

# CONTROLE SUPERVISÓRIO HIERÁRQUICO DE SISTEMAS A EVENTOS DISCRETOS: UMA ABORDAGEM BASEADA NA AGREGAÇÃO DE ESTADOS

CÉSAR R. C. TORRICO, JOSÉ E. R. CURY

*Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina  
Caixa Postal 470, Campus Trindade, CTC, 88.040-000 Fax: (55)(48)331-9934 Florianópolis, SC, Brasil  
E-mails: {torrico,cury}@lcmi.ufsc.br*

**Resumo**— Este artigo apresenta um novo modelo de controle supervisório hierárquico por agregação de estados baseado em estruturas de controle avançadas. O modelo proposto consiste de dois níveis de hierarquia, um baixo nível representado pelo modelo clássico de Ramadge-Wonham e um alto nível obtido pela agregação dos seus estados. Neste modelo, os eventos do alto nível são um subconjunto dos eventos de baixo nível, mas para síntese do controlador este nível será dotado de estruturas de controle avançadas introduzidas num trabalho anterior pelos presentes autores.

**Abstract**— This paper presents a new hierarchical control model for discrete event systems based on state aggregation and advanced control structures. The proposed model consists of two levels, a low-level represented by the classical model of Ramadge-Wonham, and a high-level, obtained by state aggregation. In this model, the high-level events are a subset of the low-level events, but for controller synthesis, this level will be endowed by advanced control structures as previously introduced by the authors.

**Key Words**— Discrete Event Systems, Supervisory Control, Hierarchical Control

## 1 Introdução

Em muitas áreas de aplicação, a complexidade dos processos incrementaram enormemente durante as últimas décadas. Isto principalmente porque a integração entre as componentes dos processos cresce constantemente para permitir que os recursos sejam usados com mais eficiência. O controle hierárquico é uma opção para manipular problemas complexos através da decomposição do problema em sub-problemas menores para depois montar as suas soluções numa estrutura hierárquica.

O formalismo sobre controle hierárquico de Sistemas a Eventos Discretos (SEDs) foi introduzido por Zhong e Wonham (1990) dentro do contexto de Ramadge e Wonham (1989). Nesta abordagem, o problema de controle hierárquico é formulado para especificações prefixo fechadas e refere-se a projetar um controlador no nível gerencial tal que o comportamento esperado neste nível seja igual ao comportamento obtido pela aplicação do controle hierárquico (*consistência hierárquica*). Em Wong e Wonham (1996) propõe-se uma versão generalizada de Zhong e Wonham (1990). Introduce-se a formulação do problema para linguagens com marcação e se estabelece a condição de *consistência hierárquica forte*. Nessas abordagens verificou-se que as mesmas apresentam condições conservadoras para garantir consistência hierárquica forte, além do que o canal de informação em geral tem que ser refinado várias vezes até se conseguir consistência hierárquica forte. Por outro lado, Caines e Hubbard (1998) desenvolveram uma abordagem para controle hierárquico por agregação de estados. Esta também apresenta uma condição de consistência hierárquica forte, porém não apresenta uma forma

sistemática de obtenção desta condição, além de só permitir especificações expressas em termos de estados proibidos.

Neste trabalho será introduzido um novo formalismo para o controle hierárquico de sistemas a eventos discretos usando dois modelos de estruturas de controle. Serão considerados dois níveis de hierarquia: para o baixo nível recorre-se ao modelo tradicional de Ramadge e Wonham; no alto nível será utilizado o modelo generalizado proposto por Cury et al. (2001), baseado em estruturas de controle avançadas e marcação dinâmica associada. O objetivo é propor um modelo para controle hierárquico baseado na agregação de estados, tal que a estrutura hierárquica seja obtida de uma forma simples e direta, garantindo consistência hierárquica forte com mínimo de refinamento.

## 2 O problema de controle hierárquico

Dada uma planta  $G$  definida sobre um alfabeto  $\Sigma$ , e dado um alfabeto de eventos relevantes  $\Sigma^A \subset \Sigma$  para possíveis especificações, o problema é obter um modelo agregado para o alto nível definido sobre  $\Sigma^A$ , tal que: *i*) para qualquer especificação realizável no alto nível, a linguagem sintetizada com o modelo agregado seja igual à imagem da implementação por meio do baixo nível; *ii*) que toda linguagem realizável no baixo nível tenha uma imagem realizável no alto nível.

Na estrutura hierárquica, o baixo nível consiste de uma planta  $G$  e um supervisor  $S$  definidos sobre um alfabeto  $\Sigma$ , e o alto nível (nível agregado) é formado por uma planta  $G^A$  e um supervisor  $S^A$  definidos sobre um alfabeto  $\Sigma^A \subset \Sigma$ . Estes são acoplados como mostrado na Fig. 1.

Na Fig. 1,  $G$  é a planta a ser controlada no

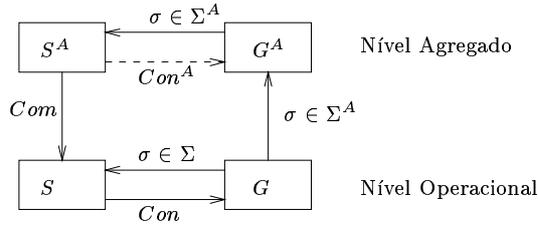


Figura 1. Arquitetura hierárquica.

mundo real por  $S$ , o operador, enquanto que  $G^A$  é uma abstração de  $G$  obtida pela agregação de seus estados que será controlada por um supervisor  $S^A$ , o gerente. A planta  $G$  informa ao nível agregado somente alguns eventos considerados relevantes ( $\sigma \in \Sigma^A$ ), atualizando o modelo  $G^A$ , mas também informa ao operador  $S$  a ocorrência de qualquer evento  $\sigma \in \Sigma$ . A cada ocorrência de um evento  $\sigma \in \Sigma^A$ , o gerente  $S^A$  envia um comando  $Com$  a ser implementado pelo operador  $S$ . O operador  $S$  recebe duas informações - o comando  $Com$  do nível agregado e a ocorrência de eventos  $\sigma \in \Sigma$  gerados pela planta  $G$  - e com estes dados envia uma entrada de controle  $Con$  para a planta  $G$ . A entrada de controle  $Con^A$ , enviada pelo gerente  $S^A$  para  $G^A$ , resulta ser uma entrada de controle virtual, uma vez que o comportamento de  $G^A$  é determinado totalmente pelo comportamento de  $G$ .

### 3 Agregação de estados

No nível operacional, o autômato da planta é dado por uma quintupla,  $G = (\Sigma, X, \delta, q_0, Q_m)$ , onde  $\Sigma$  é o conjunto finito de eventos, consistindo de eventos não-controláveis ( $\Sigma_u$ ) e eventos controláveis ( $\Sigma_c$ ), i.e.  $\Sigma = \Sigma_u \cup \Sigma_c$ ,  $X$  é o conjunto finito de estados,  $\delta : X \times \Sigma \rightarrow X$  é a função de transição de estados (em geral uma função parcialmente definida),  $q_0 \in X$  é o estado inicial e  $Q_m \subseteq X$  é o conjunto de estados marcados.

Uma partição do conjunto de estados  $X$ , denominada partição  $\pi$ , é feita de tal forma que só eventos relevantes sejam mostrados, ou seja, dois estados pertencem a um mesmo bloco se estão ligados por alguma transição não relevante. A partição  $\pi$  é uma coleção de subconjuntos  $X_k$  (chamados blocos- $\pi$ ) obtidos através da partição do conjunto de estados  $X$ , sendo formalmente denotada por  $\pi = \{X_1, X_2, \dots, X_N\}$  com  $\bigcup X_i = X$ ,  $X_i \neq \emptyset$  e  $X_i \cap X_j = \emptyset$  para  $i \neq j$ . Assim, apresenta-se um modelo abstrato definido sobre um conjunto de eventos relevantes.

**Definição 3.1 Autômato agregado- $\pi$ :** Dado um autômato  $G$ , um conjunto de eventos relevantes  $\Sigma^A \subset \Sigma$ , e uma partição  $\pi = \{X_i : x_j, x_k \in X_i \leftrightarrow x_j, x_k \in X \wedge \exists s \in (\Sigma - \Sigma^A)^*, \delta(x_i, s) = x_j\}$ , define-se o autômato agregado- $\pi$ ,

$$G^A = (\Sigma^A, \pi, \delta^A, q_0^A)$$

onde  $\Sigma^A$  é o conjunto de eventos,  $\pi$  é o conjunto finito de estados obtido após a partição de  $X$ ,  $\delta^A : \pi \times \Sigma^A \rightarrow \pi$  é a função de transição parcial

de estados tal que  $\delta^A(X_i, \sigma) = X_j$  se  $(\exists x \in X_i, \exists y \in X_j : \delta(x, \sigma) = y)$ , e  $q_0^A \in \pi$  é o estado inicial, tal que  $q_0 \in q_0^A$ .

Para obter um autômato agregado com boas propriedades, previamente é necessário tratar o problema de não determinismo que aparece pelo fato da agregação. Um possível não determinismo que surge na agregação será eliminado simplesmente renomeando os eventos  $\alpha \in \Sigma$  que geram não determinismo, em eventos  $\alpha', \alpha'', \dots$  que irão aumentar o alfabeto  $\Sigma$ . Considera-se também como não determinismo no autômato agregado, duas transições que saem com um mesmo evento de um mesmo bloco e chegam a um mesmo bloco, porém por diferentes entradas. Isto porque nesse caso, as continuções não necessariamente serão iguais. Portanto uma especificação tem que ser feita sobre o alfabeto original substituindo o evento  $\alpha$  que origina o não determinismo, por todas as instâncias renomeadas ( $\alpha' + \alpha''$ ). Este refinamento permite ao modelo eliminar qualquer ambigüidade na informação que flui pelo canal de informação para o alto nível. Isto é possível uma vez que o operador de baixo nível, conhecendo o estado atual da planta, sabe em uma situação de não determinismo no nível agregado, distinguir qual instância do evento ocorreu.

Dado um autômato  $G$  e o autômato agregado- $\pi$  correspondente ( $G^A$ ), definimos o mapeamento canônico  $\Theta : L(G) \rightarrow L(G^A)$ , como  $\Theta(\epsilon) = \epsilon$  e

$$\Theta(s\sigma) = \begin{cases} \Theta(s)\sigma & \text{se } \sigma \in \Sigma^A \\ \Theta(s) & \text{caso contrário} \end{cases}$$

para  $s \in \Sigma^*$  e  $\sigma \in \Sigma$ .

O mapeamento canônico é estendido naturalmente para linguagens como  $\Theta : 2^{L(G)} \rightarrow 2^{L(G^A)}$ ,  $\Theta(K) = \{\Theta(s) : s \in K\}$

Seja  $L^A = \Theta(L(G))$  o mapeamento para o nível agregado de uma linguagem  $L(G)$  e seja  $L(G^A)$  a linguagem gerada pelo autômato agregado- $\pi$ . Note-se que  $L^A$  é diferente de  $L(G^A)$  uma vez que algumas cadeias de  $L(G^A)$  são impossíveis de serem obtidas através do mapeamento das cadeias de baixo nível. Pode-se mostrar que  $L^A \subseteq L(G^A)$ .

### 4 Obtenção do modelo agregado

Nesta seção o modelo agregado será completado com a introdução de uma estrutura de controle com as características do modelo apresentado em (Cury et al., 2001), ou seja este modelo apresentará estruturas de controle avançados com atributos de marcação associados.

#### 4.1 Preliminares

O conjunto de estados de entrada  $I(X_i)$  de um elemento  $X_i$  de uma partição  $\pi$ , é o conjunto de estados em  $X_i$  que, ou é o estado inicial ou são estados diretamente acessíveis com algum evento relevante, isto é:

$$x \in I(X_i) \leftrightarrow [x = q_0] \vee [x \in X_i, \mathbf{e} \exists x' \in X \exists \sigma \in \Sigma^A, \delta(x', \sigma) = x] \quad (1)$$

Cada bloco da partição resulta ser uma parte do sistema que pode ser representada por um outro autômato. Para cada bloco  $X_i$ , elemento de uma partição  $\pi$ , é definido um *autômato*  $H_i$  como,

$$H_i = ((\Sigma - \Sigma^A), X_i, I(X_i), \delta_i, Q_{im}) \quad (2)$$

onde  $(\Sigma - \Sigma^A)$  é o alfabeto,  $X_i$  é o conjunto de estados,  $I(X_i)$  é o conjunto de estados iniciais,  $\delta_i : X_i \times (\Sigma - \Sigma^A) \rightarrow X_i$  é a função de transição de estados tal que  $(\delta_i(x, \sigma) = \delta(x, \sigma)$  se  $x \in X_i, \sigma \in (\Sigma - \Sigma^A)$ ), e  $Q_{im}$  é o conjunto de estados marcados tal que  $Q_{im} = X_i \cap Q_m$ .

Dado um autômato  $H_i$ , para cada entrada  $x_j \in I(X_i)$  pode ser definido um *subsistema* de  $H_i$  como um autômato,

$$H_{ij} = ((\Sigma - \Sigma^A), X_{ij}, x_j, \delta_{ij}, Q_{ijm}) \quad (3)$$

onde,  $(\Sigma - \Sigma^A)$  é o alfabeto,  $x_j$  é o estado inicial,  $X_{ij}$  é o conjunto de estados tal que  $X_{ij} = \{x \in X_i : x = \hat{\delta}_i(x_j, u), u \in (\Sigma - \Sigma^A)^*\}$ ,  $\delta_{ij}$  é a função de transição de estados tal que  $(\delta_{ij}(x, \sigma) = \delta_i(x, \sigma)$  se  $x \in X_{ij}, \sigma \in (\Sigma - \Sigma^A)$ ), e  $Q_{ijm}$  é o conjunto de estados marcados tal que  $Q_{ijm} = Q_{im} \cap X_{ij}$ .

Também define-se um *autômato aumentado*  $H_i^+$  adicionando um estado marcado  $x^+$  para receber as transições correspondentes ao conjunto ativo de eventos de  $X_i$  em  $G^A$ , i.e.,

$$H_i^+ = (\Sigma, X_i^+, I(X_i), \delta_i^+, Q_{im}^+) \quad (4)$$

onde,  $\Sigma$  é o alfabeto,  $X_i^+$  é o conjunto de estados tal que  $X_i^+ = X_i \cup \{x^+\}$ ,  $I(X_i)$  é o conjunto de estados iniciais,  $\delta_i^+$  é a função de transição de estados tal que  $(\delta_i^+(x, \sigma) = \delta_i(x, \sigma)$  se  $x \in X_i, \sigma \in (\Sigma - \Sigma^A)$ ) e  $(\delta_i^+(x, \sigma) = x^+$  se  $x \in X_i, \sigma \in \Sigma^A, \delta(x, \sigma)!$ ), e  $Q_{im}^+$  é o conjunto de estados marcados tal que  $Q_{im}^+ = Q_{im} \cup \{x^+\}$ .

Analogamente, para cada estado de entrada  $x_j \in I(X_i)$  de  $H_i^+$  define-se os *subsistemas para*  $H_i^+$

$$H_{ij}^+ = (\Sigma, X_{ij}^+, \delta_{ij}^+, x_j, Q_{ijm}^+) \quad (5)$$

onde,  $\Sigma$  é o alfabeto,  $x_j$  é o estado inicial tal que  $x_j \in I(X_i)$ ,  $X_{ij}^+$  é o conjunto de estados tal que  $X_{ij}^+ = \{x \in X_i^+ : x = \hat{\delta}_i^+(x_j, u), u \in \Sigma^*\}$ ,  $\delta_{ij}^+$  é a função de transição parcial tal que  $\delta_{ij}^+ = \hat{\delta}_i^+ / X_{ij}^+$  e  $Q_{ijm}^+$  é o conjunto de estados marcados tal que  $Q_{ijm}^+ = Q_{im}^+ \cap X_{ij}^+$ .

Seja  $H_{ij}$  e  $H_{ij}^+$  sub-autômatos de  $H_i$  e  $H_i^+$  respectivamente, define-se o *conjunto de sub-autômatos entre*  $H_{ij}$  e  $H_{ij}^+$ ,

$$\mathcal{S}_{ij} = \{ H_s : [L(H_{ij}) \subseteq L(H_s) \subseteq L(H_{ij}^+)] \wedge [L_m(H_s) = L_m(H_{ij}^+) \cap L(H_s)] \} \quad (6)$$

Para ilustrar estas definições, consideram-se os autômatos  $H_{ij}$  e  $H_{ij}^+$  mostrados na Fig. 2. Note que o conjunto de eventos relevantes é  $\{a, b, c\}$ .

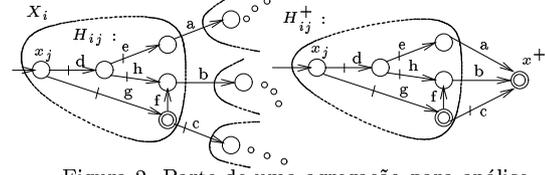


Figura 2. Parte de uma agregação para análise.

Na Fig. 3, na primeira linha de cada célula mostram-se todos os elementos de  $\mathcal{S}_{ij} = \{H_{s1}, H_{s2}, \dots, H_{s8}\}$ , considerados entre  $H_{ij}$  e  $H_{ij}^+$  da Fig. 2.

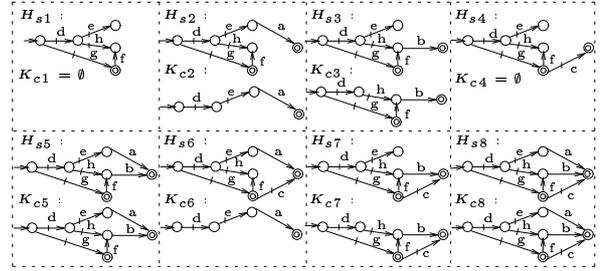


Figura 3. Conjunto de subsistemas entre  $H_{ij}$  e  $H_{ij}^+$  e suas respectivas máxima linguagem controlável.

Em cada autômato  $H_s \in \mathcal{S}_{ij}$  as transições para  $x^+ \in X_{ij}^+$  corresponde a um subconjunto do conjunto de eventos ativos de  $X_i$  entrando por  $x_j \in I(X_i)$ .

A partir de cada elemento de  $\mathcal{S}_{ij}$ , obtém-se um *conjunto de linguagens supremo controláveis em relação a*  $L(H_{ij}^+)$ ,

$$\mathcal{C}_{ij} = \{K_c \subset \Sigma^* : K_c \neq \emptyset, K_c = \sup \mathcal{C}(L_m(H_s), L(H_{ij}^+)), H_s \in \mathcal{S}_{ij}\} \quad (7)$$

Ainda continuando com a Fig. 3 também mostra-se na segunda linha de cada célula os elementos de  $\mathcal{C}_{ij}$ , sendo que neste exemplo  $\mathcal{C}_{ij} = \{K_{c2}, K_{c3}, K_{c5}, K_{c6}, K_{c7}, K_{c8}\}$ .

Também obtém-se  $\mathcal{D}_{ij}$ , retirando-se as cadeias que passam por  $L_m(H_{ij})$  de cada elemento de  $\mathcal{C}_{ij}$ . Em outras palavras, tiram-se os estados marcados dos autômatos que representam os elementos de  $\mathcal{C}_{ij}$  diferentes do estado aumentado.

$$\mathcal{D}_{ij} = \{K_d \subset \Sigma^* : K_d = K_c - \sup \mathcal{F}[(K_c - L_m(H_{ij})), L_m(H_{ij}^+)], K_c \in \mathcal{C}_{ij}\} \quad (8)$$

Nesta equação  $\sup \mathcal{F}$  representa a máxima linguagem  $L_m(H_{ij}^+)$ -fechada (Ramadge e Wonham, 1989).

Na Fig. 4 mostram-se os elementos de  $\mathcal{D}_{ij} = \{K_{d1}, K_{d2}, K_{d3}\}$  obtidos a partir de  $\mathcal{C}_{ij}$  da Fig. 3.

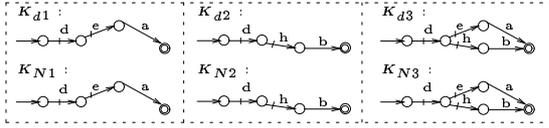


Figura 4. Conjunto de subsistemas de  $\mathcal{C}_{ij}$  sem passar por  $L_m(H_{ij})$  e suas respectivas máximas linguagens controláveis.

A partir de cada elemento de  $\mathcal{D}_{ij}$ , obtém-se um conjunto de *linguagens supremo controláveis em relação a*  $L(H_{ij}^+)$ ,

$$\mathcal{C}_{Nij} = \{K_N \subset \Sigma^* : K_N \neq \emptyset, K_N = \sup \mathcal{C}(K_d, L(H_{ij}^+)), K_d \in \mathcal{D}_{ij}\} \quad (9)$$

Na segunda linha das células da Fig. 4 mostra-se os elementos de  $\mathcal{C}_{Nij}$ , sendo  $\mathcal{C}_{Nij} = \{K_{N1}, K_{N2}, K_{N3}\}$ .

Finalmente o conjunto total de elementos supremo controlável será obtido como

$$\mathcal{C}_{Tij} = \mathcal{C}_{ij} \cup \mathcal{C}_{Nij} \quad (10)$$

O conjunto total de elementos supremo controláveis,  $\mathcal{C}_{Tij}$ , representa as máximas linguagens controláveis dos elementos de  $\mathcal{S}_{ij}$  em relação a  $L(H_{ij}^+)$ . Pode se dar o caso que dois elementos em  $\mathcal{C}_{Tij}$  possam ter as mesmas transições em  $\Sigma^A$  para alcançar o estado  $x^+$ , mas só um dos elementos terá a possibilidade de passar por algum estado marcado antes de atingir  $x^+$  e o outro necessariamente pertencerá a  $\mathcal{C}_{Nij}$ .

#### 4.2 Estrutura de controle para o nível agregado

Dado  $\mathcal{C}_{Tij}$  atribui-se ao nível agregado um conjunto de pares  $\Gamma_{ij} \in 2^{\Sigma^A} \times \{M, N\}$  definido como

$$\Gamma_{ij} = \{(\gamma, \#) : \gamma = [\Theta(K_c) - \epsilon], K_c \in \mathcal{C}_{Tij}, \text{ e } \# = M \text{ se } (\epsilon \in \Theta(K_c)), \text{ senão } \# = N\} \quad (11)$$

A estrutura de controle do alto nível é definida como um mapeamento  $\Gamma : (\pi, X) \rightarrow 2^{2^{\Sigma^A} \times \{M, N\}}$ , a qual associa a cada bloco  $X_i \in \pi$  entrando por  $x_j \in I(X_i)$  um conjunto de padrões de controle  $\Gamma_{ij} \subseteq 2^{\Sigma^A} \times \{M, N\}$ .  $(\gamma, \#) \in \Gamma_{ij}$  é um padrão de controle válido em  $X_i$  entrando por  $x_j$  com,

1.  $\gamma \subset \Sigma$  sendo um conjunto de eventos habilitados após  $X_i$ ;
2.  $\# = M$  sendo um atributo de marcação significando que a cadeia atual executada em  $L(G^A)$  é considerada uma tarefa do sistema; e
3.  $\# = N$  significando que a cadeia atual executada em  $L(G^A)$  não é considerada uma tarefa do sistema.

**Teorema 4.1**  $\Gamma$  representa uma estrutura de controle como definido em (Cury et al., 2001), isto é, cada  $\Gamma_{ij} \in \Gamma$ , satisfaz

$$1. (\gamma_1, N), (\gamma_2, N) \in \mathcal{P}_{ij} \longrightarrow (\gamma_1 \cup \gamma_2, N) \in \mathcal{P}_{ij}$$

$$2. (\gamma_1, M), (\gamma_2, \#) \in \mathcal{P}_{ij} \longrightarrow (\gamma_1 \cup \gamma_2, M) \in \mathcal{P}_{ij}, \# = M, N^a$$

Como exemplo, considerando as Figuras 3 e 4, e lembrando que os eventos relevantes são  $\{a, b, c\}$ , tem-se

$$\Gamma_{ij} = \{(\{a\}; N), (\{b\}; N), (\{b\}; M), (\{a, b\}; N), (\{a, b\}; M), (\{b, c\}; M), (\{a, b, c\}; M)\}.$$

O supervisor para alto nível,  $f^A$  é definido como um mapeamento,  $f^A : L(G^A) \rightarrow 2^\Sigma \times \{M, N\}$ . Os comportamentos gerado e marcado em malha fechada são dados como em (Cury et al., 2001).

#### 4.3 Controle supervisorio do nível agregado

A estrutura de controle  $\Gamma$  discutidas no Teorema 4.1 contem elementos  $\Gamma_{ij}$ , as quais são dependentes do bloco  $X_i$  e do estado de entrada  $x_j \in I(X_i)$ . Uma vez que a informação do estado de entrada não poderá ser vista no autômato do nível agregado, propõe-se a representação de um autômato agregado- $\pi$  com estruturas de controle estado dependente, denotado por  $(G_s^A)$ .

Para poder representar o autômato agregado- $\pi$  com estruturas de controle estado dependente, será necessário refinar este, fazendo um “split” de cada bloco para cada estado de entrada que tenha um conjunto de estruturas de controle diferente, garantindo desta forma uma única estrutura para cada bloco. O número de estados do autômato resultante  $G_s^A$ , no pior caso será igual ao somatório de todos os estados de entrada de todos os blocos.

Segundo a propor uma solução para o problema de controle supervisorio do alto nível é introduzido o conceito de  $\Gamma$ -compatibilidade (Cury et al., 2001) como segue.

**Definição 4.1  $\Gamma$ -compatibilidade:** A linguagem  $K \subseteq L(G_s^A)$  é  $\Gamma$ -compatível em relação a  $L(G_s^A)$  se, e somente se,  $K = \emptyset$  ou;

1.  $(\forall s \in K)(\exists(\gamma, M) \in \Gamma(X_i)) : \gamma \cap \Sigma_{L(G_s^A)}(s) = \Sigma_K(s), e$
2.  $(\forall s \in (\overline{K} - K))(\exists(\gamma, N) \in \Gamma(X_i)) : \gamma \cap \Sigma_{L(G_s^A)}(X_i) = \Sigma_K(s)$

onde  $\delta^A(q_0^A, s) = X_i$  e  $\Sigma_L(s)$  representa o conjunto ativo de eventos após  $s$ .

Em (Cury et al., 2001) mostra-se que  $\Gamma$ -compatibilidade é uma condição necessária e suficiente para existência de um supervisor não bloqueante.

## 5 Resultados Principais

A seguir são definidos supervisores de baixo e alto nível e mostra-se como estes estão relacionados.

Por construção das estruturas de controle do nível agregado, para cada entrada de controle

<sup>a</sup>As provas serão omitidas por falta de espaço

$(\gamma, \#) \in \Gamma_{ij}$ , existe no baixo nível uma linguagem  $K_c \subset L(H_{ij}^+)$  tal que  $K_c \in \mathcal{C}_{Tij}$  e  $\Theta(K_c) = \gamma$ . Observe que  $K_c$  é controlável em relação a  $L(H_{ij}^+)$  por ser elemento de  $\mathcal{C}_{Tij}$ . Define-se o supervisor no baixo nível o qual realiza  $K_c$  como o mapeamento,

$$f_{ij}^{K_c} : K_c \longrightarrow 2^\Sigma$$

observe também que para cada elemento  $K_c$  de  $\mathcal{C}_{Tij}$  tem-se um supervisor local  $f_{ij}^{K_c}$ .

Para uma linguagem de especificação no nível agregado,  $K^A \subset L(G^A)$ , com  $K^A$  sendo  $\Gamma$ -compatível, define-se o supervisor que realiza  $K^A$  como o mapeamento,

$$f^A : L(G^A) \longrightarrow 2^{2^\Sigma \times \{M, N\}}$$

o algoritmo a seguir traduz a implementação de  $f^A$  para um supervisor de baixo nível  $f : L(G) \longrightarrow 2^\Sigma$ .

**Algoritmo 5.1** *Obtenção do supervisor de baixo nível.*

**Entrada:**  $f^A$

1. Para  $s \in L(G)$ , tal que  $\Theta(s) = \epsilon$ ,  
 $f(s) = f_{0,0}^{K_c}(s)$ , onde  $K_c$  é tal que  $(\Theta(K_c) - \epsilon) = \gamma$  e  $\left[ (\epsilon \in \Theta(K_c) \text{ se } f^A(\epsilon) = (\gamma, M)) \text{ ou } (\epsilon \notin \Theta(K_c) \text{ se } f^A(\epsilon) = (\gamma, N)) \right]$ .
2. Para  $s \in L(G)$ , com  $s = s' \alpha s''$ ,  $s' \in \Sigma^*$ ,  $\alpha \in \Sigma^A$ ,  $s'' \in (\Sigma - \Sigma^A)^*$ , tal que  $\Theta(s) \in \overline{K^A}$ , e  $\delta(x_0, s' \alpha) = x_j \in I(X_i)$ ,  
 $f(s) = f_{ij}^{K_c}(s'')$ , onde  $K_c$  é tal que  $(\Theta(K_c) - \epsilon) = \gamma$  e  $\left[ (\epsilon \in \Theta(K_c) \text{ se } f^A(\theta(s)) = (\gamma, M)) \text{ ou } (\epsilon \notin \Theta(K_c) \text{ se } f^A(\theta(s)) = (\gamma, N)) \right]$ .

**Saída:**  $f$

Observe que a linguagem de baixo nível resultante da ação de  $f$  sobre  $G$  denotada  $L(f/G)$ , pode ser obtida pela montagem das linguagens realizadas pelos supervisores locais  $f_{ij}^{K_c}$  envolvidos. A linguagem marcada é definida como  $L_m(f/G) = L(f/G) \cap L_m(G)$ .

O Teorema a seguir apresenta as condições de consistência hierárquica.

**Teorema 5.1** *Dada uma linguagem realizada pelo supervisor do nível agregado, isto é,  $L_m(f^A/G^A) = K^A$ , e a correspondente realização no baixo nível,  $L_m(f/G) = K$ , pode-se afirmar que  $\Theta(K) = K^A$ .*

O teorema anterior garante que para qualquer linguagem  $K^A$  realizável no alto nível, existe no baixo nível uma linguagem controlável cuja imagem seja  $K^A$ , o que caracteriza a *consistência hierárquica*. Como consequência tem-se o Corolário a seguir.

**Corolário 5.1** *Dado  $K^A$ ,  $\Gamma$ -compatível,  $\exists K \subset L_m(G)$  controlável e  $L_m$ -fechada tal que  $\Theta(K) = K^A$*

**Teorema 5.2** *Toda linguagem  $K \subseteq L_m(G)$  controlável e  $L_m$ -fechada, tem imagem  $\Gamma$ -compatível no nível agregado.*

O teorema anterior garante a *consistência hierárquica forte*.

Observe que os teoremas anteriores são válidos para autômatos agregados determinísticos, uma vez que a construção de  $\Gamma$  está restrita a autômatos determinísticos.

Caso uma linguagem  $K^A$  não for  $\Gamma$ -compatível, procura-se uma sub-linguagem  $\Gamma$ -compatível contida em  $K^A$ , tal que seja minimamente restritiva.

## 6 Exemplo: Linha de transferência.

Esta abordagem será ilustrada pelo desenvolvimento de um supervisor hierárquico para uma linha de transferência apresentada por Wonham (1998) no controle hierárquico tradicional. A linha de transferência consiste de duas máquinas  $M_1$  e  $M_2$  mais uma unidade de teste  $TU$  ligadas por dois buffers  $B_1$  e  $B_2$  na seqüência  $M_1, B_1, M_2, B_2, TU$  como mostrado na Fig. 5. A peça testada por  $TU$  pode ser aceita ou rejeitada; se for aceita, esta é liberada do sistema; senão volta para o buffer  $B_1$  para re-processamento por  $M_2$

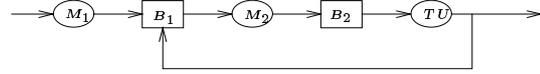


Figura 5. Linha de Transferência.

Os modelos dos componentes do sistema são mostrados na Fig. 6(a).

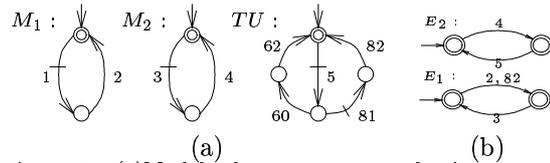


Figura 6. (a) Modelo dos componentes do sistema e (b) especificações para os buffers.

Os eventos controláveis 1 e 3 indicam o início de operação das máquinas  $M_1$  e  $M_2$ , respectivamente, e os eventos não controláveis 2 e 4 indicam o fim de operação das mesmas. O início da unidade de teste é representado pelo evento controlável 5 e os sinais de decisão de "aceito" ou "rejeitado" são representados pelos eventos 60 e 81 respectivamente. Caso a peça seja aceita pela unidade de teste, esta é enviada para fora do sistema (evento 62), caso contrário retorna para o buffer  $B_1$  (evento 82). Considera-se os buffers de capacidade unitária. Na Fig. 6(b) mostra-se a especificação para não *overflow* e não *underflow* dos buffers. A Fig. 7 apresenta o modelo de uma solução a este problema, e esta é adotada como planta  $G$  para tratar do problema de controle hierárquico.

No alto nível, a preocupação do supervisor atribui-se ao controle da relação entre peças de

entrada e peças aceitas ou recusadas. Portanto, considera-se como eventos relevantes os eventos relacionados com este fato (1, 60, 81). Com estes dados constrói-se uma agregação e o resultado conjuntamente com os padrões de controle são mostrados na Fig. 7. Observe-se que o bloco  $X_0$  apresenta um não determinismo, motivo pelo qual o evento que produz não determinismo, 1 é instanciado em  $1'$  e  $1''$ .

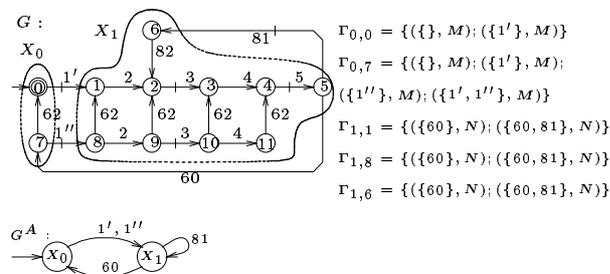


Figura 7. Planta e sua agregação.

Observa-se que o bloco  $X_0$  do autômato agregado (Fig. 7) apresenta conjuntos de estruturas de controle diferentes para os dois estados de entrada 0 e 7 ( $\Gamma_{0,0} \neq \Gamma_{0,7}$ ), motivo pelo qual o bloco  $X_0$  é dividido em dois, um associado a  $\Gamma_{0,0}$  e outro a  $\Gamma_{0,7}$ , representado pelos estados  $Y_0$  e  $Y_2$  na Fig. 8. Já o bloco  $X_1$ , para todas as entradas, apresenta um mesmo conjunto de estruturas de controle ( $\Gamma_{1,1} = \Gamma_{1,8} = \Gamma_{1,6}$ ), sendo mantido tal como está. Na Fig. 8,  $X_1$  está representado por  $Y_1$ .

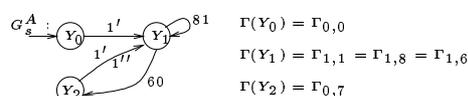


Figura 8. Planta hierárquica com estruturas de controle estado dependente.

A seguir define-se uma especificação para o sistema (Fig 9). Neste caso, toda vez que uma peça entra no sistema e for recusada duas vezes seguidas pela unidade de teste, então é forçada uma finalização e o sistema pára.

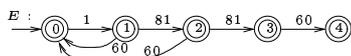


Figura 9. Especificação.

Continuando, obtém-se a linguagem alvo e aplica-se o Algoritmo para síntese da máxima linguagem  $\Gamma$ -compatível apresentado para sistemas com estruturas de controle estado dependente em (Cury et al., 2001). O resultado é mostrado na Fig. 10.

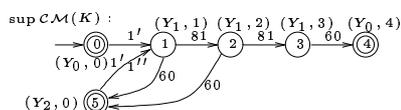


Figura 10. Máxima linguagem  $\Gamma$ -compatível.

Finalmente a Fig. 11 mostra a implementação global no baixo nível do supervisor hierárquico ótimo.

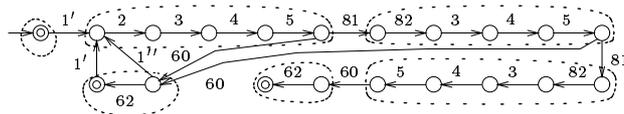


Figura 11. Implementação no baixo nível

## 7 Conclusões

O procedimento de obtenção deste modelo já apresenta a vantagem de possuir diretamente a consistência hierárquica forte. A limitação desta metodologia surge principalmente do fato que um grande número de problemas de controle supervisory locais tem que ser resolvidos para obter as estruturas de controle, mas em geral este cálculo é feito com sub-autômatos pequenos. Também a modularidade pode ser uma forma para reduzir esta complexidade, os autores atualmente estão investigando a possibilidade de incorporar os resultados sobre modularidade local proposto em (de Queiroz e Cury, 2000) ao problema de controle hierárquico.

## Agradecimentos

César Torrico foi apoiado pela CAPES. Cury foi apoiado em parte pelo CNPq, processos 3000953/93-3 e PRONEX 015/98.

## Referências Bibliográficas

- Caines, P. e Hubbard, P. (1998). A state aggregation approach to hierarchical supervisory control with applications to a transfer-line example, *Proceedings of the WODES 98*, Cagliari, Italy, pp. 2-7.
- Cury, J., Torrico, C. e da Cunha, A. (2001). A new approach for supervisory control of discrete event systems, *Proceedings of the European Control Conference 2001*.
- de Queiroz, M. H. e Cury, J. E. R. (2000). Modular supervisory control of large scale discrete event systems, *Proceedings of the WODES*.
- Ramadge, P. e Wonham, W. (1989). The control of discrete event systems, *Proceeding of the IEEE* **77**(1): 81-98.
- Wong, K. e Wonham, W. (1996). Hierarchical control of discrete-event systems, *Discrete Event Dynamical Systems* **6**: 241-273.
- Wonham, W. (1998). *Notes on Control of Discrete-Event Systems. Course Notes for ECE 1636F/1637S*, Revision 98.09.01.
- Zhong, H. e Wonham, W. (1990). On the consistency of hierarchical supervision in discrete-event systems, *IEEE Transactions on Automatic Control* **35**(10): 1125-1134.