SÍNTESE DE SUPERVISORES DE SISTEMAS A EVENTOS DISCRETOS UTILIZANDO DIÓIDES

EDUARD MONTGOMERY MEIRA COSTA E ANTONIO MARCUS NOGUEIRA LIMA

Departamento de Engenharia Elétrica
Universidade Federal da Paraíba
58109-970 Campina Grande, PB, Brasil, Caixa Postal 10.105
Email: {eduard,amnlima}@dee.ufpb.br

Resumo— Este trabalho utiliza a álgebra de dióides para construção de supervisores para SEDs. A utilização dessa álgebra dispensa a execução do algoritmo de ponto fixo usado na determinação da suprema sublinguagem controlável. As linguagens utilizadas para descrever e especificar o comportamento do SED são substituídas por matrizes de incidência construídas a partir dos eventos controláveis, dos eventos não controláveis e da especificação de comportamento. O supervisor é sintetizado através de operações matriciais que produzem o mesmo resultado obtido com o algoritmo clássico e que, desse modo, podem ser utilizadas como uma alternativa na síntese de supervisores de SEDs não temporizados.

Abstract— This paper utilizes the dioid algebra to synthesize DES supervisors. The use of this algebra avoids the use of the fix point algorithm used to determine the supremal sublanguage. The languages used to describe and to specify the DES behaviour are replaced by incidence matrices constructed from the controllable and uncontrollable events and the behaviour specification. The supervisor is synthesized through matricial operations that yield the same results of the classical algorithm, and can be utilized as an alternative in supervisor synthesis.

Key Words— Dioid Algebra, Discrete Event Systems, Supervisory Control

1 INTRODUÇÃO

Sistemas a Eventos Discretos (SEDs) (Ramadge e Wonham, 1982) são sistemas que apresentam uma evolução dinâmica descrita pela ocorrência de eventos que determinam sua interação com o ambiente e que alteram o estado do sistema. Os SEDs estão presentes em muitas aplicações do cotidiano, como redes de computadores, sistemas de manufatura e supervisão de tráfego. O estudo dos SEDs requer a utilização de uma representação adequada e que permita projetar um agente de controle automático, denominado de supervisor. A partir de tarefas especificadas para o sistema, o supervisor recebe informações dos eventos do sistema através de sensores, determina a ação de controle e envia comandos para os atuadores que inibem ou habilitam determinados eventos.

Dentre os paradigmas disponíveis para a modelagem de SEDs, os autômatos e as linguagens formais (Hopcroft e Ullman, 1979) têm sua ampla utilização. A formalização do problema de controle de SEDs utilizando autômatos e linguagens formais é denominada de Teoria de Controle Supervisório (TCS) (Ramadge e Wonham, 1989), que é uma forma elegante de resolver o problema de controle de SEDs: a partir do modelo do SED e de uma especificação funcional, determina um supervisor.

No caso dos SEDs temporizados, a álgebra de dióides (Gaubert, 1992) tem sido muito utilizada, principalmente para os sistemas que apresentam periodicidade e necessitam de sincronização (Cohen et al., 1985; Baccelli et al., 1992). A álgebra de dióides também é utilizada na avaliação de desempenho de SEDs (Gaubert, 1993; Gaubert, 1995), utilizando-se das séries formais (Berstel e Reutenauer, 1988; Gaubert, 1994b; Gaubert, 1994a). Devido a sua versatilidade, ela também é utilizada para determinar caminhos em SEDs, utilizando a matriz de incidência do grafo que representa o sistema (Gill, 1962; Gaubert, 1999).

Nesse trabalho é apresentada uma formulação alternativa do algoritmo de síntese do supervisor baseada na álgebra de dióides, que é restrita ao caso no qual a especificação de comportamento é um sub-autômato do modelo do SED. A utilização dessa álgebra dispensa a execução do algoritmo de ponto fixo usado na determinação da suprema sublinguagem controlável - $SupC(L)$ (Ramadge e Wonham, 1987). As linguagens utilizadas para descrever e especificar o comportamento do SED são substituídas por matrizes de incidência construídas a partir de Σ_c e de Σ_{uc}, e da especificação de comportamento E. O supervisor S é sintetizado através de operações matriciais que produzem o mesmo resultado obtido com o algoritmo da $SupC(L)$ clássico.

A estrutura desse trabalho é a seguinte: na Seção 2 é descrita a álgebra de dióides e as linguagens representadas por esta álgebra; a Seção 3 contém os conceitos que dão as bases para a formalização do trabalho; na Seção 4 é apresentado o algoritmo de síntese do supervisor; na Seção 5 são apresentados exemplos ilustrando a utilização do algoritmo e na Seção 6 são apresentadas as conclusões.

2 DIÓIDES E LINGUAGENS

Um dióide é definido como a seguir:

Definição 1 *Um dióide é um conjunto D dotado de duas operações: $\oplus$ (soma) e $\otimes$ (multiplicação), que satisfaz os seguintes axiomas:*

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

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

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

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

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

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 comutativa.

2.1 LINGUAGENS FORMAIS NOS DIÓIDES

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

Definição 2 *Seja* $(D,\oplus,\otimes)$ *um dióide com* $D{=}\mathcal{P}(\Sigma^*)$. *Nesse caso* $\oplus$ *e* $\otimes$, *são definidos como as operações de união e concatenação, respectivamente, de forma que dados* L_1, $L_2\in D$,

$$L_1\otimes L_2=\{s_1\otimes s_2|s_1\in L_1,s_2\in L_2\}\ e$$
$$L_1\oplus L_2=\{s_1\oplus s_2|s_1\in L_1,s_2\in L_2\}.$$

O elemento nulo ϵ *denota a linguagem vazia* $\varnothing$ *e o elemento identidade e denota o conjunto unitário* $\{\varepsilon\}$.

Assim, duas linguagens L_1 e L_2 formam uma nova linguagem L_3, através de seu produto, isto é $L_3{=}L_1{\otimes}L_2$, ou através de sua soma, se $L_1\neq L_2$.

Exemplo 1 *Para as linguagens* $L_1{=}\{a,b\}$ *e* $L_2{=}\ \{bbc\}$, *tem-se que* $L_1\oplus L_2{=}\{a,\ b,\ bbc\}$ *e* $L_1\otimes L_2{=}\{abbc,\ bbbc\}$.

Como definido para as linguagens formais, tem-se:

Definição 3 *A concatenação de uma palavra s, com a palavra vazia* ε, *é* $s\otimes\varepsilon{=}s\otimes e{=}e\otimes s{=}\varepsilon\otimes s{=}s$.

Definição 4 *A concatenação de uma linguagem L, com a linguagem vazia* ϵ *é* $L\otimes\epsilon{=}\epsilon\otimes L{=}\epsilon$.

Exemplo 2 *Seja* $L{=}\{\beta\lambda\}$. *Então,* $s=\beta\lambda\in L$. *Logo,* $s\otimes e{=}e\otimes s{=}\ \beta\lambda e{=}\ \beta\lambda$ *e,* $L\otimes\epsilon{=}\ \epsilon\otimes L{=}\ \epsilon$.

A operação $\oplus$ aplicada às linguagens regulares permite sua representação através de expressões regulares.

Exemplo 3 *A linguagem* $L_1=\alpha\beta^*$ *somada com a linguagem* $L_2=\alpha$, *é* $L_3=L_1\oplus L_2=\alpha\beta^*\oplus\alpha=\alpha\beta^*+\alpha=\alpha(\beta^*+e)$.

Todas as propriedades dos dióides não comutativos são válidas para $D{=}\mathcal{P}(\Sigma^*)$.

3 A SÍNTESE DO SUPERVISOR

Aqui é apresentada a maneira de sintetizar um supervisor para um SED, utilizando as matrizes de incidência do autômato que modela o SED, admitindo que o alfabeto de eventos é $\Sigma{=}\Sigma_c\cup\Sigma_{uc}$, $\Sigma_c\cap\Sigma_{uc}=\varnothing$.

3.1 MATRIZ DE INCIDÊNCIA

A matriz de incidência de um autômato, é formada pelos eventos que definem as transições do estado i para o estado j.

Definição 5 *A um autômato G com N estados associa-se uma matriz* $A_{N\times N}$, *denominada de matriz de incidência, onde*

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

e $\sigma\in\Sigma$ *representa o evento que provoca a transição do estado i para o estado j.*

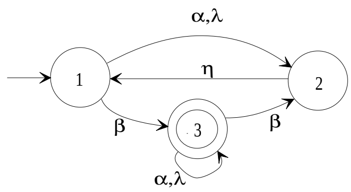

Figura 1. Autômato do Exemplo 4

Nessa representação σ é um caminho de comprimento 1, que define uma mudança do estado designado pela linha i para o estado representado pela linha j. Caso haja mais de um evento que provoque uma mudança do estado i para o estado j então a_{ij} será uma expressão regular.

Exemplo 4 *A matriz de incidência A associada ao autômato da Figura 1, é dada por*

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

O estado inicial do autômato é definido pelo estado 1, para que a primeira linha represente o estado inicial. Os estados marcados são representados por linhas da matriz A. No Exemplo 4, a linha 3 da matriz A é um estado marcado.

Para um autômato representado por uma matriz de incidência A, as condições de acessibilidade e coassecibilidade são definidas por:

Definição 6 *Um estado j de um autômato G representado por uma matriz de incidência A é acessível se* $\exists s|s=a_{1,k_1}a_{k_1,k_2}...a_{k_m,j}$ *para* $a_{1,k_1},a_{k_1,k_2},...,a_{k_m,j}\neq\epsilon$, *isto é, partindo da linha 1, existe uma seqüência s que leva à linha j.*

Definição 7 *Um estado marcado j é coacessível se* $\forall i,\exists s|s=a_{i,k_1}a_{k_1,k_2}...a_{k_m,j}$ *para* $a_{i,k_1},a_{k_1,k_2},...,\ a_{k_m,j}\neq\epsilon$.

No Exemplo 4, todos os estados são acessíveis, devido as seqüências $s{=}a_{12}$, $s{=}a_{13}a_{32}$ e, $s{=}a_{13}$. O estado 3 é coacessível, pois $a_{13}\neq\epsilon$ e $a_{33}\neq\epsilon$. Logo, A representa um autômato que é denominado de *trim*.

3.2 EVENTOS NA MATRIZ DE INCIDÊNCIA

Considerando $\Sigma{=}\Sigma_c{\cup}\Sigma_{uc}$, definem-se as matrizes A_c e A_{uc}, referentes às partes controláveis e não controláveis do autômato que modela o SED.

Definição 8 *A um autômato G definido para um alfabeto* $\Sigma{=}\Sigma_c\cup\Sigma_{uc}$, *associam-se as matrizes* A_c *e* A_{uc}, *com dimensões iguais a A, dadas por*

$$A_c{=}[(a_c)_{ij}],(a_c)_{ij}{=}\begin{cases}\sigma_c\ se\ \exists\sigma_c\ do\ estado\ i\ para\ j;\\ \epsilon\ \ \ caso\ contrário\end{cases}$$
$$A_{uc}{=}[(a_{uc})_{ij}],[a_{uc}]_{ij}{=}\begin{cases}\sigma_{uc}\ se\ \exists\sigma_{uc}\ do\ estado\ i\ para\ j;\\ \epsilon\ \ \ \ caso\ contrário,\end{cases}$$

onde $\sigma_c\in\Sigma_c,\sigma_{uc}\in\Sigma_{uc}$.

Exemplo 5 *Considerando que o autômato da Figura 1, tem* $\Sigma_c{=}\{\alpha,\eta\}$ *e* $\Sigma_{uc}{=}\{\beta,\lambda\}$, *então*

$$A_c{=}\begin{bmatrix}\epsilon\ \alpha\ \epsilon\\ \eta\ \epsilon\ \epsilon\\ \epsilon\ \epsilon\ \alpha\end{bmatrix}\quad e\quad A_{uc}{=}\begin{bmatrix}\epsilon\ \lambda\ \beta\\ \epsilon\ \epsilon\ \epsilon\\ \epsilon\ \beta\ \lambda\end{bmatrix}.$$

Observe que $A{=}A_c\oplus A_{uc}$.

3.3 A CONSTRUÇÃO DO SUPERVISOR

O algoritmo da construção da $SupC(L)$ (Ramadge e Wonham, 1987), provê uma forma de construir um supervisor para um SED. Dadas uma linguagem e uma especificação de comportamento, sua execução determina os eventos controláveis que devem ser inibidos para evitar seqüências que não pertencem à especificação. Nessa abordagem utilizando os dióides e as matrizes de incidência, a $SupC\,(L)$ é determinada através de operações matriciais. Para formalizar essas operações define-se:

Definição 9 *Dada uma especificação de comportamento E para um autômato G que é um modelo de um SED, sua matriz de incidência é definida por*

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

com dimensão $N \times N$ e, $\sigma \in \Sigma$ representa o eventos que provoca a mudança do estado i para o estado j.

A seguinte questão é formulada: *A partir da especificação E, é possível construir o supervisor ou então qual a sublinguagem controlável que pode definir um supervisor para esta especificação?*

Considerando uma dada especificação de comportamento E e a matriz de incidência dos eventos não controláveis A_{uc}, através da adição $\oplus$ é possível determinar se ela define diretamente ou não o supervisor.

Lema 1 *A adição da matriz de incidência de uma especificação de comportamento E trim com a matriz de incidência dos eventos não controláveis A_{uc}, determina a matriz de incidência do supervisor S se e só se $E \oplus A_{uc}{=}E$.*

A condição de existência para um dado supervisor S é que a matriz E contenha todos os elementos de A_{uc}. Se E é factível, o supervisor é definido.

Corolário 1 *Dada uma especificação E e a matriz de incidência A_{uc}, $S{=}E$ se e só se $E \oplus A_{uc}{=}E$.*

As condições do Lema 1 e do Corolário 1, apresentam o problema da identidade matricial. Para contornar este problema, o seguinte operador é definido.

Definição 10 *Seja A uma matriz de incidência de um autômato G. O operador $\Upsilon{:}A \rightarrow \mathbb{N}$, define o número de arcos em G, que compõem os termos da matriz A.*

Exemplo 6 *No autômato da Figura 1, $\Upsilon\,(A){=}7$, que é o número de arcos do autômato representado por A. No Exemplo 5, $\Upsilon\,(A_c){=}3$ e $\Upsilon\,(A_{uc}){=}4$.*

Assim, com este operador, tem-se:

Lema 2 *Um supervisor S para um autômato G trim é definido pela especificação de comportamento E trim se e só se $\Upsilon(E_{uc}){=}\Upsilon(A_{uc})$.*

Corolário 2 *Dada uma especificação E trim e a matriz de incidência A_{uc} de um autômato G trim, $S{=}E \oplus A_{uc}$ se e só se $\Upsilon(E_{uc}){=}\Upsilon(A_{uc})$.*

Logo, o Lema 2 sendo satisfeito, o supervisor é o autômato gerador da linguagem de E.

Exemplo 7 *Do autômato visto na Figura 2, onde $\Sigma{=}\{\alpha, \beta,\kappa,\mu\}$, $\Sigma_{uc}{=}\{\kappa\}$ e $\Sigma_c{=}\{\alpha,\beta,\mu\}$, as matrizes de incidência A e A_{uc} são dadas por*

$$A{=}\begin{bmatrix}\alpha & \beta & \mu\\ \alpha & \epsilon & \mu\\ \epsilon & \mu & \kappa\end{bmatrix} \;\; e \;\; A_{uc}{=}\begin{bmatrix}\epsilon & \epsilon & \epsilon\\ \epsilon & \epsilon & \epsilon\\ \epsilon & \epsilon & \kappa\end{bmatrix}.$$

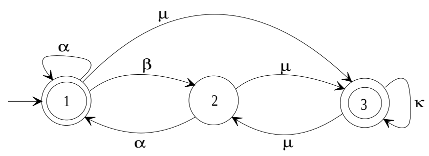

Figura 2. Autômato do Exemplo 7

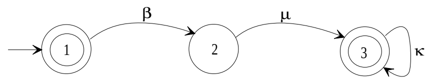

Figura 3. Supervisor do Exemplo 7

Para a especificação

$$E{=}\begin{bmatrix}\epsilon & \beta & \epsilon\\ \epsilon & \epsilon & \mu\\ \epsilon & \epsilon & \kappa\end{bmatrix},$$

$S{=}E \oplus A_{uc}{=}E$, pois $\Upsilon(E_{uc}){=}\Upsilon(A_{uc}){=}1$, que satisfaz o Lema 2. Este supervisor é visto na Figura 3.

Quando o Lema 2 não é satisfeito, é necessário encontrar a $SupC(L)$ para E. Para isto, é necessário o que se segue:

Definição 11 *Seja A uma matriz de incidência, cujos elementos a_{ij} definem caminhos de comprimento 1. Então, a matriz $A^n{=}A \otimes A \otimes ... \otimes A$ (n vezes) é uma matriz de caminhos, onde a_{ij}^n representa caminhos de comprimento n, formado por eventos controláveis e não controláveis, que levam o autômato do estado i para o estado j.*

Se não há um caminho com n eventos de i para j, $a_{ij}{=}\epsilon$. Se há, a_{ij} são palavras de comprimento n, formadas por eventos σ_{uc} e σ_c.

Exemplo 8 *Do autômato mostrado na Figura 4, tem-se*

$$A{=}\begin{bmatrix}\epsilon & \alpha & \epsilon\\ \beta & \epsilon & \mu\\ \epsilon & \beta & \alpha\end{bmatrix}.$$

A matriz de caminhos de comprimento 2 é

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

Usando a Definição 11, podem-se avaliar quais os caminhos que iniciam com um evento qualquer, seguidos só por eventos não controláveis.

Definição 12 *Seja $\Sigma{=}\Sigma_c \cup \Sigma_{uc}$, e seja um autômato G construído com símbolos de Σ. A matriz de caminhos C_{uc}^n, que define as palavras que mudam o estado do autômato de i para j, iniciados por um evento qualquer (controlável ou não controlável), e seguidos sempre de eventos não controláveis, é definida por $C_{uc}^n{=}A \otimes (A_{uc})^{n-1}$.*

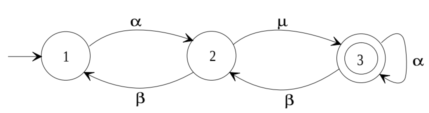

Figura 4. Autômato do Exemplo 8

Assim, os primeiros eventos das seqüências dos termos de C^n_{uc} são elementos de A (σ_{uc} ou σ_c) e os que os seguem são elementos de A_{uc}. Disso, utiliza-se a mesma idéia para definir caminhos em um dado autômato que iniciem com eventos de E, seguidos de eventos de A_{uc}, e determinar se é possível sintetizar um supervisor.

Definição 13 *Dada a matriz de incidência E, que é uma matriz de caminhos de comprimento 1, define-se a matriz de caminhos B^n_{uc}=$E \otimes (A_{uc})^{n-1}$, como sendo uma matriz de caminhos, onde o primeiro elemento de cada seqüência é um elemento de E, e os demais, de A_{uc}.*

Com esta definição, pode-se avaliar se é possível inibir alguma seqüência que leve à ocorrência de $\sigma_{uc} \notin E$. Em B^n_{uc}, um evento $\sigma_{uc} \notin E$ de $A_{uc}(i,j)$, aparece como último evento nas seqüências dos termos de uma coluna j garantindo a condição $\Sigma(\mathbf{H}(x)) \cap \Sigma_{uc}$(Ramadge e Wonham, 1987). Estes eventos são antecedidos pelos termos de $B^{n-1}_{uc}(k,i)$, para alguns k=1, ..., N. Deve-se observar que B^1_{uc}= B^1_c= $E \otimes (A_{uc})^0$= $E \otimes I$= E, com I sendo a matriz identidade. Disso, tem-se:

Teorema 1 *Dada a especificação de comportamento E e a matriz de incidência dos eventos não controláveis A_{uc}, do autômato G trim, se $\Upsilon(E_{uc}) \neq \Upsilon(A_{uc})$, então a $SupC(L)$ será determinada recursivamente por:*

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

$$B^n_{uc} = E \otimes (A_{uc})^{n-1}$$

$$S^n = [s^n_{ij}], s^n_{ij} = \begin{cases} e_{ij} & \text{se } \sigma^1\sigma^2_{uc}...\sigma^n_{uc} \in B^n_{uc} \wedge \sigma^n_{uc} \in E; \\ \epsilon & \text{se } \sigma^n_{uc} \notin E \wedge \sigma^1 \in \Sigma_c \vee e_{ij} \text{ não é alcançável} \end{cases}$$

   onde σ^n_{uc} é o n-ésimo evento da seqüência do termo B^n_{uc}, que pode ser $\sigma^n_{uc} \notin E$.

3. *Se $(n > N) \wedge (\exists\, \sigma_{uc} \notin E$ em $S^n)$, então S=$[\epsilon]$.*

Assim, se o último evento de uma seqüência em B^n_{uc} não pertençe à especificação, a inibição do primeiro evento elimina esta seqüência. A existência de outros eventos não pertencentes à especificação, é eliminada recursivamente em B^n_{uc}, para n=2, 3, ...,N, construindo o supervisor. Se algum $\sigma^n_{uc} \notin E$ não for eliminado até n=N, o supervisor não é possível para E.

4 ALGORITMO DE SÍNTESE

Algoritmo 1 *Construção da $SupC(L)$*

1. Faça S=E.
2. Se $\Upsilon(E_{uc})$=$\Upsilon(A_{uc})$ pare. Caso contrário, faça
   a) n=2, $xdif\,(k$,n-1)=i e $ydif(k$,n-1)=j (onde $A_{uc} \neq E_{uc}$), para k=1,...,M (M número de elementos diferentes entre A_{uc} e E_{uc}).
   b) Calcule B^n_{uc}.
   c) Faça para k=1 até M
      i. Procure os elementos em $B^n_{uc}(i$,$ydif(k$,n-1)), onde $\sigma^n_{uc} \notin E$ (σ^n_{uc} sendo o último elemento da seqüência).
         (1) Se $\sigma^n_{uc} \notin E$ e $\sigma^1 \in \Sigma_c$, faça $S(i$,$xdif(k$,n-1))=ϵ.
         (2) Se $\sigma^n_{uc} \notin E$ e $\sigma^1 \in \Sigma_{uc}$, faça $xdif(k$,n)=i e $ydif(k'$,n)=$ydif(k$,n-1), onde k' é o número de elementos não possíveis de controlar em B^n_{uc}.
      ii. Se $\forall i$,j=1,...,N (N número de estados), j=$ydif(k$,n-1) e $S(i$,$j)$=ϵ, faça $S(i$,$j)$=ϵ.
   d) Para k=1 até M, faça
      i. Se $S(xdif(k$,n-1),$ydif(k$,n-1))=ϵ, teste a co-acessibilidade e a acessibilidade de S. Caso contrário, faça n=n+1.
         (1) Se n<N, retorne ao passo 2.b.

Os testes de acessibilidade e de coacessibilidade são feitos pelo seguinte algoritmo:

Algoritmo 2 *Testes de acessibilidade/coacessibilidade de S*

1. Crie um vetor $vac_{1\times N}$ de estados acessíveis.
   a) Faça i e j=1 até N
      i. Para i=1, se $S(i,j) \neq \epsilon$, $vac(j)$=1.
      ii. Para i>1, se $S(i,j) \neq \epsilon$ e $vac(i)$=1, $vac(j)$=1.
   b) Repita o procedimento dos passos 1.a.i. e 1.a.ii. para i e j=N até 1.
   c) Se $\forall i$,$vac(i)$=1, S é acessível.
2. Faça i=1 até N:
   a) Se $\exists \sigma_{ik} \neq \epsilon$, onde k são as linhas que representam os estados marcados, S é coacessível.
3. Se $\forall k$, $\exists \sigma_{ik} \neq \epsilon$ e $vac(k)$=1, S é *trim.*
4. Se S é não coacessível, S não é possível para E dada.

Esse algoritmo para a síntese do supervisor, apresenta uma ordem de complexidade de $O\left(N^2\right)$, onde N é a dimensão da matriz de incidência A, ou o número de estados do autômato. Nele, se o teste de coacessibilidade falhar, não existe o supervisor para E. As matrizes $xdif$ e $ydif$ guardam os valores de i e j de E, respectivamente, para os eventos $\sigma_{uc} \notin E$. Sempre que uma seqüência em B^n_{uc} tem o último evento $\sigma_{uc} \notin E$ e o primeiro evento não controlável, estas matrizes são atualizadas com os valores de i e j de B^{n-1}_{uc}. Enquanto $n \leq N$, o algoritmo é repetido, buscando a $SupC(L)$.

5 EXEMPLOS

Exemplo 9 *No autômato visto na Figura 5, com Σ={α, β,κ,η,λ,μ} e Σ_{uc}={α,κ,λ}, a matriz de incidência A_{uc} e a*

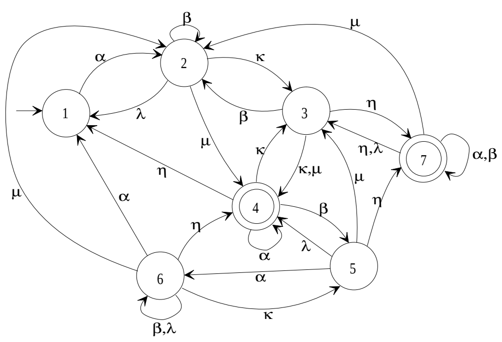

Figura 5. Autômato do Exemplo 9

especificação E (mostrada na Figura 6) são dadas por

$$A_{uc}=\begin{bmatrix} \epsilon & \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \lambda & \epsilon & \kappa & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \kappa & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \kappa & \alpha & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \lambda & \epsilon & \alpha & \epsilon \\ \alpha & \epsilon & \epsilon & \epsilon & \kappa & \lambda & \epsilon \\ \epsilon & \epsilon & \lambda & \epsilon & \epsilon & \epsilon & \alpha \end{bmatrix} \; e$$

$$E=\begin{bmatrix} \epsilon & \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \lambda & \epsilon & \kappa & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \kappa & \epsilon & \epsilon & \epsilon \\ \eta & \epsilon & \kappa & \alpha & \beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \alpha & \eta \\ \alpha & \epsilon & \epsilon & \epsilon & \epsilon & \lambda & \epsilon \\ \epsilon & \epsilon & \lambda & \epsilon & \epsilon & \epsilon & \alpha \end{bmatrix}.$$

Com essa especificação, tem-se $S^1=E$. Como $\Upsilon(E_{uc})=11$ e $\Upsilon(A_{uc})=13$, vê-se que $A_{uc}(5,4)=\lambda$ e $A_{uc}(6,5)=\kappa$ não pertencem à E. Assim, calcula-se B^2_{uc}:

$$E\otimes A_{uc}=\begin{bmatrix} \alpha\lambda & \epsilon & \alpha\kappa & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \lambda\alpha & \epsilon & \kappa\kappa & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \kappa\kappa & \kappa\alpha & \epsilon & \epsilon & \epsilon \\ \epsilon & \eta\alpha & \alpha\kappa & \kappa\kappa+\alpha\alpha+\underline{\boldsymbol{\beta\lambda}} & \epsilon & \beta\alpha & \epsilon \\ \alpha\alpha & \epsilon & \eta\lambda & \epsilon & \underline{\boldsymbol{\alpha\kappa}} & \alpha\lambda & \eta\alpha \\ \lambda\alpha & \alpha\alpha & \epsilon & \epsilon & \lambda\kappa & \lambda\lambda & \epsilon \\ \epsilon & \epsilon & \alpha\lambda & \lambda\kappa & \epsilon & \epsilon & \alpha\alpha \end{bmatrix}$$

Nessa matriz, o termo $(E\otimes A_{uc})_{4,4}$ contém a seqüência $\beta\lambda$, onde λ é o evento $A_{uc}(5,4)\neq E(5,4)$. Assim, a inibição de β em $E(4,5)$, define S^2. Observe que a inibição de β no estado 4, torna os estados 5, 6 e 7 não alcançáveis. Logo, todos os eventos nas respectivas linhas/colunas tornam-se ϵ. Assim, o supervisor, visto na Figura 7, é dado por

$$S^2=\begin{bmatrix} \epsilon & \alpha & \epsilon & \epsilon \\ \lambda & \epsilon & \kappa & \epsilon \\ \epsilon & \epsilon & \epsilon & \kappa \\ \eta & \epsilon & \kappa & \alpha \end{bmatrix}.$$

Exemplo 10 *Sejam $\Sigma=\{\alpha_1,\alpha_2,\beta\}$, $\Sigma_{uc}=\{\beta\}$, e em notação de expressões regulares (Hopcroft e Ullman, 1979): $L=\overline{(\alpha_1\beta^2+\alpha_2)\,\beta^*}$, $E=\overline{\alpha_1\beta^2}+\alpha_2\beta^*$, que são a linguagem L do autômato* trim *visto na Figura 8 e a especificação de comportamento E desejada para este autômato. Utilizando o algoritmo da $SupC(L)$ apresentado em (Ramadge e Wonham, 1987), encontra-se $K_3=\varepsilon\alpha_2+\beta^*$ como a $SupC\,(L)$.*

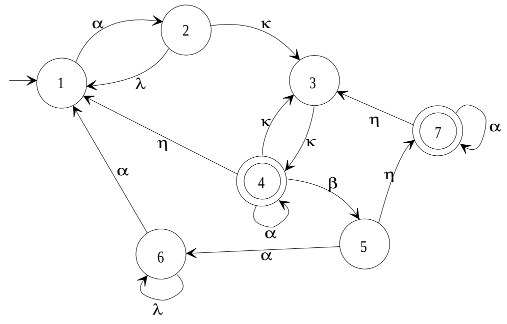

Figura 6. Especificação para o Exemplo 9

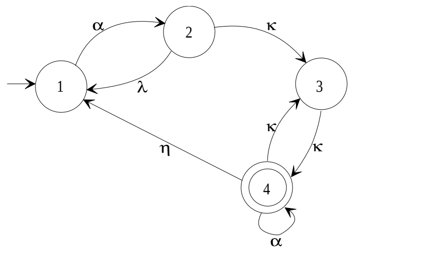

Figura 7. Supervisor do Exemplo 9

Na abordagem apresentada aqui, tem-se

$$A_{uc}=\begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \beta & \epsilon \\ \epsilon & \epsilon & \epsilon & \beta & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta \end{bmatrix} \; e \; E=\begin{bmatrix} \epsilon & \alpha_1 & \epsilon & \epsilon & \alpha_2 \\ \epsilon & \epsilon & \beta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \beta & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta \end{bmatrix}.$$

Destas matrizes, $S^1=E$. Como $\Upsilon(E_{uc})=3$ e $\Upsilon(A_{uc})=4$ vê-se que $E_{uc}\neq A_{uc}$ no termo $(4,4)$. Assim,

$$B^2_{uc}=E\otimes A_{uc}=\begin{bmatrix} \epsilon & \epsilon & \alpha_1\beta & \epsilon & \alpha_2\beta \\ \epsilon & \epsilon & \epsilon & \beta\beta & \epsilon \\ \epsilon & \epsilon & \epsilon & \underline{\boldsymbol{\beta\beta}} & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta\beta \end{bmatrix}$$

que apresenta o termo $B^2_{uc}(3,4)$ como um caminho que leva à ocorrência $\beta\notin E$. Como este termo inicia por um evento não controlável (β), $S^2=S^1$, que continua com o termo não pertencente a especificação. Então, calcula-se B^3_{uc}

$$B^3_{uc}=E\otimes(A_{uc})^2=\begin{bmatrix} \epsilon & \epsilon & \epsilon & \alpha_1\beta\beta & \alpha_2\beta\beta \\ \epsilon & \epsilon & \epsilon & \underline{\boldsymbol{\beta\beta\beta}} & \epsilon \\ \epsilon & \epsilon & \epsilon & \beta\beta\beta & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta\beta\beta \end{bmatrix}$$

que apresenta o termo $B^3_{uc}(2,4)$ definindo um caminho que leva à ocorrência de $\beta\notin E$. Contudo, como $\sigma^1_{2,4}=\beta$, B^3_{uc} não é conclusiva e $S^3=S^2$. Calculando B^4_{uc}, tem-se

$$B^4_{uc}=E\otimes(A_{uc})^3=\begin{bmatrix} \epsilon & \epsilon & \epsilon & \underline{\boldsymbol{\alpha_1\beta\beta\beta}} & \alpha_2\beta\beta\beta \\ \epsilon & \epsilon & \epsilon & \beta\beta\beta\beta & \epsilon \\ \epsilon & \epsilon & \epsilon & \beta\beta\beta\beta & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta\beta\beta\beta \end{bmatrix}$$

que tem o termo $B^4_{uc}(1,4)$ iniciando com o evento controlável α_1, seguido de β por três vezes, onde $\sigma^4_{1,4}=\beta\neq E(4,4)$, isto é $\sigma^4_{1,4}\notin E$. Com isso, a inibição de α_1 no termo

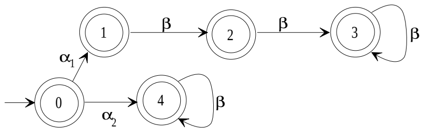

Figura 8. Autômato do Exemplo 10

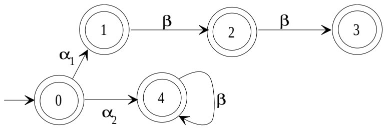

Figura 9. Especificação para o Exemplo 10

$E(1,2)$ que torna os estado 2 e 3 não acessíveis define S^4 que é o supervisor (Figura 10) dado por

$$S^4 = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \alpha_2 \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \beta \end{bmatrix}.$$

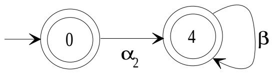

Figura 10. Supervisor do Exemplo 10

6 CONCLUSÃO

Esse trabalho mostra que é possível utilizar a álgebra de dióides na solução do problema de controle supervisório de SEDs não temporizados. A formulação do problema de controle supervisório no contexto da álgebra de dióides requer que tanto o autômato que modela o SED quanto a especificação de comportamento sejam representados através de matrizes de incidência constituídas de elementos definidos no alfabeto de eventos do SED. A solução obtida fornece resultados semelhantes ao algoritmo da $SupC(L)$ desenvolvido em (Ramadge e Wonham, 1987). Logo, o algoritmo aqui apresentado pode ser utilizado como uma alternativa à construção de supervisores para SEDs. Esse algoritmo apresenta uma ordem de complexidade de $O\left(N^2\right)$, sendo N-1 o máximo número de iterações, onde N é a dimensão da matriz de incidência. Embora nesse trabalho a síntese do supervisor esteja limitada ao tratamento de SEDs em que a especificação de comportamento é definida por um sub-autômato do modelo do sistema, uma abordagem mais ampla está sendo fundamentada, a qual inclui testes de bloqueio, controlabilidade e fechamento, além de ser aplicado à especificações não sub-autômatos. Também, os resultados obtidos com esta abordagem abrem a possibilidade de desenvolver uma ferramenta para a construção de supervisores para SED temporizados.

Referências

Baccelli, F., Cohen, G., Olsder, G. e Quadrat, J. (1992). *Synchronisation and Linearity. An Algebra for Discrete Event Systems*, John Wiley Sons.

Berstel, J. e Reutenauer, C. (1988). *Rational Series and their Languages*, Springer.

Cohen, G., Dubois, D., Quadrat, J. e Viot, M. (1985). 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.

Gaubert, S. (1992). *Théorie des Systèmes Linéaires dans les Dioïdes*, PhD thesis, École Nationale Supérieure des Mines de Paris.

Gaubert, S. (1993). Performance evaluation of timed automata, *Technical report*, Institut National de Recherche en Informatique et en Automatique.

Gaubert, S. (1994a). On rational series in one variable over certain dioids, *Technical report*, Institut National de Recherche en Informatique et en Automatique.

Gaubert, S. (1994b). 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* .

Gaubert, S. (1995). Performance evaluation of (max,+) automata, *IEEE Transactions on Automatic Control* **40**(12): 2014–2025.

Gaubert, S. (1999). Introduction aux systèmes dynamiques à Événements discrets. Notes de Cours.

Gill, A. (1962). *Introduction to the Theory of Finite-State Machines*, McGraw-Hill Electronic Sciences Series, McGRAW-HILL Book Company.

Hopcroft, J. e Ullman, J. (1979). *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, USA.

Ramadge, P. e Wonham, W. (1982). Supervision of discrete event processes, *Procedings of 21st Conference on Decision and Control* pp. 1228–1229.

Ramadge, P. e Wonham, W. (1987). On the supremal controllable sublanguage of a given language, *SIAM Journal of Control and Optimization* **25**(3): 637–659.

Ramadge, P. e Wonham, W. (1989). The control of discrete event systems, *Proceedings of the IEEE* **77**(1): 81–98.