UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

BASES MÍNIMAS PARA O DIAGNÓSTICO DE FALHAS EM SISTEMAS A EVENTOS DISCRETOS

SAULO TELLES DE SOUZA LIMA

Rio de Janeiro, RJ - Brasil

Dezembro de 2008

BASES MÍNIMAS PARA O DIAGNÓSTICO DE FALHAS EM SISTEMAS A EVENTOS DISCRETOS

SAULO TELLES DE SOUZA LIMA

PROJETO SUBMETIDO AO CORPO DOCENTE DO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO ELETRICISTA.

Aprovado por:

JOÃO CARLOS DOS SANTOS BASILIO, Ph.D.
(Orientador)

MARCOS VICENTE DE BRITO MOREIRA, D.Sc.

ANTÔNIO EDUARDO CARRILHO DA CUNHA, Dr.Eng.

Rio de Janeiro, RJ - Brasil

Dezembro de 2008

AGRADECIMENTOS

Gostaria, primeiramente, de agradecer a Deus pela vida que me concedeu, e por tudo que consegui conquistar nela, sendo ele o principal responsável pelo que sou hoje.

Em segundo lugar, gostaria de agradecer a Jesus pelo exemplo que nos passou, sendo ele meu guia e modelo para ser uma pessoa melhor.

Aos meus pais, Carlos e Maria, por toda atenção e apoio que sempre me dedicaram, dando, pelo seu esforço, todas as condições para que eu alcançasse meus objetivos, me aconselhando e incentivando nos momentos difíceis, e compartilhando os bons momentos. Aos meus irmãos Sandro e Júnior pela companhia fraterna e incentivo constantes, sempre ao meu lado na caminhada da vida, que se segue. À minha namorada Flávia, pela compreensão e carinho que sempre demonstrou, apoiando nas escolhas e decisões a serem tomadas. Aos meus amigos, que sempre trouxeram muita alegria à minha vida.

Por fim, gostaria de manifestar a minha gratidão ao professor e orientador João Carlos dos Santos Basílio, pelos ensinamentos, tanto técnicos quanto morais, passados a mim, dedicando seu tempo e paciência para me orientar.

RESUMO

Saulo Telles de Souza Lima
UFRJ - EE

Projeto de Final de Curso
Dezembro 2008

Bases Mínimas para o Diagnóstico de Falhas em Sistemas a Eventos Discretos

A grande maioria dos processos industriais existentes hoje em dia são automatizados. Este fato faz com que a automação industrial ganhe cada vez mais importância, e conseqüentemente, ferramentas matemáticas para a sua modelagem, análise e controle se tornam instrumentos de grande valia para os engenheiros projetistas destes sistemas. Uma maneira de se abordar esses problemas é através da utilização dos chamados modelos a eventos discretos. Porém, como todo processo real, os sistemas fabris não são imunes a falhas não-observáveis (falha cuja ocorrência não possui sensor capaz de detectá-la). A teoria de Sistemas a Eventos Discretos (SED) se apresenta como uma opção para a solução deste problema, uma vez que permite informar a ocorrência de uma falha não-observável a partir da verificação de ocorrências de eventos observáveis do sistema. Em sistemas de grande porte, o número de eventos possíveis de serem observados (e que, por isso, serão modelados como eventos observáveis) é considerável, e além disso, a tecnologia utilizada na fabricação destes sensores pode ser muito cara. Portanto, é desejável a redução do número de sensores para a observação de eventos, permitindo ainda que a falha possa ser detectada. Neste trabalho será apresentado um método sistemático para se encontrar os menores conjuntos de eventos observáveis que permitem que uma falha em um SED seja detectada.

Sumário

Lista de Figuras

Capítulo 1

Introdução

A grande maioria dos processos industriais que existem hoje em dia são automatizados [1]. Isto se deve a diversos motivos: complexidade de gerenciamento, alto grau de precisão, perigo no manuseio etc. Em parte destes processos nem mesmo é possível que um ser humano realize alguma manipulação, como, por exemplo, a extração de petróleo em águas profundas, ou um controle rápido e preciso, como o de um sistema automático de controle da temperatura e da pressão de fornos industriais. Este fato faz com que a automação industrial ganhe cada vez mais importância e, conseqüentemente, ferramentas matemáticas para a sua modelagem, análise e controle se tornam instrumentos de grande valia para os engenheiros projetistas de sistemas de automação industrial. Uma maneira de se abordar esses problemas é através da utilização dos chamados modelos a eventos discretos [2, 3, 4, 5]. Sistemas a Eventos Discretos (SED) foram assim denominados por possuírem o espaço de estados discreto e a transição de um estado para outro acontecer através da ocorrência assíncrona de eventos, isto é, os instantes de ocorrência dos eventos não podem ser descritos por uma função no domínio do tempo. Estas características são as mesmas apresentadas pela maioria dos sistemas de automação industrial.

Porém, como todo processo real não é imune a falhas, e, em muitos casos, não existem sensores que permitam que certas falhas sejam detectadas de forma direta. A teoria de SED [6, 7, 8] se apresenta como uma opção para a solução deste problema, pois, a partir desta análise, é possível ter a informação da ocorrência de uma falha não-observável a partir da constatação de ocorrências de eventos observáveis do sistema, quando as falhas não levam o sistema a parar.

Neste contexto, os sensores utilizados para a observação destes eventos constituem boa parte do orçamento de um projeto de diagnóstico de falhas, pois em sistemas de grande porte, o número de eventos possíveis de serem observados é consideravelmente grande, e além disso, a tecnologia utilizada na fabricação destes sensores pode ser muito cara. Assim, um projeto de diagnóstico de falhas pode se tornar algo inviável economicamente. Portanto, deve-se otimizar a alocação de sensores para a observação de eventos, de forma que a falha possa ainda ser detectada. Com esta finalidade, será apresentado, neste trabalho, um método sistemático para se encontrar os menores conjuntos de eventos observáveis que permitem que uma falha em um SED seja detectada. Estes conjuntos serão chamados de bases mínimas para o diagnóstico de falhas em SED.

Este trabalho está estruturado da seguinte forma: no capítulo 2 são apresentados os conceitos de SED mais relevantes ao entendimento do método que será proposto no capítulo 3, incluindo uma revisão da teoria de linguagens e autômatos e o diagnóstico de falhas em SED. No capítulo 3 serão apresentadas definições e teoremas acerca do diagnóstico sob observação parcial de um sistema, incluindo a construção de um diagnosticador parcial a partir do diagnosticador centralizado; será proposto um método para a identificação de todas as bases mínimas para o diagnóstico de falhas em SED, que é o objetivo principal deste trabalho. No capítulo 4 é apresentada a conclusão dos resultados deste trabalho, bem como da sua contribuição para a área de automação industrial e sistemas a eventos discretos.

Capítulo 2

Diagnóstico de Falhas em SED

Neste capítulo serão revistos os conceitos mais relevantes da teoria de Sistemas a Eventos Discretos (SED) [2] necessários ao entendimento do método que será proposto no capítulo seguinte deste trabalho, onde será proposta uma maneira sistemática de encontrar subconjuntos do conjunto de eventos observados capazes de diagnosticar a ocorrência de falhas. Na seção 2.1 são apresentados alguns fundamentos sobre linguagens e autômatos tais como autômato determinístico, linguagem gerada por um autômato e projeção de linguagens. Ainda na seção 2.1 são apresentadas as composições produto e paralela entre dois autômatos, que serão úteis em alguns teoremas e passos do método citado acima. Na seção 2.2 são introduzidos os conceitos de autômatos não-determinísticos e observadores, e o principal assunto do capítulo: diagnóstico de falhas. Na última seção são enunciados os principais conceitos sobre diagnóstico de falhas em SED, sendo apresentadas as condições necessárias e suficientes para que uma linguagem seja diagnosticável, bem como o algoritmo para a construção do diagnosticador de falhas.

2.1 Linguagens e Autômatos determinísticos

2.1.1 Linguagens

Uma possibilidade formal de se estudar o comportamento lógico de um SED é através da teoria de linguagens e autômatos. Todo SED possui um conjunto de eventos E associado a ele, e cada evento é comparado a uma letra de um alfabeto, onde uma seqüência de eventos forma uma palavra de uma linguagem sobre o alfabeto. Portanto, todo conjunto de eventos E de um sistema é como um alfabeto. Uma seqüência que não possui eventos

é formada pelo *evento nulo* ε.

Definição 1 *(Linguagem)* Uma linguagem definida sobre um conjunto de eventos E é um conjunto de seqüências formadas por eventos pertencentes a E. □

Como exemplo, suponha um conjunto de eventos $E = \{a, b, g\}$. Podem-se definir as seguintes linguagens:

$$L_1 = \{\varepsilon, a, abb\},$$

que é formada por três seqüências somente, a linguagem

$$L_2 = \{\text{todas as possíveis seqüências de tamanho 3 iniciadas pelo evento } a\},$$

que contém nove seqüências, ou a linguagem

$$L_3 = \{\text{todas as possíveis seqüências de tamanho finito iniciadas pelo evento } a\}$$

que contém um número infinito de elementos.

Denota-se por E^* o conjunto de todas as seqüências finitas formadas por elementos de E, incluindo o evento nulo ε. Esta operação é chamada de *fecho de Kleene*. Note que E^* é infinito porém contável, pois é formado por seqüências de tamanhos arbitrariamente longos. Por exemplo, se $E = \{a, b, c\}$, então

$$E^* = \{\varepsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, ...\}$$

Qualquer linguagem construída a partir de um conjunto de eventos E é um subconjunto de E^*.

Suponha agora, uma seqüência $s = tuv$, com $t, u, v \in E^*$. Então: *(i)* ε, t, tu e tuv são chamados *prefixos* de s; *(ii)* ε, t, u, v, tu, uv e tuv são *subseqüências* de s; *(iii)* ε, v, uv e tuv são *sufixos* de s. Observe que tanto ε quanto s são prefixos, subseqüências e sufixos de s.

2.1.2 Autômatos

Um autômato é um dispositivo capaz de representar uma linguagem por regras bem definidas. Formalmente, é definido da seguinte forma.

Definição 2 *(Autômato determinístico)* Um *Autômato determinístico*, denotado por G, é uma sêxtupla

$$G = (X, E, f, \Gamma, x_0, X_m)$$

sendo X o conjunto de estados, E um conjunto finito de eventos associados às transições em G, $f : X \times E \to X$ a função de transição definida como $f(x, e) = y$, e significando que existe uma transição definida pelo evento e, do estado x para o estado y, $\Gamma : X \to 2^E$, a função de eventos ativos [1] ($\Gamma(x)$ é o conjunto de todos os eventos e para os quais $f(x, e)$ é definida, e é chamado de conjunto de eventos ativos de G em x), x_0 o estado inicial e $X_m \subseteq X$ o conjunto de estados marcados. □

Observação 1

(a) Um autômato é determinístico quando para todo estado $x \in X$ existir um único estado $y \in X$ tal que $f(x, e) = y$. No caso de um autômato não-determinístico*, $f(x, e)$ pode ser um conjunto de estados, ou seja, o contra-domínio de f é o conjunto 2^X para autômatos não-determinísticos.*

(b) A função de transição de estados f pode ser parcialmente definida em seu domínio. Isto significa que em cada estado pertencente ao conjunto X, não é necessário que a função f seja definida para todos os eventos do conjunto E. Caso contrário, a função f é dita ser uma função total em seu domínio.

(c) Por conveniência, f é sempre estendida do domínio $X \times E$ para o domínio $X \times E^*$, da seguinte forma recursiva:

$$\begin{aligned} f(x, \varepsilon) &:= x \\ f(x, se) &:= f(f(x, s), e) \text{ para } s \in E^* \ \wedge \ e \in E : f(x, s) = q \\ &\qquad \text{e } f(q, e) \text{ são definidos.} \end{aligned}$$

(d) É comum omitir Γ na definição de autômato, uma vez que pode ser obtido diretamente a partir da definição de f. □

[1] Dado um conjunto A, a notação 2^A denota o conjunto potência de A, que é o conjunto formado por todos os subconjuntos de A.

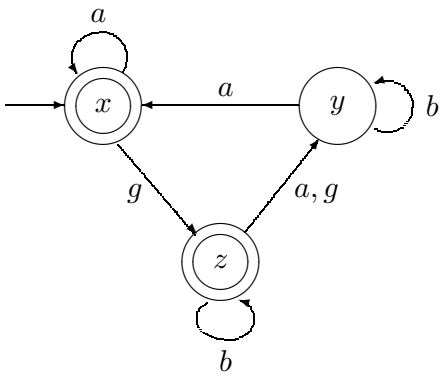

Figura 2.1: Diagrama de transição de estados do exemplo 1

O autômato G opera como se segue. Inicia no estado inicial x_0 e, após a ocorrência de um evento $e \in \Gamma(x_0)$, transita para o estado $f(x_0, e) \in X$. O processo continua baseado nas transições para as quais f é definida. O exemplo a seguir ilustra a representação e a evolução de um autômato.

Exemplo 1 *(Um autômato simples)* Considere o diagrama da figura 2.1, onde os círculos representam estados e os arcos nomeados representam transições entre os estados. Esse diagrama provê uma completa caracterização do autômato. O conjunto de estados do autômato é $X = \{x, y, z\}$ e o conjunto de eventos (alfabeto) do autômato, $E = \{a, b, g\}$ (rótulos das transições). Os arcos no diagrama são uma representação da função de transição do autômato, denotada por $f : X \times E \rightarrow X$, isto é, $f(x, a) = x$, $f(x, g) = z$, $f(y, a) = x$, $f(y, b) = y$, $f(z, b) = z$ e $f(z, a) = f(z, g) = y$. A notação $f(y, a) = x$ significa que se o autômato estiver no estado y, então após a ocorrência do evento a, o mesmo realizará uma transição instantânea para o estado x. A causa da ocorrência do evento a é irrelevante; o evento pode ser uma entrada externa para o sistema modelado pelo autômato, ou um evento espontaneamente gerado pelo próprio sistema modelado pelo autômato.

Note na figura 2.1 que o estado x possui uma seta de entrada que não é oriunda de nenhum outro estado. Essa é a notação utilizada para diferenciar o estado inicial de um estado comum em um diagrama de transição de estados. Dessa forma $x_0 = x$. Os estados representados por círculos duplos pertecem ao conjunto de estados marcados X_m.

Estados são marcados quando se deseja imprimir um significado especial à eles, e estão geralmente associados à conclusão de uma tarefa. São também referidos como estados "finais". Dessa forma, $X_m = \{x, z\}$. □

2.1.3 Linguagens gerada e marcada por um autômato

A ligação entre uma linguagem e um autômato pode facilmente ser identificada observando-se o diagrama de transição de estados de um autômato. O conjunto de todas as seqüências de eventos possíveis de serem executadas a partir do estado inicial forma a *linguagem gerada* por um autômato. O conjunto de seqüências pertencentes à linguagem gerada que levam o sistema a um estado marcado constitui a *linguagem marcada* por um autômato. As definições formais seguem abaixo.

Definição 3 *(Linguagens gerada e marcada)* A *linguagem gerada* por $G = (X, E, f, \Gamma, x_0, X_m)$ é

$$\mathcal{L}(G) := \{s \in E^* : f(x_0, s) \text{ é definida}\}.$$

A *linguagem marcada* por G é

$$\mathcal{L}_m(G) := \{s \in \mathcal{L}(G) : f(x_0, s) \in X_m\}.$$

□

Na definição acima é suposto que a função de transição estendida f tenha o seu domínio estendido para $X \times E^*$.

Exemplo 2 *(Linguagens gerada e marcada)* Suponha $E = \{a, b\}$ um conjunto de eventos. Considere a linguagem

$$L = \{a, aa, ba, aaa, aba, baa, bba, ...\}$$

consistindo de todas as seqüências formadas pelos eventos a ou b sempre terminadas pelo evento a. Essa linguagem é marcada pelo autômato de estados finitos $G = (X, E, f, \Gamma, x_0, X_m)$, representado pelo diagrama de transição da figura 2.2, em que $X = \{0, 1\}$, $x_0 = 0$, $X_m = \{1\}$ e f é definida como se segue: $f(0, a) = 1$, $f(0, b) = 0$, $f(1, a) = 1$, $f(1, b) = 0$.

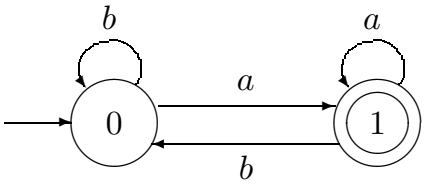

Figura 2.2: Diagrama de transição de estados para o autômato do exemplo 2

Assim, estando no estado inicial 0, a única maneira de se alcançar o estado marcado 1 é pela ocorrência do evento a, em algum momento. Alcançado este estado, o autômato somente retornará ao estado 0 após a ocorrência do evento b, repetindo-se o processo anteriormente descrito. Pode-se concluir do diagrama de transição da figura 2.2, que $\mathcal{L}_m(G) = L$. Note que f é uma função total em seu domínio, e portanto, a linguagem gerada por G é $\mathcal{L}(G) = E^*$. □

2.1.4 Projeção de linguagens

A projeção é uma função aplicada a uma determinada seqüência de eventos ou a toda uma linguagem, com relação a um conjunto de eventos. Formalmente, é definida da seguinte forma:

Definição 4 *(Projeção)* Sejam E e E_i conjuntos de eventos tais que $E_i \subset E$. A projeção de uma seqüência de eventos em E^* em E_i^* é definida da seguinte forma:

$$P_i : E^* \rightarrow E_i^*$$

satisfazendo as seguintes condições

$$\begin{aligned} P_i(\varepsilon) &:= \varepsilon \\ P_i(e) &:= \begin{cases} e & \text{se } e \in E_i \\ \varepsilon & \text{se } e \notin E_i \end{cases} \\ P_i(se) &:= P_i(s)P_i(e) \text{ para } s \in E^*,\ e \in E. \end{aligned}$$

A projeção P_i é estendida a linguagens simplesmente aplicando-se as regras acima a todas as seqüências pertencentes à linguagem da qual se deseja encontrar a projeção. Seja, portanto, uma linguagem $L \subseteq E^*$. Desta forma:

$$P_i(L) := \{t \in E_i^* : (\exists s \in L)[P_i(s) = t]\}.$$

□

Dados dois conjuntos de eventos E e E_i, em que $E_i \subset E$, tem-se que a projeção apaga os eventos presentes em seqüências formadas pelo maior conjunto E, que não pertencem ao menor conjunto E_i. Esse tipo de projeção é chamada de *projeção natural*. Pode-se definir também a projeção inversa de uma seqüência ou linguagem da seguinte forma:

Definição 5 *(Projeção inversa)* A projeção inversa de uma seqüência de eventos é definida como uma função

$$P_i^{-1} : E_i^* \rightarrow 2^{E^*}$$

sendo

$$P_i^{-1}(t) := \{s \in E^* : P_i(s) = t\}.$$

De maneira análoga à projeção, pode-se estender a definição de projeção inversa a linguagens. Para $L_i \subseteq E_i^*$,

$$P_i^{-1}(L_i) := \{s \in E^* : (\exists t \in L_i)\ [P_i(s) = t]\}.$$

□

Dada uma seqüência formada pelos eventos do menor conjunto de eventos (E_i), a projeção inversa retorna o conjunto de todas as seqüências formadas por eventos pertencentes ao maior conjunto de eventos (E), cujas projeções são a própria seqüência inicial.

A projeção inversa de uma linguagem definida em E_i e representada por um autômato pode ser implementada no diagrama de transição de estados desse autômato incluindo-se laços próprios rotulados por eventos pertencentes a $E \backslash E_i$ em todos os estados desse autômato [2].

Note que $P_i[P_i^{-1}(L)] = L$, mas, em geral, $L \subseteq P_i^{-1}[P_i(L)]$.

Exemplo 3 *(Projeção)* Considere $E = \{a, b, c\}$, $E_1 = \{a, b\}$, $E_2 = \{b, c\}$, e

$$L = \{c, ccb, abc, cacb, cabcbbca\}.$$

[2]Dados dois conjuntos A e B, a operação $A \backslash B = A - B$, isto é, o conjunto formado pelos elementos do conjunto A que não pertencem ao conjunto B.

Tem - se que:

$$\begin{aligned} P_1(L) &= \{\varepsilon, b, ab, abbba\} \\ P_2(L) &= \{c, ccb, bc, cbcbbc\} \\ P_1^{-1}(\{\varepsilon\}) &= \{c\}^* \\ P_1^{-1}(\{b\}) &= \{c\}^*\{b\}\{c\}^* \\ P_1^{-1}(\{ab\}) &= \{c\}^*\{a\}\{c\}^*\{b\}\{c\}^*. \end{aligned}$$

Observe que

$$P_1^{-1}[P_1(\{abc\})] = P_1^{-1}[\{ab\}] \supset \{abc\}$$

conforme mencionado acima. □

2.1.5 Produto de dois autômatos

O *produto* dos autômatos $G_1 = (X_1, E_1, f_1, \Gamma_1, x_{01}, X_{m1})$ e $G_2 = (X_2, E_2, f_2, \Gamma_2, x_{02}, X_{m2})$ é o autômato

$$G_1 \times G_2 := Ac(X_1 \times X_2, E_1 \cap E_2, f, \Gamma_{1\times 2}, (x_{01}, x_{02}), X_{m1} \times X_{m2})$$

em que

$$f[(x_1, x_2), e] := \begin{cases} (f_1(x_1, e), f_2(x_2, e)). & \text{se } e \in \Gamma_1(x_1) \cap \Gamma_2(x_2), \\ \text{não definido}, & \text{caso contrário} \end{cases}$$

Como conseqüência da definição acima, tem-se que $\Gamma_{1\times 2}(x_1, x_2) = \Gamma_1(x_1) \cap \Gamma_2(x_2)$. Além disso, a operação $Ac(G)$ (acessibilidade), presente na definição acima, garante que só existirão estados em $G_1 \times G_2$ que podem ser alcançados a partir do estado inicial por seqüências de eventos definidas pela função de transição de estados de $G_1 \times G_2$; tais estados são chamados de estados *acessíveis*.

Pode-se notar que no produto as transições dos autômatos envolvidos devem estar sincronizadas em um evento comum, isto é, um evento pertencente a $E_1 \cap E_2$. Um evento ocorre se, e somente se, ocorrer nos dois autômatos. Os estados de $G_1 \times G_2$ são constituídos por pares em que o primeiro componente é o estado atual de G_1 e o segundo componente

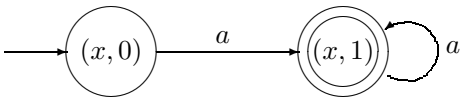

Figura 2.3: Autômato resultante do produto entre os autômatos das Figs. 2.1 e 2.2

é o estado atual de G_2. Pode-se verificar que

$$\begin{aligned} \mathcal{L}(G_1 \times G_2) &= \mathcal{L}(G_1) \cap \mathcal{L}(G_2) \\ \mathcal{L}_m(G_1 \times G_2) &= \mathcal{L}_m(G_1) \cap \mathcal{L}_m(G_2). \end{aligned}$$

Este é um resultado importante, e demonstra que é possível implementar a interseção de duas linguagens realizando-se o produto entre os autômatos que as geram.

Exemplo 4 *(Produto de dois autômatos)* O autômato da figura 2.3 é o resultado do produto entre os autômatos das figuras 2.1 e 2.2. O conjunto de eventos comuns é $\{a, b\}$. Os estados desse autômato são compostos por um estado do autômato da figura 2.1, e um estado do autômato da figura 2.2, necessariamente nesta ordem. No estado inicial $(x, 0)$, o único evento possível de ocorrer é o a, que leva de x para x e de 0 para 1; portanto $(x, 1)$ é o novo estado. Comparando-se o conjunto de eventos ativos de x e 1 em seus respectivos autômatos, conclui-se que o único evento possível de ocorrer é novamente o evento a, que leva de x para x e de 1 para 1, isto é, $(x, 1)$ novamente. Após isso, o produto está concluído. Somente transições etiquetadas pelo evento a são possíveis de ocorrer, pois o autômato da figura 2.1 nunca alcança um estado onde o evento b está ativo. Observe que o estado $(x, 1)$ é marcado, pois ambos, x e 1, são estados marcados em seus respectivos autômatos. □

2.1.6 Composição paralela de dois autômatos (composição síncrona)

A composição paralela dos autômatos $G_1 = (X_1, E_1, f_1, \Gamma_1, x_{01}, X_{m1})$ e $G_2 = (X_2, E_2, f_2, \Gamma_2, x_{02}, X_{m2})$ é o autômato

$$G_1 \| G_2 := Ac(X_1 \times X_2, E_1 \cup E_2, f, \Gamma_{1\|2}, (x_{01}, x_{02}), X_{m1} \times X_{m2}),$$

em que

$$f[(x_1,x_2),e] := \begin{cases} (f_1(x_1,e),f_2(x_2,e)), & \text{se } e \in \Gamma_1(x_1)\cap\Gamma_2(x_2) \\ (f_1(x_1,e),x_2), & \text{se } e \in \Gamma_1(x_1)\backslash E_2 \\ (x_1,f_2(x_2,e)), & \text{se } e \in \Gamma_2(x_2)\backslash E_1 \\ \text{não definido}, & \text{caso contrário} \end{cases}$$

Desta forma $\Gamma_{1\|2}(x_1,x_2) = [\Gamma_1(x_1)\cap\Gamma_2(x_2)] \cup [\Gamma_1(x_1)\backslash E_2] \cup [\Gamma_2(x_2)\backslash E_1]$.

Na composição paralela, um evento pertencente a ambos os autômatos somente poderá ser executado se os dois autômatos os executarem simultaneamente. Portanto, ao realizar essa operação, os dois autômatos são sincronizados em seus eventos comuns. Os outros eventos poderão ser executados sempre que possível, sem restrições. Se $E_1 = E_2$, então a composição paralela se reduz ao produto, pois todas as transições serão forçadas ao sincronismo. Utilizando-se da definição de projeção de linguagens, apresentada no item 2.1.4, pode-se chegar ao seguinte resultado quanto às linguagens gerada e marcada pela composição paralela:

$$\begin{aligned} \mathcal{L}(G_1\|G_2) &= P_1^{-1}[\mathcal{L}(G_1)] \cap P_2^{-1}[\mathcal{L}(G_2)] \\ \mathcal{L}_m(G_1\|G_2) &= P_1^{-1}[\mathcal{L}_m(G_1)] \cap P_2^{-1}[\mathcal{L}_m(G_2)]. \end{aligned}$$

A prova deste resultado não será mostrada. Entretanto, pode-se intuitivamente entendê-lo através da implementação da projeção inversa através de *laços próprios* e pelo produto de autômatos. A projeção inversa de uma linguagem pode ser representada incluindo-se laços próprios em todos os estados de G_1 e G_2; esses laços próprios deverão ser constituídos por eventos de $E_2\backslash E_1$ para G_1 e de $E_1\backslash E_2$ para G_2. Então, pode-se realizar o produto dos autômatos com laços próprios, e portanto chegar à interseção das projeções inversas das linguagens geradas por G_1 e G_2, que é a linguagem gerada pela composição paralela dos mesmos. Realizar a operação descrita acima é o mesmo que realizar a composição paralela, pois a ocorrência dos eventos que não pertencem a $E_1 \cap E_2$ estará garantida em qualquer estado, pela presença dos laços próprios.

Exemplo 5 *(Composição paralela)* O autômato da figura 2.4 é o resultado da composição paralela dos autômatos das Figs. 2.1 e 2.2, que serão referidos por G_1 e G_2, respectivamente, nesse exemplo. O conjunto de eventos comuns é $\{a,b\}$, e G_1 é o único que possui eventos particulares, nesse caso o evento g. Como no caso do produto, os estados de

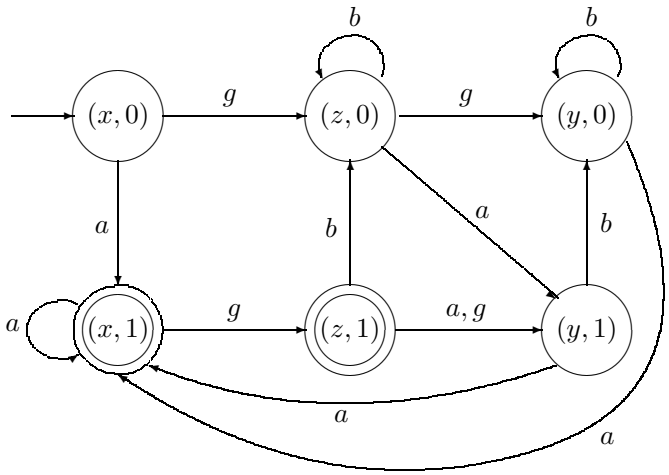

Figura 2.4: Autômato resultante da composição paralela entre os autômatos das Figs. 2.1 e 2.2

$G_1 \| G_2$ são formados por pares. No estado inicial $(x, 0)$, o evento comum a é o único possível de ocorrer e leva o sistema de $(x, 0)$ para $(x, 1)$, que por sua vez é um estado marcado, pois x é marcado em G_1 e 1 é marcado em G_2. Em contraste com $G_1 \times G_2$, uma outra transição pode ocorrer em $(x, 0)$: G_1 pode executar o evento g, sem a participação de G_2, e levar $G_1 \| G_2$ para o novo estado $(z, 0)$; após essa ocorrência, G_1 está no estado z e G_2 permanece no estado 0. O processo é repetido, encontrando-se todas as possíveis transições em $(x, 1)$ e $(z, 0)$, e em todos os novos estados gerados. Pode-se notar que todos os estados pertencentes a $X_1 \times X_2$ são acessíveis a partir do estado inicial $(x, 0)$, nesse exemplo. □

2.2 Autômatos não-determinísticos e SED parcialmente observado

2.2.1 Autômatos não-determinísticos

Na definição de autômato descrita na seção anterior, um evento e causa uma transição de um estado x para um único estado y. Suponha, entretanto, que um evento e possa ocasionar uma transição de um estado x para mais de um estado $(y_i,\ i = 1, 2, \ldots, n)$. As razões para se considerar essa hipótese são: *(i)* a não observação por um observador

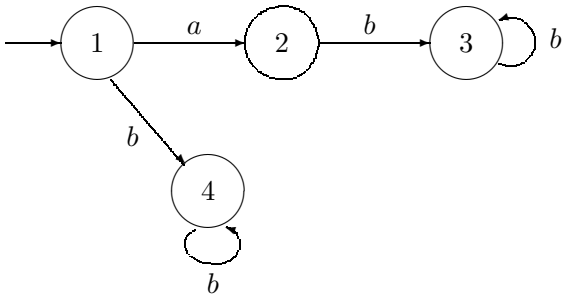

Figura 2.5: Exemplo de autômato não-determinístico para o caso em que o evento a é não-observável.

externo da ocorrência de um evento cuja ocorrência não deve ser considerada no modelo; *(ii)* a não abstração de informações do sistema real de forma a se construir um modelo que possua transições e estados intermediários ligando o estado x aos estados y_i, $i = 1, 2, \ldots, n$. Por exemplo, no autômato da figura 2.5, se o evento a não puder ser registrado e sua ocorrência não deve ser levada em conta no modelo, então, quando da observação do evento b, haverá duas possibilidades para o estado atual do sistema: os estados 3 e 4. Quando um evento e acarreta a transição de um estado x para vários estados, $f(x, e)$ deve representar um conjunto de possíveis novos estados. Além disso, pode ser conveniente relacionar uma transição a uma ocorrência do evento nulo (ε) num diagrama de transição de estados de um autômato. Novamente, essa opção é motivada pela falta de informação sobre o funcionamento do sistema, isto é, um evento que não pode ser observado por um observador externo (devido, por exemplo, à inexistência de um sensor capaz de registrar a ocorrência desse evento). Algumas vezes, há ainda a necessidade de se utilizar eventos nulos em operações de composição de autômatos que marcam linguagens regulares [2, pag. 94].

Exemplo 6 *(Autômato não-determinístico)* Considere o autômato da figura 2.6, onde o aspecto do não-determinismo está explícito por haver duas transições diferentes definidas pelo evento b no estado 1 (uma que leva o sistema do estado 1 para o estado 0 e outra que leva do estado 1 para ele mesmo). Além disso, o evento nulo ε pertence ao conjunto de eventos ativos dos estados 1 e 2. □

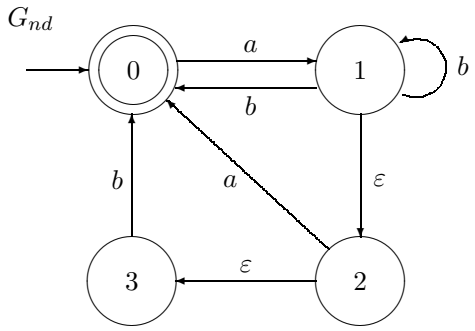

Figura 2.6: Autômato não-determinístico do exemplo 5.

Definindo-se o conjunto de eventos como sendo não mais o conjunto E mas sim o conjunto $E \cup \{\varepsilon\}$ e a função de transição de estados possuindo como domínio $X \times E \cup \{\varepsilon\}$ e como contra-domínio 2^X, chega-se à definição de autômato não-determinístico.

Definição 6 *(Autômato não-determinístico)* Um *autômato não-determinístico*, denotado por G_{nd}, é uma sêxtupla

$$G_{nd} = (X, E \cup \{\varepsilon\}, f_{nd}, \Gamma, x_0, X_m)$$

em que esses parâmetros possuem as mesmas interpretações dadas através da definição do autômato determinístico, apresentando somente duas diferenças:

1. $f_{nd} : X \times E \cup \{\varepsilon\} \rightarrow 2^X$, isto é $f_{nd} \subseteq X$ sempre que for definida.
2. O estado *inicial* pode ser um conjunto de estados, isto é $x_0 \subseteq X$. □

Pode-se estender f_{nd} para uma seqüência u, ao invés de se aplicar somente a um único evento, assim como foi feito para f em autômatos determinísticos. Em particular,

$$f_{nd}(x, ue) := \{z : z \in f_{nd}(y, e) \text{ para algum estado } y \in f_{nd}(x, u)\}.$$

Em outras palavras, identificam-se todos os estados y que são acessíveis a partir do estado x através da ocorrência da seqüência u; e, então, através da ocorrência do evento e, chega-se ao estado z pertencente ao conjunto $f_{nd}(x, ue)$. Como exemplo, no autômato da Fig 2.6, $f_{nd}(0, aa) = \{0\}$ pois $f_{nd}(0, a) = \{1, 2, 3\}$, e dentre os três estados pertencentes a $f_{nd}(0, a)$, o único que possui transição definida pelo evento a é o estado 2. Assim, como

$f_{nd}(2, a) = \{0\}$, então $f_{nd}(0, aa) = \{0\}$. Vale ressaltar que pela teoria de linguagens, $a = \{\varepsilon\}^* a = a\{\varepsilon\}^*$.

De forma similar ao autômato determinístico, o autômato não-determinístico também gera e marca linguagens, que são definidas como:

$$\begin{aligned} \mathcal{L}(G_{nd}) &= \{s \in E^* : (\exists x \in x_0)\ [f_{nd}(x, s) \text{ é definida}]\} \\ \mathcal{L}_m(G_{nd}) &= \{s \in \mathcal{L}(G_{nd}) : (\exists x \in x_0)\ [f_{nd}(x, s) \cap X_m \neq \emptyset]\} \end{aligned}$$

De acordo com a definição acima, uma seqüência pertence à linguagem gerada pelo autômato não-determinístico se existir uma trajetória no diagrama de transição de estados definida pela própria seqüência. Além disso, se uma dada trajetória levar o autômato a um estado marcado, então a seqüência que define tal trajetória pertence à linguagem marcada pelo autômato não-determinístico. Por exemplo, a seqüência ab está na linguagem marcada pelo autômato da figura 2.6, pois o mesmo pode assumir os estados 1 e 3 através do evento a e retornar ao estado 0 pela ocorrência do evento b, não importando se a mesma seqüência possa levar também a um estado que não seja marcado (estado 1 nesse caso).

2.2.2 Observador

A partir de um autômato não-determinístico pode-se obter um autômato determinístico equivalente, ou seja, que gera e marca a mesma linguagem do autômato não-determinístico. O espaço de estados do autômato determinístico equivalente será um subconjunto do conjunto potência do espaço de estados do autômato não-determinístico. O autômato determinístico equivalente é chamado de autômato observador. O observador estima o estado do autômato não-determinístico, depois de ocorrências de eventos observáveis. O procedimento para a construção do observador de um autômato não-determinístico está descrito a seguir.

Procedimento para construção do observador G_{obs} de um autômato não-determinístico G_{nd}

Seja $G_{nd} = (X, E \cup \{\varepsilon\}, f_{nd}, x_0, X_m)$ um autômato não-determinístico. Para cada estado $x \in X$ defina

$$UR(x) := f_{nd}(x, \varepsilon),$$

sendo $UR(x)$ os estados alcançados por eventos não-observáveis ou ε a partir de x. Para este procedimento, é suposto que se esteja trabalhando com a função f_{nd} estendida a sequências pertencentes a $(E \cup \{\varepsilon\})^*$. Para um conjunto $B \in 2^X \backslash \emptyset$, defina

$$UR(B) = \bigcup_{x \in B} UR(x).$$

Então $G_{obs} = (X_{obs}, E, f_{obs}, x_{0,obs}, X_{m,obs})$ é construído da seguinte forma.

Procedimento 1

Passo 1: *Defina* $x_{0,obs} = UR(x_0)$.

Passo 2: $X_{obs} = \{x_{0,obs}\}$

Passo 3: *Para cada* $x \subseteq X_{obs}$ *e* $e \in E$, *defina*

$$f_{obs}(x, e) = UR(\{y \in X : (\exists x_e \in x)\ [y \in f_{nd}(x_e, e)]\}).$$

Passo 4: *Faça* $X_{obs} = X_{obs} \cup \{f_{obs}(x, e)\}$

Passo 5: $X_{m,obs} = \{S \subseteq X : S \cap X_m \neq \emptyset\}$.

Passo 6: *Repita os passos 3 a 6 até que toda a parte acessível de* G_{obs} *tenha sido obtida.*

□

Exemplo 7 *(Observador de um autômato não-determinístico)* O autômato da figura 2.7 é o observador construído para o autômato não-determinístico da figura 2.6. O estado inicial do observador é $x_{0,obs} = UR(0) = \{0\}$. O único evento ativo neste estado é o a, e portanto $f_{obs}(0, a) = UR(1) = \{1, 2, 3\}$. No estado {1,2,3} existem dois eventos ativos, a e b. Nesse caso, a função de transição estará definida como, $f_{obs}(\{1, 2, 3\}, a) = UR(f_{nd}(2, a)) = \{0\}$ e $f_{obs}(\{1, 2, 3\}, b) = UR(f_{nd}(1, b) \cup f_{nd}(3, b)) = UR(\{1, 3\}) = \{0, 1, 2, 3\}$. O novo estado formado $\{0, 1, 2, 3\}$ também possui a e b como eventos ativos. Pela mesma análise feita

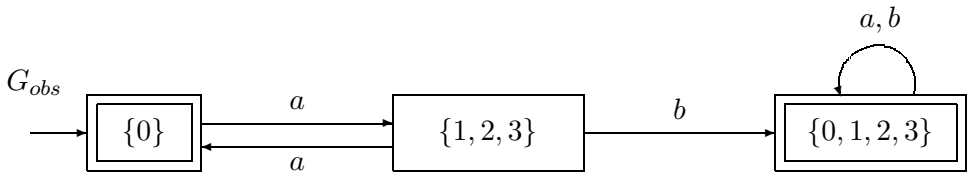

Figura 2.7: Observador do autômato não-determinístico da figura 2.6.

anteriormente, pode-se mostrar que $f_{obs}(\{0,1,2,3\}, a) = f_{obs}(\{0,1,2,3\}, b) = \{0,1,2,3\}$. O processo de construção de G_{obs} está completo, pois todos os estados criados foram analisados. Note que, de fato, G_{nd} e G_{obs} geram e marcam a mesma linguagem. Por isso são chamados de *autômatos equivalentes.* □

2.2.3 SED parcialmente observado

Conforme dito anteriormente, transições definidas por ε representam a ocorrência de um evento não-observável. Por esse motivo, ao invés de se utilizar essa representação com o evento nulo e obter um autômato não-determinístico, podem-se rotular essas transições com eventos "genuínos", porém caracterizados como eventos não-observáveis. Além disso, se não for possível que um mesmo evento rotule diversas transições definidas em um estado, então o modelo do sistema será um autômato determinístico cujo conjunto de eventos E é particionado em dois subconjuntos: E_o, conjunto de eventos observáveis, e E_{uo}, conjunto de eventos não-observáveis. Nessas condições, denomina-se esse SED como "parcialmente observado". É possível utilizar o procedimento 1 para construir um observador para um autômato determinístico com eventos não-observáveis $G = (X, E, f, x_0, X_m)$ com $E = E_o \dot{\cup} E_{uo}$, em que $\dot{\cup}$ denota partição, isto é, $E_o \cap E_{uo} = \emptyset$. Para tanto, basta tratar os eventos não-observáveis, como se fossem o evento nulo ε. O observador terá, nesse caso, E_o como conjunto de eventos. Para que o procedimento 1 possa ser utilizado, define-se:

$$UR(x) = \{y \in X : (\exists t \in E_{uo}^*)\ [f(x,t) = y]\}.$$

A definição acima é estendida para um conjunto de estados $B \subseteq X$, da seguinte forma:

$$UR(B) = \bigcup_{x \in B} UR(x).$$

Pode-se, agora, construir o observador de um autômato parcialmente observado, utilizando-se o procedimento 1, porém substituindo-se a definição de estados alcançados por eventos não-observáveis (ε) para autômatos não-determinísticos, pela definição de alcance não-observável em relação a eventos em E_{uo}.

Denotando P a projeção natural de E^* em E_o^*, tem-se, por construção, que o observador $Obs(G)$ possui as seguintes características:

- $Obs(G)$ é um autômato determinístico.
- $\mathcal{L}[Obs(G)] = P[\mathcal{L}(G)]$.
- $\mathcal{L}_m[Obs(G)] = P[\mathcal{L}_m(G)]$.

Exemplo 8 *(Autômato parcialmente observado)* Considere o autômato mostrado na figura 2.8. O conjunto de eventos não-observáveis é

$$E_{uo} = \{e_d, \sigma\}.$$

O observador $G_{obs} := Obs(G)$ está mostrado na figura 2.9. O estado $\{6, 8, 9\}$ representa o conjunto de possíveis estados em que o sistema pode estar após a observação da seqüência ac em G_{obs}. □

O conceito de evento não-observável pode ser aplicado na modelagem de sistemas práticos. Uma possível aplicação é na representação de uma falha que não cause uma mudança observável no estado de um sistema. Nesse caso, os sensores disponíveis não são capazes de detectar a ocorrência do evento que representa a falha do sistema.

Neste trabalho serão considerados somente autômatos determinísticos com eventos não-observáveis, devido ao fato de que a modelagem de um sistema, cuja transição de estados não seja conhecida inteiramente, pode ser feita por autômatos não-determinísticos ou por autômatos determinísticos com eventos não-observáveis.

2.3 Diagnóstico de falhas

Nos muitos casos em que o modelo de um sistema possui eventos não-observáveis, pode ser necessário determinar a ocorrência de algum(ns) desses eventos, após a execução de

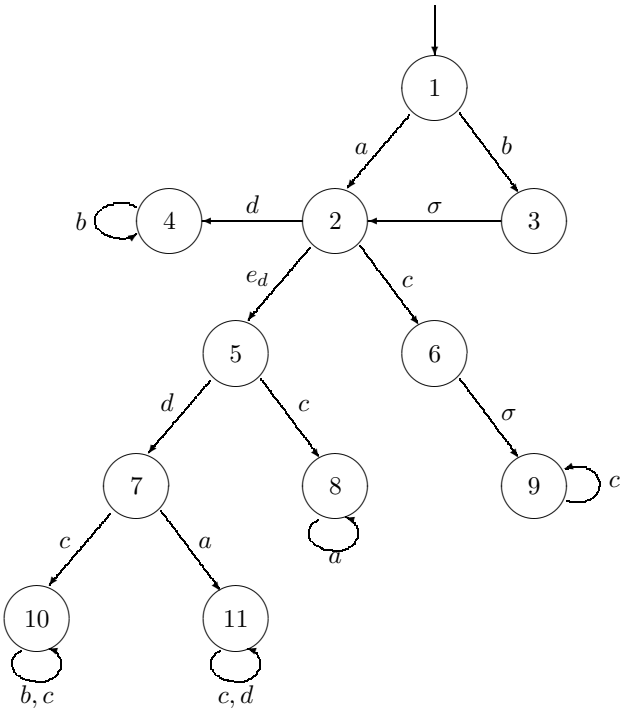

Figura 2.8: Autômato parcialmente observado do exemplo 8.

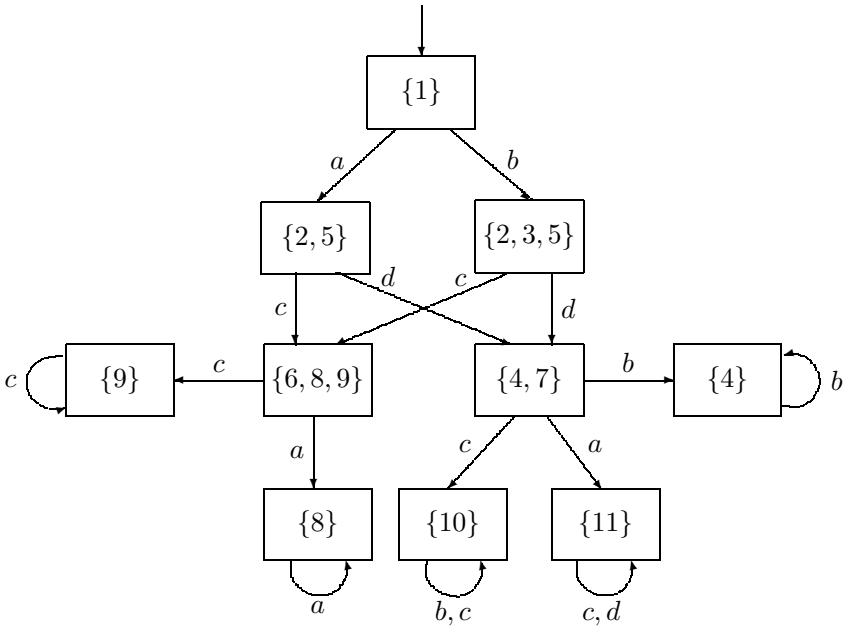

Figura 2.9: Observador do autômato da figura 2.8.

uma seqüência de eventos pelo sistema. Na teoria de diagnóstico de falhas, o evento não-observável cuja ocorrência deseja-se determinar é denominado falha do sistema. Um dispositivo capaz de informar da ocorrência (quando possível) de um determinado evento não-observável é chamado de diagnosticador [6].

2.3.1 Diagnosticador

Um diagnosticador $G_{diag} = (X_d, E_o, f_d, \Gamma_d, x_{0_d})$ é um observador modificado com o objetivo de indicar, se possível, a ocorrência de um dado evento não-observável (falha). O diagnosticador de um autômato G é denotado por $Diag(G)$ ou G_{diag}. A diferença entre um observador e um diagnosticador é o acréscimo de indicadores aos estados de G para formar os estados de G_{diag}. Esses indicadores têm como objetivo informar sobre a possibilidade ou não da ocorrência da falha (ou como será visto mais adiante, a ocorrência ou não, de fato, da falha). Para efeitos de simplicidade, serão considerados somente os casos em que um único evento não observável necessita ser detectado, $e_d \in E_{uo}$. Serão utilizados dois tipos de marcações: N para "o evento e_d ainda não ocorreu" e Y para "o evento e_d ocorreu". A notação a ser utilizada para cada estado $x \in X$ será xN e/ou xY, respectivamente. Para a construção de $Diag(G)$, utiliza-se o seguinte procedimento, que é uma modificação daquele utilizado para a construção de $Obs(G)$ (procedimento 1):

Procedimento 2

Passo 1. *Na construção de $UR(x_0)$:*

(a) Deve-se marcar com a letra N os estados que puderem ser alcançados através de uma seqüência não-observável pertencente a $[E_{uo} \backslash \{e_d\}]^*$, a partir de x_0;

(b) Deve-se marcar com a letra Y os estados que puderem ser alcançados através de uma seqüência não-observável que contém ao menos uma ocorrência de e_d, a partir de x_0;

(c) Se um estado z puder ser alcançado por, ao menos, uma seqüência que contém o evento e_d e uma seqüência que não contém o evento e_d, então deve-se criar dois elementos no estado inicial de $Diag(G)$: zY e zN.

Passo 2. *Na construção dos estados subseqüentes de $Diag(G)$:*

(a) Deve-se seguir as regras para a função de transição de $Obs(G)$, implementando-se as modificações apresentadas em M1.;

(b) A marcação Y deve ser propagada, isto é, todo estado alcançável a partir de zY deve receber a marcação Y para indicar que e_d ocorreu no processo de alcance do estado z e, portanto, no processo de alcance dos novos estados.

Passo 3. *$Diag(G)$ não possui estados marcados.* □

O $Diag(G)$ possui E_o como conjunto de eventos, sendo um autômato determinístico que gera a linguagem $\mathcal{L}[Diag(G)] = P[\mathcal{L}(G)]$. Cada estado de $Diag(G)$ é um subconjunto de $X \times \{N, Y\}$.

Exemplo 9 *(Diagnosticador)* A figura 2.10 mostra o diagnosticador G_{diag} do autômato G da figura 2.8, em que e_d é o evento a ser diagnosticado. O diagrama de transição de estados de G_{diag} mostra que após a ocorrência do terceiro evento observável de qualquer seqüência, é possível afirmar se o evento e_d ocorreu ou não, pois todos os componentes dos estados de G_{diag} possuem somente a marcação Y ou somente a marcação N. Por outro lado, enquanto o terceiro evento observável não ocorrer, o diagnosticador permanecerá em estados que possuem componentes com marcação Y e N, impossibilitando a afirmação da ocorrência do evento e_d. □

Observação 2 Pode-se construir o autômato $Diag(G)$ através da composição paralela de G com o autômato A_{label} mostrado na figura 2.10, seguido do cálculo do observador da maneira usual. Em outras palavras,

$$Diag(G) = Obs(G \| A_{label})$$

A ocorrência do evento não-observável e_d pode ser inferida pelo exame dos estados do diagnosticador. Os critérios para realização desse diagnóstico estão descritos abaixo.

- Se todos os estados de G, pertencentes ao estado presente de $Diag(G)$, possuírem a marcação N, então o evento e_d ainda não ocorreu. Esse estado é denominado *estado normal*.

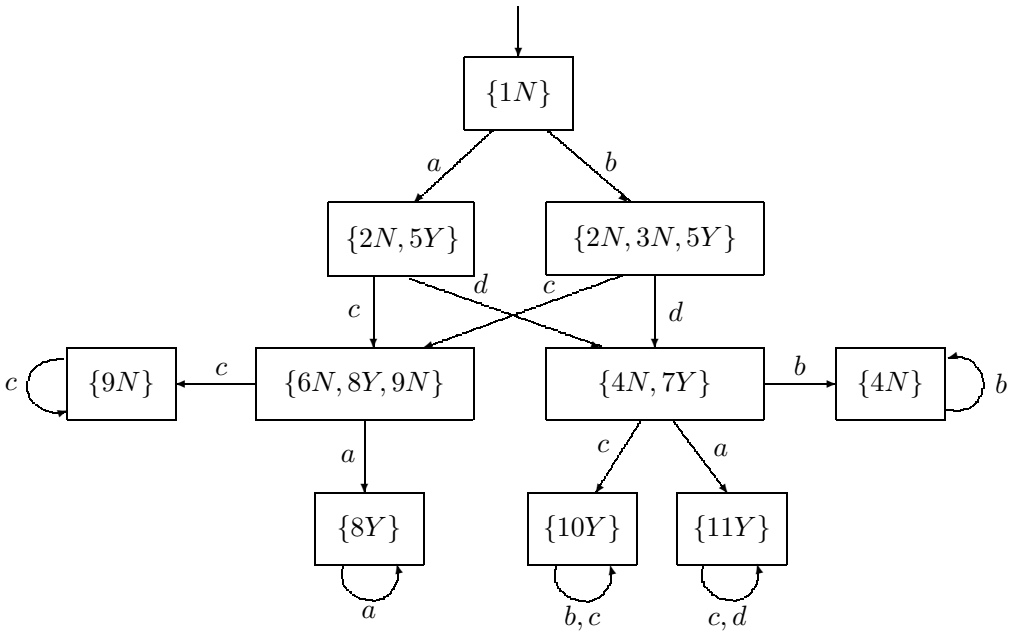

Figura 2.10: Diagnosticador do autômato da figura 2.8.

- Se todos os estados de G, pertencentes ao estado presente de $Diag(G)$, possuírem a marcação Y, então o evento e_d já ocorreu, em algum instante no passado. Esse estado é chamado de *estado certo*. Se $t \in P[\mathcal{L}(G)]$ foi observado e $f_d(x_{0_d}, t)$ é um estado certo, então todas as seqüências pertencentes a $P^{-1}(t) \cap \mathcal{L}(G)$ devem conter e_d.

- Se o estado atual de $Diag(G)$ possuir ao menos um estado de G com marcação N e um com marcação Y, então o evento e_d pode ter ocorrido ou não. Esse estado é chamado de *estado incerto*. Nesse caso, existem no mínimo duas seqüências $s_1, s_2 \in \mathcal{L}(G)$ tal que $P(s_1) = P(s_2)$ (ambas levam ao mesmo estado de $Diag(G)$) , em que s_1 contém o evento e_d, mas s_2 não.

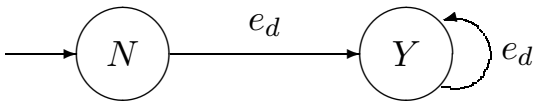

Figura 2.11: Autômato A_{label} de marcação de estados para a construção do diagnosticador

2.3.2 Diagnosticabilidade

A noção de diagnosticabilidade está baseada na possibilidade de se detectar qualquer tipo de falha em um sistema, com um atraso finito, utilizando-se as ocorrências de eventos observáveis registradas. A definição de diagnosticabilidade que se segue, pressupõe que a linguagem gerada por G seja "viva", isto é, para todo $s \in \mathcal{L}(G)$, existe $e \in E$ tal que $se \in \mathcal{L}(G)$.

Definição 7 *(Diagnosticabilidade)* Um evento não-observável e_d é não-diagnosticável em uma linguagem $\mathcal{L}(G)$ se existirem duas seqüências s_n e s_y em $\mathcal{L}(G)$ que satisfaçam às seguintes condições: (i) s_y contém e_d, mas s_n não contém; (ii) s_y possui uma continuação de tamanho arbitrariamente longo, após a ocorrência de e_d; e (iii) $P(s_n) = P(s_y)$. Quando não existirem tais seqüências, então e_d é dito ser diagnosticável em $\mathcal{L}(G)$. □

Para evitar situações em que existam ciclos de eventos não-observáveis após a ocorrência de e_d, supõe-se que G não possui tais ciclos, após a ocorrência do evento e_d. Esta hipótese é feita para que o diagnosticador possa se atualizar após a ocorrência de e_d.

Quando um diagnosticador possui um ciclo de estados incertos, a ocorrência passada de e_d pode vir a não ser diagnosticada, deixando uma incerteza sobre tal ocorrência. Corresponderá a uma violação de diagnosticabilidade, se existirem duas seqüências s_n e s_y, em que $e_d \in s_y$, $e_d \notin s_n$, $P(s_n) = P(s_y)$, e $P(s_y)$ entra e permanece em um ciclo de estados incertos de G_{diag}. Neste caso, examinando-se os autômatos G e G_{diag}, podem-se associar, no mínimo, dois ciclos diferentes em G ao ciclo de estados incertos contido no diagnosticador, nos quais o prefixo da seqüência que leva a um dos ciclos contém e_d e o prefixo da seqüência que leva ao outro ciclo não o possui. Além disso, os estados de G com marcação Y, que pertencem aos estados incertos do ciclo em G_{diag}, também formam um ciclo em G; o mesmo ocorre aos estados de G com marcação N, pertencentes aos estados incertos do ciclo em G_{diag}. Os ciclos de eventos incertos pertencentes a G_{diag} que satisfaçam às características citadas anteriormente, são chamados de *ciclos indeterminados*, e a sua presença em um diagnosticador viola as condições para a diagnosticabilidade. A violação da diagnosticabilidade também implica a presença de pelo menos um ciclo

indeterminado. Pode-se mostrar que [6] uma condição necessária e suficiente para a diagnosticabilidade de $\mathcal{L}(G)$ com relação a P_o e e_d é que o diagnosticador associado a G não possui ciclos indeterminados.

É importante salientar, que a presença de ciclos de estados incertos (porém, não indeterminados) em um diagnosticador não implica a impossibilidade do diagnóstico da ocorrência da falha em um sistema, uma vez que sempre haverá um evento que, em caso de ocorrência da falha, levará o diagnosticador a um estado certo.

Capítulo 3

Bases mínimas para o diagnóstico de falhas

No capítulo anterior, foi visto que uma falha que não pode ser observada a partir de um observador externo, por exemplo um sensor, pode ser modelada como um evento não-observável de um SED. Mesmo não sendo observada, deseja-se detectar a sua ocorrência, quando possível. O dispositivo capaz de realizar essa tarefa é o diagnosticador, que monitora o comportamento do sistema através de ocorrências de eventos observáveis e informa sobre a ocorrência da falha.

Os eventos de um sistema podem ser comandos advindo de controladores ou podem ser decorrentes da dinâmica do próprio sistema: por exemplo, uma peça manufaturada chegando através de uma esteira a uma máquina destinada a embalá-la. A fim de detectar a ocorrência desses eventos, é necessário utilizar sensores. Entretanto, em sistemas que possuem um elevado número de eventos a serem monitorados ou um evento cuja observação depende de um certo sensor de alto custo, é desejável a redução de custos através da diminuição do número de eventos a serem observados. Nesses casos, seria conveniente utilizar somente um subconjunto do conjunto de eventos observáveis para o diagnóstico da falha, eliminando a necessidade da aquisição de um sensor para cada evento possível de ser observado, e assim, permitindo que seja encontrada a alocação ótima de sensores para um dado projeto.

Neste capítulo será proposto um método sistemático para encontrar subconjuntos do conjunto de eventos observáveis que permitam o diagnóstico da falha.

A sua estrutura é a seguinte: na seção 3.1 serão apresentadas definições e enunciados

resultados preliminares que permitirão o entendimento do método proposto; na seção 3.2 será apresentado o método desenvolvido com a finalidade de encontrar os subconjuntos do conjunto de eventos observáveis que permitam o diagnóstico da falha, e na seção 3.3 será apresentado um exemplo com a finalidade de ilustrar a aplicação do método proposto.

3.1 Definições e resultados preliminares

Para o bom entendimento da teoria proposta neste trabalho, será necessário enunciar algumas definições e apresentar alguns resultados preliminares.

3.1.1 Diagnosticador parcial

Levantando-se a hipótese de que uma falha pode ser diagnosticada a partir da observação de um subconjunto E_{o_i} do conjunto de eventos observáveis E_o de um autômato G, então o diagnosticador que possui esse subconjunto como conjunto de eventos será referido como diagnosticador parcial de $\mathcal{L}(G)$, diferentemente do diagnosticador centralizado G_{diag} que possui E_o como conjunto de eventos. A definição formal de diagnosticador parcial segue abaixo [9].

Definição 8 *(Diagnosticador parcial)* Um diagnosticador parcial G_{d_i} é um autômato diagnosticador de $\mathcal{L}(G)$ cujo conjunto de eventos E_{o_i} é um subconjunto de E_o. □

Existe a possibilidade de se construir um diagnosticar parcial G_{d_i} utilizando-se o procedimento para a construção de G_{diag}, descrito no capítulo anterior, tomando-se E_{o_i} como conjunto de eventos observáveis, e $E \backslash E_{o_i}$ como conjunto de eventos não-observáveis. Porém, se houver a necessidade de se construir uma grande quantidade de diagnosticadores parciais, esta tarefa pode ser penosa, tornando-se conveniente a utilização do seguinte teorema.

Teorema 1 Suponha que $G_{diag} = (X_d, E_o, f_d, \Gamma_d, x_{0_d})$ e $G_{d_i} = (X_{d_i}, E_{o_i}, f_{d_i}, \Gamma_{d_i}, x_{0_{d_i}})$ sejam, respectivamente, o diagnosticador centralizado e um diagnosticador parcial, e que $E_{o_i} \subset E_o$ e $E_{o_i} \neq \emptyset$. Então, $Obs(G_{diag}, E_{o_i}) = (\hat{X}_d, E_{o_i}, \hat{f}_d, \hat{\Gamma}_d, \hat{x}_{0_d})$ (observador de G_{diag}

com respeito à projeção $P_{o_i} : E_o^* \rightarrow E_{o_i}^*$) e G_{d_i} são iguais, a menos da seguinte equivalência de estados:

$$\hat{x}_d = \{x_{1_d}, x_{2_d}, ..., x_{n_d}\} \in \hat{X}_d,\ x_{k_d} \in X_d \Leftrightarrow x_{d_i} = \bigcup_{k=1}^{n} x_{k_d} \in X_{d_i}.$$

□

A prova deste teorema está mostrada em [9], de modo que será suprimida neste trabalho.

De acordo com o teorema acima, o diagnosticador parcial G_{d_i}, que observa as ocorrências dos eventos pertencentes a E_{o_i} e informa sobre a ocorrência do evento não-observável e_d, que representa a falha, pode ser construído a partir do diagnosticador centralizado G_{diag} aplicando-se as seguintes regras:

1. Os estados de G_{d_i} são obtidos agrupando-se todos os estados de G_{diag} que são conectados por um evento pertencente a $E_o \backslash E_{o_i}$, em um único estado formado pela união dos conjuntos dos estados agrupados.

2. A função de transição para cada estado x_{d_i} definido pelo passo anterior é definida como

$$f_{d_i}(x_{d_i}, e) = \bigcup_{e \in E_{o_i} \cap \Gamma_d(x_d)} f_d(x_d, e),$$

em que os estados x_d são os estados de G_{diag} que foram agrupados, formando o novo estado x_{d_i}.

O exemplo a seguir ilustrará a aplicação do procedimento acima descrito.

Exemplo 10 *(Construção de um diagnosticador parcial a partir do diagnosticador centralizado)* O diagnosticador parcial mostrado na figura 3.1 foi construído calculando-se o observador do diagnosticador centralizado G_{diag} da figura 2.10, com respeito à projeção $P_{o_1} : E_o^* \rightarrow E_{o_1}^*$, para $E_{o_1} = \{a, c, d\}$, de acordo com o procedimento acima descrito e representa a situação em que o evento b se tornou não-observável. Em G_{diag}, os estados $\{2N, 3N, 5Y\}$ e $\{4N\}$ são conectados, respectivamente, aos estados $\{1N\}$ e $\{4N, 7Y\}$ através do evento b. Assim, tais pares de estados devem ser agrupados para que sejam formados, respectivamente, os novos estados $\{1N, 2N, 3N, 5Y\}$ e $\{4N, 7Y\}$ pertencentes

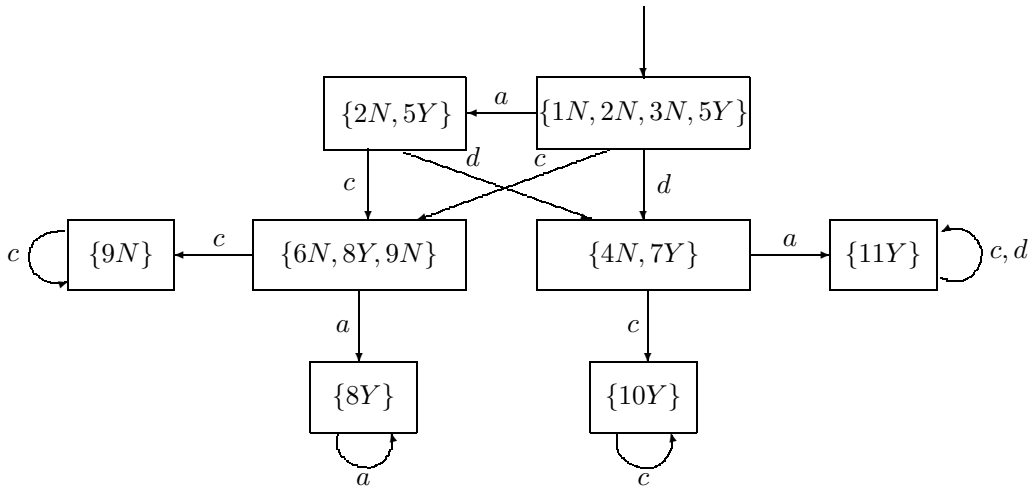

Figura 3.1: Diagnosticador parcial da linguagem gerada pelo sistema da figura 2.8, considerando $E_{o_1} = \{a, c, d\}$ como conjunto de eventos observáveis do sistema.

a X_{d_1}. Note que o estado $\{4N, 7Y\}$ não é, essencialmente, um novo estado, mas, foi assim considerado pelo fato de integrar o estado $\{4N\}$ após terem sido feitas as definições dos estados de $G_{d_1} = Obs(G_{diag}, E_{o_1})$. A função de transição para estes estados é definida da seguinte forma:

$$\begin{aligned}
f_{d_1}(\{1N, 2N, 3N, 5Y\}, a) &= f_d(\{1N\}, a) \\
&= \{2N, 5Y\} \\
f_{d_1}(\{1N, 2N, 3N, 5Y\}, c) &= f_d(\{2N, 3N, 5Y\}, c) \\
&= \{6N, 8Y, 9N\} \\
f_{d_1}(\{1N, 2N, 3N, 5Y\}, d) &= f_d(\{2N, 3N, 5Y\}, d) \\
&= \{4N, 7Y\} \\
f_{d_1}(\{4N, 7Y\}, a) &= f_d(\{4N, 7Y\}, a) \\
&= \{11Y\} \\
f_{d_1}(\{4N, 7Y\}, c) &= f_d(\{4N, 7Y\}, c) \\
&= \{10Y\}.
\end{aligned}$$

O restante dos estados de G_{d_1} são iguais aos de G_{diag}, bem como as definições associadas a esses estados na função de transição. Pode-se observar também que o ciclo formado pelo

evento b no estado $\{4N\} \in X_d$ não existe em G_{d_1}, de forma que se as seqüências $\{adb\}\{b\}^*$ ou $\{bdb\}\{b\}^*$ ocorrerem, o diagnosticador parcial permanecerá no estado $\{4N, 7Y\}$, estando impossibilitado de assumir o estado $\{4N\}$, que denota a não ocorrência de e_d. Todavia, esse fato não leva à perda de diagnosticabilidade da $\mathcal{L}(G)$ com respeito a P_{o_1} e e_d pois, se o evento e_d ocorrer e o diagnosticador parcial assumir o estado $\{4N, 7Y\}$, o evento b não será o próximo a ocorrer, como pode ser constatado pelo diagrama de transição de estados da figura 2.8. Nesse caso, o diagnosticador parcial assumirá um dos estados certos $\{10Y\}$ ou $\{11Y\}$ após a ocorrência de uma seqüência finita de eventos (mais precisamente uma seqüência formada por um único evento), diagnosticando a falha. □

Na prática, o uso do teorema 1 para a construção de diagnosticadores parciais se mostra mais eficiente que a utilização do procedimento para a construção de um diagnosticador para o novo conjunto de eventos observáveis, visto que na construção de um diagnosticador é necessário calcular estado por estado propagando-se as marcações N e Y, enquanto que no cálculo de um observador não há nenhuma marcação que necessite ser propagada.

3.1.2 Eventos definitivos

Existem eventos que levam o diagnosticador centralizado a um estado certo, quando a falha é diagnosticável. Estes eventos serão referidos como *eventos definitivos*, cuja definição formal é apresentada a seguir.

Definição 9 *(Eventos definitivos)* Um evento $e_u \in E_o$ é um evento definitivo se $\exists s e_u \in \mathcal{L}(G_{diag})$, tal que

$$f_d(x_{0_d}, s) = y \text{ em que } y \text{ é um estado incerto ou um estado certo de } G_{diag}$$

e

$$f_d(y, e_u) = z \text{ em que } z \text{ é um estado certo } G_{diag}.$$

□

No diagnosticador centralizado da figura 2.10, por exemplo, todos os eventos observáveis são também definitivos. Contudo, em sistemas com uma grande quantidade de eventos observáveis, a ocorrência deste fato se torna menos provável.

3.1.3 Bases para o diagnóstico de falhas

No capítulo anterior, foram enunciadas condições necessárias e suficientes para o diagnóstico de falhas de um determinado autômato G, que possui um conjunto de eventos observáveis E_o. Supondo-se que exista um subconjunto E_{o_i} do conjunto de eventos observáveis que permita o diagnóstico de falhas, esse subconjunto será identificado como uma *base para o diagnóstico de falhas* deste sistema. Um autômato pode ter mais de uma base, como será visto mais adiante. A definição formal de bases para o diagnóstico de falhas é apresentada a seguir.

Definição 10 *(Bases para o diagnóstico de falhas)* Dado um autômato $G = (X, E, f, \Gamma, x_0, X_m)$, em que $E = E_o \cup E_{uo}$ e $e_d \in E_{uo}$, um conjunto $E_{o_i} \subseteq E_o$ é uma base para o diagnóstico de $\mathcal{L}(G)$ com respeito a $P_{o_i} : E^* \rightarrow E_{o_i}^*$ e e_d, se existir $G_{d_i} = (X_{d_i}, E_{o_i}, f_{d_i}, \Gamma_{d_i}, x_{0_{d_i}})$ que satisfaça as condições necessárias e suficientes para o diagnostico de falhas através de um diagnosticador parcial, em que E_{o_i} é o conjunto de eventos observáveis de G_{d_i}. □

Para que as definições que ainda serão apresentadas possam ser mais facilmente compreendidas, serão definidos dois tipos de eventos: *eventos essenciais* e *eventos redundantes*. Os primeiros são eventos que, sem os quais, o diagnóstico de falhas não será possível, e os últimos são eventos cujas ausências não acarretam em perda de diagnosticabilidade. Ambas as classificações são relativas a uma certa base de eventos, podendo um evento ser essencial para uma base e redundante para outra. Abaixo seguem as definições formais.

Definição 11 *(Eventos essenciais e redundantes)* Um evento $e \in E_{o_i}$, em que E_{o_i} é uma base para o diagnóstico de falhas, é um evento redundante (essencial) de E_{o_i} se $E_{o_i} \backslash \{e\}$ for (não for) uma base para o diagnóstico de falhas. □

As bases para o diagnóstico de falhas que possuem o menor número possível de eventos são chamadas de *bases mínimas para o diagnóstico de falhas*. O objetivo final deste capítulo é apresentar um método desenvolvido com a finalidade de se encontrar, sistematicamente, todas as bases mínimas para o diagnóstico de falhas em um sistema modelado

através de um autômato. A definição formal de bases mínimas para o diagnóstico de falhas é a seguinte.

Definição 12 *(Bases mínimas para o diagnóstico de falhas)* O conjunto $E_{o_i} \subseteq E_o$ é uma base mínima para o diagnóstico de falhas em G se E_{o_i} for uma base para o diagnóstico de falhas e possuir somente eventos essenciais. □

No capítulo anterior, foram enunciadas condições necessárias e suficientes para que a linguagem $\mathcal{L}(G)$, gerada pelo autômato $G = \{X, E_o \cup E_{uo}, f, \Gamma, x_0, X_m\}$, seja diagnosticável com respeito a $P_o : E^* \rightarrow E_o^*$ e $e_d \in E_{uo}$, através do diagnosticador centralizado $G_{diag} = (X_d, E_o, f_d, \Gamma_d, x_{0_d})$. A partir da informação de que o conjunto de eventos observáveis E_o é uma base para o diagnóstico de falhas, o interesse se volta para o problema de se descobrir se um conjunto $E_{o_i} \subset E_o$ é também uma base para o diagnóstico de falhas. Esse problema é equivalente a encontrar condições necessárias e suficientes para que a mesma linguagem $\mathcal{L}(G)$ seja diagnosticável, com respeito a $P_{o_i} : E^* \rightarrow E_{o_i}^*$ e $e_d \in E_{uo}$, através do diagnosticador parcial $G_{d_i} = (X_{d_i}, E_{o_i}, f_{d_i}, \Gamma_{d_i}, x_{0_{d_i}})$. Para que se possa apresentar o teorema que define tais condições, é necessário definir o autômato

$$G_{teste_i} = G_{diag} \| G_{d_i},$$

cujo conjunto de estados $X_{teste_i} \subseteq X_d \times X_{d_i}$, e portanto é formado por estados do tipo $x_k = (x_{k_1}, x_{k_2})$, em que $x_{k_1} \in X_d$ e $x_{k_2} \in X_{d_i}$. É necessário também que se definam *estados incertos* e *ciclos indeterminados de* G_{teste_i}.

Definição 13 *(Estados incertos de G_{teste_i})* Um estado $x_k = (x_{k_1}, x_{k_2}) \in X_{teste_i}$ é um estado incerto se x_{k_1} for um estado certo de G_{diag} e x_{k_2} for um estado incerto de G_{d_i}. □

A presença de um estado incerto em G_{teste_i} denota que o evento definitivo que leva o diagnosticador centralizado de um estado incerto a um estado certo é um evento não-observável para o diagnosticador parcial.

Logo, após a ocorrência deste evento, a componente do estado atual de G_{teste_i}, referente ao diagnosticador centralizado, assumirá um estado certo, enquanto que a componente associada ao diagnosticador parcial permanecerá em um estado incerto por não ter observado o evento definitivo.

Definição 14 *(Ciclos indeterminados de G_{teste_i})* Um ciclo indeterminado de G_{teste_i} é definido como sendo um ciclo de eventos observáveis pertencentes a E_o no qual todos os estados envolvidos são estados incertos, e que está associado a um ciclo indeterminado em G_{d_i} ou a um ciclo formado por estados certos em G_{diag} e que se tornou invisível (não existe) em G_{d_i}.

Um ciclo indeterminado de G_{teste_i} existirá: (i) se a não observação de um evento acarretar na impossibilidade de G_{d_i} distinguir entre, no mínimo, duas seqüências de eventos pertencentes a $\mathcal{L}(G)$, sendo que uma possui a falha e a outra não, caracterizando a presença de um ciclo indeterminado em G_{d_i}; ou (ii) se a não observação de eventos definitivos ocultar ciclos em estados certos de G_{diag}, podendo levar à falsa impressão de que a ocorrência da falha pode ser apontada em G_{d_i}, sendo que, na realidade, no caso de ocorrência da falha, o dignosticador parcial poderá ficar indefinidamente em um estado incerto, não indicando a ocorrência da falha.

Após os conceitos e definições necessários terem sido apresentados, será apresentado agora o teorema que define condições necessárias e suficientes para a diagnosticabilidade de um sistema parcialmente observado.

Teorema 2 Dado um autômato $G = (X, E_o \dot{\cup} E_{uo}, f, \Gamma, x_0, X_m)$ e sendo sua linguagem gerada $\mathcal{L}(G)$ diagnosticável com respeito a $P_o : E^* \rightarrow E_o^*$ e $e_d \in E_{uo}$, então um subconjunto $E_{o_i} \subset E_o$ é uma base para o diagnóstico de falhas se e somente se não existirem ciclos indeterminados em $G_{teste_i} = G_{diag} \| G_{d_i}$, em que G_{diag} é o diagnosticador centralizado e G_{d_i} é o diagnosticador parcial que possui como conjunto de eventos o conjunto $E_{o_i} \subseteq E_o$. □

A prova do teorema 2 não será apresentada formalmente neste trabalho, visto que já foi mostrada em [9]. Entretanto, a mesma pode ser entendida da seguinte forma: sendo $\mathcal{L}(G)$ diagnosticável com respeito a P_o e e_d, e tendo em vista que os eventos de $E_o \backslash E_{o_i}$ se tornaram não-observáveis, então as duas únicas possibilidades para que $\mathcal{L}(G)$ não seja diagnosticável com respeito a P_{o_i} e e_d, para $E_{o_i} \subset E_o$, são: (i) o aparecimento de ciclos indeterminados em G_{d_i} e/ou; (ii) a perda de observabilidade de eventos que

levam G_{diag} de um estado incerto para um estado certo formando ciclos não-observáveis. Para o primeiro caso, não é necessário calcular G_{teste_i}, bastando verificar se G_{d_i} possui algum ciclo indeterminado. Para o segundo caso, como o ciclo em questão se tornou não-observável (escondido), quando da ocorrência de uma falha, tem-se que o diagnosticador parcial não atingirá o estado certo correspondente, permanecendo em um estado incerto, e por conseguinte, não apontando a ocorrência da falha. Como $\mathcal{L}(G_{diag} \| G_{d_i}) = \mathcal{L}(G_{diag})$ pelo fato de que $E_{o_i} \subset E_o$ e $\mathcal{L}(G_{d_i}) = P_{o_i}[\mathcal{L}(G_{diag})]$, em que $P_{o_i} : E_o^* \rightarrow E_{o_i}^*$, então esses ciclos, que se tornaram invisíveis para G_{d_i}, aparecerão em G_{teste_i}, definindo, portanto, ciclos indeterminados.

Observação 3 De acordo com o exemplo 10, tem-se que ao se construir um diagnosticador parcial, alguns ciclos de eventos que eram observáveis podem se tornar não-observáveis, fazendo com que o diagnosticador parcial permaneça em um estado incerto sem se atualizar. Este fato faz com que a ocorrência da falha possa se tornar impossível de ser informada. Além disso, o ciclo que se tornou não-observável simplesmente desaparece do diagnosticador parcial, deixando a entender que, ao se utilizar a condição de que não exista ciclos indeterminados, a linguagem permanecerá diagnosticável com relação a P_{o_i} e e_d, quando esta, de fato, não o é. Logo, devem-se encontrar condições necessárias e suficientes para que uma linguagem parcialmente observada seja diagnosticável com relação a P_{o_i} e e_d. Essas condições devem envolver tanto o caso do aparecimento de ciclos indeterminados quanto o caso da não observação de ciclos de eventos que eram observáveis, que são identificados através da composição paralela entre G_{diag} e G_{d_i}. □

Uma vez verificado que $\mathcal{L}(G)$ é diagnosticável com respeito a P_o e e_d, pode-se descobrir, através do teorema 2, se um dado subconjunto $E_{o_i} \subset E_o$ é uma base para o diagnóstico de falhas do sistema modelado pelo autômato G, seguindo os passos descritos abaixo:

1. Calcula-se o diagnosticador parcial G_{d_i} como descrito na seção 3.1.1;

2. Verifica-se a existência de ciclos indeterminados no diagnosticador parcial G_{d_i}, calculado no passo anterior. Caso a resposta seja afirmativa, o subconjunto E_{o_i} não é

uma base mínima para o diagnóstico de falhas do sistema. Caso contrário, passar para o passo seguinte;

3. Calcula-se o autômato $G_{teste_i} = G_{diag} \| G_{d_i}$. Se houver ciclos constituídos por estados incertos em G_{teste} (ciclos indeterminados), então o conjunto E_{o_i} não é uma base mínima para o diagnóstico de falhas do sistema, pois estes ciclos são formados por eventos que se tornaram não-observáveis em G_{d_i} (por isso não foram identificados no passo anterior) e ocorrem após a falha, impedindo que o diagnosticador parcial se atualize e alcance um estado certo, violando assim, a condição de diagnosticabilidade, i.e., que não pode haver uma seqüência, arbitrariamente longa, formada somente por eventos não-observáveis após a ocorrência da falha. Caso contrário, o subconjunto E_{o_i} é uma base para o diagnóstico de falhas do sistema.

O teste realizado através da composição paralela, descrito no passo 3 do procedimento acima, é necessário pois há casos em que o diagnosticador G_{d_i} não apresenta nenhum ciclo indeterminado, e portanto, aparentemente, o conjunto E_{o_i} seria uma base para o diagnóstico de falhas do sistema. Entretanto, com a perda de observabilidade de alguns eventos pertencentes ao conjunto de eventos observáveis, o sufixo de certas seqüências se tornam *invisíveis* para o diagnosticador parcial G_{d_i}. Logo, se a falha ocorrer, o diagnosticador parcial poderá não ser capaz de apontar sua ocorrência, devido à falta de informação para a atualização de seus estados. Com a composição paralela, esses sufixos estarão presentes em G_{teste_i}, levando-o a um ciclo constituído por estados em que a primeira componente é um estado certo de G_{diag}, e a segunda componente é um estado incerto de G_{d_i}, definidos como estados incertos de G_{teste_i}. Se nenhum estado pertencente a um ciclo de estados incertos em G_{teste_i} possuir uma transição para um estado em que ambas as componentes sejam estados certos de G_{diag} e G_{d_i}, então este ciclo está associado às seqüências do tipo $s_y, s_n \in \mathcal{L}(G)$, arbitrariamente longas, em que $e_d \in s_y$, $e_d \notin s_n$ e $P_{o_i}(s_y) = P_{o_i}(s_n)$, já que, por hipótese, $\mathcal{L}(G)$ é diagnosticável em relação a P_o e e_d. Nesse caso, o conjunto E_{o_i} não será uma base para o diagnóstico de falhas do sistema, como pode ser comprovado pelo passo 3.

O seguinte exemplo ilustra a aplicação do procedimento, descrito acima, para verificar se um subconjunto E_{o_i} é uma base para o diagnóstico de falhas de um sistema.

Exemplo 11 *(Bases para o diagnóstico de falhas)* Nesse exemplo, será averiguado se dois conjuntos $E_{o_1} = \{a, c, d\}$ e $E_{o_2} = \{a, b, d\}$ são bases para o diagnóstico de falhas do sistema apresentado no exemplo 8, cujo diagnosticador centralizado está mostrado na figura 2.10. O conjunto de eventos desse sistema é $E_o = \{a, b, c, d\}$. O diagnosticador parcial G_{d_1}, que utiliza o conjunto E_{o_1} como conjunto de eventos observáveis de G, já foi calculado no exemplo 10, e está mostrado na figura 3.1. Por inspeção, pode-se concluir que tal diagnosticador não possui ciclos indeterminados. Com isso, os passos 1 e 2 do procedimento já estão concluídos. O terceiro passo é calcular o autômato $G_{teste_1} = G_{diag} \| G_{d_1}$, que está mostrado na figura 3.2, e verificar se há ciclos constituídos por estados incertos de G_{teste_1}. Pela observação deste, pode-se concluir que não há tais ciclos, e portanto, o conjunto E_{o_1} é uma base para o diagnóstico de falhas em G. Deseja-se agora verificar se o conjunto E_{o_2} também é uma base para o diagnóstico de falhas. O processo realizado anteriormente deve ser repetido, iniciando-se pelo cálculo do diagnosticador parcial G_{d_2}, que está mostrado na figura 3.3. Pode-se notar a presença de um ciclo no estado $\{4N, 10Y\}$ formado pelo evento b. Este ciclo está associado às seqüências $s_{n_1} = adb\{b\}^*$, $s_{y_1} = ae_d dcb\{b\}^*\{c\}^*$, $s_{n_2} = b\sigma db\{b\}^*$ e $s_{y_2} = b\sigma e_d dcb\{b\}^*\{c\}^*$, em que $P_{o_2}(s_{n_1}) = P_{o_2}(s_{y_1})$ e $P_{o_2}(s_{n_2}) = P_{o_2}(s_{y_2})$. Logo, o ciclo citado é um ciclo indeterminado, e o conjunto E_{o_2} não é uma base para o diagnóstico de falhas em G. Embora a não diagnosticabilidade do sistema já tenha sido determinada, para que se possa tirar o máximo proveito desse exemplo, o autômato $G_{teste_2} = G_{diag} \| G_{d_2}$ será também considerado (figura 3.4). Pode-se notar a existência de dois ciclos indeterminados, um no estado $\{10Y; 4N, 10Y\}$ e o segundo no estado $\{10Y; 4N, 7Y, 10Y\}$. O primeiro foi reconhecido através da análise de G_{d_2}, mas o segundo não pôde ser identificado diretamente a partir de G_{d_2}, pois o evento que o compõe é um evento não-observável. Portanto, ao se calcular o diagnosticador parcial para o conjunto E_{o_2}, este ciclo, que é formado pelo evento não-observável c, se tornou invisível. Utilizando-se a composição paralela, este ciclo volta a ser observável e, pelo fato de o diagnosticador G_{d_2} não alcançar um estado certo antes da ocorrência deste ciclo

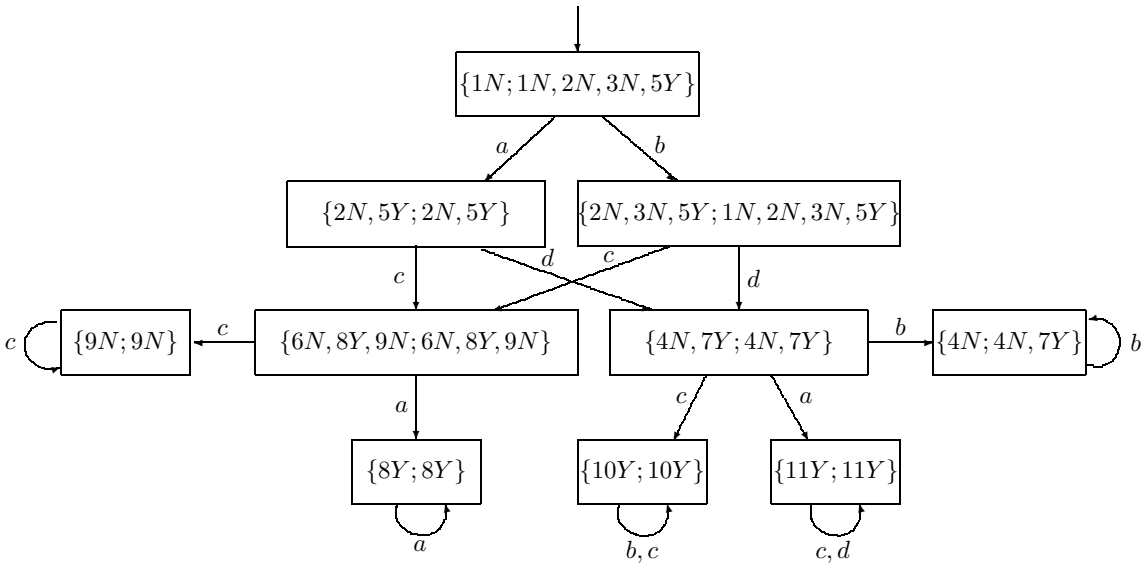

Figura 3.2: Autômato $G_{teste_1} = G_{diag} \| G_{d_1}$.

pode-se concluir que sua presença viola a condição de diagnosticabilidade. Tais ciclos são chamados de *ciclos escondidos indeterminados*. Pode-se observar, tanto em G_{teste_1} quanto em G_{teste_2}, que existem ciclos escondidos associados a estados de G_{diag} que indicam que a falha não ocorreu, como o ciclo presente no estado $\{4N; 4N, 7Y\}$ de G_{teste_1}. Como estes ciclos estão em seqüências de G que não contêm e_d, não há violação nas condições de diagnosticabilidade, uma vez que se a falha realmente ocorrer, o diagnosticador parcial assumirá um estado certo. Portanto, os ciclos que satisfazem essas condições são ciclos escondidos, porém não indeterminados. □

A presença de ciclos escondidos pode ser detectada na construção do diagnosticador parcial, comparando-se este último com o diagnosticador centralizado e observando se todos os ciclos pertencentes ao diagnosticador centralizado possuem um ciclo associado presente no diagnosticador parcial. Do contrário, o ciclo presente no diagnosticador centralizado que não possui o seu correspondente no diagnosticador parcial é um ciclo escondido neste último. Além disso, se o estado do diagnosticador parcial em que este ciclo se encontra for um estado incerto então ele será um ciclo escondido indeterminado. Apesar dos ciclos escondidos não terem sido identificados nos diagnosticadores parciais do exemplo 11, para efeito didático, ambos os ciclos escondidos determinados e indeter-

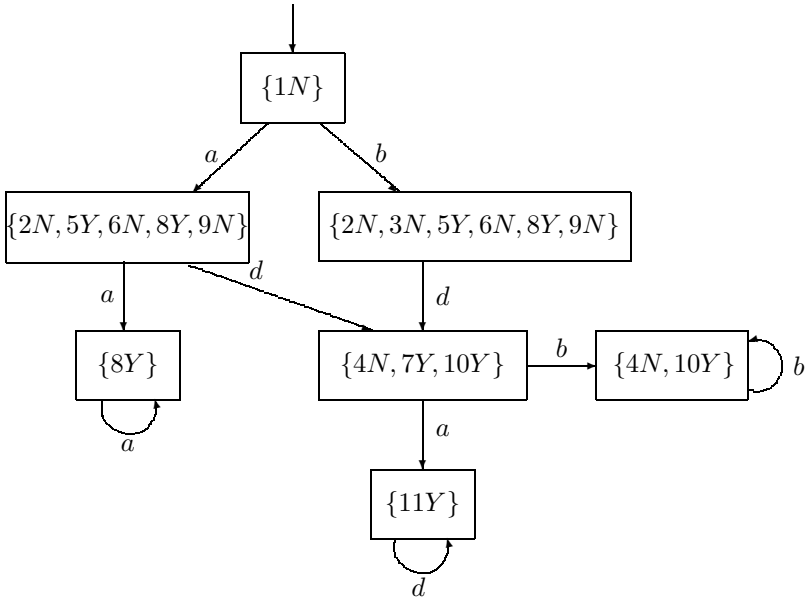

Figura 3.3: Diagnosticador parcial G_{d_2}, calculado considerando-se $E_{o_2} = \{a, b, d\}$ como conjunto de eventos observáveis do sistema.

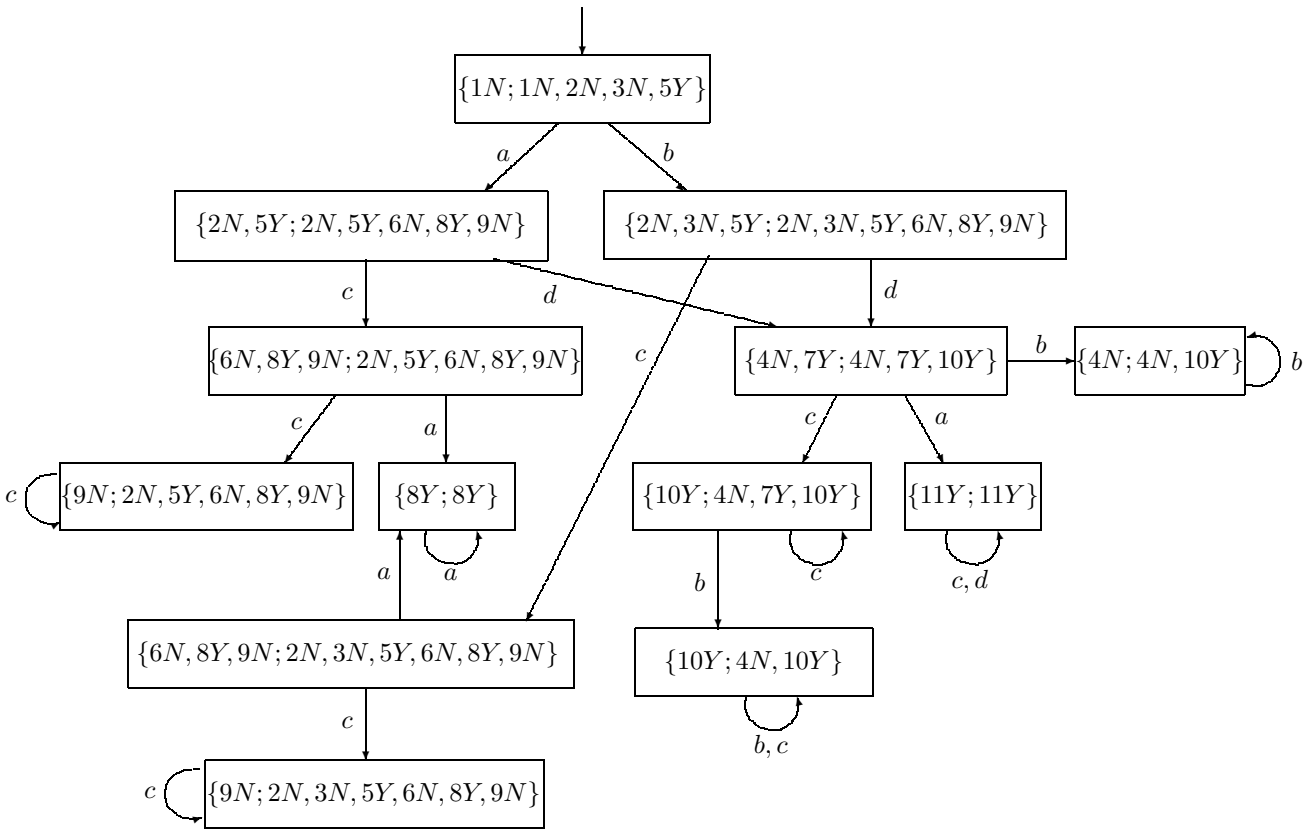

Figura 3.4: Autômato $G_{teste_2} = G_{diag} \| G_{d_2}$.

minados devem ser identificados através de arcos tracejados, nos respectivos estados do diagnosticador parcial pelos quais foram agrupados. Este artifício será utilizado daqui para frente a fim de se verificar a diagnosticabilidade de uma linguagem utilizando-se somente o diagnosticador parcial, sem a necessidade de se construir o autômato G_{teste_i}. Esse ponto será ilustrado em um exemplo na próxima seção.

Através do exemplo 11, pode-se observar que mesmo em um sistema com poucos eventos, a quantidade de combinações de eventos observáveis a serem testadas pode ser grande, mais precisamente $2^n - 2$, em que n é a cardinalidade (número de elementos) de E_o. Além disso, o autômato G_{teste_i} possui, no mínimo, o mesmo número de estados de G_{diag}, podendo chegar a um número elevado destes. Assim, torna-se necessário encontrar uma maneira de se testar o menor número possível de combinações de eventos observáveis, e de forma mais objetiva. Um método sistemático de escolha dos subconjuntos de eventos a serem testados e de determinação das bases mínimas para o diagnóstico de falhas será proposto na seção seguinte.

3.2 Método sistemático para identificação de bases mínimas para o diagnóstico de falhas em SED

A partir dos resultados obtidos neste capítulo, pode-se concluir que qualquer método que seja proposto com a finalidade de se encontrar todas as bases mínimas para o diagnóstico de falhas de um sistema deve ser capaz de identificar a presença de ciclos escondidos indeterminados e a presença de ciclos de eventos observáveis que se tornaram indeterminados após a perda da observabilidade de um ou mais eventos.

Para o desenvolvimento de um método para a busca das bases mínimas para o diagnóstico de falhas, a definição de *seqüências Y* e *seqüências N* é necessária.

Definição 15 *(Seqüências Y e seqüências N)* Suponha um diagnosticador centralizado $G_{diag} = (X_d, E_o, f_d, \Gamma_d, x_{0_d})$, em que x_{d_Y}, x_{d_N} e $x_{d_{YN}}$ são, respectivamente, estados certo, normal e incerto de G_{diag}.

Sejam,

$$s \in \mathcal{L}(G_{diag}) : f_d(x_{0_d}, s) = x_{d_{YN}}$$

e defina o conjunto E_Y da seguinte forma:

$$E_Y = e \in \Gamma(x_{d_{YN}}) : f_d(x_{d_{YN}}, e) = x_{d_Y}.$$

Sejam $L_d = \mathcal{L}(G_{diag})$ e $L_d/s = \{t \in E_o^* : st \in L_d\}$. Então as seqüências Y associadas a um estado incerto $x_{d_{YN}}$ e a um evento $e \in E_Y$ são definidas como

$$S_Y(x_{d_{YN}}, e) = \{e\}\{t \in L_d/se\}.$$

Então, o conjunto de todas as seqüências Y de $x_{d_{YN}}$ é definido por

$$L_Y(x_{d_{YN}}) = \bigcup_{e \in E_Y} S_Y(x_{d_{YN}}, e).$$

Da mesma forma, podem ser definidas as seqüências N (S_N) e o conjunto L_N substituindo-se x_{d_Y} e E_Y por x_{d_N} e E_N, respectivamente, nas definições acima. □

Pode-se definir também *trajetórias Y e N*.

Definição 16 *(Trajetória Y e trajetória N)* Trajetórias Y e N são seqüências de comprimento finito do diagnosticador centralizado que se iniciam em um estado incerto e terminam, respectivamente, em ciclos formados somente por estados certos ou somente por estados normais (estados que indicam que a falha não ocorreu). Além disso, o primeiro estado incerto deve também ser o único.

Como exemplo, considere o caso do diagnosticador centralizado da figura 2.10, que possui três trajetórias Y que se iniciam nos estados incertos $\{6N, 8Y, 9N\}$ e $\{4N, 7Y\}$ e cujos ciclos finais [1] são formados pelo estado $\{8Y\}$ para o primeiro estado incerto, e $\{10Y\}$ e $\{11Y\}$ para o segundo, e duas trajetórias N, que se iniciam nos mesmos estados incertos acima, e têm os ciclos finais compostos pelos estados $\{9N\}$ e $\{4N\}$, respectivamente. Pode-se mostrar que a causa do aparecimento de um ciclo indeterminado observável é a formação de estados incertos quando alguns dos eventos pertencentes às seqüências Y e N se tornam não-observáveis. O aparecimento de um ciclo escondido indeterminado possui duas causas: uma delas corresponde à mesma citada para o aparecimento de um ciclo

[1] Ciclos finais são ciclos formados somente por estados certos ou estados normais que existem em todo diagnosticador centralizado, tendo em vista que a linguagem gerada pelo autômato é, por hipótese, viva.

indeterminado observável, e a outra é o fato de todos os eventos de uma seqüência Y, que leva o diagnosticador de um estado incerto a algum ciclo de estados certos, e também os eventos que compõem este ciclo, tornarem-se não-observáveis. Essas condições podem ser observadas no exemplo 11, em que o conjunto de eventos observáveis E_{o_2} permitiu o aparecimento de um ciclo indeterminado observável e de um ciclo escondido indeterminado pela perda da observabilidade do evento c. Através do diagnosticador centralizado da figura 2.10, pode-se notar que existe uma seqüência cujo único evento é o evento c, que o leva do estado incerto $\{4N, 7Y\}$ ao estado certo $\{10Y\}$, e um ciclo formado pelo mesmo evento, neste estado. Portanto, com a perda da observabilidade deste evento, a condição para o aparecimento de um ciclo escondido é satisfeita. A condição para o aparecimento do ciclo indeterminado observável não pode ser vista pela simples análise do diagnosticador centralizado, mas sim pela análise do diagnosticador parcial correspondente ao conjunto E_{o_2}.

A primeira exigência que o método deve atender é a objetividade da busca. Considere um autômato $G = (X, E_o \dot{\cup} E_{uo}, f, \Gamma, x_0, X_m)$ cuja linguagem gerada $\mathcal{L}(G)$ seja diagnosticável com respeito a $P_o : E^* \rightarrow E_o^*$ e $e_d \in E_{uo}$. Portanto, para otimizar o processo de busca de bases mínimas para o diagnóstico de falhas, deve-se utilizar o seguinte resultado: se uma seqüência Y, que leva o diagnosticador centralizado, através da trajetória Y associada, do estado incerto, a algum ciclo de estados certos, incluindo os eventos do ciclo, se tornar não-observável em um diagnosticador parcial, então todos os estados certos, alcançados pela seqüência Y que se tornou não-observável, serão agrupados no estado incerto (origem da trajetória), levando ao aparecimento de um ciclo escondido indeterminado, uma vez que não será observado nenhum evento que leve este diagnosticador parcial de um estado incerto para um estado certo desta trajetória.

Com isso, sabendo-se que o conjunto E_o é uma base para o diagnóstico de falhas do sistema, então, para que seja possível encontrar, de forma objetiva, todas as bases mínimas desse sistema, os subconjuntos iniciais de eventos a serem testados devem ser formados pelas combinações dos eventos definitivos das trajetórias Y, cada uma dessas combinações contendo somente um evento definitivo de cada trajetória Y. Entretanto, em

alguns casos, a identificação dessas trajetórias pode não se dar de forma direta como no exemplo da figura 2.10, feito no início desta seção. A dificuldade reside na possibilidade de existirem ciclos finais que possuam mais de um "caminho" de chegada através de estados incertos diferentes, conforme será visto adiante.

Após a aplicação de regras a serem ainda apresentadas, se for constatado que uma das combinações de eventos definitivos não é uma base mínima para o diagnóstico de falhas, deverão ser acrescentados a esta combinação, eventos pertencentes ao conjunto $E_o \backslash E_{o_i}$, com a finalidade de torná-la uma base mínima, fazendo com que os ciclos indeterminados observáveis e/ou os ciclos escondidos indeterminados deixem de existir. A inserção desses eventos deve também ser feita de maneira sistemática, através de critérios a serem pré-estabelecidos.

As medidas apresentadas nos parágrafos anteriores levam a uma diminuição considerável na quantidade de operações e verificações em diagosticadores. Para que seja possível a aplicação da otimização do diagnóstico de falhas em um SED de grande porte, deve-se lançar mão de ferramentas computacionais para a realização desta tarefa. Portanto, torna-se conveniente a utilização de medidas que diminuam o tempo de processamento dos cálculos necessários na identificação de todas as bases mínimas de um sistema. A seguir, será apresentado um algoritmo para encontrar os conjuntos de eventos definitivos de cada trajetória Y do diagnosticador centralizado, que serão utilizados em um método sistemático de identificação de bases mínimas para o diagnóstico de falhas em SED, que será proposto nesta seção.

Algoritmo 1 *(Determinação dos conjuntos de eventos definitivos)*

1. Identificar todos os estados incertos de G_{diag} que possuem uma transição para um estado certo.

2. Para cada estado incerto identificado, construir uma árvore da seguinte forma: os ramos que saem deste estado são associados aos seus eventos ativos, e chegam a novos nós, associados aos estados certos que são alcançados a partir deste estado. Continuar esse processo até que um novo nó esteja associado a um estado que já

tenha sido visitado. Quando isto ocorrer, interromper o processo. Quando todas as possibilidades de novos ramos estiverem extintas, a árvore estará completa.

3. Obter as trajetórias Y do diagnosticador e enumerá-las da seguinte forma: $T_{Y_1}, T_{Y_2}, \ldots, T_{Y_m}$.

4. Compor os conjuntos de eventos definitivos E_{d_i} associado à trajetória T_{Y_i} com cada evento pertencente à trajetória T_{Y_i}.

Exemplo 12 *(Aplicação do algoritmo 1)* Considere o autômato da figura 3.5 e seu diagnosticador centralizado mostrado na figura 3.6. Este diagnosticador não possui ramos bem definidos como o diagnosticador da figura 2.10, onde foram identificadas as trajetórias Y sem a necessidade da construção das árvores definidas pelo algoritmo 1. No problema considerado neste exemplo, as árvores devem ser construídas a fim de serem identificados todas as trajetórias Y e os eventos que realizam as transições entre seus estados. Os estados incertos $\{1N, 2Y\}$, $\{3Y, 5N\}$ e $\{4Y, 5N\}$ possuem transição para estados certos. E, portanto, deverá ser construída uma árvore para cada um destes estados. Estas árvores estão mostradas nas figuras 3.7, 3.8 e 3.9. Para cada nó final existe uma trajetória única associada, formada por uma seqüência que leva a árvore do nó inicial ao nó final. Ao todo existem doze trajetórias, três da primeira árvore, três da segunda árvore e três da terceira árvore. Pode-se então compor os conjuntos de eventos definitivos de cada trajetória identificada. Neste exemplo, os conjuntos são: $E_{d_1} = \{d, e\}$, $E_{d_2} = \{b, c, e\}$, $E_{d_3} = \{a, b, c, e\}$, $E_{d_4} = \{a, b\}$, $E_{d_5} = \{a, b, c\}$, $E_{d_6} = \{a, b, d\}$, $E_{d_7} = \{b, c\}$, $E_{d_8} = \{b, c, d\}$, $E_{d_9} = \{a, b, c\}$, $E_{d_{10}} = \{d\}$, $E_{d_{11}} = \{b, c, d\}$ e $E_{d_{12}} = \{a, b, c, d\}$. □

A partir deste algoritmo, é possível gerar os conjuntos iniciais de eventos observáveis que serão candidatos a bases mínimas para o diagnóstico de falhas de um SED, tendo em vista que se todos os eventos de um conjunto definitivo se tornarem não-observáveis, o diagnosticador parcial que não registrar a ocorrência desses eventos apresentará um ciclo escondido indeterminado. Utilizando-se este resultado e as definições de diagnosticabilidade parcial apresentadas neste capítulo, será apresentado, a seguir, um método capaz de encontrar todas as bases mínimas para o diagnóstico de falhas em SED. Para tanto,

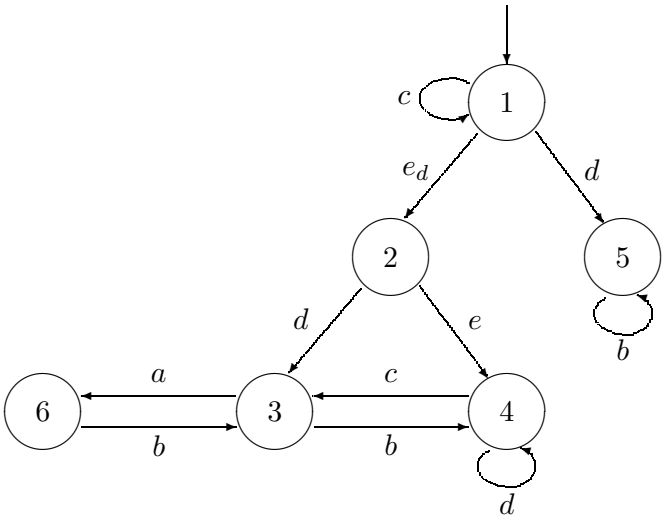

Figura 3.5: Autômato do exemplo 12.

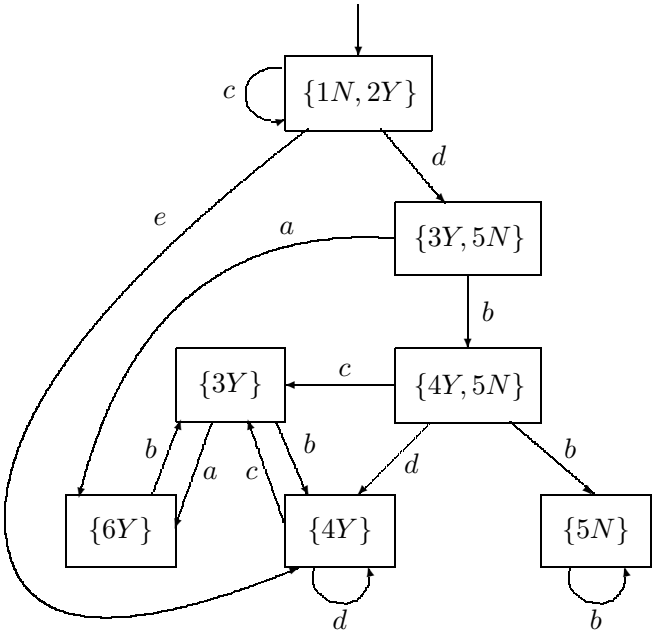

Figura 3.6: Diagnosticador centralizado do autômato do exemplo 12.

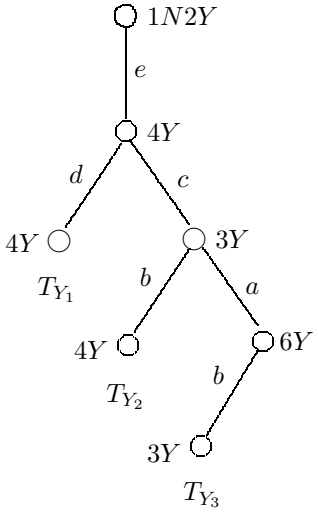

Figura 3.7: Árvore criada a partir do estado {1N,2Y}.

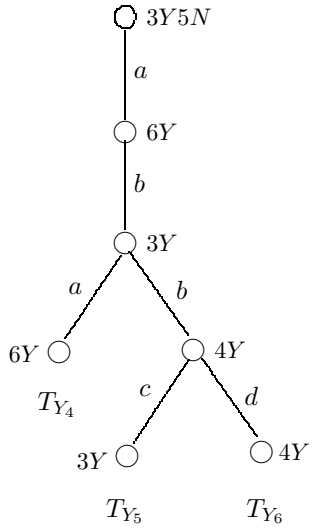

Figura 3.8: Árvore criada a partir do estado {3Y,5N}.

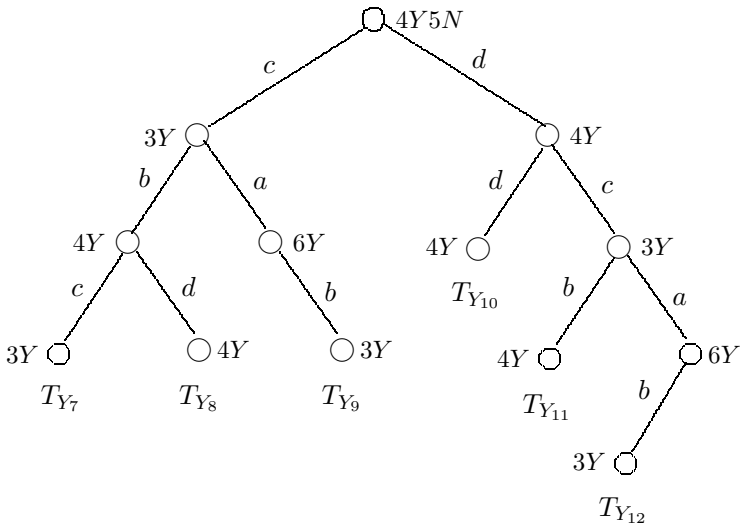

Figura 3.9: Árvore criada a partir do estado {4Y,5N}.

considere um autômato $G = (X, E_o \dot{\cup} E_{uo}, f, \Gamma, x_0, X_m)$ cuja linguagem gerada $\mathcal{L}(G) = L$ seja diagnosticável com respeito a $P_o : E^* \rightarrow E_o^*$ e $e_d \in E_{uo}$; ou equivalentemente, seu diagnosticador centralizado $G_{diag} = (X_d, E_o, f_d, \Gamma_d, x_{0_d})$ não possui ciclos indeterminados. Defina

$$E_{d_1} \dot{\times} E_{d_2} \dot{\times} \ldots \dot{\times} E_{d_n} = \{\tilde{E}_o = E_{e_1} \cup E_{e_2} \cup \ldots \cup E_{e_n} : (E_{e_1}, E_{e_2}, \ldots, E_{e_n}) \in 2_1^{E_{d_1}} \times 2_1^{E_{d_2}} \times$$

$$\times \ldots \times 2_1^{E_{d_n}}\}$$

sendo $2_1^B = \{\Sigma \in 2^B : |\Sigma| = 1\}$ e $|\Sigma|$ a cardinalidade. O conjunto de todas as bases mínimas para o diagnóstico de falhas em G pode ser obtido de acordo com o seguinte algoritmo.

Algoritmo 2 *(Determinação das bases mínimas para o diagnóstico de falhas em SED)*

Passo 1: Obtenha todos os conjuntos de eventos definitivos E_{d_i}, $i = 1, \ldots, n$, de acordo com o algoritmo 1, em que n é o número de trajetórias Y, e faça $E_{bmd} = \emptyset$.

Passo 2: Componha, da seguinte forma, os conjuntos iniciais de eventos observáveis E_{o_i}:

1. Calcule $\tilde{E}_{cbd} = E_{d_1} \dot{\times} E_{d_2} \dot{\times} \ldots \dot{\times} E_{d_n}$.
2. Calcule $\bar{E}_{cbd} = \{\bar{E} \in \tilde{E}_{cbd} : (\exists \tilde{E} \in \tilde{E}_{cbd})[\tilde{E} \subseteq \bar{E}]\}$.

3. Calcule $E_{cbd} = \tilde{E}_{cbd} \backslash \bar{E}_{cbd} = \{E_{o_1}, E_{o_2}, \ldots, E_{o_i}\}$.

Passo 3: Seja E_{o_1} o elemento de E_{cbd} com menor cardinalidade. Verifique se L é diagnosticável com relação a $P_{o_1} : E^* \rightarrow E^*_{o_1}$ e $e_d \in E_{uo}$.

Passo 4: Se L for diagnosticável com relação a P_{o_1} e $e_d \in E_{uo}$, então:

A.1 Faça $E_{bmd} \leftarrow E_{bmd} \cup \{E_{o_1}\}$.

A.2 Faça $E_{cbd} \leftarrow E_{cbd} \backslash \{E_{o_1}\}$.

A.3 Se existir $E_{o_j} \in E_{cbd}$ tal que $E_{o_i} \subseteq E_{o_j}$, então faça $E_{cbd} \leftarrow E_{cbd} \backslash \{E_{o_j}\}$.

Caso contrário:

B.1 Faça $E_{cbd} \leftarrow E_{cbd} \backslash \{E_{o_1}\}$.

B.2 Calcule o autômato $G_{teste} = G_{diag} \| G_{d_1}$ e proceda da seguinte forma:

(a) Encontre todos os estados (ou grupos de estados) incertos de $G_{teste} = G_{diag} \| G_{d_i}$ que formem ciclos indeterminados, e faça $j = 0$.

(b) Identifique também todos os estados de G_{teste} em que a primeira componente é N e as segundas componentes sejam iguais às dos estados incertos determinados em (a) e formem ciclos.

(c) Para cada estado incerto (ou grupo de estados incertos, no caso de um ciclo indeterminado observável) obtido na etapa (a), identifique dentre todos os estados obtidos em (a) e (b) aqueles cujas segundas componentes forem iguais às do estado incerto considerado.

(d) Para cada estado (ou grupo de estados) identificado em (c), construa uma árvore iniciando-se pelo estado citado (ou pelo último estado alcançado, no caso de um grupo de estados) e termine no estado inicial de G_{teste}.

(e) Registre todos os eventos presentes na árvore em um conjunto E_{Y_i}, para $i = 1 \ldots m$, em que m é o número de estados em que a primeira componente é um estado certo. No caso da primeira componente deste estado ser um estado normal, guarde os eventos em um conjunto E_{N_k}, para $k = 1 \ldots n$,

em que n é o número de estados em que a primeira componente é um estado normal.

(f) Faça:

(i) $E'_{Y_i} = E_{Y_i} \backslash E_{o_1}$, $i = 1 \ldots m$.

(ii) $E'_{N_k} = E_{N_k} \backslash E_{o_1}$, $k = 1 \ldots n$.

(iii) $E_{T_Y} = E'_{Y_1} \dot{\times} E'_{Y_2} \dot{\times} \ldots \dot{\times} E'_{Y_m}$.

(iv) $E''_{N_k} = E'_{N_k} \backslash \{E'_{Y_1} \cup E'_{Y_2} \cup \ldots \cup E'_{Y_m}\}$, $k = 1 \ldots n$.

(v) $E_{T_N} = E''_{N_1} \dot{\times} E''_{N_2} \dot{\times} \ldots \dot{\times} E''_{N_n}$

(vi) $E_T = E_{T_Y} \cup E_{T_N}$

(g) Faça:

1. $j = j + 1$
2. $E_{T_j} = E_T$

(h) Se ainda houver estados incertos com ciclos indeterminados, retorne à etapa (c).

B.3 Componha os novos subconjuntos de eventos observáveis E'_{o_1} da seguinte forma:

1. Calcule $E_{n_T} = E_{T_1} \dot{\times} E_{T_2} \dot{\times} \ldots \dot{\times} E_{T_j}$.
2. Calcule $\tilde{E}'_{o_1} = \{E = E_{o_1} \cup E_n : \ E_n \in E_{n_T}\}$.
3. Calcule $\bar{E}'_{o_1} = \{\bar{E} \in \tilde{E}'_{o_1} : (\exists \tilde{E} \in \tilde{E}'_{o_1})[\tilde{E} \subset \bar{E}]\}$.
4. $E'_{o_1} = \tilde{E}'_{o_1} \backslash \bar{E}'_{o_1}$

B.4 Faça $E_{cbd} \leftarrow E_{cbd} \cup (E'_{o_i} \backslash E_{bmd})$.

Passo 5: Se $|E_{cbd}| = 0$ então o processo está finalizado. Caso contrário, retorne ao passo 3. □

Os passos do algoritmo 2 podem ser justificados da seguinte forma.

1. O passo 2 é o responsável por criar um conjunto (E_{cbd}) contendo todos subconjuntos de eventos observáveis com menor cardinalidade possível, que podem ser uma base mínima, isto é, subconjuntos formados por apenas um evento definitivo de cada

trajetória Y identificada no algoritmo do passo 1. Além disso, não podem haver elementos em E_{cbd} que sejam subconjuntos de outros elementos em E_{cbd}. (itens 2 e 3 do passo 2).

2. No passo 3, o subconjunto de menor cardinalidade de E_{cbd} é selecionado e verifica-se se a linguagem gerada por G permanece diagnosticável supondo este subconjunto como sendo o conjunto de eventos observáveis de G.

3. Se o subconjunto testado for uma base para o diagnóstico de falhas, no passo 4, insere-se este subconjunto no conjunto E_{bmd} e retira-se este subconjunto do conjunto E_{cbd} (passos A.1 a A.3). Caso contrário, retira-se este subconjunto do conjunto E_{cbd} e calcula-se o autômato $G_{teste} = G_{diag} \| G_{d_1}$ com a finalidade de verificar quais seqüências Y e N possuem projeções iguais, tendo em vista que essas seqüências estarão associadas àquelas que levam o autômato G_{teste} a um ciclo indeterminado. Com essas seqüências identificadas, verificam-se quais os eventos pertencentes a elas, que adicionados ao conjunto que está sendo testado, poderão diferenciar as seqüências Y das seqüências N. Um ciclo escondido indeterminado aparece quando um ciclo de estados certos se agrupa em um estado incerto. Logo, somente inserindo-se eventos pertencentes à seqüência que leva G_{diag} ao ciclo de estados certos pode-se resolver esta ambigüidade. Entretanto, ciclos indeterminados observáveis envolvem tanto ciclos de estados certos quanto ciclos de estados normais de G_{diag}. Logo, pode-se resolver esta ambigüidade inserindo-se tanto eventos da seqüência que leva o diagnosticador ao ciclo de estados certos quanto eventos pertencentes às seqüências que levam o diagnosticador a ciclos de estados normais. Constroem-se subconjuntos contendo um evento de cada seqüência Y e subconjuntos contendo um evento de cada seqüência N identificadas pelas árvores construídas em B.2(c) e unem-se estes subconjuntos em E_T. Faz-se o mesmo para todos os ciclos indeterminados de G_{teste}, para compor os novos candidatos à base para o diagnóstico de falhas, inserindo-se um subconjunto de cada conjunto E_{T_j} ao conjunto E_{o_1} testado, em que j é o número de ciclos indeterminados do diagnosticador parcial. Inserem-se estes novos candidatos em E_{cbd}, excluindo o caso em que um novo candidato contiver uma base

mínima.

4. Quando E_{cbd} se tornar vazio, isto é, quando não houverem mais candidatos à base mínima, o algoritmo se encerra.

Observação 4 Dado que o autômato G gera a linguagem $\mathcal{L}(G)$ diagnosticável com respeito a $P_o : E^* \rightarrow E_o^*$ então o método deve encontrar pelo menos o conjunto E_o como uma base mínima para o diagnóstico de falhas; neste caso, os subconjuntos de E_o não são bases para o diagnóstico de falhas, e a segunda parte do passo 4 fará com que o método convirja para este resultado.

Exemplo 13 *(Procedimento para identificação das bases mínimas)* O diagnosticador centralizado mostrado no capítulo anterior e reproduzido na figura 3.10 será utilizado para ilustrar a aplicação do procedimento proposto para a identificação das bases mínimas para o diagnóstico de falhas.

Inicialmente, deve-se identificar os conjuntos de eventos definitivos de cada trajetória Y. Para isso, serão construídas árvores cujos nós iniciais serão os estados incertos que possuem ao menos uma transição para um estado certo. As árvores correspondentes aos estados incertos do diagnosticador mostrado na figura 3.10, estão representadas nas Figs. 3.11 e 3.12. Os conjuntos de eventos definitivos são, portanto, $E_{d_1} = \{a\}$, $E_{d_2} = \{b, c\}$, $E_{d_3} = \{c\}$, $E_{d_4} = \{a, d\}$ e $E_{d_5} = \{a, c\}$.

O conjunto de candidatos à base mínima é formado seguindo o passo 2 do algoritmo 2 da seguinte forma: 1. $\tilde{E}_{cbd} = \{\{a, b, c\}, \{a, c\}, \{a, b, c, d\}, \{a, c, d\}\}$; 2. $\bar{E}_{cbd} = \{\{a, b, c\}, \{a, b, c, d\}, \{a, c, d\}\}$ e 3. $E_{cbd} = \{a, c\}$. Portanto, somente o conjunto $E_{o_1} = \{a, c\}$ deve ser testado inicialmente.

De acordo com o passo 3, deve-se agora construir o diagnosticador parcial que possui os eventos a e c como observáveis. Este diagnosticador está mostrado na figura 3.13. Nota-se que há ciclos indeterminados observáveis nos estados $\{9N, 10Y\}$ e $\{9N, 10Y, 11Y\}$, o que determina a não-diagnosticabilidade da linguagem L com relação a $P_{o_1} : E^* \rightarrow E_{o_1}^*$, para $E_{o_1} = \{a, c\}$. Pode-se notar também a existência de ciclos escondidos em todos os estados que possuem como subconjuntos os estados $\{10Y\}$ e $\{11Y\}$, ciclos estes gerados

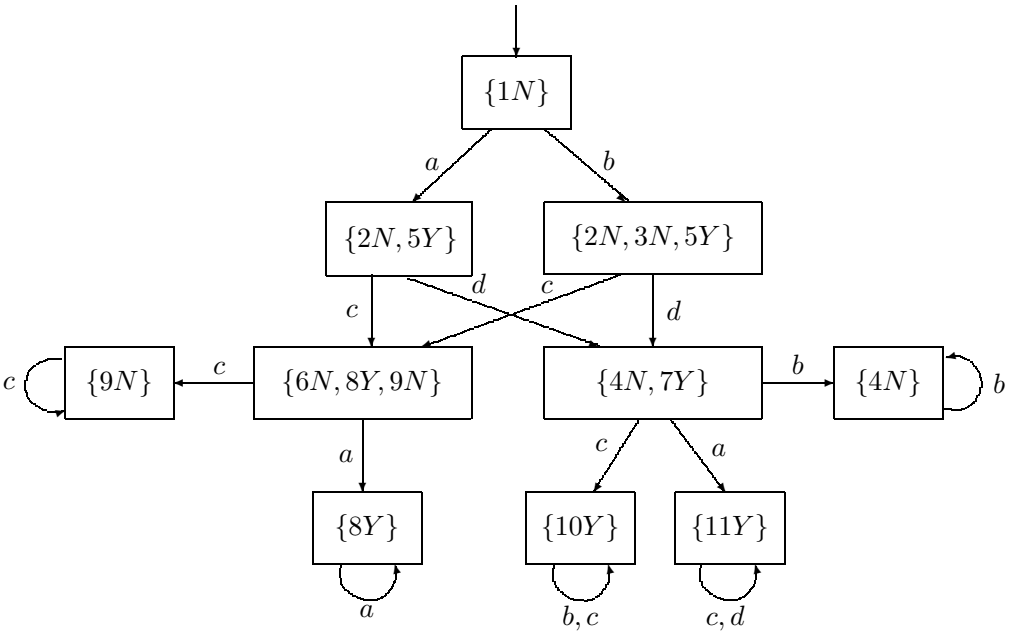

Figura 3.10: Diagnosticador centralizado do exemplo 13.

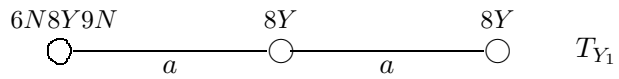

Figura 3.11: Árvore criada a partir do estado $\{6N, 8Y, 9N\}$.

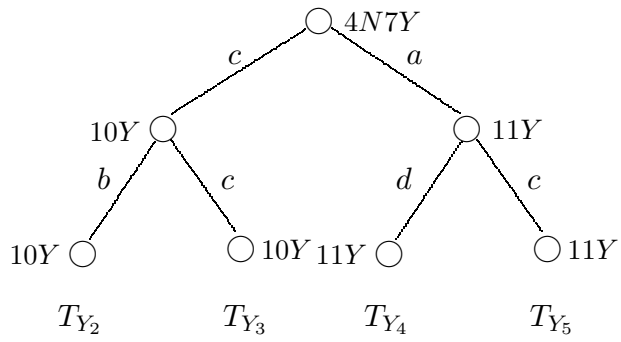

Figura 3.12: Árvore criada a partir do estado $\{4N, 7Y\}$.

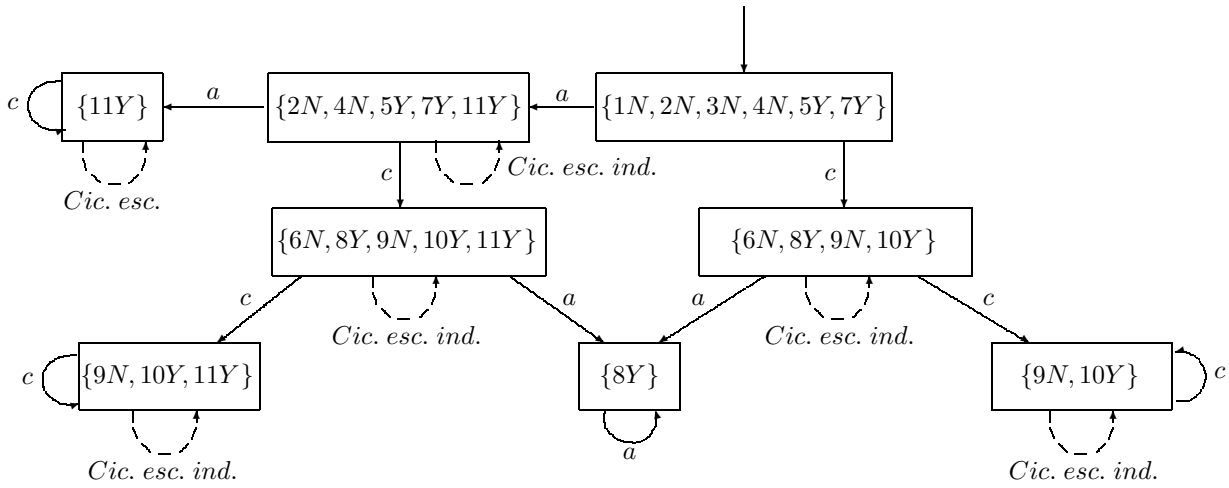

Figura 3.13: Diagnosticador parcial para o conjunto de eventos observáveis $\{a, c\}$.

pela não-observabilidade dos eventos b e d. Todos estes ciclos são indeterminados, exceto o ciclo presente no estado $\{11Y\}$, dado que este se encontra em um estado certo. Assim, de acordo com o passo B.2, retira-se E_{o_1} do conjunto de candidatos à base mínima para o diagnóstico, resultando em $E_{cbd} = \emptyset$.

De acordo com o passo B.2, deve-se, então, calcular $G_{teste} = G_{diag} \| G_{d_1}$, cujo diagrama de transição de estados está representado na figura 3.14, e verificar quais são as seqüências que geram ambigüidades no diagnóstico, para então incluir os eventos no conjunto E_{o_1} que permitam resolver essas ambigüidades e diagnosticar a falha. De acordo com o passo 4 (B.2(a)), devem-se identificar todos os estados incertos de G_{teste} que formam ciclos indeterminados, sendo eles: $x_1 = \{10Y; 6N, 8Y, 9N, 10Y, 11Y\}$, $x_2 = \{10Y; 9N, 10Y, 11Y\}$, $x_3 = \{10Y; 6N, 8Y, 9N, 10Y\}$, $x_4 = \{10Y; 9N, 10Y\}$, $x_5 = \{11Y; 2N, 4N, 5Y, 7Y, 11Y\}$, $x_6 = \{11Y; 6N, 8Y, 9N, 10Y, 11Y\}$, $x_7 = \{11Y; 9N, 10Y, 11Y\}$. Deve-se agora identificar, segundo o passo B.2(b), todos os estados cuja primeira componente é N e as segundas componentes sejam iguais às dos estados identificados em B.2(a) e que formem ciclos; são eles: $x_8 = \{9N; 9N, 10Y, 11Y\}$ e $x_9 = \{9N; 9N, 10Y\}$. De acordo com o passo B.2(c), para cada um dos estados obtidos em B.2(a), devem ser identificados entre todos os estados obtidos (desde x_1 até x_9) aqueles que possuem suas segundas componentes iguais. Iniciando-se pelo estado x_1, obtém-se: $x_{11} = x_1$ e $x_{12} = x_6$. A próxima etapa é construir uma árvore para cada um destes estados e identificar os eventos presentes nas

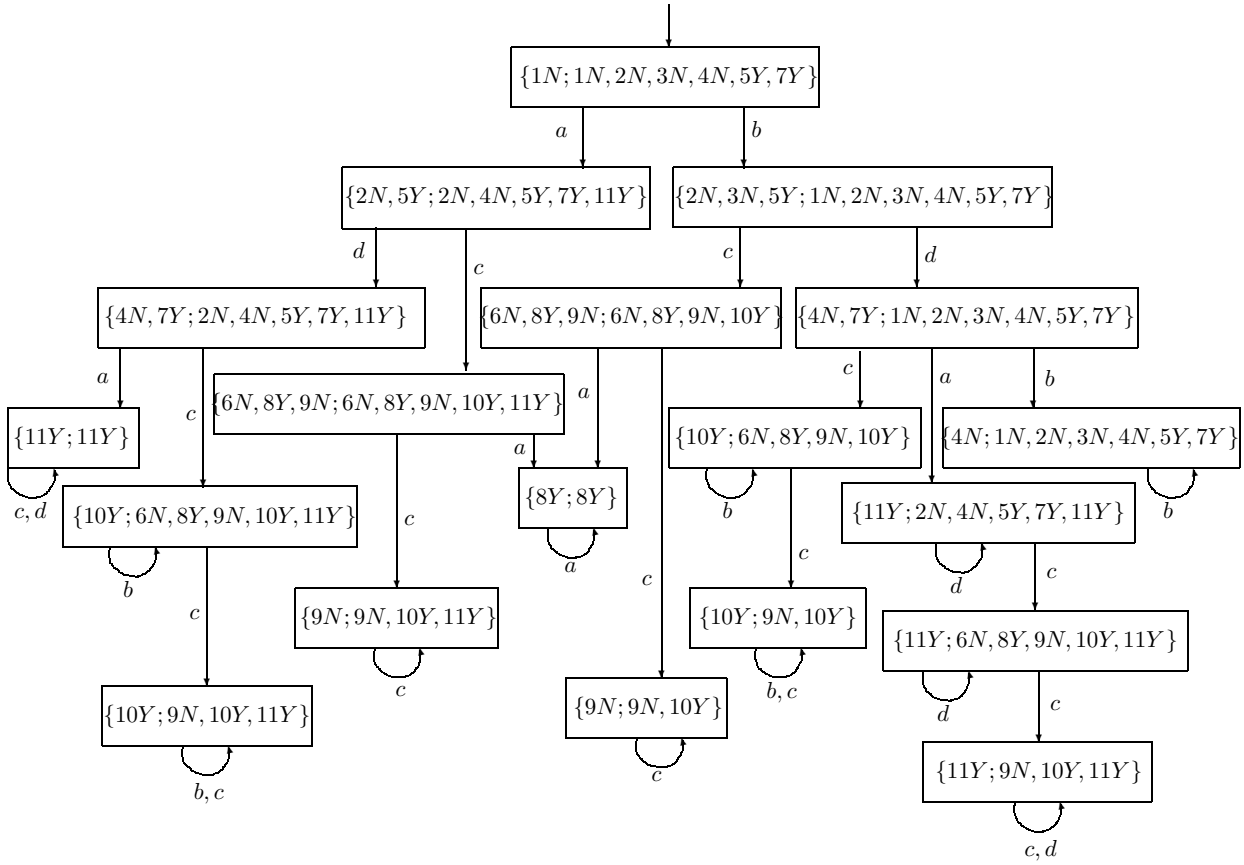

Figura 3.14: $G_{teste_1} = G_{diag} \| G_{d_1}$.

seqüências que levam ao estado inicial de G_{teste}. Entretanto, é possível identificá-los no seu próprio diagrama de transição de estados, sendo $E_{Y_1} = E_{Y_2} = \{a, b, c, d\}$. O próximo passo é calcular $E'_{Y_1} = E'_{Y_2} = E_{Y_1} \setminus \{a, c\} = \{b, d\}$ e

$$E_{T_Y} = E'_{Y_1} \dot{\times} E'_{Y_2} = \{\{b\}, \{d\}, \{b, d\}\}.$$

Portanto, $E_{T_1} = E_T = E_{T_Y}$. O próximo estado a ser estudado é $x_2 = \{10Y; 9N, 10Y, 11Y\}$. Os estados a ele associados são $x_{21} = x_2$, $x_{22} = x_7$ e $x_{23} = x_8$. Os conjuntos de eventos identificados são $E_{Y_1} = \{a, b, c, d\}$, $E_{Y_2} = \{a, c, d\}$, $E_{Y_3} = \{a, b, c, d\}$, $E_{Y_4} = \{a, b, c, d\}$ e $E_{N_1} = \{a, c\}$, respectivamente. Calculam-se

$$\begin{aligned} E'_{Y_1} = E'_{Y_3} = E'_{Y_4} &= E_{Y_1} \setminus \{a, c\} = \{b, d\} \\ E'_{N_1} &= \{a, c\} \setminus \{a, c\} = \emptyset \\ E'_{Y_2} &= E'_{Y_2} \setminus \{a, c\} = \{d\}. \end{aligned}$$

Então,

$$E_{T_Y} = E'_{Y_1} \dot{\times} E'_{Y_2} \dot{\times} E'_{Y_3} \dot{\times} E'_{Y_4} = \{\{d\}, \{b, d\}\} = E_T$$

que resulta em $E_{T_2} = E_T = E_{T_Y}$. O próximo estado é $x_3 = \{10Y; 6N, 8Y, 9N, 10Y\}$. Encontra-se, associado a este, somente o próprio estado x_3, e portanto, $x_{31} = x_3$. Com isso, o conjunto de eventos será

$$E_{T_3} = \{b, c, d\} \setminus \{a, c\} = \{\{b\}, \{d\}\}.$$

O estado subseqüente é $x_4 = \{10Y; 9N, 10Y\}$, que possui $x_{41} = x_4$ e $x_{42} = x_9$ como estados associados. Formam-se os conjuntos $E_{Y_1} = \{b, c, d\}$, $E_{Y_2} = \{b, c, d\}$ e $E_{N_1} = \{b, c\}$, e calculam-se

$$\begin{aligned} E'_{Y_1} = E'_{Y_2} &= E_{Y_1} \setminus \{a, c\} = \{b, d\} \\ E'_{N_1} &= \{b, c\} \setminus \{a, c\} = \{b\}. \end{aligned}$$

Logo,

$$\begin{aligned} E_{T_Y} &= E'_{Y_1} \dot{\times} E'_{Y_2} = \{\{b\}, \{d\}, \{b, d\}\} \\ E_{T_N} &= E'_{N_1} \setminus E'_{Y_1} \cup E'_{Y_2} = \emptyset. \end{aligned}$$

Com isso, tem-se que $E_{T_4} = E_T = E_{T_Y}$. O estado $x_5 = \{11Y; 2N, 4N, 5Y, 7Y, 11Y\}$ também possui somente ele mesmo como estado associado, e com isso, o conjunto de estados $E_{T_5} = \{a, b, d\}\backslash\{a, c\} = \{\{b\}, \{d\}\}$. Os estados x_6 e x_7 possuem as mesmas segundas componentes que os estados x_1 e x_2, e portanto não precisam ser analisados. Os novos candidatos a base mínima serão determinados pelo conjunto

$$\begin{aligned} E_{n_T} &= E_{T_1} \dot{\times} E_{T_2} \dot{\times} E_{T_3} \dot{\times} E_{T_4} \dot{\times} E_{T_5} \\ &= \{\{b\}, \{d\}, \{b, d\}\} \dot{\times} \{\{d\}, \{b, d\}\} \dot{\times} \{\{b\}, \{d\}\} \dot{\times} \{\{b\}, \{d\}, \{b, d\}\} \dot{\times} \{\{b\}, \{d\}\} \\ &= \{\{d\}, \{b, d\}\}. \end{aligned}$$

Tem-se como novos candidatos à base mínima os conjuntos pertencentes a

$$\begin{aligned} E'_{o_1} &= \{\{a, c, d\}, \{a, b, c, d\}\}\backslash\{a, b, c, d\} \\ &= \{a, c, d\} \end{aligned}$$

chegando a

$$E_{cbd} = \{a, c, d\}.$$

Com o primeiro ciclo finalizado, o conjunto E_{cbd} que no início do ciclo anterior possuía somente um subconjunto formado pelos eventos a e c, continua possuindo somente um subconjunto, formado pelos eventos a, c e d. Retornando ao passo 3 do algoritmo 2, constrói-se o seu diagnosticador parcial mostrado na figura 3.15; observa-se que não há ciclos indeterminados, tanto observáveis quanto escondidos, e portanto, o conjunto E_{o_2} é uma base mínima para o diagnóstico de falhas em G. Com isso, o conjunto de candidatos à base mínima se reduz a $E_{cbd} = \emptyset$, e de acordo com o passo 5 do algoritmo 2, o processo deve ser finalizado. Assim, o conjunto de bases mínimas encontrado para este sistema é $E_{bmd} = \{a, c, d\}$.

O resultado encontrado no exemplo acima já era esperado, pois este sistema foi analisado no Exemplo 11, tendo os subconjuntos $E'_o = \{a, c, d\}$ e $E''_o = \{a, b, d\}$ sido testados diretamente pela composição paralela entre o diagnosticador parcial e o diagnosticador centralizado. O resultado obtido foi positivo para o subconjunto E'_o. Este fato comprova

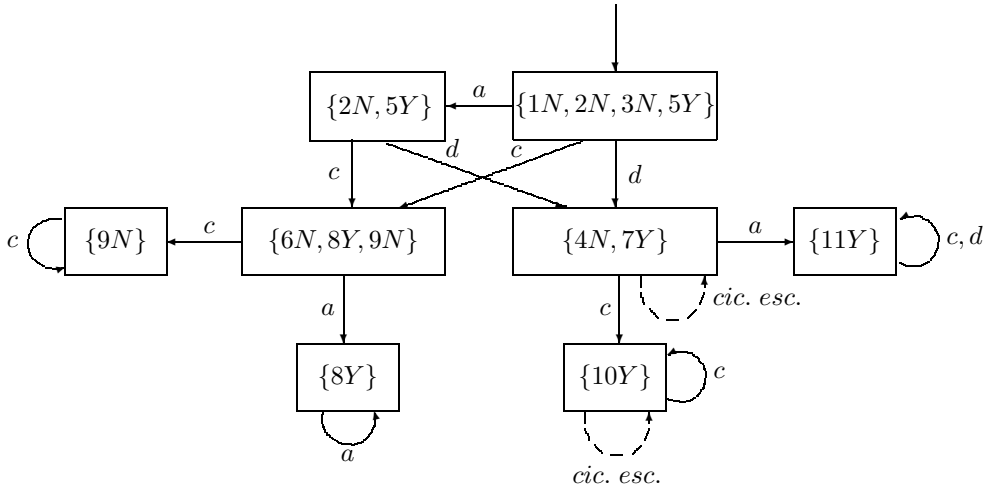

Figura 3.15: Diagnosticador parcial para o conjunto de eventos observáveis $\{a, c, d\}$.

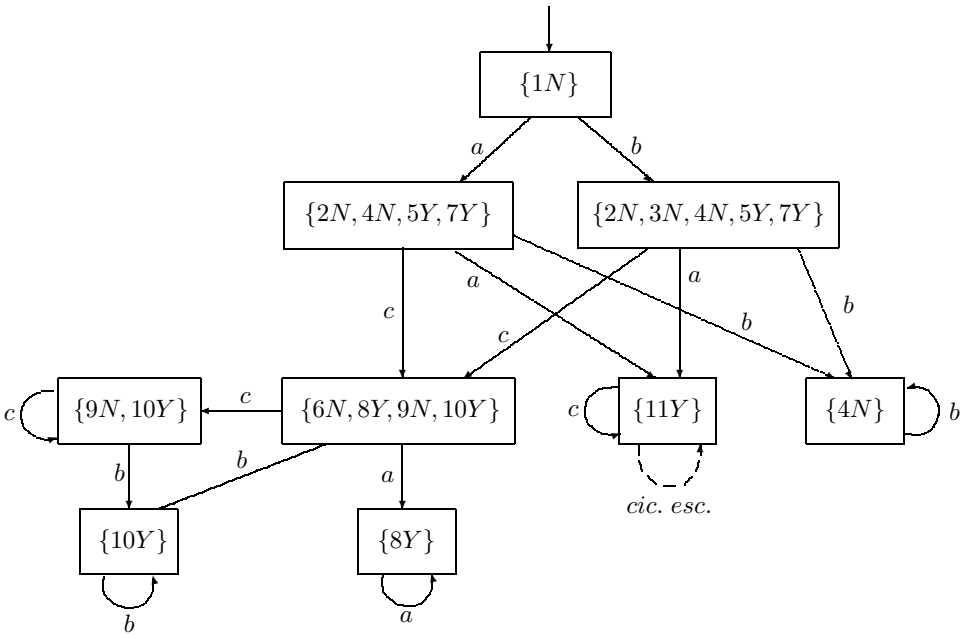

Figura 3.16: Diagnosticador parcial para o conjunto de eventos observáveis $\{a, b, c\}$.

que o evento c é um evento essencial para o conjunto E_o. O mesmo aconteceria se o evento a fosse retirado do conjunto de eventos observáveis. Isso ocorre, pois tanto o evento a quanto o evento c são eventos definitivos, e uma vez que sua observabilidade é perdida, um ciclo escondido indeterminado irá aparecer no diagnosticador parcial relacionado ao subconjunto que não possui um destes eventos. Logo, com a identificação dos eventos essenciais, e visto que somente a observação destes não permite o diagnóstico de falhas no sistema estudado, só resta testar os subconjuntos $\{a, c, d\}$ e $\{a, b, c\}$. Através de uma análise do diagnosticador parcial que possui este último como seu conjunto de eventos observáveis (mostrado na figura 3.16), verifica-se que somente a inclusão do evento b não garante a solução de ambigüidades de seqüências. Um exemplo disso é a ambigüidade formada pelas famílias de seqüências $s_Y = adc\{c\}^n$ e $s_N = ac\{c\}^n$, que para o conjunto $E_{o_3} = \{a, b, c\}$, possuem projeções $P_{o_3}(s_1) = P_{o_3}(s_2) = ac\{c\}^n$, comprovando uma vez mais que este subconjunto não é uma base mínima para o diagnóstico de falhas em G. Por fim, o subconjunto $E_{o_2} = \{a, c, d\}$ comprovadamente é uma base mínima para o diagnóstico de falhas, confirmando a validade do método proposto para identificar as bases mínimas para o diagnóstico de falhas em SED.

Capítulo 4

Conclusão

Neste trabalho apresentou-se um método sistemático com a finalidade de se encontrar todas as bases mínimas para o diagnóstico de falhas em Sistemas a Eventos Discretos, permitindo que o diagnóstico de um sistema que contenha falhas modeladas como eventos não-observáveis seja realizado com o menor número de eventos observáveis possível. Este fato faz com que o número de sensores a serem utilizados para o diagnóstico de falhas neste sistema seja reduzido ao máximo, já que este número está diretamente ligado à quantidade de eventos a serem observados, levando a uma diminuição considerável nos custos de um projeto para o diagnóstico de falhas em um processo industrial.

O método proposto pode ser implementado através de diversas linguagens de programação computacional, permitindo que este seja aplicado a sistemas com número elevado de estados e eventos, que são característicos em grandes processos industriais.

Como trabalhos futuros, pode-se destacar:

1. Posicionamento ótimo de sensores. A partir da obtenção das bases mínimas para o diagnóstico de falhas, pode-se também aplicar diferentes critérios para definição dos sensores a serem utilizados, como, por exemplo, estabelecer um compromisso entre o número de sensores e o atraso na deteção da falha (número de eventos que deverão ocorrer até que a falha seja detectada de fato pelo diagnosticador). Neste caso, o atraso no diagnóstico da falha acarretaria em perdas consideráveis no processo, até que uma ação fosse tomada pelo controle supervisório, a partir da informação enviada pelo diagnosticador da ocorrência da falha.

2. Diagnose robusta à perda definitiva de sensores. Dado que o método proposto retorna todas as bases mínimas para o diagnóstico de falhas, e que um dado sistema possui mais de uma base mínima, pode-se pensar em aumentar a confiabilidade deste sistema de deteção de falhas utilizando-se mais de um diagnosticador, cada um possuindo uma base mínima como conjunto de eventos observáveis. Com isso, o sistema de deteção de falhas se tornará mais robusto a falhas em sensores, isto é, caso um sensor venha a falhar, o diagnosticador parcial que utilizava a informação deste sensor para detectar a falha perderá sua validade, porém, ainda restarão outros diagnosticadores que não possuíam como observável o evento cuja observação era feita pelo sensor defeituoso, permitindo assim que, em alguns casos, o sistema continue a detectar a falha caso ela venha a ocorrer.

Referências Bibliográficas

[1] John W. Webb e Ronald A. Reis, *Programmable Logic Controllers: Principles e Applications*, Prentice-Hall, Inc., Upper Saddle River, USA, 4th edition, 1999.

[2] C. G. Cassandras e S. Lafortune, *Introduction to Discrete Event Systems*, Springer, New York, 2nd edition, 2007.

[3] Paulo Eigi Miyagi, *Controle Programável: Fundamentos do Controle de Sistemas a Eventos Discretos*, Edgard Blücher LTDA., São Paulo, Brasil, 1st edition, 1996.

[4] P. J. Ramadge e W. M. Wonham, "The control of discrete-event systems," *Proceedings of the IEEE*, vol. 77, pp. 81–98, 1989.

[5] James L. Peterson, *Petri Net Theory e the Modeling of Systems*, Prentice-Hall, Inc., Englewood Cliffs, N.J., USA, 1st edition, 1981.

[6] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, e D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Trans. on Automatic Control*, vol. 40, pp. 1555–1575, 1995.

[7] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, e D. Teneketzis, "Failure diagnosis using discrete event models," *IEEE Trans. on Control Systems Technology*, vol. 4, pp. 105–124, 1996.

[8] R. Debouk, S. Lafortune, e D. Teneketzis, "Coordinated decentralized protocols for failure diagnosis of discrete event systems," *Discrete Event Dynamic Systems*, vol. 10, pp. 33–86, 2000.

[9] J. C. Basilio e S. Lafortune, "Robust diagnosis of discrete event systems," Tech. Rep., UMICH, Department of Electrical Engineering e Computer Science, 2008.