COPPE/UFRJ

DIAGNOSE ROBUSTA DE SISTEMAS A EVENTOS DISCRETOS SUJEITOS À PERDA PERMANENTE DE SENSORES

Saulo Telles de Souza Lima

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: João Carlos dos Santos Basilio

Rio de Janeiro
Março de 2010

DIAGNOSE ROBUSTA DE SISTEMAS A EVENTOS DISCRETOS SUJEITOS À PERDA PERMANENTE DE SENSORES

Saulo Telles de Souza Lima

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. João Carlos dos Santos Basilio, D. Phil.

Prof. Marcos Vicente de Brito Moreira, D. Sc.

Prof. Gilberto Oliveira Corrêa, Ph. D.

RIO DE JANEIRO, RJ – BRASIL
MARÇO DE 2010

Lima, Saulo Telles de Souza

Diagnose robusta de sistemas a eventos discretos sujeitos à perda permanente de sensores/Saulo Telles de Souza Lima. – Rio de Janeiro: UFRJ/COPPE, 2010.

XII, 102 p.: il.; 29, 7cm.

Orientador: João Carlos dos Santos Basilio

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2010.

Referências Bibliográficas: p. 99 – 102.

1. Diagnose de falhas. 2. Falhas permanentes em sensores. 3. Sistemas a eventos discretos. I. Basilio, João Carlos dos Santos. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

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, pelos seus esforços, 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.

Gostaria de agradecer a co-orientação informal do professor Stéphane Lafortune, do Departamento de Engenharia Elétrica e Ciência da Computação da Universidade de Michigan, nos Estados Unidos, que contribuiu profundamente com sua vasta experiência e conhecimento sobre o tema da presente dissertação, sendo autor de artigos e livros sobre o assunto.

Por fim, gostaria de manifestar meu carinho e 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, se mostrando, acima de tudo, um grande amigo.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

DIAGNOSE ROBUSTA DE SISTEMAS A EVENTOS DISCRETOS SUJEITOS À PERDA PERMANENTE DE SENSORES

Saulo Telles de Souza Lima

Março/2010

Orientador: João Carlos dos Santos Basilio

Programa: Engenharia Elétrica

O objetivo de um sistema de diagnose de falhas é inferir e informar a ocorrência de falhas em um sistema a partir de informações recebidas em tempo real sobre o comportamento desse sistema. Uma das maneiras de se abordar esse problema é através da construção de um modelo a eventos discretos do sistema cuja ocorrência de falhas deve ser diagnosticada. Nesse caso a decisão sobre a ocorrência da falha (diagnose) é tomada considerando-se somente os eventos que tenham sido observados, isto é, registrados pelos sensores. Na prática, isso é feito utilizando-se diagnosticadores. Diagnosticadores são autômatos determinísticos cujos estados são conjuntos formados pelos estados do autômato do sistema (planta), juntamente com marcações que indicam se a sequência de eventos ocorrida possui ou não o evento associado à falha. A seguinte pergunta pode, então, ser feita: todos os eventos observáveis que estão sendo utilizados são realmente necessários para diagnosticar a ocorrência da falha em questão? Esse trabalho não somente responde a essa pergunta como também propõe uma forma sistemática de se encontrar todos os subconjuntos (bases para a diagnose) do conjunto de eventos observáveis que são essenciais para a diagnose de falha em um sistema a evento discreto. Além disso, é proposto um diagnosticador robusto que emprega a redundância resultante da utilização das bases para ampliar o diagnosticador de modo a garantir a diagnose de falhas mesmo quando há perda definitiva de sensores na planta.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

ROBUST CENTRALIZED FAILURE DIAGNOSIS OF DISCRETE EVENT SYSTEMS SUBJECT TO PERMANENT SENSOR FAILURES

Saulo Telles de Souza Lima

March/2010

Advisor: João Carlos dos Santos Basilio

Department: Electrical Engineering

The main purpose of a fault diagnosis system is to infer and inform the fault occurrence in a system based on online information on the system behavior. One approach to fault diagnosis is through the construction of a discrete-event model of the system whose fault occurrence must be diagnosed. In this case, the decision on the fault occurrence (diagnosis) is taken by considering only the observed events, i.e., those events whose occurrence can be recorded by sensors. In practice, this is carried out by using diagnosers. Diagnosers are deterministic automata whose states are sets formed with the states of the system automaton (plant) together with labels that indicate if the trace occurred so far possesses or not the fault event. The following question can then be raised: are all the observable events being used really necessary to diagnose the fault occurrence? This work not only answers this question but also proposes a systematic way to find all subsets (bases for diagnosability) of the set of observable events that are necessary to diagnose the fault in a discrete-event system. It is also proposed a robust diagnoser that deploys the resulting redundancy to augment the diagnoser with a view to guarantee the fault diagnosis even when there are sensor losses in the plant.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Diagnose de falhas em sistemas a eventos discretos (SEDs) tem despertado grande interesse nos últimos anos [1–14]. Uma das razões para esse interesse é o fato de modelos a eventos discretos poderem ser aplicados não só a sistemas em que esses modelos são os mais apropriados (sistemas de computação, redes de comunicação e de manufatura, por exemplo), como também a diversos sistemas dinâmicos de variáveis contínuas (SDVC), uma vez que SVDC sistemas podem também ser modelados como SEDs dependendo do grau de abstração.

Dois paradigmas norteiam a diagnose de falhas em SEDs:

1. As falhas a serem diagnosticadas são eventos não observáveis, isto é, eventos cujas ocorrências não podem ser registradas por sensores;

2. A ocorrência de falhas altera o comportamento do sistema, porém não necessariamente leva o sistema a uma parada (em sistemas de manufatura, por exemplo, a ocorrência de uma falha não diagnosticada pode levar a uma degradação da produção e, consequentemente, à perda de qualidade).

Assim, o objetivo de um sistema de diagnose de falhas é inferir e informar a ocorrência de falhas tendo como base somente os eventos que tenham sido observados, isto é, registrados pelos sensores.

O projeto de um sistema que permite diagnosticar falhas em SEDs pode ser dividido em duas etapas principais: *(i)* a construção de um modelo a eventos discretos do sistema que se deseja realizar a diagnose de falhas; *(ii)* O desenvolvimento de um conjunto de regras (protocolo) a serem seguidas para a identificação e a diagnose de falhas. A primeira parte consiste no desenvolvimento de um modelo que capture tanto o comportamento normal quanto o comportamento do sistema levando-se em consideração a ocorrência da falha [1]. Os modelos mais comumente utilizados são os autômatos [2, 15] e as redes de Petri [16–18]. A segunda parte é calcada em um arcabouço teórico introduzido por LIN [19] e SAMPATH *et al.* [3] e desenvolvido nas duas últimas décadas. Grande parte dessa base teórica será revista neste trabalho, uma vez que é a base para os resultados aqui apresentados.

O problema da diagnose de falhas foi trazido para o contexto de SED por LIN [19], que introduziu o conceito da capacidade de se diagnosticar a ocorrência de uma falha em um sistema. Logo a seguir, SAMPATH *et al.* [3] apresentaram condições necessárias e suficientes para a diagnose de SED e propuseram a construção de um autômato diagnosticador que permite tanto inferir sobre a capacidade de diagnosticar as falhas presentes no sistema quanto ser usado para realizar a diagnose de falhas em tempo real. O diagnosticador é, por definição, um autômato no qual cada estado é formado por um conjunto de estados do autômato a ser analisado. Os estados do diagnosticador são construídos de forma a representar uma estimativa do estado atual do sistema, após a ocorrência de um evento observável e possui marcações que indicam se a sequência de eventos ocorrida possui ou não a falha. Logo, o diagnosticador infere, a partir do registro da ocorrência somente de eventos observáveis, qual sequência de eventos (incluindo eventos não-observáveis) do sistema ocorreu.

Para tornar possível a diagnose de uma falha em SEDs cujos modelos não satisfazem as condições para diagnosticabilidade apresentadas em SAMPATH *et al.* [3], as seguintes abordagens podem ser seguidas:

1. Introdução de mais sensores no sistema. Essa abordagem tem a desvantagem de introduzir outros sensores além daqueles realmente necessários para a operação normal do sistema. É, em geral, rejeitada por razões econômicas.

2. Introdução dos chamados sensores virtuais [4]. Sensores virtuais são usados para aumentar a quantidade de informações fornecidas pelos sensores reais do sistema, sendo as novas informações obtidas analiticamente.

3. Uso de ações de controle para alterar a propriedade de diagnosticabilidade de um sistema [5], restringindo-se o comportamento de um sistema não-diagnosticável através de ações de controle apropriadas para torná-lo diagnosticável. Essa abordagem, diferentemente das soluções 1) e 2) acima, que tratam a diagnose de falhas como passiva, combina observação e controle, sendo esse último problema formulado e resolvido utilizando a teoria de controle supervisório [20].

O diagnosticador proposto por SAMPATH *et al.* [3] requer que todos os eventos observáveis do sistema estejam acessíveis em um determinado ponto. Contudo na prática, devido à natureza descentralizada de alguns sistemas e grandes diferenças entre as distâncias entre o diagnosticador e os sensores que registram a ocorrência de eventos, o que pode levar a erros na sequência de ocorrência dos eventos informada ao diagnosticador, não é sempre possível utilizar o diagnosticador proposto por SAMPATH *et al.* [3]. Assim, inspirado nos resultados de LIN e WONHAM [21] para controle supervisório descentralizado, DEBOUK *et al.* [6] propuseram uma arquitetura descentralizada com coordenação, denominada codiagnose, que consiste de módulos locais capazes de observar a ocorrência de parte dos eventos observáveis do sistema. Esses módulos locais se comunicam com um coordenador, que é

responsável pela diagnose das falhas que venham a ocorrer no sistema. A noção de diagnosticabilidade introduzida por SAMPATH *et al.* [3] é estendida em DEBOUK *et al.* [6] levando ao conceito de diagnose descentralizada. Em um trabalho posterior, CONTANT *et al.* [7] introduzem o conceito de diagnosticabilidade modular em sistemas que podem ser modelados pela composição paralela de autômatos, em que cada autômato representa um componente local (ou subsistema, ou módulo) do sistema global. É mostrado que se o sistema for modularmente diagnosticável, isto é, se cada subsistema for diagnosticável, então a diagnose de falha do sistema global será obtida utilizando-se somente os diagnosticadores locais (*i.e.*, os diagnosticadores projetados para cada um dos subsistemas). Mais recentemente, BASILIO e LAFORTUNE [22] apresentam o conceito de codiagnose robusta, segundo a qual, uma arquitetura descentralizada diagnosticável será robusta se e somente se continuar diagnosticável mesmo com a perda de comunicação entre um ou mais módulos e o coordenador. Condições necessárias e suficientes para codiagnose robusta são apresentadas em BASILIO e LAFORTUNE [22].

O diagnosticador proposto por SAMPATH *et al.* [3], embora intuitivo e com aplicabilidade para diagnose em tempo real apresenta, no que se refere à sua utilização na análise da diagnosticabilidade, a deficiência de ter complexidade computacional exponencial para o cálculo do espaço de estados do diagnosticador em relação à cardinalidade do espaço de estados do autômato cuja linguagem gerada se deseja diagnosticar. Para contornar esse problema, JIANG *et al.* [23] e YOO e LAFORTUNE [24] propõem um novo método para verificar a diagnosticabilidade de SEDs baseado na construção de autômatos não determinísticos denominado verificadores, cujos espaços de estados são polinomiais na cardinalidade do espaço de estados do modelo do sistema. Mais recentemente, QIU e KUMAR [11] e WANG *et al.* [25] estenderam esses verificadores para a codiagnose, levando aos chamados verificadores descentralizados.

Como pode ser visto nos parágrafos precedentes, a maioria dos trabalhos em diagnose de falhas modela os SEDs utilizando autômatos. Conforme mencionado anteriormente, SEDs podem também ser modelados utilizando redes de Petri. Os principais trabalhos que abordam a diagnose de falhas num contexto de redes de Petri são os de USHIO *et al.* [26], CHUNG *et al.* [27], GIUA e SEATZU [28], RAMIREZ-TREVINO *et al.* [29], GENC e LAFORTUNE [30] e MANYARI-RIVERA *et al.* [31]. Contudo, embora redes de Petri sejam utilizadas na modelagem dos SEDs, os diagnosticadores propostos são ainda autômatos determinísticos.

Neste trabalho, os SEDs considerados serão modelados por autômatos e serão utilizados diagnosticadores para realizar a diagnose de falhas. O diagnosticador, por sua vez, utiliza a sequência de eventos observáveis realizada pelo autômato para inferir sobre a ocorrência da falha ou não.

Contudo, uma questão vem à tona: será possível diagnosticar a ocorrência de uma falha considerando-se como eventos observáveis somente parte daqueles que realmente podem ser observados? Se a resposta for sim, pode-se concluir que somente parte dos sensores que anteriormente necessitariam ser utilizados, deverão realmente ser utilizados para indicar a ocorrência de eventos para que a falha seja diagnosticada. De fato, para muitos SEDs, a resposta é sim. Nesse contexto, este trabalho irá propor uma forma sistemática de se encontrar todos os subconjuntos do conjunto de eventos observáveis que devem ser observados para que a falha seja diagnosticada em um SED. Esses subconjuntos serão referidos como bases para a diagnose de um SED.

O problema de se encontrar um subconjunto do conjunto de eventos observáveis que permite a diagnose de falhas em SEDs já foi considerado em alguns trabalhos [8, 32, 33]. Em DEBOUK *et al.* [33] e em JIANG *et al.* [8], a formulação do problema é feita no sentido de se encontrar a seleção de sensores que minimiza um funcional de custo (seleção ótima), sendo que essa seleção precisa manter uma dada propriedade; por exemplo, a diagnosticabilidade da linguagem gerada pelo autômato. Além disso, em DEBOUK *et al.* [33], o método proposto retorna a quantidade mínima de cálculos a serem realizados para se afirmar que a seleção de sensores que minimiza o custo possui a propriedade especificada. Já em TRAVÉ-MASSUYÈS *et al.* [32], o objetivo é encontrar um conjunto de sensores redundantes que somados ao conjunto de sensores já existentes levem a um grau desejado de diagnosticabilidade e de discriminabilidade, que é a capacidade de distinguir entre duas falhas distintas. A busca é feita de forma exaustiva utilizando-se o modelo do sistema.

O objetivo deste trabalho difere de todos os outros citados, já que o interesse é encontrar todos os subconjuntos do conjunto de eventos observáveis que permitam a diagnose de falhas em um SED, situação ainda não abordada em publicações na área de Diagnose de Falhas em SEDs. Além disso, o método desenvolvido neste trabalho não é baseado em busca exaustiva, e sim num conjunto de condições necessárias que norteiam o processo da busca, indicando os subconjuntos que satisfazem a essas condições, e que, portanto, são candidatos a bases para a diagnose de falhas do SED.

Se a diagnose de falhas puder ser realizada através da observação de um subconjunto do conjunto de eventos observáveis, então pode-se dizer que o diagnosticador centralizado com observação total utiliza sensores redundantes, e não aproveita as vantagens introduzidas por essa redundância. Uma das principais vantagens de se introduzir sensores redundantes é o aumento da confiabilidade da diagnose de falhas com relação à perda permanente de um sensor por defeito. Assim, nesse trabalho será proposto um diagnosticador que seja robusto à perda definitiva de um sensor utilizando-se subconjuntos do conjunto de eventos observáveis que permitem a diagnose de falhas.

Em resumo, o objetivo deste trabalho é fornecer ferramentas teóricas para que o diagnosticador robusto seja construído para um dado SED. Isso envolve os seguintes passos: *(i)* o desenvolvimento de um método sistemático para se encontrar todas as bases para a diagnose, *i.e.*, os possíveis conjuntos de eventos a serem observados, que permitem a diagnose de falhas do SED; *(ii)* o desenvolvimento do algoritmo de construção do diagnosticador robusto; *(iii)* e o estabelecimento de condições necessárias e suficientes para que eventos observáveis possam ser ignorados permanentemente (falha no sensor) sem interferir na diagnose de falhas do SED através do diagnosticador robusto.

Para atingir os objetivos traçados, este trabalho está estruturado da seguinte forma: no capítulo 2 são revistos os principais conceitos de SEDs necessários ao entendimento dos resultados sobre diagnose de falhas, apresentados ao final deste mesmo capítulo. No capítulo 3 analisa-se a diagnose de falhas sob observação parcial, sendo apresentadas as condições necessárias e suficientes para essa diagnose. Ainda no capítulo 3, são apresentadas definições e resultados básicos para a apresentação de um algoritmo para a busca das bases mínimas para a diagnose de falhas em SEDs. O algoritmo de construção do diagnosticador robusto a falhas permanentes de sensores é apresentado no capítulo 4, bem como o grau de robustez desse diagnosticador (à perda de quais sensores o diagnosticador é robusto). Finalmente, no capítulo 5, são feitas as conclusões e propostas de trabalhos futuros sobre o tema.

Capítulo 2

Diagnose de falhas em SEDs

Neste capítulo serão revistos os conceitos e resultados mais relevantes, para este trabalho, da teoria de Sistemas a Eventos Discretos (SED) aplicados à diagnose de falhas. O objetivo principal do capítulo é dar subsídios para um bom entendimento dos formalismos teóricos introduzidos nos capítulos seguintes deste trabalho. Maiores detalhes podem ser obtidos em CASSANDRAS e LAFORTUNE [2].

Este capítulo está estruturado da seguinte forma. 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, é apresentada a composições paralela (também chamada de composição síncrona) entre dois autômatos, que será largamente utilizada em alguns teoremas e algoritmos apresentados no capítulo seguinte deste trabalho. Na seção 2.2, são introduzidos os conceitos de autômatos não-determinísticos e observadores, e o principal assunto do capítulo: diagnose de falhas. Na última seção, são enunciados os principais conceitos sobre diagnose de falhas em SED, sendo apresentadas as condições necessárias e suficientes para a diagnosticabilidade de SED [3], bem como um algoritmo para a construção de diagnosticadores 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 a ele associado, e cada evento é comparado a uma letra de um alfabeto, no qual uma seqüência de eventos forma uma palavra de uma linguagem sobre o alfabeto. Portanto, o conjunto de eventos E de um dado sistema é como um alfabeto. Uma seqüência que não possui eventos é formada pela *sequência vazia* ε.

Definição 2.1 *(Linguagem) Uma linguagem definida sobre um conjunto de eventos*

E é um conjunto de seqüências formadas por eventos pertencentes a E. □

A título de ilustração da definição acima, suponha um conjunto de eventos $E = \{a, b, g\}$. Pode-se definir a linguagem:

$$L_1 = \{\varepsilon, aa, aabb\},$$

que é formada por três seqüências somente, a linguagem L_2, formada por todas as possíveis seqüências de tamanho 2 iniciadas pelo evento g, ou seja,

$$L_2 = \{ga, gb, gg\},$$

que também contém três seqüências, ou a linguagem

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

que contém um número infinito de elementos.

Definição 2.2 *(Fecho de Kleene) O Fecho de Kleene de um conjunto de eventos E é o conjunto de todas as seqüências finitas formadas por elementos de E, incluindo o sequência vazia ε. É denotado por E^*.* □

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.

Definição 2.3 *(Fecho do prefixo) Seja uma linguagem L definida sobre um conjunto de eventos E. O fecho do prefixo de L é o conjunto*

$$\overline{L} = \{u \in E^* : (\exists s \in L)[u \text{ é um prefixo de } s]\}.$$

□

Note que $L \subseteq \overline{L}$. No caso específico em que $L = \overline{L}$, a linguagem L é dita ser *de prefixo fechado.* Como ilustração, suponha uma linguagem $L = \{ac, abc\}$. Nesse caso, tem-se que

$$\overline{L} = \{\varepsilon, a, ac, ab, abc\}.$$

2.1.2 Autômatos

Um autômato é um dispositivo capaz de representar uma dada linguagem através de regras bem definidas. Sua definição formal segue abaixo.

Definição 2.4 *(Autômato determinístico ou simplesmente autômato) Um autômato, denotado por G, é uma sêxtupla*

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

em que

- *X é o conjunto de estados,*
- *E é um conjunto finito de eventos associados às transições em G,*
- *$f : X \times E \to X$ é a função de transição definida como $f(x, e) = y$, que denota a existência de uma transição definida pelo evento e, do estado x para o estado y,*
- *$\Gamma : X \to 2^E$ é a função de eventos ativos[1] definida como $\Gamma(x) = \{e \in E : f(x, e)$ é definida$\}$, chamado de conjunto de eventos ativos de G em x,*
- *x_0 o estado inicial, e*
- *$X_m \subseteq X$ é o conjunto de estados marcados.*

□

Observação 2.1

(a) Um autômato será denominado determinístico quando para todo estado $x \in X$ e para todo $e \in \Gamma(x)$ 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.*

(b) A função de transição de estados f pode ser parcialmente definida em seu domínio. Isso 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.

[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.

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

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

(d) É comum omitir Γ na definição de autômato, uma vez que este pode ser obtido em cada estado $x \in X$ diretamente a partir das transições definidas em x por f. □

Um autômato G opera da seguinte forma: partindo do estado inicial x_0 e, após a ocorrência de um evento $e \in \Gamma(x_0)$, faz a transição 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 2.1 *Considere o diagrama da figura 2.1, no qual os círculos representam os estados e os arcos rotulados representam as 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, $E = \{a, b, g\}$ (rótulos das transições). Os arcos no diagrama são uma representação da função de transição de G, denotada por $f : X \times E \to X$, 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 ocorrência do evento a pode ser devido a uma entrada externa ao sistema modelado pelo autômato ou a uma ação gerada espontaneamente 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

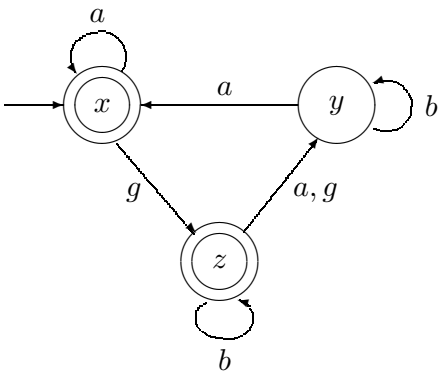

Figura 2.1: Diagrama de transição de estados do exemplo 1

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

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

A linguagem marcada por G *é definida como*

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

□

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

O exemplo a seguir ilustra a determinação das linguagens gerada e marcada por um autômato.

Exemplo 2.2 *(Linguagens gerada e marcada) Suponha o conjunto de eventos* $E = \{a, b\}$*, e considere a linguagem*

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

que consiste de todas as seqüências formadas pelos eventos a ou b sempre terminadas pelo evento a. Pode-se verificar que 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* $X = \{0, 1\}$*,* $x_0 = 0$ *e* $X_m = \{1\}$*. Para esse autômato,* f *está*

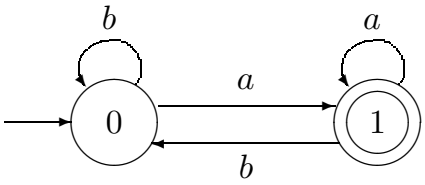

Figura 2.2: Diagrama de transição de estados para o autômato do exemplo 2

definida da seguinte forma: $f(0,a) = 1$, $f(0,b) = 0$, $f(1,a) = 1$, $f(1,b) = 0$. *Assim, estando o sistema no seu estado inicial* 0*, a única maneira de se alcançar o estado marcado* 1 *é pela ocorrência do evento* a*, após a ocorrência de qualquer subseqüência formada por uma seqüência em* $\{b\}^*$*. Alcançado este estado, o autômato somente retornará ao estado* 0 *após a ocorrência do evento* b*, depois da ocorrência de uma subseqüência formada por qualquer seqüência em* $\{a\}^*$*. Com isso, este processo pode ser repetido indefinidamente, donde se pode concluir que* $\mathcal{L}_m(G) = L$*. O processo descrito pode facilmente ser retirado do diagrama de transição de estados de* G *mostrado na figura 2.2. 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 de linguagens é uma função muito utilizada em diagnose de falhas de SED, pois sua definição é tal que ao aplicá-la em uma seqüência, somente os eventos escolhidos (observáveis, por exemplo) serão considerados, sendo apagados todos os eventos restantes (não-observáveis, por exemplo) dessa sequência. Será também utilizada na definição de linguagens gerada e marcada pela composição paralela entre dois autômatos, que será apresentada na seção 2.1.6.

A projeção é uma função aplicada a uma determinada sequência de eventos ou a uma linguagem, tendo como base um conjunto de eventos. Sua definição formal é a seguinte.

Definição 2.6 *(Projeção) Sejam* E *e* E_i *conjuntos de eventos, em que* $E_i \subseteq E$*. A projeção de uma sequência 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 sequências pertencentes à linguagem da qual se deseja encontrar a projeção. Seja, portanto, uma linguagem $L \subseteq E^*$. Desta forma:

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

□

O tipo de projeção apresentada na definição 2.6 é chamada de *projeção natural*. Pode-se definir também a projeção inversa de uma sequência ou linguagem da seguinte forma:

Definição 2.7 *(Projeção inversa) A projeção inversa de uma sequência de eventos é definida como uma função*

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

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 $L_i \subseteq E_i^*$,

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

□

De acordo com a definição 2.7, dada uma sequência formada pelos eventos do conjunto de menor cardinalidade (E_i), a projeção inversa retorna o conjunto de todas as sequências formadas por eventos pertencentes ao conjunto de maior cardinalidade (E), cujas projeções são a própria sequência 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 autolaços rotulados por eventos pertencentes a $E \backslash E_i$ em todos os estados desse autômato[2].

[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.

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

Exemplo 2.3 *(Projeção) Considere* $E = \{a, b, c\}$, $E_1 = \{a, b\}$, $E_2 = \{b, c\}$, *e*

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

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 anteriormente. □

2.1.5 Produto de dois autômatos

Definição 2.8 *(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*

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

em que

$$f_{1\times 2}[(x_1, x_2), e] := \begin{cases} (f_1(x_1, e), f_2(x_2, e)). & \text{se } e \in \Gamma_1(x_1) \cap \Gamma_2(x_2), \\ \text{não definido,} & \text{caso contrário} \end{cases}$$

□

Como consequência 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 sequências 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

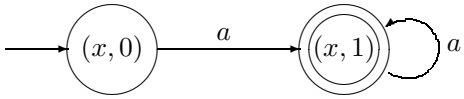

Figura 2.3: Autômato resultante do produto entre os autômatos das Figs. 2.1 e 2.2

$G_1 \times G_2$ são constituídos por pares em que o primeiro componente é o estado atual de G_1 e o segundo componente 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 2.4 *(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 evento* a*, que leva de* x *para* x *no primeiro autômato, e de* 0 *para* 1 *no segundo; 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 rotuladas 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 os estados,* x *e* 1*, são estados marcados em seus respectivos autômatos.* □

2.1.6 Composição paralela de dois autômatos (composição síncrona)

Esta operação terá um papel fundamental nos capítulos seguintes deste trabalho, e sua definição completa e formal segue abaixo.

Definição 2.9 *(Composição paralela de dois autômatos) 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*

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

em que

$$f_{1\|2}[(x_1,x_2),e] := \begin{cases} (f_1(x_1,e), f_2(x_2,e)), & \textit{se } e \in \Gamma_1(x_1) \cap \Gamma_2(x_2) \\ (f_1(x_1,e), x_2), & \textit{se } e \in \Gamma_1(x_1) \backslash E_2 \\ (x_1, f_2(x_2,e)), & \textit{se } e \in \Gamma_2(x_2) \backslash E_1 \\ \textit{não definido}, & \textit{caso contrário} \end{cases}$$

□

Pela definição acima, pode-se concluir que

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

Na composição paralela, um evento pertencente a ambos os autômatos somente poderá ser executado se os dois autômatos o 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 a definição de projeção de linguagens, apresentada na subseção 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 autolaços e pelo produto de autômatos. A projeção inversa de uma linguagem pode ser representada incluindo-se autolaços em todos os estados de G_1 e G_2; esses autolaços deverão ser rotulados 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 os autolaços, 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 autolaços.

Exemplo 2.5 *(Composição paralela) O autômato da figura 2.4 é o resultado da composição paralela dos autômatos das figuras 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 $G_1 \| G_2$ são formados por pares, cujas primeiras componentes*

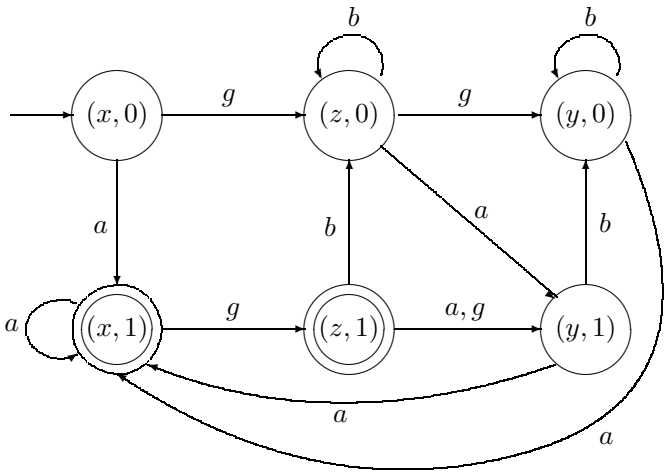

Figura 2.4: Autômato resultante da composição paralela entre os autômatos das figuras 2.1 e 2.2

pertencem a X_1 *e segundas componentes pertencem a* X_2*. 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* $(x, 1)$*, em* $(z, 0)$*, 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.* □

2.1.7 Modelo por eventos discretos de uma célula de manufatura

Para ilustrar a aplicação da teoria de SED, mais precisamente de autômatos, na modelagem de um sistema real, considere uma célula de manufatura formada por duas máquinas (M_1 e M_2) e um robô que transporta as peças de M_1 para M_2. A máquina M_1 recebe peças brutas e quando as peças estão prontas são recolhidas pelo robô. Caso o robô esteja ocupado, a máquina M_1 retém a peça até que o robô esteja completamente livre. Caso uma outra peça chegue enquanto a máquina M_1 estiver processando/retendo alguma peça, a máquina M_1 rejeita a peça recebida. Quando o robô recebe uma peça de M_1, inicia o transporte desta até a máquina M_2. No momento em que chegar a M_2, o robô somente entregará a peça à máquina M_2 se esta estiver livre; caso contrário reterá a peça até M_2 ficar disponível. Após

Elemento	Estados	Eventos
Máquina M_1	M_1 disponível:I_1 M_1 processando: P_1 M_1 retendo peça pronta: H_1 $X_1 = \{I_1, P_1, H_1\}$	Chegada de peça a M_1: a_1 Fim de processamento: t_1 Entrega de peça ao robô: e_1 $E_1 = \{a_1, t_1, e_1\}$
Robô	Robô disponível: I, Transportando $M_1 \rightarrow M_2$: T_{12} Esperando em M_2: H Retornando para M_1: R $X_r = \{I, T_{12}, H, R\}$	Entrega de peça ao robô: e_1 Chegada a M_2: c_2 Entrega/chegada de peça a M_2: a_2 Chegada a M_1: r_1 $E_r = \{e_1, c_2, a_2, r_1\}$
Máquina M_2	M_2 disponível: I_2 M_2 processando: P_2 $X_2 = \{I_2, P_2\}$	Entrega/chegada de peça em M_2: a_2 Fim de processamento: t_2 $E_2 = \{a_2, t_2\}$

Tabela 2.1: Os estados e os eventos das máquinas M_1, M_2 e do robô.

entregar a peça a M_2, o robô retorna à máquina M_1. A máquina M_2 recebe a peça do robô e a processa.

A tabela 2.1 descreve os estados e os eventos das máquinas M_1 e M_2 e do robô. Note que os eventos e_1 (entrega de peça ao robô) e a_2 (entrega/chegada de peça em M_2) pertencem a dois subsistemas: máquina M_1 e robô, e robô e máquina M_2, respectivamente. É importante notar que, para que o evento e_1 ocorra, a máquina M_1 deverá estar no estado H_1 e o robô no estado I. Para que o evento a_2 ocorra, o robô deverá estar no estado H e a máquina M_2 deverá estar no estado I_2. Para os demais estados dos sistemas, isto é, aqueles que estão presentes em somente um dos subsistemas, a ocorrência não dependerá do estado em que os demais subsistemas estiverem, sendo determinada somente pelo estado atual do subsistema; por exemplo, a ocorrência do evento t_1 (fim de processamento da peça em M_1) dependerá apenas da máquina M_1 estar no estado P_1, independentemente de quais estados estiverem o robô e a máquina M_2. □

Com o auxílio da tabela 2.1 é possível gerar os diagramas de transições dos autômatos das máquina M_1 e M_2, e do Robô, denominados por G_1, G_2 e G_r, que estão representados na figura 2.5.

O modelo do sistema pode ser obtido pela composição síncrona de G_1, G_2 e G_r. Para fins didáticos, a análise da composição síncrona de somente dois subsistemas é mais interessante do que a da composição dos três subsistemas, visto que o autômato

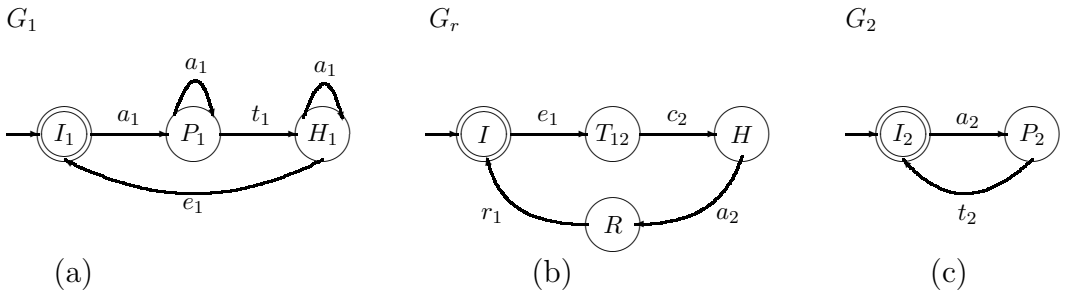

Figura 2.5: Máquina M_1 (a); Robô (b); Máquina M_2 (c).

resultante possui muitos estados e transições. A figura 2.6 mostra a composição $G_r||G_2$. Note que o evento a_2 é um evento comum dos autômatos G_r e G_2 e esse evento somente poderá ocorrer caso G_r e G_2 estejam em estados cujos conjuntos dos eventos ativos tenham o evento a_2 como elemento. Note ainda que o evento a_2 não pertence ao conjunto dos eventos ativos do estado I de G_r e, portanto, não poderá ocorrer no estado (I, I_2) de $G_r||G_2$.

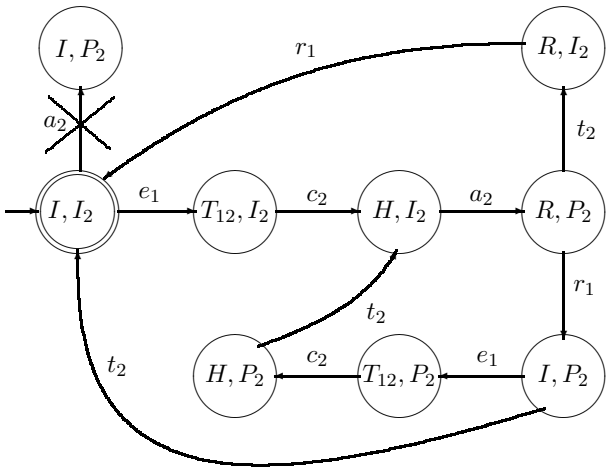

Figura 2.6: Composição síncrona de G_r e G_2.

2.2 Autômatos não-determinísticos e SEDs parcialmente observados

2.2.1 Autômatos não-determinísticos

Na definição de autômato apresentada 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 externo, de um evento cuja ocorrência não deve ser considerada no modelo; *(ii)* a não abstração de informações do sistema real de forma

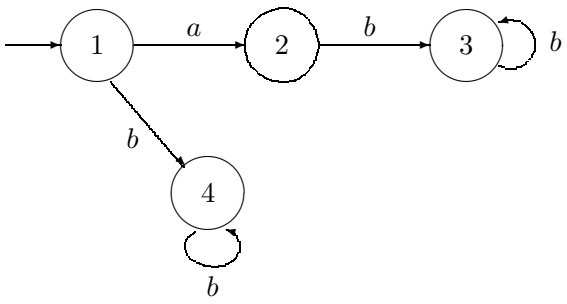

Figura 2.7: Exemplo de autômato não-determinístico para o caso em que o evento a é não-observável.

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 ou se 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 da palavra 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 a sequência vazia em operações de composição de autômatos que marcam linguagens regulares [2, pag. 94].

Exemplo 2.6 *(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, a sequência vazia* ε *pertence ao conjunto de eventos ativos dos estados* 1 *e* 2. □

Para uma perfeita caracterização da função de transição de estados de um autômato não-determinístico, é necessário definir o alcance-ε de um estado x, denotado por $\varepsilon R(x)$, da seguinte forma:

$$\begin{aligned} \varepsilon R(x) \;=\; & \{x' \in X : x' \text{ pode ser alcançado a partir de } x \\ & \text{seguindo transições rotuladas por } \varepsilon\}. \end{aligned}$$

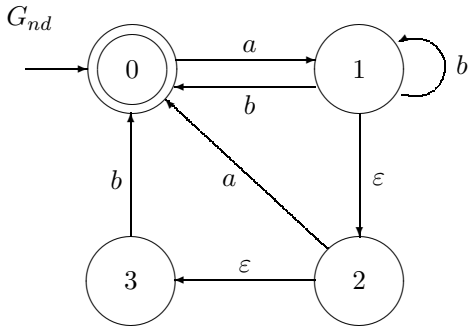

Figura 2.8: Autômato não-determinístico do exemplo 5.

No caso de um estado $B \in 2^X$, tem-se que

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

Torna-se necessário, ainda, definir a seguinte função de transição de estados.

$$\begin{aligned} f'_{nd} \quad : \quad & X \times E \to 2^X \\ & (x, e) \mapsto y = \{x' \in X : \text{existe uma transição rotulada por } e \\ & \text{do estado } x \text{ para o estado } x'\}. \end{aligned}$$

Definindo-se o conjunto de eventos como $E \cup \{\varepsilon\}$ 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 2.10 *(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 pela definição de autômato determinístico, apresentando somente duas diferenças:

- *A função de transição de estados é definida da seguinte forma:*

$$f_{nd} : X \times E \cup \{\varepsilon\} \to 2^X$$

$$f_{nd}(x, e) = \begin{cases} \varepsilon R(x), & e = \varepsilon \\ \varepsilon R[f'_{nd}(x, e)], & e \neq \varepsilon \end{cases}$$

- *O estado* inicial *pode ser um conjunto de estados, isto é $x_0 \subseteq X$.*

□

Pode-se estender f_{nd} para uma sequência 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 sequência 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 figura 2.8, $f_{nd}(0, aa) = \{0\}$ 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 sequência pertence à linguagem gerada pelo autômato não-determinístico se existir uma trajetória no diagrama de transição de estados definida pela própria sequência. Além disso, se uma dada trajetória levar o autômato a um estado marcado, então a sequência que define tal trajetória pertence à linguagem marcada pelo autômato não-determinístico. Por exemplo, a sequência ab está na linguagem marcada pelo autômato da figura 2.8, 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 sequência possa levar também a um estado que não seja marcado (estado 1, nesse caso).

2.2.2 SEDs parcialmente observados

Conforme citado anteriormente, transições definidas por ε representam, em sua maioria, mudanças de estado advindas da ocorrência de um evento não-observável. Por esse motivo, em vez de se utilizar a representação dessas transições marcadas com a sequência vazia e obter um autômato não-determinístico, podem-se rotular essas transições com eventos "genuínos", porém classificados como eventos não-observáveis. Além disso, se por algum motivo um mesmo evento não puder rotular várias transições não-observáveis (por exemplo, se um estado tiver mais de uma transição não-observável definida para estados diferentes), 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".

Um autômato que representa um SED parcialmente observado será definido como $G = (X, E, f, \Gamma, x_0, X_m)$, sendo $E = E_o \dot{\cup} E_{uo}$, em que $\dot{\cup}$ denota a partição de um conjunto, isto é, $E_o \cap E_{uo} = \emptyset$ e $E_o \cup E_{uo} = E$.

2.2.3 Observador

As transições rotuladas por eventos não-observáveis em um autômato geram uma ambiguidade na informação do estado atual em que o autômato se encontra. Isto se deve ao fato do observador externo não registrar a ocorrência de tais eventos, e portanto, a incerteza sobre o estado atual em que o SED se encontra fica caracterizada.

É possível construir um autômato determinístico capaz de registrar somente os eventos observáveis do SED e estimar o seu estado atual. Este autômato é denominado *observador*. Os estados do observador são uma estimativa do estado atual do autômato parcialmente observado, atualizada após a ocorrência de eventos observáveis. Com isso, seu espaço de estados será um subconjunto do conjunto potência do espaço de estados do autômato parcialmente observado.

Um procedimento para a construção do observador de um autômato parcialmente observado está descrito a seguir. Para tanto, seja $G = (X, E_o \dot{\cup} E_{uo}, f, x_0, X_m)$ um autômato parcialmente observado. Para cada estado $x \in X$ defina

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

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

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

Então $G_{obs} = (X_{obs}, E_o, f_{obs}, x_{0,obs}, X_{m,obs})$ é construído através do seguinte algoritmo.

Algoritmo 2.1
Passo 1: *Defina* $x_{0,obs} = UR(x_0)$ *e faça* $X_{obs} = \{x_{0,obs}\}$ *e* $\tilde{X}_{obs} = X_{obs}$.
Passo 2: $\hat{X}_{obs} = \tilde{X}_{obs}$ *e* $\tilde{X}_{obs} = \emptyset$.
Passo 3: *Para cada* $B \in \hat{X}_{obs}$,

- $\Gamma_{obs}(B) = (\bigcup_{x \in B} \Gamma(x)) \cap E_o$;

- *Para cada* $e \in \Gamma_{obs}(B)$,

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

- $\tilde{X}_{obs} \leftarrow \tilde{X}_{obs} \cup f_{obs}(B, e)$.

Passo 4: $X_{obs} \leftarrow X_{obs} \cup \tilde{X}_{obs}$
Passo 5: *Repita os passos 2 a 4 até que toda a parte acessível de* G_{obs} *tenha sido construída.*
Passo 6: $X_{m,obs} = \{B \in X_{obs} : B \cap X_m \neq \emptyset\}$. □

Observação 2.2

1. *No algoritmo 2.1 foi suposto que a função* f *tenha sido estendida a sequências pertencentes a* E^* *na definição de* $UR(x)$.
2. *Note que* $f(x, \varepsilon) = x$ *e, portanto,* $UR(x) \neq \emptyset$, *para todo* $x \in X$.
3. *No caso de um autômato não-determinístico, o procedimento para a obtenção do observador é o mesmo, apenas substituindo-se os eventos não-observáveis pela sequência vazia* ε. □

Denotando por P_o 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 totalmente observado.
- $\mathcal{L}[Obs(G)] = P_o[\mathcal{L}(G)]$.
- $\mathcal{L}_m[Obs(G)] = P_o[\mathcal{L}_m(G)]$.

Exemplo 2.7 *(Autômato parcialmente observado) Considere o autômato mostrado na figura 2.9. O conjunto de eventos não-observáveis é*

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

Aplicando-se o algoritmo 2.1, obtém-se o observador $G_{obs} := Obs(G)$, *mostrado na figura 2.10, da seguinte forma: (i)* $x_{0,obs} = UR(x_0) = \{1\}$, $X_{obs} = \{\{1\}\}$ *e* $\tilde{X}_{obs} = \{\{1\}\}$*; (ii)* $\hat{X}_{obs} = \{\{1\}\}$ *e* $\tilde{X}_{obs} = \emptyset$*; (iii)* $\Gamma_{obs}(\{1\}) = \{a, b\}$, $f_{obs}(\{1\}, a) = UR(\{2\}) = \{2, 5\}$, $f_{obs}(\{1\}, b) = UR(\{3\}) = \{2, 3, 5\}$ *e* $\tilde{X}_{obs} = \{\{2, 5\}, \{2, 3, 5\}\}$*; (iv)* $X_{obs} = \{\{1\}, \{2, 5\}, \{2, 3, 5\}\}$. *Retorna-se ao passo 2 do algoritmo até que toda a parte acessível de* G_{obs} *seja construída. A título de ilustração, considere o estado* $\{6, 8, 9\}$. *Este estado representa o conjunto de possíveis estados em que o SED pode estar após a observação da sequência ac em* G_{obs}. □

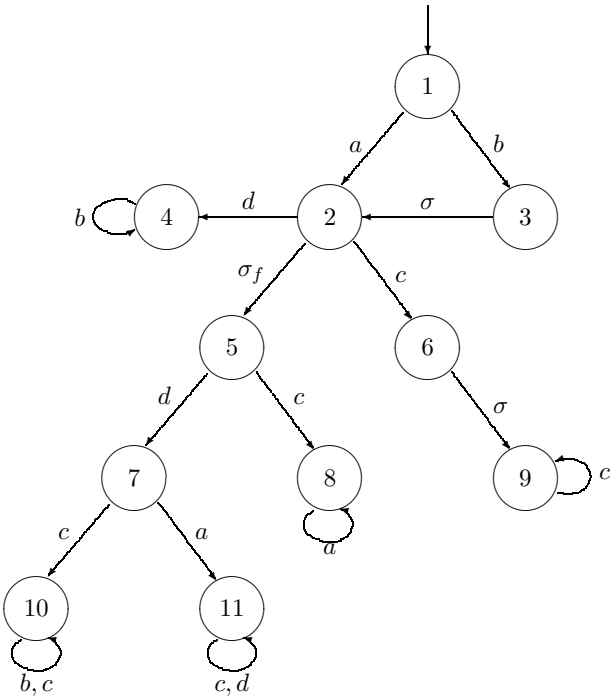

Figura 2.9: Autômato parcialmente observado do exemplo 2.7.

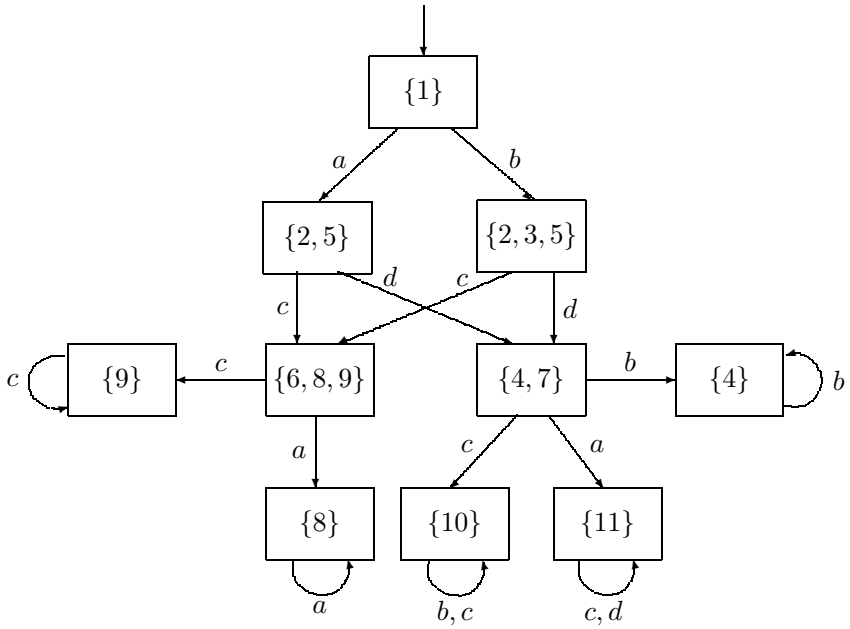

Figura 2.10: Observador do autômato da figura 2.9.

2.2.4 Obtenção da união de linguagens através de operações entre autômatos

No capítulo 4 deste trabalho, será necessário encontrar um autômato que gere uma linguagem obtida através da união de linguagens geradas por outros autômatos, *i.e.*, dados n autômatos diferentes G_1, G_2,..., G_n, que geram as linguagens L_1, L_2,..., L_n, deseja-se encontrar um autômato G_u tal que $\mathcal{L}(G_u) = \bigcup_{i=1}^{n} L_i$. Esse problema já foi abordado em CASSANDRAS e LAFORTUNE [2], e pode ser resolvido através da aplicação do seguinte algoritmo.

Algoritmo 2.2

Passo 1 *Construa o autômato G'_u da seguinte forma:*

1. *Crie o estado x_{0_u}.*
2. *Crie uma transição do estado x_{0_u} para cada um dos estados iniciais dos autômatos G_1, G_2,..., G_n e rotule-as com a sequência vazia ε.*
3. *Defina o estado x_{0_u} como estado inicial do autômato G'_u.*

Passo 2 *Faça $G_u = Obs(G'_u)$.*

Para ilustrar a aplicação do algoritmo 2.2, considere o exemplo a seguir.

Exemplo 2.8 *Considere os autômatos G_1, G_2 e G_3, mostrados na figura 2.11. Deseja-se encontrar o autômato G_u que gera a união das linguagens geradas pelos três autômatos citados. Para tanto, deve-se aplicar o algoritmo 2.2. No passo 1, constrói-se o autômato G'_u, mostrado na figura 2.12 em três etapas: (i) cria-se o estado x_{0_u}; (ii) cria-se uma transição de x_{0_u} para o estado inicial de cada um dos autômatos que se deseja obter a união das linguagens G_1, G_2 e G_3, que são: 0, A e x, e rotula-se-as com a sequência vazia ε; (iii) define-se o estado x_{0_u} como o estado inicial do autômato G'_u. O passo 2 consiste em calcular o observador do autômato G'_u, que é o próprio autômato G_u (mostrado na figura 2.12) que gera a união das linguagens $\mathcal{L}(G_1)$, $\mathcal{L}(G_2)$ e $\mathcal{L}(G_3)$, como se pode facilmente constatar.* □

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 modela 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.

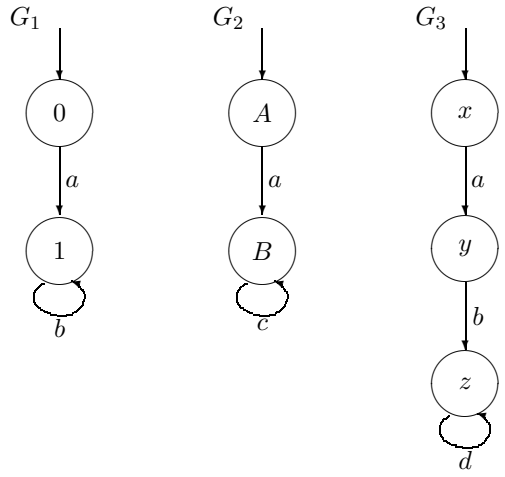

Figura 2.11: Autômatos G_1, G_2 e G_3 do exemplo 2.8.

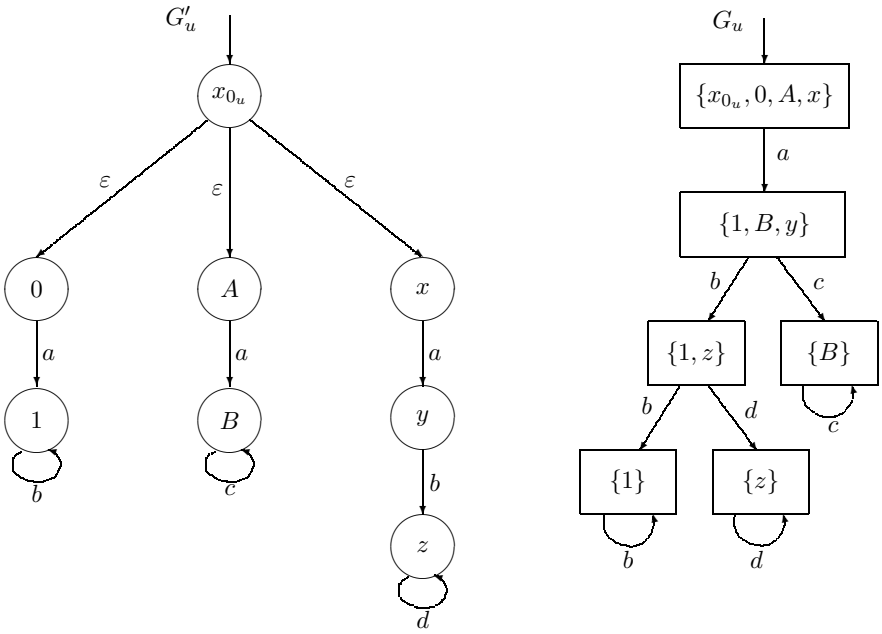

Figura 2.12: Autômatos G'_u e G_u do exemplo 2.8.

2.3 Diagnose de falhas

Em alguns casos pode ser necessário determinar a ocorrência de um ou mais eventos não-observáveis, em um SED parcialmente observado. Essa hipótese é motivada pela possibilidade de tal SED possuir um evento que representa uma falha do sistema. Nos casos mais simples, a falha é observada por um sensor, e portanto, é um evento observável e sua ocorrência pode ser detectada imediatamente. Porém, em muitos casos, não existe um sensor capaz de perceber a ocorrência dessa falha, e portanto, tal falha deve ser modelada através de um evento não-observável em um autômato parcialmente observado, por exemplo. Através da teoria de diagnóstico de falhas em SEDs, pode-se verificar se a ocorrência de uma falha em um SED pode ser detectada, ou não. Além disso, é possível construir um dispositivo capaz de informar a ocorrência (quando possível) de um determinado evento não-observável (falha). Este dispositivo é chamado de diagnosticador [3].

2.3.1 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. Nos trabalhos envolvendo diagnose de falhas em SED [1, 3], as seguintes hipóteses são feitas:

A1. A linguagem gerada por G é "viva", *i.e.*, $\Gamma(x_i) \neq \emptyset$ para todo $x_i \in X$;

A2. O autômato G não possui nenhum ciclo formado somente por eventos não-observáveis, *i.e.*, $\forall ust \in L$, $s \in E_{uo}^*$, $\exists n_0 \in \mathbb{N}$ tal que $\|s\| \leq n_0$, em que $\|s\|$ denota o comprimento da sequência s;

A3. Existe somente um único tipo de falha, *i.e.*, $E_f = \{\sigma_f\}$

A hipótese A2 é usualmente feita desde os primeiros trabalhos envolvendo diagnose de falhas em SED [3], porém, será removida no capítulo seguinte deste trabalho.

Antes de enunciar a definição de diagnosticabilidade, é necessário apresentar as definições abaixo.

Definição 2.11
A. *A linguagem de L após s, denotada por L/s, é definida como*

$$L/s = \{t \in E^* : st \in L\}.$$

B. *O operador projeção inversa em L, denotado por $P_{o_L}^{-1}$, é definido como*

$$P_{o_L}^{-1}(y) = \{s \in L : P_o(s) = y\},$$

em que a sequência $y \in E_o^*$.
C. *Suponha que* $\Psi(E_f)$ *denote o conjunto de todas as sequências de* L *que terminam com o evento* σ_f *associado à falha que se deseja diagnosticar. Formalmente, se* s_f *denota o último evento de uma sequência* s*, então,*

$$\Psi(E_f) = \{s \in L : s_f \in E_f\}.$$

Com um ligeiro abuso de notação, dada a sequência s*, a relação de pertinência* $E_f \in s$ *pode ser usada para denotar que* $\overline{s} \cap \Psi(E_f) \neq \emptyset$*, na qual* $\overline{s}$ *denota o fecho do prefixo de* s.
D. *(Sequência que contém uma falha) A sequência* $s \in L$ *é uma sequência que contém uma falha se* $E_f \in s$. □

Informalmente, diz-se que a linguagem gerada por um autômato é diagnosticável em relação a um conjunto de eventos observáveis E_o e um conjunto de eventos de falhas E_f se a ocorrência de qualquer evento de E_f puder ser detectada após um atraso finito da ocorrência dessa falha usando somente sequências de eventos observáveis. Formalmente, a diagnose de falhas é definida da seguinte forma [3].

Definição 2.12 *Seja* L *uma linguagem gerada por um autômato* G *e suponha que* L *seja viva e de prefixo fechado. Então* L *é diagnosticável com relação à projeção* P_o *e* $E_f = \{\sigma_f\}$ *se a seguinte condição for verificada:*

$$(\exists n \in \mathbb{N})(\forall s \in \Psi(E_f))(\forall t \in L/s)(\|t\| \geq n \Rightarrow D), \tag{2.1}$$

sendo a condição de diagnosticabilidade D *expressa por*

$$(\forall \omega \in P_{o_L}^{-1}(P_o(st)))(E_f \in \omega). \tag{2.2}$$

□

2.3.2 Diagnosticador

Um diagnosticador $G_d = (X_d, E_o, f_d, \Gamma_d, x_{0_d})$ é um observador modificado com o objetivo de indicar, se possível, a ocorrência de um dado evento não-observável (falha). A diferença entre um observador e um diagnosticador é o acréscimo, nesse último, de indicadores aos estados de G que formam os estados de G_{obs}, formando assim, os estados de G_d. 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). Serão utilizados dois tipos de marcações: N que indica que "o evento σ_f ainda não ocorreu" e Y para indicar que "o evento σ_f

ocorreu". A notação a ser utilizada para cada estado $x \in X$ será xN e/ou xY, respectivamente.

Dependendo de como as informações sobre a evolução dinâmica do sistema é disponibilizada, isto é, centralizada em um único sistema de aquisição ou distribuída como no caso de redes de comunicação, sistemas de manufaturas, e sistemas elétricos de potência, podem-se definir duas estruturas para a diagnose de falhas em SED:

1. Diagnosticador centralizado, que utiliza um único diagnosticador que tem acesso a todos os eventos observáveis do sistema;

2. Codiagnosticadores (diagnosticadores descentralizados) nos quais a leitura dos sensores não é centralizada, mas sim distribuída em diferentes módulos; cada módulo observando o comportamento de parte do sistema utilizando um subconjunto do conjunto de eventos observáveis do sistema.

O presente trabalho lidará somente com a diagnose centralizada de falhas. Para a construção do diagnosticador centralizado G_d, utiliza-se o seguinte algoritmo, que é uma modificação daquele utilizado para a construção de G_{obs} (algoritmo 2.1):

Algoritmo 2.3

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 sequência não-observável pertencente a $[E_{uo}\backslash\{\sigma_f\}]^*$, a partir de x_0;

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

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

Passo 2. *Na construção dos estados subsequentes de G_d:*

(a) Devem-se seguir as regras para a função de transição de G_{obs}, implementando-se as modificações apresentadas no passo 1.;

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

Passo 3. *G_d não possui estados marcados.* □

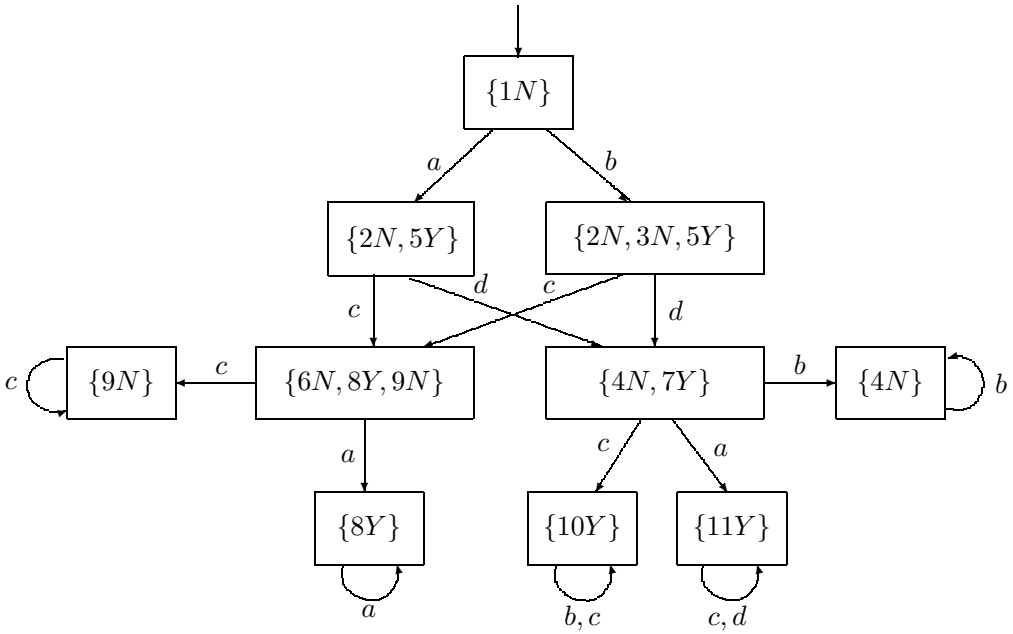

Figura 2.13: Diagnosticador do autômato da figura 2.9.

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

Exemplo 2.9 *(Diagnosticador) A figura 2.13 mostra o diagnosticador G_d do autômato G da figura 2.9, em que σ_f é o evento a ser diagnosticado. O diagrama de transição de estados de G_d mostra que após a ocorrência do terceiro evento observável de qualquer sequência, é possível afirmar se o evento σ_f ocorreu ou não, pois todos os componentes dos estados de G_d 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 σ_f.* □

Observação 2.3 *Pode-se construir o autômato G_d através da composição paralela de G com o autômato A_{label} mostrado na figura 2.14, seguido do cálculo do observador da maneira usual. Em outras palavras,*

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

Tendo em vista que $G_d = Obs(G \| A_{label})$, é fácil verificar que uma vez que o diagnosticador tiver certeza da ocorrência da falha, todos os estados seguintes permanecem indicando a falha. Contudo, é possível para um diagnosticador mudar de um estado de não-falha para duvidoso ou certo. Pode-se, então, classificar os estados do diagnosticador quanto à presença de rótulos Y e N [3] da seguinte forma,

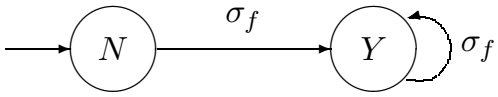

Figura 2.14: Autômato A_{label} de marcação de estados para a construção do diagnosticador

Definição 2.13 *Um estado $x_d \in X_d$ é denominado certo (de falha), se $\ell = Y$ para todo $(x, \ell) \in x_d$, e normal (ou de não-falha) se $\ell = N$ para todo $(x, \ell) \in x_d$. Se existir $(x, \ell), (y, \tilde{\ell}) \in x_d$, x não necessariamente distinto de y, tal que $\ell = Y$ e $\tilde{\ell} = N$, então x_d é um estado incerto de G_d.* □

Usando as definições 2.12 e 2.13, é possível estabelecer as seguintes relações entre os estados do diagnosticador e os sequências da linguagem gerada por G [3].

Lema 2.1
(i) *Seja $x_d = f_d(x_{0_d}, s)$, $s \in \mathcal{L}(G_d)$. Se x_d for um estado certo, então para todo $\omega \in P_{o_L}^{-1}(s)$, $E_f \in \omega$.*
(ii) *Se x_d for um estado incerto, então existem $s_1, s_2 \in L$ tais que $E_f \in s_1$ e $E_f \notin s_2$, porém $P_o(s_1) = P_o(s_2)$ e $f_d(x_{0_d}, P_o(s_1)) = f_d(x_{0_d}, P_o(s_2)) = x_d$.* □

Uma consequência imediata da definição 2.12 e do lema 2.1 é que a linguagem gerada por G será diagnosticável em relação a E_f e P_o se, e somente se, o diagnosticador sempre alcançar um estado certo para toda sequência arbitrariamente longa de L que contiver o evento σ_f. Isso não irá ocorrer se, e somente se, existir uma sequência de L que faça com que o diagnosticador fique preso indefinidamente em um laço formado por estados incertos. Para que resultados mais expressivos sobre esse problema possam ser enunciados, considere as seguintes definições [3].

Definição 2.14 *Seja $L = \mathcal{L}(G)$, $L \in E^*$, e considere que $L(G, x_1) = \{v \in E^* : [\exists uv \in L](f(x_0, u) = x_1)\}$. Um conjunto de estados $\{x_1, x_2, \ldots, x_n\} \subset X$ forma um ciclo em um autômato G se existir uma sequência $s = \sigma_1\sigma_2 \ldots \sigma_n \in L(G, x_1)$ tal que $f(x_l, \sigma_l) = x_{l+1}$, $l = 1, \ldots, n-1$, e $f(x_n, \sigma_n) = x_1$.* □

Definição 2.15 *Um conjunto de estados incertos $\{x_{d_1}, x_{d_2}, \ldots, x_{d_p}\} \subset X_d$ forma um ciclo indeterminado se as seguintes condições forem satisfeitas:*
1) $x_{d_1}, x_{d_2}, \ldots, x_{d_p}$ forma um ciclo em G_d, isto é, existem $\sigma_l \in E_o$, $l = 1, 2, \ldots, p$, tais que $f_d(x_{d_l}, \sigma_l) = x_{d_{l+1}}$, $l = 1, 2, \ldots, p-1$, e $f_d(x_{d_p}, \sigma_p) = x_{d_1}$;
2) $\exists (x_l^{k_l}, Y), (\tilde{x}_l^{r_l}, N) \in x_{d_l}$, $x_l^{k_l}$ não necessariamente distinto de $\tilde{x}_l^{r_l}$, $l = 1, 2, \ldots, p$, $k_l = 1, 2, \ldots, m_l$, e $r_l = 1, 2, \ldots, \tilde{m}_l$ de tal sorte que as sequências de estados $\{x_l^{k_l}\}$, $l = 1, 2, \ldots, p$, $k_l = 1, 2, \ldots, m_l$ e $\{\tilde{x}_l^{r_l}\}$, $l = 1, 2, \ldots, p$, $r_l = 1, 2, \ldots, \tilde{m}_l$ podem

ser rearranjadas para formar ciclos em G, cujas sequências correspondentes s e $\tilde{s}$, formados com os eventos que definem a evolução dos ciclos, têm como projeção $\sigma_1\sigma_2\ldots\sigma_p$, em que σ_1, σ_2,...,σ_p são definidos de acordo com o item 1. □

Utilizando as definições 2.12, 2.14 e 2.15 e o lema 2.1, pode-se enunciar a seguinte condição necessária e suficiente para a diagnose de uma linguagem.

Teorema 2.1 *Uma linguagem L gerada por um autômato G será diagnosticável em relação à projeção P_o e $E_f = \{\sigma_f\}$ se, e somente se, o seu diagnosticador G_d não tiver ciclos indeterminados.* □

A ocorrência do evento não-observável σ_f 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 G_d, possuírem a marcação N, então o evento σ_f ainda não ocorreu. Esse estado é denominado *estado normal* de G_d.

- Se todos os estados de G, pertencentes ao estado presente de G_d, possuírem a marcação Y, então o evento σ_f já ocorreu em algum instante no passado. Esse estado é chamado de *estado certo* de G_d. Se $t \in P_o[\mathcal{L}(G)]$ tiver sido observado e $f_d(x_{0_d}, t)$ for um estado certo, então todas as sequências pertencentes a $P_{o_L}^{-1}(t)$ devem conter σ_f.

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

Como exemplo, considere o autômato da figura 2.9 e seu diagnosticador, mostrado na figura 2.13. Note que G_d não possui ciclos indeterminados, podendo-se concluir que a linguagem L é diagnosticável com relação a P_o e E_f.

Observação 2.4 *A presença de ciclos em G_d formados somente por estados incertos não implica diretamente que L não seja diagnosticável com relação a P_o e E_f. Para que a impossiblidade de se diagnosticar uma falha seja caracterizada, é necessário que G possua um ciclo de estados formado após a ocorrência da falha, que seja correspondente ao ciclo de estados incertos em G_d [3].*

Neste ponto, a seguinte questão pode ser levantada: será possível fazer com que a linguagem L seja diagnosticável com relação a $P'_o : E^* \rightarrow E'^*_o$ e E_f, em que

$E'_o \subset E_o$? Se a resposta for afirmativa, será possível realizar a diagnose de falhas em um sistema modelado a eventos discretos com um número menor de sensores, e até mesmo utilizar diferentes conjuntos de sensores para detectar a mesma falha, agregando confiabilidade ao sistema de diagnose de falhas. Os resultados que serão apresentados no capítulo seguinte respondem a essa questão. Além disso, será proposto um algoritmo que pode ser utilizado para encontrar todos os conjuntos mínimos de eventos observáveis (sem redundâncias) que permitem que uma falha seja diagnosticada.

Capítulo 3

Bases mínimas para a diagnose de falhas em SEDs

Neste capítulo serão apresentadas condições necessárias e suficientes para que uma linguagem seja diagnosticável com relação a $P'_o : E^* \rightarrow E'^*_o$ e E_f, em que $E'_o \subset E_o$, bem como um algoritmo de construção do diagnosticador que utiliza somente os eventos pertencentes a E'_o para a diagnose da falha a partir do diagnosticador que utiliza os eventos pertencentes a E_o. Na sequência serão enunciados teoremas e definições de forma a embasar os passos do algoritmo de busca dos conjuntos mínimos de eventos observáveis que permitem a diagnose da falha num SED.

O presente capítulo está estruturado da seguinte forma: na seção 3.1 são apresentados os principais resultados obtidos para a diagnose centralizada com observação parcial [22], tais como, um algoritmo para construção do diagnosticador centralizado com observação parcial a partir do diagnosticador centralizado com observação total e uma condição necessária e suficiente para se verificar a diagnosticabilidade de uma linguagem $\mathcal{L}(G)$ com relação a $P'_o : E^* \rightarrow E'^*_o$ e E_f, em que $E'_o \subset E_o$, através da análise do diagnosticador parcial considerando-se E'_o como conjunto de eventos observáveis. Na seção 3.2 são introduzidos os conceitos de bases para a diagnose de falhas e bases mínimas para a diagnose de falhas em SEDs, bem como definições e teoremas necessários ao desenvolvimento de um algoritmo para a busca das bases mínimas para a diagnose de falhas em um SED. Na seção 3.3 é proposto um algoritmo para a busca das bases mínimas para a diagnose de falhas em SEDs utilizando-se os resultados obtidos na seção 3.2. Comentários finais sobre esse capítulo são feitos na seção 3.4.

3.1 Diagnose centralizada com observação parcial

A definição de diagnosticabilidade de uma linguagem L apresentada no capítulo 2 leva em consideração não somente a linguagem gerada, mas também o conjunto de eventos observáveis e a partição dos eventos de falha. A dependência da diagnosticabilidade em relação ao conjunto de eventos observáveis sugere que pode ser possível que uma linguagem L seja diagnosticável com relação a $P'_o : E^* \rightarrow E'^*_o$ e E_f, em que $E'_o \subset E_o$. Esse problema é denominado diagnose centralizada com observação parcial. Para que se possa abordar esse problema, ao lado das hipóteses A1-A3 feitas no capítulo anterior, a seguinte hipótese deve ser considerada:

A4. L é diagnostícavel com relação a $P_o : E^* \rightarrow E^*_o$ e E_f.

Seja $G'_d = (X'_d, E'_o, f'_d, \Gamma'_d, x'_{0_d})$ um diagnosticador supondo observação parcial, *i.e.*, G'_d é capaz de observar somente os eventos pertencentes a $E'_o \subset E_o$; por essa razão, G'_d será referido como diagnosticador centralizado com observação parcial ou simplesmente diagnosticador parcial. Com isso, o seguinte resultado, cuja demonstração será omitida nesse trabalho, pode ser enunciado [22]

Teorema 3.1 *Sejam $G_d = (X_d, E_o, f_d, \Gamma_d, x_{0_d})$ e $G'_d = (X'_d, E'_o, f'_d, \Gamma'_d, x'_{0_d})$ diagnosticadores supondo observação total e parcial, respectivamente,* i.e., *$E'_o \subset E_o$ e $E'_o \neq \emptyset$. Então, $Obs(G_d, E'_o) = (\hat{X}_d, E'_o, \hat{f}_d, \hat{\Gamma}_d, \hat{x}_{0_d})$ (o observador de G_d com relação a projeção $P_{oo'} : E^*_o \rightarrow E'^*_o$) e G'_d são iguais considerando-se a seguinte equivalência de estados:*

$$\hat{x}_d = \{x_{d_1}, x_{d_2}, \ldots, x_{d_n}\} \in \hat{X}_d, x_{d_i} \in X_d \Leftrightarrow x'_d = \bigcup_{i=1}^{n} x_{d_i} \in X'_d.$$

Além disso, $f'_d = \hat{f}_d$. □

De acordo com o teorema 3.1, o diagnosticador parcial G'_d que observa eventos pertencentes ao subconjunto E'_o do conjunto dos eventos observáveis E_o pode ser construído diretamente a partir do diagnosticador centralizado G_d calculando-se $Obs(G_d, E'_o)$ e substituindo-se cada um dos estados de $Obs(G_d, E'_o)$ pela união dos conjuntos que formam cada um desses estados. Além disso, como $\mathcal{L}(G'_d) = \mathcal{L}(Obs(G_d, E'_o)) = P_{oo'}(G_d)$, sendo $P_{oo'} : E^*_o \rightarrow E'^*_o$, então, embora a linguagem gerada pelo diagnosticador centralizado com observação total seja, por hipótese, viva, a linguagem gerada por um diagnosticador parcial não é necessariamente viva. Isso ocorre sempre que os eventos que formam um ciclo em G_d se tornam não-observáveis no diagnosticador parcial; não é difícil verificar que quando isso acontece, esse ciclo se reduz a um único estado em G'_d. Quando, após o sistema alcançar um determinado estado, ocorrer um ciclo de estados ligados por eventos não-observáveis, não haverá

mudança de estado no diagnosticador parcial, embora os estados reais do autômato mudem de forma cíclica. Nesse caso, diz-se que esse diagnosticador parcial possui um ciclo escondido nesse estado. Considere a seguinte definição [22].

Definição 3.1 *(Ciclos escondidos e ciclos escondidos indeterminados) Suponha que* $x'_d \in X'_d$ *tenha sido obtido agrupando-se os estados* $x_{d_1}, x_{d_2}, \ldots, x_{d_n} \in X_d$*. Então existe um ciclo escondido em* x'_d *se para algum* $\{i_1, i_2, \ldots, i_k\} \subseteq \{1, 2, \ldots, n\}$, $\{x_{d_{i_1}}, x_{d_{i_2}}, \ldots, x_{d_{i_k}}\}$ *forma um ciclo em* G_d*. Além disso, se* x'_d *é um estado incerto e todos os estados* $x_{d_{i_1}}, x_{d_{i_2}}, \ldots, x_{d_{i_k}}$ *são certos, então o ciclo escondido é denominado indeterminado.* □

Observação 3.1 *De agora em diante no texto, os ciclos indeterminados que não são escondidos serão referidos como ciclos observados indeterminados.* □

Em função da hipótese A4, não é difícil perceber que os estados $x_{d_{i_k}}$ que formam um ciclo escondido em x'_d devem ser todos normais, certos ou incertos que não formam um ciclo indeterminado. Os ciclos escondidos serão representados nos diagramas de transição de estados dos diagnosticadores parciais por laços tracejados: os ciclos escondidos indeterminados serão rotulados como *ihc* (do inglês *indeterminate hidden cycle*) e os ciclos escondidos em estados certos, em estados normais ou em estados incertos em que o ciclo escondido é formado por estados normais ou por estados incertos de G_d que não formam um ciclo indeterminado serão rotulados simplesmente como *hc*, uma vez que, conforme será visto mais a frente, eles não levam a perda de diagnosticabilidade quando se tem observação parcial dos eventos.

De acordo com a condição para diagnose dada na definição 2.12, todas as sequências $s \in \Psi(E_f)$ devem ser diagnosticadas pelo diagnosticador parcial G'_d. Uma sequência $s \in \Psi(E_f)$ que não pode ser diagnosticada pelo diagnosticador parcial é chamada de sequência ambígua [3].

Definição 3.2 *(Sequência ambígua) Uma sequência* $s \in L$ *é uma sequência ambígua em relação à projeção* $P'_o : E^* \rightarrow E'^*_o$ *e* E_f *se existir uma sequência* $s' \in L$ *tal que* $E_f \in s$*, porém* $E_f \notin s'$*, e* $P'_o(s) = P'_o(s')$*.* □

O teorema a seguir provê uma condição necessária e suficiente para a diagnose de falhas sob observação parcial [22].

Teorema 3.2 *Suponha que a linguagem* L *seja diagnosticável em relação à projeção* P_o *e* E_f*. Então* L *será também diagnosticável em relação à projeção* $P'_o : E^* \rightarrow E'^*_o$*,* $E'_o \subset E_o$*, e* $E_f = \{\sigma_f\}$ *se, e somente se,* G'_d *não tiver nenhum ciclo indeterminado (observados e escondidos).*

Demonstração: Uma condição necessária e suficiente para a diagnose foi obtida no teorema 2.1 para o caso em que G_d não possui ciclos escondidos. Note que o diagnosticador parcial G'_d pode ser visto como um diagnosticador centralizado com uma nova partição do conjunto de eventos $E = E'_o \dot{\cup} E'_{uo}$. Logo, se G'_d não possuir ciclos escondidos, então o teorema 2.1 se aplicará a ele. Falta, então, considerar o caso em que G'_d possui ciclos escondidos.

($\Rightarrow$) Seja $x'_{d_{unc}} = UR(x_{d_{unc}}, E_o \setminus E'_o) \in X'_d$ para algum $x_{d_{unc}} \in X_d$, e suponha que, para $k = 1, \ldots, n$, os estados $x^{(k)}_{d_{cert}} \in x'_{d_{unc}}$ formam um ciclo escondido indeterminado em $x'_{d_{unc}}$. Não é difícil verificar que, nessas condições, existe uma sequência $stu_k \in L$ que satisfaz as seguintes condições:

1) $s \in \Psi(E_f)$ e $f_d(x_{0_d}, P_o(s)) = x_{d_{unc}}$;

2) $t \in (E_o \setminus E'_o)^*$, $\|t\| > n_t$, em que n_t pode ser arbitrariamente grande, tal que $f_d(x_{0_d}, P_o(st)) = x^{(1)}_{d_{cert}}$;

3) $u_k \in (E_o \setminus E'_o)^*$, $\|u_k\| = k \geq 0$, e $f_d(x^{(1)}_{d_{cert}}, u_k) = x^{(k+1)}_{d_{cert}}$, $k = 1, \ldots, n-1$ e $f_d(x^{(1)}_{d_{cert}}, u_n) = x^{(1)}_{d_{cert}}$;

Portanto, é fácil concluir que $f'_d(x'_{0_d}, P'_o(stu_k)) = x'_{d_{unc}}$, o que implica que, uma vez que $f_d(x_{0_d}, P_o(s)) = x_{d_{unc}}$, então existe $w \in P'^{-1}_{o_L}(stu_k)$ tal que $E_f \notin w$, o que viola a condição para diagnose.

($\Leftarrow$) Suponha que a linguagem L seja não-diagnosticável em relação à projeção P'_o e E_f. Então baseado na definição 2.12, tem-se que $\exists (s,t) \in (\Psi(E_f) \times L/s)$, $\|t\|$ arbitrariamente longo e $\exists \omega \in P'^{-1}_{o_L}(P'_o(st))$ tal que $E_f \notin \omega$.

É fácil concluir que w pode ser de comprimento arbitrariamente longo ou finito. O primeiro caso é abordado pelo teorema 2.1 e o segundo caso será tratado a seguir. Suponha que $f_d(x_{0_d}, P_o(w)) = x_{d_{norm}}$ seja um estado normal em G_d e exista um ciclo $\{x^{(1)}_{d_{cert}}, x^{(2)}_{d_{cert}}, \ldots, x^{(n)}_{d_{cert}}\}$ ao longo de st em G_d de estados certos. Como $f'_d(x'_{0_d}, P'_o(st)) = f'_d(x'_{0_d}, P'_o(w)) = x'_d$, então x'_d é um estado incerto em G'_d. Se $t \in (E_o \setminus E'_o)^*$, então $x'_d = \{x_{d_{norm}}, x^{(1)}_{d_{cert}}, x^{(2)}_{d_{cert}}, \ldots, x^{(n)}_{d_{cert}}\}$, o que, pela definição 3.1, define um ciclo escondido indeterminado. □

Exemplo 3.1 *Para ilustrar o resultado do teorema 3.2, considere o autômato* $G = (X, E, f, \Gamma, x_0, X_m)$ *cujo diagrama de transição de estados está representado na figura 3.1. Suponha que* $E = \{a, b, c, d, \sigma, \sigma_f\}$, $E_o = \{a, b, c, d\}$, $E_{uo} = \{\sigma, \sigma_f\}$ *e* $E_f = \{\sigma_f\}$. *O diagnosticador* G_d *associado a* G *pode ser visto na figura 3.2(a), de onde se pode notar que* L *pode ser diagnosticada em relação a* P_o *e* E_f, *uma vez que* G_d *não tem nenhum ciclo indeterminado. Considere agora o problema de verificar se* L *é também diagnosticável em relação à projeção* $P'_o : E^* \rightarrow E'^*_o$ *e* E_f, *sendo* $E'_o = \{c, d\} \subset E_o$. *O diagnosticador parcial* G'_d *correspondente ao conjunto de eventos observáveis* E'_o *está representado na figura 3.2(b), de onde se pode ver que* G'_d *tem um ciclo escondido indeterminado no estado* $\{3N, 4N, 6Y\}$. *Como consequência,* L *é não diagnosticável em relação a* P'_o *e* E_f.

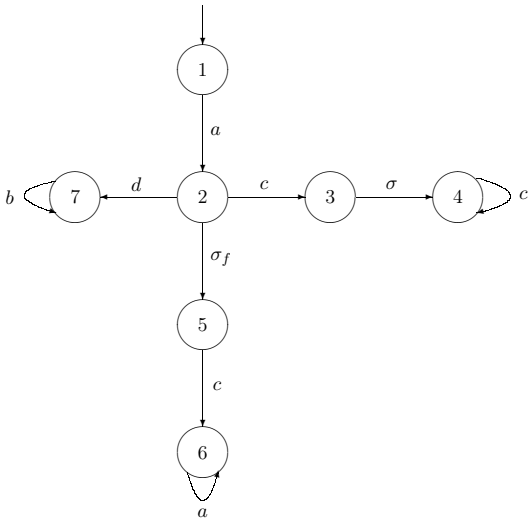

Figura 3.1: Autômato G cuja ocorrência do evento σ_f deve ser diagnosticada.

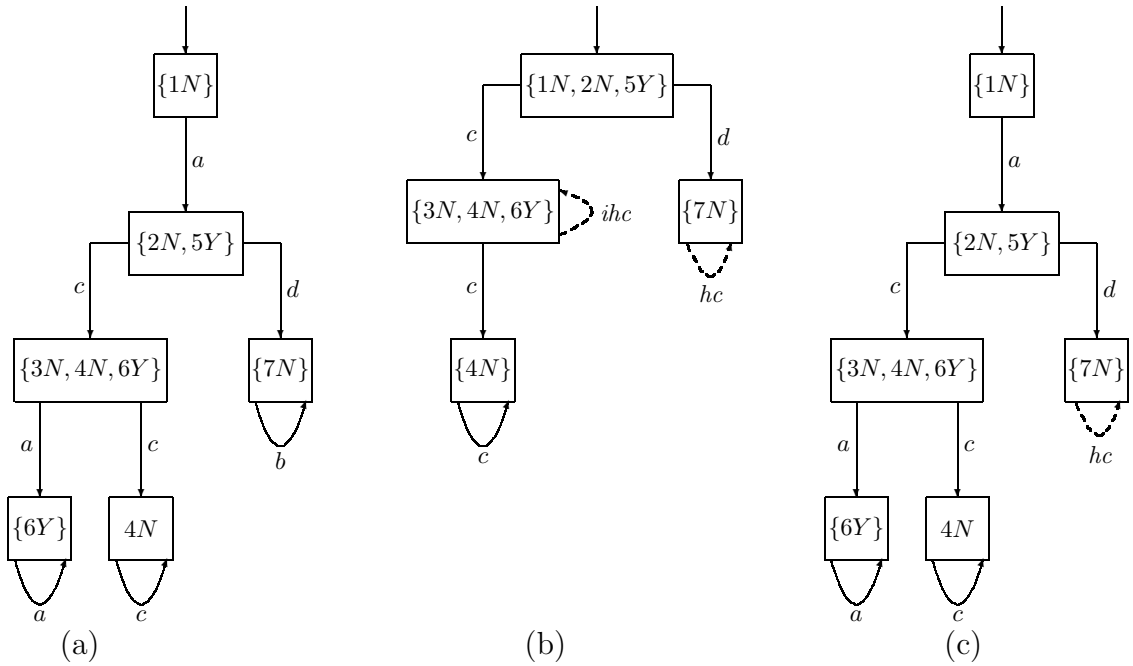

Figura 3.2: Diagnosticador G_d (a) e os diagnosticadores parciais G'_d (b) e G''_d (c) para os conjuntos de eventos observáveis $E'_o = \{c, d\}$ e $E''_o = \{a, c, d\}$, respectivamente.

A justificativa para a não diagnose de L com relação a P'_o é a existência da sequência $s = a\sigma_f ca^n$, $n \in \mathbb{N}$, que contém o evento de falha σ_f, e que possui a mesma projeção em relação a P'_o que uma sequência normal $s' = ac$, isto é, $P'_o(s) = P'_o(s') = c$; consequentemente s é uma sequência ambígua (s é, na verdade, a única sequência ambígua nesse exemplo). Para se obter um novo subconjunto de E_o que faça com que L seja diagnosticável com observação parcial, note que $a \in s$, porém $a \notin E'_o$. Dessa forma, acrescentando-se o evento a ao conjunto de eventos observáveis E'_o, e formando um novo conjunto de eventos observáveis $E''_o = \{a, c, d\}$, é esperado que L se torne diagnosticável em relação $P''_o : E^* \rightarrow E''^*_o$ e E_f. Isso, de fato, acontece, uma vez que G''_d não possui ciclos indeterminados (observados e escondidos), conforme pode ser visto na figura 3.2(c). □

3.2 Bases para a diagnose centralizada de falhas

3.2.1 Conjuntos de eventos elementares para a diagnose

O resultado apresentado no teorema 3.2 traz as seguintes questões: é possível encontrar diferentes subconjuntos do conjunto de eventos observáveis que também permitem que a linguagem gerada por um autômato seja diagnosticável? Qual o subconjunto de eventos observáveis de menor cardinalidade necessário para a diagnose de falhas centralizada? Será possível, de forma sistemática, encontrar todos os subconjuntos E'_o do conjunto de eventos observáveis capazes de permitir a diagnose de uma linguagem gerada por um autômato?

De forma a responder as questões postas acima, considere o autômato $G = (X, E, f, \Gamma, x_0, X_m)$, em que $E = E_o \cup E_{uo}$, e suponha que a linguagem L gerada por G seja diagnosticável em relação a $P_o : E^* \rightarrow E_o^*$ e $E_f = \{\sigma_f\}$. Além disso, suponha que $E'_o \subset E_o$, $E'_o \neq \emptyset$, e que G_d e G'_d denotem, respectivamente, os diagnosticadores centralizados para G supondo observação total e parcial de eventos observáveis. Nesse ponto, as seguintes definições são necessárias.

Definição 3.3 *(Bases para a diagnose) O conjunto E'_o é uma base para a diagnose se L for diagnosticável com relação a $P'_o : E^* \rightarrow E'^*_o$ e $E_f = \{\sigma_f\}$.* □

Definição 3.4 *(Bases mínimas para a diagnose) O conjunto E'_o é uma base mínima para a diagnose se E'_o é uma base para a diagnose e se, para todo subconjunto não vazio E''_o de E'_o, L não for diagnosticável com relação a $P''_o : E^* \rightarrow E''^*_o$ e $E_f = \{\sigma_f\}$.* □

Utilizando as definições 3.3 e 3.4, o problema de se encontrar todos os conjuntos $E'_o \subset E_o$ para o qual L é diagnosticável com relação a $P'_o : E^* \rightarrow E'^*_o$ pode também

ser formulado da seguinte forma: dado um autômato $G = (X, E, f, \Gamma, x_0, X_m)$, em que $E = E_o \dot{\cup} E_{uo}$, e supondo que E_o seja uma base para a diagnose, encontre todos os conjuntos $E'_o \in 2^{E_o} \setminus \{E_o, \emptyset\}$ que também sejam bases para a diagnose de L. É importante ressaltar que esse problema já foi considerado em outros contextos que não SED. Especificamente, o trabalho de TRAVÉ-MASSUYÈS *et al.* [32], apresenta métodos para a análise da diagnosticabilidade de um sistema, indicando os sensores que devem ser incluídos para alcançar um grau de diagnosticabilidade desejado. Isso é feito através da caracterização de diferentes propriedades da diagnosticabilidade de um sistema, propondo-se um método que busca um modelo de forma a alcançar o grau de discriminabilidade desejado, *i.e.*, dado um conjunto de sensores e o número de falhas que podem ser discriminadas, determina-se o conjunto mínimo de sensores adicionais que garantem o grau de diagnosticabilidade desejado. Entretanto, em TRAVÉ-MASSUYÈS *et al.* [32], a busca é feita de uma forma exaustiva.

De acordo com as definições 3.3 e 3.4, a diferença principal entre uma base para a diagnose e uma base mínima para a diagnose é com relação à natureza dos eventos. Os eventos de uma base mínima são todos essenciais, no sentido de que a falta de um dos eventos dessa base implica na perda de diagnosticabilidade da linguagem gerada. Por outro lado, uma base para a diagnose (não mínima) possui eventos redundantes, no sentido de que nem todos os eventos da base são necessários para diagnosticar a ocorrência da falha. Essas observações são formalizadas na definição a seguir.

Definição 3.5 *(Conjuntos de eventos redundantes) Suponha que o conjunto E'_o seja uma base para a diagnose e que $E'_{o_i} \subset E'_o$, $i = 1, 2, \ldots, N_b$, denote todas as bases mínimas para a diagnose que podem ser formadas com os eventos pertencentes à E'_o. Então, o conjunto formado por todos os conjuntos de eventos redundantes de E'_o é*

$$E_{red}(E'_o) = \{E'_o \setminus E'_{o_1}, E'_o \setminus E'_{o_2}, \ldots, E'_o \setminus E'_{o_{N_b}}\}.$$

□

Para ilustrar a definição 3.5 considere o seguinte caso. Suponha que um autômato G possua um conjunto de eventos observáveis $E'_o = \{a, b, c, d, e\}$, sendo este conjunto uma base para a diagnose de falhas da linguagem L gerada por G. Além disso, suponha que $E'_{o_1} = \{a, b, c\}$ e $E'_{o_2} = \{c, d, e\}$ sejam as únicas bases mínimas para a diagnose de L. Portanto, o conjunto formado pelos conjuntos de eventos redundantes de E'_o é $E_{\text{red}}(E'_o) = \{\{d, e\}, \{a, b\}\}$.

Suponha, agora, que $x_{d_{YN}}, x_{d_Y}, x_{d_N} \in X_d$ denotem, respectivamente, estados incertos, certos e normais de G_d. Sob a hipótese de que L é diagnosticável com

relação a P_o e E_f, é sempre possível definir o seguinte subconjunto de X_d:

$$X_{YN}^{Y} = \{x_{d_{YN}} \in X_d : (\exists x_{d_Y} \in X_d \wedge \exists \sigma \in E_o)[f_d(x_{d_{YN}}, \sigma) = x_{d_Y}]\}.$$

Note que para cada estado de X_{YN}^{Y} é sempre possível definir, pelo menos, uma trajetória $P_Y = (x_{d_{YN}}, \sigma_0, x_{d_{Y,1}}, \sigma_1, \ldots, \sigma_{n-1}, x_{d_{Y,n}})$ que satisfaz às seguintes condições: *(i)* $x_{d_{Y,n}} = x_{d_{Y,i}}$ para algum $i \in \{1, 2, \ldots, n-1\}$, *i.e.*, os estados $(x_{d_{Y,i}}, x_{d_{Y,i+1}}, \ldots, x_{d_{Y,n}})$ formam um ciclo; *(ii)* $(x_{d_{Y,i}}, x_{d_{Y,i+1}}, \ldots, x_{d_{Y,n}})$ é o único ciclo existente na trajetória. O conjunto X_{YN}^{Y} será referido como conjunto de estados-origem de trajetórias de falha (CEOTF) e a trajetória P_Y como trajetória de falha. Os elementos de X_{YN}^{Y} são chamados estados-origem de trajetórias de falha (ou simplesmente estados-origem, quando o contexto assim o permitir).

Definição 3.6 *(Evento de uma trajetória de falha, conjunto de eventos de uma trajetória de falha)*
A. Um evento $\sigma \in E_o$ é um evento de uma trajetória de falha se σ pertence a qualquer trajetória de falha definida para algum estado pertencente à X_{YN}^{Y}.
B. Um conjunto de eventos de uma trajetória de falha (CETF), denotado por E_{etf}, é um conjunto formado por todos os eventos de uma trajetória de falha. □

A definição de conjunto de eventos de uma trajetória de falha permite que se chegue à uma condição necessária para que um conjunto $E_o' \subset E_o$ seja uma base para a diagnose, como demonstrado abaixo.

Teorema 3.3 *Suponha que N_{cetf} denote o número de conjuntos de eventos de uma trajetória de falha de G_d. Então, uma condição necessária para que $E_o' \subset E_o$ seja uma base para a diagnose de L (linguagem gerada por G) e $E_f = \{\sigma_f\}$ é que*

$$E_o' \bigcap E_{etf,i} \neq \emptyset, i = 1, 2, \ldots, N_{cetf}. \tag{3.1}$$

Demonstração: Suponha que E_o' seja uma base para a diagnose e que, para algum $k \in \{1, 2, \ldots, N_{cetf}\}$, $E_{etf,k} \cap E_o' = \emptyset$. Então, para algum $x_{d_{YN}} \in X_{YN}^{Y}$, existirá uma trajetória de falha $P_Y^k = (x_{d_{YN}}, \sigma_0^k, x_{d_{Y,1}}^k, \sigma_1^k, \ldots, \sigma_{n-1}^k, x_{d_{Y,n}}^k)$, satisfazendo $x_{d_{Y,j}}^k = x_{d_{Y,n}}^k$ para algum $j \in \{1, 2, \ldots, n-1\}$. É fácil verificar que $x_{d_{Y,j}}^k, x_{d_{Y,j+1}}^k, \ldots, x_{d_{Y,n}}^k$ forma um ciclo escondido indeterminado no estado $x'_{d_{YN}} \in X_d'$ que contém $UR(x_{d_{YN}}, E_o \setminus E_o')$, que, de acordo com o teorema 3.2, implica que L é não-diagnosticável com relação a $P_o' : E^* \rightarrow E_o'^*$ e $E_f = \{\sigma_f\}$; contradizendo assim, a hipótese de que E_o' é uma base para a diagnose. □

Observação 3.2 *Note que a condição imposta pelo teorema 3.3 não é suficiente. Como será esclarecido em exemplos apresentados mais à frente, é possível que a*

condição (3.1) seja satisfeita, mas que E'_o não seja uma base para a diagnose. A condição necessária e suficiente para que E'_o seja uma base para a diagnose foi apresentada no teorema 3.2. □

Está claro que, para que a ocorrência da falha seja diagnosticada, pelo menos um evento de cada trajetória de falha deve ser observável. Este fato leva à definição de conjuntos de eventos elementares para a diagnose.

Definição 3.7 *(Conjunto de eventos elementares para a diagnose) Suponha que $E_{etf,i}$, $i = 1, \ldots, N_e$ seja um conjunto de eventos de uma trajetória de falha de G_d. O conjunto de todos os conjuntos de eventos elementares para a diagnose de G_d é definido como se segue:*

$$\begin{aligned} E_{eed} \;=\; & \{E_e = E_{e,1} \cup E_{e,2} \cup \ldots \cup E_{e,N_e} : (E_{e,1}, E_{e,2}, \ldots, E_{e,N_e}) \in 2_1^{E_{etf,1}} \times \\ & \times 2_1^{E_{etf,2}} \times \ldots \times 2_1^{E_{etf,N_e}}\}, \end{aligned}$$

em que $2_1^{E_{etf,i}} = \{E_e \in 2^{E_{etf,i}} : |E_e| = 1\}$. □

O algoritmo abaixo sugere uma forma sistemática de se encontrar todos os conjuntos de eventos elementares para a diagnose de G_d.

Algoritmo 3.1

Passo 1 *Construa o diagnosticador centralizado G_d e encontre o conjunto de estados-origem de trajetórias de falha (X_{YN}^Y) de G_d. Defina $|X_{YN}^Y| = N_{YN}$.*

Passo 2 *Para cada estado-origem $x_{d_{YN,i}} \in X_{YN}^Y$, $i = 1, 2, \ldots, N_{YN}$ construa uma árvore*[1] *com raiz $x_{d_{YN,i}}$, como se segue:*
(i) Defina $\Gamma_d^Y(x_{d_{YN,i}}) = \{\sigma \in \Gamma_d(x_{d_{YN,i}}) : f_d(x_{d_{YN,i}}, \sigma) = x_{d_Y}\}$ e suponha que $|\Gamma_d^Y(x_{d_{YN,i}})| = n_{YN,i}$. Crie $n_{YN,i}$ descendentes de $x_{d_{YN,i}}$ e rotule-os como x_{d_Y}, em que $x_{d_Y} = f_d(x_{d_{YN,i}}, \sigma)$, $\sigma \in \Gamma_d^Y(x_{YN,i})$. Rotule os ramos $(x_{d_{YN,i}}, x_{d_Y})$ como σ;
(ii) Um nó rotulado como x_{d_Y}, definido na árvore, será uma folha se o estado x_{d_Y} já tiver rotulado algum ancestral de x_{d_Y}. Caso contrário, defina $|\Gamma_d(x_{d_Y})| = n_Y$. Crie n_Y descendentes de x_{d_Y} e rotule-os como $x_{d_{Y,new}}$, em que $x_{d_{Y,new}} = f_d(x_{d_Y}, \sigma)$, $\sigma \in \Gamma_d(x_{d_{YN,i}})$. Rotule os ramos $(x_{d_Y}, x_{d_{Y,new}})$ como σ.

Passo 3 *Para cada árvore T_i, $i = 1, 2, \ldots, N_{YN}$, identifique suas folhas $x_{d_{Y,i}}^\ell$, $\ell = 1, \ldots, L_i$, em que L_i é o número de folhas da árvore T_i. Forme trajetórias $P_{Y,i}^\ell$,*

[1] De forma rigorosa, o grafo a ser construído no Algorithm 3.1 não é uma árvore pois nós distintos podem ter o mesmo rótulo de marcação. A principal razão para a marcação de dois nós diferentes com o mesmo rótulo vem do fato que o diagnosticador possui ciclos e, portanto, é possível que exista mais de uma trajetória que leva de um estado-origem a um estado certo.

$\ell = 1, \ldots, L_i$, *iniciando em* $x_{d_{YN,i}}$ *e terminando em* $x^{\ell}_{d_{Y,i}}$, $\ell = 1, \ldots, L_i$ *(essas trajetórias são, na verdade, as trajetórias de falha que se iniciam em* $x_{d_{YN,i}}$*).*

Passo 4 *Forme os conjuntos de eventos de uma trajetória de falha (CETFs)* $E^{\ell}_{etf,i}$, $i = 1, \ldots, N_{YN}$, $\ell = 1, \ldots, L_i$ *utilizando as trajetória* $P^{\ell}_{Y,i}$ *obtidas no passo anterior.*

Passo 5 *Com os CETFs obtidos no passo 4, construa o conjunto formado pelos conjuntos de eventos elementares para a diagnose, de acordo com a equação (3.2).* □

O exemplo apresentado abaixo ilustra o cálculo de todos os conjuntos de eventos elementares para a diagnose de falhas a partir um dado diagnosticador centralizado.

Exemplo 3.2 *Considere o autômato* $G = (X, E, f, \Gamma, x_0, X_m)$ *mostrado na Fig. 3.3(a) e suponha que* $E_o = \{a, b, c, d, e\}$ *e* $E_f = \{\sigma_f\}$*. Pode-se notar que o diagnosticador centralizado* G_d*, mostrado na Figura 3.3(b), não possui ciclos indeterminados e, portanto, a linguagem* L *gerada por* G *é diagnosticável com relação a* P_o *e* E_f*. De acordo com o Algoritmo 3.1, para encontrar os conjuntos de eventos elementares para a diagnose, o primeiro passo é identificar os estados-origem de* G_d*. Da Figura 3.3(b) pode-se concluir que* $X^{Y}_{YN} = \{x_{d_{YN,1}}, x_{d_{YN,2}}, x_{d_{YN,3}}\}$*, em que* $x_{d_{YN,1}} = \{1N, 2Y\}$*,* $x_{d_{YN,2}} = \{4Y, 5N\}$ *e* $x_{d_{YN,3}} = \{3Y, 5N\}$*. O próximo passo do Algoritmo 3.1 é construir uma árvore para cada estado-origem acima, que estão mostradas na Figura 3.4. Baseado nessas árvores, é possível, como descrito no passo 3 do Algoritmo 3.1, identificar suas folhas, e na sequência formar as trajetórias que se iniciam na raíz e que terminam nas folhas. Em particular, a árvore da Figura 3.4(b) possui 6 folhas, que definem as seguintes trajetórias:* $P^1_{Y,2} = (\{4Y, 5N\}, c, \{3Y\}, b, \{4Y\}, c, \{3Y\})$*,* $P^2_{Y,2} = (\{4Y, 5N\}, c, \{3Y\}, b, \{4Y\}, d, \{4Y\})$*,* $P^3_{Y,2} = (\{4Y,\ 5N\}, c, \{3Y\}, a, \{6Y\}, b, \{3Y\})$*,* $P^4_{Y,2} = (\{4Y, 5N\}, d, \{4Y\}, d, \{4Y\})$*,* $P^5_{Y,2} = (\{4Y, 5N\}, d, \{4Y\},\ c, \{3Y\}, b, \{4Y\})$*, e* $P^6_{Y,2} = (\{4Y, 5N\}, d, \{4Y\}, c, \{3Y\},\ a, \{6Y\}, b, \{3Y\})$*. Procedendo desta forma, 6 outras trajetórias podem ser obtidas a partir das árvores das Figuras 3.4(a) e (c). Portanto, os conjuntos de eventos de uma trajetória de falha (CETFs) de* G_d *são dados por:* $E^1_{etf,1} = \{d, e\}$*,* $E^2_{etf,1} = \{b, c, e\}$*,* $E^3_{etf,1} = \{a, b, c, e\}$*,* $E^1_{etf,2} = \{b, c\}$*,* $E^2_{etf,2} = \{b, c, d\}$*,* $E^3_{etf,2} = \{a, b, c\}$*,* $E^4_{etf,2} = \{d\}$*,* $E^5_{etf,2} = \{b, c, d\}$*,* $E^6_{etf,2} = \{a, b, c, d\}$*,* $E^1_{etf,3} = \{a, b\}$*,* $E^2_{etf,3} = \{a, b, c\}$*, e* $E^3_{etf,3} = \{a, b, d\}$*. Finalmente, procedendo de acordo com o* 5^o *passo do Algoritmo 3.1, os seguintes conjuntos de eventos elementares para a diagnose (CEEDs) são obtidos:*

$$\begin{aligned} E_{eed} \;=\; & \{\{a, b, d\}, \{a, b, c, d\}, \{b, d\}, \{b, c, d\}, \{a, b, d, e\}, \{a, b, c, d, e\}, \{b, d, e\}, \\ & \{b, c, d, e\}, \{a, c, d\}, \{a, c, d, e\}\}. \end{aligned} \tag{3.2}$$

□

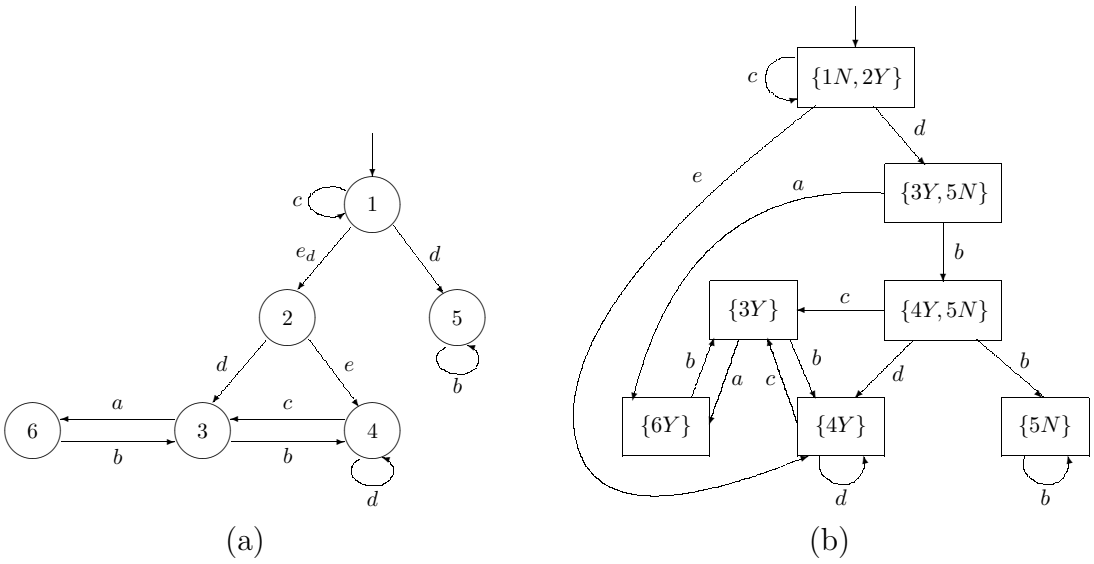

Figura 3.3: Autômato e correspondente diagnosticador centralizado do exemplo 3.2.

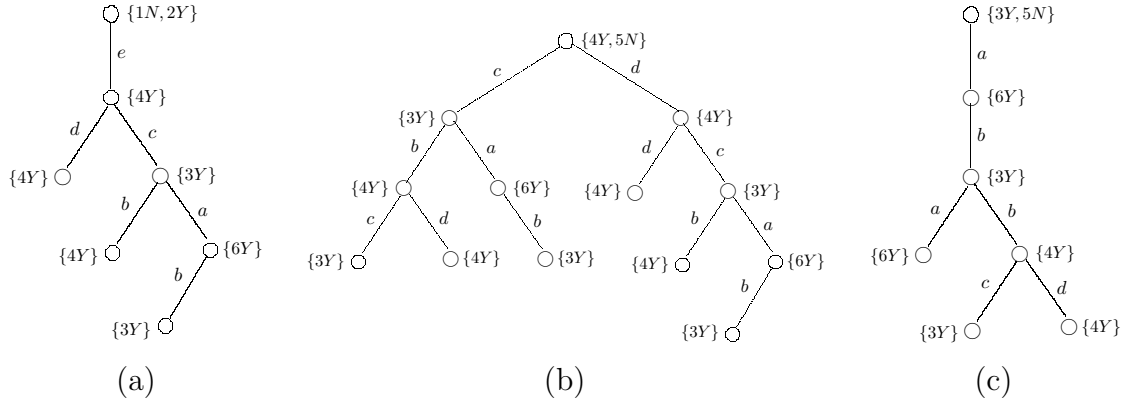

Figura 3.4: Árvores correspondentes aos estados-origem $x_{d_{YN},1} = \{1N, 2Y\}$ (a), $x_{d_{YN},2} = \{4Y, 5N\}$ (b), e $x_{d_{YN},3} = \{3Y, 5N\}$ (c).

3.2.2 Uma nova condição para a diagnose de SEDs com observação parcial

De acordo com o Teorema 3.3, cada conjunto de eventos elementares para a diagnose possui o menor número de eventos necessários para a diagnose da ocorrência de σ_f. Entretanto, para que se possa avaliar a diagnosticabilidade de L em relação a $P'_o : E^* \rightarrow E'^*_o$ e $E_f = \{\sigma_f\}$, em que $E'_o \in E_{eed}$, é necessário que se construa o diagnosticador parcial G'_d. De acordo com o Teorema 3.2, L é também diagnosticável com relação a P'_o e $E_f = \{\sigma_f\}$ se, e somente se, o diagnosticador parcial G'_d não possuir ciclos indeterminados observados e escondidos. Se para algum conjunto E'_o, L for diagnosticável, então E'_o será uma base mínima para a diagnose. Porém, se L não for diagnosticável com relação a P'_o e $E_f = \{\sigma_f\}$, será necessário adicionar novos eventos à E'_o. Como o interesse está na busca por bases mínimas para a diagnose, a inserção de eventos deve ser feita de forma criteriosa, de forma a evitar que eventos redundantes sejam inseridos.

Uma forma imediata de se encontrar os conjuntos de eventos cuja união com E'_o seja uma base mínima é através de uma busca exaustiva nos subconjuntos de $2^{E_o \setminus E'_o} \setminus \{\emptyset, E_o \setminus E'_o\}$. Entretanto, essa solução requer cálculos em demasia, e portanto, um método diferente de busca será proposto nesse trabalho. A idéia básica é formar novos conjuntos $E''_o = E'_o \cup \{\sigma\}$, em que σ é um evento pertencente a uma sequência ambígua (s_a) de L com relação à projeção P'_o ou um evento pertencente a uma sequência normal (s_N) que satisfaça $P'_o(s_N) = P'_o(s_a)$; neste último caso, quando o evento escolhido pertencer somente a s_N ou aparecer em certas posições de s_a e de s_N fazendo com que $P''_o(s_a) \neq P''_o(s_N)$, $P''_o : E^*_o \rightarrow E''^*_o$, a inserção deste evento será certamente uma opção válida. Entretanto, sequências ambíguas não aparecem claramente em G'_d, e portanto, uma outra forma de se identificar essas sequências deve ser desenvolvida.

Considere o seguinte autômato:

$$G_{\text{teste}} = G'_d \| G_d = (X_t, E_o, f_t, \Gamma_t, x_{0_t}, X_{m_t}). \tag{3.3}$$

Note que o estado x_t de G_{teste} possui a seguinte estrutura:

$$x_t = (x'_d, x_d),$$

em que $x'_d \in X'_d$ e $x_d \in X_d$.

Definição 3.8 *Um estado x_t de G_{teste} é um estado certo se x'_d e x_d forem ambos estados certos, e incerto se x_d for um estado certo e x'_d for um estado incerto.* □

Definição 3.9 *Um ciclo em G_{teste} será um ciclo indeterminado se o ciclo correspondente (observado ou escondido) em G'_d for indeterminado.* □

Da definição de G_{teste} dada pela Eq. (3.3), não é difícil de se observar que

$$\mathcal{L}(G_{\text{teste}}) = P^{-1}_{oo'}(\mathcal{L}(G'_d)) \bigcap \mathcal{L}(G_d) = \mathcal{L}(G_d),$$

em que a projeção inversa $P^{-1}_{oo'}$ é em relação a E_o e não em relação a E. É sabido que uma condição necessária e suficiente para que a linguagem L seja não-diagnosticável com relação a P'_o e E_f é a existência de sequências ambíguas s_a em relação a P'_o e E_f. O resultado seguinte mostra que G_{teste} pode ser utilizado não somente como um teste para a diagnosticabilidade, mas também para se encontrar todas as sequências ambíguas $s_a \in L$ com relação a P'_o e E_f.

Teorema 3.4 *Suponha que a linguagem L, gerada por $G = (X, E, f, \Gamma, x_0, X_m)$, em que $E = E_o \cup E_{uo}$, seja diagnosticável com relação a $P_o : E^* \to E_o^*$ e $E_f = \{\sigma_f\} \subset E_{uo}$. Então, L será diagnosticável com relação a $P'_o : E^* \to E_o'^*$, $E'_o \subset E_o$, e $E_f = \{\sigma_f\}$ se, e somente se, G_{teste} não possuir ciclos indeterminados, em que G_{teste} é definido de acordo com a Eq. (3.3).*

Demonstração: ($\Rightarrow$) Suponha que G_{teste} possua um ciclo indeterminado e que a sequência $st \in L$ satisfaz às seguintes condições: (i) $s \in \Psi(E_f)$; (ii) $\|t\| > n_t$, em que n_t pode ser arbitrariamente longo; (iii) $P_o(st)$ gira em um ciclo indeterminado de G_{teste}. Seja $s't' = P'_o(st)$. Então, devido à estrutura de G_{teste}, ou $s't'$ gira em um ciclo observado indeterminado de G'_d ou leva a um estado incerto de G'_d (quando o ciclo indeterminado de G_{teste} estiver relacionado com um ciclo escondido indeterminado de G'_d). Isso implica que $\exists w \in P'^{-1}_{o_L}[P'(st)]$ tal que $E_f \notin w$, o que viola a condição de diagnosticabilidade (Eq. 4.2), ou equivalentemente, que L é não-diagnosticável com relação a P'_o e $E_f = \{\sigma_f\}$.

($\Leftarrow$) Suponha que G_{teste} não possua ciclos indeterminados e considere uma sequência $s \in \Psi(E_f)$. Como L é diagnosticável com relação a P_o e E_f, existe uma sequência t arbitrariamente longa, *i.e.*, $\|t\| > n_t$ (n_t arbitrariamente grande) tal que $P_o(st)$ leva G_{teste} a um estado $x_t = (x'_d, x_d)$ com x_d certo; a componente correspondente x'_d de x_t pode ser um estado certo ou incerto. Porém, como G_{teste} não possui ciclos indeterminados, então t pode ser estendido de forma a fazer com que x'_d seja também um estado certo. Isso implica que $\exists s't' = P'_o(st)$ que leva a um estado certo de G'_d. Como $s \in L$ é arbitrário, então L é também diagnosticável com relação a P'_o e $E_f = \{\sigma_f\}$. □

Como L é diagnosticável com relação a P_o e $E_f = \{\sigma_f\}$, pode-se, a partir do Teorema 3.4, chegar ao seguinte resultado.

Corolário 3.1 *Sob as mesmas hipóteses do Teorema 3.4, uma sequência arbitrariamente longa $s_t \in \mathcal{L}(G_{teste})$, que gira em um ciclo indeterminado de G_{teste}, é tal que o conjunto $P_{o_L}'^{-1}\{P_o'[P_{o_L}^{-1}(s_t)]\}$ possui ambos, uma sequência normal e uma sequência de falha.* □

Uma das consequências do Corolário 3.1 é que mesmo que s_t não seja, em geral, uma sequência de L, uma vez que definida em E_o^*, esta sequência possui uma relação próxima com as sequências ambíguas de L em relação a P_o' e E_f. De forma a justificar este fato, denote por L_d a linguagem gerada por G_d. De acordo com o teorema 3.1, $P_o'(L) = P_{oo'}(L_d)$, e como L é diagnosticável em relação a P_o e E_f, então a análise da diagnosticabilidade de L com relação a P_o' e E_f pode ser feita utilizando-se L_d e $P_{oo'}(L_d)$ em vez de L e $P_o'(L)$. Além disso, a não-diagnosticabilidade de L com relação a P_o' e E_f é devido à existência de ciclos indeterminados observados ou escondidos em G_d'. Na sequência, será estabelecida uma ligação entre esses ciclos de G_d' e suas projeções inversas em G_{teste}.

3.2.3 Trajetórias primas e cobertura para uma trajetória com ciclos inerentes

Considere, inicialmente, os ciclos observados indeterminados de G_d'. Pode-se facilmente concluir que devam existir duas sequências arbitrariamente longas $s_Y, s_N \in L_d$ que satisfaçam às seguintes condições: *(i)* $f_d(x_{0_d}, s_Y) = x_{d_Y}$ e $f_d(x_{0_d}, s_N) = x_{d_N}$, em que x_{d_Y} (x_{d_N}) é um estado certo (respectivamente, normal or uncertain) de G_d que pertence a um ciclo de estados certos (respectivamente, normais ou incertos, mas que não formam um ciclo indeterminado) ; *(ii)* $P_{oo'}(s_Y) = P_{oo'}(s_N) = s'_{YN}$, em que s'_{YN} é tal que $f'_d(x'_{0_d}, s'_{YN}) = x'_{d_{YN}}$, com $x'_{d_{YN}}$ pertencente a um ciclo indeterminado de G_d'. Portanto, para cada ciclo observado indeterminado de G_d', devem existir, pelo menos, dois ciclos em G_{teste}, com as seguintes características: *(i)* um ciclo formado por estados cujas primeiras componentes sejam estados $x'_{d_{YN}}$ de G_d', alcançados através de sequências s'_{YN} e, cujas segundas componentes sejam estados certos x_{d_Y} de G_d, alcançados através da sequência s_Y; *(ii)* outro ciclo formado por estados cujas primeiras componentes sejam as mesmas do ciclo anterior ($x'_{d_{YN}}$), e cujas segundas componentes sejam estados normais x_{d_N} de G_d ou estados incertos de G_d que não são estados de um ciclo indeterminado, ambos alcançados através da sequência s_N.

Não é difícil ver que uma condição necessária para que L seja diagnosticável com relação a $P_o'' : E_o^* \rightarrow E_o''^*$ e E_f, em que $E_o'' = E_o' \cup E_{ies}$ ($E_{ies} \subseteq E_o \setminus E_o'$) é que E_{ies} possua um evento da sequência s_Y ou um evento da sequência s_N que faça com que $P_{oo''}(s_Y) \neq P_{oo''}(s_N)$. É importante notar que como um ciclo pode possuir outros ciclos dentro dele, então podem existir diversas sequências s_Y e s_N, mesmo

no caso em que existe uma única sequência que conecta o estado inicial de G_{teste} ao primeiro estado do ciclo; por exemplo, se os estados $x_{t_1}, x_{t_2}, x_{t_3}, x_{t_2}$ formam um ciclo, então é possível definir diversos ciclos de estados com os estados deste ciclo (*e.g.* $(x_{t_1}, x_{t_2}, x_{t_3}, x_{t_2})$, (x_{t_1}, x_{t_2}) e $(x_{t_1}, x_{t_2}, x_{t_3}, x_{t_2}, x_{t_3}, x_{t_2})$) e, consequentemente, ao menos as sequências que se iniciam no estado inicial e giram em cada um dos ciclos definidos acima podem ser definidos. Portanto, a escolha de E_{ies}, quando baseada em sequências ambíguas não é uma tarefa direta, pois requer que todas as sequências que giram em um ciclo indeterminado de G_{teste} sejam levados em conta. Essa dificuldade será sobreposta pela substituição de sequências arbitrariamente longas por trajetórias de tamanho finito, como mostrado a seguir.

Suponha duas trajetórias fechadas (que possuem um ciclo ao seu final) P_Y^c e P_N^c associadas a dois ciclos de G_{teste},

$$P_Y^c = (x_{t_{Y,q}}, \sigma_{Y,q}, x_{t_{Y,q+1}}, \sigma_{Y,q+1}, \ldots, \sigma_{Y,n-1}, x_{t_{Y,n}}, \sigma_{Y,n}, x_{t_{Y,q}}), \tag{3.4}$$

$$P_N^c = (x_{t_{N,r}}, \sigma_{N,r}, x_{t_{N,r+1}}, \sigma_{N,r+1}, \ldots, \sigma_{N,m-1}, x_{t_{N,m}}, \sigma_{N,m}, x_{t_{N,r}}), \tag{3.5}$$

em que k não é necessariamente igual a r, e que possuem as seguintes propriedades: *(i)* P_Y^c e P_N^c começam e terminam, respectivamente, nos estados $x_{t_{Y,q}}$ e $x_{t_{N,r}}$ de G_{teste}; *(ii)* as segundas componentes dos estados de P_Y^c são estados certos de G_d; *(iii)* as segundas componentes dos estados de P_N^c são estados normais de G_d ou estados incertos de G_d que não são estados de um ciclo indeterminado; *(iv)* as primeiras componentes dos estados de P_Y^c e de P_N^c são estados incertos de G_d' que formam um dos ciclos indeterminados responsáveis pela perda de diagnosticabilidade; *(v)* $x_{t_{Y,i}}$ ($x_{t_{N,i'}}$) não são necessariamente diferentes de $x_{t_{Y,j}}$ ($x_{t_{N,j'}}$) para qualquer $i, j \in \{q, q+1, \ldots, n\}$ ($i', j' \in \{r, r+1, \ldots, m\}$) (*i.e.* essas trajetórias podem ter um ou mais ciclos internos).

De forma a associar sequências com trajetórias é necessário estender P_Y^c e P_N^c para trás, até que se alcance o estado inicial de G_{teste}, como se segue:

$$P_{0_Y} = (x_{t_0}, \sigma_{Y,0}, x_{t_{Y,1}}, \sigma_{Y,1}, \ldots, x_{t_{Y,q-1}}, \sigma_{Y,q-1}, P_Y^c), \tag{3.6}$$

$$P_{0_N} = (x_{t_0}, \sigma_{N,0}, x_{t_{N,1}}, \sigma_{N,1}, \ldots, x_{t_{r-1}}, \sigma_{N,r-1}, P_N^c). \tag{3.7}$$

Note que definindo-se s_Y e s_N como

$$s_Y = \sigma_{Y,0}\sigma_{Y,1}\ldots\sigma_{Y,q-1}(\sigma_{Y,q}\sigma_{Y,q+1}, \ldots, \sigma_{Y,n-1}\sigma_{Y,n})^\ell,$$

$$s_N = \sigma_{N,0}\sigma_{N,1}\ldots\sigma_{N,r-1}(\sigma_{N,r}\sigma_{N,r+1}, \ldots, \sigma_{N,m-1}\sigma_{N,m})^\ell,$$

em que $\ell \in \mathbb{N}$, pode-se fazê-los arbitrariamente longos, e portanto, fica claro que $P_{oo'}(s_Y) = P_{oo'}(s_N)$. Logo, as trajetórias definidas pelas Equações (3.6) e (3.7)

guardam as mesmas informações que as sequências s_Y e s_N, necessárias para o acréscimo de eventos à E_o' com o intuito de tornar L diagnosticável com relação a P_o'' e E_f. Considere as seguintes definições.

Definição 3.10 *(Trajetórias primas) Considere o autômato* $G = (X, E, f, \Gamma, x_0, X_m)$ *que satisfaz às hipóteses A1–A3. Uma trajetória* $(x_0, \sigma_0, x_1, \sigma_1, \ldots, \sigma_{n-1}, x_n)$ *de* G *que se inicia no estado inicial* x_0 *é uma trajetória prima se satisfizer às seguintes condições:*

1. $x_i \neq x_j$ *para todo* $i \neq j$ *e* $i, j \in \{0, 1, 2, \ldots, n-1\}$*;*
2. $\exists k \in \{0, 1, 2, \ldots, n-1\}$ *tal que* $x_n = x_k$. □

Todas as trajetórias primas de G podem ser encontradas através da construção de uma árvore T com raíz x_0, de forma similar à árvore obtida de acordo com o Algoritmo 3.1, como se segue.

Algoritmo 3.2 *(Algoritmo para a obtenção de todas as trajetórias primas de um autômato)*

Passo 1 *Rotule a raíz de T com* x_0.

Passo 2 *Defina* $|\Gamma(x_0)| = n_0$ *e* $x = f(x_0, \sigma)$, $\sigma \in \Gamma(x_0)$. *Crie* n_0 *descendentes de* x_0 *e rotule-os com* x *e os correspondentes ramos* (x_0, x) *com* σ.

Passo 3 *Um nó rotulado com* x*, definido na árvore, será uma folha se o estado* x *já tiver rotulado algum ancestral de* x. *Caso contrário, defina* $|\Gamma(x)| = n$ *e* $x_{new} = f(x, \sigma)$, $\sigma \in \Gamma(x)$. *Crie* n *descendentes de* x *e rotule-os com* x_{new} *e os correspondentes ramos* (x, x_{new}) *com* σ. *Repita esse passo até que todos os estados* x_{new} *sejam folhas.*

Passo 4 *Identifique todas folhas* x_l *de* T *e forme todas as possíveis trajetórias que se iniciam na raíz e terminam em* x_l. □

Considere

$$P_l^c = (x_l, \sigma_l, x_{l+1}, \sigma_{l+1}, \ldots, \sigma_{n-1}, x_n, \sigma_n, x_l) \tag{3.8}$$

uma trajetória que se inicia e termina no mesmo estado de G, em que x_i não é necessariamente diferente de x_j, $i \neq j$, $i, j \in \{l, l+1, \ldots, n\}$ e defina uma trajetória

$$P_0 = \{x_0, \sigma_0, x_1, \sigma_1, \ldots, x_{l-1}, \sigma_{l-1}, P_l^c). \tag{3.9}$$

Definição 3.11 *(Trajetórias primas para cobertura) Considere a trajetória com ciclos inerentes* P_0 *definida na Equação 3.9. Uma trajetória prima para cobertura de* P_0 *é qualquer trajetória prima que pode ser obtida de* P_0. □

Definição 3.12 *(Cobertura para uma trajetória com ciclos inerentes). Denote por* $C(P_0) = \{P_{0,cpp1}, P_{0,cpp2}, \ldots, P_{0,cpp\ell}\}$ *o conjunto formado por* ℓ *trajetórias primas para cobertura de* P_0 *e suponha que* $E_{0,cppi}$, $i = 1, 2, \ldots, \ell$, *e* E_{P_0} *sejam os conjuntos de eventos que aparecem em* $P_{0,cppi}$ *e* P_0, *respectivamente. Então* $C(P_0)$ *será uma cobertura para* P_0 *se e somente se* $\cup_{i=1}^{\ell} E_{0,cppi} = E_{P_0}$. □

Observação 3.3 *Das definições 3.10, 3.11 e 3.12 pode-se concluir que qualquer trajetória prima é uma trajetória prima para cobertura, e consequentemente, uma cobertura para si própria.*

Lema 3.1 *Suponha duas trajetórias do autômato* G, P_l^c *e* P_0 *definidas de acordo com as Equações (3.8) e (3.9), respectivamente. Então, sempre existirá uma cobertura* $C(P_0)$ *para* P_0.

Demonstração: A demonstração é construtiva. Comece, a partir do estado inicial, seguindo a árvore construída para G de acordo com o Algoritmo 3.2, através de P_0, até alcançar uma folha; esse procedimento define a primeira trajetória prima. Volte ao ancestral de mesmo rótulo que a folha alcançada no passo anterior e continue seguindo a árvore, através de P_0, até alcançar outra folha. A trajetória desde a raíz até essa folha define a segunda trajetória prima. Como no passo anterior, volte ao ancestral de mesma marcação que a folha alcançada e repita o processo até que o último estado de P_0 (x_l) seja alcançado. Se esse estado corresponder a uma folha da árvore, então a última trajetória prima foi encontrada; caso contrário, volte ao primeiro estado de P_l^c e continue seguindo a árvore através de P_0 até alcançar outra folha. Esse procedimento é sempre possível de ser realizado pois existe pelo menos um ciclo inerente em P_0. A trajetória que se inicia na raíz e termina na folha encontrada é a última trajetória prima de P_0. Note que, como todos os eventos e estados de P_0 foram utilizados para formar as trajetórias primas para cobertura e nenhum outro estado ou evento externo à P_0 foi utilizado, fica claro que a condição $E_{P_0} = \cup_{i=1}^{N} E_{cpp,i}$ é satisfeita. □

O exemplo a seguir ilustra o procedimento de formação de uma cobertura para uma dada trajetória com ciclos inerentes.

Exemplo 3.3 *Considere a parte de um autômato mostrada na Figura 3.5(a) e a árvore correspondente, na Figura 3.5(b). Considere a trajetória com ciclos inerentes* $P_l^c = (3, \sigma_{32}, 2, \sigma_{22}, 2, \sigma_{23}, 3, \sigma_{31}, 1, \sigma_{12}, 2, \sigma_{23}, 3)$. *Embora existam muitas formas de se definir* P_0, *o procedimento para a obtenção de sua cobertura é o mesmo. Considere agora uma dessas possibilidades,* $P_0 = (0, u, 1, \sigma_{13}, P_l^c)$. *Seguindo a árvore através de* P_0, *a primeira trajetória prima a ser formada é* $P_{cpp,1} = (0, u, 1, \sigma_{13}, 3, \sigma_{32}, 2, \sigma_{22}, 2)$, *que termina na folha rotulada com* $2\checkmark$. *A segunda trajetória prima é obtida como se segue: retorne até que o primeiro nó rotulado com* 2 *seja alcançado (rotulado*

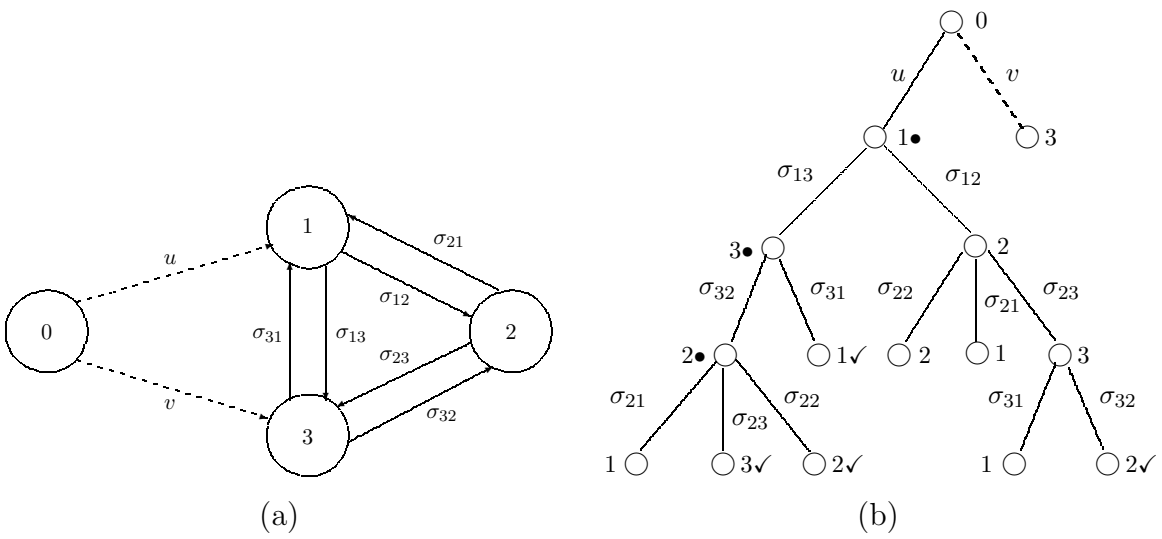

Figura 3.5: Autômato simplificado e árvore para o cálculo da cobertura para uma trajetória com ciclos inerentes.

na Figura 3.5(b) com $2\bullet$ *para termos de entendimento) e, na sequência, siga a árvore através da continuação de* P_0 *depois do estado 2 (estado de* P_0 *em que a busca parou no passo anterior) até que uma nova folha seja alcançada (rotulada com* $3\checkmark$ *na Figura 3.5(b)). A nova trajetória prima encontrada é, portanto,* $P_{cpp,2} = (0, u, 1, \sigma_{13}, 3, \sigma_{32}, 2, \sigma_{23}, 3)$*. De forma a obter a próxima trajetória prima, é necessário retornar ao ancestral de* 3 *com mesmo rótulo (rotulado na Figura 3.5(b) com* $3\bullet$*), e proceder como no passo anterior. No presente caso, a nova folha alcançada é* $1\checkmark$*, que leva a* $P_{cpp,3} = (0, u, 1, \sigma_{13}, 3, \sigma_{31}, 1)$*. Como anteriormente, se faz necessário retornar ao ancestral de* 1 *com mesmo rótulo (nó rotulado com* $1\bullet$*, na Figura 3.5(b)) e siga a árvore como descrito por* P_0 *até que uma nova folha seja alcançada. Porém, embora o último estado de* P_0 *tenha sido alcançado, ele não corresponde a uma folha da árvore. Nesse caso, é necessário voltar ao estado de* P_0 *que corresponde ao primeiro estado de* P_l *e continuar através de* P_0 *até que uma folha seja alcançada (folha rotulada com* $2\checkmark\checkmark$ *na Figura 3.5(b)). Portanto, foi encontrada a última trajetória prima de* P_0*, que é* $P_{cpp,4} = (0, u, 1, \sigma_{12}, 2, \sigma_{23}, 3, \sigma_{32}, 2)$*. O conjunto* $C(P_0) = \{P_{cpp,1}, P_{cpp,2}, P_{cpp,3}, P_{cpp,4}\}$ *é uma cobertura para* P_0 *pois todos os eventos de* P_0 *foram utilizados. É importante observar que se o processo tivesse sido continuado, a próxima trajetória prima a ser encontrada seria* $P_{ccp,5} = (0, u, 1, \sigma_{12}, 2, \sigma_{22}, 2)$*, que, claramente, é diferente de todas as trajetórias primas da cobertura encontrada. Entretanto, como os eventos* σ_{12} *e* σ_{22} *já foram levados em conta, a trajetória* $P_{ccp,5}$ *é uma trajetória prima redundante para* $C(P_0)$*.* □

3.3 Busca pelas bases mínimas para a diagnose centralizada de falhas

3.3.1 Resultados básicos

Retornemos ao problema de se encontrar um conjunto de eventos inovadores $E_{ei} \subseteq E_o \setminus E'_o$ de forma a fazer L diagnosticável com relação a $P''_o : E^* \rightarrow E''^*_o$ ($E''_o = E'_o \cup E_{ies}$) e E_f. Supondo que L não seja diagnosticável com relação a P'_o e E_f, então tanto G'_d como G_{teste} terão um ou mais ciclos indeterminados (o primeiro pode também conter ciclos escondidos indeterminados). Como mencionado anteriormente, para cada ciclo observado indeterminado de G'_d, corresponderão, ao menos, dois ciclos em G_{teste}: um ciclo cujas primeiras componentes dos estados que formam estes ciclos são os estados do ciclo observado indeterminado de G'_d e as segundas componentes serão todas estados certos, e um outro ciclo cujas primeiras componentes são os estados do ciclo indeterminado de G'_d e as segundas componentes são todas formadas por estados normais ou incertos que não formam um ciclo indeterminado de G_d. Isso implica que existe, em G_{teste}, trajetórias como aquelas definidas nas Equações (3.6) e (3.7), e, como consequência, trajetórias primas de cobertura podem ser encontradas para cada uma dessas trajetórias.

Definição 3.13 *(Trajetórias primas-Y e trajetórias primas-N) Uma trajetória prima-Y é uma trajetória prima de G_{teste} cujos estados do seu único ciclo formam um ciclo indeterminado em G_{teste}. Uma trajetória prima-N é uma trajetória prima de G_{teste} cujo único ciclo é formado por estados de G_{teste} cujas primeiras componentes são estados incertos de G'_d, e, cujas segundas componentes são estados normais ou incertos de G_d que não sejam estados de um ciclo indeterminado.* □

Considere, inicialmente, as trajetórias de G_{teste} que possuem ciclos formados por estados cujas primeiras componentes formam ciclos observados indeterminados em G'_d.

Teorema 3.5 *Considere a sequência s' formada por eventos de uma trajetória prima de G'_d cujo único ciclo seja observado indeterminado. Então, é sempre possível encontrar um par de sequências, s_Y e s_N, formados, respectivamente, por eventos de uma trajetória prima-Y e de uma trajetória prima-N de G_{teste} tais que $s' \in \overline{P_{oo'}(s_Y)}$ e $s' \in \overline{P_{oo'}(s_N)}$.*

Demonstração: Por hipótese, L é não-diagnosticável com relação a P'_o e E_f, o que implica que G'_d possui ciclos observados indeterminados (a existência de ciclos escondidos indeterminados foi descartada pois somente ciclos observados estão sendo

levados em conta). Considere uma sequência s' formada por eventos de uma trajetória prima de G'_d cujo único ciclo seja observado indeterminado, e suponha que não seja possível encontrar um par de sequências s_Y e s_N, formadas, respectivamente, por eventos de uma trajetória prima-Y e de uma trajetória prima-N de G_{teste} que satisfaça $s' \in \overline{P_{oo'}(s_Y)}$ e $s' \in \overline{P_{oo'}(s_N)}$. Para que tal par (s_Y, s_N) não exista, uma das seguintes condições deve ser satisfeita: (i) $\exists s_N \in \mathcal{L}(G_{\text{teste}}) : s' \in \overline{P_{oo'}(s_N)}$ e $\nexists s_Y \in \mathcal{L}(G_{\text{teste}}) : s' \in \overline{P_{oo'}(s_Y)}$; (ii) $\exists s_Y \in \mathcal{L}(G_{\text{teste}}) : s' \in \overline{P_{oo'}(s_Y)}$ e $\nexists s_N \in \mathcal{L}(G_{\text{teste}}) : s' \in \overline{P_{oo'}(s_N)}$; (iii) $\nexists s_N \in \mathcal{L}(G_{\text{teste}}) : s' \in \overline{P_{oo'}(s_N)}$ e $\nexists s_Y \in \mathcal{L}(G_{\text{teste}}) : s' \in \overline{P_{oo'}(s_Y)}$.

Vamos supor, inicialmente, que a condição (i) acima seja satisfeita. Não é difícil verificar que, como $G_{\text{teste}} = G'_d \| G_d$ e $\mathcal{L}(G_{\text{teste}}) = \mathcal{L}(G_d)$, então um estado em uma trajetória de G_{teste} será revisitado através de s somente se ambas as condições seguintes forem satisfeitas: (1) um estado de G'_d é revisitado através de $P_{oo'}(s)$ e (2) um estado de G_d é revisitado através de s. Consequentemente, para cada s_Y associada a uma trajetória prima-Y de G_{teste}, deve corresponder uma trajetória com ciclos de G_d que possui ciclos formados por estados certos e uma sequência $s'_Y = P_{oo'}(s_Y)$ formada por eventos de uma trajetória com ciclos inerentes de G'_d. Portanto, como por hipótese, não há $s_Y \in \mathcal{L}(G_{\text{teste}})$ tal que $s' \in \overline{P_{oo'}(s_Y)}$, e somente trajetórias que possuem ciclos formados por estados normais de G_d cujos eventos formam a sequência s_N tal que $s' \in \overline{P_{oo'}(s_N)}$ podem ser encontradas em G_{teste}, então não é possível obter s' associada a uma trajetória prima de G'_d cujo único ciclo seja observado indeterminado, que é uma contradição. O mesmo raciocínio pode ser utilizado pata provar que quando a condição (ii) for satisfeita, também haverá uma contradição. Por fim, quando a condição (iii) for satisfeita, a contradição será trivialmente verificada. □

O teorema 3.5 estabelece que para qualquer sequência s' formada pelos eventos de uma trajetória prima de G'_d cujo único ciclo seja observado indeterminado é sempre possível encontrar, ao menos, um par de sequências s_Y e s_N, formadas, respectivamente, pelos eventos de uma trajetória prima-Y e de uma trajetória prima-N de G_{teste}, tais que $s' \in \overline{P_{oo'}(s_Y)}$ e $s' \in \overline{P_{oo'}(s_N)}$. Esse fato não implica que, para cada trajetória prima-Y de G_{teste}, cuja sequência associada seja s_Y, seja sempre possível encontrar uma sequência s' formada pelos eventos de uma trajetória prima de G'_d cujo único ciclo seja observado indeterminado, tal que, para uma sequência $v \in \overline{s_Y}$, $s' = P_{oo'}(v)$. A mesma conclusão se estende ao caso de trajetórias primas-N de G_{teste}. Em vista desses fatos, o seguinte resultado será enunciado.

Teorema 3.6 *Considere a sequência s formada pelos eventos de uma trajetória prima-Y ou de uma trajetória prima-N de G_{teste}. Então, existe sempre uma sequência $u \in \overline{s}$ tal que $P_{oo'}(u) = s'$, em que s' é uma sequência formada pelos eventos de uma trajetória prima de G'_d (formada a partir de um ciclo observado) cujo único ciclo satisfaz uma das seguintes condições: (i) é indeterminado; (ii) é formado por*

estados normais; (iii) é formado por estados incertos que não dão origem a um ciclo indeterminado.

Demonstração Para um estado ser revisitado em uma trajetória de G_{teste}, um estado da trajetória de G'_d deve ser revisitado através dos eventos da trajetória correspondente de G_{teste}, que pertençam à E'_o. Isso implica que a projeção de uma trajetória prima de G_{teste} não necessariamente leva a uma trajetória prima de G'_d, mas sempre leva a uma trajetória com ciclos inerentes. Logo, qualquer sequência s associada a uma trajetória prima-Y ou a uma trajetória prima-N de G_{teste} possui um prefixo u, tal que $P_{oo'}(u) = s'$, em que s' é uma sequência associada a uma trajetória prima de G'_d, que não necessariamente contém um ciclo indeterminado. Isso ocorre porque um diagnosticador pode "girar" em um ciclo de estados normais ou incertos, que não formam um ciclo indeterminado, antes de "girar" em um ciclo indeterminado. Note que s' não pode ser uma sequência associada a uma trajetória prima cujo único ciclo seja formado por estados certos, pois não é possível que um diagnosticador passe de um estado certo para um estado incerto. □

Considere agora os ciclos escondidos de G'_d. De modo a obter resultados para ciclos escondidos indeterminados similares aos encontrados para ciclos observados indeterminados, é necessário realizar algumas observações. De acordo com a definição 3.1, um ciclo escondido é um ciclo formado por estados de G_d que se juntaram em um único estado de G'_d, e cujas transições são rotuladas por eventos pertencentes a $E_o \setminus E'_o$, ou seja, um ciclo escondido fica inteiramente dentro de um estado de G'_d. Este fato leva à seguinte definição.

Definição 3.14 *(Trajetória com ciclos escondidos inerentes) Uma trajetória*

$$P'_{hc} = (x'_{0_d}, \sigma_{d_0}, x'_{1_d}, \sigma_{d_1}, \ldots, x'_{n_d})$$

de G'_d, em que x'_{0_d} é o estado inicial de G'_d, é uma trajetória com ciclos escondidos inerentes se existir um estado $x'_{k_d} \in P'_{hc}$ que possua um ciclo escondido. □

Observação 3.4 *Note que uma trajetória com ciclos escondidos inerentes pode ou não conter ciclos observados (aqueles cujas transições entre seus estados são rotuladas por eventos pertencentes a E'_o).* □

Se G'_d possuir ciclos escondidos indeterminados, então, necessariamente, devem existir duas sequências $s_Y, s_N \in \mathcal{L}(G_d)$, s_Y arbitrariamente longa e s_N com tamanho finito, para as quais, $s' = P_{oo'}(s_Y) = P_{oo'}(s_N)$ será sempre uma sequência finita. Isso ocorre pelos seguintes motivos: *(i)* para que um ciclo escondido seja indeterminado, este deve ser constituído por estados certos que formem um ciclo em G_d, garantindo a existência de uma sequência s_Y arbitrariamente longa, e; *(ii)* para que o ciclo

escondido seja indeterminado, os estados certos que compõem o ciclo em G_d devem se juntar com estados normais ou incertos de G_d, gerando um estado incerto em G'_d (que contém o ciclo escondido indeterminado), garantindo a existência de uma sequência finita s_N que leva G_d do estado inicial a um estado normal ou incerto.

Finalmente, note que, como $\mathcal{L}(G_{\text{teste}}) = \mathcal{L}(G_d)$ e os eventos que rotulam transições entre esses estados de um ciclo escondido pertencem à $E_o \setminus E'_o$, os ciclos que são escondidos em G'_d possuem ciclos correspondentes em G_{teste}, cujos estados têm a mesma primeira componente. Esse fato sugere que pode ser possível estabelecer uma correlação entre trajetórias primas-Y de G_{teste} e trajetórias de G'_d com ciclos escondidos indeterminados inerentes. Essa correlação é estabelecida no teorema seguinte.

Teorema 3.7 *Considere $x_t^\star = (x_d'^\star, x_d^\star)$ como sendo o único estado revisitado de uma trajetória prima-Y de G_{teste}, e que a sequência s_Y seja formada pelos eventos dessa trajetória prima-Y. Além disso, suponha que $s_Y = uv$, em que $v \in (E_o \setminus E'_o)^*$, $x_t^\star = f_t(x_{0_t}, u)$, e $f_t(x_t^\star, v) = x_t^\star$. Então,*

(1) $s' = P_{oo'}(s_Y)$ é uma sequência formada pelos eventos de uma trajetória com ciclos escondidos indeterminados inerentes de G'_d, no estado $x_d'^\star = f'_d(x'_{0_d}, s')$;

(2) existe, ao menos, uma sequência s_N, formada pelos eventos de uma trajetória prima de G_{teste} cujo único estado revisitado é $x_t^\# = (x_d'^\#, x_d^\#) = f_t(x_{0_t}, s_N)$, em que $x_d'^\#$ é um estado incerto ou normal de G'_d e $x_d^\#$ é um estado normal ou incerto que não faz parte de um ciclo indeterminado de G_d, que possui um prefixo $\hat{s}_N$ ($\hat{s}_N \in \overline{s_N}$) tal que $P_{oo'}(\hat{s}_N) = s'$.

Demonstração: Como $G'_d = Obs(G_d)$ (levando em consideração a equivalência de estados) e $G_{\text{teste}} = G'_d \| G_d$, não é difícil verificar que para todo estado $x_t = (x'_d, x_d) \in X_t$, tem-se que $x_d \subseteq x'_d$. Além disso, como por hipótese, $v \in (E_o \setminus E'_o)^*$ e satisfaz $f_t(x_t^\star, v) = x_t^\star$, então, as primeiras componentes de todos os estados alcançados após a ocorrência de u são todas iguais a $x'_d = f'_d(x'_{0_d}, s')$. Portanto, como $\mathcal{L}(G_{\text{teste}}) = \mathcal{L}(G_d)$ e as segundas componentes dos estados da trajetória prima-Y alcançados após a ocorrência de u são todas subconjuntos de x'_d, então x'_d possui um ciclo escondido indeterminado, o que demonstra a primeira parte do lema.

Para provar a segunda parte, considere que $x_d'^\star$ seja, de acordo com (1), um estado incerto de G'_d que contém um ciclo escondido indeterminado e suponha que não exista uma sequência s_N, formada pelos eventos de uma trajetória prima de G_{teste} cujo único estado revisitado $x_t^\# = (x_d'^\#, x_d^\#)$ seja tal que $x_d'^\#$ seja um estado incerto ou normal de G'_d e $x_d^\#$ um estado normal ou incerto de G_d que não seja um estado de um ciclo indeterminado, que possui um prefixo $\hat{s}_N$ que satisfaz $P_{oo'}(\hat{s}_N) = s'$. Note que $\mathcal{L}(G_{\text{teste}}) = \mathcal{L}(G_d)$, e, para que um estado $x'_d \in X'_d$ possua um ciclo

escondido indeterminado é necessário que existam sequências $\tilde{s}_Y, \tilde{s}_N \in \mathcal{L}(G_{\text{teste}})$, $\tilde{s}_Y$ formada pelos eventos de uma trajetória com ciclos de estados certos inerentes de G_d, e $\tilde{s}_N$ uma sequência de tamanho finito, satisfazendo às seguintes condições: (i) $\tilde{x}_d = f_d(x_{0_d}, \tilde{s}_N)$ é um estado normal ou incerto de G_d; (ii) $P_{oo'}(\tilde{s}_N) = P_{oo'}(\tilde{s}_Y) = \tilde{s}'$. Portanto, como, por hipótese, não existe s_N que leva G_d do estado inicial a um estado normal ou incerto, pode-se concluir que $x_d'^{\star} = f_d'(x'_{0_d}, s')$ é um estado certo de G'_d, o que contradiz o fato de $x_d'^{\star}$ ser um estado incerto. □

A partir de agora, é possível identificar aquelas trajetórias primas de G_{teste} que estão diretamente ligadas a ciclos observados indeterminados em G'_d, e aquelas que estão diretamente ligadas a ciclos escondidos indeterminados em G'_d. Com isso, está estabelecida a correlação entre as trajetórias primas de G_{teste} e as trajetórias com ciclos indeterminados inerentes (observados e escondidos) de G'_d, como se buscava.

3.3.2 Lidando com ciclos observados indeterminados de G'_d

Como dito anteriormente, sendo L diagnosticável com relação a P_o e E_f, a presença de ciclos observados indeterminados em G'_d é determinada pela existência de, pelo menos, duas sequências arbitrariamente longas $s_Y, s_N \in \mathcal{L}(G_d)$, s_Y associada a uma trajetória com ciclos inerentes formados por estados certos de G_d, e s_N associada a uma trajetória que possui ciclos inerentes formados por estados normais de G_d, tais que $P_{oo'}(s_Y) = P_{oo'}(s_N)$. De forma a evitar a existência de tal ciclo observado indeterminado em G''_d, em que $E''_o = E'_o \cup E_{ei}$, $E_{ei} \cap E'_o = \emptyset$, é necessário e suficiente que $G''_{\text{teste}} = G''_d \| G_d$ não possua nenhum ciclo indeterminado, como enunciado no Teorema 3.4. Essa condição pode ser alcançada através da formação do conjunto E_{ei}, contendo eventos pertencentes a $E_o \setminus E'_o$, tal que $P_{oo''}(s_Y) \neq P_{oo''}(s_N)$, para todo $s_Y, s_N \in \mathcal{L}(G_d)$ que satisfaçam às condições citadas acima.

Como as análises de G_{teste} e de G'_d são equivalentes no que diz respeito à análise de diagnosticabilidade de L com relação a E_f e P'_o, podem-se utilizar as trajetórias primas de G_{teste} para aumentar o conjunto E'_o, fazendo com que L seja diagnosticável em relação a $P''_o : E \rightarrow E''^{*}_o$, em que $E''_o = E'_o \cup E_{ei}$, e E_f. De forma a realizar essa tarefa, note que, de acordo com o Teorema 3.6, toda trajetória prima-Y e toda trajetória prima-N, de G_{teste}, possuem associadas a si as sequências s_Y e s_N, respectivamente, cujas projeções em E'_o são iguais ou possuem como prefixo uma sequência s' associada a uma trajetória prima de G'_d formada a partir de um ciclo observado indeterminado, de um ciclo de estados normais ou de ciclo de estados incertos que não seja indeterminado. Considere $x_t^{\star} = (x_d'^{\star}, x_d^{\star})$ como sendo o único estado revisitado de de uma trajetória prima-Y de G_{teste} e que $s_Y = uv$ seja a sequência formada pelos eventos dessa trajetória. Suponha que u e v satisfaçam às seguintes condições: $x_t^{\star} = f_t(x_{0_t}, u)$ e $f_t(x_t^{\star}, v) = x_t^{\star}$. Portanto, para que essa

trajetória prima-Y seja associada a um ciclo indeterminado, que não é escondido em G'_d, ao menos um evento de v deve pertencer à E'_o. A mesma condição se aplica a uma trajetória prima-N de G_{teste}.

Três casos são possíveis de ocorrer:

1. Existe um par (s_Y, s_N) associados a uma trajetória prima-Y e a uma trajetória prima-N de G_{teste}, respectivamente, e uma sequência s' associada a uma trajetória prima de G'_d cujo único ciclo é observado indeterminado, tais que $P_{oo'}(s_Y) = P_{oo'}(s_N) = s'$;

2. Para algum $s_Y = uv$ (ou $s_N = uv$) associado a uma trajetória prima-Y (ou, respectivamente trajetória prima-N) de G_{teste}, existe uma sequência s' associada a uma trajetória prima de G'_d cujo único ciclo é observado indeterminado, tal que $P_{oo'}(u) = s'$ e $P_{oo'}(v) \neq \varepsilon$;

3. Para algum $s_Y = uv$ (ou $s_N = uv$) associado a uma trajetória prima-Y (ou, respectivamente trajetória prima-N) de G_{teste}, existe uma sequência s' associada a uma trajetória prima de G'_d cujo único ciclo não é indeterminado (i.e. é formado por estados normais ou é um ciclo de estados incertos que não indeterminado), tal que $P_{oo'}(u) = s'$ e $P_{oo'}(v) \neq \varepsilon$.

Nesse trabalho somente o caso 1 será considerado.

Teorema 3.8 *Suponha que a linguagem L não seja diagnosticável com relação a P'_o e E_f, e seja $E''_o = E'_o \cup E_{ei}$, $E_{ei} \subseteq E_o \setminus E'_o$. Suponha que G''_d denote o diagnosticador parcial para L considerando E''_o como conjunto de eventos observáveis. Além disso, suponha que exista um par de sequências (s_Y, s_N) associadas a uma trajetória prima-Y e a uma trajetória prima-N de G_{teste}, respectivamente, e uma sequência s' associada a uma trajetória prima de G'_d cujo único ciclo seja observado indeterminado, e que satisfaçam $P_{oo'}(s_Y) = P_{oo'}(s_N) = s'$. Uma condição necessária para que a trajetória prima associada a s' não seja uma trajetória prima de G''_d cujo único ciclo é observado indeterminado, é que $E_{ei} \cap [(E_{s_Y} \cup E_{s_N}) \setminus E'_o] \neq \emptyset$, em que E_{s_Y} e E_{s_N} denotem, respectivamente, o conjunto formado pelos eventos das sequências s_Y e s_N.*

Demonstração: Considere $E_{ies} = \{\sigma\}$, em que $\sigma \notin (E_{s_Y} \cup E_{s_N}) \setminus E'_o$ e suponha que todas as sequências $s'' = P_{oo''}[P^{-1}_{oo'}(s') \cap \mathcal{L}(G_d)]$ sejam associadas a trajetórias G''_d cujos ciclos inerentes não sejam indeterminados. Entretanto, como $P^{-1}_{oo'}(s') \cap \mathcal{L}(G_d) \supseteq \{s_Y, s_N\}$, então $P_{oo''}(s_Y) = P_{oo''}(s_N) = s'$, o que contradiz a hipótese de que s'' não é uma sequência associada a uma trajetória de G''_d com ciclos indeterminados inerentes. □

Observação 3.5 *Note que a condição estabelecida pelo Teorema 3.8 é somente necessária, pois se um evento comum a s_Y e a s_N pertencer à E_{ei} e esse conjunto for*

tal que $P_{oo''}(s_Y) = P_{oo''}(s_N) = s''$*, então* s'' *estará associada a uma trajetória com ciclos indeterminados inerentes.* □

De acordo com o Teorema 3.8, para que um par de sequências (s_Y, s_N), associadas a uma trajetória prima-Y e a uma trajetória prima-N de G_{teste}, que satisfaz $P_{oo'}(s_Y) = P_{oo'}(s_N) = s'$, não leve a trajetórias com ciclos indeterminados inerentes em G''_d, é necessário incluir, ao menos, um evento de s_Y ou um evento de s_N em E_{ies}. Portanto, essa condição deve ser satisfeita para todos os pares de sequências (s_Y, s_N) de G_{teste} cujas projeções em E'_o levam a alguma trajetória prima associada a s', como o caso mencionado acima. Além disso, como toda trajetória prima-Y e -N são coberturas para si próprias, e qualquer trajetória-Y e -N com ciclos inerentes possuem, respectivamente, trajetórias primas-Y e -N como trajetórias primas para cobertura, assim se um evento de cada par (s_Y, s_N) for incluído em E_{ies}, então, ao menos, um evento de cada trajetória-Y ou -N com ciclos inerentes, que possuem, respectivamente, sequências associadas $\hat{s}_Y$ e $\hat{s}_N$ que satisfaçam $P_{oo'}(\hat{s}_Y) = P_{oo'}(\hat{s}_N)$, também será incluído em E_{ei}.

Após a identificação de todas as sequências s_Y e s_N formadas com os eventos, respectivamente, de uma trajetória prima-Y e de uma trajetória prima-N associadas a ciclos observados indeterminados de G'_d, tais que $P_{oo'}(s_Y) = P_{oo'}(s_N) = s'$, será necessário separar essas sequências em dois conjuntos: $P_{Y,i}$ e $P_{N,i}$, $i = 1, \ldots, p$, em que p é o número de trajetórias primas de G'_d formadas a partir de ciclos observados indeterminados. De acordo com o teorema 3.5 e considerando a hipótese de que $P_{oo'}(s_Y) = P_{oo'}(s_N) = s'$, pode-se afirmar que será sempre possível definir p conjuntos de trajetórias primas-Y e p conjuntos de trajetórias primas-N, em que as sequências $s^i_{Y,k}$ que pertencem à $P_{Y,i}$ e as sequências $s^i_{N,l}$ que pertencem à $P_{N,i}$ sejam tais que $P_{oo'}(s^i_{Y,k}) = P_{oo'}(s^i_{N,l})$, para todo k e todo l. Portanto, incluindo-se somente um evento de cada sequência $s^i_{Y,k} \in P_{Y,i}$ ou somente um evento de cada sequência $s^i_{N,l} \in P_{N,i}$, pertencentes à $E_o \setminus E'_o$, para $i = 1, \ldots, p$, será possível criar conjuntos de eventos E''_o que serão candidatos a bases mínimas para a diagnose de falhas, pois somente condições necessárias estão sendo satisfeitas. Antes de apresentar o algoritmo que permite encontrar o conjunto E_{ei}, como tentativa de se evitar ciclos observados indeterminados em G''_d, a definição de uma nova operação entre conjuntos se faz necessária.

Definição 3.15 *(Produto ponto-cartesiano)*
A. *Considere os conjuntos* $E_i \subseteq E$*,* $i = 1, 2, \ldots, n$*. O produto ponto-cartesiano entre os conjuntos* E_i*,* $i = 1, 2, \ldots, n$*, denotado por* $E_1 \dot{\times} E_2 \dot{\times} \ldots \dot{\times} E_n$*, é definido*

como se segue:

$$E_1 \dot{\times} E_2 \dot{\times} \ldots \dot{\times} E_n = \{E_e \;\; = \;\; E_{e,1} \cup E_{e,2} \cup \ldots \cup E_{e,n} : (E_{e,1}, E_{e,2}, \ldots, E_{e,n}) \in 2_1^{E_1} \times \times 2_1^{E_2} \times \ldots \times 2_1^{E_n}\},$$

em que $2_1^E = \{E_e \in 2^E : |E_e| = 1\}$.
B. *Considere os conjuntos* $E_i \subseteq 2^E$, $i = 1, 2, \ldots, n$, *e defina o seguinte conjunto:*

$$E_\times = \{E_e = E_{e,1} \cup E_{e,2} \cup \ldots \cup E_{e,n} : (E_{e,1}, E_{e,2}, \ldots, E_{e,n}) \in 2_1^{E_1} \times 2_1^{E_2} \times \ldots \times 2_1^{E_n}\}.$$

Considere que $|E_\times| = p$ *e que os elementos de* $E_\times$ *sejam denotados por* $E_{\times,i}$, $i = 1, 2, \ldots, p$. *O produto ponto-cartesiano entre os conjuntos* E_i, $i = 1, 2, \ldots, n$, *é definido como se segue:*

$$E_1 \dot{\times} E_2 \dot{\times} \ldots \dot{\times} E_n = \{\tilde{E}_{\times,1}, \tilde{E}_{\times,2}, \ldots; \tilde{E}_{\times,p} : (\tilde{E}_{\times,i} = \cup_{E \in E_{\times,i}} E) \wedge (E_{\times,i} \in E_\times)\}.$$

□

Para ilustrar a operação apresentada na definição 3.15, considere que $E_1 = \{a, b\}$, $E_2 = \{b, c\}$, $E_3 = \{b\}$, e $E_4 = \{a, c\}$. Como $2_1^{E_1} = \{\{a\}, \{b\}\}$, $2_1^{E_2} = \{\{b\}, \{c\}\}$, $2_1^{E_3} = \{\{b\}\}$, e $2_1^{E_4} = \{\{a\}, \{c\}\}$, não é difícil verificar que

$$E_1 \dot{\times} E_2 \dot{\times} E_3 \dot{\times} E_4 = \{\{a, b\}, \{b, c\}, \{a, b, c\}\}.$$

Suponha, agora, que $E_a = \{E_1, E_2\}$, $E_b = \{E_3, E_4\}$, e $E_c = \{E_4\}$. Então, $E_\times$ será dado por

$$E_\times = \{\{E_1, E_4\}, \{E_2, E_4\}, \{E_1, E_3, E_4\}, \{E_2, E_3, E_4\}\},$$

que implica que

$$E_a \dot{\times} E_b \dot{\times} E_c = \{E_1 \cup E_4, E_2 \cup E_4, E_1 \cup E_3 \cup E_4, E_2 \cup E_3 \cup E_4\} = \{\{a, b, c\}\}.$$

Com as definições acima, é possível apresentar o seguinte algoritmo para encontrar o conjunto E_{ei}.

Algoritmo 3.3

Passo 1 *Forme os seguintes conjuntos:*

- $S' = \{s' \in E_o'^* : s'$ *é uma sequência formada com os eventos de uma trajetória prima de* G_d' *formada a partir de um ciclo observado indeterminado. Pode-se escrever esse conjunto como* $S' = \{s_1', s_2', \ldots, s_p'\}$, *em que* $p = |S'|$.

- $S_Y = \{s_Y \in E_o^* : s_Y$ *é uma sequência formada com os eventos de uma trajetória prima-Y de* G_{test} *associada a uma trajetória com ciclos observados indeterminados inerentes de* $G'_d\}$.
- $S_N = \{s_N \in E_o^* : s_N$ *é uma sequência formada com os eventos de uma trajetória prima-N de* G_{test} *associada a uma trajetória com ciclos observados indeterminados inerentes de* $G'_d\}$.

Passo 2 *Para cada* $s'_i \in S'$, $i = 1, \ldots, p$, *forme os seguintes conjuntos:*

- $S_Y^i = \{s_Y \in S_Y : P_{oo'}(s_Y) = s'_i\}$.
- $S_N^i = \{s_N \in S_N : P_{oo'}(s_N) = s'_i\}$.

Passo 3 *Para cada* $s_{Y,k}^i \in S_Y^i$ *forme um conjunto* $E_{Y,k}^i$ *com os eventos de* $s_{Y,k}^i$ *que não pertençam à* E'_o. *Para cada* $s_{N,l}^i \in S_N^i$ *forme um conjunto* $E_{N,l}^i$ *com os eventos de* $s_{N,l}^i$ *que não pertençam à* E'_o.

Passo 4 *Para* $i = 1, \ldots, p$, *calcule:*

- $E_{ei,i}^Y = E_{Y,1}^i \dot{\times} E_{Y,2}^i \dot{\times} \ldots \dot{\times} E_{Y,k}^i$, *em que* $k = |S_Y^i|$.
- $E_{ei,i}^N = E_{N,1}^i \dot{\times} E_{N,2}^i \dot{\times} \ldots \dot{\times} E_{N,l}^i$, *em que* $l = |S_N^i|$.
- $E_{ei,i} = E_{ies,i}^Y \cup E_{ies,i}^N$.

Passo 5 *Calcule* $E_{ei} = E_{ei,1} \dot{\times} E_{ei,2} \dot{\times} \ldots \dot{\times} E_{ei,p}$. □

Observação 3.6 *Note que quando* E_{ei} *tiver um subconjunto que contenha um outro subconjunto de* E_{ei} *(isto é,* $E'_{ei}, E''_{ei} \in E_{ei}$ *tais que* $E'_{ei} \subset E''_{ei}$*). Nesse caso, o conjunto* E''_{ei} *deverá ser retirado de* E_{ei}, *pois o interesse reside somente nas bases mínimas. Se* E'_{ei} *não for uma base para a diagnose, então o algoritmo 3.3 será aplicado a esse conjunto (em uma nova iteração da busca), e o conjunto* E''_{ei} *será novamente formado, se ele for realmente uma base mínima para a diagnose de falhas.* □

O algoritmo 3.3 retorna um conjunto E_{ei} que consiste de subconjuntos de $E_o \setminus E'_o$ que devem ser unidos a E'_o de forma a criar novos candidatos a bases mínimas para a diagnose de falhas. Para ilustrar a aplicação do algoritmo 3.3, considere o seguinte exemplo.

Exemplo 3.4 *(Aplicação do algoritmo 3.3) Considere o autômato G mostrado na figura 3.6. O conjunto* $E_{uo} = \{\sigma, \sigma_f\}$ *denota o conjunto de eventos não-observáveis, enquanto que* $E_f = \{\sigma_f\}$ *denota o conjunto de falhas do sistema. A figura 3.7 mostra o diagnosticador* G_d *para* $L = \mathcal{L}(G)$, *de onde se pode verificar que L é diagnosticável em relação a* $P_o : E^* \rightarrow E_o^*$ *e* E_f. *Utilizando-se o algoritmo 3.1, calcula-se o conjunto de eventos elementares para a diagnose que é dado por* $E_{eed} = \{\{a, c\}, \{a, b, c\}\}$.

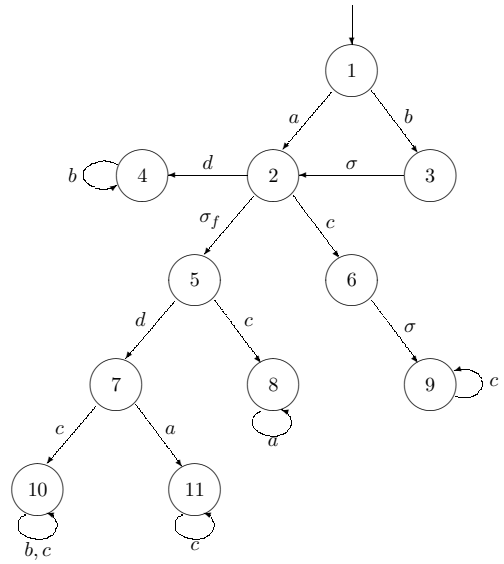

Figura 3.6: Autômato G.

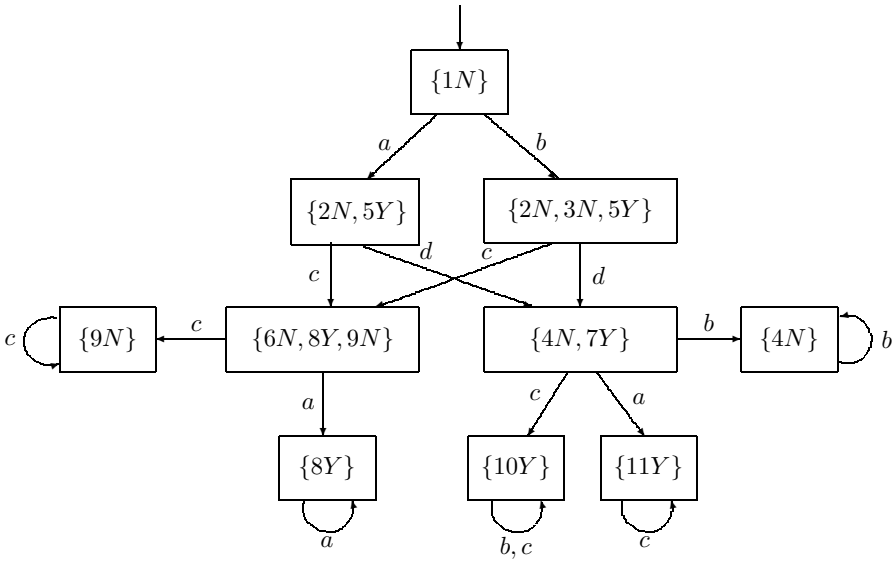

Figura 3.7: Diagnosticador G_d.

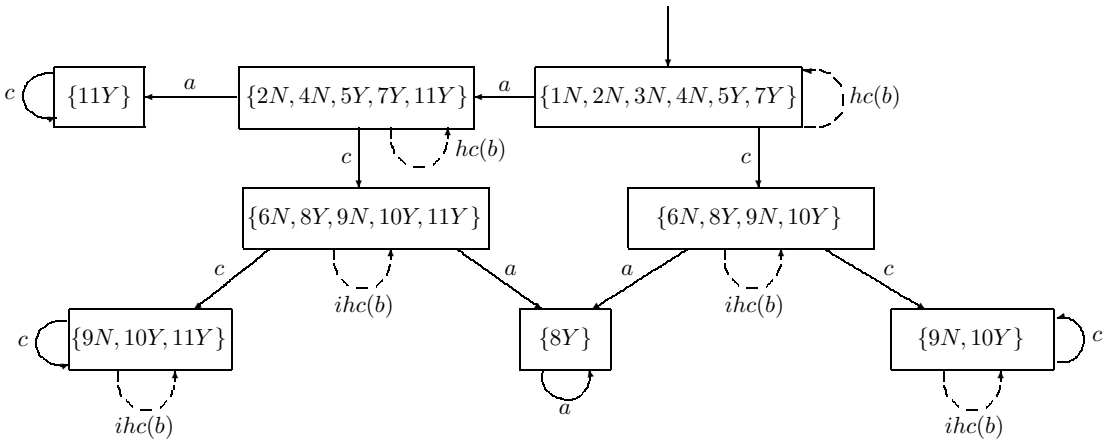

Figura 3.8: Diagnosticador parcial G'_d.

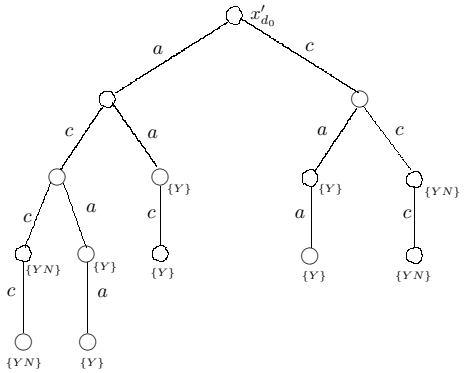

Figura 3.9: Árvore para G'_d.

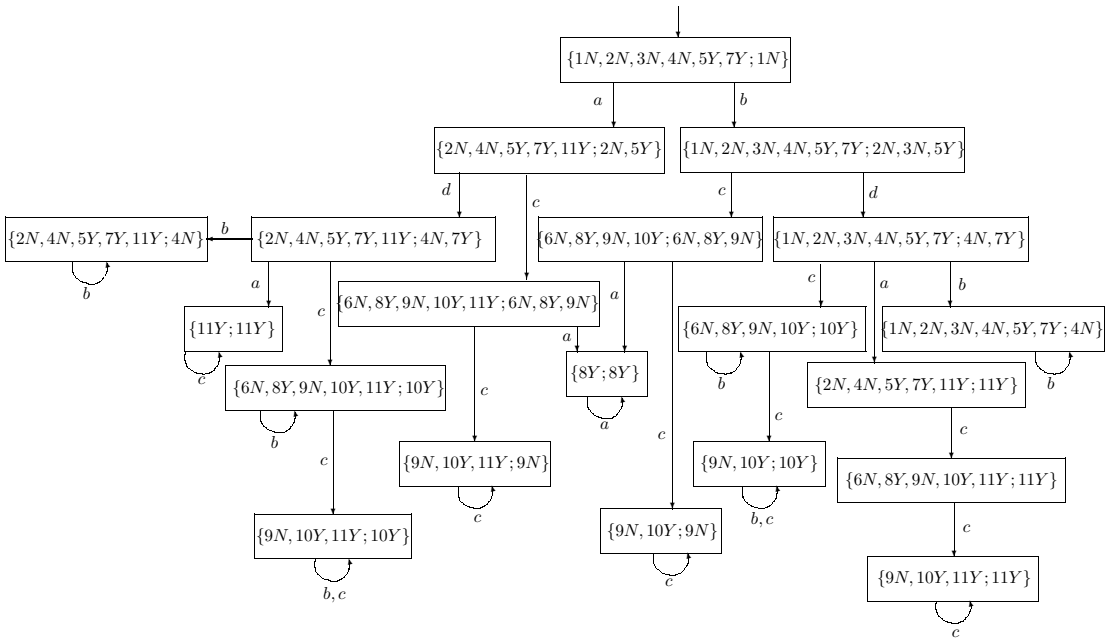

Figura 3.10: Autômato $G_{\text{teste}} = G'_d \| G_d$.

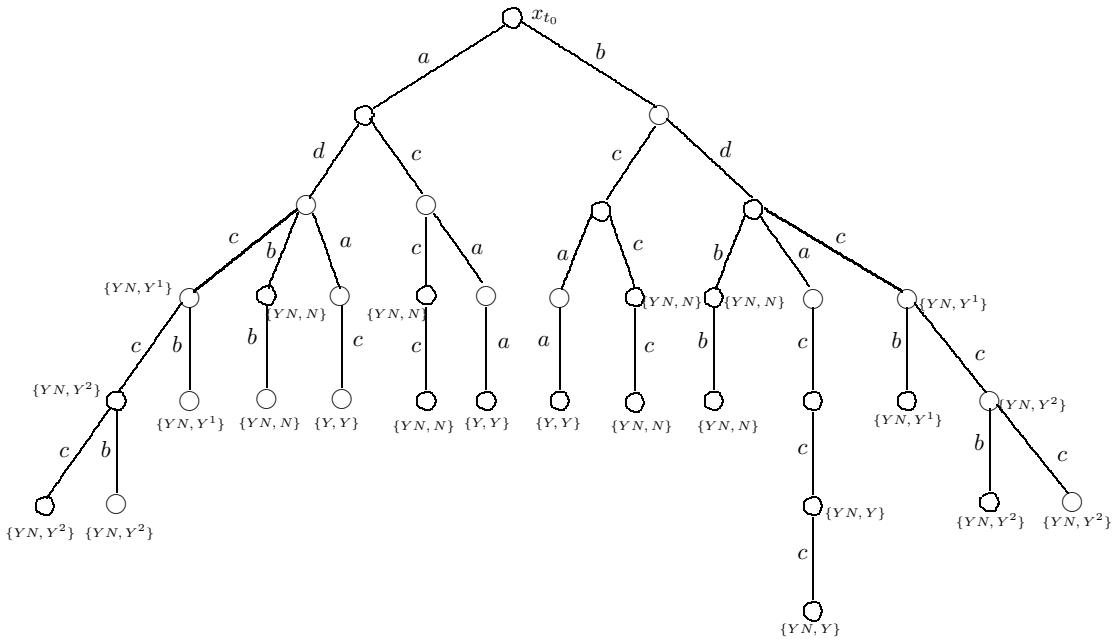

Figura 3.11: Árvore para G_{teste}. Os nós $\{YN, Y^1\}$ e $\{YN, Y^2\}$ correspondem a estados incertos de G_{teste} diferentes numa mesma trajetória.

Inicialmente, verificar-se-á se $E'_o = \{a, c\}$ *é uma base mínima para a diagnose de* $\mathcal{L}(G)$ *construindo-se o diagnosticador parcial* G'_d *considerando* E'_o *como conjunto de eventos observáveis e verificando se há ciclos indeterminados observados e/ou escondidos. Esse diagnosticador parcial está mostrado na figura 3.8. Pode-se observar que* G'_d *possui tanto ciclos escondidos indeterminados (representados por laços próprios formados por arcos pontilhados e rotulados por ihc) quanto ciclos observados indeterminados (nos estados* $(9N, 10Y)$ *e* $(9N, 10Y, 11Y)$*). De maneira a se tentar evitar os ciclos observados indeterminados em* G''_d*, deve-se aplicar o algoritmo 3.3. No passo 1, deve-se formar o conjunto* S' *com as sequências construídas com os eventos de uma trajetória prima de* G'_d *ligada a um ciclo observado indeterminado. Para tanto, deve-se construir a árvore para* G'_d*, mostrada na figura 3.9, e identificar as trajetórias primas que possuem como folha um estado incerto de* G'_d *pertencente a um ciclo observado indeterminado. Pela figura 3.9, somente duas trajetórias satisfazem essa condição, sendo as sequências associadas a essas trajetórias* $s'_1 = accc$ *e* $s'_2 = ccc$*. Logo,*

$$S' = \{s'_1, s'_2\} = \{accc, ccc\}.$$

Para formar os conjuntos S_Y *e* S_N *deve-se construir a árvore para* G_{teste} *de acordo com o algoritmo 3.2 e identificar as trajetórias primas-Y e -N associadas a ciclos observados indeterminados de* G'_d*. Através da árvore para* G_{teste}*, mostrada na figura 3.11, pode-se verificar que as sequências associadas a trajetórias primas-Y são as seguintes:* $s_{Y,1} = adcc(c)$*,* $s_{Y,2} = adcc(b)$*,* $s_{Y,3} = adc(b)$*,* $s_{Y,4} = bdacc(c)$*,* $s_{Y,5} = bdc(b)$*,* $s_{Y,6} = bdcc(b)$*,* $s_{Y,7} = bdcc(c)$*. Porém, somente as sequências* $s_{Y,1}$*,* $s_{Y,4}$ *e*

$s_{Y,7}$ são associadas a trajetórias ligadas a ciclos observados indeterminados, pois, ao menos, um evento da subsequência que leva a trajetória do primeiro alcance do único estado revisitado até seu segundo alcance (destacadas entre parênteses nas sequências) pertencente à E'_o. Logo,

$$S_Y = \{adccc, bdaccc, bdccc\}.$$

As sequências associadas a trajetórias primas-N são: $s_{N,1} = adb(b)$, $s_{N,2} = bcc(c)$, $s_{N,3} = bdb(b)$ e $s_{N,4} = acc(c)$. A mesma condição aplicada para o caso das trajetórias primas-Y deve ser aplicada para as trajetórias primas-N, levando ao conjunto

$$S_N = \{bccc, accc\}.$$

De acordo com o passo 2, devem-se formar os conjuntos

$$\begin{aligned} S_Y^1 &= \{adccc, bdaccc\} \\ S_Y^2 &= \{bdccc\} \\ S_N^1 &= \{accc\} \\ S_N^2 &= \{bccc\} \end{aligned}$$

em que as projeções das sequências pertencentes aos conjuntos com índices 1 e 2 em E'_o são iguais às sequências s'_1 e s'_2 pertencentes à S', respectivamente. No passo 3, formam-se os conjuntos

$$\begin{aligned} E_{Y,1}^1 &= \{d\} \\ E_{Y,2}^1 &= \{b, d\} \\ E_{Y,1}^2 &= \{b, d\} \\ E_{N,1}^1 &= \emptyset \\ E_{N,1}^2 &= \{b\} \end{aligned}$$

com os eventos das sequências que pertencem aos conjuntos criados no passo ante-

rior, e que não pertençam à E'_o. De acordo com o passo 4, calcula-se

$$\begin{aligned}
E^Y_{ei,1} &= E^1_{Y,1}\dot{\times}E^1_{Y,2} = \{\{b,d\},\{d\}\} \\
E^Y_{ei,2} &= \{\{b\},\{d\}\} \\
E^N_{ei,1} &= \emptyset \\
E^N_{ei,2} &= \{\{b\}\} \\
E_{ei,1} &= E^Y_{ei,1} \cup E^N_{ei,1} = \{\{b,d\},\{d\}\} \\
E_{ei,2} &= E^Y_{ei,2} \cup E^N_{ei,2} = \{\{b\},\{d\}\}.
\end{aligned}$$

Por fim, no passo 5, calcula-se

$$E^{ic}_{ei} = E_{ei,1}\dot{\times}E_{ei,2} = \{\{b,d\},\{d\}\}.$$

□

Observação 3.7 *Como demonstrado no exemplo anterior, no caso dos conjuntos S^2_Y e S^2_N, se um conjunto S^i_Y ou um conjunto S^i_N possuir somente uma sequência, então o conjunto $E^Y_{ei,i}$ ou o conjunto $E^N_{ei,i}$ serão formados por subconjuntos contendo somente um evento cada; evento este que pertence, respectivamente, à única sequência do conjunto S^i_Y ou à única sequência do conjunto S^i_N, e que não pertence a E'_o.*
□

3.3.3 Lidando com ciclos escondidos indeterminados de G'_d

Como citado anteriormente, a presença de ciclos escondidos indeterminados em G'_d é caracterizada pela existência de, ao menos, uma sequência arbitrariamente longa $t_Y \in \mathcal{L}(G_d)$ associada a uma trajetória com ciclos de estados certos inerentes, de G_d, e uma sequência de comprimento finito $t_N \in \mathcal{L}(G_d)$ que leva G_d do seu estado inicial a um estado normal ou incerto, satisfazendo $P_{oo'}(t_Y) = P_{oo'}(t_N) = s'$, em que s' é uma sequência formada pelos eventos de uma trajetória com ciclos escondidos indeterminados inerentes. Uma condição necessária e suficiente para se evitar esse ciclo indeterminado escondido em G''_d (o diagnosticador de L supondo $E''_o = E'_o \cup E_{ei}$, $E_{ei} \subseteq E_o \setminus E'_o$), é que $G''_{\text{teste}} = G''_d \| G_d$ não possua ciclos indeterminados. Como no caso dos ciclos observados indeterminados, essa condição pode ser alcançada incluindo-se eventos de $E_o \setminus E'_o$ de forma a satisfazer $P_{oo''}(t_Y) \neq P_{oo''}(t_N)$ para todo $t_Y, t_N \in \mathcal{L}(G_d)$ que satisfaçam às condições de existência de um ciclo escondido indeterminado em G'_d.

Considere agora a sequência $s_Y = uv$, $v \in (E_o \setminus E'_o)^*$, formada com os eventos de uma trajetória prima-Y de G_{teste} cujo único estado revisitado seja $x^\star_t = (x'^\star_d, x^\star_d)$, em que $x^\star_t = f_t(x_{0_t}, u)$ e $f_t(x^\star_t, v) = x^\star_t$. De acordo com o teorema 3.7, essa trajetó-

ria prima-Y está associada a uma trajetória com ciclos escondidos indeterminados inerentes de G'_d no estado $x'^{\star}_d = f'_d(x'_{0_d}, s')$, em que $s' = P_{oo'}(s_Y)$. Uma condição necessária para que uma trajetória com ciclos escondidos indeterminados de G'_d associada a uma trajetória prima-Y de G_{teste} não seja uma trajetória de G''_d é dada pelo seguinte teorema.

Teorema 3.9 *Suponha que a linguagem L não seja diagnosticável com relação a P'_o e E_f e considere $E''_o = E'_o \cup E_{ei}$, $E_{ei} \subseteq E_o \setminus E'_o$. Seja G''_d o diagnosticador parcial de L considerando E''_o como conjunto de eventos observáveis. Além disso, suponha que exista um par (s_Y, s_N), em que s_Y é uma sequência formada pelos eventos de uma trajetória prima-Y de G_{teste} associada a uma trajetória com ciclos escondidos indeterminados, e s_N é uma sequência formada pelos eventos de uma trajetória prima de G_{teste} cujo único estado revisitado é $x_t^{\#} = (x'^{\#}_d, x^{\#}_d)$, com $x^{\#}_d$ sendo um estado normal ou incerto de G_d. Finalmente, suponha que para uma sequência $\hat{s}_N \in \overline{s_N}$, $P_{oo'}(s_Y) = P_{oo'}(\hat{s}_N) = s'$. Uma condição necessária para que a trajetória com ciclos escondidos indeterminados associada a s' não seja uma trajetória de G''_d, é que $E_{ei} \cap [(E_{s_Y} \cup E_{\tilde{s}_N}) \setminus E'_o] \neq \emptyset$, em que E_{s_Y} e $E_{\tilde{s}_N}$ denotam, respectivamente, o conjunto formado pelos eventos da sequência s_Y e $\tilde{s}_N$, com $\tilde{s}_N$ um prefixo de $\hat{s}_N$ tal que $P_{oo'}(\tilde{s}_N) = P_{oo'}(\hat{s}_N)$ e cujo último evento $\tilde{s}_{N_f} \in E'_o$.*

Demonstração: Considere $E_{ies} = \{\sigma\}$, em que $\sigma \notin (E_{s_Y} \cup E_{\hat{s}_N}) \setminus E'_o$, e que todas as sequências $s'' = P_{oo''}[P^{-1}_{oo'}(s') \cap \mathcal{L}(G_d)]$ estejam associadas a trajetórias de G''_d que não possuam ciclos escondidos indeterminados inerentes. Entretanto, $P_{oo''}(s_Y) = P_{oo''}(\hat{s}_N) = s'$, pois $P^{-1}_{oo'}(s') \cap \mathcal{L}(G_d) \supseteq \{s_Y, \hat{s}_N\}$, o que contradiz a hipótese de que s'' não seja uma sequência associada a uma trajetória de G''_d com ciclos escondidos indeterminados inerentes. □

Utilizando a definição de $\tilde{s}_N$, pode-se escrever $\hat{s}_N = \tilde{s}_N\hat{w}$, em que $\hat{w} \in (E_o \setminus E'_o)^*$. Suponha que $\sigma \in E_{ei}$, $\sigma \in \hat{w}$ mas $\sigma \notin s_Y$, e $\sigma \notin \tilde{s}_N$. Como $P_{oo'}(s_Y) = P_{oo'}(\hat{s}_N) = P_{oo'}(\tilde{s}_N) = s'$, é trivial verificar que $P_{oo''}(s_Y) = P_{oo''}(\tilde{s}_N) = s'$, mostrando que, nesse caso, G''_d também possui o ciclo escondido indeterminado.

Observação 3.8

1. *Nesse ponto, poder-se-ia questionar se seria possível resolver o problema do ciclo escondido indeterminado selecionando-se eventos de $(E_{s_Y} \cup E_{\hat{s}_N}) \setminus E'_o$ para formar E_{ei}, em vez de se restringir a eventos do conjunto $(E_{s_Y} \cup E_{\tilde{s}_N}) \setminus E'_o$. Porém, um evento $\sigma \in E_{\hat{s}_N}$, em que $\sigma \notin E_{\tilde{s}_N}$ e $\sigma \notin E_{s_Y}$, não seria suficiente. Nesse caso, embora $P^{-1}_{oo'}(s') \cap \mathcal{L}(G_d) \supseteq \{s_Y, \hat{s}_N, \tilde{s}_N\}$ e $P_{oo''}(\hat{s}_N) \neq s'$, $P_{oo''}(s_Y) = P_{oo''}(\tilde{s}_N) = s'$, o que, novamente, levaria a um ciclo escondido indeterminado em G''_d.*

2. *Embora a condição imposta pelo Teorema 3.9 seja somente necessária, ela pode se tornar suficiente se hipóteses adicionais forem feitas. Por exemplo, considere que* $\sigma \in E_{ei}$, $\sigma \in \tilde{s}_N$ *e* $\sigma \notin s_Y$, *e escreva* $\hat{s}_N = \tilde{s}_N\hat{w} = \tilde{t}\sigma\tilde{w}\hat{w}$. *Não é difícil constatar que* $P_{oo'}(s_Y) = P_{oo'}(\tilde{s}_N) = P_{oo'}(\tilde{t})P_{oo'}(\tilde{w})$, $P_{oo''}(s_Y) = P_{oo'}(s_Y) = P_{oo'}(\tilde{t})P_{oo'}(\tilde{w})$ *e* $P_{oo''}(\tilde{s}_N) = P_{oo''}(\tilde{t})\sigma P_{oo''}(\tilde{w})$. *Como consequência, mesmo quando* $\sigma \notin \tilde{t}$ *ou* $\sigma \notin \tilde{w}$, *tem-se que* $P_{oo''}(s_Y) \neq P_{oo''}(\tilde{s}_N)$ *e* $P_{oo''}(s_Y) \neq P_{oo''}(\hat{s}_N)$. *Além disso, não é possível encontrar um prefixo de* $\tilde{s}_N$ *com a mesma projeção em* $E_o''^*$ *que* s_Y, *que implica que* $P_{oo''}[P_{oo'}^{-1}(s') \cap \mathcal{L}(G_d)]$ *não leva a ciclos escondidos indeterminados.* □

Como no caso de ciclos observados indeterminados, para que cada par de sequências (s_Y, s_N), que satisfaça as condições do teorema 3.9, não leve a trajetórias com ciclos escondidos indeterminados inerentes em G_d'', é necessário incluir, ao menos, um evento de s_Y ou um evento de $\tilde{s}_N$ em E_{ei}. Portanto, essa condição deve ser satisfeita para todos os pares de sequências (s_Y, s_N) de G_{teste} que sejam responsáveis pelo aparecimento de trajetórias com ciclos escondidos indeterminados inerentes em G_d'. Além disso, como toda trajetória prima-Y é uma cobertura de si própria, e qualquer trajetória-Y com ciclos inerentes possui as trajetórias primas-Y como trajetórias primas de cobertura, então se um evento de cada sequência s_Y for incluído em E_{ei}, então, ao menos, um evento de cada trajetória-Y com ciclos inerentes que possuem sequências associadas $\hat{s}_Y$, que satisfaçam $P_{oo'}(\hat{s}_Y) = P_{oo'}(\tilde{s}_N)$, também será incluído em E_{ei}. Além disso, como as sequências $\tilde{s}_N$ são de tamanho finito e não estão associadas a trajetórias com ciclos inerentes em G_d, não se faz necessário o uso dos conceitos de cobertura e trajetórias primas para se justificar que somente essas sequências são necessárias para satisfazer à condição para que G_d'' não possua ciclos escondidos indeterminados.

Portanto, após identificar todas as sequências s_Y e $\tilde{s}_N$ que satisfaçam às condições impostas pelo teorema 3.9, em que $P_{oo'}(s_Y) = P_{oo'}(\tilde{s}_N) = s'$, torna-se necessário separá-las em dois conjuntos: $P_{Y,i}$ e $P_{N,i}$, $i = 1, \ldots, p$, em que p é o número de sequências distintas $s' = P_{oo'}(s_Y)$ que podem ser definidas utilizando-se todas as sequências s_Y formadas com os eventos de trajetórias primas-Y de G_{teste} associadas a trajetórias de G_d' com ciclos escondidos indeterminados inerentes. Logo, incluindo-se somente um evento de cada sequência $s_{Y,k}^i \in P_{Y,i}$ ou somente um evento de cada sequência $\tilde{s}_{N,l}^i \in P_{N,i}$, que seja um evento de $E_o \setminus E_o'$, para $i = 1, \ldots, p$, torna-se possível formar conjuntos de eventos candidatos a bases mínimas para a diagnose de falhas, pois somente condições necessárias são satisfeitas. O algoritmo abaixo permite que se encontre o conjunto E_{ei}, como tentativa de se evitar o aparecimento de ciclos escondidos indeterminados em G_d''.

Algoritmo 3.4

Passo 1 *Forme os seguintes conjuntos:*

- $S_{Yh} = \{s_Y \in E_o^* : s_y$ *é uma sequência formada com os eventos de uma trajetória prima-Y de* G_{test} *associada a uma trajetória com ciclos escondidos indeterminados inerentes de* $G_d'\}$.
- $S_{Yh}' = \{s \in E_o'^* : (\exists s_Y \in S_{Yh})[P_{oo'}(s_Y) = s]\}$. *Considere* $p = |S_{Yh}'| \leq |S_{Yh}|$. *Então o conjunto* S_{Yh}' *pode ser escrito por* $S_{Yh}' = \{s_1', s_2', \ldots, s_p'\}$.

Passo 2 *Para cada* $s_i' \in S_{Yh}'$, $i = 1, \ldots, p$, *forme o seguinte conjunto:*

$$S_{Yh}^i = \{s_Y \in S_{Yh} : P_{oo'}(s_Y) = s_i'\}.$$

Passo 3 *Utilizando a mesma árvore construída para se obter as trajetórias primas-Y de* G_{teste}, *forme as seguintes conjuntos:*

- $S_{Nh} = \{s_N \in E_o^* : s_N$ *é uma sequência formada com os eventos de uma trajetória prima de* G_{test} *cujo único estado revisitado possui como segunda componente um estado normal ou um estado incerto de* $G_d\}$.
- $S_{Nh}' = \{s \in E_o'^* : (\exists s_N \in S_{Nh})[P_{oo'}(s_N) = s]\}$.

Passo 4 *Para cada* $s_i' \in S_{Yh}'$, $i = 1, \ldots, p$, *forme os seguintes conjuntos:*

- $S_{Nh}'^i = \{s \in S_{Nh}' : s_i' \in \overline{s}\}$.
- $S_{Nh}^i = \{s_N \in S_{Nh} : (\exists s \in S_{Nh}'^i)[P_{oo'}(s_N) = s]\}$.
- $S_{\tilde{N}h}^i = \{s \in \overline{S_{Nh}^i} : (P_{oo'}(s) = s_i') \wedge (s_f \in E_o')\}$, *em que* s_f *denota o último evento de* s.

Passo 5 *Para cada* $s_{Y,k}^i \in S_{Yh}^i$ *forme o conjunto* $E_{Y,k}^i$ *com os eventos de* $s_{Y,k}^i$ *que não pertencem à* E_o'. *Para cada* $\tilde{s}_{N,q}^i \in S_{\tilde{N}h}^i$ *forme o conjunto* $E_{\tilde{N},q}^i$ *com os eventos de* $\tilde{s}_{N,q}^i$ *que não pertencem à* E_o'.

Passo 6 *Para* $i = 1, \ldots, p$, *calcule* $E_{ei,i}^Y = E_{Y,1}^i \dot{\times} E_{Y,2}^i \dot{\times} \ldots \dot{\times} E_{Y,m}^i$ *e* $E_{ei,i}^{\tilde{N}} = E_{\tilde{N},1}^i \dot{\times} \dot{\times} E_{\tilde{N},2}^i \dot{\times} \dot{\times} \ldots \dot{\times} E_{\tilde{N},r}^i$, *em que* m *e* r *denotam, respectivamente, a cardinalidade de* S_{Yh}^i *e a cardinalidade de* $S_{\tilde{N}h}^i$.

Passo 7 *Para* $i = 1, \ldots, p$, *calcule* $E_{ei,i} = E_{ei,i}^Y \cup E_{ei,i}^{\tilde{N}}$, .

Passo 8 *Calcule* $E_{ei} = E_{ei,1} \dot{\times} E_{ei,2} \dot{\times} \ldots \dot{\times} E_{ei,p}$. □

Observação 3.9 *Como no caso do algoritmo 3.3, se* E_{ei} *possuir um subconjunto que contenha um outro subconjunto de* E_{ei} *(isto é,* $E_{ei_m}, E_{ei_n} \in E_{ei}$ *tais que* $E_{ei_m} \subset E_{ei_n}$*), então o subconjunto* E_{ei_n} *deve ser retirado de* E_{ies}, *pois o interesse reside na identificação somente das bases mínimas para a diagnose de falhas.*

Para ilustrar a aplicação do algoritmo 3.4, considere o exemplo abaixo.

Exemplo 3.5 *(Aplicação do algoritmo 3.4) Considere novamente o autômato G da figura 3.6, seu diagnosticador centralizado (figura 3.7), seu diagnosticador parcial considerando $E'_o = \{a, c\}$ como conjunto de eventos observáveis (figura 3.8) e o autômato $G_{teste} = G'_d \| G_d$ (figura 3.10). Segundo o passo 1 do algoritmo 3.4, deve-se formar os conjuntos S_{Yh} e S'_{Yh}, respectivamente, com as sequências associadas a trajetórias primas-Y de G_{teste} ligadas a trajetórias com ciclos escondidos indeterminados de G'_d e com as projeções em E'^*_o das sequências de S_{Yh}. No exemplo 3.4, foram identificadas todas as trajetórias primas-Y associadas a ciclos observados indeterminados, de forma que as demais trajetórias primas-Y só podem estar associadas a ciclos escondidos indeterminados. Além disso, pode-se identificá-las utilizando-se suas subsequências que levam a trajetória do primeiro alcance do único estado revisitado ao seu segundo alcance, em que, no caso de ciclos escondidos indeterminados, devem ser formadas somente por eventos em $E_o \setminus E'_o$. Portanto, formam-se*

$$\begin{aligned} S_{Yh} &= \{adccb, adcb, bdcb, bdccb\} \\ S'_{Yh} &= \{s'_1, s'_2, s'_3, s'_4\} = \{acc, ac, c, cc\}. \end{aligned}$$

Segundo o passo 2, para cada $s'_i \in S'_{Yh}$ deve-se formar um conjunto S^i_{Yh} com as sequências de S_{Yh} que possuem projeção em E'^*_o igual à sequência s'_i. Formam-se então

$$\begin{aligned} S^1_{Yh} &= \{adccb\} \\ S^2_{Yh} &= \{adcb\} \\ S^3_{Yh} &= \{bdcb\} \\ S^4_{Yh} &= \{bdccb\}. \end{aligned}$$

No passo 3, devem-se identificar todas as sequências associadas a trajetórias primas de G_{teste} cujo único estado revisitado seja um estado normal ou incerto de G_d. Através da árvore da figura 3.11, pode-se concluir que o conjunto formado pelas sequências que satisfazem às condições citadas é o conjunto

$$S_{Nh} = \{adbb, accc, bccc, bdbb\}.$$

É fácil verificar que o conjunto formado pelas projeções em E'^*_o das sequências do conjunto S_{Nh} é

$$S'_{Nh} = \{a, accc, ccc, \varepsilon\}.$$

Através do passo 4, formam-se, então, os seguintes conjuntos:

- $S'^1_{Nh} = \{accc\}$, $S'^2_{Nh} = \{accc\}$, $S'^3_{Nh} = \{ccc\}$ *e* $S'^4_{Nh} = \{ccc\}$;
- $S^1_{Nh} = \{accc\}$, $S^2_{Nh} = \{accc\}$, $S^3_{Nh} = \{bccc\}$ *e* $S^4_{Nh} = \{bccc\}$;
- $S^1_{\tilde{N}h} = \{acc\}$, $S^2_{\tilde{N}h} = \{ac\}$, $S^3_{\tilde{N}h} = \{bc\}$ *e* $S^4_{\tilde{N}h} = \{bcc\}$;

em que S'^i_{Nh} *é formado por sequências do conjunto* S'_{Nh} *que possuem como prefixo a sequência* $s'_i \in S'_{Yh}$; S^i_{Nh} *é formado por sequências do conjunto* S_{Nh} *que possuem como projeção em* E'^*_o *uma sequência de* S'^i_{Nh}; *e* $S^i_{\tilde{N}h}$ *é formado pelos prefixos das sequências de* S^i_{Nh} *que possuem projeções iguais a sequência* s'_i *e cujo último evento pertence a* E'_o, *para* $i = 1, \ldots, 4$. *Através do passo 5, formam-se os seguintes conjuntos:*

- $E^1_{Y,1} = \{b, d\}$, $E^2_{Y,1} = \{b, d\}$, $E^3_{Y,1} = \{b, d\}$ *e* $E^4_{Y,1} = \{b, d\}$;
- $E^1_{\tilde{N},1} = \emptyset$, $E^2_{\tilde{N},1} = \emptyset$, $E^3_{\tilde{N},1} = \{b\}$ *e* $E^4_{\tilde{N},1} = \{b\}$,

com os eventos das sequências pertencentes aos conjuntos S^i_{Yh} *e* $S^i_{\tilde{N}h}$ *que não pertencem a* E'_o, *em que o índice superior denota que os eventos do conjunto pertencem a uma sequência do conjunto* S^i_{Yh} *ou do conjunto* $S^i_{\tilde{N}h}$ *e o índice inferior se refere à ordem da sequência no conjunto. No passo 6, formam-se os seguintes conjuntos:*

- $E^Y_{ei,1} = \{\{b\}, \{d\}\}$, $E^Y_{ei,2} = \{\{b\}, \{d\}\}$, $E^Y_{ei,3} = \{\{b\}, \{d\}\}$ *e* $E^Y_{ei,4} = \{\{b\}, \{d\}\}$;
- $E^{\tilde{N}}_{ei,1} = \{\emptyset\}$, $E^{\tilde{N}}_{ei,2} = \{\emptyset\}$ $E^{\tilde{N}}_{ei,3} = \{\{b\}\}$ *e* $E^{\tilde{N}}_{ei,4} = \{\{b\}\}$.

Como cada conjunto S^i_{Yh} *e* $S^i_{\tilde{N}h}$ *possui somente uma sequência, os conjuntos* $E^Y_{ies,i}$ *e* $E^{\tilde{N}}_{ies,i}$ *são formados por subconjuntos de um único evento, como destacado na observação 3.7. De acordo com o passo 7, formam-se os conjuntos*

$$\begin{aligned} E_{ei,1} &= E^Y_{ei,1} \cup E^{\tilde{N}}_{ei,1} = \{\{b\}, \{d\}\} \\ E_{ei,2} &= E^Y_{ei,2} \cup E^{\tilde{N}}_{ei,2} = \{\{b\}, \{d\}\} \\ E_{ei,3} &= E^Y_{ei,3} \cup E^{\tilde{N}}_{ei,3} = \{\{b\}, \{d\}\} \\ E_{ei,4} &= E^Y_{ei,4} \cup E^{\tilde{N}}_{ei,4} = \{\{b\}, \{d\}\}. \end{aligned}$$

Finalmente, seguindo o passo 8, o seguinte conjunto é formado:

$$E^{ihc}_{ei} = E_{ei,1} \dot{\times} E_{ei,2} \dot{\times} E_{ei,3} \dot{\times} E_{ei,4} = \{\{b\}, \{d\}, \{b, d\}\}.$$

□

Com os formalismos desenvolvidos, nesta seção, que fornecem condições necessárias para se eliminar os ciclos indeterminados observados e escondidos presentes em G'_d, torna-se possível propor um procedimento sistemático para a busca das bases mínimas para a diagnose de falhas em SED.

3.3.4 Procedimento para a busca das bases mínimas para a diagnose de falhas

Embasado nos resultados apresentados nas seções 3.2.1, 3.3.2 e 3.3.3 um algoritmo para a busca das bases mínimas para a diagnose centralizada de falhas em SED é apresentado a seguir.

Algoritmo 3.5

Passo 1 *Calcule G_d e verifique se há ciclos indeterminados observados ou escondidos. Se não houver ciclos indeterminados, avance ao passo 2. Caso contrário, o algoritmo está encerrado, pois não existem bases para a diagnose de falhas em G.*

Passo 2 *Utilizando o algoritmo 3.1, encontre os Conjuntos de Eventos Elementares para a Diagnose de Falhas (E_{eed}).*

Passo 3 *Faça $E_{cbd} = E_{eed}$ e $E_{bmd} = \emptyset$.*

Passo 4 *Calcule $\bar{E}_{cbd} = \{\bar{E} \in E_{cbd} : (\exists \tilde{E} \in E_{eed})[\tilde{E} \subseteq \bar{E}]\}$ e faça $E_{cbd} \leftarrow E_{cbd} \setminus \bar{E}_{cbd}$.*

Passo 5 *Faça $E'_o = E_{cbd_{min}}$, em que $E_{cbd_{min}}$ é o subconjunto de E_{cbd} que possui menor cardinalidade.*

Passo 6 *Calcule G'_d.*

Passo 7 *Se G'_d não possuir ciclos indeterminados observados ou escondidos então*

- $E_{bmd} \leftarrow E_{bmd} \cup \{E'_o\}$
- $E_{cbd} \leftarrow E_{cbd} \setminus \{E'_o\}$

Caso contrário

- $E_{cbd} \leftarrow E_{cbd} \setminus \{E'_o\}$
- *Aplique o algoritmo 3.3 e encontre E^{ic}_{ei}.*
- *Aplique o algoritmo 3.4 e encontre E^{ihc}_{ei}.*
- *Calcule $E_{ei} = E^{ic}_{ei} \dot{\times} E^{ihc}_{ei}$.*
- *Para $i = 1, \ldots, n$ faça $E_{cbd} \leftarrow E_{cbd} \cup \{E'_o \cup E_{ei_i}\}$, em que $n = \|E_{ei}\|$ e E_{ei_i} denota um elemento (conjunto) de E_{ei}.*

Passo 8 *Calcule $\hat{E}_{cbd} = \{\bar{E} \in E_{cbd} : (\exists \tilde{E} \in E_{bmd})[\tilde{E} \subseteq \bar{E}]\}$ e faça $E_{cbd} \leftarrow E_{cbd} \setminus \bar{E}_{cbd}$. Se $E_{cbd} = \{\emptyset\}$ então o algoritmo está encerrado. Caso contrário, retorne ao passo 4.* □

Para ilustrar a aplicação do algoritmo 3.5 considere os dois exemplos a seguir.

Exemplo 3.6 *(Aplicação do algoritmo 3.5) Considere o autômato da figura 3.6. Como citado no exemplo 3.4, seu conjunto de eventos é* $E = \{a, b, c, d, \sigma, \sigma_f\}$ *particionado em* $E_o = \{a, b, c, d\}$ *e* $E_{uo} = \{\sigma, \sigma_f\}$*. Deseja-se encontrar todas as bases mínimas para a diagnose de falhas em* G*. Inicialmente, é necessário verificar se* $L = \mathcal{L}(G)$ *é diagnosticável com relação a* P_o *e* $E_f = \{\sigma_f\}$*. Esta análise foi realizada no exemplo 3.4, onde se obteve resposta afirmativa à essa questão, baseando-se no fato de que* G_d *(figura 3.7) não possui ciclos indeterminados observados ou escondidos. A tarefa a ser realizada no passo 2 do algoritmo 3.5, que consiste em determinar o conjunto de eventos elementares para a diagnose, foi cumprida no exemplo 3.3, levando a* $E_{eed} = \{\{a, c\}, \{a, b, c\}\}$*. Como* $\{a, c\} \subset \{a, b, c\}$*, então o conjunto* $\{a, b, c\}$ *deve ser removido de* E_{eed}*, isto é,* $E_{cbd} = E_{eed} \setminus \{\{a, b, c\}\} = \{\{a, c\}\}$*. A seguir, define-se* $E'_o = \{a, c\}$*, pois esse é o único conjunto de* E_{cbd}*. De acordo com o passo 6, deve-se calcular o diagnosticador parcial* G'_d*, considerando* E'_o *como o conjunto de eventos observáveis de* G*. No exemplo 3.4,* G'_d *foi construído e está mostrado na figura 3.8. Como* G'_d *possui tanto ciclos observados indeterminados quanto ciclos escondidos indeterminados, deve-se aplicar os algoritmos 3.3 e 3.4. Nesse caso, encontra-se* $E_{ei}^{ic} = \{\{b, d\}, \{d\}\}$ *e* $E_{ei}^{ihc} = \{\{b\}, \{d\}, \{b, d\}\}$*, como mostrado nos exemplos 3.4 e 3.5, respectivamente. Portanto, pode-se calcular*

$$E_{ei} = E_{ei}^{ic} \dot{\times} E_{ei}^{ihc} = \{\{b, d\}, \{d\}\}.$$

Atualiza-se o conjunto E_{cbd}*, que passa a ser* $E_{cbd} = \{\{a, c, d\}, \{a, b, c, d\}\}$*. De acordo com o passo 8, como* $E_{bmd} = \emptyset$ *e* $E_{cbd} \neq \emptyset$*, retorna-se ao passo 4. Nesse ponto, faz-se* $E_{cbd} = \{\{a, c, d\}, \{a, b, c, d\}\} \setminus \{\{a, b, c, d\}\} = \{\{a, c, d\}\}$*, dado que* $\{a, c, d\} \subset \{a, b, c, d\}$*. Isso faz com que, nessa iteração,* $E'_o = \{a, c, d\}$*. Como descrito no passo 6, deve-se calcular* G'_d *para o novo conjunto* E'_o*, que está representado na figura 3.12. Note que* G'_d *não possui nenhum ciclo indeterminado, e portanto,* L *é diagnosticável com relação a* P'_o *e* E_f*. De acordo com o passo 7, faz-se* $E_{bmd} = \{\{a, c, d\}\}$ *e* $E_{cbd} = \{\{a, c, d\}\} \setminus \{\{a, c, d\}\} = \emptyset$*. Como* $E_{cbd} = \{\emptyset\}$*, então o algoritmo está encerrado e o conjunto de bases mínimas para a diagnose contém somente um elemento,* $\{a, c, d\}$*.* □

Exemplo 3.7 *(Outro exemplo de aplicação do algoritmo 3.5) Considere o autômato* G *representado na figura 3.13, sendo* $E = \{a, b, c, d, f, \sigma_f\}$ *o conjunto de eventos do autômato, em que* $E_{uo} = E_f = \{\sigma_f\}$*. Através da análise do diagnosticador centralizado mostrado na figura 3.14, pode-se concluir que* L *é diagnosticável em relação a* P_o *e* E_f*, e portanto, pode-se passar ao passo 2 do algoritmo 3.5, em busca de bases mínimas para a diagnose de falhas em* G*. No passo 2*

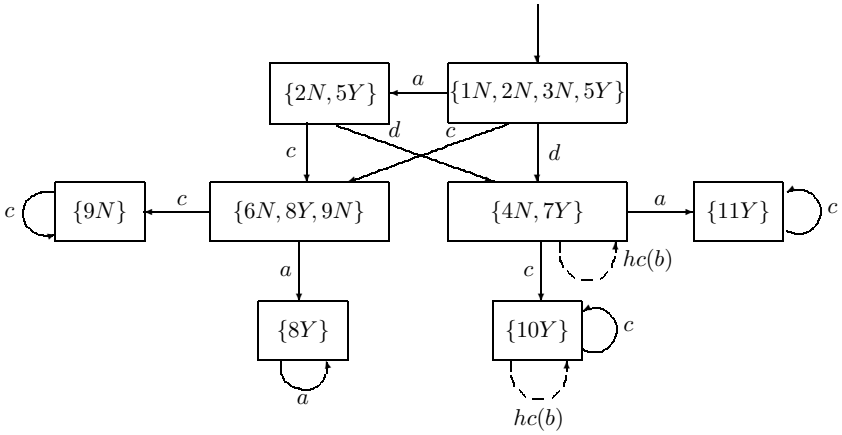

Figura 3.12: Diagnosticador parcial G'_d, para $E'_o = \{a, c, d\}$.

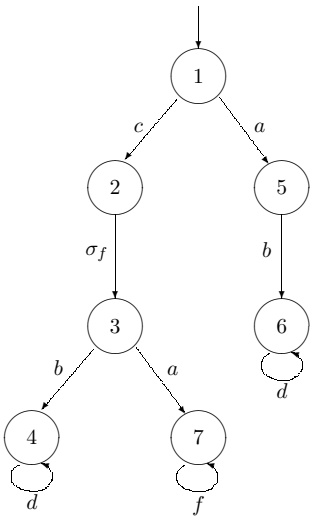

Figura 3.13: Autômato G.

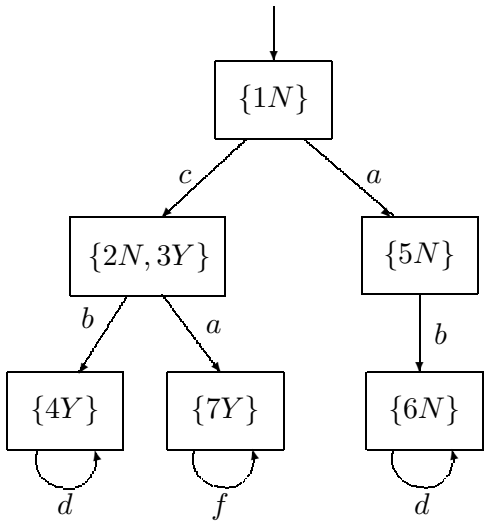

Figura 3.14: Diagnosticador centralizado G_d.

obtém-se $E_{eed} = \{\{a,b\},\{a,d\},\{b,f\},\{d,f\}\}$. *Além disso, como* $\bar{E}_{cbd} = \emptyset$, *então* $E_{cbd} = \{\{a,b\},\{a,d\},\{b,f\},\{d,f\}\}$. *Levando-se em consideração que todos os subconjuntos de* E_{cbd} *possuem igual cardinalidade, arbitra-se* $E'_o = \{a,b\}$. *No passo 6, calcula-se o diagnosticador parcial* G'_d, *que está mostrado na figura 3.15, de onde se pode notar a presença de um ciclo escondido indeterminado devido à falta de observabiblidade do evento* f, *no estado* $\{5N, 7Y\}$. *Logo, no passo 7, faz-se* $E_{cbd} = \{\{a,b\},\{a,d\},\{b,f\},\{d,f\}\} \setminus \{\{a,b\}\} = \{\{a,d\},\{b,f\},\{d,f\}\}$. *A seguir, como* G'_d *possui somente ciclos indeterminados escondidos, basta que o algoritmo 3.4 seja aplicado, não sendo necessário utilizar o algoritmo 3.3. Aplicando-se o algoritmo indicado, encontra-se* $E_{ei} = E_{ei}^{ihc} = \{\{c\},\{f\}\}$. *Com isso, o conjunto* E_{cbd} *deve ser atualizado para* $E_{cbd} = \{\{a,d\},\{b,f\},\{d,f\},\{a,b,c\},\{a,b,f\}\}$. *De acordo com o passo 8, retorna-se ao passo 4, já que* $E_{bmd} = \emptyset$ *e* $E_{cbd} \neq \emptyset$. *No passo 4, calcula-se* $\bar{E}_{cbd} = \{\{a,b,f\}\}$ *e* $E_{cbd} = \{\{a,d\},\{b,f\},\{d,f\},\{a,b,c\},\{a,b,f\}\} \setminus \{\{a,b,f\}\} = \{\{a,d\},\{b,f\},\{d,f\},\{a,b,c\}\}$. *Com um novo conjunto de candidatos, escolhe-se aquele que possui menor cardinalidade. Como três dos quatro subconjuntos possuem a mesma cardinalidade, sendo esta a menor do conjunto* E_{cbd}, *escolhe-se, de forma arbitrária,* $E'_o = \{d,f\}$. *De acordo com o passo 6, calcula-se o diagnosticador parcial* G'_d *para o novo conjunto* E'_o *a ser testado. Esse diagnosticador está mostrado na figura 3.16, onde se pode notar que* G'_d *possui somente um ciclo observado indeterminado. Por esse motivo, no passo 7, torna-se desnecessário aplicar o algoritmo 3.4, bastando que se utilize o algoritmo 3.3. Antes disso, deve-se atualizar o conjunto de candidatos a bases mínimas fazendo* $E_{cbd} = \{\{a,d\},\{b,f\},\{d,f\},\{a,b,c\}\} \setminus \{\{d,f\}\} = \{\{a,d\},\{b,f\},\{a,b,c\}\}$. *Aplicando-se o algoritmo referido, chega-se ao conjunto* $E_{ei} = E_{ei}^{hc} = \{\{a\},\{b\},\{c\}\}$ *e, pode-se atualizar o conjunto* E_{cbd} *para* $E_{cbd} = \{\{a,d\},\{b,f\},\{a,b,c\},\{a,d,f\},\{b,d,f\},\{c,d,f\}\}$

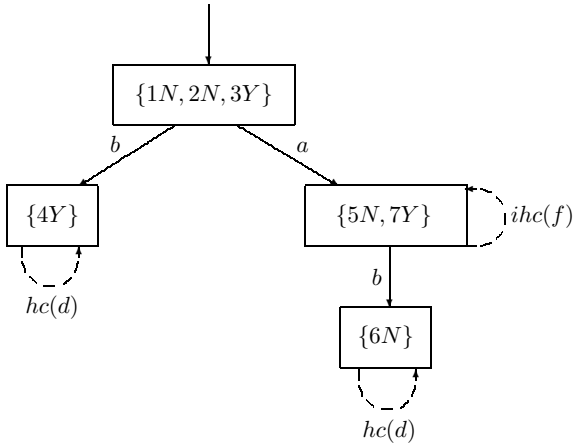

Figura 3.15: Diagnosticador parcial G'_d para $E'_o = \{a, b\}$.

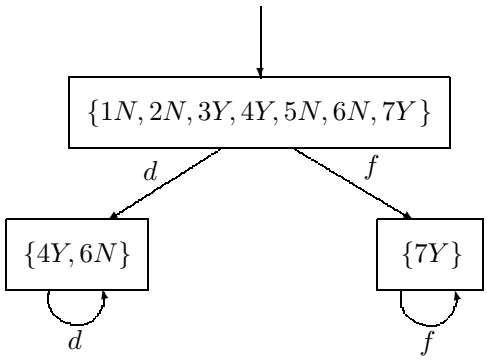

Figura 3.16: Diagnosticador parcial G'_d para $E'_o = \{d, f\}$.

e seguir para o passo 8. Como o conjunto E_{bmd} permanece vazio, pode-se voltar diretamente ao passo 4. Nesse passo, calcula-se $\bar{E}_{cbd} = \{\{a, d, f\}, \{b, d, f\}\}$ e $E_{cbd} = \{\{a, d\}, \{b, f\}, \{a, b, c\}, \{a, d, f\}, \{b, d, f\}, \{c, d, f\}\} \setminus \{\{a, d, f\}, \{b, d, f\}\} = \{\{a, d\}, \{b, f\}, \{a, b, c\}, \{c, d, f\}\}$. Deve-se então escolher agora um dentre os dois candidatos de menor cardinalidade para ser testado. Considere, portanto, $E'_o = \{a, d\}$. Pode-se verificar que esse conjunto também não é uma base para a diagnose de falhas; além disso, dá origem a mais dois candidatos a bases mínimas, pois ao aplicar os algoritmos do passo 7, obtém-se $E_{ei} = \{\{c\}, \{f\}\}$. Com isso, o conjunto de candidatos é atualizado para $E_{cbd} = \{\{b, f\}, \{a, b, c\}, \{c, d, f\}, \{a, c, d\}, \{a, d, f\}\}$. Fazendo $E'_o = \{b, f\}$, pode-se verificar que esse conjunto também não representa uma base mínima para a diagnose de falhas, dando origem a mais três candidatos, pois, para esse caso, $E_{ei} = \{\{a\}, \{c\}, \{d\}\}$. Com isso, o novo conjunto de candidatos passa a ser $E_{cbd} = \{\{a, b, c\}, \{c, d, f\}, \{a, c, d\}, \{a, d, f\}, \{a, b, f\}, \{b, c, f\}, \{b, d, f\}\}$. Pode-se verificar que dentre todos os subconjuntos de E_{dbc}, somente o conjunto $\{b, d, f\}$ não é uma base mínima para a diagnose de falhas, e, além disso, dá origem a mais dois candidatos, que são os conjuntos $\{a, b, d, f\}$ e $\{b, c, d, f\}$. Porém, como $\{a, d, f\}$ e $\{b, d, f\}$ são bases mínimas para a diagnose e estão con-

tidos, respectivamente, em $\{a, b, d, f\}$ *e em* $\{b, c, d, f\}$*, então, no passo 8, faz-se* $E_{cbd} = \{\{a, b, d, f\}, \{b, c, d, f\}\} \setminus \{\{a, b, \ d, f\}, \{b, c, d, f\}\} = \emptyset$*. Logo, o algoritmo está encerrado, obtendo-se seguinte resultado:*

$$E_{bmd} = \{\{a, b, c\}, \{c, d, f\}, \{a, c, d\}, \{a, d, f\}, \{a, b, f\}, \{b, c, f\}\}.$$

□

3.4 Comentários finais

Na subseção 3.3.2 foi constatado que três casos eram possíveis na associação das trajetórias primas-Y e -N de G_{teste} com as trajetórias primas de G'_d. Porém, somente o primeiro caso foi abordado. Isso porque o algoritmo de busca das bases para os outros dois casos possui características peculiares e ainda estão em fase de desenvolvimento. Entretanto, a grande maioria dos casos recai sobre o primeiro, dado que para satisfazer as condições impostas pelos outros dois casos, uma combinação de estados e sequências de eventos específica deve ser satisfeita. Logo, a probabilidade de se ter esses casos na prática é muito pequena.

Após serem encontradas todas bases mínimas para a diagnose de falhas em um SED, pode-se cogitar a hipótese de se construir um diagnosticador centralizado que seja robusto à perda de observabilidade de alguns eventos considerados, anteriormente, observáveis. A intenção é que, com o diagnosticador robusto projetado, se acontecer de algum sensor responsável pela detecção da ocorrência de um evento falhar permanentemente, ainda assim, o diagnosticador seja capaz de informar a ocorrência da falha, caso ela venha a acontecer. Com o diagnosticador centralizado, essa hipótese não poderia ser levada em conta, pois, dado que qualquer sensor venha a falhar, o diagnosticador ficará "preso" em um estado ou avançará por uma trajetória incorreta, fornecendo informações errôneas sobre a ocorrência da falha, em ambos os casos. O diagnosticador robusto à perda permanente de sensores será tema do capítulo a seguir.

Capítulo 4

Diagnose robusta à perda permanente de sensores

Neste capítulo será apresentado o algoritmo para a construção do diagnosticador robusto à perda permanente de sensores e uma condição necessária e suficiente para que esse diagnosticador seja robusto à perda de um dado sensor antes da primeira ocorrência do evento cujo sensor é responsável por detectar sua ocorrência.

Este capítulo está estruturado da seguinte forma: na seção 4.1 são discutidas as principais propriedades desejadas para um diagnosticador robusto à perda permanente de sensores. Na seção 4.2 são apresentadas a formalização teórica e o algoritmo para as principais causas construção do diagnosticador robusto. Ainda na seção 4.2 são discutidas as principais causas de perda de diagnosticabilidade, tais como ciclos indeterminados observados e escondidos, que um diagnosticador candidato a robusto pode vir a apresentar. A solução para essas perdas de diagnosticabilidade é apresentada na seção 4.3, bem como um exemplo de diagnosticador robusto. Comentários finais sobre esse capítulo são feitos na seção 4.4.

4.1 Diagnosticador robusto: propriedades desejadas e definições básicas

Após serem obtidas todas as bases mínimas para a diagnose de falhas de um dado SED, a intenção é utilizar essas bases de forma a se projetar um diagnosticador de falhas centralizado que seja robusto à falhas permanentes de sensores. Para ilustrar essa idéia, suponha que a linguagem gerada por um autômato seja diagnosticável em relação a $P_{o_i} : E_o \rightarrow E_{o_i}$, $i = 1, 2$, e E_f, sendo $E_{o_1} = \{a, b\}$ e $E_{o_2} = \{b, c\}$ duas bases mínimas e $E_o = \{a, b, c\}$ o conjunto de eventos observáveis. Pela definição de base mínima, pode-se afirmar que com a observação dos eventos do conjunto E_{o_1} ou dos eventos do conjunto E_{o_2} é possível detectar a ocorrência da falha. Considere, então,

o seguinte problema: construir um diagnosticador que detecte todas as ocorrências da falha caso as seguintes situações ocorram: *(i)* nenhum sensor falhe; *(ii)* o sensor responsável pela observação do evento a falhe; ou *(iii)* o sensor responsável pela observação do evento c falhe. A esse diagnosticador dar-se-á o nome de diagnosticador robusto à perda permanente, mas não simultânea, de observabilidade do evento c e do evento a. Note que, como o evento b é comum a todas as bases para a diagnose desse SED, não há como construir um diagnosticador que seja robusto a perda de observabilidade desse evento.

O diagnosticador com observação total realiza a função de detectar a ocorrência de uma falha considerando que todos os eventos observáveis permanecem observáveis, enquanto os diagnosticadores parciais detectam as mesmas ocorrências de falhas considerando como observáveis somente um subconjunto do conjunto de eventos observáveis. O diagnosticador robusto deve unir essas duas propriedades. A intuição direta diz, então, que o diagnosticador robusto deve gerar a união das linguagens geradas pelos diagnosticadores parciais considerando como conjunto de eventos observáveis uma base para a diagnose e da linguagem gerada pelo diagnosticador com observação total de eventos. Além disso, as marcações Y e N dos estados dos diagnosticadores devem ser mantidas, para que as sequências de falha possam ser identificadas.

Considere, agora, a seguinte situação: suponha que a linguagem gerada por um autômato seja diagnosticável em relação a P_{o_i}, $i = 1, 2$ e, E_f, em que $E_{o_1} = \{a, b\}$ e $E_{o_2} = \{b, c\}$ são bases mínimas e $E_o = \{a, b, c, d\}$ o conjunto de eventos observáveis. Logo, esse SED possui como bases para a diagnose, os seguintes conjuntos: $E_{o_1} = \{a, b\}$, $E_{o_2} = \{b, c\}$, $E_{o_3} = \{a, b, c\}$, $E_{o_4} = \{a, b, d\}$, $E_{o_5} = \{b, c, d\}$ e $E_o = \{a, b, c, d\}$. Com isso, um diagnosticador robusto pode ser construído considerando, inicialmente, os cinco diagnosticadores parciais e o diagnosticador centralizado. Assim como no caso anterior, o diagnosticador não será robusto à perda de observabilidade do evento b, uma vez que todas as bases possuem esse evento.

Para que o diagnosticador robusto sugerido nesse trabalho possa retornar informações corretas sobre a ocorrência da falha, a seguinte hipótese sobre a perda de observabilidade de eventos deve ser feita.

A5. A perda do sensor ocorre antes da primeira ocorrência do evento a ele associado.

Ao se construir um diagnosticador robusto cuja linguagem gerada seja a união das linguagens geradas pelos diagnosticadores parciais e pelo diagnosticador centralizado com observação total, é possível que uma mesma sequência de eventos leve a um estado certo em um diagnosticador e a um estado normal em outro; por exemplo, $s = bb$, pode levar a um estado certo no diagnosticador parcial para E_{o_1} e a um

estado normal no diagnosticador parcial para E_{o_2}. Essa situação é indesejada, pois, dado que não se detém a informação de qual sensor falhou, não será possível dizer se a falha ocorreu ou não. Logo, é necessário verificar qual perda de observabilidade levou a essa situação, isto é, qual evento que deixou de ser observado na sequência de falha e na sequência "normal" ocasionou tal situação. Esse fato leva à definição de diagnosticabilidade de uma linguagem sob perda permanente de observabilidade de eventos.

Definição 4.1 *(Diagnosticabilidade sob perda permanente de observabilidade de eventos) Seja L uma linguagem viva e de prefixo fechado gerada por um autômato G e suponha que os conjuntos $E'_{o_1}, E'_{o_2}, \ldots, E'_{o_n}$ sejam bases para a diagnose de L. Então L é diagnosticável sob a perda permanente de observabilidade dos conjuntos de eventos redundantes $E_{red}(E'_{o_1}) \cup E_{red}(E'_{o_2}) \cup \ldots \cup E_{red}(E'_{o_n})$ em relação às projeções $P_{o_1}, P_{o_2}, \ldots, P_{o_n}$ e $E_f = \{\sigma_f\}$, em que $P_{o_i} : E^* \rightarrow E^*_{o_i}$, se a seguinte condição for verificada:*

$$(\exists n \in \mathbb{N})(\forall s \in \Psi(E_f))(\forall t \in L/s)(\|t\| \geq n \Rightarrow D_p), \tag{4.1}$$

sendo a condição de diagnosticabilidade D_p expressa por

$$(\forall i, j \in [1, 2, \ldots, n])(i \neq j)(\forall \omega \in P^{-1}_{o_{j_L}}(P_{o_i}(st)))(E_f \in \omega). \tag{4.2}$$

□

As proprieadades desejadas para o diagnosticador robusto podem, então, ser resumidas.

1. O diagnosticador robusto deve conter o maior número possível de diagnosticadores parciais cujos eventos observáveis são as bases (mínimas e não-mínimas) para a diagnose de falhas do SED considerado;

2. A linguagem gerada por esse diagnosticador deve ser a união das linguagens geradas pelos diagnosticadores considerando como conjuntos de eventos observáveis as bases para a diagnose de falhas do SED considerado;

3. O diagnosticador robusto deve manter as marcações Y e N dos estados dos diagnosticadores que o compuseram;

4. Os estados dos diagnosticadores parciais considerando-se como conjunto de eventos observáveis um conjunto E'_o devem carregar uma marcação informando quais sensores falharam para que o diagnosticador alcançasse tal estado através da sequência registrada, isto é, o momento em que os eventos pertencentes à

$E_o \setminus E'_o$ deveriam ser observados e não foram. Além disso, essa marcação deve ser propagada pelos estados subsequentes.

A partir da propriedade 1 do diagnosticador procurado, é possível concluir que esse diagnosticador deve ser robusto à perda de observabilidade dos conjuntos de eventos redundantes de todas as bases não-mínimas para a diagnose, definidos de acordo com a definição 3.5. Isso leva à definição da máxima robustez teórica de um diagnosticador.

Definição 4.2 *(Máxima robustez alcançável) Seja* $E_{bnmd} = \{E_{o_1}, E_{o_2}, \ldots, E_{o_m}\}$ *o conjunto de bases não-mínimas para a diagnose do mesmo SED. A máxima robustez alcançável por um diagnosticador em relação à perda permanente de sensores, é o conjunto formado pelos conjuntos de eventos redundantes (ver definição 3.5) de todas as bases não-mínimas para a diagnose do SED, isto é,*

$$E_{rob}^{max} = E_{red}(E_{o_1}) \cup E_{red}(E_{o_2}) \cup \ldots \cup E_{red}(E_{o_m}).$$

□

Logo, a construção do diagnosticador desejado deve levar em conta todas as bases para a diagnose de falhas, sendo elas mínimas ou não-mínimas (que possuam eventos redundantes), como será visto na seção seguinte.

4.2 Diagnose robusta à perda permanente de observabilidade de eventos

A construção do diagnosticador a ser proposto será dividida, basicamente, em duas etapas: a primeira etapa é a construção dos diagnosticadores considerando as bases para a diagnose de falhas como conjuntos de eventos observáveis. Essa etapa inclui a marcação que indicará quais sensores falharam na trajetória de chegada ao estado atual do diagnosticador. Logo, essa marcação deve ser feita estado a estado. A segunda etapa é a construção do diagnosticador propriamente dito, que será realizada através de um algoritmo, levando a um autômato cuja linguagem gerada é a união das linguagens geradas pelos autômatos que o compuseram, que, nesse caso, são os diagnosticadores parciais e o diagnosticador centralizado.

4.2.1 Diagnosticadores com marcações de perdas de sensores

A primeira etapa da construção do diagnosticador robusto consiste na construção dos diagnosticadores parciais considerando-se as bases para a diagnose de falhas

como conjunto de eventos observáveis. Além disso, devem-se incluir as marcações de indicação de falhas dos sensores, estado a estado como descrito abaixo.

1. Os estados do diagnosticador centralizado receberão a marcação S_n, indicando que todos os sensores estão em funcionamento normal em todos os estados.

2. Seja σ o evento que se tornou não-observável devido à perda do sensor. Os estados dos diagnosticadores parciais serão rotulados da seguinte forma.

   - Se um estado do diagnosticador centralizado for alcançado através de uma sequência que não contém o evento σ (que se tornou não-observável), então esse estado receberá a marcação $S_{\bar{\sigma}}$, indicando que não houve falha no sensor que registra a ocorrência do evento σ; se um estado do diagnosticador centralizado for alcançado através de uma sequência que contém o evento que se tornou não-observável, então o estado receberá a marcação S_σ, indicando que houve uma falha no sensor que detecta o evento σ.
   - Se um estado x do diagnosticador centralizado puder ser alcançado por uma sequência que contém o evento σ e por uma sequência que não o contém, então deve-se criar uma componente para cada marcação, sendo $\{xS_{\bar{\sigma}}, xS_\sigma\}$.
   - Caso o diagnosticador parcial considere a perda de mais de um sensor, por exemplo, dos sensores que detectam a ocorrência dos eventos σ_1 e σ_2, então os estados poderão assumir as marcações $S_{\bar{\sigma}_1\bar{\sigma}_2}$, $S_{\sigma_1\bar{\sigma}_2}$, $S_{\bar{\sigma}_1\sigma_2}$ e $S_{\sigma_1\sigma_2}$.
   - O diagnosticador parcial será obtido utilizando o teorema 3.1.

Com o intuito de formalizar o processo de marcação dos estados dos diagnosticadores parciais, considere as seguintes definições.

Definição 4.3

A. *Seja o conjunto $E'_o \subset E_o$ uma base para a diagnose e considere o conjunto $E'_{uo} = E_o \setminus E'_o = \{\sigma'_1, \sigma'_2, \ldots, \sigma'_n\}$, sendo $n = |E'_{uo}|$. O conjunto de índices de marcações de perdas de sensores, denotado por I, é definido como*

$$I = \{\sigma'_1, \bar{\sigma}'_1\} \times \{\sigma'_2, \bar{\sigma}'_2\} \times \ldots \times \{\sigma'_n, \bar{\sigma}'_n\}.$$

B. *O conjunto de marcações dos estados do diagnosticador parcial, denotado por M, é definido como*

$$M = \{S_m : m \in I\}.$$

□

Definição 4.4 *(Função de propagação de marcação de perdas de sensores) Seja novamente o conjunto $E'_o \subset E_o$ uma base para a diagnose e considere o conjunto $E'_{uo} = E_o \setminus E'_o = \{\sigma'_1, \sigma'_2, \ldots, \sigma'_n\}$, em que $n = |E'_{uo}|$. A função de propagação de marcação de perdas de sensores é definida da seguinte forma:*

$$S : X_d \times M \times E_o \to M$$

$$S(x_d, S_m, \sigma) = \begin{cases} S_{\tilde{m}} : \sigma'_i \in \tilde{m} & se\ (\exists \sigma'_i \in E'_{uo} : \sigma = \sigma'_i) \\ S_m & caso\ contrário \end{cases}$$

em que $\sigma \in \Gamma_d(x_d)$, $x_d \in X_d \setminus \{x_{0_d}\}$ e $S_m, S_{\tilde{m}} \in M$. □

Pode-se estender a definição acima para uma sequência $s\sigma$ pertencente a $\mathcal{L}(G)$, da seguinte forma recursiva:

$$S(x_{0_d}, S_{\bar{\sigma}'_1,\bar{\sigma}'_2,\ldots,\bar{\sigma}'_n}, s\sigma) = S(f_d(x_{0_d}, s), S(x_{0_d}, S_{\bar{\sigma}'_1,\bar{\sigma}'_2,\ldots,\bar{\sigma}'_n}, s), \sigma).$$

Definição 4.5 *(Diagnosticador centralizado com marcação de perda de sensores para a construção do diagnosticador parcial) Seja o conjunto $E'_o \subset E_o$ uma base para a diagnose e considere o conjunto $E'_{uo} = E_o \setminus E'_o = \{\sigma'_1, \sigma'_2, \ldots, \sigma'_n\}$, em que $n = |E'_{uo}|$. O diagnosticador centralizado marcado para a construção do diagnosticador parcial, é definido como*

$$\tilde{G}_d = (\tilde{X}_d, E_o, f_d, \Gamma_d, \tilde{x}_{0_d}),$$

em que $\tilde{X}_d \subseteq X_d \times M$ e $\tilde{x}_{0_d} = x_{0_d} S_{\bar{\sigma}'_1,\bar{\sigma}'_2,\ldots,\bar{\sigma}'_n}$. As marcações dos estados de $\tilde{G}_d$, exceto a do estado inicial, são propagadas pela função de propagação de marcação de perda de sensores. □

Com as definições realizadas, pode-se apresentar um algoritmo que fornece uma forma sistemática de se obter os diagnosticadores parciais para a construção do diagnosticador robusto.

Algoritmo 4.1 *Seja E'_o uma base para a diagnose de L.*

Passo 1 *Calcule o diagnosticador centralizado marcado para a construção do diagnosticador parcial utilizando as definições 4.5 e 4.4.*

Passo 2 *Utilizando o teorema 3.1 construa o diagnosticador parcial considerando E'_o como conjunto de eventos observáveis.*

□

Para ilustrar a aplicação do algoritmo 4.1, considere o seguinte exemplo.

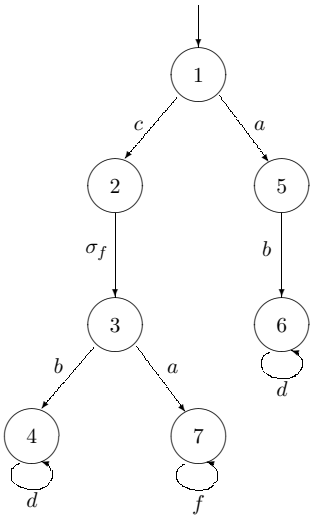

Figura 4.1: Autômato G.

Exemplo 4.1 *(Aplicação do algoritmo 4.1) Considere o autômato G da figura 4.1 cujo conjunto de eventos observáveis é $E_o = \{a, b, c, d, f\}$. O correspondente diagnosticador centralizado está mostrado na figura 3.14. De acordo com o exemplo 3.7, as bases mínimas para a diagnose de $\mathcal{L}(G)$ são:*

$$E_{bmd} = \{\{a, b, c\}, \{c, d, f\}, \{a, c, d\}, \{a, d, f\}, \{a, b, f\}, \{b, c, f\}\}.$$

Considere $E'_{o_1} = \{a, c, d\}$. Nesse caso, $E_o \setminus E'_o = \{b, f\}$. Pelo passo 1 do algoritmo 4.1, devem-se adicionar aos estados de G_d marcações de perdas de sensores S, que, nesse caso, será feita com os eventos b e f que se tornaram não-observáveis. Antes de inserir as marcações de perdas de sensores nos estados de G_d, é conveniente renomeá-los de forma a simplificar as notações. Defina $x_0 = \{1N\}$, $x_1 = \{2N, 3Y\}$, $x_2 = \{5N\}$, $x_3 = \{4Y\}$, $x_4 = \{7Y\}$ e $x_5 = \{6N\}$. Como o diagnosticador centralizado não considera perda de observabilidade, então todos os estados receberão a marcação S_n, conforme mostrado na figura 4.2. Esse diagnsoticador já está devidamente marcado para a construção do diagnosticador robusto. Assim sendo, de acordo com o passo 2 do algoritmo 4.1, o diagnosticador centralizado marcado para a construção do diagnosticador parcial, $\tilde{G}_d$, levando em conta possíveis perdas permanentes de observabilidade dos eventos b e f pode ser construído, sendo mostrado na figura 4.3. Na figura 4.13, está mostrado o diagnosticador parcial considerando a perda de observabilidade dos eventos b e f, como descrito pelo passo 3. Note que não há ciclos indeterminados tanto observados como escondidos, garantindo que a linguagem L é diagnosticável em relação a P'_{o_1} e E_f. □

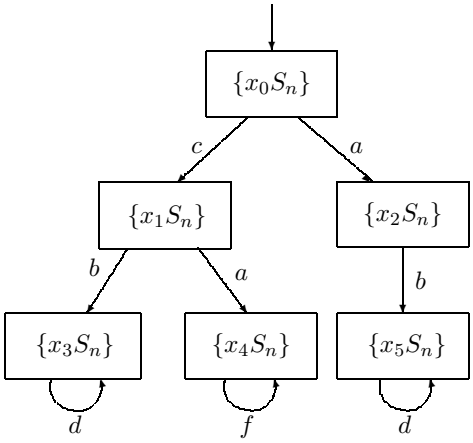

Figura 4.2: Diagnosticador centralizado G_d.

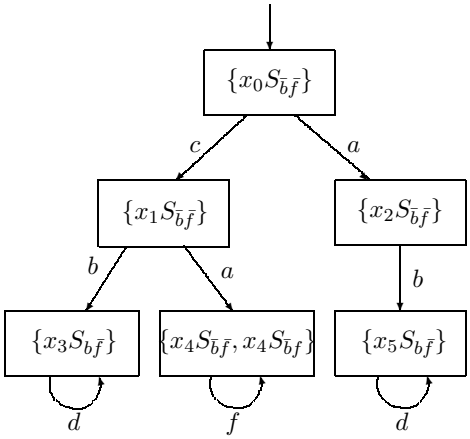

Figura 4.3: Diagnosticador centralizado com estados marcados para construção do diagnosticador robusto considerando $E'_o = \{a, c, d\}$.

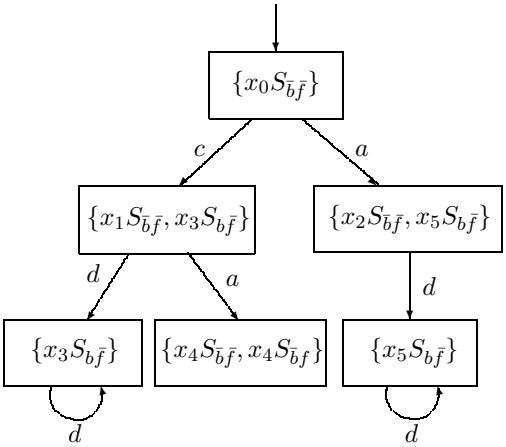

Figura 4.4: Diagnosticador parcial para construção do diagnosticador robusto considerando a perda de observabilidade dos eventos b e f.

4.2.2 Diagnosticabilidade sob perda permanente de observabilidade de eventos: condições necessárias e suficientes

A idéia a ser seguida na construção de um diagnosticador que possua a máxima robustez alcançável (definição 4.2), é a utilização do maior número de bases possível, e, com isso, gerar um número maior de possibilidades de perdas de sensores que serão cobertas pelo diagnosticador.

Uma vez encontradas todas as bases mínimas para a diagnose de falhas do SED considerado, é possível obter todas as bases ditas *não-mínimas* para a diagnose, isto é, as bases para a diagnose formadas acrescentando-se eventos redundantes às bases mínimas. Com isso, o maior número possível de combinações de eventos observáveis que permitem que a diagnose de L seja realizada estará sendo utilizado. Tal fato leva à seguinte definição.

Definição 4.6 *(Diagnosticador união) Seja E_{bd} o conjuntos de todas as bases para a diagnose de um SED e sejam os conjuntos $E_{o_1}, E_{o_2}, \ldots, E_{o_n}$ bases para a diagnose de falhas, isto é, $E_{o_i} \in E_{bd}$, para $i = 1, \ldots, n$, sendo $n \leq |E_{bd}|$. Denote por $\tilde{G}_{d_i}$, para $i = 1, \ldots, n$, os diagnosticadores parciais com marcações de perdas de sensores. O diagnosticador união com relação a $E_{o_1}, E_{o_2}, \ldots, E_{o_n}$ é o diagnosticador cuja linguagem gerada é a união das linguagens geradas pelos diagnosticadores $\tilde{G}_{d_i}$, para $i = 1, \ldots, n$.* □

O exemplo a seguir ilustra a construção de um diagnosticador união.

Exemplo 4.2 *(Construção de um diagnosticador união) Considere o mesmo autômato do exemplo 4.1, mostrado na figura 4.1. As bases mínimas para a diagnose de $\mathcal{L}(G)$ foram calculadas, e formam o seguinte conjunto:*

$$E_{bmd} = \{\{a, b, c\}, \{c, d, f\}, \{a, c, d\}, \{a, d, f\}, \{a, b, f\}, \{b, c, f\}\}.$$

Acrescentar às bases mínimas os eventos que não pertencem a elas, mas que estão no conjunto de eventos observáveis do autômato, obtém-se as bases não-mínimas, agrupadas no seguinte conjunto:

$$E_{bnmd} = \{\{a, b, c, d\}, \{a, b, c, f\}, \{a, b, c, d, f\}, \{a, c, d, f\}, \{b, c, d, f\}, \{a, b, d, f\}\}.$$

Suponha que se deseje encontrar o diagnosticador união com relação a todos os elementos do conjunto $E_{bmd} \cup E_{bnmd}$. Para tanto, o primeiro passo é encontrar todos os diagnosticadores parciais considerando as bases para a diagnose de falhas como eventos observáveis, inserindo as marcações de perda de observabilidade de eventos,

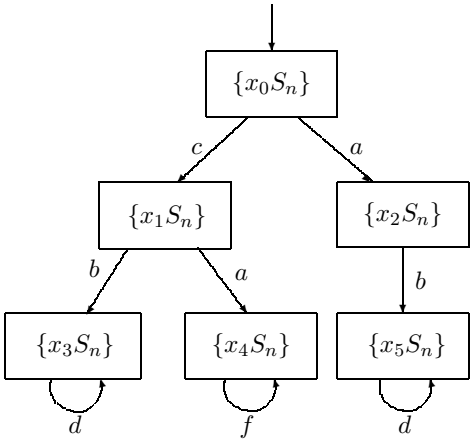

Figura 4.5: Diagnosticador centralizado G_d.

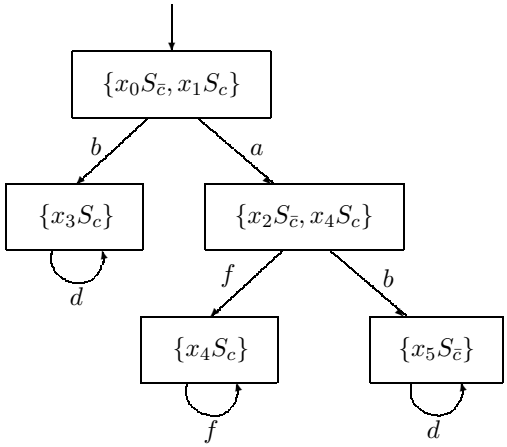

Figura 4.6: Diagnosticador parcial para a base $E'_o = \{a, b, d, f\}$ (falha de observação do evento c).

como descrito no algoritmo 4.1. Tais diagnosticadores estão mostrados nas figuras de 4.5 a 4.16. Com os diagnosticadores parciais e centralizado construídos, a próxima etapa consiste em gerar o diagnosticador união utilizando-se esses diagnosticadores. Aplicando-se o algoritmo 2.2, obtém-se o diagnosticador união com relação aos elementos de $E_{bmd} \cup E_{bnmd}$, mostrado na figura 4.17. É importante ressaltar que todas as bases para a diagnose de falhas foram utilizadas na construção desse diagnosticador. □

No exemplo 4.2, pode-se notar a presença de autolaços nos estados $\{x_3S_c; x_5S_a\} = \{4YS_c; 6NS_a\}$ e $\{x_5S_{ab}; x_3S_{bc}\} = \{6NS_{ab}; 4YS_{bc}\}$. Se o diagnosticador união considerado alcançar esses estados, a ocorrência da falha não poderá ser afirmada, pois o estado x_3 é um estado certo e o estado x_5 é um estado normal de G_d. Além disso, no estado $\{x_3S_c; x_5S_a; x_3S_{c\bar{d}}, x_3S_{cd}; x_5S_{a\bar{d}}, x_5S_{ad}\}$, as componentes referentes aos diagnosticadores parciais considerando falhas na observabilidade dos eventos c e d, e a e d não possuem eventos ativos, o que significa que nunca alcan-

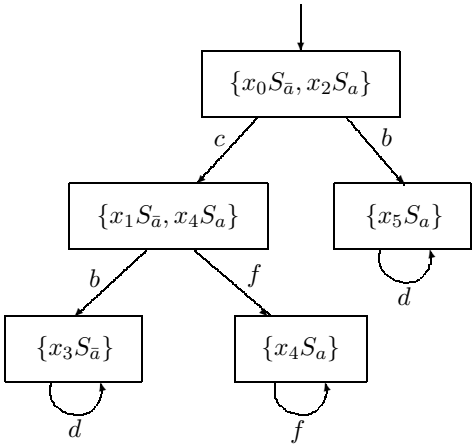

Figura 4.7: Diagnosticador parcial para a base $E'_o = \{b, c, d, f\}$ (falha de observação do evento a).

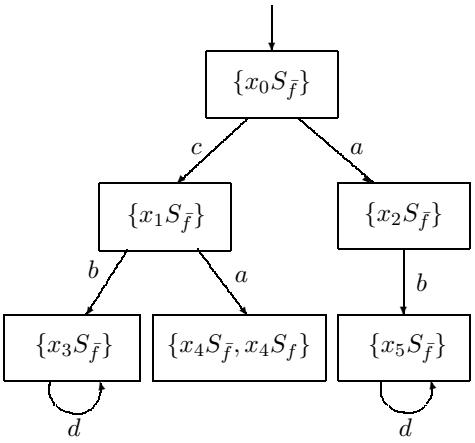

Figura 4.8: Diagnosticador parcial para a base $E'_o = \{a, b, c, d\}$ (falha de observação do evento f).

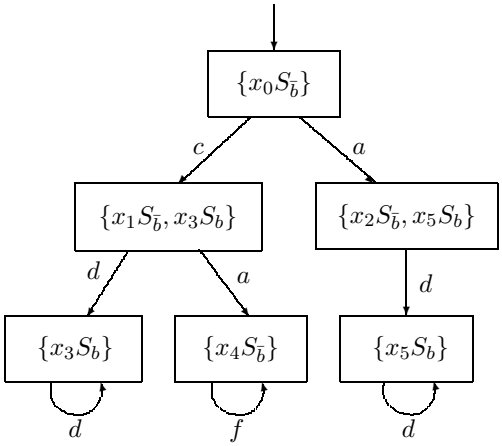

Figura 4.9: Diagnosticador parcial para a base $E'_o = \{a, c, d, f\}$ (falha de observação do evento b).

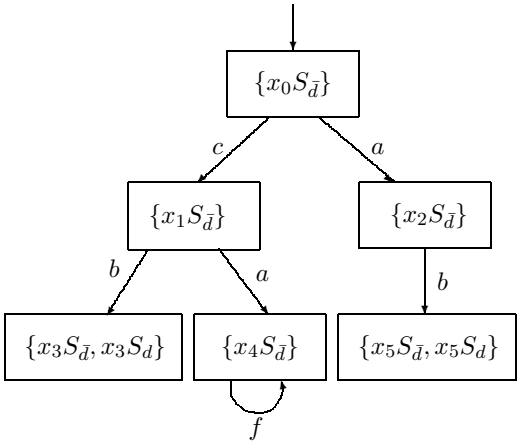

Figura 4.10: Diagnosticador parcial para a base $E'_o = \{a, b, c, f\}$ (falha de observação do evento d).

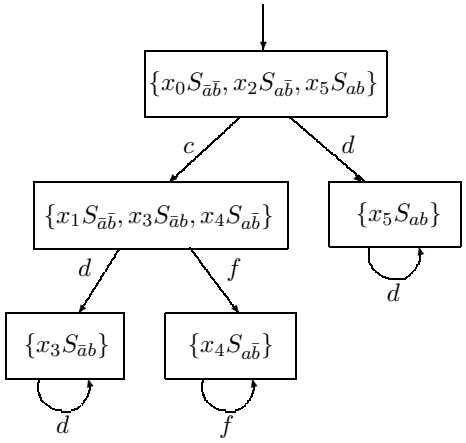

Figura 4.11: Diagnosticador parcial para a base $E'_o = \{c, d, f\}$ (falha de observação dos eventos a e b).

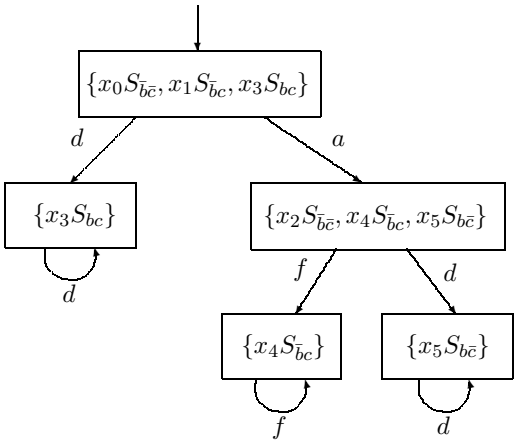

Figura 4.12: Diagnosticador parcial para a base $E'_o = \{a, d, f\}$ (falha de observação dos eventos b e c).

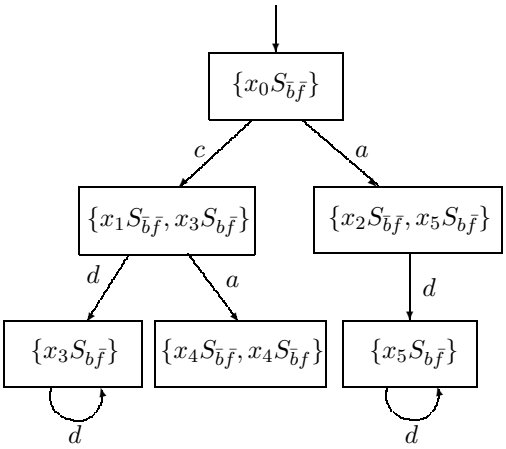

Figura 4.13: Diagnosticador parcial para a base $E'_o = \{a, c, d\}$ (falha de observação dos eventos b e f).

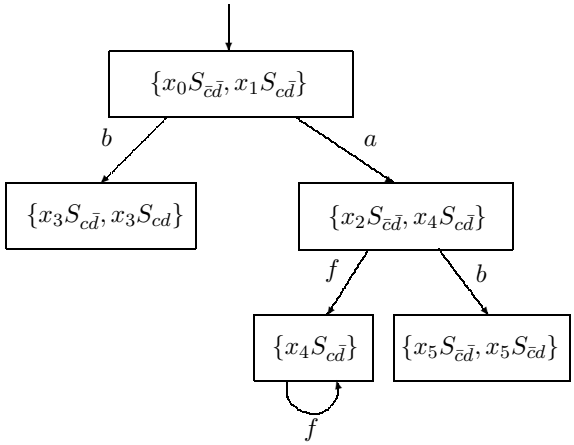

Figura 4.14: Diagnosticador parcial para a base $E'_o = \{a, b, f\}$ (falha de observação dos eventos c e d).

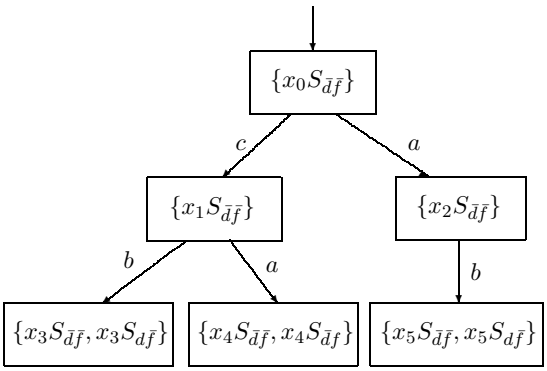

Figura 4.15: Diagnosticador parcial para a base $E'_o = \{a, b, c\}$ (falha de observação dos eventos d e f).

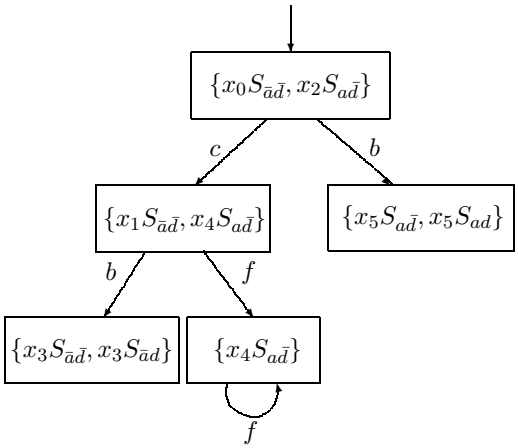

Figura 4.16: Diagnosticador parcial para a base $E'_o = \{b, c, f\}$ (falha de observação dos eventos a e d).

çariam um estado certo no caso da falha ter ocorrido. Esse é mais um indício de perda de diagnosticabilidade, uma vez que os estados dos diagnosticadores parciais que não possuem eventos ativos, possuem, pelo menos, um ciclo escondido. Com isso, pode-se definir, formalmente, estado incerto e ciclo indeterminado (observado e escondido) de um diagnosticador união.

Definição 4.7 *(Estado incerto de um diagnosticador união) Um estado do diagnosticador união G_{d_u} será incerto se for composto por, pelo menos, um estado certo e um estado normal ou incerto de diagnosticadores parciais diferentes.* □

Definição 4.8 *(Ciclos indeterminados observados e escondidos do diagnosticador união)*

A. *Um ciclo observado indeterminado do diagnosticador união é um ciclo observado do referido diagnosticador composto por estados incertos.*

B. *Existe um ciclo escondido indeterminado em um estado incerto do diagnosticador união se este estado incerto contiver um estado certo de algum diagnosticador parcial no qual há um ciclo escondido.* □

Como no caso da diagnosticabilidade centralizada sem perda de observabilidade de eventos (teorema 2.1), é possível obter condições necessárias e suficientes que um diagnosticador união deve conter para que a linguagem L seja diagnosticável sob a perda permanente de observabilidade de eventos redundantes das bases não-mínimas utilizadas na construção do diagnosticador união considerado. Essas condições são impostas dadas teorema a seguir.

Teorema 4.1 *Seja G um autômato e suponha que $E_{o_1}, E_{o_2}, \ldots, E_{o_n} \subset E_o$ e que a linguagem L gerada por G seja diagnosticável em relação a $P_{o_1}, P_{o_2}, \ldots, P_{o_n}$ e E_f,*

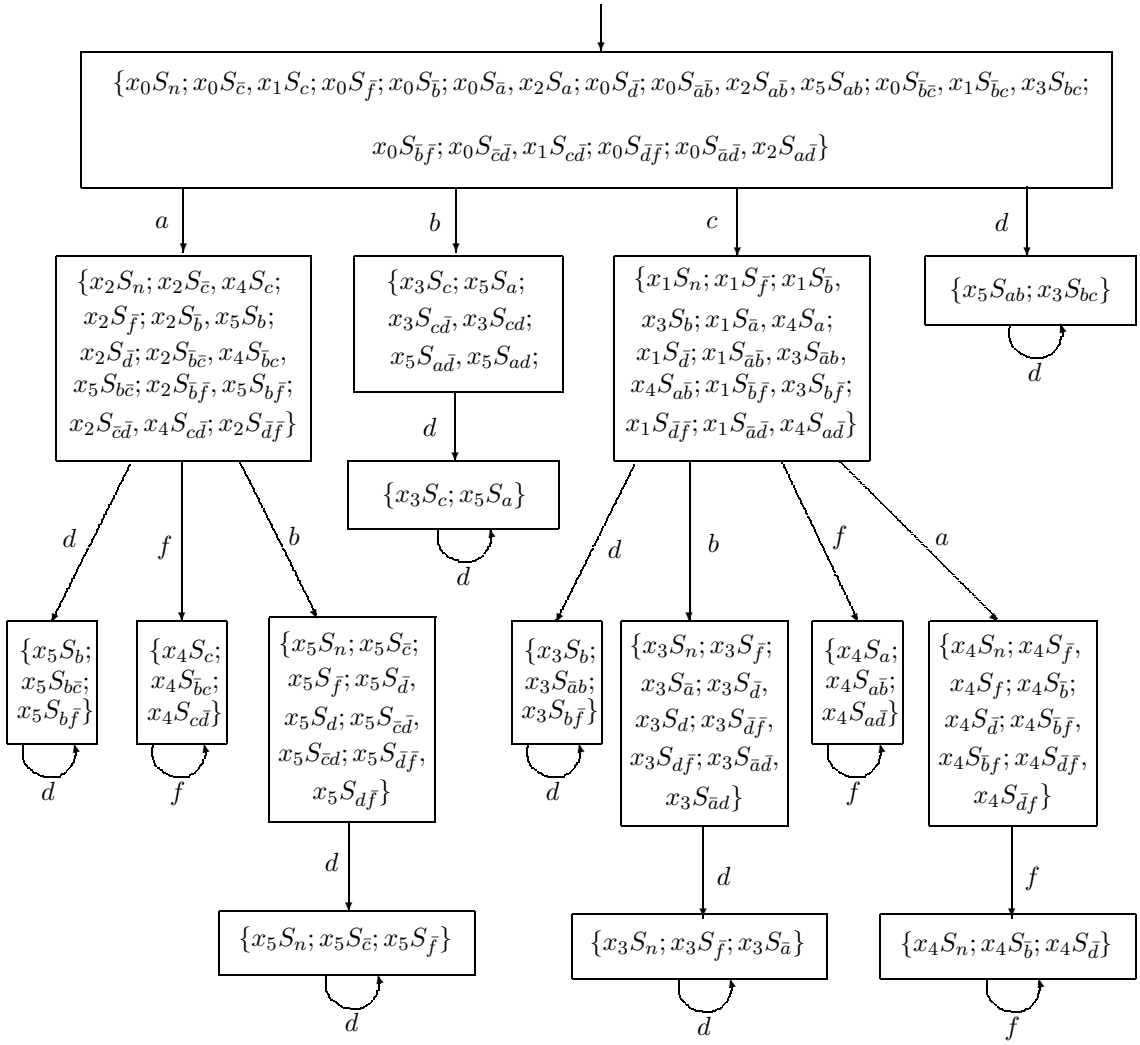

Figura 4.17: Diagnosticador união.

individualmente. Então, L será diagnosticável em relação à perda permanente de observabilidade dos conjuntos de eventos pertencentes a $E_{rob} = E_{red}(E_{o_1}) \cup E_{red}(E_{o_2}) \cup \ldots \cup E_{red}(E_{o_n})$*, em relação às projeções* $P_{o_1}, P_{o_2}, \ldots, P_{o_n}$ *e* $E_f = \{\sigma_f\}$ *se e somente se o diagnosticador união em relação aos conjuntos* $E_{o_1}, E_{o_2}, \ldots, E_{o_n}$ *não tiver nenhum ciclo indeterminado (observado ou escondido).*

Demonstração: ($\Rightarrow$) Suponha que o diagnosticador união considerado não possua ciclos indeterminados (observados ou escondidos) e que a linguagem L não seja diagnosticável sob a perda permanente de observabilidade em relação a $P_{o_1}, P_{o_2}, \ldots, P_{o_n}$ e $E_f = \{\sigma_f\}$. Portanto, existe uma sequência s', arbitrariamente longa, pertencente a L, tal que σ_f pertence a s' e uma sequência s'' pertencente a $P^{-1}_{o_{j_L}}(P_{o_i}(s'))$, para algum i e algum j, $i \neq j$, tal que σ_f não pertence a s''. Como s'' pertence a $P^{-1}_{o_{j_L}}(P_{o_i}(s'))$, pela definição da projeção inversa, $P_{o_j}(s'') = P_{o_i}(s')$.

Seja $L_i = P_{o_i}(L)$ a linguagem gerada pelo diagnosticador parcial que considera a base E_{o_i} como conjunto de eventos observáveis e considere a sequência $P_{o_i}(s')$

pertencente a L_i que leva esse diagnosticador a girar em um ciclo de estados certos ou leva a um estado certo (ciclo escondido, porém não indeterminado), dado que esse conjunto é uma base para a diagnose de L. Por outro lado, a linguagem gerada pelo diagnosticador parcial que considera a base E_{o_j} como conjunto de eventos observáveis é $L_j = P_{o_j}(L)$, e a sequência $P_{o_j}(s'')$ pertencente L_j leva esse diagnosticador a girar em um ciclo de estados normais ou de estados incertos (que não seja indeterminado) ou leva a um estado normal ou incerto (ciclos escondidos, porém não indeterminados). Como a linguagem de um diagnosticador união é a união das linguagens geradas pelos diagnosticadores parciais e centralizado que o compõem, e pela forma de construção de um diagnosticador união, pode-se concluir que a sequência $P_{o_i}(s') = P_{o_j}(s'')$ girará em um ciclo indeterminado observado ou levará a um estado incerto que possui um ciclo indeterminado escondido nesse diagnosticador, o que leva a uma contradição.
($\Leftarrow$) Suponha que a linguagem L seja diagnosticável sob a perda permanente de observabilidade de eventos em relação a $P_{o_1}, P_{o_2}, \ldots, P_{o_n}$ e $E_f = \{\sigma_f\}$, e que, inicialmente, o diagnosticador união em relação a $E_{o_1}, E_{o_2}, \ldots, E_{o_n}$ possua um ciclo indeterminado observado. Isso implica que existe uma sequência arbitrariamente longa s tal que, ao menos, em um diagnosticador parcial essa sequência gira em um ciclo de estados certos (diagnosticador cujo conjunto de eventos observáveis é E_{o_i}), enquanto que em outro diagnosticador parcial essa sequência gira em um ciclo de estados normais ou em um ciclo de estados incertos que não seja indeterminado (diagnosticador cujo conjunto de eventos observáveis é E_{o_j}). Logo, existe uma sequência s' pertencente a L, que possui σ_f, e uma sequencia s'' pertencente a L, que não possui σ_f, tais que $P_{o_i}(s') = P_{o_j}(s'') = s$, o que viola a condição de diagnosticabilidade sob perda permanente de observabilidade de eventos. Esse fato leva a uma contradição.

Suponha agora que a linguagem L seja diagnosticável sob a perda permanente de observabilidade de eventos em relação a $P_{o_1}, P_{o_2}, \ldots, P_{o_n}$ e $E_f = \{\sigma_f\}$, e que o diagnosticador união construído utilizando-se as bases $E_{o_1}, E_{o_2}, \ldots, E_{o_n}$ possua um ciclo escondido indeterminado. Isso implica que existe uma sequencia finita s que leva um diagnosticador parcial a um estado certo (diagnosticador cujo conjunto de eventos observáveis é E_{o_i}) e que leva outro diagnosticador parcial a um estado normal ou incerto (diagnosticador cujo conjunto de eventos observáveis é E_{o_j}). Como L é uma linguagem viva, é possível definir uma sequência arbitrariamente longa pertencente a $P^{-1}_{o_{i_L}}(s)$ que possui σ_f. Além disso, é possível definir uma sequência finita ou arbitrariamente longa pertencente a $P^{-1}_{o_{j_L}}(s)$ que não possui σ_f. Esse fato viola a condição de diagnosticabilidade sob perda permanente de observabilidade de eventos, levando a uma contradição. □

Portanto, se um diagnosticador união não possuir ciclos indeterminados (obser-

vados ou escondidos), então este pode ser utilizado na diagnose de falhas do SED para a qual foi construído. Além disso, esse diagnosticador será robusto à perda dos sensores responsáveis pelo registro da ocorrência dos eventos pertencentes aos conjuntos que constituem $E_{rob} = E_{red}(E'_{o_1}) \cup E_{red}(E'_{o_2}) \cup \ldots \cup E_{red}(E'_{o_n})$. Esse fato leva à seguinte definição.

Definição 4.9 *(Diagnosticador robusto) O diagnosticador união construído utilizando-se os diagnosticadores que consideram os conjuntos $E_{o_1}, E_{o_2}, \ldots, E_{o_n}$ como conjuntos de eventos observáveis será robusto à perda de observabilidade de um conjunto de eventos de $E_{rob} = E_{red}(E_{o_1}) \cup E_{red}(E_{o_2}) \cup \ldots \cup E_{red}(E_{o_n})$ se e somente se não tiver ciclos indeterminados (observados ou escondidos).* □

Uma vez que o diagnosticador união construído utilizando-se todas as bases para a diagnose de um SED pode não ser robusto, segundo a definição 4.9, na próxima seção será apresentada uma solução para se encontrar o diagnosticador que possua a máxima robustez possível a partir do diagnosticador união construído utilizando-se todas as bases para a diagnose do SED.

4.3 Diagnosticador de máxima robustez

Como a linguagem L do SED é, por hipótese, diagnosticável, então a presença de ciclos indeterminados, tanto observados como escondidos, no diagnosticador união formado utilizando-se todas as bases para a diagnose de falhas (máxima robustez teórica) se deve a uma incompatibilidade de diagnosticadores parciais. Essa incompatibilidade se dá pelo fato de um dado diagnosticador parcial alcançar um estado certo e um outro diagnosticador parcial alcançar um estado normal através da observação da mesma sequência de eventos, ou pela presença de ciclos escondidos em estados certos de diagnosticadores parciais que compõem estados incertos do diagnosticador união considerado.

Porém, como a linguagem L é originalmente diagnosticável, ao menos um diagnosticador sem ciclos indeterminados existe para essa linguagem. Esse diagnosticador é o diagnosticador centralizado construído para se verificar a diagnosticabilidade de L em relação a P_o e E_f. Portanto, uma alternativa para se chegar a um diagnosticador que seja robusto a partir do diagnosticador união que utiliza todas as bases para a diagnose é deixar de utilizar as bases que levam à ambiguidade nesse diagnosticador união.

Nesse ponto, as marcações de perda de observabilidade dos eventos inicialmente observáveis definidas na construção do diagnosticador união serão justificadas. Elas indicarão as bases que estão gerando os ciclos indeterminados no diagnosticador robusto. Assim, escolhendo-se algumas bases para serem retiradas, gera-se novamente

o diagnosticador cuja linguagem gerada é a união das linguagens geradas pelos diagnosticadores parciais considerando as bases restantes como eventos observáveis. Portanto, esse novo diagnosticador será robusto somente à perda de observabilidade dos conjuntos de eventos redundantes de cada base não-mínima para a diagnose que o compõe.

Considere novamente o caso do exemplo 4.2, no exemplo abaixo.

Exemplo 4.3 *(Diagnosticador de máxima robustez para o SED do exemplo4.2) Considere o diagnosticador união mostrado na figura 4.17. Como mencionado anteriormente, os estados* $\{x_3S_c; x_5S_a\}$ *e* $\{x_5S_{ab}; x_3S_{bc}\}$ *formam ciclos observados indeterminados devido às bases* $\{a, b, d, f\}$ *e* $\{b, c, d, f\}$ *para o primeiro ciclo e, às bases* $\{c, d, f\}$ *e* $\{a, d, f\}$ *para o segundo ciclo. Já o estado* $\{x_3S_c; x_5S_a; x_3S_{c\bar{d}}, x_3S_{cd}; x_5S_{a\bar{d}}, x_5S_{ad}\}$ *possui ciclos escondidos indeterminados devido às bases* $\{a, b, f\}$, $\{b, c, f\}$ *e* $\{b, c, d, f\}$, *já que o estado* $\{x_3S_{c\bar{d}}, x_3S_{cd}\}$ *possui um ciclo escondido no diagnosticador parcial referente a essa base, e os estados referentes aos diagnosticadores das outras duas bases levam a um estado normal após o evento b ocorrer nos respectivos estados iniciais. Logo, devem-se retirar uma das bases que geram cada ciclo observado indeterminado e base que geram o ciclo escondido indeterminado de forma a tornar o estado incerto que possui o ciclo escondido um estado normal ou certo. Para os ciclos observados indeterminados retirar-se-ão as bases que informam que a falha não ocorreu, por questão de filosofia de diagnose, dado que pode ser mais seguro informar que a falha ocorreu quando, de fato, não ocorreu, do que o contrário. Logo, as bases a serem retiradas serão* $\{b, c, d, f\}$ *e* $\{c, d, f\}$, *para extinguir esses ciclos observados indeterminados. Já para o ciclo escondido indeterminado, devem-se retirar todas as bases que façam com que o estado incerto seja incerto, tornando-o um estado certo ou um estado normal. Para tanto, nesse caso, retirar-se-á a base* $\{b, c, f\}$, *para fazer com que este estado incerto tenha somente componentes que sejam estados certos. Desse modo, as bases para a diagnose robusta serão*

$$\begin{aligned} E_{bdr} \;=\; & \{\{a,b,c\},\{a,c,d\},\{a,d,f\},\{a,b,f\},\{a,b,c,d\},\{a,b,c,f\},\\ & \{a,b,d,f\},\{a,c,d,f\}\{a,b,c,d,f\}\}. \end{aligned}$$

Note que o diagnosticador a ser construído com as bases acima relacionadas não será robusto à perda de observabilidade dos eventos a e c, dado que todas as bases do conjunto E_{bdr} *possuem esses dois eventos. O diagnosticador robusto sem ciclos indeterminados está mostrado na figura 4.18. Como era de se esperar, os ciclos que geravam indeterminação sobre a ocorrência da falha não estão presentes nesse diagnosticador. Note que o diagnosticador sempre alcança um estado certo caso a falha tenha ocorrido, no caso de perda somente dos eventos cujas bases estão*

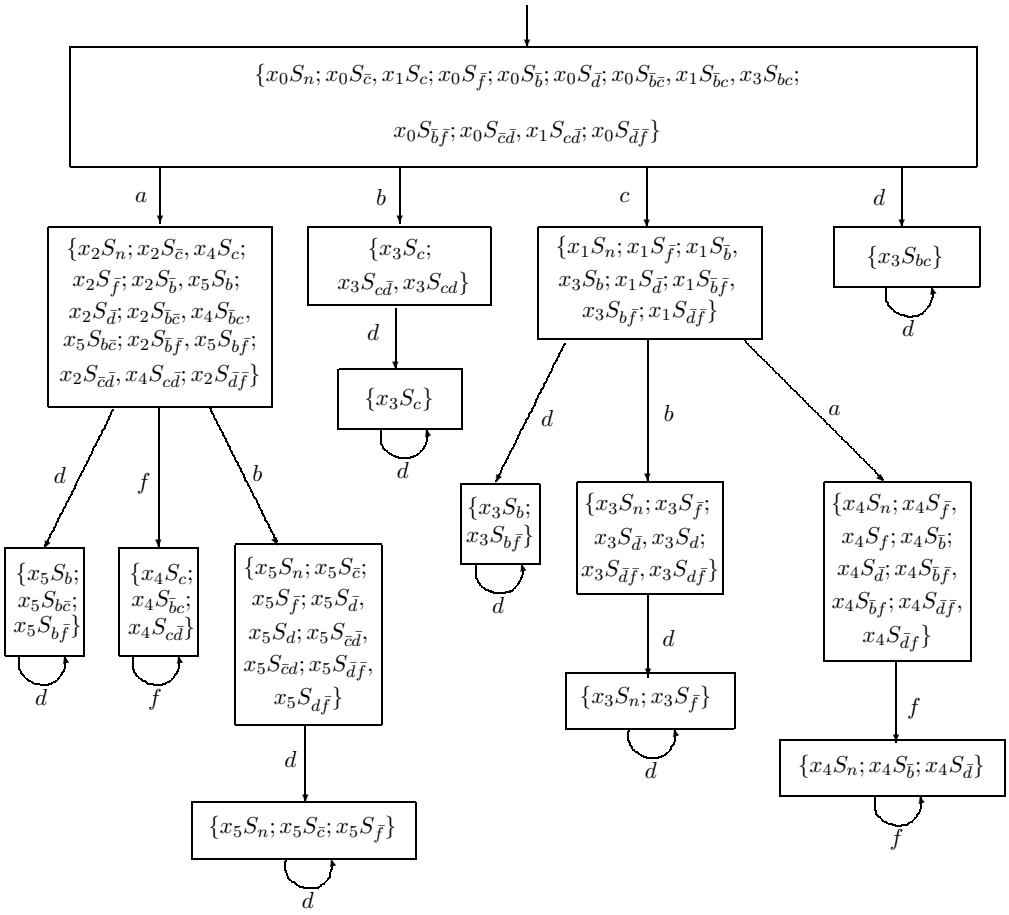

Figura 4.18: Diagnosticador de máxima robustez.

presentes nesse diagnosticador, ou seja, em caso de perda de observação dos seguintes conjuntos de eventos: $E_{rob} = \{\{b\}, \{c\}, \{d\}, \{f\}, \{c, d\}, \{b, c\}, \{b, f\}, \{d, f\}\}$*, que são, justamente, os conjuntos de eventos redundantes das bases para a diagnose que compõem o diagnosticador robusto.* □

Observação 4.1 *Note que, no diagnosticador centralizado com observação total mostrado na figura 4.5, se o sensor que verifica a ocorrência do eventos c falhar, e ocorrer a sequência ca, esse diagnosticador informará, de forma errada, que a falha ocorreu, indo para o estado* x_4*. O mesmo pode ocorrer no diagnosticador robusto, se um sensor responsável por detectar a ocorrência de um evento que não seja redundante vier a falhar. Por exemplo, se o sensor de detecção de ocorrência do evento a falhar e ocorrer a sequência abd, o diagnosticador robusto informará, de forma equivocada, que a falha ocorreu, indo para o estado* $\{x_3S_c\}$*. Da mesma forma que o diagnosticador centralizado com observação total não é robusto à falha de nenhum sensor, o diagnosticador proposto neste trabalho não será robusto à perda de sensores responsáveis por detectar eventos que não sejam redundantes.* □

4.4 Comentários finais

Como pôde-se observar no exemplo 4.3, o diagnosticador robusto é capaz de informar sobre a ocorrência da falha mesmo com a perda de funcionamento de alguns sensores. Além disso, esse diagnosticador utiliza o mesmo número de sensores que o diagnosticador centralizado necessita para a diagnose de L. Logo, fica clara a vantagem de se utilizar esse diagnosticador robusto ao invés do diagnosticador centralizado com observação total.

Capítulo 5

Conclusões e trabalhos futuros

Neste trabalho foi proposto um algoritmo para a busca sistemática de todas as bases mínimas para a diagnose de falhas de um SED, permitindo assim que a diagnose de falhas de um SED seja realizada com um número mínimo de sensores, o que pode representar uma grande economia de recursos para o projeto de diagnose de falhas de um sistema real. Todos os passos do algoritmo de busca foram teoricamente justificados com base em condições necessárias.

Uma outra contribuição desse trabalho é a utilização das bases para a diagnose de falhas no projeto de um diagnosticador que seja robusto à perda permanente de sensores Além disso, pode-se cogitar a hipótese de inserção de sensores redundantes a fim de aumentar a confiabilidade da observação de um dado evento essencial para a diagnose de falhas. Os resultados sobre o diagnosticador robusto apresentados no capítulo 4 demonstram que a ocorrência da falha em um SED pode ser detectada mesmo com a perda de observabilidade de algun(s) evento(s) observável(eis), utilizando-se os mesmos sensores que seriam necessários para se realizar a diagnose centralizada através do diagnosticador centralizado apresentado por SAMPATH *et al.* [3]. Com isso, pode-se concluir que, no diagnosticador robusto, a informação recebida pelos sensores é organizada de forma a permitir a perda da informação sobre a ocorrência de alguns eventos.

Além das contribuições já citadas, este trabalho apresenta novos conceitos, definições e resultados a serem incorporados à teoria de SEDs, como trajetórias primas, trajetórias com ciclos inerentes, cobertura para trajetórias com ciclos inerentes, conjuntos de eventos elementares para a diagnose, bases para a diagnose, diagnosticabilidade sob perda permanente de observabiidade de eventos etc. Esses novos conceitos e resultados podem, futuramente, também ser utilizados em outros contextos além da diagnose de falhas, como por exemplo, em controle supervisório.

Pode-se citar como continuações imediatas deste trabalho: *(i)* o desenvolvimento de algoritmos capazes de lidar com os dois possíveis casos de ciclos observados indeterminados apresentados na subseção 3.3.2, não abordados neste trabalho; *(ii)*

o aprofundamento dos estudos sobre a robustez do diagnosticador apresentado no capítulo 4; e *(iii)* o cálculo da complexidade computacional do algoritmo de busca pelas bases mínimas para a diagnose proposto nesse texto. Outros trabalhos podem ser desenvolvidos mais a frente, como a utilização do diagnosticador robusto para o apontamento dos sensores que falharam durante o processo de diagnose de falhas, contribuindo para manutenção do sistema de diagnose, e a utilização do diagnosticador robusto no processo de recuperação do SED após uma falha ser detectada (*fault recovery*).

Referências Bibliográficas

[1] SAMPATH, M., SENGUPTA, R., LAFORTUNE, S., et al. "Failure diagnosis using discrete event models", *IEEE Trans. on Control Systems Technology*, v. 4, pp. 105–124, 1996.

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

[3] SAMPATH, M., SENGUPTA, R., LAFORTUNE, S., et al. "Diagnosability of discrete-event systems", *IEEE Trans. on Automatic Control*, v. 40, pp. 1555–1575, 1995.

[4] SAMPATH, M. "A Hybrid Aproach to Failure Diagnosis of Industrial Systems". In: *Proc. of the American Control Conference*, pp. 2077–2082, Arlington, VA, 2001.

[5] SAMPATH, M., LAFORTUNE, S., , et al. "Active diagnosis of discrete-event systems", *IEEE Trans. on Automatic Control*, v. 43, pp. 908–929, 1998.

[6] DEBOUK, R., LAFORTUNE, S., TENEKETZIS, D. "Coordinated decentralized protocols for failure diagnosis of discrete event systems", *Discrete Event Dynamic Systems: Theory and Applications*, v. 10, pp. 33–86, 2000.

[7] CONTANT, O., LAFORTUNE, S., TENEKETZIS, D. "Diagnosability of discrete event systems with modular structure", *Discrete Event Dynamic Systems-Theory And Applications*, v. 16, n. 1, pp. 9–37, 2006.

[8] JIANG, S., KUMAR, R., GARCIA, H. E. "Optimal sensor selection for discrete-event systems with partial observation", *IEEE Transactions on Automatic Control*, v. 48, n. 3, pp. 369–381, March 2003.

[9] LUNZE, J., SCHRODER, J. "State observation and diagnosis of discrete-event systems described by stochastic automata", *Discrete Event Dynamic Systems-Theory And Applications*, v. 11, n. 4, pp. 319–369, 2001.

[10] THORSLEY, D., TENEKETZIS, D. "Diagnosability of Stochastic Discrete-Event Systems", *IEEE Trans. on Automatic Control*, v. 50, pp. 476–492, 2005.

[11] QIU, W., KUMAR, R. "Decentralized failure diagnosis of discrete event systems", *IEEE Transactions on Systems, Man and Cybernetics, Part A*, v. 36, n. 2, pp. 384–395, March 2006. doi: 10.1109/TSMCA.2005.853503.

[12] JIANG, S., KUMAR, R., GARCIA, H. E. "Diagnosis of repeated/intermittent failures in discrete event systems", *IEEE Transactions on Robotics and Automation*, v. 19, n. 2, pp. 310–323, April 2003. doi: 10.1109/TRA.2003.809590.

[13] KILIC, E. "Diagnosability of fuzzy discrete event systems", *Information Sciences*, v. 178, n. 3, pp. 858–870, 2008.

[14] ZAD, S. H., KWONG, R., WONHAM, W. "Fault diagnosis in discrete-event systems: incorporating timing information", *Automatic Control, IEEE Transactions on*, v. 50, n. 7, pp. 1010–1015, July 2005. ISSN: 0018-9286. doi: 10.1109/TAC.2005.851444.

[15] HOPCCROFT, J. E., MOTWANI, R., ULLMAN, J. D. *Introduction to automata theory, languages, and computation*. 3rd ed. Boston, Addison Wesley, 2007.

[16] PETERSON, J. *Petri net theory and the modeling of systems*. Englewood Cliffs, NJ, Prentice Hall, 1981.

[17] MURATA, T. "Petri nets - properties, analysis and applications", *Proceedings of the IEEE*, v. 77, n. 4, pp. 541–580, 1989.

[18] DAVID, R., ALLA, H. *Discrete, Continuous, and Hybrid Petri Nets*. New York, NY, Springer, 2005.

[19] LIN, F. "Diagnosability of discrete event systems and its applications", *Journal of Discrete Event Dynamic Systems*, v. 4, pp. 197–212, 1994.

[20] RAMADGE, P. J., WONHAM, W. M. "The control of discrete-event systems", *Proceedings of the IEEE*, v. 77, pp. 81–98, 1989.

[21] LIN, F., WONHAM, W. M. "Supervisory control and coordination of discrete-event systems with partial observation", *IEEE Transactions on Automatic Control*, v. 35, pp. 1330–1337, 1990.

[22] BASILIO, J. C., LAFORTUNE, S. "Robust codiagnosability of discrete event systems". In: *Proceedings of the American Control Conference*, pp. 2202–2209, St. Louis, Missouri, 2009.

[23] JIANG, S., HUANG, Z., CHANDRA, V., et al. "A polynomial algorithm for testing diagnosability of discrete-event systems", *IEEE Transactions on Automatic Control*, v. 46, n. 8, pp. 1318–1321, Aug. 2001. doi: 10.1109/9.940942.

[24] YOO, T.-S., LAFORTUNE, S. "Polynomial-time verification of diagnosability of partially observed discrete-event systems", *IEEE Transactions on Automatic Control*, v. 47, n. 9, pp. 1491–1495, Sep 2002. doi: 10.1109/TAC.2002.802763.

[25] WANG, Y., YOO, T. S., LAFORTUNE, S. "Diagnosis of discrete event systems using decentralized architectures", *Discrete Event Dynamic Systems-Theory And Applications*, v. 17, n. 2, pp. 233–263, jun. 2007. doi: 10.1007/s10626-006-0006-8.

[26] USHIO, T., ONISHI, I., OKUDA, K. "Fault detection based on Petri net models with faulty behaviors". In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, v. 1, pp. 113–118, San Diego, 1998.

[27] CHUNG, S.-L., WU, C.-C., JENG, M. "Failure diagnosis: A case study on modeling and analysis by Petri nets". In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, v. 3, pp. 2727–2732, Washington, 2003.

[28] GIUA, A., SEATZU, C. "Fault detection for discrete event systems using Petri nets with unobservable transitions". In: *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, CDC-ECC 2005*, v. 2005, pp. 6323–6328, Seville, Spain, 2005.

[29] RAMIREZ-TREVINO, A., RUIZ-BELTRAN, E., RIVERA-RANGEL, I., et al. "Diagnosability of discrete event systems. A Petri Net based approach". In: *Proceedings of IEEE International Conference on Robotics and Automation*, v. 2004, pp. 541–546, New Orleans, LA, 2004.

[30] GENC, S., LAFORTUNE, S. "Distributed diagnosis of place-bordered Petri nets", *IEEE Transactions on Automation Science and Engineering*, v. 4, n. 2, pp. 206–219, 2007.

[31] MANYARI-RIVERA, M., BASILIO, J. C., BHAYA, A. "Integrated fault diagnosis based on Petri net models". In: *Proceedings of the 16th IEEE International Conference on Control Applications*, pp. 958–963, Singapore, 2007.

[32] TRAVÉ-MASSUYÈS, L., ESCOBET, T., OLIVE, X. "Diagnosability analysis based on component-supported analytical redundancy relations", *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, v. 36, pp. 1146–1160, 2006.

[33] DEBOUK, R., LAFORTUNE, S., TENEKETZIS, D. "On an optimization problem in sensor selection", *Discrete Event Dynamic Systems: Theory and Applications*, v. 12, pp. 417–445, 2002.