

# UTILIZANDO LÓGICA TEMPORAL E REDES DE PETRI NA SÍNTESE DE SUPERVISORES DE SISTEMAS A EVENTOS DISCRETOS

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**— A lógica temporal de tempo ramificado é utilizada para especificar o comportamento desejado e para representar a dinâmica de sistemas a eventos discretos modelados por redes de Petri multiplexadas. As redes de Petri multiplexadas tem a mesma expressividade das redes de Petri lugar/transição e simplificam a utilização da lógica temporal. As funções associadas às transições de uma rede de Petri multiplexada podem ser transformadas em fórmulas CTL, para facilitar a construção do supervisor para sistemas a eventos discretos. O supervisor tem a mesma estrutura da rede de Petri multiplexada que modela o sistema diferenciando-se apenas pelas fórmulas lógicas que definem a semântica desejada.

**Abstract**— The computational tree logic is employed for specifying the desired behavior and the dynamics of discrete event systems modelled with multiplexed Petri nets. The multiplexed Petri nets have the same expressiveness of the place/transition nets but simplifies the use of temporal logic for discrete event system applications. The functions associated to the transitions of a multiplexed Petri net can be converted into logical formulas for simplifying the synthesis of the supervisor for discrete event systems. The supervisor has the same structure of the model of the supervised discrete event system but has a different set of logical formulas that define the semantics desired.

**Key Words**— Petri Nets, Temporal Logic, Discrete Event Systems

## 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-se um supervisor. A solução do problema de controle supervisório utilizando a TCS é resolvida em três fases distintas: a modelagem, a especificação de comportamento e a síntese do supervisor.

Um dos paradigmas para a modelagem de SEDs são as redes de Petri (Cofrancesco et al., 1991; DiCesare et al., 1993; Sreenivas, 1997). Em Barroso (Barroso, 1996), este paradigma é utilizado para modelar o SED, onde através do comportamento especificado e da árvore de alcançabilidade da rede, constrói-se um conjunto de funções para as transições da rede supervisora, que permite ou bloqueia os eventos do sistema de acordo com as marcações atingidas,

ou seja, a rede supervisora é uma rede de Petri com funções de habilitação de transições (Papelis e Casavant, 1992).

Para a especificação de comportamento, dois formalismos matemáticos podem ser utilizados: *i*) as linguagens formais (Hopcroft e Ullman, 1979), na qual se associam símbolos aos eventos e, desse modo, a tarefa que se deseja realizar é representada por uma linguagem; *ii*) a lógica temporal (Pnueli, 1977; Finger e Gabbay, 1993; Galton, 1987), na qual se associam proposições aos eventos (Uchihira e Honiden, 1990) ou estados (Racloz e Buchs, 1997) e, desse modo, o comportamento desejado é expresso através de fórmulas lógicas construídas com essas proposições (Costa, 1997).

Nesse trabalho, o paradigma utilizado para a modelagem do SED e para a construção do supervisor são as redes de Petri multiplexadas (RPM) (Costa e Lima, 1999). O uso desse tipo de rede simplifica a obtenção da fórmula lógica que expressa o comportamento do SED já que a semântica da rede é definida através da lógica temporal de tempo ramificado (CTL). Considerando que o comportamento desejado também é expresso através de uma fórmula CTL, há duas alternativas básicas para realizar a síntese da rede supervisora: *i*) efetuar a verificação e validação da fórmula que expressa o comportamento desejado em relação à fórmula que descreve o comportamento do SED ou *ii*) construir a árvore de alcançabilidade da RPM e verificar nessa árvore a validade da fórmula que expressa o comportamento desejado. Efetuada a verificação e validação, as partes da fórmula que define o comportamento desejado que não são factíveis em relação ao comportamento do SED devem ser eliminadas para obter o supervisor. Desse modo, o supervisor terá a mesma estrutura do modelo do SED, diferenciando apenas nas fórmulas CTL associadas às transições.

O artigo é apresentado como a seguir: na seção 2, é apresentada a CTL e as RPM; na seção 3 é apresentada a transformação das funções de transição da RPM em fórmulas CTL e as condições para uma especificação de comporta-

mento ser validada; na seção 4 é apresentado um exemplo de síntese do supervisor nesse contexto e, na seção 5 são apresentadas as conclusões do trabalho.

## 2 PRELIMINARES

**Definição 1** *Lógica temporal de tempo ramificado (CTL)*

1. Sintaxe: Fórmulas CTL são construídas de um conjunto de proposições atômicas  $Prop = \{p_1, \dots\}$ , conectivos lógicos  $\vee, \wedge, \neg$  e operadores temporais  $X$  (próximo),  $U$  (até que),  $F$  (eventualmente),  $A$  (necessariamente),  $E$  (possivelmente) e  $G$  (sempre). Os operadores temporais podem ser combinados apenas em pares, isto é, i)  $A$  seguido por  $F, G, U$  ou  $X$  ou ii)  $E$  seguido por  $F, G, U$  ou  $X$ . As regras de construção de fórmulas são: uma proposição atômica  $p \in Prop$  é uma fórmula e, se  $f, g$  são fórmulas, também são  $A(fUg)$ ,  $f \wedge g$ ,  $Aff$ ,  $AGg$ ,  $\neg f$ ,  $EXg$ .
2. Semântica:  $EXq$  – possivelmente  $q$  será verdadeira no próximo estado que sucede o estado  $q$ ;  $E(qU(f \wedge p))$  – possivelmente  $q$  será verdadeira em todos os estados até que o estado onde  $f$  e  $p$  sejam verdadeiras ao mesmo tempo;  $AGq$  –  $q$  é necessariamente verdadeira para todo estado alcançado (ver (Racloz e Buchs, 1997; McMillan, 1992; Costa, 1997)).

Define-se uma linguagem gerada de uma fórmula CTL, da seguinte maneira:

**Definição 2** Se  $f$  é uma fórmula CTL então  $L_s(f)$  é uma linguagem gerada da fórmula e que representa caminhos em uma árvore de decisão binária.

**Exemplo 1** Considere o conjunto  $Prop = \{\alpha, \beta\}$ . Dada a fórmula  $f = A(\alpha U \beta) \vee AG\beta$ , sua linguagem  $L_s(f)$  pode ser representada pela expressão regular  $(\alpha^* + \beta^*)\beta^*$  cujos caminhos são indicados na Figura 1.

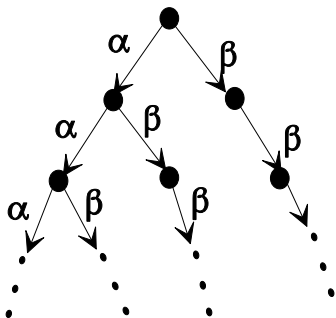


Figura 1. Árvore de caminhos do Exemplo 1

### 2.1 REDES DE PETRI E CTL

A CTL, pode descrever o comportamento dinâmico de uma rede de Petri através de fórmulas se proposições atômicas forem associadas às transições da rede. Esse mapeamento é feito de forma que cada proposição atômica corresponde a uma transição da rede de Petri de acordo com:  $Prop = \Omega(T)$ , onde  $\Omega$  é o operador que mapeia cada proposição atômica e  $T$  é conjunto de transições da rede. Logo, uma proposição atômica  $f = \Omega(t)$  é verdadeira se e só se a transição  $t \in T$  está habilitada ou dispara.

A linguagem de uma rede de Petri é expressa através de fórmulas em CTL de acordo com:

**Definição 3** Seja  $E_{rpe} = (P, T, A, K, W, \Omega)$  uma estrutura de rede de Petri etiquetada, em que  $P = \{p_1, \dots, p_m\}$  é um conjunto de lugares,  $T = \{t_1, \dots, t_n\}$  é um conjunto de transições,  $A \subseteq (T \times P) \cup (P \times T)$  é um conjunto de arcos,  $K : P \rightarrow \mathbb{N} \cup \{\infty\}$  é a função de capacidade,  $W : A \rightarrow \mathbb{N}^+$  é a função de ponderação e  $\Omega : T \rightarrow Prop$  a função de etiquetagem das transições sobre um conjunto de proposições atômicas  $Prop$ , e seja  $RPE = (E_{rpe}, M_0)$  uma rede de Petri etiquetada, onde  $M_0 : P \rightarrow \mathbb{N}$  é a função de marcação inicial, que satisfaz  $\forall p \in P, M_0(p) \leq K(p)$ . Então define-se

$$L(RPE, f) \equiv L_\infty(RPE) \cap L_s(f)$$

como sendo a linguagem da rede de Petri etiquetada em termos de uma fórmula CTL, onde  $L_\infty(RPE)$  é a linguagem gerada da rede de Petri etiquetada e  $L_s(f)$  é a linguagem gerada da fórmula CTL  $f$ .

Uma fórmula é uma sentença que será verdadeira se existir um caminho factível na árvore de alcançabilidade da rede de Petri. No exemplo seguinte, a CTL é usada para descrever o comportamento de um SED:

**Exemplo 2** Considere um SED cuja rede de Petri é mostrada na Figura 2. A árvore de alcançabilidade dessa rede é apresentada na Figura 3. Considerando que  $t_1, t_2$  e  $t_3$  são proposições atômicas, então:

- A fórmula  $A(t_1 U t_2)$  é verdadeira, pois representa a linguagem  $(t_1 t_2)^*$ , a qual define a seqüência de disparos das transições  $t_1$  e  $t_2$ , nessa ordem, e que pertence à linguagem gerada pela rede que, nesse caso é sempre possível;
- A fórmula CTL  $A(t_1 U t_2) \wedge AX(\neg t_1 \wedge t_3)$  exprime a seqüência  $t_1 t_2 t_3$ . Contudo, esta fórmula não é factível para a marcação inicial desse exemplo, pois não existe esta seqüência (ver árvore de alcançabilidade). Logo, a fórmula é falsa.

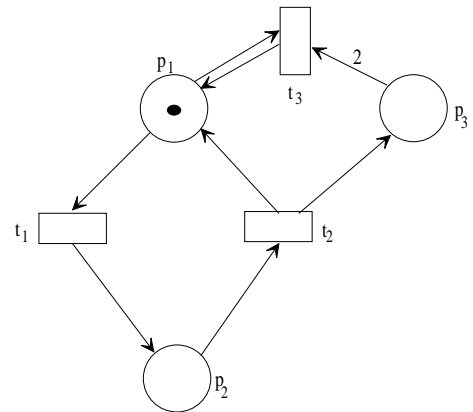


Figura 2. Rede de Petri utilizada no exemplo 2

### 2.2 REDES DE PETRI MULTIPLEXADAS

**Definição 4** Uma rede de Petri multiplexada, é uma quintupla  $RPM = (E_{rp}, R_A, R_T, M_0, \Psi)$ , onde:  $E_{rp} = (P, T, A, K, W)$  é uma estrutura de rede de Petri;  $R_A : A \rightarrow \Sigma$  é a função que etiqueta os arcos, onde  $\Sigma$  é um alfabeto;  $R_T : T \rightarrow \Sigma$  é a função que etiqueta as transições,  $\Psi : \mathcal{R} \rightarrow \{0, 1\}$  é a função de habilitação das transições para o conjunto de arcos da rede e  $\mathcal{R}$  representa o conjunto de marcações alcançáveis da rede a partir da marcação inicial  $M_0$ .

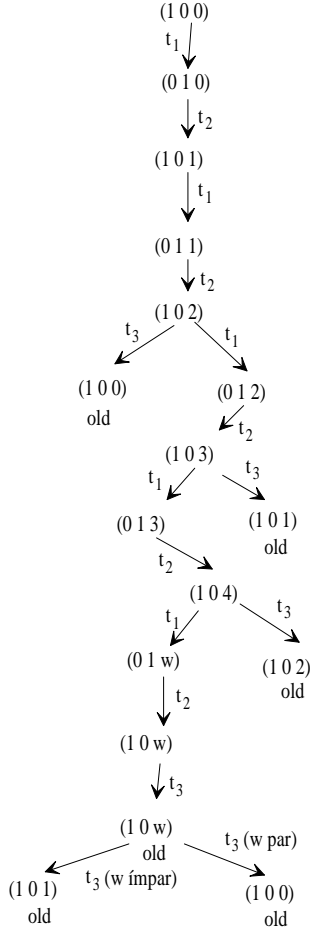


Figura 3. Árvore de alcançabilidade da RP, do exemplo 2.

As funções  $\psi_k$ , são funções associadas às transições  $t_k$ , compostas por subfunções  $\tau_l(M(p_j), w_i a_i)$ , dependentes da marcação do lugar  $p_j$ , que influencia o arco  $a_i$  (arco entre uma transição e um lugar), e  $w_i$  é o peso deste arco. Desse modo,  $\psi_k$  é definida por

$$\psi_k = \bigvee_{l=1, L} \tau_l \left( \bigwedge_j M(p_j), \bigwedge_i w_i a_i \right),$$

$$p_j \in (t^\bullet)_k \cup (\bullet t)_k,$$

$$a_i \in (t_k \times (t^\bullet)_k) \cup ((\bullet t)_k \times t_k)$$

com  $(\bullet t)_k$  e  $(t^\bullet)_k$  representando os conjuntos dos lugares de entrada e de saída da transição  $t_k$ , respectivamente.

Sua dinâmica é descrita pela seguinte regra:

**Definição 5** A marcação de uma rede de Petri multiplicada, evolui de acordo com a seguinte regra de disparos de transições:

1. Uma transição  $t_k$  é dita habilitada (pronta para disparar) na marcação  $M$ , para um arco  $a_i$ , se e somente se: i)  $W(p, t_k) \leq M(p)$ ,  $\forall p \in (\bullet t)_k$ , ii)  $M(p) \leq K(p) - W(t_k, p)$ ,  $\forall p \in (t^\bullet)_k$  e iii)  $\psi_k = 1$ .
2. Uma transição  $t_k$  habilitada, pode ou não disparar;
3. O disparo de uma transição  $t_k \in T$ , habilitada na marcação  $M$ , é instantânea e resulta em uma nova marcação  $M'$ , dada pela equação

$$M'(p) = M(p) - W(p, t_k) + W(t_k, p), \forall p \in (t^\bullet)_k \cup (\bullet t)_k;$$

4. A ocorrência do disparo de  $t_k$ , que modifica a marcação  $M$  da rede para uma nova marcação  $M'$ , é denotada por  $M[t_k]M'$ .

A marcação da rede varia com a função da transição que define quais arcos devem ser habilitados de um conjunto de lugares para outro. Quando uma transição dispara, não depende necessariamente de todo o conjunto de lugares de entrada e de saída da transição, devido a sua função lógica associada. Se em uma transição  $t$  a função tem sempre valor verdadeiro ( $\psi=1$ ), define que  $t$  está sempre habilitada para todos os seus arcos. Todas as propriedades das redes de Petri lugar/transição são válidas para as RPMs, consideradas as restrições impostas pelas subfunções  $\tau$  que compõem as funções  $\psi$ . Também, a RPM pode incluir uma função de habilitação de transição como uma RPFHT (Papelis e Casavant, 1992).

Na figura 4(a), é mostrada uma RPM, com suas equivalentes RPFHT (Fig. 4(b)) e rede de Petri lugar/transição (fig. 4(c)). Esta última, apresenta capacidade em alguns lugares, o que impede das transições  $t_1$  e  $t_3$  dispararem, quando esta marcação é alcançada. Na RPFHT, as funções associadas a estas transições são definidas por  $\varphi_1 = \varphi_2 = [M(p_1) < 4]$  e  $\varphi_3 = [M(p_4) < 5]$ , e garantem suas desabilitações, eliminando a função de capacidade. Na RPM, há uma compactação das transições da rede, dadas por  $t'_2 = t_1 \cup t_2$  e  $t'_3 = t_3 \cup t_4$ , com as funções dadas por  $\psi_2 = \{(M(p_1) = 3 \Rightarrow \neg t_2) \vee (M(p_3) = 0 \Rightarrow t_2 : \neg a_5)\}$  e  $\psi_3 = \{((M(p_4) \leq 4 \wedge M(p_2) > 0 \wedge M(p_3) > 0) \Rightarrow t_3 : a_3 \wedge a_6 \wedge a_7 \wedge a_8) \vee ((M(p_2) > 0) \Rightarrow t_3 : a_3 \wedge a_7)\}$ , respectivamente, onde essas funções são constituídas por subfunções  $\tau$ , dependentes das marcações de alguns lugares e dos arcos ligados a estas transições. Assim, na marcação  $M_0 = [2 \ 0 \ 0 \ 0]^T$ , as transições habilitadas são  $t_5$  e  $t_2$ , com  $t_2$  valendo apenas para o arco  $a_4$ , o que equivale a transição  $t_1$  da RPFHT. Na marcação  $M = [0 \ 0 \ 1 \ 4]^T$ , as transições habilitadas da RPM são  $t'_2$  e  $t'_3$ , onde  $t'_3$  só está habilitada para os arcos  $a_3$  e  $a_7$ , procedendo ao que faz a função  $\varphi_3$  da RPFHT.

### 2.3 SEQÜÊNCIA DE DISPAROS

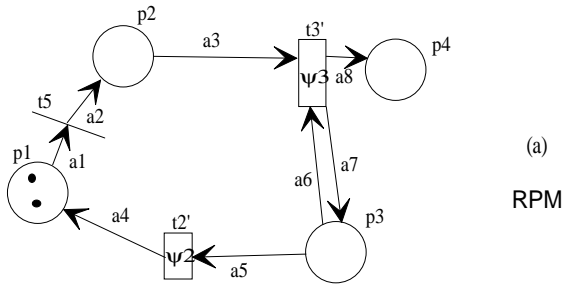
Uma RPM tem uma linguagem associada que é representada por seqüências de disparos de transições. Essa linguagem é definida associando-se palavras às subfunções e às transições nas quais  $\psi = 1$ . Isso define um caminho que leva de uma marcação para outra. Na Figura 5 é vista uma RPM e sua linguagem.

## 3 TRANSIÇÕES E FÓRMULAS CTL

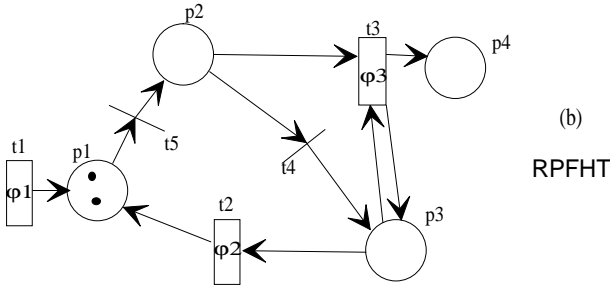
Uma transição numa RPM pode ser associada a uma função lógica que define o seu disparo em relação aos arcos. Essa função é dependente da marcação dos lugares da rede. Considerando que a CTL permite descrever caminhos numa árvore, os operadores da CTL podem ser utilizados para descrever as funções lógicas associadas às transições de uma RPM.

**Definição 6** A interpretação dos operadores da CTL em relação à dinâmica da rede é feita da seguinte forma:

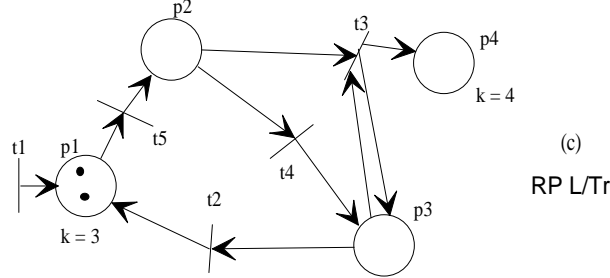
- $AGt$  -  $t$  sempre dispara em qualquer marcação;
- $AFt$  -  $t$  sempre pode disparar em qualquer marcação;
- $AXt$  -  $t$  sempre dispara na próxima marcação;



(a)  
RPM



(b)  
RPFHT



(c)  
RPL/Tr

Figura 4. Exemplo de RPM com suas equivalentes RPFHT e RPL/Tr

- $A(t_i U t_j)$  -  $t_i$  sempre dispara em qualquer marcação até que  $t_j$  dispare em alguma marcação;
- $EGt$  -  $t$  sempre dispara em pelo menos uma marcação;
- $EFt$  -  $t$  sempre pode disparar em pelo menos uma marcação;
- $EXt$  -  $t$  sempre pode disparar na próxima marcação;
- $E(t_i U t_j)$  -  $t_i$  sempre pode disparar em qualquer marcação até que  $t_j$  dispare em alguma marcação;

**Exemplo 3** Considere a RPM da Figura 6. A função lógica associada à sua transição é

$$\Psi = \{ \underbrace{(M(p_1) = 1 \Rightarrow t : a_1 \wedge a_3)}_{\tau_1} \vee \underbrace{(M(p_2) = 1 \Rightarrow t : a_4 \wedge a_5)}_{\tau_2} \vee \underbrace{(M(p_3) = 1 \Rightarrow t : a_6 \wedge a_2)}_{\tau_3} \}$$

Essa função determina uma seqüência repetitiva que pode ser representada pela fórmula

$$f = A(A((a_1 \wedge a_3) U (a_4 \wedge a_5)) U (a_6 \wedge a_2))$$

para  $M_0 = [1 \ 0 \ 0]^T$ .

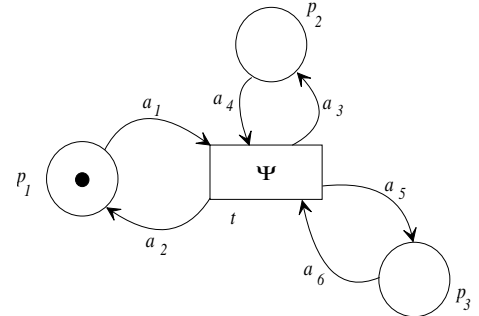


Figura 6. RPM com seqüência repetitiva.

**Exemplo 4** Considere a RPM da Figura 7. A função lógica associada à sua transição é

$$\Psi = \{ (t : a_2) \vee (M(p_1) = 5 \Rightarrow t : 5a_1 \wedge a_3) \vee (M(p_2) = 2 \Rightarrow t : 2a_4) \}$$

Essa função pode ser representada pela fórmula

$$f = AGa_2 \vee EF(5a_1 \wedge a_3) \vee AX(2a_4),$$

onde os números que antecedem os arcos definem seu peso.

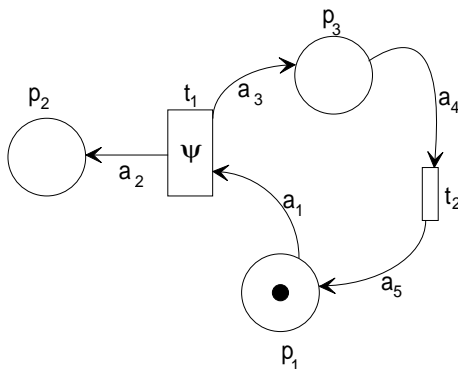


Figura 5. Uma RPM que gera a linguagem  $L = (\tau_1 t_2)^* (\tau_1 \tau_2^3 + \epsilon)$

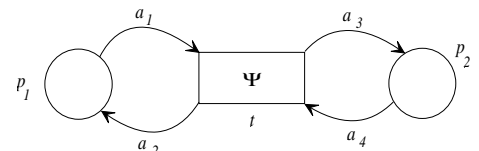


Figura 7. RPM para o Exemplo 4.

**Exemplo 5** Considere a RPM da Figura 8. A função lógica associada à sua transição é

$$\Psi = \{(M(p_1) = 1 \Rightarrow t : a_1 \wedge a_3) \vee (M(p_2) = 1 \Rightarrow t : a_2 \wedge a_4) \vee (M(p_3) = 1 \wedge M(p_4) = 1 \Rightarrow t : a_7 \wedge a_5 \wedge a_6 \wedge a_8)\}.$$

Essa função que determina uma concorrência seguida de uma sincronização, pode ser representada pela fórmula

$$\mathbf{f} = A(F(a_1 \wedge a_3) \vee F(a_2 \wedge a_4)) \wedge AX(a_7 \wedge a_5 \wedge a_6 \wedge a_8).$$

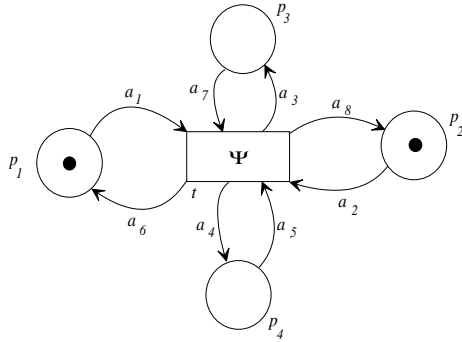


Figura 8. RPM com concorrência seguida de sincronização.

Os exemplos acima ilustram como definir o comportamento da rede através de uma fórmula CTL. Desse modo, uma determinada especificação de comportamento também pode ser expressa através de uma fórmula CTL desde que a linguagem representada por essa fórmula esteja contida na linguagem gerada pelo SED. A linguagem gerada pelo SED é determinada através da construção de sua árvore de alcançabilidade que pode ser representada de forma compacta através de uma fórmula CTL.

**Definição 7** Uma especificação de comportamento definida por uma fórmula CTL e é dita válida para um SED representado por uma RPM, que tem sua dinâmica descrita por uma fórmula CTL  $\mathbf{f}$ , se e somente se  $\mathbf{e} \subseteq \mathbf{f}$ .

A condição  $\mathbf{e} \subseteq \mathbf{f}$  significa que a especificação de comportamento é um subconjunto das seqüências existentes na árvore de alcançabilidade da rede.

#### 4 SÍNTESE DE SUPERVISOR PARA UM SISTEMA DE MANUFATURA

Considere um sistema de manufatura simplificado, onde há seis usuários e oito recursos disponíveis e os usuários compartilham estes recursos. Os recursos são limitados a um número máximo de 5 por usuário. As fichas colocadas no lugar  $p_1$  são as peças que serão processadas pelas outras estações (os lugares  $p_2, p_3, p_4, p_5$  e  $p_6$ ), e o lugar  $p_7$  contém as ferramentas para o processamento dos itens (fichas em  $p_1$ ). Observa-se que, para o término do processamento, o sistema está com 5 ferramentas em uso. Após seu término, todas as ferramentas são devolvidas. Pode-se ver que mais de uma peça pode ser processada ao mesmo tempo. O modelo deste sistema é mostrado na Figura 9. A função de transição é dada por

$$\psi = \{((M(p_1) \geq 1 \wedge M(p_7) \geq 1) \Rightarrow t : a_1 \wedge a_{12} \wedge a_3) \vee ((M(p_2) \geq 1 \wedge M(p_7) \geq 2) \Rightarrow t : a_2 \wedge 2a_{12} \wedge a_5) \vee ((M(p_3) \geq 1) \Rightarrow t : a_4 \wedge a_{11} \wedge a_7) \vee ((M(p_4) \geq 1 \wedge M(p_7) \geq 3) \Rightarrow t : a_6 \wedge 3a_{12} \wedge a_9) \vee ((M(p_5) \geq 1) \Rightarrow t : a_8 \wedge 5a_{11} \wedge a_{10})\}$$

e pode ser representada pela fórmula CTL

$$\mathbf{f} = AF(a_1 \wedge a_{12} \wedge a_3) \vee EF(a_2 \wedge 2a_{12} \wedge a_5) \wedge E((a_4 \wedge a_{11} \wedge a_7)U(a_6 \wedge 3a_{12} \wedge a_9)) \vee AX(a_8 \wedge 5a_{11} \wedge a_{10})$$

que representa a evolução do sistema.

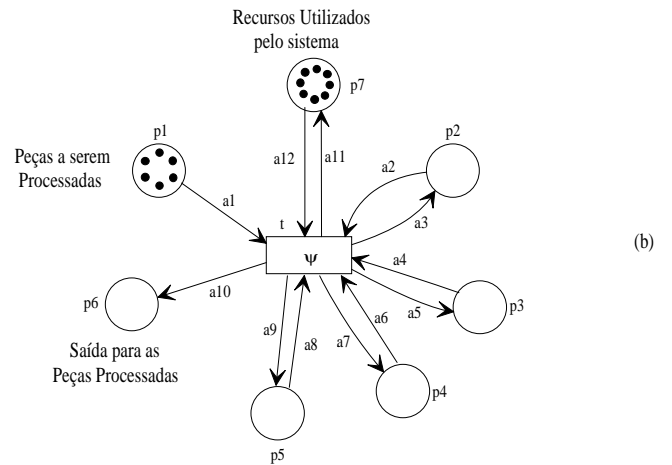
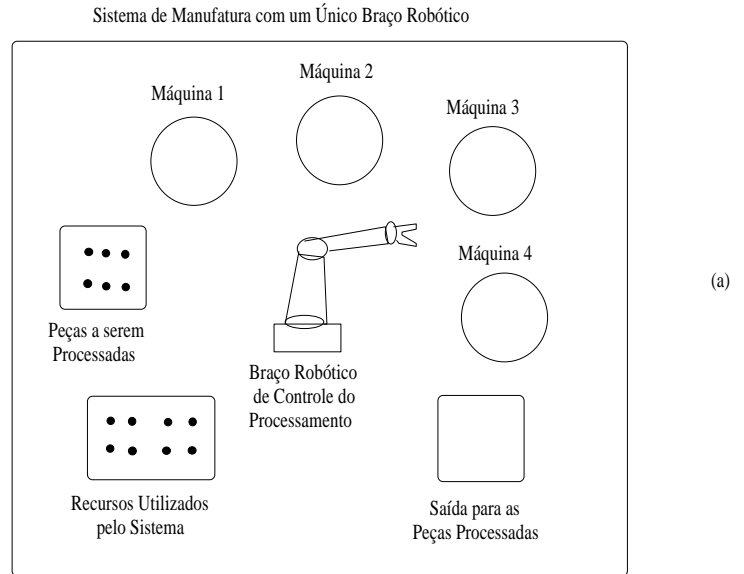


Figura 9. Modelo de RPM de um sistema de manufatura com recursos compartilhados.

Utilizando o algoritmo AMAR<sub>RPM</sub> (Costa e Lima, 1999) constrói-se a árvore de alcançabilidade desse modelo que contém 162 marcações alcançáveis distintas. Interessa aqui, atingir a marcação  $M = [0\ 0\ 0\ 0\ 6\ 8]^T$ , que define que todos os usuários completaram a tarefa. A especificação de comportamento definida é de que todas as peças sejam processadas e que no máximo duas peças sejam processadas em paralelo, ou seja,

$$\mathbf{e} = AF(a_1 \wedge a_{12} \wedge a_3) \wedge EX(a_1 \wedge a_{12} \wedge a_3) \wedge AF(a_2 \wedge 2a_{12} \wedge a_5) \wedge A((a_4 \wedge a_{11} \wedge a_7)U(a_6 \wedge 3a_{12} \wedge a_9)) \wedge AX(a_8 \wedge 5a_{11} \wedge a_{10}).$$

Essa especificação deve ser interpretada da seguinte maneira: o processamento deve ser iniciado para uma peça ( $AF(a_1 \wedge a_{12} \wedge a_3)$ ) e em seguida o processamento de uma segunda peça pode ser iniciado ( $EX(a_1 \wedge a_{12} \wedge a_3)$ ). As peças que começaram a serem usinadas devem passar pelas demais etapas de processamento ( $AF(a_2 \wedge 2a_{12} \wedge a_5) \wedge$

$A((a_4 \wedge a_{11} \wedge a_7)U(a_6 \wedge 3a_{12} \wedge a_9)))$  sem que nenhuma outra peça entre no sistema de manufatura até que a usinagem de uma das peças seja concluída ( $AX(a_8 \wedge 5a_{11} \wedge a_{10})$ ).

É possível mostrar que  $e \subset f$  e, desse modo, a especificação pode ser utilizada diretamente para descrever o comportamento do supervisor.

## 5 CONCLUSÃO

Esse trabalho mostra que é possível utilizar a lógica temporal de tempo ramificado para simplificar a especificação e eliminar a ambiguidade na descrição do comportamento de sistemas a eventos discretos modelados por redes de Petri multiplexadas. Essa abordagem facilita a especificação de comportamento para esses sistemas, desde que a especificação pode ser obtida diretamente da fórmula da transição da rede. A semântica operacional que define o comportamento desejado é definida pela escolha das funções de habilitação de transições da rede multiplexada que são formalizadas com operadores da lógica de tempo ramificado. Quando a especificação de comportamento desejada não é validada, os caminhos que ela representa não existem na árvore de alcançabilidade da rede. Desse modo, a fórmula que representa a especificação de comportamento desejada deve ser alterada através da eliminação das partes que a tornam não factível.

## Referências

- Barroso, G. (1996). *Uma Nova Abordagem para a Síntese de Supervisores de Sistemas a Eventos Discretos*, Tese de Doutorado. Universidade Federal da Paraíba, Campus II, Campina Grande, PB, Brasil.
- Cofrancesco, P., Cristoforetti, A. e Scattolini, R. (1991). Petri nets based approach to software development for real-time control, *IEE Proceedings-D Control Theory and Applications* **138**(5): 474–478.
- Costa, E. (1997). *Contribuição ao Uso da Lógica Temporal na Especificação de Comportamentos de Sistemas a Eventos Discretos*, Dissertação de Mestrado. Universidade Federal da Paraíba, Campus II, Campina Grande, PB, Brasil.
- Costa, E. e Lima, A. (1999). Utilizando redes de petri multiplexadas na síntese de supervisores de sistemas a eventos discretos, *Anais do IV Simpósio Brasileiro de Automação Inteligente - SBAI* pp. 533–538.
- DiCesare, F., Harhalakis, G., Proth, J., Silva, M. e Verna-dat, F. (1993). *Practice of Petri Nets in Manufacturing*, Chapman and Hall.
- Finger, M. e Gabbay, D. (1993). *Adding a Temporal Dimension to a Logic System*, Springer-Verlag.
- Galton, A. (1987). *Temporal Logics and their Applications*, Academic Press.
- Hopcroft, J. e Ullman, J. (1979). *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, USA.
- McMillan, K. (1992). *Symbolic Model Checking: An Approach to the State Explosion Problem*, PhD thesis, Carnegie Mellon University.
- Papelis, A. e Casavant, T. (1992). Specification and analysis of parallel/distributed software and systems by petri nets with transition enabling functions, *IEEE Transactions on Software Engineering* **18**(3): 252–261.
- Pnueli, A. (1977). The temporal semantics of concurrent programs, *18th Symposium on Foundations of Computer Science*.
- Racloz, P. e Buchs, D. (1997). Symbolic proof of ctl formulae over petri nets, *Technical report*, C.U.I. University of Geneva.
- Ramadge, P. e Wonham, W. (1982). Supervision of discrete event processes, *Proceedings of 21st Conference on Decision and Control* pp. 1228–1229.
- Ramadge, P. e Wonham, W. (1989). The control of discrete event systems, *Proceedings of the IEEE* **77**(1): 81–98.
- Sreenivas, R. (1997). On the existence of supervisory policies that enforces liveness in discrete-event dynamic systems modeled by controlled petri nets, *IEEE Transactions on Automatic Control* **42**(7): 928–945.
- Uchihira, N. e Honiden, S. (1990). Verification and synthesis of concurrent programs using petri nets and temporal logic, *Technical report*, Institute for New Generation Computer Technology.