{S} = [\epsilon]$.

Exemplo 42 Considerando que o autômato mostrado na Figura 0.35 tem

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

em que $\Sigma = \{\alpha, \beta, \mu, \kappa\}$ e $\Sigma_{uc} = \{\kappa\}$, é dada a especificação

$$\mathbf{Et} = \begin{bmatrix} \epsilon & 4\beta & 3\mu \\ \alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & \epsilon \end{bmatrix}, \quad \theta(\mathbf{Et}) = \begin{bmatrix} 4 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{Et}) = \begin{bmatrix} 5 \\ \epsilon \\ \epsilon \end{bmatrix}$$

a qual está apresentada na Figura 0.38. Calculando $ACES(\mathbf{Et} \oplus \mathbf{At}_{uc})$, encontra-se

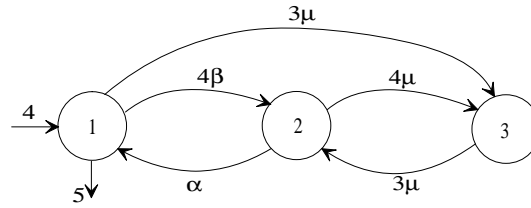


Figura 0.38: Especificação não controlável para síntese do supervisor.

$$ACES \left(\left(\begin{bmatrix} \epsilon & 4\beta & 3\mu \\ \alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & \epsilon \end{bmatrix} \oplus \begin{bmatrix} \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 5\kappa \end{bmatrix} \right) \right) =$$

$$ACES \left(\begin{bmatrix} \epsilon & 4\beta & 3\mu \\ \alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{bmatrix} \right) = \begin{bmatrix} \epsilon & 4\beta & 3\mu \\ \alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & 5\kappa \end{bmatrix} \triangleright \mathbf{Et}$$

Então,

$$\mathbf{St}^1 = \mathbf{Et} = \begin{bmatrix} \epsilon & 4\beta & 3\mu \\ \alpha & \epsilon & 4\mu \\ \epsilon & 3\mu & \epsilon \end{bmatrix}.$$

Calculando \mathbf{Bt}_{uc}^2 , tem-se

$$\mathbf{Bt}_{uc}^2 = \mathbf{Et} \otimes \mathbf{At}_{uc} = \begin{bmatrix} \epsilon & \epsilon & 8\mu\kappa \\ \epsilon & \epsilon & 9\mu\kappa \\ \epsilon & \epsilon & \epsilon \end{bmatrix}$$

que, utilizando o Teorema 4, encontram-se os termos $(bt_{uc}^2)_{1,3} = 8\mu\kappa$ e $(bt_{uc}^2)_{2,3} = 9\mu\kappa$, que contêm o evento controlável μ seguido do evento não controlável κ , que não pertence à especificação. Como μ é controlável, então fazendo $st_{1,3}^2 = st_{2,3}^2 = \epsilon$, elimina-se o estado 3. Logo, tornando-se o estado 3 não acessível, $st_{3,2}^2 = st_{3,3}^2 = \epsilon$. Assim, o supervisor é determinado por

$$\mathbf{St} = TRIM(\mathbf{St}^2) = \begin{bmatrix} \epsilon & 4\beta & \epsilon \\ \alpha & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \theta(\mathbf{St}) = \begin{bmatrix} 4 & \epsilon & \epsilon \end{bmatrix}, \quad \phi(\mathbf{St}) = \begin{bmatrix} 5 \\ \epsilon \\ \epsilon \end{bmatrix}$$

o qual está mostrado na Figura 0.39. Este supervisor determina que o SED reconheça a linguagem temporizada

$$\begin{aligned} L_m(\mathbf{St}||\mathbf{At}) &= (y|(\beta\alpha)^0)(\beta\alpha)^0 \oplus (y|(\beta\alpha)^1)(\beta\alpha)^1 \oplus \dots \\ L_m(\mathbf{St}||\mathbf{At}) &= 9 + 14\beta\alpha + 19\beta\alpha + \dots \end{aligned}$$

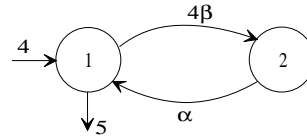


Figura 0.39: Supervisor para a especificação não controlável.

Observe no Exemplo 42 que, se $\phi_1(\mathbf{Et}) = \epsilon$, então $\mathbf{S} = [\epsilon]$.

O algoritmo para a síntese do supervisor de SEDs temporizados utilizando a álgebra de dióides é apresentado a seguir.

Algoritmo 4 Construção da $\sup C(L)$

1. Se $\mathbf{Et} \not\leq \mathbf{At}$, Construa $\mathbf{At}^\#$ e $\mathbf{Et}^\#$ a partir de \mathbf{At} e \mathbf{Et} , e faça $\mathbf{Et} = \mathbf{Et}^\#$ e $\mathbf{At} = \mathbf{At}^\#$.
2. Faça $\mathbf{Dt} = ACES(\mathbf{Et} \oplus \mathbf{At}_{uc})$.
3. Se $\mathbf{Dt} = \mathbf{Et}$, faça $\mathbf{St} = TRIM(\mathbf{Et})$ e pare.
4. Se $\mathbf{Dt} \triangleright \mathbf{Et}$, faça $n = 1$:
 - a) $\mathbf{St}^n = \mathbf{Et}$, $xdif(k, n-1) = i$ e $ydif(k, n-1) = j$ (onde $\mathbf{At}_{uc} \neq \mathbf{Et}_{uc}$), para $k = 1, \dots, M$ (M número de elementos diferentes entre \mathbf{At}_{uc} e \mathbf{Et}_{uc}).
 - b) Calcule \mathbf{Bt}_{uc}^n .
 - c) Faça para $k = 1$ até M
 - i. Procure os elementos em $\mathbf{Bt}_{uc}^n(i, ydif(k, n-1))$, onde $\sigma_{uc}^n \notin \mathbf{Et}$ (σ_{uc}^n sendo o último elemento da seqüência).

- (1) Se $\sigma_{uc}^n \notin \mathbf{Et}$ e $\sigma^1 \in \Sigma_c$, faça $\mathbf{St}^n(i, xdif(k, n-1)) = \epsilon$ e calcule $COACES(\mathbf{St}^n)$ e $ACES(\mathbf{St}^n)$;
- (2) Se $\sigma_{uc}^n \notin \mathbf{Et}$ e $\sigma^1 \in \Sigma_{uc}$, faça $xdif(k, n) = i$ e $ydif(k, n) = ydif(k, n-1)$.
- d) Se $\forall k = 1$ até M , $\mathbf{St}^n(xdif(k, n-1), ydif(k, n-1)) = \epsilon$ pare. Caso contrário, faça $n = n + 1$ e retorne ao passo 4.b.
- i) Se $n > N$ pare (\mathbf{Et} não é factível).

Nesse algoritmo, as matrizes $xdif$ e $ydif$ guardam os valores de i e j de \mathbf{Et} , respectivamente, para os eventos σ_{uc} acessíveis, mas que não devem ocorrer em \mathbf{Et} . Sempre que uma seqüência em \mathbf{Bt}_{uc}^n tem o último evento σ_{uc} acessível, porém não pertencente a \mathbf{Et} e o primeiro evento não controlável, estas matrizes são atualizadas com os valores de i e j de \mathbf{Bt}_{uc}^{n-1} . Enquanto $n \leq N$ (dimensão de \mathbf{At}), o algoritmo é repetido, buscando a $\sup C(L)$. Neste algoritmo, considerando que todos os tempos de vida dos eventos são $t_\sigma = e$, resolve-se o problema de Ramadge e Wonham [28], isto é, o caso não temporizado.

A seguir são apresentados dois exemplos da síntese do supervisor: um para o caso temporizado e outro para o caso não temporizado.

Exemplo 43 Considere o Exemplo 37, que é uma extensão do exemplo clássico do sistema das duas máquinas e um buffer de Ramadge e Wonham (a máquina 1 é representada pelo autômato G_1 e a máquina 2 é representada pelo autômato G_2 , apresentados na Figura 0.40, e a composição síncrona é apresentada na Figura 0.41) Sendo $\Sigma = \{\alpha_1, \alpha_2, \beta_1, \beta_2\}$ e $\Sigma_{uc} = \{\beta_1, \beta_2\}$, para a

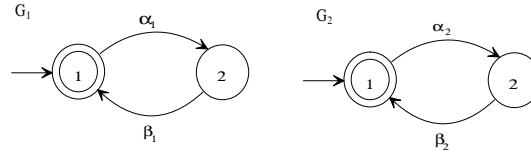


Figura 0.40: Autômatos G_1 e G_2 .

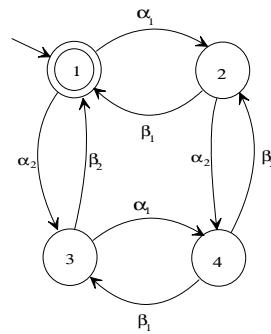


Figura 0.41: Autômato G_3 : composição síncrona de G_1 e G_2 .

especificação expandida $\mathbf{Et}^\#$ em forma de submatriz de $\mathbf{At}^\#$ (vista na figura 0.42), encontra-se

$$ACES(\mathbf{Et}^\# \oplus \mathbf{At}^\#) \triangleright \mathbf{Et}^\#,$$

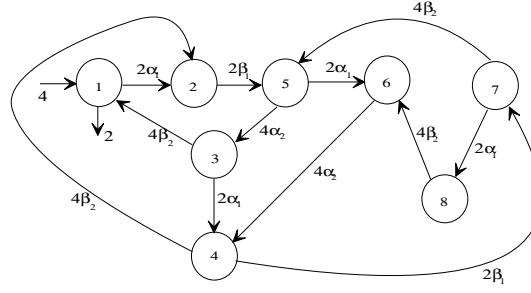


Figura 0.42: Especificação temporizada para o exemplo das duas máquinas e um buffer.

em que os elementos $et_{6,9}^\#$ e $et_{8,11}^\#$ não pertencem à $\mathbf{Et}^\#$, e não devem ocorrer. Assim, faz-se $\mathbf{St}^1 = \mathbf{Et}^\#$, $\theta(\mathbf{St}^1) = \theta(\mathbf{Et}^\#)$ e $\phi(\mathbf{St}^1) = \phi(\mathbf{Et}^\#)$, e calculando $\mathbf{Bt}_{uc}^2 = \mathbf{Et}^\# \otimes \mathbf{At}_{uc}^\#$, encontram-se os elementos $(bt_{uc}^2)_{5,9} = 4\alpha_1\beta_1$, $(bt_{uc}^2)_{7,11} = 4\alpha_1\beta_1$ e $(bt_{uc}^2)_{8,9} = 6\beta_2\beta_1$, que são os termos resultantes das multiplicações $et_{5,6}^\# \otimes (at_{uc}^\#)_{6,9}$, $et_{7,8}^\# \otimes (at_{uc}^\#)_{8,11}$ e $et_{8,6}^\# \otimes (at_{uc}^\#)_{6,9}$, respectivamente. Assim, a inibição dos eventos que antecedem β_1 em $st_{5,6}^1$ e $st_{7,8}^1$ tornam a linha 8 não acessível, não necessitando avaliar o termo $(bt_{uc}^2)_{8,9}$. Isso define a matriz \mathbf{St}^2 , dada por

$$\mathbf{St}^2 = \begin{bmatrix} \epsilon & 2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 & \epsilon & \epsilon \\ 4\beta_2 & \epsilon & \epsilon & 2\alpha_1 & \epsilon & \epsilon & \epsilon \\ \epsilon & 4\beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 \\ \epsilon & \epsilon & 4\alpha_2 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 4\alpha_2 & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & \epsilon \end{bmatrix}$$

$$\theta(\mathbf{St}^2) = \begin{bmatrix} 4 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix},$$

$$\phi(\mathbf{St}^2) = \begin{bmatrix} 2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}^T$$

Fazendo $\mathbf{St}^2 = \text{COACES}(\mathbf{St}^2)$, para eliminar os estados não coacessíveis e $\mathbf{St}^2 = \text{ACES}(\mathbf{St}^2)$, para eliminar os estados não acessíveis, encontra-se

$$\mathbf{St}^2 = \begin{bmatrix} \epsilon & 2\alpha_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 & \epsilon & \epsilon \\ 4\beta_2 & \epsilon & \epsilon & 2\alpha_1 & \epsilon & \epsilon & \epsilon \\ \epsilon & 4\beta_2 & \epsilon & \epsilon & \epsilon & \epsilon & 2\beta_1 \\ \epsilon & \epsilon & 4\alpha_2 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 4\alpha_2 & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 4\beta_2 & \epsilon & \epsilon \end{bmatrix}$$

$$\theta(\mathbf{St}^2) = \begin{bmatrix} 4 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix},$$

$$\phi(\mathbf{St}^2) = \begin{bmatrix} 2 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}^T$$

que é o supervisor apresentado na figura 0.43. Este supervisor garante que os eventos sejam atrasados para só ocorrerem após os tempos determinados, definindo que a linguagem do sistema

supervisionado seja sua própria linguagem temporizada, isto é

$$L(\mathbf{St}||\mathbf{At}^\#) = L(\mathbf{St}).$$

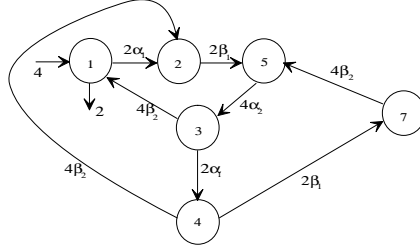


Figura 0.43: Supervisor.

Exemplo 44 No autômato visto na Figura 0.44, com $\Sigma = \{\alpha, \beta, \kappa, \eta, \lambda, \mu\}$ e $\Sigma_{uc} = \{\alpha, \lambda\}$, as matrizes de incidência \mathbf{At} e \mathbf{At}_{uc} são dadas por

$$\mathbf{At} = \begin{bmatrix} \epsilon & \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \lambda & \beta & \kappa & \mu & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \kappa + \mu & \epsilon & \epsilon & \eta \\ \epsilon & \epsilon & \epsilon & \alpha & \beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \mu & \lambda & \epsilon & \alpha & \eta \\ \alpha & \mu & \epsilon & \eta & \kappa & \beta + \lambda & \epsilon \\ \epsilon & \mu & \eta + \lambda & \epsilon & \epsilon & \epsilon & \alpha + \beta \end{bmatrix}$$

$$\theta(\mathbf{At}) = \begin{bmatrix} e & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix},$$

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

e

$$\mathbf{At}_{uc} = \begin{bmatrix} \epsilon & \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \lambda & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \alpha & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \lambda & \epsilon & \alpha & \epsilon \\ \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \lambda & \epsilon \\ \epsilon & \epsilon & \lambda & \epsilon & \epsilon & \epsilon & \alpha \end{bmatrix}$$

$$\theta(\mathbf{At}_{uc}) = \theta(\mathbf{At})$$

$$\phi(\mathbf{At}_{uc}) = \phi(\mathbf{At})$$

onde deve-se entender que cada estado marcado define um arco de saída com o valor $\phi_{q_m} = 0 = e$, o estado inicial é um arco de entrada com o valor $\theta_q = 0 = e$ e todos os arcos se apresentam com

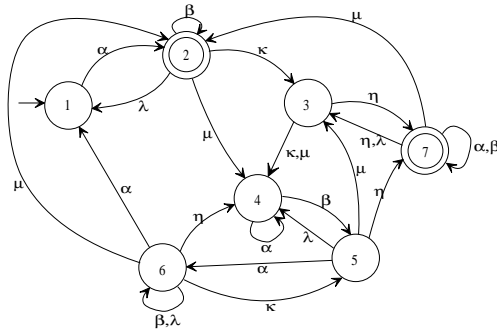


Figura 0.44: Autômato para o Exemplo 44.

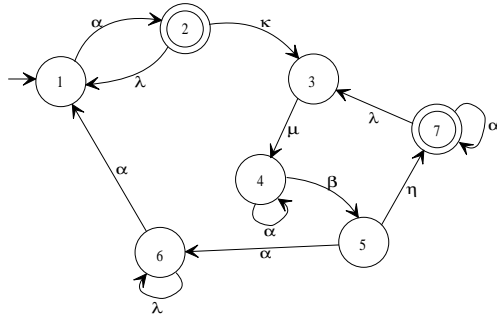


Figura 0.45: Especificação para o autômato do Exemplo 44.

$T_{q,\sigma,q^{prime}} = 0 = e$. Assim, definindo a especificação de comportamento pela matriz de incidência

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

$$\theta(\mathbf{Et}) = \begin{bmatrix} e & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix},$$

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

a qual está mostrada graficamente na figura 0.45, Vê-se que \mathbf{E} é válida, e que $\mathbf{E} \triangleleft \mathbf{A}$, mas a condição de controlabilidade falha, desde que

$$ACES(\mathbf{E} \oplus \mathbf{A}_{uc}) \triangleright \mathbf{E}.$$

O termo que não pertence à \mathbf{E} é $(a_{uc})_{5,4} = \lambda$. De acordo com o Teorema 4, faz-se $\mathbf{S}^1 = \mathbf{E}$ e calcula-se $\mathbf{B}_{uc}^2 = \mathbf{E} \otimes \mathbf{A}_{uc}$ para avaliar quais os eventos que devem ser desabilitados para

encontrar uma linguagem restrita para o supervisor.

$$\mathbf{Bt}_{uc}^2 = \begin{bmatrix} \alpha\lambda & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \lambda\alpha & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \mu\alpha & \epsilon & \epsilon & \epsilon \\ \epsilon & \eta\alpha & \epsilon & \alpha\alpha + \underline{\beta\lambda} & \epsilon & \beta\alpha & \epsilon \\ \alpha\alpha & \epsilon & \eta\lambda & \epsilon & \epsilon & \alpha\lambda & \eta\alpha \\ \lambda\alpha & \alpha\alpha & \epsilon & \epsilon & \epsilon & \lambda\lambda & \epsilon \\ \epsilon & \epsilon & \alpha\lambda & \epsilon & \epsilon & \epsilon & \alpha\alpha \end{bmatrix}$$

com $\theta(\mathbf{Bt}_{uc}^2) = \theta(\mathbf{At}) = \theta(\mathbf{Et})$ e $\phi(\mathbf{Bt}_{uc}^2) = \phi(\mathbf{At}) = \phi(\mathbf{Et})$. Essa matriz de caminhos define as palavras de comprimento 2, que terminam com um evento não controlável e que levam o autômato do estado i para o estado j iniciando com eventos da especificação. Como o interesse é inibir o evento que antecede o evento que torna a condição de controlabilidade falsa, vê-se que o termo $(bt_{uc}^2)_{4,4}$ contém a seqüência $\beta\lambda$, que leva o autômato através do evento controlável β do estado 4 para o estado 5, e depois retorna para o estado 4 pelo evento não controlável λ , que não faz parte da especificação \mathbf{Et} . Tornando o elemento $st_{4,5}^1 = \epsilon$ determina-se \mathbf{St}^2 . Calculando $\text{COACES}(\mathbf{St}^2)$, encontra-se a matriz

$$\mathbf{St}^2 = \text{COACES} \left(\begin{pmatrix} \begin{bmatrix} \epsilon & \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \lambda & \epsilon & \kappa & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \mu & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \alpha & \underline{\epsilon} & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \alpha & \eta \\ \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \lambda & \epsilon \\ \epsilon & \epsilon & \lambda & \epsilon & \epsilon & \epsilon & \alpha \end{bmatrix} & \begin{matrix} \epsilon \\ e \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \\ e \end{matrix} \end{pmatrix} \right) = \begin{bmatrix} \epsilon & \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \lambda & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \alpha & \eta \\ \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \lambda & \epsilon \\ \epsilon & \epsilon & \lambda & \epsilon & \epsilon & \epsilon & \alpha \end{bmatrix} \begin{matrix} \epsilon \\ e \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \\ e \end{matrix}$$

Calculando $\text{ACES}(\mathbf{St}^2)$, encontra-se

$$\mathbf{St} = \text{ACES}(\mathbf{St}^2) = \begin{bmatrix} \epsilon & \alpha \\ \lambda & \epsilon \end{bmatrix} \begin{matrix} \epsilon \\ e \end{matrix} \quad (0.91)$$

e está mostrado na figura 0.46. Observe que a matriz \mathbf{St} se apresenta com uma dimensão 2, contudo sua dimensão é a mesma de \mathbf{St}^2 , em que os elementos de todas as linhas e colunas $k = 3, \dots, 7$ são iguais a ϵ .

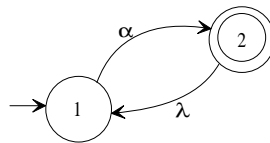


Figura 0.46: Supervisor que gera a $\text{sup } C(L)$ construído através da especificação dada no Exemplo 21.

Aplicação da Álgebra de Dióides aos SEDs Temporizados Modelados por RPT

Como visto anteriormente, a descrição da dinâmica de uma RPT pode ser formalizada através de um sistema matricial baseado na álgebra de dióides. Dessa forma, essas matrizes permitem fazer análises sobre as características da rede como periodicidade, análise de desempenho, etc. Considerando que a RPT seja um modelo de um SED temporizado, é possível avaliar suas características, de maneira a determinar seu desempenho (por exemplo, a taxa de produção em sistemas de manufatura) e construir supervisores para melhorar sua performance.

As características das matrizes das RPT são: o circuito crítico, os auto-valores e auto-vetores e a periodicidade. Para analisar estas características, a seguir é apresentado um exemplo de um sistema de fila modelado por uma RPT para tornar mais simplificada a descrição do problema.

Exemplo 45 Considere a Figura 0.47. Esta figura é um modelo de um simples sistema de fila com conjunto de lugares $P = \{Q, I, B\}$. O estado inicial, isto é, $\mathbf{x} = [0 \ 1 \ 0]$, indica que a fila está vazia e o servidor está inativo. Nesse caso, as únicas transições temporizadas são a e d, correspondentes à chegada e à saída de clientes do servidor. A transição s, sendo

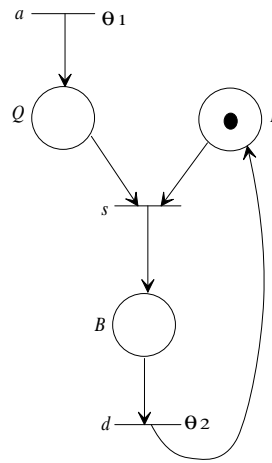


Figura 0.47: Modelo de um sistema de fila utilizando RPT.

não temporizada, não impõe atraso em seu disparo, o que implica que o serviço inicia assim que um cliente chega na fila. A estrutura de tempo desse modelo consiste das seqüências $\theta_1 = \{\theta_{1,1}, \theta_{1,2}, \dots\}$ e $\theta_2 = \{\theta_{2,1}, \theta_{2,2}, \dots\}$, com $\theta_{i,j}$ significando o j -ésimo tempo θ_i . Para descrever sua evolução dinâmica, tem-se que a_k é o k -ésimo tempo de chegada, d_k é o k -ésimo tempo de saída, s_k é o k -ésimo tempo de início de serviço e $t_{Q,k}$ é o tempo quando Q recebe sua k -ésima ficha, $t_{I,k}$ é o tempo quando I recebe sua k -ésima ficha e $t_{B,k}$ é o tempo quando B recebe sua k -ésima ficha, com a condição inicial dada por $t_{I,1} = 0$ (I recebe sua primeira ficha no tempo 0). Assim,

encontra-se que

$$\begin{aligned} a_k &= a_{k-1} + \theta_{1,k}, & k = 1, 2, \dots & \quad a_0 = 0 \\ s_k &= \max(t_{Q,k}, t_{I,k}) & k = 1, 2, \dots & \\ d_k &= t_{B,k} + \theta_{2,k} & k = 1, 2, \dots & \end{aligned} \quad (0.92)$$

e

$$\begin{aligned} t_{Q,k} &= a_k, & k = 1, 2, \dots & \\ t_{I,k} &= d_{k-1} & k = 1, 2, \dots & \quad t_{I,1} = 0 \\ t_{B,k} &= s_k & k = 1, 2, \dots & \end{aligned} \quad (0.93)$$

Combinando a segunda equação de (0.92) com a terceira equação de (0.93) para eliminar $t_{Q,k}$, $t_{I,k}$ e $t_{B,k}$, encontra-se

$$\begin{aligned} s_k &= \max(a_k, d_{k-1}) & k = 1, 2, \dots & \quad d_0 = 0 \\ d_k &= s_k + \theta_{2,k} & k = 1, 2, \dots & \end{aligned}$$

que dá

$$d_k = \max(a_k, d_{k-1}) + \theta_{2,k} \quad k = 1, 2, \dots \quad d_0 = 0$$

que é uma simples relação recursiva caracterizando os tempos de saída dos clientes. Pode-se observar que a k -ésima saída ocorre $\theta_{2,k}$ unidades de tempo após a $(k-1)$ -ésima entrada, exceto quando $a_k > d_{k-1}$. Essa última condição ocorre quando a saída em d_{k-1} leva a uma fila vazia. Neste caso, o servidor deve esperar o próximo tempo de chegada no tempo a_k e gerar a próxima saída no tempo $a_k + \theta_{2,k}$. Disso, tem-se que, para $k = 2, 3, \dots$

$$\begin{aligned} a_k &= a_{k-1} + \theta_{1,k} & a_0 &= 0 \\ d_k &= \max(a_{k-1} + \theta_{1,k}, d_{k-1}) + \theta_{2,k} & d_0 &= 0 \end{aligned} \quad (0.94)$$

que determina o espaço de estados do modelo. O sistema evolui com a seqüência de tempos $\theta_{1,k}$ e $\theta_{2,k}$, e sua saída consiste das seqüências de tempo de chegadas e saídas $\{a_1, a_2, \dots\}$ e $\{d_1, d_2, \dots\}$ geradas através das equações (0.94). Considerando por simplicidade que $\theta_{1,k} = C_a$ e $\theta_{2,k} = C_d$, $\forall k = 1, 2, \dots$, onde C_a e C_d são constantes, e assumindo que $C_a > C_d$, esse sistema de equações pode ser escrito na álgebra $(\max, +)$ como

$$\begin{aligned} a_{k+1} &= (a_k \otimes C_a) \oplus (d_{k-1} \otimes -L) & a_1 &= C_a \\ d_k &= (a_k \otimes C_d) \oplus (d_{k-1} \otimes C_d) & d_0 &= 0 \end{aligned} \quad (0.95)$$

onde $-L$ representa um número negativo suficientemente negativo, tal que

$$(a_k \otimes C_a) \oplus (d_{k-1} \otimes -L) = (a_k \otimes C_a).$$

Em notação matricial, encontra-se

$$\begin{bmatrix} a_{k+1} \\ d_k \end{bmatrix} = \begin{bmatrix} C_a & -L \\ C_d & C_d \end{bmatrix} \begin{bmatrix} a_k \\ d_{k-1} \end{bmatrix}.$$

Definindo

$$\mathbf{x}_k = \begin{bmatrix} a_{k+1} \\ d_k \end{bmatrix}, \mathbf{A} = \begin{bmatrix} C_a & -L \\ C_d & C_d \end{bmatrix},$$

tem-se

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k, \quad \mathbf{x}_0 = \begin{bmatrix} C_a \\ 0 \end{bmatrix}. \quad (0.96)$$

Utilizando esse exemplo serão estudadas as características das matrizes.

Circuito Crítico de uma Matriz

A uma matriz \mathbf{A} associa-se um grafo direcionado como o mostrado na Figura 0.48. O número

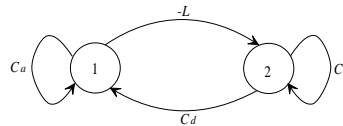


Figura 0.48: Grafo de uma matriz.

de nós é igual à dimensão da matriz \mathbf{A} . Cada arco corresponde um termo (i, j) da matriz, denominado peso. No caso do Exemplo 45, a matriz do grafo da Figura 0.48 é

$$\mathbf{A} = \begin{bmatrix} C_a & -L \\ C_d & C_d \end{bmatrix}.$$

Quando $a_{i,j} = \epsilon$, indica a inexistência de um arco apontando do nó i para o nó j .

Todo laço fechado no grafo forma um circuito. Na Figura 0.48, um exemplo de circuito é dado por $(1, 2, 1)$, representando o conjunto de vértices que o forma.

O comprimento de um circuito é o número de arcos formando o circuito, cujo peso é a soma dos arcos que o forma. Assim, o comprimento do circuito $(1, 2, 1)$ é 2. O peso do circuito $(1, 2, 1)$ é $C_d - L$.

O peso médio de um circuito é seu peso dividido por seu comprimento. Para o caso do circuito $(1, 2, 1)$, tem-se que seu peso médio é $(C_d - L)/2$.

O circuito crítico de uma matriz é, então, o circuito com máximo peso médio. Considerando que na Figura 0.48, $C_a > C_d$, o circuito crítico é $(1, 1)$, desde que $C_a > C_d > (C_d - L)/2$. Note que o peso médio desse circuito é $C_a/1 = C_a$, e que o peso médio do circuito $(2, 2)$ é $C_d/1 = C_d$.

Auto-Valores e Auto-Vetores

A determinação de auto-valores e auto-vetores utiliza o mesmo formalismo da teoria de controle clássico. Isto é, dada uma matriz \mathbf{A} , encontrar λ (auto-valor) e \mathbf{u} (auto-vetor), tal que a seguinte relação seja satisfeita:

$$\lambda \otimes \mathbf{u} = \mathbf{A} \otimes \mathbf{u},$$

em que o vetor \mathbf{u} , não tenha todos os seus elementos iguais a ϵ .

Para o caso do Exemplo 45, tem-se

$$\lambda \otimes \mathbf{u} = \begin{bmatrix} \lambda \otimes u_1 \\ \lambda \otimes u_2 \end{bmatrix}, \quad \mathbf{A} \otimes \mathbf{u} = \begin{bmatrix} C_a & -L \\ C_d & C_d \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} C_a \otimes u_1 \\ C_d \otimes u_1 \oplus u_2 \end{bmatrix}$$

Se \mathbf{u} é um auto-vetor de \mathbf{A} , e λ é um auto-valor, tem-se que

$$\begin{aligned} \lambda &= C_a \\ u_2 \otimes \lambda &= C_d \otimes u_1 \oplus u_2. \end{aligned}$$

Assim, colocando $u_1 = u_2 \otimes C_a \oslash C_d$, onde $C_a > C_d$, encontra-se para um $u \in \mathbb{R}_{\max}$

$$\mathbf{u} = \begin{bmatrix} u \otimes C_a \oslash C_d \\ u \end{bmatrix}$$

sendo um auto-vetor, com $\lambda = C_a$ sendo um auto-valor. Deve-se observar que λ é precisamente o peso médio do circuito crítico. Isto é sempre verdadeiro para o caso da álgebra $(\max, +)$.

Periodicidade

A periodicidade é um valor para o qual uma dada matriz multiplicada por ela mesma não altera os valores de seus elementos. Em termos de sistemas, ela representa a repetição da realização de uma tarefa determinada. Para avaliá-la, será retomado o Exemplo 45.

Assim, tomando a equação (0.96), pode-se escrevê-la como:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} = \mathbf{A}^2\mathbf{x}_{k-2} = \mathbf{A}^k\mathbf{x}_0, \quad \mathbf{x}_0 = \begin{bmatrix} C_a \\ 0 \end{bmatrix}, \quad (0.97)$$

onde $\mathbf{A}^k = \mathbf{A} \otimes \mathbf{A} \otimes \dots \otimes \mathbf{A}$ (k vezes).

Para calcular \mathbf{A}^k e, então solucionar \mathbf{x}_k , reescreve-se a matriz \mathbf{A} como

$$\mathbf{A} = \begin{bmatrix} C_a & -L \\ C_d & C_d \end{bmatrix} = C_a \otimes \begin{bmatrix} e & -L' \\ C_d \oslash C_a & C_d \oslash C_a \end{bmatrix} \quad (0.98)$$

em que $-L' = -L \otimes C_a$ e $C_d \oslash C_a < 0$. Coloca-se então, $\mathbf{A} = C_a \otimes \mathbf{B}$, com

$$\mathbf{B} = \begin{bmatrix} e & -L' \\ C_d \oslash C_a & C_d \oslash C_a \end{bmatrix}$$

e calcula-se $\mathbf{B}^k, k = 2, 3, \dots$

$$\mathbf{B}^2 = \begin{bmatrix} e & -L' \\ C_d \oslash C_a & b_2 \end{bmatrix}, \quad b_2 = -L' \otimes (C_d \oslash C_a) \oplus 2(C_d \oslash C_a),$$

com $2(C_d \oslash C_a) = (C_d \oslash C_a) \otimes (C_d \oslash C_a)$. Procedendo assim, encontra-se

$$\mathbf{B}^k = \begin{bmatrix} e & -L' \\ C_d \oslash C_a & b_k \end{bmatrix}, \quad b_k = -L' \otimes (C_d \oslash C_a) \oplus k(C_a \oslash C_d).$$

Para o valor de b_k , que é o único que muda na potência de \mathbf{B} , pode-se observar que para algum valor k' , encontrar-se-á o valor $k \otimes (C_a \otimes C_d)$ tão pequeno que garante que

$$b_k = -L' \otimes (C_d \otimes C_a), \forall k > k'.$$

Então, para um valor de k suficientemente grande, tem-se

$$\mathbf{B}^k = \begin{bmatrix} e & -L' \\ C_d \otimes C_a & -L' \otimes (C_d \otimes C_a) \end{bmatrix} = \mathbf{P}$$

em que \mathbf{P} é uma matriz fixa. Assim, define-se:

Definição 68 *Uma matriz \mathbf{M} é dita ser n -periódica na álgebra $(\max, +)$ se e somente se existe um inteiro k' tal que*

$$\mathbf{M}^{k+n} = \mathbf{M}^k, \forall k > k'.$$

Observe que estas características podem ser utilizadas para análise de desempenho (por exemplo, a taxa de produção em um sistema de manufatura) em SEDs.

Considerações sobre o Formalismo de Controle para as RPTs

Quando se utilizam as RPTs no formalismo de controle de SEDs temporizados descrevendo-se sua evolução dinâmica através de uma equação de estados do tipo

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k \oplus \mathbf{B}\mathbf{u}_k \quad (0.99)$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k, \quad (0.100)$$

utiliza-se o formalismo apresentado da álgebra de dióides e as características das matrizes para encontrar os valores de entrada para \mathbf{u} , tal que a saída satisfaça os pré-requisitos de uma especificação de comportamento dada, similarmente ao formalismo usado na teoria clássica de controle. Dessa forma, a especificação desejada para o sistema é definida pela saída \mathbf{y} e, por meio dela, determina-se valores específicos para o controlador \mathbf{u} que atrasa os valores da entrada \mathbf{x} (inibe temporalmente eventos), de modo que o sistema siga o comportamento requerido. Este formalismo é apresentado na Figura 0.49.

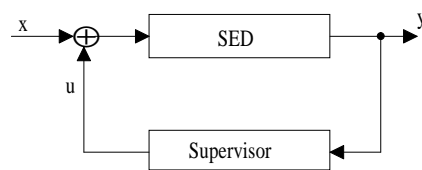


Figura 0.49: Controle de SEDs temporizados utilizando o formalismo dos dióides.

Também, esta mesma fundamentação teórica pode ser utilizada para descrever o comportamento de autômatos $(\max, +)$.

Deve-se levar em consideração que as RPTs em conjunto com a álgebra de dióides podem ser utilizadas para modelar sistemas que exibem sincronização de ações, enquanto que os autômatos podem ser utilizados para sistemas não determinísticos. Assim, nem todos os casos que podem ser tratados quando modela-se o SED por meio de uma RPT podem ser tratados com os autômatos, e *vice-versa*.

De forma semelhante, a álgebra $(\min, +)$ é utilizada para os mesmos desígnios, considerando a formulação da evolução dinâmica do sistema definida através dos contadores.

Bibliografia

- [1] S. Gaubert. *Théorie des Systèmes Linéaires dans les Dioïdes*. PhD thesis, École Nationale Supérieure des Mines de Paris, 1992.
- [2] E. Emre and P.P. Khargonekar. Regulation of split linear systems over rings: Coefficient assignment and observers. *IEEE Transactions on Automatic Control*, AC-27(1):104–113, 1982.
- [3] S. Gaubert, M. Akian, G. Cohen, R. Nikoukhah, and J.P. Quadrat. Linear systems in $(\max, +)$ algebra. In *Proc. Of the 29th Conference on Decision and Control*, 1990.
- [4] S.M. Aji and R.J. McEliece. The generalized distributive law. *IEEE Transactions on Information Theory*, 46(2):325–343, March 2000.
- [5] M. Vidyasagar. *Control Systems Synthesis - a Factorization Approach*. MIT Pres, 1985.
- [6] V. Kucera. *Analysis and Design of Discrete Linear Control Systems*. Academia, 1991.
- [7] 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.
- [8] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. *Synchronisation and Linearity. An Algebra for Discrete Event Systems*. John Wiley Sons, 1992.
- [9] 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.
- [10] E. Menguy. *Contribution à la Commande des Systèmes Linéaires dans les Dioïdes*. Thèse de Doctorat, École Doctorale Sciences Pour L’Ingénieur de Nantes, 1997.
- [11] 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.
- [12] S. Gaubert. Introduction aux systèmes dynamiques à Événements discrets. Notes de Cours, 1999.

- [13] G. Cohen, P. Moller, J.P. Quadrat, and M. Viot. Algebraic tools for the performance evaluation of discrete event systems. *Proceedings of the IEEE*, 77(1):39–56, january 1989.
- [14] S. Gaubert. Performance evaluation of timed automata. Technical report, Institut National de Recherche en Informatique et en Automatique, 1993.
- [15] G. Cohen, S. Gaubert, and J.P. Quadrat. Algebraic system analysis of timed petri nets. In Cambridge University Press J. Gunawardena Ed., editor, *Idempotency - Collection of the Isaac Newton Institute*, 1995.
- [16] S. Gaubert and J. Mairesse. Task resource models and $(\max, +)$ automata. In J. Gunawardena Ed., editor, *Idempotency - Collection of the Isaac Newton Institute*, 1995.
- [17] R. Huuck. *Transformation Von Echtzeit-Bedingungs-/Ereignissystemen in Echtzeitautomaten: Ein Weg Zur Automatischen Verifikation*. PhD thesis, Universität zu Kiel, Juni 1998.
- [18] A. Benveniste, C. Jard, and S. Gaubert. Algebraic techniques for timed systems. 1998. Esprit LTR-SYRF project.
- [19] J. Berstel and C. Reutenauer. *Rational Series and their Languages*. Springer, 1988.
- [20] 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.
- [21] S. Gaubert. On rational series in one variable over certain dioids. Technical report, Institut National de Recherche en Informatique et en Automatique, 1994.
- [22] S. Gaubert. Rational series over dioids and discrete event systems. *Proceedings of the 11th Int. Conf. on Analysis and Optimization of Systems: Discrete Event Systems, Sophia Antipolis, Lectures Notes in Control and Information Sciences, Springer*, (199), June 1994.
- [23] A. Benveniste, C. Jard, and S. Gaubert. Monotone rational series and max-plus algebraic models of real-time systems. *Proceedings of 4th Workshop on Discrete Event Systems, WODES95, Cagliari, Italy*, 1995.
- [24] I. Klimann. *Langages, Séries et Contrôle de Trajectoires*. PhD thesis, l’Université Denis Diderot - Paris 7, Juin 1999.
- [25] 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.
- [26] 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.

- [27] 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.
- [28] P.J.G. Ramadge and W.M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, 1989.
- [29] J.N. Tsitsiklis. On the control of discrete-event dynamical systems. *Mathematics of Control, Signals, and Systems*, 2:95–107, 1989.
- [30] 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.
- [31] J.G. Thistle. Supervisory control of discrete event systems. *Mathl. Comput. Modeling*, 23(11/12):25–53, 1996.
- [32] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, USA, 1979.
- [33] W.M. Wonham. Sed notes. Course notes, 1999.
- [34] A. Gill. *Introduction to the Theory of Finite-State Machines*. McGraw-Hill Electronic Sciences Series. McGRAW-HILL Book Company, 1962.
- [35] J.E. Hopcroft and J.D. Ullman. *Formal Languages and Their Relation to Automata*. Addison-Wesley Publishing Company, 1969.
- [36] 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.
- [37] F. Jahanian and A.K. Mok. Modechart: A specification language for real-time systems. *IEEE Transactions of Software Engineering*, 20(12):933–947, December 1994.
- [38] P. Caspi E. Asarin and O. Maler. A kleene theorem for timed automata. *Proc. of LICS'97*, pages 160–171, 1997.
- [39] E. Asarin. Equations on timed languages. *Proc. of Hybrid'98*, 1998.
- [40] 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.
- [41] R. Alur, S. La Torre, and G.J. Pappas. *Optimal Paths in Weighted Timed Automata*. University of Pennsylvania, Bell Labs and Università degli Studi di Salerno, 2000.

- [42] S. Gaubert. Performance evaluation of $(\max,+)$ automata. *IEEE Transactions on Automatic Control*, 40(12):2014–2025, December 1995.
- [43] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.
- [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] B.A. Brandin and W.M. Wonham. Supervisory control of timed discrete-event systems. *IEEE Transactions on Automatic Control*, 39(2):329–342, 1994.
- [46] 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.
- [47] S. Tripakis. *L'Analyse Formelle Des Systèmes Temporisés En Pratique*. PhD thesis, L'Université Joseph Fourier, Décembre 1998.
- [48] L. Fribourg. *A Closed-Form Evaluation for Extended Timed Automata*. Research Report LSV-98-2, Laboratoire Spécification et Vérification, Ecole Normale Supérieure de Cachan, France, 1998.
- [49] 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.
- [50] R. Alur, C. Courcoubetis, and D. Dill. Model checking in dense real-time. *Information and Computation*, 104(1):2–24, 1993.
- [51] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, (126):183–235, 1994.
- [52] R. Alur and D. Dill. Automata-theoretic verification of real-time systems. Technical report, Computing Science Research (Bell Labs) and Computer Science Department (Stanford University), 1995.
- [53] F. Balarin. Approximate reachability analysis of timed automata. *IEEE - 1052-8725/96*, pages 52–61, 1996.
- [54] R. Alur. Timed automata. *Proc. NATO-ASI Summer School, Antalaya, Turkey*, 1997.
- [55] K. Okano, S. Hattori, A. Yamamoto, T. Higashino, and K. Tanigushi. Specification of real-time systems using a timed automata model with shared variables and verification of partial-deadlock freeness. *IEEE - 0-7695-0358-5/99*, pages 576–581, 1999.

- [56] P. Chrétienne. *Les Réseaux de Petri Temporisés*. Thèse d'état. Université Paris VI, Paris, 1983.
- [57] J.L. Peterson. *Petri Net Theory and Modeling of Systems*. Prentice Hall, 1981.
- [58] W. Reisig. *Petri Nets: an Introduction*. Springer-Verlag Berlin Heidelberg, 1985.
- [59] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.
- [60] W. Reisig. *A Primer in Petri Net Design*. Springer-Verlag, 1992.
- [61] M.C. Zhou and F. DiCesare. *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Kluwer Academic Publishers, 1993.
- [62] 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.
- [63] E.M.M. Costa. *Contribuição ao Uso da Lógica Temporal na Especificação de Comportamentos de Sistemas a Eventos Discretos*. Dissertação de Mestrado. Universidade Federal da Paraíba, Campus II, Campina Grande, PB, Brasil, Outubro 1997.
- [64] R. Smedinga. *Control of Discrete Events*. PhD thesis, Rijksuniversiteit Groningen, 20 januari 1989.
- [65] 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.
- [66] R. Smedinga. Discrete event systems. Technical report, University of Groningen, October 1993.
- [67] H. Marchand and M. Le Borgne. Partial order control and optimal control of discrete event systems modeled as polynomial dynamical systems over galois fields. Technical Report 3273, Institut National De Recherche en Informatique et en Automatique - INRIA, Octobre 1997.
- [68] 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.
- [69] C.R.C. Torrico. Implementação de controle supervisorio de sistemas a eventos discretos aplicado a processos de manufatura. Technical report, Universidade de Santa Catarina, Santa Catarina, Brasil, 15 de Março 1999.

- [70] O. Boivineau H. Marchand and S. Lafortune. Optimal control of discrete event systems under partial observation. Technical Report 4046, Institut National de Recherche en Informatique et en Automatique - INRIA, Novembre 2000.
- [71] K.Q. Pu. Modeling and control of discrete event systems with hierarchical abstraction. Master's thesis, University of Toronto, 2000.
- [72] C. Jard A. Benveniste, E. Fabre and S. Haar. Diagnosis of asynchronous discrete event systems, a net unfolding approach. Technical Report 4181, Institut National de Recherche en Informatique et en Automatique - INRIA, Mai 2001.
- [73] J.E.R. Cury, C.R.C. Torrico, and A.E.C. Da Cunha. A new approach for supervisory control of discrete event systems. 2001.
- [74] Z. Zhang. Smart TCT: An efficient algorithm for supervisory control design. Master's thesis, University of Toronto, June 2001.
- [75] X.R. Cao and Y.C. Ho. Models of discrete event dynamic systems. *IEEE Control System Magazine*, 10(4):69–76, 1990.
- [76] R. Milner. *A Calculus of Communicating Systems*. Springer Verlag, New York, USA, 1980.
- [77] C.A.R. Hoare. *Communicating Sequential Process*. Prentice-Hall, London,UK, 1985.
- [78] F. DiCesare, G. Harhalakis, J.M. Proth, M. Silva, and F.B. Vernadat. *Practice of Petri Nets in Manufacturing*. Chapman and Hall, 1993.
- [79] J.A. Beachy. *Abstract Algebra II*. Waveland Press, Inc., 1996.
- [80] R. Kumar and V.K. Garg. Optimal control of discrete event dynamical systems using network flow techniques. *Proceedings of 1991 Annual Allerton Conference*, pages 705–714, October 1991.
- [81] P.A.S. Veloso. *Máquinas e Linguagens. Uma Introdução à Teoria de Autômatos*. Escola de Computação Instituto de Matemática e Estatística da USP - São Paulo, 1979.
- [82] B.A. Brandin, W.M. Wonham, and B. Benhabib. Manufacturing cell supervisory control - a timed discrete event system approach. In *Proceedings of the 1992 IEEE International Conf. On Robotics and Automation*, pages 931–936, May 1992.
- [83] M.S. Lawford. *Model Reduction of Discrete Real-Time Systems*. PhD thesis, University of Toronto, 1997.
- [84] 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.

- [85] 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.
- [86] D.D. Cofer and V.K. Garg. Supervisory control of timed event graphs. *IEEE - 0-7803-2129-4/94*, 1994.
- [87] 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.
- [88] S. Gaubert. On the burnside problem for semigroups of matrices in the $(\max,+)$ algebra. *Semigroup Forum, Springer-Verlag New York Inc.*, 52:271–292, 1996.