Universidade Federal de Sergipe

DIAGNÓSTICO DE FALHAS BASEADO EM AUTÔMATOS TEMPORIZADOS: APLICAÇÃO EM UM SISTEMA MODULAR DE MANUFATURA

Wellington Alves Santana Júnior

2016

DIAGNÓSTICO DE FALHAS BASEADO EM AUTÔMATOS TEMPORIZADOS: APLICAÇÃO EM UM SISTEMA MODULAR DE MANUFATURA

Wellington Alves Santana Júnior

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica – PROEE, da Universidade Federal de Sergipe, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Carlos A. V. Cardoso

São Cristóvão – SE, Brasil
Agosto de 2016

DIAGNÓSTICO DE FALHAS BASEADO EM AUTÔMATOS TEMPORIZADOS: APLICAÇÃO EM UM SISTEMA MODULAR DE MANUFATURA

Wellington Alves Santana Júnior

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA – PROEE DA UNIVERSIDADE FEDERAL DE SERGIPE COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA.

Examinada por:

_____

Prof. Dr. Lucas Molina
Presidente da Banca

_____

Prof. Dr. Douglas Bressan Riffel
Examinador Interno

_____

Prof. Dr. Jânio Coutinho Canuto
Examinador Externo

SÃO CRISTÓVÃO – SE, BRASIL
AGOSTO DE 2016

Agradecimentos

Ao Deus de minha compreensão, por expor minha consciência a mais uma experiência nobre.

À minha esposa, Histéfani Almeida Alves, pelo carinho e enorme apoio.

À minha filha, Sara Almeida Alves, pela alegria contagiante.

Ao meu orientador, Carlos Alberto Villacorta Cardoso, pela confiança e orientação.

À minha família e amigos pelas palavras de incentivo.

Resumo da Dissertação apresentada ao PROEE/UFS como parte dos requisitos necessários para a obtenção do grau de Mestre (Me.)

DIAGNÓSTICO DE FALHAS BASEADO EM AUTÔMATOS TEMPORIZADOS: APLICAÇÃO EM UM SISTEMA MODULAR DE MANUFATURA

Wellington Alves Santana Júnior

Agosto/2016

Orientador: Prof. Dr. Carlos A. V. Cardoso

Programa: Engenharia Elétrica

O problema do diagnóstico de falhas, utilizando a teoria de Sistemas a Eventos Discretos, tem sido largamente abordado pela comunidade acadêmica. Entretanto, a aplicação desta teoria a sistemas reais é um campo onde há muito a ser explorado. O problema proposto, neste trabalho, é o de diagnosticar falhas permanentes ou intermitentes de dispositivos (sensores e atuadores) pertencentes a estações de um sistema flexível de manufatura didático, denominado Sistema Modular de Produção - MPS, fabricado pela empresa Festo. Este objetivo será alcançado por meio de uma modelagem e simulação que permitam uma futura implementação no sistema. Serão apresentados três métodos para diagnóstico de falhas escritos na linguagem do software UPPAAL que se baseia no formalismo autômatos seguros temporizados, conforme proposto por ALUR e DILL (1994) e HENZINGER *et al* (1994). O primeiro método é uma implementação do diagnosticador proposto em TRIPAKIS (2002). Os outros dois métodos, elaborados nesta pesquisa, são inspirados no diagnosticador TRIPAKIS (2002) e são diagnosticáveis pelos critérios apresentados em TRIPAKIS (2002) e I-diagnosticabilidade conforme SAMPATH *et al.* (1995). As estratégias para detecção de falhas incluem a utilização de uma rede de autômatos seguros temporizados, composta pelo autômato que descreve o comportamento do processo e por autômatos diagnosticadores para cada tipo de falha. Os diagnosticadores detectam as falhas a partir da observação de atrasos de determinadas transições do autômato G (processo) e as isolam por meio de observações dos estados dos sensores. Eventos indicadores de falhas servem para anunciar falhas e sincronizar o autômato G com os diagnosticadores.

Abstract of Dissertation presented to PROEE/UFS as a partial fulfillment of the requirements for the degree of Master

FAULT DIAGNOSIS BY TIMED AUTOMATA: APPLICATION ON MODULAR PRODUCTION SYSTEM

Wellington Alves Santana Júnior

August/2016

Advisor: Prof. Dr. Carlos A. V. Cardoso

Department: Electrical Engineering

The problem of fault diagnosis has been widely discussed by the academic community using the theory of Discrete Event Systems. However, the application of this theory to real systems is a field where there is a lot to be explored. The problem proposed in this work is to diagnose permanent or intermittent failures in devices (sensors and actuators) belonging to stations of a didactic flexible manufacturing system, called Modular Production System - MPS, produced by Festo company. The objective will be achieved through a modeling and simulation that allow for future implementation in the system. Three methods will be presented on fault diagnosis written in UPPAAL software language which is based on the timed safety automata formalism, as proposed by ALUR and DILL (1994) and HENZINGER et al (1994). The first method is an implementation of TRIPAKIS (2002) diagnoser. The other two methods developed in this research are inspired by TRIPAKIS (2002) and are diagnosable by definitions presented in TRIPAKIS (2002) and I-diagnosability presented in Sampath et al. (1995). The strategies for fault detection include the use of a network of timed safety automata, composed of the automaton that describes the process behavior and the diagnosers automata for each type of failure. The diagnosers detect failures from the observation of delays of certain transitions in the automaton G (process) and isolate them through observations of the sensors states. Fault indicators events serve to announce failures and synchronize the automaton G with the diagnosers.

Sumário

1 Introdução

A manufatura passou por muitas transformações ao longo da história. Começando com a produção manual intensiva e as linhas de produção em massa, as primeiras tentativas de automação industrial começam com a implementação de linhas automatizadas de produção adotadas pela indústria automobilística no século XIX (MORALES, 2009).

Posteriormente são introduzidos os *flow shops* com o objetivo de aumentar a produtividade. A produção foi adaptada para executar tarefas repetitivas e cíclicas; os *job shops* e a manufatura celular com o intuito de produzir de forma eficiente vários tipos de produtos com algum grau de similaridade de maneira mais eficaz. Mais tarde, a introdução de recursos computacionais possibilitou o desenvolvimento de ferramentas capazes de aumentar a flexibilidade e viabilizar alterações no que é produzido e como é produzido. Assim, foi concebida a ideia de sistema flexível de manufatura (*Flexible Manufacturing System* – FMS em inglês).

Sistemas flexíveis de manufatura podem ser projetados para atender a diferentes linhas de produção. Estes são denominados Sistemas Modulares de Produção (*Modular Production System* - MPS). Possuem um conceito de arquitetura aberta que permite a evolução contínua de seu arranjo físico: é possível agregar novas unidades, conforme haja necessidade de atender novos processos de manufatura que envolve diferentes funcionalidades de acordo com o perfil desejado.

O funcionamento do FMS apresenta uma natureza assíncrona, envolvendo em geral, processos concorrentes e paralelos que incluem vários componentes e subsistemas que compartilham recursos limitados e que interagem entre si de maneira relativamente complexa e em tempo real. Dois problemas são comuns em tais sistemas: possibilidade de *deadlock*, e possibilidade de conflito (SANTOS FILHO, 2000). O *deadlock* se define como um estado a partir do qual o sistema não pode evoluir mais, e no qual permanece por tempo indefinido. O conflito por sua vez é definido como um estado no qual dois ou mais processos requerem simultaneamente um mesmo recurso do sistema. Estes problemas podem ter sua origem em uma falha de algum componente físico do sistema, como por exemplo, um sensor de fim de curso.

Falhas em componentes físicos de um sistema podem ser detectadas através de duas abordagens (FRANK, 1992 e CHANG *et al.*, 1991):

- Mediante a utilização de sensores para monitoração de sinais, variáveis ou parâmetros específicos de um equipamento. Por exemplo, a ruptura de uma correia pode ser detectada através de um sensor que monitore este evento.
- Aplicação de alguma técnica ou método (ferramenta matemática) para substituir a utilização de um sensor físico (MORALES, 2009).

Um dos métodos de detecção e identificação de faltas, baseado em modelos matemáticos, utiliza a teoria de Sistemas a Eventos Discretos (SEDs) (SAMPATH *et al.*, 1995; LAFORTUNE *et al.*, 2001). Este método tem como base trabalhos que abordaram o problema de observação parcial e observabilidade que foram desenvolvidos na década de 80 e início da década de 90 (ZAYTOON, 2013). Em LIN (1994) é proposto uma abordagem baseada em estados, para diagnosticabilidade de SEDs. Em seu trabalho, Lin propôs algoritmos para computar sequências de comandos de teste para diagnosticar faltas. Métodos de otimização de sensores com fins de diagnóstico foram propostos ao mesmo tempo por BAVISH e CHONG (1994). Modelos Redes de Petri de SEDs, para fins de diagnóstico de faltas, foram também desenvolvidos na década de 80 e 90 (ZAYTOON, 2013). SAMPATH *et al.* (1995) apresentaram condições necessárias e suficientes para a diagnose de faltas de SEDs e propuseram a construção de um autômato diagnosticador que permite tanto inferir sobre a capacidade de diagnosticar as faltas presentes no sistema quanto ser usado para realizar a diagnose de faltas em tempo real. Neste trabalho, SAMPATH *et al.* (1995) define formalmente as noções e condições necessárias e suficientes para Diagnosticabilidade e I-diagnosticabilidade de SEDs tanto para o caso da ocorrência de faltas não múltiplas como múltiplas. Em um trabalho correlacionado, SAMPATH *et al.* (1996) consideraram o problema de diagnóstico de faltas de um sistema de condicionamento de ar.

A inclusão da informação de tempo em SEDs resultou nos chamados autômatos temporizados. Nos modelos temporizados, os traços (trajetórias) não são especificados somente em termos de sequências de estados ou eventos, mas incluem a informação do tempo entre dois eventos. ALUR e DILL (1994) propuseram o chamado Autômato Temporizado com guarda ou Autômato de tempo contínuo que emprega um conjunto de

clocks ou relógios para restringir transições entre estados. Os *clocks* assumem valores reais e estão associados a dinâmica das transições. Esta dinâmica é definida por expressões denominadas guardas, compostas por *clocks* e operadores lógicos. HENZINGER *et al.*. (1994) criam uma variação do autômato temporizado introduzido por ALUR e DILL (1994), denominada de Autômatos Seguros Temporizados (*Timed Safety Automata*) a qual especifica propriedades de progresso usando condições de invariante local. TRIPAKIS (2002) estendeu os resultados de SAMPATH *et al.* (1995) para SEDs modelados pelos autômatos temporizados com guarda de ALUR e DILL (1994), sendo a diagnose de faltas baseada não somente nas sequências de eventos observáveis, mas também nos intervalos de tempo decorridos entre dois eventos sucessivos. BENGTSSON e YI (2004) apresentam resultados de trabalhos com autômatos temporizados focando em semântica e algoritmos empregados em ferramentas de verificação de modelos. Um dos tópicos do trabalho é dedicado à ferramenta de verificação de modelos baseada em autômatos temporizados UPPAAL, a mesma ferramenta utilizada no presente trabalho.

Outras abordagens para o problema da diagnose de faltas em SEDs temporizados foram apresentadas por CHEN e PROVAN (1997), ZAD *et al.* (1999) e ZAD *et al.* (2005) que consideraram a diagnose de faltas em modelos a tempo-discreto. Nestes trabalhos, o tempo decorrido entre eventos é modelado tendo como base um evento observável especial denominado "*clock tick*", sendo o problema da diagnose de falta resolvido utilizando-se técnicas de modelos não-temporizados. Além dessas abordagens, é importante mencionar o trabalho de HOLLOWAY e CHAND (1996), que propôs uma nova técnica para diagnose de falta distribuída denominada monitoração de padrões, que utiliza conjuntos de temporizações e relações sequenciais para determinar quando estão previstas as ocorrências dos eventos e para precisar se um evento ocorreu ou não.

As publicações citadas acima contribuíram com importantes fundamentos sobre diagnosticabilidade, construção de diagnosticadores e a teoria de SEDs. Entretanto, um importante esforço permanece necessário para aplicar esta teoria em sistemas reais (ZAYTOON, 2013). A aplicação da teoria de sistemas a eventos discretos para fins de diagnóstico de faltas em sistemas de manufatura é apresentada em RIASCOS (2002), G. MORALES (2009) e PHILIPPOT, SAYED-MOUCHAWEH e CARRÉ-MÉNÉTRIER (2008). RIASCOS (2002) introduz uma metodologia baseada no conceito de redes de

Petri, que além da modelagem e análise dos processos normais, permite a detecção e o tratamento de faltas em sistemas de manufatura de uma forma hierárquica e modularizada através da abordagem “top-down” e “bottom-up” e a utilização de supervisores distribuídos nos equipamentos de chão de fábrica. G. MORALES (2009) propõe a utilização de redes Bayesianas e redes de Petri para diagnóstico de faltas em sistemas flexíveis de manufatura. Em PHILIPPOT, SAYED-MOUCHAWEH e CARRÉ-MÉNÉTRIER (2008) é proposto um diagnosticador descentralizado para detectar e isolar faltas em sensores e atuadores de uma planta de manufatura. Cada diagnosticador local é um modelo de SED booleano que infere a ocorrência de faltas a partir da sequência de eventos, atrasos no tempo entre eventos correlacionados e condições de estados. No que diz respeito à aplicação de ferramentas formais de verificação de modelos com base na teoria de autômatos temporizados, pode-se citar o trabalho de HSIEH e CHANG (2016). Estes fazem uso de autômatos temporizados e da ferramenta de verificação de modelos UPPAAL para construir possíveis cenários de propagação de faltas e seus traços (sequências) de eventos observáveis em três exemplos de processo à batelada. O objetivo do trabalho é de obter um modelo ótimo de teste de processos, para cada traço de eventos observáveis utilizando um controlador supervisório que executa a solução de diagnóstico de faltas.

O objetivo do presente trabalho consiste em modelar, simular e diagnosticar faltas induzidas em cinco subprocessos de um FMS didático, denominado Sistema Modular de Produção - MPS, fabricado pela empresa Festo. As faltas a serem diagnosticadas estão concentradas nos dispositivos (sensores e atuadores) das estações (subprocessos) responsáveis pela fabricação de relógios, termômetros e higrômetros na planta didática. Devido ao comportamento dinâmico do sistema modular de produção ser característico de sistemas cuja natureza é do tipo a eventos discretos, este trabalho aplicou o formalismo de autômatos temporizados para alcançar o objetivo citado.

A escolha desta ferramenta foi norteada por dois critérios:

I. A quantidade de técnicas e métodos desenvolvidos e consolidados a respeito de diagnóstico de faltas, usando modelos a eventos discretos, baseado no formalismo de autômatos.
II. A possibilidade de incluir informações da dinâmica temporal na modelagem dos processos. O acréscimo da variável de tempo no modelo matemático

permite que haja uma maior aproximação entre simulação computacional e o funcionamento real dos sistemas.

O formalismo autômato não temporizado foi escolhido inicialmente tanto para modelar os processos quanto para aplicar uma possível estratégia de diagnóstico das faltas induzidas nos sensores e atuadores. Entretanto, como será mostrado no capítulo 4, a aplicação do autômato tradicional mostrou-se inapropriada. Isto porque a técnica aplicada, neste trabalho, para isolar as faltas dos sensores e atuadores depende da observação da mudança de estado dos sensores envolvidos em uma determinada operação. Um desvio no comportamento esperado dos sensores indicará a manifestação de alguma falta.

É importante mencionar que falhas são resultantes de faltas. Faltas representam desvios indesejados no comportamento esperado de um sistema. Estes desvios degradam o funcionamento do sistema podendo levá-lo a um estado de falha. A falha é a interrupção permanente da capacidade de um sistema executar a função para o qual foi projetado sob condições específicas (ISERMANN e BALLÉ, 1997). Nesta pesquisa, as faltas induzidas nos processos, utilizados como estudo de caso, levam a um estado de bloqueio ou *deadlock*. Os dois tipos de faltas a serem diagnosticadas, a saber, faltas permanentes e faltas intermitentes, levam a estados de falhas. Por este motivo, dentro do escopo deste trabalho, os dois termos serão tratados como sinônimos.

Um modelo de diagnosticador baseado em autômatos temporizados, proposto em TRIPAKIS (2002), será abordado no capítulo 2. No capítulo 4, este diagnosticador será interpretado, adaptado para linguagem do software UPPAAL e aplicado a uma operação típica de processos de manufatura. Esta mesma operação servirá como exemplo para apresentação, no capítulo 4, dos métodos elaborados nesta pesquisa para diagnosticar falhas permanentes e intermitentes de sensores e atuadores. Os dois métodos são apropriados para o mesmo tipo de sistema e configuração dos dispositivos. São inspirados no diagnosticador proposto em TRIPAKIS (2002) e são diagnosticáveis pelos critérios apresentados em TRIPAKIS (2002) e I-diagnosticabilidade conforme SAMPATH *et al.* (1995). Ambos métodos são implementáveis através da linguagem UPPAAL. Conforme será tratado no Capítulo 2, esta linguagem utiliza o formalismo de autômatos seguros temporizados (ALUR e DILL, 1994; HENZINGER *et al.*., 1994) estendida para uma rede de autômatos temporizados com modificações para aproximá-la de uma linguagem de programação de alto nível que lida com vários tipos de dados.

Embora haja muitos trabalhos relacionados a diagnóstico de falhas, utilizando a teoria de sistemas a eventos discretos, e alguns dedicados a aplicação desta teoria em sistemas de manufatura, não foram encontrados trabalhos que modelem sistemas de manufaturas utilizando uma linguagem baseada no formalismo de autômatos seguros temporizados e ao mesmo tempo proponha um diagnosticador *on-line* para falhas permanentes ou intermitentes de sensores e atuadores.

O método para diagnóstico de falhas proposto em TRIPAKIS (2002) foi aplicado à toda estação de distribuição. O método desenvolvido nesta pesquisa para diagnóstico de falhas intermitentes, foi aplicado somente ao sistema apresentado no capítulo 4. Por fim, o método proposto para diagnosticar falhas permanentes foi empregado em todas as estações pertencentes ao processo de fabricação de relógios, termômetros e higrômetros do Sistema Modular de Produção (FESTO).

2 Referencial Teórico

2.1 Sistemas a Eventos Discretos

Sistemas a Eventos Discretos (SEDs) são sistemas dinâmicos que possuem espaços de estados discretos, e a transição entre estados ocorre em função de eventos, em geral, assíncronos no tempo. Os eventos representam a ocorrência de algum fenômeno ou a execução de alguma ação ou conjunto de ações específicas que podem afetar o comportamento do sistema. Os estados estão associados a condições que o sistema alcança quando da ocorrência de um determinado evento. Estas condições podem ser valores simbólicos, como, por exemplo, {aberto, fechado} ou discretos tais como valores numéricos. Qualquer tipo de sistema físico pode ser modelado como um sistema a eventos discretos. Entretanto, há sistemas que possuem naturalmente um comportamento dinâmico característico de SEDs. A título de exemplo, de tais sistemas, pode-se citar os sistemas de manufatura.

O comportamento de um SED pode ser descrito por linguagens definidas sobre um alfabeto $\sum$. $\sum$, pois denota um conjunto finito e não vazio de símbolos que estão associados aos eventos de um sistema. Sendo assim, uma linguagem L sobre $\sum$ é definida como um conjunto de palavras formadas pelas sequências de símbolos pertencentes a $\sum$. As palavras são também conhecidas como cadeias ou traços. Uma palavra s possui um comprimento, isto é, quantidade de eventos que a compõe, denotado por |s|. Uma sequência que não possui eventos é denominada sequência vazia, sendo identificada por ε. Formalmente, o conceito de linguagem é definido como a seguir:

Definição 2.1. *(Linguagem) Uma linguagem definida sobre um conjunto de eventos $\sum$ é um conjunto formado por sequências de comprimento finito construídas a partir de eventos pertencentes a $\sum$ (MAAS, 2004).*

A fim de facilitar o entendimento da definição 2.1, considere a linguagem $L_1 = \{\mu\alpha, \alpha\beta\}$, definida sobre um conjunto de eventos $\sum$. L_1 é um exemplo de uma linguagem composta por apenas duas palavras.

Como uma linguagem representa um conjunto, as operações e propriedades válidas na teoria de conjunto como interseção, união e outras são também aplicáveis às

linguagens. Uma das operações comuns sobre linguagens é a operação de concatenação. Esta operação é formalmente definida conforme abaixo (CASSANDRAS e LAFORTUNE, 2008):

Definição 2.2. *(Concatenação) Sejam* $L_1, L_2 \subseteq \Sigma^*$*,então a concatenação* L_1L_2 *é definida como:*

$$L_1L_2 := \{s \in \Sigma^* : (s = s_1s_2)[s_1 \in L_1 e\ s_2 \in L_2\} \qquad (2.1)$$

A operação de concatenação consiste em unir duas ou mais sequências para formar uma nova sequência. Considere a cadeia s formada pela concatenação dos traços s_1, s_2 e s_3. A cadeia $s = s_1s_2s_3$ tem como prefixo a cadeia s_1; s_2 é a subcadeia de s; s_3 é o sufixo de s.

A operação *Fecho de Kleene* sobre uma linguagem L é dado pela concatenação de um número finito de cadeias de L, incluindo a cadeia vazia ε. Formalmente esta operação é definida por:

Definição 2.3. *(Fecho de Kleene) Seja* $L \subseteq \Sigma^*$*, então o Fecho de Kleene de L denotado por* L^**, é definido como:*

$$L^* := \{\varepsilon\} \cup L \cup LL \cup LLL \cup \ldots \qquad (2.2)$$

Outras operações definidas sobre linguagens são as operações de prefixo fechamento, pós linguagem, projeção e projeção inversa cujas definições são apresentadas a seguir:

Definição 2.4. *(Prefixo Fechamento) Seja* $L \subseteq \Sigma^*$*, então o prefixo fechamento de L, denotado por* $\overline{L}$ *é definido como (MAAS, 2004):*

$$\overline{L} := \{s \in \Sigma^* : (\exists t \in \Sigma^*)[st \in L]\} \qquad (2.3)$$

Se $L = \overline{L}$*, diz-se que a linguagem L é prefixo-fechada.*

Definição 2.5. *(Pós Linguagem) Seja* $L \subseteq \Sigma^*$ *e* $s_1 \in L_1$ *então a pós linguagem de L após* s*, denotada por* L/s*, é a linguagem (MAAS, 2004):*

$$L/s := \{t \in \Sigma^* : st \in L\} \qquad (2.4)$$

Definição 2.6. *(Projeção) Seja* $\sum$ *e* $\sum_O$ *conjuntos de eventos, onde* $\sum_O \subset \sum$*. A projeção* P_0 *de uma sequência de eventos* $\sum$ *em* $\sum_O$ *é definida como (*RAMADGE e WONHAM, 1989; CASSANDRAS e LAFORTUNE, 2008)*:*

$$P_0: \Sigma^* \to \Sigma^*{}_0 \tag{2.5}$$

Com as seguintes propriedades:

$$\begin{gathered} P_0(\varepsilon) \coloneqq \varepsilon, \\ P_0(\sigma) \coloneqq \begin{cases} \sigma, se\ \sigma \in \sum_O, \\ \varepsilon, se\ \sigma \in \sum \backslash \sum_O, \end{cases} \\ P_0(s\sigma) \coloneqq P_0(s)P_0(\sigma), para\ s \in\ \Sigma^*, e\ \sigma \in \sum \end{gathered} \tag{2.6}$$

onde $\backslash$ *denota a diferença entre conjuntos. Assim a operação* $\sum \backslash \sum_{O,} = \sum - \sum_O$*, isto é, o conjunto formado pelos elementos de* $\sum$ *que não pertencem ao conjunto* $\sum_O$*.*

O operador projeção aplicado sobre uma sequência s de eventos elimina de s os eventos não observáveis. Vale lembrar que duas sequências distintas podem ter a mesma projeção. O operador projeção pode ser também estendido para uma linguagem L , aplicando as definições mostradas acima a todas as sequências pertinentes à linguagem *L* (CARVALHO, 2011). Assim, se $L \subset \Sigma^*$então:

$$P_0(L) \coloneqq \{t \in \Sigma_0^*: (\exists s \in L)[P_0(s) = t]\} \tag{2.7}$$

Definição 2.7. *(Projeção Inversa) A projeção inversa de uma sequência de eventos é definida como uma função (MAAS, 2004):*

$$\begin{gathered} P_0^{-1}: \Sigma_0^* \to 2^{\Sigma^*} \\ P_0^{-1}(t) \coloneqq \{s \in \sum^*: P_0(s) = t\} \end{gathered} \tag{2.8}$$

A projeção inversa também pode ser estendida para linguagens de modo que para $L \subset \Sigma_0^*$, tem-se:

$$P_0^{-1}(L_0) \coloneqq \{s \in \Sigma^*: (\exists t \in L_0)[P_0(s) = t]\} \tag{2.9}$$

2.2 Autômatos

Autômatos é um formalismo utilizado para modelagem de SEDs. São dispositivos capazes de representar linguagens por meio de regras bem definidas (MARCOVISTZ, 2014). Segundo CASSANDRAS e LAFORTUNE (2008), um autômato determinístico é definido formalmente como se segue:

Definição 2.8. *(Autômato Determinístico) Um autômato determinístico G é uma sêxtupla:*

$$G = \{X, \Sigma, f, \Gamma, x_0, X_m\}, \qquad (2.10)$$

em que X é o espaço de estados, Σ *é o conjunto finito de eventos,* $f(X \times \Sigma) \rightarrow X$ *é a função de transição de estados,* $\Gamma: X \rightarrow 2^{\Sigma}$ *é a função dos eventos ativos (viáveis),* x_0 *é o estado inicial e* $X_m \subseteq X$ *é o conjunto de estados marcados.*

Graficamente, autômatos são construídos utilizando-se circunferências e arcos rotulados. As circunferências representam estados e, os arcos, eventos. Os estados (circunferências) são conectados por eventos (arcos) e, assim, formam um diagrama de transição de estados. Os estados marcados são identificados por duas circunferências concêntricas. Eles estão associados ao cumprimento de uma tarefa realizada pelo sistema modelado pelo autômato. O estado que recebe uma seta não oriunda de qualquer outro estado é o estado inicial (MARCOVISTZ, 2014).

Observação 2.1. *A função de transição de estados* f *é usualmente estendida com relação ao seu domínio, isto é,* $f : X \times \sum^* \rightarrow X$*, em que:*

$$\begin{gathered} f(x, \varepsilon) = x \\ f(x, se) = f(f(x, s), e), \forall\, s \in \Sigma^*, \forall\, e \in \sum \end{gathered} \qquad (2.11)$$

Observação 2.2. *Um autômato é denominado determinístico quando para todo estado* $x \in X$ *e para todo evento* $\sigma \in \Gamma$*, existe um único estado* $y \in X$ *tal que* $f(x, \sigma) = y$*. Note que, se um autômato for não-determinístico,* $f(x, \sigma)$ *poderá levar não só a um novo estado, mas sim a um conjunto de estados.*

Todas as sequências de eventos possíveis, partindo de um estado inicial, representam a linguagem gerada por um autômato (identificada por L). O subconjunto,

de L, constituído por sequências terminadas em um estado marcado, representa a linguagem marcada L_m de um autômato. (MARCOVISTZ, 2014). A definição formal destas linguagens é apresentada a seguir:

Definição 2.9. *(Linguagem gerada) A linguagem gerada por um autômato* $G = (X, \Sigma, f, \Gamma, x_0, X_m)$ *é definida como:*

$$L(G) = \{s \in \Sigma^* : f(x_0, s) \text{ é } definida\}, \qquad (2.12)$$

em que a linguagem gerada por G *é prefixo fechada, ou seja,* $L(G) = \overline{L(G)}$.

Definição 2.10. *(Linguagem marcada) A linguagem marcada por um autômato* $G = (X, \Sigma, f, \Gamma, x_0, X_m)$ *é definida como:*

$$L(G) = \{s \in \Sigma^* : f(x_0, s) \in X_m\} \qquad (2.13)$$

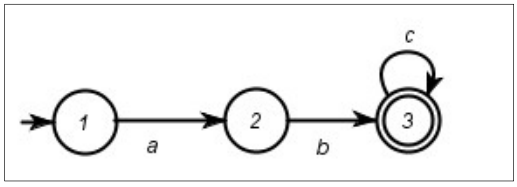

Figura 2.1 - Exemplo de um Autômato determinístico

O grafo da Figura 2.1, representa o diagrama de transição de estados de um autômato do tipo $G = (X, \Sigma, f, \Gamma, x_0, X_m)$, onde:

- $X = \{1,2,3\}$;
- $\sum = \{a, b, c\}$;
- $X_m = \{3\}$;
- $x_0 = 1$;
- $f(1, a) = 2$; $f(2, b) = 3$; $f(3, c) = 3$;
- $f\Gamma(1) = \{a\}$; $\Gamma(2) = \{b\}$; $\Gamma(3) = \{c\}$.

No autômato exibido na figura 2.1, há um auto-laço no estado 3. Auto-laços ocorrem quando existem eventos ativos, em um ou mais estados de um autômato, que não geram mudanças de estado, isto é, o estado de destino é o mesmo de origem. Esta condição ocorre no estado 3 do autômato da Figura 2.1. Por outro lado, quando um

estado não possui eventos ativos, o denominamos como um estado de bloqueio ou *deadlock*.

2.3 Autômatos Temporizados

Autômatos temporizados de tempo contínuo é um formalismo para modelagem e verificação de sistemas de tempo real. Exemplo de outros formalismos com este mesmo propósito são Redes de Petri temporizadas, Álgebra de processos temporizados, e Lógica de tempo real (BERTHOMIEU e DIAZ, 91; REED e ROSCO, 1988; YI, 1991; NICOLLIN e SIFAKIS, 1994; ALUR e HENZINGER, 1994; CHAOCHEN, 1999).

Na teoria original proposta por ALUR e DILL (1994) um autômato temporizado é um autômato de estado finito de Büchi estendido por um conjunto de variáveis de *clocks* reais e expressões booleanas que impõe restrições ao progresso de transições. HENZINGER *et al.*. (1994) criam uma variação do autômato temporizado introduzido por ALLUR e DILL (1994), denominada de Autômatos Seguros Temporizados (*Timed Safety Automata*) a qual especifica propriedades de progresso usando condições de invariante local.

Um autômato temporizado é essencialmente um autômato de estado finito (um grafo contendo um conjunto finito de nós ou lugares de controle e um conjunto finito de arcos rotulados) estendido com variáveis que assumem valores reais. Tal autômato pode ser considerado como um modelo abstrato de um sistema temporizado. Todos os relógios do sistema são inicializados por zero e evoluem sincronamente a uma mesma taxa. Restrições de *clock*, isto é, guardas nos arcos de transição, são usadas para restringir o comportamento do autômato. Uma transição representada por um arco pode ocorrer quando os valores de *clock* satisfazem a guarda que rotula o arco de transição. c*locks* podem ser zerados quando uma transição é realizada.

2.3.1 Sintaxe

A definição de autômato seguro temporizado, extraído de (BERHMANN *et al.*, 2006), é apresentada a seguir:

Um autômato temporizado é a tupla $\langle L, l_0, C, A, E, I \rangle$, onde L é o conjunto de lugares (estados), $l_0 \in L$ é a localização inicial, C é o conjunto de *clocks* de valores

reais, A é o conjunto de ações, co-ações e ações τ internas, $E \subseteq L \times A \times B(C) \times 2^{C} \times L$ é conjunto de arcos entre os lugares com uma ação, uma guarda e um conjunto de *clocks* a serem zerados, B(C) é o conjunto das restrições de *clock*, e I: N$\rightarrow B(C)$ associa invariantes aos lugares. As invariantes são restrições de tempos que podem ser associados a estados e tem por objetivo determinar o tempo máximo que o autômato pode permanecer nestes estados. As restrições de *clock*, guardas, são expressões booleanas com restrições atômicas do tipo $x \sim n \; ou \; x - y \sim n$, para $x, y \in C, \sim \in \{\leq, <, =, \geq, >\}$ e $n \in \mathbb{N}$.

2.3.2 Semântica

A semântica de um autômato temporizado é definida como um sistema de transição, onde um estado ou configuração consiste no lugar de controle corrente (l) e nos valores correntes dos relógios (u), isto é, no par (l, u). Existem dois tipos de transição entre estados: uma transição por ação, quando um arco habilitado é executado, e uma transição por tempo decorrido, quando o lugar de controle permanece o mesmo e o valor dos relógios é incrementado.

As regras de transição pertencentes a semântica de autômatos temporizados, estão definidas matematicamente abaixo (BENGTSSON e YI, 2004):

Transição por tempo decorrido: $(l, u) \xrightarrow{d} (l, u + d) \; se \; u \in I(l) \; e \; u + d \in I(l)$ para um real não negativo $d \in \mathbb{R}_{+}$.

Transição por ação:

$$(l, u) \xrightarrow{a} (l', u') \; se \; l \xrightarrow{g,a,r} l', u \in g, u' = [r \rightarrow 0]u \; e \; u' \in I(l'),$$

onde l e l'∈ L são lugares de controle; g ∈ B(C) é uma guarda associada a um arco; a ∈ A é uma ação; r é um subconjunto de C cujos relógios são zerados quando da realização da transição.

Considere o autômato seguro temporizado exibido na Figura 2.2. Este autômato possui apenas dois estados ou lugares, ligado e desligado, que estão conectados por também dois arcos. O autômato transitará do estado inicial (desligado) para o estado (ligado) quando a guarda (t ≥ 3) for satisfeita e retornará para o estado (desligado) assim que a invariante (t ≥ 5) for satisfeita. Neste exemplo tem-se a seguinte configuração:

L (conjunto de lugares) = {Desligado, Ligado};

l_0 (estado inicial) = {desligado}, indicado pela dupla circunferência;

C (conjunto de *clocks*) = {t};

Exemplo de transição por ação = $\{\text{Desligado}, \text{t} = 0\} \xrightarrow{t \geq 3} \{\text{Ligado}, \text{t=0}\}$;$\{\text{Ligado}, \text{t} \geq 5\} \rightarrow \{\text{Desligado, t=0}\}$

Exemplo de transição por tempo decorrido: {Desligado, t=0} $\xrightarrow{t=1}$ {Desligado, t=1}

I (invariantes) = t ≥ 5.

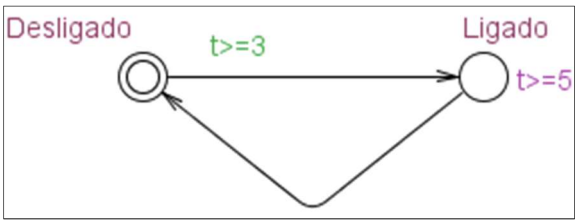

Figura 2.2 - Exemplo de autômato seguro temporizado.

2.3.3 Rede de Autômatos Temporizados

Autômatos temporizados são frequentemente compostos por uma rede de autômatos temporizados sobre um conjunto comum de *clock* e ações, consistindo de n autômatos temporizados $A_i = \langle L_i, l_{0_i}, C, A, E_i, I_i \rangle, 1 \leq i \leq n$. O vetor de locação é um vetor $\bar{l} = (l_1, \ldots, l_n)$. A função de invariante é composta por uma função comum sobre os vetores de lugares $I(\bar{l}) = \wedge_i I_i(l_i)$. Escreve-se $\bar{l}[l_i'/l_i]$ para denotar o vetor onde o i-ésimo elemento $l_i$de $\bar{l}$ é substituído por l_i'. A semântica de uma rede de autômatos temporizados é semelhante à de um único autômato. As definições matemáticas dos tipos de transições da rede podem ser consultadas em BENGTSSON e YI (2004).

2.4 UPPAAL

Autômatos temporizados de tempo contínuo é uma teoria para modelagem de sistemas de tempo real. Seguindo o trabalho de ALLUR e DILL (1994) e posteriormente HENZINGER *et al.* (1994) muitas ferramentas destinadas à verificação de modelos de tempo real foram desenvolvidas, como por exemplo, o UPPAAL.

A ferramenta é dividida em três partes principais: linguagem de descrição, simulador e verificação de modelo. A linguagem de descrição pode ser inserida tanto de forma gráfica (grafo) quanto textual. O seu núcleo é a rede de autômatos temporizados, onde cada autômato temporizado é um autômato seguro baseado no modelo proposto em HENZINGER *et al.*. (1994). Um autômato seguro temporizado no UPPAAL pode conter guardas formadas não somente por variáveis do tipo *clock*, mas também variáveis inteiras ou boleanas. Esta modificação aproxima a linguagem do UPPAAL a uma linguagem de programação de alto nível que lida com vários tipos de dados.

O ambiente de simulação permite ao usuário examinar o comportamento dinâmico de um sistema, ao contrário do verificador de modelo que explora todos os espaços de estados alcançáveis do sistema.

2.4.1 Sintaxe (UPPAAL)

Para exemplificar a sintaxe básica adotada pelo UPPAAL, considere o modelo exibido na Figura 2.3 extraído de LARSEN *et al.*., 1997.

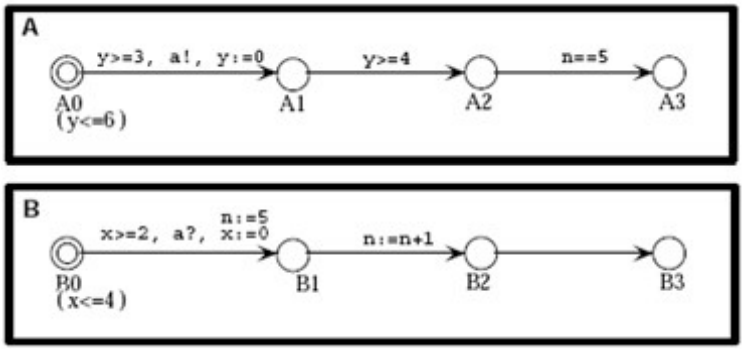

Figura 2.3 -Um exemplo de modelo UPPAAL (LARSEN *et al.*, 1997).

A rede exibida no modelo é composta por dois autômatos (processo) formado pelos seguintes conjuntos de estados $A = \{A_0, A_1, A_2, A_3\}$ e $B = \{B_0, B_1, B_2, B_3\}$. O sistema utiliza dois *clocks* x e y, uma variável inteira n e um canal a.

As transições do autômato são decoradas com três tipos de rótulos: uma guarda, uma ação de sincronização e resets de *clocks* ou atribuições a variáveis inteiras. A guarda expressa uma condição com valores de *clock* ou variáveis inteiras que precisa ser satisfeita para que a transição ocorra. A título de exemplo, na Figura 2.3, para que ocorra a transição entre A_0 e A_1 o valor do *clock* y tem que ser maior ou igual a 3. Similarmente, a transição entre A_2 e A_3 só pode ocorrer quando o valor da variável n for igual a 5. Na mesma figura, uma ação de sincronização é executada quando a transição entre A_0 e A_1 ocorre. No UPPAAL, um autômato pode se comunicar via variáveis inteiras ou por canais de sincronização. No modelo da Figura 2.3, os processos estão sincronizados e podem se comunicar através do canal a. A ação com notação $a!$ representa uma ação de envio e $a?$ de recebimento. A ausência de uma ação de sincronização indica uma transição τ interna. Quando uma transição ocorre *clocks* ou variáveis inteiras estão sujeitos a resets ou atribuições. Uma operação de reset em uma variável *clock* tem que ser do tipo $x := m$, onde m é um número natural. Operação de reset em uma variável inteira deve ser do tipo $i := c * i + c'$, onde c e c'são constantes inteiras ($c\ e\ c'$podem ser zero ou negativo). Na Figura 2.3, o *clock* y é resetado para zero quando a transição entre A_0 e A_1 é realizada. Da mesma forma, a variável inteira n é incrementada quando a transição entre B_1 e B_2 é executada. Todos os três tipos de rótulos são opcionais.

Os estados ou lugares também são rotulados com as chamadas invariantes, que são condições que impõe restrições às variáveis de *clocks* com o objetivo de controlar a permanência do autômato em um estado particular. Esta condição está indicada no estado A_0 da Figura 2.3, onde o autômato só pode permanecer neste estado enquanto y não é maior que 6.

2.4.2 Semântica (UPPAAL)

Formalmente, estados no UPPAAL são da forma $(\bar{l}, u)$, onde $\bar{l}$ é um vetor de controle indicando o nó atual de controle para cada componente da rede e u atribui um valor atual a cada *clock* e variável inteira. Seguindo a semântica de uma rede de

autômatos temporizados, o UPPAAL possui dois tipos de transição: por tempo decorrido e por ação. Os dois tipos de transição estão descritos a seguir:

Transição por tempo decorrido: Enquanto nenhuma invariante de um nó de controle no estado atual é violada, o tempo pode passar sem afetar o vetor de nó de controle e incrementar todas as variáveis de *clock* com a duração de tempo que foi executada. Na Figura 2.3, do estado inicial ((A_0, B_0); x = 0; y = 0; n = 0) o tempo pode passar 1,5 unidades levando ao estado ((A_0, B_0); x = 1,5; y = 1,5; n = 0). Entretanto, o tempo não pode passar de 5 unidades, pois isso violaria a invariante B_0. Em uma transição por tempo decorrido, o autômato pode mudar de estado sem necessariamente mudar de lugar de controle.

Transições por ação: Se dois rótulos de arcos de dois componentes diferentes estão habilitados em um estado, eles podem sincronizar. Sendo assim, no estado ((A_0, B_0); x = 3,5; y = 3,5; n = 0) os dois componentes podem sincronizar levando a um novo estado e lugar ((A_1, B_1); x = 0; y = 0; n = 5). Nesta transição os valores de x, y e n foram atualizados apropriadamente. Se um componente tem um arco interno habilitado, a transição pode ocorrer sem que haja qualquer sincronização. Então, no estado ((A_1, B_1); x = 0; y = 0; n = 5), a componente B pode realizar uma transição sem qualquer sincronização com a, levando ao estado ((A_1, B_2); x = 0; y = 0; n = 6).

2.5 Fundamentos de Diagnóstico de Faltas em SEDs

Como já foi mencionado, faltas causam desvios não desejados no comportamento esperado ou normal de um sistema ou em um de seus componentes. As faltas podem levar a um estado de falha que necessariamente precisará ser diagnosticado para que o sistema ou componente possa voltar ao seu estado normal.

Para diagnosticar faltas busca-se alcançar três objetivos: detecção, isolamento e identificação. Detecção da falta é uma funcionalidade que decide se um sistema está trabalhando em condições normais ou se uma falta ocorreu. Caso uma falta tenha ocorrido, busca-se seu isolamento para localizar o componente que causou a falta no sistema. Identificação da falta é dedicado a reconhecer a natureza específica da falta (tamanho, criticidade, importância, etc.). Detecção de faltas normalmente requer um modelo do comportamento normal do sistema, enquanto que isolação e identificação

requer um modelo do comportamento do sistema submetido às faltas consideradas (ZAYTOON e LAFORTUNE, 2013).

Faltas podem ser permanentes (abruptas ou progressivas) ou intermitentes. Faltas permanentes abruptas surgem repentinamente e mudam o comportamento do processo rapidamente, como por exemplo, a ruptura de uma correia em uma esteira transportadora. Faltas permanentes progressivas decorrem de pequenos desvios do comportamento normal do sistema, como por exemplo, uma pequena fuga de corrente que compromete gradativamente a performance de um motor. Faltas intermitentes se alternam no tempo, como por exemplo, o mal contato de um fio que pode fazer com que um equipamento ligue e desligue indesejadamente. O comportamento temporal destes tipos de faltas é exibido na Figura 2.4.

A classificação de faltas, com respeito ao componente do sistema que as originou, é a seguinte:

- Falta em sensores, tal como *offset* de sensores, sensor travado no estado ligado ou desligado que ocasionará discrepâncias entre os valores medidos e os valores reais das variáveis do sistema.

- Falta em atuadores, como por exemplo, atuador travado no estado ligado ou desligado, que terá como efeito a não operação devida do atuador sobre o processo.

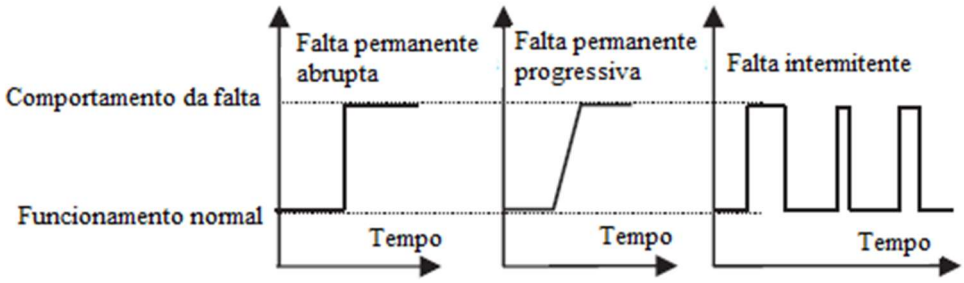

Figura 2.4 - Tipos de faltas (Adaptado de ZAYTOON e LAFORTUNE, 2013).

- Falta em plantas, tal como vazamento de tanques ou tubulações obstruídas, que resultam em mudanças no comportamento dinâmico do sistema.

- Falta na comunicação ou implementação de controle, devido a problemas de hardware ou software.

As definições e algoritmos propostos em SAMPATH *et al.* (1995, 1996) proveram os conceitos básicos e fundamentos formais de diagnóstico de faltas e análise de diagnosticabilidade de SEDs para a literatura dedicada a este tópico. A abordagem proposta por SAMPATH *et al.* (1995,1996) consiste em diagnosticar faltas, eventos não observáveis, a partir da observação de uma sequência de eventos. Para tanto, a ferramenta matemática desenvolvida para tratar este problema é o autômato observador ou diagnosticador que basicamente é um autômato determinístico que estima os estados do autômato global ou local e discrimina aqueles onde ocorreu falta, não ocorreu falta ou não há certeza da ocorrência de falta. Posteriormente, outros pesquisadores desenvolveram novos modelos, novos métodos, novos algoritmos e eficientes soluções para o diagnóstico de faltas em SEDs. Em ZAYTOON e LAFORTUNE (2013) é apresentado uma visão geral dos métodos de detecção de faltas em SEDs desenvolvidos até então. De acordo com este trabalho, os métodos de detecção de faltas podem ser classificados com respeito a diferentes critérios tais como compilação de faltas, ferramentas de modelagem, representação da falta, estrutura de decisão e arquitetura. A seguir será apresentado um resumo destas classificações e uma apresentação mais detalhada apenas para os tópicos que foram diretamente aplicados no presente trabalho.

2.5.1 Classificação de Métodos de Diagnóstico com Respeito a Compilação da Falta

Diagnóstico de faltas podem ser conseguidos usando um diagnosticador *off-line* ou um diagnosticador *on-line* (*On-the-fly*). No caso *off-line*, considera-se que o sistema a ser diagnosticado está em uma bancada de teste, isto é, não está ainda desempenhando sua função em um processo. O diagnosticador desejado é compilado baseado em possíveis cenários de operação do sistema. Compilação *off-line* do diagnosticador provê uma completa caracterização *off-line* do problema de diagnóstico. Entretanto, diagnóstico *off-line* requer um modelo completo e correto do modelo do sistema com faltas, que é difícil de ser alcançado em sistemas reais complexos, além de demandar enorme esforço computacional.

Outra abordagem consiste na detecção, isolamento e identificação *on-line* de um conjunto de faltas que podem ocorrer após cada nova observação do sistema durante sua operação. Cálculos computacionais complexos podem ser necessários para conseguir um diagnóstico *on-line*. Essa estratégia de diagnóstico requer maior demanda

computacional em termos de tempo de resposta, entretanto resulta em um considerável ganho de memória pois não é necessário armazenar a diagnose completa.

2.5.2 Classificação do Método de Diagnóstico com Respeito ao Formalismo de Modelagem

Muitos formalismos de modelagem foram usados para projetar diagnosticadores, incluindo:

Autômato (SAMPATH *et al.*, 1995) e suas extensões temporizadas e probabilísticas.

Redes de Petri (BASILE, CHIACCHIO e DE TOMMASI, 2008, 2009; CABASINO, GIUA e SEATZU, 2010; entre outros).

Statecharts e máquinas de estado hierárquicas (IDGHAMISHI e ZAD, 2004; PAOLI e LAFORTUNE, 2008).

2.5.2.1 Autômatos (Máquinas de Estados Finitos)

Seja um sistema a ser diagnosticado, modelado como um Autômato G (Máquina de Estado Finito) definido pela equação 2.10.

Suponha que o conjunto de eventos de G possa ser particionado em $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$, sendo Σ_o e Σ_{uo} conjuntos de eventos observáveis e não observáveis, respectivamente. Seja $\Sigma_f \subseteq \Sigma$, o conjunto de eventos de faltas que precisam ser diagnosticadas. Pode-se assumir, sem perda de generalidade que $\Sigma_f \subseteq \Sigma_{uo}$, caso contrário um evento de falta seria trivialmente diagnosticado. O conjunto de eventos de faltas pode ser particionado em conjuntos disjuntos de diferentes tipos de faltas:

$$\Sigma_f = \Sigma_{f1} \cup \ldots \cup \Sigma_{fm} \qquad (2.14)$$

As seguintes notações são importantes para compreensão das definições pertencentes aos teoremas sobre diagnosticabilidade:

Uma linguagem gerada por G é viva se $\Gamma(x_i) \neq 0$, para todo $x_i \in X$, i.e., o sistema não pode chegar a um ponto em que nenhum evento é possível.

$$\Psi(\Sigma_{fi}) = \{s\sigma_f \in L: \sigma_f \in \Sigma_{fi}\} \qquad (2.15)$$

i.e., $\Psi(\Sigma_{fi})$ denota o conjunto de todos os traços (sequências) de L que termina com um evento de falta pertencente a Σ_{fi}.

$$L/s = \{t \in \Sigma^*: st \in L\}, \qquad (2.16)$$

i.e., continuação da linguagem L após um traço s.

No artigo de SAMPATH *et al.*. (1995) são apresentados modelos de diagnosticadores de faltas baseados em eventos. Para este tipo de diagnose, diz-se que uma linguagem L é diagnosticável se é possível detectar com um delay finito a ocorrência de faltas de qualquer tipo usando registro de eventos observáveis. No mesmo artigo é apresentado duas definições de diagnosticabilidade. A primeira é conhecida como diagnosticabilidade e a segunda como I-diagnosticabilidade. A diagnosticabilidade é definida como:

Definição 2.11 *Uma linguagem viva e prefixo fechada L é dita diagnosticável com respeito à projeção P e com respeito à partição Π_f de Σ_f se a seguinte condição for verificada:*

$$(\forall i \in \Pi_f)(\exists n_i \in \mathbb{N})[\forall s \in \Psi(\Sigma_{fi})](\forall t \in \mathrm{L/s})\left[\|t\| \geq n_i \Rightarrow D\right],$$

onde a condição de diagnosticabilidade D é

$$\omega \in P_L^{-1}[P(st)] \Rightarrow \Sigma_{fi} \in \omega.$$

A condição de Diagnosticabilidade D implica que todo evento de falta, pertencente a um traço gerado pelo sistema, seja detectado por eventos observáveis distintos o suficiente para permitir identificar unicamente o tipo da falta que ocorreu com delay finito. A condição de I-diagnosticabilidade, adotada no presente trabalho, não requer que a condição de diagnosticabilidade D se mantenha para todos os traços que contenha um evento de falta, mas somente para aqueles em que um evento de falta é seguido por um evento observável indicador do tipo de falta. A definição de I-diagnosticabilidade conforme SAMPATH *et al.*. (1995), é apresentada abaixo:

Definição 2.12 *Uma linguagem viva e prefixo fechada L é dita I-diagnosticável com respeito a projeção P e com respeito a partição Π_f de Σ_f, e o indicador I se a seguinte condição for verificada:*

$$\left(\forall i \in \Pi_f\right)(\exists n_i \in \mathbb{N})\left[\forall s \in \Psi\left(\Sigma_{fi}\right)\right]\left(\forall t_1 t_2 \in L/s:\ st_1 \in \Psi\left[I(\Sigma_{fi})\right]\right)\left[\|t_2\| \geq n_i \Rightarrow D\right],$$

onde a condição de diagnosticabilidade D é

$$\omega \in P_L^{-1}[P(st_1t_2)] \Rightarrow \Sigma_{fi} \in \omega.$$

Para o caso da I-diagnosticabilidade é requerido que a ocorrência de eventos de falta do tipo F_i sejam detectados em até n_i transições do sistema após a ocorrência de um evento indicador do conjunto $\mathrm{I}(\Sigma_{fi})$. As faltas do tipo F_i são identificadas por um conjunto de rótulos de faltas $\Delta_f = \{F_1, F_2, \ldots, F_m\}$ após eventos observáveis identificados por rótulos do conjunto $\Delta_I = \{I_1, I_2, \ldots, I_m\}$. O conjunto completo dos possíveis rótulos é:

$$\Delta^I = \{N\} \cup 2^{\Delta_f \cup \Delta_i} \qquad (2.33)$$

com a restrição $(\forall l \in \Delta^I)\ I_i \in l \Rightarrow F_i \in l$.

Finalmente, o diagnosticador adequado a I-diagnosticabilidade é dado por:

$$G_d^I = \left(Q_d^I,\ \Sigma_0, \sigma_d^I, q_0\right), \qquad (2.34)$$

onde Q_d^I é o espaço de estados, Σ_0 é o conjunto de eventos, σ_d^I é a função de transição e $q_0 = \{(x_0, \{N\})\}$ é o estado inicial rotulado pela função de propagação $LP^I(x, l, s)$, definida por:

$$LP^I(x, \ell, s) = \begin{cases} \{N\} & \textbf{Se } \ell = \{N\} \wedge \forall i[\Sigma_{fi} \notin s] \\ \{F_i: F_i \in \ell \vee \Sigma_{fi} \in s\} & \\ \quad \bigcup\{I_i: I_i \in \ell \vee [I(\Sigma_{fi}) & \\ \quad \in s \wedge (F_i \in \ell \vee \Sigma_{fi} \in s)]\} & \end{cases}$$

onde $x \in X_0, l \in \Delta^I, e\ s \in L_0(G, x)$.

Os estados q_d de G_d^I são da forma $q_d = \{(x_1, l_1), \ldots, (x_n, l_n)\}$ e os rótulos I_i *e* F_i se propagam de estado para estado.

A condição necessária e suficiente para que a linguagem L seja I-diagnosticável, apresentada em SAMPATH. *et al.*. (1995), é verificar se em L existe dois traços s_1 e s_2 de comprimento arbitrariamente longo, tal que ambos possuem a mesma projeção observável, e s_1 contém um evento de falta do conjunto Σ_{fi} seguido por um evento indicador do conjunto I(Σ_{fi}) enquanto s_2 não contém qualquer evento do conjunto Σ_{fi}, conforme apresentado no Lema 3 em SAMPATH *et al.*. (1995).

2.5.2.2 Diagnóstico de autômatos temporizados e probabilísticos

Métodos de diagnósticos para sistemas temporizados baseados em autômatos temporizados foram propostos por BOUYER *et al.* (2005); CASSEZ (2009); CHEN e PROVAN (1997); JIANG e KUMAR (2006); TRIPAKIS (2002); ZAD, KWONG, e WONHAM (2005), entre outros. Essas contribuições foram principalmente baseadas na definição de diagnosticabilidade temporizada, que requer que a condição de diagnosticabilidade seja válida após um intervalo limitado de tempo, ao invés de um número limitado de eventos.

No presente trabalho é aplicado o diagnosticador e a verificação de diagnosticabilidade apresentados em TRIPAKIS (2002). Segundo o referido autor, em SEDs onde não há a variável *clock* ou tempo associado a estados ou eventos, a diagnosticabilidade é definida com respeito a um parâmetro n que é o máximo *delay* requerido para o diagnosticador detectar a falta. Este *delay* corresponde a uma certa quantidade de eventos observáveis que ocorrem após o evento de falta. Para o caso de um autômato temporizado, em TRIPAKIS (2002) é definido um *delay* Δ que corresponde a um intervalo de tempo requerido para que uma falta seja detectada após a sua ocorrência. A definição de diagnosticabilidade, proposta por TRIPAKIS (2002), é a seguinte:

Definição 2.13 *Considere um autômato temporizado A. Nós dizemos que A é* $\Delta-$ *diagnosticável para um número natural* $\Delta \in \mathbb{N}$*, se para qualquer duas sequências finitas* ρ_1, ρ_2*, de A, se* ρ_1 *possui uma* $\Delta-$*,falta então* ρ_2 *possui uma falta ou* $P(\rho_1, \Sigma_o) \neq P(\rho_2, \Sigma_o)$*. Nós dizemos que A é diagnosticável se existe algum* $\Delta \in \mathbb{N}$*,tal que A é* Δ - diagnosticável.

Na definição 2.13, ρ_1 e ρ_2 representam sequências temporais. Uma sequência temporal é válida se entre dois quaisquer eventos há somente um único *delay*, inclusive

0, pertencente a $\mathbb{R}$. Por exemplo, se a, b e c são eventos, a, 0, b, 3, c é uma sequência válida. Considere o seguinte exemplo, apresentado em TRIPAKIS (2002):

Assumindo que os eventos *a* e *b* são observáveis, *f* e *u* são não observáveis e *f* é a falta, o autômato temporizado da Figura 2.5 é 3-diagnosticável. Por outro lado, o autômato ligeiramente modificado da Figura 2.6 não é diagnosticável. De fato, as duas sequências temporais *a*, 2.5, *f*, 0.1, *b* e *a*, 2.5, *u*, 0.1, *b* têm a mesma projeção *a*, 2.6, *b*, mas somente na primeira ocorreu uma falta. Além disto, um intervalo arbitrário de tempo passou após *b* em ambas sequências temporais, e as suas projeções permaneceram idênticas.

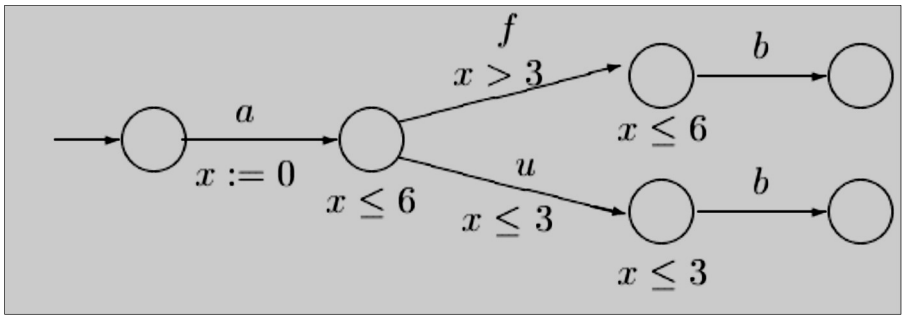

Figura 2.5 – Autômato temporizado diagnosticável (TRIPAKIS, 2002).

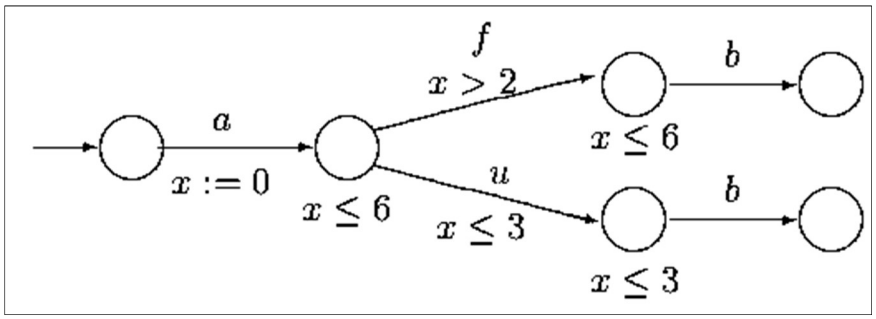

Figura 2.6 – Autômato temporizado não diagnosticável (TRIPAKIS, 2002).

No mesmo Artigo, é apresentado um modelo de diagnosticador que atua como estimador de estados. Por este motivo, antes de apresentar o diagnosticador, é apresentado um método para dividir os estados do autômato em estados de falha e de não falha. Por exemplo, considere um autômato A e seus estados discretos Q. O conjunto Q pode ser particionado em dois conjuntos disjuntos de estados discretos: $Q = Q_f \cup (Q - Q_f)$, tal que qualquer sequência temporal de A que contenha uma falta estará

contida em Q_f. Sendo assim, uma vez que uma falta ocorra, o autômato A se movimentará para o conjunto Q_f e não mais sairá deste conjunto de estados discretos, enquanto não houver falta, A se movimentará dentro de $Q - Q_f$. Qualquer autômato pode ser transformado em um autômato que satisfaça a condição descrita acima. A Figura 2.7 demonstra um exemplo desta transformação. O diagnosticador proposto por TRIPAKIS (2002), é apresentado na definição 2.14.

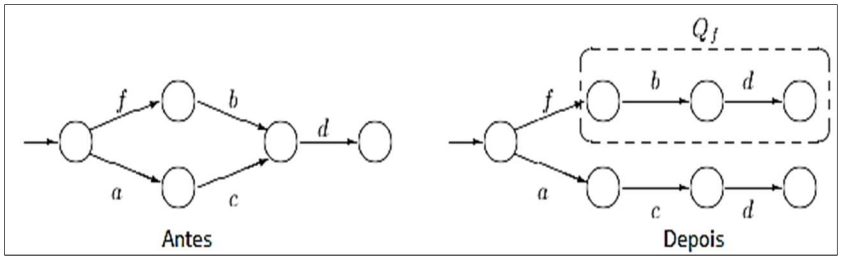

Figura 2.7 Transformando um autômato (TRIPAKIS, 2002).

Definição 2.14 *Seja um autômato temporizado A sobre* Σ *com conjuntos de eventos observáveis e não observáveis, isto é,* Σ_o *e* Σ_u *respectivamente. Considerando que* $\Sigma_u \subseteq \Sigma, e\ \Delta \in \mathbb{N}$, *um* Δ*-diagnosticador para A é a função:*

$$D: FTS_o \longrightarrow \{0,1\}$$

Tal que:

1. *Para qualquer sequência temporal finita (sem falta)* ρ *de A,* $D(P(\rho, \Sigma_u)) = 0$.
2. *Para qualquer sequência temporal finita* $\Delta - falt \quad \rho$ *de A,* $D(P(\rho, \Sigma_u)) = 1$.

Se um $\Delta -$*diagnosticador existe para A então dizemos que A é* $\Delta -$*diagnosticável.*

2.5.2.3 Diagnóstico com Redes de Petri

O objetivo dos métodos de diagnóstico baseados em Redes de Petri é usar a estrutura, as capacidades analíticas e a natureza intrinsicamente distribuída dos modelos em Redes de Petri para reduzir a complexidade computacional dos problemas de diagnóstico pela eliminação de uma exaustiva enumeração de espaço de estados do sistema, assim como lidar com algumas classes de sistemas de estados infinitos (linguagens não regulares). Alguns métodos de diagnósticos baseados em Redes de Petri consideram que a marcação de certos lugares é observável (CHUNG, 2005; GHAZEL, BIGAND, e TOGUYÉNI, 2005; HERNANDEZ-FLORES, LOPEZ-

MELLADO, e RAMIREZ-TREVINO, 2011); enquanto que outros se baseiam não em uma rede de marcações observáveis, mas em um conjunto observável de transições (BASILE, CHIACCHIO, e DETOMMASI, 2009; BENVENISTE, FABRE, HAAR, e JARD, 2003; CABASINO, GIUA, POSSI, e SEATZU, 2011).

2.5.3 Diagnóstico utilizando modelos que incluem o comportamento de falta

Modelos com faltas são baseados em diferentes tipos de representação de faltas, tal como a execução de um evento (diagnóstico baseado em eventos), alcançando um estado de falta (diagnóstico baseado em estados), a execução de um padrão de supervisão, ou a verificação de restrições temporais parciais.

Diagnóstico baseado em eventos, tal como a abordagem de SAMPATH *et al.* (1995), decide se uma falta ocorreu e o seu tipo baseado somente na execução de uma sequência de eventos. Métodos de diagnóstico baseado em eventos podem ser usados para diagnosticar faltas intermitentes pois consideram faltas como ocorrências de eventos não observáveis (CONTANT, LAFORTUNE, e TENEKETZIS, 2004). Nestas abordagens é necessário que o diagnosticador seja inicializado junto com o modelo do sistema porque o diagnosticador toma suas decisões com base em sequências de eventos observáveis. Esta condição de inicialização simultânea não é fácil de ser realizada em sistemas reais, que pode resultar na necessidade de incluir eventos não observáveis adicionais no estágio de modelagem.

Diagnóstico baseado em estados tem como base a partição do espaço de estados do sistema de acordo com o status de falta. As abordagens propostas por ZAD, KWONG, e WONHAM (2003) e LIN (1994) são relacionadas a sistemas com entradas e saídas binárias. Cada estado é rotulado com um vetor binário de suas saídas associadas e o diagnosticador usa a sequência dos vetores de saída binária associada com estados do sistema para detectar e isolar faltas. O método de diagnose baseado em estados é adequado para detecção de faltas permanentes, pois consideram que a falta alcançará um estado de falha. Entretanto, esses métodos não são apropriados para detecção de faltas intermitentes. Uma vez que o diagnosticador baseado em estado pode determinar a ocorrência de um modo de falta tendo como referência uma saída gerada por um estado, nenhuma informação sobre o estado ou sobre o status de falta do sistema é requerido antes da inicialização do diagnosticador. Portanto, a vantagem do diagnóstico

baseado em estado é que não é necessário que o diagnosticador seja inicializado junto com o sistema.

Uma falta pode também ser representada como a execução de um padrão de supervisão, que é uma propriedade temporal relacionada com a ocorrência de um conjunto de trajetórias/eventos que precisam ser diagnosticados (JÉRON, MARCHAND, PINCHINAT, e CORDIER, 2006). A noção de padrão de supervisão é genérica o bastante para cobrir uma importante classe de objetivos de diagnose, incluindo detecção de faltas permanentes, faltas incipientes, múltiplas faltas, assim como também uma complexa sequência de eventos.

2.5.4 Diagnose usando modelos livres de faltas

Diagnose com modelos livres de faltas são baseadas na comparação das saídas do sistema com a saída nominal do modelo. A falta é detectada se um comportamento observado do sistema não pode ser reproduzido pelo seu modelo. Entretanto, isolação de falta e identificação pode não ser possível neste caso porque o modelo não inclui o comportamento de falta e, portanto, a diagnosticabilidade de uma dada falta não é garantida. A modelagem baseada na abordagem livre de faltas proposta por PANDALAI e HOLLOWAY (2000) usa modelos (*templates*) de condição para determinar se um sistema gera eventos na ordem correta ou dentro de um intervalo de tempo pré-definido. Uma falta é detectada quando existem reações erradas ou perdidas no processo. Nesses casos, os eventos relacionados com o modelo (*template*) ajudam a isolar a falta.

2.5.5 Classificação de métodos de diagnóstico com relação à estrutura de decisão

Três estruturas principais de processamento, ou arquiteturas, são usadas para calcular a decisão do diagnóstico de falta: centralizada, descentralizada (BOELEVAN SCHUPPEN, 2002; CHAKIB e KHOUMSI, 2012; DEBOUK, LAFORTUNE, e TENEKETZIS, 2000; QIU e KUMAR, 2006; etc.) e distribuída (FABRE *et al.*, 2005; GENC e LAFORTUNE, 2007). A distinção entre a descentralizada e a distribuída é algumas vezes confusa. Em termos gerais, a abordagem descentralizada possui um conjunto de diagnosticadores, cada um com diferentes capacidades de observação, mas todos considerando o modelo global do sistema em seu modelo de inferência. Na

abordagem distribuída, os diagnosticadores individuais somente usam modelos dos sistemas locais, ao contrário do modelo de sistema global.

2.5.5.1 Diagnose Centralizada

Na estrutura centralizada, o diagnóstico é calculado usando um diagnosticador (monolítico) global, que é construído usando o modelo global do sistema a ser diagnosticado. A vantagem principal do diagnóstico centralizado é seu diagnóstico preciso e sua simplicidade conceitual. Entretanto, sua principal desvantagem é sua complexidade computacional proibitiva, uma vez que requer o modelo centralizado da planta para gerar o diagnosticador centralizado. Os modelos resultantes podem se tornar muito grandes para serem fisicamente armazenados quando um sistema de larga escala é considerado.

2.5.5.2 Diagnose Descentralizada com Coordenação

Na estrutura descentralizada, o sistema é particionado em um número de lugares. Cada lugar (*site*) conhece o modelo inteiro do sistema, tem observações locais, e usa um diagnosticador local que computa uma decisão de diagnóstico local baseada em sua observação parcial do sistema inteiro. Um coordenador provê a decisão final de diagnóstico como uma função das decisões de diagnóstico local que são comunicadas para ele. Os diagnosticadores locais e o coordenador são construídos usando o modelo global do sistema. Os diagnosticadores locais podem não se comunicar diretamente um com o outro, e usualmente somente uma comunicação limitada entre eles e o coordenador é permitida. O principal problema para enfrentar na arquitetura descentralizada é sobre como os lugares (*sites*) podem, em conjunto, descobrir a ocorrência de uma falta, sabendo que a informação disponível pode ser ambígua, incompleta, atrasada, e possivelmente errônea. O coordenador deve, portanto, ter alguma memória e capacidade de processamento para coordenar a requerida troca de informação entre os diagnosticadores locais para resolver ambiguidades das decisões locais. Entretanto, essas capacidades devem ser limitadas, senão, a estrutura centralizada pode ser replicada no lugar (*site*) do coordenador através da comunicação de todas as observações para ele, o que invalidaria o propósito da estrutura descentralizada.

2.5.5.3 Diagnose Distribuída

Abordagens distribuídas (BENVENISTE *et al.*, 2003; PANDALAI e HOLLOWAY, 2000; PENCOLÉ e SUBIAS, 2009; SU e WONHAM, 2004) conseguem

diagnósticos usando um conjunto de modelos locais sem se referir a um modelo global. O objetivo é melhorar a escalabilidade e a robustez das metodologias de diagnóstico. Cada subsistema sabe somente sua parte do modelo global e possui seu diagnosticador local com o objetivo de realizar a diagnose localmente. Essa computação de diagnóstico é baseada no modelo local e na informação comunicada diretamente para ela pelos diagnosticadores locais através de um protocolo de comunicação. A troca de informação entre diagnosticadores locais é usada para atualizar sua própria informação e compensar a observação parcial. Um protocolo de comunicação deve ser definido para garantir consistência entre diagnosticadores locais no caso de conflito entre suas decisões. O desafio da diagnose distribuída é como realizar diagnose local que seja equivalente, se possível, ao modelo global, usando um protocolo de comunicação escalável, e sem precisar usar o modelo global.

3 Estudo de Caso

3.1 Sistema Modular de Produção - MPS (Festo)

O Sistema Modular de Produção ("*Modular Production System*" MPS®) da empresa Festo, representa um ambiente didático de simulação de uma linha de produção industrial, que contempla os mais comuns sub-processos de um sistema de produção automatizado: Distribuição, Separação, Processamento, Montagem, Classificação, Robô, dentre outras. Cada sub-processo ou célula de produção é representada por um módulo ou estação. As estações possuem quatro partes: Planta, Painel de Controle, gabinete móvel e controlador. A Figura 3.1 exibe uma das estações e sua estrutura dividida em quatro partes.

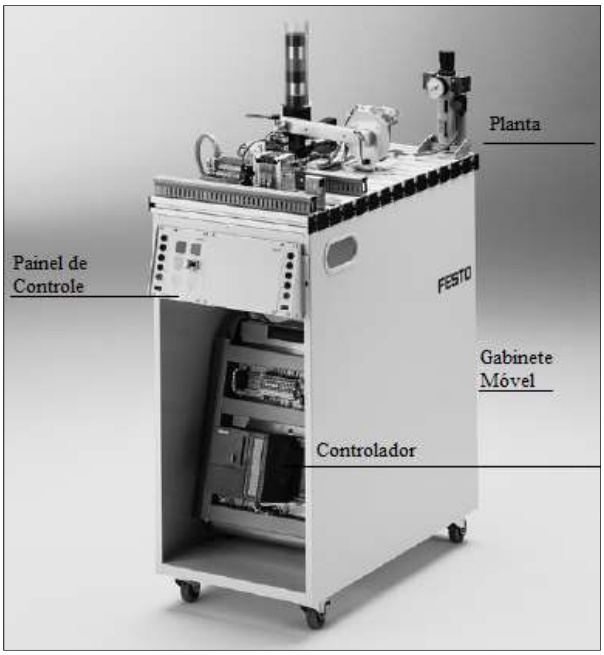

Figura 3.1 - Estrutura das estações (adaptado de Festo, 2006).

O MPS utilizado como estudo de caso pertence ao curso de Engenharia de Produção da UFS (Universidade Federal de Sergipe). O sistema é composto por um total de 8 estações: Distribuição, Separação, Classificação, Coleta e encaixe, Músculo fluído, Classificação, Processamento, Robô manipulador e Armazenamento, onde são produzidas 2 famílias distintas de produtos: cilindros; e relógios, termômetros e higrômetros. O *layout* que exibe o arranjo das estações necessário para produzir as duas famílias de produtos pode ser visto na Figura 3.2.

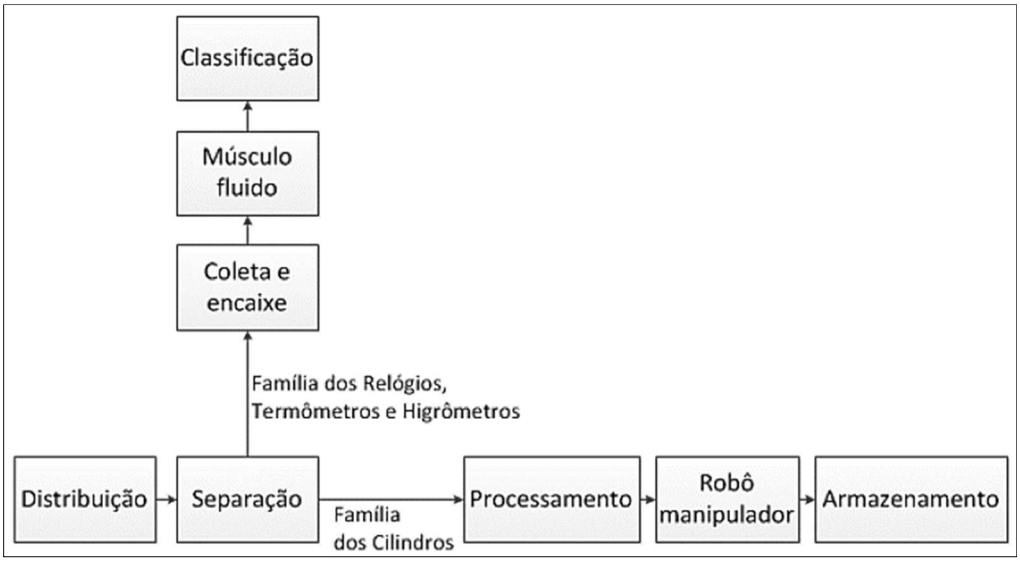

Figura 3.2- Layouts de produção.

No presente trabalho, as estações pertencentes ao *layout* de produção dos relógios, termômetros e higrômetros serão utilizadas como estudo de caso para modelagem e diagnóstico de faltas. A seguir as estações Distribuição, Separação, Coleta e encaixe, Músculo fluido e Classificação serão apresentadas.

3.1.1 Estação de Distribuição

A estação de Distribuição é um dispositivo do tipo alimentador, os quais são definidos como unidades que realizam a função de abastecimento e alimentação dos componentes ou peças a serem processadas. A estação de Distribuição é formada por duas partes: um cilindro de armazenamento com um atuador linear acoplado, denominado *stack magazine*, e um módulo atuador giratório ou braço rotativo, denominado *changer module*. A função do *stack magazine* é de armazenar até 8 peças e disponibilizá-las para serem transportadas pelo braço rotativo (*changer module*). A presença de peças no *stack magazine* é monitorada por um sensor de barreira. A separação das peças é feita por um cilindro de dupla ação (atuador linear) que empurra ou retira a peça de posição mais inferior da *stack magazine* para uma posição de espera. Dois sensores indutivos de fim de curso monitoram a posição do cilindro de dupla ação. O braço rotativo com um succionador a vácuo, irá capturar a peça que se encontra na posição de espera "*magazine*" e levá-la para a posição de "próxima estação" que, no caso atual, corresponde ao início da esteira da estação de Separação. O braço rotativo é capaz de realizar um movimento de até 180°. Suas posições de fim de curso são

monitoradas por chaves e o movimento de transporte da peça somente é iniciado após uma chave, que se encontra no dispositivo succionador, detectar que a peça foi realmente capturada. A Figura 3.3 exibe a estação completa de Distribuição, a Figura 3.4 exibe a planta da estação, a Figura 3.5 exibe o dispositivo *stack magazine* e a Figura 3.6 o braço rotativo (*changer module*).

Figura 3.3- Estação de Distribuição (Festo didatic, 2006).

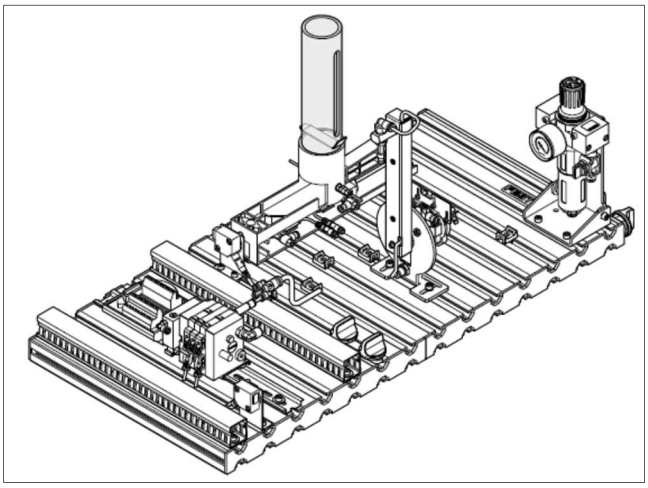

Figura 3.4- Planta da estação de Distribuição (Festo didatic, 2006).

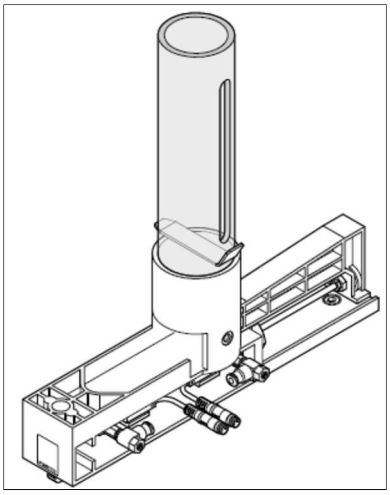

Figura3.5- *Stack magazine* (Festo didatic, 2006).

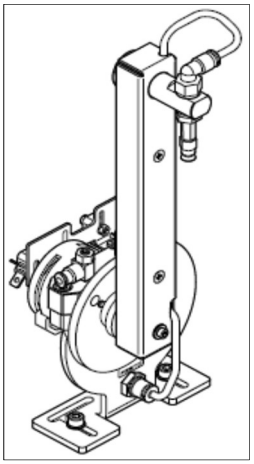

Figura 3.6- Braço rotativo (*changer module*) (Festo didatic, 2006).

As modelagens de todas as estações foram baseadas em suas sequências de funcionamento e, em alguns casos, informações adicionais sobre a dinâmica temporal de funcionamento foram extraídas da programação Grafcet da planta. Tanto as sequências de funcionamento quanto algumas programações estão presentes nos manuais individuais das estações. Todas as estações possuem pré-requisitos para que entrem em funcionamento. Além disto, toda estação possui uma posição inicial que representa a configuração adequada dos dispositivos para que possam entrar em funcionamento seguindo a sequência correta de operação. Caso a estação esteja inicialmente em um estado que não represente a posição inicial correta para a operação, o controlador da planta solicitará que o usuário reinicie a planta apertando o botão *reset* situado no painel de controle. Após apertar o botão *reset*, o controlador enviará comandos para a planta e uma sequência de operações será executada para que a mesma

atinja a posição inicial. Uma vez na posição inicial, a planta inicia sua operação assim que o botão *start* é pressionado. A sequência de funcionamento, os pré-requisitos e a posição inicial da planta da estação de distribuição estão descritos abaixo:

Sequência de funcionamento:

1. O braço rotativo se movimentará para a posição "próxima estação" se as peças de trabalho são identificadas no *stack magazine* e o botão *Start* é pressionado.
2. O atuador linear (*stack magazine*) empurra uma peça para fora do *magazine*.
3. O atuador linear recua. Obs.: Na sequência de funcionamento descrita no manual, o braço rotativo gira para a posição de espera "*magazine*" antes do atuador linear recuar. Entretanto, seguindo esta sequência haveria uma colisão entre o braço rotativo e o atuador linear. Sendo assim, a sequência correta, considerada na modelagem, é recuar o atuador linear antes do braço rotativo girar para a posição de espera "*magazine*".
4. O braço rotativo gira para a posição "*magazine*" (posição para coleta da peça que foi empurrada do *stack magazine*).
5. O vácuo é ligado assim que o sensor (fim de curso) detectar que o braço está na posição *magazine*. Quando a peça é capturada o sensor de detecção de peça presente no succionador mudará para o estado ligado.
6. Após o sensor de detecção de peça detectar que a peça foi capturada, o braço rotativo gira para a posição de "próxima estação".
7. O vácuo é desligado assim que o sensor de fim de curso "próxima estação" detectar a presença do braço rotativo.
8. Quando o sensor de peça detectada mudar para o estado desligado (peça liberada), o braço rotativo irá girar para a posição "*magazine*". Obs.: Esta ação será realizada caso a planta esteja no modo de funcionamento automático ou o usuário queira que a planta continue em operação. Caso contrário, o processo termina assim que a peça é liberada na posição "próxima estação".

Pré-requisitos:

1. Haver peças no *stack magazine*.

Posição inicial:

1. Atuador linear recuado.
2. Braço rotativo na posição "*magazine*".

3.1.2 Estação de Separação

A estação de Separação realiza as funções de manipulação de mudanças de quantidade e verificação. Isto significa que ela aloca e separa quantidades de diferentes partes de acordo com as informações adquiridas (atual) em comparação com as características especificadas (requerido). A estação de separação separa o fluxo de material das peças de trabalho corpo (família dos cilindros) e invólucro (família dos relógios, termômetros e higrômetros), distribuindo estas peças em duas esteiras diferentes. A estação de separação é formada por três módulos funcionais: diferenciação, esteiras e desvio. O módulo de diferenciação realiza a detecção das peças por meio de um dispositivo sensitivo fotoelétrico de distância. O sensor difuso deste módulo, tem opção de saída analógica ou digital e é responsável por diferenciar as peças das duas diferentes famílias. O módulo de esteiras é utilizado para realizar o transporte e distribuição das peças de trabalho detectadas, onde um sensor difuso detecta se existe uma peça de trabalho disponível no início da esteira, fazendo com que a mesma ligue. A peça inserida é parada por meio de um anteparo giratório (módulo de desvio). A identificação e diferenciação das peças acontecem por meio de um sensor difuso no módulo de diferenciação. Dependendo da classificação um braço para separação de peças (módulo de desvio) é ativado, ou seja, avança. As posições do módulo de desvio são monitoradas por um sensor de fim de curso. Uma vez que a peça é liberada pelo anteparo, ela é transportada para a outra esteira ou para o final da esteira atual. As peças de trabalho no final das esteiras são detectadas por meio de sensores óticos de proximidade (sensores de barreira). As esteiras são ativadas por motores de corrente contínua. A Figura 3.7 exibe a estação completa de Separação, a Figura 3.8 exibe a planta da estação, a Figura 3.9 exibe o Módulo de diferenciação, a Figura 3.10 o Módulo de esteiras e a figura 3.11 o Módulo de desvio.

Figura 3.7- Estação de Separação (Festo didatic, 2006).

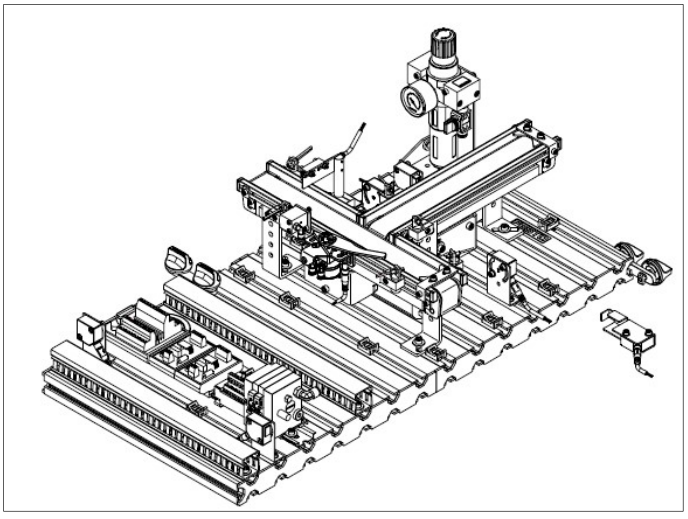

Figura 3.8- Planta da estação de Separação (Festo didatic, 2006).

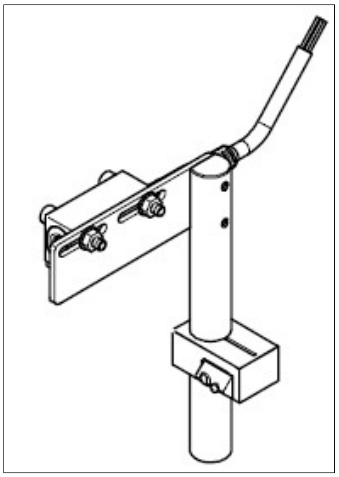

Figura 3.9- Módulo de diferenciação (Festo didatic, 2006).

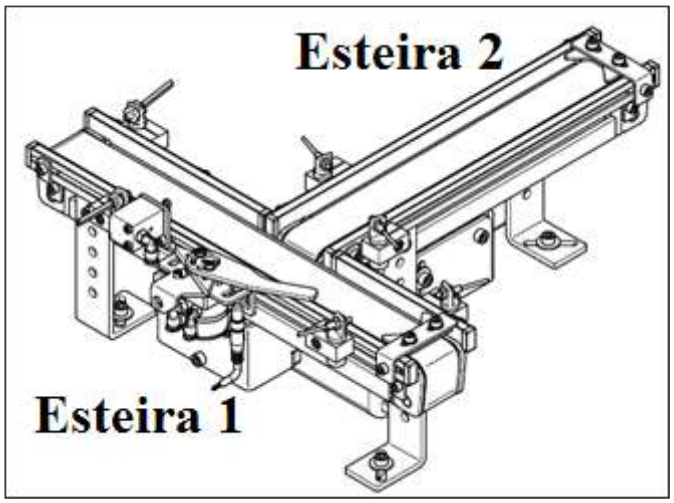

Figura 3.10- Módulo de esteiras da estação de separação (Festo didatic, 2006).

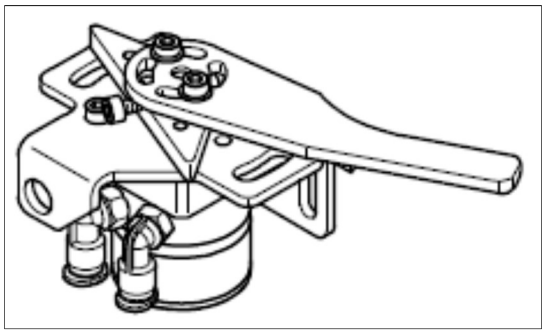

Figura 3.11- Módulo de desvio (anteparo giratório) (Festo didatic, 2006).

A sequência de funcionamento, os pré-requisitos e a posição inicial da planta da estação de Separação estão descritos abaixo:

Sequência de funcionamento:

1. O motor da esteira 1 é ligado se uma peça é detectada pelo sensor localizado no início da esteira e o botão *start* é pressionado. A peça é transportada até o anteparo (atuador giratório), onde está o módulo de desvio e o módulo de diferenciação.
2. O motor da esteira 1 é desligado se o sensor difuso, localizado na parte frontal do anteparo (atuador giratório), detecta a presença de uma peça. Neste momento o sensor difuso avalia as dimensões da peça e realiza a diferenciação. Caso a peça detectada possua as dimensões esperadas para o corpo da família dos cilindros, a peça será transportada até o final da esteira 1 (passos 3 e 4). Caso a peça possua as dimensões do invólucro da família dos relógios, termômetros e higrômetros, a peça será transportada para a esteira 2 (passos 5, 6 e 7).
3. O anteparo (atuador giratório) retrai e o motor da esteira 1 é ligado se o sensor do final da esteira estiver desligado, indicando que não há peça no ponto de coleta.
4. O motor da esteira 1 é desligado e o anteparo (atuador giratório) avança assim que o corpo do cilindro é detectado pelo sensor do final da esteira (ponto de coleta) por pelo menos 2s.
5. O anteparo (atuador giratório) retrai, o braço (módulo de desvio) avança e o motor da esteira 1 é ligado.
6. O braço (módulo de desvio) retrai, o anteparo (atuador giratório) avança, o motor da esteira 1 desliga se o sensor do início da esteira 2 detecta a presença do invólucro.
7. O motor da esteira 2 é ligado assim que o braço é recuado e o anteparo é retraído.
8. O motor da esteira 2 é desligado se o sensor do final da esteira (ponto de coleta) detecta a presença do invólucro por pelo menos 4s.

Pré-requisitos:

1. Não haver peça no início da esteira 1.

Posição inicial:

1. Anteparo (atuador giratório) avançado.
2. Braço (módulo de desvio) retraído.
3. Motores das esteiras desligados.

3.1.3 Estação de Coleta e Encaixe

A estação Coleta e Encaixe – *Pick & Place* – é um dispositivo de inserção automatizado que basicamente encaixa uma peça em outra. Os tipos de peças disponíveis na estação são: relógio, termômetro e higrômetro. Esta estação possui três partes importantes: uma esteira transportadora, acionada por um motor DC; um deslizador (módulo deslizador), que alimenta as peças a serem encaixadas; e por último, um manipulador (módulo de coleta e encaixe), controlado inteiramente por atuadores eletropneumáticos. A esteira transportará a peça (invólucro) desde início da estação até o local onde o manipulador encaixará a parte móvel (*insert*), neste local (anteparo separador) é acionada uma trava que prenderá a peça maior (invólucro) até que a parte móvel (*insert*) seja encaixada pelo manipulador. O tipo de controle do manipulador *Pick & Place* são os pistões pneumáticos. As peças sairão apenas encaixadas sendo necessária uma força adicional dada pela estação conhecida como músculo fluido, que finalizará a montagem. A Figura 3.12 exibe a estação completa de Coleta e Encaixe, a Figura 3.13 exibe a planta da estação, a Figura 3.14 exibe o Módulo de Coleta e Encaixe, a Figura 3.15 o Módulo da esteira e a figura 3.16 o Módulo deslizador.

Figura 3.12- Estação de Coleta e encaixe (Festo didatic, 2006).

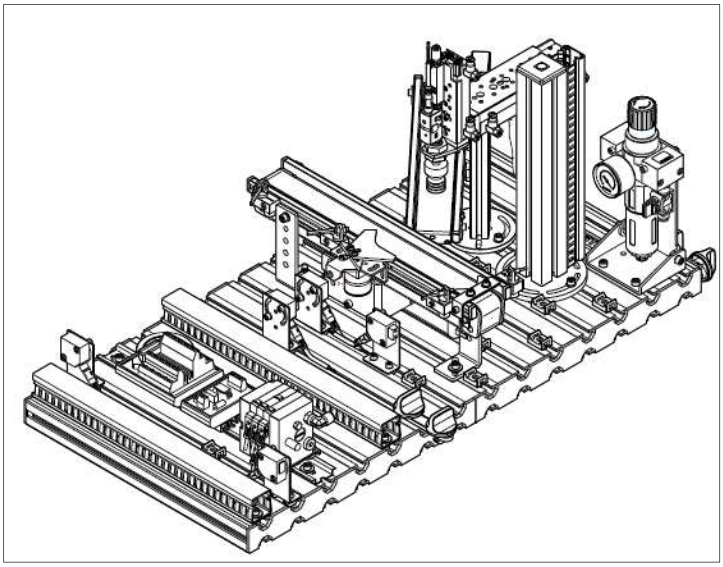

Figura 3.13 - Planta da estação de Coleta e encaixe (Festo didatic, 2006).

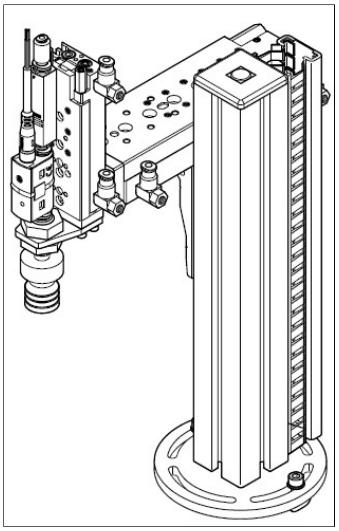

Figura 3.14- Módulo de coleta e encaixe (Festo didatic, 2006).

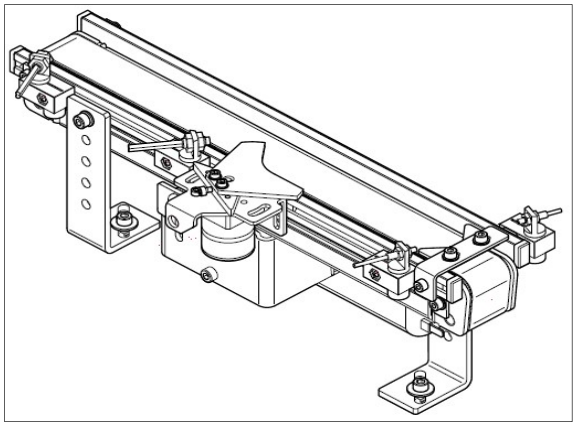

Figura 3.15- Módulo da esteira da estação de coleta e encaixe (Festo didatic, 2006).

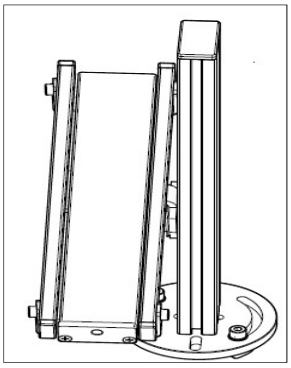

Figura 3.16- Módulo deslizador da estação de coleta e encaixe (Festo didatic, 2006).

A sequência de funcionamento, os pré-requisitos e a posição inicial da planta da estação de Coleta e encaixe estão descritos abaixo:

Sequência de funcionamento:

1. O motor da esteira é ligado se uma peça (invólucro) é detectada no início da esteira e o botão *start* é pressionado. A peça é transportada até o anteparo separador.
2. O motor da esteira é desligado se o sensor difuso, localizado na parte frontal do anteparo separador, detecta a presença de uma peça.
3. O mini-deslizador (módulo coleta e encaixe) movimenta-se para baixo e liga o vácuo assim que o sensor de fim de curso (baixo) detectar que ele está na posição de captura de peça.
4. O sensor de detecção de peça, do módulo de coleta e encaixe, mudará para o estado ligado caso a peça (*insert*) tenha sido capturada. Após capturada a peça, o mini-deslizador (módulo de coleta e encaixe) movimenta-se para cima e assim que o sensor de fim de curso (alto) detectar que o mine-slide está na posição alta ele avança (estende).
5. O mine-slide (módulo de coleta e encaixe) movimenta-se para baixo assim que o sensor de fim de curso (estendido) detectar a sua presença.
6. O vácuo é desligado, liberando o *insert* sobre o invólucro, assim que o sensor de fim de curso (baixo) detectar que o mine-slide está na posição baixa.
7. O mine-slide (módulo de coleta e encaixe) movimenta-se para cima assim que o sensor de peça capturada muda para o estado desligado.
8. O mine-slide (módulo de coleta e encaixe) recua (retrai) assim que o sensor de fim de curso (alto) detectar a sua presença.
9. O anteparo separador retrai e assim que o sensor de fim de curso " anteparo separador retraído" mudar para o estado ligado, o motor da esteira é ligado.
10. O motor da esteira é desligado e o anteparo separador é estendido assim que o sensor de fim de curso da esteira "posição próxima estação" detectar a presença da peça. Fim do ciclo.

Pré-requisitos:

1. Não haver peça no início da esteira.

Posição inicial:

1. Anteparo separador estendido.
2. Motor da esteira desligado.
3. Mine-slide (módulo de coleta e encaixe) na posição alta.
4. Mine-slide (módulo de coleta e encaixe) na posição recuada.
5. Vácuo desligado.

3.1.4 Estação de Músculo fluido

O Músculo Fluido funciona como uma prensa pneumática. A função da estação do Músculo fluido é a de prensar peças nos seus devidos encaixes e transferi-las para a próxima estação. No caso desta planta, as peças podem ser relógios, termômetros ou higrômetros. É importante salientar que as peças, *insert* e invólucro, já vêm acopladas desde a estação anterior (Coleta e encaixe). Mas, a união definitiva ocorre com a prensagem na estação de Músculo fluido. Esta estação é composta pelos módulos de prensa pneumática (músculo fluido) e de deslocamento (rotativo e linear) que possui uma garra para segurar as peças. Inicialmente o módulo rotativo linear coleta a peça, fechando a garra, e a transporta (giro de 90°) até a posição de prensagem. O módulo músculo fluido prensa o *insert* no invólucro. Um sensor de pressão gera uma saída binária quando a pressão de *set* tiver sido atingida. Depois o conjunto é transportado (giro de 90°) para a posição de "próxima estação". O atuador linear rotativo se estende depois libera a peça, abrindo a garra, para a próxima estação.

A pressão fornecida para o módulo Músculo fluido é medida por um sensor de pressão analógico e exibida em um *display*. A pressão de fornecimento, a velocidade e profundidade da prensa pode ser ajusta mecanicamente, através da válvula reguladora de pressão e da válvula de controle de fluxo de ar, respectivamente. Tanto a pressão como a velocidade podem também ser ajustados eletronicamente através do regulador proporcional de pressão. A Figura 3.17 exibe a estação completa Músculo fluido, a Figura 3.18 exibe a planta da estação, a Figura 3.19 exibe o Módulo de Músculo fluido e a Figura 3.20 o Módulo rotativo e linear.

Figura 3.17- Estação Músculo fluido (Festo didatic, 2006).

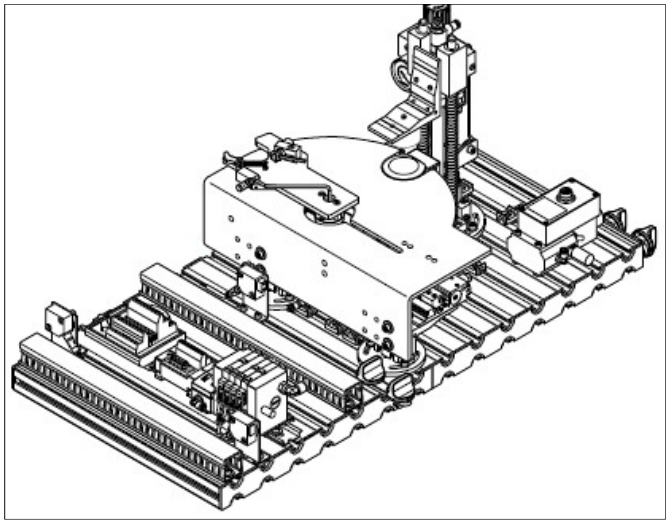

Figura 3.18- Planta da estação Músculo fluido (Festo didatic, 2006).

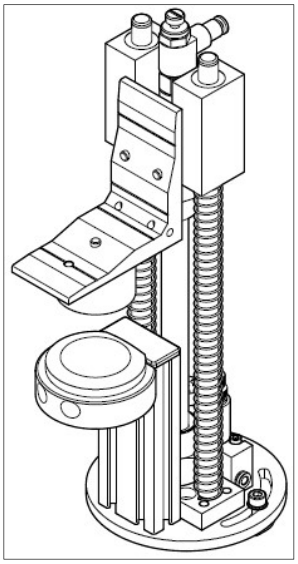

Figura 3.19- Módulo de músculo fluido (Festo didatic, 2006).

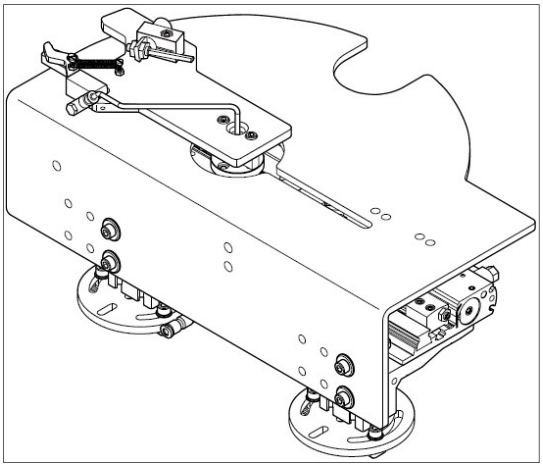

Figura 3.20- Módulo rotativo e linear da estação Músculo fluido (Festo didatic, 2006).

A sequência de funcionamento, os pré-requisitos e a posição inicial da planta da estação Músculo fluido estão descritos abaixo:

Sequência de funcionamento:

1. Se uma peça é detectada pelo sensor da garra, pertencente ao módulo rotativo e linear, e o botão *start* é pressionado, a garra fecha segurando a peça.
2. O módulo rotativo linear gira 90° transportando a peça para a posição de processamento.

3. Assim que o sensor de fim de curso detecta que a peça se encontra na posição de processamento, o módulo músculo fluido na posição alta (sensor de pressão indicando saída 0) desce e prensa a peça.
4. 1 segundo após a prensa atingir a pressão pré-ajustada de prensagem (sensor de pressão gera "1" como saída), o módulo músculo fluido sobe e o módulo rotativo linear gira mais 90° transportando a peça para a posição de "próxima estação". Obs.: O manual não considera a saída binária do sensor de pressão em sua sequência de funcionamento e programação Grafcet. A modelagem realizada no presente trabalho, considerou a saída do sensor de pressão como um recurso adicional para verificar se o conjunto insert e invólucro foram prensados conforme especificação requerida pelo usuário (*set point*) e para ter uma referência de posição do módulo músculo fluido (levantado ou abaixado). Além disto, ao incluir o sensor de pressão na modelagem é possível também verificar uma possível falta neste dispositivo.
5. Assim que o sensor de fim de curso detecta que a peça se encontra na posição "próxima estação", o módulo rotativo linear se estende.
6. Assim que o sensor de fim de curso do módulo rotativo linear detectar que o mesmo está na posição estendida, a garra é aberta e a peça liberada para a próxima estação.
7. Quando o sensor da garra detectar que não há mais peça na garra, o módulo rotativo linear recua.
8. Após o sensor de fim de curso detectar que o módulo rotativo linear está recuado, o mesmo gira 180° no sentido de retornar à posição de coleta de peça.
9. O sensor de fim de curso detecta que o módulo rotativo linear retornou para a posição inicial. Fim do ciclo.

Pré-requisitos:

1. Nenhuma peça presente na garra do módulo rotativo linear.

Posição inicial:

1. Módulo rotativo linear retraído.
2. Módulo rotativo linear na posição de coleta de peças.

3. Módulo músculo fluido levantado.

3.1.5 Estação de Classificação

A estação Classificação é composta pelos módulos de transporte e de escorregadores. Ela possui a função de classificar as peças através de três escorregadores de acordo com as características das peças (preta, não preta e metálica). Um sensor difuso detecta as peças no início da esteira depois estas são detectadas por sensores na frente do atuador que funciona como ponto de parada para avaliação. Após esta constatação, as peças classificadas são direcionadas para os escorregadores apropriados via atuadores que operam como separadores. Por fim, um sensor retro reflexivo monitora o nível de enchimento dos escorregadores. A Figura 3.21 exibe a estação completa de Classificação, a Figura 3.22 exibe a planta da estação, a Figura 3.23 exibe o Módulo de esteira e a Figura 3.24 o Módulo de escorregadores.

Figura 3.21- Estação Classificação (Festo didatic, 2006).

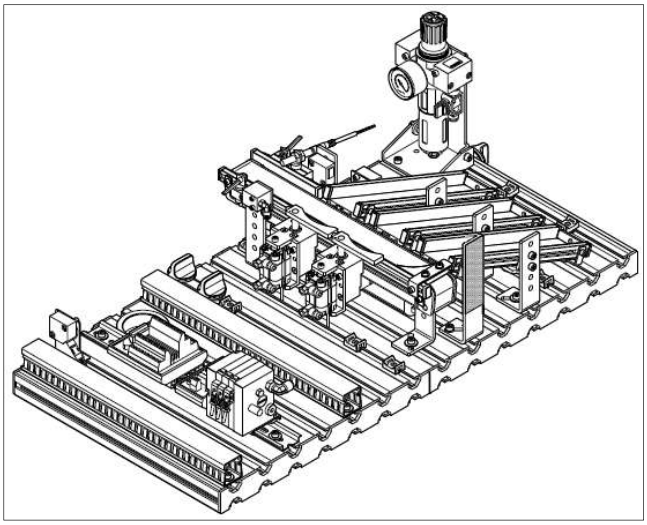

Figura 3.22- Planta Classificação (Festo didatic, 2006).

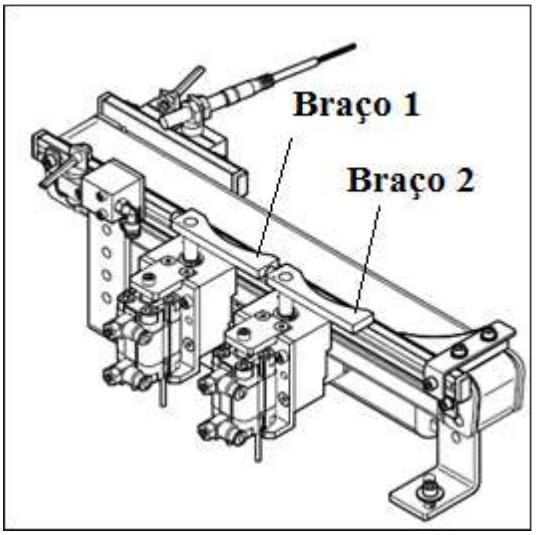

Figura 3.23- Módulo de esteira (Festo didatic, 2006).

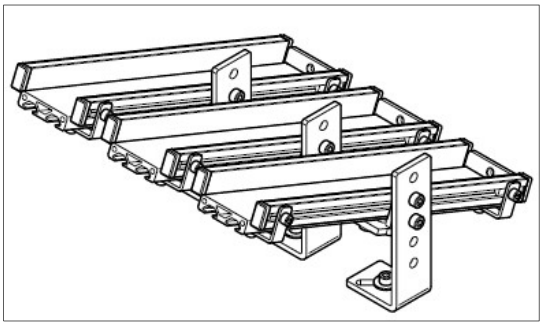

Figura 3.24- Módulo de escorregadores (Festo didatic, 2006).

A sequência de funcionamento, os pré-requisitos e a posição inicial da planta da estação Classificação estão descritos abaixo:

Sequência de funcionamento:

1. Se o escorregador de peças pretas, o escorregador de peças pratas e o escorregador de peças vermelhas não estiverem cheios; assim como também, se uma peça é detectada pelo sensor do início da esteira e o botão start é pressionado, o motor da esteira é ligado.
2. A peça alcança o anteparo. Permanece em avaliação pelos sensores por 2s.
3. Se a peça for preta, o anteparo é retraído. A peça segue até o final da esteira e depois alcança o escorregador de peças pretas. Após 3s, o anteparo é estendido e o motor da esteira desligado. Fim do ciclo.
4. Se a peça for vermelha, o braço 1 é avançado e o anteparo é retraído. A peça segue até alcançar o escorregador de peças vermelhas. Após 3s, o anteparo é estendido, o braço 1 é retraído e o motor da esteira desligado. Fim do ciclo.
5. Se a peça for prata, o braço 2 é avançado e o anteparo é retraído. A peça segue até alcançar o escorregador de peças pratas. Após 3s, o anteparo é estendido, o braço 2 é retraído e o motor da esteira desligado. Fim do ciclo.

Pré-requisitos:

1. Peça no início da esteira.

Posição inicial:

1. Braço 1 retraído.
2. Braço 2 retraído.
3. Anteparo estendido.
4. Motor da esteira desligado.

4 Metodologia

4.1 Modelagem do processo

As estações apresentadas no capítulo 3, subprocessos típicos de sistemas de manufatura, seguem uma sequência de operação para cumprir a sua função no sistema de produção (planta). Conforme foi apresentado, as operações são executadas essencialmente por atuadores elétricos ou eletropneumáticos que têm suas posições (estados) monitoradas por sensores usualmente indutivos ou óticos. Os sensores, assim como os atuadores, são dispositivos que possuem somente os estados discretos (ligado ou desligado). Estes estados são informados para o controlador lógico da estação. A função do controlador é de coordenar as ações, direcionando comandos para os atuadores e observando os estados dos sensores. Os estados observados podem não estar conforme o esperado, caracterizando uma falta. Esta resultará na interrupção da sequência de funcionamento do processo, isto é, ocasionará uma falha.

Considera-se que em sensores e atuadores discretos encontrados em processos de manufatura, há apenas dois tipos de falhas: travado no estado ligado ou travado no estado desligado. Estas falhas podem ser manifestadas de modo abrupto e permanente ou intermitente.

Antes de abordar os métodos de diagnóstico de falhas utilizados nesta pesquisa, serão apresentados os passos e premissas empregados para modelagem do processo das estações, exceto a parte relacionada ao diagnóstico de falhas, na linguagem do software UPPAAL:

1. Consulta do manual da estação para obter a sequência de funcionamento do processo da estação. As sequências de funcionamento das estações foram abordadas no Capítulo 3. Como já foi mencionado, em alguns casos, quando disponível, a linguagem Grafcet da planta foi consultada para obter detalhes da dinâmica temporal de algumas operações. Por exemplo, na estação de Coleta e encaixe, após apertado o botão de partida da planta o motor da esteira somente será ligado 3 segundos após o sensor de peça, localizado no início da esteira, detectar a presença de uma peça. Este *delay* de 3 segundos não é informado na sequência de funcionamento do manual.

2. Transformação da sequência de funcionamento diretamente em um autômato temporizado. Para exemplificar como uma expressão, contendo sequências de operações, podem ser transformadas em um autômato, considere a sequência de funcionamento da estação de distribuição, exibida abaixo:

   Exemplo 2.1: O braço rotativo se movimentará para a posição "próxima estação" se as peças de trabalho são identificadas no *stack magazine* e o botão *star*t é pressionado. A operação que está sendo realizada é o movimento do braço para a próxima estação, que representa um comando do controlador lógico para um atuador (braço rotativo). Entretanto, para que ocorra esta operação é necessário que as condições botão s*tart* pressionado e peça detectada no *stack magazine* tenham sido satisfeitas, caso contrário, o controlador não irá enviar o comando para o braço rotativo se movimentar.

   A dinâmica do autômato que representa o processo evolui conforme a habilitação de transições. Estas transições, que são representados por arcos na linguagem de autômatos, estão diretamente relacionados aos estados dos dispositivos que compõem a planta, isto é, sensores e atuadores. Os estados ou lugares, na linguagem de autômatos temporizados, representam o fim ou início de uma transição. Diferente dos arcos ou transições, que devem estar associados aos estados dos sensores e atuadores, os estados ou lugares podem ser rotulados conforme critério definido pelo projetista durante a modelagem. Neste trabalho, os estados foram rotulados com o critério de apenas indicar a quantidade de etapas que compõe o funcionamento do processo e a ocorrência de uma falha ou não. O autômato temporizado, escrito na linguagem adotada pelo UPPAAL, referente a operação descrita acima pode ser visto na Figura 4.1.

Figura 4.1- Construção do autômato.

Conforme pode ser visto na Figura 4.1, o autômato mudará do estado E1 para o estado E2 caso a condição ou guarda “SP==1” seja satisfeita. “SP==1” representa a condição: botão *start* pressionado (SP), onde “==1” representa a verificação da condição ligado. Em seguida é verificado se há peça no *stack magazine*, isto é, se o sensor de peça está habilitado “Speca==1”. Caso esteja, a ação mover braço para a próxima estação será habilitada “MBpe=1”. A expressão “MBpe=1” indica a atribuição do valor 1 (ligar ou habilitar) a variável inteira MBpe, caso a expressão de guarda “Speca==1” seja satisfeita. A condição “SP==1” *start* pressionado, em todas as estações, indica o início do ciclo de funcionamento da estação e por este motivo foi verificada antes da condição de existência de peça “Speca==1”. Seguindo este procedimento é possível construir todo o autômato do processo, seguindo a sua sequência de funcionamento.

3. Declaração de todas as variáveis inteiras que representam os estados dos sensores e atuadores. Nas expressões de guarda, o valor lógico 1 representa o estado ligado ou habilitado e o valor lógico 0 representa o estado desligado ou desabilitado.
4. As expressões de guarda foram formadas pelas variáveis que representam os estados dos sensores e restrições de tempo para a ocorrência da transição. A verificação do estado ligado de um sensor é do tipo “==1” e do estado desligado “==0”. Os termos “&” e “or” são utilizados nas expressões de guarda para representar os operadores booleanos “e” e “ou”. A habilitação de uma guarda pode ocasionar o disparo de uma ação (evento) sincronizante, uma atribuição a variável inteira (comando para um atuador), o *reset* de variáveis inteiras ou do tipo *clock* ou uma combinação destes elementos. Há arcos que que contêm comandos para atuadores sem a presença de guardas.

4.2 Métodos de diagnóstico de falhas

Os métodos de detecção ou diagnóstico de falhas que serão apresentados nas próximas seções são aplicáveis a qualquer sistema semelhante ao da Figura 4.2. Esta figura exibe uma operação genérica, comum em processos de manufatura, que envolve um par de sensores de fim de curso e um atuador, ambos possuem somente os estados discretos (ligado ou desligado).

Na Figura 4.2, o sensor S1 está inicialmente ligado e, o sensor S2, desligado. O controlador envia um comando para o atuador mover-se para P2. O atuador alcançará P2 em algum tempo situado entre 0,2 e 0,3 segundos, depois de ter recebido o comando. Assim que o atuador sai de P1, o sensor S1 muda seu estado para desligado. Após o atuador alcançar P2, o sensor S2 muda seu estado para ligado. Os eventos que representam os possíveis estados e ações desta operação estão indicados abaixo:

- **m_At_$P_{1,2}$** (mover atuador de P1 para P2).
- **S1_D** (sensor S1 desligado).
- **S2_L** (sensor S2 ligado).
- **S2_D** (sensor S2 desligado).
- **S1_L** (sensor S1 ligado).
- **f1** (atuador travado na posição P1).
- **f2** (sensor S1 travado no estado ligado).
- **f3** (sensor S2 travado no estado desligado).

Há 3 falhas que podem ocorrer nesta operação. Estas falhas podem ser detectadas, observando o estado dos sensores após transcorrido o tempo esperado para conclusão da operação, que é no máximo 0,3 segundos. A primeira falha, f1, indica que o atuador não respondeu ao sinal do controlador, isto é, permaneceu na posição P1. Neste caso, o sensor S1 permaneceu ligado enquanto que S2 permaneceu desligado. A segunda falha, que será indicada como f2, representa o sensor S1 travado no estado ligado, mesmo após o atuador ter saído da posição P1. A terceira falha, f3, representa o sensor S2 travado no estado desligado, pois o atuador alcançou a posição P2 e, o sensor S2 não mudou para o estado ligado.

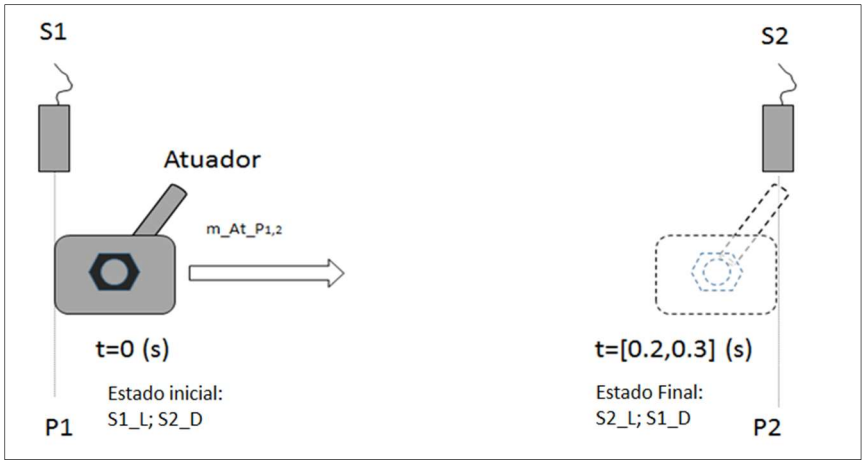

Figura 4.2- Sistema típico de processos de manufatura

4.2.1 Diagnóstico de falhas utilizando autômatos não temporizados

Para modelar a operação e duas das possíveis falhas descritas anteriormente, considere o autômato G, exibido na Figura 4.3, constituído pelos elementos abaixo:

- Conjunto de estados: X= {1, 2, 3, 4, 5, 6, 7, 8};
- Conjunto de eventos: $\Sigma = \{m\_At\_P_{1,2}, S1\_D, S2\_D, S1\_L, S2\_D, f1, f3\}$, onde o conjunto de eventos observáveis é $\Sigma_o = \{m\_At\_P_{1,2}, S1\_D, S2\_D, S1\_L, S2\_D\}$ e o conjunto de falhas é $\Sigma_f = \{f1, f3\}$, com $\Sigma_f \subseteq \Sigma_{uo}$(eventos não observáveis);
- Estado marcado: $X_m = \{4\}$ e estado inicial $x_0 = \{1\}$.

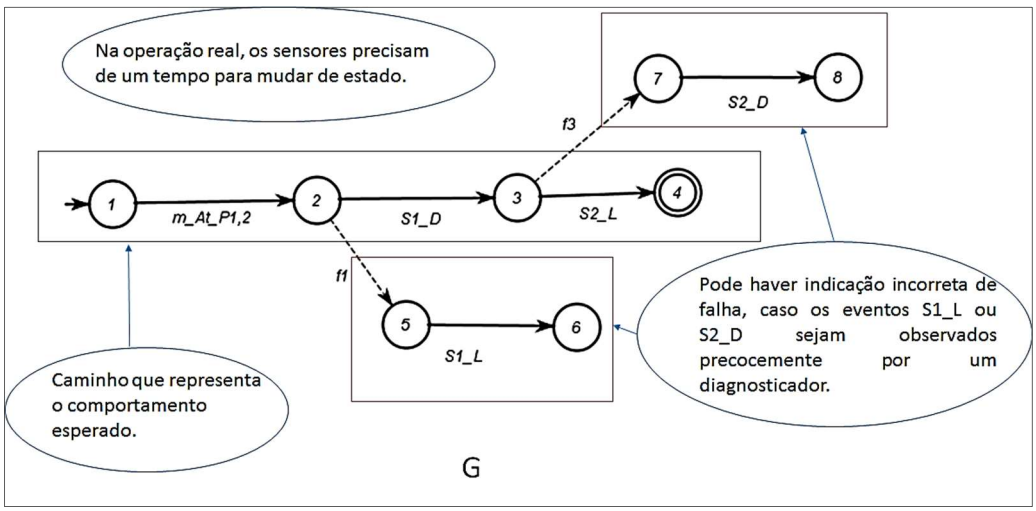

Figura 4.3- Possível estratégia para diagnóstico de falhas utilizando autômato não temporizado.

O evento que conecta o estado 1 ao estado 2, m_At_$P_{1,2}$, representa o comando enviado pelo controlador para o atuador mover de P1 para P2. O evento que conecta o estado 2 ao estado 3 indica o desligamento do sensor S1, após o atuador sair do seu campo de detecção. Este é o comportamento esperado para S1, considerando que o atuador respondeu ao comando e está, de fato, se movendo. O evento que conecta o estado 3 ao estado 4 representa o fim da operação, uma vez que o atuador alcançou a posição P2 fazendo com que o sensor S2 ligasse.

Se os sensores se comportarem diferente do que foi descrito anteriormente, significa que houve alguma falha. Por exemplo, caso o atuador esteja travado na posição P1, ele não responderá ao sinal m_At_$P_{1,2}$. Esta condição pode ser inferida fazendo a leitura do sensor S1, após o evento m_At_$P_{1,2,}$ para verificar se está ligado. Entretanto, como não há uma variável do tipo *clock* para definir um atraso na leitura do sensor S1, o evento que estará ativo e será disparado é S1_L. Isto acontece porque S1_L é o estado inicial do sensor S1. Para que o evento S1_D torne-se ativo e dispare, é necessário aguardar um tempo, uma vez que o atuador precisa de um período para se deslocar e sair do campo de detecção de S1. Sendo assim, caso esta estratégia de detecção de falha fosse implementada no sistema real, através de um autômato diagnosticador, haveria falsos anúncios de falhas sendo gerados para o supervisório. O mesmo vale para a detecção da falha f3. Neste caso, S2_D será sempre o primeiro evento ativo no estado 3, pois é o estado inicial do sensor S2.

A análise apresentada revela que autômatos não temporizados seriam inadequados para serem aplicados como diagnosticadores, considerando o método de diagnóstico de falhas proposto nesta pesquisa.

4.2.2 Método de detecção de falhas utilizando o Diagnosticador proposto por TRIPAKIS (2002)

Em TRIPAKIS (2002), é proposto um método para diagnosticar falhas, utilizando autômatos temporizados de tempo contínuo. Conforme já apresentado na seção 2.5.2.2, o diagnosticador proposto por TRIPAKIS (2002) atua como um estimador de estados do autômato que representa o processo. Desta forma, é necessário que o autômato do processo esteja dividido em dois conjuntos de estados: conjunto de estados de falhas e conjunto de estados de não falha. O diagnosticador, apresentado pela definição 2.14, é uma função que gera o valor 1 toda vez que observa um atraso

característico de um evento de falha. Como consequência, é possível inferir que o autômato do processo alcançou algum estado pertencente ao conjunto de estados de falhas.

Para exemplificar como ocorre a detecção de falhas, observando atrasos entre transições, considere o autômato exibido na Figura 4.4. Este autômato representa a modelagem do sistema exibido na Figura 4.2. Além dos eventos apresentados nas seções precedentes, há uma variável *clock* t nas expressões de guarda que restringem as transições e inclui a informação da dinâmica temporal da operação. Conforme se pode observar, a sequência temporal normal de operação é dada por: m_At_$P_{1,2}$, 0.1, S1_D, 0.2, S2_L. Esta sequência indica que o sensor S1 desliga no máximo 0.1 segundos depois do comando m_At_$P_{1,2}$ e, o sensor S2 liga em até 0.2 segundos depois do mesmo comando. Quando há um atraso de 0.3 segundos, e os sensores ainda não mudaram de estado, considera-se que houve uma falha (evento não observável). Somente 0.5 segundos e 0.8 segundos após m_At_$P_{1,2}$, os sensores S1 e S2 mudam de estado respectivamente. A sequência que contém o evento de falha (f) possui atrasos muito superiores aos da sequência normal. O diagnosticador observa estes atrasos e anuncia a ocorrência da falha.

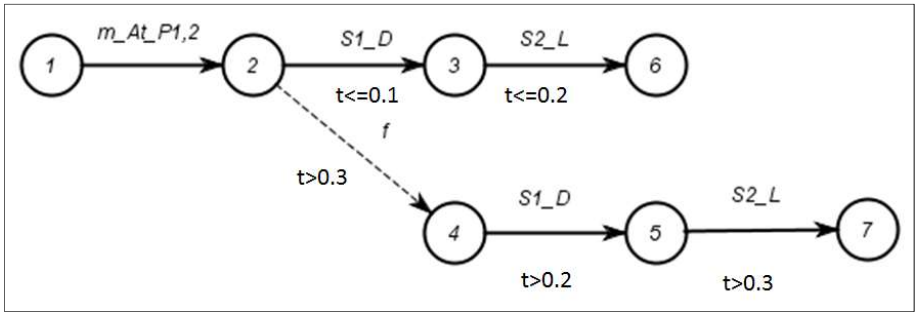

Figura 4.4- Autômato temporizado para exemplificar método de detecção de falhas baseado nos atrasos das transições do sistema.

No presente trabalho, o método de diagnóstico de falha proposto por TRIPAKIS (2002) é aplicado usando o software UPPAAL. O autômato temporizado, escrito na linguagem do UPPAAL, que representa o sistema da Figura 4.2, está exibido na Figura 4.5. O diagnosticador, baseado no método de TRIPAKIS (2002), está exibido na Figura 4.6.

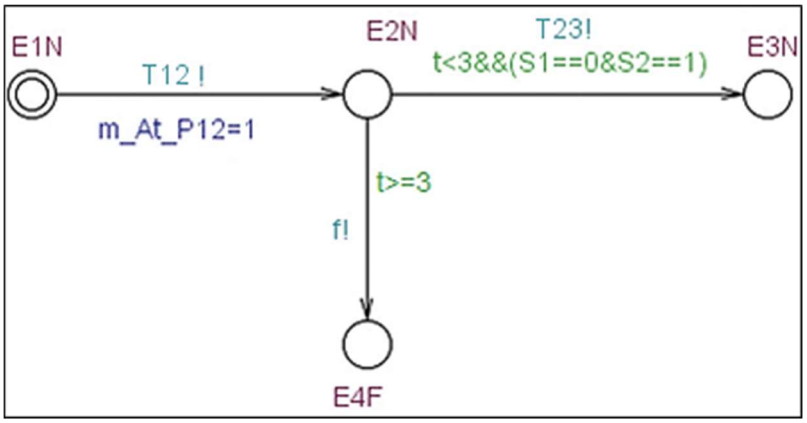

Figura 4.5- Autômato temporizado do processo escrito em UPPAAL.

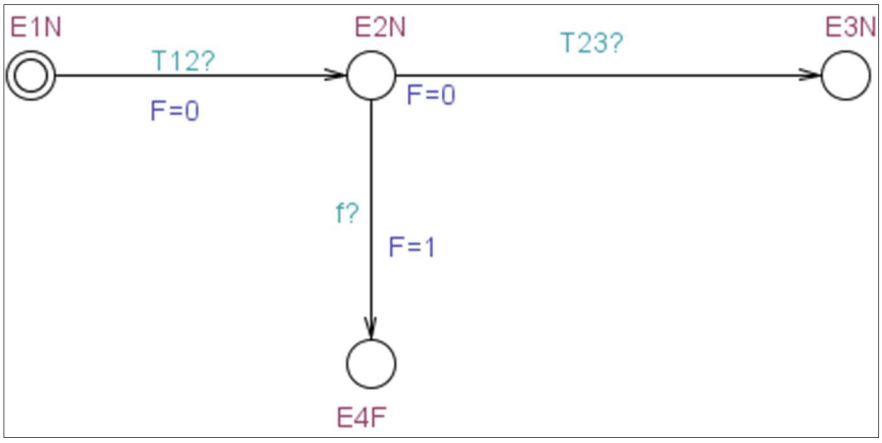

Figura 4.6- Diagnosticador Autômato temporizado escrito em UPPAAL.

O autômato do processo (G), exibido na Figura 4.5, é composto pelos seguintes conjuntos:

- Lugares: L={E1, E2, E3, E4};
- Lugar inicial: l_0={E1};
- Variáveis inteiras: V_{int}={m_At_P12, S1, S2}.
- Variável *clock*: C={t}.
- Canais de comunicação ou ações sincronizantes: A={T12, T23}.

O autômato diagnosticador, exibido na Figura 4.6, é composto pelos seguintes conjuntos:

- Lugares: L={E1N, E2N, E3N, E4F}, onde o conjunto de lugares normais é LN={ E1N, E2N, E3N} e o conjunto de lugares de falhas é LF={E4F}.

- Lugar inicial: l_0={E1N};
- Variáveis inteiras: V_{int}={F}.
- Canais de comunicação ou ações sincronizantes: A={T12, T23}.

No UPPAAL, as variáveis inteiras, canais de comunicação e variável *clock* são declarados, globalmente, conforme exibido na Figura 4.7.

Na Figura 4.5, a expressão "m_At_P12=1" representa o comando para movimentar o atuador. O canal de comunicação (emissor) "T12!", executado após o comando para o atuador, é utilizado para informar ao diagnosticador (observador) que o autômato do processo está se movendo do estado 1 para o estado 2.

```
int [0,1] m_At_P12, F, S1, S2;
clock t;
chan T12;
chan T23;
chan f;
```

Figura 4.7- Declaração de variáveis do autômato G e diagnosticador no UPPAAL.

O canal de comunicação (receptor) "T12?" fará com que o diagnosticador transite do estado 1 para o estado 2 sincronamente ao autômato do processo. Como não houve falha, o diagnosticador irá atribuir o valor 0 à variável F. Esta variável serve para informar se houve falha ou não. Quando não houver falha, o sensor S1 mudará para o estado desligado e, o sensor S2 mudará para o estado ligado em um intervalo de tempo inferior a 3 segundos. Esta condição está representada na guarda "t<3&&(S1=0&S2=1)" que, quando satisfeita, executa a ação sincronizante "T23!". O tempo de 3 segundos representa 10 vezes a duração da operação normal e é utilizado como um tempo limite, entre comportamento aceitável e não aceitável, do processo. Caso haja falha, a guarda "t>=3" será satisfeita fazendo com que a ação sincronizante "f!" seja executada. O diagnosticador observa o atraso de 3 segundos, por intermédio do canal receptor "f?" e anuncia a falha atribuindo o valor 1 à variável F.

4.2.3 Método proposto para o diagnóstico de faltas permanentes abruptas

O método para diagnosticar faltas permanentes abruptas, proposto nesta pesquisa, é inspirado no modelo em UPPAAL do diagnosticador proposto em TRIPAKIS (2002), discutido na seção anterior e no conceito de I-diagnosticabilidade conforme proposto por SAMPATH *et al.*. (1995). A diferença entre o método de diagnóstico de falhas que será apresentado nesta seção e o método de detecção de falhas apresentado na Seção 4.2.2 é que no segundo não há isolamento de falhas. Para isolar as faltas permanentes (falhas), o diagnosticador observará também os estados dos sensores, conforme já mencionado no início da seção 4.2.

O método de diagnóstico de faltas permanentes possui as seguintes premissas e definições:

1. As falhas de sensores e atuadores são do tipo: 1- Sensor travado no estado desligado; 2- Sensor travado no estado ligado; 3- Atuador travado ou não responde a sinais de controle.
2. O autômato G (processo) deve ser construído de modo que o conjunto de estados em que não houve falha seja disjunto do conjunto de estados onde ocorreu uma falha. Uma vez que tenha ocorrido uma falha, o autômato G transitará entre os estados de falha e não mais sairá deste conjunto de estados. O processo não se recupera de uma falha e uma vez detectada e isolada não deixará de ser anunciada pelo diagnosticador.
3. O método de diagnose assume que há dois conjuntos de eventos observáveis indicadores de falha: conjunto de eventos indicadores de falha indefinida e o conjunto de eventos indicadores de falha específica. O conjunto de eventos indicadores de falha indefinida foi definido como ΔIFI={IFIpf1, IFIpf12, IFIpf123,..., IFIpf123...M} onde cada elemento de ΔIFI é uma partição, com até M elementos do conjunto de eventos de falhas Δσf={σf1, σf2, σf3, σf4, σf5, σf6,..., σfm}. O conjunto de eventos indicadores de falha específica foi definido como ΔIF={IF1, IF2,..., IFm}. Os dois tipos de eventos devem ser declarados no UPPAAL como canais de comunicação. A nomenclatura destes dois tipos de eventos (canais de comunicação) assim como também de estados, devem

respeitar as restrições de nomenclatura impostas pelo UPPAAL. A título de exemplo, considere o sistema exibido na Figura 4.2. Após o comando enviado para o atuador mover-se de P1 para P2, poderá ocorrer três falhas distintas pertencentes a uma mesma partição de falhas. Neste caso, supondo que as falhas que podem ocorrer são as 3 primeiras do conjunto de falhas, isto é, σf1, σf2, σf3, a partição de falhas será formada por três elementos, por exemplo: pf123={σf1, σf2, σf3}. O evento indicador de falha indefinida será o IFIpf123, que no UPPAAL representa um canal de comunicação formado pelas ações ou eventos IFIpf123! (emissor) e IFpf123? (receptor/observador). O evento IFIpf123! é disparado no autômato G após a guarda de detecção de falha, tipo restrição de *clock,* ter sido satisfeita, enquanto que o evento IFIpf123? é disparado, sincronamente ao evento IFIpf123!, nos diagnosticadores da falha 1, falha 2 e falha 3. Considerando que neste mesmo exemplo, a partição de falhas é formada pelas falhas 1, 2 e 3 então haverá três eventos indicadores de falhas específicas (emissores) IF1!, IF2! e IF3!, presentes no autômato G e IF1?, IF2? e IF3? (observadores) presentes nos seus respectivos diagnosticadores.

4. Os eventos indicadores de falhas indefinidas foram declarados, no UPPAAL, como canais de comunicação do tipo *broadcast chan*, enquanto que os eventos indicadores de falha específica como canais de comunicação do tipo *chan*. Considera-se que há um diagnosticador para cada tipo de falha e estes diagnosticadores e o autômato G formam uma rede de autômatos temporizados. É importante ressaltar que o disparo de um evento indicador de falha indefinida não significa que os diagnosticadores observam diretamente o evento de falha, pois falhas são consideradas eventos não observáveis. Neste método de diagnose de falhas, a detecção de uma falha inicia com o disparo de um evento indicador de falha indefinida. Este evento é disparado quando uma guarda formada por uma restrição de *clock* é satisfeita, no estado ou lugar logo após um comando ter sido enviado para um atuador. Em outras palavras, os diagnosticadores observam o tempo em que o autômato G permanece no estado (lugar) posterior a um comando ter sido enviado para um atuador. Se o autômato G superar o tempo máximo permitido

para ele permanecer em um determinado estado (restrição de *clock*) significa que há uma anormalidade no processo, então um evento indicador de falha indefinida é habilitado fazendo com que o autômato G mude para um estado de falha pertencente ao conjunto de estados de falhas.

5. Cada diagnosticador é formado por três estados cujas identificações foram adicionados os rótulos N, Falhapf (1...M) e Falha (1...M) dependendo da quantidade de falhas da partição e com $M \in \mathbb{N}$. No primeiro estado do diagnosticador ainda não ocorreu a falha, logo possui o rótulo N (normal). Com a ocorrência de uma falha no processo, um evento de falha indefinida é disparado, fazendo com que o diagnosticador transite para o estado com rótulo pertencente ao conjunto de rótulos de partições de falhas:

   $\Delta Fp = \{Falhapf1, Falhapf12, Falhapf123, \dots, Falhapf123 \dots M\}$. Para o exemplo citado no item 3, em que há três falhas (σf1, σf2, σf3), o rótulo do segundo estado nos diagnosticadores 1, 2 e 3 será Falhapf123. O disparo de um evento ou ação sincronizante de falha indefinida representa a etapa de detecção da falha. Caso não haja informação suficiente para isolar a falha, o evento indicador de falha indefinida será novamente disparado no segundo estado do(s) diagnosticador(es) e permanecerá sendo disparado, em intervalos de tempo, no terceiro estado. Caso haja informação suficiente para isolar o tipo de falha, que ocorreu em uma partição contendo mais de uma falha, um dos diagnosticadores destas falhas transitará para o terceiro estado quando for disparado, no autômato G, o seu correspondente evento indicador de falha específica.

6. Um evento indicador de falha específica é habilitado caso a sua guarda (expressão booleana) seja satisfeita. A guarda dos eventos de falha específica são formadas por estados dos sensores que fazem parte da operação/atividade que está sendo executada. Após disparado o evento indicador de falha específica, é atribuído o valor lógico 1 a variável que identifica a falha. Cada variável de falha foi declarada como do tipo inteira e identificadas por F1, F2,...,Fm. Caso a falha não tenha sido isolada, será atribuído valor lógico 1 a variáveis, do tipo inteira,

identificadas por FI1,FI2,...,FIM. A operação de atribuir valor 1 as variáveis correspondentes as falhas específicas, representa a etapa de isolação da falha. O terceiro estado do diagnosticador possui rótulo pertencente a um conjunto de rótulos de falhas específicas definido por $\Delta F = \{\text{falha } 1, \text{falha } 2, \ldots, \text{falha M}\}$ ou terá o mesmo rótulo do segundo estado, caso a falha não tenha sido isolada. Neste estado a falha específica ou indefinida é anunciada em intervalos de tempo, devido a invariante.

7. Caso ocorram múltiplas falhas, apenas uma falha será diagnosticada por vez. A localização atual do autômato G determinará a falha que será diagnosticada primeiro, isto é, será detectado a falha no sensor ou atuador danificado em que sua atuação seja solicitada primeiro pelo processo. Após a primeira falha ter sido diagnosticada e tratada, o processo poderá continuar do estado imediatamente anterior a falha ou ser reiniciado caso o controlador deseje. A segunda falha também será detectada tão logo o sensor ou atuador danificado seja solicitado pelo processo. A Estação permanecerá executando as operações que fazem parte do processo enquanto o controlador não solicitar a atuação do sensor ou atuador danificado.

Os autômatos exibidos nas figuras 4.8 e 4.9 são modelos para serem escritos na linguagem do *software* UPPAAL. Na figura 4.8 é representado um autômato G (processo) contendo os elementos necessários para o diagnóstico de falhas de dispositivos (sensores e atuador). Este autômato é uma aplicação do método de diagnóstico de faltas permanentes (abruptas) a um sistema semelhante ao da Figura 4.2. O modelo genérico proposto para diagnosticadores de faltas permanentes está exibido na Figura 4.9.

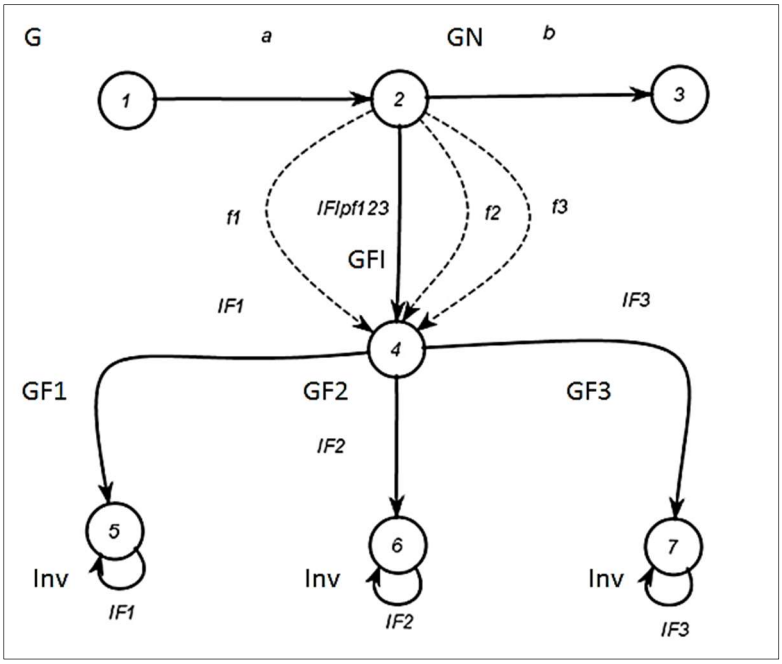

Figura 4.8- Modelo de autômato G, considerando um atuador e um par de sensores discretos, para diagnóstico de faltas permanentes

O autômato exibido na figura 4.8 é constituído pelos seguintes conjuntos e elementos:

- Conjunto de Lugares ou estados: L={1, 2, 3, 4, 5, 6, 7};
- Conjunto de Eventos ou ações observáveis: Σ_o ={a, b, IFIpf123,IF1, IF2, IF3};
- Conjunto de Eventos ou ações não observáveis: Σ_{uo} ={f1, f2, f3} e $\Sigma_{uo} = \Sigma_f$.
- Conjunto de eventos ou ações indicadoras de falha indefinida ΔIFI={ IFIpf123} e a partição de falhas é pf123={f1, f2, f3}.
- Conjunto de eventos indicadores de falhas específicas ΔIF={IF1, IF2, IF3}.
- Evento a: Representa um comando para um atuador;
- Evento b: Representa a habilitação e disparo da guarda GN; Após disparo do evento b, as variáveis de *clock* e inteira (a) devem ser zeradas.
- Evento IFIpf123 (indicador de falha indefinida): Evento sincronizante (canal emissor) que ocorre quando a guarda GFI é habilitada e disparada; Após disparado, as variáveis de *clock* e inteira (a) devem ser zeradas.

- Eventos IF1, IF2, IF3 (indicadores de falhas específicas): Eventos sincronizantes (canais emissores) que ocorrem quando as guardas GF1 ou GF2 ou GF3 são habilitadas e disparadas respectivamente. São disparados também pela violação das invariantes.
- GN (guarda): Expressão boleana formada pelos estados esperados (comportamento normal) dos sensores S1 e S2 e restrição de tempo (*clock*).
- GFI (guarda): Expressão boleana formada por uma variável do tipo *clock.* Restrição que indica o limite de tempo aceitável para operação. Caso a guarda seja satisfeita, significa que há uma falha em algum dispositivo. Neste momento, a falha é do tipo indefinida.
- GF1, GF2, GF3 (guardas): Expressões boleanas formadas pelos estados dos sensores S1 e S2. Indicam um desvio de comportamento (falha) do atuador ou do sensor 1 ou do sensor 2. Cada guarda é uma combinação de estados dos sensores e representa a falha de um dos dispositivos.
- Inv: Invariante que define a frequência de disparo dos eventos de falhas específicas. Quando não houver isolação da falha, a invariante fará com que um evento indicador de falha indefinida seja disparado periodicamente.

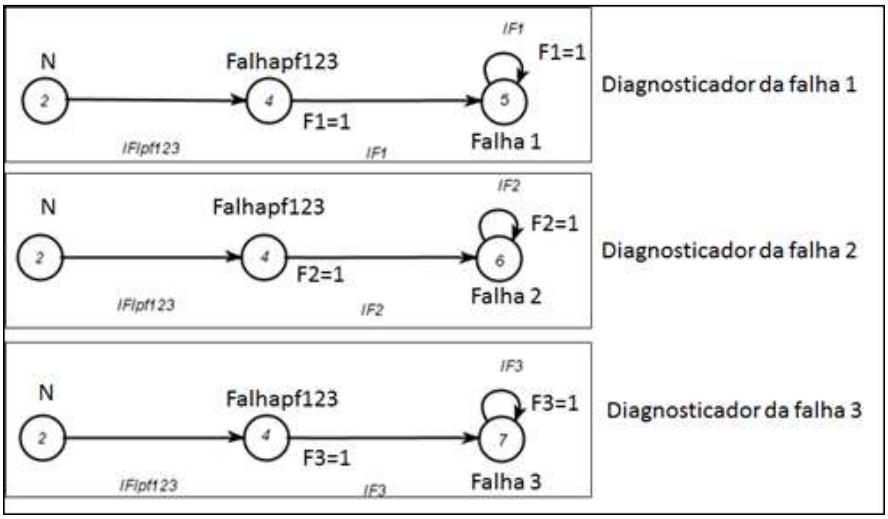

Figura 4.9- Modelo de diagnosticadores de faltas permanentes

O autômato exibido na figura 4.9 é constituído pelos seguintes conjuntos e elementos:

- Conjunto de Lugares ou estados: L={2N, 4 Falhapf123, 5 Falha1, 6 Falha2, 7 Falha3};
- Conjunto de eventos ou ações indicadoras de falha indefinida ΔIFI={ IFIpf123} e a partição de falhas é pf123={f1, f2, f3}.
- Conjunto de eventos indicadores de falhas específicas ΔIF={IF1, IF2, IF3}.
- Evento IFIpf123 (indicador de falha indefinida): Evento (canal receptor) que é disparado sincronamente a ação IFIpf123 (canal emissor);
- Eventos IF1, IF2, IF3 (indicadores de falhas específicas): Eventos (canais receptores) que são disparados sincronamente as ações IF1, IF2 e IF3 (canais emissores) respectivamente;
- Conjunto de variáveis inteiras: V_{int}= {F1, F2, F3}.
- Conjunto de rótulos de partições de falhas ΔFp={Falhapf123}.
- Conjunto de rótulos de falhas ΔF={Falha 1, Falha 2, Falha 3}.

Um exemplo de aplicação do método de detecção de faltas permanentes abruptas será tratado detalhadamente na seção 4.2.4.1.

4.2.4 Método proposto para o diagnóstico de faltas intermitentes

O método proposto para detecção de faltas intermitentes tem como base as mesmas premissas e definições apresentadas na seção 4.2.1. Uma diferença fundamental, com relação ao método para diagnóstico de faltas permanentes, reside no fato de que o autômato G poderá alcançar um estado de falha e depois retornar para um estado normal. Esta mudança implica autômatos G e diagnosticadores mais complexos que consequentemente demandam maior esforço computacional para serem implementados.

Em concordância com o que já foi exposto, faltas intermitentes são faltas que se alternam em intervalos não determinísticos ao longo do tempo. O sistema poderá conviver com a falta até alcançar um estado de falha. Antes de alcançar este estado, a qualidade da produção e a segurança do processo podem ter sido comprometidas.

A falha intermitente, assim como a falha permanente, poderá também ser detectada a partir da observação de um desvio no tempo esperado para uma operação.

Entretanto, para que sejam isoladas, o diagnosticador irá observar se o sensor permanece no estado de falha por um tempo limite ou se ele volta a funcionar conforme esperado. Este tempo limite é um atraso que irá definir se a falha é do tipo permanente, caso o estado do sensor continue irregular, ou intermitente, caso o estado do sensor mude para o esperado. O valor deste período de observação foi considerado, de modo arbitrário, igual ao tempo determinado para disparar o evento indicador de falha indefinida. Este método de diagnóstico está presente nos diagnosticadores dos sensores e do atuador.

Outro método, pertencente exclusivamente aos diagnosticadores dos sensores, consiste na observação de mudanças de estados dos sensores, mesmo após a operação ter sido concluída sem desvios.

Assim como ocorreu na seção 4.2.1, nesta seção será apresentado o modelo genérico de um autômato G (processo) e diagnosticadores para o caso de falhas intermitentes, considerando um sistema semelhante ao apresentado na Figura 4.2. A Figura 4.10 exibe o autômato G (processo), e a Figura 4.11 e 4.12, os autômatos diagnosticadores das falhas 1 e 2 respectivamente. O diagnosticador da falha 3 é similar ao da falha 2, por isso não será exibido.

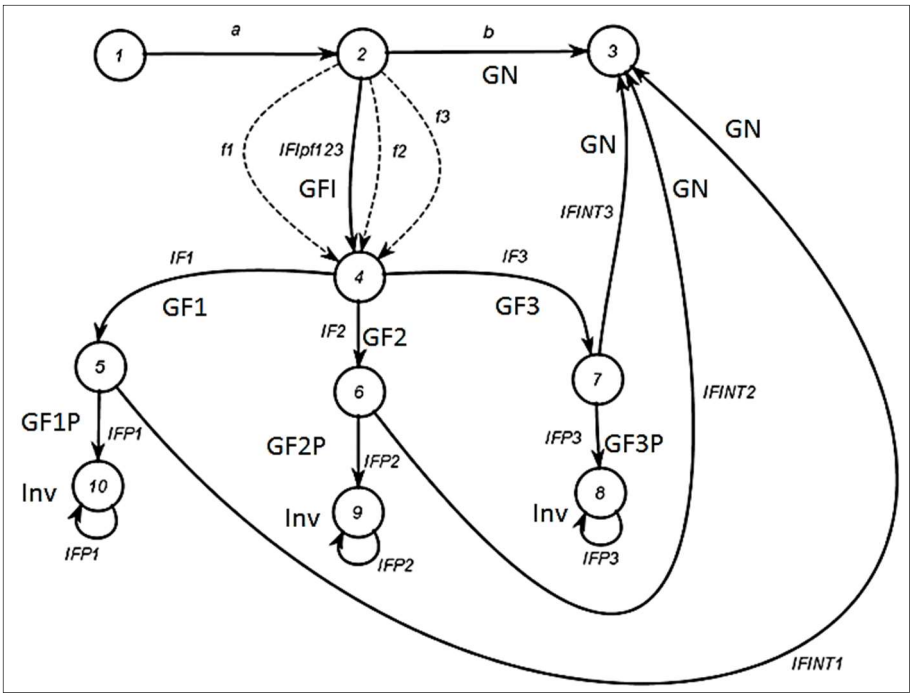

Figura 4.10- Modelo (genérico) de autômato G para diagnóstico de faltas intermitentes e permanentes

O autômato exibido na figura 4.10 é constituído pelos seguintes conjuntos e elementos:

- Conjunto de Lugares ou estados: L={1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
- Conjunto de Eventos ou ações observáveis: Σ_o ={a, b, IFIpf123,IF1, IF2, IF3, IFP1, IFP2, IFP3, IFINT1, IFINT2, IFINT3};
- Conjunto de Eventos ou ações não observáveis: Σ_{uo} ={f1, f2, f3} e $\Sigma_{uo} = \Sigma_f$.
- Conjunto de eventos ou ações indicadoras de falha indefinida ΔIFI={ IFIpf123} e a partição de falhas é pf123={f1, f2, f3}.
- Conjunto de eventos indicadores de faltas específicas ΔIF={IF1, IF2, IF3}.
- Conjunto de eventos indicadores de faltas específicas permanentes ΔIFP={IFP1, IFP2, IFP3}.
- Conjunto de eventos indicadores de faltas específicas intermitentes ΔIFINT={IFINT1, IFINT2, IFINT3}.
- Evento a: Representa um comando para um atuador;
- Evento b: Evento sincronizante (canal emissor) que representa a habilitação e disparo da guarda GN; Após disparo do evento b, as variáveis de *clock* e inteira (a) devem ser zeradas.
- Evento IFIpf123 (indicador de falha indefinida): Evento sincronizante (canal emissor) que ocorre quando a guarda GFI é habilitada e disparada; Após disparado, as variáveis de *clock* e inteira (a) devem ser zeradas.
- Eventos IF1, IF2, IF3 (indicadores de falhas específicas): Eventos sincronizantes (canais emissores) que ocorrem quando as guardas GF1 ou GF2 ou GF3 são habilitadas e disparadas respectivamente;
- Eventos IFP1, IFP2, IFP3 (indicadores de faltas específicas permanentes): Eventos sincronizantes (canais emissores) que ocorrem quando as guardas GF1P ou GF2P ou GF3P são habilitadas e disparadas respectivamente. Após disparados, a variável de *clock* deve ser zerada. São disparados também pela violação das invariantes.
- Eventos IFINT1, IFINT2, IFINT3 (indicadores de falhas específicas intermitentes): Eventos sincronizantes (canais emissores) que ocorrem

quando a guarda GN é habilitada e disparada após eventos indicadores de falhas específicas. Após disparados, a variável de *clock* deve ser zerada

- GN (guarda): Expressão boleana formada pelos estados esperados (comportamento normal) dos sensores S1 e S2 e restrição tempo (*clock*).
- GFI (guarda): Expressão boleana formada por uma variável do tipo *clock.* Restrição que indica o limite de tempo aceitável para operação. Caso a guarda seja satisfeita, significa que há uma falha em algum dispositivo. Esta falha é do tipo indefinida.
- GF1, GF2, GF3 (guardas): Expressões boleanas formadas pelos estados dos sensores S1 e S2. Indicam um desvio de comportamento (falha) do atuador ou do sensor 1 ou do sensor 2. Cada guarda representa a falha de um único dispositivo.
- GF1P, GF2P, GF3P (guardas): Expressões boleanas formadas pelos estados dos sensores S1, S2 e restrição de tempo (atraso). Este atraso é necessário para definir se a falha é permanente ou intermitente. Quando as guardas são satisfeitas, indicam uma falha permanente no atuador ou no sensor 1 ou no sensor 2. Cada guarda representa a falha de um único dispositivo.
- Inv: Invariante que define a frequência de disparo dos eventos de falhas específicas.

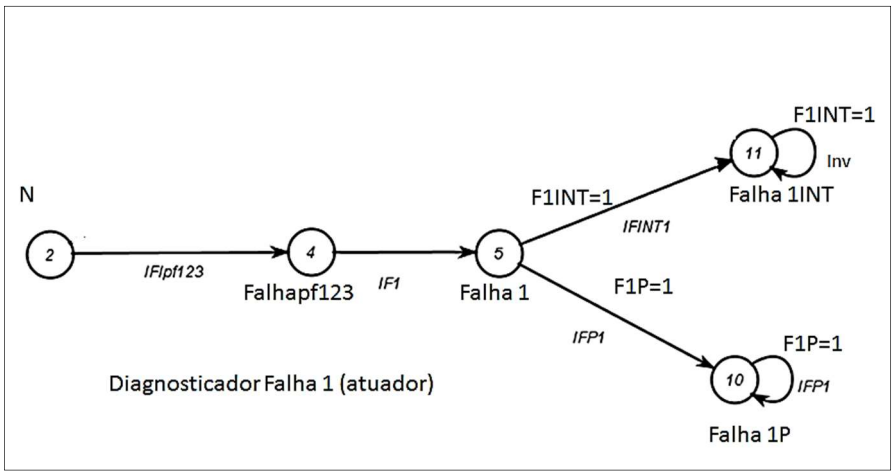

Figura 4.11- Modelo (genérico) de diagnosticador da falta intermitente e permanente do atuador (falha 1)

O autômato exibido na Figura 4.11 é constituído pelos seguintes conjuntos e elementos:

- Conjunto de Lugares ou estados: L={2N, 3N, 4 Falhapf123, 5 Falha1, 10 Falha 1P, 11 Falha 1INT};
- Conjunto de eventos ou ações indicadoras de falha indefinida ΔIFI={ IFIpf123} e a partição de falhas é pf123={f1, f2, f3}.
- Conjunto de eventos indicadores de falhas específicas ΔIF={IF1}.
- Conjunto de eventos indicadores de faltas específicas permanentes ΔIFP={IFP1 }.
- Conjunto de eventos indicadores de falhas específicas intermitentes ΔIFP={IFINT1 }.
- Evento IFIpf123 (indicador de falha indefinida): Evento (canal receptor) que é disparado sincronamente a ação IFIpf123 (canal emissor);
- Evento IF1 (indicador de falha específica): Evento sincronizante (canal receptor) que é disparado sincronamente a ação IF1 (canal emissor);
- Eventos IFP1 (indicador de falha específica permanente): Evento sincronizante (canal receptor) que é disparado sincronamente a ação IFP1 (canal emissor). É disparado também pela violação da invariante.
- Evento IFINT1 (indicador de falha específica intermitente): Evento sincronizante (canal receptor) que é disparado sincronamente a ação IFINT1 (canal emissor).
- Conjunto de variáveis inteiras: V_{int}= {F1P, F1INT}.
- Conjunto de rótulos de partições de falhas ΔFp={Falhapf123}.
- Conjunto de rótulos de falhas ΔF={Falha 1}.
- Conjunto de rótulos de falhas permanentes ΔF={Falha 1P}.
- Conjunto de rótulos de falhas intermitentes ΔF={Falha 1Int}.
- Inv: Invariante que define a frequência de disparo dos eventos de falhas específicas.

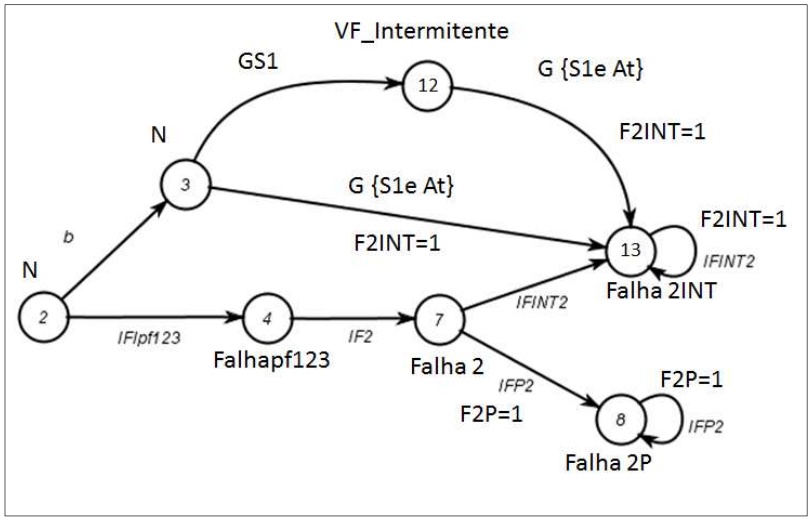

Figura 4.12- Modelo (genérico) de diagnosticador da falta intermitente e permanente do sensor 1 (falha 2)

O autômato exibido na Figura 4.12 é constituído pelos seguintes conjuntos e elementos:

- Conjunto de Lugares ou estados: L={2N, 3N, 4 Falhapf123, 12 VF (Intermitente), 7 Falha2, 8 Falha 2P, 13 Falha 2INT};
- Conjunto de eventos ou ações indicadoras de falha indefinida ΔIFI={ IFIpf123} e a partição de falhas é pf123={f1, f2, f3}.
- Conjunto de eventos indicadores de falhas específicas ΔIF={IF2}.
- Conjunto de eventos indicadores de falhas específicas permanentes ΔIFP={IFP2 }.
- Conjunto de eventos indicadores de falhas específicas intermitentes ΔIFP={IFINT2 }.
- Evento b: Evento sincronizante (canal receptor) que representa a habilitação e disparo da guarda GN;
- Evento IFIpf123 (indicador de falha indefinida): Evento (canal receptor) que é disparado sincronamente a ação IFIpf123 (canal emissor);
- Evento IF2 (indicador de falha específica): Evento sincronizante (canal receptor) que é disparado sincronamente a ação IF2 (canal emissor);
- Evento IFP2 (indicador de falha específica permanente): Evento sincronizante (canal receptor) que é disparado sincronamente a ação IFP2 (canal emissor). É disparado também pela violação da invariante.

- Evento IFINT2 (indicador de falha específica intermitente): Evento sincronizante (canal receptor) que é disparado sincronamente a ação IFINT2 (canal emissor). Após disparado, a variável *clock* deve ser zerada.
- GS1: Expressão boleana formada com o estado atual do sensor S1 após a operação ter sido concluída.
- G {S1 e At}: Expressão boleana com o estado atual do sensor S1 e do atuador após a operação ter sido concluída. Esta guarda representa uma mudança de estado do sensor 1 sem que o atuador tenha sido acionado.
- Conjunto de variáveis inteiras: V_{int}= {F2P, F2INT}.
- Conjunto de rótulos de partições de falhas ΔFp={Falhapf123}.
- Conjunto de rótulos de falhas ΔF={Falha 2}.
- Conjunto de rótulos de falhas permanentes ΔF={Falha 2P}.
- Conjunto de rótulos de falhas intermitentes ΔF={Falha 2Int}.
- Inv: Invariante que define a frequência de disparo dos eventos de falhas específicas.

Um exemplo de aplicação do método de detecção de faltas intermitentes será tratado na seção 4.2.5.2

4.2.5 Aplicação dos métodos de diagnóstico de faltas à uma das operações da estação de distribuição

Para exemplificar uma aplicação, no sistema modular de produção (Festo), dos métodos de diagnóstico de faltas apresentados, será considerada, sem perda de generalidade, uma das operações que é executada na Estação de Distribuição. Este exemplo tratará do caso mais comum de detecção de faltas, em que há um par de sensores e um atuador. Outras operações que fazem parte do processo desta Estação ou das demais são equivalentes à deste exemplo, portanto seguirão os mesmos procedimentos de modelagem e sujeitas aos métodos de diagnóstico de faltas apresentados. A seguir serão apresentados detalhadamente a operação e o método de diagnóstico das faltas permanentes. Na seção 4.2.5.2, serão apresentados e discutidos brevemente os autômatos do processo e diagnosticadores para o caso de diagnóstico de faltas intermitentes.

Não serão apresentados os autômatos, escritos em linguagem UPPAAL, referentes ao método de detecção de falhas baseado em TRIPAKIS (2002). Na Seção 4.2.2 já foi exibida e discutida a aplicação do referido método a um sistema similar ao que será utilizado como estudo de caso nesta seção.

4.2.5.1 Método de diagnóstico de faltas permanentes

Considere o atuador linear e seus sensores de fim de curso, exibido na Figura 4.13. Uma das operações do processo da estação de distribuição, consiste em recuar o atuador para que uma peça seja retirada do *magazine*. Para isto, o Controlador envia o comando RAL (Recuar atuador linear). Inicialmente o atuador encontra-se na posição P1 e o sensor 1 está ativado. Assumindo que não há faltas no controlador ou na comunicação entre controlador e dispositivos, espera-se que após recebido o comando RAL o atuador recue imediatamente. O recuo do atuador fará com que o sensor 1 seja desabilitado e o sensor 2 habilitado, indicando que o atuador se encontra na posição P2. Nesta operação, a variável RAL será ativada, assumindo o valor lógico 1. A variável SAav (Sensor atuador avançado), que representa o estado do sensor 1, será desativada (valor lógico 0) e a variável SAre (Sensor atuador recuado) será ativada (valor lógico 1). Há três possíveis falhas nesta operação: Falha 1 – O atuador linear ficar travado na posição avançada, isto é, não responder ao sinal RAL; Falha 2– O sensor 1 ficar travado no estado ligado mesmo após o atuador ter recuado; Falha 3 – O atuador recuar e o sensor 2 não detectar a presença do atuador, por estar travado no estado desligado. Embora possam ocorrer as falhas sensor atuador avançado travado no estado desligado ou sensor atuador recuado travado no estado ligado, estas se confundem com o comportamento esperado destes sensores após o comando RAL ter sido executado. Caso estes sensores estejam nestes estados de falha, a detecção destas falhas ocorrerá quando o controlador solicitar ao atuador a execução da operação inversa, que é de avançar o atuador linear. O autômato G que modela a operação de recuar o atuador linear, incluindo falhas, é exibido na Figura 4.14. Considere que as variáveis SAav, SAre, RAL, F1, F2, F3 são declaradas como variáveis inteiras; t é uma variável do tipo *clock* que assume valores reais; IFIpf123!, IF1!, IF2! e IF3! são ações ou eventos do tipo emissor (canais de comunicação); Para efeito de análise, a execução das expressões "RAL:=1" e "t<3&&SAav==0&SAre==1" equivalem ao disparo dos eventos α e β respectivamente; $\sigma f1, \sigma f2$ e $\sigma f3$ são eventos não observáveis que representam as faltas 1, 2 e 3 respectivamente;

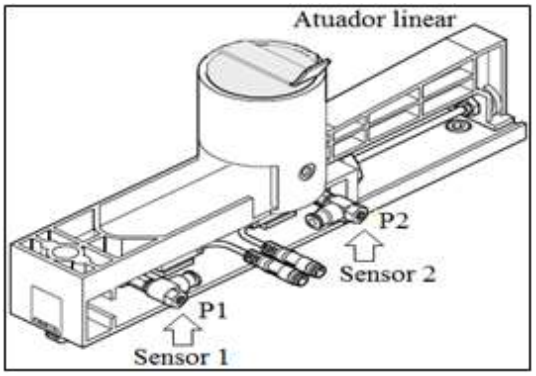

Figura 4.13 - Atuador Linear (Festo Didactic , 2006).

Seja Q o conjunto de estados, Qf o conjunto de estados de faltas e Q-Qf o conjunto de estados onde não ocorreu falha. Inicialmente o autômato encontra-se no estado E1. Quando o controlador envia o comando para recuar o atuador linear, a variável RAL assume o valor lógico 1, representado pela atribuição "RAL:=1". Isto equivale ao disparo de um evento, neste caso α, e o autômato transita para E2. Assumindo que a operação de recuar dura em média 0,5 segundos, o tempo de 3 segundos será utilizado para indicar um desvio inaceitável de comportamento, isto é, uma falha. Assumindo que o atuador recue em menos de 3s "t<3", e que os sensores 1 e 2 mudarão para o estado desligado e ligado respectivamente, o autômato transitará para o estado E7, executando um *reset* (atribuindo valor lógico 0) a variável de *clock* t e RAL. Esta condição está expressa na guarda que conecta o estado E2 ao estado E7 e quando satisfeita, equivale ao disparo do evento β. Caso o autômato alcance no mínimo 3s de permanência no estado E2 o evento indicador de falta indefinida IFIpf123! é disparado e o *clock* t é zerado. Neste momento sabe-se somente que ocorreu algum dos três tipos de faltas da partição de faltas pf123. Os diagnosticadores 1, 2 e 3 observam o disparo do evento indicador de falta indefinida e transitam, cada um, para os seus respectivos segundos estados com rótulo (Faltapf123). Neste estado, será verificado qual das três faltas ocorreu a partir da leitura dos estados dos sensores 1 e 2, que estão representados pelas variáveis SAav e SAre. Por exemplo, caso o atuador esteja travado será disparado o evento IF1!. Esta condição é deduzida pelo fato do sensor atuador avançado ainda estar habilitado "SAav==1" e o sensor atuador recuado não ter sido habilitado "SAre==0", após o comando de recuar o atuador linear ter sido executado. Caso o sensor atuador avançado ainda esteja habilitado e o sensor atuador recuado seja ativado, será disparado IF2!, indicando que ocorreu a falta 2. Neste caso o atuador recuou e o sensor 1 não desabilitou. Finalmente, caso o sensor atuador avançado seja

desabilitado e o sensor atuador recuado não habilite, significa que ocorreu a falta 3 e o evento IF3! será disparado. O diagnosticador para a falta 1 está exibido na Figura 4.15. Os diagnosticadores para as faltas 2 e 3 são equivalentes ao da falta 1. O diagnosticador observou o evento de falta indefinida IFIpf123! assim como também o evento de falta específica IF1! e alcançou o estado de falta 1, a partir do disparo de IFIpf123? e IF1? respectivamente. A variável F1 assume o valor lógico 1 após o disparo de IF1?. Esta variável aponta que o diagnosticador atingiu o estado de falta 1 e registra a ocorrência desta falta. Finalmente, no terceiro estado do diagnosticador, a falta 1 é anunciada a cada intervalo de segundo, por causa da invariante (t<=1) ocasionar o disparo do auto-laço formado por IF1?.

Analisando o autômato diagnosticador proposto, pode ser observado que o mesmo não possui ciclos indeterminados, logo a linguagem L gerada pelo autômato G, é I-diagnosticável. Além disso, o autômato G é Δ-diagnosticável pois não há duas sequências temporais que possuam os mesmos eventos observáveis e em somente uma delas tenha ocorrido um evento de falta. O diagnosticador é capaz de detectar um evento de falta após pelo menos um intervalo de tempo Δ depois de sua ocorrência. A divisão do autômato G em conjuntos disjuntos de estados de falta e de estados normais, garante a não existência de ciclos indeterminados, uma vez que, o autômato G ao entrar no conjunto de estados de faltas não mais sairá do mesmo.

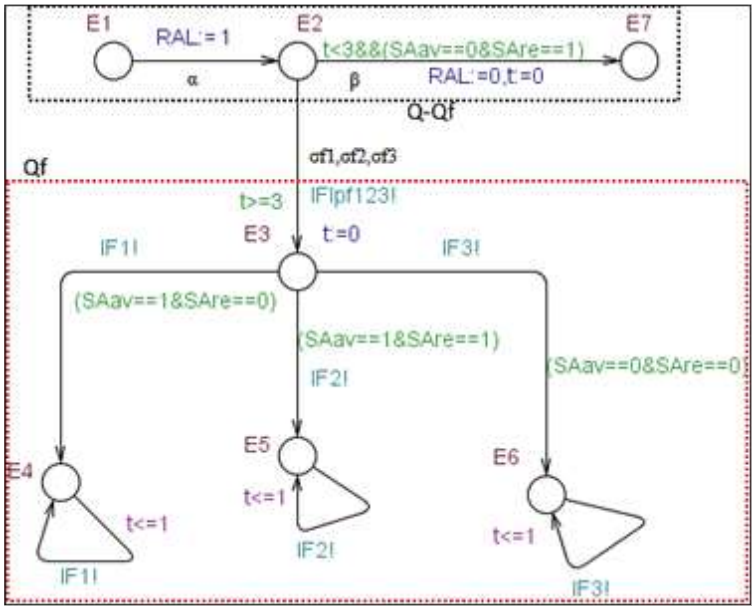

Figura 4.14 – Autômato que representa a operação de recuar o atuador linear pertencente ao Autômato G da estação de Distribuição.

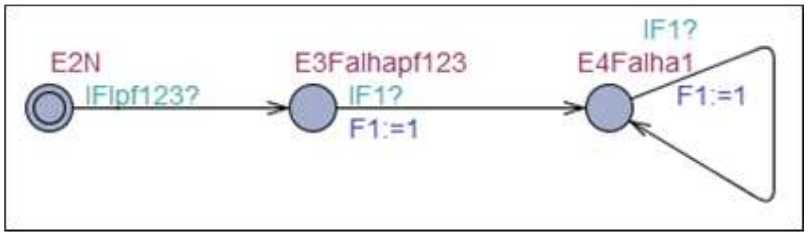

Figura 4.15 - Diagnosticador da falta 1.

Há casos em que não é possível isolar ou identificar o tipo de falta que ocorreu. O diagnosticador detecta que houve uma anormalidade no processo, isto é, detecta uma falta, mas não é capaz de definir qual tipo de falta ocorreu por informação insuficiente. Este são casos que não há sensores para monitorar o estado do atuador, antes e após a execução da ação solicitada pelo controlador. Esta situação ocorre na estação de distribuição quando o controlador envia o comando para ligar ou desligar o vácuo "LDV:=1", afim de que o braço rotativo capture ou libere a peça de trabalho. Não há sensores para detectar os estados ligado e desligado do dispositivo succionador (vácuo). Ainda assim, nesta operação é possível definir entre dois tipos possíveis de faltas, pois há um sensor de detecção de peça no succionador. Caso seja enviado o comando para ligar o vácuo e o sensor de peça capturada não mude para o estado ligado em até 3s "t<3&&SPseg==1", então houve uma falta. Esta falta pode ser do sensor de detecção de peça do succionador travado no estado desligado ou falta no atuador que liga o vácuo, não é possível identificar qual destas faltas ocorreu. Sendo assim haverá somente o disparado do evento indicador de falta indefinida IFIpf13!. O usuário, utilizando uma tabela com a identificação de todos os tipos de faltas da estação, saberá ao menos que a falta detectada é no sensor de peça capturada ou no atuador de vácuo. A Figura 4.16 extraída do autômato G da estação de distribuição, exibe esta operação. A Figura 4.17 exibe o diagnosticador da falta indefinida identificada por FI13.

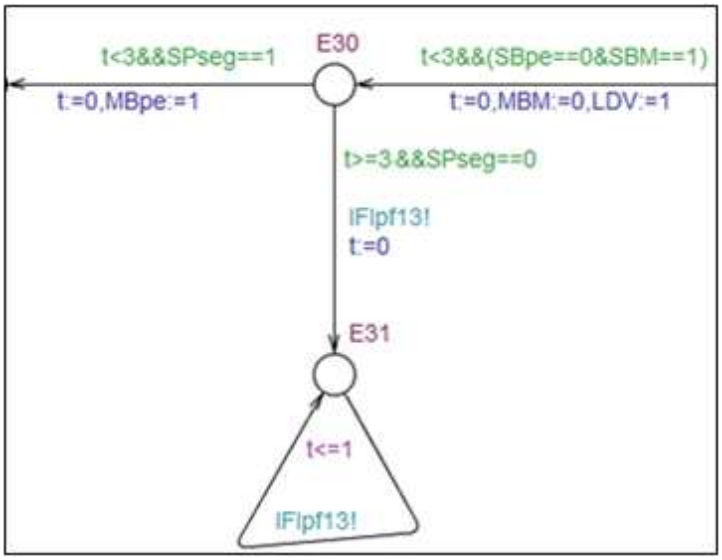

Figura 4.16 - Falta indefinida ao ligar vácuo.

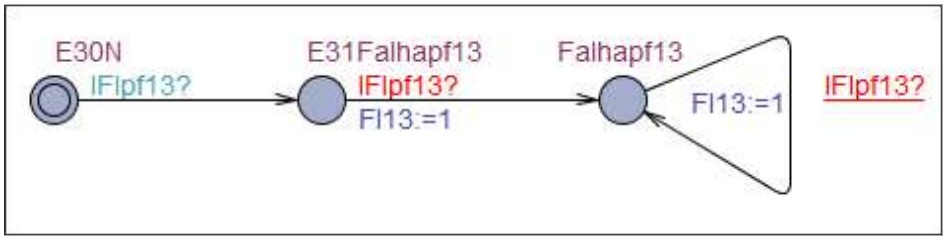

Figura 4.17 – Diagnosticador da Falta indefinida FI13.

4.2.5.2 Método de diagnóstico de falhas intermitentes

Em relação ao exemplo discutido na seção precedente, considere agora que os dispositivos SAre, SAav e atuador podem apresentam um comportamento de falha intermitente. Esta mudança faz com que o autômato do processo possa alcançar um estado de falha específica (E4F, E5F ou E6F), pertencente ao conjunto de estados de falhas e depois retorne a um estado normal ou esperado E7N. O autômato G (processo), modelado para o diagnóstico de falhas intermitentes e permanentes, está exibido na Figura 4.18. Os diagnosticadores para as falhas 1, 2 e 3 intermitentes ou permanentes, estão exibidos nas Figuras 4.19, 4.20 e 4.21, respectivamente.

Inicialmente será discutida a falha intermitente do atuador. Considere que há dois possíveis comportamentos que o atuador pode apresentar, após receber o comando RAL, quando estiver em falha intermitente. O primeiro é o atuador permanecer parado por 3 segundos, superando o tempo limite para conclusão da operação e depois

conseguir alcançar a posição P2. Inicialmente o evento indicador de falha indefinida "IFIpf123!" será disparado, depois a guarda "SAav==1&SAre==0" será habilitada fazendo com que o evento "IF1!" seja executado. Após estes eventos, o autômato G estará situado no estado E4. Caso os sensores mudem de estado, habilitando a guarda "SAav==0&SAre==1", o autômato G alcançará o estado E7N. Esta mudança de estados dos sensores irá fazer com que seja disparado o evento indicador de falha 1 intermitente, isto é, "IFINT1!". O diagnosticador 1 irá observar o evento indicador de falha 1 intermitente e irá atribuir o valor 1 a variável F1INT. Ainda no estado E4, caso os sensores permaneçam nos estados "SAav==1&SAre==0" por 3 segundos, correspondente a falha 1, o autômato irá transitar para o estado E10, disparando o evento "IFP1!". O diagnosticador muda para o estado de falha 1 permanente, sincronamente ao evento "IFP1!".

O segundo comportamento é o atuador iniciar o movimento, parar em alguma posição entre os sensores SAav e SAre e depois conseguir alcançar P2. Caso ocorra esta situação, o diagnosticador não será capaz de identificar que houve falha intermitente no atuador. Se depois de ter permanecido entre P1 e P2, o atuador alcançar P2 em menos de 3 segundos, o autômato G irá realizar transições semelhantes às realizadas em operação normal. Se o atuador permanecer parado entre P1 e P2, isto é, "SAav" desligado e "SAre" também desligado, o diagnosticador anunciaria que o sensor "SAre" está travado no estado desligado. Sendo assim, o diagnosticador não é capaz de identificar uma falha intermitente, no atuador, que resulte neste comportamento.

As falhas intermitentes nos sensores podem ser diagnosticadas de dois modos distintos. Assim como no caso do atuador, caso os sensores não mudem para o estado esperado representado pela guarda "SAav==0&SAre==1" em menos de 3 segundos, será disparado o evento indicador de falha indefinida, e depois, o de falha específica "IF2!" ou "IF3!". Nos estados E5 ou E6 será verificado se os sensores mudam para o estado esperado ou se permanecem no estado de falha. Os eventos indicadores de falha intermitentes "IFINT2!" ou "IFINT3!" serão disparados se os sensores mudarem para o estado esperado antes de um novo período de 3 segundos. Quando isto acontecer, os diagnosticadores irão alcançar os seus respectivos estados de falha intermitente e registrarão a ocorrência destas falhas, isto é, "F2INT=1" ou "F3INT=1".

O outro modo de detecção de falha intermitente dos sensores acontece após a operação de recuar, o atuador ter sido finalizado. Nesta situação, o diagnosticador

observa que a operação ocorreu normalmente através do evento “N23!”. Para o caso do diagnóstico da falha 2, por exemplo, considere que o sensor atuador avançado “SAav” estava desligado após o evento “N23!” e depois ligou sem que o atuador tenha recebido um comando para avançar “AAL=0”. Outro comportamento, que irá indicar a falha 2 intermitente, é o sensor “SAav” ligar imediatamente após “N23!” e sem que um comando tenha sido enviado para o atuador avançar. O primeiro comportamento é característico de uma falha intermitente que resulte em mudanças de estados lentas. O segundo comportamento, seria para mudanças de estado mais rápidas do que os ciclos de varreduras do controlador. Estas mesmas formas de detecção de falha intermitente são válidas para o sensor “SAre”, ou seja, para o diagnóstico da falha 3 intermitente.

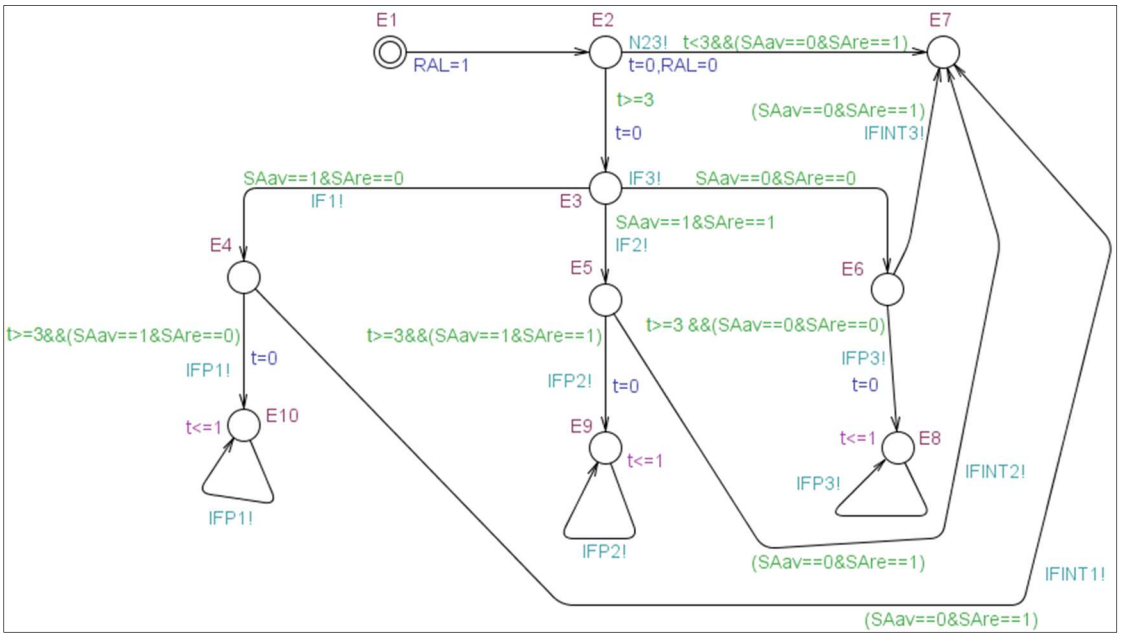

Figura 4.18 – Exemplo de autômato G para diagnóstico de falhas intermitentes e permanentes.

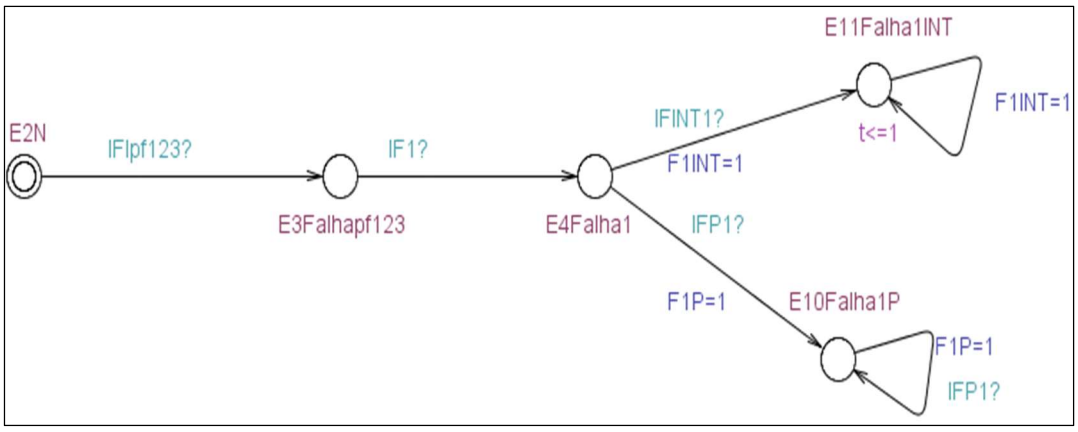

Figura 4.19 – Diagnosticador da falha 1 (intermitente ou permanente).

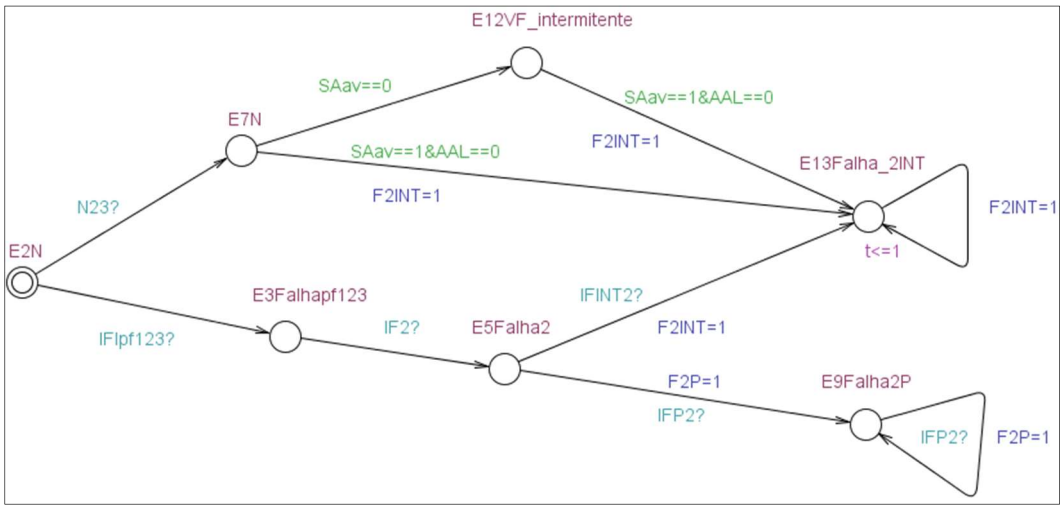

Figura 4.20 – Diagnosticador da falha 2 (intermitente ou permanente).

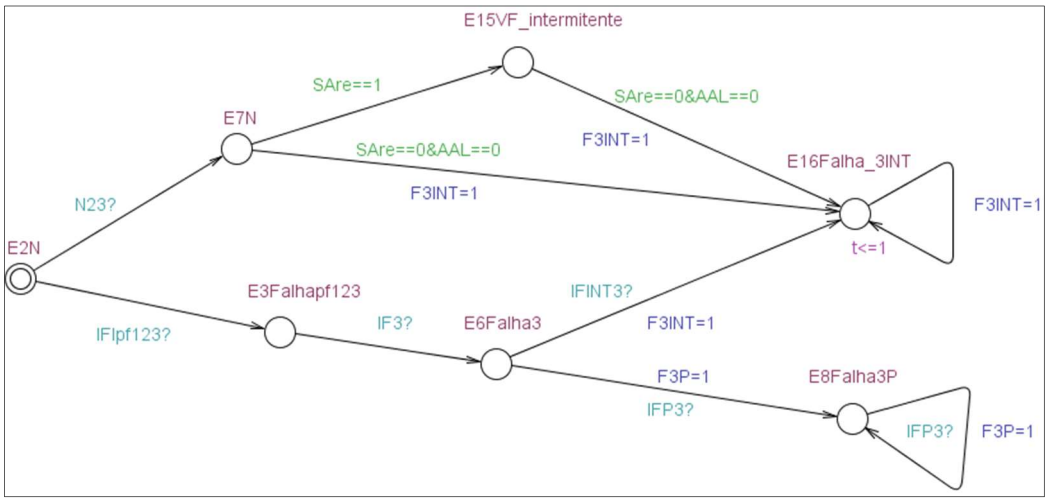

Figura 4.21 – Diagnosticador da falha 3 (intermitente e permanente).

4.3 Resultados

Para simular o comportamento dinâmico das estações e as faltas, foi utilizado o ambiente *Simulator* do software UPPAAL. Antes de iniciar a simulação é necessário descrever os autômatos no ambiente *Editor*. O autômato global e cada diagnosticador são inseridos como *templates* distintos. Também foram criados *templates* para representar o comportamento dos sensores. A fim de compor uma rede de autômatos temporizados, todos os *templates* devem ser declarados como um sistema no campo *system declarations*. Todas as variáveis relacionadas aos estados dos sensores, atuadores, tipos de faltas, *clock* e canais de comunicação (eventos sincronizantes) devem ser declarados como variáveis globais. Após realizado estas configurações, o sistema pode ser simulado.

A simulação da dinâmica do processo (autômato G), comportamento dos diagnosticadores e sensores, foi realizada de duas formas distintas. Uma forma empregou transições controladas deterministicamente, isto é, a evolução do autômato G e as mudanças de estados dos sensores (ligado ou desligado) foram determinadas arbitrariamente. O intuito foi o de testar todas as situações de falhas possíveis. A outra forma de simulação foi através de transições automáticas e não determinísticas, ou seja, o comportamento dos sensores e as mudanças de estados do autômato G eram aleatoriamente realizados pelo programa. Os dois procedimentos alcançaram resultados similares. É importante ressaltar que não foi considerada a possibilidade de falha dos dispositivos em sua primeira solicitação pelo processo. Isto é, as estações partem com todos os sensores funcionando, pois pressupõe-se que houve o comissionamento do sistema.

Para exemplificar como os diagnosticadores detectam falhas no ambiente de simulação, sem perda de generalidade, serão apresentadas figuras para demonstrar a detecção e isolamento de apenas uma falta permanente pertencente à estação de distribuição. A simulação de qualquer outra falta desta estação ou de outra, seguiu o mesmo procedimento descrito anteriormente e gerou resultados que serão discutidos sinteticamente nas próximas seções.

4.3.1 Método de detecção de falhas proposto por TRIPAKIS (2002)

O método proposto em TRIPAKIS (2002) foi aplicado somente à estação de distribuição. Ele obteve sucesso na detecção de todas as falhas simuladas por intermédio de atrasos de transição. Entretanto, este método não é capaz de isolar o dispositivo em falha e só é aplicável para faltas permanentes. A vantagem deste diagnosticador reside em sua fácil implementação e o baixo consumo de processamento exigido.

4.3.2 Método de detecção de faltas permanentes

O método proposto para diagnóstico de faltas permanentes foi aplicado a todas as estações tratadas no Capítulo 3. Todas as faltas permanentes simuladas em operações compostas por um atuador e um par de sensores, arranjo semelhante ao da figura 4.2, foram detectadas e isoladas. Nos casos em que havia somente um sensor, as falhas foram somente detectadas. Este método resulta em autômatos G e diagnosticadores mais complexos, contudo capazes de realizar detecção e isolamento das falhas, ao invés de somente detecção, como ocorre no método proposto em TRIPAKIS (2002).

4.3.3 Método de detecção de faltas intermitentes

O método proposto para diagnóstico de faltas intermitentes e permanentes foi aplicado somente a operação de recuar o atuador linear, pertencente à estação de distribuição. Para uma simulação mais completa foram considerados também os estados compreendidos entre o recuo e avanço do atuador linear. Estes estados adicionais não foram exibidos na Figura 4.18, entretanto foram incluídos na simulação para abranger todos os possíveis comportamentos de falhas intermitentes discutidos na Seção 4.2.5.2.

As faltas intermitentes simuladas nos sensores foram todas detectadas e isoladas pelos diagnosticadores. As falhas no atuador também foram diagnosticadas, porém restritas ao comportamento em que o atuador funciona depois de no mínimo 3 segundos parado (condição discutida na Seção 4.2.5.2)..

A seguir será apresentada a construção do autômato G de toda a estação de distribuição, a partir das etapas da sequência de funcionamento indicada no capítulo 3, aplicando o método de diagnóstico de faltas permanentes.

4.3.4 Aplicação do método de detecção de faltas permanentes e construção detalhada do autômato G de todo o processo da Estação de Distribuição

A partir das etapas e definições indicadas na seção 4.1 é possível construir o autômato G da estação de distribuição baseando-se na sua sequência de funcionamento. Inicialmente é necessário definir todas as variáveis que serão associadas aos sensores e atuadores. As falhas são definidas na medida em que as operações, pertencente ao autômato G, são modeladas. A tabela 4.1, exibe todas as variáveis associadas aos sensores e atuadores da estação, assim como as falhas e as suas respectivas expressões booleanas de detecção. A declaração de todas as variáveis globais da estação, no programa UPPAAL, estão exibidas nas Figuras 4.22 e 4.23. Todas as variáveis associadas aos sensores, atuadores e tipos de falta foram declaradas como variáveis do tipo inteiras. Os eventos indicadores de falha específica foram declarados como canais de comunicação (*chan*) e os eventos indicadores de falha indefinida como canais de comunicação (*broadcast chan*). Os canais de comunicação tipo *chan* são usados quando existe apenas um par emissor/receptor. Por exemplo, considere que o evento indicador de falha específica IF1! é disparado no autômato G. Somente no diagnosticador da falha 1 existirá o evento IF1?, que estará sincronizado com IF1!. Isto garante que somente o diagnosticador da falha 1 seja capaz de observar o evento indicador de falha específica IF1! e detectar esta falha. No caso dos canais tipo *broadcast chan*, há um evento (ação sincronizante) emissor para mais de um receptor. Por exemplo, o evento IFIpf123? está presente nos diagnosticadores das falhas 1, 2 e 3 e está sincronizado com o evento IFIpf123! do autômato G. Desta forma, o evento indicador de falha indefinida IFIpf123 deve ser declarado como um canal do tipo *broadcast chan*.

O autômato G, modelado do estado 1 ao 15, está exibido na figura 4.24. No primeiro estado (E1Estacaoligada) é feito a verificação se a estação está na posição inicial. A posição inicial da estação de distribuição é a configuração atuador linear recuado e braço rotativo na posição "*magazine*". A guarda associada ao arco que liga o estado (E1Estacaoligada) ao estado E2, verifica se o sensor atuador linear avançado ou o sensor braço rotativo na posição magazine estão ligados, representado pela expressão "t<3&&(SAav==1 or SBpe==1)". Em funcionamento normal esta verificação deve ser executada em menos de 3 segundos (período arbitrariamente definido para ser muito

superior ao tempo médio esperado para operação). Caso o autômato G permaneça no estado (E1Estacaoligada) por mais de 3 segundos, significa que os sensores do atuador linear e braço rotativo estão em falta, mas estas faltas não serão detectadas pelo autômato neste momento pois, conforme já explicado, considera-se que a estação entra em funcionamento com todos os dispositivos funcionando. Então, para este caso, o usuário deverá desligar a estação e comissionar corretamente os dispositivos e demais componentes da planta. Esta condição vale para a modelagem das demais estações. Caso o sensor atuador avançado esteja habilitado ou o sensor braço rotativo esteja na posição *magazine*, o autômato transitará para o estado E2. Se a estação já estiver na configuração que representa a posição inicial, a guarda que representa esta condição, isto é, "t<3&&(SAre==1&SBM==1)" será habilitada levando o autômato do estado (E1Estacaoligada) ao estado E15. Vale lembrar que a posição inicial é a configuração sensor atuador recuado ligado, indicando que o atuador linear está recuado, e sensor braço na posição magazine habilitado, indicando que o braço está na posição "*magazine*". Nas transições (E1Estacaoligada) para E2 ou (E1Estacaoligada) para E15, ocorre o *reset* da variável *clock* t.

Conforme já exposto na metodologia, após habilitada uma transição cuja guarda possui a variável *clock*, é executado o comando *reset* nesta variável. A expressão que realiza o *reset* da variável *clock*, em todas as modelagens, é "t:=0" ou "t=0". No estado E2, há um auto-laço cuja guarda é "RP==0". Este auto-laço é utilizado para manter o autômato no estado E2 enquanto o botão *reset* da estação não é pressionado, isto é, "RP==0". Quando o usuário apertar o botão de *reset*, a guarda "RP==1" será habilitada fazendo com que o autômato transite para o estado E3. A partir deste momento, o controlador irá enviar os comandos necessários para levar a estação para a configuração de posição inicial. A transição que conecta o estado E3 ao E4, é uma atribuição a variável inteira RAL, que representa o comando do controlador para o atuador linear recuar. A expressão "RAL=1" executa a ação de habilitar a variável RAL, que fisicamente é dado por um sinal de tensão que chegará até o motor do atuador linear, fazendo com que o mesmo ligue e recue o atuador. Caso o autômato permaneça, no mínimo, 3 segundos no estado E4 significa que houve uma de três possíveis falhas.

A explicação de como são detectadas as faltas da operação de recuar o atuador linear foi abordada na metodologia. As demais faltas da estação seguem o mesmo raciocínio, mudando apenas as expressões de detecção. Sendo assim, as detecções das

próximas faltas não serão explicadas detalhadamente, caso a caso. A guarda do arco que conecta o estado E4 ao estado E10 representa a condição esperada dos sensores SAav (sensor atuador avançado) e SAre (sensor atuador recuado) após o atuador linear ter recuado. Na operação de recuar o atuador linear, o sensor atuador avançado será desligado e o sensor atuador recuado será ligado. Após habilitada a referida guarda "t<3&&(SAav==0&SAre==1)", a variável RAL será *resetada*, ou seja, será desligado o sinal para recuar o atuador linear e ao mesmo tempo o controlador deverá enviar o comando para mover o braço para magazine, isto é, "MBM:=1". Sempre que possível, concomitante ao *reset* de uma variável que representa um comando para um atuador, será ativada a variável que representa o comando para o próximo atuador da sequência de funcionamento. No estado E10 poderá ocorrer as falhas F4, F5 ou F6 (ver tabela 4.1). Caso não ocorram falhas, o sensor de braço rotativo na posição "*magazine*" será habilitado e o sensor braço rotativo na posição "próxima estação" será desabilitado, logo, a guarda "t<3&&(SBpe==0&SBM==1)" será satisfeita fazendo com que o autômato transite para o estado E15, *resetando* as variáveis *clock* e MBM. No estado E15 será verificado se o botão *start* foi pressionado "SP==1" ou não "SP==0".

Continuando a interpretação do autômato G, a figura 4.25 exibe desde o estado E16 até o estado E28. O autômato permanece no estado E15 até que o botão *start* seja pressionado, levando o autômato para o estado E16. A partir do estado E15 começa a sequência de funcionamento da estação, conforme indicado no manual. As condições indicadas na etapa 1 da sequência de funcionamento da estação de distribuição, são representadas pelas guardas dos arcos que conectam o estado E15 ao E16 e E16 ao E18. A restrição botão *start* pressionado "SP==1" e presença de peças no *stack magazine* "Speca==1" resultará no comando "MBpe:=1", indicado no arco que conecta E16 ao E18, fazendo com que o braço rotativo movimente-se para a posição de "próxima estação". Caso não haja peças no *stack magazine*, o autômato sairá do estado E16 para o estado E17 e permanecerá neste estado até que sejam inseridas peças "Speca==1". Da mesma forma, no arco que conecta o estado E17 ao E18 também será executado a habilitação da variável MBpe, assim que a variável Speca for habilitada. No estado E18 poderá ocorrer as falhas F7, F8 ou F9. Se não houver falhas a guarda "t<3&&(SBM==0&SBpe==1)" será habilitada, indicando que o braço rotativo se encontra na posição "próxima estação". Nesta operação o sensor braço na posição "*magazine*" foi desligado "SBM==0" e o sensor braço rotativo na posição "próxima

estação” foi ligado “SBpe==1”. Ainda no arco que conecta o estado E18 ao estado E23 é executado o comando avançar atuador linear, representado pela atribuição “AAL:=1”. Esta operação é a segunda etapa da sequência de funcionamento, conforme pode ser visto no capítulo 3. O atuador linear avança com o objetivo de retirar uma peça da posição mais inferior do *stack magazine*.

No estado E23 é possível que ocorra as falhas F10, F11 ou F12. Em funcionamento normal, após o comando “AAL:=1” ocorrerá o desligamento do sensor atuador linear recuado “SAre==0” e o sensor atuador linear avançado será habilitado “SAav==1”. Esta condição implica que a guarda “t<3&&(SAre==0&SAav==1)” foi satisfeita e os comandos para recuar o atuador linear “RAL:=1” e *resetar* as variáveis AAL e t, tenham sido executados. O comando “RAL:=1” representa a etapa 3 da sequência de funcionamento. Em seguida, o autômato alcança o estado E28, onde pode ocorrer novamente as falhas F1, F2 ou F3. Neste estado, há um arco que o conecta ao estado E5 onde ocorre a detecção das falhas F1, F2 ou F3. Sendo assim, o mesmo evento indicador de falta indefinida, as mesmas restrições e atribuições que existem no arco que conecta E4 ao E5, estão presentes no arco que liga E28 ao E5. Isto, tem como objetivo não repetir estados de detecção de falta que já estão presentes no autômato. O arco que interliga o estado E28 ao estado E29 possui a guarda “t<3&&(SAav==0&SAre==1)” que representa o comportamento esperado dos sensores atuador linear recuado, atuador linear avançado e indica que o atuador realmente recuou. O comando mover braço rotativo para posição “*magazine*” “MBM:=1” é executado após a guarda do arco que conecta E28 ao E29 ter sido satisfeita, representando a etapa 4 da sequência de funcionamento da estação.

O último conjunto de estados do autômato G, estado E29 até E36, está exibido na figura 4.26. No estado E29 pode ocorrer também as falhas F4, F5 ou F6. Caso não ocorram, o sensor braço rotativo na posição “*magazine*” será habilitado e o sensor braço rotativo na posição de “próxima estação” será desabilitado. Esta condição está expressa na guarda “t<3&&(SBpe==0&SBM==1)”. Uma vez que o braço rotativo está na posição “*magazine*” e há uma peça na posição de espera para ser coletada, o próximo passo da sequência de funcionamento da estação é ligar o atuador de vácuo para aprisionar a peça de trabalho no braço rotativo. Esta condição é representada pela etapa 5, e o comando de ligar vácuo “LDV:=1” está presente no arco que conecta o estado E29 ao estado E30. No estado E30 pode ocorrer uma falha no sensor de peça capturada

ou no atuador de vácuo. É possível detectar que houve uma falha, mas não é possível isolar alguma destas duas falhas, conforme já foi tratado no capítulo 4. Caso a peça tenha sido capturada normalmente, a guarda “t<3&&SPseg==1” será satisfeita e o comando para mover o braço rotativo para a posição “próxima estação” será executado, isto é, “MBpe:=1”. Vale ressaltar que neste arco a variável que representa o comando ligar ou desligar o vácuo (LDV) não será resetada, pois caso o fosse, a peça seria liberada antes de chegar na posição de “próxima estação”.

A etapa 6 da sequência de funcionamento da estação, foi executada na transição que conecta o estado E30 ao E32, isto é, a peça foi transportada da posição “*magazine*” para a posição “próxima estação”. No estado E32 pode ocorrer as falhas F7, F8 ou F9. Considerando que o braço rotativo não apresentou falha, o sensor braço rotativo na posição “próxima estação” foi habilitado e o sensor braço rotativo na posição “*magazine*” foi desligado. Sendo assim, a guarda “t<3&&(SBM==0&SBpe==1)” foi satisfeita e consequentemente o atuador de vácuo recebeu o comando para ser desligado “LDV:=0”. Esta operação representa a etapa 7 da sequência de funcionamento da estação. Finalmente o autômato alcança o estado E33, onde pode ocorrer uma falha no atuador de vácuo travado no estado ligado ou sensor de peça detectada travado no estado ligado. Não é possível isolar estas faltas. Caso não ocorram estas faltas, a guarda “t<3&&SPseg==0” será satisfeita indicando que a peça foi liberada na posição de “próxima estação” e o autômato chegará ao último estado (E35Fimdociclo).

Se o botão automático do painel de controle estiver habilitado, representado pela guarda “Automatico==1”, o autômato transitará para o estado E15 fazendo com que seja reiniciado o ciclo de funcionamento. Caso o botão automático esteja desabilitado “Automatico==0” a estação finalizará o seu ciclo de funcionamento.

A simulação da ocorrência da falha 2, isto é, sensor de posição atuador avançado travado no estado ligado, está exibida nas figuras 4.27, 4.28 e 4.29. A Figura 4.27 demonstra o evento indicador de falha indefinida (IFIpf123) sendo disparado (em vermelho) devido a permanência do autômato no estado (E4) durante 3s. A Figura 4.28 exibe o caminho percorrido pelo autômato G, em azul claro, até chegar ao estado E7, onde o evento indicador da falha 2 é anunciado em intervalos de 1s. A Figura 4.29 demonstra o estado dos diagnosticadores 1, 2 e 3 após o diagnóstico da falha 2.

Tabela 4.1- Variáveis e faltas da estação de distribuição

Variável	Descrição da variável	Falta associada	Descrição da falta/ indicador	Expressão booleana de detecção
SAre	Sensor atuador linear recuado	F3	Travado no estado desligado/ IF3!	Após RAL=1; (SAav==0&SAre==0)
		F11	Travado no estado ligado/ IF11!	Após AAL=1; (SAav==1&SAre==1)
SAav	Sensor atuador linear avançado	F2	Travado no estado ligado/ IF2!	Após RAL=1; (SAav==1&SAre==1)
		F12	Travado no estado desligado/ IF12!	Após AAL=1; (SAre==0&SAav==0)
RAL	Comando Recuar atuador linear	F10	Atuador linear travado na posição recuada/ IF10!	Após AAL=1; (SAre==1&SAav==0)
AAL	Comando avançar atuador linear	F1	Atuador linear travado na posição avançada/ F1!	Após RAL=1; (SAav==1&SAre==0)
SBpe	Sensor braço rotativo na próxima estação	F5	Travado no estado ligado/ IF5!	Após MBM=1; (SBpe==1&SBM==1)
		F9	Travado no estado desligado/ IF9!	Após MBpe=1; (SBM==0&SBpe==0)
SBM	Sensor braço rotativo na Magazine	F6	Travado no estado desligado/ IF6!	Após MBM=1; (SBpe==0&SBM==0)
		F8	Travado no estado ligado/ IF8!	Após MBpe=1; (SBM==1&SBpe==1)
MBM	Mover braço para Magazine	F7	Atuador braço rotativo travado na posição magazine/ IF7!	Após MBpe=1; (SBM==1&SBpe==0)
MBpe	Mover braço para próxima estação	F4	Atuador braço rotativo travado na posição próxima estação/ IF4!	Após MBM=1; (SBM==1&SBpe==0)
LDV	Comando Ligar/Desligar vácuo	FI13	Falta no sensor de peça travado no estado desligado ou atuador de vácuo travado no estado desligado/ IFIpf13!	Após LDV=1; t==3&&SPseg==0

		FI14	Falta no sensor de peça travado no estado ligado ou atuador de vácuo travado no estado ligado/ IFIpf14!	Após LDV=0; t==3&&SPseg==1
R	Botão Reset	-	-	-
S	Botão Start	-	-	-

```
// Place global declarations here.
int [0,1] SAre;
int [0,1] SBM;
int [0,1] SPseg;
int [0,1]Speca;
clock t;
int [0,1] RAL;
int [0,1] MBM;
int [0,1] MBpe;
int [0,1] LDV;
int [0,1] AAL;
int [0,1] SBpe=1;
int [0,1] SAav;
int [0,1] SP=1;
int [0,1] RP=1;
int [0,1] Automatico=1;
broadcast chan IFIpf123;
broadcast chan IFIpf456;
broadcast chan IFIpf789;
broadcast chan IFIpf101112;
broadcast chan IFIpf13;// Pode indicar uma falha no atuador de vácuo (travado no estado desligado) ou no sensor de peça capturada (travado em 0)
broadcast chan IFIpf14;// Pode indicar uma falha no atuador de vácuo (travado no estado ligado) ou no sensor de peça capturada (travado em 1)
chan IF1;
chan IF2;
chan IF3;
chan IF4;
chan IF5;
chan IF6;
chan IF7;
chan IF8;
chan IF9;
chan IF10;
```

Figura 4.22 - Declaração das variáveis globais da estação de distribuição (parte 1).

```
chan IF8;
chan IF9;
chan IF10;
chan IF11;
chan IF12;
int [0,1] F1; // Falha atuador linear travado na posição avançada.
int [0,1] F2; // Falha sensor atuador avançado travado no estado ligado.
int [0,1] F3; // Falha sensor atuador recuado travado no estado desligado.
int [0,1] F4; // Falha atuador braço rotativo travado na posição de próxima estação.
int [0,1] F5; // Falha sensor braço rotativo na posição próxima estação travado no
estado ligado.
int [0,1] F6; // Falha sensor braço rotativo na posição magazine travado no estado
desligado.
int [0,1] F7; // Falha atuador braço rotativo travado na posição magazine.
int [0,1] F8; // Falha sensor braço rotativo na posição magazine travado no estado
ligado.
int [0,1] F9; // Falha sensor braço rotativo na posição próxima estação travado no
estado desligado.
int [0,1] F10; // Falha atuador linear travado na posição recuada.
int [0,1] F11; // Falha sensor atuador linear recuada travado no estado ligado.
int [0,1] F12; // Falha sensor atuador linear avançado travado no estado desligado.
int [0,1] FI13; //pode indicar uma falha no atuador de vácuo (travado no estado
desligado) ou no sensor de peça capturada (travado em 0)
int [0,1] FI14; // Pode indicar uma falha no atuador de vácuo (travado no estado
ligado) ou no sensor de peça capturada (travado em 1)
```

Figura 4.23 - Declaração das variáveis globais da estação de distribuição (parte 2).

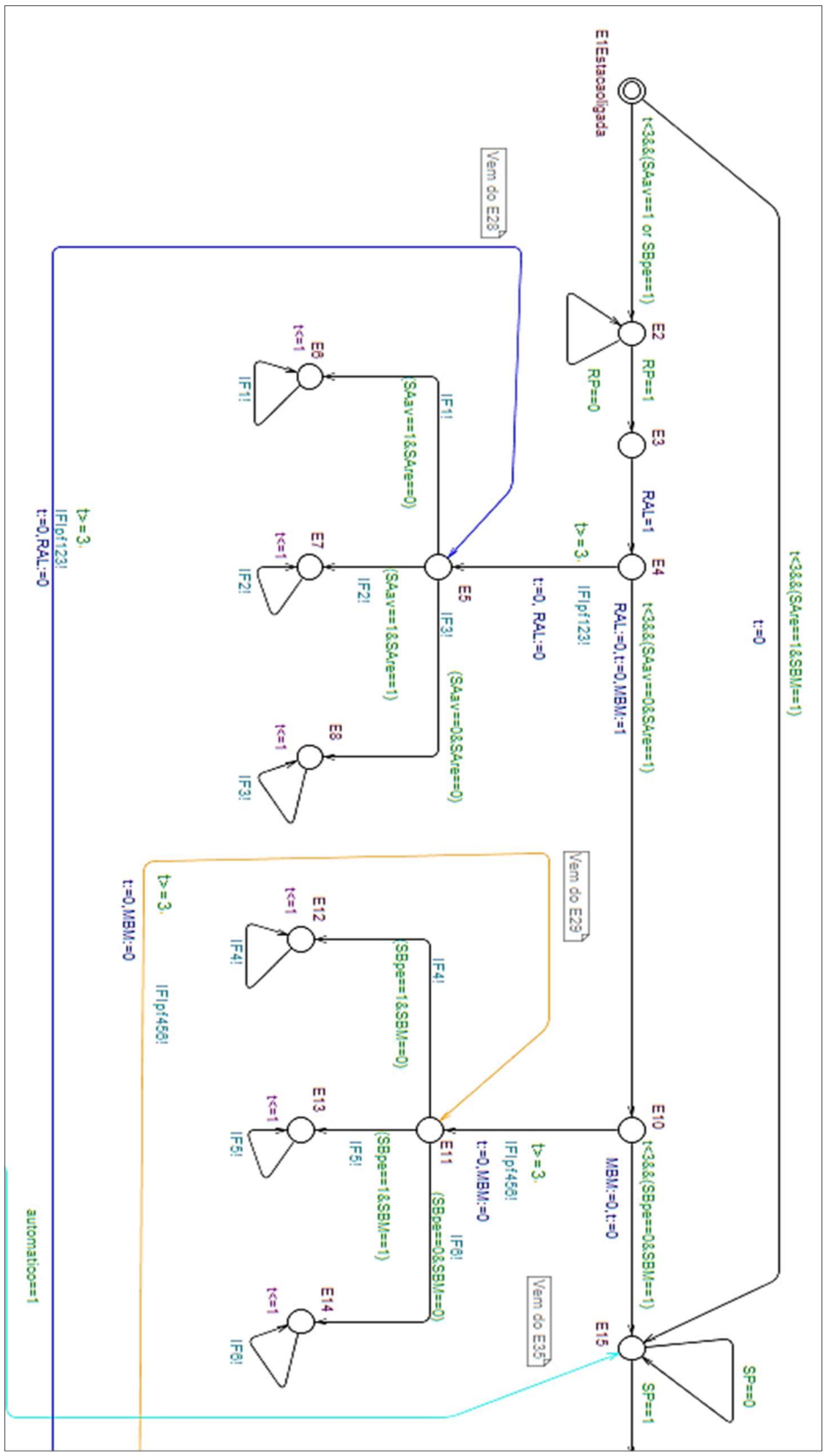

Figura 4.24 - Autômato G da estação de distribuição (estado 1 ao 15).

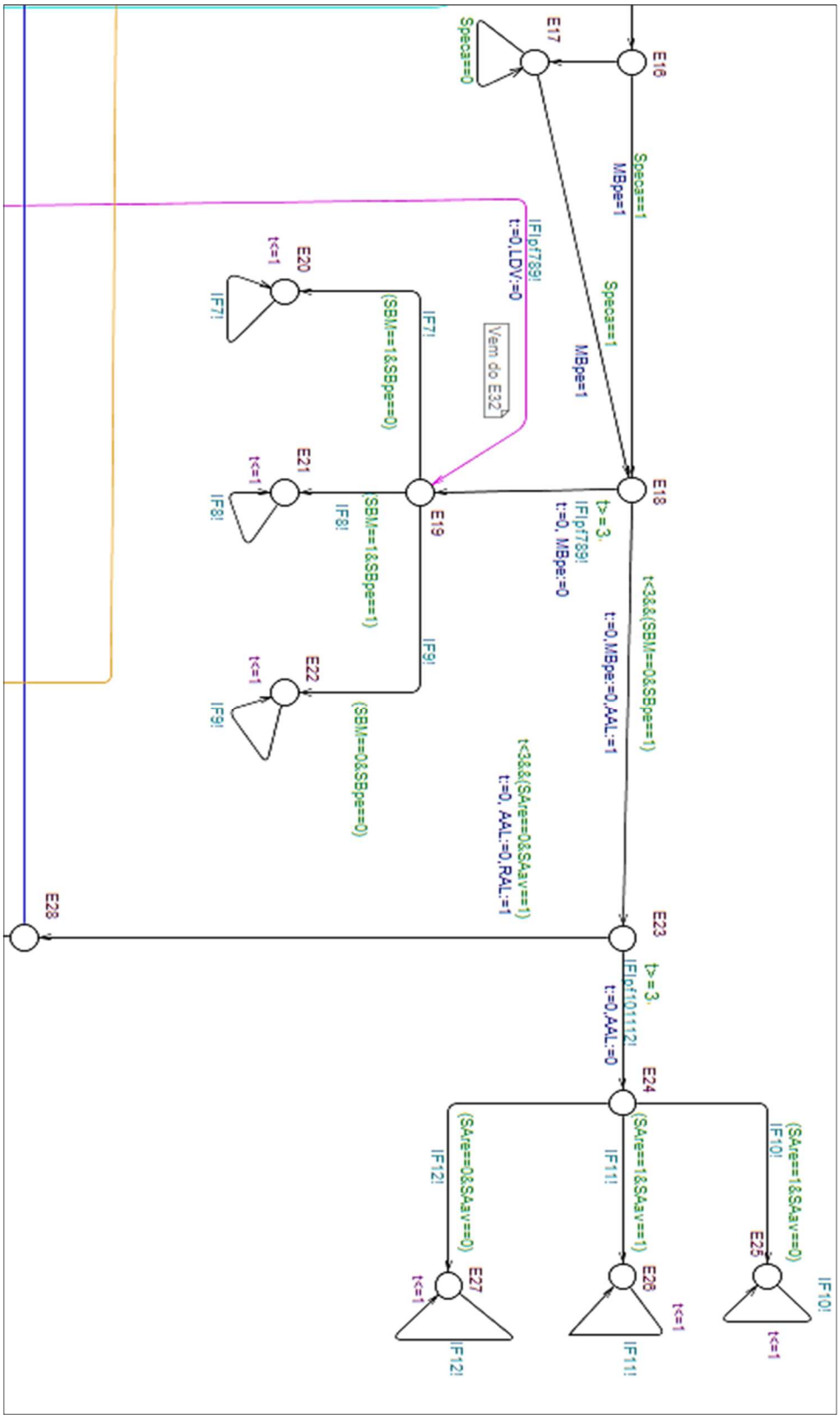

Figura 4.25 - Autômato G da estação de distribuição (estado 16 ao 28).

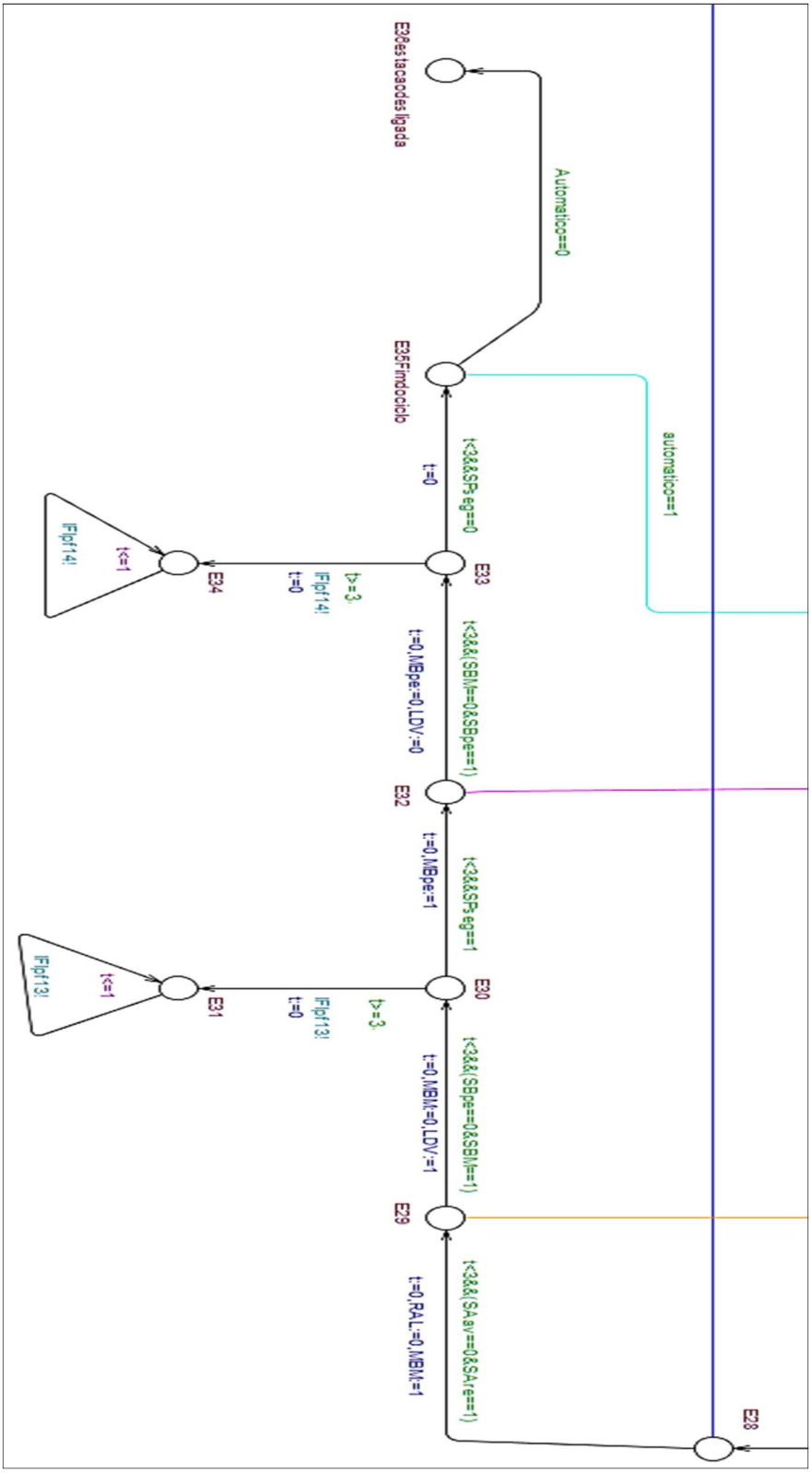

Figura 4.26 - Autômato G da estação de distribuição (estado 29 ao 36).

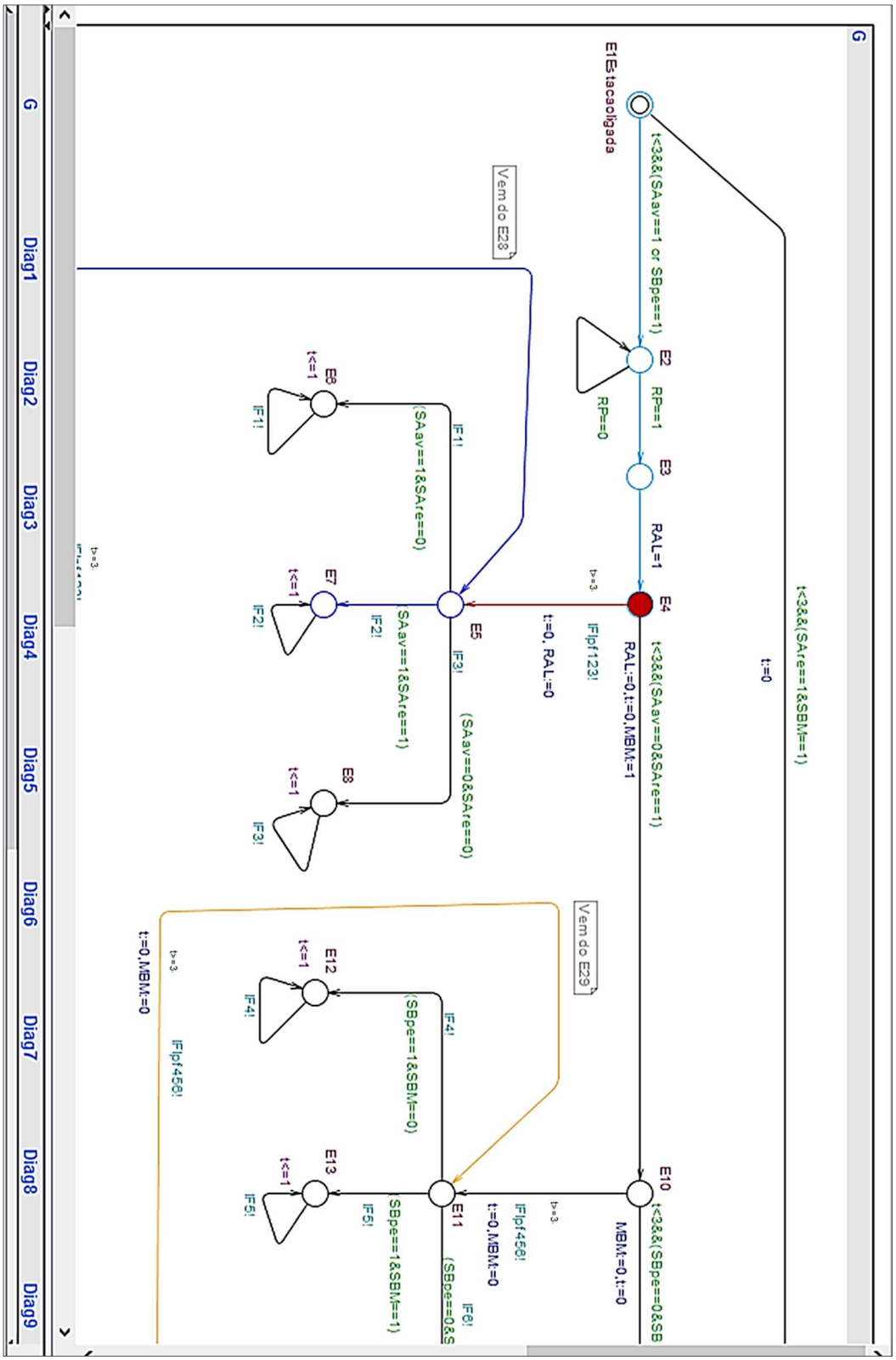

Figura 4.27 - Disparo do evento indicador de falha indefinida (pf123) (Estação de Distribuição).

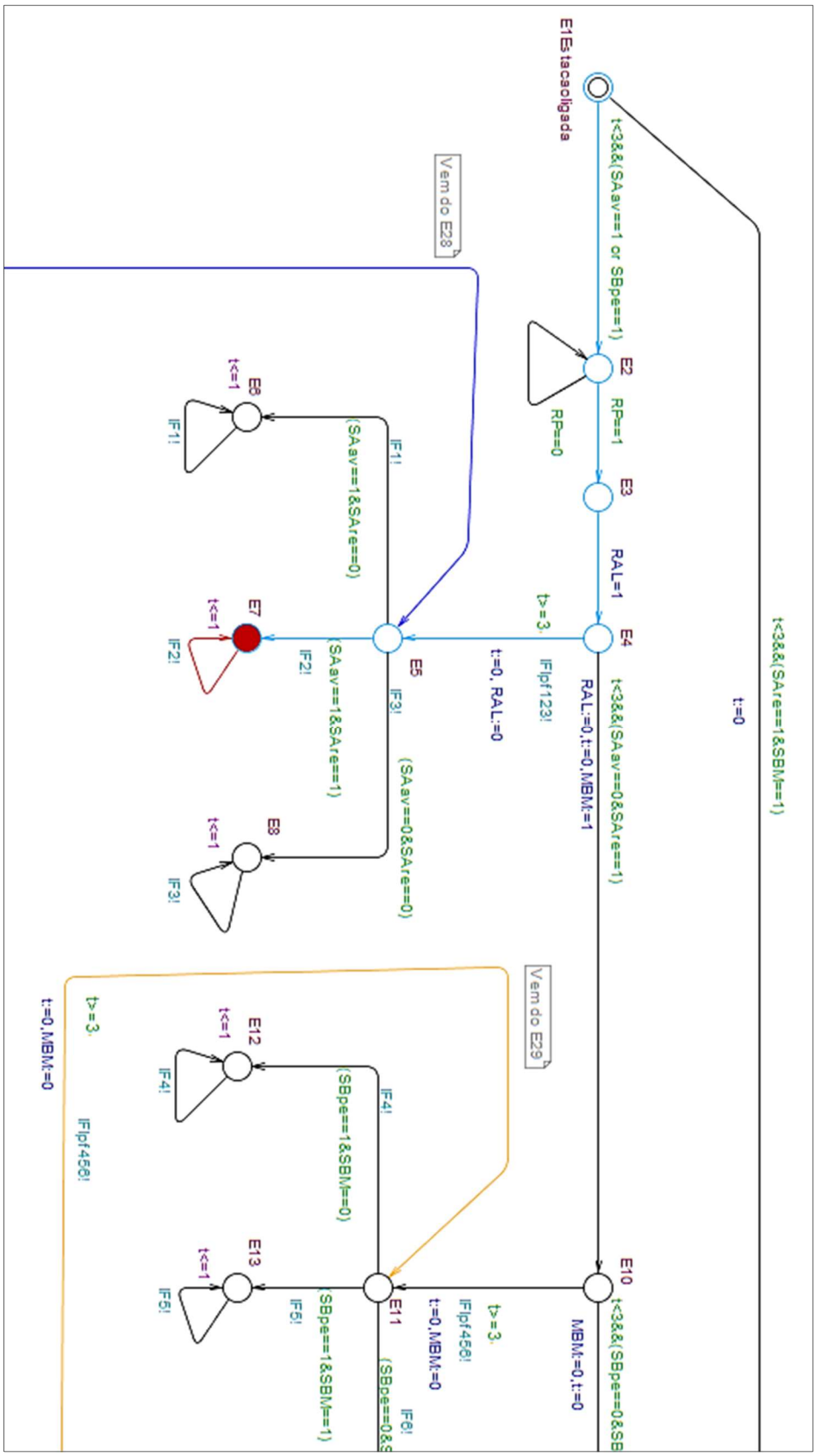

Figura 4.28 Disparo e isolamento do evento indicador de falha específica (IF2) (Estação de Distribuição).

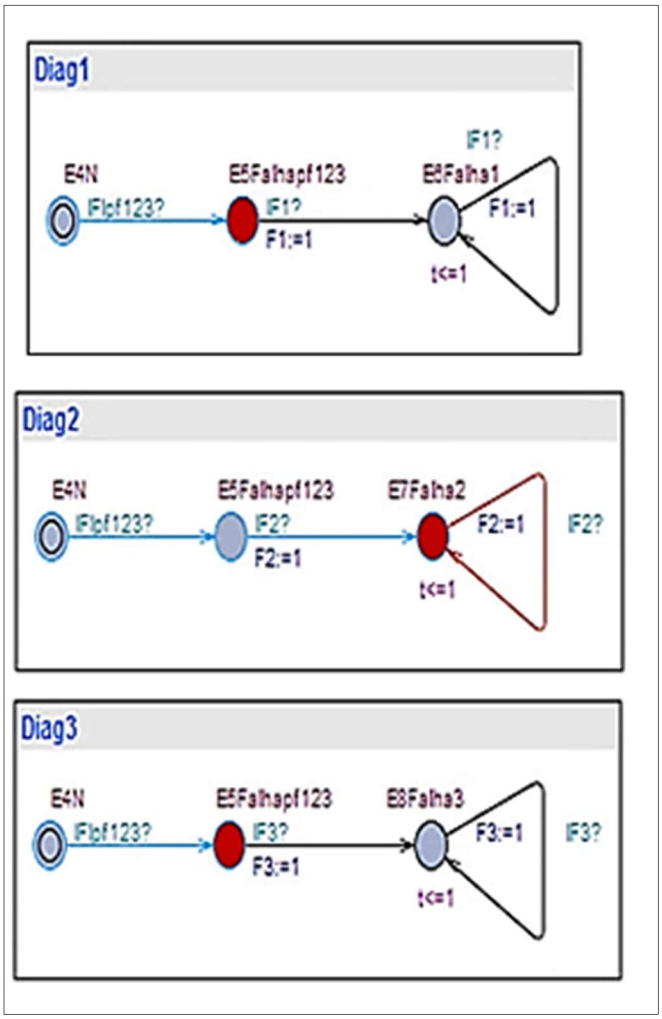

Figura 4.29 - Diagnosticadores 1, 2 e 3 da estação de distribuição (detecção da falta 2 pelo diagnosticador 2).

O autômato global, sujeito a faltas permanentes, das demais estações estão exibidos nos anexos. Os diagnosticadores de falhas não serão exibidos, pois são similares aos diagnosticadores da estação de distribuição.

No anexo 1 exibe o autômato global e a tabela de variáveis construídos para a Estação de separação. O anexo 2 exibe o autômato global e a tabela de variáveis construídos para Estação de Coleta e encaixe. O anexo 3 exibe o autômato global e a tabela de variáveis construídos para a Estação de Músculo Fluido. O anexo 4 exibe o autômato global e a tabela de variáveis construídos para a Estação de Classificação.

5 Conclusões e Recomendações

A motivação deste trabalho foi modelar, simular e realizar o diagnóstico on-line de falhas induzidas em um sistema modular de manufatura didático. As falhas a serem diagnosticadas estavam concentradas nos sensores e atuadores do sistema e podiam ser do tipo permanente ou intermitente. O diagnosticador proposto em TRIPAKIS (2002), baseado em autômatos temporizados, foi adaptado para linguagem UPPAAL e aplicado a um sistema e operação típicos dos processos abordados no estudo de caso. Foi demonstrado que este diagnosticador não é capaz de isolar falhas e é adequado para detectar somente falhas permanentes.

Com base no diagnosticador apresentado em TRIPAKIS (2002) e no conceito de I-diagnosticabilidade (SAMPATH *et al.*, 1995), foram elaborados dois métodos, na linguagem do UPPAAL, para diagnosticar as falhas referidas anteriormente. O primeiro método é capaz de diagnosticar falhas do tipo permanentes. O segundo método resultou de modificações inseridas no primeiro método para torná-lo adequado ao diagnóstico de falhas permanentes ou intermitentes. Ambos métodos são eficientes na detecção e isolamento das falhas para o qual foram projetados, considerando sistemas em que o atuador tem seus estados monitorados antes e depois de realizar uma operação (Ver Seção 4.2). Para configurações de dispositivos que não atendem a condição citada anteriormente, as falhas não poderão ser isoladas por nenhum dos métodos. Há também um tipo de comportamento de falha intermitente, em atuadores, que não é diagnosticável pelo método elaborado neste trabalho. Este caso foi discutido na Seção 4.2.5.2.

Os métodos de diagnóstico de falhas apresentados podem ser utilizados em simulações ou até mesmo implementados em controladores de diversos processos ou sub-processos de plantas de manufatura pois, de um modo geral, há muitas operações com atuadores e sensores de estados discretos (ligado ou desligado), nestes sistemas. Antes de implementá-los é necessário converter a linguagem de autômatos em qualquer uma das linguagens de programação definidas pela IEC- 61131-3, isto é, Lista de instruções, Diagrama de contatos, Texto estruturado, Diagrama de blocos funcionais e Cartas de funções sequenciais. Outra forma de implementação, seria utilizar um software equivalente ao UPPAAL, mas com a possibilidade de se comunicar, em tempo

real, com o controlador lógico programável da planta. As entradas e saídas do controlador estariam associadas a variáveis, do tipo inteiras, definidas na modelagem do sistema, sendo que os estados das variáveis (ligado ou desligado) mudariam conforme os estados dos dispositivos conectados ao controlador lógico programável. As transições do autômato G (processo) estariam relacionadas aos estados dos dispositivos e a sequência de operação da planta. Desta forma, este software atuaria como um sistema supervisório, mas com um diagnosticador *on-line* incorporado.

Para trabalhos futuros, a principal proposta é aplicar e adaptar os métodos de diagnostico apresentados neste trabalho a um sistema de tempo contínuo. Outra proposta é aplicar técnicas de verificação de modelos para simular o maior número possível de comportamentos de faltas através de uma abordagem *off-line*.

6 Referências

ALUR, R. E DILL, D. L. (1994).*A Theory Of Timed Automata*. Theoretical Computer Science, 126(2):183–235.

ALUR, RAJEEV; COURCOUBETIS,COSTAS E HENZINGER, THOMAS A.(1994). *The Observational Power Of Clocks.* In International Conference on Concurrency Theory, pages 162–177, 1994.

ANDIA, Angela Catarine Arana (2014). *Diagnóstico de Falhas em um Forno Aquecedor de Óleo de uma Planta de Tratamento de Hidrocarbonetos.* Dissertação de Mestrado. Universidade Federal do Rio de Janeiro.

BASILE, F., CHIACCHIO, P., & DE TOMMASI, G. (2008). *Sufficient Conditions For Diagnosability Of Petri Nets*. In Proc. 9th international workshop on discrete event systems (WODES'08) (pp. 436–442).

BASILE, F., CHIACCHIO, P., & DE TOMMASI, G. (2009). *An Efficient Approach For Online Diagnosis Of Discrete Event Systems.* IEEE Transactions Automatic Control, 54(4),748–759.

BASILIO, J. C., CARVALHO, L. K., MOREIRA, M. V. (2010). *Diagnose de falhas em sistemas a eventos discretos*. Revista Controle e Automação, SBA.

BAVISHI, S., & CHONG, E. (1994).*Automated Fault Diagnosis Using A Discrete Event Systems Framework.* In Proc. 9th IEEE int. symp. intelligent contr. (pp. 213–218).

BENGTSSON, J. E YI, W. (2004).*Timed automata: Semantics, Algorithms and Tools*. Theoretical Computer Science, 126(2):183–235.

BENVENISTE, A., FABRE, E., HAAR, S., & JARD, C. (2003). *Diagnosis Of Asynchronous Discrete Event Systems*: A net unfolding approach. IEEE Transactions Automatic Control, 48(5), 714–727.

BEHRMANN, G, DAVID, A, LARSEN, KG.(2006).*A tutorial on UPPAAL 4.0*. Denmark: Aalborg University.

BOEL, R.K., & VAN SCHUPPEN, J.H. (2002). *Decentralized Failure Diagnosis For Discrete event Systems With Costly Communication Between Diagnosers* .In Proc. 6thinternational workshop on discrete event systems (WODES'02).

BOUYER, P., CHEVALIER, F., & D'SOUZA, D. (2005).*Fault Diagnosis Using Timed Automata*. In Proc. 8th International Conference on Foundations of Software Science and Computation Structures. LNCS (3441, pp. 219–233). Springer.

CABASINO, M. P., GIUA, A., POSSI, M., & SEATZU, C. (2011). *Discrete Event Diagnosis Using Labeled Petri Nets.* An application to manufacturing systems. Control Engineering Practice, 19(9), 989–1001.

CABASINO, M. P., GIUA, A. N., & SEATZU, C. (2010). *Fault Detection For Discrete Event Systems Using Petri Nets With Unobservable Transitions*. Automatica, 46(9),1531–1539.

CONTANT, O., LAFORTUNE, S., & TENEKETZIS, T. (2004). *Diagnosis Of Intermittent Faults.* Discrete Event Dynamic Systems, 14(2), 171–202.

CARVALHO, L. K. (2011). *Diagnose Robusta de Sistemas a Eventos Discretos.* Tese de Doutorado. Universidade Federal do Rio de Janeiro.

CASSANDRAS, C. G., LAFORTUNE, S. (2008). *Introduction to Discrete Event Systems*.2nd ed. Boston, Kluwer Academic Publishers.

CASSEZ, F. (2009).*A Note On Fault Diagnosis Algorithms*. In Proc. 48th IEEE conference on decision and control and 28th Chinese control conference.

CHAKIB, H., & KHOUMSI, A. (2012). *Multi-Decision Diagnosis: Decentralized Architectures Cooperating For Diagnosing The Presence Of Faults In Discrete Event Systems.* Discrete Event Dynamic Systems, 22(3), 333–380.

CHANG S.-J.; DiCESARE F., GOLDBOGEN, G. (1991). *Failure propagation trees for diagnosis in manufacturing systems*. Transactions on Systems, Man and Cybernetics, v. 21.

CHEN, Y.L., & PROVAN, G. (1997). *Modelling And Diagnosis Of Timed Discrete Event Systems*. In Proc. American control conference (pp. 31–36).

CHAOCHEN, ZHOU (1999). *Duration Calculus, A Logical Approach To Real-Time Systems.* Lecture Notes in Computer Science, 1548:1–7.

CHUNG, S. L. (2005*). Diagnosing PN-Based Models With Partial Observable Transitions*. International Journa of Computer Integrated Manufacturing, 18,158–169.

CURY, J. E. (2001). *Teoria de Controle Supervisório de Sistemas a Eventos Discretos*. V Simpósio de Automação Inteligente.

DEBOUK, R., LAFORTUNE, S., & TENEKETZIS, D. (2000).*Coordinated Decentralized Protocols For Failure Diagnosis Of Discrete Event Systems*. Discrete event dynamic systems, 10, 33–86.

FABRE, E., BENVENISTE, A., HAAR, S., & JARD, C. (2005). *Distributed Monitoring Of Concurrent And Asynchronous Systems.* Discrete Event Dynamic Systems, 15(1), 33–84.

FESTO (2006).*Manual Festo Didatic MPS 200*. Denkendorf.

FRANK, P.(1992). *Principles of model-based fault detection*. In: Proceedings of International Symposium on AI in Real-time Control, Delft.

GENC, S., & LAFORTUNE, S. (2007). *Distributed Diagnosis Of Place-Bordered PetriNets*. IEEE Transactions Automation Science and Engineering, 4(2), 206–219.

GHAZEL, M., BIGAND, M., & TOGUYÉNI, A. (2005). *A Temporal-Constraint Based Approach For Monitoring Of Dess Under Partial Observation*. In Proc. 16th IFAC world congress

HALGAMUGE, S. (1996).*Advanced Methods for Fusion of Fuzzy System and Neural Networks in Intelligent Data Processing*. Fortschr. Ber. VDIReihe 10, Nr. 401, VDI-Verlag - Diisseldorf.

HERNANDEZ-FLORES, E., LOPEZ-MELLADO, E., & RAMIREZ-TREVINO, A. (2011).*Diagnosticability Analysis Of Partially Observable Deadlock-Free Petri Nets*. InProc.3rd int. workshop on dependable control of discrete systems (pp. 176–181).

HOPCROFT, J. E., MOTWANI, R., ULLMAN, J. D (1939). *Introduction to automata theory, languages, and computation.* Pearson Education India.

HSIEH, WEI-CHUN; CHANG, CHUEI-TIN. (2016) *Timed-Automata Based Method For Synthesizing Diagnostic Tests In Batch Processes.* Computers & Chemical Engineering, v. 84, p. 12-27.

IDGHAMISHI, A.M., & ZAD, S.H. (2004*). Fault Diagnosis In Hierarchical Discrete-Event Systems*. In Proc. 43rd IEEE conference on decision and control (pp. 63–68).

ISERMANN, R. e BALLÉ, P. (1997). *Trends in the application of model-based fault detection and diagnosis of technical processes*. Control Engineering Practice, v. 5.

JÉRON, T., MARCHAND, H., PINCHINAT, S., & CORDIER, M. (2006). *Supervision Patterns In Discrete Event Systems Diagnosis.* In Proc. 8th international workshop on discrete event systems (WODES 2006).

JIANG, S., & KUMAR, R. (2006). *Diagnosis Of Repeated Failures For Discrete Event Systems With Linear-Time Temporal-Logic Specifications*. IEEE Transactions Automation Science and Engineering, 3(1), 47–59.

JIROVEANU, G., BOEL, R.K., e DE SCHUTTER, B. (2006). *Fault diagnosis for time Petrinets*. In Proc. WODES 06, Ann Arbor - USA.

LAFORTUNE, S., TENEKETZIS, D., SAMPATH, M., SENGUPTA, R., e SINNAMOHIDEEN, K. (2001). *Failure diagnosis of dynamic systems: An approach based on discrete event systems.* Proceedings of the American Control Conference, Arlington, USA.

Larsen, K. G., Pettersson, P., Yi, W. (1997): *Uppaal In A Nutshell.* Int. Journal on Softwares Tools for Technology Transfer 1(1−2), 134−152.

LIN, F. (1994). *Diagnosability of discrete event systems and its applications*. Discrete Event Dynamic Systems, 4(1), 197–212.

MAAS, Daniel Gumiero Noranha (2004). *Diagnóstico de Falhas Multicamadas de Sistemas Embarcados Modelados por SEDs.* Dissertação de Mestrado. Universidade do Estado de Santa Catarina – UDESC, Joinville.

MARCOVISTZ , Ighor(2014). *Diagnóstico de Falhas no Sistema de Resfriamento de um Sistema de Células a Combustível de Membrana Polimérica.* Dissertação de Mestrado. Universidade Federal do Rio de Janeiro.

MORALES, Roy Andes Gomez. (2009). *Modelagem e Análise de Sistemas Flexíveis de Manufatura Tolerantes à Falhas Baseado em Rede Bayesiana e Rede de Petri.* Dissertação de Mestrado. Escola Politécnica – Universidade Federal de São Paulo, Brasil.

PANDALAI, D. N., & HOLLOWAY, L. E. (2000). *Template Languages For Fault Monitoring Of Timed Discrete Event Processes.* IEEE Transactions Automatic Control, 45(5),868–882.

PAOLI, A., & LAFORTUNE, S. (2008). *Diagnosability Analysis Of A Class Of Hierarchical State Machines*. Discrete Event Dynamic Systems, 18, 385–413.

PENCOLÉ, Y., & SUBIAS, A. (2009).*A Chronicle-Based Diagnosability Approach For Discrete Timed-Event Systems: Application To Web-Services*. Journal of Universal Computer Science, 15(17), 3246–3272.

QIU, W., & KUMAR, R. (2006). *Decentralized Failure Diagnosis Of Discrete Event Systems*. IEEE Transactions on Systems, Man and Cybernetics: Part A, 36(2),628–643.

RAMADGE, P. J.; WONHAM, W. M. (1989). *The Control of Discrete Event Systems*. Proceedings of IEEE, Vol. 77, n. 1, 1989.

SAMPATH, M., SENGUPTA, R., LAFORTUNE, S., SINNAMOHIDDEN, K., & TENEKETZIS, D. (1995). *Diagnosability of discrete event systems.* IEEE Transactions Automatic Control,40, 1555–1575.

SAMPATH, M., SENGUPTA, R., LAFORTUNE, S., SINNAMOHIDEEN, K., & TENEKETZIS, D. (1996). *Failure Diagnosis Using Discrete Event Models*. IEEE TransactionsControlSystemsTechnology, 4(2), 105–124.

SANTOS FILHO, D. J. (2000).*Aspectos do Projeto de Sistemas Produtivos*. Tese de Livre Docência, Escola Politécnica – Universidade de São Paulo.

SAYED-MOUCHAWEH, M., PHILIPPOT, A., & CARRÉ-MÉNÉTRIER, V. (2008).*Decentralized Diagnosis By Boolean Discrete Event System Model: Application On Manufacturing Systems.* International Journal of Production Research, 46(19), 5469–5490.

SU, R., & WONHAM, W.M. (2004).*A Model Of Component Consistency In Distributed Diagnosis.* In Proc. 7th international workshop on discrete event systems(WODES'04) (pp. 427-432).

TEIXEIRA, E. (1993). *Diagnóstico Inteligente de Falhas em um Processo de Separação Óleo-Gás em Plataformas Off shore*. Dissertação de Mestrado,COPPE/UFRJ.

THOMAS A. HENZINGER, XAVIER NICOLLIN, JOSEPH SIFAKIS, E SERGIO YOVINE (1994). *Symbolic Model Checking For Real-Time Systems.* Journal of Information and Computation, 111(2):193–244.

TRIPAKIS, S. (2002). *Fault Diagnosis For Timed Automata. In Proc. Int. Conf. On Formal Techniques In Real Time And Fault Tolerant Systems*. LNCS (Vol. 24, pp. 205–224).Springer.

YI, WANG (1991). *CCS + Time = An Interleaving Model For Real Time Systems*. In Proceedings, Eighteenth International Colloquium on Automata, Languages and Programming, volume 510 of Lecture Notes in Computer Science. Springer-Verlag.

ZAD, S. H., KWONG, R. H., & WONHAM, W. M. (2003). *Fault Diagnosis In Discrete-Event Systems: Framework And Model Reduction*. IEEE Transactions Automatic Control,48(7), 1199–1212.

ZAD, S. H., KWONG, R. H., & WONHAM, W. M. (2005). *Fault Diagnosis In Discrete-Event Systems: Incorporating Timing Information*. IEEE Transactions Automatic Control,50(7), 1010–1015.

ZAYTOON, J. A E LAFORTUNE, S. (2013). *Overview Of Fault Diagnosis Methods For Discrete Event Systems*. Annual Reviews in Control

7 Anexos

7.1 Anexo 1: Estação de Separação

Este anexo exibe o autômato global que modela o comportamento da estação de Separação. Tanto o autômato global quanto todas as possíveis faltas permanentes detectáveis de sensores e atuadores estão exibidos nas Figuras 7.1.1, 7.1.2, 7.1.3, 7.1.4, 7.1.5 e 7.1.6. A tabela 7.1 exibe todas as variáveis da estação e suas faltas associadas.

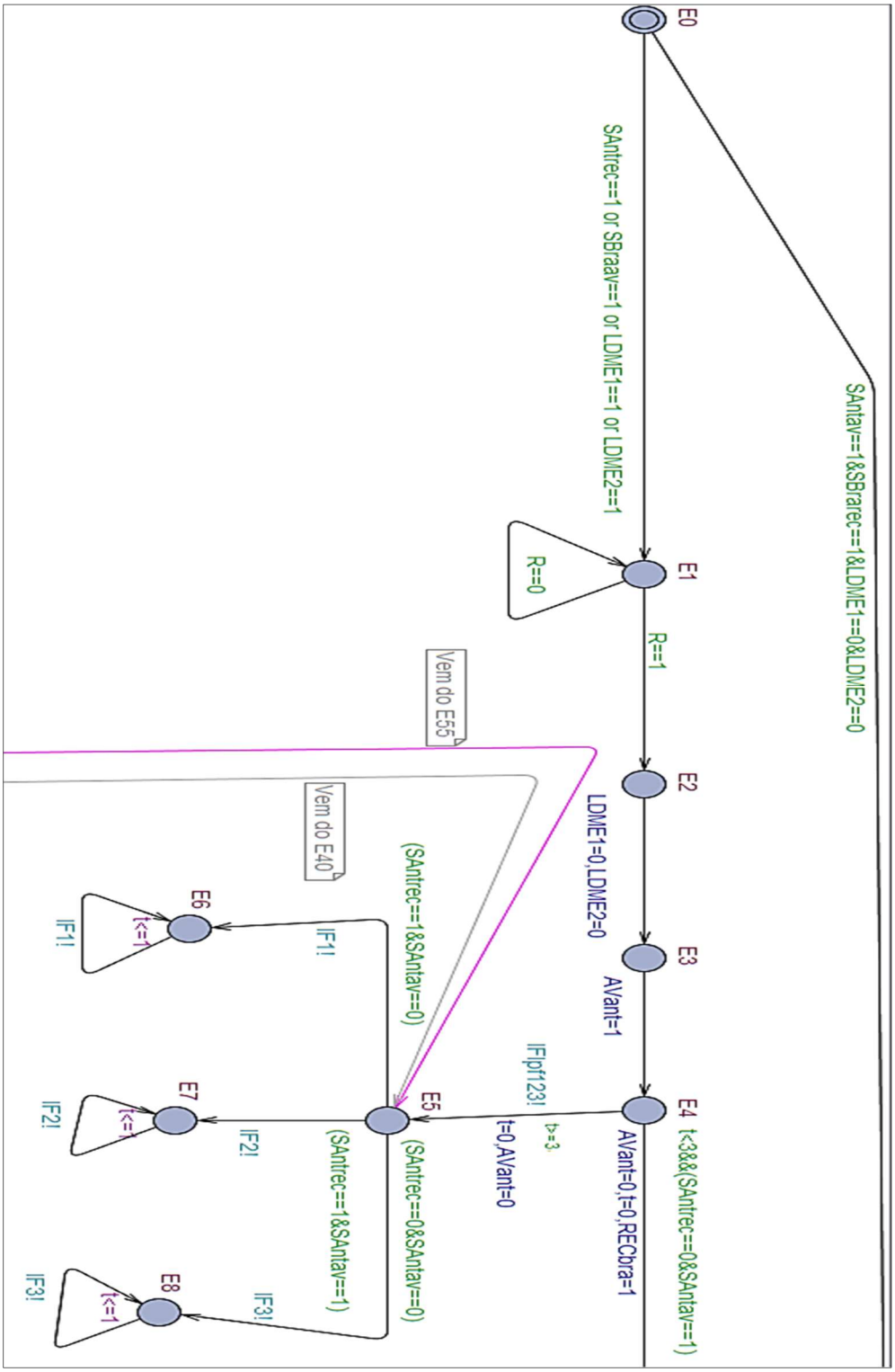

Figura 7.1.1 - Autômato G da estação de Separação (estado 0 ao 8).

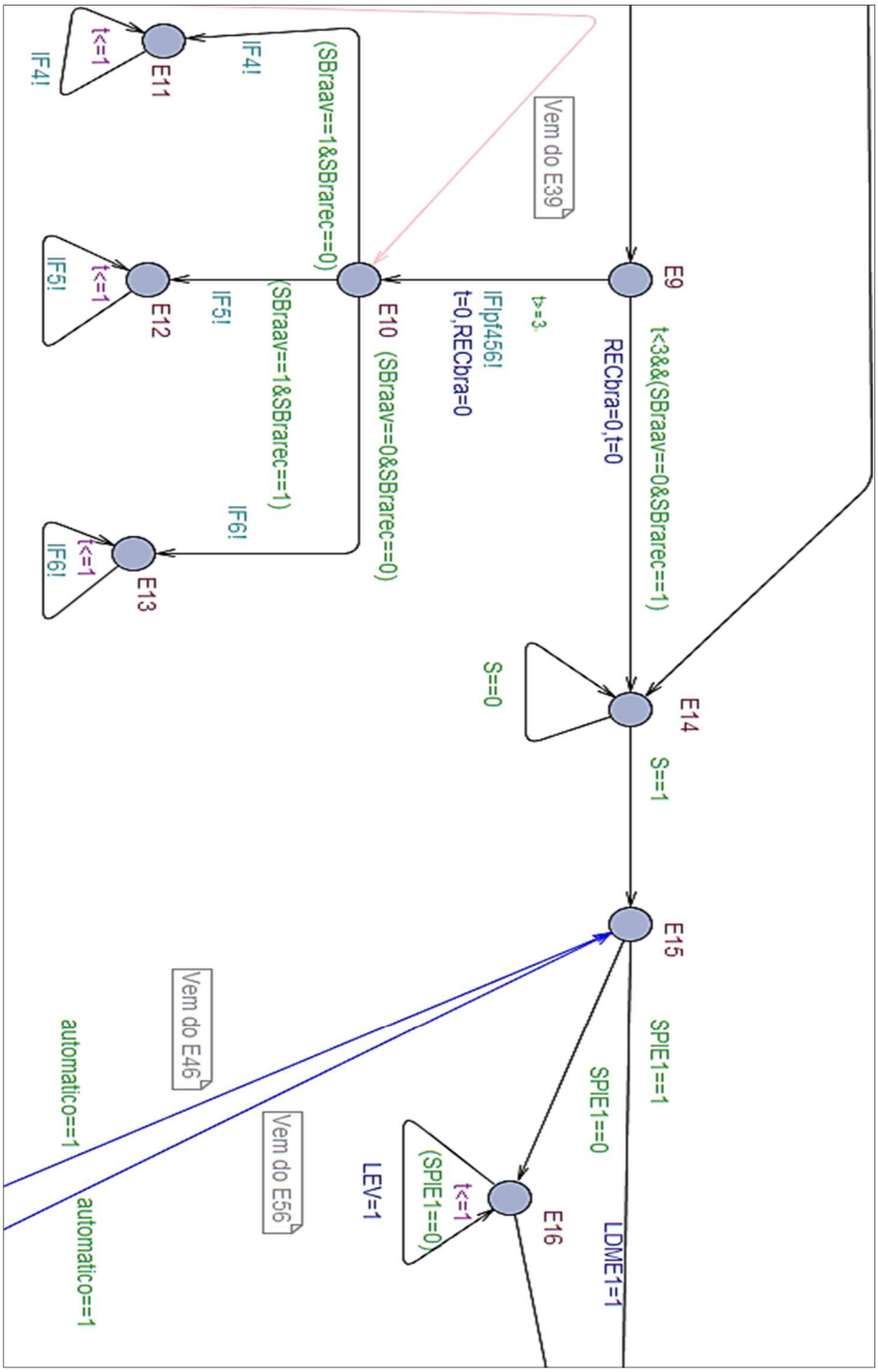

Figura 7.1.2 Autômato G da estação de Separação (estado 9 ao 16).

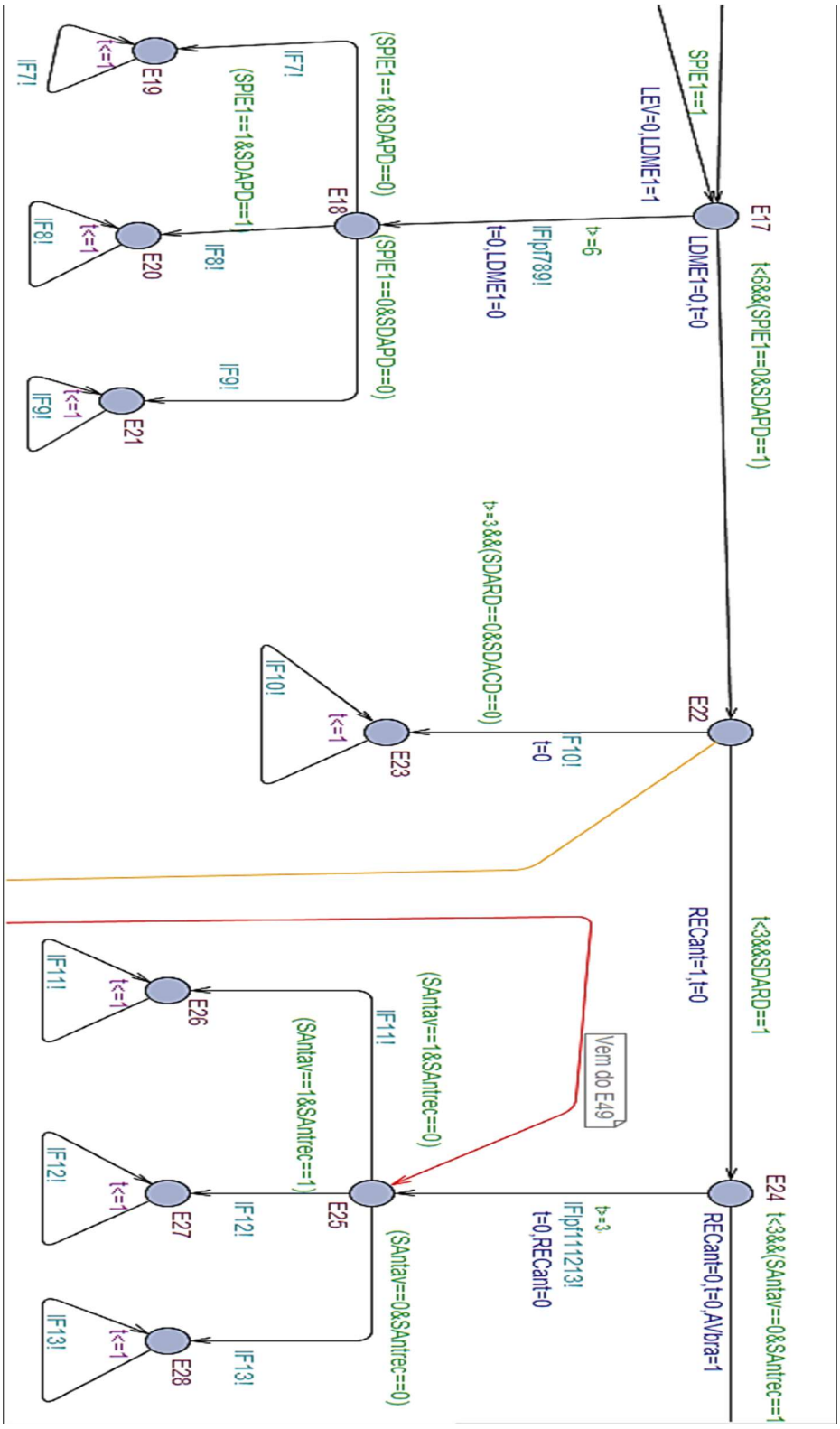

Figura 7.1.3 - Autômato G da estação de Separação (estado 17 ao 28).

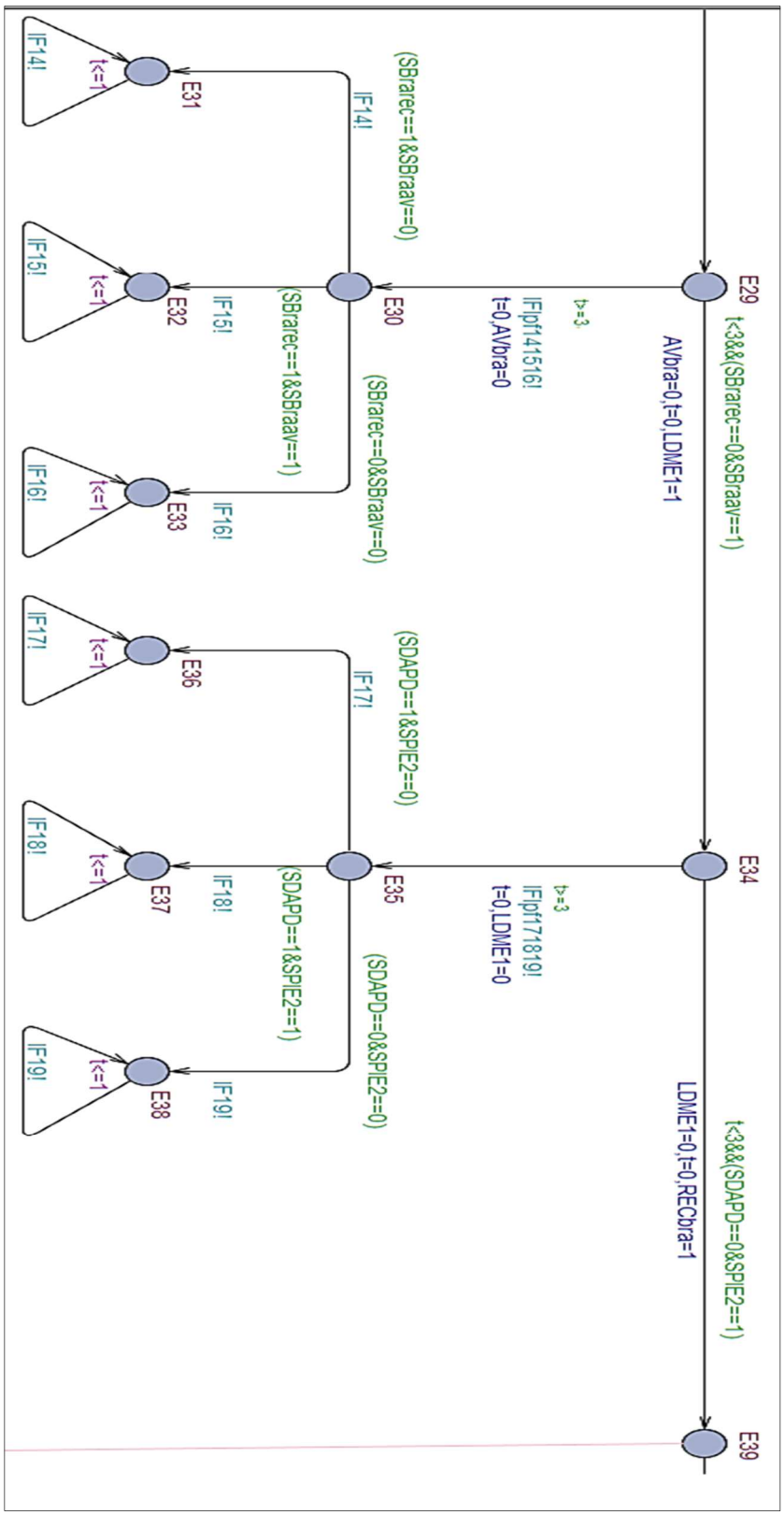

Figura 7.1.4 - Autômato G da estação de Separação (estado 29 ao 39).

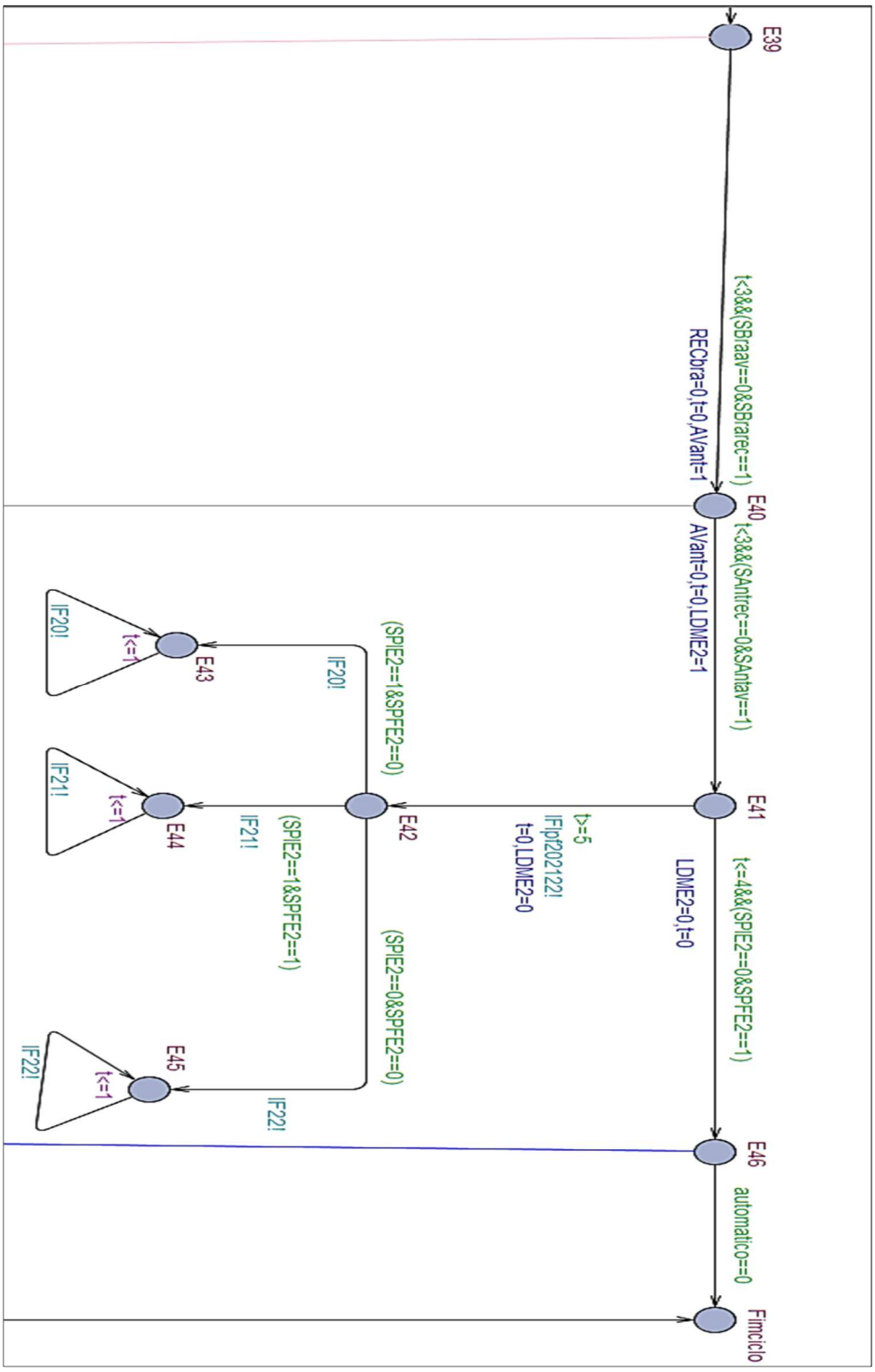

Figura 7.1.5 - Autômato G da estação de Separação (estado 39 ao 46/Fim ciclo).

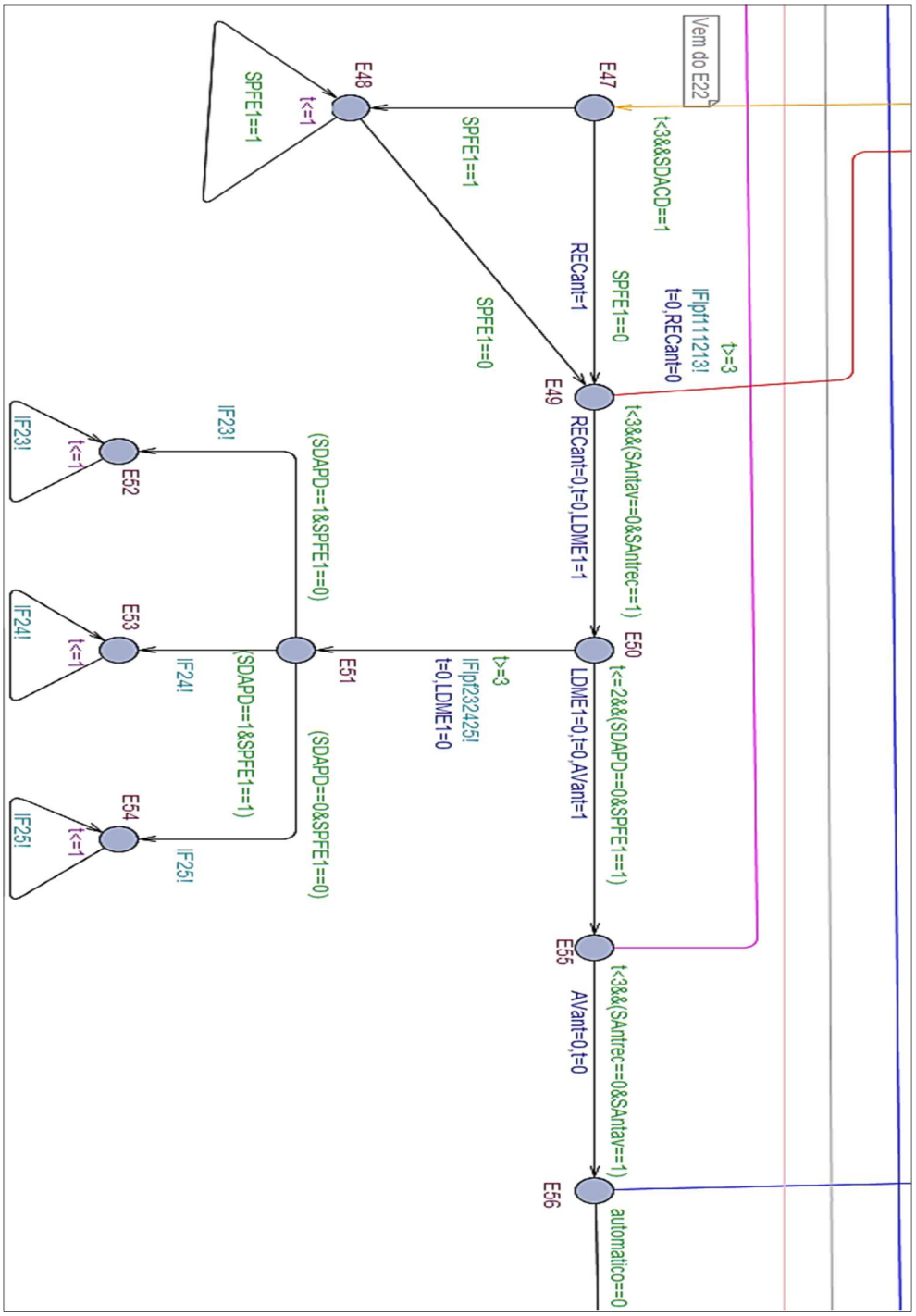

Figura 7.1.6 - Autômato G da estação de Separação (estado 47 ao 56).

Tabela 7.1- Variáveis e Faltas da Estação de Separação

Variável	Descrição da variável	Falta associada	Descrição da falta/ indicador	Expressão booleana de detecção
SAntav	Sensor Anteparo avançado	F3	Travado no estado desligado/ IF3!	Após Avant=1; (SAntrec==0&SAntav==0)
		F12	Travado no estado ligado/ IF12!	Após RECant=1; (SAntav==1&SAntrec==1)
SAntrec	Sensor Anteparo recuado	F2	Travado no estado ligado/ IF2!	Após AVant=1; (SAntrec==1&SAntav==1)
		F13	Travado no estado desligado/ IF13!	Após RECant=1; (SAntav==0&SAntrec==0)
LDME1	Comando Ligar/desligar motor da esteira 1	F7/F17/ F23	Motor da esteira travado no estado desligado/ IF7!/IF17!/IF23!	Após LDME1=1; (SPIE1==1&SDAPD==0) (SDAPD==1&SPIE2==0) (SDAPD==1&SPFE1==0)
LDME2	Comando Ligar/desligar motor da esteira 2	F20	Motor da esteira travado no estado desligado/ IF20!	Após LDM2=1; (SPIE2==1&SPFE2==0)
AVant	Comando Avançar anteparo	F11	Anteparo travado na posição avançada/ IF11!	Após RECant=1; (SAntav==1&SAntrec==0)
SBraav	Sensor Braço avançado	F5	Travado no estado ligado/ IF5!	Após RECbra=1; (SBraav==1&SBrarec==1)
		F16	Travado no estado desligado/IF16!	Após AVbra=1; (SBrarec==0&SBraav==0)
SBrarec	Sensor Braço recuado	F6	Travado no estado desligado/ IF6!	Após RECbra=1; (SBraav==0&SBrarec==0)
		F15	Travado no estado ligado/ IF15!	Após AVbra=1;(SBrarec==1&SBraav==1)
SDACD	Sensor difuso do anteparo cilindro detectado	F10	Travado no estado desligado/ IF10!	Após LDME1=1; t==3&&(SDARD==0&SDACD==0)
SDARD	Sensor difuso do anteparo relógio detectado	F10	Travado no estado desligado/ IF10!	Após LDME1=1; t==3&&(SDARD==0&SDACD==0)
SDAPD	Sensor difuso do anteparo peça detectada	F9	Travado no estado desligado/ IF9!	Após LDME1=1; (SPIE1==0&SDAPD==0)
		F18/F2	Travado no estado	Após LDME1=1;

		4	ligado/ IF18!/IF24!	(SDAPD==1&SPIE2==1) (SDAPD==1&SPFE1==1)
SPIE1	Sensor peça início da esteira 1	F8	Travado no estado ligado/ IF8!	Após LDME1=1; (SPIE1==1&SDAPD==1)
SPIE2	Sensor peça início da esteira 2	F19	Travado no estado desligado/ IF19!	Após LDME1=1; (SDAPD==0&SPIE2==0)
		F21	Travado no estado ligado/ IF21!	Após LDME2=1; (SPIE2==1&SPFE2==1)
SPFE2	Sensor peça fim da esteira 2	F22	Travado no estado desligado/ IF22!	Após LDME2=1; (SPIE2==0&SPFE2==0)
SPFE1	Sensor peça fim da esteira 1	F25	Travado no estado desligado/ IF25!	Após LDME1=1; (SDAPD==0&SPFE1==0)
RECant	Comando recuar anteparo	F1	Anteparo travado na posição recuada/ IF1!	Após Avant=1; (SAntrec==1&SAntav==0)
AVbra	Comando avançar braço	F4	Braço travado na posição avançada/ IF4!	Após RECbra=1; (SBraav==1&SBrarec==0)
RECbra	Comando recuar braço	F14	Braço travado na posição recuada/ IF14!	Após AVbra=1; (SBrarec==1&SBraav==0)
R	Botão reset	-	-	-
S	Botão start	-	-	-

7.2 Anexo 2: Estação de Coleta e encaixe

Este anexo exibe o autômato global que modela o comportamento da estação de Coleta e encaixe. Tanto o autômato global quanto todas as possíveis faltas permanentes detectáveis de sensores e atuadores estão exibidos nas Figuras 7.2.1, 7.2.2, 7.2.3, 7.2.4, 7.2.5 e 7.2.6. A tabela 7.2 exibe todas as variáveis da estação e suas faltas associadas.

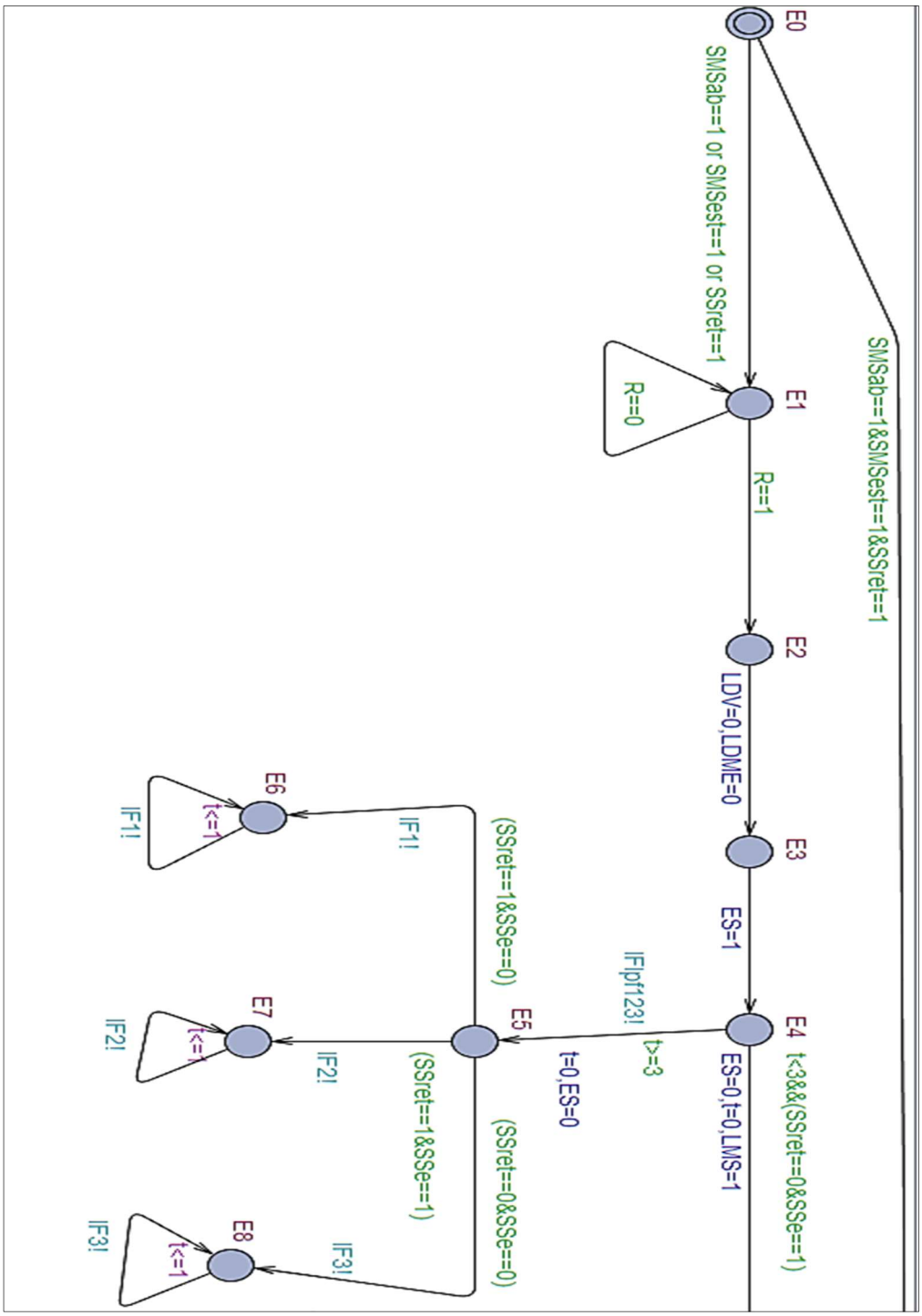

Figura 7.2.1 - Autômato G da estação de Coleta e Encaixe (estado 0 ao 8).

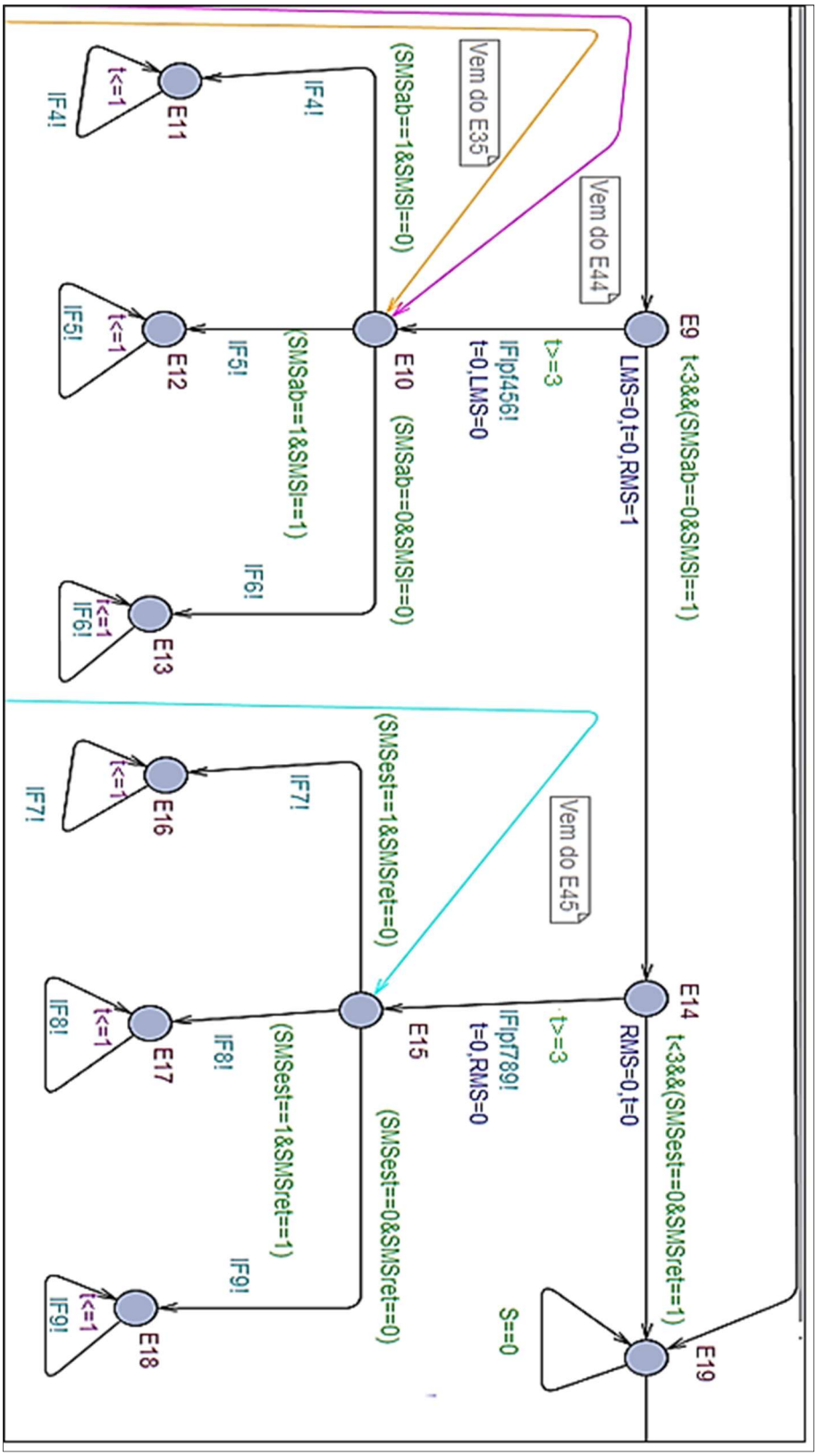

Figura 7.2.2 - Autômato G da estação de Coleta e Encaixe (estado 9 ao 19).

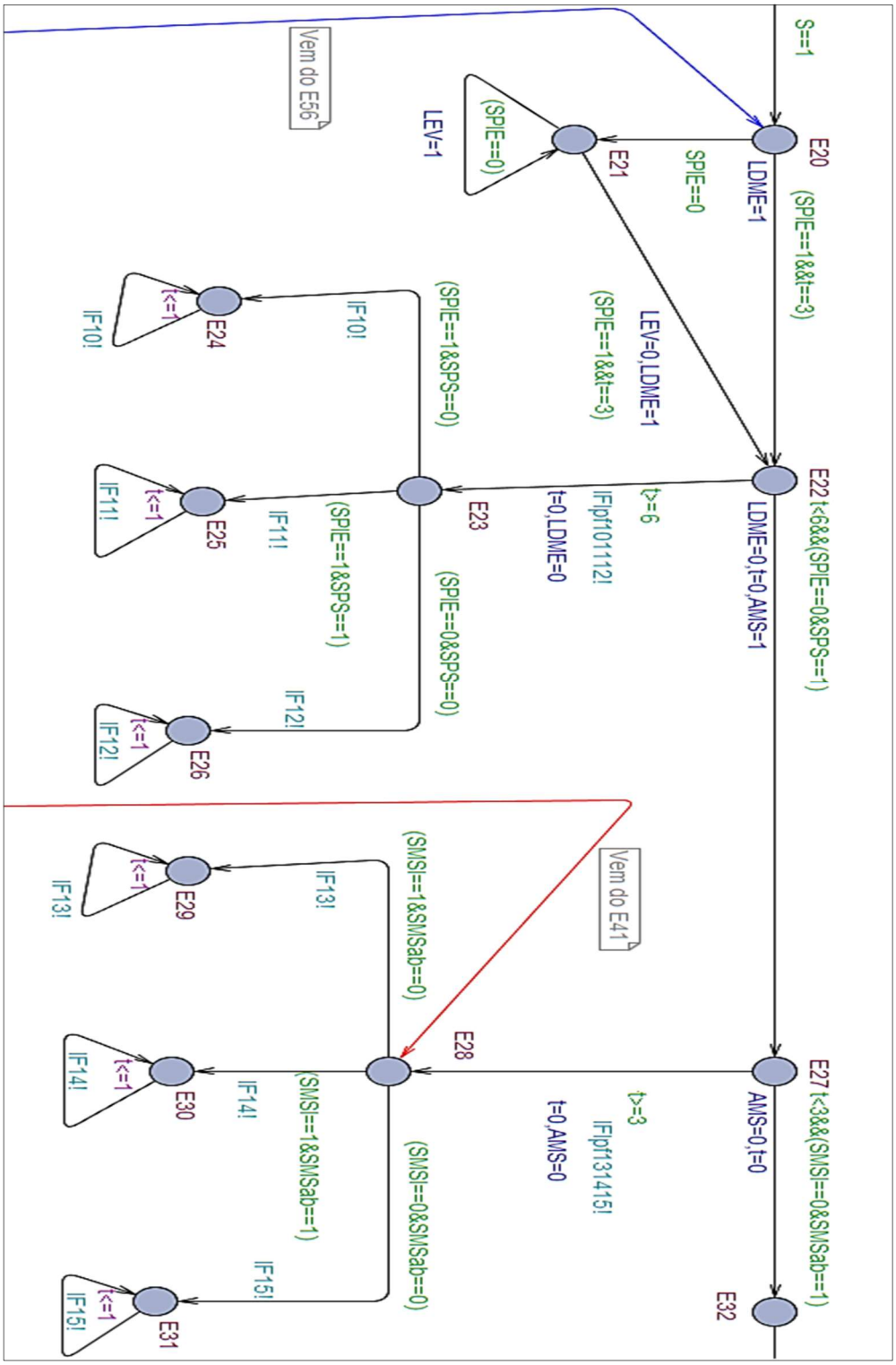

Figura 7.2.3 - Autômato G da estação de Coleta e Encaixe (estado 20 ao 32).

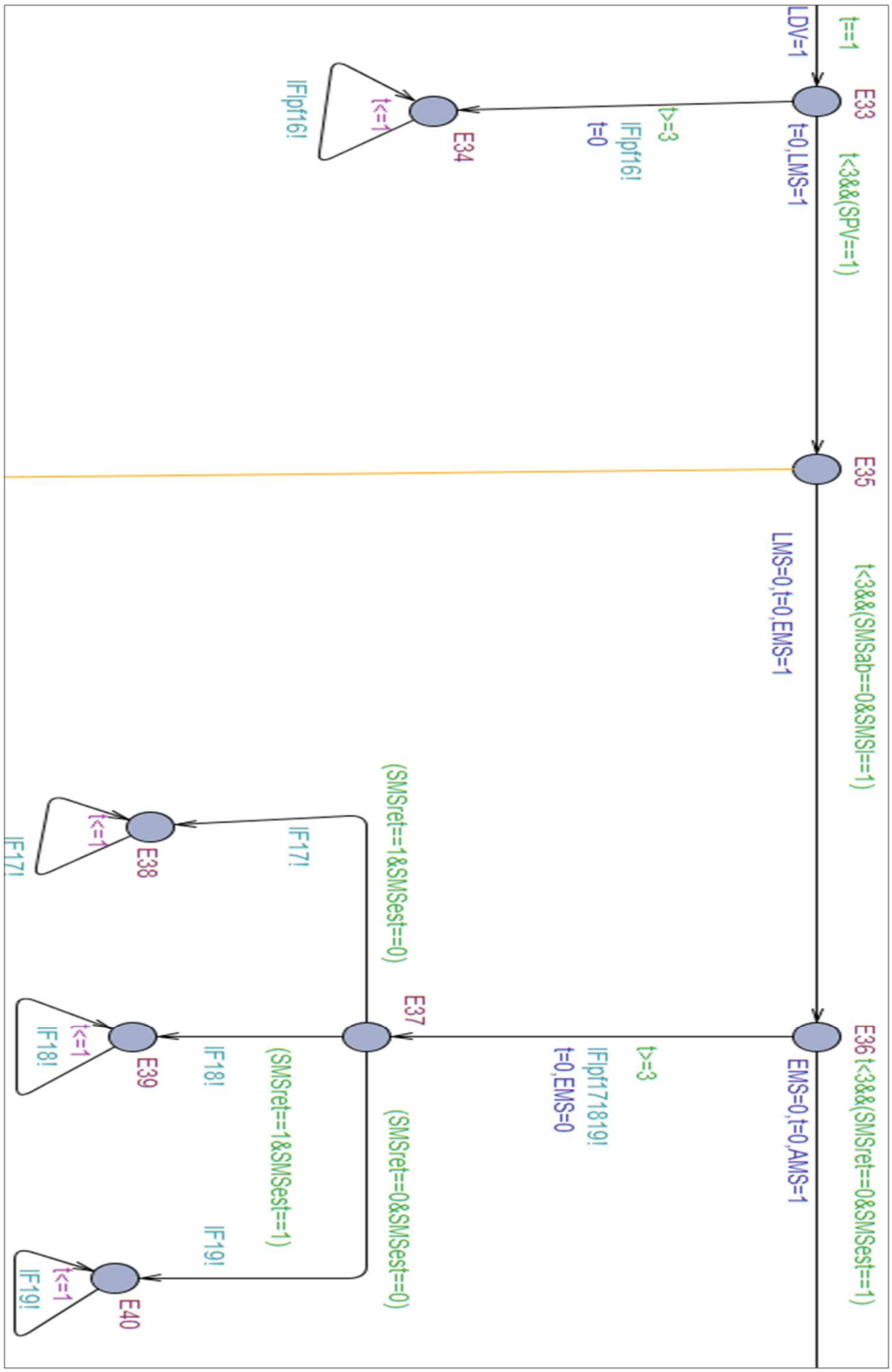

Figura 7.2.4 - Autômato G da estação de Coleta e Encaixe (estado 33 ao 40).

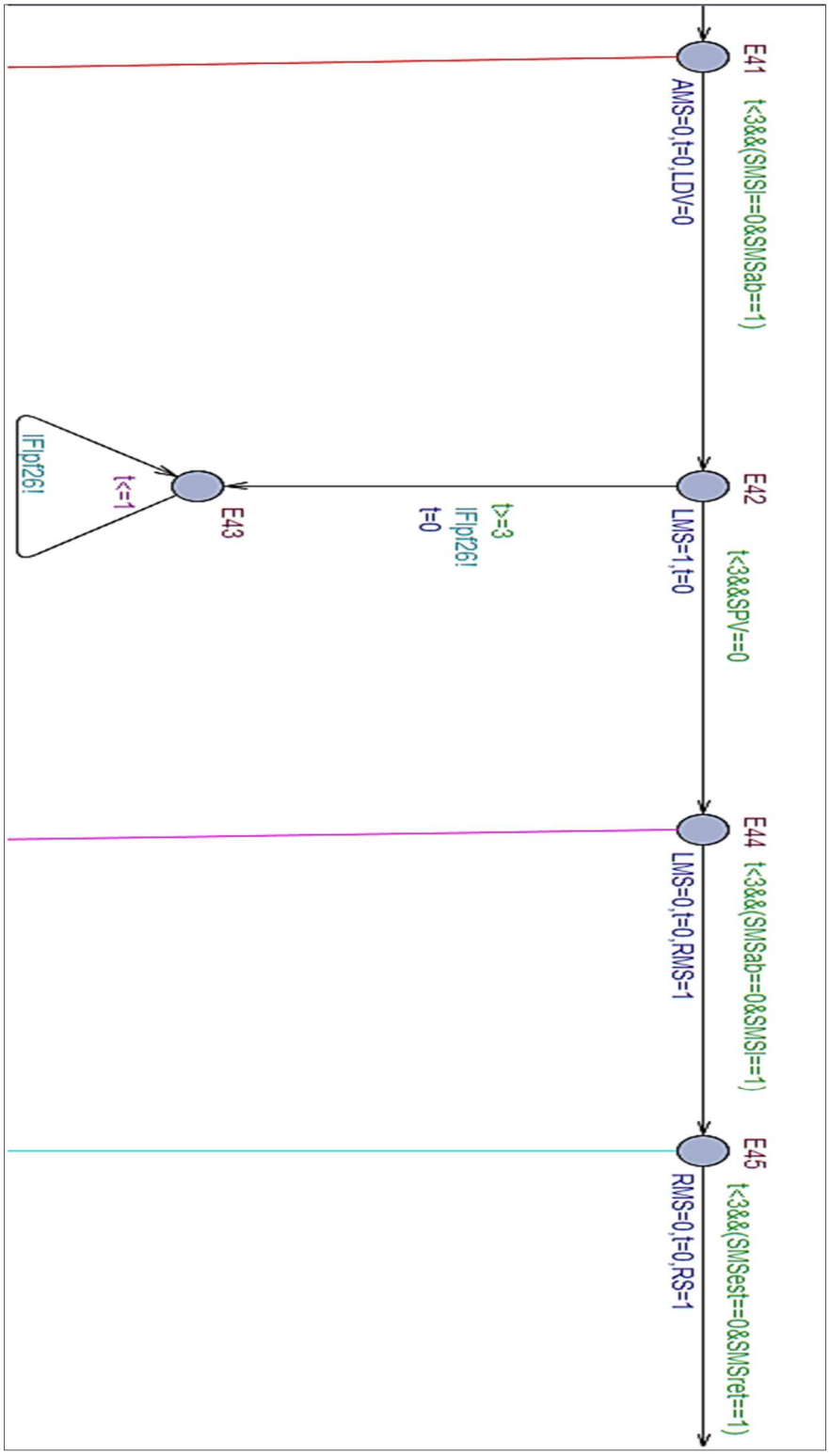

Figura 7.2.5 - Autômato G da estação de Coleta e Encaixe (estado 41 ao 45).

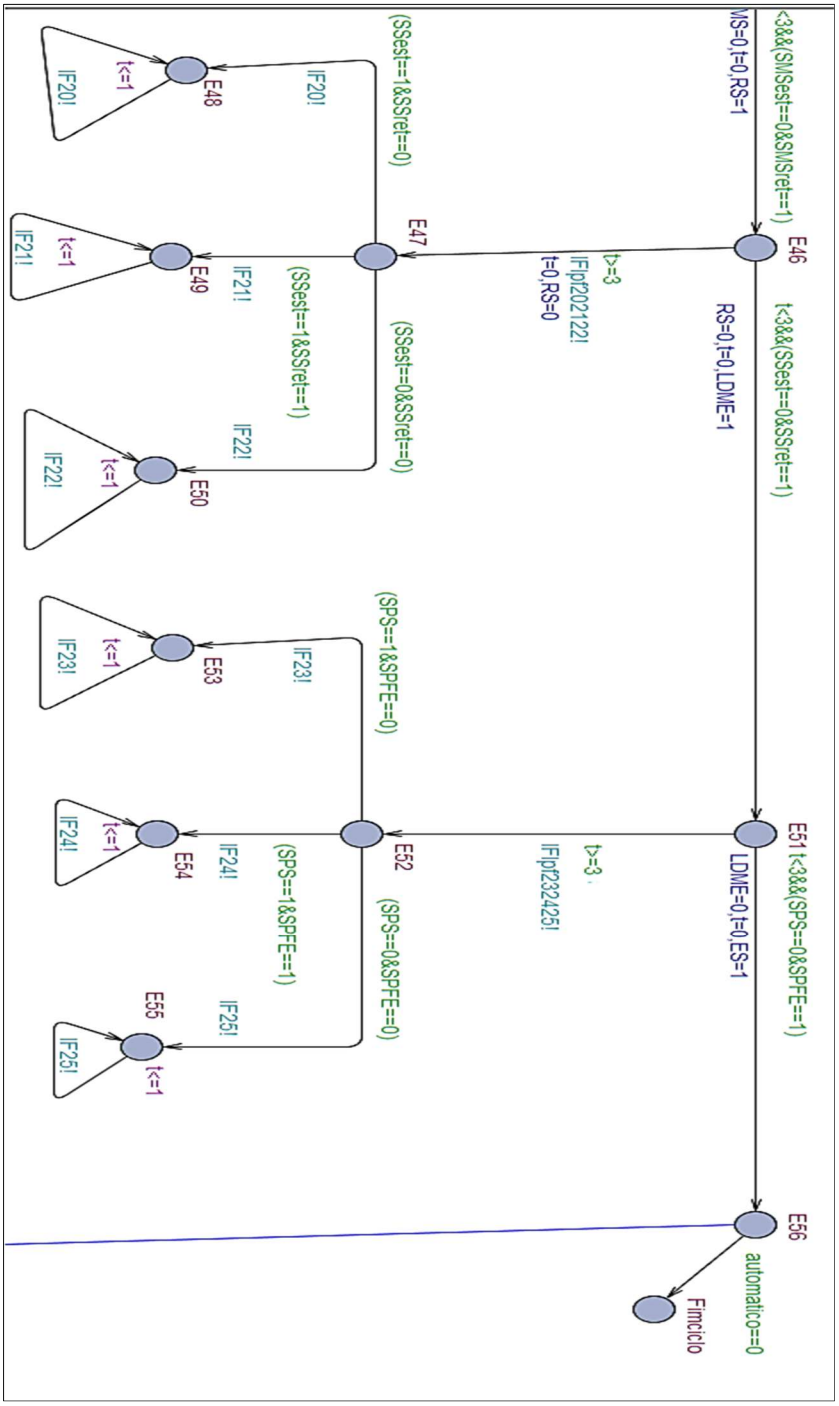

Figura 7.2.6 - Autômato G da estação de Coleta e Encaixe (estado 46 ao 56).

Tabela 7.2- Variáveis e faltas da estação de Coleta e encaixe

Variável	Descrição da variável	Falta associada	Descrição da falta/ indicador	Expressão booleana de detecção
SMSab	Sensor mine slide abaixado	F5	Travado no estado ligado/ IF5!	Após LMS=1; (SMSab==1&SMSl==1)
		F15	Travado no estado desligado/ IF15!	Após AMS=1; (SMSl==0&SMSab==0)
SMSL	Sensor mine slide levantado	F6	Travado no estado desligado/ IF6!	Após LMS=1; (SMSab==0&SMSl==0)
		F14	Travado no estado ligado/ IF14!	Após AMS=1; (SMSl==1&SMSab==1)
LDME	Comando Ligar/desligar motor da esteira	F10/F23	Motor da esteira travado no estado desligado/ IF10!/IF23!	Após LDME=1; (SPIE==1&SPS==0) (SPS==1&SPFE==0)
LMS	Comando Levantar mine slide	F13	Mine slide travado na posição levantada/ F13!	Após AMS=1; (SMSl==1&SMSab==0)
AMS	Comando abaixar mine slide	F4	Mine slide travado na posição	Após LMS=1; (SMSab==1&SMSl==0)
SMSret	Sensor mine slide retraído	F9	Travado no estado desligado/ IF9!	Após RMS=1; (SMSest==0&SMSret==0)
		F18	Travado no estado ligado/ IF18!	Após EMS=1; (SMSret==1&SMSest==1)
SMSest	Sensor mine slide estendido	F8	Travado no estado ligado/ IF8!	Após RMS=1; (SMSest==1&SMSret==1)
		F19	Travado no estado desligado/ IF19!	Após EMS=1; (SMSret==0&SMSest==0)
SSe	Sensor separador estendido	F3	Travado no estado desligado/ IF3!	Após RS=1; (SSret==0&SSe==0)
		F21	Travado no estado ligado/ IF21!	Após ES=1; (SSest==1&SSret==1)
LDV	Comando Ligar/Desligar vácuo	FI16	Falta no sensor de peça travado no estado desligado ou atuador de vácuo travado no estado desligado/ IFIpf16!	Após LDV=1; t==3&&SPV==0
		FI26	Falta no sensor de peça travado no estado ligado ou atuador de vácuo travado no estado	Após LDV=0; t==3&&SPV==1

			ligado/IFIpf26!	
SSret	Sensor separador retraído	F2	Travado no estado ligado/ IF2!	Após ES=1; (SSret==1&SSe==1)
		F22	Travado no estado desligado/ IF22!	Após RS=1; (SSest==0&SSret==0)
EMS	Estender mine slide	F7	Mine slide travado na posição estendida/ F7!	Após RMS=1; (SMSest==1&SMSret==0)
RMS	Retrair mine slide	F17	Mine slide travado na posição retraída/ F17!	Após EMS=1; (SMSret==1&SMSest==0)
ES	Estender separador	F20	Separador travado na posição estendida/ IF20!	Após RS=1; (SSest==1&SSret==0)
RS	Retrair separador	F1	Separador travado na posição retraída/ IF1!	Após ES=1; (SSret==1&SSe==0)
SPIE	Sensor peça início da esteira	F11	Travado no estado ligado/ IF11!	Após LDME=1; (SPIE==1&SPS==1)
SPS	Sensor peça no separador	F12	Travado no estado desligado/ IF12!	Após LDME=1; (SPIE==0&SPS==0)
		F24	Travado no estado ligado/ IF24!	Após LDME=1; (SPS==1&SPFE==1)
SPFE	Sensor peça no final da esteira	F25	Travado no estado desligado/ IF25!	Após LDME=1; (SPS==0&SPFE==0)
R	Botão reset	-	-	-
S	Botão start	-	-	-

7.3 Anexo 3: Estação Músculo fluido

Este anexo exibe o autômato global que modela o comportamento da estação de Músculo fluido. Tanto o autômato global quanto todas as possíveis faltas permanentes detectáveis de sensores e atuadores estão exibidos nas Figuras 7.3.1, 7.3.2, 7.3.3, 7.3.4 e 7.3.5. A tabela 7.3 exibe todas as variáveis da estação e suas faltas associadas.

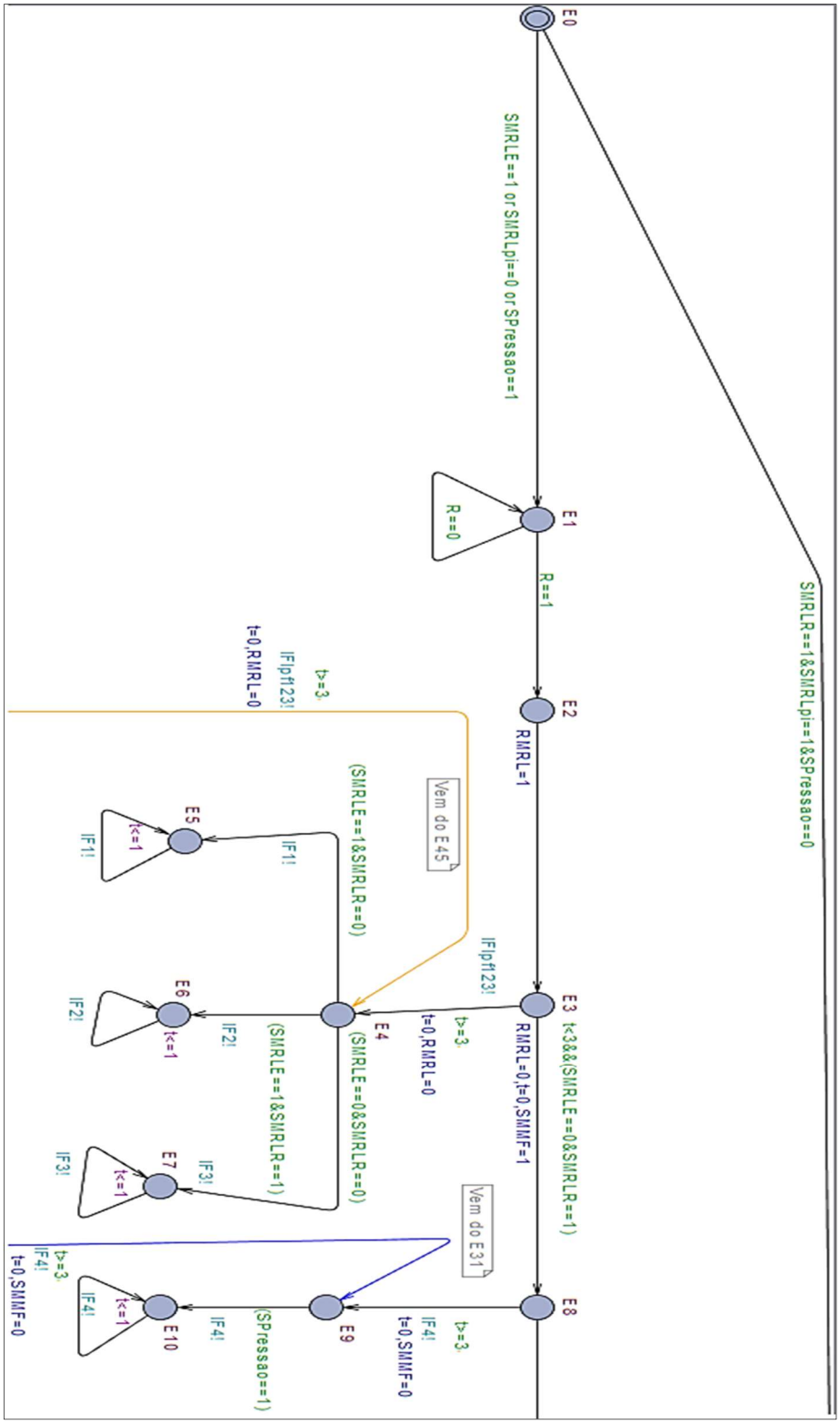

Figura 7.3.1 - Autômato G da estação de Músculo fluido (estado 0 ao 10).

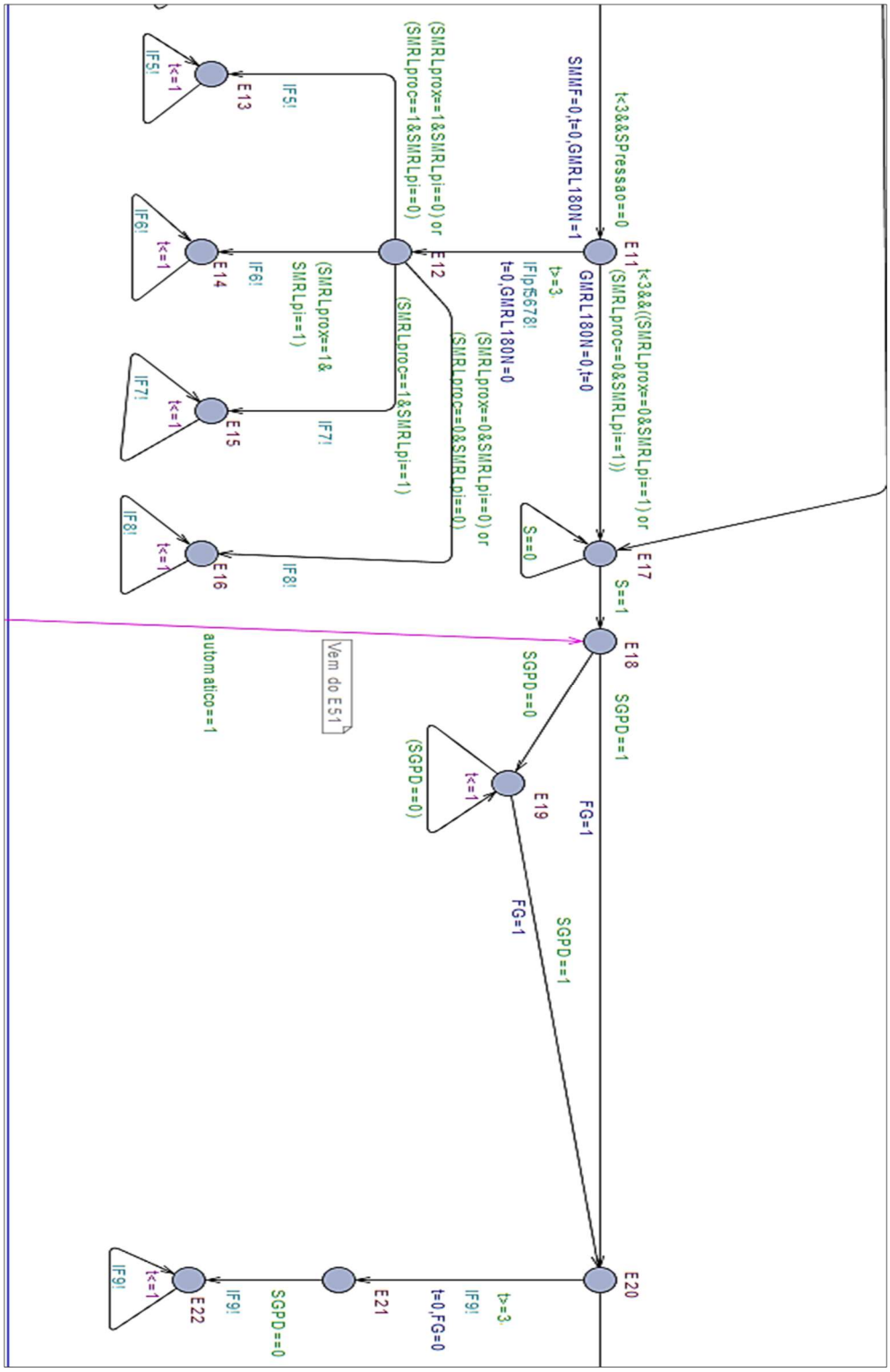

Figura 7.3.2 - Autômato G da estação de Músculo Fluido (estado 11 ao 22).

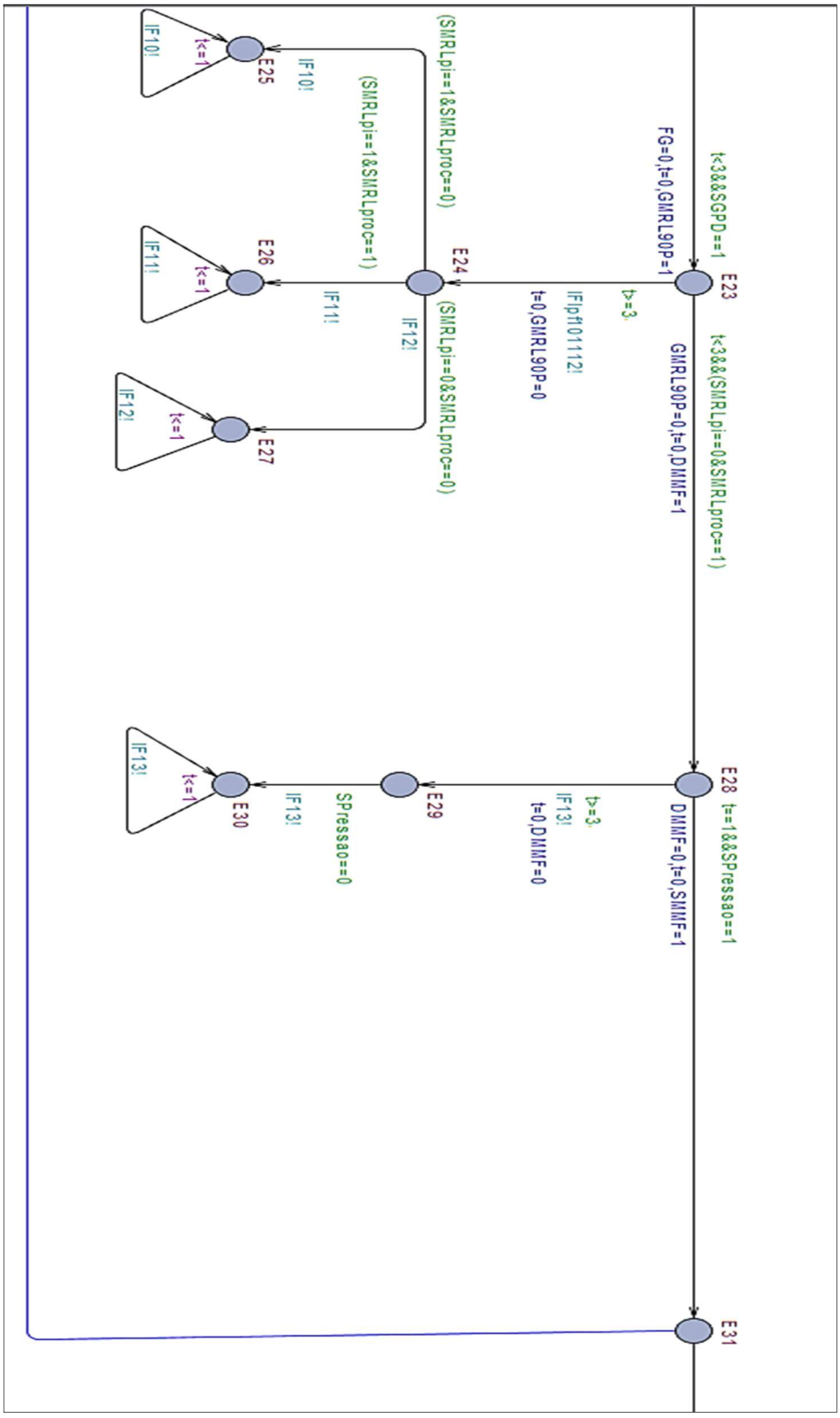

Figura 7.3.3 - Autômato G da estação de Músculo Fluido (estado 23 ao 31).

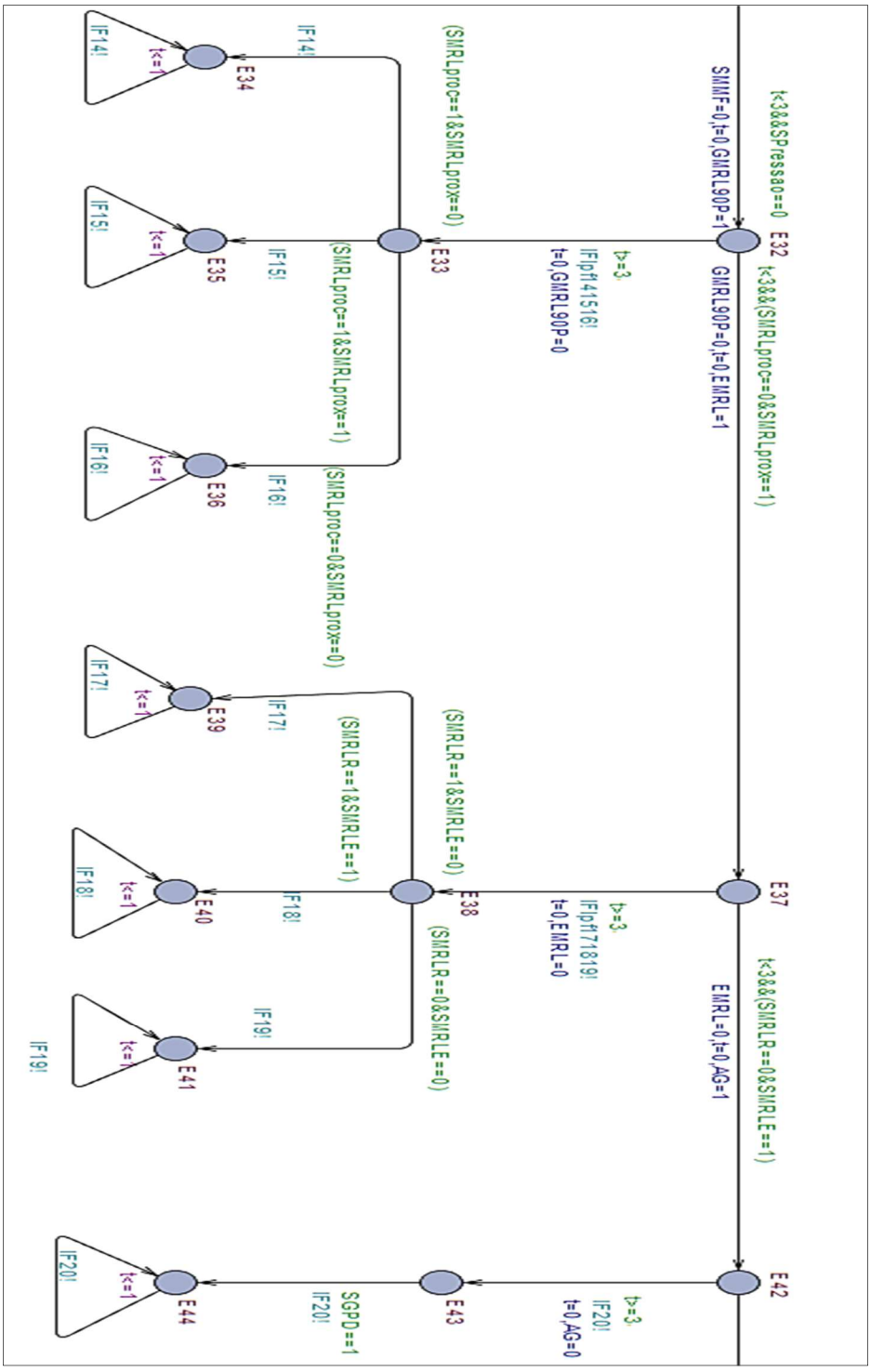

Figura 7.3.4 - Autômato G da estação de Músculo fluido (estado 32 ao 44).

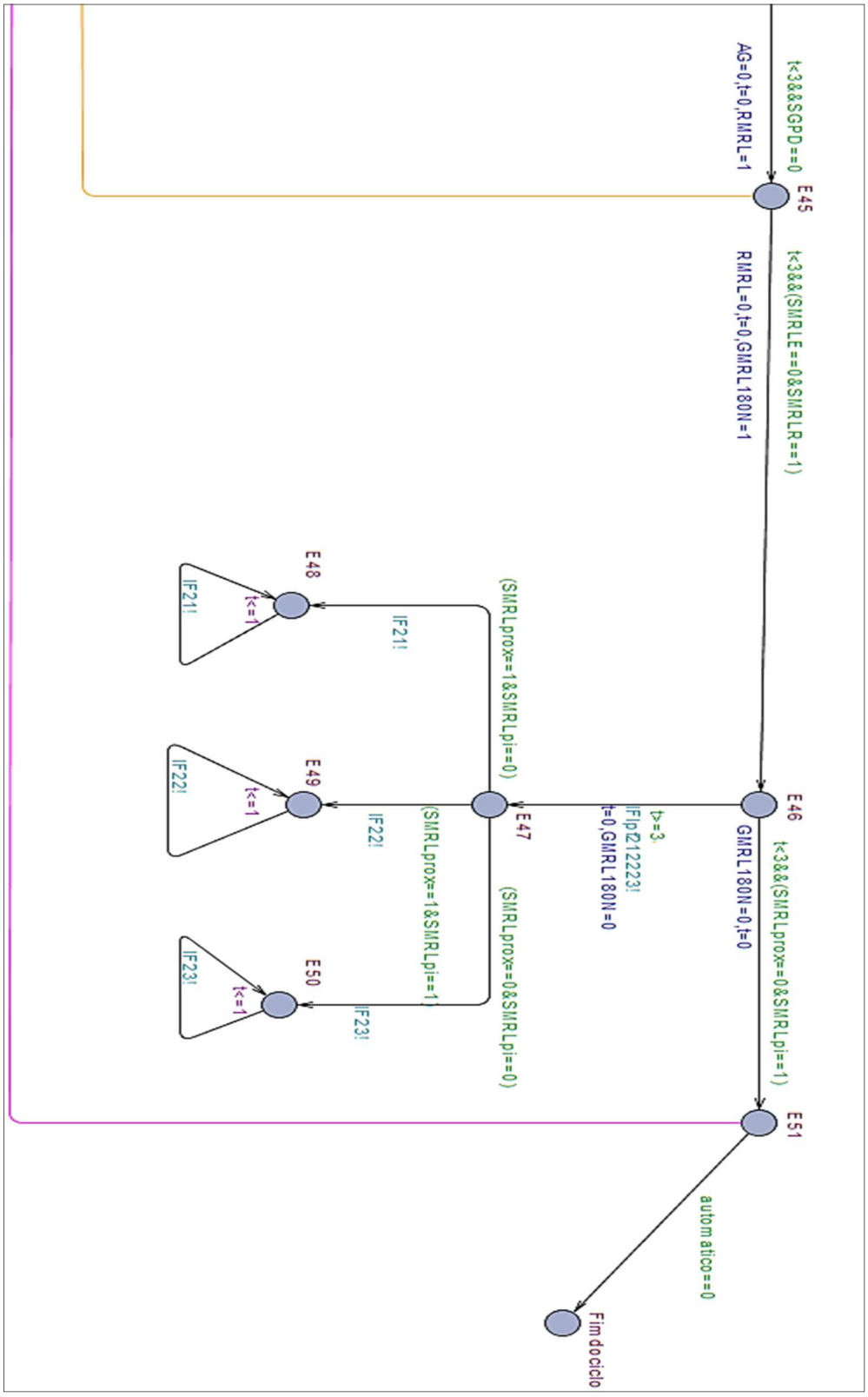

Figura 7.3.5 - Autômato G da estação de Músculo fluido (estado 45 ao Fim do ciclo).

Tabela 7.3- Variáveis e Faltas da Estação Músculo Fluido

Variável	Descrição da variável	Falta associada	Descrição da falta/ indicador	Expressão booleana de detecção
SGPD	Sensor Garra peça detectada	FI9	Falta sensor garra peça detectada (travado no estado desligado) ou atuador da garra travado no estado aberto/ IFIpf9!	Após FG=1; SGPD==0
		FI20	Falta sensor garra peça detectada (travado no estado ligado) ou atuador da garra travado no estado fechado/ IFIpf20!	Após AG=1; SGPD==1
SMRLproc	Sensor módulo rotativo linear posição processamento	F7/F15	Travado no estado ligado/ IF7/IF15!!	Após GMRL180N=1; (SMRLproc==1&SMRLpi==1) (SMRLproc==1&SMRLprox==1)
		F12	Travado no estado desligado/ IF12!	Após GMRL90P=1; (SMRLpi==0&SMRLproc==0)
SMRLprox	Sensor módulo rotativo linear posição próxima estação	F6/F22	Travado no estado ligado/ IF6!/IF22!	GMRL180N=1; (SMRLprox==1&SMRLpi==1) (SMRLprox==1&SMRLpi==1)
		F16	Travado no estado desligado/ IF16!	GMRL90P=1; (SMRLproc==0&SMRLprox==0)
SPressao	Sensor de Pressão	FI4	Falta no sensor de pressão (travado no estado ligado) ou no módulo músculo fluido/IFIpf4!	Após SMMF=1; (SPressao==1)
		FI13	Falta no sensor de pressão (travado no estado desligado) ou no módulo músculo fluido/IFIpf13!	Após DMMF=1; SPressao==0
SMRLpi	Sensor módulo rotativo linear posição inicial	F8/F23	Travado no estado desligado/ IF8!/ IF23!	GMRL180N=1; (SMRLprox==0&SMRLpi==0) or (SMRLproc==0&SMRLpi==0);

				(SMRLprox==0&SMRLpi==0)
		F11	Travado no estado ligado/ IF11!	GMRL90P=1; (SMRLpi==1&SMRLproc==1)
SMRLE	Sensor módulo rotativo linear estendido	F2	Travado no estado ligado/ IF2!	RMRL=1; (SMRLE==1&SMRLR==1)
		F19	Travado no estado desligado/ IF19!	EMRL=1; (SMRLR==0&SMRLE==0)
SMRLR	Sensor módulo rotativo linear retraído	F18	Travado no estado ligado/ IF18!	EMRL=1; (SMRLR==1&SMRLE==1)
		F3	Travado no estado desligado/ IF3!	RMRL=1; (SMRLE==0&SMRLR==0)
GMRL90P	Girar módulo rotativo linear 90° positivo	F10/F14	Falta no módulo rotativo linear travado na posição inicial/ Falta no módulo rotativo linear travado na posição de processamento	GMRL90P=1; (SMRLpi==1&SMRLproc==0) (SMRLproc==1&SMRLprox==0)
GMRL180N	Girar módulo rotativo linear 180° negativo	F5/F21	Falta no módulo rotativo linear travado na posição de processamento ou próxima estação/IF5!/ IF21!	GMRL180N=1; (SMRLprox==1&SMRLpi==0) or (SMRLproc==1&SMRLpi==0); (SMRLprox==1&SMRLpi==0)
EMRL	Estender módulo rotativo Linear	F1	Módulo rotativo linear travado na posição avançada/IF1!	RMRL=1; (SMRLE==1&SMRLR==0)
RMRL	Recuar módulo rotativo linear	F17	Módulo rotativo linear travado na posição recuada/IF17!	EMRL=1; (SMRLR==1&SMRLE==0)
DMMF	Descer módulo musculo fluido e prensar peça	FI4	Falta no sensor de pressão (travado no estado ligado) ou no módulo músculo fluido/IFIpf4!	SMMF=1; (SPressao==1)
SMMF	Subir módulo musculo fluido	FI13	Falta no sensor de pressão (travado no estado desligado) ou no módulo	Após DMMF=1; SPressao==0

			músculo fluido/IFIpf13!	
FG	Fechar garra	FI9	Falta sensor garra peça detectada (travado no estado desligado) ou atuador da garra travado no estado aberto/ IFIpf9!	Após FG=1; SGPD==0
AG	Abrir garra	FI20	Falta sensor garra peça detectada (travado no estado ligado) ou atuador da garra travado no estado fechado/ IFIpf20!	Após AG=1; SGPD==1
R	Botão reset	-	-	-
S	Botão start	-	-	-

7.4 Anexo 4: Estação de Classificação

Este anexo exibe o autômato global que modela o comportamento da estação de Classificação. Tanto o autômato global quanto todas as possíveis faltas permanentes detectáveis de sensores e atuadores estão exibidos nas Figuras 7.4.1, 7.4.2, 7.4.3, 7.4.4, 7.4.5 e 7.4.6. A tabela 7.4 exibe todas as variáveis da estação e suas faltas associadas.

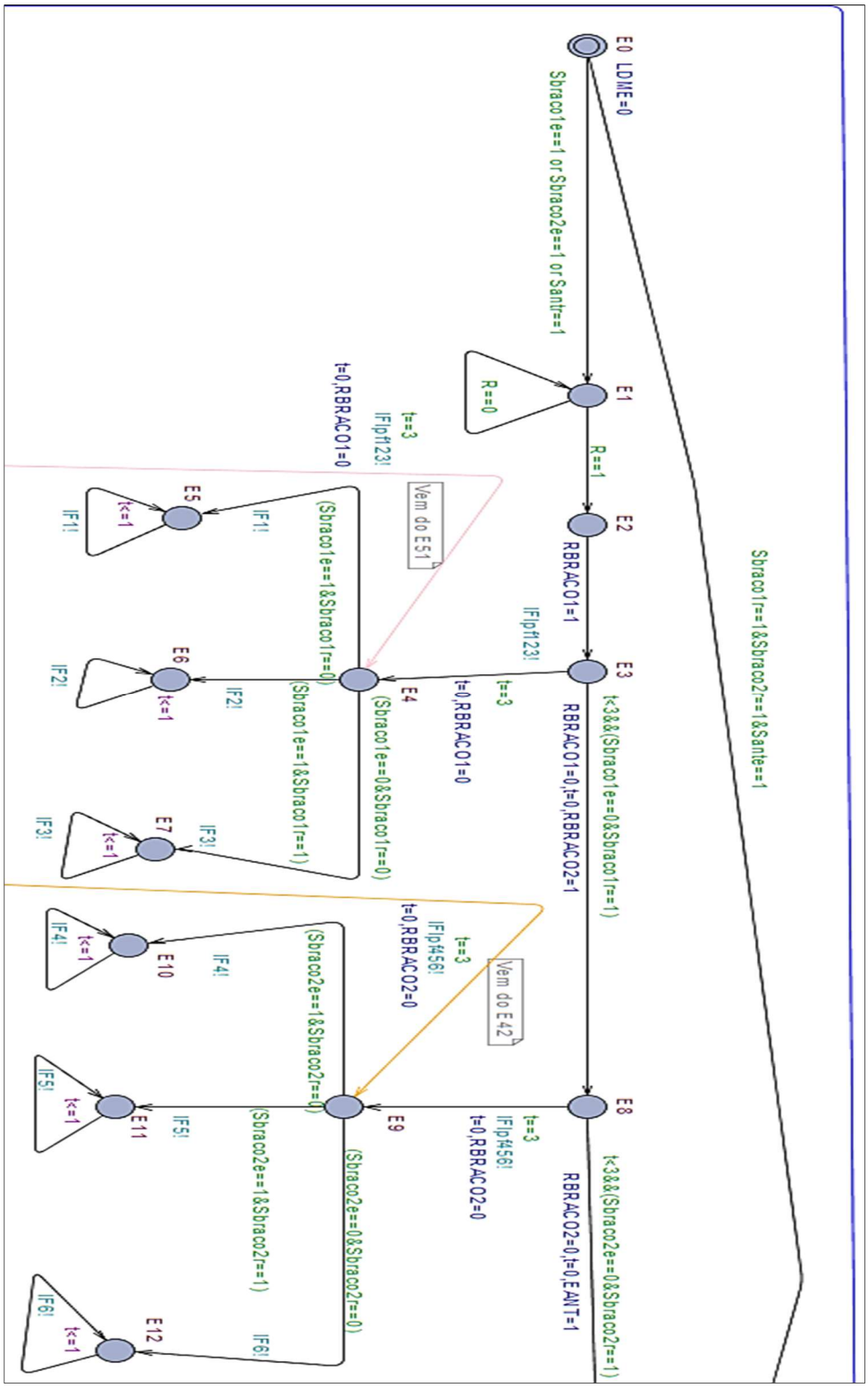

Figura 7.4.1 - Autômato G da estação de Classificação (estado 0 ao 12).

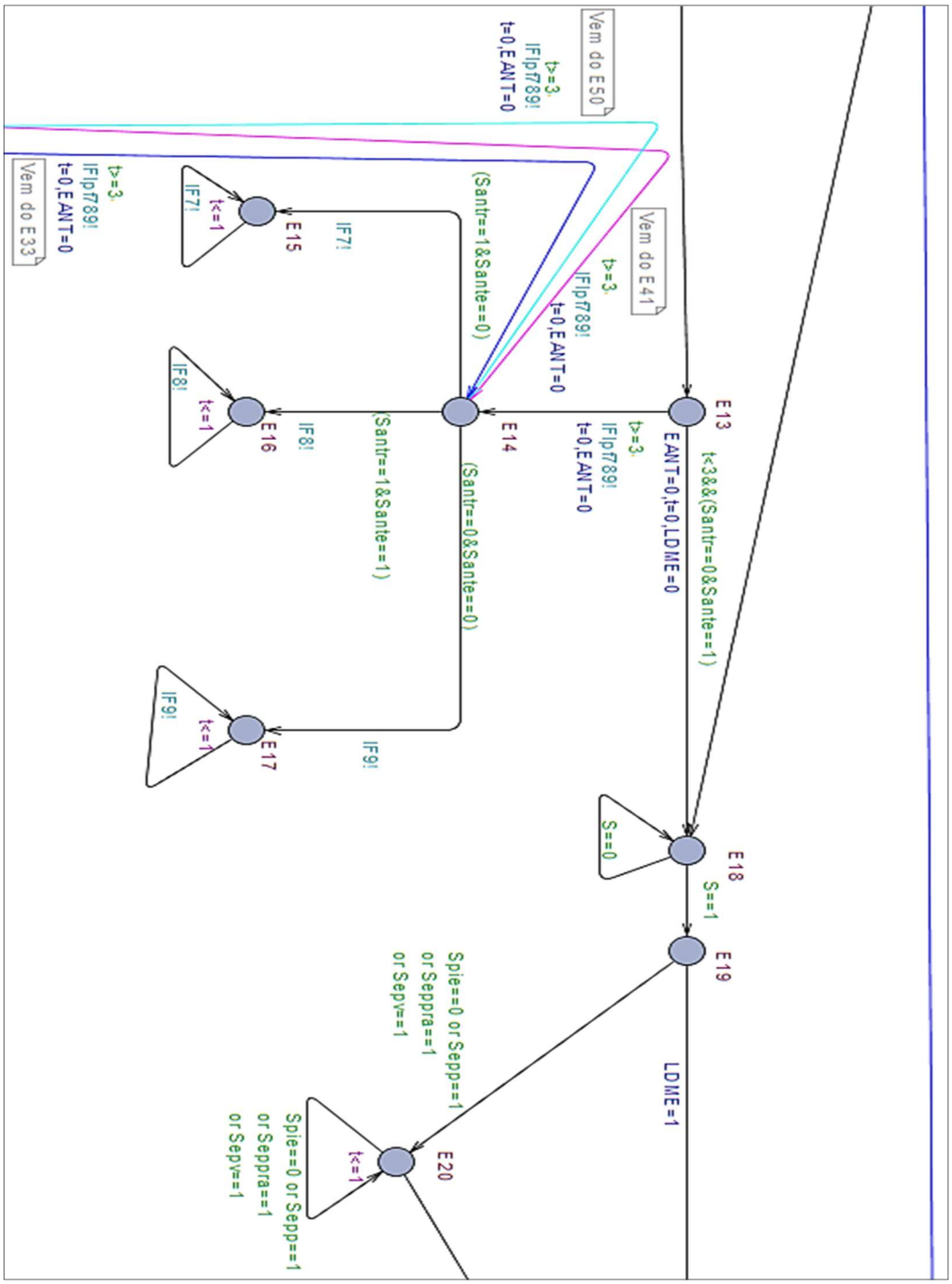

Figura 7.4.2 - Autômato G da estação de Classificação (estado 13 ao 20).

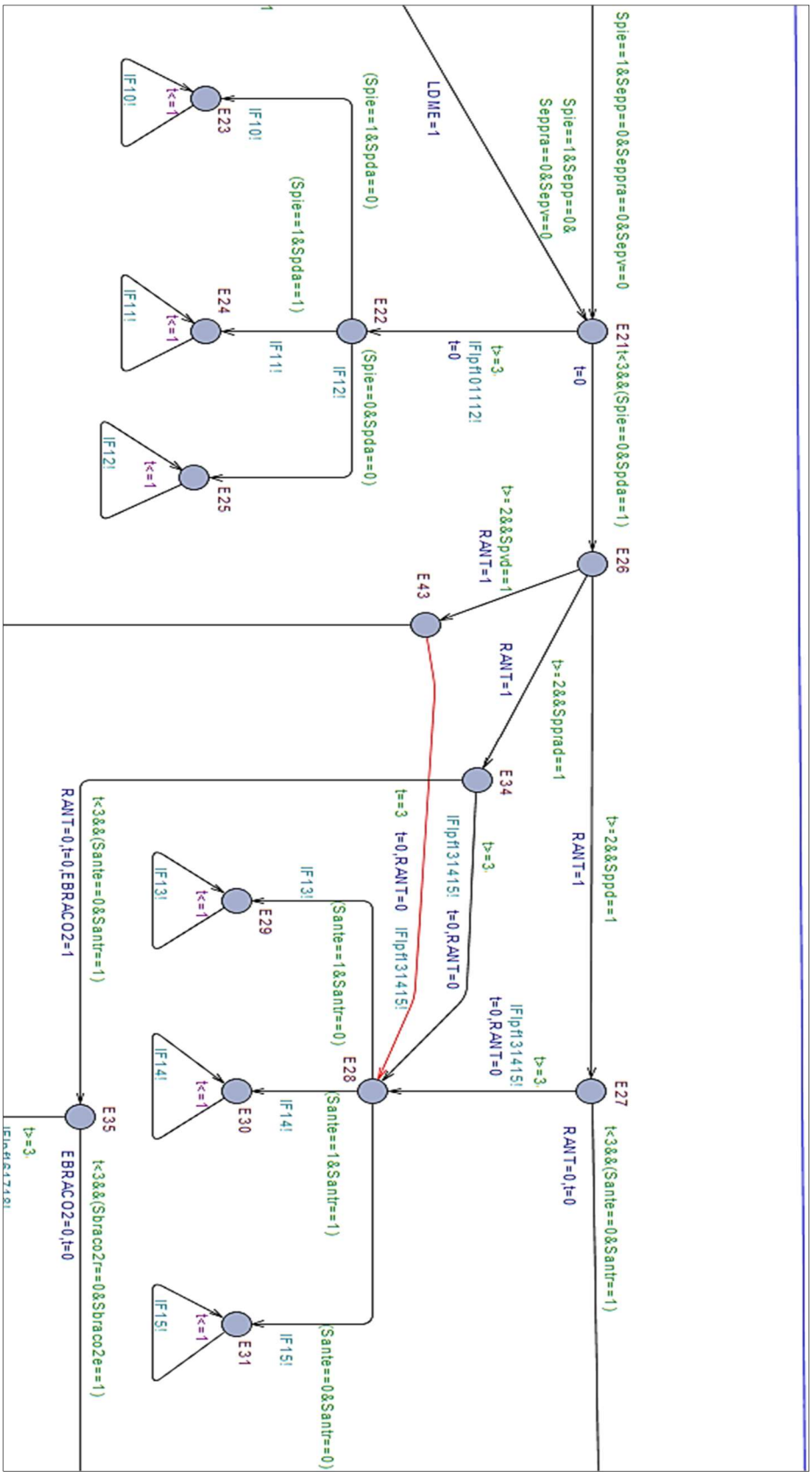

Figura 7.4.3 - Autômato G da estação de Classificação (estado 21 ao 31).

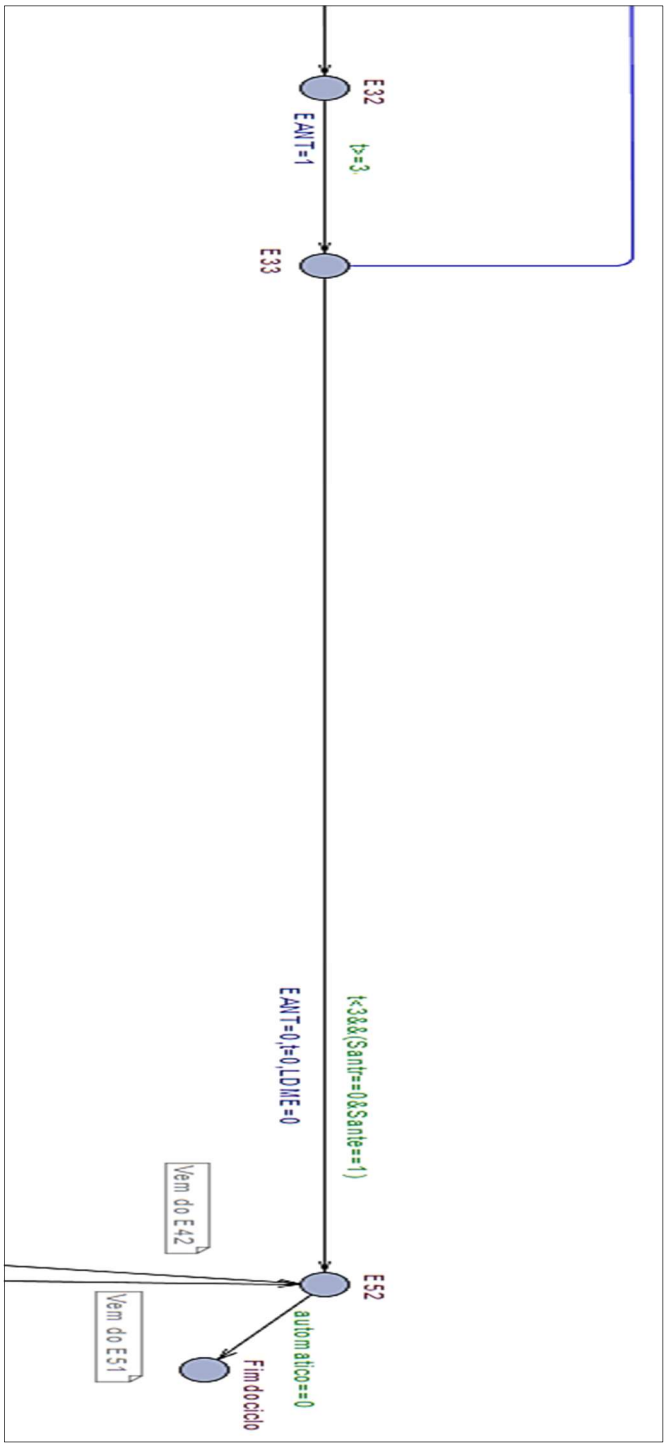

Figura 7.4.4 - Autômato G da estação de Classificação (estado 32 ao Fim do ciclo).

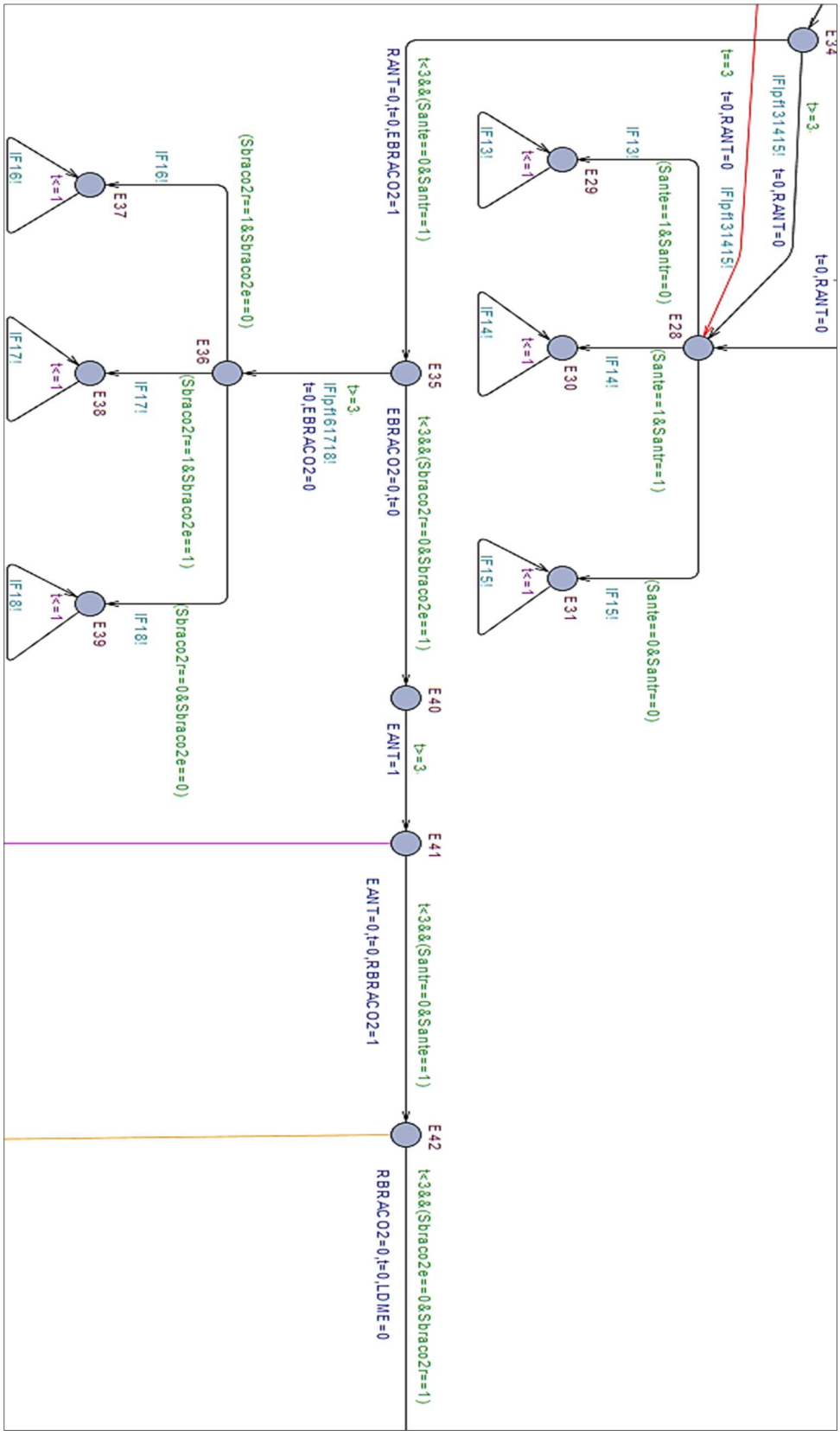

Figura 7.4.5 - Autômato G da estação de Classificação (estado 34 ao 42).

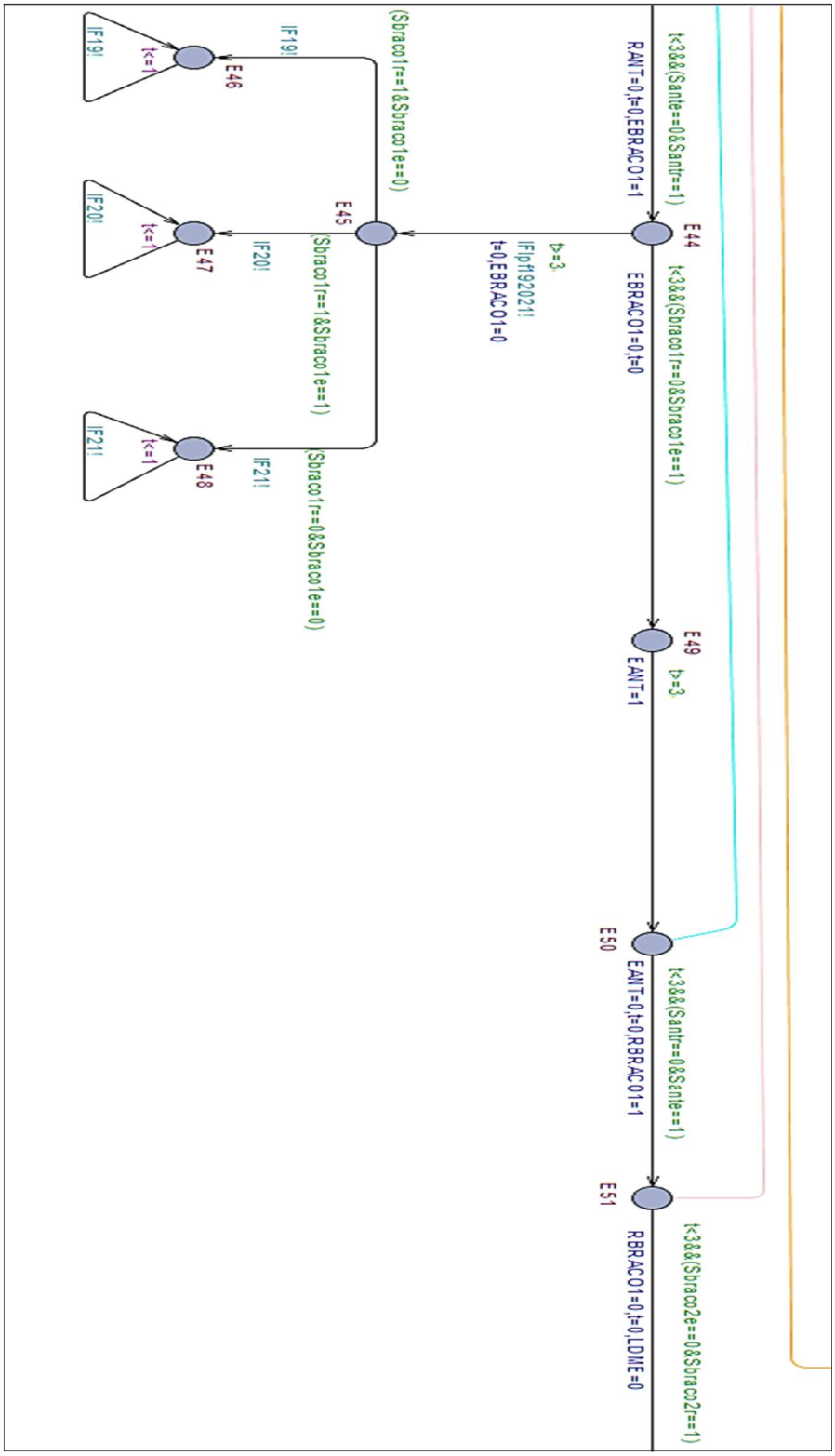

Figura 7.4.6 - Autômato G da estação de Classificação (estado 44 ao 51).

Tabela 7.4-Variáveis e Faltas da Classificação

Variável	Descrição da variável	Falta associada	Descrição da falta/ indicador	Expressão booleana de detecção
Sbraco1e	Sensor braço 1 estendido	F2	Travado no estado ligado/ IF2!	Após RBRACO1=1; (Sbraco1e==1&Sbraco1r==1)
		F21	Travado no estado desligado/ IF21!	Após EBRACO1=1; (Sbraco1r==0&Sbraco1e==0)
Sbraco1r	Sensor braço 1recuado	F3	Travado no estado desligado/ IF3!	Após RBRACO1=1; (Sbraco1e==0&Sbraco1r==0)
		F20	Travado no estado ligado/ IF20!	Após EBRACO1=1; (Sbraco1r==1&Sbraco1e==1)
Sbraco2e	Sensor braço 2 estendido	F5	Travado no estado ligado/ IF5!	Após RBRACO2=1; (Sbraco2e==1&Sbraco2r==1)
		F18	Travado no estado desligado/ IF18!	Após EBRACO2=1; (Sbraco2r==0&Sbraco2e==0)
Sbraco2r	Sensor braço 2recuado	F6	Travado no estado desligado/ IF6!	Após RBRACO2=1; (Sbraco2e==0&Sbraco2r==0)
		F17	Travado no estado ligado/ IF17!	Após EBRACO2=1; (Sbraco2r==1&Sbraco2e==1)
Sante	Sensor anteparo estendido	F9	Travado no estado desligado/ IF9!	Após EANT=1; (Santr==0&Sante==0)
		F14	Travado no estado ligado/ IF14!	Após RANT=1; (Sante==1&Santr==1)
Santr	Sensor anteparo recuado	F8	Travado no estado ligado/ IF8!	Após EANT=1; (Santr==1&Sante==1)
		F15	Travado no estado desligado/ IF14!	Após RANT=1; (Sante==0&Santr==0)
Spie	Sensor peça início da esteira	F11	Travado no estado ligado/ IF11!	LDME=1; (Spie==1&Spda==1)
Spda	Sensores peça detectada anteparo	F12	Travado no estado desligado/ IF12!	Após LDME=1; (Spie==0&Spda==0)
Sppd	Sinal peça preta detectada	Este é um sinal de identificação gerado pelo Spda. Caso haja falta, esta	-	-

		já terá ocorrido na etapa de detecção da peça.		
Spvd	Sinal peça vermelha detectada	Idem anterior.	-	-
Spprad	Sinal peça prata detectada	Idem anterior.	-	-
Sepp	Sensor nível máx. escorregador peça preta	Não é possível detectar faltas	-	-
Seppra	Sensor nível máx. escorregador peça prata	Não é possível detectar faltas	-	-
Sepv	Sensor nível máx. escorregador peça vermelha	Não é possível detectar faltas	-	-
EBRACO1	Estender braço 1	F1	Braço 1 travado na posição estendida/ IF1!	RBRACO1=1; (Sbraco1e==1&Sbraco1r==0)
RBRACO1	Recuar braço 1	F19	Braço 1 travado na posição recuada/ IF19!	EBRACO1=1; (Sbraco1r==1&Sbraco1e==0)
EBRACO2	Estender braço 2	F4	Braço 2 travado na posição estendida/IF4!	RBRACO2=1; (Sbraco2e==1&Sbraco2r==0)
RBRACO2	Recuar braço 2	F16	Braço 2 travado na posição recuada/IF16!	EBRACO2=1; (Sbraco2r==1&Sbraco2e==0)
LDME	Ligar e desligar motor da esteira	F10	Motor da esteira travado no estado desligado/IF10!	LDME=1; (Spie==1&Spda==0)
EANT	Estender anteparo	F13	Anteparo travado na posição estendida/IF13!	RANT=0; (Sante==1&Santr==0)
RANT	Recuar anteparo	F7	Anteparo travado na posição recuada/IF7!	EANT=0; (Santr==1&Sante==0)
R	Botão Reset	-	-	-
S	Botão Start	-	-	-