

Apostila
Álgebra de Dióides

Eduard Montgomery Meira Costa, DSc
UFBA

©Eduard Montgomery Meira Costa, DSc
UFBA, 2002

Introdução

A álgebra de dióides 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 [1, 2, 3, 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:

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

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

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

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 $s = cat_{\Sigma}(\alpha, \beta) = \alpha\beta$, 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*

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

onde

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

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

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

sua concatenação define uma nova palavra que é

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

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

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

Um conjunto de palavras é definido como:

Definição 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

$$\begin{aligned} cat_L(\{\varepsilon\}, L) &= cat_L(L, \{\varepsilon\}) = L, L \subset \Sigma^* \\ cat_L(L_1, L_2) &= \{s_1 s_2 | s_1 \in L_1 \wedge s_2 \in L_2\}. \end{aligned}$$

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

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

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

e

$$\emptyset + L = L,$$

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

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

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

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

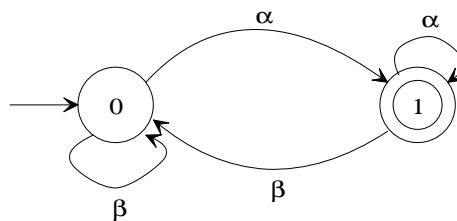


Figura 0.1: Autômato com linguagem marcada $L_m(A) = \{\alpha, \alpha\alpha, \beta\alpha, \alpha\beta\alpha, \beta\alpha\alpha, \beta\beta\alpha, \dots\}$.

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 $L_1, L_2 \in D$,*

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

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

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

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

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

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:

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

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 $L = \{\varepsilon, \alpha, \alpha\beta, \beta\alpha, \alpha\alpha, \beta\beta, \beta\alpha\beta\}$ é representada pela série formal

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

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

$$\begin{aligned} Y_L &= e\varepsilon \oplus e\alpha \oplus e\alpha\beta \oplus e\beta\alpha \oplus e\alpha\alpha \oplus e\beta\beta \oplus e\beta\alpha\beta & \text{ou} \\ Y_L &= \varepsilon \oplus \alpha \oplus \alpha\beta \oplus \beta\alpha \oplus \alpha\alpha \oplus \beta\beta \oplus \beta\alpha\beta \end{aligned}$$

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

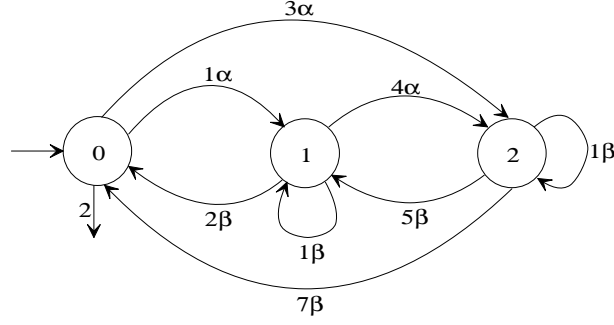


Figura 0.2: Autômato $(\max, +)$.

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:

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

- x. Estando no estado 3, a passagem de 2 unidades de tempo habilita β , que permanece habilitado até sua ocorrência.
- xi. Se ocorre β no estado 3, o autômato alcança o estado 1.
- xii. Novamente no estado 1, a passagem de mais uma unidade de tempo permite o reconhecimento da palavra $11\alpha\kappa\beta$.

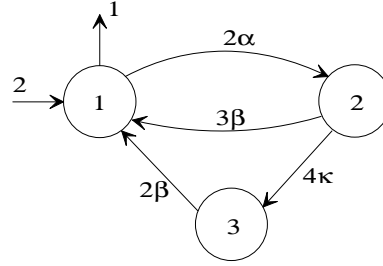


Figura 0.3: Autômato $(\max, +)$.

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*

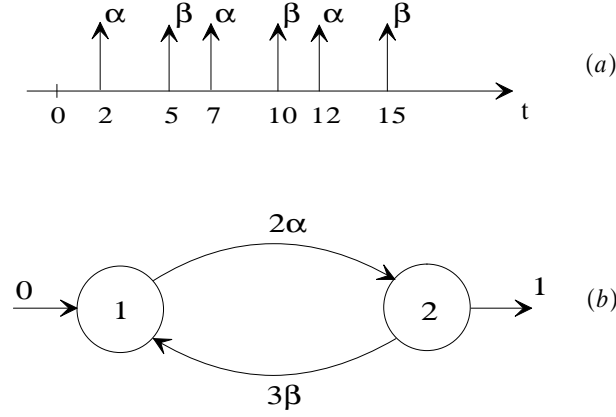


Figura 0.4: (a) Autômato $(\max, +)$ e (b) Diagrama temporal de ocorrência de seus eventos α e β .

12, enquanto que os menores tempos de ocorrência do evento β são 5, 10 e 15. Dessa forma, encontram-se as seqüências

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

Logo, o vetor x é definido por

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

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

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

A evolução dinâmica do autômato $(\max, +)$ pode ser descrita por meio de um sistema linear através da matriz de tempos de transição \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.*

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*

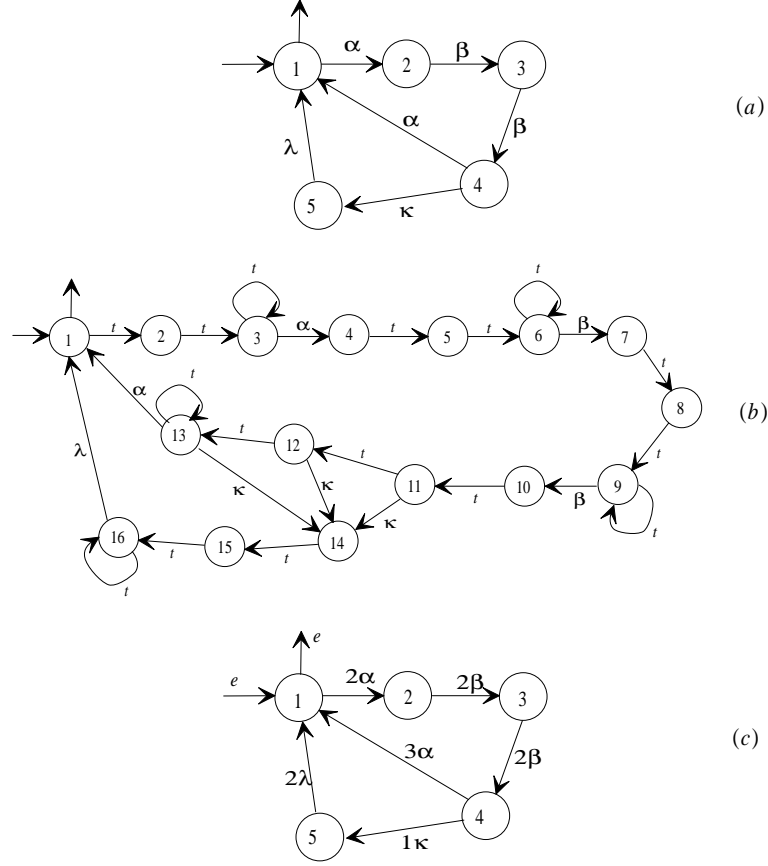


Figura 0.5: (a) Grafo de transições de atividades, (b) Grafo de transições temporizadas e (c) Autômato $(\max, +)$.

$t_s, t_{s'}, \dots \in \mathbb{R}_{\max}$, 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

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

ou

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

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

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

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

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

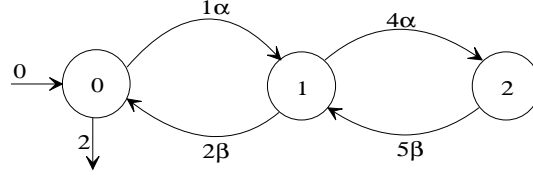


Figura 0.6: Autômato $(\max, +)$ determinístico.

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

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

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

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

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 .

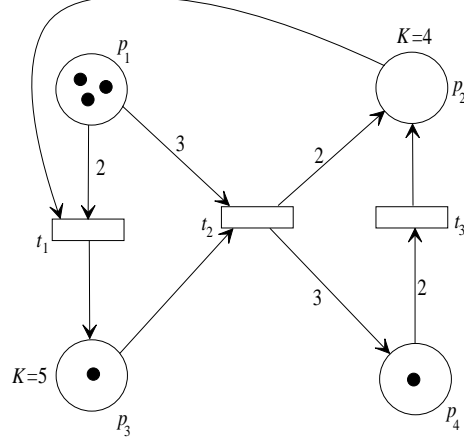


Figura 0.7: Exemplo de rede de Petri.

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

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

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 $M[t] M'$.

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

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

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

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

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

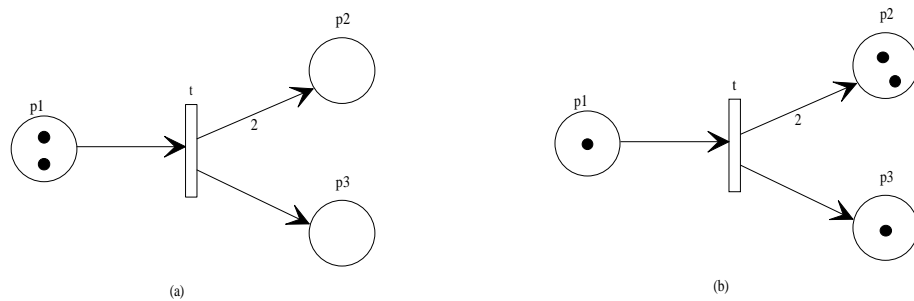


Figura 0.8: Exemplo de uma rede de Petri antes e após o disparo de sua transição.

Observação: Utiliza-se também a notação $M_k[t]$ 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)).

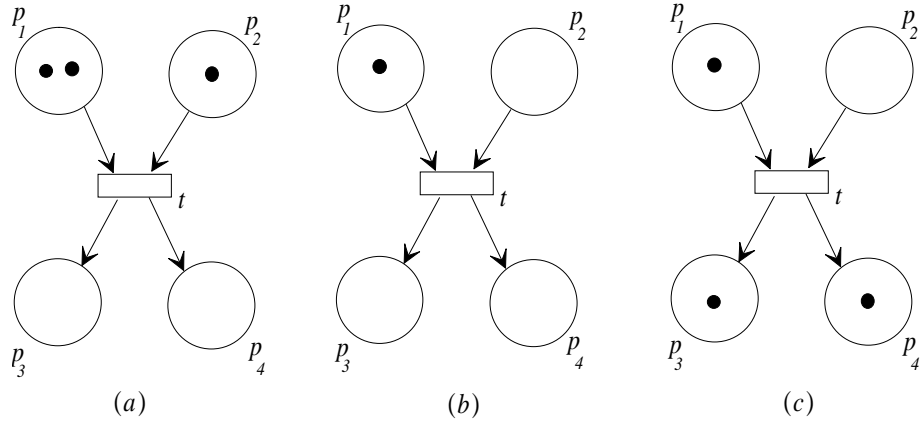


Figura 0.9: Exemplo de uma rede de Petri temporizada antes e após o disparo de sua transição.

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 ($\theta(t_1) = \theta(t_2) = 0$) e com o valor da temporização passado ao lugar introduzido como $\theta(p_1) = \mathbf{t}$. 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 .

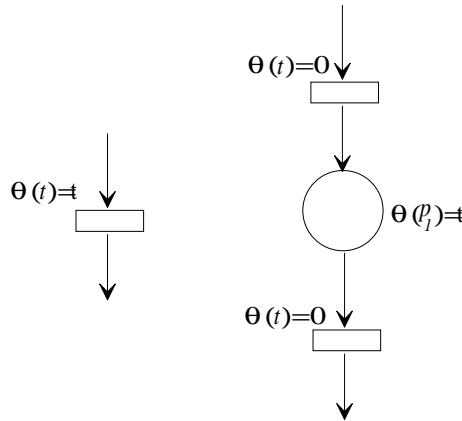


Figura 0.10: Exemplo de transformação de transições temporizadas.

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.

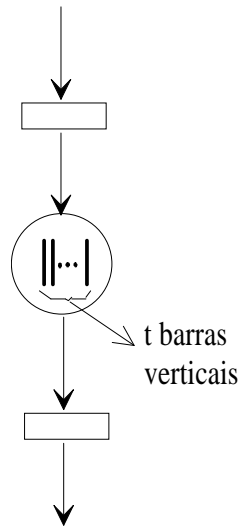


Figura 0.11: Representação utilizada para GETs.

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, pode-se ver que o lugar entre as transições x_2 e x_3 é temporizado em 3 unidades de tempo, e o lugar entre as transições x_1 e x_3 , em 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

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

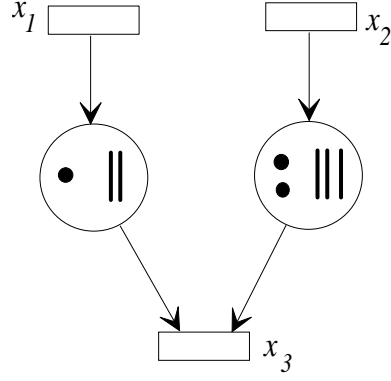


Figura 0.12: RPT.

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:

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

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

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

que matricialmente pode ser escrita como

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

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

que é um sistema da forma

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

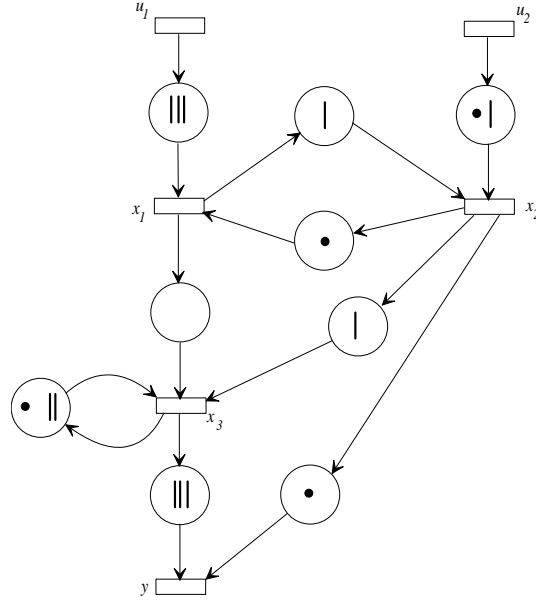


Figura 0.13: RPT para formulação da equação dinâmica na álgebra $(\max, +)$.

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 $\max(3, 2) = 3$.

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:

$$\forall a \in \mathbb{R}, a \oplus \epsilon = a \oplus (+\infty) = \min \{a, +\infty\} = a = \min \{+\infty, a\} = (+\infty) \oplus a = \epsilon \oplus a. \quad (0.50)$$

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

$$\forall a \in \mathbb{R}, a \otimes e = a + 0 = a = 0 + a = e \otimes a. \quad (0.51)$$

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,

$$D = \mathbb{R}_{\min}^{n \times n}. \quad (0.52)$$

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

$$\begin{aligned} x_1(t) &\leq \min(u_1(t-3), x_2(t) + 1) \\ x_2(t) &\leq \min(1 + u_2(t-1), x_1(t-1)) \\ x_3(t) &\leq \min(x_1(t), x_2(t-1), 1 + x_3(t-2)) \\ y(t) &\leq \min(1 + x_2(t), x_3(t-3)), \end{aligned} \quad (0.53)$$

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

$$\begin{aligned}
 x_1(t) &\succeq u_1(t-3) \oplus 1x_2(t) \\
 x_2(t) &\succeq 1u_2(t-1) \oplus x_1(t-1) \\
 x_3(t) &\succeq x_1(t) \oplus x_2(t-1) \oplus 1x_3(t-2) \\
 y(t) &\succeq 1x_2(t) \oplus x_3(t-3).
 \end{aligned} \tag{0.54}$$

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 (especificamente 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.

Bibliografia

- [1] 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.
- [2] 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.
- [3] S. Gaubert. *Théorie des Systèmes Linéaires dans les Dioïdes*. PhD thesis, École Nationale Supérieure des Mines de Paris, 1992.
- [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, Antalya, 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.