

# REGRAS DE CONTROLE PARA ALOCAÇÃO DE RECURSOS EM SISTEMAS PRODUTIVOS COM PROCESSOS CONCORRENTES

FRANCISCO YASTAMI NAKAMOTO

*Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos  
Escola Politécnica da Universidade de São Paulo  
Avenida Professor Mello Moraes, 2231, São Paulo, SP, 05508-900  
E-mails: francisco.nakamoto@poli.usp.br*

NEWTON MARUYAMA, PAULO EIGI MIYAGI, DIOLINO JOSÉ DOS SANTOS FILHO

*Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos  
Escola Politécnica da Universidade de São Paulo  
Avenida Professor Mello Moraes, 2231, São Paulo, SP, 05508-900  
E-mails: maruyama@usp.br, pemiyagi@usp.br, diolinos@usp.br*

**Resumo**— Sistemas produtivos flexíveis, que executam múltiplos processos simultâneos com compartilhamento de recursos, podem apresentar indeterminismo em relação ao tempo e ao sequenciamento das atividades no contexto global. Tais sistemas podem ser classificados como Sistemas Produtivos Complexos e modelados através de Redes de Petri e suas derivadas quando abordados como sendo uma classe de Sistemas a Eventos Discretos. Neste contexto, se a alocação de recursos compartilhados não for gerenciada adequadamente, é possível ocorrer um auto-travamento ou ‘deadlock’ do sistema devido a falta de informação quanto ao sequenciamento do processo. Uma abordagem efetiva para este problema global realizado no sistema é o método ‘deadlock avoidance’ que consiste na detecção do estado pré-deadlock conjugada a uma reação que evita a evolução do sistema para o estado de travamento. Este método baseia-se no modelo GAR (Grafos de Alocação de Recursos). Este trabalho propõe uma sistematização deste método que possibilita a geração automática, através de ferramentas computacionais do modelo de controle de recursos.

**Abstract**— Flexible productive systems which execute multiple simultaneous processes using the same resources set is not possible to determine when which events occur regarding time and the sequence of the activities in the global context. Such systems can be classified as Complex Productive Systems and modeled through Petri's Net and its derivatives. In this context, if the allocation of shared resources is not managed adequately, is possible to occur deadlock of the system due to information lack regarding the sequence of the activities of the process. An effective approach for this global problem accomplished in the system is the method deadlock avoidance that consists in the detection of the previous state to deadlock conjugated to a reaction that avoids the evolution of the system for the state deadlock indeed. This method it bases on model RAG (Resource Allocation Graph). This work proposes an operations sequence of this method that enables the automatic generation, through computational tools of the resources control model.

**Keywords**— Deadlock Avoidance, Petris Net, Production System Control, Resource Allocation.

## 1 Introdução

De uma forma genérica podemos conceituar sistemas produtivos (SPs) como aqueles capazes de realizarem um conjunto de atividades de acordo com objetivos pré-determinados e que apresentam certas peculiaridades em seu comportamento dinâmico. Por exemplo, freqüentemente depara-se com características do tipo paralelismo, concorrência e assincronismo, presentes em sua evolução dinâmica.

Considerando-se o fato de haver uma tendência marcante no sentido de tornar-se cada vez menor o ciclo de vida de um produto manufaturado em um SPs, é conveniente enfatizar a importância que o conceito de flexibilidade assume para garantir um nível de produtividade adequado a estes sistemas. Desta forma, observa-se uma migração dos tradicionais SPs baseados no conceito de produção em lote para a arquitetura em que há uma maior flexibilidade, permitindo a execução de processos simultâneos caracterizados como Sistemas Produtivos Flexíveis

(SPFs). Os SPFs produzem uma variedade de produtos modificando suas configurações rapidamente de acordo com o planejamento da produção. Esta flexibilidade permite uma alocação mais adequada dos recursos, mas incrementa a complexidade do controle do sistema.

O estado tecnológico atual oferece equipamentos cada vez mais eficientes, porém, esta eficiência restringe-se as atividades inerentes a cada processo. Aprimorar a eficiência dos processos de uma forma global requer um maior controle sobre a utilização dos equipamentos. O controle sobre um único processo é relativamente simples, mas em um sistema produtivo com múltiplos processos simultâneos e compartilhamento de um conjunto finito de recursos é possível a ocorrência de ‘deadlock’ ou auto-travamento do sistema. Este fenômeno ocorre quando o fluxo dos processos são permanentemente impedidos e/ou quando as operações dos processos não podem ser executados. Basicamente existem três métodos para abordar o problema de ‘deadlock’: (i) Prevenir (‘Prevent’), (ii) Evitar (‘Avoid’) e (iii) Detectar e Restabelecer (‘Detect and Resolution’). Neste

contexto, este trabalho baseia-se no método de ‘deadlock avoidance’. Banaszak (1990) propõe um algoritmo de restrições que age sobre o sistema utilizando buffers dos recursos. Viswanadham (1990) utiliza um algoritmo que age sobre um grafo de atingibilidade do sistema. Os dois métodos utilizam a Redes de Petri para modelar o sistema. Nestes trabalhos podemos verificar que a interação entre o controle sobre o processo e o controle sobre os recursos é realizada em um mesmo plano, aumentando o grau de complexidade de acordo com o tamanho do sistema.

O método proposto por Santos Filho para SPFs, aplica uma arquitetura hierárquica sobre o sistema de controle (Santos Filho, 2000a). Este método utiliza ferramentas gráficas e matemática Enhanced Mark Flow Graph (E-MFG), derivada de Redes de Petri, e Grafos de Alocação de Recursos (GAR), associando-se regras de produção para representar o conjunto de restrições ao sistema de controle afim de evitar o auto-travamento. Portanto, o objetivo deste trabalho é apresentar a possibilidade de utilização de recursos computacionais para a implementação de algoritmos para a geração automática de regras de controle conforme o método proposto em (Santos Filho, 2000a) e, como parte integrante da arquitetura de uma ferramenta de auxílio ao projeto de sistemas de controle de SPFs.

## 2 Sistema Produtivo Flexível

Em um sistema produtivo flexível com compartilhamento de recursos finitos entre múltiplos processos, a competição de recursos pelos processos pode levar o sistema a um auto-travamento ou ‘deadlock’. Esta situação pode ocorrer quando dois ou mais processos competem por um número limitado de recursos equivalente a situação em que processos aguardam a liberação de recursos utilizados por outros processos (Holliday, 1998). Este fenômeno ocorre em muitos sistemas onde recursos e informações são compartilhados. Por exemplo, um processo **A** aguarda a liberação do recurso alocado pelo processo **B** e o processo **B** aguarda a liberação do recurso alocado pelo processo **A**. Neste estado, só uma intervenção externa poderá tirar o sistema do estado de ‘deadlock’, conforme ilustrado na figura 1.

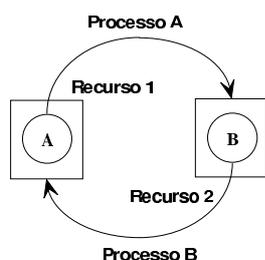


Figura 1. Exemplo de ‘deadlock’.

A interação entre processo e recurso é realizada em três etapas fundamentais: (i) Alocação de Recurso; (ii) Utilização de Recurso; (iii) Liberação do

Recurso. Um processo que tenta alocar um recurso que não esteja disponível, causa um bloqueio no sistema. Existe uma diferença entre um sistema bloqueado e um sistema travado. Um sistema bloqueado aguarda até que um determinado evento ocorra, enquanto que no ‘deadlock’ o processo aguarda um evento que nunca irá ocorrer.

Em Cho (1993) são descritos os tipos de ‘deadlocks’ em sistemas produtivos como sendo:

- ‘Deadlock’ de Fluxo de Processos (‘Part Flow deadlock’);
- ‘Deadlock’ de Recursos (‘Processing Resource deadlock’);
- ‘Deadlock’ de Materiais (‘Material Handler Deadlock’);

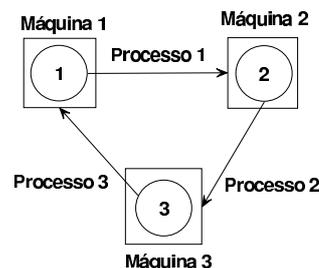


Figura 2. ‘Deadlock’ de Fluxo de Processo.

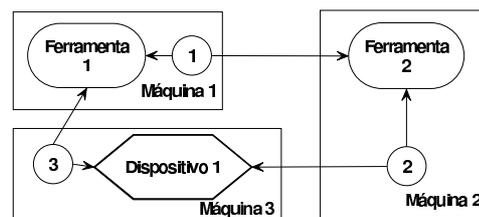


Figura 3. ‘Deadlock’ de Recursos.

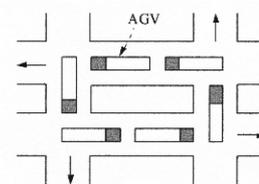


Figura 4. ‘Deadlock’ de Materiais.

Considerando o problema de ‘deadlock’ de fluxo de processos são necessários 4 (quatro) condições para a ocorrência de travamento onde há processos concorrentes (Santos Filho 2000, 1998):

- Mútua exclusão: Cada processo requisita o uso exclusivo de um recurso;
- Retenção enquanto espera: Enquanto aguarda a liberação de um recursos, o processo não libera o recurso alocado por ele;
- Não há preempção: Um recurso poderá ser liberado somente pelo processo que o alocou;
- Espera circular: É um cadeia cíclica fechada de processos aguardando a liberação dos recursos alocados por outros processos;

Neste contexto, para evitar o ‘deadlock’ em um sistema é suficiente garantir-se que pelo menos uma das condições acima nunca irá ocorrer. Em SPFs

inseridos na classe de SED as três primeiras condições são sempre verdadeiras, logo, resta apenas a condição de espera circular para ser controlada. Basicamente existem três métodos para abordar o problema de ‘deadlock’ (Cho, 1993):

- **Prevenir (Prevention).** Na prevenção, um esquema estruturado do sistema garante que pelo menos uma das quatro condições para a ocorrência de ‘deadlock’ nunca ocorrerá. A esquematização é estática e portanto não podem ser alteradas em tempo real.
- **Evitar (Avoidance).** Consiste em uma técnica baseada em algoritmos que supervisionam os processos quanto à alocação de recursos e realiza decisões em tempo real para evitar o ‘deadlock’.
- **Detectar e Restabelecer (Detection and Resolution).** Consiste em uma técnica baseada em algoritmos que detectam o estado de ‘deadlock’ e realizam ações para sair deste estado. Para tirar o sistema do estado de ‘deadlock’, o método para-lisa ou cancela um ou mais processos. Os processos travados podem perder parte ou todas as atividades que estavam sendo realizados durante a ocorrência do travamento.

Os dois primeiros métodos fornecem resultados satisfatórios, porém, ambos possuem um elevado custo de implementação. O último método apresenta o inconveniente de possibilitar a ocorrência de situações em que o tempo de recuperação do sistema, após a detecção de ‘deadlock’, é relativamente longo, comprometendo o desempenho global do sistema. O grau de dificuldade de implementação utilizando-se o primeiro método é proporcional à complexidade do sistema, assim, adota-se neste trabalho a segunda abordagem, o ‘deadlock avoidance’.

### 3 Utilização de Recursos e Sequênciação dos Processos em SPFs

Evitar o ‘deadlock’ é uma forma de controle essencial requerido em operações automatizadas nos sistemas de tecnologia avançada que envolvem um intenso compartilhamento de recursos. Na abordagem ‘deadlock avoidance’ deve-se determinar o estado anterior ao ‘deadlock’ e então tomar uma decisão em relação a ação a ser executada para que o sistema evolua para um estado que não determine o travamento.

#### 3.1 Arquitetura Hierárquica de Controle

A estrutura hierárquica de controle proposta em Santos Filho (Santos Filho, 2000) é uma divisão do controle em níveis semanticamente diferenciados, conforme apresentado na figura 5:

- **Controle de processos:** Tem por objetivo garantir a dinâmica do sequenciamento de atividades inerente a cada processo;

- **Controle de recursos:** Tem por objetivo ampliar o potencial de gerar diferentes estratégias de controle de utilização dos recursos considerando o aspecto de modelagem de sistemas complexos.

O controle orientado a processo atua diretamente sobre o objeto de controle. Por sua vez, em um nível superior situa-se o controle de recursos que atua e impõe restrições no controle de processos e pode monitorar o comportamento dinâmico do objeto de controle.

A estrutura hierárquica proposta possibilita a aplicação do método de ‘deadlock avoidance’ no sistema. Para aplicar o método utiliza-se o modelo de restrições (Santos Filho, 2000), isto é, deve-se descobrir estados anteriores ao ‘deadlock’ e evitá-los.

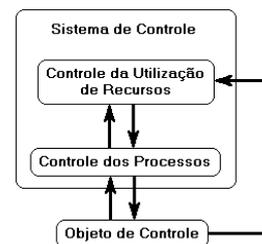


Figura 5. Arquitetura hierárquica de controle de SPFs.

#### 3.2 Modelagem do Sistema de Controle

Os grafos possibilitam analisar e realizar simulações no modelo, fornecendo dados importantes sobre o comportamento dinâmico do sistema. Uma ferramenta eficaz na modelagem, análise e controle de SPFs são as Redes de Petri (Peterson, 1981; Holliday, 1998; Miyagi, 2000). Atualmente uma série de variações de Redes de Petri estão sendo propostos para diferentes propósitos (Holliday, 1998).

Para este trabalho são utilizados as seguintes ferramentas: (i) Enhanced Mark Flow Graph (E-MFG), que é uma derivada da Redes de Petri; (ii) Grafos de Alocação de Recursos (GAR), apresentados a seguir.

##### 3.2.1 Enhanced Mark Flow Graph (E-MFG)

O E-MFG é uma extensão do MFG (Mark Flow Graph) (Miyagi, 2000; Santos Filho, 1998). O MFG é uma rede interpretada derivada de Redes de Petri. Basicamente é composto pelos seguintes elementos estruturais: Transições, Boxes, Arcos orientados, Portas habilitadoras, inibidoras e arcos de sinal de saída (figura 6).

Um elemento externo pode habilitar ou inibir o disparo de uma transição de acordo com o estado do sistema real, ou ainda, enviar alguma informação para o sistema real. É através do elemento externo que é possível a troca de informações entre o sistema de controle e o objeto de controle.

No MFG as marcas fornecem informações booleanas, isto é, um determinado recurso pode estar sendo utilizado ou não, por exemplo. Enquanto que no E-MFG as marcas são individuais e possuem

atributos, isto é, cada marcação representa um determinado processo, por exemplo. O controle das alterações de fluxo é realizado através de sinais de portas provenientes de elementos externos.

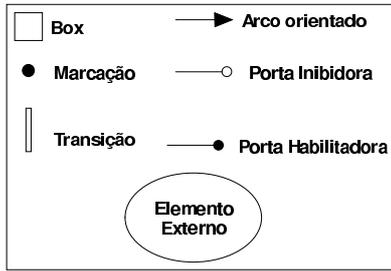


Figura 6. Elementos Estruturais.

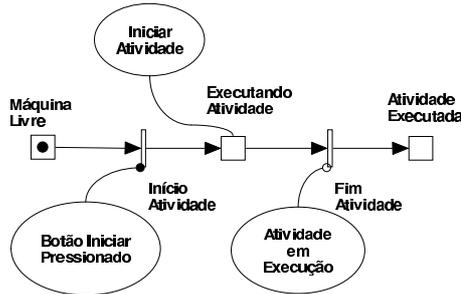


Figura 7. Exemplo de MFG.

### 3.2.2 Grafo de Alocação de Recursos (GAR)

O GAR é um grafo orientado bi-partido constituído por nós e arcos (Santos Filho, 1998; Santos Filho, 2000a; Santos Filho, 2000b). Um par contendo um nó e o respectivo arco de saída é denominado par de alocação e, o par contendo o nó e um arco de entrada é denominado par de requisição, conforme ilustrada na figura 8. O GAR é utilizado para modelar a seqüência de utilização de recursos pelos processos.

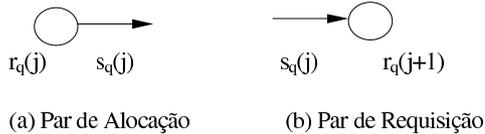


Figura 8. Par de Alocação e Requisição no GAR.

No par de alocação, o nó representa um determinado recurso  $r_q(j)$  e o arco de saída possui uma inscrição que representa a etapa do processo  $s_q(j)$  que aloca  $r_q(j)$  para ser executada; Por sua vez, em um par de requisição o nó também representa um determinado recurso  $r_q(j+1)$  e o arco orientado de entrada possui uma inscrição que representa a etapa do processo  $s_q(j)$  que requisita  $r_q(j+1)$ .

Os ciclos fechados em um GAR são denominados CFE (Ciclos Fechados de Espera) e representa compartilhamentos de recursos. Há dois tipos básicos de ciclos fechados:

- Quando a rota de um determinado processo cruza-se sobre si mesma antes de atingir a etapa final da seqüência de produção;
- Quando as rotas de processos diferentes cruzam-se antes de atingirem as etapas finais de cada seqüência de produção.

Num CFE, define-se:

- Intersecção. Nó correspondente à intersecção é denominado nó de intersecção;
- Par de alocação de intersecção (PAI). Corresponde ao Par de alocação que contém o nó;
- Par de alocação terminal (PAT). Corresponde ao Par de alocação que contém o arco que aponta para o nó de intersecção;
- Seguimento. Conexão contínua de um determinado PAI até o próximo PAI.

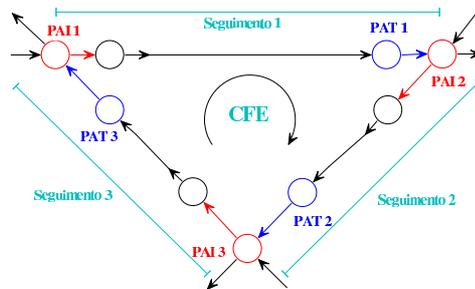


Figura 9. Exemplo de Modelo GAR com CFE.

## 4 'Deadlock Avoidance' em SPFs

### 4.1 Método Proposto

Dado o conjunto de recursos e a seqüência de utilização de recursos pelos processos, os passos do método proposto são:

- (i) Modelagem de cada processo através do GAR;
- (ii) Obtenção do modelo GAR de utilização de recursos através da técnica de fusão de lugares (Santos Filho, 2000a; Santos Filho, 1998).
- (iii) Determinação dos ciclos fechados de espera (CFE) pelo modelo GAR global;
- (iv) Geração das regras de controle para alocação de recursos, baseado na abordagem 'deadlock avoidance' a partir dos ciclos fechados;
- (v) Tradução do GAR global para o E-MFG (Miyagi, 2000; Santos Filho, 1998), inserindo-se as regras de controle.

Os passos (i), (ii) e (iii) são apresentados em Santos Filho, 2000a. Para a determinação dos ciclos fechados, é adotado o método apresentado em Santos Filho, 2000a. A contribuição deste trabalho é focalizado no passo (iv) e (v). Cada um dos passos é detalhado através de um exemplo de aplicação. A seguir, cada passo do método é apresentado detalhadamente no projeto do controle de um SPF exemplo.

Modelagem de cada processo através do GAR. O SPF exemplo possui um conjunto de recursos  $r = \{r_1, r_2, r_3\}$ , um conjunto de processos  $P = \{P_1, P_2, P_3\}$  e a seqüência de utilização dos recursos de cada processo:  $P_1 = \{r_1, r_2\}$ ;  $P_2 = \{r_2, r_3\}$ ;  $P_3 = \{r_3, r_1\}$ . A figura 10 apresenta o GAR dos quatro processos do SPF exemplo.

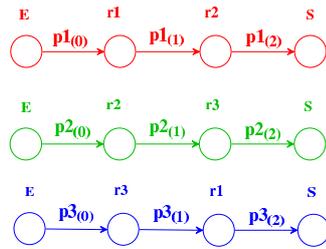


Figura 10. Modelos GAR de cada processo do SPF exemplo.

O processo 1 (P1) utilizará os recursos: r1 e r2. Entre o nó de entrada (E) e o primeiro recurso (nó do recurso r1) constrói o arco p1(0). Após o término da etapa p1(1), o processo p1 requisita o recurso r2, assim, constrói-se o arco p1(1) que liga o nó r1 com o r2 e assim por diante.

O próximo passo é a utilização da técnica de fusão de lugares (Santos Filho, 2000a; Santos Filho, 1998) para criar um modelo global da utilização de recursos (GAR). A figura 11 é uma representação da rede em forma de matriz e a figura 12 é apresentado o modelo GAR Global. Do modelo GAR global, deduz o modelo E-MFG global de utilização de recursos.

$$M = \begin{matrix} & r_1 & r_2 & r_3 \\ \begin{matrix} r_1 \\ r_2 \\ r_3 \end{matrix} & \begin{bmatrix} 0 & P1(1) & 0 \\ 0 & 0 & P2(1) \\ P3(1) & 0 & 0 \end{bmatrix} \end{matrix}$$

Figura 11 Modelo GAR representado em forma de Matriz

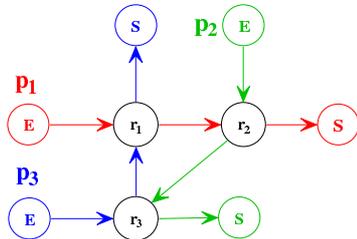


Figura 12. Modelo GAR Global.

Para a determinação dos ciclos fechados, aplica-se o algoritmo para determinar os ciclos fechados conforme em (Santos Filho, 2000b), assim temos o seguinte CFE: (p1(1).p2(1).p3(1)), ilustrado na figura 15. O algoritmo utilizado para determinar os ciclos fechados é apresentado na figura 15.

Este algoritmo baseia-se na potenciação da matriz de adjacência M do grafo GAR. O elemento  $m_{ij}$  da matriz M representa todas as atividades que conectam  $R_i$  e  $R_j$  no modelo GAR. Quando um elemento não nulo é obtido na diagonal principal, a expressão contida neste elemento representa todos os caminhos possíveis que iniciam e retornam para  $R_i$ , indicando a existência de um ciclo fechado.

Para a geração das regras de Controle dos Recursos, utiliza-se o método de ‘deadlock avoidance’ que verifica os possíveis estados de ‘pré-deadlock’ em CFEs, criando-se assim as regras para que o sistema não trave. O estado de ‘pré-deadlock’ em um CFE é caracterizado quando apenas um único recurso

está disponível e os demais estão sendo utilizados por processos que participam do CFE.

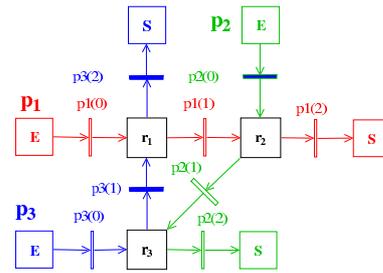


Figura 13. Modelo E-MFG Global

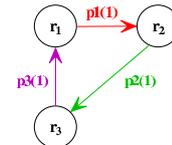


Figura 14. O Ciclo Fechado de Espera (CFE).

Inicializa G (matriz Associada ao GAR)
W = I (Matriz Identidade)
j = N (Ordem da Matriz G)
Enquanto (W<>0) e (j>0)
W = W * G
j = j - 1
i = 0
Enquanto i <= N
Wii <> 0
SIM
acrescenta Wii na lista de CEFs
Elimina Wii da diagonal principal de W
NÃO
i = i + 1

Figura 15. Algoritmo para determinar os CFE.

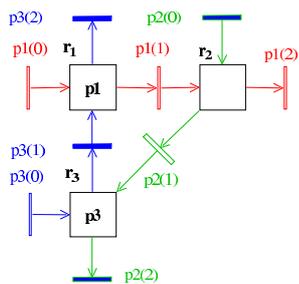
Os processos que podem alocar esse recurso disponível devem pertencer ao mesmo CFE. Então, a ação de controle que deve ser aplicada é de inibir a entrada desse novo processo no ciclo. As regras de controle são elaborados a partir da lista de ciclos fechados e do modelo E-MFG global de utilização de recursos aplicando-se a cada Ciclo Fechado de Espera o seguinte procedimento:

- Determina todos os possíveis estados de ‘pré-deadlock’ no ciclo;
- Para cada estado possível, inibe-se a entrada de processo que participam do CFE no recurso disponível.
- Adiciona-se cada regra obtida à lista de regras.

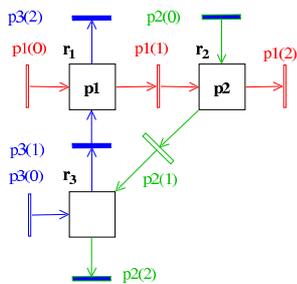
Este procedimento pode ser estruturado em um algoritmo. Propõe-se então, o algoritmo de geração de regras de controle de recursos baseado na abordagem ‘deadlock avoidance’, apresentado na figura 16.

Para cada CFE
Para cada estado de pré-deadlock
Inibir a entrada do processo que participa do CFE no recurso disponível.
Adiciona a regra em uma lista

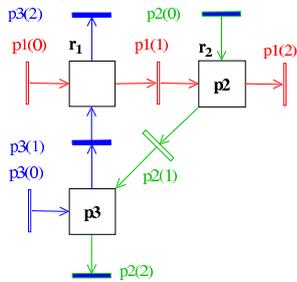
Figura 16. Algoritmo para geração de regras de controle.



IF  $r1=p1$  AND  $r3=p3$  THEN  $p2(0)=FALSE$



IF  $r2=p2$  AND  $r1=p1$  THEN  $p3(0)=FALSE$



IF  $r3=p3$  AND  $r2=p2$  THEN  $p1(0)=FALSE$

Figura 17. Os CFE do exemplo.

#### 4.2 Geração Automática de Regras de Controle

O Algoritmo de geração automática de regras de controle é baseada no método proposto por Santos Filho (Santos Filho, 2000a) que será utilizado em uma ferramenta de auxílio ao projeto de sistemas de controle. Esta ferramenta é composta por módulos. Para o presente trabalho será descrita apenas o módulo de geração das regras. Os dados de entrada da ferramenta são as informações sobre os recursos disponíveis e os processos que utilizarão estes recursos. O método é aplicado sobre as informações e através de simulação, a ferramenta fornecerá um conjunto de resultados para análise e avaliação do sistema em estudo.

### 5 Conclusão

A sistematização do procedimento de modelagem da utilização dos recursos e do procedimento de obtenção de regras de controle de recursos é o primeiro passo na geração automática dos modelos de controle de SPFs. Está em desenvolvimento um protótipo da ferramenta de auxílio ao projeto de controle de SPF que objetiva a geração dos modelos de controle e sua análise por meio da simulação que aplica o método proposto.

Este trabalho apresenta a sistematização do método proposto viabilizando a geração automática de regras de controle para alocação de recursos.

### Agradecimentos

Os autores agradecem a CAPES e CNPq pelo apoio a este trabalho.

### Referências Bibliográficas

- Cho, H., (1993). "An Intelligent Workstation Controller for Computer Integrated Manufacturing", Doctor Dissertation, Texas A&M University, pp.83-88.
- Banaszak, Z.A., Krogh, B.H., (1990), "Deadlock Avoidance in Flexible Manufacturing Systems with Concurrently Competing Process Flows", IEEE Transactions on Robotics and Automation, vol.6, no.6.
- Holliday, J.L., Abbadi, A.E., (1998) "Distributed Deadlock Detection", University of California at Santa Barbara.
- Miyagi, P.E., Maruyama, N., Santos Filho, D.J., (2000), "An Anthropocentric Approach for the Design of Production Systems Control", Proceeding of Mechatronics 2000. 7<sup>th</sup> Mechatronics Forum International Conference, Atlanta.
- Miyagi, P.E., (1996), "Controle Programável – Fundamentos do Controle de Sistemas a Eventos Discretos", Ed. Edgard Blücher Ltda, São Paulo, Brazil, pp. 1-5, 71-148.
- Peterson, J.L., (1981), "Petri Net Theory and the Modeling of System", Englewood Cliffs, N.J., Prince-Hall, pp.1-78.
- Santos Filho, D.J., (2000a). "Aspectos do Projeto de Sistemas Produtivos", Tese (Livre Docência), Universidade de São Paulo – Escola Politécnica – Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos, pp.8-21, 61-95, 101-111.
- Santos Filho, D.J., Motohashi Matsusaki, C.T., Nelli Silva, E.C., Miyagi, P.E., (2000b), "Automatic Generation of Metamodels to Design Production Systems Control", Proceeding of Mechatronics 2000. 7<sup>th</sup> Mechatronics Forum International Conference, Atlanta.
- Santos Filho, D.J., (1998). "Controle de Sistemas Antropocêntricos de Produção Baseados em Redes de Petri Interpretadas", Tese (Doutorado), Universidade de São Paulo – Escola Politécnica – Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos, pp. 84-119.
- Viswanadham, N., Narahari, Y., Johnson, T.L., (1990), "Deadlock Prevention and Deadlock Avoidance in Flexible Manufacturing Systems Using Petri Net Models", IEEE Transactions on Robotics and Automation, vol.6, no.6.