

DANIEL BALIEIRO SILVA

**APLICAÇÃO DA TEORIA DE CONTROLE
SUPERVISÓRIO NO PROJETO DE
CONTROLADORES PARA SISTEMAS DE
ROTA VARIÁVEL CENTRADO EM ROBÔ
PPGEPS**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção e Sistemas da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Engenharia de Produção e Sistemas.

CURITIBA
Setembro 2007

DANIEL BALIEIRO SILVA

**APLICAÇÃO DA TEORIA DE CONTROLE
SUPERVISÓRIO NO PROJETO DE
CONTROLADORES PARA SISTEMAS DE
ROTA VARIÁVEL CENTRADO EM ROBÔ
PPGEPS**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção e Sistemas da Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Engenharia de Produção e Sistemas.

Área de Concentração: *Automação e Sistemas.*

Orientador: Prof. Dr. Eduardo Alves Portela Santos

Co-orientador: Prof. Dr. Marco A. Buseti de Paula

Prof. Dr. Agnelo Denis Vieira

CURITIBA

2007

Silva, Daniel Balieiro

APLICAÇÃO DA TEORIA DE CONTROLE SUPERVISÓRIO NO PROJETO DE CONTROLADORES PARA SISTEMAS DE ROTA VARIÁVEL CENTRADO EM ROBÔ. Curitiba, 2007. 147p.

Dissertação – Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Engenharia de Produção e Sistemas.

1. Sistema a Eventos Discretos 2. Autômatos de Estados Finitos 3. Teoria de Controle Supervisório 4. Controladores Lógico Programáveis. I. Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e de Tecnologia. Programa de Pós-Graduação em Engenharia de Produção e Sistemas.

À minha esposa Fátima e minha filha Daniela, pelo permanente incentivo e
compreensão nos momentos em que ficamos distantes.

Razões para que eu não desista nunca.

À minha mãe Adélia e à meu pai Guilhermino (*in memoriam*)
por me darem o exemplo de luta, dedicação e perseverança.

À todas as minhas irmãs e irmãos, por seguirem
o exemplo de meus pais e acreditarem
que os sonhos se realizam através da luta.

Agradecimentos.....	ii
Sumário.....	iii
Lista de Figuras.....	v
Lista de Tabelas.....	viii
Lista de Símbolos.....	ix
Lista de Abreviaturas.....	xi
Resumo.....	xii
Abstract.....	xiii

Agradecimentos

À Pontifícia Universidade Católica do Paraná pela oportunidade de atuar como aluno bolsista do Programa de Pós-graduação. O meu respeito por esta grande e importante Universidade.

À todos os Professores do PPGEPS pela oportunidade oferecida e por acreditarem na realização do trabalho.

À Universidade Tecnológica Federal do Paraná (UTFPR) pelo apoio e ao contribuinte brasileiro.

Faço um agradecimento especial aos Professores: Prof^o. Dr. Marco Antônio Buseti, Prof^o. Dr. Eduardo Alves Portela Santos e Prof^o. Dr. Agnelo Denis Vieira, pelo apoio incondicional, compreensão, paciência e conhecimentos transmitidos. Levo-os como exemplo para a minha nova etapa profissional como professor.

Ao Prof^o. Luiz Fernando Brum Teixeira pelo apoio nos trabalhos de programação do robô Comau. Um grande companheiro.

À todos os meus colegas de mestrado e laboratório (Ricardo Diogo, Nilson e Jairo), que de forma direta ou indireta contribuíram na elaboração deste trabalho.

À todos os meus amigos que vivenciaram todas as dificuldades e sempre me incentivaram nesta longa e difícil jornada.

Sumário

Capítulo 1	i
Introdução	1
1.1 Objetivos do Trabalho	3
1.2 Objetivos Específicos do Trabalho	3
1.3 Metodologia do Trabalho	3
1.4 Justificativas do Trabalho	4
1.5 Organização do Trabalho	7
Capítulo 2	9
Sistemas de Manufatura Flexível	9
2.1 Sistemas de Manufatura	9
2.2 Classificação dos Sistemas de Manufatura	12
2.3 Sistemas de Manuseio e Transporte de Material	14
2.4 Flexibilidade dos Sistemas de Manufatura	19
2.5 Conclusão ao capítulo	28
Capítulo 3	30
Teoria de Controle Supervisório	30
3.1 Teoria de Linguagens	31
3.2 Operações sobre linguagens	31
3.3 Teoria de Autômatos	32
3.4 Linguagens Geradas e Marcadas	35
3.5 Operações sobre Autômatos	36
3.5.1 Acessibilidade	37
3.5.2 Co-acessibilidade	38
3.5.3 Operação Trim	38
3.5.4 Composição Paralela	39
3.6 Teoria de Controle Supervisório	40
3.6.1 Supervisão Centralizada (Controle Monolítico)	41
3.6.1.1 Controlabilidade	44
3.6.1.2 Problema de Controle Supervisório	45
3.6.2 Supervisão Modular	45
3.7 Conclusão ao capítulo	49
Capítulo 4	51
Modelagem do Sistema Físico	51
4.1 Configuração Generalizada do Sistema Físico	51
4.2 Caso específico a modelar	53
4.3 Análise do Primeiro Modelo para o Sistema de Transporte	55
4.4 Análise do Segundo Modelo para o Sistema de Transporte	58
4.5 Análise do Terceiro Modelo para o Sistema de Transporte	59
4.6 Análise do Quarto Modelo para o Sistema de Transporte	60
4.7 Conclusão ao capítulo	62
Capítulo 5	66
Modelagem das Especificações de Controle e Síntese dos Supervisores	66
5.1 Especificações de Controle Generalizadas	67
5.2 Especificações de Controle do Caso Específico	74
5.3 Síntese dos Supervisores	80
5.4 Conclusão ao capítulo	85

Capítulo 6	87
Implementação do Sistema de Controle Supervisório	87
6.1 Introdução.....	87
6.2 Arquitetura de Controle Supervisório.....	88
6.3 Modelo para Implementação da Arquitetura de Controle Supervisório.....	90
6.4 Obtenção dos SFC's g_i no conjunto $\{g_i \mid i \in I\}$ e funções lógicas no conjunto $\{dg_i \mid i \in I\}$	94
6.4.1 Definição das variáveis do CLP	95
6.4.2 Conversão dos Autômatos G_i em Autômatos H_i	96
6.4.3 Obtenção dos SFCs g_i correspondentes aos Autômatos H_i	101
6.4.4 Obtenção da Função lógica dg_i	106
6.5 Obtenção dos SFC's $\{s_j \mid j \in J\}$ correspondentes aos Supervisores Reduzidos.	107
6.5.1 Definição das variáveis do CLP	108
6.5.2 Obtenção dos SFC $s_j = (X^{s_j}, T^{s_j}, x 0^{s_j})$, $j \in J$	109
6.6 Obtenção dos SFC's $\{o_\sigma \mid \sigma \in \Sigma\}$ correspondentes aos Procedimentos Operacionais .	115
6.7 Conclusão ao capítulo.....	118
Capítulo 7	120
Conclusões	120
Referências Bibliográficas	125
Apêndice	129
Modelagem de Outras Seqüências de Processamento.....	129
A.1 Seqüência de Processamento com Linha de Transferência	129
A.2 Seqüência de Processamento com Divergência e Convergência de Estações.....	135
A.3 Seqüência de Processamento com Divergência de Estações.....	139
A.4 Seqüência de Processamento com Retrabalho.....	142

Lista de Figuras

Figura 2.1: Composição de um sistema de produção. Groover (2001).....	10
Figura 2.2: Tipos de rotas em sistemas de manufatura com múltiplas estações, (a) rotas variáveis e (b) rotas fixas. (Groover, 2001).....	11
Figura 2.3: Classificação dos Sistemas de Manufatura quanto ao nível de automação. (Groover, 2001).....	13
Figura 2.4: Leiaute em linha com rotas fixas Groover (2001).	15
Figura 2.5: Leiaute em linha com rotas flexíveis Groover (2001).	16
Figura 2.6: Leiaute em circuito fechado formato elíptico Groover (2001).	16
Figura 2.7: Leiaute em circuito fechado formato retangular Groover (2001).	17
Figura 2.8: Leiaute escada Groover (2001).	18
Figura 2.9: Leiaute de Célula Centrada em robô Groover (2001).....	19
Figura 2.10: a) Níveis de flexibilidade de sequenciamento NFS = 0, b) NFS = 1, c) NFS= 2.21	
Figura 2.11: Flexibilidades na manufatura de uma peça.	22
Figura 2.12a: Processo seqüencial, concorrência zero.	25
Figura 2.12b: Sobreposição de processos, concorrência parcial.	25
Figura 2.12c: Processos Paralelos, concorrência máxima.	26
Figura 2.13: Sistema de controle e flexibilidade dos processos.	26
Figura 2.14: Hierarquia da flexibilidade nos sistemas.	28
Figura 3.1: Exemplo de um autômato determinístico.	33
Figura 3.2: Autômato que marca a linguagem consistindo de todas as palavras iniciadas por a e b seguidas por a	36
Figura 3.3: Autômato não-acessível e não co-acessível	36
Figura 3.4: Autômato acessível.	37
Figura 3.5: Autômato Trim.....	38
Figura 3.6: Autômatos G_1 e G_2	39
Figura 3.7: Autômato $G_1 G_2$	40
Figura 3.8: Arquitetura do controle monolítico (Ramadge e Wonham, 1989).....	43
Figura 3.9: Arquitetura de controle modular local (Queiroz e Cury, 2000).....	47
Figura 4.1: Sistema automatizado com rotas variáveis, centrado em robô (Adaptação de Groover, 2001).	52
Figura 4.2: Sistema Flexível de Manufatura Centrado em Robô, caso específico para $n=3$	53
Figura 4.3: Modelo das Máquinas M_1 , M_2 e M_3	54
Figura 4.4: Modelo da esteira transportadora G_4	54
Figura 4.5: G_{ST1} , primeiro modelo candidato ao sistema de transporte.....	55
Figura 4.6: E_{ST1} , especificação de rotas.....	56
Figura 4.7: G_{ST2} , segundo modelo candidato ao sistema de transporte.....	58
Figura 4.8: G_{ST3} , terceiro modelo para o sistema de transporte.....	59
Figura 4.9: G_{ST4} , quarto modelo para o sistema de transporte.....	61
Figura 4.10: E_{ST2} , especificação de rotas.....	62
Figura 5.1: Especificação generalizada do primeiro correspondente à um <i>buffer</i> de saída.....	67
Figura 5.2: Especificação generalizada do primeiro conjunto para um <i>buffer</i> de entrada.	68
Figura 5.3: Especificação generalizada do segundo conjunto.....	69

Figura 5.4: Especificação generalizada do quarto conjunto.....	71
Figura 5.5: Especificações do primeiro conjunto: a) E_0 , b) E_1 , c) E_2 , d) E_3 , e) E_4 , f) E_5 , g) E_6	74
Figura 5.6: Especificações do segundo conjunto E_7 , E_8 , E_9 e E'_9	77
Figura 5.7: Especificação do terceiro conjunto para o sistema de transporte, E_{10}	78
Figura 5.8: Especificação do quarto conjunto, para a coordenação conjunta da esteira, máquina M_1 e sistema de transporte, E_{11}	79
Figura 5.9: Estrutura de transição de estado dos supervisores reduzidos S_0 a S_9	82
Figura 5.10: Estrutura de transição de estados do supervisor reduzido S_{10}	83
Figura 5.11: Estrutura de transição de estados do supervisor reduzido S_{11}	83
Figura 6.1: Arquitetura de Controle Supervisório (Queiroz e Cury, 2002).....	89
Figura 6.2: Programa principal SFC Main (Vieira <i>et al.</i> , 2007).....	91
Figura 6.3: Autômato H_4 correspondente ao módulo G_4	98
Figura 6.4: Autômato H_1 correspondente ao módulo G_1	100
Figura 6.6: Autômato H_3 correspondente ao módulo G_3	100
Figura 6.8: SFC g_1 da máquina M_1	104
Figura 6.9: SFC g_2 da máquina M_2	105
Figura 6.10: SFC g_3 da máquina M_3	105
Figura 6.11: Função lógica $dg1$	106
Figura 6.12: Função lógica $dg2$	107
Figura 6.13: Função lógica $dg3$	107
Figura 6.14: Função lógica $dg4$	107
Figura 6.15: Função lógica $dg0$	107
Figura 6.16: Função de desabilitação de evento controlável $am1ad$	109
Figura 6.17: SFC s_0	111
Figura 6.18: SFC s_1	111
Figura 6.19: SFC s_2	111
Figura 6.20: SFC s_3	112
Figura 6.21: SFC s_4	112
Figura 6.22: SFC s_5	112
Figura 6.23: SFC s_6	112
Figura 6.24: SFC s_7	113
Figura 6.25: SFC s_8	113
Figura 6.26: SFC s_9	113
Figura 6.27: SFC s_{10}	114
Figura 6.28: SFC s_{11}	114
Figura 6.29: Simulação das máquinas M_t	115
Figura 6.30: SFC om_1	116
Figura 6.31: SFC om_2	116
Figura 6.32: SFC om_3	116
Figura 6.33: SFC oes	117
Figura 6.34: SFC $oa23$	118
Figura A.1: Modelo do Sistema de Transporte para linha de transferência.....	130
Figura A.2: Modelo das máquinas M_t , autômatos G_1 , G_2 e G_3	130
Figura A.3: Modelo da esteira transportadora de entrada G_4	131

Figura A.4: Especificações dos <i>buffers</i> B_0, B_1, B_2, B_3, B_4 e B_5	131
Figura A.5: Especificação E_6	132
Figura A.6: Especificação E_7	133
Figura A.7: Especificação E_8	133
Figura A.8: Especificação E_9	134
Figura A.9: Autômatos G_1, G_2 e G_3 das máquinas M_t	135
Figura A.10: Modelo do sistema de transporte $G_0, PA(M1M3)$ e $PB(M2M3)$	136
Figura A.11: Especificações do primeiro e segundo conjuntos.....	137
Figura A.12: Especificação do sistema de transporte $PA(M1M3)$ e $PB(M2M3)$	138
Figura A.13: Modelo do sistema de transporte $PA(M1M2)$ e $PB(M1M3)$	139
Figura A.14: Modelos das máquinas $PA(M1M2)$ e $PB(M1M3)$	140
Figura A.15: Especificações de controle processo $PA(M1M2)$ e $PB(M1M3)$	140
Figura A.16: Especificação do quarto conjunto $PA(M1M2)$ e $PB(M1M3)$	141
Figura A.18: Modelos dos autômatos G_1, G_2, G_3 e G_4 , para o processo com retrabalho.	143
Figura A.19: Modelo do sistema de transporte G_0 para o processo com retrabalho.	143
Figura A.20: Especificações do primeiro e segundo conjuntos para o processo de retrabalho.	144
Figura A.21: Especificação do terceiro conjunto, E_8	145
Figura A.22: Especificação do quarto conjunto para 3 peças.....	147

Lista de Tabelas

Tabela 2.1: Equipamentos utilizados nos sistemas de manuseio primário de material.....	15
Tabela 5.1: Generalização da especificação do quarto conjunto.....	72
Tabela 5.2: Semântica dos estados para a Especificação E_{11}	80
Tabela 5.3 – Resultados da abordagem modular local.....	81
Tabela 5.4 – Ação de controle dos supervisores reduzidos.....	84
Tabela 6.1 - Program Organization Units (POU).....	93
Tabela A.1 – Resultados da abordagem modular local.....	134
Tabela A.2 – Resultados da abordagem modular local.....	138
Tabela A.3 – Resultados da abordagem modular local.....	142
Tabela A.4 – Resultados da abordagem modular local.....	146

Lista de Símbolos

ε	Palavra vazia
L	Linguagem
\bar{L}	Prefixo-fechamento de L
s, t, u, v	Cadeias de eventos
\in	Pertence a
\notin	Não pertence a
Σ	Alfabeto de eventos
Σ^*	Conjunto de todas as cadeias finitas compostas dos elementos de Σ
\subseteq	Está contido em
\exists	Existe
\cup	União de conjuntos
\cap	Intersecção de conjuntos
\emptyset	Conjunto vazio
δ, ξ	Função de transição de estados
δ_c	Função de transição de estados estendida
X, Q	Conjunto de estados
x, y, z	Estados
$\alpha, \beta, \gamma, \mu$	Eventos
X_m, Q_m	Conjunto de estados marcados
G	Gerador
X_0	Conjunto de estados iniciais
x_0, q_0	Estado inicial
Γ	Função de eventos ativos
$\Gamma(x)$	Função de eventos ativos para o estado x
f	Supervisor
γ	Padrão de controle
$L_m(G)$	Linguagem marcada por G
$L(G)$	Linguagem gerada por G
$ $	Tal que
\parallel	Operação de composição paralela de autômatos (produto síncrono)
Σ_u	Conjunto dos eventos não-controláveis
Σ_c	Conjunto dos eventos controláveis
f/G	G sob a supervisão de f
Φ	Função que mapeia as entradas de controle para cada estado de S
T	Autômato do supervisor

x', q'	Estados futuros
Trim	Autômato acessível e co-acessível
$\sup C(M, G)$	Máxima linguagem contida em M e controlável em relação a G
$C(M, G)$	Classe de linguagens controláveis em relação a G .
K	Linguagem
\neq	Diferente
L_c	Linguagem controlável
i, j	Conjunto de índices
E_{xj}	Especificações locais
G_{xj}	Plantas locais
\parallel_{as}	Produto assíncrono de linguagens
!	A função de transição de estados é definida
$\neg!$	A função de transição de estados não é definida

Lista de Abreviaturas

AGV	Veículo Auto Guiado
CLP	Controlador Lógico Programável
FMS	Sistema Flexível de Manufatura
FS	Flexibilidade de Sequenciamento
GRAIL	Nome do Software com interface gráfica para simulação de SED
P_i	Processo
RSP	Representação Sistema Produto
SED	Sistema a Eventos Discretos
SFC	Sequential Function Chart
NFS	Nível de Flexibilidade de Sequenciamento
TCS	Teoria de Controle Supervisório
TCT	Nome do Software para simulação de SED
GRAF CET	Gráfico Funcional de Comandos Etapas e Transições

Resumo

Os modernos sistemas de manufatura se caracterizam pela fabricação de uma extensa de linha de produtos o que os identificam como sistemas multi-produtos. Sendo assim, tais sistemas são marcados por um alto grau de flexibilidade e complexidade presente em seus processos. A flexibilidade advém do elevado número de tarefas possíveis de serem realizadas e a complexidade é resultado das ações de coordenação que devem ser estabelecidas para se atingir os objetivos de produção. Dentre as ações de coordenação em sistemas multi-produtos estão os roteiros flexíveis de produção a serem desenvolvidos pelo sistema de transporte, de acordo com cada família de produto, para processamento nas diferentes estações de trabalho. Os gargalos de produção estão cada vez mais sendo transferidos dos processos produtivos para os sistemas de controle que coordenam as ações entre os subsistemas como, por exemplo, máquinas e sistemas de transporte. Os processos para desenvolvimento da lógica de controle em tais casos são fundamentalmente intuitivos. A utilização de um formalismo no projeto e re-projeto da lógica de controle facilita as tarefas de modificação, manutenção e identificação de erros pelos operadores e projetistas. Este trabalho tem por objetivo a implementação de uma abordagem metodológica que consiste da utilização de ferramentas formais para o projeto lógico de controladores como a Teoria de Controle Supervisório de acordo com a abordagem Modular Local. Esta metodologia compõe-se de três etapas: a primeira consiste da modelagem dos subsistemas e especificações de controle, a segunda etapa consiste da síntese dos supervisores através de ferramentas computacionais, a terceira etapa constitui-se da simulação e implementação, em sistemas reais, das estruturas de controle em Controladores Lógicos Programáveis Industriais. O modelo de implementação da Arquitetura de Controle Supervisório em CLP's está de acordo com Vieira *et al.* (2007).

Palavras-Chave: Sistemas a Eventos Discretos, Autômatos de Estados Finitos, Teoria de Controle Supervisório, Controladores Lógico Programáveis.

Abstract

Modern manufacturing systems are characterized by the fabrication of an extensive amount of products, which identifies them as multi-product systems. These systems are marked by a high degree of flexibility and complexity in their processes. Flexibility comes from the high number of possible tasks, and the complexity is a result of coordination actions which should be established to obtain the desired production goals. Amongst coordination actions in multi-product systems are flexible production routes to be supported by the transport system, in accordance to each product family, for processing in the different workstations. The production bottlenecks are being transferred from the production process itself to the control systems that coordinate the actions between subsystems, for example machinery and transport systems. The processes for control logic development in these cases are basically intuitive. Using formalisms in the project and re-project of the control logic facilitates modification tasks, maintenance and error identification by the operators and process designers. This work has as an objective the implementation of a methodological approach which consists using formal tools for the logical project of controllers as Supervisory Control Theory. This methodology is composed by three steps: First step consists on subsystem and specification modeling, the second step consists on supervisor synthesis through computer tools, and finally the third step consists on simulation and implementation, in real world systems of the control structures in industrial Programmable Logic Controllers. The implementation model on the Supervisory Control Architecture in PLCs is developed according to Vieira *et al.*(2007).

Keywords: Discret Event Systems, Finite State Automata, Supervisory Control Theory, Programmable Logic Controller

Capítulo 1

Introdução

A evolução da sociedade impõe ao ser humano uma nova forma de observar os processos e buscar melhorias para tudo aquilo que o cerca. A crescente escassez de recursos e as exigências nos padrões de qualidade, obrigam os gestores de processo a alcançarem maiores níveis de eficiência sejam estes relacionados aos produtos ou aos seus processos de fabricação.

A elevada competitividade e diminuição no ciclo de vida dos produtos fazem com que a racionalidade dos processos seja um fator de vantagem competitiva quando se quer atingir altas taxas de produção, baixo consumo de insumos energéticos e matéria-prima, alta confiabilidade e segurança nas operações e instalações produtivas, bem como melhorias nos prazos de entrega e desenvolvimento de projetos e produtos (Miyagi, 1996) e (Moraes e Castrucci, 2001).

Os modernos sistemas de manufatura são um exemplo de tais necessidades, uma vez que o elevado número de produtos com os quais operam e a complexidade das operações desenvolvidas, como por exemplo, pelos sistemas de transporte, lhes conferem uma importância maior. A complexidade dos sistemas de transporte advém das várias possibilidades de movimentação para o transporte de peças ou produtos entre as estações de processamento, podendo ser realizado através de atuadores pneumáticos, robôs ou veículos auto-guiados (AGVs).

O projeto lógico para a correta coordenação entre o sistema de transporte e os vários subsistemas presentes no processo exige um grande esforço na sua elaboração. Sendo assim, a observação e estudo dos processos possibilitam a formulação matemática do seu comportamento com vias a entendê-los e aprimorá-los, de forma que os resultados possam ser utilizados em seu benefício com o objetivo de se atingir maior precisão e qualidade associadas aos produtos (Geromel e Palhares, 2004).

Pelas razões anteriormente expostas e de acordo com Dorf (2001), o estudo e a prática da engenharia de controle possibilitam a modelagem dos sistemas de manufatura com propósitos

voltados à otimização dos tempos de projeto lógicos, com a conseqüente melhoria nos tempos de processamento através da correta coordenação das tarefas a serem desenvolvidas entre os diferentes subsistemas constituintes. Esta redução de tempos e insumos na execução dos produtos e processos é alcançada com as constantes inovações tecnológicas. Os produtos são caracterizados por sua atração, recursos e funcionalidade; e os processos por equipamentos e dispositivos com elevado grau de inteligência tornando-os mais flexíveis e capazes de realizarem um grande número de atividades ou tarefas. Os processos de projeto lógico têm a sua melhoria através da utilização de ferramentas formais e softwares.

Quando se observa uma grande variedade de produtos, torna-se necessário o compartilhamento dos recursos de manufatura, e a execução plena e correta das atividades é função da coordenação entre estes recursos. Esta função de coordenação é executada pelo sistema de controle que, por sua vez, é projetado por profissionais que, em sua grande maioria, não fazem uso de ferramentas que possibilitem uma estruturação do projeto, de modo a ganhar-se tempo em sua elaboração e facilitar as atividades de manutenção e re-configuração dos sistemas (Gouyon *et al.*, 2004).

As estratégias presentes na realização da lógica de controle são fundamentalmente intuitivas e baseadas em processos que possuam algum grau de semelhança. Esta forma de tratamento dificulta a re-configuração da lógica de controle dos sistemas quando da fabricação de um novo produto, sendo necessária a alteração da lógica como um todo, mesmo que a mudança esteja restrita a uma pequena parte do processo (Chandra *et al.*, 2003) e (Hasdemir *et al.* 2004).

No desenvolvimento dos projetos atuais, a complexidade para a elaboração da lógica de controle é proporcional ao número de subsistemas envolvidos. Os processos de modelagem do comportamento dos subsistemas, quando existem, são realizados com dificuldade e os controladores obtidos são elementos únicos, e isto os tornam de difícil interpretação em função do grande número de informações que agregam. Do mesmo modo, a aplicação de tais procedimentos ao ambiente industrial moderno é prejudicada em razão de não atender à todos os aspectos de segurança e controle.

A Teoria de Controle Supervisório (TCS) formulada por Ramadge e Wonham (1987) para os Sistemas Automatizados de Manufatura, auxilia no projeto de controladores através da utilização de ferramentas formais e computacionais, o que garante maior confiabilidade na realização das regras de controle. Uma adequada estruturação na síntese dos elementos de

controle surgiu com a abordagem Modular Local (Queiroz e Cury, 2000). Esta abordagem possibilita a obtenção de vários elementos de controle menos complexos, uma vez que as informações contidas nos mesmos, refere-se à uma pequena parte do processo. A utilização destes elementos de controle menores oferece mais clareza das informações, o que facilita interpretações.

1.1 Objetivos do Trabalho

Este trabalho tem por objetivo aplicar a Teoria de Controle Supervisório, abordagem Modular Local (Queiroz e Cury, 2000), no projeto de controladores para sistemas automatizados de manufatura com rotas variáveis de produção e compartilhamento de recursos como os sistemas de transporte centrado em robô.

1.2 Objetivos Específicos do Trabalho

- Modelar o comportamento dos subsistemas envolvidos;
- Modelar as especificações de controle;
- Estabelecer uma generalização das especificações de controle;
- Sintetizar os supervisores modulares;
- Implementar uma Arquitetura de Controle Supervisório de acordo com o Modelo Queiroz e Cury (2002);
- Implementar esta Arquitetura de Controle em Controladores Lógicos Programáveis (CLP's) conforme o Modelo Vieira *et al.* (2007).

1.3 Metodologia do Trabalho

A metodologia de trabalho a ser utilizada para a implementação desta abordagem se caracteriza por um ciclo de desenvolvimento composto de três etapas: a primeira consiste da modelagem dos subsistemas e especificações de controle, a segunda constitui-se da síntese dos supervisores, e finalmente a terceira etapa é a implementação das estruturas de controle em tecnologias de manufatura como Controladores Lógicos Programáveis.

Como forma de caracterizar estas três etapas tem-se que na primeira etapa deve-se primeiramente buscar o modelo que representa o comportamento desejado para cada subsistema e, em segundo, definir as especificações de controle que determinam a coordenação entre os mesmos. Possíveis alterações neste comportamento podem estar restritas à apenas um subsistema, como por exemplo o sistema de transporte, não sendo necessária a alteração dos outros. Uma abordagem mais estruturada no projeto destes sistemas facilitaria de forma considerável as atividades de re-projeto.

A segunda etapa que corresponde à síntese dos supervisores, verifica o correto comportamento dos subsistemas sujeitos às especificações de controle, através da obtenção dos elementos de controle chamados supervisores. O objetivo deste procedimento é eliminar existência de bloqueios garantindo a controlabilidade das linguagens que determinam a operação contínua dos sistemas, assim como obter um supervisor que seja de fácil implementação e alteração na lógica de controle. Na terceira etapa, implementam-se todas as estruturas de controle, obtidas nas duas etapas anteriores, em controladores lógicos programáveis, tendo como objetivo principal a facilitação de futuras modificações quando forem alteradas as regras de comportamento do sistema.

1.4 Justificativas do Trabalho

A utilização de uma abordagem metodológica para a integração dos sistemas de manufatura, confere aos projetos maior confiabilidade, segurança, eficiência e objetividade no desenvolvimento dos trabalhos. Como parte dos objetivos está a elaboração de uma biblioteca de modelos dos subsistemas e especificações de controle, que atenda às necessidades de modificação. Como um importante exemplo de aplicação da abordagem Modular Local (Queiroz e Cury, 2000) cita-se o caso de sistemas multi-produtos em que a inserção de um novo produto determina a re-configuração para as novas rotas do sistema, ou uma nova operação a ser realizada por uma das máquinas. Algumas destas modificações são de difícil solução quando não se faz uso de uma estruturação na lógica de controle, advinda da utilização de ferramentas formais de projeto.

Como os modernos equipamentos cada vez mais são dotados de recursos próximos à inteligência artificial, os gargalos de produção não mais se limitam à quantidade de equipamentos ou às tarefas que estes podem desempenhar. Desta forma, os gargalos são

transferidos aos sistemas de controle que coordenam as operações entre os subsistemas, e dentre estas operações encontram-se os roteiros de produção (Wadhwa *et al.* 2005).

Elaborar uma lógica de controle mais estruturada e que facilite alterações futuras, principalmente relacionadas às mudanças dos roteiros de produção é de fundamental importância para os propósitos de re-configurabilidade dos sistemas. Estas rotas, caso mal projetadas, podem conduzir o processo à um estado crítico de bloqueio, interrompendo de forma contínua a evolução dos sinais, e por conseqüência a conclusão das tarefas.

As alterações na lógica de controle são tarefas bastante trabalhosas quando o desenvolvimento segue as estratégias heurísticas atualmente utilizadas. Sendo assim, a não utilização de técnicas baseadas em ferramentas formais para o projeto lógico de controladores, conduz a dificuldades no entendimento, manutenção, alteração e detecção de erros no programa de controle, sendo de grande importância aqueles relacionados à segurança operacional do processo (Chandra *et al.*, 2003).

Este trabalho se justifica em razão da grande evolução, importância e crescimento dos processos classificados como Sistemas a Eventos Discretos (SED). É uma linha de pesquisa relativamente nova se considerarmos os estudos realizados a respeito dos já tradicionais processos de variáveis contínuas. Neste sentido tem ocorrido o desenvolvimento de muitos trabalhos voltados ao estudo e análise de SED's que abordam principalmente os sistemas de rotas fixas sem, no entanto, considerar os aspectos de implementação em tecnologias industriais.

Observa-se também na literatura acadêmica e trabalhos de pesquisa, uma abordagem pouco voltada ao tratamento do compartilhamento de recursos em processos concorrentes com propósitos de implementação tomando como base o formalismo a ser utilizado neste trabalho. Sendo assim, a Teoria de Controle Supervisório (TCS) desenvolvida por (Ramadge e Wonham, 1987) é particularmente adequada para estes propósitos, já que o processo para a obtenção das estruturas de controle é automático, e o procedimento de síntese determina que as especificações de controle sempre serão atendidas (Kumar *et al.*, 2005).

A principal contribuição deste trabalho diz respeito ao processo de implementação de uma Arquitetura de Controle Supervisório em Controladores Lógicos Programáveis, utilizados nas modernas instalações industriais, para os sistemas automatizados com rotas variáveis e compartilhamento dos recursos de fabricação para processos concorrentes.

Os estudos serão desenvolvidos considerando o sistema operando sob condições normais, uma vez que agregando estados representativos de falhas o esforço computacional torna-se

substancialmente maior, inviabilizando o cálculo para a síntese dos supervisores. No entanto, um primeiro direcionamento aos leitores que desejam estudos abordando os eventos de falhas em Sistemas a Eventos Discretos considerando a TCS pode ser encontrado em Yalcin (2004). O aspecto de análise sob condições normais ou anormais deve ser levado em consideração na realização de todos os trabalhos de pesquisa em Sistemas a Eventos Discretos envolvendo a Teoria de Controle Supervisório, uma vez que a explosão do número de estados é um fator importante para a viabilidade de uma implementação.

Conforme relatado o propósito maior deste trabalho é a modelagem, implementação e simulação de um sistema flexível de manufatura hipotético abordando os roteiros flexíveis de produção. Uma revisão bibliográfica do contexto atual dos trabalhos indica que tais roteiros de produção e os diferentes tipos de flexibilidades associadas aos processos, têm sido intensamente estudados sob o ponto de vista de avaliação de desempenho, considerando o aspecto quantitativo da engenharia de produção nos processos produtivos. Estes estudos possibilitam determinar a melhor configuração de células (Kochikar e Narendran, 1998), alternativas de rotas e máquinas (Tsubone e Horikawa, 1999), (Caprihan e Wadhwa, 1997), seqüência de processamento (Wadhwa *et al.*, 2005) e uma série de algoritmos que levam em conta estas características para tornar as respostas destes sistemas mais efetivas e eficientes.

No contexto da engenharia de automação, os roteiros de produção de acordo com Ramirez-Serrano *et al.* (2000a) definem a capacidade do sistema de transporte em fazer um roteamento adequado das peças através do processo, nas diferentes células de manufatura, para a realização de operações específicas e necessárias. Este fator de flexibilidade de roteamento é visto como um importante problema da engenharia de controle e representativo de pesquisas, já que tem um impacto fundamental nas taxas de produção e define em grande parte a utilização e disponibilidade dos recursos das células de manufatura, em processos com elevados índices de concorrência nas atividades (Kochikar e Narendran (1998).

Ao observarem tais problemas, Ramirez-Serrano *et al.* (2000a e 2000b), desenvolveram trabalhos para a modelagem e síntese de controladores utilizando *Autômatos de Moore Expandidos* e a Teoria de Ramadge-Wonham (1987) como as ferramentas formais para o projeto de controladores. Abordando a Teoria clássica de Autômatos Finitos (Gouyon, 2004a), explora as características de projetos dirigidos ao produto, e os aspectos relacionados ao processo nas etapas de modelagem e síntese das estruturas de controle.

Explorando a modularidade física dos sistemas de manufatura e das especificações de controle (Queiroz e Cury, 2000) contribuíram com a abordagem modular local da TCS possibilitando um novo conceito de implementação. Implementação que teve em Brandin (1996), Fabian e Hellgren (1998) e Queiroz *et al.* (2001) contribuições para a aplicação da TCS em controladores lógicos programáveis (CLP's) através de diagramas escada.

Os trabalhos acima citados seguem os propósitos de fornecer uma maior estruturação ao projeto de controladores, através da incorporação de algoritmos computacionais de síntese. A utilização destes algoritmos proporciona a diminuição no tempo de desenvolvimento, aumentam a confiabilidade, facilitam a manutenção e entendimento por parte dos mantenedores dos sistemas, e propiciam uma rápida alteração dos controladores responsáveis pelas rotas de produção.

Durante o processo de síntese dos supervisores, a utilização de ferramentas computacionais como TCT (Wonham, 1999) e GRAIL (Raymond e Wood, 1996), tornam esta metodologia de projeto diferenciada das tradicionais técnicas manuais e intuitivas utilizadas como ferramentas nos trabalhos atuais de projeto lógico.

Os objetivos de implementação deste trabalho estão de acordo com a abordagem Modular Local da TCS (Queiroz e Cury, 2000) a Arquitetura de Controle Supervisório também proposta por (Queiroz e Cury, 2002) e o modelo para implementação em CLP's de Vieira *et al* (2007) que se utiliza, além do diagrama escada, a programação em GRAFCET através de estruturas de controle concentradas e também distribuídas.

1.5 Organização do Trabalho

O presente trabalho divide-se da seguinte forma: no capítulo 2 apresentam-se algumas definições que dizem respeito às diferentes classificações dos sistemas de manufaturas. O sistema de transporte responsável pela movimentação de peças entre as estações de processamento é exemplificado de modo a observar como se realiza a integração entre as várias estações de trabalho, e os possíveis roteiros de produção objetivando o processamento do produto final. Uma conceituação dos diferentes tipos de flexibilidades é elaborada de modo a identificar importantes relações que contribuam para o aumento da concorrência entre os processos.

O capítulo 3 apresenta a fundamentação teórica sobre Linguagens Formais e Autômatos, utilizados para representação e estudo dos sistemas de manufatura. A Teoria de Controle

Supervisório, modelo Ramadge e Wonham (1989), de acordo com as abordagens Centralizada e Modular Local (Queiroz e Cury, 2000), é explanada estabelecendo as diferenças conceituais com vistas aos objetivos de implementação. O capítulo 4 apresenta o Sistema Físico a ser modelado tendo como objetivo principal a modelagem do sistema de transporte, uma vez que apresenta uma complexidade maior em função das características de operação que exige um sistema flexível de manufatura. Desenvolve-se então uma explicação de todos os modelos considerados no trabalho, concluindo-se por aquele que representa da melhor o comportamento do sistema em questão.

No capítulo 5 é realizada a modelagem das especificações de controle, que são divididas em quatro conjuntos distintos, assim como o desenvolvimento de alguns procedimentos referentes à concepção das referidas especificações com vistas a sua generalização. O processo de síntese dos supervisores utiliza-se da abordagem modular local da TCS e é concluído com a obtenção dos supervisores reduzidos.

No capítulo 6 desenvolve-se o estudo da aplicação real através da conversão em código de CLP's das estruturas de controle anteriormente obtidas. Este processo se realiza de acordo com a abordagem modular local da TCS (Queiroz e Cury, 2002) e o Modelo para Implementação CLP's (Vieira *et al.*, 2007). No capítulo 7 são apresentadas as contribuições alcançadas através da realização deste trabalho.

Capítulo 2

Sistemas de Manufatura Flexível

Neste capítulo são apresentados os conceitos referentes aos sistemas de manufatura, considerando as suas classificações em função do nível e tipo de automação presente nos processos produtivos, que se subdividem em manual, automático e híbrido. As diferentes configurações de leiaute dos sistemas de manipulação e transportes são apresentadas de modo a estabelecer uma correlação com o trabalho desenvolvido. Este importante aspecto de configuração proporciona aos sistemas de manufatura a possibilidade de processar diferentes famílias de produtos, seguindo diferentes roteiros de produção, através do que se classifica de sistemas de rotas flexíveis.

Uma definição e caracterização dos diferentes atributos de flexibilidade são desenvolvidas com o objetivo de estabelecer as suas correlações com os tipos de produtos e processos. Alguns destes atributos são classificados como flexibilidades de volume, flexibilidade de rotas, flexibilidade de produto, flexibilidade de sequenciamento e flexibilidade de máquinas.

2.1 Sistemas de Manufatura

Segundo Groover (2001) um sistema de manufatura é um conjunto composto de recursos humanos e equipamentos integrados cuja principal função é desempenhar uma série de atividades, nas quais se desenvolvem operações de montagem e processamento sobre uma matéria-prima, peça ou conjunto de peças de modo a obter-se um produto final. Os equipamentos integrados incluem uma categoria de dispositivos e máquinas inteligentes cuja integração, coordenação e controle se realiza através de sistemas computacionais.

A figura 2.1 mostra como se compõe um sistema de produção típico e a posição ocupada pelo sistema de manufatura no processo produtivo. Observa-se o suporte provido à esta atividade

de manufatura através das tecnologias de manipulação de material e tecnologias de automação e controle.

As tecnologias de manipulação de material compreendem os equipamentos responsáveis pelo transporte e manuseio do material em processo através das estações de trabalho. Uma metodologia voltada ao projeto conceitual dos sistemas de manipulação e montagens automatizados é abordada em Santos (2003).

As tecnologias de automação e controle constituem-se dos controladores programáveis e sistemas computacionais baseados em software de forma a coordenar as atividades do sistema produtivo. Este trabalho visa a modelagem do sistema de transporte e a sua implementação em tecnologias de automação e controle.

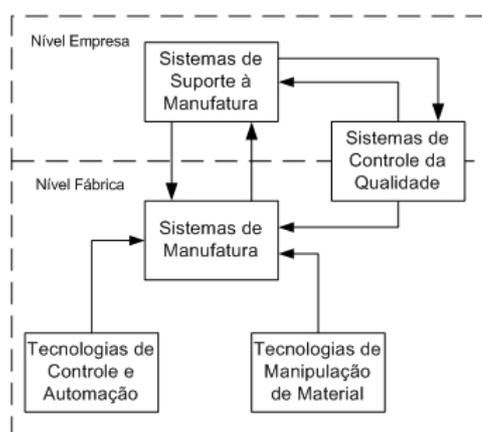


Figura 2.1: Composição de um sistema de produção. Groover (2001)

Os modernos sistemas de manufatura são compostos de múltiplas estações de processamento ou trabalho. Dependendo do seu arranjo e função, podem constituir o que se convencionou chamar de uma linha de produção ou célula de montagem (Groover, 2001).

As tarefas de interligação e deslocamentos entre as estações podem ser desenvolvidas por equipamentos como veículos auto-guiados (AGV), esteiras transportadoras, atuadores pneumáticos, sistemas mono-trilho ou robôs. Este sistema é definido como Sistema Primário de Transporte.

Para o posicionamento, carregamento e descarregamento de peças nas máquinas conceituado de Sistema Secundário de Transporte utilizam-se com frequência os robôs industriais. Um robô industrial é definido por Slack (2001) como: "...um manipulador

automático multifunção que tem diversos graus de liberdade, capaz de manusear materiais, peças, ferramentas ou dispositivos especializados através de movimentos programados variáveis, para desempenho de uma variedade de tarefas...”.

O leiaute do processo que define a disposição entre as estações de processamento, contribui para a escolha do melhor sistema ser adotado e caracteriza as possibilidades de rotas entre as estações.

A figura 2.2 exemplifica duas das possíveis configurações entre as estações e as possíveis rotas de produção características de cada sistema produtivo. Na situação (a) têm-se um sistema que aceita uma variedade de rotas para a manufatura de um determinado produto, ou seja, uma mesma operação pode ser desenvolvida em diferentes máquinas. No caso (b) a rota de produção é fixa e a seqüência de processamento sempre se repete.

Segundo Zahran *et al.* (1990) a rota de processamento ou produção de um produto é o conjunto de operações necessárias para o processamento deste produto. O conjunto de operações a serem desenvolvidas sobre este produto é realizado de acordo com uma seqüência tecnológica.

Um conjunto de máquinas alternativas pode existir para desempenhar cada uma destas operações e a disponibilidade de múltiplas alternativas conduzirá à uma flexibilidade nos roteiros de produção.

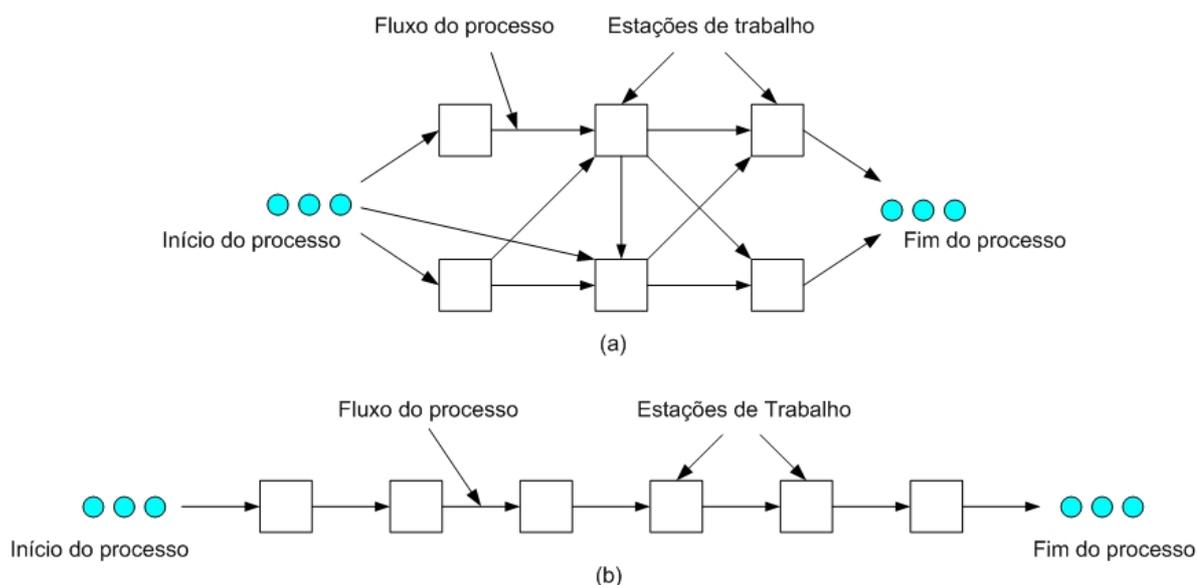


Figura 2.2: Tipos de rotas em sistemas de manufatura com múltiplas estações, (a) rotas variáveis e (b) rotas fixas. (Groover, 2001)

2.2 Classificação dos Sistemas de Manufatura

De acordo com Groover (2001) são quatro os fatores principais que distinguem e definem os diferentes tipos de processos com vistas à uma classificação sendo:

- 1) *Tipos de operações desenvolvidas*: a) Operações de processamento sobre unidades individuais e; b) Operações de montagem de modo a combinar peças individuais em entidades montadas. As tecnologias utilizadas para estas atividades são relevantes para a classificação.
- 2) *Número de operações e configuração do sistema*: O arranjo entre as estações e o número destas, definem as possibilidades de rotas para as tarefas de processamento. A figura 2.2a identifica um arranjo com rotas variáveis, enquanto a figura 2.2b destaca uma configuração em linha cujo roteiro de produção é fixo.

Este fator subdivide-se em três níveis:

2.1) Tipo I, *Simples estação*: Constitui-se de apenas uma máquina de produção que pode ser operada manualmente, semi-automatizada ou plenamente automatizada;

2.2) Tipo II, *Múltiplas estações com rotas variáveis*: Este sistema de manufatura compõe-se de duas ou mais estações, trabalha com diferentes famílias de peças ou produtos realizando operações de montagem ou processamento. Groover (2001) define uma família de peças como uma coleção de peças que são similares pela forma geométrica ou tamanho, ou requerem etapas de processamento similares em sua manufatura;

2.3) Tipo III, *Múltiplas estações com rota fixa*: Este sistema constitui-se de duas ou mais estações arranjadas em linha.

- 3) *Nível de Automação*: Caracteriza as estações do sistema de manufatura como operadas manualmente, semi-automatizadas ou automatizadas.

A figura 2.3 representa os diferentes tipos de células a partir do nível de automação (Groover, 2001):

- a) Tipo I M: *Simples estação e célula manual*: a máquina é operada manualmente, o trabalhador permanece continuamente, gerenciando o processo.
- b) Tipo I A: *Simples estação e célula automatizada*: A máquina é completamente automatizada e o trabalhador apenas carrega e descarrega a máquina em períodos longos de tempo.

- c) Tipo II M: *Múltiplas estações com sistema manual e rotas variáveis*: As estações são operadas manualmente ou semi-automatizadas. Ao sistema de transporte é possível a

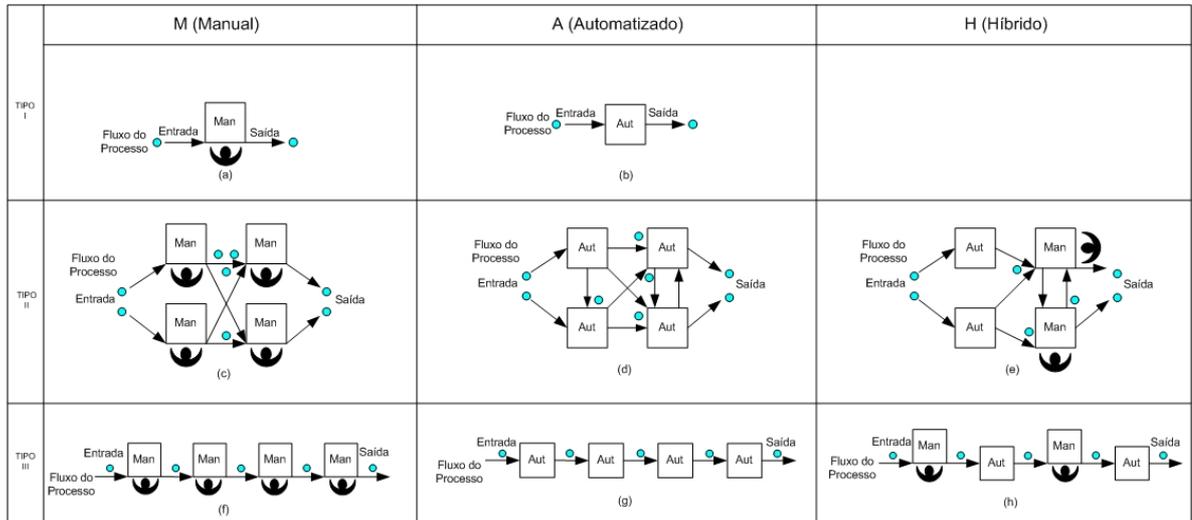


Figura 2.3: Classificação dos Sistemas de Manufatura quanto ao nível de automação. (Groover, 2001)

realização de várias rotas, assim como às peças e produtos. O transporte entre estações é manual ou mecanizado.

- d) Tipo II A: *Múltiplas estações com sistema automatizado e rotas variáveis*: Diferencia-se do anterior pelo fato do transporte e operação das estações serem completamente automatizados.
- e) Tipo II H: *Múltiplas estações com sistema híbrido e rotas variáveis*: Tanto o sistema de transporte como a operação das estações, podem ser automatizados, manual ou misto (híbrido).
- f) Tipo III M: *Múltiplas estações com sistema manual e rotas fixas*: O número de máquinas e operadores é maior ou igual a dois. O processo se desenvolve de forma seqüencial e o sistema de transporte pode ser manual ou mecanizado.
- g) Tipo III A: *Múltiplas estações com sistema automatizado e rota fixa*: O processo se desenvolve de forma semelhante ao anterior, diferenciando-se apenas pela operação e transporte que são completamente automatizados.

- h) Tipo III H: *Múltiplas estações com sistema híbrido e rota fixa*: Tanto a operação e o sistema de transporte podem ser realizados de forma manual, automática ou mista (híbrida).
- 4) *Variedade de Peças e Produtos*: Capacidade do sistema de trabalhar com diferentes peças ou produtos. Em razão desta variedade o sistema poderá ter uma flexibilidade maior ou menor.

2.3 Sistemas de Manuseio e Transporte de Material

O manuseio e transporte de material são importantes tarefas desenvolvidas na execução dos sistemas de manufatura. Nos modernos sistemas produtivos, os elevados volumes de produção e o grande número de operações nos produtos determinam a necessidade de roteiros alternativos na movimentação de material entre as estações de processamento de forma à atender a variedade imposta pelo mercado consumidor. O desempenho destes sistemas é objeto de intensos estudos científicos e são bastante responsáveis pelos elevados índices de produtividade e flexibilidade obtidos nos sistemas produtivos.

Como observado na seção anterior Groover (2001) estabelece uma classificação aos diferentes sistemas de manufatura e as possibilidades de movimentação de material entre as estações caracterizam a sua operação como manual, automatizado ou híbrido.

Dentre os equipamentos e mecanismos clássicos de transferência e transporte de peças entre as estações estão os robôs e as esteiras transportadoras. No entanto, esta função de transporte segundo Groover (2001) é compartilhada entre dois sistemas: (1) Sistema de manuseio primário e (2) Sistema de manuseio secundário.

O sistema de manuseio primário estabelece a configuração básica do sistema de manufatura e é responsável pelo transporte de peças entre as estações. Os equipamentos mais comumente utilizados para este fim estão descritos na Tabela 2.1 (Groover, 2001).

O sistema de manuseio secundário consiste de dispositivos de transferência automáticos localizados nas estações de processamento. A função deste sistema é a transferência de peças do sistema primário às máquinas ferramentas ou outras estações de processamento, armazenagem de peças, assim como o posicionamento destas de forma precisa e repetitiva para as operações de montagem e outros processos.

Configuração de Leiaute	Sistema de Manuseio de Material Típico
Leiaute em Linha (linhas de transferência)	Sistema de transferência em linha, sistema de esteiras, sistemas de veículos guiados sobre trilhos.
Leiaute em Circuito Fechado (círculo ou retangular)	Sistemas de esteiras, <i>In-floor towline carts</i> .
Leiaute Escada	Sistemas de esteiras, sistemas de veículos auto-guiados (AGV's), sistemas de veículos guiados sobre trilhos.
Leiaute de Campo Aberto	Sistemas de veículos auto-guiados (AGV's), <i>In-floor towline carts</i> .
Leiaute Centrado em Robô	Robôs industriais

De acordo com Groover (2001), os sistemas de manuseio de material determinam o leiaute dos sistemas de manufatura e podem ser divididos em cinco categorias principais conforme a tabela 2.1.

Na configuração *leiaute em linha*, as máquinas e o sistema de manuseio são arranjados em linha reta, conforme ilustrado nas figuras 2.4 e 2.5 a seguir. É a forma mais simples, e as peças seguem um fluxo contínuo de uma estação à outra numa seqüência bem definida.

Sua operação se assemelha à uma linha de transferência, o fluxo de trabalho sempre acontece numa única direção e nunca no sentido contrário como se observa na figura 2.4. As peças seguem uma mesma rota de processamento embora possam passar por várias operações de processamento em cada estação. A classificação para este tipo de sistema é tipo III A.

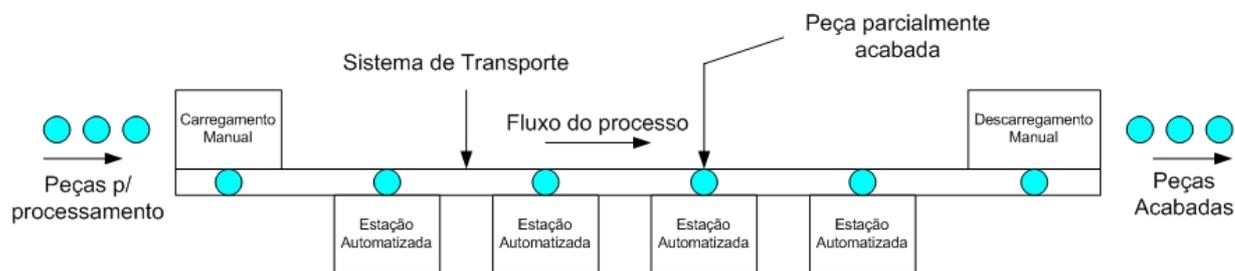


Figura 2.4: Leiaute em linha com rotas fixas Groover (2001).

Uma outra possibilidade para esta configuração em linha é como apresentado na figura 2.5 em que os roteiros de produção podem ser variados, uma vez que o transportador pode

movimentar-se em ambas as direções atendendo às várias estações. Este tipo de sistema é classificado como do tipo II A.

A configuração *Leiaute em Circuito Fechado* organiza as estações em círculo ou retângulo formando em circuito fechado através das estações com o sistema primário de transporte seguindo o mesmo formato. O fluxo das peças segue em uma única direção e o sistema de transporte secundário presente em cada estação é capaz de manipular as peças sem interromper o fluxo normal.

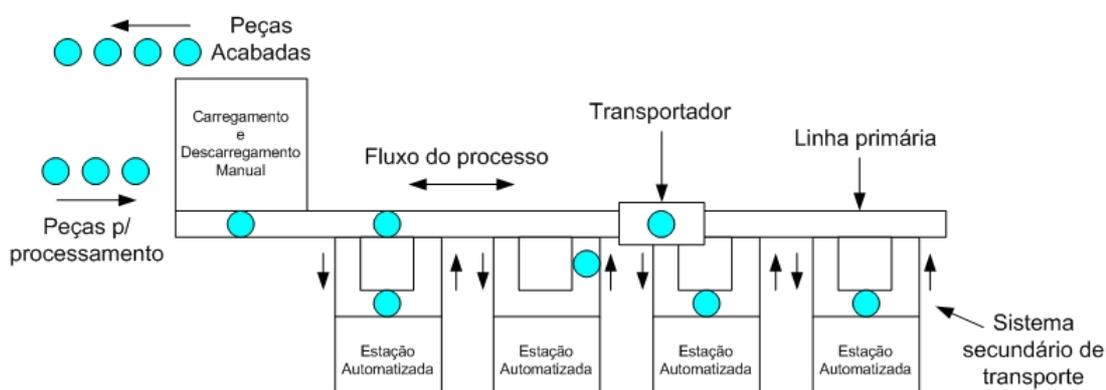


Figura 2.5: Leiaute em linha com rotas flexíveis Groover (2001).

A representação esquemática da configuração de *Leiaute em Circuito Fechado* pode ser observada nas figuras 2.6 e 2.7 com uma distribuição elíptica e retangular das estações respectivamente.

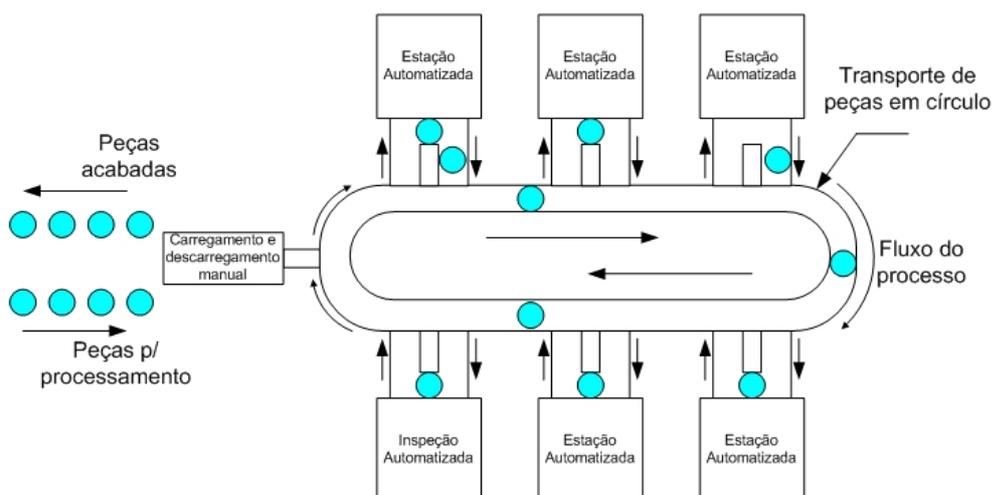


Figura 2.6: Leiaute em circuito fechado formato elíptico Groover (2001).

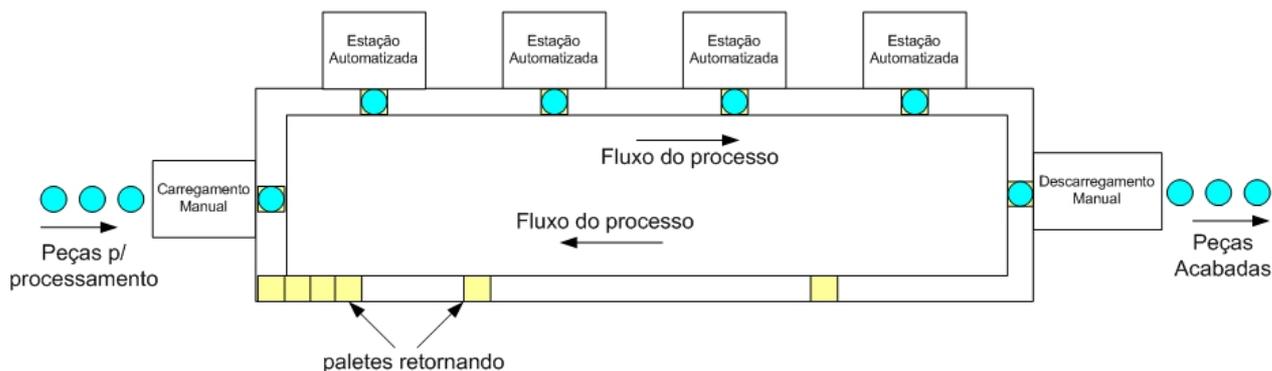


Figura 2.7: Leiaute em circuito fechado formato retangular Groover (2001).

Um *leiaute em escada* se assemelha ao leiaute retangular com uma diferença na sua parte interna onde se localizam as estações de trabalho. Estes anéis retangulares internos às seções retas do sistema primário de transporte propiciam caminhos alternativos às peças quando se deslocam de uma estação à outra, reduzindo o tempo de transporte entre as estações.

Um sistema secundário de transporte também se faz necessário sem que haja a obstrução do fluxo que segue em uma única direção. Esta configuração pode ser observada na figura 2.8.

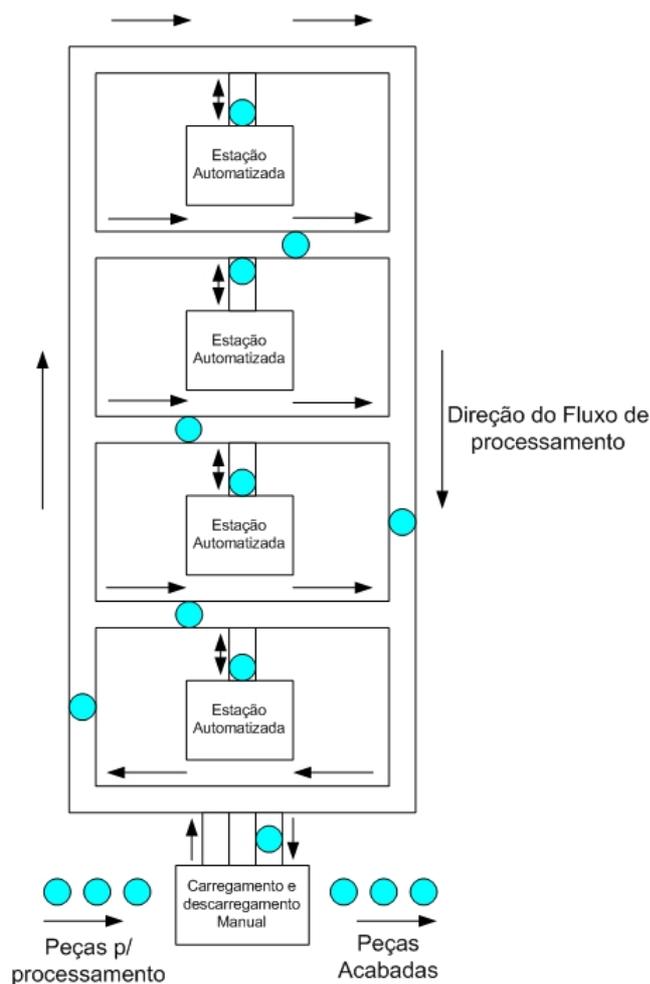


Figura 2.8: Leiaute escada Groover (2001).

Uma combinação das duas últimas configurações, circuito fechado e escada, consiste do *leiaute em campo aberto*. Este tipo de configuração é apropriado ao processamento de uma grande família de peças em razão das várias rotas possíveis no sistema.

A configuração de *leiaute centrado no robô* utiliza um ou mais robôs como sistema de manuseio e transporte de material entre as diferentes estações. Esta configuração será objeto de modelagem deste trabalho por três motivos: 1) em razão da complexidade envolvida para a obtenção da lógica de controle; 2) por ser uma configuração muito utilizada na indústria; 3) é o sistema utilizado no laboratório da PUC-PR. Uma correlação ao sistema de transporte da figura 2.5 é perfeitamente admissível. Em processos com este sistema de transporte uma grande família de peças pode ser trabalhada e as rotas de produção são variáveis. Uma desvantagem deste tipo

de transporte é que apenas uma peça pode ser transportada por vez. A figura 2.9 ilustra o leiaute centrado no robô.

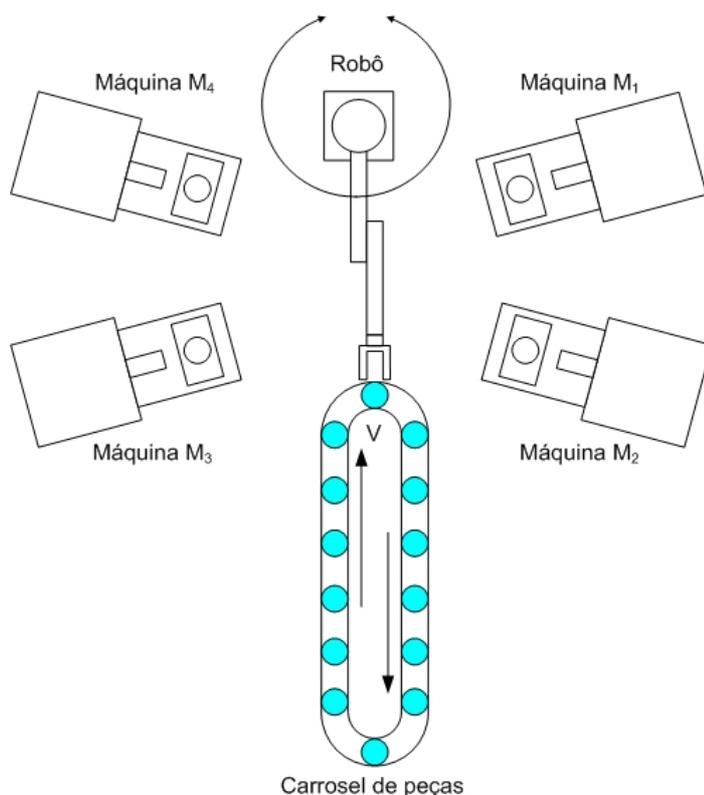


Figura 2.9: Leiaute de Célula Centrada em robô Groover (2001).

Estas características relacionadas aos sistemas de manipulação e transporte de peças agregam flexibilidades aos processos de forma a atender os requisitos de volume e variedade dos processos produtivos. Desta forma é necessário que a definição de alguns conceitos a cerca das diferentes flexibilidades associadas à estes processos sejam colocadas para melhor compreender o que se convencionou chamar de Sistemas Flexíveis de Manufatura.

2.4 Flexibilidade dos Sistemas de Manufatura

A busca por eficiência e competitividade conduz as empresas a automatizarem seus processos, que ultimamente têm aumentado consideravelmente a sua complexidade. Quanto mais complexo um processo, maiores serão os tempos envolvidos em seu projeto e implementação, tornando as corporações menos ágeis em suas estratégias de mercado. Nos

sistemas de produção o binômio volume-variedade tem influência direta na escolha das tecnologias utilizadas para a fabricação, assim como leiaute de fábrica e características do processo, dentre as quais o fator flexibilidade é de grande importância tanto para os objetivos de respostas rápidas, bem como nas atividades de projeto relacionadas ao produto e ao processo. Observa-se, com frequência, que nas operações de grande variedade de produtos, o volume costumeiramente é baixo.

Para Slack (2001), a forma como a empresa irá visualizar os seus objetivos de desempenho e projeto são diretamente determinados ou influenciados por estes dois aspectos do processo produtivo, volume e variedade. Slack (2001) define flexibilidade como a capacidade que um sistema possui de alterar as operações ou estado, de modo regular e sincronizado, ou seja, alterar o que se faz, como faz ou quando faz, sem ocasionar perdas ao sistema. Kochikar & Narendran (1998) adiciona à esta definição a habilidade de resposta rápida e eficiente. Wadhwa *et al.*(2005) considera que a regra de flexibilidade num sistema de manufatura é habilitá-lo a gerenciar as mudanças (certas ou incertas), de modo efetivo e eficiente.

Brandin (1996) observa que, além do elevado grau de automação e integração presente nos Sistemas Automatizados de Manufatura, os aspectos relacionados à flexibilidade podem assumir diferentes formas:

- flexibilidade de volume, que é a capacidade de gerenciar alterações no volume de produção, ou seja, alterar os níveis de saída;
- flexibilidade de rotas, que é a capacidade de rotear peças através do sistema de forma dinâmica levando em conta, quebra de máquinas, tipos de operações a serem desenvolvidas nas peças, etc;
- flexibilidade de produto, que é a capacidade de tratar requisitos para uma larga variedade de produtos, incluindo a capacidade de re-configurar o sistema para trabalhar a produção de diferentes produtos.

A flexibilidade de rotas é definida segundo Caprihan e Wadhwa (1997) e, Tsubone e Horikawa (1999) como a habilidade de tratar quebras e continuar a produzir um dado conjunto de tipos de peças. Isto necessariamente implica na existência de múltiplas seqüências de rotas para cada tipo de peças através de máquinas multifunções e um sistema de manuseio de material versátil, com o propósito de melhorar o desempenho do sistema.

De acordo com Kochikar e Narendran (1998) flexibilidade de rotas é a habilidade de variar a seqüência de visitação das máquinas . Isto é, uma mesma operação pode ser executada

em diferentes máquinas e pode-se assim alterar a seqüência nas quais as operações são realizadas.

Tsubone e Horikawa (1999) e, Kochikar e Narendran (1998), ainda conceituam a flexibilidade de máquinas como a capacidade de uma máquina desempenhar diferentes operações requeridas para um dado conjunto de tipo de peças. Adiciona-se a este conceito a preparação rápida de máquinas.

No sentido de suprir as diferentes máquinas do processo para execução das operações, Kochikar e Narendran (1998) definem a flexibilidade de transferência de material como a eficácia do Sistema de Manuseio de Material em realizar a transferência entre máquinas de um dado conjunto de peças.

Um outro tipo de flexibilidade que se encontra presente nos sistemas de manufatura, é a flexibilidade de sequenciamento (FS). Segundo Wadhwa *et al.*(2005) esta refere-se a capacidade do sistema de alterar a ordem das operações de processamento sobre um determinado produto. Através da figura 2.10 (Wadhwa *et al.*, 2005), ilustra-se algumas das alternativas de sequenciamento em função das características do produto e operações a serem desenvolvidas nas diferentes estações de processamento:

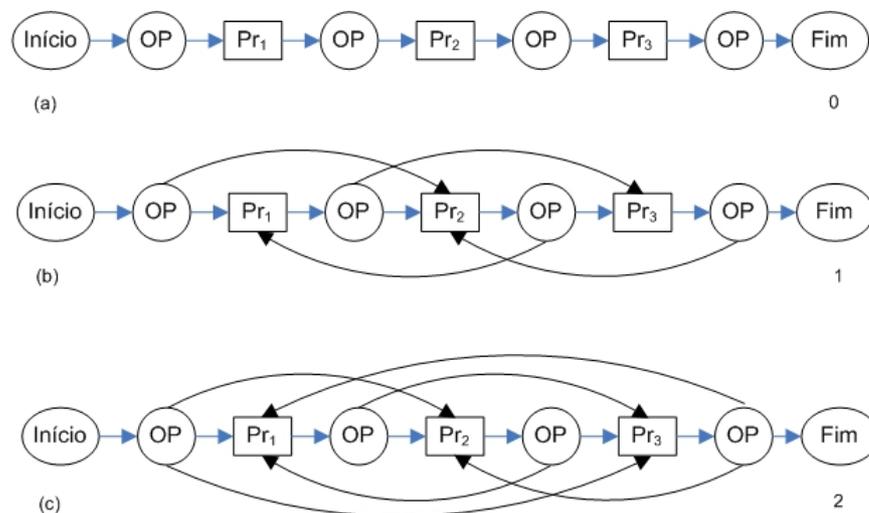


Figura 2.10: a) Níveis de flexibilidade de sequenciamento NFS = 0, b) NFS = 1, c) NFS= 2.

Na figura 2.10a, observa-se um processo com nível de sequenciamento 0(zero), pois existe uma única ordem na qual as operações sobre o produto serão realizadas. A figura 2.10b, apresenta um sistema no qual o grau de flexibilidade do produto é maior, podendo as operações,

serem realizadas em diferentes ordens. A figura 2.10c, apresenta um sistema com várias possibilidades de ordenamento, característico dos sistemas multi-produto.

Alguns autores se referem à flexibilidade de rotas como flexibilidade de operações e flexibilidade de sequenciamento. Neste sentido Benjaafar e Ramakrishnan (1996) consideram que três tipos principais de flexibilidade estão associadas a manufatura de uma peça que são:

- Flexibilidade de operações: relaciona a possibilidade de desempenhar uma operação em mais de uma máquina;
- Flexibilidade de sequenciamento: refere-se à possibilidade de intercambiar a seqüência na qual as operações de manufatura são executadas;
- Flexibilidade de processamento: é determinada pela possibilidade de produzir o mesmo aspecto de manufatura com operações alternativas ou seqüências de operações.

A figura 2.11 (Benjaafar e Ramakrishnan, 1996) ilustra estes tipos de flexibilidades.

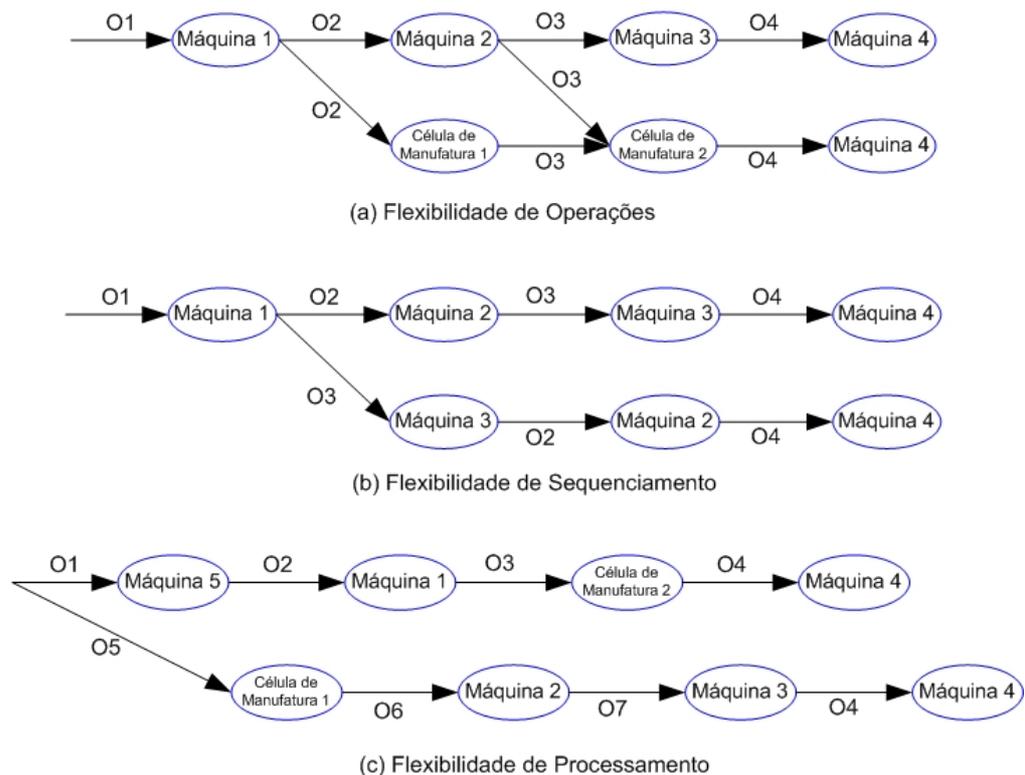


Figura 2.11: Flexibilidades na manufatura de uma peça.

A figura 2.11a representa um sistema em que uma operação (O2 ou O3) pode ser realizada em mais de uma máquina. Estas máquinas alternativas não necessariamente precisam

ser iguais, mas têm vários graus de flexibilidade. A figura 2.11b ilustra um processo em que a ordem das operações pode ser variável. As operações O2 e O3 podem ser realizadas em qualquer ordem após a operação O1.

Num sistema de manufatura altamente flexível, um trabalho sobre uma peça pode ser simultaneamente executado por um conjunto de planos de processos alternativos. Isto quer dizer que a composição do conjunto de operações requeridas para a manufatura de uma peça pode ser variável. Por exemplo, na figura 2.11c as operações O1 e O2 podem ser substituídas pela operação O5, enquanto a operação 3 pode ser substituída pelas operações O6 e O7.

Fundamentado nas características anteriormente relacionadas, Groover (2001) classifica os Sistemas Automatizados de Manufatura, que possuem diferentes graus de flexibilidade e sequenciamento, em três categorias de acordo com o tipo de automação:

- 1) Automação Fixa, na qual a disposição dos equipamentos determina a seqüência de operações, podendo-se trabalhar com altas taxas de produção e pequena variedade de produtos, caracterizando o sistema como relativamente inflexível;
- 2) Automação Programável, o equipamento de produção é projetado com a capacidade de alterar a seqüência de operações e produzir diferentes configurações de produtos. As taxas de produção são menores que a automação fixa e possui a flexibilidade para trabalhar com as variações e alterações na configuração do produto, sendo estes sistemas adequados para se trabalhar com baixos e médios volumes de produção;
- 3) Automação Flexível, é uma extensão da automação programável, este sistema é capaz de produzir uma variedade de peças (ou produtos), sem praticamente perdas de tempo na fabricação de um tipo de produto a outro. É possível a produção contínua de uma variedade de produtos, as taxas de produção são medianas e existe a flexibilidade em lidar com variações no *design* dos produtos.

Desta forma Groover (2001) afirma que a palavra flexibilidade é um termo muito utilizado nos sistemas de manufatura para definir as diferentes características dos processos. Este atributo permite aos processos trabalharem com uma gama de diferentes produtos, considerando-se um certo grau de variação no estilo da peça ou produto, sem interrupção na produção quando se alteram os modelos a serem produzidos. Sendo assim os sistemas de manufatura que possuem esta versatilidade nos processos de execução através da presença dos diferentes tipos de flexibilidades e automação enunciados anteriormente, são chamados de *Sistemas Flexíveis de*

Manufatura ou *FMS (Flexible Manufacturing Systems)*, sigla em inglês mundialmente utilizada para referenciar-se a este tipo de sistema.

Alguns dos benefícios que podem advir da implantação de um FMS, segundo Groover (2001) são:

- aumento na utilização das máquinas;
- requer menos máquinas;
- redução de espaço físico no chão de fábrica;
- melhor resposta às mudanças;
- redução de inventário;
- tempo de processamento (*lead times*) menores.

Diante de tantos atributos e benefícios imagina-se que para um sistema flexível de manufatura não haja limites de produção ou produtos em processo. Groover (2001) considera que nenhum sistema de manufatura possa ser completamente flexível. Há limites para os tipos de peças ou produtos que possam ser feitos em um *FMS*. Estes sistemas são projetados para produzir peças ou produtos dentro de uma faixa definida que abrange os estilos, geometria, tamanho e tipo de processamento. Isto significa que em um *FMS* há limites à produção de famílias de peças.

A razão para que um *FMS* seja então chamado de flexível é a capacidade de processar uma variedade de diferentes estilos de peças simultaneamente em várias estações de trabalho ou processamento (Groover, 2001). Este fato implica em um grande número de atividades que se desenvolvem de forma concorrente ou paralela.

A definição de concorrência pode ser entendida como o ato de se executar as operações de forma simultânea ou conjunta. Num sistema de manufatura, a concorrência refere-se a realizar múltiplos processos em múltiplas máquinas (flexibilidade de máquinas) simultaneamente, desta forma quanto mais processos são realizados concorrentemente, o tempo de processo (*lead-time*) diminuirá proporcionalmente.

Dois processos ‘A’ e ‘B’ (Wadhwa *et al.*, 2005) são realizados de forma seqüencial e não concorrentemente, por algumas das seguintes razões, e que são vistas como fatos que restringem a concorrência nos processos:

1. por razões de determinada prática ou hábito, ou por conveniência;
2. devido a condições funcionais;
3. ‘B’ requer informações de ‘A’;

4. 'B' requer decisões de 'A';
5. 'B' requer material de 'A';
6. 'B' requer recursos que estão sendo utilizados concorrentemente por 'A';
7. Há a necessidade de sincronização de alguns eventos de 'A' e 'B', que podem resultar em atrasos nas relações de precedência entre 'A' e 'B'.

As figuras 2.12 (Wadhwa *et al.*, 2005), ilustram o grau de concorrência entre os processos 'A' e 'B'. Na figura 2.12a, verifica-se a inexistência de atividades paralelas no desenvolvimento de cada um dos processos, enquanto que na figura 2.12b o paralelismo nas operações pode ser observado de forma parcial. A figura 2.12c representa a concorrência total de atividades entre dois processos.

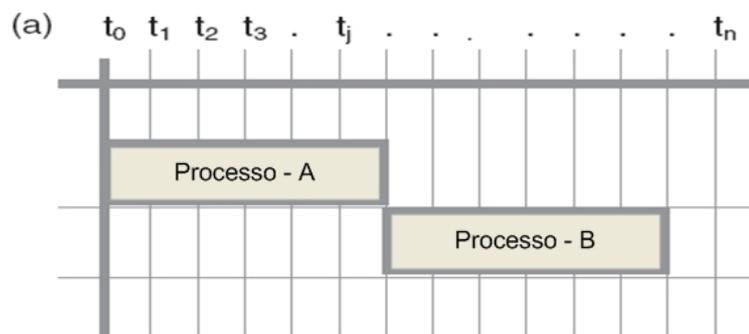


Figura 2.12a: Processo seqüencial, concorrência zero.

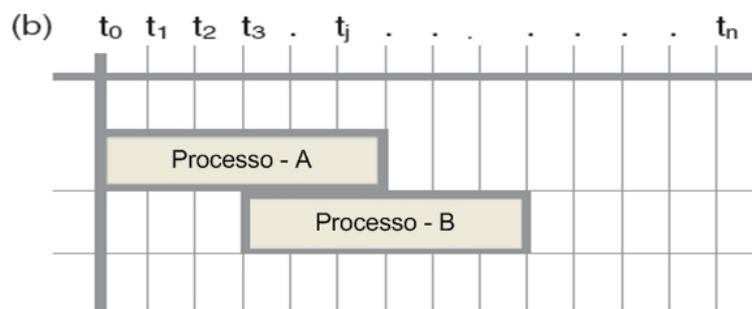


Figura 2.12b: Sobreposição de processos, concorrência parcial.

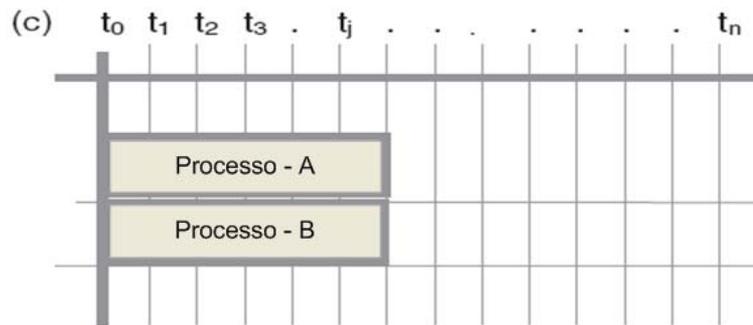


Figura 2.12c: Processos Paralelos, concorrência máxima.

Observa-se na figura 2.13 (Wadhawa *et al.*, 2005) à medida que se aumenta a concorrência entre os processos e diminui o desbalanceamento de carga, reduz-se o tempo de processo e diminui a inflexibilidade do sistema. Desta forma, o sistema de controle tem grande importância sobre o nível de flexibilidade final dos processos, os benefícios então, dependem de como o sistema de controle pode ser capaz de explorar esta flexibilidade.

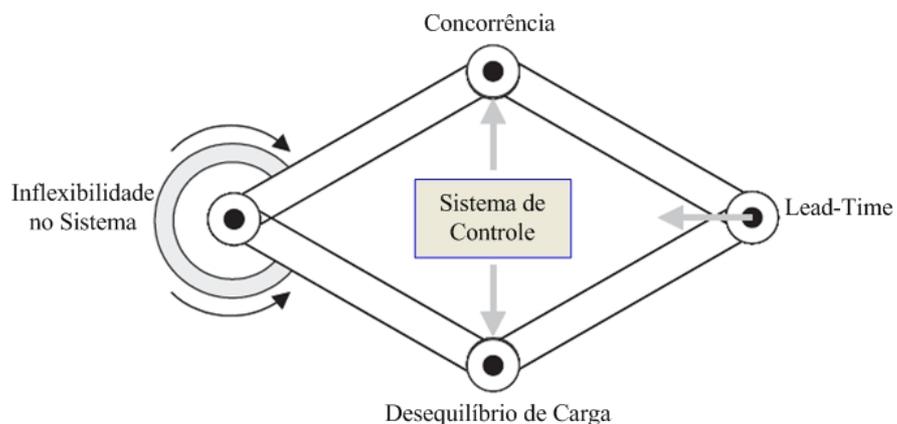


Figura 2.13: Sistema de controle e flexibilidade dos processos.

De acordo com Kochikar e Narendran (1998) uma motivação maior que diferencia os *FMS* dos sistemas convencionais é a racionalização no sistema de manuseio e transporte de material. A versatilidade dos equipamentos de manuseio e transporte de material em conjunto com os equipamentos de produção é que permitem aos *FMS* operar com um espectro grande de tipos de peças. Agregada à esta condição estão as várias opções dos roteiros de produção para

cada tipo de peça. Isto assegura alta flexibilidade e melhora o desempenho do sistema como um todo.

Isto permite definir um outro tipo de flexibilidade denominada *Flexibilidade de Configuração* que é resultado da integração do sistema de manuseio de material automatizado. A larga variedade de produtos em processo impõe variações nos roteiros de produção, que ocorrem porque diferentes materiais seguem diferentes fluxos na fábrica, e se estas diferenças existem, o sistema de manuseio do material deve ser flexível o suficiente para lidar com esta tarefa.

A flexibilidade de rotas ou a flexibilidade de máquinas (Wadhwa *et al.*, 2005) podem ser exploradas na redução do *lead-time*, como mostra a figura 2.13, num sistema de manufatura multi-produto. Dentre os vários fatores que podem influenciar o tempo de processo na manufatura, dois fatores, o nível de concorrência entre os processos e o nível de desbalanceamento de carga entre os processos, são fatores de maior significância.

Um maior nível de concorrência entre os processos pode ser obtido através da lógica de controle do sistema de transporte. Neste sentido, uma estruturação nos processos de modelagem e implementação, facilita a re-configuração do sistema de controle tornando menos dispendiosa esta atividade.

Para Yalcin (2004) as flexibilidades associadas às peças e recursos na célula têm sido estudadas dentro do conceito de flexibilidade de rotas. Na maior parte das abordagens para programar sistemas flexíveis de manufatura, a desejada rota da peça é pré-determinada, e durante a operação, cada tipo de peça tem somente um roteiro através do sistema. Uma programação fixa da rota de peças não é muito adequada para o restabelecimento dos distúrbios do sistema em tempo-real, tais como uma quebra de máquina e outros eventos aleatórios que podem frustrar os propósitos da programação.

O presente trabalho considerará para efeito de estudos, apenas as situações em que o sistema opera sob condições normais, ou seja, sem a observação dos eventos relacionados à falha dos equipamentos. O fato que justifica esta decisão é a explosão do número de estados que se agregaria aos modelos dos subsistemas e o conseqüente esforço computacional que seria de difícil solução.

Como se pode observar, à medida que o número de estações aumenta, torna-se necessário uma abordagem diferenciada no tratamento das questões relacionadas ao projeto dos sistemas de manuseio (Santos, 2003) e transporte. Esta tarefa de re-projeto ou alteração dos roteiros já

existentes, deve contribuir de forma positiva para a elevação do nível de concorrência no processo e elevar o grau de flexibilidade do sistema.

A figura 2.14 (Benjaafar, 1996) apresenta de forma resumida os diferentes tipos de flexibilidades presentes nos sistemas flexíveis de manufatura e associadas aos produtos e processos de execução. Uma discussão mais detalhada sobre estes diferentes tipos de flexibilidades pode ser encontrada em Benjaafar (1995).

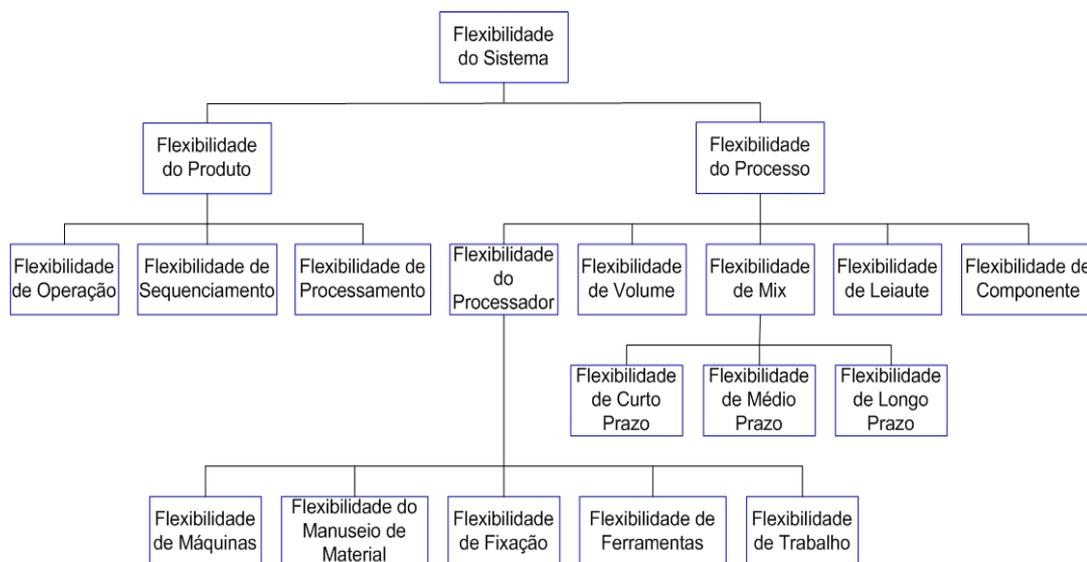


Figura 2.14: Hierarquia da flexibilidade nos sistemas.

2.5 Conclusão ao capítulo

Uma classificação dos sistemas de manufatura torna-se necessária para a distinção dos diferentes aspectos relacionados aos seus produtos e processos. Quando se processam peças ou produtos de diferentes famílias é importante a caracterização de cada fator responsável pelo desempenho do sistema.

Assim os diferentes tipos de flexibilidades ganham uma nova dimensão inerente e necessária a cada um dos processos, pois em razão destas características definem-se os objetivos de projeto para os diferentes sistemas de manufatura. Requisitos como flexibilidade de rotas e nível de sequenciamento das operações impõe ao sistema de controle uma maior importância aos

processos, uma vez que o seu poder de re-configuração contribui para a diminuição do tempo de processo e auxilia no aumento da concorrência entre os processos.

As máquinas e equipamentos atualmente utilizados são versáteis o bastante para desempenham várias operações. Sendo assim as tarefas de coordenação entre os subsistemas ganha a sua relevância em função da flexibilidade voltada aos sistemas de controle, pois as flexibilidades relacionadas aos sistemas físicos encontram-se em um ponto de gargalo. Deste modo, uma alternativa para se obter flexibilidade é atuando no objeto de inteligência do processo, neste caso, o sistema de controle.

Capítulo 3

Teoria de Controle Supervisório

Neste capítulo serão apresentados conceitos fundamentais e necessários ao entendimento, estudo, formalização e modelagem dos Sistemas a Eventos Discretos. As Teorias de Linguagens e Autômatos são definidas e descritas como as ferramentas formais para a concretização de tais propósitos. Como parte deste processo de estudo para a análise qualitativa dos sistemas de manufaturas são apresentadas as principais operações possíveis de serem realizadas sobre linguagens e autômatos. Como aspecto qualitativo pode-se considerar a distinção entre os estados de bloqueios e os estados considerados desejados à evolução do processo.

A não ocorrência de bloqueios e controlabilidade na linguagem gerada por uma sistema em malha fechada é verificada através da obtenção de um elemento supremo chamado *SupC* ou Máxima Linguagem Controlável. Este elemento representa que a linguagem gerada por um sistema sujeito à ação de um supervisor é controlável e com ausência de bloqueios. Sendo assim definem-se os elementos necessários para a obtenção de um supervisor, assim como as condições de controlabilidade da linguagem gerada sob supervisão.

Os elementos de controle chamados supervisores são obtidos de acordo com a Teoria de Controle Supervisório para os Sistemas a Eventos Discretos proposta por Ramadge e Wonham (1987) conforme a abordagem Modular Local (Queiroz e Cury, 2000). No desenvolvimento deste capítulo são consideradas duas abordagens: a centralizada, em que o conjunto das especificações determina a síntese de um supervisor monolítico com base numa planta global; e a abordagem modular local, a qual cada especificação determina a síntese de um supervisor modular de acordo com as respectivas plantas locais. As vantagens e desvantagens destas abordagens são relatadas considerando vários aspectos.

3.1 Teoria de Linguagens

Os sistemas de manufatura são regidos por eventos que sinalizam o início ou fim de uma atividade. No desenvolvimento destas atividades ocorre uma série de eventos que contribuem para que o sistema alcance um novo estado. Neste sentido, a representação da alternância ou seqüência de eventos através de linguagens tem o significado de descrever o comportamento dos sistemas a eventos discretos (Brandin, 2004).

Para Cassandras e Lafortune (1999), uma das maneiras formais de estudar o comportamento lógico de um SED é baseada na Teoria de Linguagens (Carroll e Long, 1989). *Um SED (Sistema a Eventos Discretos) é um sistema dinâmico que evolui de acordo com a ocorrência abrupta, em intervalos de tempos irregulares e desconhecidos, dos eventos físicos* (Ramadge e Wonham, 1987).

Dado um conjunto de eventos distintos Σ como o alfabeto de um SED, e assume-se que Σ seja finito. Entende-se por Σ^* o conjunto de todas as palavras possíveis e factíveis formadas pelos eventos constituintes deste alfabeto. Uma seqüência de eventos sobre este alfabeto forma uma "palavra". Uma palavra que consiste apenas do evento ε é chamada de "palavra vazia" (Cassandras e Lafortune, 1999).

Uma linguagem definida sobre um alfabeto Σ é um conjunto de palavras de comprimento finito sobre este alfabeto. Seja $\Sigma = \{a, b, g\}$ o conjunto de eventos, define-se a linguagem $L1 = \{\varepsilon, a, abb\}$ que consiste de três palavras.

Seja $s \in \Sigma^*$ sendo $s = tuv$ com $t, u, v \in \Sigma^*$, então:

- t é chamado de *prefixo* de s ,
- u é chamado de *subpalavra* de s , e
- v é chamado de *sufixo* de s .

3.2 Operações sobre linguagens

O processo de análise das linguagens se desenvolve de acordo com um conjunto de operações. Tais operações são em função do alfabeto de eventos característicos de cada sistema. Cassandras e Lafortune (1999), definem três operações sobre linguagens que são:

a) O *Fechamento de Kleene* de $L \subseteq \Sigma^*$ é definido como a união de todas as possíveis palavras, formadas pelos símbolos pertencentes a um alfabeto, incluindo a palavra vazia ε , e pode ser expresso por:

$$L^* := \{\varepsilon\} \cup L \cup LL \cup LLL \cup \dots$$

ou seja, $A^* = \bigcup_{n=0}^{\infty} A^n$ onde: $A^0 = \{\varepsilon\}$

$$A^n = A^{n-1}A$$

b) A *concatenação* de duas linguagens $L_a, L_b \subseteq \Sigma^*$, é definida (Carroll e Long, 1989) como a justaposição de duas linguagens, dando origem a uma linguagem maior, que é composta dos símbolos pertencentes a primeira linguagem, imediatamente seguidos dos símbolos pertencentes a segunda linguagem, e pode ser expressa por:

$$L_a L_b := \{s \in \Sigma^* : (s = s_a s_b) \text{ e } (s_a \in L_a) \text{ e } (s_b \in L_b)\}$$

c) O *prefixo – fechamento* de uma linguagem $L \subseteq \Sigma^*$ é definido como o conjunto de todos os prefixos de uma dada linguagem incluindo a palavra vazia ε , e pode ser definido por,

$$\bar{L} := \{s \in \Sigma^* : \exists t \in \Sigma^* (st \in L)\}$$

logo $\bar{\bar{L}} = \{ \varepsilon, a, ab, abb \}$, em geral $L \subseteq \bar{L}$ e, L é dito ser *prefixo – fechada* se $L = \bar{L}$.

3.3 Teoria de Autômatos

Um autômato finito é definido como o modelo matemático de uma máquina que aceita um conjunto particular de palavras sobre um determinado alfabeto Σ (Carroll e Long, 1989).

De acordo com Carroll e Long (1989), dado um alfabeto de entrada Σ , para um Autômato Finito Determinístico G , cada símbolo sucessivo em uma palavra causará uma transição do estado presente para um outro estado na máquina. Há somente um estado para cada transição de estado em cada combinação de símbolos $e \in \Sigma$, e um estado $q \in Q$. O termo estado representa as condições nas quais um processo se encontra em determinado instante como em repouso ou em atividade desenvolvendo algum tipo de operação.

Um Gerador (Ramadge e Wonham, 1987), pode ser descrito da seguinte forma:

$$G = (Q, \Sigma, \delta, Q_m, q_0)$$

Q é conjunto de estados q .

Σ é o conjunto finito de eventos associados às transições em G .

$\delta: Q \times \Sigma \rightarrow Q$ é a *função de transição*: $\delta(x, e) = y$ significa que há uma transição rotulada por um evento e do estado x para o estado y .

$Q_m \subseteq Q$ é o conjunto de estados marcados (que representam tarefas executadas) e $q_0 \in Q$ é o estado inicial.

Considere o grafo da figura 3.1 (Cassandras e Lafortune, 1999), onde os nós representam os *estados* e os arcos rotulados representam *transições* entre estes estados. Este grafo dá uma descrição completa de um autômato. O conjunto de nós é o conjunto de estados do sistema automatizado, $Q = \{x, y, z\}$. O conjunto de rótulos para as transições é o conjunto de eventos (alfabeto) do autômato, $\Sigma = \{a, b, g\}$. Os arcos no grafo dão uma representação gráfica da *função de transição* do autômato, que denota-se como $\delta: Q \times \Sigma \rightarrow Q$:

$$\begin{array}{ll} \delta(x, a) = x & \delta(x, g) = z \\ \delta(y, a) = x & \delta(y, b) = y \\ \delta(z, b) = z & \delta(z, a) = f(z, g) = y \end{array}$$

A notação $\delta(y, a) = x$ quer dizer que se o autômato está no estado y , então quando da ocorrência do evento a , o autômato fará uma transição instantânea para o estado x .

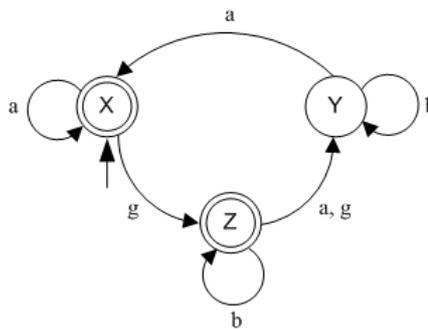


Figura 3.1: Exemplo de um autômato determinístico.

Três observações devem ser feitas com relação a este exemplo. Primeiro, um evento pode ocorrer sem mudança de estado, como em $\delta(x, a) = x$. Segundo, dois eventos distintos podem

ocorrer num dado estado causando a mesma transição como em $\delta(z, a) = \delta(z, g) = y$. Terceiro, a função δ é uma função parcial sobre seu domínio $Q \times \Sigma$, isto é, não é necessário uma transição ser definida para cada evento em Σ para cada estado de Q , por exemplo, $\delta(x, b)$ e $\delta(y, g)$ não são definidas.

O estado inicial, denotado por q_0 , e o subconjunto Q_m que representa os estados *marcados* de Q , são elementos essenciais para uma definição completa de um autômato. Os estados são marcados quando se deseja associar a este um significado especial, como por exemplo, a conclusão de uma tarefa. Os estados marcados são também referenciados como estados *finais*.

Em todos os estados q e Q_m , pertencentes ao conjunto Q , é preciso definir a função $\Gamma: Q \rightarrow 2^\Sigma$ que de acordo com Cassandras e Lafortune (1999), é a *função de eventos ativos* (ou função de eventos factíveis); $\Gamma(q)$ é o conjunto de todos os eventos e para os quais $\delta(q, e)$ é definida e chamada de *conjunto de eventos ativos* (ou conjunto de eventos factíveis) de G em q .

Um autômato G conforme Cassandras e Lafortune (1999) opera como segue. No seu estado inicial q_0 e após a ocorrência de um evento $e \in \Gamma(q_0) \subseteq \Sigma$ ele fará uma transição para o estado $\delta(q_0, e) \in Q$. Este processo então continua fundamentado nas transições para as quais δ é definida. A função δ é sempre expandida do domínio $Q \rightarrow \Sigma$ para o domínio $Q \rightarrow \Sigma^*$ da seguinte forma:

$$\delta(q, \varepsilon) := q$$

$$\delta(q, se) := \delta(\delta(q, s), e) \text{ para } s \in \Sigma^* \text{ e } e \in \Sigma.$$

Retornando ao autômato da figura (3.1), tem-se como exemplo que:

$$\delta(y, \varepsilon) = y$$

$$\delta(x, gba) = \delta(\delta(x, gb), a) = \delta(\delta(\delta(x, g), b), a) = \delta(\delta(z, b), a) = \delta(z, a) = y$$

$$\delta(x, aagb) = z$$

$$\delta(z, b^n) = z \text{ para todo } n \geq 0$$

Onde b^n denota n ocorrências consecutivas do evento b .

3.4 Linguagens Geradas e Marcadas

O comportamento de uma planta é caracterizado por duas linguagens $L(G)$ e $L_m(G)$. $L(G)$ representa o comportamento livre da planta, ou seja, todas as seqüências de eventos que ela pode gerar chamada de *linguagem gerada*, sendo formalmente definida de acordo com Ramadge e Wonham (1987) como:

$$L(G) := \{ s \in \Sigma^* : \delta(q_0, s) \text{ é definida} \}$$

A linguagem $L_m(G)$ representa o comportamento marcado, ou seja, tarefas completas executadas pela planta chamada de *linguagem marcada*, sendo a sua definição conforme Ramadge e Wonham (1987) é:

$$L_m(G) := \{ s \in L(G) : \delta(q_0, s) \in Q_m \},$$

estas duas linguagens são executadas sobre o alfabeto Σ e fazem parte do conjunto Σ^* . Diz-se que um autômato é *não-bloqueante* quando, $\overline{L_m(G)} = L(G)$ (Ramadge e Wonham, 1987).

Exemplo 3.1

Seja $\Sigma = \{a, b\}$ um conjunto de eventos (Cassandras e Lafortune, 1999). Considere a linguagem L como sendo $L = \{a, aa, ba, aaa, aba, baa, bba, \dots\}$, consistindo de todas as palavras de a e b sempre seguidas por um evento a . Esta linguagem é marcada pelo autômato de estado finito $G = (Q, \Sigma, \delta, Q_m, q_0)$ onde $Q = \{0, 1\}$, $q_0 = 0$, $Q_m = \{1\}$ e δ é definido como: $\delta(0, a) = 1$, $\delta(0, b) = 0$, $\delta(1, a) = 1$, $\delta(1, b) = 0$.

No autômato, apresentado na figura 3.2, tem-se o estado 0 (zero) como estado inicial e o estado 1 (um) como estado marcado. Verifica-se que $L_m(G) = L$. Nota-se que δ é uma função total, ou seja, para todo evento de Σ é definida uma transição em cada estado de Q e além disso a linguagem gerada por G é $L(G) = \Sigma^*$.

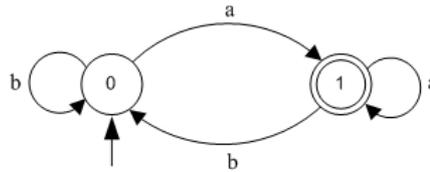
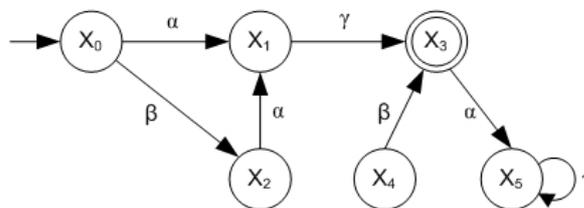


Figura 3.2: Autômato que marca a linguagem consistindo de todas as palavras iniciadas por a e b seguidas por a .

Ao remover-se do autômato acima o auto-laço referente ao evento b no estado 0 (zero), isto é, tornar $\delta(0, b)$ indefinida, então $L(G)$ agora consiste daquelas palavras em Σ^* que iniciam com o evento a e nas quais não existe a ocorrência consecutiva de b . Neste caso, ou o evento b é o último evento da palavra, ou ele será seguido imediatamente por um evento a .

3.5 Operações sobre Autômatos

De modo a estudar os Sistemas a Eventos Discretos modelados por Autômatos, necessita-se de um conjunto de operações que permitam combinar ou compor dois ou mais autômatos, bem como analisar as propriedades destes. No autômato da figura 3.3 pode-se atentar para dois estados em particular, X_4 e X_5 , em função dos quais serão definidos os conceitos de acessibilidade e co-acessibilidade respectivamente:



X_4 : estado *não - acessível*

X_5 : estado *não co - acessível*

Figura 3.3: Autômato não-acessível e não co-acessível .

3.5.1 Acessibilidade

Um estado é *acessível* (Ramadge e Wonham, 1987) se este pode ser alcançado através de qualquer cadeia $s \in L(G)$ tendo como origem o estado $q_0 \in Q$. Das definições de $L(G)$ e $L_m(G)$, observa-se que é possível apagar de G todos os estados que não são *acessíveis* ou *alcançáveis* a partir de q_0 através de alguma palavra de $L(G)$, sem afetar as linguagens geradas e marcadas por G .

Um autômato acessível é aquele em que todos os estados podem ser alcançados a partir do estado inicial q_0 .

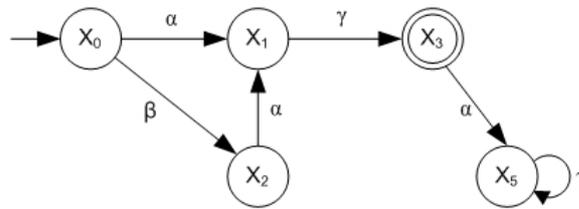


Figura 3.4: Autômato acessível.

Quando se apaga um estado, também são extintas todas as transições ligadas àquele estado. Chama-se esta operação por $Ac(G)$, onde Ac representa a parte *acessível*. Formalmente,

$$Ac(G) := (Q_{ac}, \Sigma, \delta_{ac}, q_0, Q_{ac,m})$$

$$Q_{ac} = \{q \in Q : \exists s \in \Sigma^* (\delta(q_0, s) = q)\}$$

$$Q_{ac,m} = Q_m \cap Q_{ac}$$

$$\delta_{ac} = \delta |_{Q_{ac} \times \Sigma \rightarrow Q_{ac}}$$

A notação $\delta |_{Q_{ac} \times \Sigma \rightarrow Q_{ac}}$ quer dizer que restringe-se δ ao menor domínio dos estados acessíveis Q_{ac} . Claramente, a operação Ac não tem efeito sobre $L(G)$ e $L_m(G)$. Então assume-se sempre que um autômato é *acessível* se $G = Ac(G)$.

3.5.2 Co-acessibilidade

Um estado q de G é dito ser *co-acessível* para Q_m , se há uma cadeia em $L_m(G)$ com início neste estado q e final em um estado $q \in Q_m$ (Ramadge e Wonham, 1987). No exemplo anterior tem-se que o estado X_5 é não co-acessível, já que não existe um caminho de X_5 para um estado marcado. Denota-se a operação de apagar todos os estados de G que não são *co-acessíveis* por $CoAc(G)$; onde $CoAc$ leva em conta a parte *co-acessível* (Cassandras e Lafortune, 1999).

$$CoAc(G) := (Q_{coac}, \Sigma, \delta_{coac}, q_{0,coac}, Q_m)$$

$$Q_{coac} = \{q \in Q : \exists s \in \Sigma^* (\delta(q, s) \in Q_m)\}$$

$$q_{0,coac} = \begin{cases} q_0 & \text{se } q_0 \in Q_{coac} \\ \text{indefinido de outra forma} \end{cases}$$

A operação $CoAc$ pode alterar $L(G)$, desde que podem ser apagados estados que são acessíveis. Contudo a operação $CoAc$ não afeta $L_m(G)$, desde que algum estado apagado não esteja em algum caminho de q_0 para Q_m . Se $G = CoAc(G)$, então G é dito ser *co-acessível*, neste caso $L(G) = \overline{L_m(G)}$ (Cassandras e Lafortune, 1999).

3.5.3 Operação Trim

Um autômato *acessível* e *co-acessível* ao mesmo tempo é dito *Trim* (Ramadge e Wonham, 1987). Ele representa a ausência de bloqueios no sistema, isto é, a partir do estado inicial q_0 ou de qualquer outro estado, sempre existirá um caminho que conduz a um estado marcado. Observa-se este comportamento na figura 3.5 abaixo,

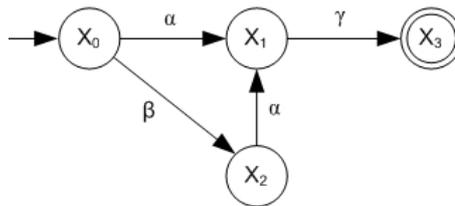


Figura 3.5: Autômato Trim.

Define-se a operação *Trim* como segue:

$$\text{Trim}(G) := \text{CoAc}[\text{Ac}(G)] = \text{Ac}[\text{CoAc}(G)]$$

3.5.4 Composição Paralela

A composição paralela é uma operação que representa o comportamento sincronizado entre dois autômatos. Nesta operação um evento comum aos dois alfabetos de eventos, é permitido ocorrer se e somente se ele ocorre em ambos os autômatos simultaneamente. No caso de um evento pertencer a somente um dos alfabetos, ele sempre estará habilitado a ocorrer no autômato resultante quando se realizar a composição. Desta forma o sincronismo se observa somente quando analisamos eventos comuns a ambos os alfabetos (Cassandras e Lafortune, 1999).

Um estado do autômato composto será marcado se ele resultar de estados marcados em ambos os geradores. A composição paralela de dois autômatos G_1 e G_2 é definida como abaixo,

$$G_1 \parallel G_2 := \text{Ac}(Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \delta, (q_{01}, q_{02}), Q_{m1} \times Q_{m2})$$

onde

$$f((x_1, x_2), e) := \begin{cases} (f_1(x_1, e), f_2(x_2, e)) & \text{se } e \in \Gamma_1(x_1) \cap \Gamma_2(x_2) \\ (f_1(x_1, e), x_2) & \text{se } e \in \Gamma_1(x_1) \setminus \Sigma_2 \\ (x_1, f_2(x_2, e)) & \text{se } e \in \Gamma_2(x_2) \setminus \Sigma_1 \\ \text{indefinida de outra forma} & \end{cases}$$

Sejam os autômatos G_1 e G_2 mostrados na figura 3.6, sendo o estado inicial 0(zero) significando o estado de repouso e o estado 1(um), o estado representativo do sistema em operação. Os eventos (α_1 , α_2) representam início de operação, e os eventos (β_1 , β_2) significam final de operação.



Figura 3.6: Autômatos G_1 e G_2 .

Na composição paralela de G_1 e G_2 , obtém-se o autômato da figura 3.7 :

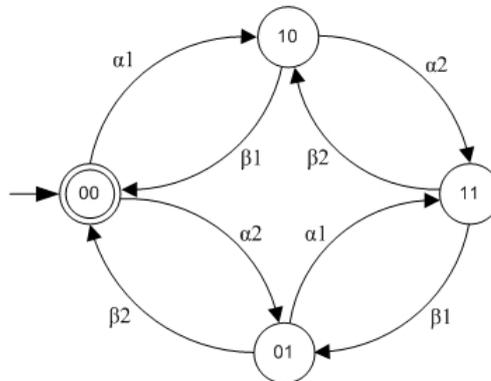


Figura 3.7: Autômato $G_1||G_2$.

Observa-se na figura 3.7 que em função dos autômatos G_1 e G_2 serem assíncronos, ou seja, não possuem eventos em comum, a evolução do autômato resultante determina que todos os eventos definidos em ambos os autômatos também estarão definidos em $(Q_1 \times Q_2)$. Por exemplo, o estado (01) representa a sincronização do estado 0(zero) de G_1 , com o estado 1(um) de G_2 . Isto significa que os eventos α_1 e β_2 deverão estar definidos.

3.6 Teoria de Controle Supervisório

Os sistemas de manufatura são constituídos de vários subsistemas que através de uma inter-relação dinâmica buscam atingir objetivos globais para o seu bom funcionamento. A tarefa de coordenação entre estes subsistemas é o objetivo principal do controle supervisório de modo a diminuir o tempo de execução das operações globais, garantir a ausência de bloqueios e não violar as especificações de segurança (Queiroz e Cury, 2002).

De acordo com Brandin e Charbonnier (1994) duas estratégias principais são utilizadas para os propósitos de um sistema de controle supervisório:

- 1) Supervisão Centralizada, a qual implica que toda a tarefa do sistema supervisório é executada por um único supervisor que incorpora o comportamento em malha fechada da planta. Além disto, tal supervisor é provido de informações completas sobre a ocorrência de eventos na planta.
- 2) Supervisão Modular, a qual implica na divisão de toda a tarefa do sistema supervisório em dois ou mais supervisores que possuem uma visão parcial da planta, atuando concorrentemente.

No modelo de controle supervisorio proposto por Ramadge e Wonham (1989), o sistema é dividido em duas estruturas, a planta e um supervisor. Assume-se que a planta gera eventos de forma espontânea e assíncrona no tempo. O supervisor, por sua vez, exercerá controle sobre a planta não permitindo dinamicamente a ocorrência de um conjunto mínimo de eventos e executando as especificações desejadas. Sendo assim, um supervisor é projetado para ser *maximamente permissível* na Teoria de Controle Supervisorio (Kumar *et al.*, 2005). Este modelo é particularmente adequado para o propósito da tarefa de projeto de supervisores, uma vez que o supervisor resultante sempre garante satisfazer os objetivos de controle.

3.6.1 Supervisão Centralizada (Controle Monolítico)

O objetivo do controle supervisorio monolítico é projetar um único supervisor cuja função é habilitar ou desabilitar eventos controláveis de acordo com a seqüência de eventos observados na planta. Desta forma o sistema em malha fechada impõe algumas regras operacionais especificadas (Queiroz e Cury, 2002a).

De modo a modelar um sistema de controle Ramadge e Wonham (1989) particionam o conjunto de eventos Σ em dois conjuntos, *eventos controláveis* (Σ_c) e *eventos não-controláveis* (Σ_u), de modo que $\Sigma = \Sigma_c \cup \Sigma_u$. Os eventos Σ_c podem ser desabilitados em qualquer momento pelo supervisor, ao passo que os eventos Σ_u não são afetados pela ação do supervisor.

Um subconjunto de eventos γ que estão aptos a ocorrer em qualquer momento, na operação da planta, é definido através da determinação das entradas de controle. Uma entrada de controle para um gerador G consiste do subconjunto $\gamma \subseteq \Sigma$ que satisfaça $\Sigma_u \subseteq \gamma$. Esta condição $\Sigma_u \subseteq \gamma$ significa que qualquer evento não-controlável sempre será permitido ocorrer, ou seja, não está sujeito à ação de desabilitação por parte do supervisor. O conjunto das entradas de controle é denotado por $\Gamma \subseteq 2^\Sigma$ (Ramadge e Wonham, 1989).

Formalmente um supervisor é um mapa que corresponde à seguinte função $f: L(G) \rightarrow \Gamma$, que associa à cada cadeia $s \in L(G)$ gerada pela planta, uma entrada de controle $\gamma = f(s)$ que restringe o comportamento da planta até a ocorrência de um novo evento. Quando um SED está sob a ação de controle de um supervisor f , ele obedece às restrições adicionais. Após a geração de uma palavra w pela planta, o próximo evento deve ser um elemento de

$f(w) \cap \Sigma(\delta(w, q_0))$. Define-se $\Sigma(q)$ como conjunto de eventos factíveis no estado q com $\Sigma(q) \subseteq \Sigma$, de modo que para cada $\sigma \in \Sigma(q)$, $\delta(\sigma, q)$ é definida. Portanto o evento a ser gerado após a palavra w deverá ter simultaneamente a sua entrada de controle habilitada pelo supervisor e a função de transição definida na planta satisfazendo $f(w) \cap \Sigma(\delta(w, q_0))$ (Ramadge e Wonham, 1989).

O comportamento do sistema composto pela planta G , sujeita à ação de um supervisor f , é denotado por (f/G) de acordo com Ramadge e Wonham (1989). Seja $L(f/G)$ a linguagem que representa o comportamento do sistema sob supervisão, sendo $L(f/G) \subseteq L(G)$ definida formalmente como segue:

i) $\varepsilon \in L(f/G)$

ii) $w\sigma \in L(f/G)$ se e somente se $w \in L(f/G)$, $\sigma \in f(w)$ e $w\sigma \in L(G)$

A linguagem marcada $L_m(f/G)$ e que representa o comportamento marcado do sistema sob supervisão é definida como:

$$L_m(f/G) = L_m(G) \cap L(f/G)$$

Ou seja, ela representa o conjunto de tarefas completas realizadas pela planta e que sobrevivem à ação de controle do supervisor.

Um supervisor f é dito ser *não-bloqueante* para um gerador G , se $\overline{L_m(f/G)} = L(f/G)$, o que significa que a linguagem gerada pelo sistema sob supervisão deverá ser marcada e prefixo-fechada. Isto implica que, as palavras ou seqüências de eventos geradas pelo sistema sob supervisão sempre deverão evoluir no sentido de concluir-se uma tarefa imposta pelo sistema.

O supervisor que afeta o comportamento da planta ao habilitar e desabilitar eventos controláveis pode ser descrito, ou representado na prática, por um autômato T (Ramadge e Wonham, 1987):

$$T = (X, \Sigma, \xi, X_m, x_0)$$

Onde X é o conjunto de estados x , Σ é o alfabeto utilizado por G , ξ função de transição, X_m é o conjunto de estados marcados e x_0 é o estado inicial.

Seja $\Phi : X \rightarrow \Gamma$ a função que mapeia os estados de T em entradas de controle para G . Diz-se que o par (T, Φ) executa o supervisor f se para cada $w \in L(f/G)$

$$\Phi(\xi : X \times \Sigma^* \rightarrow X) = f(w)$$

Isto significa que o valor de f para uma dada seqüência de eventos w gerada pela planta G é obtido aplicando-se w ao autômato T , a partir do seu estado inicial x_0 , obtendo-se um novo estado x' , cuja função de transição é definida por $x' = \xi : X \times \Sigma^* \rightarrow X$. Assim a função Φ fornece a entrada de controle que deverá ser aplicada a seqüência de eventos w , ou seja, $f(w) = \gamma = \Phi(x')$.

De modo prático, o supervisor f será representado graficamente pelo autômato T , onde as ações de controle de f sobre a planta G estarão definidas na estrutura de transição de T . Os eventos não habilitados pelo supervisor f não aparecerão na estrutura de transição de estados do autômato T , assim, na operação de composição síncrona, como os alfabetos são comuns, um evento só poderá ocorrer, se ele ocorrer simultaneamente em ambos os autômatos de T e G . Desta forma, dado o autômato T que representa f , se uma dada cadeia $s \in L(f/G)$, então $s \in L(T)$ e $s\sigma \in L(T)$ somente se $\sigma \in f(s)$. Ainda assim, se $s \in L(f/G)$, $s\sigma \in L(G)$, e $\sigma \in f(s)$, então $s\sigma \in L(T)$. A primeira afirmação assegura que as transições desabilitadas por f , não aparecem na estrutura de transição de T , ao passo que a segunda, assegura que as transições habilitadas por f , e que são factíveis em G , aparecerão na estrutura de transição de T .

A figura 3.8 mostra a estrutura de um sistema de controle a eventos discretos com supervisão monolítica. Observa-se que a planta gera os eventos controláveis e não-controláveis, cabendo ao supervisor a ação de desabilitar somente os eventos controláveis.



Figura 3.8: Arquitetura do controle monolítico (Ramadge e Wonham, 1989).

Assume-se que os autômatos T (Supervisor) e G (planta), são executados de forma concorrente, ou seja, estando os autômatos em um determinado estado $(x, q) \rightarrow (x', q')$, a

ocorrência de um evento σ determina que o autômato T evolui de $x \rightarrow x'$, e o autômato G de $q \rightarrow q'$.

3.6.1.1 Controlabilidade

A controlabilidade de uma linguagem diz respeito à ação do sistema de controle (supervisão) sobre a planta, mantendo-o dentro de um padrão de comportamento considerado desejado sob o ponto de vista de cumprimento das especificações comportamentais. Este aspecto é considerado de relevância quando se deseja alterar o comportamento em malha aberta da planta para se atingir os objetivos de desempenho qualitativos, como por exemplo, ausência de bloqueios quando se observa o sistema em malha fechada.

Dado um autômato $G = (Q, \Sigma, \delta, Q_m, q_0)$ (por definição *Trim*) que representa o comportamento controlado e considerando-se que $K \subset \Sigma^*$, $L \subset \Sigma^*$ sejam linguagens arbitrárias. Por definição, $\Sigma = \Sigma_u \cup \Sigma_c$ e sendo $\Sigma_u = \Sigma - \Sigma_c$ o conjunto de eventos não-controláveis não sujeitos à ação do supervisor, pode-se dizer que K é controlável se $K \subset L(G)$ e $\overline{K}\Sigma_u \cap L(G) \subset \overline{K}$ (Ramadge e Wonham, 1987).

Sabe-se que \overline{K} é uma linguagem que consiste das palavras de K e todos os prefixos de K , incluindo a palavra vazia ε . Uma linguagem K contida em L é *L-fechada* se os prefixos de K são também palavras em L .

A linguagem $\overline{K}\Sigma_u \cap L$ consiste de todas as palavras $s' = s\sigma$, sendo $s' \in L$, $s \in \overline{K}$ e $\sigma \in \Sigma_u$. Seja L a representação do comportamento fisicamente possível e \overline{K} o comportamento desejado sob supervisão, toda palavra s aceita pela planta e por \overline{K} , imediatamente seguida de um evento não-controlável σ , deverá manter-se dentro do conjunto de \overline{K} . Desta forma $s\sigma$ deverá representar o comportamento fisicamente possível e desejado de \overline{K} (Ramadge e Wonham, 1987).

A classe de linguagens controláveis contidas em M , sendo $M \subseteq \Sigma^*$, é denotada por $C(M, G) = \{K \mid K \subseteq M \text{ e } K \text{ é controlável e.r.a } G\}$ sendo $K \subseteq \Sigma^*$, é fechada sobre o conjunto união e contém um (único) elemento supremo, chamado $\sup C(M, G)$. Este elemento é a

maior linguagem controlável, ou seja, a menos restritiva, dentre as linguagens contidas em K , e denotada por $K^\uparrow \subseteq K$ (Ramadge e Wonham, 1987).

3.6.1.2 Problema de Controle Supervisório

Sejam as linguagens $L_a, L_g \subset \Sigma^*$, onde L_a representa o comportamento mínimo admissível pela planta e L_g o comportamento desejado para a planta, com $\emptyset \neq L_a \subset L_g \subset L_m(G)$. O *Problema de Controle Supervisório*, de acordo com Ramadge e Wonham (1987), consiste em sintetizar um supervisor tal que a linguagem do sistema sob supervisão, e que sobrevive à ação do supervisor $L_c(f/G)$ seja,

$$L_a \subset L_c(f/G) \subset L_g$$

Este problema tem solução quando para uma sublinguagem K de L_g e com um comportamento mínimo $L_a \subseteq L_m(G)$, temos que o elemento supremo $\sup C(K, G)$ representativo da linguagem minimamente restritiva sob supervisão, aceita $L_a \subseteq \sup C(K, G)$. E o *problema de controle supervisório* é dito ter solução.

3.6.2 Supervisão Modular

De acordo com Ramadge e Wonham (1989) o processo de síntese de um supervisor monolítico apresenta algumas limitações que inviabilizam a sua implementação que são:

- i. O número de especificações de controle e subsistemas envolvidos aumentam, de forma exponencial, a complexidade dos cálculos computacionais para a síntese de um supervisor;
- ii. O número de estados do supervisor obtido também aumenta de forma exponencial quanto ao número de especificações de controle e subsistemas considerados. Isto dificulta interpretações de comportamentos relativas ao sistema, devido ao elevado número de informações presentes;
- iii. Para qualquer alteração que seja necessária, referente à inclusão ou exclusão de subsistemas e modificação de especificações, é preciso restabelecer novamente todos os

procedimentos para a obtenção de um novo supervisor, uma vez que todo o modelo deve ser alterado;

- iv. Todas estas limitações citadas dificultam propósitos de implementação em dispositivos de controle, assim como facilidades para manutenção e re-projetos.

Com a supervisão modular procura-se resolver o problema de explosão do espaço de estado através da síntese de um supervisor para cada especificação de comportamento. Na abordagem monolítica, a controlabilidade das especificações é verificada observando-se a planta global, ao passo que na abordagem modular local este procedimento é determinado verificando-se a planta local. Uma planta local é constituída dos subsistemas que contém os eventos que definem uma especificação. Sob este aspecto consideram-se os subsistemas da planta em sua Representação Sistema Produto (RSP), em que os mesmos são assíncronos, isto é, não compartilham eventos. Em ambas as abordagens a atuação conjunta e concorrente dos vários supervisores terá o mesmo efeito de um supervisor monolítico. Neste sentido será preciso verificar a modularidade de cada uma das máximas linguagens controláveis geradas pelas especificações locais.

Controle Modular Local

De acordo com (Queiroz e Cury, 2002), o projeto de sistemas de manufatura envolve uma grande quantidade de especificações e a abordagem de controle modular pode ser freqüentemente utilizada para a síntese de controladores. Ao invés de se projetar um único supervisor monolítico que satisfaça todas as especificações, procura-se construir um supervisor para cada especificação. Neste sentido, deseja-se que o comportamento conjunto e concorrente dos supervisores modulares têm a mesma ação de controle que a de um supervisor monolítico.

A ação de controle conjunta dos supervisores é verificada através da composição síncrona de todos os supervisores modulares resultantes e tal procedimento é denominado teste de modularidade. Quando a propriedade de modularidade é verificada, a abordagem de controle modular é bastante vantajosa no sentido de promover maior flexibilidade, maior eficiência computacional e segurança na aplicação do controle.

Um funcionamento conjunto dos supervisores localmente modulares é ilustrado na figura 3.9, observa-se que os supervisores locais observam e controlam apenas o comportamento dos subsistemas afetados pela especificação genérica. Assim, pode-se afirmar que a abordagem de

controle modular local induz uma estrutura de controle descentralizada que surge naturalmente do processo de síntese.

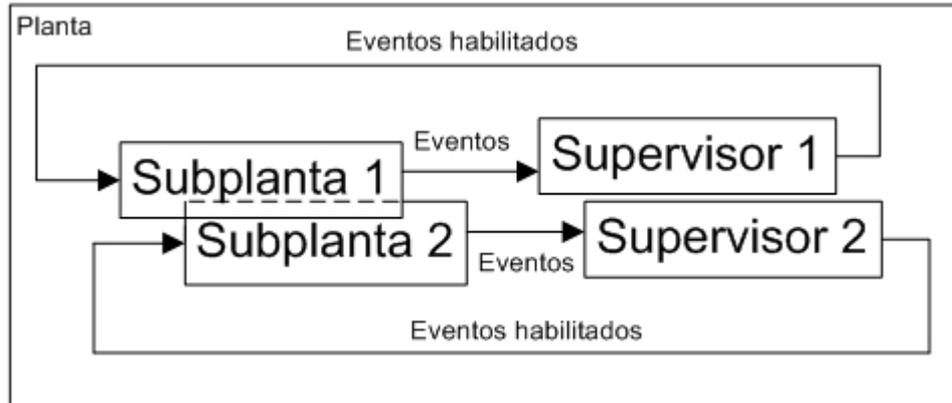


Figura 3.9: Arquitetura de controle modular local (Queiroz e Cury, 2000).

Seja a Representação por Sistema Produto (Ramadge e Wonham, 1989) de um sistema de manufatura formada por subsistemas $G_i = (\Sigma_i, Q_i, \delta_i, q_{0i}, Q_{mi})$, para $i \in I = \{1, \dots, n\}$. Para tal sistema obtém-se um número $j \in J = \{1, \dots, m\}$ de especificações genéricas. Sejam as especificações genéricas locais E_{x_j} definidas em $\Sigma_{x_j} \subseteq \Sigma$. Para $j=1, \dots, m$ a planta local $G_{x_j} = (\Sigma_{x_j}, Q_{x_j}, \delta_{x_j}, q_{0x_j}, Q_{mx_j})$, associada a especificação E_{x_j} é definida por $G_{x_j} = \parallel_{i \in I_{x_j}} G_i$, com $I_{x_j} = \{k \in I \mid \Sigma_k \cap \Sigma_{x_j} \neq \emptyset\}$.

Assim, sejam as linguagens $L_i \subseteq \Sigma_i^*$, $i \in I = \{1, \dots, n\}$. O conjunto de linguagens $\{L_i, i=1, \dots, n\}$ é dito *localmente modular* se $\parallel_{i=1}^n \bar{L} = \overline{\parallel_{i=1}^n L_i}$ (Queiroz e Cury, 2002a). Se para $i=1, \dots, n$ as linguagens forem marcadas por geradores *trim* G_i , é fácil provar que a modularidade local é verificada se e somente se $G = \parallel_{i=1}^n G_i$ for um gerador *trim*, o que implica a ausência de estados ou ciclos de bloqueio. Pode-se mostrar também que, para quaisquer $L_i \subseteq \Sigma_i^*$, $i \in I = \{1, \dots, n\}$, é sempre verdade que $\overline{\parallel_{i=1}^n L_i} \subseteq \parallel_{i=1}^n \bar{L}_i$, o que significa que um conjunto de linguagens $\{L_i, i=1, \dots, n\}$ é localmente modular se e somente se $\parallel_{i=1}^n \bar{L}_i \subseteq \overline{\parallel_{i=1}^n L_i}$.

O principal resultado da abordagem de controle modular local (Queiroz e Cury, 2002a) é enunciado abaixo da seguinte forma: *Dados um sistema com RSP formada por G_j , $j = 1, \dots, m$, e as especificações genéricas E_{xj} , $j = 1, \dots, m$. Sejam E_{xj} , K_{xie} , G_{xi} e G_e definidos como nesta seção. Se $SupC(E_{xi}, L_m(G_{xi}))$, $i = 1, \dots, n$ são localmente modulares, então $SupC(\prod_{i=1}^n K_{xie}, L_m(G_e)) = \prod_{i=1}^n SupC(E_{xi}, L_m(G_{xi}))$.*

Assim, se o conjunto de supervisores locais calculados forem localmente modulares, o comportamento obtido pela supervisão das plantas locais é não bloqueante e equivalente ao comportamento obtido por supervisão monolítica. O procedimento de síntese de um supervisor considerando a abordagem modular local de acordo com Vieira (2004) apresenta vantagens em relação à abordagem monolítica que são:

- i. A complexidade computacional envolvida para o cálculo de síntese de um supervisor local é consideravelmente menor que a de um supervisor monolítico (Queiroz 2000);
- ii. Facilidade de interpretação do comportamento imposto pelas especificações devido ao menor número de estados do supervisor obtido;
- iii. O menor número de estados possibilita a obtenção de supervisores reduzidos que desempenham as mesmas ações de controle;
- iv. A inclusão ou exclusão de subsistemas ou a modificação de uma especificação de controle, somente determina a alteração do supervisor afetado não sendo necessária à alteração dos outros supervisores;
- v. A exclusão de uma especificação somente implica na remoção do supervisor local correspondente;
- vi. A abordagem modular local preserva a modularidade física dos sistemas e das especificações de controle, uma vez que somente as partes afetadas estão sujeitas às alterações.

A metodologia proposta por Queiroz e Cury (2002), que consiste na síntese de supervisores para sistemas automatizados de manufatura pode ser descrita sinteticamente pela seqüência de passos abaixo:

1. Modelar cada componente elementar do sistema isoladamente, evitando incluir eventos e estados desnecessários;
2. Calcular a mais refinada representação por sistema produto (RSP), fazendo a composição dos subsistemas síncronos, que compartilham eventos;

3. Modelar cada especificação isoladamente, considerando apenas os eventos relevantes;
4. Obter a planta local para cada especificação compondo os subsistemas da RSP que tenham eventos em comum com a especificação;
5. Calcular a linguagem de cada planta local que satisfaça a especificação, através do produto síncrono de cada planta local com sua respectiva especificação genérica;
6. Calcular a máxima linguagem controlável contida em cada especificação local;
7. Verificar a modularidade local das linguagens resultantes;
8. Se não forem modulares, resolver os conflitos definindo novas especificações ou eliminando os estados bloqueantes, por exemplo;
9. Se forem modulares, implementar um supervisor reduzido para cada linguagem controlável.

3.7 Conclusão ao capítulo

Apresentou-se neste capítulo as formas de representação matemática do comportamento dos Sistemas a Eventos Discretos. Tais representações são de acordo com as Teorias de linguagens e Autômatos. Sendo assim, as principais operações sobre linguagens e autômatos foram abordadas, para que seja possível uma interpretação quando forem analisados os comportamentos de vários subsistemas de forma conjunta. Estes sistemas compostos representam o comportamento da planta física.

Duas estratégias de controle foram apresentadas para confinar o comportamento da planta física um comportamento desejado. A primeira diz respeito à abordagem monolítica, que resulta num esforço computacional maior quando se desenvolve o processo de síntese do supervisor. Este supervisor é resultado da composição síncrona de todo o conjunto de especificações de controle para a planta em questão. Esta planta por sua vez é global, uma vez que considera o comportamento simultâneo de todos os subsistemas constituintes. O elemento de controle a ser obtido resulta em um elevado número de estados, o que dificulta aspectos de interpretação, re-configuração e implementação em CLP's.

A segunda estratégia é de acordo com a abordagem Modular Local que será objeto de utilização no desenvolvimento deste trabalho. Esta abordagem possibilita a utilização de elementos de controle reduzidos, uma vez que considera plantas locais, que envolvem um número menor de subsistemas. Ao invés de um único supervisor, obtém-se vários supervisores

menores, sendo um para cada especificação, mas que possuem a mesma ação de controle, caso a sua modularidade seja verificada.

A dificuldade computacional, para a verificação de tal propriedade, se restabelece pois envolve um número elevado de estados na operação. No entanto, as razões para a utilização desta estratégia são a clareza de interpretação das ações impostas pelos supervisores modulares e as facilidades para novas re-configurações, o que resulta em vantagens no processo de implementação. Algumas destas vantagens são as facilidades para a detecção de erros e mudanças nas regras de controle, através da exclusão ou inclusão dos novos supervisores que realizam cada especificação.

Capítulo 4

Modelagem do Sistema Físico

Neste capítulo é apresentado o sistema flexível de manufatura a ser modelado. Considera-se que este sistema tem a capacidade de processar diferentes famílias de produtos em diferentes máquinas, sendo que cada família de produto segue um roteiro de produção distinto para a sua manufatura. No desenvolvimento deste processo ocorre o compartilhamento de recursos como máquinas e sistema de transporte.

O processo de análise se desenvolve a partir de um caso específico cujo objetivo é a obtenção de modelos generalizados que possam atender a diferentes aplicações. Define-se uma seqüência de processamento para dois diferentes produtos e a partir daí realiza-se um estudo de análise para os diferentes modelos que representam o comportamento do sistema de transporte.

Analisa-se quatro modelos alternativos para o sistema de transporte, comparando as características e fatores de limitação para sua aplicação. Isto visa observar o que melhor atende aos objetivos de flexibilidade, quando forem necessárias alterações no processo. O modelo a ser adotado deve facilitar as atividades de projeto e re-projeto associadas às novas aplicações dos processos.

O presente sistema de manufatura a ser modelado de acordo com o número de máquinas e as características do sistema de transporte que pode atender a todas as máquinas, é classificado como de múltiplas estações com rotas variáveis.

4.1 Configuração Generalizada do Sistema Físico

O sistema tratado no presente trabalho é classificado, de acordo com Groover (2001), como um sistema automatizado de múltiplas estações com rotas variáveis conforme apresentado no capítulo 2. Uma configuração genérica do sistema de manufatura a ser tratado neste trabalho é apresentada na figura 4.1.

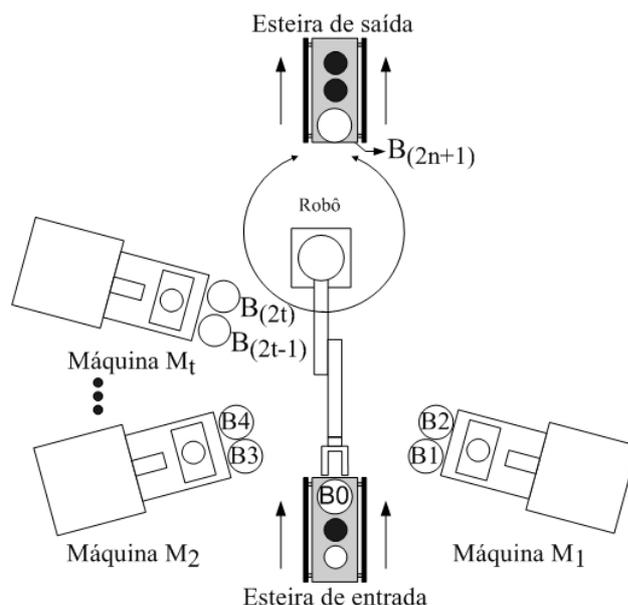


Figura 4.1: Sistema automatizado com rotas variáveis, centrado em robô
(Adaptação de Groover, 2001).

O sistema é composto de uma esteira transportadora de entrada de peças e outra de saída de peças. Cada máquina M_t representa uma estação de trabalho, sendo n o número de máquinas e $t=1\dots n$. O transporte de peças entre os *buffers* B_z , com $z = \{0, \dots, (2t+1)\}$, das estações é realizado por um robô sendo permitido o seu deslocamento entre todos os *buffers*. Segundo Groover (2001), o leiaute para este sistema e configuração é denominado de centrado em robô. A figura 2.9 representa esta situação.

Em cada estação de trabalho é possível realizar diferentes operações sobre as peças. Cada estação de trabalho possui um *buffer de entrada* $[B_{(2t-1)}]$ o qual recebe as peças a serem processadas e um *buffer de saída* $[B_{(2t)}]$ que recebem as peças processadas pelas máquinas M_t . Tais *buffers* possuem capacidade unitária. A esteira transportadora de entrada realiza o transporte de peças brutas para o processo e possui apenas um *buffer de saída* B_0 com capacidade unitária. O transporte de produtos acabados é realizado pela esteira transportadora de saída que possui capacidade infinita.

4.2 Caso específico a modelar

Para o caso exemplo é considerado um sistema operando com três máquinas ($n=3$). Os *buffers* são identificados como $B_1, B_2, B_3, B_4, B_5, B_6$ e B_7 . É proposto o processamento de dois diferentes tipos de peças identificadas como tipo A e B. A figura 4.2 representa o sistema a ser modelado

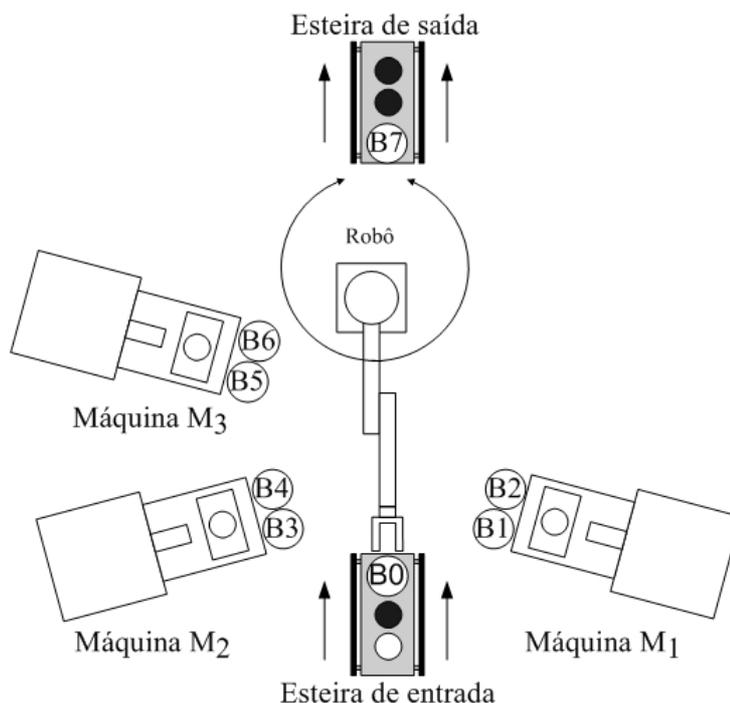


Figura 4.2: Sistema Flexível de Manufatura Centrado em Robô, caso específico para $n=3$.

Para o modelo das máquinas M_i , considera-se o número de operações que cada uma realizará durante o processo. O plano de processo define que a peça tipo A terá uma seqüência de operações que solicita as máquinas M_1, M_2 e M_3 , nesta ordem. A peça tipo B solicitará para a sua manufatura as máquinas M_1 e M_3 , nesta ordem. As máquinas M_1 e M_3 realizam dois tipos de operações, sendo adotado um autômato de três estados: 1 estado inicial (máquina em repouso) e dois estados correspondentes às duas operações. Para a máquina M_2 , que realiza uma única operação é adotado um autômato contendo apenas dois estados. A figura 4.3 apresenta os autômatos G_1, G_2 e G_3 correspondentes aos modelos das máquinas M_1, M_2 e M_3 , respectivamente.

Os estados iniciais dos autômatos mostrados na figura 4.3 representam as máquinas M_i em repouso. Os estados 1 e 2 representam as máquinas em operação processando peças tipo A e B, respectivamente. Os eventos α_{mtA} e α_{mtB} denotam o início de processamento das peças A e B, respectivamente. Os eventos β_{mtA} e β_{mtB} denotam o fim de processamento das peças A e B, respectivamente. Como regra geral os eventos denotados por α correspondem aos eventos controláveis, e os eventos denotados por β correspondem aos eventos não-controláveis.

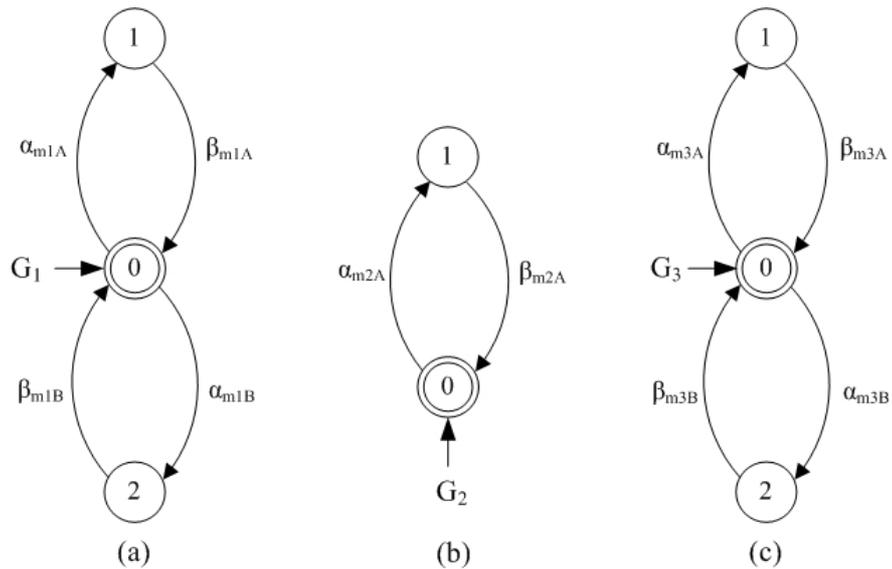


Figura 4.3: Modelo das Máquinas M_1 , M_2 e M_3 .

O modelo da esteira transportadora de entrada, G_4 , é apresentado na figura 4.4. O evento α_{es} representa o início de operação da esteira. Os eventos β_{pA} e β_{pB} , representam a chegada de uma peça tipo A ou B, respectivamente, ocasionando a parada da esteira.

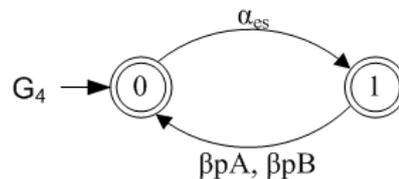


Figura 4.4: Modelo da esteira transportadora G_4 .

No sistema tratado, o robô pode movimentar-se livremente entre todas as posições de entrada e saída das máquinas. Para evitar movimentos desnecessários e tornar o robô mais eficiente, são permitidos dois tipos de deslocamentos:

- Estando o robô em um *buffer de saída*, somente será permitido o seu deslocamento em direção à um *buffer de entrada*.
- Estando o robô em um *buffer de entrada*, somente será permitido o seu deslocamento em direção à um *buffer de saída*.

Supondo o plano de processo anteriormente citado, têm-se os seguintes roteiros para as peças A e B:

- Peça A: $B_0 \rightarrow B_1 \rightarrow M_1 \rightarrow B_2 \rightarrow B_3 \rightarrow M_2 \rightarrow B_4 \rightarrow B_5 \rightarrow M_3 \rightarrow B_6 \rightarrow B_7$.
- Peça B: $B_0 \rightarrow B_1 \rightarrow M_1 \rightarrow B_2 \rightarrow B_5 \rightarrow M_3 \rightarrow B_6 \rightarrow B_7$.

De acordo com os dois tipos de deslocamentos considerados têm-se um total de 21 deslocamentos relevantes para esta aplicação que são: $\alpha_{01}, \alpha_{10}, \alpha_{30}, \alpha_{50}, \alpha_{70}, \alpha_{12}, \alpha_{32}, \alpha_{52}, \alpha_{72}, \alpha_{14}, \alpha_{34}, \alpha_{54}, \alpha_{74}, \alpha_{16}, \alpha_{36}, \alpha_{56}, \alpha_{76}, \alpha_{23}, \alpha_{25}, \alpha_{45}, \alpha_{67}$. Nas próximas seções realiza-se a análise de alguns modelos para o sistema de transporte. Estes modelos podem representar o comportamento de tal sistema, em atendimento às necessidades de movimentação entre *buffers*.

4.3 Análise do Primeiro Modelo para o Sistema de Transporte

O primeiro modelo é apresentado na figura 4.5. Este modelo representa o robô em dois possíveis estados: em repouso, em algum *buffer*, ou em movimento entre dois possíveis *buffers*. O conjunto de eventos $\Sigma_d = \{\forall \alpha_{xy} \mid x \neq y\}$, sendo $x, y \in \{0, \dots, 7\}$, para $x \neq y$, determinam a movimentação do robô. O conjunto de eventos $\Sigma_b = \{\forall \beta_z \mid z = 0, \dots, 7\}$, denotam a parada do robô em um *buffer*.

$\alpha_{01}, \alpha_{02}, \alpha_{03}, \alpha_{04}, \alpha_{05}, \alpha_{06}, \alpha_{07}, \alpha_{10}, \alpha_{12}, \alpha_{13}, \alpha_{14}, \alpha_{15}, \alpha_{16}, \alpha_{17},$
 $\alpha_{20}, \alpha_{21}, \alpha_{23}, \alpha_{24}, \alpha_{25}, \alpha_{26}, \alpha_{27}, \alpha_{30}, \alpha_{31}, \alpha_{32}, \alpha_{34}, \alpha_{35}, \alpha_{36}, \alpha_{37},$
 $\alpha_{40}, \alpha_{41}, \alpha_{42}, \alpha_{43}, \alpha_{45}, \alpha_{46}, \alpha_{47}, \alpha_{50}, \alpha_{51}, \alpha_{52}, \alpha_{53}, \alpha_{54}, \alpha_{56}, \alpha_{57},$
 $\alpha_{60}, \alpha_{61}, \alpha_{62}, \alpha_{63}, \alpha_{64}, \alpha_{65}, \alpha_{67}, \alpha_{70}, \alpha_{71}, \alpha_{72}, \alpha_{73}, \alpha_{74}, \alpha_{75}, \alpha_{76}$

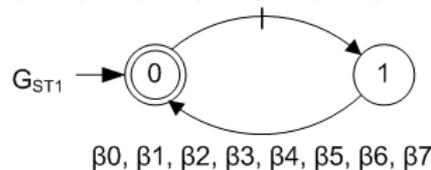


Figura 4.5: G_{ST1} , primeiro modelo candidato ao sistema de transporte.

Este modelo é genérico no sentido que prevê todo e qualquer deslocamento do robô. Entretanto, na produção de um determinado produto, nem todas as rotas consideradas no modelo são necessárias. Neste sentido, é preciso elaborar uma especificação que evite as rotas que não são previstas no plano de processo.

De acordo com os roteiros para as peças A e B, e os deslocamentos relevantes definidos na seção 4.2 têm-se a especificação da figura 4.6. Esta especificação ordena os 21 deslocamentos relevantes a partir de cada um dos *buffers* B_z , considerados para este processo.

Os estados 0, 1, 2, 3, 4, 5, 6 e 7, representam o robô em repouso no *buffer* correspondente e de mesmo número. Por exemplo, o estado 0 representa o robô em repouso no *buffer* B_0 . Os estados 8, 9, 10, 11, 12, 13, 14 e 15 representam o robô em movimento entre dois *buffers*. Tais *buffers* correspondem aos índices x e y do evento α_{xy} , que conduz ao referido estado. Por exemplo, o evento α_{67} conduz o autômato ao estado 8; este estado representa o robô em movimento entre os *buffers* B_6 e B_7 . O estado inacessível de número 16 corresponde à exclusão dos movimentos não-relevantes para esta aplicação.

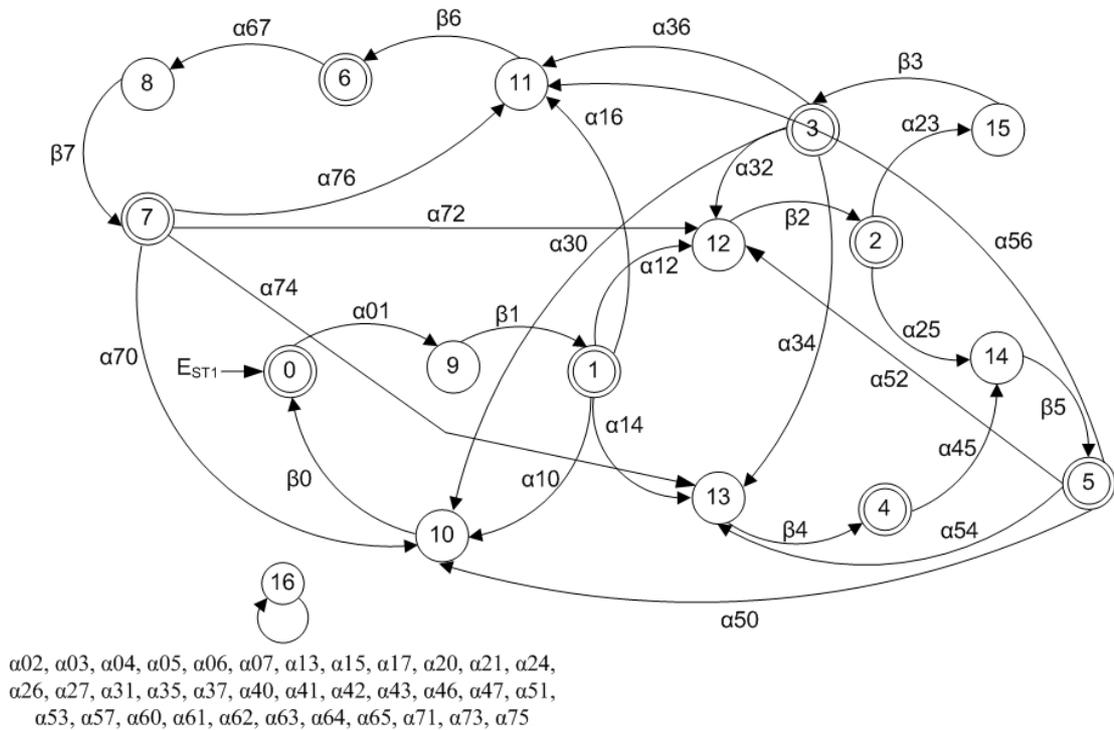


Figura 4.6: E_{ST1} , especificação de rotas.

Mesmo atendendo às necessidades do processo, no sentido de ordenar os deslocamentos a partir de cada um dos *buffers*, é preciso observar o comportamento que esta especificação impõe ao processo em relação ao que é previsto pelo modelo do sistema de transporte. Por exemplo, quando ocorre o evento α_{01} , é de se esperar que o robô, ao iniciar o seu deslocamento a partir do *buffer* B_0 em direção ao *buffer* B_1 , venha a parar nesta última posição através do evento β_1 . Observando o modelo do sistema de transporte (G_{ST1}) da figura 4.5, a parada em qualquer outra posição é perfeitamente possível, pois uma vez em movimento a ocorrência de qualquer evento β_z determina a sua parada na posição B_z correspondente ($z = 0, \dots, 7$).

A especificação para os roteiros desejados, mostrada na figura 4.6, prevê que após a ocorrência do referido evento α_{01} , somente estará definido o evento β_1 em seu estado de número 9. Isto implica na não definição dos outros eventos não-controláveis $\beta_0, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6$ e β_7 , que estão definidos no estado de número 1 do modelo do sistema de transporte. Isto representa um problema de controlabilidade da linguagem a ser gerada e pode ser explicado como segue.

Seja a cadeia de eventos constituída de um único evento $s = \alpha_{01}$ a ser observada nos geradores G_{Planta} e $G_{Planta||E_{ST1}}$ sendo $G_{Planta} = G_1 || G_2 || G_3 || G_4 || G_{ST1}$ e $G_{Planta||E_{ST1}} = G_{Planta} || G_{E_{ST1}}$. Sendo $\delta_{G_{Planta}}(0, s) = 1$ e $\delta_{G_{Planta||E_{ST1}}}(0, s) = 1$ as funções de transições de estado. As funções de eventos ativos dos respectivos geradores, G_{Planta} e $G_{Planta||E_{ST1}}$, são descritas como segue:

$$\Gamma_{G_{Planta}}(1) = \{\alpha_{es}, \alpha_{m1a}, \alpha_{m1b}, \alpha_{m2}, \alpha_{m3a}, \alpha_{m3b}, \beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7\} \quad e$$

$$\Gamma_{G_{Planta||E_{ST1}}}(1) = \{\alpha_{es}, \alpha_{m1a}, \alpha_{m1b}, \alpha_{m2}, \alpha_{m3a}, \alpha_{m3b}\}.$$

Observa-se que no estado 1 do gerador G_{Planta} o evento β_1 está sendo definido. Já no estado 1 da especificação, $G_{Planta||E_{ST1}}$, este mesmo evento β_1 não está definido, assim como todos outros eventos β_z . Desta forma, a não-definição de eventos não-controláveis que estão previstos no modelo da planta, e não estão definidos na linguagem resultante da composição síncrona ($G_{Planta} || G_{E_{ST1}}$), implica na não-controlabilidade desta linguagem. Sendo assim, na obtenção da máxima linguagem controlável ($\sup C(G_{Planta}, G_{Planta||E_{ST1}})$), o supervisor reconhecerá uma linguagem na qual a planta somente poderá operar sem o robô, pois o mesmo jamais poderá ser colocado em movimento.

A desativação dos eventos β_z faz com que o supervisor desabilite o evento controlável que deu origem a sua ocorrência, neste caso α_{01} , de forma a evitar uma ação de controle sobre um evento não-controlável β_z . Portanto, a linguagem resultante desta composição será não-controlável e isto inviabiliza a utilização do modelo G_{ST1} para a representação do comportamento livre do sistema de transporte.

4.4 Análise do Segundo Modelo para o Sistema de Transporte

A figura 4.7 apresenta o segundo modelo que representa o comportamento do sistema de transporte. Este modelo é mais refinado em relação ao primeiro modelo, uma vez que representa o sistema de transporte em repouso, em cada um dos respectivos *buffers* B_z . Observa-se tal fato através dos estados 0, 1, 2, 3, 4, 5, 6 e 7 do referido modelo G_{ST2} , no qual considera o *buffer* B_0 como a posição inicial do robô. O estado de número 8 representa o robô em movimento.

Este modelo, apesar de representar a localização do robô nos *buffers* B_z das estações de trabalho, ainda preserva a limitação apresentada pelo primeiro modelo. Uma vez em movimento, é possível que o robô pare em qualquer um dos *buffers* B_z . A razão para isto é que no estado de número 8 somente se define a ocorrência dos eventos não-controláveis β_z .

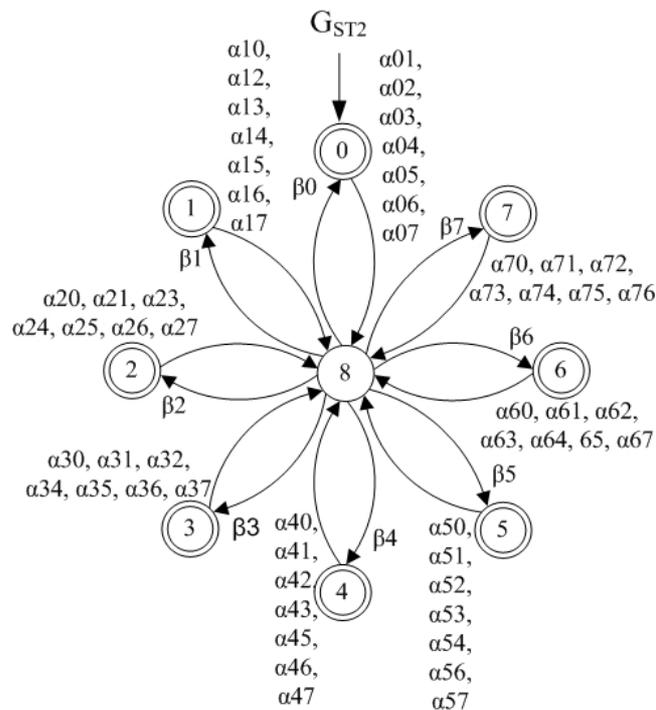


Figura 4.7: G_{ST2} , segundo modelo candidato ao sistema de transporte.

A utilização de uma especificação que venha ordenar os deslocamentos necessários para a manufatura de um determinado produto (E_{ST1}), conduzirá ao mesmo processo considerado para justificar tal limitação. Deste modo, as dificuldades relatadas para o primeiro modelo encontram-se presentes neste segundo modelo.

4.5 Análise do Terceiro Modelo para o Sistema de Transporte

A figura 4.8 representa o terceiro modelo para o sistema de transporte. Este modelo possui um maior número de estados, uma vez que representa o robô tanto em movimento, em direção a cada um dos *buffers* B_z , bem como em repouso nos respectivos *buffers*. De modo geral este modelo representa a ordem de ocorrência de todos os movimentos necessários para a realização do processo. A sua representação e a semântica dos estados é idêntica à realizada para a especificação E_{ST1} . Os deslocamentos não-relevantes são desconsiderados para este modelo.

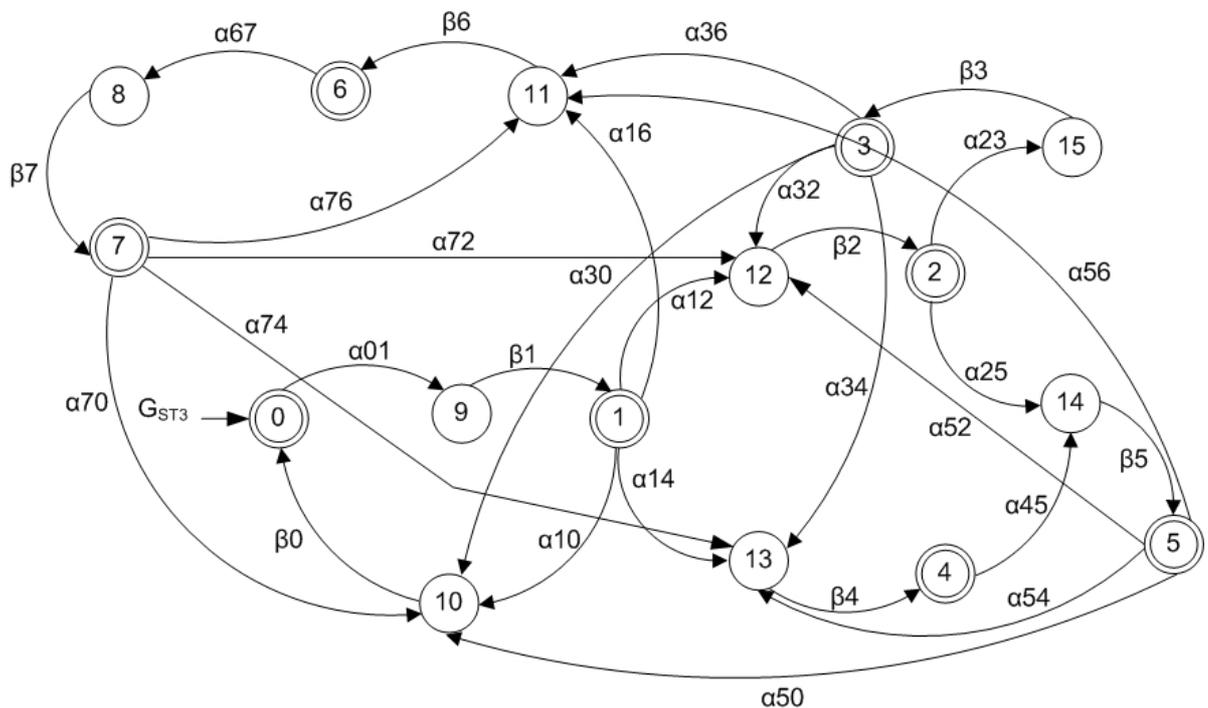


Figura 4.8: G_{ST3} , terceiro modelo para o sistema de transporte.

Apesar do elevado número de estados representar uma desvantagem, a sua utilização dispensa a elaboração de uma especificação para ordenamento dos movimentos. Uma vez o

robô em repouso em cada um dos *buffers*, somente são permitidos os movimentos necessários a partir de cada uma destas posições. Sendo assim, para a elaboração deste modelo deve-se considerar o plano de processo dos produtos e as suas respectivas rotas de produção.

A incorporação de um grande número de informações num único modelo, ocasiona uma outra desvantagem a este modelo quando se trata da sua alteração. A representação de forma integral do plano de processo, dificulta os propósitos de re-projeto quando ocorrer a manufatura de um novo produto. Isto decorre do processo de refinamento do modelo, que contém informações relativas à ordenação dos deslocamentos, representação do estado do transportador e a sua localização. Esta atividade de re-projeto exigirá alterações mais amplas na lógica de controle, inviabilizando as reduções de tempo, advindas de uma lógica mais estruturada. Deste modo, quando a complexidade de rotas aumenta, a utilização deste modelo torna difícil a sua utilização e implementação.

4.6 Análise do Quarto Modelo para o Sistema de Transporte

O quarto modelo a ser analisado, denominado G_{ST4} , apresentado na figura 4.9, segue uma representação semelhante ao segundo caso (fig. 4.7). No entanto, a semântica associada aos estados é alterada. Neste caso, considera-se para o estado inicial o robô em repouso, não especificando o *buffer* em que o mesmo se encontra. Os estados 0, 1, 2, 3, 4, 5, 6 e 7, representam o robô em movimento e em direção aos *buffers* B_0 , B_1 , B_2 , B_3 , B_4 , B_5 , B_6 e B_7 , respectivamente. Os eventos α_{xy} são agrupados de modo que representam o deslocamento em direção à um único *buffer* B_z . Por exemplo, os eventos que conduzem ao *buffer* B_0 são α_{10} , α_{20} , α_{30} , α_{40} , α_{50} , α_{60} e α_{70} , e ao *buffer* B_2 são α_{02} , α_{12} , α_{32} , α_{42} , α_{52} , α_{62} e α_{72} .

Para este modelo, nos estados que representam o robô em movimento, define-se apenas o evento não-controlável β_z que corresponde a sua chegada à um determinado *buffer* B_z . Esta característica é também observada no modelo da figura 4.8. Este modelo G_{ST4} apresenta um menor número de estados, uma vez que a informação referente ao ordenamento dos movimentos não foi considerada.

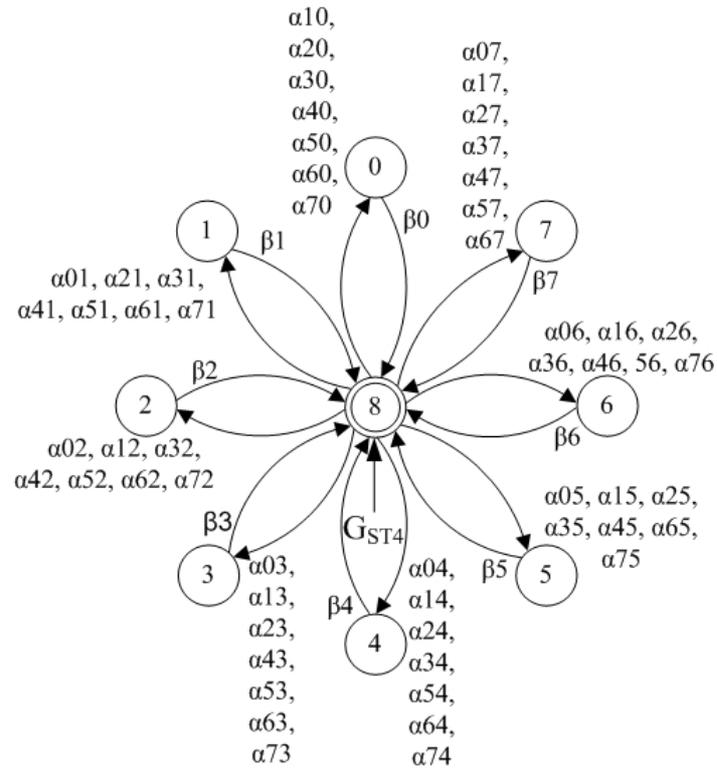


Figura 4.9: G_{ST4} , quarto modelo para o sistema de transporte.

Este modelo representa um avanço quando se trata da alteração do referido plano de processo, pois a inclusão ou exclusão dos *buffers* e dos deslocamentos se realiza de forma menos trabalhosa. Por exemplo, caso seja necessária a adição de uma nova máquina ao processo, o modelo do sistema de transporte não terá grandes alterações. A adição de dois novos *buffers* se realiza através da adição de dois novos estados ao modelo. O procedimento para a adição dos novos deslocamentos ocorre da mesma forma. Consideram-se os possíveis deslocamentos entre todos os *buffers* e realiza-se a sua adição ao modelo, através das novas transições de estados correspondentes.

O ordenamento destes deslocamentos deve ser detalhado através de uma especificação E_{ST2} , apresentada na figura 4.9. De acordo com o plano de processo considerado na seção 4.2, esta especificação define a partir de cada um dos *buffers*, os possíveis deslocamentos. Os estados 0, 1, 2, 3, 4, 5, 6 e 7 representam o robô em direção aos *buffers* B_0 , B_1 , B_2 , B_3 , B_4 , B_5 , B_6 e B_7 , respectivamente. O estado de número 8, representa a exclusão dos deslocamentos que não são relevantes para a aplicação em questão.

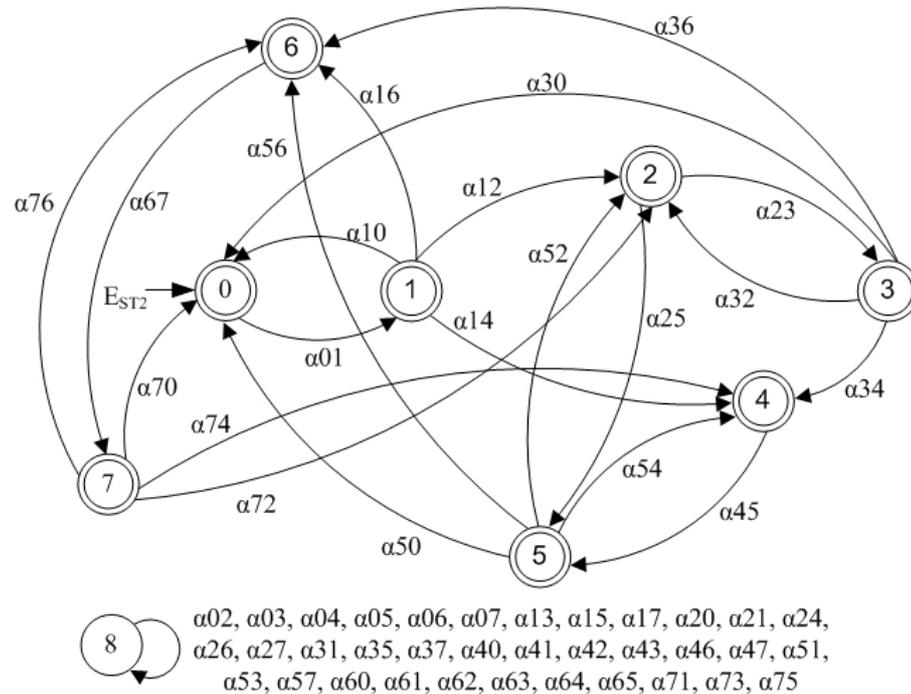


Figura 4.10: E_{ST2} , especificação de rotas.

A composição síncrona dos autômatos, $G_{ST4} \parallel E_{ST2}$, dará origem a um autômato que reconhece o mesmo par de linguagens (linguagem gerada e linguagem marcada) reconhecidas pelo autômato da figura 4.8. No entanto, uma vantagem na utilização deste modelo está na possibilidade da adição e remoção de novos deslocamentos e *buffers*, se realizar de forma mais simples. Outra vantagem está na tarefa de alteração da lógica de controle gerada no CLP, que será menos dispendiosa em razão da mesma ser apenas parcial.

4.7 Conclusão ao capítulo

Neste capítulo são apresentados os modelos que representam o comportamento dos subsistemas constituintes de um sistema de manufatura flexível hipotético. As características de um sistema multi-produtos e com rotas flexíveis, impõe que seja utilizada uma abordagem metodológica que privilegie modificações futuras. Sendo assim, para os modelos das máquinas M_i foi utilizado um modelo que considera o número de operações desenvolvidas, além do seu estado inicial. Ou seja, se a máquina desenvolve um número x de operações, o autômato que representa o seu comportamento terá $x+1$ estados. Deste modo, quando for preciso a realização

por alguma máquina de uma nova operação, basta que se adicione ao seu modelo o estado correspondente à esta nova operação. Processo semelhante se realiza para a exclusão de operações, através da remoção de estados.

O processo para obtenção do modelo do sistema de transporte exige uma maior observação, em razão das várias possibilidades de movimentação desempenhadas pelo transportador. A abordagem metodológica considerada para este processo, cujo sistema de transporte é centrado em robô, também pode ser adotada para um sistema com leiaute em linha, de rotas fixas ou variáveis. Estes dois últimos leiautes podem ser observados nas figuras 2.4 e 2.5 do capítulo 2.

O primeiro modelo para o sistema de transporte possui uma vantagem que é o número reduzido de estados. No entanto, este fato acarreta uma desvantagem que é a falta de informação quanto a localização do robô no processo. Uma limitação deste modelo está na impossibilidade de se determinar o local exato de parada do robô. Isto porque todos os eventos que sinalizam as paradas em cada um dos *buffers*, estão definidos simultaneamente em um único estado. Tal fato, compromete os objetivos de controlabilidade da linguagem gerada pelo modelo.

O segundo modelo incorpora a informação referente à localização do robô, ausente no primeiro modelo. Para tanto, utiliza-se de um maior número de estados, o que é uma desvantagem considerando o esforço computacional para a síntese e implementação da máxima linguagem controlável. A linguagem em questão será obtida, sem que se permita a entrada em operação do sistema de transporte conforme se relata na seção 4.3. Este fato permite que as máquinas M_i e esteiras transportadoras sejam colocadas em operação, porém o transporte das peças será realizado de forma manual. A razão para isto é a mesma limitação apresentada pelo primeiro modelo. Tal limitação inviabiliza a utilização de ambos os modelos.

Uma outra limitação presente para os dois casos é a inexistência de informações relativas à ordem de realização dos movimentos. No entanto, esta última limitação não determina a impossibilidade de utilização dos modelos e pode ser solucionada através de uma especificação de rotas.

O terceiro modelo, obtido a partir da análise de uma especificação de rotas (E_{ST1}), agrega um grande número de informações, tanto do sistema de transporte como do processo. Tais informações dizem respeito ao estado do robô, se em movimento ou em repouso, a sua localização e o ordenamento dos movimentos para a realização do processo de manufatura. Em razão disto, o número de estados para este modelo é maior.

Uma vantagem do terceiro modelo é que elimina a limitação de utilização presente nos dois primeiros casos, é a definição de apenas um evento não-controlável nos estados que representam o robô em movimento. Este evento representa a parada do robô no respectivo *buffer*, para o qual estava se deslocando. Neste sentido, tem-se a certeza do local de parada do mesmo, inviabilizando paradas indesejadas.

Outro importante aspecto que resulta da definição de apenas um evento não-controlável, diz respeito à controlabilidade da linguagem gerada por este modelo. Desta forma, o supervisor exercerá ação de controle apenas sobre determinados grupos de eventos controláveis, e não sobre todo o conjunto de eventos controláveis que determinam a movimentação do robô. Por exemplo, nos dois primeiros modelos, os quais se define em um único estado todos os eventos não-controláveis, o supervisor exercerá ação de controle sobre toda a cadeia de eventos controláveis que colocam o robô em movimento. Isto significa que nenhum evento controlável será permitido ocorrer.

Quando se considera um modelo em que se define apenas um evento não-controlável em um único estado, significa que o supervisor não será forçado a exercer ação de controle sobre o evento controlável que originou tal evento β_z . Sendo assim, o supervisor exercerá ação de controle somente sobre alguns eventos controláveis e não sobre a sua totalidade. Isto implica na obtenção de uma máxima linguagem controlável considerando o robô em movimento. A utilização deste modelo mostrou-se plenamente viável, no entanto, quando se observam os propósitos de implementação e re-projeto ele oferece uma desvantagem relativa ao quarto que será apresentada a seguir.

O quarto modelo proporciona vantagens citadas para o segundo e terceiro caso, que é a utilização de um estado para representar a localização do robô nos *buffers*. Também apresenta a mesma característica de controlabilidade da linguagem gerada, como relatado para o terceiro modelo e apresentado no parágrafo anterior. A utilização deste é viável, para tanto é preciso a utilização de uma especificação de controle para o ordenamento dos movimentos.

Uma vantagem deste modelo, em relação ao terceiro, é a separação entre informações referente ao sistema de transporte das informações referentes ao processo de produção. Ou seja, o modelo principal incorpora informações pertinentes apenas ao sistema de transporte. Já a especificação define a ordenação dos movimentos para a realização do processo de produção. Esta abordagem facilita as alterações futuras, uma vez que é realizada em autômatos distintos.

Isto proporciona, também, facilidades no processo de alteração da lógica de controle, pois tal atividade se realiza de forma pontual e não na lógica como um todo.

Capítulo 5

Modelagem das Especificações de Controle e Síntese dos Supervisores

Neste capítulo é apresentada uma sistematização do processo de modelagem das especificações de controle, responsáveis pela coordenação geral do sistema de manufatura descrito no capítulo 4. Estas especificações estão divididas em quatro conjuntos: o primeiro conjunto coordena o comportamento do sistema de acordo com a presença ou não de peças em cada *buffer* sem considerar o tipo de peça a ser processada; o segundo conjunto é responsável pelas ações de coordenação relacionadas ao tipo de peça a ser processada; o terceiro conjunto coordena as movimentações do sistema de transporte e o quarto conjunto coordena as ações dos subsistemas envolvidos de acordo com a presença de um tipo específico de peça ao observar simultaneamente o estado de dois *buffers*.

Os modelos generalizados referente à cada conjunto de especificação e os procedimentos para obtenção dos mesmos são apresentados. Em seguida é realizada a apresentação das especificações de controle obtidas para o caso específico do processo de modelagem, e são descritos os objetivos de controle a serem impostos ao processo através de cada uma delas. Neste contexto define-se a semântica associada aos estados e eventos das especificações, de acordo com o caso específico a ser tratado.

O processo de síntese dos supervisores modulares esta de acordo com a abordagem modular local (Queiroz e Cury, 2000). Na seqüência, desenvolve-se a redução dos supervisores modulares sintetizados. Este último processo, conclui os procedimentos da etapa de síntese dos supervisores. Por fim, são apresentadas as conclusões pertinentes ao processo de modelagem e síntese realizado.

5.1 Especificações de Controle Generalizadas

Uma especificação de controle é um requisito a ser imposto ao sistema e visa coordenar as ações a serem desenvolvidas entre dois ou mais subsistemas. Nesta seção é apresentado o conjunto de especificações $E_j, j \in J = \{0, \dots, m\}$, necessárias para a coordenação geral do sistema de manufatura. Para atender os requisitos operacionais a serem impostos ao sistema de manufatura tratado, propõe-se a construção de quatro conjuntos de especificações.

De um modo geral, o primeiro conjunto de especificações, que representa o estado de cada um dos *buffers*, pode ser generalizado considerando tanto os *buffers* de entrada bem como os *buffers* de saída. Em função dos subsistemas esteiras transportadoras, de entrada e de saída, não serem considerados estações de processamento de peças e por possuírem apenas um *buffer*, os mesmos não fazem parte do conjunto de subsistemas M_t , pois estes subsistemas possuem tanto *buffer* de entrada como de saída.

No entanto, para efeito de generalização de especificação, é necessário agregar a especificação que coordena as ações dos subsistemas em função do estado do *buffer* de saída da esteira de entrada à um conjunto específico que compreende os *buffers* de saída. Para tanto utiliza-se de uma variável t' , que representará tanto os subsistemas máquinas M_t bem como o subsistema esteira transportadora de entrada.

Considere os *buffers* de saída de um subsistema sendo identificados como $B_{(2i')}$ com $i'=t'=\{0, \dots, n\}$ sendo n o número de máquinas. Sendo $i'=0$, obtém-se então o comportamento do *buffer* B_0 (saída da esteira de entrada) para todo $t'=\{0, \dots, n\}$. Para $i'=1$, calcula-se o comportamento do *buffer* de saída da máquina M_t , identificado como B_2 , também para todo $t'=\{0, \dots, n\}$. A figura 5.1 ilustra a generalização de uma especificação para um *buffer* de saída.

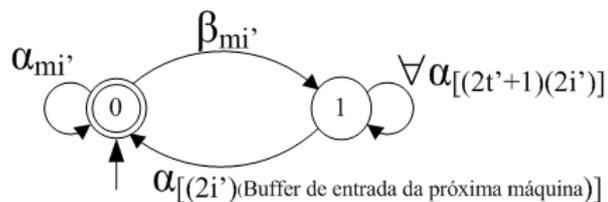


Figura 5.1: Especificação generalizada do primeiro correspondente à um *buffer* de saída.

Este primeiro conjunto de especificação generalizada tem por objetivo principal a coordenação de dois tipos de ações. A primeira corresponde ao fim de operação de um subsistema qualquer, com a solicitação de deslocamento do sistema de transporte até o *buffer* $B_{(2i')}$. Em seguida, realizar o transporte da peça em processo, do referido *buffer de saída* $B_{(2i')}$ da máquina M_i , até o *buffer de entrada* da próxima máquina. O segundo tipo de ação, coordena o início de operação de uma máquina M_i com a chegada de uma peça qualquer ao seu respectivo *buffer de entrada* $B_{(2i-1)}$, em que $t=\{1,\dots,n\}$ para todo $t'=\{0,\dots,n\}$.

O estado de um *buffer* de entrada $B_{(2i-1)}$ pode ser generalizado de acordo com o modelo de especificação da figura 5.2. Sendo assim, o estado do *buffer* B_1 pode ser obtido fazendo $t=1$ para todo $t'=\{0,\dots,n\}$. Do mesmo modo o estado do *buffer* B_3 pode ser obtido fazendo $t=2$ para todo $t'=\{0,\dots,n\}$.

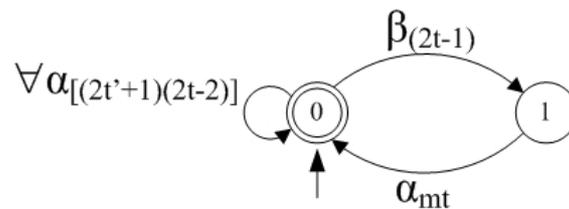


Figura 5.2: Especificação generalizada do primeiro conjunto para um *buffer* de entrada.

No estado inicial deste tipo de especificação, a definição de que eventos habilitar para a movimentação do sistema de transporte dependerá dos objetivos de controle a serem impostos para cada processo. Por exemplo, caso o robô se encontre em um *buffer de entrada* e este mesmo *buffer* esteja cheio, é recomendável que o robô não seja enviado a um *buffer de saída* para buscar uma peça que será transportada para o *buffer de entrada* em questão. Isto se explica uma vez que, em um *buffer de entrada* o robô possui uma liberdade de deslocamento maior, pois é facultado ao mesmo se deslocar para a saída de qualquer subsistema. Quando o robô se encontra em um *buffer de saída* somente será permitido o seu deslocamento a uma única direção. Sendo assim, permite-se que o sistema opere com maior liberdade, pois caso a máquina, cujo *buffer de entrada* encontra-se com peça, venha a quebrar, o robô terá a liberdade de movimentar-se para qualquer outra posição não bloqueando o sistema.

Caso uma determinada máquina quebre com o seu *buffer de entrada* cheio e o robô se encontrar em *buffer de saída* cuja peça deverá ser transportada para a referida máquina, isto

ocasionará um bloqueio no sistema até que o referido *buffer de entrada* seja esvaziado. Para evitar tal bloqueio é recomendável que o robô se desloque para a saída de um subsistema, cuja peça deverá ser transportada para a máquina em questão, somente se esta mesma máquina já estiver em operação, ou seja, com o seu *buffer de entrada* vazio.

Este primeiro conjunto de especificações tem por objetivo geral a coordenação das atividades das máquinas em função da presença ou ausência de peças nos *buffers* B_z , com $z = \{0, \dots, 2t+1\}$, independente do tipo de peça presente. No caso em questão, o estado do *buffer* $B_{(2t+1)}$, que corresponde à entrada da esteira transportadora de saída, não está sendo modelado, uma vez que o mesmo possui capacidade infinita de peças.

O segundo conjunto de especificações refere-se à um tipo específico de peça. Por exemplo, qual operação deverá ser desenvolvida por uma máquina M_t ou para onde esta peça deverá ser transportada se a mesma for do tipo Y. Estas especificações dizem respeito mais claramente à coordenação do sistema de transporte com o final ou início de processamento de uma peça específica, em uma determinada máquina M_t . A figura 5.3 ilustra duas generalizações para esta classe de especificação de coordenação.

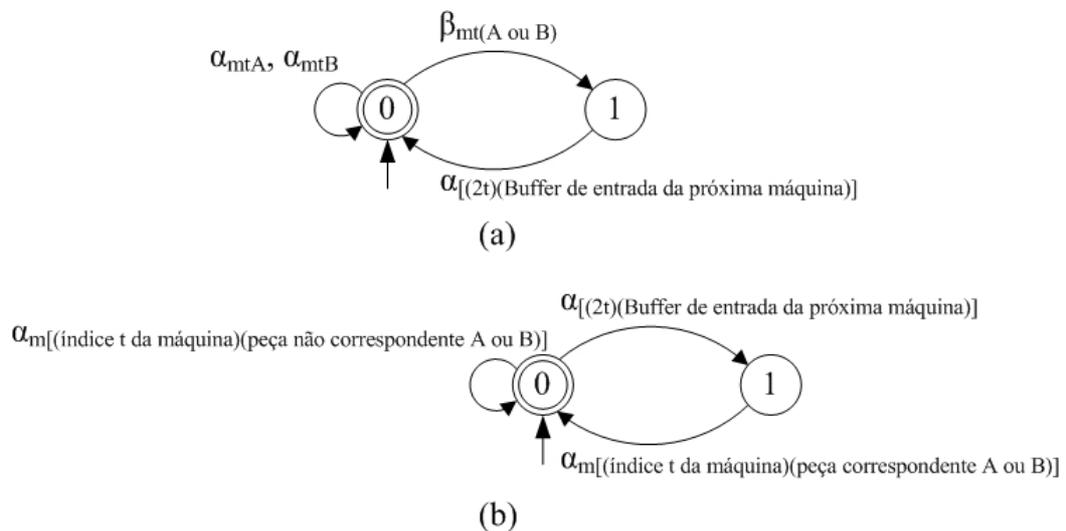


Figura 5.3: Especificação generalizada do segundo conjunto.

A figura 5.3a representa uma especificação generalizada que coordena o final de uma operação de processamento $[\beta_{mt}(A, B)]$ por uma máquina qualquer e o início do transporte da peça para uma outra máquina M_t . A habilitação dos eventos α_{mtA} e α_{mtB} no estado inicial evita

uma nova operação da máquina antes que se faça o transporte da peça. A figura 5.3b representa uma especificação generalizada que coordena o transporte de uma peça tipo A ou B, com a habilitação do evento correspondente (αmtA ou αmtB) para início de processamento. A habilitação do evento αmt no estado inicial evita que seja gerada uma operação indevida para uma determinada peça. Por exemplo, se a peça transportada for do tipo A, deverá ser habilitado o evento αmtA no estado de número 1. No estado inicial deverá ser habilitado o evento αmtB . Isto garante que se no *buffer de entrada* de uma máquina M_i tiver uma peça tipo A, jamais se desenvolverá uma operação para peça tipo B.

O terceiro conjunto de especificações é formado por uma única especificação, que impõe a correta seqüência de movimentações para o processamento das peças tipo A e B. Esta especificação restringe o comportamento do sistema de transporte aos deslocamentos necessários para o cumprimento das rotas de produção pré-estabelecidas. Esta especificação se generaliza a partir das condições para movimentação impostas ao sistema transporte e do plano processo em questão, sendo que ambas as situações foram descritas nas seções 4.2 e 4.3. Sendo assim, esta especificação não pode ser generalizada de acordo com um modelo único, pois cada peça ou produto poderá ter o seu próprio plano de processo. No entanto deve-se ter como ponto de partida para a sua elaboração as condições citadas nas seções 4.2 e 4.3. No caso geral, esta especificação terá $(2n+3)$ estados, sendo n o número total de máquinas no processo.

O quarto conjunto de especificações é formado por uma única especificação, apresentada na figura 5.4, que tem por objetivo coordenar o comportamento conjunto de dois subsistemas (duas máquinas, ou uma máquina e uma esteira) com o sistema de transporte. Para tanto, deve-se observar simultaneamente o estado de dois *buffers*. Um destes *buffers* é o *buffer de saída* do subsistema origem ou fornecedor de peças, o outro é o *buffer de entrada* do subsistema destino ou consumidor de peça.

Essa especificação, no caso geral, deve sempre ser construída quando se considera uma situação em que dois diferentes tipos de peças, saem de um único subsistema e se direcionam para processamento também em um único subsistema. Ou seja, existe uma relação unívoca entre o subsistema fornecedor (origem) e o subsistema consumidor (destino).

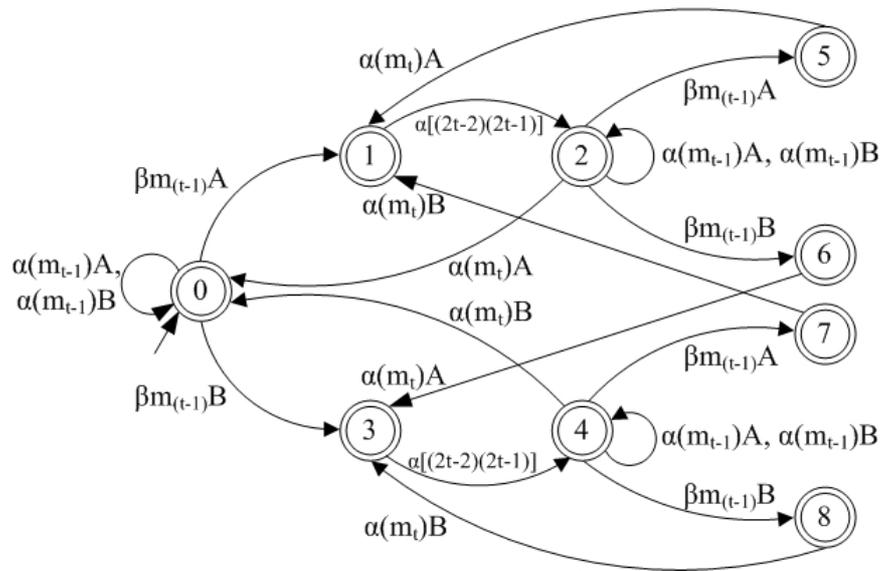


Figura 5.4: Especificação generalizada do quarto conjunto.

Esta especificação pode ser generalizada de acordo com o número de máquinas M_t , para $t=\{1, \dots, n\}$, quando se considera um sistema de linha de transferência em que as peças saem da máquina M_1 (Subsistema fornecedor) e seguem para a máquina M_2 (Subsistema consumidor). Na seqüência de processamento, estas peças saem da máquina M_2 (Subsistema fornecedor) e vão para a máquina M_3 (Subsistema consumidor). Neste caso, em que os índices t dos subsistemas são seqüenciais e em ordem crescente o modelo se aplica integralmente considerando-se apenas os índices associados às máquinas M_t e aos eventos a serem habilitados.

Em sistemas com rotas variáveis os índices das máquinas M_t já não mais serão seqüenciais e em ordem crescente. Por exemplo, um sistema em que dois diferentes tipos de peças saem da máquina M_1 (Subsistema fornecedor) e sempre terão como destino à máquina M_3 (Subsistema consumidor). Neste caso, a adoção dos índices dos subsistemas do modelo geral não mais se aplica, no entanto a semântica dos estados e as funções de eventos ativos nos referidos estados continuam as mesmas. Portanto, o modelo da especificação do quarto conjunto pode ser generalizado se forem consideradas apenas as funções de eventos ativos em cada estado, porém os índices relacionados aos eventos dos subsistemas devem ser objeto de análise.

A tabela 5.1 apresenta uma generalização dos eventos a serem habilitados nos respectivos estados da especificação do quarto conjunto. A primeira coluna representa os estados. A segunda coluna indica quais eventos deverão ser habilitados no estado correspondente. Por exemplo,

considere o caso de um sistema com rotas variáveis em que as peças sempre saem da máquina M_1 (Subsistema fornecedor) e serão transportadas para a máquina M_3 (Subsistema consumidor).

No estado inicial da especificação devem ser habilitados quatro eventos sendo $\alpha m1A$, $\alpha m1B$, $\beta m1A$ e $\beta m1B$. Os eventos $\alpha m1A$ e $\alpha m1B$ serão do tipo auto-laço, ou seja, não ocasionam a transição do estado inicial para nenhum outro estado. Tais eventos, como já definido, representam o início de operação do subsistema máquina M_1 . Os eventos $\beta m1A$ e $\beta m1B$ são do tipo transição de estado, ou seja, ocasionam a transição do estado inicial para os estados 1 e 3 respectivamente. Tais eventos representam o fim de operação da máquina M_1 .

Nos estados de números 1 e 3 é definido o evento que representa a movimentação do robô (sistema de transporte), para o transporte de uma peça do *buffer de saída* (B_2) da máquina M_1 para o *buffer de entrada* (B_3) da máquina M_3 . O evento em questão é o $\alpha 25$ e é do tipo transição de estado. Tal evento ocasiona a transição do estado 1 para o estado 2, e a transição do estado 3 para o estado 4 da referida especificação.

Tabela 5.1: Generalização da especificação do quarto conjunto.				
Estado	Evento a ser habilitado	Subsistema ao qual se refere	Semântica do Evento	Tipo de Evento (Transição de estado ou Auto-laço)
0	$\alpha(m_{fornecedor})A$	Fornecedor	Início de operação	Auto-laço
	$\alpha(m_{fornecedor})B$	Fornecedor	Início de operação	Auto-laço
	$\beta(m_{fornecedor})A$	Fornecedor	Fim de operação	Transição ($0 \rightarrow 1$)
	$\beta(m_{fornecedor})B$	Fornecedor	Fim de operação	Transição ($0 \rightarrow 3$)
1	$\alpha(buffer_fornecedor,buffer_consumidor)$	Sistema de Transporte	Transporte de peça do buffer de saída do subsistema fornecedor para o buffer de entrada do subsistema consumidor.	Transição ($1 \rightarrow 2$)

2	$\alpha(m_{fornecedor})A$	Fornecedor	Início de operação	Auto-laço
	$\alpha(m_{fornecedor})B$	Fornecedor	Início de operação	Auto-laço
	$\beta(m_{fornecedor})A$	Fornecedor	Fim de operação	Transição (2 → 5)
	$\beta(m_{fornecedor})B$	Fornecedor	Fim de operação	Transição (2 → 6)
	$\alpha(m_{consumidor})A$	Consumidor	Início de operação Peça A	Transição (2 → 0)
3	$\alpha(buffer_fornecedor,buffer_consumidor)$	Sistema de Transporte	Transporte de peça do buffer de saída do subsistema fornecedor para o buffer de entrada do subsistema consumidor.	Transição (3 → 4)
4	$\alpha(m_{fornecedor})A$	Fornecedor	Início de operação	Auto-laço
	$\alpha(m_{fornecedor})B$	Fornecedor	Início de operação	Auto-laço
	$\beta(m_{fornecedor})A$	Fornecedor	Fim de operação	Transição (4 → 7)
	$\beta(m_{fornecedor})B$	Fornecedor	Fim de operação	Transição (4 → 8)
	$\alpha(m_{consumidor})B$	Consumidor	Início de operação Peça B	Transição (4 → 0)
5	$\alpha(m_{consumidor})A$	Consumidor	Início de operação Peça A	Transição (5 → 1)
6	$\alpha(m_{consumidor})A$	Consumidor	Início de operação Peça A	Transição (6 → 3)
7	$\alpha(m_{consumidor})B$	Consumidor	Início de operação Peça B	Transição (7 → 1)
8	$\alpha(m_{consumidor})B$	Consumidor	Início de operação Peça B	Transição (8 → 3)

No apêndice deste trabalho, seção A.5, é apresentada uma especificação para um caso em que considera o processamento de 3 peças. A partir deste exemplo pode-se generalizar o número de estados que constitui este tipo de especificação. Para o caso em que o número de peças x é igual a 1 ($x=1$), tem-se uma especificação de três estados. Para os casos em que o número de peças é $x \geq 2$, a especificação terá $(x+1)^2$ estados.

5.2 Especificações de Controle do Caso Específico

Para o caso tratado na seção 4.2, os quatro conjuntos de especificações de controle podem ser obtidos de acordo com o exposto a seguir. O primeiro conjunto é formado pelas especificações de E_0 a E_6 e representam o estado dos *buffers* B_0 à B_6 respectivamente, e são apresentadas na figura 5.5. De forma geral, o estado inicial destas especificações representa os *buffers* sem peça, e o estado de número 1 representa os *buffers* com peça, não sendo feita distinção quanto ao tipo de peça presente nos *buffers*. Tais especificações estabelecem os seguintes requisitos:

- Evitar a ocorrência de *overflow* e *underflow* dos *buffers* B_z ;
- Reconhecer como tarefa completa o estado vazio de um determinado *buffer*;
- Coordenar o início e final de operação dos subsistemas, máquinas M_i e esteira, com o sistema de transporte.

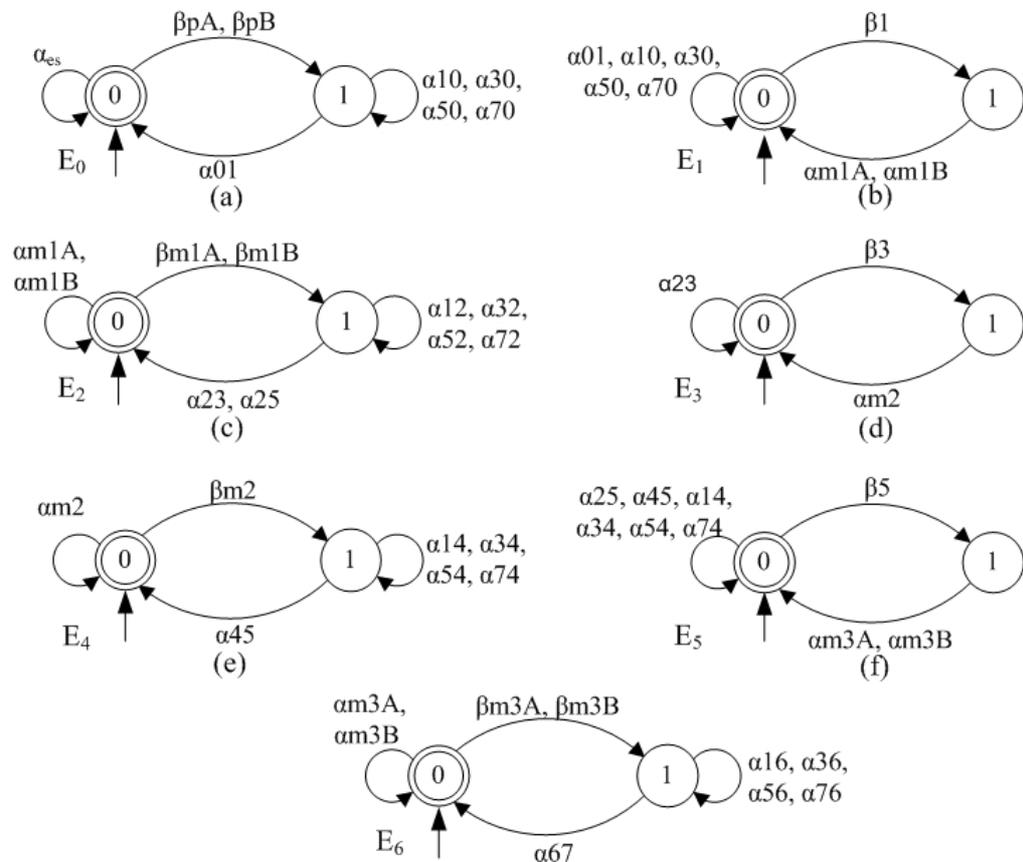


Figura 5.5: Especificações do primeiro conjunto: a) E_0 , b) E_1 , c) E_2 , d) E_3 , e) E_4 , f) E_5 , g) E_6 .

A especificação E_0 referente ao estado do *buffer* B_0 e mostrada na figura 5.5a, estabelece os seguintes requisitos:

- Evitar a operação da esteira transportadora entrada, estando o *buffer* B_0 com peça;
- Evitar deslocamentos do robô em direção ao *buffer* B_0 , se o mesmo estiver sem peça;
- Evitar deslocamento do robô a partir do *buffer* B_0 para o *buffer* B_1 , se não houver peça em B_0 a ser transportada.

A especificação E_1 referente ao estado do *buffer* B_1 e mostrada na figura 5.5b, estabelece os seguintes requisitos:

- Evitar que o robô busque peça no *buffer* B_0 , se houver peça no *buffer* B_1 ;
- Evitar que o robô se desloque do *buffer* B_0 para o *buffer* B_1 , se houver peça no *buffer* B_1 ;
- Evitar o início de operação da máquina M_1 , se não houver peça em B_1 .

A especificação E_2 referente ao estado do *buffer* B_2 e mostrada na figura 5.5c, estabelece os seguintes requisitos:

- Evitar que o robô se desloque em direção ao *buffer* B_2 , se o mesmo estiver vazio;
- Evitar o início de operação da máquina M_1 , se houver peça no *buffer* B_2 ;
- Evitar que o robô se desloque, a partir do *buffer* B_2 , em direção aos *buffers* B_3 ou B_5 , sem peças A ou B, respectivamente.

A especificação E_3 referente ao estado do *buffer* B_3 e mostrada na figura 5.5d, estabelece os seguintes requisitos:

- Evitar deslocamentos do robô a partir do *buffer* B_2 , em direção ao *buffer* B_3 , se houver peça no *buffer* B_3 ;
- Evitar início de operação da máquina M_2 , se não houver peça em B_3 .

A especificação E_4 referente ao estado do *buffer* B_4 e mostrada na figura 5.5e, estabelece os seguintes requisitos:

- Evitar o início de operação da máquina M_2 , se houver peça no *buffer* B_4 ;
- Evitar deslocamentos do robô em direção ao *buffer* B_4 , se o mesmo estiver vazio;
- Evitar deslocamentos do robô a partir do *buffer* B_4 , em direção ao *buffer* B_5 se não houver peça em B_4 .

A especificação E_5 referente ao estado do *buffer* B_5 e mostrada na figura 5.5f, estabelece os seguintes requisitos:

- Evitar deslocamentos do robô em direção ao *buffer* B_5 , se houver peça no mesmo;
- Evitar que o robô busque peça no *buffer* B_4 , se houver peça no *buffer* B_5 ;
- Evitar o início de operação da máquina M_3 , se o *buffer* B_5 estiver vazio.

A especificação E_6 referente ao estado do *buffer* B_6 e mostrada na figura 5.5g, estabelece os seguintes requisitos:

- Evitar o início de operação da máquina M_3 , se houver peça no *buffer* B_6 ;
- Evitar deslocamentos do robô em direção ao *buffer* B_6 , se o mesmo estiver vazio;
- Evitar deslocamentos do robô a partir do *buffer* B_6 , em direção ao *buffer* B_7 se não houver peça em B_6 .

Uma observação deve ser feita quanto às especificações que modelam o estado dos *buffers* de entrada das máquinas M_i . Não há um número mínimo ou máximo de eventos a serem habilitados quando o *buffer* estiver vazio. A habilitação de um ou outro evento dependerá do grau de liberdade que se quer dar ao sistema de transporte (robô), pois uma vez que um *buffer* de entrada estiver com peça isto poderá restringir a movimentação do robô.

Por exemplo, na especificação E_3 , que coordena o *buffer* de entrada da máquina M_2 , defini-se apenas o evento α_{23} em seu estado inicial para que o robô tenha a liberdade de buscar peças na saída da máquina M_1 caso estas sejam do tipo B. Desta forma o robô pode continuar a alimentar a máquina M_3 com peças tipo B. Caso tivessem sido habilitados os eventos $\alpha_{12}, \alpha_{32}, \alpha_{52}$ e α_{72} , o robô ficaria preso no *buffer* B_3 até que se concluísse a operação da máquina M_2 , e somente assim sairia em direção ao *buffer* B_4 . Deste modo, o conjunto de eventos a serem habilitados no estado inicial das especificações, que modelam o estado dos *buffers* de entrada das máquinas M_i , depende dos objetivos de controle do próprio projetista, o qual deverá buscar a maior flexibilidade possível.

O segundo conjunto de especificações para esta aplicação, é formado pelas especificações E_7, E_8 e E_9 conforme representadas na figura 5.6.

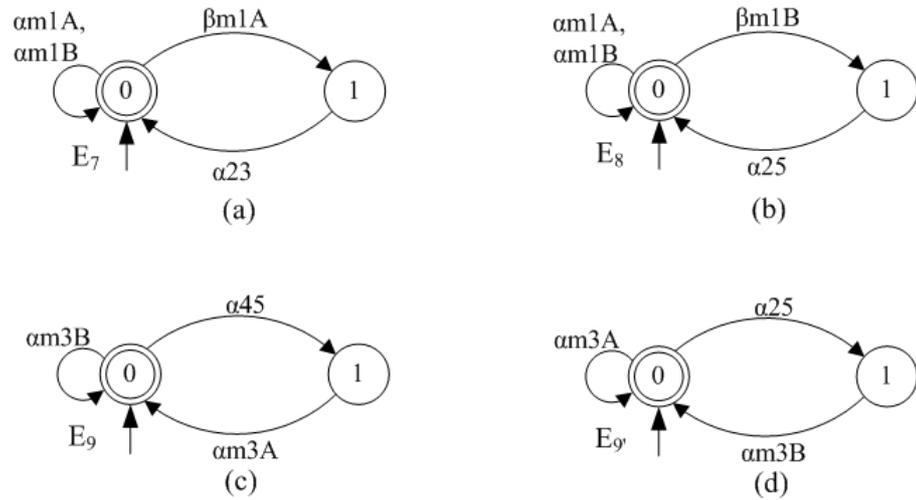


Figura 5.6: Especificações do segundo conjunto E_7 , E_8 , E_9 e E'_9 .

As especificações E_7 e E_8 coordenam o final de processamento de peça tipo A e B, respectivamente, pela máquina M_1 , com o início de operação do robô. Sendo assim, a especificação E_7 da figura 5.6a, estabelece os seguintes requisitos:

- Evitar o início de operação da máquina M_1 , se houver peça no *buffer* B_2 ;
- Evitar deslocamento do robô a partir do *buffer* B_2 , em direção ao *buffer* B_3 , se não existir peça tipo A no *buffer* B_2 .

A especificação E_8 , apresentada na figura 5.6b, estabelece os seguintes requisitos:

- Evitar o início de operação da máquina M_1 , se houver peça no *buffer* B_2 ;
- Evitar deslocamento do robô a partir do *buffer* B_2 , em direção ao *buffer* B_5 , se não existir peça tipo B no *buffer* B_2 .

A especificação E_9 coordena o transporte de uma peça tipo A, do *buffer* de saída da máquina M_2 para o *buffer* de entrada da máquina M_3 , e o início de operação da mesma máquina M_3 para processamento de uma peça tipo A. A especificação E'_9 coordena o transporte de uma peça tipo B, do *buffer* de saída da máquina M_1 para o *buffer* de entrada da máquina M_3 , e o início de operação da mesma máquina M_3 para processamento de uma peça tipo B. Estas duas especificações fazem com que a máquina M_3 inicie a sua operação de acordo com o tipo de peça presente em seu *buffer* de entrada. No entanto, elas são mutuamente excludentes uma vez que a especificação E_5 define em parte os objetivos para início de operação da máquina M_3 . Sendo assim, a especificação E'_9 foi excluída para os cálculos de obtenção dos supervisores, sendo considerada apenas a especificação E_9 .

A especificação E_9 , mostrada na figura 5.6c, estabelece os seguintes requisitos:

- Evitar o início de operação de processamento para peça tipo B, pela máquina M_3 , se houver peça tipo A no *buffer* B_5 ;

A ação conjunta das três especificações E_5 , E_8 e E_9 , estabelecem o seguinte requisito:

- Evitar o início de operação de processamento para peça tipo A, pela máquina M_3 , se houver peça tipo B no *buffer* B_5 ;

O terceiro conjunto de especificações é composto de uma única especificação. Esta especificação representa os possíveis deslocamentos do robô a partir de cada um dos *buffers* B_z . Os estados 0, 1, 2, 3, 4, 5, 6 e 7, representam o robô movimentando-se em direção aos respectivos *buffers* B_0 , B_1 , B_2 , B_3 , B_4 , B_5 , B_6 e B_7 . Esta especificação do terceiro conjunto é denominada E_{10} e é ilustrada na figura 5.7. No estado não acessível de número 8, são colocados em auto-laço os eventos correspondentes aos deslocamentos não desejados.

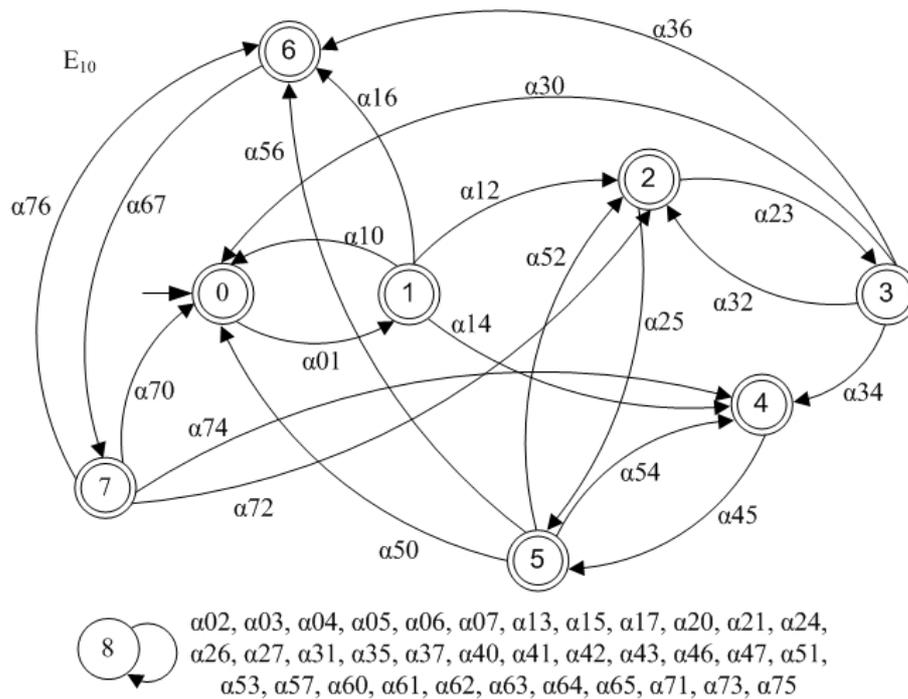


Figura 5.7: Especificação do terceiro conjunto para o sistema de transporte, E_{10} .

O quarto conjunto de especificações é formado por uma única especificação, que tem por objetivo coordenar o comportamento conjunto dos subsistemas esteira transportadora de entrada,

máquina M_I e robô. De acordo com os roteiros de produção do exemplo tratado, as duas peças, A e B, devem ser processadas inicialmente pela mesma máquina M_I . Desta forma, é possível operar o sistema com uma peça sendo processada na máquina M_I , uma peça no *buffer* B_0 , e outra no *buffer* B_I . A especificação em questão, denominada E_{11} , é apresentada na figura 5.8.

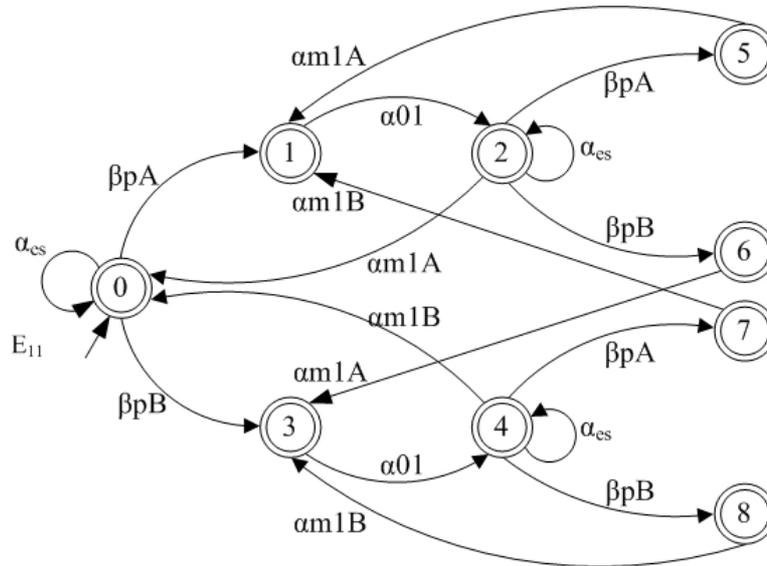


Figura 5.8: Especificação do quarto conjunto, para a coordenação conjunta da esteira, máquina M_I e sistema de transporte, E_{11} .

Para o caso tratado, ao se considerar os procedimentos da especificação generalizada e fazendo $t=1$, obtém-se uma coordenação de ações que considera simultaneamente o estado do *buffer de saída* (B_0) da esteira transportadora de entrada e o estado do *buffer de entrada* (B_I) da máquina M_I . Neste caso, como o evento para início de operação da esteira transportadora de entrada independe do tipo de peça, tem-se apenas um evento α_{m_0} que é então associado ao evento α_{es} .

Os eventos β_{m_0A} e β_{m_0B} , que determinam a parada de operação da esteira transportadora de entrada, são associados aos eventos β_{pA} e β_{pB} , que significam a chegada de peças tipo A e B, respectivamente. O evento $\alpha[(2t-2)(2t-1)]$, para $t=1$ resulta em α_{01} , o representa o transporte de peça do *buffer de saída* (B_0) da esteira transportadora de entrada para o *buffer de entrada* (B_I) da máquina M_I . Por fim, obtém-se α_{m_1A} e α_{m_1B} que representam o início de operação da máquina M_I para processamento de peça tipo A e B, respectivamente.

A semântica dos estados associada à esta especificação pode ser observada na tabela 5.2.

Tabela 5.2: Semântica dos estados para a Especificação E_{11}		
Estado	Semântica dos Estados (Tipo de peça presente nos <i>Buffer B_z</i>)	
	<i>Buffer B₀</i>	<i>Buffer B₁</i>
0	Não há peça A ou B	Não há peça A ou B
1	Há peça tipo A	Não há peça A ou B
2	Não há peça A ou B	Receberá peça Tipo A
3	Há peça tipo B	Não há peça A ou B
4	Não há peça A ou B	Receberá peça Tipo B
5	Há peça tipo A	Há peça tipo A
6	Há peça tipo B	Há peça tipo A
7	Há peça tipo A	Há peça tipo B
8	Há peça tipo B	Há peça tipo B

5.3 Síntese dos Supervisores

O procedimento de síntese dos supervisores segue a abordagem Modular Local. Para o caso considerado tem-se um conjunto de 12 supervisores modulares. As plantas locais, afetadas

por cada um destes supervisores, são definidas na tabela 1. Como ferramenta computacional para a obtenção dos supervisores foi utilizado o software TCT (Wonham, 1999).

Este conjunto de estruturas é apresentado na tabela 5.3, sendo que a coluna c1 apresenta a composição das plantas locais $G_{loc,j}$. A coluna c2 apresenta as células de controle. Denomina-se célula de controle (Vieira *et al.*, 2007) ao conjunto formado por um supervisor local e os subsistemas que correspondem a planta local. Na coluna c3 são apresentados os números de estados e transições dos supervisores modulares locais (SM_{Ej}). A coluna c4 informa o número de estados e transições dos supervisores modulares reduzidos. Um supervisor modular reduzido será identificado como (S_j), representado por um par (S_j, Φ_j), conforme apresentado na tabela 5.4. O conjunto Φ representa as ações de controle, ou função de desabilitação de eventos controláveis, de cada um dos supervisores modulares reduzidos, nos seus respectivos estados. Estes supervisores modulares reduzidos terão um número menor ou igual de estados, porém possuem a mesma ação de controle sobre o comportamento da planta livre local, que os correspondentes supervisores modulares locais.

Tabela 5.3 – Resultados da abordagem modular local			
c1	c2	c3 (SM_{Ej})	c4 (S_j)
$G_{loc0}=G_0 G_4$	$CC_0=\{SM_{E0}, G_0, G_4\}$	27x209	2x126
$G_{loc1}=G_0 G_1$	$CC_1=\{SM_{E1}, G_0, G_1\}$	51x398	2x122
	$CC_2=\{SM_{E2}, G_0, G_1\}$	36x274	2x126
	$CC_7=\{SM_{E7}, G_0, G_1\}$	36x289	2x131
	$CC_8=\{SM_{E8}, G_0, G_1\}$	36x289	2x131
$G_{loc2}=G_0 G_2$	$CC_3=\{SM_{E3}, G_0, G_2\}$	34x265	2x123
	$CC_4=\{SM_{E4}, G_0, G_2\}$	27x200	2x125
$G_{loc3}=G_0 G_3$	$CC_5=\{SM_{E5}, G_0, G_3\}$	36x277	2x122
	$CC_6=\{SM_{E6}, G_0, G_3\}$	36x289	2x127
	$CC_9=\{SM_{E9}, G_0, G_3\}$	54x435	2x133
$G_{loc5}=G_0$	$CC_{10}=\{SM_{E10}, G_0\}$	16x29	8x29
$G_{loc4}=G_0 G_4 G_1$	$CC_{11}=\{SM_{E11}, G_0, G_4, G_1\}$	324x2805	9x602

A figura 5.9 mostra a estrutura de transição de estados dos supervisores reduzidos S_0 à S_9 obtidos e apresentados na tabela 3 a partir da abordagem modular local, assim como as ações de controle dos respectivos supervisores sobre os eventos controláveis em cada estado.

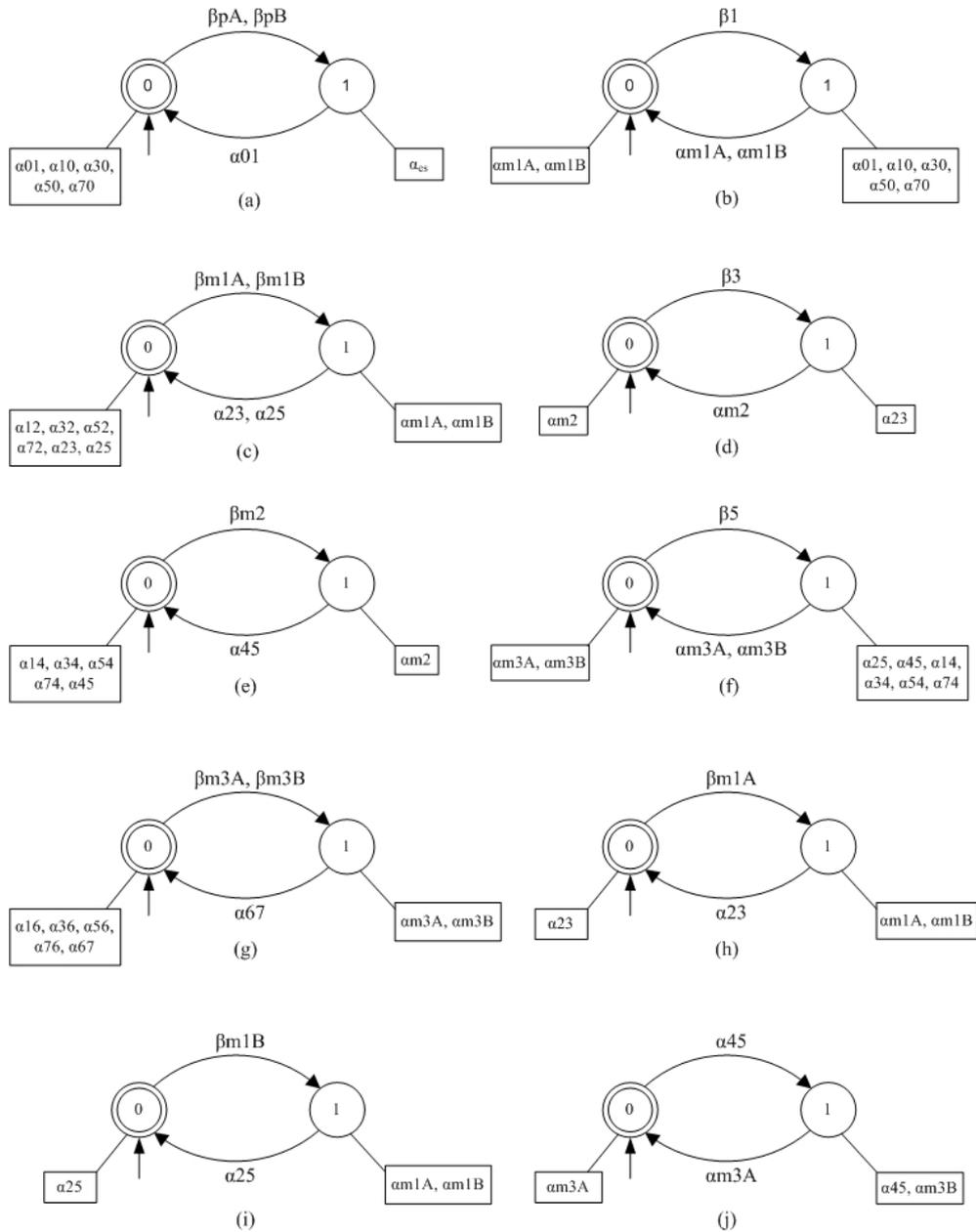


Figura 5.9: Estrutura de transição de estado dos supervisores reduzidos S_0 à S_9 .

A figura 5.10 apresenta o supervisor reduzido S_{10} referente às movimentações a serem desenvolvidas pelo robô. Nota-se uma semelhança deste supervisor com um dos modelos propostos no capítulo 4 para o sistema de transporte, mais especificamente o segundo modelo apresentado na seção 4.4. Internamente aos retângulos estão representadas as ações de controle nos respectivos estados.

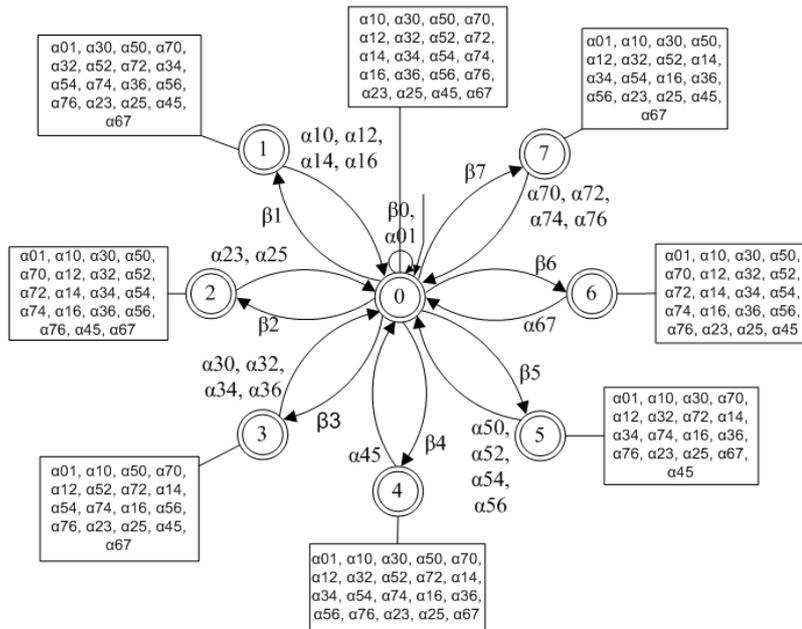


Figura 5.10: Estrutura de transição de estados do supervisor reduzido S_{10} .

A figura 5.11 apresenta a estrutura de transição do supervisor reduzido S_{11} que coordena ações dos subsistemas esteira transportadora de entrada, máquina M_1 e sistema de transporte, em função do estado simultâneo dos *buffers* B_0 e B_1 .

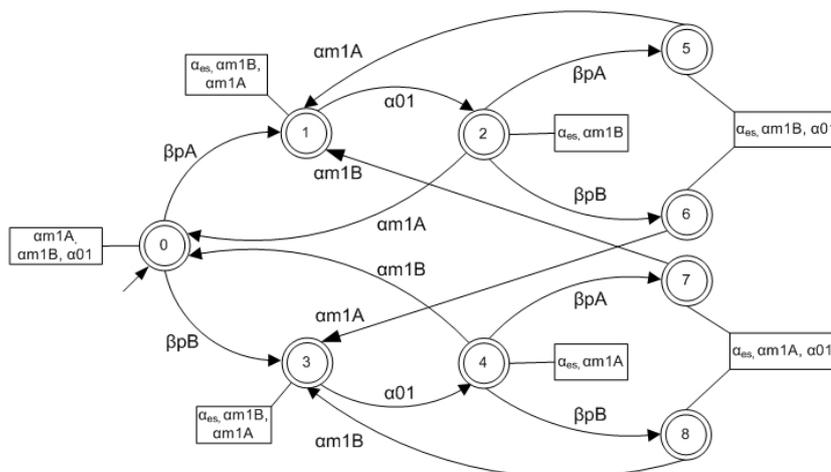


Figura 5.11: Estrutura de transição de estados do supervisor reduzido S_{11} .

A tabela 5.4 resume as ações de controle desenvolvidas pelos supervisores reduzidos (S_j) em cada um dos seus respectivos estados.

Tabela 5.4 – Ação de controle dos supervisores reduzidos	
S_0	$\Phi_0(0)=\{\alpha 01, \alpha 10, \alpha 30, \alpha 50, \alpha 70\}; \Phi_0(1)=\{\alpha_{es}\}$
S_1	$\Phi_1(0)=\{\alpha_{m1A}, \alpha_{m1B}\}; \Phi_1(1)=\{\alpha 01, \alpha 10, \alpha 30, \alpha 50, \alpha 70\}$
S_2	$\Phi_2(0)=\{\alpha 12, \alpha 32, \alpha 52, \alpha 72, \alpha 23, \alpha 25\}; \Phi_2(1)=\{\alpha_{m1A}, \alpha_{m1B}\}$
S_3	$\Phi_3(0)=\{\alpha_{m2}\}; \Phi_3(1)=\{\alpha 23\}$
S_4	$\Phi_4(0)=\{\alpha 14, \alpha 34, \alpha 54, \alpha 74, \alpha 45\}; \Phi_4(1)=\{\alpha_{m2}\}$
S_5	$\Phi_5(0)=\{\alpha_{m3A}, \alpha_{m3B}\}; \Phi_5(1)=\{\alpha 14, \alpha 34, \alpha 54, \alpha 74, \alpha 45\}$
S_6	$\Phi_6(0)=\{\alpha 16, \alpha 36, \alpha 56, \alpha 76, \alpha 67\}; \Phi_6(1)=\{\alpha_{m3A}, \alpha_{m3B}\}$
S_7	$\Phi_7(0)=\{\alpha 23\}; \Phi_7(1)=\{\alpha_{m1A}, \alpha_{m1B}\}$
S_8	$\Phi_8(0)=\{\alpha 25\}; \Phi_8(1)=\{\alpha_{m1A}, \alpha_{m1B}\}$
S_9	$\Phi_9(0)=\{\alpha_{m3A}\}; \Phi_9(1)=\{\alpha 45, \alpha_{m3B}\}$
S_{10}	$\Phi_{10}(0)=\{\alpha 10, \alpha 30, \alpha 50, \alpha 70, \alpha 12, \alpha 32, \alpha 52, \alpha 72, \alpha 14, \alpha 34, \alpha 54, \alpha 74, \alpha 16, \alpha 36, \alpha 56, \alpha 76, \alpha 23, \alpha 25, \alpha 45, \alpha 67\};$ $\Phi_{10}(1)=\{\alpha 01, \alpha 30, \alpha 50, \alpha 70, \alpha 32, \alpha 52, \alpha 72, \alpha 34, \alpha 54, \alpha 74, \alpha 36, \alpha 56, \alpha 76, \alpha 23, \alpha 25, \alpha 45, \alpha 67\};$ $\Phi_{10}(2)=\{\alpha 01, \alpha 10, \alpha 30, \alpha 50, \alpha 70, \alpha 12, \alpha 32, \alpha 52, \alpha 72, \alpha 14, \alpha 34, \alpha 54, \alpha 74, \alpha 16, \alpha 36, \alpha 56, \alpha 76, \alpha 45, \alpha 67\};$ $\Phi_{10}(3)=\{\alpha 01, \alpha 10, \alpha 50, \alpha 70, \alpha 12, \alpha 52, \alpha 72, \alpha 14, \alpha 54, \alpha 74, \alpha 16, \alpha 56, \alpha 76, \alpha 23, \alpha 25, \alpha 45, \alpha 67\};$ $\Phi_{10}(4)=\{\alpha 01, \alpha 10, \alpha 30, \alpha 50, \alpha 70, \alpha 12, \alpha 32, \alpha 52, \alpha 72, \alpha 14, \alpha 34, \alpha 54, \alpha 74, \alpha 16, \alpha 36, \alpha 56, \alpha 76, \alpha 23, \alpha 25, \alpha 67\};$ $\Phi_{10}(5)=\{\alpha 01, \alpha 10, \alpha 30, \alpha 70, \alpha 12, \alpha 32, \alpha 72, \alpha 14, \alpha 34, \alpha 74, \alpha 16, \alpha 36, \alpha 76, \alpha 23, \alpha 25, \alpha 67, \alpha 45\};$ $\Phi_{10}(6)=\{\alpha 01, \alpha 10, \alpha 30, \alpha 50, \alpha 70, \alpha 12, \alpha 32, \alpha 52, \alpha 72, \alpha 14, \alpha 34, \alpha 54, \alpha 74, \alpha 16, \alpha 36, \alpha 56, \alpha 76, \alpha 23, \alpha 25, \alpha 45\};$ $\Phi_{10}(7)=\{\alpha 01, \alpha 10, \alpha 30, \alpha 50, \alpha 12, \alpha 32, \alpha 52, \alpha 14, \alpha 34, \alpha 54, \alpha 16, \alpha 36, \alpha 56, \alpha 23, \alpha 25, \alpha 45, \alpha 67\}$
S_{11}	$\Phi_{11}(0)=\{\alpha_{m1A}, \alpha_{m1B}, \alpha 01\}; \Phi_{11}(1)=\Phi_{11}(3)=\{\alpha_{m1A}, \alpha_{m1B}, \alpha_{es}\};$ $\Phi_{11}(2)=\{\alpha_{m1B}, \alpha_{es}\}; \Phi_{11}(4)=\{\alpha_{m1A}, \alpha_{es}\};$ $\Phi_{11}(5)=\Phi_{11}(6)=\{\alpha 01, \alpha_{m1B}, \alpha_{es}\}; \Phi_{11}(7)=\Phi_{11}(8)=\{\alpha 01, \alpha_{m1A}, \alpha_{es}\}$

5.4 Conclusão ao capítulo

Neste capítulo foram apresentadas as especificações de controle para a coordenação das atividades de um sistema de manufatura multi-produtos. A sistematização dos procedimentos para a obtenção das especificações possibilita a utilização dos mesmos para outras aplicações, uma vez que é possível a generalização. Esta generalização associada ao estado individual e conjunto dos *buffers* facilita a obtenção de um grupo de especificações que coordenam as ações das plantas locais. As especificações que coordenam o estado dos *buffers* ficam restritas à um número definido de estados que varia entre dois e nove. As especificações de coordenação geral, e que tratam das operações específicas para cada peça, ficam restritas a dois estados. A especificação que coordena as movimentações do sistema de transporte depende do número de *buffers* presentes no processo que é função do número de máquinas. Sendo o número de *buffers* igual a z , tal especificação terá um número de estados igual a $z+1$.

Uma contribuição deste processo de modelagem é referente ao número aproximado de especificações que se faz necessária para cada aplicação. Este número se define a partir dos seguintes aspectos:

- i. Quantidade de *buffers* envolvidos no processo: no caso tratado foram modelados o estado de 7 *buffers*. Isto implica em uma especificação para cada *buffer*, o que resulta em 7 especificações;
- ii. Número de ramificações a partir de uma máquina específica: como exemplo cita-se a saída da máquina M_1 em que ocorrem duas divergências. A máquina M_1 fornece peças para as máquinas M_2 e M_3 . Isto implica em duas especificações;
- iii. Número de convergências para uma máquina específica: como exemplo cita-se o caso da máquina M_3 . Esta máquina recebe peças de duas outras máquinas, M_1 e M_2 . Isto implica em duas especificações;
- iv. Caso em que as peças saem de um único subsistema e são transportadas para um único subsistema: esta situação ocorre apenas entre a esteira de alimentação e a máquina M_1 . Sendo assim é necessária apenas uma especificação. Nos casos em que não ocorre esta situação as especificações ficam restritas aos itens (ii) e (iii);
- v. Ordenamento dos movimentos: este tipo de especificação depende do tipo de processo e fica restrita à uma especificação apenas.

No caso tratado são necessárias aproximadamente 13 especificações, como ocorreu uma exclusão mútua entre as especificações E_9 e E'_9 , foram necessárias apenas 12 especificações na obtenção do processo controlado. Esta proposta de trabalho que considera a abordagem modular local da Teoria de Controle Supervisório para a implementação em CLP's, demonstra as vantagens de utilização de um modelo cuja Arquitetura de Controle Supervisório é dividida em níveis. Esta característica possibilita alterações e re-configurações da lógica de controle de forma pontual. O teste de modularidade local realizado através da composição síncrona dos supervisores modulares revelou a inexistência de bloqueio, o que permite a implementação dos supervisores que terão a mesma ação de controle de um supervisor monolítico. O autômato obtido em tal teste resultou em 115986 estados e 651355 transições.

Capítulo 6

Implementação do Sistema de Controle Supervisório

Desenvolver um Sistema de Controle Supervisório significa estabelecer regras e procedimentos ao comportamento livre de um sistema. Estas ações são desenvolvidas por estruturas que imponham um comportamento desejado à um sistema de manufatura. Tais estruturas, chamadas supervisores, atuam de forma conjunta de modo a satisfazer as especificações comportamentais de rotas, sequenciamento ou seguranças (Brandin, 1994).

Os modelos dos subsistemas bem como dos supervisores, modeladas e sintetizadas nos dois capítulos anteriores, são agora implementadas em um controlador lógico programável. A arquitetura de controle supervisório utilizada para a implementação do sistema de controle atende ao modelo Queiroz e Cury (2002), na qual as estruturas de controle estão divididas em três níveis: Supervisores Modulares, Sistema Produto e Procedimentos Operacionais.

O modelo para implementação da Arquitetura de Controle Supervisório em CLP está de acordo com Vieira *et al.* (2007). As estruturas de controle em Autômatos de Estados Finitos, obtidas através da Teoria de Controle Supervisório são convertidas para *SFC's* (*Sequential Function Chart*). Para tanto, realiza-se o detalhamento dos procedimentos que auxiliam no processo de conversão e são apresentados os modelos obtidos a partir de tal processo.

6.1 Introdução

De acordo com Ramadge e Wonham (1989) uma arquitetura de controle (figura 3.9) constitui-se de duas estruturas: uma chamada planta, que gera os eventos de forma espontânea e assíncrona no tempo, e a outra constituída pelos supervisores, os quais exercem a ação de controle sobre o comportamento livre da planta. Esta ação de controle se baseia nos estados ativos dos supervisores, que são atualizados de acordo com a ocorrência de eventos gerados

na planta. Sendo assim, em cada estado ocorrerá uma ação de desabilitação de eventos controláveis que não serão permitidos ocorrer na planta.

A tarefa de implementação de um sistema de controle supervísório não é nada simples e tampouco direta, uma vez que se deve estabelecer a sincronia entre duas estruturas que atuam de formas bastante distintas: a planta, cujo comportamento na geração dos eventos é espontâneo, assíncrono e não simultâneo; e os CLP's, nos quais serão implementados os supervisores, cuja forma de operar é síncrona e baseada em ciclos de atualização. Neste sentido, Fabian e Hellgren (1998) enumeram alguns problemas que surgem quando da implementação dos supervisores em CLP's através das linguagens de programação considerando os tradicionais diagramas escadas: (i) Sinais e eventos, (ii) Causalidade, (iii) Escolha e (iv) Sincronização não exata.

Segundo Queiroz e Cury (2002), outro problema que surge é a abstração dos modelos em autômatos referentes aos subsistemas, utilizados na fase de modelagem para evitar a explosão do espaço de estados e facilitar o processo de síntese dos supervisores. A explosão do número de estados dos modelos referentes aos subsistemas é fator de grande importância para o sucesso da implementação de um Sistema de Controle Supervísório em CLP's. Durante a fase de modelagem este aspecto é tratado considerando a maior abstração possível das atividades referentes aos modelos. Sendo assim, consideram-se principalmente os eventos relevantes que sejam necessários às tarefas de coordenação entre os subsistemas. Deste modo, as atividades intermediárias à cada processo não são representadas nos referidos modelos.

6.2 Arquitetura de Controle Supervísório

Alguns dos problemas citados na seção anterior, os quais dizem respeito à implementação de uma Arquitetura de Controle Supervísório em CLP's, podem ser resolvidos com a inclusão de uma interface ao modelo proposto por Ramadge e Wonham (1989). Esta interface foi concebida por Queiroz e Cury (2002), e é constituída de dois níveis chamados: Sistema Produto e Procedimentos Operacionais.

O nível Sistema Produto são implementados os módulos que descrevem o comportamento de cada subsistema isoladamente em sua forma assíncrona, ou seja, não compartilhando eventos entre si. Neste nível são tratadas as questões pertinentes à simultaneidade e espontaneidade de eventos. As abstrações dos modelos dos subsistemas, as

quais estão relacionadas ao conjunto de atividades a serem desenvolvidas nas etapas intermediárias aos eventos de relevância, são descritas através dos procedimentos operacionais. Como eventos de relevância citam-se os eventos de início e fim de operação.

Por exemplo, um autômato que possa representar a operação de um torno, pode ser composto de apenas dois estados, um estado representando a situação da máquina em repouso, e um outro estado representando o torno em operação. Quando se considera o estado que representa o torno em operação, o modelo abstrato do autômato não representa as atividades que, por ventura, estejam se desenvolvendo (Vieira *et al.*, 2007). O desenvolvimento destas atividades é realizado através do nível dos procedimentos operacionais, que serão disparados a partir da ocorrência dos eventos relevantes e serão representados através de *Seqüencial Function Charts* (SFC). Para cada módulo do sistema produto é preciso um módulo de procedimento operacional.

A figura 6.1, representa a arquitetura de controle supervísório composta dos níveis supervisores modulares, sistema produto e procedimentos operacionais.

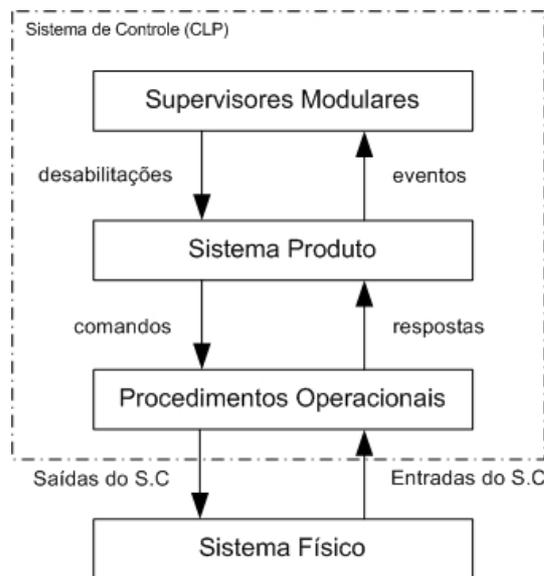


Figura 6.1: Arquitetura de Controle Supervísório (Queiroz e Cury, 2002).

O conjunto dos supervisores será composto de vários elementos, pois na Teoria de Controle Supervísório seguindo a abordagem Modular Local, cada supervisor realiza uma especificação e observa o comportamento de uma planta local. E cada planta local pode ser

composta de um ou mais módulos do sistema produto. De acordo com Vieira *et al.* (2004), esta arquitetura de controle supervísório possui algumas vantagens que são:

- i) Preserva a modularidade do sistema físico ao garantir que a inclusão ou exclusão de algum subsistema, não irá afetar o comportamento das estruturas anteriormente concebidas nos níveis sistema produto e procedimentos operacionais.
- ii) Preserva a modularidade das especificações por garantir que para a alteração das regras de controle, basta a inclusão ou exclusão do supervisor modular que realiza uma determinada especificação.
- iii) Permite a utilização de modelos mais abstratos, o que evita a explosão do número de estados na etapa de síntese dos supervisores, uma vez que o detalhamento necessário é feito no nível procedimentos operacionais.

6.3 Modelo para Implementação da Arquitetura de Controle Supervísório

O modelo para a implementação da arquitetura é como descrito em Vieira *et al.* (2007), o qual descreve a necessidade de alguns dos seguintes elementos de acordo com Queiroz e Cury (2002):

- O comportamento livre de cada subsistema será representado por um autômato $G_i = (Q_i, \Sigma_i, \delta_i, q_{0i}, Q_{mi})$, que corresponde aos módulos do nível sistema produto, com $i \in I = \{1, \dots, n\}$, onde n representa o número de subsistemas. O conjunto $\Sigma_c = \cup_{\forall i \in I} \Sigma_c^{G_i}$, $\Sigma_{uc} = \cup_{\forall i \in I} \Sigma_{uc}^{G_i}$ e $\Sigma = \Sigma_c \cup \Sigma_{uc}$.
- O conjunto dos procedimentos operacionais $(\{o_\sigma \mid \sigma \in \Sigma\})$, que descreve o detalhamento das operações referente aos módulos do sistema produto.
- As especificações de controle que definem as funções de coordenação entre os subsistemas, são realizadas por um conjunto de supervisores modulares reduzidos $S = \{S_j = (S'_j, \Phi_j)\}$.

De acordo com o modelo Vieira *et al.* (2007) o processo de implementação de um sistema de controle supervísório em CLP's requer considerar as seguintes condições:

1. cada subsistema físico que compõe a planta será representado por um autômato G_i com $i \in I$, e será associado à um módulo do sistema produto;

2. cada subsistema físico G_i será representado através de um SFC na seção de código da lógica do CLP. Tais SFC's serão identificados por um conjunto $\{g_i \mid i \in I\}$;
3. deverá ser associado um comando, a cada evento controlável do gerador G_i ;
4. deverá ser associada uma resposta, a cada evento não-controlável do gerador G_i ;
5. a cada evento $\sigma \in \Sigma$, poderá ser associado um procedimento operacional;
6. cada procedimento operacional e cada supervisor será representado através de um SFC na seção de código da lógica do CLP. Os SFC's referentes aos supervisores serão identificados por um conjunto $\{s_j \mid j \in J\}$.

A seção de código da lógica do CLP que constitui o programa principal, identificado como SFC Main, de acordo com Vieira *et al.*(2007) é apresentada na figura 6.2.

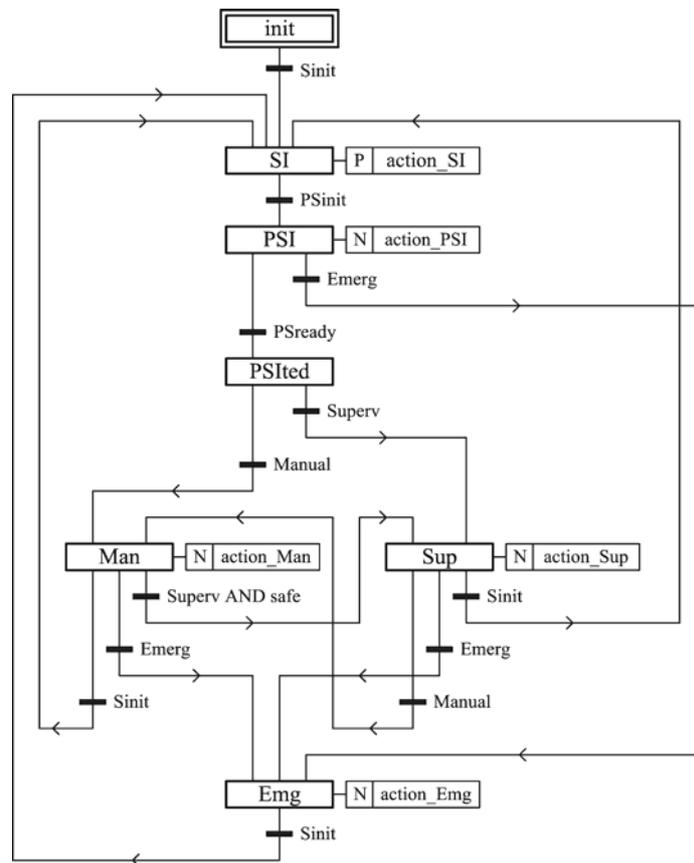


Figura 6.2: Programa principal SFC Main (Vieira *et al.*, 2007)

De acordo com Vieira *et al.* (2007), este programa estabelece seis modos distintos de operação para o sistema de controle, conforme o passo ativo do SFC que são:

- *Software Initialization* (SI);
- *Physical System Initialization* (PSI);
- *Manual* (Man);
- *Supervised* (Sup);
- *Emergency* (Emg);
- *Idle* (Init e PSIted).

Quando o sistema de controle encontra-se no modo *Software Initialization* (SI), a ação *action_SI* é executada apenas uma vez e todos os SFC's nos conjuntos $\{g_i | i \in I\}$, $\{s_j | j \in J\}$, $\{o_\sigma | \sigma \in \Sigma\}$ são inicializados, assim como todas as variáveis envolvidas no programa assumem um determinado valor.

A inicialização do sistema físico é realizada pelo passo PSI (*Physical System Initialization*). A ativação deste passo faz com que o sistema a ser controlado vá para o estado inicial desejado. A ação *action_PSI* é executada uma vez a cada ciclo de atualização do CLP.

No instante em que o estado inicial do sistema físico é alcançado, a variável booleana "PSready" assume valor lógico verdadeiro. Isto faz com que o passo PSIted do SFC Main seja ativado conduzindo o sistema de controle ao modo *Idle*.

Enquanto o sistema de controle estiver no modo de operação *Idle*, existem duas formas possíveis de se controlar o sistema físico: Uma é através da ativação do modo *Manual* (Man) e a outra forma é pelo modo *Supervisionado* (Sup).

No modo *Manual*, o operador conduz a realização das atividades a serem executadas pelo sistema físico. Isto é possível através da desativação seletiva dos sinais de desabilitação associados aos eventos controláveis, uma vez que neste modo de operação todos estes sinais são ativados, o que impede a geração espontânea de qualquer evento controlável. Este modo de operação é particularmente adequado na fase de teste do sistema do sistema de controle, na qual pode-se conduzir o sistema físico à um estado desejado, através de uma determinada cadeia de eventos.

Quando o sistema de controle encontra-se no modo supervisionado (passo Sup ativado) a coordenação das ações a serem desenvolvidas pelo sistema físico é de acordo com o estado ativo do conjunto de supervisores. Isto é realizado através da ativação do sinal de desabilitação associado aos eventos controláveis.

A ativação do modo *Emergency* (passo Emg) conduz o sistema físico a um estado apropriado de acordo com as características de cada processo e a conseqüente suspensão das atividades que até então estavam sendo realizadas.

A ativação de alguns destes passos do SFC *Main* resulta na chamada ordenada de alguns *Functions Blocks*. Tais *Functions Blocks* são concebidos para a realização das tarefas pertinentes ao sistema de controle como desempenhar as ações dos supervisores, realização dos procedimentos operacionais e módulos do sistema produto, assim como a suspensão ou autorização para tratamento de eventos. A tabela 6.1 (Vieira *et al.*, 2007) apresenta a organização destas estruturas.

Tabela 6.1 - Program Organization Units (POU)		
nome	tipo	comentários
s_j $j \in J$	FB	Representa o supervisor S_j
MS	FB	Estrutura lógica que executa a chamada ordenada de todos os FB's em $\{s_j \mid j \in J\}$
g_i $i \in I$	FB	Representa o módulo do sistema produto associado ao subsistema G_i
dg_i $i \in I$	FB	Estrutura lógica que suspende ou autoriza o tratamento de eventos em Σ^{G_i}
PS	FB	Estrutura lógica que executa a chamada ordenada de todos os FB's em $\{g_i \mid i \in I\}$ e $\{dg_i \mid i \in I\}$
o_σ $\sigma \in \Sigma$	FB	Representa o procedimento operacional o_σ
OP	FB	Estrutura lógica que executa a chamada ordenada de todos os FB's em $\{o_\sigma \mid \sigma \in \Sigma\}$
Main	<i>Program</i>	Coordena a operação da arquitetura de controle supervisão

O *Function Block* $\{s_j \mid j \in J\}$ representa cada um dos supervisores modulares reduzidos S_j obtidos no processo de síntese. O *Function Block* MS é a estrutura lógica que executa a chamada ordenada de cada FB em $\{s_j \mid j \in J\}$. Isto permite a atualização do estado ativo do conjunto de supervisores modulares reduzidos num único ciclo de atualização do

CLP. O FB MS em conjunto com os FB's em $\{s_j | j \in J\}$ constituem o nível dos Supervisores Modulares (MS) da Arquitetura de Controle Supervisório.

O *Function Block* em $\{g_i | i \in I\}$ representa o módulo do sistema associado ao subsistema G_i . O *Function Block* em $\{dg_i | i \in I\}$ é uma estrutura lógica que permite a suspensão ou autorização do tratamento de eventos em Σ^{G_i} . Tal procedimento é necessário para evitar o tratamento simultâneo de eventos dos subsistemas G_i que compartilham uma mesma célula de controle.

Estes dois conjuntos de FB's, $\{g_i | i \in I\}$ e $\{dg_i | i \in I\}$, juntamente com o FB PS constituem o nível Sistema Produto (PS) da Arquitetura de Controle Supervisório. A função do FB PS é realizar a chamada ordenada dos dois conjuntos de FB's $\{g_i | i \in I\}$ e $\{dg_i | i \in I\}$, sendo que a chamada de cada FB em $\{dg_i | i \in I\}$ é executada imediatamente antes de se realizar a chamada do correspondente FB em $\{g_i | i \in I\}$.

O conjunto de FB's em $\{o_\sigma | \sigma \in \Sigma\}$ os quais representam os procedimentos operacionais associados aos eventos $\sigma \in \Sigma$ juntamente com o FB OP, constituem o nível Procedimentos Operacionais (OP) da Arquitetura de Controle Supervisório. A função do FB OP é executar a chamada ordenada de todos os FB's em $\{o_\sigma | \sigma \in \Sigma\}$.

Nas seções a seguir serão descritos os procedimentos para a obtenção das estruturas referente a cada um dos três elementos citados no início desta seção, ou seja, os módulos do sistema produto, os procedimentos operacionais e os supervisores modulares.

6.4 Obtenção dos SFC's g_i no conjunto $\{g_i | i \in I\}$ e funções lógicas no conjunto $\{dg_i | i \in I\}$.

Esta seção apresenta os procedimentos sistemáticos para a conversão de cada autômato no conjunto $\{G_i | i \in I\}$ no correspondente SFC g_i . Para tanto, são desenvolvidos os procedimentos que auxiliam na definição dos conjuntos de variáveis a serem utilizadas no processo de implementação. São apresentadas as propriedades a serem satisfeitas para a conversão dos autômatos G_i nos correspondentes autômatos H_i . Esta seção é concluída com a obtenção da função lógica dg_i , $i \in I$.

6.4.1 Definição das variáveis do CLP

Inicialmente é preciso definir um conjunto de variáveis a serem utilizadas no processo de implementação. De acordo com Vieira *et al.* (2007) deve-se associar uma variável booleana do CLP chamada “CED” (Desabilitação de Evento Controlável) com a Representação do Sistema Produto $\{G_i \mid i \in I\}$. A ativação desta variável torna falsa todas as condições de transição dos SFCs g_i , $i \in I$, associadas aos eventos controláveis em Σ^{G_i} . Isto permite priorizar o tratamento de eventos não-controláveis em detrimento da geração de eventos controláveis.

O modelo proposto por Vieira *et al.* (2007) propõe ainda a utilização de duas outras variáveis booleanas do CLP que devem estar associadas a cada módulo G_i do sistema produto, com $i \in I$. Estas variáveis são identificadas como " $g_i\text{evt}$ " e " dg_i ". A utilização destas variáveis evita a simultaneidade de eventos, sendo que a ativação da variável " $g_i\text{evt}$ " sinaliza o tratamento de eventos em Σ^{G_i} , e a ativação da variável " dg_i " suspende o tratamento de eventos em Σ^{G_i} .

Um total de três outras variáveis booleanas deverá ser associada a cada evento controlável $\sigma \in \Sigma^{G_i}$, $i \in I$. Estas variáveis fazem parte de três conjuntos distintos sendo " σ " $\in E_i$, " σd " $\in D_i$ e " $\text{cmd } \sigma$ " $\in C_i$. O conjunto E_i representa o conjunto de variáveis booleanas do CLP que sinalizam a ocorrência de um evento $\sigma \in \Sigma_c^{G_i}$. O conjunto D_i representa as variáveis booleanas do CLP que sinalizam a desabilitação de sinal de um evento controlável. O conjunto C_i representa as variáveis booleanas do CLP que sinalizam os comandos associados a cada evento controlável em $\sigma \in \Sigma_c^{G_i}$. A ativação desta última variável dispara os procedimentos operacionais associados a tais comandos. Aos eventos não-controláveis $\sigma \in \Sigma_{uc}^{G_i}$, $i \in I$, associa-se uma variável inteira e sem sinal do CLP, que fará parte do conjunto " $\text{rsp } \sigma$ " $\in R_i$.

Os quatro conjuntos de variáveis booleanas do CLP E_i , D_i , C_i e R_i são tais que $|E_i| = |\Sigma^{G_i}|$, $|D_i| = |C_i| = |\Sigma_c^{G_i}|$ e $|R_i| = |\Sigma_{uc}^{G_i}|$ (Vieira *et al.*, 2007).

6.4.2 Conversão dos Autômatos G_i em Autômatos H_i .

Cada módulo do sistema produto é representado por um autômato $G_i = (\Sigma^{G_i}, Q^{G_i}, \delta^{G_i}, q0^{G_i}, Q_m^{G_i})$. Todo autômato G_i será convertido no correspondente autômato $H_i = (\Sigma^{H_i}, Q^{H_i}, \delta^{H_i}, q0^{H_i}, Q_m^{H_i})$ em que:

Σ^{H_i} - alfabeto de eventos, $\Sigma^{H_i} = \Sigma^{G_i}$;

$q0^{H_i}$ - estado inicial;

Q^{H_i} - conjunto de estados, com $Q^{H_i} \subseteq \{Q^{G_i} \times \Sigma^{G_i}\} \cup \{q0^{H_i}\}$;

δ^{H_i} - função de transição de estados, $\delta^{H_i}: Q^{H_i} \times \Sigma^{H_i} \rightarrow Q^{H_i}$;

$Q_m^{H_i}$ - conjunto de estados marcados.

A conversão do autômato $G_i, i \in I$, no autômato H_i é realizada de modo que H_i satisfaça as seguintes propriedades:

- i. $L(H_i) = L(G_i)$;
- ii. $\forall((q1, q2 \in Q^{H_i}), (\sigma a, \sigma b \in \Sigma^{H_i}))$
 $\delta^{H_i}(q1, \sigma a) = \delta^{H_i}(q2, \sigma b)$ somente se $\sigma a = \sigma b$.
- iii. $\forall((q \in Q^{H_i}), (\sigma \in \Sigma^{H_i}))$
 $\delta^{H_i}(q, \sigma) \neq q0^{H_i}$;
- iv. $\forall((q \in Q^{H_i}), (\sigma \in \Sigma^{H_i}))$
 $\delta^{H_i}(q, \sigma) \neq q$.

A propriedade (i) estabelece que linguagem gerada pelo autômato H_i é exatamente a mesma que pode ser gerada pelo autômato G_i . Esta propriedade faz referência a capacidade do autômato H_i não reconhecer tarefas completas. Sendo assim, ambos os autômatos descrevem o mesmo comportamento livre dos correspondentes subsistemas. A propriedade (ii) estabelece que cada estado de H_i será sempre atingido através da ocorrência do mesmo evento.

A propriedade (iii) significa que a ocorrência de qualquer evento $\sigma \in \Sigma^{H_i}$, jamais conduzirá autômato H_i ao seu estado inicial $q0^{H_i}$. A propriedade (iv) implica que a ocorrência de qualquer evento em $\sigma \in \Sigma^{H_i}$ deve resultar na transição de um q à um estado q' , ou seja, o autômato H_i não possui auto-laços. Isto somente ocorrerá se autômato G_i , também não possuir auto-laços.

O autômato H_i é obtido como segue:

- i. $q0^{H_i} = (q0^{G_i}, \sigma_{dummy})$, tal que $\sigma_{dummy} \notin \Sigma$;
- ii. $\forall((q \in Q^{G_i}), (\sigma_1 \in \Sigma^{G_i}), (\sigma_2 \in \Sigma^{G_i} \cup \{\sigma_{dummy}\}))$
 $\delta^{H_i}((q, \sigma_2), \sigma_1) = (\delta^{G_i}(q, \sigma_1), \sigma_1)$;
- iii. $Qm^{H_i} = \emptyset$

A definição do item (iii) acima é baseada no fato de que, embora o conjunto de estados marcados do autômato empregado para descrever o comportamento livre do sistema físico e das especificações de controle seja importante para a síntese dos supervisores, não se considera tal informação para a obtenção dos SFCs no conjunto $\{g_i \mid i \in I\}$.

Exemplo 5.1

Considere a figura 4.4 na qual é apresentado o autômato referente a esteira transportadora de entrada, subsistema G_4 . Este ilustra a obtenção do autômato H_4 .

Seja $\Sigma^{G_4} = \{\alpha_{es}, \beta pA, \beta pB\}$ com $\Sigma_c^{G_4} = \{\alpha_{es}\}$ e $\Sigma_{uc}^{G_4} = \{\beta pA, \beta pB\}$. Sendo $Q^{G_4} = \{0, 1\}$ obtém-se $Q^{G_4} \times \Sigma^{G_4} = \{(0, \alpha_{es}), (0, \beta pA), (0, \beta pB), (1, \alpha_{es}), (1, \beta pA), (1, \beta pB)\}$. Para $q0^{H_4} = (0, \sigma_{dummy})$ tem-se $Q^{H_4} \subseteq \{Q^{G_4} \times \Sigma^{G_4}\} \cup \{q0^{H_4}\}$.

A obtenção das funções de transição de estados $\delta^{H_4} : Q^{H_4} \times \Sigma^{H_4} \rightarrow Q^{H_4}$ é como segue:

$\delta^{H_4} = ((0, \sigma_{dummy}), \alpha_{es}) = (\delta^{G_4}(0, \alpha_{es}), \alpha_{es}) = (1, \alpha_{es})!$, a função é definida.

$\neg \delta^{H_4} = ((0, \sigma_{dummy}), \beta pA) = (\delta^{G_4}(0, \beta pA), \beta pA)!$, a função não é definida.

$\neg \delta^{H_4} = ((0, \sigma_{dummy}), \beta pB) = (\delta^{G_4}(0, \beta pB), \beta pB)!$, a função não é definida.

$\delta^{H_4} = ((0, \sigma_{es}), \alpha_{es}) = (\delta^{G_4}(0, \alpha_{es}), \alpha_{es}) = (1, \alpha_{es})!$, a função é definida.

$\neg \delta^{H_4} = ((0, \sigma_{es}), \beta pA) = (\delta^{G_4}(0, \beta pA), \beta pA)!$, a função não é definida.

$\neg \delta^{H_4} = ((0, \sigma_{es}), \beta pB) = (\delta^{G_4}(0, \beta pB), \beta pB)!$, a função não é definida.

$\delta^{H_4} = ((0, \beta pA), \alpha_{es}) = (\delta^{G_4}(0, \alpha_{es}), \alpha_{es}) = (1, \alpha_{es})!$, a função é definida.

$\neg \delta^{H_4} = ((0, \beta pA), \beta pA) = (\delta^{G_4}(0, \beta pA), \beta pA)!$, a função não é definida.

$\neg \delta^{H_4} = ((0, \beta pA), \beta pB) = (\delta^{G_4}(0, \beta pB), \beta pB)!$, a função não é definida.

$\delta^{H_4} = ((0, \beta pB), \alpha_{es}) = (\delta^{G_4}(0, \alpha_{es}), \alpha_{es}) = (1, \alpha_{es})!$, a função é definida.

$\neg\delta^{H_4} = ((0, \beta pB), \beta pA) = (\delta^{G_4}(0, \beta pA), \beta pA)!$, a função não é definida.

$\neg\delta^{H_4} = ((0, \beta pB), \beta pB) = (\delta^{G_4}(0, \beta pB), \beta pB)!$, a função não é definida.

$\neg\delta^{H_4} = ((1, \sigma_{es}), \alpha_{es}) = (\delta^{G_4}(1, \alpha_{es}), \alpha_{es})!$, a função não é definida.

$\delta^{H_4} = ((1, \sigma_{es}), \beta pA) = (\delta^{G_4}(1, \beta pA), \beta pA) = (0, \beta pA)!$, a função é definida.

$\delta^{H_4} = ((1, \sigma_{es}), \beta pB) = (\delta^{G_4}(1, \beta pB), \beta pB) = (0, \beta pB)!$, a função é definida.

$\neg\delta^{H_4} = ((1, \beta pA), \alpha_{es}) = (\delta^{G_4}(1, \alpha_{es}), \alpha_{es})!$ a função não é definida.

$\delta^{H_4} = ((1, \beta pA), \beta pA) = (\delta^{G_4}(1, \beta pA), \beta pA) = (0, \beta pA)!$, a função é definida.

$\delta^{H_4} = ((1, \beta pA), \beta pB) = (\delta^{G_4}(1, \beta pB), \beta pB) = (0, \beta pB)!$, a função é definida.

$\neg\delta^{H_4} = ((1, \beta pB), \alpha_{es}) = (\delta^{G_4}(1, \alpha_{es}), \alpha_{es})!$, a função não é definida.

$\delta^{H_4} = ((1, \beta pB), \beta pA) = (\delta^{G_4}(1, \beta pA), \beta pA) = (0, \beta pA)!$, a função é definida.

$\delta^{H_4} = ((1, \beta pB), \beta pB) = (\delta^{G_4}(1, \beta pB), \beta pB) = (0, \beta pB)!$, a função é definida.

A verificação das funções de transição de estados $\delta^{H_4} : Q^{H_4} \times \Sigma^{H_4} \rightarrow Q^{H_4}$, resulta que os estados do autômato H_4 são $Q^{H_4} = \{(0, \sigma_{dummy}), (0, \beta pA), (0, \beta pB), (1, \alpha_{es})\}$. A figura 6.3 apresenta o autômato H_4 .

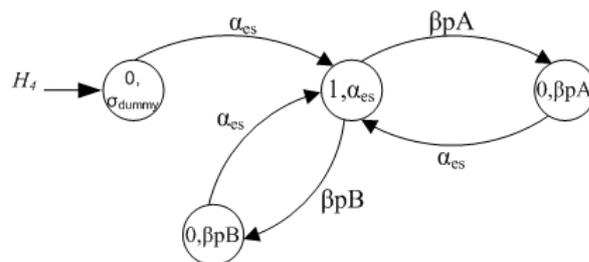


Figura 6.3: Autômato H_4 correspondente ao módulo G_4 .

Exemplo 5.2

Considere a figura 4.3, a qual representa os autômatos das máquinas M_i . Toma-se como exemplo, para ilustrar os procedimentos de conversão de um autômato G_i em um autômato H_i , o autômato G_I da máquina M_I .

Para este subsistema G_I tem-se que:

$$Q^{G_1} = \{0, 1, 2\}$$

$$\Sigma^{G_1} = \{\alpha_{m1A}, \alpha_{m1B}, \beta_{m1A}, \beta_{m1B}\} \text{ sendo } \Sigma_c^{G_1} = \{\alpha_{m1A}, \alpha_{m1B}\} \text{ e } \Sigma_{uc}^{G_1} = \{\beta_{m1A}, \beta_{m1B}\}$$

$$q0^{H_1} = (0, \sigma_{dummy})$$

$$Q^{G_1} \times \Sigma^{G_1} = \{(0, \alpha_{m1A}), (0, \alpha_{m1B}), (0, \beta_{m1A}), (0, \beta_{m1B}), (1, \alpha_{m1A}), (1, \alpha_{m1B}), (1, \beta_{m1A}), (1, \beta_{m1B}), (2, \alpha_{m1A}), (2, \alpha_{m1B}), (2, \beta_{m1A}), (2, \beta_{m1B})\}$$

Conforme exposto nesta seção $Q^{H_1} \subseteq \{Q^{G_1} \times \Sigma^{G_1}\} \cup \{q0^{H_1}\}$.

A função de transição de estados $\delta^{H_1} : Q^{H_1} \times \Sigma^{H_1} \rightarrow Q^{H_1}$ é obtida através de:

$$\delta^{H_1} = ((0, \sigma_{dummy}), \alpha_{m1A}) = (\delta^{G_1}(0, \alpha_{m1A}), \alpha_{m1A}) = (1, \alpha_{m1A})!, \text{ a função é definida.}$$

$$\delta^{H_1} = ((0, \sigma_{dummy}), \alpha_{m1B}) = (\delta^{G_1}(0, \alpha_{m1B}), \alpha_{m1B}) = (2, \alpha_{m1B})!, \text{ a função é definida.}$$

$$\delta^{H_1} = ((0, \alpha_{m1A}), \alpha_{m1A}) = (\delta^{G_1}(0, \alpha_{m1A}), \alpha_{m1A}) = (1, \alpha_{m1A})!, \text{ a função é definida.}$$

$$\delta^{H_1} = ((0, \alpha_{m1A}), \alpha_{m1B}) = (\delta^{G_1}(0, \alpha_{m1B}), \alpha_{m1B}) = (2, \alpha_{m1B})!, \text{ a função é definida.}$$

$$\delta^{H_1} = ((0, \alpha_{m1B}), \alpha_{m1A}) = (\delta^{G_1}(0, \alpha_{m1A}), \alpha_{m1A}) = (1, \alpha_{m1A})!, \text{ a função é definida.}$$

$$\delta^{H_1} = ((0, \alpha_{m1B}), \alpha_{m1B}) = (\delta^{G_1}(0, \alpha_{m1B}), \alpha_{m1B}) = (2, \alpha_{m1B})!, \text{ a função é definida.}$$

$$\delta^{H_1} = ((0, \beta_{m1A}), \alpha_{m1A}) = (\delta^{G_1}(0, \alpha_{m1A}), \alpha_{m1A}) = (1, \alpha_{m1A})!, \text{ a função é definida.}$$

$$\delta^{H_1} = ((0, \beta_{m1A}), \alpha_{m1B}) = (\delta^{G_1}(0, \alpha_{m1B}), \alpha_{m1B}) = (2, \alpha_{m1B})!, \text{ a função é definida.}$$

$$\delta^{H_1} = ((0, \beta_{m1B}), \alpha_{m1A}) = (\delta^{G_1}(0, \alpha_{m1A}), \alpha_{m1A}) = (1, \alpha_{m1A})!, \text{ a função é definida.}$$

$$\delta^{H_1} = ((0, \beta_{m1B}), \alpha_{m1B}) = (\delta^{G_1}(0, \alpha_{m1B}), \alpha_{m1B}) = (2, \alpha_{m1B})!, \text{ a função é definida.}$$

$$\delta^{H_1} = ((1, \alpha_{m1A}), \beta_{m1A}) = (\delta^{G_1}(1, \beta_{m1A}), \beta_{m1A}) = (0, \beta_{m1A})!, \text{ a função é definida.}$$

$$\delta^{H_1} = ((1, \beta_{m1A}), \beta_{m1A}) = (\delta^{G_1}(1, \beta_{m1A}), \beta_{m1A}) = (0, \beta_{m1A})!, \text{ a função é definida.}$$

$$\delta^{H_1} = ((1, \beta_{m1B}), \beta_{m1A}) = (\delta^{G_1}(1, \beta_{m1A}), \beta_{m1A}) = (0, \beta_{m1A})! \text{ a função é definida.}$$

$$\delta^{H_1} = ((1, \alpha_{m1B}), \beta_{m1A}) = (\delta^{G_1}(1, \beta_{m1A}), \beta_{m1A}) = (0, \beta_{m1A})!, \text{ a função é definida.}$$

$$\delta^{H_1} = ((2, \alpha_{m1A}), \beta_{m1B}) = (\delta^{G_1}(2, \beta_{m1B}), \beta_{m1B}) = (0, \beta_{m1B})!, \text{ a função é definida.}$$

$$\delta^{H_1} = ((2, \alpha_{m1B}), \beta_{m1B}) = (\delta^{G_1}(2, \beta_{m1B}), \beta_{m1B}) = (0, \beta_{m1B})!, \text{ a função é definida.}$$

$$\delta^{H_1} = ((2, \beta_{m1A}), \beta_{m1B}) = (\delta^{G_1}(2, \beta_{m1B}), \beta_{m1B}) = (0, \beta_{m1B})!, \text{ a função é definida.}$$

$$\delta^{H_1} = ((2, \beta_{m1B}), \beta_{m1B}) = (\delta^{G_1}(2, \beta_{m1B}), \beta_{m1B}) = (0, \beta_{m1B})!, \text{ a função é definida.}$$

Para todos os outros casos em que $\delta^{G_1} = (q, \sigma 1)$ não está definida, a função de transição de estados $\delta^{H_1} : Q^{H_1} \times \Sigma^{H_1} \rightarrow Q^{H_1}$ não é definida. Isto resulta que $Q^{H_1} = \{(0, \sigma_{dummy}), (0, \beta_{m1A}), (0, \beta_{m1B}), (1, \alpha_{m1A}), (2, \alpha_{m1B})\}$.

A figura 6.4 apresenta o autômato H_1 referente ao subsistema G_1 (máquina M_1).

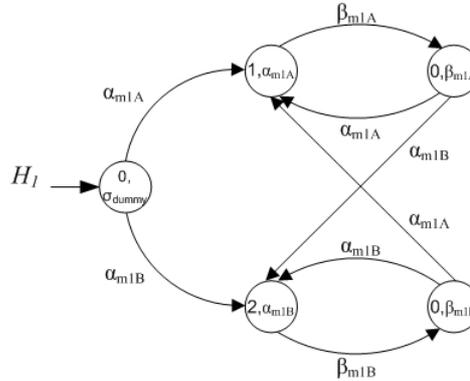


Figura 6.4: Autômato H_1 correspondente ao módulo G_1 .

Através dos procedimentos realizados nos exemplos 5.1 e 5.2 obtém os autômatos H_2 e H_3 apresentados nas figuras 6.5 e 6.6 respectivamente.

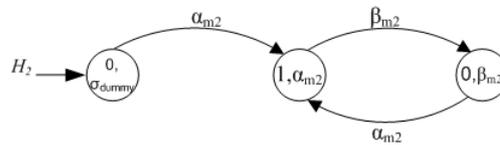


Figura 6.5: Autômato H_2 correspondente ao módulo G_2 (máquina M_2).

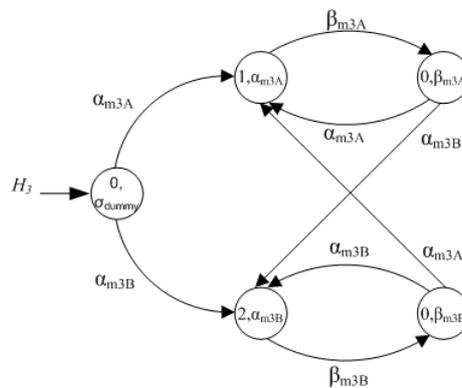


Figura 6.6: Autômato H_3 correspondente ao módulo G_3 .

O autômato H_0 correspondente ao módulo G_0 , referente ao sistema de transporte não está sendo representado em função do seu elevado número de estados que corresponde a 30 estados, e também devido ao elevado número de transições que totalizam 210 transições.

6.4.3 Obtenção dos SFCs g_i correspondentes aos Autômatos H_i

De acordo com o modelo Vieira *et al.* (2007), uma vez realizada a conversão dos autômatos G_i no correspondente autômato H_i , deve-se prosseguir na obtenção do SFC g_i , com $i \in I$. Tal SFC corresponde aos módulos do sistema produto a ser implementado no CLP, e poderá ser representado como SFC $g_i = (X^{g_i}, T^{g_i}, x_0^{g_i})$, $i \in I$.

Vieira *et al.* (2007) descrevem os passos de (i) a (iii) a serem desenvolvidos para a obtenção do SFC g_i , $i \in I$, com $T^{g_i} = T^{H_i} \cup T^{G_i}$.

- i) Um passo $xq \in X^{g_i}$ é associado a cada estado $q \in Q^{H_i}$, de tal modo que $|X^{g_i}| = |Q^{H_i}|$;
- ii) $x_0^{g_i}$ é o passo associado a $q_0^{H_i}$;
- iii) Uma transição $(xq, xq') \in T^{g_i}$ é associada a cada tripla ordenada (q, σ, q') de H_i , onde $q' = \delta^{H_i}(q, \sigma)$ e os passos $xq, xq' \in X^{g_i}$ são associados, respectivamente, aos estados $q, q' \in Q^{H_i}$. Esta associação tem de ser feita de modo que $|T^{g_i}|$ seja igual ao número da tripla ordenada;

Pode-se observar que a transição $(xq, xq') \in T^{g_i}$, $i \in I$, é associada ao evento $\sigma \in \Sigma^{G_i}$ através da tripla ordenada (q, σ, q') . A condição de transição associada com a transição no conjunto T^{g_i} , $i \in I$, é a expressão booleana (i) ou (ii) abaixo. Estas expressões assumem que o evento σ é associado a tal transição bem como às variáveis do CLP σd ou $\text{rsp } \sigma$.

i) NOT σd AND NOT CED;

ii) $\text{rsp } \sigma > 0$.

A expressão (i) deve ser empregada quando o evento associado é controlável com $\sigma \in \Sigma_c^{G_i}$ e a expressão (ii) é associada a um evento não-controlável $\sigma \in \Sigma_{uc}^{G_i}$.

Desde que o estado $q \in Q^{H_i}$, sendo $q \neq q_0^{H_i}$ e $i \in I$, é um par ordenado em $\{Q^{G_i} \times \Sigma^{G_i}\}$ pode-se dizer que tal estado está associado a um evento $\sigma \in \Sigma^{G_i}$ que constitui o

segundo elemento de tal par ordenado. Como uma consequência pode-se dizer que o passo $xq \in X^{E_i}$, que está associado a tal estado, é também associado ao evento σ .

Exemplo 5.3

Este exemplo ilustra a obtenção do SFC g_i , correspondente ao subsistema esteira transportadora de entrada, autômato G_4 (figura 4.4). De acordo com Vieira *et al.* (2007) e conforme exposto na seção 6.4.1.1 associa-se uma variável ao subsistema G_4 a qual segue a seguinte notação (G_4 , $g4evt$). Também realiza-se a mesma associação de acordo com os eventos que pertencem ao subsistema sendo: (α_{es}, aes) , $(\alpha_{es}, cmdaes)$, $(\alpha_{es}, aesda)$, $(\beta pA, bpa)$, $(\beta pA, rspbpa)$ e $(\beta pB, bpb)$, $(\beta pB, rspbpb)$. Os conjuntos obtidos são $C_4=\{cmdaes\}$, $D_4=\{aesd\}$, $E_4=\{aes, bpa, bpb\}$ e $R_4=\{rspbpa, rspbpb\}$. Do exposto nesta seção, deve-se realizar a associação dos estados do autômato H_4 com os passos do correspondente SFC g_4 o que resulta em: $((0, \sigma_{dummy}), x0)$, $((1, \alpha_{es}), x1)$, $((0, \beta pA), x2)$ e $((0, \beta pB), x3)$.

A obtenção das triplas ordenadas, associadas ao autômato H_4 (figura 6.3), resulta em: $((0, \sigma_{dummy}), \alpha_{es}, (1, \alpha_{es}))$, $((1, \alpha_{es}), \beta pA, (0, \beta pA))$, $((1, \alpha_{es}), \beta pB, (0, \beta pB))$, $((0, \beta pA), \alpha_{es}, (1, \alpha_{es}))$, $((0, \beta pB), \alpha_{es}, (1, \alpha_{es}))$.

As triplas ordenadas associadas ao SFC g_4 são: $(x0, aes, x1)$, $(x1, bpa, x2)$, $(x1, bpb, x3)$, $(x2, aes, x1)$ e $(x3, aes, x1)$. As expressões booleanas associadas aos eventos são: $(aes, NOT aesda AND NOT CED)$, $(bpa, rspbpa > 0)$, $(bpb, rspbpb > 0)$. O SFC g_4 resultante é apresentado na figura 6.7.

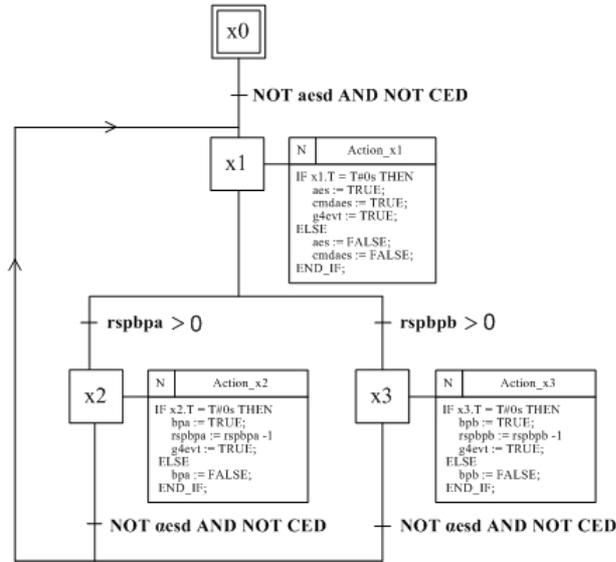


Figura 6.7: SFC g_4 da esteira transportadora de entrada.

Exemplo 5.4

Este exemplo ilustra a obtenção do SFC g_1 , correspondente ao subsistema máquina M_1 , autômato G_I (figura 4.3a). A associação da variável do CLP ao módulo G_I é $(G_1, g1evt)$. A associação das variáveis do CLP aos eventos pertencentes ao subsistema G_I são: $(\alpha_{m1A}, am1a)$, $(\alpha_{m1A}, cmdam1a)$, $(\alpha_{m1A}, am1ad)$, $(\alpha_{m1B}, am1b)$, $(\alpha_{m1B}, cmdam1b)$, $(\alpha_{m1B}, am1bd)$, $(\beta_{m1A}, bm1a)$, $(\beta_{m1A}, rspbm1a)$, $(\beta_{m1B}, bm1b)$, $(\beta_{m1B}, rspbm1b)$. Os conjuntos obtidos são $C_4 = \{cmdam1a, cmdam1b\}$, $D_4 = \{am1ad, am1bd\}$, $E_4 = \{am1a, am1b, bm1a, bm1b\}$ e $R_4 = \{rspbm1a, rspbm1b\}$. A associação dos estados do autômato H_I (figura 6.4) com os passos do correspondente SFC g_1 resulta em: $((0, \sigma_{dummy}), x_0)$, $((1, \alpha_{m1A}), x_1)$, $((2, \alpha_{m1B}), x_3)$, $((0, \beta_{m1A}), x_2)$, $((0, \beta_{m1B}), x_4)$.

A obtenção das triplas ordenadas associadas ao autômato H_I resulta em: $((0, \sigma_{dummy}), \alpha_{m1A}, (1, \alpha_{m1A}))$, $((0, \sigma_{dummy}), \alpha_{m1B}, (2, \alpha_{m1B}))$, $((1, \alpha_{m1A}), \beta_{m1A}, (0, \beta_{m1A}))$, $((2, \alpha_{m1B}), \beta_{m1B}, (0, \beta_{m1B}))$, $((0, \beta_{m1A}), \alpha_{m1A}, (1, \alpha_{m1A}))$, $((0, \beta_{m1A}), \alpha_{m1B}, (2, \alpha_{m1B}))$, $((0, \beta_{m1B}), \alpha_{m1A}, (1, \alpha_{m1A}))$, $((0, \beta_{m1B}), \alpha_{m1B}, (2, \alpha_{m1B}))$.

As triplas ordenadas associadas ao SFC g_1 são: $(x_0, am1a, x_1)$, $(x_1, bm1a, x_2)$, $(x_0, am1b, x_3)$, $(x_3, bm1b, x_4)$, $(x_2, am1a, x_1)$, $(x_2, am1b, x_3)$, $(x_4, am1a, x_1)$,

(x4, am1b, x3). As expressões booleanas associadas aos eventos são: (am1a, NOT am1ad AND NOT CED), (am1b, NOT am1bd AND NOT CED), (bm1a, rspbm1a>0), (bm1b, rspbm1b>0). O SFC g_1 resultante é apresentado na figura 6.8.

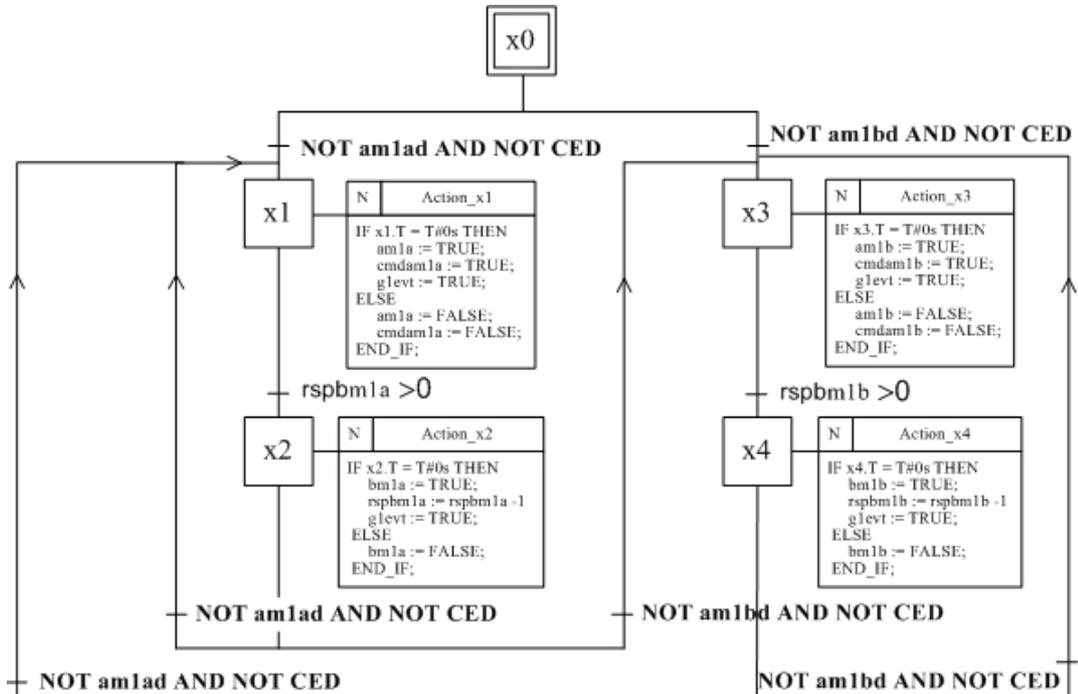


Figura 6.8: SFC g_1 da máquina M_1 .

Por razões de tamanho o SFC g_0 que corresponde ao Sistema de Transporte cujo comportamento foi modelado através do autômato G_0 (figura 4.9), não é apresentado. Tal SFC g_0 possui 30 passos e 210 transições. As figuras 6.9 e 6.10 apresentam os SFC's g_2 que corresponde ao autômato G_2 (figura 4.3b) e o SFC g_3 que correspondente ao autômato G_3 (figura 4.3c), obtidos através da aplicação dos procedimentos descritos nos exemplos 5.3 e 5.4.

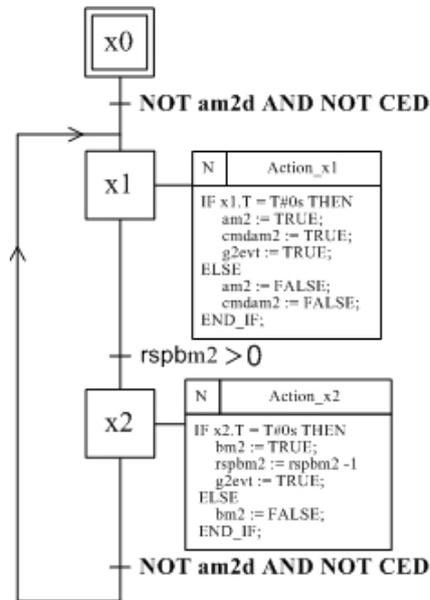


Figura 6.9: SFC g_2 da máquina M_2 .

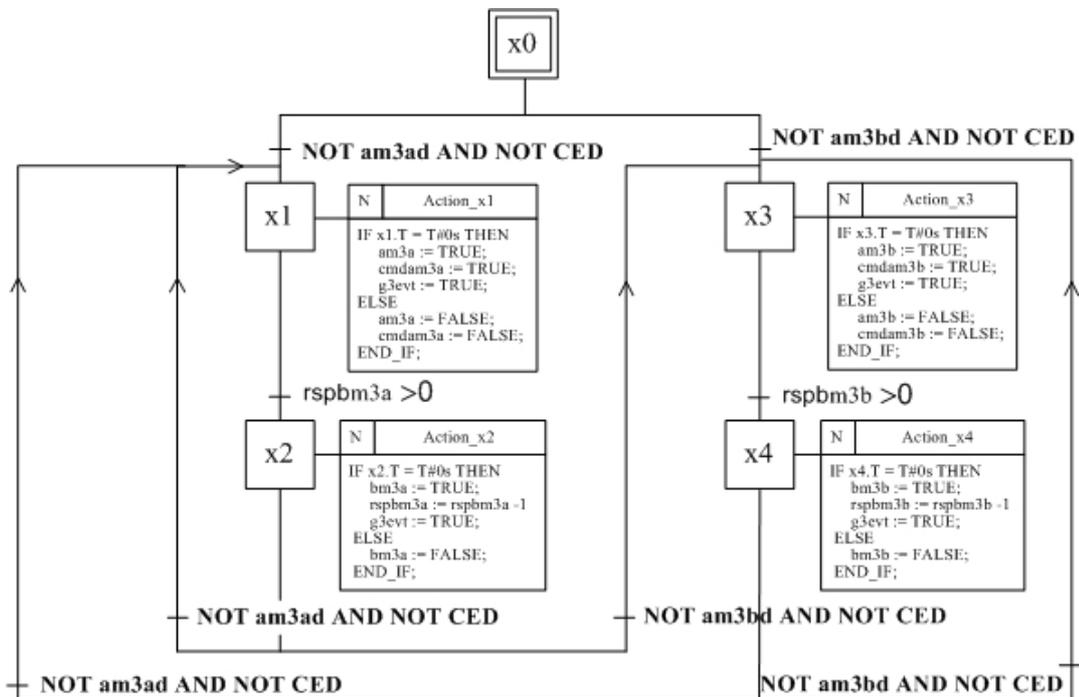


Figura 6.10: SFC g_3 da máquina M_3 .

6.4.4 Obtenção da Função lógica dg_i .

De acordo com Vieira *et al.* (2007) a definição da função lógica dg_i , $i \in I$, tem por objetivo evitar o tratamento simultâneo de eventos, em subsistemas que compartilham uma mesma célula de controle. Esta suspensão no tratamento de eventos deve ocorrer até que todos os supervisores associadas à uma mesma planta local sejam atualizados. Sendo assim define-se o conjunto $\{G_k | k \in K_i, K_i \subseteq I\}$ como sendo composto por todos os subsistemas em $\{G_i | i \in I\}$ que compartilham alguma célula de controle com o subsistema G_i . Na seção de código do FB dg_i , $i \in I$, a variável do CLP g_i deve ser ativada quando alguma variável do CLP em $\{g_k evt | k \in K_i\}$ tem valor lógico VERDADEIRO caso contrário assumirá valor FALSO.

Exemplo 5.5

No caso específico, ao observar a tabela 5.3, tem-se a planta local identificada como G_{loc4} , em que os subsistemas G_0 , G_1 e G_4 compartilham uma mesma célula de controle e a planta local G_{loc1} em que G_1 compartilha uma célula de controle com G_0 . Sendo assim pode-se obter uma função lógica $dg1$ de acordo com os subsistemas que constituem uma célula de controle com G_1 que são $\{G_0, G_1, G_4\}$. O conjunto K_1 resulta em $K_1 = \{0, 1, 4\}$. As associações das variáveis do CLP aos referidos subsistemas resultam em $(G_0, g0evt)$, $(G_1, g1evt)$ e $(G_4, g4evt)$. A figura 6.11 representa esta função em diagrama ladder.

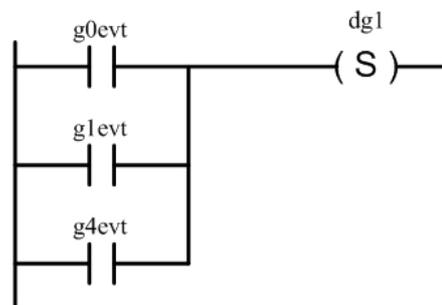


Figura 6.11: Função lógica $dg1$.

As figuras 6.12 a 6.15 representam as funções lógicas dg2, dg3, dg4 e dg0 respectivamente.

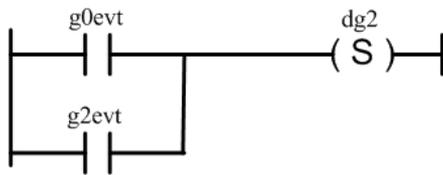


Figura 6.12: Função lógica dg2.

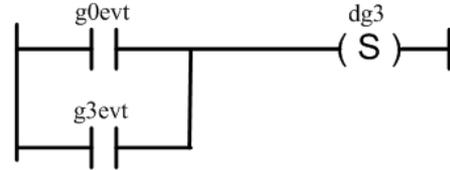


Figura 6.13: Função lógica dg3.

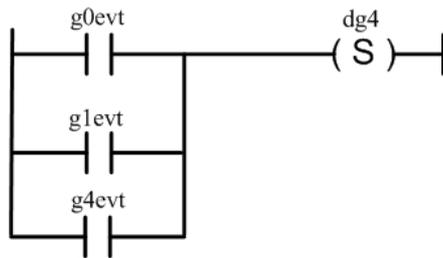


Figura 6.14: Função lógica dg4.

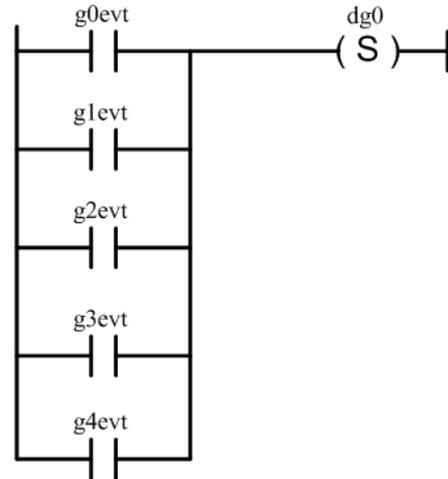


Figura 6.15: Função lógica dg0.

6.5 Obtenção dos SFC's $\{s_j | j \in J\}$ correspondentes aos Supervisores Reduzidos.

Esta seção apresenta os procedimentos sistemáticos para a conversão de cada supervisor no conjunto $\{S_j | j \in J\}$ no correspondente SFC s_j . Desenvolve-se primeiramente os conceitos genéricos para obtenção dos conjuntos de variáveis e FB's correspondentes aos SFC's. Ao final de cada subseção ilustra-se com um exemplo a aplicação dos procedimentos. São apresentados todos os SFC's s_j obtidos para a realização do processo de implementação.

6.5.1 Definição das variáveis do CLP

Sendo a ação de controle dos supervisores modulares baseada em seus estados ativos e nas respectivas funções de desabilitação de eventos controláveis Φ_j , Vieira *et al.* (2007) definem $\Sigma_p^{S_j} = \cup_{q \in QS_j} \Phi_j(q)$ como sendo o conjunto de eventos passíveis de serem desabilitados pelo supervisor S_j , $j \in J$. A cada evento $\sigma \in \Sigma_p^{S_j}$ a ser desabilitado pelo conjunto de supervisores é associada uma variável do CLP " σds_j " $\in P_j$, $j \in J$. Este conjunto de variáveis do CLP P_j é tal que $|P_j| = |\Sigma_p^{S_j}|$.

A variável σds_j é ativada quando o supervisor S_j desabilita a ocorrência do evento σ . Deste modo um conjunto de variáveis " σds_j " $\in P_j$, ativam as variáveis " σd " $\in D_i$ do CLP associadas à desabilitação dos eventos, as quais fazem parte do conjuntos $\{D_i | i \in I\}$. Se pelo menos uma variável $\sigma ds_j \in P_j$, $j \in J$ tem valor lógico VERDADEIRO a variável do CLP $\sigma d \in D_i$, $i \in I$, associada a cada evento $\sigma \in \Sigma_c^{G_i}$, também deverá ser ativada para VERDADEIRO. No caso contrário, se nenhuma variável $\sigma ds_j \in P_j$, $j \in J$ assumir valor lógico VERDADEIRO, a variável $\sigma d \in D_i$, $i \in I$, deverá ser mantida em valor FALSO.

Exemplo 5.6

No caso específico representa-se a função de desabilitação de evento controlável referente ao evento *am1ad*. Este evento está sujeito à uma ação de controle dos supervisores S_1 , S_2 , S_7 , S_8 e S_{11} quando se observa a tabela 5.4. Sendo assim é possível estabelecer uma associação do referido evento com o respectivo supervisor e a variável do CLP como segue:

- Supervisor S_1 : (α_{m1A} , am1ads1);
- Supervisor S_2 : (α_{m1A} , am1ads2);
- Supervisor S_7 : (α_{m1A} , am1ads7);
- Supervisor S_8 : (α_{m1A} , am1ads8);
- Supervisor S_{11} : (α_{m1A} , am1ads11).

A figura 6.16 representa a função de desabilitação *amlad* através de diagramas ladder.

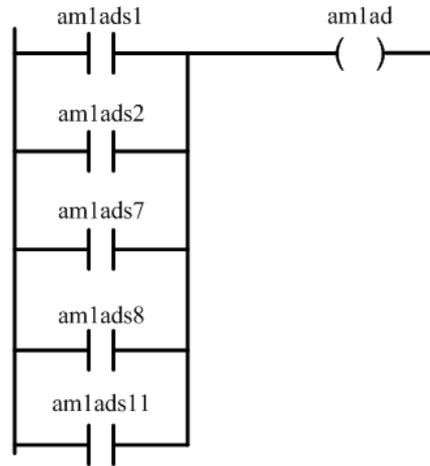


Figura 6.16: Função de desabilitação de evento controlável *amlad*.

6.5.2 Obtenção dos SFC $s_j = (X^{s_j}, T^{s_j}, x0^{s_j})$, $j \in J$

Considere o conjunto de supervisores modulares reduzidos locais $S_j = (S_j, \Phi_j)$, $j \in J$, com $S_j = (\Sigma^{S_j}, Q^{S_j}, \delta^{S_j}, q0^{S_j}, Q_m^{S_j})$ e $\Phi_j: Q^{S_j} \rightarrow 2^{\Sigma^c}$. De acordo com Vieira *et al.* (2007) $\Sigma q, q^{s_j} = \{\sigma \in \Sigma \mid (q' = \delta^{S_j}(q, \sigma)) \wedge (q \neq q')\}$ é definido como o conjunto de eventos cuja ocorrência causa a transição do estado q para um estado distinto q' do autômato S_j .

A obtenção dos SFCs s_j , $j \in J$, com $T^{s_j} = T^{rs_j} \cup T^{ns_j}$, de acordo com Vieira *et al.* (2007), é realizada desenvolvendo-se os itens (i) até (iii) abaixo.

- i) Um passo $xq \in X^{s_j}$ é associado à cada estado $q \in Q_j^S$, de modo que $|X^{s_j}| = |Q^{S_j}|$;
- ii) $x0^{s_j}$ é o passo associado à $q0^{S_j}$;
- iii) Uma transição $(xq, xq') \in T_j^{rs}$ é associada a cada par ordenado $(q, q') \in \{Q^{S_j} \times Q^{S_j}\}$ quando $\Sigma q, q^{rs_j} \neq \emptyset$, onde $xq, xq' \in X^{s_j}$ estão respectivamente associados à q, q' . Esta associação tem de ser feita de modo que $|T^{rs_j}|$ seja igual ao número de par ordenado com $\Sigma q, q^{rs_j} \neq \emptyset$;

A condição de transição associada com a transição no conjunto $T^{s_j}, j \in J$ está na expressão (i) abaixo. Esta expressão assume que $\Sigma q, q^{s_j} = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$ onde (q, q') é o par ordenado que origina tal transição. Assume-se também que os eventos $\sigma_1, \sigma_2, \dots, \sigma_n$ estão respectivamente associados às seguintes variáveis do CLP $\sigma_1, \sigma_2, \dots, \sigma_n$.

i) $(\sigma_1 \text{ OR } \sigma_2 \text{ OR } \dots \text{ OR } \sigma_n)$;

A ação associada à $xq \in X^{s_j}$ é baseada na suposição que:

i) $xq \in X^{s_j}$ está associada a $q \in Q^{s_j}$;

ii) $\Phi_j(q) = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$;

iii) os eventos $\sigma_1, \sigma_2, \dots, \sigma_n$ estão respectivamente associados às variáveis do CLP $\sigma_1 ds_j, \sigma_2 ds_j, \dots, \sigma_n ds_j$.

Exemplo 5.7

Considere as ações de controle impostas pelo supervisor S_0 , conforme a tabela 5.4, $\Phi_0(0) = \{\alpha_{01}, \alpha_{10}, \alpha_{30}, \alpha_{50}, \alpha_{70}\}$; $\Phi_0(1) = \{\alpha_{es}\}$. O conjunto de eventos $\Sigma_p^{S_0} = \Phi_0(0) \cup \Phi_0(1) = \{\alpha_{01}, \alpha_{10}, \alpha_{30}, \alpha_{50}, \alpha_{70}, \alpha_{es}\}$. A associação dos eventos com as variáveis do CLP resulta em: $(\alpha_{01}, a_{01} ds_0), (\alpha_{10}, a_{10} ds_0), (\alpha_{30}, a_{30} ds_0), (\alpha_{50}, a_{50} ds_0), (\alpha_{70}, a_{70} ds_0), (\alpha_{es}, a_{es} ds_0)$. O conjunto de variáveis do CLP P_j tal que $|P_j| = |\Sigma_p^{S_j}|$ resulta em: $P_0 = \{a_{01} ds_0, a_{10} ds_0, a_{30} ds_0, a_{50} ds_0, a_{70} ds_0, a_{es} ds_0\}$.

Os eventos associados às transições de estado do supervisor S_0 apresentado na Figura 5.9a são: $\Sigma_{0,1}^{S_0} = \{\beta pA, \beta pB\}$, $\Sigma_{1,0}^{S_0} = \{\alpha_{01}\}$. As associações dos eventos com as variáveis do CLP são: $(\beta pA, bpa), (\beta pB, bpb), (\alpha_{01}, a_{01})$. A associação dos estados de S_0 com os passos do SFC s_0 é como segue: $(0, x_0), (1, x_1)$. A figura 6.17 apresenta o SFC s_0 correspondente ao supervisor S_0 .

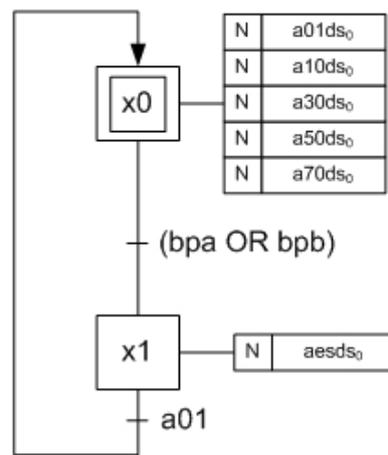


Figura 6.17: SFC s_0

As figuras 6.18 a 6.28 representam, respectivamente, os SFC s_1 ao SFC s_{11} .

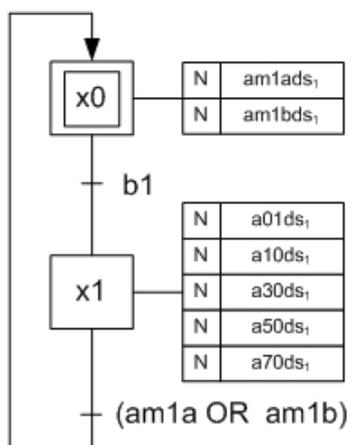


Figura 6.18: SFC s_1

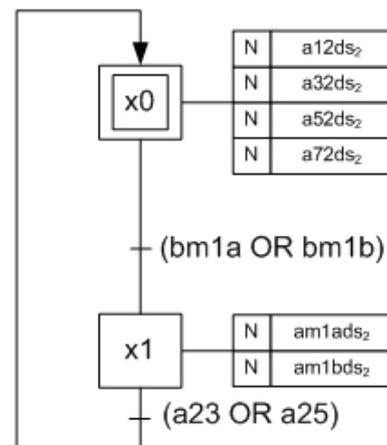
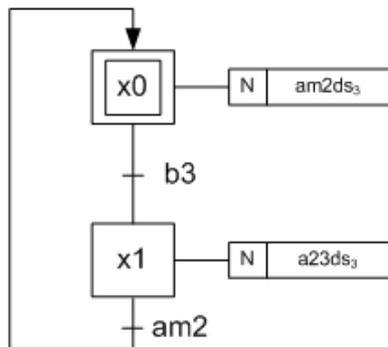
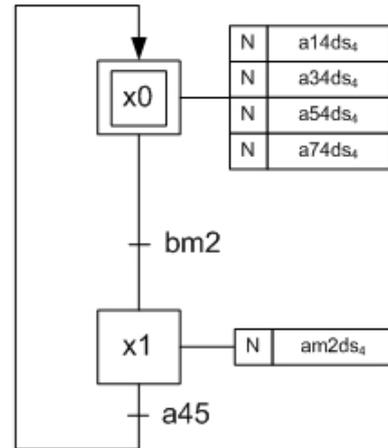
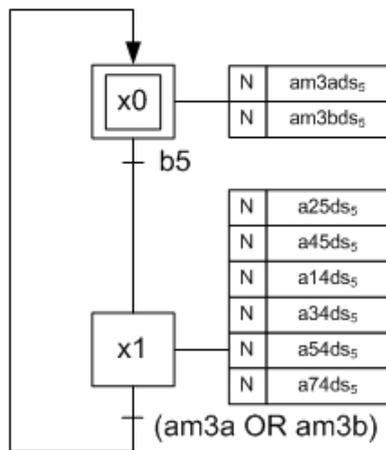
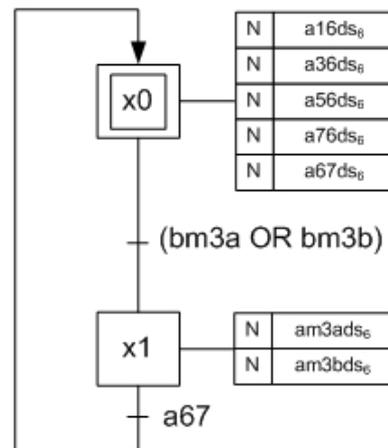
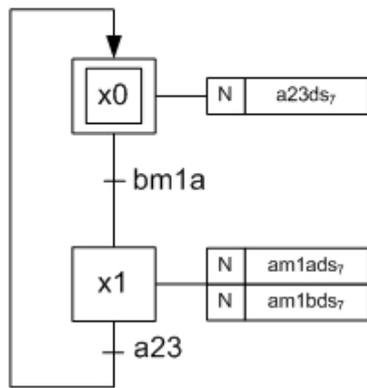
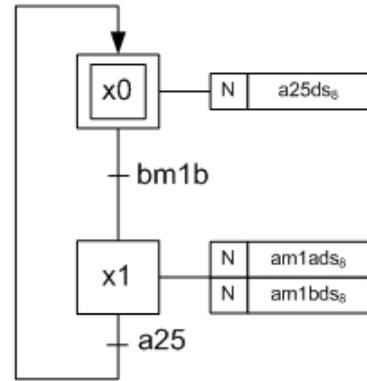
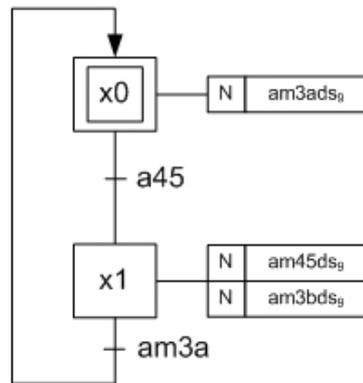


Figura 6.19: SFC s_2

Figura 6.20: SFC s_3 Figura 6.21: SFC s_4 Figura 6.22: SFC s_5 Figura 6.23: SFC s_6

Figura 6.24: SFC s_7 Figura 6.25: SFC s_8 Figura 6.26: SFC s_9

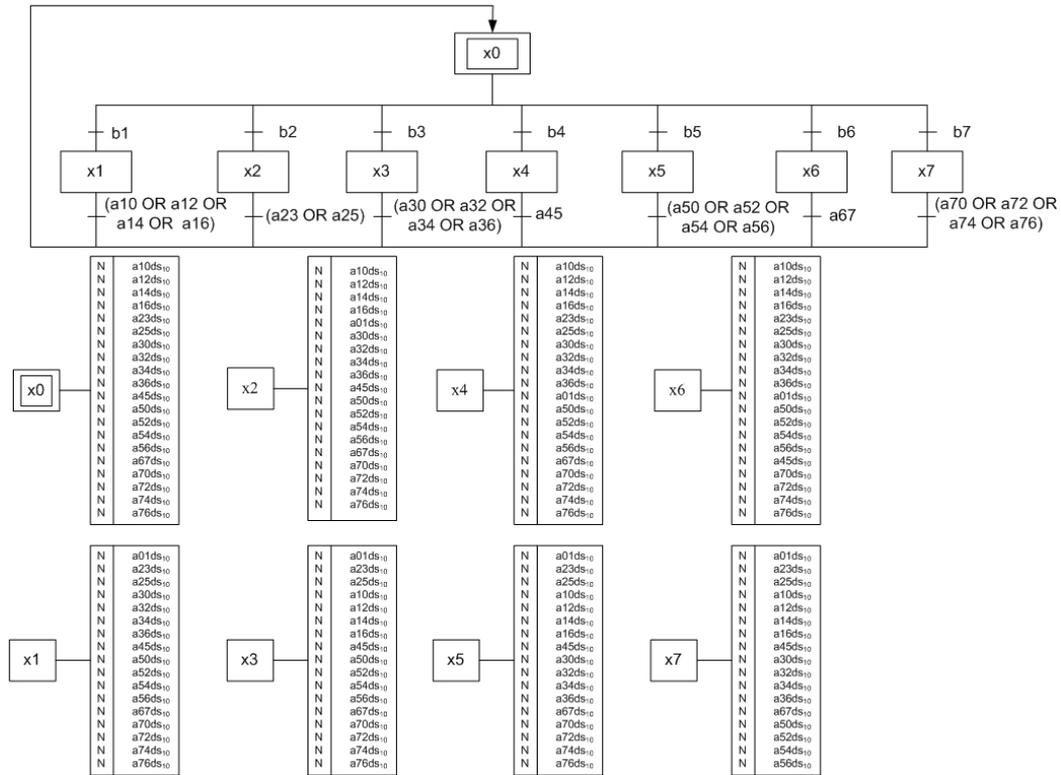


Figura 6.27: SFC s_{10}

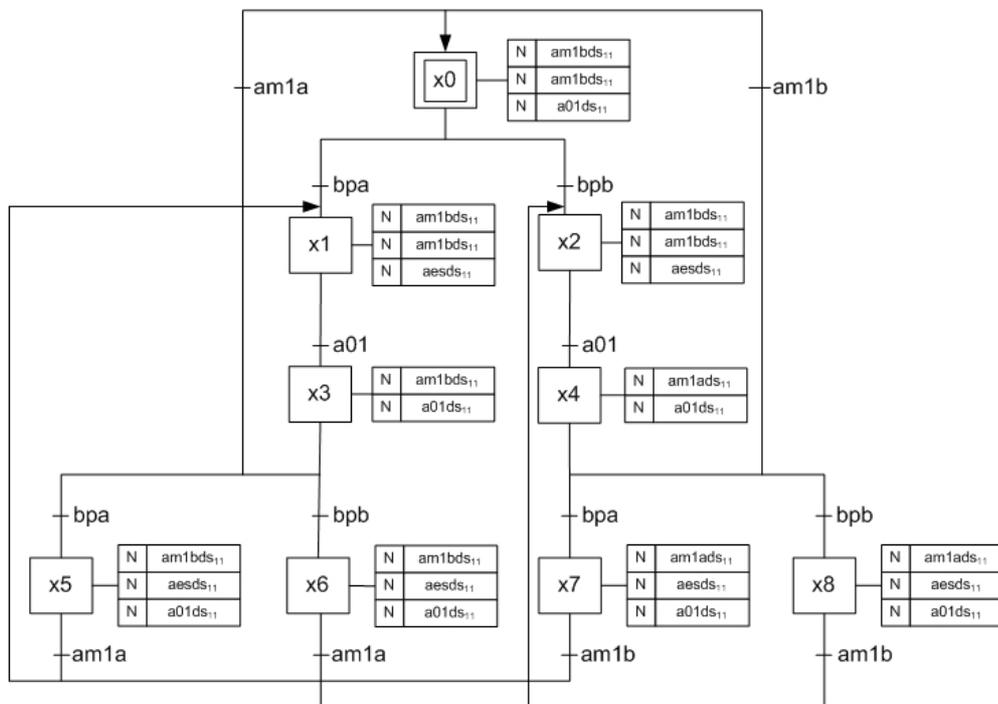


Figura 6.28: SFC s_{11}

6.6 Obtenção dos SFC's $\{o_\sigma | \sigma \in \Sigma\}$ correspondentes aos Procedimentos Operacionais

Esta seção apresenta os modelos dos SFC's $\{o_\sigma | \sigma \in \Sigma\}$ obtidos a partir dos procedimentos sugeridos por Vieira *et al.* (2007). Esta etapa do processo de modelagem resulta em particularidades que são características de cada processo, como por exemplo, as operações desenvolvidas em cada máquina e forma como estas operações se desenvolvem.

A figura 6.29 ilustra o sistema de acionamento das máquinas M_t , $t = \{1, 2, 3\}$, que neste caso tratado foram realizadas através de cilindros de dupla ação comandados por uma válvula 5/2 simples solenóide (Y_t).

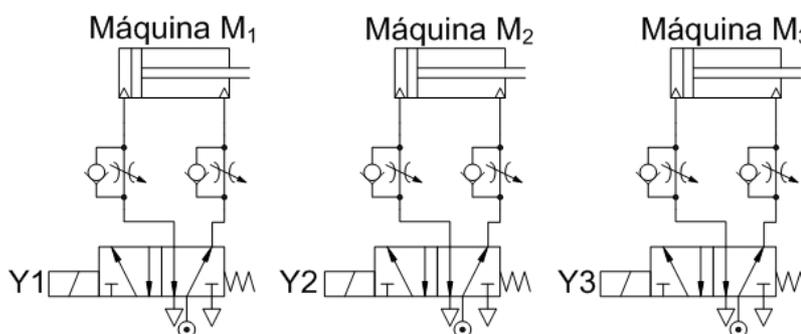


Figura 6.29: Simulação das máquinas M_t .

Conforme definido no capítulo 4, seção 4.2 na figura 4.3, o conjunto de eventos controláveis pertinentes a cada subsistema é: $\Sigma_c^{G_1} = \{\alpha_{m1A}, \alpha_{m1B}\}$, $\Sigma_c^{G_2} = \{\alpha_{m2}\}$, $\Sigma_c^{G_3} = \{\alpha_{m3A}, \alpha_{m3B}\}$. Os comandos associados a tais eventos, que correspondem às variáveis do CLP, são: $C_1 = \{\text{cmdam1a}, \text{cmdam1b}\}$, $C_2 = \{\text{cmdam2}\}$, $C_3 = \{\text{cmdam3a}, \text{cmdam3b}\}$. O conjunto de procedimentos operacionais associados à tais eventos são: $O = \{\text{om1}, \text{om2}, \text{om3}\}$. Os SFC's correspondentes à estes procedimentos são apresentados nas figura 6.30 a 6.32.

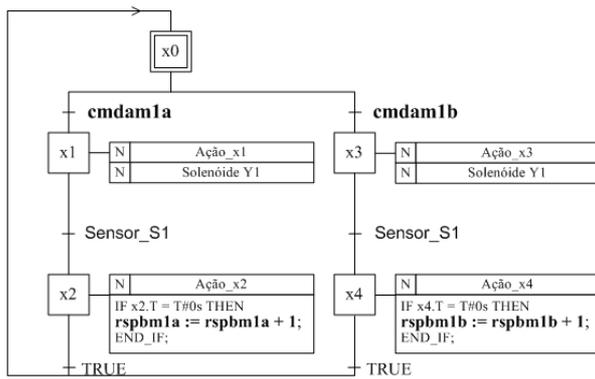


Figura 6.30: SFC om_1

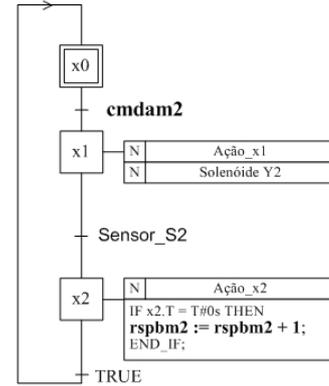


Figura 6.31: SFC om_2

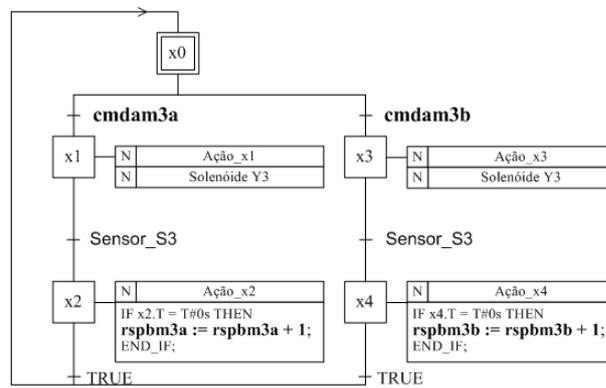


Figura 6.32: SFC om_3

O procedimento operacional associado ao evento controlável constituente do subsistema G_4 (esteira transportadora de entrada), $\Sigma_c^{G_4} = \{\alpha_{es}\}$, é apresentado na figura 6.33. O conjunto que constitui a variável do CLP associada à este evento é $C_4 = \{cmdaes\}$. O SFC oes representa o procedimento operacional associado.

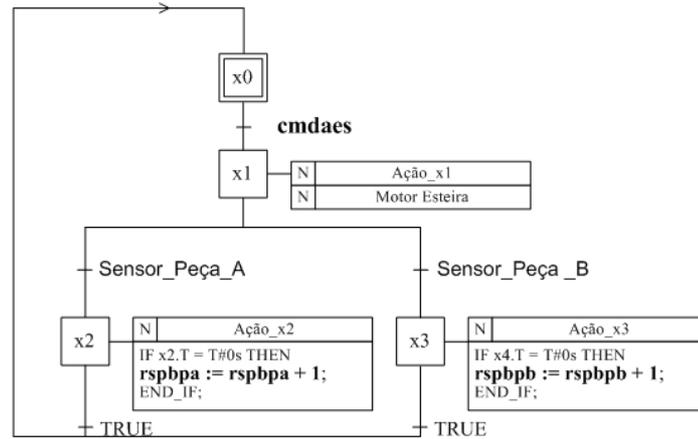


Figura 6.33: SFC *oes*.

Os procedimentos operacionais associados ao sistema de transporte foram realizados de modo a obter um SFC para cada evento controlável α_{xy} considerado no desenvolvimento do plano de processo. Os eventos controláveis que constitui tal plano de processo é conforme exposto ao final da seção 4.2 do capítulo 4. Tal conjunto é $\Sigma_c^{G_0} = \{ \alpha_{01}, \alpha_{10}, \alpha_{30}, \alpha_{50}, \alpha_{70}, \alpha_{12}, \alpha_{32}, \alpha_{52}, \alpha_{72}, \alpha_{14}, \alpha_{34}, \alpha_{54}, \alpha_{74}, \alpha_{16}, \alpha_{36}, \alpha_{56}, \alpha_{76}, \alpha_{23}, \alpha_{25}, \alpha_{45}, \alpha_{67} \}$.

O conjunto que relaciona as variáveis do CLP associadas a tais eventos é $C_{\theta} = \{ \text{cmda01}, \text{cmda10}, \text{cmda30}, \text{cmda50}, \text{cmda70}, \text{cmda12}, \text{cmda32}, \text{cmda52}, \text{cmda72}, \text{cmda16}, \text{cmda36}, \text{cmda56}, \text{cmda76}, \text{cmda23}, \text{cmda25}, \text{cmda45}, \text{cmda67} \}$. Sendo assim, o conjunto de procedimentos operacionais disparados por tais comandos é $O = \{ \text{oa01}, \text{oa10}, \text{oa30}, \text{oa50}, \text{oa70}, \text{oa12}, \text{oa32}, \text{oa52}, \text{oa72}, \text{oa14}, \text{oa34}, \text{oa54}, \text{oa74}, \text{oa23}, \text{oa25}, \text{oa45}, \text{oa67} \}$.

A figura 6.34 representa o procedimento operacional, SFC *oa23*, associado ao evento controlável α_{23} . Neste caso a ocorrência do comando (*cmda23*) dispara tal procedimento, no qual o passo x1 determina a ativação das entradas do controlador do robô para a realização do referido movimento.

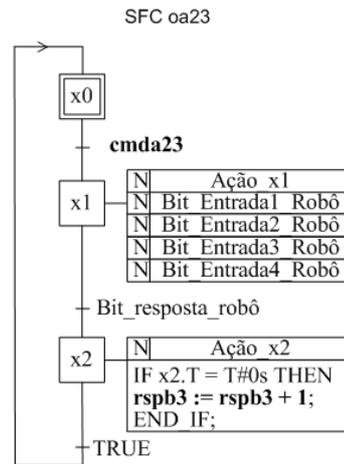


Figura 6.34: SFC *oa23*.

A condição de transição associada ao evento `Bit_resposta_robô`, que determina a evolução do SFC *oa23* do passo `x1` para o passo `x2`, sinaliza a chegada do robô ao *buffer* B_3 . Esta transição assumindo valor lógico VERDADEIRO, incrementa a resposta associada ao evento não-controlável β_3 (`rspb3`) de uma unidade.

6.7 Conclusão ao capítulo

A utilização de um modelo para implementação de uma arquitetura de controle supervisorio em CLP conforme proposto por Vieira *et al.* (2007) auxilia de forma importante a atingir os objetivos para a obtenção de um sistema de controle com lógica re-configurável.

Com a fundamentação do modelo Vieira *et al.* (2007) de acordo com a Teoria de Controle Supervisorio (Ramadge e Wonham (1987), seguindo a abordagem Modular Local (Queiroz e Cury, 2000) e a Arquitetura de Controle Supervisorio (Queiroz e Cury, 2002), que contempla a modularidade física dos sistemas de manufatura, consegue-se estabelecer a sistematização de procedimentos para as atividades de projeto e re-projetos em sistemas multi-produtos. Sendo assim, qualquer alteração referente à reconfiguração do sistema físico ou mudança nos roteiros dos produtos, não reflete de forma significativa na estrutura de controle. As alterações neste caso são realizadas de forma localizada e isto facilita os propósitos de elaboração, manutenção e detecção de erros na lógica de controle, diminuindo o tempo necessário para a realização de tais tarefas.

Importante relatar que a inclusão de novos roteiros, associados à colocação de novos *buffers* tem um impacto importante não na complexidade da lógica de controle, mas sim na complexidade computacional e dimensão das estruturas associadas ao sistema de transporte como o módulo do sistema produto, por exemplo. A inclusão de dois *buffers* ao processo, resultou num aumento de 50% no número de estados do autômato H_0 referente ao sistema de transporte, e num aumento de 102% das transições associadas à este mesmo autômato. Isto resulta em um SFC de 30 passos e 210 transições para o sistema de transporte. Considerando o fato de que um SFC utiliza mais recursos de memórias dos CLP's do que os tradicionais diagramas Ladder, é um fator a ser considerado quando da implementação. Tal observação não inviabiliza a utilização deste modelo uma vez que existem mecanismos para a conversão dos SFC's em diagramas Ladder.

O modelo proposto por Queiroz e Cury (2002) e Vieira *et al.* (2007) em conjunto com ferramenta computacional TCT (Wonham, 1999), se apresentam como uma importante alternativa de projeto nos casos em que exigem flexibilidade de reconfiguração dos sistemas diante de mudanças nos roteiros de produção. Mesmo com as facilidades proporcionadas com a utilização destas ferramentas formais, um problema ainda persiste em função do modo síncrono de operação dos CLP's: Tal problema diz respeito à escolha do evento conforme relatado por Fabian e Hellgren (1998) sendo citado na seção de introdução deste capítulo.

De acordo com Fabian e Hellgren (1998) um comportamento da planta com a maior liberdade possível é conseguido através de um supervisor *minimamente restritivo*. A planta decidirá que caminho escolher dentre as alternativas que a estrutura do supervisor oferecer. Contudo, quando o supervisor gera algumas mudanças de sinal, ele pode ter de escolher entre estas alternativas num único estado. Dependendo da ordem de implementação, internamente no CLP, diferentes escolhas serão feitas e para cada estado com diferentes alternativas de eventos para a ocorrência, a escolha será sempre a mesma. Esta escolha deverá ser feita pelo programador, ou de outro modo pela execução seqüencial do programa pelo CLP. No modelo Vieira *et al.* (2007) a razão para tal problema reside na ordem em que o FB referente ao nível Sistema Produto chama cada FB do conjunto $\{g_i \mid i \in I\}$.

Capítulo 7

Conclusões

Os Sistemas Flexíveis de Manufatura Multi-produtos compreendem um elevado número de máquinas controladas por computador e dispositivos de manuseio e transporte de materiais, que se interagem com o objetivo de produzir diferentes peças ou produtos com pouco ou nenhum *setup*. Os sistemas automáticos de manuseio e transporte de materiais são uma das principais áreas objetos de pesquisa, no sentido de melhorar o desempenho de tais sistemas (Sivagnanavelu, 2000).

Alguns dos diferentes tipos de flexibilidades presentes nos sistemas de manufatura como flexibilidade de operações, flexibilidade sequenciamento e flexibilidade de processamento dependem do bom desempenho destes sistemas e conferem aos mesmos uma complexidade maior. Esta complexidade diz respeito à racionalização do manuseio e transporte de material em processo entre as células de manufatura (Kochikar e Narendran, 1998).

O objetivo deste trabalho é aplicar a Teoria de Controle Supervisório (Ramadge e Wonham, 1987) utilizando a abordagem Modular Local (Queiroz e Cury, 2000) na modelagem da lógica de controle para os sistemas flexíveis de manufatura multi-produtos. De acordo com Wadhwa *et al.* (2005), tais sistemas têm sido caracterizados por um gargalo de produção que nos últimos anos tem se deslocado para as áreas de inteligência dos processos que compreendem os seus sistemas de controle. Algumas das atividades a serem desenvolvidas exigem uma complexidade maior na coordenação entre os subsistemas, como por exemplo, os sistemas de transporte que suprem as estações de processamento. A grande variedade de movimentos exigidos na realização dos processos e as diferentes formas de execução considerando por exemplo atuadores pneumáticos ou robôs, fez deste tipo de subsistema o objetivo maior deste estudo.

As atuais técnicas heurísticas de concepção da lógica de controle para este tipo de sistema podem oferecer um esforço maior na sua elaboração, além disto, não se tem a garantia de que todos os aspectos para o bom funcionamento serão atendidos. A utilização de uma abordagem metodológica fundamentada em técnicas formais e ferramentas computacionais de síntese contribui de forma importante para o desenvolvimento do projeto da lógica de controle. Deste modo os processos de re-projetos que envolvem alteração da lógica, detecção de erros e implementação são contemplados de forma positiva pela estruturação advinda da adoção de um formalismo como a Teoria de Controle Supervisório para Sistemas a Eventos Discretos (Ramadge e Wonham, 1987).

A utilização da abordagem Modular Local (Queiroz e Cury, 2000) da Teoria de Controle Supervisório com a Arquitetura de Controle Supervisório proposta por Queiroz e Cury (2002) e o Modelo para Implementação em CLPs (Vieira *et al.* 2007) oferecem a estruturação requerida para os modernos Sistemas Flexíveis de Manufatura que operam com diferentes tipos de produtos. As constantes alterações presentes nestes sistemas, tanto nas operações a serem desenvolvidas pelas máquinas como pelos roteiros de produção associados aos produtos, foram amplamente favorecidas através da adoção desta abordagem metodológica.

Este favorecimento é advindo da sistematização dos procedimentos que caracterizou cada uma das etapas de modelagem das especificações, síntese dos supervisores e implementação nos dispositivos de controle. Todo este processo de sistematização contribui para a diminuição do tempo necessário ao desenvolvimento de cada projeto lógico. Por exemplo, modificações que dizem respeito à inserção de uma nova operação a ser desenvolvida por uma máquina qualquer, ou a inclusão de uma nova máquina no processo, se realizam de forma simples através da adição dos novos elementos nos níveis Sistema Produto e Procedimentos Operacionais, não sendo necessária a alteração das estruturas anteriormente concebidas.

A razão para o aumento de facilidades no processo de re-configuração da lógica de controle é também devida à modularidade das especificações de controle proposta por Queiroz e Cury (2000). Esta modularidade foi transferida de modo eficaz para o nível Supervisores Modulares. Em casos de alteração nas regras de controle, grande parte das estruturas já concebidas na lógica de controle pode ser novamente reutilizada, uma vez que as alterações estão restritas à partes específicas do programa. Sendo assim, a organização da

Arquitetura de Controle em três níveis possibilita uma flexibilidade maior na elaboração da lógica de controle, pois permite que as alterações sejam realizadas de forma rápida e pontual.

O modelo de Arquitetura de Controle Supervisório proposto por Queiroz e Cury (2002), possibilita que o trabalho de modelagem dos subsistemas e especificações de controle e síntese dos supervisores, seja realizado através de modelos com um menor número de estados observando apenas os eventos de maior relevância. Isto diminui a complexidade computacional, uma vez que o detalhamento das atividades intermediárias será realizado via os procedimentos operacionais. Tal fato propicia evitar a explosão exponencial do número de estados, o que pode inviabilizar os processos de síntese dos supervisores e traz dificuldades adicionais para os propósitos de implementação. A obtenção de supervisores reduzidos facilita a implementação em CLP's e auxilia na interpretação da regras de controle, facilitando alterações futuras.

O Modelo para Implementação em CLPs de Vieira *et al.* (2007) mostrou-se bastante flexível para as tarefas de alteração da lógica de controle. Este modelo resolve alguns dos problemas citados por Fabian e Hellgren (1998), como a simultaneidade de sinais e o efeito avalanche, que surgem durante os processos de implementação quando a linguagem de programação utilizada era os diagramas escadas.

Como principal objetivo atingido tem-se a obtenção de modelos que representam o comportamento de uma classe de sistema de transporte utilizado nos sistemas flexíveis de manufatura. Este sistema de transporte atende aos sistemas de rotas variáveis e podem ser observados tanto na figura 2.5, assim como na figura 2.9 que foi objeto deste estudo. O processo de modelagem atende à ambas as situações, uma vez que as possibilidades de deslocamento são iguais. A única diferença reside no elemento realizador do sistema de transporte, que no primeiro caso (figura 2.5) é um atuador pneumático e no segundo caso (figura 2.9) é um robô.

Como um dos aspectos a serem resolvidos está o problema citado por Fabian e Hellgren (1998), e que diz respeito à escolha do evento que deverá ocorrer. Tal problema ainda persiste e depende da ordem de implementação do programa. O modo síncrono como os CLP's operam determina que um evento tenha prioridade com relação à outros que estejam habilitados em um determinado estado do sistema. Este aspecto não é propriamente um problema da metodologia adotada, e sim da natureza síncrona de atualização de dados, que envolve todo dispositivo de controle.

No desenvolvimento deste trabalho, uma limitação encontrada com a adoção desta metodologia ainda diz respeito à explosão exponencial de estados. Este é um aspecto relevante para o sucesso dos objetivos finais a serem atingidos. Fatores como o número de máquinas utilizadas, número de operações a serem desenvolvidas pelas máquinas e tipos de peças a serem processadas, influenciam de forma direta na escolha do sistema a ser modelado.

Por exemplo, num primeiro caso considerando apenas 6 *buffers* no sistema, o número de rotas relevantes era de apenas 13. Ao elevar o número de *buffers* do sistema para 8, o número de rotas relevantes aumentou para 21. Sendo assim, o SFC que representa o modelo do sistema de transporte saltou de 20 passos para 30 passos, e o número de transições subiu de 104 para 210 transições.

Ou seja, a adição de dois novos *buffers*, o que corresponde à apenas uma máquina, aumentou de forma considerável o tamanho do modelo do sistema de transporte. A adoção de mais um tipo de peça a ser processada, como por exemplo uma peça Tipo C, implicaria na adição de novos estados aos modelos das máquinas, aumentando ainda mais a complexidade computacional. A operação de teste de modularidade dos supervisores envolve um número elevado de estados. Trabalhar com modelos maiores pode comprometer o processo de verificação do teste.

Este elevado espaço de estados, característico dos processos de manufatura, continua a ser o objeto de limitação da aplicação da Teoria de Controle Supervisório ao contexto industrial. Este aspecto compromete o fator tecnológico envolvido na utilização da metodologia proposta, uma vez que os recursos de *hardware* devem ser robustos tanto nas fases de síntese como de implementação. Na fase de síntese dos supervisores são necessários microcomputadores robustos face à operação de modularidade, que é da ordem de 2×10^{19} estados. Na fase de implementação, exige-se que se trabalhe com um CLP mais robusto, uma vez que a programação em SFC solicita pelo menos 3 vezes mais memória para o processamento do programa.

Tal obstáculo será resolvido com o avanço tecnológico dos próximos anos, e a abordagem proposta por Ramadge e Wonham (1987), Queiroz e Cury (2002) e Vieira *et al.* (2007) torna-se mais uma importante ferramenta de projeto lógico a disposição dos projetistas de Controle de Sistemas a Eventos Discretos, uma vez que os sistemas têm operado com sucesso sob tal supervisão.

Como perspectivas para trabalhos futuros, sugere-se a modelagem envolvendo um maior número de produtos, que por consequência solicita mais máquinas e operações, aumentando a quantidade de roteamentos. No sentido de aumento da complexidade observam-se os sistemas de transporte cooperativos em que, por exemplo, dois robôs podem realizar os trabalhos de movimentação de peças entre células compartilhando uma mesma zona de operação. Pode-se ainda, realizar a distribuição de toda a estrutura de controle supervisorio, que no presente trabalho foi implementada de forma concentrada em um único CLP. Isto possibilitaria trabalhar com CLP's menos robustos, diminuindo os custos de implementação.

Referências Bibliográficas

- BENJAAFAR, S.; RAMKRISHNAN, R.; 1996 Modeling, measurement and evaluation of sequencing flexibility in manufacturing systems. Department of Mechanical Engineering , University of Minnesota. *International Journal of Production Research*,.
- BENJAAFAR, S.; 1995 Process Planning Flexibility: Models, Measurement, and Evaluation, Working Paper. *Department of Mechanical Engineering* , University of Minnesota.
- BRANDIN, B.A.; CHARBONNIER, F.E.; 1994. The supervisory control of the automated manufacturing system of the AIP. *In: 4th International Conference on Computer Integrated Manufacturing and Automation Technology*, p. 319-324.
- BRANDIN, B. A.; MALIK, R.; MALIK, P., 2004 Incremental Verification and Synthesis of Discrete-Event Systems Guided by Counter Examples *Transactions on Control Systems Technology IEEE* Vol. 12, Nº 3, May.
- BRANDIN, B. A., 1996 The Real-Time Supervisory Control of an Experimental Manufacturing Cell *Transactions on Robotics and Automation IEEE Proc.*, vol. 12, Nº 1, February.
- CAPRIHAN, R.; WADHWA, S., Impact of Routing Flexibility on the Performance of an FMS – A Simulation Study. *The International Journal of Flexible Manufacturing Systems*. 9(1997): 273-298. Kluwer Academic Publishers, Boston.
- CARROL, J.; LONG, D. 1989 Theory of finite automata. *Prentice-Hall International Editions*.
- CASSANDRAS, G.C.; LAFORTUNE, S., 1999 Introduction to Discrete Event Systems. KLUWER Academic Publishers, Massachusetts, USA.
- CHANDRA, V.; HUANG, Z.; KUMAR, R., 2003 Automated Control Synthesis for an Assembly Line Using Discrete Event System Control Theory *Transactions on Systems, Man and Cybernetics IEEE* Vol. 33, Nº 2, May.
- DORF, R.; BISHOP, R., 2001 Sistemas de Controle Modernos *Ed. LTC 8^a ed.* Rio de Janeiro.
- FABIAN, M.; HELLGREN, A., 1998 PLC – based Implementation of Supervisory Control for Discrete Event Systems. *Proceedings of the 37^o IEEE Conference on Decision & Control*, Tampa, Flórida USA, December.
- GEROMEL, J.C; PALHARES, A.G.B, 2004. Análise Linear de Sistemas Dinâmicos. *Editores Edgard Blücher Ltda*, São Paulo

- GOUYON, D.; PETIN, J.F.; MOREL, G., 2004a Control Synthesis for Product-Driven Automation *Proceedings IFAC*.
- GOUYON, D.; PETIN, J. F.; GOUIN, A., 2004 Pragmatic approach for modular control synthesis and implementation *International Journal of Production Research* Vol.42, N° 14, July.
- GROOVER, M.P., 2001. Automation, Production Systems and Computer Integrated Manufacturing *Ed. Prentice Hall 2ª ed.* New Jersey.
- HASDEMIR, T., KURTULAN, S., GÖREN, L.; 2004. Implementation of Local ModularSupervisory Control for a pneumatic system using PLC. *In: 7th International Workshop on Discrete Event Systems*, p. 25-30.
- KOCHIKAR, V.P., NARENDRAN, T.T; Logical Cell Formation in FMS, Using Flexibility-Based Criteria. *The International Journal of Flexible Manufacturing Systems*. 10(1998): 163-182. Kluwer Academic Publishers, Boston.
- KUMAR, R.; TAKAI, S.; FABIAN, M.; USHIO, T., 2005. Maximally permissive mutually and globally nonblocking supervision with application to switching control. *AUTOMATICA* vol. 41, May.
- MIYAGI, P. E., 1996 Controle Programável: Fundamentos do Controle de Sistemas a Eventos Discretos *Ed. Edgard Blücher Ltda 1ª ed* São Paulo.
- MORAES, C.C; CASTRUCCI, P.B.L., 2001 Engenharia de Automação Industrial *Ed. LTC 1ª ed* Rio de Janeiro.
- QUEIROZ, M. H; CURY, J. E. R., 2000. Controle Supervisório Modular de Sistemas de Grande Porte. *Dissertação de mestrado em Engenharia Elétrica*. Universidade Federal de Santa Catarina.
- QUEIROZ, M. H; CURY, J. E. R., 2002. Controle Supervisório Modular de Sistemas de Manufatura *Revista Controle & Automação*, Vol. 13, N° 2, Agosto.
- QUEIROZ, M. H. DE ; CURY, J. E. R, 2002a . Controle Supervisório Modular de Sistemas de Manufatura. SBA. Sociedade Brasileira de Automática, Campinas, v. 13, n. 2, p. 123-133.
- RAMADGE, P. J.; WONHAM, W. M., 1987 Supervisory control of a class of discrete event processes *SIAM Journal on Control and Optimization* Vol. 25 N° 01.
- RAMADGE, P. J.; WONHAM, W. M., 1989. *The control of discrete event systems Proceedings of IEEE, Special Issue on Discret Event Dynamics Systems*.

- RAMIREZ-SERRANO, A.; SRISKANDARAJAH, C.; BENHABIB, B., 2000a Automated-Based Modeling and Control Synthesis for Manufacturing Workcells with Part-Routing Flexibility *Transactions on Robotics and Automation IEEE*, vol. 16, Nº 6, December.
- RAMIREZ-SERRANO, A.; SRISKANDARAJAH, C.; BENHABIB, B., 2000b Supervisory Control of Multiworkcell Manufacturing Systems with Shared Resources *Transactions on Systems, Man and Cybernetics IEEE*, vol. 30, Nº 5, October.
- RAYMOND, D.; WOOD, D., *Grail: Engineering automata in C++: Version 2.5*, 1996 <http://www.csd.uwo.ca/research/grail/> verificada em 31 de outubro de 2005.
- SANTOS, E.A.P., 2003. Contribuições ao projeto conceitual de sistemas de manipulação e montagem automatizados. *Tese de Doutorado (Engenharia Elétrica)*, Universidade Federal de Santa Catarina.
- SIVAGNANAVELU, D.G.; 2000. Dynamic Scheduling of Flexible Manufacturing Systems: A Study of Machine and Material Handling Control Strategies. *A Thesis in the Department of Mechanical Engineering Presented in Partial Fulfillment of the Requirements for the Degree of Master of Applied Science*. Concordia University – Montreal ,Quebec – Canada.
- SLACK, N.; CHAMBERS S.; HARLAND, C.; HARRISON, A.; JOHNSTON, R., 2001 *Administração da Produção Ed. Atlas 2ª ed.* São Paulo.
- TSUBONE, H.; HORIKAWA, M., A; 1999 Comparison Between Machine Flexibility and Routing Flexibility. *The International Journal of Flexible Manufacturing Systems*. 11(1999): 83-101. Kluwer Academic Publishers, Boston.
- VIEIRA, A.D., Contribuições à Implementação de Sistemas de Controle Supervisório, *Proposta de Tese de Doutorado, Programa de Pós-Graduação em Engenharia Elétrica*, UFSC, Florianópolis, Dezembro de 2004a.
- VIEIRA, A.D; CURY, J.E.R.; QUEIROZ, M.H. de; 2007. A model for PLC implementation of supervisory control of discrete event systems. *Control Engineering Practice*. submetido para publicação.
- WADHWA, S.; RAO, K. S.; CHAN, T. S; August 2005. Flexibility-enabled lead-time reduction in flexible systems *International Journal of Production Research*, vol. 43, Nº 15, 1, 3131-3162.
- WONHAM, W. M., 1999 Notes on control of discrete event systems. *Course notes for ECE 1636F/1637S. Dept. of Electrical and Computer Engineering*. University of Toronto, Canada.
- YALCIN, A., 2004. Supervisory control of automated manufacturing cells with resource failures *Robotics and Computer-Integrated Manufacturing*, Elsevier vol. 20.

ZAHNAN, I.M.; ELMAGRABY, A.S.; SHALABY, M.A., 1990. Evaluation of flexibility in Manufacturing Systems. *IEEE*.

Apêndice

Modelagem de Outras Seqüências de Processamento

Este anexo apresenta os resultados obtidos para outras seqüências de processamento durante a etapa de modelagem. Nesta etapa são observadas alternativas de rotas que podem estar presentes em alguns sistemas de manufatura. Algumas destas alternativas compreende os processos em que ocorre a divergência de peças a partir de uma estação específica para diferentes máquinas. Outra possibilidade é a convergência das peças a partir de várias máquinas em direção à uma única máquina. São apresentados os modelos dos subsistemas e as especificações de controle. Como observação comum a todo o processo de modelagem, considera-se que o estado do *buffer de saída* da última máquina de processamento não será modelado. Neste caso tais *buffers* são considerados de capacidade infinita, sendo assim não é apresentada a especificação que representa o seu estado, se cheio ou vazio. Todos os outros *buffers* do processo são considerados de capacidade unitária da mesma forma como desenvolvido durante todo este trabalho. Nos modelos do robô e das especificações para a coordenação das movimentações do robô, apenas são considerados os movimentos relevantes para o processo de manufatura. Os resultados consideram a abordagem monolítica e a abordagem modular local.

A.1 Seqüência de Processamento com Linha de Transferência

Neste caso é modelado um processo em se opera com dois diferentes tipos de produtos ou peças, sendo identificados como dos tipos A e B. O número de máquinas para processamento é igual a três, sendo identificadas como Máquina M_1 , Máquina M_2 e Máquina M_3 . Cada máquina poderá processar tanto as peças tipo A como as peças tipo B. Para cada operação será adotado um estado no modelo do autômato, que representa o comportamento das Máquinas M_i .

Supondo um plano de processo, tem-se que o roteiro de produção para as peças A e B será o mesmo sendo:

- Peça A e B: $B_0 \rightarrow B_1 \rightarrow M_1 \rightarrow B_2 \rightarrow B_3 \rightarrow M_2 \rightarrow B_4 \rightarrow B_5 \rightarrow M_3$.

O modelo do robô é como apresentado na figura A.1 e identificado como G_0 .

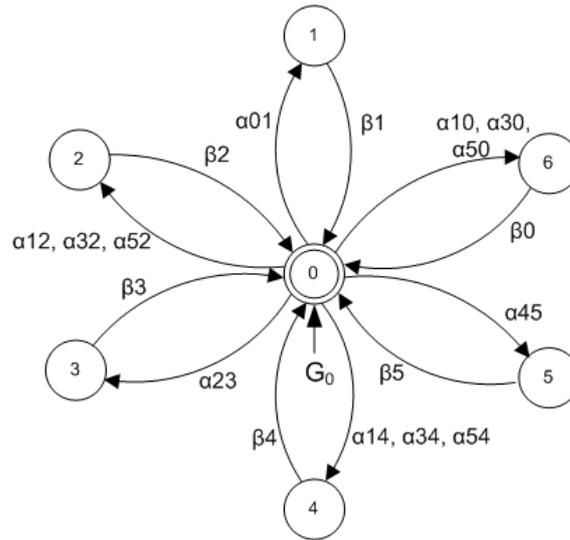


Figura A.1: Modelo do Sistema de Transporte para linha de transferência.

Os autômatos referentes aos modelos das máquinas M_t são como apresentados na figura A.2. Os autômatos G_1 , G_2 e G_3 representam respectivamente as máquinas M_1 , M_2 e M_3 .

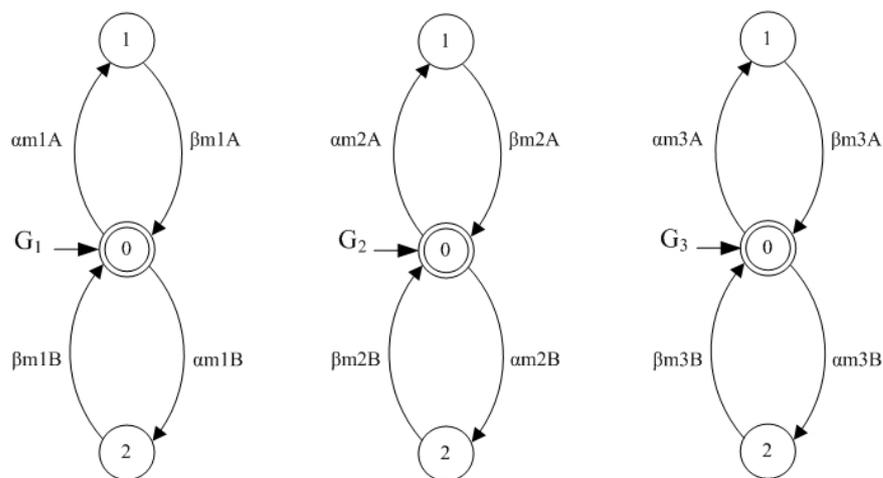


Figura A.2: Modelo das máquinas M_t , autômatos G_1 , G_2 e G_3 .

O autômato que representa o modelo da esteira transportadora de entrada é identificado como G_4 . A figura A.3 representa tal modelo.

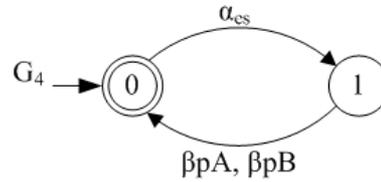


Figura A.3: Modelo da esteira transportadora de entrada G_4 .

As especificações pertencentes ao primeiro conjunto são apresentadas na figura A.4. Tais especificações determinam a coordenação do robô com o início e fim de operação das máquinas M_t , em função do estado dos *buffers* de entrada e saída das estações de processamento. O estado cheio dos *buffers*, independente do tipo de peça. As especificações E_0, E_1, E_2, E_3, E_4 e E_5 modelam o estado dos *buffers* de mesmo índice.

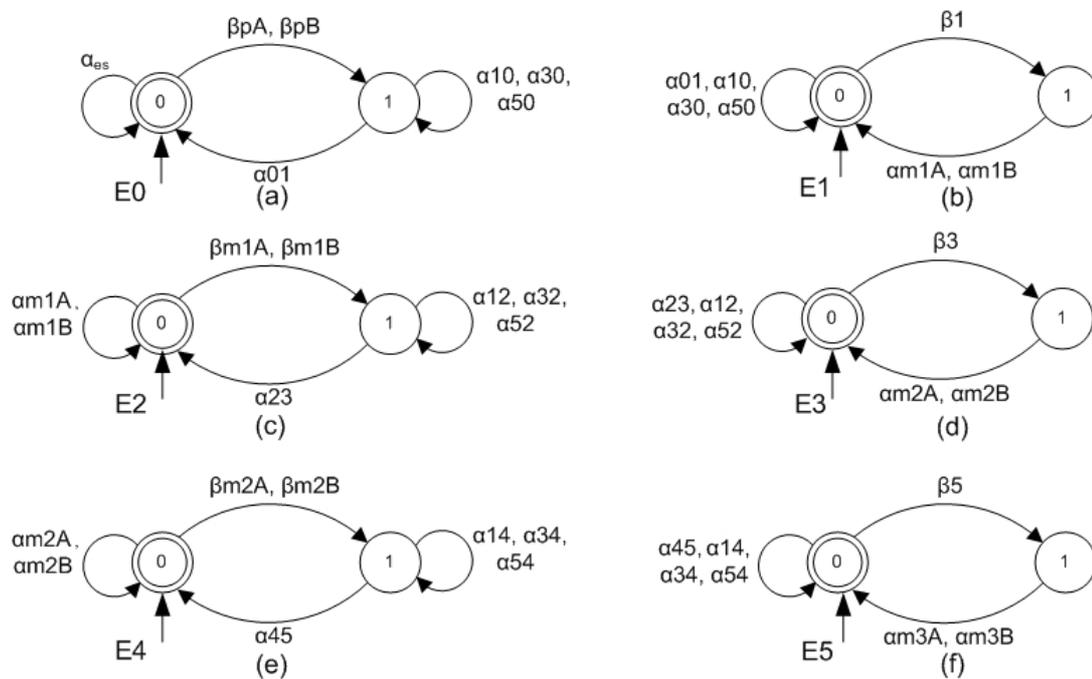


Figura A.4: Especificações dos *buffers* B_0, B_1, B_2, B_3, B_4 e B_5 .

As especificações E_6 , E_7 e E_8 coordenam as ações de três subsistemas simultaneamente. Esta coordenação depende de um tipo específico de peça e considera o estado de dois *buffers*, também simultaneamente. Estes *buffers* são os *buffers* de saída das estações fornecedoras, e de entrada das estações consumidoras. Neste caso específico, as especificações que fazem parte do segundo conjunto, são contempladas neste quarto conjunto. A figura A.5 apresenta a especificação E_6 . Tal especificação coordena as ações dos subsistemas esteira transportadora de entrada, máquina M_1 e robô considerando o estado dos *buffers* B_0 e B_1 .

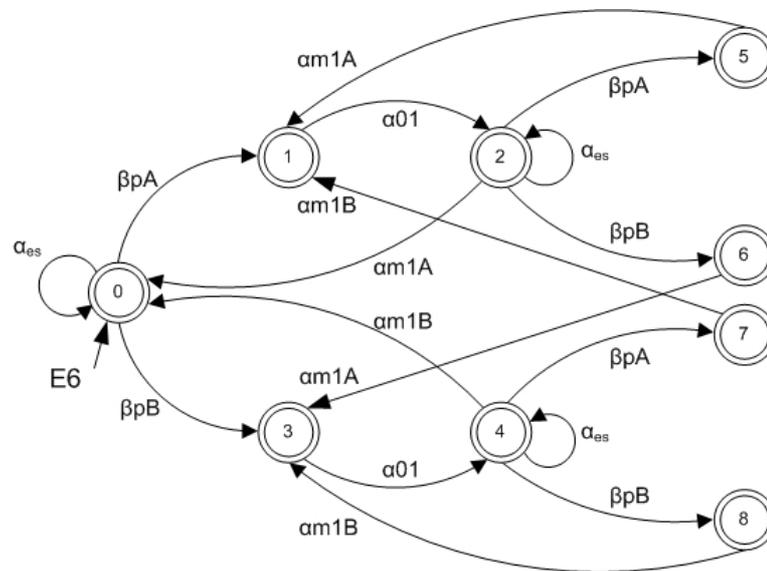


Figura A.5: Especificação E_6 .

A especificação E_7 , apresentada na figura A.6, coordena as ações referentes aos subsistemas máquinas M_1 , robô e máquina M_2 . Considerando o estado dos *buffers* B_2 e B_3 , também simultaneamente.

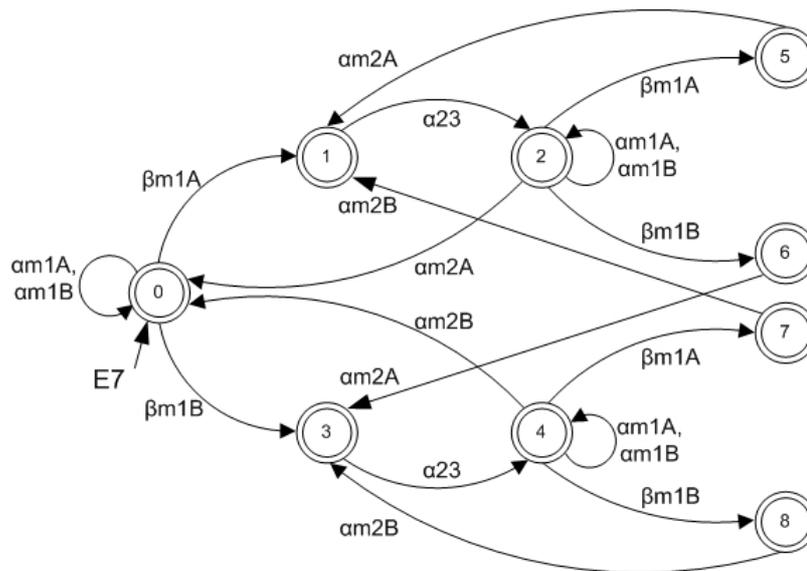


Figura A.6: Especificação E₇.

A especificação E₈, apresentada na figura A.7, coordena as ações referentes aos subsistemas máquinas M₂, robô e máquina M₃. Considerando o estado dos *buffers* B₄ e B₅, também simultaneamente.

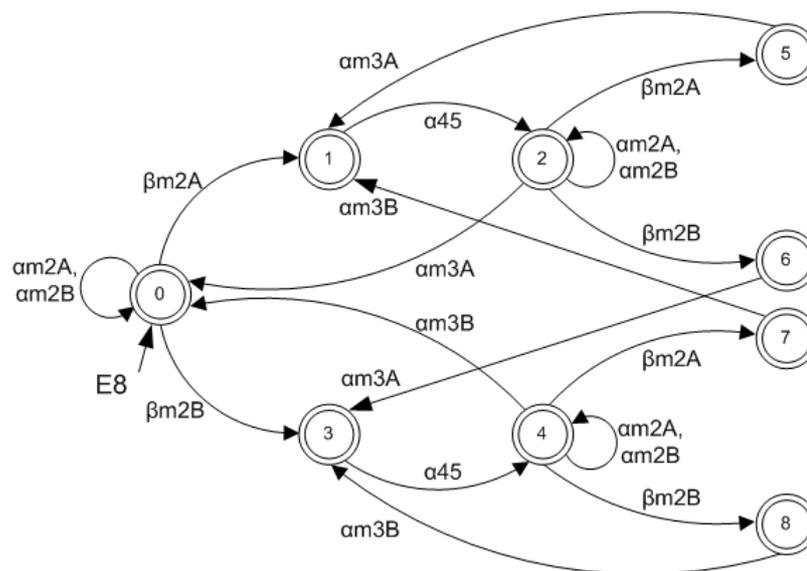


Figura A.7: Especificação E₈.

A especificação E_9 , coordena a movimentação do robô para a realização de todo o processo. Tal especificação faz parte do terceiro conjunto é apresentada na figura A.8 a seguir.

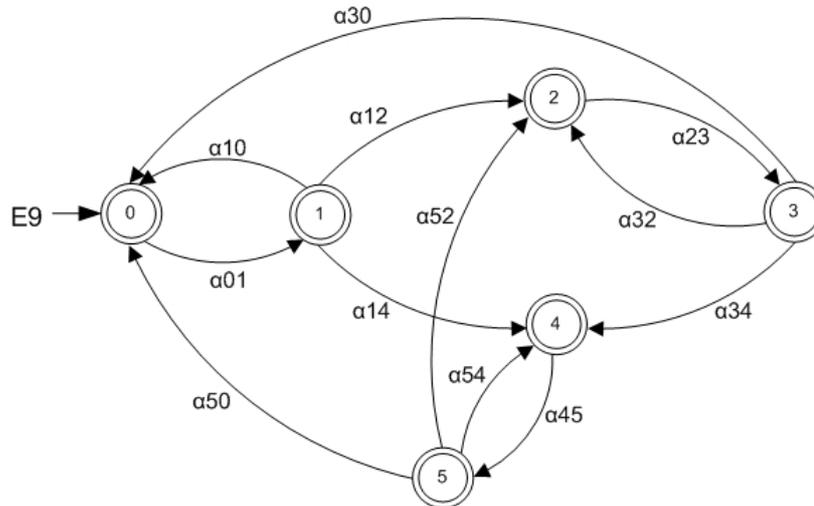


Figura A.8: Especificação E_9 .

A tabela A.1, apresenta o resultado referente à síntese dos supervisores modulares e dos supervisores reduzidos para esta seqüência de processamento.

Tabela A.1 – Resultados da abordagem modular local			
c1	c2	c3 (SM_{E_j})	c4 (S_j)
$G_{loc3}=G_0 G_4$	$CC_0=\{SM_{E0}, G_0, G_4\}$	19x62	2x34
$G_{loc4}=G_0 G_1$	$CC_1=\{SM_{E1}, G_0, G_1\}$	36x124	2x36
	$CC_2=\{SM_{E2}, G_0, G_1\}$	25x81	2x35
$G_{loc5}=G_0 G_2$	$CC_3=\{SM_{E3}, G_0, G_2\}$	36x124	2x36
	$CC_4=\{SM_{E4}, G_0, G_2\}$	25x81	2x35
$G_{loc6}=G_0 G_3$	$CC_5=\{SM_{E5}, G_0, G_3\}$	36x124	2x36
$G_{loc0}=G_0 G_1 G_4$	$CC_6=\{SM_{E6}, G_0, G_1, G_4\}$	252x1031	9x188
$G_{loc1}=G_0 G_1 G_2$	$CC_7=\{SM_{E7}, G_0, G_1, G_2\}$	315x1303	9x191
$G_{loc2}=G_0 G_1 G_3$	$CC_8=\{SM_{E8}, G_0, G_1, G_3\}$	315x1303	9x191
$G_{loc7}=G_0$	$CC_9=\{SM_{E9}, G_0\}$	12x18	6x18

Considerando a abordagem monolítica, foi obtido um supervisor com 32330 estados e 106038 transições.

A.2 Seqüência de Processamento com Divergência e Convergência de Estações

A seqüência de processamento analisada nesta seção apresenta um sistema com rotas alternativas em que ocorre a divergência das peças a partir de uma única máquina, neste caso a esteira transportadora de entrada, que fornece peças para as máquina M_1 e máquina M_2 . Também ocorre a convergência de peças para uma única máquina. Esta convergência se realiza através das peças que são processadas nas máquinas M_1 e M_2 e se direcionam para uma última operação de processamento na máquina M_3 . Sendo assim, o plano de processo para as peças tipo A e B são de acordo com o exposto a seguir:

- Peça A: $B_0 \rightarrow B_1 \rightarrow M_1 \rightarrow B_2 \rightarrow B_5 \rightarrow M_3$.
- Peça B: $B_0 \rightarrow B_3 \rightarrow M_2 \rightarrow B_4 \rightarrow B_5 \rightarrow M_3$.

O modelo do subsistema esteira transportadora de entrada, G_4 , é como definido para o primeiro caso e apresentado na figura A.3. O modelo das máquinas M_t é conforme a figura A.9. Sendo que as máquinas M_1 e M_2 , processam apenas um tipo de peça, já a máquina M_3 processa peças tipo A e B.

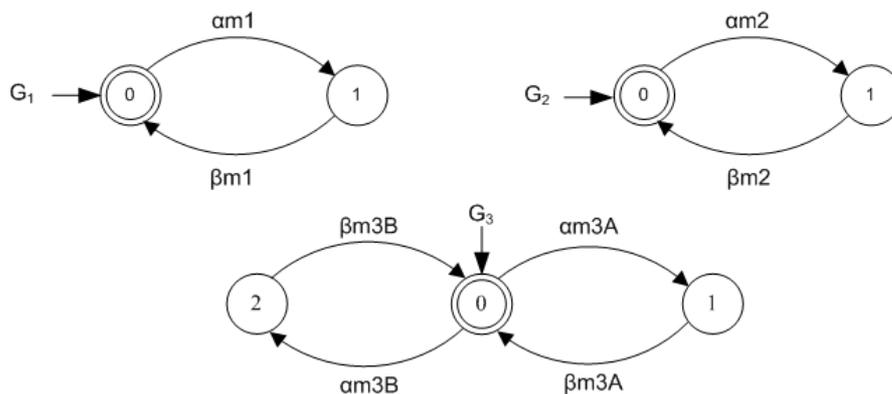


Figura A.9: Autômatos G_1 , G_2 e G_3 das máquinas M_t .

O modelo do robô é apresentado na figura A.10.

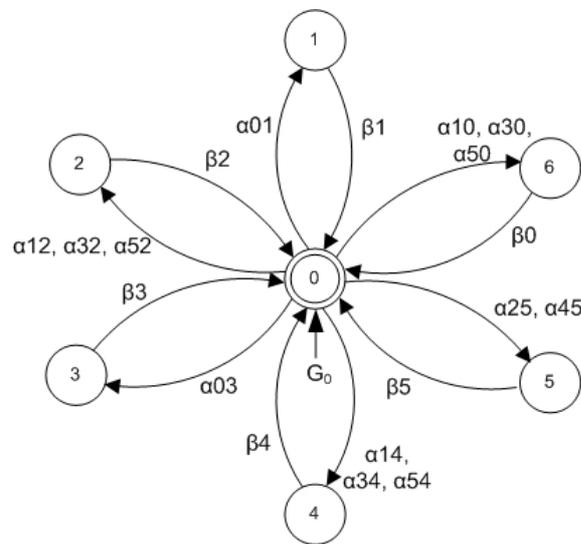


Figura A.10: Modelo do sistema de transporte G_0 , PA(M1M3) e PB(M2M3)..

As especificações de controle referentes ao primeiro e segundo conjuntos, são apresentadas na figura A.11. Sendo que as especificações que correspondem ao primeiro conjunto são as identificadas de E_0 à E_5 . As especificações E_6 à E_9 fazem parte do segundo conjunto e definem uma ação em função de um tipo específico de peça.

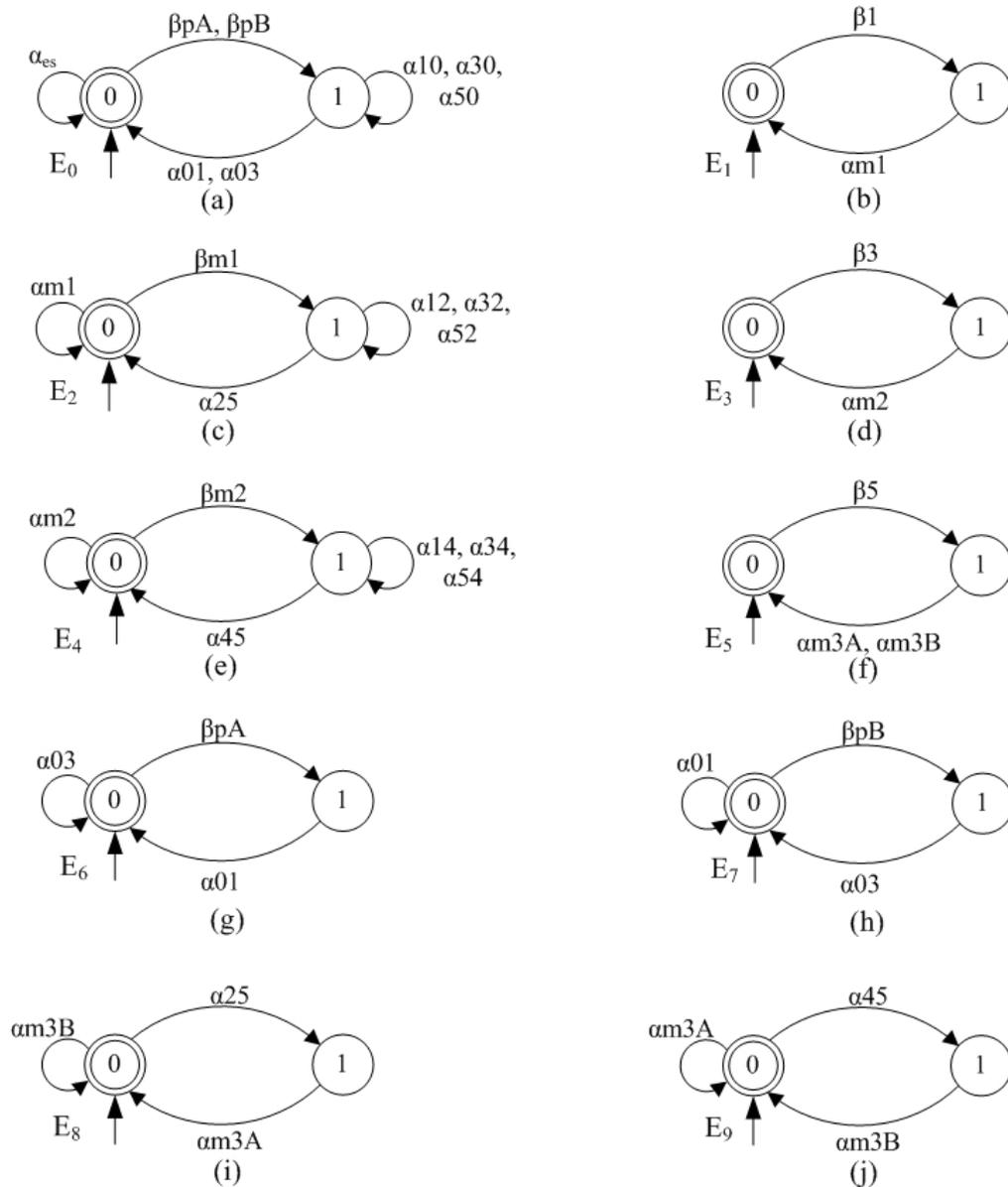


Figura A.11: Especificações do primeiro e segundo conjuntos.

Importante ressaltar que esta seqüência de processamento não necessita do quarto conjunto de especificações, uma vez que não há uma relação unívoca entre os subsistemas em questão. Isto decorre do fato que a esteira alimentadora de entrada fornece peças para as máquinas M_1 e M_2 , e a máquina M_3 recebe peças das máquinas M_1 e M_2 .

A especificação E_{10} que coordena as movimentações do sistema de transporte é apresentada na figura A.12.

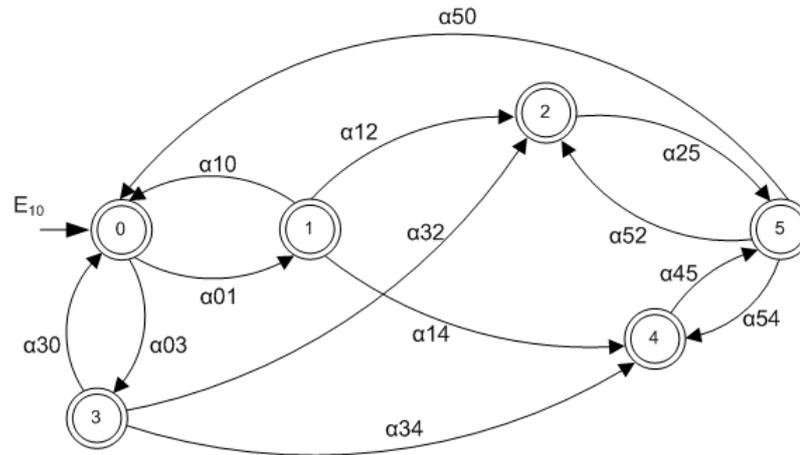


Figura A.12: Especificação do sistema de transporte PA(M1M3) e PB(M2M3).

A tabela A.2, apresenta o resultado referente à síntese dos supervisores modulares e dos supervisores reduzidos para esta seqüência de processamento.

Tabela A.2 – Resultados da abordagem modular local			
c1	c2	c3 (SM_{Ej})	c4 (S_j)
$G_{1oc0}=G_0 G_4$	$CC_0=\{SM_{E0}, G_0, G_4\}$	19x63	2x35
$G_{1oc1}=G_0 G_1$	$CC_1=\{SM_{E1}, G_0, G_1\}$	26x91	2x39
	$CC_2=\{SM_{E2}, G_0, G_1\}$	19x59	2x35
$G_{1oc2}=G_0 G_2$	$CC_3=\{SM_{E3}, G_0, G_2\}$	26x91	2x39
	$CC_4=\{SM_{E4}, G_0, G_2\}$	19x59	2x35
$G_{1oc3}=G_0 G_3$	$CC_5=\{SM_{E5}, G_0, G_3\}$	39x143	2x41
$G_{1oc0}=G_0 G_4$	$CC_6=\{SM_{E6}, G_0, G_4\}$	21x75	2x39
$G_{1oc0}=G_0 G_4$	$CC_7=\{SM_{E7}, G_0, G_4\}$	21x75	2x39
$G_{1oc3}=G_0 G_3$	$CC_8=\{SM_{E8}, G_0, G_3\}$	42x153	2x43
$G_{1oc3}=G_0 G_3$	$CC_9=\{SM_{E9}, G_0, G_3\}$	42x153	2x43
$G_{1oc4}=G_0$	$CC_{10}=\{SM_{E10}, G_0\}$	12x19	6x19

A abordagem monolítica resultou em um supervisor com 7522 estados e 25123 transições.

A.3 Seqüência de Processamento com Divergência de Estações

Nesta seção é desenvolvida a modelagem de um processo em que são processados dois tipos de peças como nos dois casos anteriores. As peças tipo A e B chegam através da esteira transportadora de entrada e seguem sempre em direção à máquina M_1 para uma primeira operação. Sendo assim, a máquina M_1 é capaz de realizar dois diferentes tipos de operação. Após a primeira operação na máquina M_1 , as peças seguem para máquinas distintas para uma última operação. A peça tipo A segue para a máquina M_2 e a peça tipo B segue para a máquina M_3 .

As máquinas M_2 e M_3 realizam apenas um tipo de operação. De forma resumida o plano de processo é como descrito abaixo.

- Peça A: $B_0 \rightarrow B_1 \rightarrow M_1 \rightarrow B_2 \rightarrow B_3 \rightarrow M_2$.
- Peça B: $B_0 \rightarrow B_1 \rightarrow M_1 \rightarrow B_2 \rightarrow B_5 \rightarrow M_3$.

Como modelo para os subsistemas tem se que a esteira transportadora de entrada G_4 é como considerado nos dois casos anteriores. O modelo do robô G_0 é como apresentado na figura A.13.

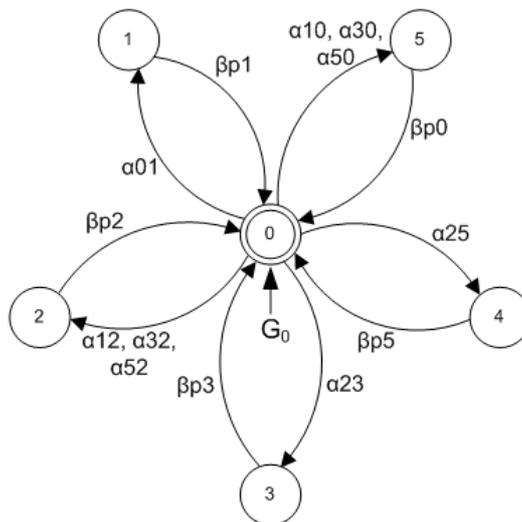


Figura A.13: Modelo do sistema de transporte PA(M_1M_2) e PB(M_1M_3).

Os modelos das máquinas M_1 , M_2 e M_3 são apresentados na figura A.14.

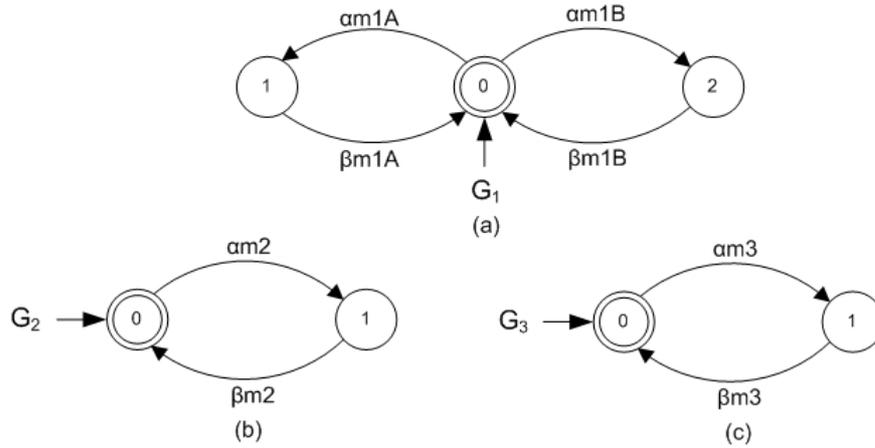


Figura A.14: Modelos das máquinas PA(M1M2) e PB(M1M3).

As especificações de controle referentes ao primeiro e segundo conjunto são apresentadas na figura A.15.

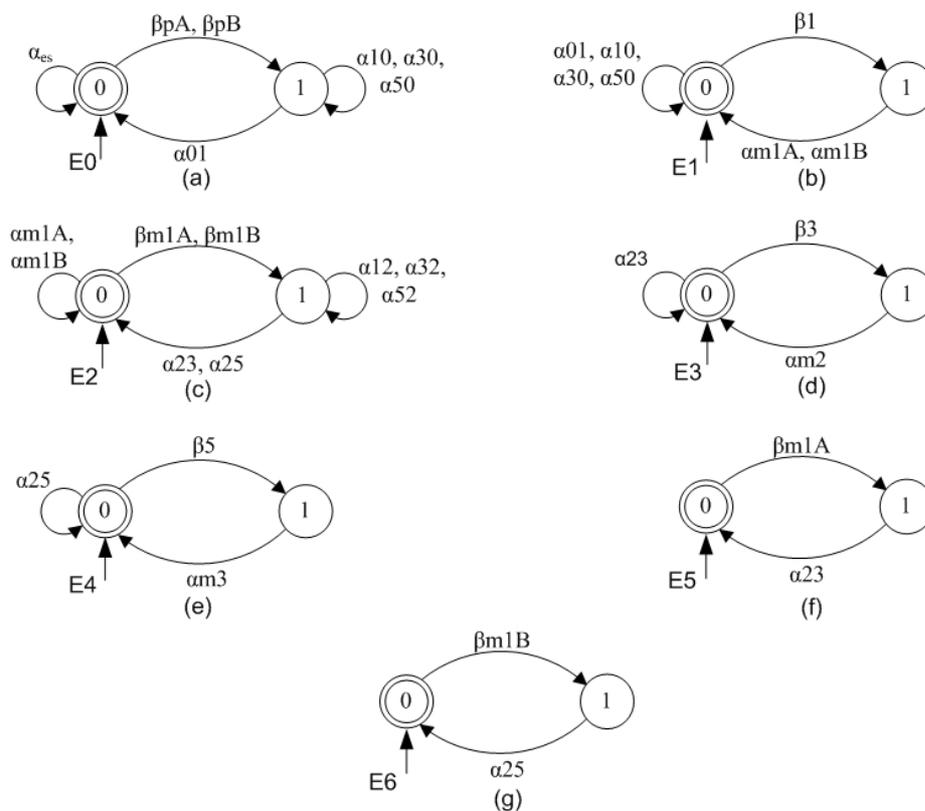


Figura A.15: Especificações de controle processo PA(M1M2) e PB(M1M3).

As especificações E_0, E_1, E_2, E_3 e E_4 correspondem à modelagem do estado dos *buffers* B_0, B_1, B_2, B_3 e B_5 respectivamente e fazem parte do primeiro conjunto. As especificações E_5 e E_6 fazem parte do segundo conjunto.

A especificação E_7 representa o quarto conjunto e é apresentada na figura A.16.

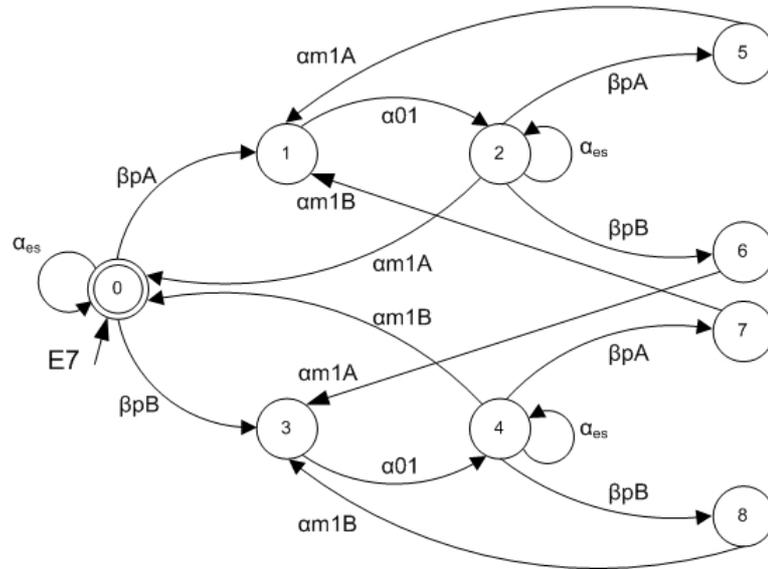


Figura A.16: Especificação do quarto conjunto PA(M1M2) e PB(M1M3).

O terceiro conjunto de especificações é representado pela especificação E_8 na figura A.17.

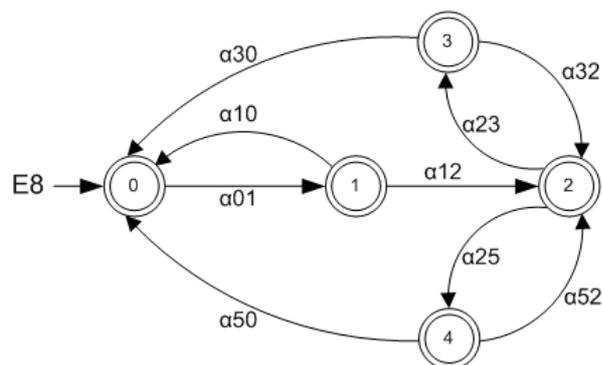


Figura A.17: Especificação para o sistema de transporte PA(M1M2) e PB(M1M3).

A tabela A.3 apresenta os resultados referentes à abordagem modular local.

Tabela A.3 – Resultados da abordagem modular local			
c1	c2	c3 (SM_{Ej})	c4 (S_j)
$G_{loc0}=G_0 G_4$	$CC_0=\{SM_{E0}, G_0, G_4\}$	16x47	2x26
$G_{loc1}=G_0 G_1$	$CC_1=\{SM_{E1}, G_0, G_1\}$	30x94	2x28
	$CC_2=\{SM_{E2}, G_0, G_1\}$	24x68	2x28
	$CC_5=\{SM_{E5}, G_0, G_1\}$	30x103	2x33
	$CC_6=\{SM_{E6}, G_0, G_1\}$	30x103	2x33
$G_{loc2}=G_0 G_2$	$CC_3=\{SM_{E3}, G_0, G_2\}$	22x68	2x29
$G_{loc3}=G_0 G_3$	$CC_4=\{SM_{E4}, G_0, G_3\}$	22x68	2x29
$G_{loc4}=G_0 G_1 G_4$	$CC_7=\{SM_{E7}, G_0, G_1, G_4\}$	216x828	9x152
$G_{loc5}=G_0$	$CC_8=\{SM_{E8}, G_0\}$	10x14	5x14

Na abordagem monolítica resultou em um supervisor com 4514 estados e 15571 transições.

A.4 Seqüência de Processamento com Retrabalho

Esta seção apresenta os resultados referente à um processo em que ocorre uma mesma operação sobre uma peça, porém em instantes distintos. A primeira operação ocorre quando a peça é inserida no sistema vinda do subsistema esteira transportadora de entrada. A segunda operação ocorre quando a peça após ser processada na máquina M_2 retornará à máquina M_1 para ser novamente processada. As máquinas M_t realizam apenas uma operação cada. Neste exemplo é considerado o processamento de apenas uma peça.

Um plano de processo para este caso pode ser descrito de acordo com o exposto na linha abaixo.

- Peça A: $B_0 \rightarrow B_1 \rightarrow M_1 \rightarrow B_2 \rightarrow B_3 \rightarrow M_2 \rightarrow B_4 \rightarrow B_1 \rightarrow M_1 \rightarrow B_2 \rightarrow B_5 \rightarrow M_3$.

Os modelos da esteira transportadora de entrada e das máquinas M_1 , M_2 e M_3 são como representados na figura A.18.

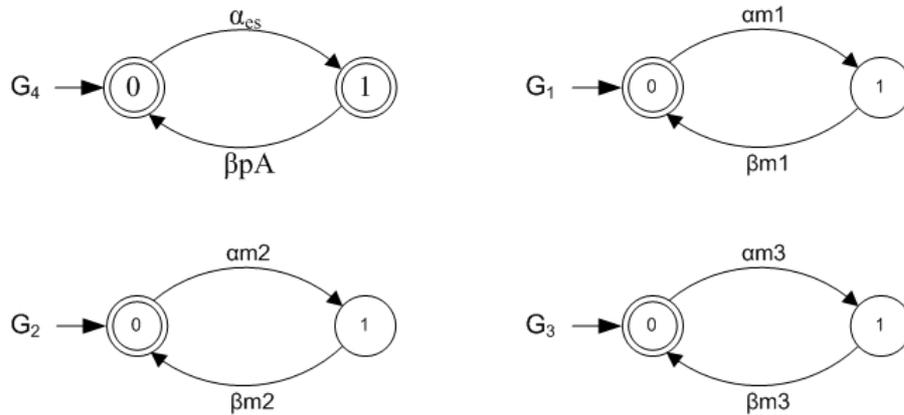


Figura A.18: Modelos dos autômatos G_1 , G_2 , G_3 e G_4 , para o processo com retrabalho.

O modelo do sistema de transporte (robô) representado pelo autômato G_0 é como apresentado na figura A.19.

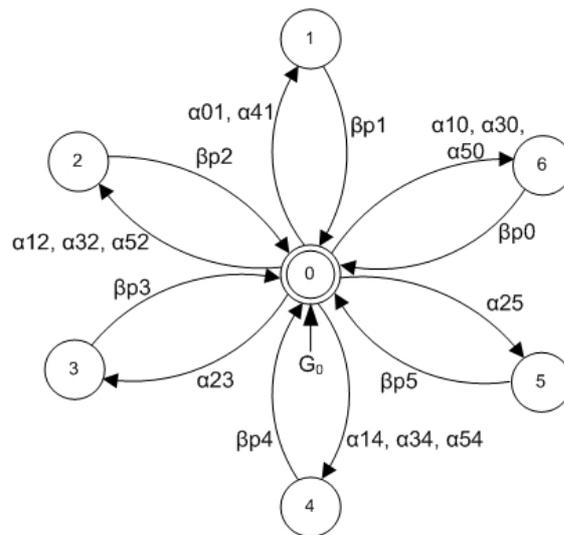


Figura A.19: Modelo do sistema de transporte G_0 para o processo com retrabalho.

As especificações de controle pertencentes ao primeiro conjunto são apresentadas na figura A.20. O estado dos *buffers* B_0 à B_5 assim como a coordenação de algumas ações é imposta pelas especificações E_0 à E_5 respectivamente.

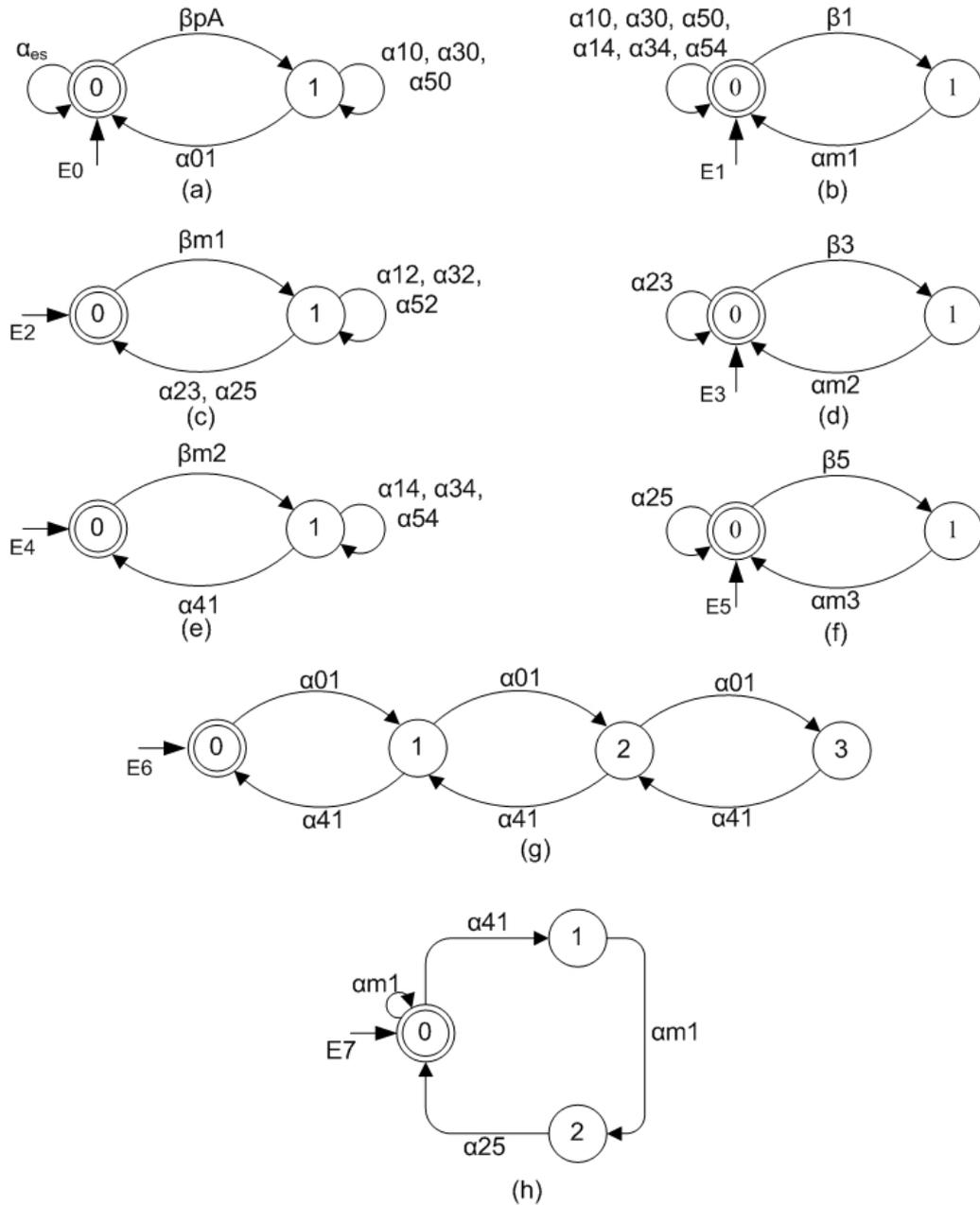


Figura A.20: Especificações do primeiro e segundo conjuntos para o processo de retrabalho.

Neste primeiro conjunto, observações devem ser feitas sobre as especificações E_6 e E_7 das figuras A.20g e A.20h, respectivamente. A especificação E_6 evita que sejam colocadas 4 peças consecutivas no sistema, ou seja, exerce a ação de um contador com valor de contagem igual a 3. Isto se explica, a primeira parte do ciclo de processamento da peça será $M_1 \rightarrow M_2 \rightarrow M_1$, como as máquinas M_1 e M_2 possuem um *buffer de entrada* e um *buffer de saída*, é possível que haja uma peça em cada *buffer*. No entanto, esta hipótese inviabilizaria que a primeira peça a entrar no sistema para processamento, sofra a segunda operação

(retrabalho) na máquina M_1 , se o *buffer de entrada* da máquina M_1 estiver ocupado. Sendo assim, permite-se que haja o transporte de apenas 3 peças brutas para o sistema. Deste modo, a especificação E_6 garante que o *buffer de entrada* da máquina M_1 estará vazio, quando uma peça precisar sofrer a segunda operação (retrabalho) em M_1 .

A especificação E_7 da figura A.20h, garante que após a segunda operação em M_1 a peça seja transportada para a máquina M_3 .

O terceiro conjunto de especificações é representado pela especificação E_8 da figura A.21. Esta especificação coordena as movimentações do robô para a realização do processo em questão.

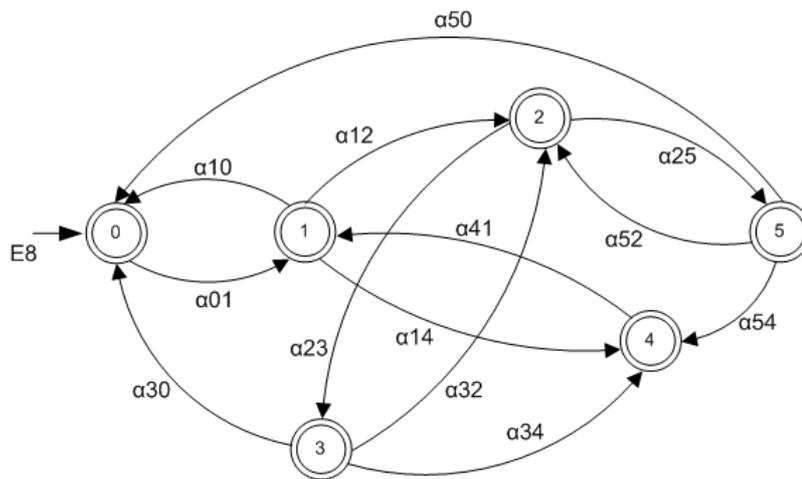


Figura A.21: Especificação do terceiro conjunto, E_8 .

Os resultados da abordagem modular local obtidos no processo de síntese dos supervisores são apresentados na tabela A.4.

A utilização da abordagem monolítica resultou em um supervisor de 3186 estados e 9638 transições.

Tabela A.4 – Resultados da abordagem modular local			
c1	c2	c3 (SM_{Ei})	c4 (S_j)
$G_{1oc0}=G_0 G_4$	$CC_0=\{SM_{E0}, G_0, G_4\}$	19x59	2x35
$G_{1oc1}=G_0 G_1$	$CC_1=\{SM_{E1}, G_0, G_1\}$	22x69	2x30
	$CC_2=\{SM_{E2}, G_0, G_1\}$	19x57	2x34
	$CC_7=\{SM_{E7}, G_0, G_1\}$	38x134	3x56
$G_{1oc2}=G_0 G_2$	$CC_3=\{SM_{E3}, G_0, G_2\}$	26x91	2x39
	$CC_4=\{SM_{E4}, G_0, G_2\}$	19x59	2x35
$G_{1oc3}=G_0 G_3$	$CC_5=\{SM_{E5}, G_0, G_3\}$	26x91	2x39
$G_{1oc4}=G_0$	$CC_6=\{SM_{E6}, G_0\}$	28x74	4x74
	$CC_8=\{SM_{E8}, G_0\}$	12x19	6x19

A figura A.22 apresenta uma especificação referente ao quarto conjunto para o processamento de três diferentes tipos de peças. As considerações sobre tal especificação já foram apresentadas no capítulo 5 na seção 5.1.

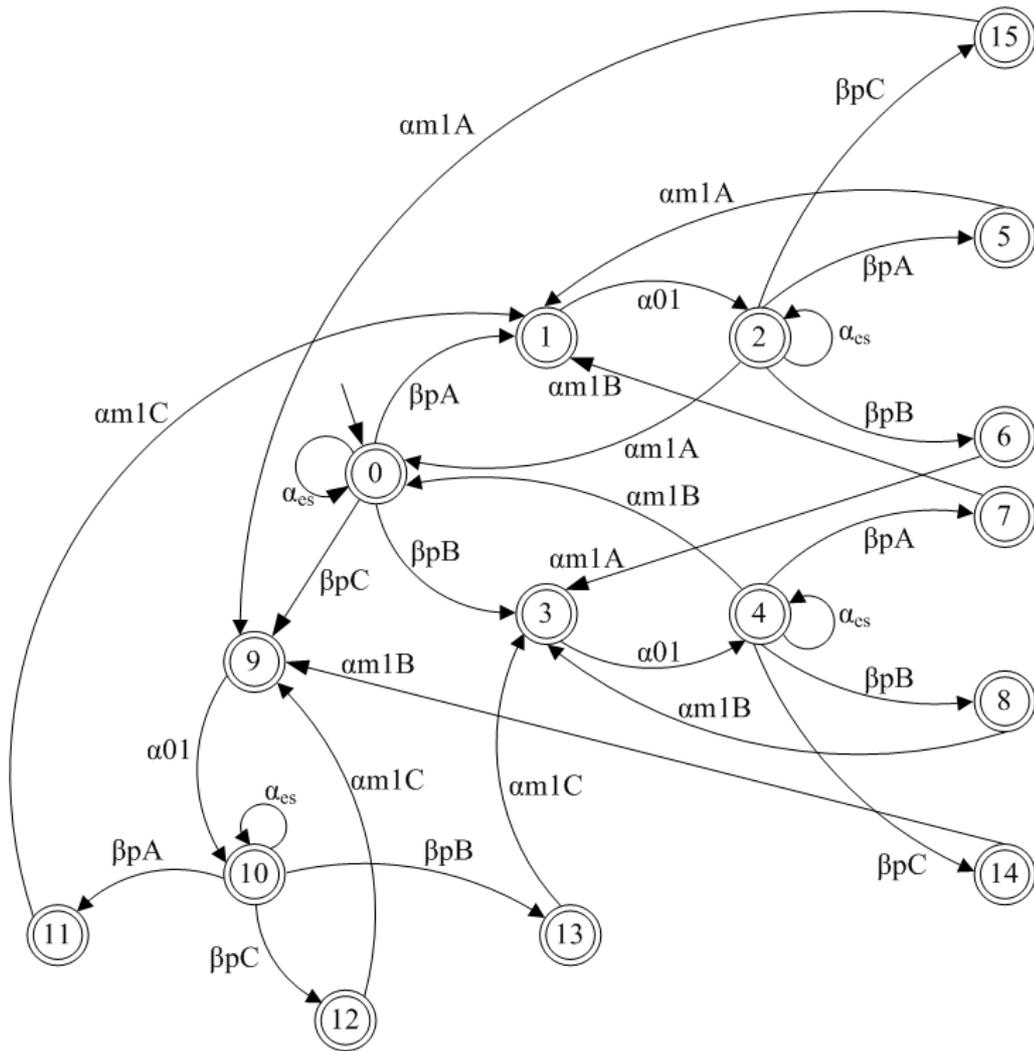


Figura A.22: Especificação do quarto conjunto para 3 peças.

