

AUTÔMATOS COM TEMPORIZAÇÃO VARIÁVEL

TIMING VARYING AUTOMATON

Eduard Montgomery Meira Costa
Departamento de Engenharia Elétrica
Universidade Federal da Bahia
Salvador, BA, Brasil
Email: eduard@dee.eng.ufba.br

ABSTRACT

The timing varying automaton is introduced in this paper. This automaton ensure systems modelling that present state transitions which associated time is variable. The formalization of its languages is presented in two ways: through formal series and based in incidence matrix. Both formalizations are defined on the dioid algebra.

KEY-WORDS: Timed Automata, Systems Modelling

RESUMO

O autômato com temporização variável é introduzido nesse artigo. Esse autômato possibilita modelar sistemas que apresentam transições de estados cujo tempo associado é variável. A formalização de suas linguagens são apresentadas de duas maneiras: através das séries formais e das matrizes de incidência. Ambas formalizações são definidas sobre a álgebra de dióides.

PALAVRAS-CHAVE: Autômatos Temporizados, Modelagem de Sistemas

1 INTRODUÇÃO

Autômatos [1] são modelos de máquinas de estado que podem representar algumas classes de sistemas. Dentre esses sistemas, encontram-se os Sistemas a Eventos Discretos (SEDs) [2], que são sistemas cuja evolução dinâmica é descrita pela ocorrência de eventos que alteram o estado do sistema.

Os autômatos permitem estruturar modelos de sistemas em que é necessário estudar seu funcionamento lógico através de sua evolução dinâmica descrita na forma de linguagens formais [3]. Exemplo disto é a formalização do problema de controle de SEDs, que é denominada de Teoria de Controle Supervisório [4].

Quando se necessita expressar uma representação temporal em um modelo de um sistema, um paradigma que pode ser utilizado é o autômato temporizado [5–7]. Esse autômato permite incluir representações de tempo em sua estrutura, tal que os símbolos ocorrem de acordo com instantes de tempo específicos, determinados em um ou mais relógio global. No caso do autômato temporizado, a linguagem é representada por um conjunto de pares (tempo, símbolo), sendo denominada de linguagem temporizada [8–10].

Outro formalismo utilizado para a representação temporal nos autômatos são os grafos de transições de atividades (GTAs) [11], que apresentam uma estrutura formal seme-

lhante aos autômatos não temporizados. Porém, os GTAs incluem as definições dos tempos máximo e mínimo associados aos símbolos, em que um símbolo é definido como uma tripla $(t_{\min}, \text{evento}, t_{\max})$, em que t_{\min} é o menor tempo em que um símbolo torna-se habilitado e t_{\max} é o maior tempo em que o símbolo pode ocorrer. A evolução dinâmica dos GTAs pode ser representada através dos grafos de transições temporizadas (GTTs) [12], que incluem o símbolo t (tick) que é sincronizado a um relógio global. Esta representação gráfica apresenta um aumento de estados que ocorre devido à inclusão deste símbolo. Este tipo de autômato é utilizado por Brandin e Wonham [13] para controle de SEDs temporizados. Os GTAs em conjunto com os GTTs permitem representar sistemas que apresentam transições de estados com limites máximo e mínimo de ocorrência. A linguagem associada é uma linguagem formal que inclui o símbolo t .

Para a representação de sistemas em que os tempos são definidos como apenas o limite mínimo em que um símbolo pode ocorrer, o que representa um modelo livre de controle, encontra-se o autômato $(\max, +)$ [14], que é similar ao GTA, mas apresentando as definições dos tempos associados aos símbolos apresentados diretamente nos arcos (representação gráfica das funções de transição). Este tipo de autômato temporizado é utilizado em conjunto com sua representação matricial [15] para tratar o problema de controle de SEDs temporizados [16, 17].

Dentre os vários formalismos de autômatos temporizados, nenhum se apresenta para representar diretamente modelos de sistemas em que o tempo associado ao símbolo varie de acordo com o último tempo decorrido em uma transição passada. Geralmente, os formalismos citados apresentam tempos constantes (como o autômato $(\max, +)$) ou definidos por intervalos de tempo (como os GTAs). Assim, para sistemas em que, a cada momento que ocorre um retorno a um estado anterior, o tempo associado ao símbolo neste estado recorrente adquire um novo valor dependente do tempo decorrido na ocorrência da transição que levou o autômato a este estado, estes modelos não satisfazem tais condições. Logo, não podem representar o sistema diretamente, podendo o modelo do sistema se apresentar como não determinístico, ou como infinito. Dessa forma, esses casos necessitam da definição de um paradigma mais amplo, tal que os tempos associados aos símbolos sejam expressos por funções dependentes do último tempo passado. Com essa definição, é possível representar sistemas que apresentem essas características. Assim, esse artigo apresenta este paradigma, o qual é denominado de autômato com temporização variável (ATV), sua semântica e o formalismo das matrizes de incidência e linguagens associadas.

Esse artigo é organizado como segue: na Seção 2 é apresentada a definição do ATV, sua representação gráfica, sua semântica e o formalismo dos datadores associados e linguagens; na Seção 3 é apresentada a representação matricial do ATV e o formalismo para a construção de suas linguagens através desta representação; na Seção 4 é apresentado um exemplo de modelo de sistema utilizando o ATV e na Seção 5 são apresentadas as conclusões.

2 AUTÔMATOS COM TEMPORIZAÇÃO VARIÁVEL

Um autômato temporizado inclui uma representação do tempo na sua estrutura de transição, em que um intervalo de tempo está relacionado com cada função de transição, denominado tempo de vida do evento. O tempo de vida expressa o menor tempo para que o evento se torne habilitado. Um autômato temporizado reconhece uma linguagem temporizada [18], que é uma linguagem formalizada sobre pares (*evento*, *tempo*). Dentre as estruturas de autômatos temporizados, encontra-se a formalização do autômato $(\max, +)$ [14] que apresenta além de uma função de transição, um tempo de vida associado ao arco, bem como tempos de atrasos iniciais e atrasos finais. Este autômato pode ser utilizado para modelagem de sistemas em que o tempo de vida do evento seja definido como o instante de tempo em que o evento se torna habilitado, permanecendo habilitado até sua ocorrência [17]. No entanto, para casos em que em um determinado estado alcançado o tempo de vida do evento varia de acordo com o último tempo decorrido para o autômato chegar a este estado, os formalismos existentes não são suficientes para tal descrição. Dessa forma, é introduzido a

seguir o autômato com temporização variável, que comporta essa condição:

Definição 1 Um autômato finito ATV sobre um alfabeto Σ é uma sextupla

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

em que:

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

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

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

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

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

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

$$t_f = f(t_{i-1}) \neq \epsilon, \quad (2)$$

com t_f definido de forma igual à t_i .

Exemplo 1 Na Figura 1 é visto um exemplo de um ATV. Observe que cada arco tem uma função temporal que determina o tempo de vida do evento dependente do tempo decorrido anteriormente, seja ele o tempo de atraso inicial (no estado inicial), seja pelo tempo de vida do último evento ocorrido no ATV. As funções de tempo de transição são $f_1 = t^2 - t$ e $f_2 = t$, e o tempo de atraso final é definido pela função $t_f = \sqrt{t}$. Assim, decorrido o tempo de atraso inicial $t_0 = 3$, o evento α tem um tempo de vida dado por $f_1 = 3^2 - 3 = 6$. Com a ocorrência do evento

α , o estado 2 é alcançado, e o tempo de vida do evento β é $f_2 = 6$. Com sua ocorrência, o estado inicial é novamente alcançado e o novo tempo de vida do evento α é dado por $f_1 = 6^2 - 6 = 30$, e assim por diante. Os tempos de atraso final para uma palavra ser reconhecida (quando o estado 2 é alcançado) são dependentes do tempo decorrido no evento α , isto é, ao ocorrer a primeira vez o evento α , $t_f = \sqrt{6}$. Na segunda vez que o evento α ocorre, $t_f = \sqrt{30}$ e assim por diante.

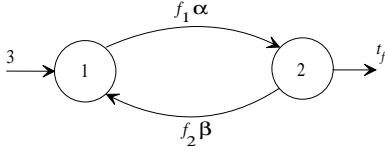


Figura 1: Exemplo de um ATV.

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

Com a definição do ATV, sua semântica é definida de maneira similar ao autômato $(\max, +)$, considerando as variações nos tempos das transições, como a seguir: 1) Há um relógio global sendo incrementado; 2) O tempo de vida t_i de um evento σ , é uma função f do último tempo decorrido t_{i-1} de um evento σ^{prime} , que levou o ATV para o estado em que o evento σ é definido, e seu valor calculado é o tempo mínimo necessário para sua habilitação; 3) Para iniciar a execução do autômato (alcançar o estado inicial) é transcorrido um tempo t_0 no relógio global; 4) Estando no estado inicial (ou em qualquer outro), os contadores dos eventos definidos neste estado (valores calculados de cada função de tempo de vida de evento - $t_i(\sigma)$) vão sendo decrementados; 5) Quando um contador de um dos eventos definidos no estado é zerado, o evento torna-se habilitado e pode ocorrer a qualquer instante; 6) Se com a incrementação do relógio global, for zerado o contador de um outro evento definido neste estado, ele também torna-se habilitado; 7) A ocorrência de um evento habilitado reinicializa seu contador e muda o estado do autômato, desabilitando os outros eventos; 8) Ao atingir um estado marcado, ao transcorrer o tempo de atraso final, o ATV reconhece este estado, reconhecendo assim a palavra que o levou do estado inicial até este estado; 9) Eventos iguais com diferentes funções de tempos de vida definem não determinismo no ATV.

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

Exemplo 2 Seja $\Sigma = \{\alpha, \beta\}$. O autômato com conjunto de estados $Q = \{1, 2, 3\}$, atraso inicial $t_0 = 2$, funções de tempos de transições $f_1(t) = t^2 - t$, $f_2(t) = 3t$, $f_3(t) = 4$,

$f_4(t) = 1 + 2t$ e função de atraso final $f_5(t_{i-1}) = 5 - \frac{2}{t}$ (os outros valores de $f(t_i)$ e $f(t_f)$ são ϵ) está apresentando na Figura 2. O valor de t é interpretado como t_{i-1} , isto é, o último tempo decorrido para atingir um dado estado. Assim, o tempo de cada transição em um determinado instante de tempo é sempre calculado em relação ao tempo decorrido para atingir o estado em que o autômato se encontra. Assim, o valor do primeiro tempo de transição no arco $q_1 \xrightarrow{\alpha} q_2$ é $t_1 = f_1(2) = 2$, desde que $t_0 = 2$. t_1 é o tempo mínimo para a ocorrência do evento α que muda o estado do autômato de q_1 para q_2 . O tempo final para reconhecer a palavra vazia ϵ é $t_f = f_5(2) = 4$, tal que seu reconhecimento tem um tempo total $t = 6$. Na ocorrência de α , o autômato vai para o estado 2, onde o tempo de transição para α é $t_2 = f_2(2) = 6$ e para β é $t_2 = f_4(2) = 5$. Se ocorre α , o autômato vai para o estado 3, onde o tempo de transição para β é $t_3(5) = 4$. Ainda no estado 2, se ocorre β , o autômato vai para o estado 1, onde o novo tempo de transição para α é $t_3(6) = 30$ e para reconhecer a palavra $\alpha\beta$, o atraso final é $t_f = 4,67$, tal que seu reconhecimento tem um tempo total $t = 13,67$.

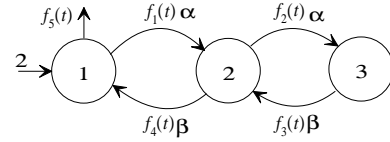


Figura 2: Autômato com Temporização Variável

Pode-se observar que o reconhecimento de uma palavra se dá quando um estado marcado é atingido e o tempo de atraso final é transcorrido. Este reconhecimento pode ser descrito por meio das seqüências de eventos através dos datadores definidos sobre o conjunto $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$, do dióide $(\max, +)$ [19]. Para isto, é necessário as seguintes definições:

Definição 2 Um dióide é um conjunto D dotado de duas operações: \oplus (soma) e \otimes (multiplicação), que satisfaz os seguintes axiomas:

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

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

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

Axioma 4: Distributividade de \otimes sobre \oplus :

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

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

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

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

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

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

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

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

$$\forall a \in \mathbb{R}, \quad a \oplus \epsilon = a \oplus (-\infty) = \max \{a, -\infty\} = a = \max \{-\infty, a\} = (-\infty) \oplus a = \epsilon \oplus a \quad (3)$$

e

$$\forall a \in \mathbb{R}, \quad a \otimes e = a + 0 = a = 0 + a = e \otimes a. \quad (4)$$

Da equação (4), vê-se que a álgebra $(\max, +)$ é um dióide comutativo.

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

$$D = \mathbb{R}_{\max}^{n \times n}. \quad (5)$$

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

$$(\mathbf{A} \oplus \mathbf{B})_{i,j} = \mathbf{A}_{i,j} \oplus \mathbf{B}_{i,j} \quad (6)$$

$$(\mathbf{A} \otimes \mathbf{B})_{i,j} = \bigoplus_{i=1}^n (\mathbf{A}_{i,j} \otimes \mathbf{B}_{i,j}). \quad (7)$$

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

Definição 3 Um datador é um mapa

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

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

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

O datador define os tempos das palavras de Σ^* . Assim, o autômato *ATV* leva um tempo Y para percorrer uma sequência s , partindo de um estado q para um outro estado q' , onde $(y|s)$ denota o valor de Y na palavra s [20].

Definição 4 Um datador Y é dito reconhecível se existir um *ATV* tal que $(y|s) \neq \epsilon$.

Exemplo 3 No Exemplo 2, vê-se que $s = \alpha\beta$ é uma palavra reconhecida, pois

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

3 REPRESENTAÇÃO MATRICIAL

Dentre as representações dos autômatos, a matriz de incidência é uma importante definição. Esta representação é utilizada para formalismos de construção de linguagens em autômatos não temporizados [15], bem como para autômatos temporizados $(\max, +)$ [17]. No caso dos *ATVs*, esse formalismo é semelhante ao formalismo dos autômatos $(\max, +)$. A definição da matriz de incidência de um *ATV* é apresentada a seguir:

Definição 5 Seja *ATV* um autômato com temporização variável. Define-se a matriz de incidência temporizada, denotada por \mathbf{A} , como

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

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

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

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

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

e os estados marcados são representados pelo vetor coluna

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

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

Exemplo 4 O autômato *ATV* apresentado na Figura 2, tem sua matriz de incidência dada por

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

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

3.1 LINGUAGENS

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

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

3.1.1 Séries Formais

As séries formais provêm um formalismo para tratar as linguagens formais, sejam elas temporizadas ou não temporizadas. Uma série formal é definida como [20, 21]:

Definição 6 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 .

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

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

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

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

$$(y_1 \oplus y_2|s) = (y_1|s) \oplus (y_2|s) \quad (8)$$

$$(y_1 \otimes y_2|s) = \bigoplus_{uv=s} (y_1|u) \otimes (y_2|v), \quad (9)$$

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

As séries formais permitem descrever linguagens através da equação (9). Uma linguagem não temporizada pode ser descrita por meio de uma série formal definindo $D = \mathbb{B} = \{\epsilon, e\}$, que denota o semi-anel *binário*, como a seguir.

Definição 7 Uma linguagem regular $L = \{s, s', \dots\} \subseteq \Sigma^*$, é representada pela série formal

$$Y_{\mathbb{B}\langle\langle\Sigma\rangle\rangle} = \bigoplus_{s \in \Sigma^*} (y|s) s \quad (10)$$

onde $\mathbb{B}\langle\langle\Sigma\rangle\rangle$ é o semi-anel das séries formais com coeficientes em \mathbb{B} e variáveis não comutativas em Σ .

Exemplo 5 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

$$Y_{\mathbb{B}\langle\langle\Sigma\rangle\rangle} = e\epsilon \oplus e\alpha \oplus e\alpha\beta \oplus e\beta\alpha \oplus e\alpha\alpha \oplus e\beta\beta \oplus e\beta\alpha\beta \oplus \underbrace{e\alpha\alpha\alpha \oplus \dots \oplus e\beta\beta\beta \oplus \dots}_{\Sigma^* - L} \quad (11)$$

onde $(y|s) = e, \forall s \in L$, e $(y|s) = \epsilon, \forall s \in \Sigma^* - L$. A série formal da equação (11) pode ser escrita como

$$Y_{\mathbb{B}\langle\langle\Sigma\rangle\rangle} = e\epsilon \oplus e\alpha \oplus e\alpha\beta \oplus e\beta\alpha \oplus e\alpha\alpha \oplus e\beta\beta \oplus e\beta\alpha\beta \quad \text{ou} \\ Y_{\mathbb{B}\langle\langle\Sigma\rangle\rangle} = \epsilon \oplus \alpha \oplus \alpha\beta \oplus \beta\alpha \oplus \alpha\alpha \oplus \beta\beta \oplus \beta\alpha\beta$$

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

Deve-se observar que para uma linguagem L descrita por uma série formal $Y_{\mathbb{B}\langle\langle\Sigma\rangle\rangle}$, as operações de união e concatenação de linguagens se apresentam como a soma e a multiplicação de suas respectivas séries formais.

Quando se considera $D = \mathbb{R}_{\max}\langle\langle\Sigma\rangle\rangle$, as séries formais permitem descrever uma linguagem temporizada [8–10] que é uma linguagem que apresenta em cada palavra $s \in L$ um valor numérico t_s associado. Este valor numérico representa um intervalo de tempo decorrido para a palavra ser reconhecida por um autômato temporizado [5–7].

Definição 8 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

$$Y_{\mathbb{R}_{\max}\langle\langle\Sigma\rangle\rangle} = \bigoplus_{s \in \Sigma^*} (y|s) s \quad (12)$$

onde $\mathbb{R}_{\max}\langle\langle\Sigma\rangle\rangle$ é o semi-anel das séries formais com coeficientes em \mathbb{R}_{\max} e variáveis não comutativas em Σ .

Exemplo 6 Dado o alfabeto $\Sigma = \{\alpha, \beta\}$, a linguagem $L = \{3\epsilon, 4\alpha, 2\alpha\beta, 3\beta\alpha, 5\alpha\alpha, 2\beta\beta, \beta\alpha\beta\}$ é representada pela série formal

$$Y_{\mathbb{R}_{\max}\langle\langle\Sigma\rangle\rangle} = 3\epsilon \oplus 4\alpha \oplus 2\alpha\beta \oplus 3\beta\alpha \oplus 5\alpha\alpha \oplus 2\beta\beta \oplus e\beta\alpha\beta \oplus \underbrace{e\alpha\alpha\alpha \oplus \dots \oplus e\beta\beta\beta \oplus \dots}_{\text{seqüências de eventos não pertencentes à } L}$$

ou

$$Y_{\mathbb{R}_{\max}\langle\langle\Sigma\rangle\rangle} = 3\epsilon \oplus 4\alpha \oplus 2\alpha\beta \oplus 3\beta\alpha \oplus 5\alpha\alpha \oplus 2\beta\beta \oplus e\beta\alpha\beta \\ Y_{\mathbb{R}_{\max}\langle\langle\Sigma\rangle\rangle} = 3 \oplus 4\alpha \oplus 2\alpha\beta \oplus 3\beta\alpha \oplus 5\alpha\alpha \oplus 2\beta\beta \oplus \beta\alpha\beta$$

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.

3.1.2 Descrição das Linguagens dos ATVs por Séries Formais

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

Definição 9 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

$$Y_{\mathbb{R}_{\max}(\langle \Sigma \rangle)} = \bigoplus_{s \in \Sigma^*} (y|s) s,$$

onde $\mathbb{R}_{\max}(\langle \Sigma \rangle)$ é o semi-anel das séries formais com coeficientes em \mathbb{R}_{\max} e variáveis não comutativas em Σ .

Exemplo 7 Dado o alfabeto $\Sigma = \{\alpha, \beta\}$, a linguagem $L = \{3\epsilon, 4\alpha, 6, 3\alpha\beta, 3\beta\alpha\}$ é representada pela série formal

$$Y_{\mathbb{R}_{\max}(\langle \Sigma \rangle)} = 3\epsilon \oplus 4\alpha \oplus 6, 3\alpha\beta \oplus 3\beta\alpha = 3 \oplus 4\alpha \oplus 6, 3\alpha\beta \oplus 3\beta\alpha$$

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

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

Definição 10 A linguagem temporizada L de um ATV é definida por uma série formal como

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

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

Exemplo 8 Considerando o autômato do Exemplo 2, a série formal $Y_{\mathbb{R}_{\max}(\langle \Sigma \rangle)}$ identificada com a linguagem temporizada do autômato $L(ATV)$ admite a expressão racional

$$Y_{\mathbb{R}_{\max}(\langle \Sigma \rangle)} = 6 \oplus 13,67\alpha\beta \oplus 74,95\alpha\beta\alpha\beta \oplus 27,78\alpha\alpha\beta\beta \oplus \dots$$

3.1.3 Construção das Linguagens dos ATV através das Matrizes de Incidência

A construção das linguagens das matrizes de incidência temporizadas necessita da seguinte definição:

Definição 11 Seja \mathbf{A} uma matriz de incidência. A matriz

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

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

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

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

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

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

Exemplo 9 Do autômato ATV mostrado na Figura 3, sua representação matricial é dada por

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

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

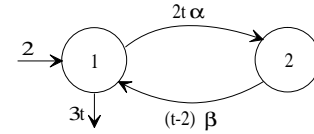


Figura 3: Autômato para exemplo da matriz de caminhos

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

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

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

Definição 12 Para uma matriz de incidência \mathbf{A} , sua linguagem é definida como

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

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

A linguagem marcada é definida como:

Definição 13 Para uma matriz de incidência \mathbf{A} , sua linguagem marcada é definida como

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

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

Exemplo 10 Seja o autômato apresentado na Figura 3. Para determinar sua linguagem, utiliza-se a matriz de incidência, a matriz de caminhos \mathbf{A}^2 encontrada no Exemplo 9, e calcula-se

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

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

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

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

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

4 EXEMPLO

Exemplo 11 Considere um suposto sistema em que há dois tipos de peças (p_1 e p_2). Cada tipo de peça necessita ter sua temperatura elevada em um valor específico, para poder ser processada. Também, cada um tipo tem um tempo específico para ser processado. Ao ser processada uma peça, esta é enviada a um buffer onde é consumida, e o sistema retorna ao estado inicial. A primeira peça a ser processada sempre é do tipo p_1 . Se a peça que está para ser processada não atingir a temperatura ideal, ela deve ser reesquentada para poder ser processada. Se após o processamento esta peça ficar defeituosa, ela é enviada a um lugar de peças defeituosas, em que é refeita e devolvida ao buffer de entrada, onde pode ser reprocessada. Este suposto sistema é apresentado na Figura 4. O ATV que modela este sistema está apresentado na Figura 5(a), em que as funções temporais dos arcos são definidas por: $f_1 = t^2 - 1$, $f_2 = 2t + 1$, $f_3 = 2 \vee 3$, $f_4 = \sqrt{t+1}$, $f_5 = 2t$ e $f_6 = \sqrt{(\frac{t}{2} + 1)}/2$. A função f_3 define uma condição aleatória, cujo valor pode

ser 2 ou 3 unidades de tempo. A função de atraso final é $f_7 = t + 2$ e $t_0 = 2$. Com essas funções, observa-se que é possível diferenciar o processo no autômato (qual tipo de peça está sendo processada), pelos tempos específicos de vida dos eventos, embora os eventos sejam os mesmos para ambos tipos de peças. Este sistema é mais complexo se for

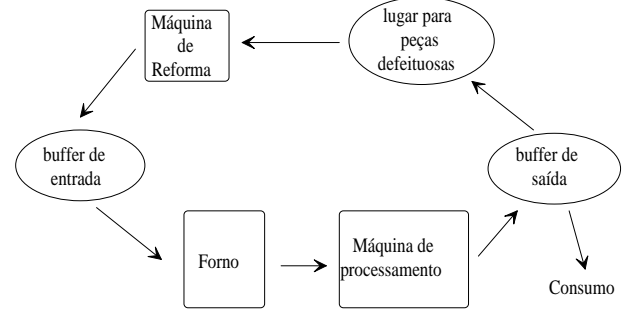


Figura 4: Sistema.

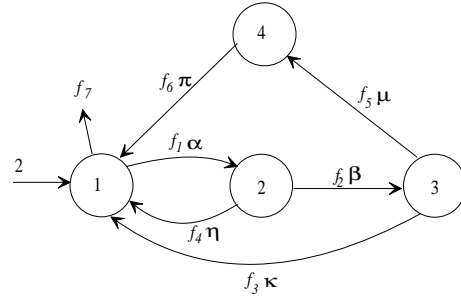


Figura 5: (a) Sistema modelado por um ATV e (b) sistema modelado por um autômato (max,+).

modelá-lo por um autômato (max,+), desde que existe a necessidade da diferenciação de alguns eventos e dos tempos de vida, para poder se reconhecer qual o processo em andamento. O número de estados, bem como o número de eventos naturalmente são maiores, podendo apresentar não determinismo (por causa da função f_3). Assim, o modelo em ATV deste sistema mostra-se bastante simplificado.

5 CONCLUSÕES

Esse artigo apresenta o autômato com temporização variável com os formalismos de representação gráfica e matricial, bem como sua dinâmica, aplicando-o à modelagem de sistemas. Com o ATV alguns modelos de sistemas que apresentam variação nos tempos de vida dos eventos podem ser construídos devido à introdução das funções de tempo. Os cálculos das linguagens são mais complexos que a construção da linguagem de um autômato (max,+) [17] devido à necessidade da separação explícita dos tempos nas matrizes de caminhos. O exemplo apresentado mostra sua eficácia na representação de sistemas com as características apresentadas.

Este autômato pode ser utilizado para modelagem e síntese de supervisores de sistemas a eventos discretos que apresentem tais características pela abordagem de Costa [17], como consequência direta da sua representação matricial.

AGRADECIMENTOS

O autor agradece ao CNPq e à UFBA pelo suporte dado a esta pesquisa.

Referências

- [1] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, USA, 1979.
- [2] 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.
- [3] A. Salomaa. *Formal Languages*. ACM Monograph Series. Academic Press, Inc. New York, 1973.
- [4] P.J.G. Ramadge and W.M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, 1989.
- [5] 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.
- [6] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, (126):183–235, 1994.
- [7] R. Alur. Timed automata. *Proc. NATO-ASI Summer School, Antalya, Turkey*, 1997.
- [8] 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.
- [9] P. Caspi E. Asarin and O. Maler. A kleene theorem for timed automata. *Proc. of LICS'97*, pages 160–171, 1997.
- [10] E. Asarin. Equations on timed languages. *Proc. of Hybrid'98*, 1998.
- [11] 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.
- [12] M.S. Lawford. *Model Reduction of Discrete Real-Time Systems*. PhD thesis, University of Toronto, 1997.
- [13] B.A. Brandin and W.M. Wonham. Supervisory control of timed discrete-event systems. *IEEE Transactions on Automatic Control*, 39(2):329–342, 1994.
- [14] S. Gaubert. Performance evaluation of (max,+) automata. *IEEE Transactions on Automatic Control*, 40(12):2014–2025, December 1995.
- [15] A. Gill. *Introduction to the Theory of Finite-State Machines*. McGraw-Hill Electronic Sciences Series. McGRAW-HILL Book Company, 1962.
- [16] E.M.M. Costa and A.M.N. Lima. Síntese de supervisores de sistemas a eventos discretos temporizados. *Anais do V Simpósio Brasileiro de Automação Inteligente (SBAI)*, 2001.
- [17] 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.
- [18] 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.
- [19] J.A. Beachy. *Abstract Algebra II*. Waveland Press, Inc., 1996.
- [20] J. Berstel and C. Reutenauer. *Rational Series and their Languages*. Springer, 1988.
- [21] I. Klimann. *Langages, Séries et Contrôle de Trajectoires*. PhD thesis, l'Université Denis Diderot - Paris 7, Juin 1999.

DADOS BIOGRÁFICOS

EDUARD MONTGOMERY MEIRA COSTA, nasceu em Esperança, PB, em 12 de Junho de 1970. Formou-se em Engenharia Elétrica pela UFPB, em 1995. Obteve os títulos de Mestre e Doutor em Ciências no domínio da Engenharia Elétrica também pela UFPB, em 1997 e 2001, respectivamente. Atualmente é bolsita DCR/CNPq na UFBA, atuando na área de Sistemas a Eventos Discretos.