Este trabalho trata o problema da controlabilidade segura de sistemas a eventos discretos, o qual consiste em evitar a ocorrência de eventos ou sequências de eventos que podem levar o sistema a um comportamento proibido após a ocorrência de uma falha. Para tanto, utiliza-se uma abordagem por cadeias e introduzem-se os conceitos de cadeia diagnosticável, cadeia diagnosticável segura, cadeia prognosticável, cadeia controlável segura pela diagnose e cadeia controlável segura pela prognose. São apresentadas também condições necessárias e suficientes para garantir tais propriedades. A partir desses conceitos, define-se a controlabilidade segura de uma linguagem pela diagnose ou prognose, segundo a qual uma linguagem é DP-Controlável Segura se cada uma das cadeias que contém a falha é controlável segura pela diagnose ou é controlável segura pela prognose. Por fim, são apresentadas condições necessárias e suficientes para que uma linguagem seja DP-Controlável Segura.

Orientador: Dr. André Bittencourt Leal

JOINVILLE, 2019

ANO
2019

ANA TERUKO YOKOMIZO WATANABE | CONTROLABILIDADE SEGURA DE SISTEMAS A EVENTOS DISCRETOS UTILIZANDO DIAGNOSE E PROGNOSE ONLINE

UDESC

UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA ELÉTRICA

TESE DE DOUTORADO

CONTROLABILIDADE SEGURA DE SISTEMAS A EVENTOS DISCRETOS UTILIZANDO DIAGNOSE E PROGNOSE ONLINE

ANA TERUKO YOKOMIZO WATANABE

JOINVILLE, 2019

ANA TERUKO YOKOMIZO WATANABE

CONTROLABILIDADE SEGURA DE SISTEMAS A EVENTOS DISCRETOS UTILIZANDO DIAGNOSE E PROGNOSE ONLINE

Tese submetida ao Curso de Pós-Graduação em Engenharia Elétrica, do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, para a obtenção do Grau de Doutor em Engenharia Elétrica.

Orientador: Prof. Dr. André Bittencourt Leal

JOINVILLE

2019

Ficha catalográfica elaborada pelo programa de geração automática da

Biblioteca Setorial do CCT/UDESC,

com os dados fornecidos pelo(a) autor(a)

Watanabe, Ana Teruko Yokomizo
CONTROLABILIDADE SEGURA DE SISTEMAS A EVENTOS DISCRETOS UTILIZANDO DIAGNOSE E PROGNOSE ONLINE / Ana Teruko Yokomizo Watanabe. -- 2019.
144 p.

Orientador: André Bittencourt Leal
Tese (doutorado) -- Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, Programa de Pós-Graduação em Engenharia Elétrica, Joinville, 2019.

1. Diagnose de falhas. 2. Prognose de falhas. 3. Controlabilidade segura. 4. Controle tolerante a falhas. 5. Sistemas a Eventos Discretos. I. Leal, André Bittencourt. II. Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

Controlabilidade Segura de Sistemas a Eventos Discretos Utilizando Diagnose e Prognose Online

por

Ana Teruko Yokomizo Watanabe

Esta tese foi julgada adequada para obtenção do título de

DOUTOR EM ENGENHARIA ELÉTRICA

Área de concentração em "Sistemas Eletroeletrônicos"
e aprovada em sua forma final pelo

CURSO DE DOUTORADO EM ENGENHARIA ELÉTRICA
DO CENTRO DE CIÊNCIAS TECNOLÓGICAS DA
UNIVERSIDADE DO ESTADO DE SANTA CATARINA.

Banca Examinadora:

Prof. Dr. André Bittencourt Leal
CCT/UDESC (Orientador/Presidente)

Prof. Dr. João Carlos dos Santos Basilio - UFRJ

Prof. Dr. Antonio Eduardo Carrilho da Cunha - IME

Prof. Dr. Ademir Nied
CCT/UDESC

Prof. Dr. Douglas Wildgrube Bertol
CCT/UDESC

Joinville / SC, 25 de março de 2019.

Eu dedico este trabalho a Deus que me deu condições físicas, mentais, emocionais e ainda colocou pessoas maravilhosas no meu caminho para poder realizar!

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus pela vida, pela saúde e por me dar condições de fazer este trabalho. A Ele toda honra e toda glória por essa conquista! Houve momentos em que a Paz que somente Ele pode nos dar é que me fez continuar nessa caminhada até chegar ao objetivo final. Agradeço ao meu esposo Edson que me ajudou de forma incondicional. Ele me ajudou em todos os aspectos, dando incentivos, ideias, lavando louças e até preparando deliciosos almoços e jantares. Agradeço aos nossos filhos que compreenderam minha ausência em muitos momentos sendo que cada um também teve durante essa trajetória seus desafios na vida acadêmica, emocional e espiritual. Por isso, fica aqui minha gratidão ao Rodrigo, Guilherme e Daniel pela compreensão. Não poderia deixar de agradecer de forma especial ao Daniel que me ajudou a fazer um video para uma conferência internacional. Agradeço aos meus pais que me mostraram desde criança a importância da dedicação ao trabalho e ao estudo fazendo tudo que estava ao alcance deles para nos encaminhar para o caminho do bem. Agradeço de coração à minha sogra-mãe, que sempre cuidou de mim como uma filha e ainda cuida de meus filhos em São Paulo, onde nossos filhos foram estudar. Agradeço muito ao meu orientador, amigo e professor Dr. André, que se mostrou ser um verdadeiro orientador. Ele sempre me auxiliou nesta área tão complexa e nova para mim, com muita paciência e dedicação em qualquer momento que eu solicitava seu precioso auxílio. Agradeço aos professores da pós-graduação que me ensinaram coisas inovadoras e importantes dentro de cada disciplina cursada. Aos amigos do Departamento da Engenharia Elétrica que sempre me apoiaram e deram palavras de encorajamento e motivação. Aos amigos de luta nesse doutorado em especial o prof. Fabrício (desde o início sempre me motivando e ajudando!), prof. Renan, prof. Benjamin, prof. Ricardo, prof. Douglas. Passamos alguns momentos tensos, porém criamos mais vínculos de amizade e companheirismo. Agradeço aos membros da banca examinadora, prof. Dr. João Carlos dos Santos Basilio, prof. Dr. Antonio Eduardo Carrilho da Cunha, prof. Dr. Antonio Heronaldo de Sousa, prof. Dr. Douglas Wildgrube Bertol e prof. Dr. Ademir Nied pela dedicação e contribuições enriquecedoras. Agradeço também a dedicação e atenção do secretário Marshel da Pós-Graduação da Engenharia Elétrica, e da Pós Graduação CCT, Francine e Valdinei. Agradeço a FUMDES - UNIEDU-SC pela bolsa de doutorado concedida durante 3 anos.

"Se procurar a sabedoria como se procura a prata e buscá-la como quem busca um tesouro escondido, então você entenderá o que é temer ao Senhor e achará o conhecimento de Deus. Pois, o Senhor é quem dá sabedoria; de sua boca procedem o conhecimento e o discernimento." Provérbios 2:4-6

RESUMO

Este trabalho considera o problema da controlabilidade segura de sistemas a eventos discretos. Basicamente, esse problema consiste em evitar a ocorrência de eventos ou sequências de eventos que podem levar o sistema a um comportamento ilegal ou proibido após a ocorrência de uma falha. Originalmente, o conceito de controlabilidade segura foi associado à diagnose de falhas. Segundo o mesmo, a falha precisa ser diagnosticada antes da ocorrência de qualquer evento ilegal e, após a diagnose, deve haver sempre um evento controlável que possa impedir qualquer evento ilegal. Nesta tese, introduz-se o conceito de controlabilidade segura pela prognose, segundo o qual as ações de controle podem ser tomadas até mesmo antes da ocorrência da falha, a partir de sua prognose. Esses dois conceitos foram concebidos no âmbito de linguagens, de forma que uma linguagem precisa ser, toda ela, controlável segura pela diagnose ou controlável segura pela prognose para que se tenha a controlabilidade segura da linguagem. Buscando ampliar a noção de controlabilidade segura, propõe-se uma nova abordagem baseada em cadeias ao invés de linguagens. Nessa abordagem, as propriedades são analisadas para cada cadeia que contém o evento de falha e não sobre o conjunto de cadeias. São introduzidos então os conceitos de cadeia diagnosticável, cadeia diagnosticável segura, cadeia prognosticável, cadeia controlável segura pela diagnose e cadeia controlável segura pela prognose. São apresentadas também condições necessárias e suficientes para garantir tais propriedades. A partir desses conceitos, define-se a controlabilidade segura de uma linguagem pela diagnose ou prognose, denominada DP-Controlabilidade Segura. Nesse caso, uma linguagem será DP-Controlável Segura se cada uma das cadeias que contém a falha for controlável segura pela diagnose ou for controlável segura pela prognose. Dessa forma, uma linguagem que não é, toda ela, controlável segura por um mesmo mecanismo de detecção de falha (diagnose ou prognose), pode ser DP-Controlável Segura. Por fim, são apresentadas condições necessárias e suficientes para que uma linguagem seja DP-Controlável Segura.

Palavras-chave: Diagnose de falhas, Prognose de falhas, Controlabilidade segura, Controle tolerante a falhas, Sistemas a Eventos Discretos.

ABSTRACT

This work considers the problem of safe controllability of Discrete Event Systems. Basically, this problem consists of avoiding the occurrence of events or sequences of events that can lead the system to illegal or prohibited behavior after a fault has occurred. Originally, the concept of safe controllability was associated with fault diagnosis. According to it, the fault must be diagnosed before any illegal event occurs and, after diagnosis, there must always be a controllable event that can prevent any illegal event. In this thesis, the concept of safe controllability by prognosis is introduced, according to which control actions can be taken even before the occurrence of the fault, based on its prognosis. These two concepts have been conceived in the context of languages, so that to ensure safe controllability of a given language, it must be (completely) safe controllable by diagnosis, or (fully) safe controllable by prognosis. Seeking to expand the notion of safe controllability, a new approach based on strings rather than languages is proposed. In this approach, properties are parsed over each string that contains the fault event, rather than over the set of strings that contain the fault. Then, the concepts of diagnosable string, safe diagnosable string, prognosable string, safe controllable string by diagnosis, and safe controllable string by prognosis are introduced. Necessary and sufficient conditions to guarantee such properties are also presented. From these concepts, the safe controllability of a language by diagnosis or prognosis is defined, which is called DP-Safe Controllability. In this case, a language is DP-controllable if each of the strings containing the fault is safe controllable by diagnosis or is safe controllable by prognosis. Thus, a language that is not completely controllable by the same fault detection mechanism (diagnosis or prognosis) can be DP-controllable. Finally, necessary and sufficient conditions for a language to be DP-controllable are presented.

Keywords: Fault diagnosis, Fault prognosis, Safe controllability, Fault tolerant control, Discrete Event Sytems.

Lista de Figuras

LISTA DE TABELAS

LISTA DE DEFINIÇÕES

LISTA DE PROPOSIÇÕES

LISTA DE TEOREMAS

LISTA DE ABREVIATURAS E SIGLAS

ADP	*(Active Diagnosis Problem)* Problema da Diagnose Ativa
CTF	Controle Tolerante a Falhas
CTFA	Controle Tolerante a Falhas Ativo
CTFP	Controle Tolerante a Falhas Passivo
NSE	*(Negative State Estimate)* Estimativa de Estado Negativo
PSE	*(Positive State Estimate)* Estimativa de Estado Positivo
SCTF	Sistema Controle Tolerante a Falhas
SDVC	Sistemas Dinâmicos de Variáveis Contínuas
SED	Sistema a Eventos Discretos
SEDF	Sistema a Eventos Discretos *Fuzzy*
TCS	Teoria de Controle Supervisório

LISTA DE SÍMBOLOS

Σ	Conjunto de eventos
Σ_o	Conjunto de eventos observáveis, $\Sigma_o \subset \Sigma$
Σ_{uo}	Conjunto de eventos não-observáveis, $\Sigma_{uo} \subset \Sigma$
Σ_c	Conjunto de eventos controláveis, $\Sigma_c \subset \Sigma$
Σ_{uc}	Conjunto de eventos não-controláveis, $\Sigma_{uc} \subset \Sigma$
Σ^{n+f}	Conjunto de eventos da planta incluindo evento de falha, $\Sigma^{n+f} = \Sigma + \Sigma_f$
Σ^*	Fecho de *Kleene*: Conjunto de todas as possíveis cadeias de comprimento finito sobre Σ, inclusive o ε
ε	Cadeia vazia
Σ^+	$\Sigma^* - \{\varepsilon\}$
$\overline{s}$	Fecho do prefixo de uma cadeia s
$t \leq s$	t é prefixo de s
$t < s$	t é prefixo estrito de s, sendo $t \neq s$
st	Concatenação das cadeias s e t
$\|\|t\|\|$	Comprimento de uma cadeia t
f	Evento de falha, $f \in \Sigma_{uo} \cup \Sigma_{uc}$
$\Psi_L(f)$	Conjunto de todas as cadeias de L que terminam com o evento f, $\Psi_L(f) = \{rf \in L : r \in \Sigma^*,\ f \in \Sigma_{uo} \cup \Sigma_{uc}\}$
L	Linguagem gerada por um autômato, $L \subseteq \Sigma^*$, linguagem sobre Σ
$\overline{L}$	Fecho de prefixo de L: Conjunto formado por todos os prefixos das cadeias de L
$L = \overline{L}$	Linguagem prefixo-fechada
L/s	Continuação da linguagem L após sequência s, $L/s = \{t \in \Sigma^* : st \in L\}$
G	Autômato finito determinístico, $G = (Q, \Sigma, \delta, q_0)$
Q	Conjunto de estados de um autômato G
Q^B	Conjunto de maus estados
Q_d	Conjunto de estados de um diagnosticador
Q_d^N	Conjunto de estados normais de um diagnosticador
Q_d^U	Conjunto de estados incertos *(uncertain)* de um diagnosticador
Q_d^C	Conjunto de estados certos de um diagnosticador

Q_{sd}	Conjunto de estados de um diagnosticador seguro
Q_{sd}^N	Conjunto de estados normais de um diagnosticador seguro
Q_{sd}^U	Conjunto de estados incertos *(uncertain)* de um diagnosticador seguro
Q_{sd}^C	Conjunto de estados certos de um diagnosticador seguro
δ	Função de transição de G, $\delta : Q \times \Sigma \to Q$
$\hat{\delta}$	Função de transição extendida de G, $\hat{\delta} : Q \times \Sigma^* \to Q$
q_0	Estado inicial de um autômato
$Obs(G, \Sigma_o)$	Autômato observador de G em relação a Σ_o
Obs^c	Autômato observador com alcance não-observável
Obs^s	Autômato observador sem alcance não-observável
P_o	Projeção de elementos de Σ^* em Σ_o^*
P_o^{-1}	Projeção inversa de elementos de $\Sigma_o^* \to 2^{\Sigma^*}$
$G_1 \| G_2$	Composição paralela de G_1 e G_2
$G_1 \times G_2$	Composição produto de G_1 e G_2
G_d	Autômato diagnosticador
G_d^c	Autômato c-diagnosticador com alcance não-observável
G_d^s	Autômato s-diagnosticador sem alcance não-observável
G_{sd}	Autômato diagnosticador seguro
G_{sd}^c	Autômato c-diagnosticador seguro com alcance não-observável
G_{sd}^s	Autômato s-diagnosticador seguro sem alcance não-observável
A_l	Autômato rotulador de falhas
G^{nom}	Planta nominal sem falhas
S^{nom}	Supervisor nominal
H^{nom}	Autômato da especificação nominal
$\mathscr{K}^{nom}$	Linguagem da especificação nominal $\mathscr{K}^{nom} = L(H^{nom})$
G_{sup}^{nom}	Autômato de planta nominal sob a ação do supervisor $G_{sup}^{nom} = G^{nom} \| S^{nom}$
G^{n+f}	Autômato de planta nominal com falhas
G_{sup}^{n+f}	Autômato de planta nominal com falhas sob a ação do supervisor
$G_i^{deg} (i = 1, ...m)$	Autômato da i-ésima planta degradada pós-falha
S_i^{deg}	i-ésimo supervisor degradado pós-falha
Φ	Conjunto finito de cadeias proibidas pós-falha

ξ	Cadeia (Sequência de eventos) proibida ou ilegal, $\xi \in \Phi$
$\mathscr{K}_f$	Linguagem ilegal pós-falha que contém a cadeia proibida Φ
$\mathscr{K}_i^{deg}$	i-ésima especificação degradada pós-falha
H_i^{deg}	Autômato que representa a i-ésima especificação degradada $\mathscr{K}_i^{deg}$
$\mathscr{D}$	Condição de diagnosticabilidade de uma falha
$\mathscr{P}$	Condição de prognosticabilidade de uma falha
$Ac(G,q')$	Acessibilidade de um estado q' de G
$\mathscr{C}$	Todos os ciclos em $Ac(G,q)$ que são estados certos no diagnosticador
$\mathcal{FC}$	Conjunto dos primeiros estados certos de falha alcançados no diagnosticador
$FC(s)$	Função que mapeia uma cadeia $s \in \Psi_L(f)$ nos primeiros estados certos de falha no diagnosticador alcançados a partir do estado inicial
F_D	Conjunto de estados normais no diagnosticador que possuem um sucessor imediato que não seja normal
$\mathcal{FU}$	Conjunto dos primeiros estados incertos no diagnosticador alcançados a partir do estado inicial
$FU(s)$	Função que mapeia uma cadeia $s \in \Psi_L(f)$ no primeiro estado incerto no diagnosticador alcançado a partir do estado inicial
$\mathcal{FP}$	Conjunto dos primeiros estados normais ou incertos no diagnosticador que asseguram a prognose
$FP(s)$	Função que mapeia uma cadeia $s \in \Psi_L(f)$ no primeiro estado que assegura prognose no diagnosticador, o qual é alcançado a partir do estado inicial
$FB(s)$	Função que mapeia uma cadeia $s \in \Psi_L(f)$ no primeiro mau estado no diagnosticador, o qual é alcançado a partir do estado inicial

SUMÁRIO

1 INTRODUÇÃO

A vulnerabilidade a falhas dos sistemas automatizados e o aumento da exigência por confiabilidade e desempenho em todos os segmentos, seja em sistemas industriais, sistemas de comunicação, computacionais, equipamentos médicos, ou mesmo residenciais, têm levado a constantes estudos de novas metodologias e tecnologias cada vez mais sofisticadas e sistemáticas para obter um diagnóstico mais preciso e rápido das falhas nos sistemas. Esse é de fato um problema importante a resolver, pois as falhas em sensores, atuadores ou controladores podem resultar em aumento de custos operacionais, perdas de produções, linhas de produção paradas e até mesmo impactos ambientais desastrosos. Diante dessa realidade, o problema de diagnose de falhas tem despertado grande interesse em diversas áreas de pesquisa, a fim de estudar metodologias para identificar e caracterizar possíveis falhas em partes predeterminadas da planta. De uma forma geral, a diagnose de falhas envolve três aspectos: detecção, isolamento e identificação da falha (ZAYTOON; LAFORTUNE, 2013). Neste trabalho, agrupam-se essas três tarefas sob a terminologia genérica de diagnose de falhas. A verificação da diagnosticabilidade de uma linguagem poderá ser feita através da análise de condições necessárias e suficientes. Além, da diagnosticabilidade, na literatura é apresentada a diagnosticabilidade segura. A diagnosticabilidade segura requer que a diagnose da falha ocorra antes da execução de um conjunto de cadeias proibidas para poder evitar situações perigosas. Outro tema de pesquisa ligado à diagnose de falhas que também tem sido alvo de muitas pesquisas é a prognose de falhas. Nessa, ao invés de apontar a ocorrência de uma falha que já ocorreu, busca-se inferir sobre futuras ocorrências de uma falha e, com isso, tomar decisões importantes antecipadas à ocorrência de falhas, como parar o sistema ou ativar medidas preventivas ou corretivas para evitar situações indesejáveis, aumentando o desempenho e a confiabilidade do mesmo.

Após a diagnose segura ou prognose da falha, o próximo passo consiste em atuar no sistema de controle a fim de garantir que, em malha fechada, a planta possa continuar sua operação de maneira segura, evitando a sua interrupção total ou a ocorrência de acidentes ou avarias graves. Assim, para que isso seja possível, o sistema deve ser controlável seguro, o que significa que, em malha fechada, ele deve ser capaz de impedir a ocorrência de certos eventos considerados proibidos após a ocorrência de uma falha. Nesse contexto, é introduzido o conceito de controlabilidade segura. Originalmente, a controlabilidade segura pode ser alcançada através de uma linguagem controlável segura pela diagnose, ou seja, a falha precisa ser diagnosticada antes da ocorrência de um evento proibido, e

somente após a diagnose, deve haver um evento controlável que possibilite evitar esse evento ilegal. Depois disso, surgiu o conceito de controlabilidade segura pela prognose, ou seja, uma ação de controle que poderia ser tomada até mesmo antes da ocorrência da falha, a partir de sua prognose. Essas duas abordagens são conceituadas sobre linguagens, de tal forma que uma linguagem precisa ser toda ela controlável segura pela diagnose ou controlável segura pela prognose para se ter a controlabilidade segura da linguagem. E se essa característica não ocorrer, não se tem um sistema controlável. A grande lacuna que este trabalho vem resolver é utilizando uma abordagem baseada em cadeias, ao invés de linguagens, resolver o problema através da controlabilidade segura de uma linguagem pela diagnose ou prognose. Ou seja, uma linguagem não precisa ser totalmente controlável segura por um mesmo mecanismo de detecção de falha, seja diagnose ou prognose. Uma vez obtida uma linguagem controlável segura pode-se buscar um Controle Tolerante a Falha Ativo (CTFA). Ou seja, a partir da diagnose segura ou prognose da falha, o controlador que satisfaz as especificações de controle na operação nominal é reconfigurado para um controlador degradado, o qual é capaz de evitar a ocorrência de eventos proibidos. A seguir, define-se o escopo desta tese.

1.1 DEFINIÇÃO DO ESCOPO

Para dar conta dos problemas relacionados à diagnose de falhas, encontram-se na literatura desde métodos baseados em modelos matemáticos, os quais são de interesse particular desta tese, até métodos baseados em inteligência artificial e em sistemas especialistas, que por sua vez fogem ao escopo desta tese. Segundo Zaytoon e Lafortune (2013), os métodos de diagnóstico de falhas baseados em modelos podem ser classificados de acordo com a representação da falha em: diagnose baseada em modelos que contemplam o comportamento faltoso do sistema; e diagnose usando modelos livres de falhas (*fault-free models*). De acordo com Sampath, Lafortune e Teneketzis (1998), sob o ponto de vista conceitual, a maioria dos métodos existentes para a diagnose de falhas pode ser classificada em: i) métodos baseados em árvores de falhas; ii) métodos baseados em modelos analíticos e qualitativos; iii) sistemas especialistas; iv) métodos de raciocínio baseados em modelos; e v) métodos baseados em sistemas a eventos discretos (SEDs). Nesta tese, o interesse reside nos métodos baseados em SEDs, os quais capturam o comportamento lógico e sequencial de sistemas usando modelos de estados discretos dirigido a eventos (CASSANDRAS; LAFORTUNE, 2008). Ainda, segundo Zaytoon e Lafortune (2013), no âmbito de SEDs, diversos formalismos de modelagem têm sido usados para tratar de problemas de diagnose de falhas e suas aplicações para controle. Dentre essas, destacam-se:

(i) autômatos de estados finitos (SAMPATH et al., 1995) e suas extensões (autômatos temporizados e probabilísticos); (ii) redes de Petri (FANTI et al., 2014; CABRAL, 2014).

Nesta tese, consideram-se os problemas de diagnose e prognose de falhas em SEDs modelados por autômatos de estados finitos.

Segundo Basilio, Carvalho e Moreira (2010), há dois paradigmas que orientam a diagnose de falhas em SEDs: (a) as falhas a serem diagnosticadas são eventos não-observáveis, isto é, eventos cujas ocorrências não podem ser registradas por sensores; (b) a ocorrência de falhas altera o comportamento do sistema, porém não necessariamente leva o sistema a uma parada. Estes paradigmas são válidos também para o contexto da prognose de falhas e para a controlabilidade segura, os quais são objeto de interesse deste trabalho.

De acordo com Zaytoon e Lafortune (2013), em SEDs as falhas podem ser permanentes, graduais ou intermitentes. Neste trabalho consideram-se apenas as falhas permanentes. Destaca-se, entretanto, que mesmo sendo permanente, a falha pode não levar o sistema a uma parada completa, conforme citado anteriormente, podendo levar o sistema a apresentar o que se chama de comportamento degradado.

Neste momento, pode-se finalmente apresentar o escopo em que esta tese é desenvolvida: problema de controlabilidade segura em SEDs utilizando diagnose e prognose de falhas online. Para tanto, serão utilizados autômatos de estados finitos como formalismo de modelagem e as falhas serão consideradas como permanentes e serão incluídas de forma explícita na modelagem da planta. Esta tese utiliza a abordagem da controlabilidade segura pela diagnose ou prognose para verificar se a linguagem é controlável segura.

Antes de apresentar os objetivos deste trabalho, é importante destacar que as metodologias desenvolvidas para o controle de SEDs podem ser aplicadas não apenas para sistemas cuja evolução de estados depende estritamente da ocorrência de eventos discretos, mas também em muitos sistemas dinâmicos de variáveis contínuas, uma vez que, considerando um nível maior de abstração, também podem ser modelados como SEDs. Dessa forma, a presente proposta de tese pode ser aplicada à classe de sistemas cujo comportamento pode, em algum nível de abstração, ser representado como SEDs.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Apresentar novos métodos ou abordagens para o problema da controlabilidade segura de SEDs modelados por autômatos de estados finitos.

1.2.2 Objetivos Específicos

No intuito de alcançar o objetivo geral da tese, foram objetivos específicos:
a) Realizar revisão bibliográfica sobre o tema de diagnose de falhas em SEDs;
b) Desenvolver estudo sobre prognose de falhas em SEDs;
c) Analisar o estado da arte sobre o controlabilidade segura em SEDs;
d) Identificar as lacunas existentes na literatura sobre controlabilidade segura utilizando prognose de falhas;
e) Combinar os conceitos de diagnose e prognose de falhas para fins de controlabilidade segura.

1.3 TRABALHOS RELACIONADOS

Este trabalho abrange três áreas distintas no contexto de SEDs. A primeira temática é a diagnose de falhas. O trabalho desenvolvido por Lin (1994) trouxe pela primeira vez a temática da diagnose de falhas no contexto de SEDs. Nesse, foram introduzidos os conceitos sobre capacidade de diagnosticar uma falha em sistemas modelados por SEDs. Posteriormente, Sampath et al. (1995) apresentaram condições necessárias e suficientes para a diagnose de falhas e propuseram a construção de autômato diagnosticador, o qual permite inferir sobre a capacidade de se diagnosticar as falhas presentes no sistema e realizar a diagnose de falhas em tempo real. Na sequência, diversos trabalhos surgiram na literatura. A importância da área de diagnose de falhas é refletida no número de publicações em conferências internacionais e jornais conforme citado em Zaytoon e Lafortune (2013). Dentre os quais, os trabalhos mais relacionados com a nossa pesquisa voltada para diagnose de falhas utilizando autômatos com abordagem estrutural centralizada, ou seja, usando um modelo global (monolítico) do sistema a ser diagnosticado são: (BASILIO; CARVALHO; MOREIRA, 2010) e (BASILIO et al., 2012). Em relação à diagnosticabilidade segura tem-se o trabalho de Paoli e Lafortune (2005) que trata da detecção da falha antes da execução de um comportamento inseguro.

Outra temática de especial interesse deste trabalho consiste na prognose de falhas em SEDs. Durante a revisão bibliográfica do tema, observou-se que há duas terminologias na literatura, predição (preditibilidade) e prognose (prognosticabilidade). Não foi detectado uma tendência do uso de um termo em restrição do outro ao longo das publicações. No entanto, no desenvolvimento deste trabalho, será adotado o termo prognose (prognosticabilidade). O trabalho sobre prognose de eventos em SEDs modelados por linguagens regulares desenvolvido por Genc e Lafortune (2006) foi o que inspirou este trabalho na área

de prognose de falhas. Depois, os autores introduziram condições necessárias e suficientes para que as ocorrências de um evento sejam prognosticáveis em um sistema modelado por linguagens regulares e a prognosticabilidade se mostra como uma condição mais forte do que a diagnosticabilidade de uma linguagem.

Por fim, a terceira temática envolvida neste trabalho consiste na controlabilidade segura em SEDs, a qual, segundo Paoli, Sartini e Lafortune (2011), recebeu menos atenção por parte da comunidade científica. Um dos primeiros trabalhos relacionando diagnose de falhas e controle no âmbito de SEDs foi desenvolvido por Sampath, Lafortune e Teneketzis (1998). A obtenção de um controlador através da controlabilidade segura pela diagnose foi tratada por Paoli, Sartini e Lafortune (2011). Nesse, os autores introduzem o conceito de diagnosticabilidade segura, o qual está relacionado com a capacidade do sistema diagnosticar a ocorrência da falha e verificar a controlabilidade antes da execução de alguma sequência proibida de eventos. Watanabe et al. (2017a) apresentam uma abordagem de controlabilidade segura utilizando prognose *online*. Nesse trabalho, os autores apresentam condições necessárias e suficientes para que uma linguagem possa ser controlável segura pela prognose.

Assim, a partir da revisão bibliográfica desenvolvida, e apresentada aqui de forma resumida, não se identificou na literatura nenhum trabalho envolvendo o uso de diagnose e prognose de falhas para fins de controlabilidade segura, o que indica o ineditismo desta tese.

1.4 RESUMO DAS CONTRIBUIÇÕES

Em termos gerais, o trabalho desenvolvido estende a abordagem de Controlabilidade Segura pela Diagnose e Prognose a SEDs. De forma mais pontual, as contribuições deste trabalho podem ser resumidas conforme Figura 1.1.

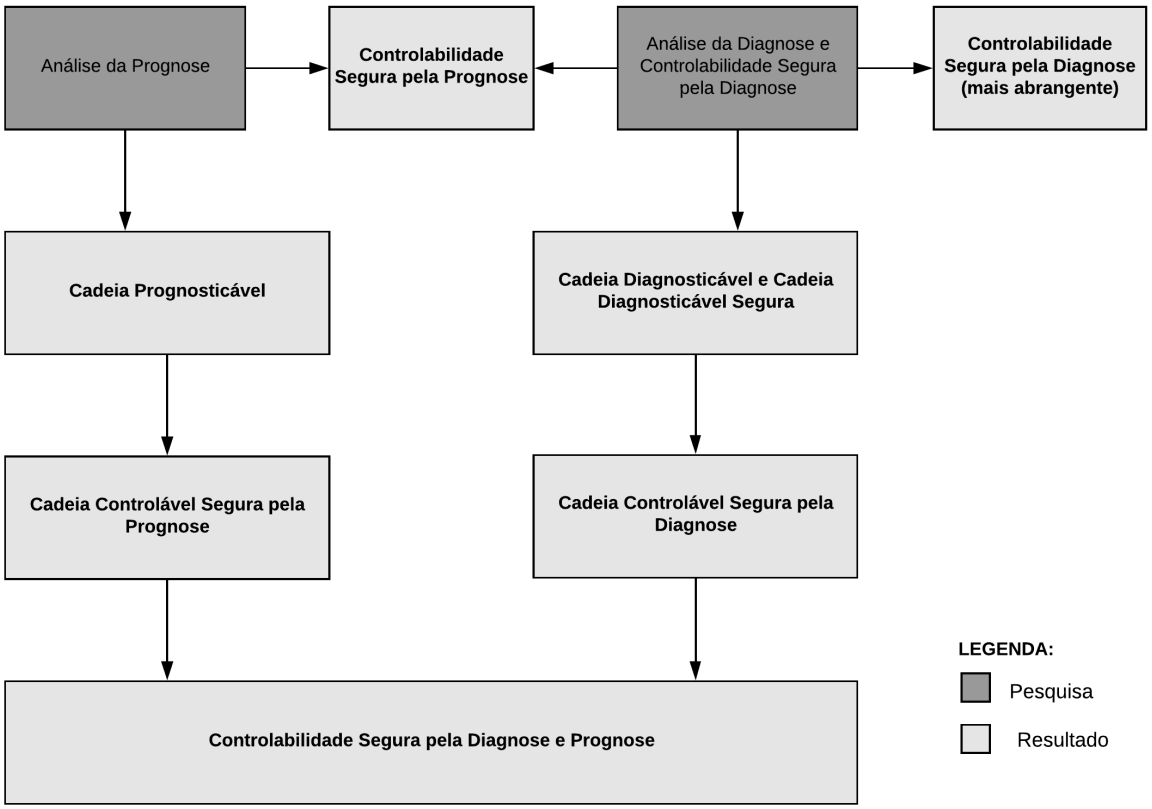
A partir da pesquisa na área da diagnose de falhas e controlabilidade segura de uma linguagem pela diagnose foram apresentadas as seguintes contribuições:
1) Controlabilidade Segura pela Diagnose:

- Novas condições para diagnosticabilidade segura;
- Proposta de melhoria na definição de controlabilidade segura apresentada por Paoli, Sartini e Lafortune (2011);

2) Cadeia Diagnosticável:

- Introdução das definições de cadeia diagnosticável e cadeia diagnosticável segura;

Figura 1.1 – Principais contribuições



Fonte: (Autor.)

– Estabelecimento de condições necessárias e suficientes para cadeia diagnosticável segura;

A partir da introdução de cadeia diagnosticável segura foi definida a controlabilidade segura de uma cadeia pela diagnose:

3) Cadeia Controlável Segura pela Diagnose:

– Introdução da definição de controlabilidade segura de uma cadeia pela diagnose;

– Estabelecimento de condições necessárias e suficientes para controlabilidade segura pela diagnose;

Ao analisar a controlabilidade segura de uma linguagem pela diagnose, foi pesquisada a prognose de falhas a fim de verificar a possibilidade de se obter a controlabilidade segura de uma linguagem pela prognose.

4) Controlabilidade Segura pela Prognose:

– Condições para Prognosticabilidade;

– Introdução da definição de controlabilidade segura pela prognose;

- Algoritmo do conjunto denominado FPpara controlabilidade segura pela prognose;
- Estabelecimento de condições necessárias e suficientes para a controlabilidade segura pela prognose;

A partir da pesquisa da prognose de falhas foram introduzidas a cadeia prognosticável e a cadeia controlável segura pela prognose:
5) Cadeia Prognosticável:

- Introdução da definição de cadeia prognosticável;
- Estabelecimento de condições necessárias e suficientes para cadeia prognosticável;

6) Cadeia Controlável Segura pela Prognose:

- Introdução da definição de controlabilidade segura de uma cadeia pela prognose;
- Estabelecimento de condições necessárias e suficientes para controlabilidade segura pela prognose;

Finalmente, a partir da controlabilidade segura de uma cadeia pela diagnose e pela prognose, foi introduzida uma nova abordagem baseada em cadeias de controlabilidade segura de uma linguagem pela diagnose ou prognose.
7) Controlabilidade Segura pela Diagnose ou Prognose:

- Generalização da definição de controlabilidade segura, a qual engloba os conceitos de cadeia controlável segura pela diagnose e cadeia controlável segura pela prognose;
- Estabelecimento de condições necessárias e suficientes para controlabilidade segura pela diagnose ou prognose.

1.5 ORGANIZAÇÃO DOS CAPÍTULOS

Este documento está estruturado da seguinte forma. No Capítulo 2 é feita uma breve revisão dos conceitos básicos da teoria de linguagens e autômatos em SEDs. No Capítulo 3 é feita uma revisão bibliográfica sobre os principais conceitos e fundamentos relacionados à diagnose de falhas em SEDs modelados por autômatos, dentre os quais se destacam os conceitos de diagnose de falhas e diagnose segura de falhas. São apresentados os diagnosticadores usados para a análise da diagnosticabilidade e da diagnosticabilidade segura de uma dada linguagem, bem como condições necessárias e suficientes para se garantir tais propriedades. São introduzidas novas definições de cadeia diagnosticável e

cadeia diagnosticável segura. Por fim, são apresentados os diagnosticadores usados para a análise da diagnosticabilidade e da diagnosticabilidade segura de uma dada cadeia, bem como condições necessárias e suficientes para se garantir tais propriedades. No Capítulo 4 é feita uma revisão bibliográfica sobre os principais conceitos relacionados à prognose de falhas em SEDs, com destaque para o conceito de prognosticabilidade. Discute-se sobre as formas de se verificar a prognosticabilidade de uma linguagem e apresenta-se a construção do diagnosticador usado para este fim. São apresentadas condições necessárias e suficientes para que a ocorrência de um determinado evento seja prognosticável em uma linguagem. É introduzida uma definição de cadeia prognosticável. São apresentados os diagnosticadores usados para a análise da prognosticabilidade de uma dada cadeia, bem como condições necessárias e suficientes para se garantir tal propriedade. O Capítulo 5 apresenta uma melhoria na definição de controlabilidade segura (PAOLI; SARTINI; LAFORTUNE, 2011) a qual foi denominada de controlabilidade segura pela diagnose e são apresentadas novas condições para a controlabilidade segura no contexto da diagnose. Introduz-se a definição de controlabilidade segura pela prognose e são estabelecidas condições sob as quais uma linguagem é controlável segura pela prognose. Apresenta-se a definição de cadeia controlável segura pela diagnose e condições necessárias e suficientes para a que a ocorrência de falha numa cadeia seja controlável segura pela diagnose. É introduzida a definição de cadeia controlável segura pela prognose e condições necessárias e suficientes para a que a ocorrência de falha numa cadeia seja controlável segura pela prognose. Além disso, é apresentada uma generalização da definição de controlabilidade segura, a qual engloba os conceitos de controlabilidade segura pela diagnose e pela prognose através das cadeias e são estabelecidas condições necessárias e suficientes para a controlabilidade segura definida anteriormente. Por fim apresenta-se a generalização dos resultados anteriores, propondo um exemplo da aplicação do que denominamos de *DP-Controller*. Esse controlador serve para chavear entre um supervisor nominal e um banco de supervisores degradados pós-diagnose ou pós-prognose de falha, os quais são projetados com o objetivo de evitar comportamentos proibidos depois da ocorrência de uma falha.

2 CONCEITOS BÁSICOS DE SISTEMAS A EVENTOS DISCRETOS

Sistemas a eventos discretos (SEDs) são sistemas compostos por estados discretos cuja transição de estados é realizada por eventos em instantes em geral em intervalos irregulares. Estados discretos consistem em estados que podem assumir valores simbólicos (por exemplo, aceso, apagado, ligado, desligado, etc.) e valores discretos como valores numéricos pertencentes aos conjuntos $\mathbb{N}$, $\mathbb{Z}$ ou subconjuntos enumeráveis do conjunto $\mathbb{R}$. Os eventos discretos podem ser ações como ligar uma máquina ou resultado de alguma ação como a chegada de uma peça numa esteira. Já em sistemas dinâmicos de variáveis contínuas (SDVC) as trajetórias dos estados são descritas em função do tempo (por exemplo, bobina de reaquecimento em sistemas de aquecimento, resposta de sensores, etc.) e, portanto, são modelados por equações diferenciais. Na literatura são encontrados vários formalismos para modelamento dos SEDs. Os mais conhecidos são autômatos e linguagens, redes Petri, teoria das filas, controle supervisório, álgebra Max-Plus, cadeias de Markov, simulação de eventos discretos, análise de perturbação e outros. A teoria de linguagens e autômatos foram adotados neste trabalho como formalismo de modelagem, pois dispõem de recursos necessários para o desenvolvimento deste projeto.

O objetivo deste capítulo é apresentar uma revisão bibliográfica dos assuntos importantes relacionados a SEDs a serem usados neste trabalho e está organizado da seguinte forma: Na seção 2.1 é apresentada a teoria de linguagens formais e na seção 2.2 é feita uma breve revisão sobre os autômatos. Na seção 2.3 são descritas as considerações finais.

2.1 LINGUAGENS

Na linguagem natural existe um conjunto de palavras que são coerentes ou corretas no contexto do idioma. Tais palavras são compostas unicamente por letras (símbolos) do seu alfabeto. Mas, também é possível formar composições de novas palavras, embora elas possam não ser reconhecidas pelo seu vocabulário formal. No contexto de SEDs, as linguagens são as entidades que definem esse vocabulário, ou seja, uma linguagem é um conjunto de sequências (também chamadas de cadeias) de comprimento finito formadas pelos eventos. O conjunto finito e não vazio desses símbolos é chamado de alfabeto, o qual é representado por Σ. Uma sequência finita de eventos que pertencem ao alfabeto Σ é denominada cadeia. Por exemplo, considerando que $\Sigma = \{a, b, c\}$ é um conjunto de eventos e que a linguagem L contém todas as possíveis cadeias de tamanho 2 que iniciam com evento b, tem-se que a linguagem $L = \{ba, bb, bc\}$. O conjunto de todas as cadeias

finitas formados por eventos Σ, incluindo a cadeia ε é representado por Σ^*. ε é uma cadeia especial chamada cadeia vazia, equivalente a uma sequência sem nenhum evento, ou seja, uma cadeia $s = \varepsilon s = s\varepsilon$. Σ^* é também chamado de fecho de Kleene de Σ. Por exemplo, se $\Sigma = \{a,b,c\}$, então: $\Sigma^* = \{\varepsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, cc, aaa, ...\}$. Outra forma de descrever linguagens regulares é utilizando expressões compactas chamadas de expressões regulares (CASSANDRAS; LAFORTUNE, 2008). Por exemplo, a expressão regular $(a+b)c^*$ denota a linguagem $L = \{a, b, ac, bc, acc, bcc, accc, bccc, ...\}$ a qual consiste de todas as cadeias que iniciam com evento a ou b, seguidas ou não pelo evento c ou sua repetição. Pode-se observar que embora L tenha uma infinidade de elementos, a expressão regular correspondente permite representar L de forma compacta. A construção de cadeias e, por conseguinte, de linguagens envolve uma operação chamada concatenação de eventos. Por exemplo a cadeia $s = ba \in L$ é obtida pela concatenação dos eventos a e b. O comprimento de uma cadeia $s \in \Sigma^*$ é representada por $\|s\|$ que indica a quantidade de eventos inclusive repetições na cadeia s, por exemplo, $\|abbde\| = 5$. Por convenção, o comprimento da cadeia vazia ε é zero. Dadas duas cadeias t, $s \in \Sigma^*$, t é um prefixo de s ($t \leq s$) se existe $u \in \Sigma^*$ tal que a concatenação de $tu = s$. Denota-se $t < s$, um caso especial de $t \leq s$, em que $s \neq t$. Nesse caso, diz-se que t é um prefixo estrito de s. Além disso, $\overline{s}$ expressa o conjunto de prefixos de s. Dado um evento $\sigma \in \Sigma$ e uma cadeia $s \in \Sigma^*$, é usada a notação $\sigma \in s$ para expressar que σ aparece pelo menos uma vez em s. Uma linguagem é viva se todas as cadeias em L podem ser estendidas para outra cadeia em L.

Como as linguagens são conjuntos, podem-se considerar válidas as operações convencionais de conjuntos. A seguir são apresentadas algumas operações para linguagens: concatenação, fecho de Kleene, fecho do prefixo, pós-linguagem, projeção e projeção inversa (CASSANDRAS; LAFORTUNE, 2008).

- Concatenação: A concatenação entre duas linguagens L_1 e $L_2 \subseteq \Sigma^*$ é dada da seguinte forma: $L_1L_2 := \{s \in \Sigma^* : (s = s_1s_2) \text{ e } (s_1 \in L_1) \text{ e } (s_2 \in L_2)\}$.

- Fecho de Kleene: Seja $L \subseteq \Sigma^*$, então o fecho de Kleene de L é definido como: $L^* := \{\varepsilon\} \cup L \cup LL \cup LLL \cup ...$

- Fecho do prefixo: O fecho de prefixo de uma linguagem L (denotado por $\overline{L}$) é o conjunto formado por todos os prefixos das sequências de L, ou seja:
  $\overline{L} := \{s \in \Sigma^* : (\exists t \in \Sigma^*)[st \in L]\}$. Uma linguagem L tal que $L = \overline{L}$ é dita ser prefixo-fechada.

- Pós-linguagem: A continuação da linguagem L após cadeia s, é expressa por $L/s := \{t \in \Sigma^* : st \in L\}$.

Para apresentar o conceito de projeção e projeção inversa considere que o conjunto de eventos de uma linguagem L é particionado como $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$, sendo que Σ_o representa o conjunto de eventos observáveis e Σ_{uo} representa o conjunto de eventos não-observáveis. O símbolo $\dot{\cup}$ significa uma união disjunta, ou seja, a interseção entre esses conjuntos é vazia. Eventos não-observáveis são aqueles cuja ocorrência não é registrada ou percebida por sensores, como por exemplo a ocorrência de eventos de falhas.

- Projeção: Informalmente, pode-se dizer que na projeção P_o de uma cadeia definida em Σ^* sobre outro conjunto de cadeias definidas em Σ_o^* são apagados os eventos da cadeia original que não estejam no alfabeto projetado; nesse caso seriam os eventos não-observáveis (elementos de Σ_{uo}) em s. Formalmente, a projeção $P_o : \Sigma^* \to \Sigma_o^*$, é definida como:

  $P_o(\varepsilon) := \varepsilon$;

  $$P_o(\sigma) := \begin{cases} \sigma, & \text{se} \quad \sigma \in \Sigma_o, \\ \varepsilon, & \text{se} \quad \sigma \in \Sigma \backslash \Sigma_o. \end{cases}$$

  $P_o(s\sigma) := P_o(s)P_o(\sigma)$, $s \in \Sigma^*$ e $\sigma \in \Sigma$.

- Projeção inversa: A projeção inversa $P_o^{-1} : \Sigma_o^* \to 2^{\Sigma^*}$ é definida da seguinte maneira:

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

  As projeções de cadeias podem ser estendidas para linguagens de forma natural aplicando as propriedades de projeção a todas as cadeias da linguagem. Assim, sendo $L \subseteq \Sigma^*$, então, a projeção de L é definida por:

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

  A projeção inversa de uma linguagem $L_o \subseteq \Sigma_o^*$ é dado por:

  $P_o^{-1}(L_o) := \{s \in \Sigma^* : (\exists t \in L_o)[P_o(s) = t]\}$.

  A seguir, um exemplo para ilustrar operações de linguagens:

Exemplo 1 *Considere o conjunto de eventos* $\Sigma = \{a, b, c\}$ *e a linguagem* $L_1 = \{\varepsilon, b\}$ *e linguagem* $L_2 = \{a, ab, acc\}$. $\overline{L_1} = \{\varepsilon, b\}$, *como* $L_1 = \overline{L_1}$, *a linguagem* L_1 *é prefixo-fechada. Porém,* $\overline{L_2} = \{\varepsilon, a, ab, ac, acc\}$, *então* $L_2 \neq \overline{L_2}$. *Portanto, a linguagem* L_2 *não é prefixo-fechada. Além disso, pode-se verificar que:*

$L_1^* = \{\varepsilon, b, bb, bbb...\}$

$L_1L_2 = \{a, ab, acc, ba, bab, bacc\}$

$L_2L_1 = \{a, ab, acc, abb, accb\}$

$L_2/a = \{\varepsilon, b, cc\}$

Considerando que $\Sigma_{uo} = \{c\}$*, tem-se que:*
$P_o(abc) = ab$
$P_o^{-1}(ab) = \{c^*\}\{a\}\{c^*\}\{b\}\{c^*\}$
$P(L_2) = \{a, ab\}$
$P_o^{-1}(L_1) = \{\{c^*\}, \{c^*\}\{b\}\{c^*\}\}$

2.2 AUTÔMATOS

Os autômatos ou máquinas de estados finitos ou geradores são formas de modelar SEDs representando uma linguagem seguindo regras bem definidas. Os autônomos podem ser determinísticos ou não-determinísticos:

2.2.1 Autômatos Determinísticos

A seguir é apresentada definição de um autômato determinístico como uma quíntupla:

$$G = (Q, \Sigma, \delta, \Gamma, q_0), \tag{2.1}$$

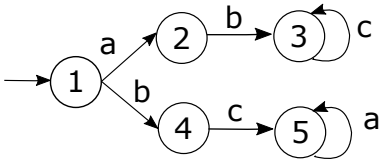
em que Q denota o conjunto de estados, Σ o conjunto finito de eventos, $\delta : Q \times \Sigma \to Q$ a função de transição de estados, Γ é a função de eventos ativos e q_0 estado inicial de G. A função de transição pode ser intuitivamente estendida do domínio $Q \times E$ para o domínio $Q \times E^*$, extensão essa que será denotada por $\hat{\delta}$: $Q \times \Sigma^* \to Q$.

O comportamento do sistema é descrito pela linguagem prefixo-fechada $L(G)$ gerada pelo autômato G. Para fins de simplificação, daqui em diante $L(G)$ será denotada como L. Os autômatos são representados graficamente por meio de diagramas de transição de estados, sendo que os estados são representados por nós circulares e as transições de estados por arcos rotulados com símbolos que representam os eventos. O estado inicial é identificado por uma seta apontando para o nó circular inicial sem conexão com outros estados. Quando uma transição não gera a mudança de estado, é chamada de auto-laço.

A seguir, é dado um exemplo que ilustra um autômato determinístico.

Exemplo 2 *Considere o Autômato* G_2 *mostrado na Figura 2.1. A linguagem gerada por* G_2 *é* $L_2 = \overline{abc^* + bca^*}$*, sendo que todos os eventos são observáveis, denotados como* $\Sigma_o = \{a, b, c\}$.

Figura 2.1 – Exemplo de um autômato determinístico. Autômato G_2.



Fonte: (Autor.)

Na Figura 2.1, a linguagem L_2, gerada pelo autômato G_2, apresenta o conjunto de estados $Q = \{1,2,3,4,5\}$, o conjunto de eventos $\Sigma = \{a,b,c\}$ e a função dos eventos ativos $\Gamma(1) = \{a\}$, $\Gamma(2) = \{b\}$, $\Gamma(3) = \{c\}$, $\Gamma(4) = \{c\}$ e $\Gamma(5) = \{a\}$. Observando as funções de transições de estados $\delta(1,a) = 2, \delta(1,b) = 4, \delta(2,b) = 3, \delta(3,c) = 3, \delta(4,c) = 5$ e $\delta(5,a) = 5$, percebe-se a certeza na evolução dinâmica do autômato, ilustrando, portanto, um autômato determinístico.

2.2.2 Autômatos Não-determinísticos

A seguir é apresentada formalmente a definição de um autômato não-determinístico:

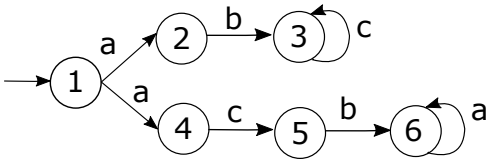
$$G = (Q, \Sigma, f_{nd}, \Gamma, q_0), \tag{2.2}$$

sendo que Q denota o conjunto de estados, Σ o conjunto finito de eventos, $f_{nd} : Q \times \Sigma \to 2^Q$, a função não-determinística sendo 2^Q o conjunto de todos os subconjuntos de Q, Γ é a função de eventos ativos e q_0 estado inicial de G.

A seguir, apresenta-se um exemplo para ilustrar um autômato não-determinístico.

Exemplo 3 *Considere o Autômato G_3 mostrado na Figura 2.2. A linguagem gerada por G_3 é $L_3 = \overline{abc^* + acba^*}$, sendo que $\Sigma = \Sigma_o = \{a,b,c\}$.*

Figura 2.2 – Exemplo de um autômato não-determinístico. Autômato G_3.



Fonte: (Autor.)

O autômato G_3 mostrado na Figura 2.2 apresenta o conjunto de estados $Q = \{1,2,3,4,5,6\}$, o conjunto de eventos $\Sigma = \{a,b,c\}$ e a função dos eventos ativos $\Gamma(1) = \{a\}$, $\Gamma(2) = \{b\}$,

$\Gamma(3) = \{c\}$, $\Gamma(4) = \{c\}$, $\Gamma(5) = \{b\}$ e $\Gamma(6) = \{a\}$. Observando a função $f_{nd}(1,a) = \{2,4\}$, verifica-se a incerteza na evolução dinâmica do sistema, tendo portanto, um exemplo de autômato não determinístico.

Neste trabalho, vamos modelar problemas na planta com falhas que ocorrem inesperadamente sem serem, portanto, observáveis. Portanto, serão tratados autômatos determinísticos com eventos não-observáveis.

2.2.3 Autômato determinístico com eventos não-observáveis

Autômato determinístico com observação parcial é todo autômato determinístico que possui algum evento não observável. esta subseção, consideraremos o caso de DES parcialmente observado, ou seja, quando alguns eventos não podem ter suas ocorrências vistas por um observador externo. Esta falta de observabilidade pode ser devido à ausência de um sensor para registrar a ocorrência do evento ou para o fato de que o evento ocorre em um local remoto, mas não é comunicada ao site sendo modelado. Neste caso, alguma forma de estimativa de estado torna-se necessário ao analisar o comportamento do sistema. Para este fim, o conjunto de eventos Σ é particionado no conjunto de eventos observáveis Σ e conjunto de eventos não-observáveis Σ_{uo}. O autômato correspondente será determinístico, e é referido como autômato determinístico com eventos não observáveis.

Para se obter um autômato determinístico com observação parcial, é necessário introduzir o conceito de alcance não-observável de um estado $q' \in Q$, denotado por $UR(q')$:

$$UR(q') = \{q \in Q : (\exists t \in \Sigma_{uo}^*)(\hat{\delta}(q',t) = q)\}. \tag{2.3}$$

Observe que $UR(q')$ retorna todos os estados alcançáveis a partir de q' através de transições rotuladas por eventos não-observáveis.

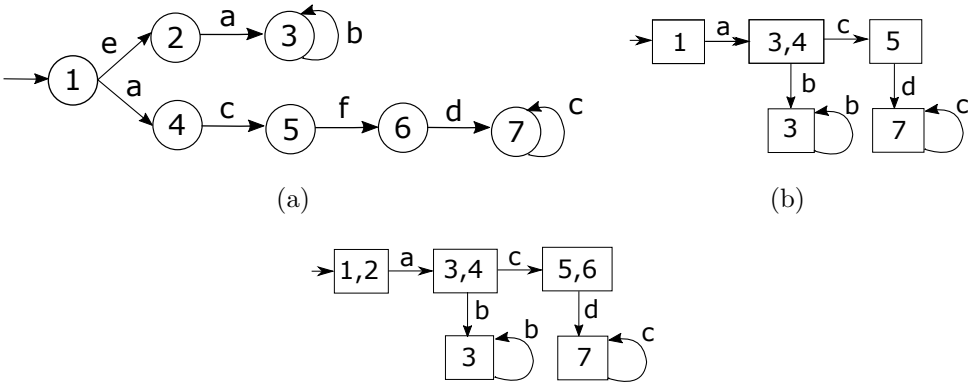
O observador para G, denotado por $Obs(G,\Sigma_o)$, é definido da seguinte forma: $Obs(G,\Sigma_o) = (Q_{obs},\Sigma_o,f_{obs},\Gamma_{obs},q_{0obs})$, sendo $Q_{obs} \in 2^Q$.

No cálculo do observador pode-se adotar a estratégia de incluir (BASILIO; LAFORTUNE, 2009), (BASILIO; CARVALHO; MOREIRA, 2010) e (CARVALHO, 2011) ou não incluir o alcance não-observável (SAMPATH et al., 1995).

A seguir, é dado um exemplo para ilustrar a construção de um autômato observador considerando a não inclusão e a inclusão do alcance não-observável.

Exemplo 4 *Seja o autômato G_4 mostrado na Figura 2.3(a), cuja linguagem gerada é dada por $L_4 = \overline{eab^* + acfdc^*}$, sendo $\Sigma_{uo} = \{e, f\}$, $\Sigma_o = \{a, b, c, d\}$.*

Figura 2.3 – Exemplo de construção de um autômato observador. (a) Autômato G_4; (b) Autômato s-observador Obs_4^s; (c) Autômato c-observador Obs_4^c.



Fonte: (Autor.)

A construção do observador sem incluir o alcance não-observável Obs_4^s, parte do estado inicial (1) de G_4, do qual existe uma transição de saída com um evento não observável e. Portanto, o estado (2), não faz parte do observador Obs_4^s, conforme Figura 2.3b. Observa-se que como o evento f que parte do estado (5) também é um evento não-observável, o estado (6) também não é considerado no Obs_4^s. Já na construção do observador incluindo o alcance não-observável Obs_4^c, conforme Figura 2.3c, os estados (2) e (6) são inclusos.

A partir da construção do observador, pode-se concluir que a linguagem gerada por $Obs(G)$ é a projeção da linguagem de G sobre o conjunto de eventos observáveis, isto é, $L(Obs(G)) = P_o[L(G)]$.

A seguir são apresentadas algumas operações com autômatos que serão utilizadas no decorrer deste trabalho.

2.2.4 Operações com Autômatos

- Parte acessível: Um $q \in Q$ de um determinado autômato G é acessível, se $\exists s \in \Sigma^*$, tal que $\delta(q_0, s) = q$. Por outro lado, q é um estado não-acessível. A operação da parte acessível remove todos os não-acessíveis estados do autômato G. A definição é: $Ac(G) := (Q_{ac}, \Sigma, \delta_{ac}, \Gamma_{ac}, q_0)$, sendo que $Q_{ac} = \{q \in Q : (\exists s \in \Sigma^*)(\delta(q_0, s) = q)\}$ e $\delta_{ac} = \delta | Q_{ac} \times \Sigma \to Q_{ac}$.

A parte acessível de G em relação à q' é denotada por $A_c(G,q') = (Q_{ac}, \Sigma, \delta_{ac}, q')$, sendo $Q_{ac} = \{q \in Q : (\exists s \in \Sigma^*)(\delta(q',s) = q)\}$, e $\delta_{ac} = \delta | Q_{ac} \times \Sigma \to Q_{ac}$. Em palavras, a parte acessível de um autômato é a operação que elimina todos os estados de G que não são alcançáveis a partir do estado q'.

Sejam $G_1 = (Q_1, \Sigma_1, \delta_1, \Gamma_1, q_{01})$ e $G_2 = (Q_2, \Sigma_2, \delta_2, \Gamma_2, q_{02})$ autômatos distintos e acessíveis. Duas operação de composição de autômatos são apresentadas a seguir.

- Composição paralela ou síncrona: A composição paralela entre dois autômatos G_1 e G_2 mapeia o comportamento síncrono entre os mesmos, ou seja, um evento σ, comum aos dois autômatos, somente poderá ser executado se ocorrer simultaneamente nos dois autômatos, enquanto os eventos particulares, isto é, $\sigma \in (\Sigma_1 \backslash \Sigma_2) \cup (\Sigma_2 \backslash \Sigma_1)$ poderão ser executados livremente sempre que forem possíveis. A composição paralela de G_1 e G_2 é o autômato $G_1 || G_2 := Ac(Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \delta_{1||2}, \Gamma_{1||2}, (q_{01}, q_{02}))$, em que

$$\delta((q_1,q_2),\sigma) = \begin{cases} (\delta_1(q_1,\sigma), \delta_2(q_2,\sigma)), & \text{se } \sigma \in \Gamma_1(q_1) \cap \Gamma_2(q_2) \\ (\delta_1(q_1,\sigma), q_2), & \text{se } \sigma \in \Gamma_1(q_1) \backslash \Sigma_2 \\ (q_1, \delta_1(q_2,\sigma)), & \text{se } \sigma \in \Gamma_2(q_2) \backslash \Sigma_1 \\ \text{indefinida}, & \text{caso contrário} \end{cases}$$

além disso,

$\Gamma_{1||2}(q_1,q_2) = [\Gamma_1(q_1) \cap \Gamma_2(q_2)] \cup [\Gamma_1(q_1) \backslash \Sigma_2] \cup [\Gamma_2(q_2) \backslash \Sigma_1]$

- Produto: Na operação produto um evento ocorre no autômato resultante do produto $G_1 \times G_2$ se e somente se ocorrer em ambos os autômatos. O produto de G_1 e G_2 é o autômato $G_1 \times G_2 := Ac(Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \delta_{1\times 2}, \Gamma_{1\times 2}, (q_{01}, q_{02}))$, em que

$$\delta((q_1,q_2),\sigma) = \begin{cases} (\delta_1(q_1,\sigma), \delta_2(q_2,\sigma)), & \text{se } \sigma \in \Gamma_1(q_1) \cap \Gamma_2(q_2) \\ \text{indefinida}, & \text{caso contrário} \end{cases}$$

além disso,

$\Gamma_{1\times 2}(q_1,q_2) = \Gamma_1(q_1) \cap \Gamma_2(q_2)$.

2.3 CONSIDERAÇÕES FINAIS

Neste capítulo foi feita uma breve revisão sobre SEDs e sobre a teoria de Autômatos e Linguagens para dar embasamento aos próximos capítulos. Vale lembrar que existem duas

maneiras para construção do observador, uma delas incluindo o alcance não observável e outra não. As implicações dessas duas abordagens são tratadas nos capítulos 3 e 4, nos quais são abordados os temas de Diagnose e Prognose de falhas, respectivamente.

3 DIAGNOSE DE FALHAS EM SEDS

Neste capítulo, além de apresentar definições e fundamentos já estabelecidos sobre diagnose de falhas em SEDs modelados por autômatos de estados finitos, são introduzidos, como contribuições desta tese, os conceitos de cadeia diagnosticável, cadeia diagnosticável segura, bem como são apresentadas condições necessárias e suficientes para garantir a diagnosticabilidade e a diagnosticabilidade segura de cadeias.

Este capítulo está organizado da seguinte forma: Na seção 3.1 é apresentada uma revisão bibliográfica sobre diagnose de falhas. Na seção 3.2 é formulado o problema da diagnose de falhas e na seção 3.3 é introduzido o conceito de cadeia diagnosticável. Na seção 3.4 é apresentado o diagnosticador para verificar diagnosticabilidade. Condições necessárias e suficientes para que a linguagem seja diagnosticável são apresentadas na seção 3.5. Na seção 3.6 são apresentadas condições necessárias e suficientes para que uma cadeia seja diagnosticável. Na seção 3.7 é apresentada a definição de diagnosticabilidade segura e, a partir dessa definição, é introduzido o conceito de diagnosticabilidade segura em cadeias na seção 3.8. Na seção 3.9 é apresentado o diagnosticador seguro e condições necessárias e suficientes para a diagnosticabilidade segura de uma linguagem são apresentadas na seção 3.10. Na seção 3.11 são introduzidas condições necessárias e suficientes para se obter diagnosticabilidade segura numa cadeia. Finalmente, na seção 3.12 são apresentadas as considerações finais.

3.1 REVISÃO BIBLIOGRÁFICA SOBRE DIAGNOSE DE FALHAS

Esta seção contempla uma revisão de artigos publicados na área de diagnose de falhas. Portanto, o leitor que tiver conhecimento dos trabalhos relacionados poderá se dirigir à seção seguinte, sem prejuízos para o entendimento do trabalho. Em Zaytoon e Lafortune (2013) é apresentada uma ampla revisão sobre a diagnose de falhas. Se o leitor quiser se aprofundar nessa revisão poderá consultar essa obra. Após as publicações dos trabalhos feitos por Lin (1994), Sampath et al. (1995) e Sampath et al. (1996), nos quais foram apresentados os conceitos fundamentais da diagnose de falhas no contexto de SEDs, surgiram muitos trabalhos nessa área. Conforme Basilio, Carvalho e Moreira (2010), as falhas a serem diagnosticadas são eventos não-observáveis; e a ocorrência de falhas não necessariamente leva o sistema a uma parada; por exemplo, em sistemas de manufatura, a ocorrência de uma falha não diagnosticada pode levar a uma degradação da eficácia global dos equipamentos (disponibilidade, eficiência e qualidade) sem ocasionar a

parada do sistema. Miyagi e Riascos (2006) apresentaram que em sistemas de manufatura ocorrem dois tipos de detecção de falhas em relação a equipamentos: 1) A detecção de falha através do monitoramento do parâmetro de um dispositivo específico, como a utilização de um sensor para monitorar o nível de algum líquido. Neste caso, não é necessário um diagnóstico de falhas. 2) As falhas não podem ser detectadas diretamente pelo monitoramento, necessitando de um tipo de diagnóstico de falhas. Considerando para o estudo deste trabalho o segundo tipo, um evento de falha poderá ser diagnosticado se a sua ocorrência puder ser detectada após a ocorrência de um número finito de eventos observáveis. Para tal, são construídos sistemas para a diagnose de falhas cujo objetivo é inferir e informar a ocorrência de falhas tendo como base somente os eventos que tenham sido observados (BASILIO; CARVALHO; MOREIRA, 2010).

Na sequência, diversos trabalhos abordaram a temática de diagnose de falhas em SEDs, sendo que os de maior relevância para o contexto desta tese são discutidos a seguir.

Zaytoon e Lafortune (2013) classificaram em três estruturas ou arquiteturas o cálculo da diagnose de falhas: centralizada, descentralizada e distribuída. Neste trabalho, utiliza-se a abordagem de diagnose centralizada.

Diagnose Centralizada: A estrutura de diagnose centralizada é baseada em um modelo global (monolítico) do sistema a ser diagnosticado. Essa estrutura apresenta vantagens quanto à precisão do diagnóstico e simplicidade conceitual. Porém, sua principal desvantagem é a sua complexidade computacional, pois é requerido um modelo de planta centralizado para gerar o diagnosticador centralizado.

Diagnose Descentralizada: A arquitetura de diagnose descentralizada consiste em diagnosticadores locais com capacidade de observação parcial sobre o sistema como um todo. Tais módulos locais têm comunicação com um coordenador, que é responsável pela diagnose das falhas que venham a ocorrer no sistema. O propósito da abordagem descentralizada é vencer o problema da alta complexidade computacional da abordagem centralizada. (DEBOUK; LAFORTUNE; TENEKETZIS, 2000), (LIU; WU, 2018), (VIANA; BASILIO, 2019).

Diagnose Distribuída: Nesta estrutura cada subsistema conhece somente sua própria parte do modelo global. O diagnosticador local é associado a cada subsistema para realizar a diagnose localmente (PENCOLÉ, 2004), (SU; WONHAM, 2004). A comunicação somente é realizado através de protocolos de comunicação. A abordagem distribuída consiste em realizar o diagnóstico descentralizado usando um conjunto de modelos locais, sem se referir a um modelo de planta centralizado. No entanto, um protocolo de comunicação deve ser definido para garantir a consistência entre diagnosticadores locais. Se os modelos

locais (subsistemas) não interagirem de forma hierárquica ou de árvore, o protocolo de comunicação exigirá tempo computacional e grande espaço para os estados.

Com a finalidade de evitar a complexidade exponencial do sistema centralizado, muitos métodos com estrutura modular foram propostos: (DEBOUK; MALIK; BRANDIN, 2002); (CONTANT; LAFORTUNE; TENEKETZIS, 2006); e (ZHOU; KUMAR; SREENIVAS, 2008). Nesses trabalhos, um diagnosticador local é construído para cada módulo do sistema global e o diagnóstico é realizado baseado apenas nas observações do módulo. Contant, Lafortune e Teneketzis (2006) introduziram o conceito de diagnosticabilidade modular em sistemas.

Diagnose Modular: Nesta estrutura, são construídos diagnosticadores locais a partir dos modelos dos subsistemas que compõem a planta e o diagnóstico é realizado apenas com base nas observações do módulo. Se o sistema for modularmente diagnosticável, então a diagnose de falha do sistema global pode ser feita utilizando-se somente os diagnosticadores locais, sem a necessidade de construir um diagnosticador global. Esta arquitetura apresenta um coordenador e o modelo da planta não é centralizado.

Muitos trabalhos importantes surgiram dentro do conceito da diagnosticabilidade robusta.

Diagnose robusta com perdas permanentes e intermitentes: O conceito de diagnosticabilidade robusta foi introduzido por Basilio e Lafortune (2009) no contexto de diagnosticabilidade descentralizada considerando que a comunicação entre um módulo e o coordenador não é confiável. Muitas abordagens relacionadas a diagnose robusta foram sendo publicadas como no contexto de diagnosticabilidade centralizada a perdas permanentes de sensores (LIMA et al., 2010), diagnosticador robusto (TAKAI, 2010), metodologias probabilísticas (ATHANASOPOULOU; LINGXI; HADJICOSTIS, 2010) e (THORSLEY; YOO; GARCIA, 2008), utilização de diagnosticadores e verificadores (BASILIO; CARVALHO; MOREIRA, 2010), perdas intermitentes de sensores (BASILIO; LAFORTUNE, 2009), estocásticos (THORSLEY; YOO; GARCIA, 2008), (YIN et al., 2019), entre outros.

No trabalho de Sampath, Lafortune e Teneketzis (1998) é tratado a diagnose ativa que engloba a diagnose de falhas com controle no âmbito de SEDS.

Diagnose passiva e ativa: O termo diagnose passiva é usado quando o papel do diagnosticador é simplesmente observar o comportamento do sistema e fazer inferências sobre possíveis falhas enquanto diagnose ativa é uma combinação de observação e ação de controle para alterar a propriedade de diagnosticabilidade de um sistema. Em Sampath, Lafortune e Teneketzis (1998) é apresentada uma abordagem integrada de controle e diagnose, ou seja, através de um apropriado controlador é projetado um sistema diagnosticável. É mostrado o desenvolvimento de um controlador, denominado *diagnostic*

controller, baseado na teoria de diagnose de falhas para SEDs e nos resultados existentes de controle supervisório sob observações parciais. Os autores Sampath, Lafortune e Teneketzis (1998) desenvolvem o conceito de diagnose ativa através do Problema da Diagnose Ativa (Active Diagnosis Problem - ADP).

Outros trabalhos relevantes: Um trabalho relacionado à detecção e prevenção de intrusos em sistemas de controle supervisório foi apresentado por Carvalho et al. (2016). Nesse trabalho é apresentado a partir de um modelo matemático estratégias para detectar ataques online e desabilitar todos os atuadores controláveis após obter certeza de ataque. Os autores criam uma variação da controlabilidade segura de Paoli, Sartini e Lafortune (2011) para representar a capacidade de prevenir o sistema de alcançar um evento inseguro após detectar o ataque. De acordo com Carvalho (2011), de forma resumida basta definir estados não seguros e exigir que haja um evento controlável após detectar o ataque. Em Zhao, Liu e Liu (2017) foi discutido sobre a diagnosticabilidade relativa em SEDs e um algoritmo baseado em opacidade é desenvolvido para testar a diagnosticabilidade relativa. Recentemente, outra condição necessária e suficiente para a codiagnosticabilidade em SEDs foi apresentada (VIANA; BASILIO, 2019). No trabalho de Lafortune (2019) é discutido diagnosticabilidade e opacidade no contexto de SEDs parcialmente observáveis.

3.2 DIAGNOSE E DIAGNOSTICABILIDADE DE FALHAS DE UMA LINGUAGEM

O presente trabalho está baseado nas definições e conceitos básicos de diagnose e diagnosticabilidade de falhas em SEDs propostos por Sampath et al. (1995), Sampath et al. (1996), Sampath, Lafortune e Teneketzis (1998) e Basilio, Carvalho e Moreira (2010). A diagnose de falhas está fortemente ligada ao problema de observabilidade dos estados, o qual consiste em construir um autômato determinístico, denominado diagnosticador G_d, que será explanado nesta seção.

De acordo com Carvalho (2011), as seguintes hipóteses normalmente são adotadas nos trabalhos envolvendo diagnose de falhas em SEDs:

H1) A linguagem L gerada por G é viva;

H2) Nenhum ciclo de estados do autômato G é composto somente por eventos não-observáveis; e

H3) Considerado apenas um tipo de falha.

Em relação a H1, deve haver uma transição definida para cada estado $q \in Q$, $\Gamma(q) \neq \emptyset$. Em relação a H2, é para evitar que a ocorrência de uma falha possa vir a não ser detectada caso o sistema fique preso num ciclo somente de estados conectados por eventos não-observáveis após sua ocorrência. A hipótese H3 é feita por simplicidade, uma vez que a

análise de diagnosticabilidade é a mesma aplicada para um único tipo de falha. Embora o conjunto Σ_f seja particionado em diferentes subconjuntos $\Sigma_{f_i}, i = 1,2,...,m$, sendo que cada conjunto Σ_{f_i} é formado por eventos que modelam falhas correlacionadas, será considerado somente um único tipo de falha, i.e., $\Pi_f = \{\Sigma_f\}$, em que $\Sigma_f = \{f\}$.

Denota-se por $\Psi_L(f)$ o conjunto de todas as cadeias de L que terminam com o evento f. Formalmente, tem-se: $\Psi_L(f) = \{rf \in L : r \in \Sigma^*,\ f \in \Sigma\}$.

A seguir é apresentada a definição formal de Diagnosticabilidade de uma linguagem, segundo Sampath et al. (1995):

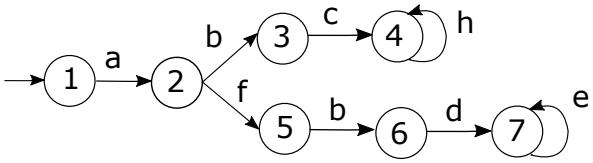
Definição 1 *(Diagnosticabilidade (SAMPATH et al., 1995)).* Uma linguagem L prefixo-fechada que é viva e não contém ciclos de eventos não-observáveis, é dita diagnosticável em relação à projeção P_o e ao evento f se a seguinte condição for verificada: $(\exists n \in \mathbb{N})(\forall s \in \Psi_L(f))(\forall t \in L/s)(\|t\| \geq n \Rightarrow \mathscr{D})$, sendo que a condição de diagnosticabilidade $\mathscr{D}$ é expressa como: $\forall \omega \in P_o^{-1}[P_o(st)] \cap L \Rightarrow f \in \omega$.

Em palavras, essa definição estabelece que uma linguagem é diagnosticável se for possível detectar a ocorrência do evento f com um número finito de eventos depois da ocorrência de f, utilizando somente sequências de eventos observáveis. A diagnosticabilidade requer que cada evento de falha conduza a observações distintas o suficiente para permitir a identificação única da falha f com um atraso finito de transições do sistema.

Os exemplos a seguir ilustram uma linguagem diagnosticável e uma linguagem não-diagnosticável.

Exemplo 5 *Considere o Autômato G_5 ilustrado na Figura 3.1, cuja linguagem é dada por $L_5 = \overline{a(bch^* + fbde^*)}$, sendo $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,c,d,e,h\}$ e $\Sigma_f = \{f\}$. Note que, após a cadeia de eventos observáveis abd, é possível concluir que a falha ocorreu.*

Figura 3.1 – Exemplo de linguagem diagnosticável. Autômato G_5.

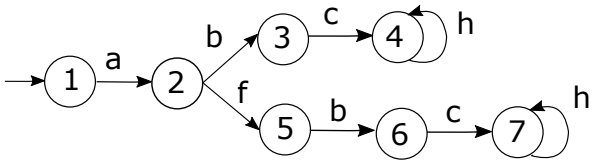


Fonte: (Autor.)

Exemplo 6 *Considere o Autômato G_6 ilustrado na Figura 3.2, cuja linguagem é dada por $L_6 = \overline{a(bch^* + fbch^*)}$, sendo $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,c,h\}$ e $\Sigma_f = \{f\}$. Observe que não é*

possível identificar a ocorrência da falha f mesmo após a sequência bch^* *indefinidamente.*

Figura 3.2 – Exemplo de linguagem não-diagnosticável. Autômato G_6.



Fonte: (Autor.)

3.3 DIAGNOSTICABILIDADE DE UMA CADEIA

Com base no conceito de diagnosticabilidade de uma linguagem, introduz-se a seguir o conceito de diagnosticabilidade de uma cadeia.

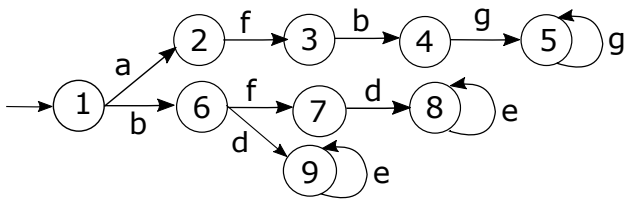
Definição 2 *(Cadeia Diagnosticável).* A ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é diagnosticável em relação a P_o se $(\exists n \in \mathbb{N})(\forall t \in L/s)(\|t\| \geq n \Rightarrow \mathscr{D})$, sendo que a condição de diagnosticabilidade $\mathscr{D}$ é expressa como: $\forall v \in P_o^{-1}[P_o(st)] \cap L \Rightarrow f \in v$.

Em palavras, a ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é diagnosticável se é possível identificar esta particular ocorrência de f usando somente sequências de eventos observáveis dentro de um atraso finito, independentemente se a falha é diagnosticável ou não em outra cadeia.

O exemplo a seguir ilustra a noção de cadeia diagnosticável.

Exemplo 7 *Considere o autômato* G_7 *ilustrado na Fig. 3.3 e sua linguagem* $L_7 = \overline{afbgg^* + b(de^* + fde^*)}$*, sendo que* $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,d,e,g\}$ *e* $\Sigma_f = \{f\}$.

Figura 3.3 – Exemplo de cadeia diagnosticável e não-diagnosticável. Autômato G_7.



Fonte: (Autor.)

Existem duas cadeias $s \in \Psi_{L_7}(f)$ em L_7, isto é, $s_1 = af$ e $s_2 = bf$. A cadeia s_1 é diagnosticável, pois observando a sequência ab (com $n = 1$ e $t_1 = b \in L_7/s_1$), tem-se a

certeza da ocorrência da falha. Entretanto, a cadeia s_2 não é diagnosticável, pois para a cadeia $t_2 = de^* \in L_7/s_2$, $\nexists n \in \mathbb{N}$ tal que a condição $\mathscr{D}$ é satisfeita, uma vez que há uma cadeia $v_2 = bde^*$ tal que $v_2 \in P_o^{-1}[P_o(s_2t_2)] \cap L_7$ na qual $f \notin v_2$.

A partir da Definição 2, pode-se reescrever a definição original de diagnosticabilidade, apresentada por Sampath et al. (1995), conforme segue.

Uma linguagem L prefixo-fechada, que é viva e não contém ciclos de eventos não-observáveis, é dita diagnosticável em relação à projeção P_o e ao evento f se a ocorrência do evento f é diagnosticável em todas as cadeias $s \in \Psi_L(f)$.

3.4 VERIFICAÇÃO DA DIAGNOSTICABILIDADE

Com o intuito de verificar se uma linguagem é diagnosticável, pode-se usar um autômato determinístico chamado diagnosticador, denotado por G_d. Esse diagnosticador possui dois objetivos: verificar *offline* se a linguagem gerada pelo autômato G é diagnosticável; e realizar a diagnose do evento de falha f a partir da observação do comportamento do sistema *online*. A estrutura do sistema a ser considerada é a centralizada, na qual um único diagnosticador é construído a partir do modelo global da planta G, o qual tem acesso a todos os eventos observáveis do sistema.

O diagnosticador G_d possui o alfabeto igual ao conjunto dos eventos observáveis de G e os estados apresentam os rótulos F e N para indicar se o evento f ocorreu ou não.

Formalmente, G_d é representado como:

$$G_d = (Q_d, \Sigma_o, \delta_d, q_{d,0}), \tag{3.1}$$

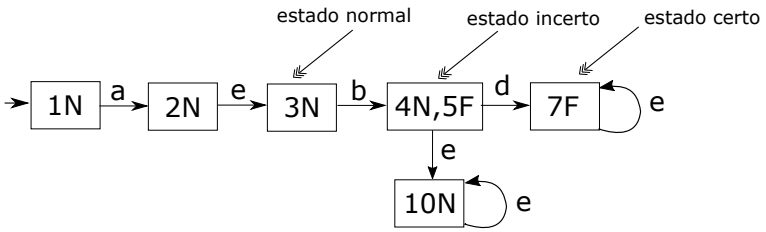
sendo que Q_d denota o conjunto de estados do diagnosticador, Σ_o é o conjunto finito de eventos observáveis, $\delta_d : Q_d \times \Sigma_o \to Q_d$ é a função de transição de estado do diagnosticador, e $q_{d,0} \in Q_d$ é o estado inicial do diagnosticador. O espaço de estados do diagnosticador Q_d é um subconjunto de $2^{Q \times \{N,F\}}$. O estado $q_d \in Q_d$ é da forma $q_d = \{(q_1,l_1),...,(q_m,l_m)\}$, sendo $q_i \in Q$ e $l_i \in \{N,F\}$ para $i = 1,...,m$. Seja $q'_d \in Q_d$ um estado no diagnosticador G_d tal que q'_d é alcançado a partir de q_d por $\sigma_o \in \Sigma_o$, ou seja, $q'_d = \delta_d(q_d, \sigma_o)$. Sejam $q_d = \{(q_1,l_1),...,(q_m,l_m)\}$ e $q'_d = \{(q'_1,l'_1),...,(q'_n,l'_n)\}$. Para todo $i \in \{1,...,m\}$, existe $j \in \{1,...,n\}$ tal que $q'_i = \delta(q_j, s)$, sendo $s = t\sigma_o$ e $t \in \Sigma_{uo}^*$, e

$$l'_i = \begin{cases} F, & \text{se} \quad l_j = F \quad \text{ou} \quad (f \in s), \\ N, & \text{se} \quad l_j = N \quad \text{e} \quad (f \notin s). \end{cases}$$

Diz-se que um estado $q_d = \{(q_1, l_1), ..., (q_m, l_m)\} \in Q_d$ para $m \in \mathbb{N}$ é: normal se $l_j = N$ para todo $j = 1, ..., n$; certo de falha se $l_i = F$ para todo $i = 1, ..., n$; e incerto de falha se existe $l_j = N$ e $l_i = F$ para algum $i, j \in \{1, ..., n\}$, $i \neq j$.

Em palavras, G_d pode ser usado para fazer a diagnose *online* de falhas ao observar o comportamento de G através dos rótulos nos estados. Se um estado apresentar todos os rótulos do tipo F, significa que houve uma falha e denomina-se este de estado certo de falha. Se um estado apresentar pelo menos um rótulo F e pelo menos um do tipo N, então é denominado um estado incerto de falha. Se um estado apresentar todos os rótulos do tipo N, significa que não houve falha, e é denominado um estado normal. A Figura 3.4 ilustra os três tipos de estados do diagnosticador G_d. Para simplificar a notação, é usual representar os estados de G_d como qN e qF ao invés de (q,F) e (q,N), respectivamente. A seguir é mostrado como G_d pode ser construído em dois passos:

Figura 3.4 – Tipos de estados do diagnosticador G_d.

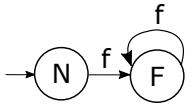


Fonte: (Autor.)

1) Obter a composição paralela $G||A_l$, sendo A_l o autômato rotulador de falhas de dois estados mostrado na Figura 3.5;
2) Calcular o autômato Observador $Obs(G||A_l, \Sigma_o)$, conforme Cassandras e Lafortune (2008).

Como visto no Capítulo 2, têm-se duas estratégias para calcular o autômato Observador: incluir ou não incluir o alcance não-observável, o que resulta em dois diagnosticadores diferentes.

Figura 3.5 – Autômato rotulador de falhas A_l.



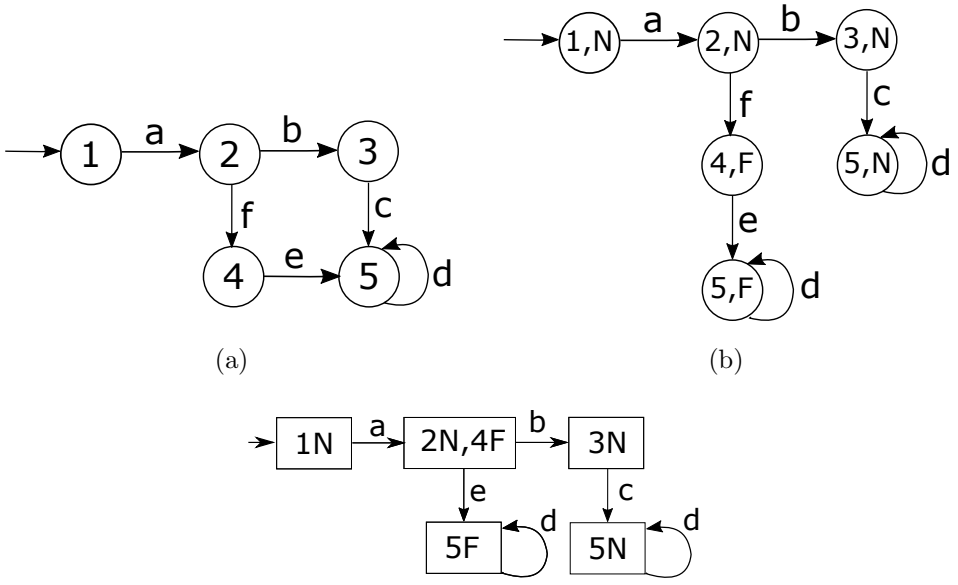
Adaptado de Carvalho (2011).

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

Exemplo 8 *Seja o autômato G_8 mostrado na Figura 3.6(a), cuja linguagem é dada por $L_8 = \overline{a(fe+bc)d^*}$, sendo $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,c,d,e\}$ e $\Sigma_f = \{f\}$. Na Figura 3.6(b)*

mostra-se o resultado obtido com a composição paralela $G_8||A_l$; e na Figura 3.6(c) apresenta-se o autômato diagnosticador G_{d8} calculando o observador da composição paralela, ou seja, $G_{d8} = Obs(G_8||A_l, \Sigma_o)$.

Figura 3.6 – Exemplo de diagnosticador. (a) Autômato G_8; (b) Composição paralela $G_8||A_l$; (c) Autômato diagnosticador $G_{d8} = Obs(G_8||A_l, \Sigma_o)$.



(a) (b)

(c)

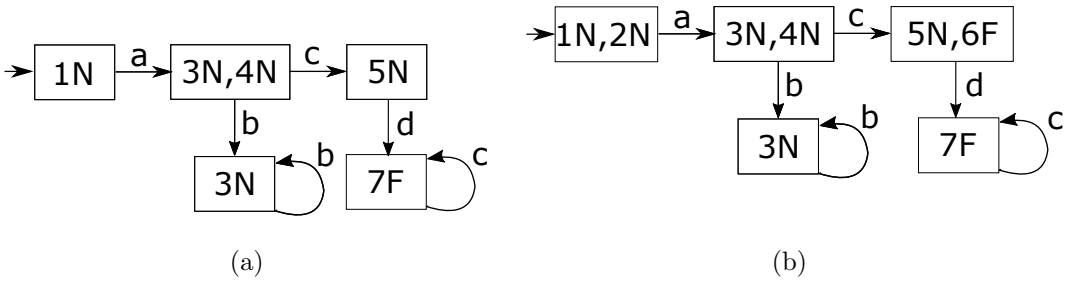
Fonte: (Autor.)

Note que o autômato da Figura 3.6(b), obtido após a composição paralela realizada no primeiro passo, gera a mesma linguagem que G e os estados de $G||A_l$ são da forma (q,F) ou (q,N), dependendo se f está ou não na sequência que leva q_0 até q; consequentemente $q_d \in 2^{Q\times\{N,F\}}$. A Figura 3.6(c) mostra o diagnosticador $G_{d8} = Obs(G_8||A_l,\Sigma_o)$. Note que o estado (5) de G_8 se divide nos estados $(5F)$ e $(5N)$ devido à existência de duas sequências distintas $s_1 = afe$ e $s_2 = abc$, sendo que somente a sequência s_1 contém o evento de falha f.

Conforme visto anteriormente, pode-se obter o diagnosticador sem e com alcance não-observável nos estados do diagnosticador. Para simplificar, neste trabalho adotam-se os nomes s-diagnosticador e c-diagnosticador para denotar os diagnosticadores sem e com alcance não-observável, respectivamente (WATANABE et al., 2017b). Na Figura 3.7 (a) mostra-se o s-diagnosticador G^s_{d4} e na Figura 3.7 (b) ilustra-se o c-diagnosticador G^c_{d4} para o autômato G_4 da Figura 2.3.

O s-diagnosticador é utilizado em diversos trabalhos encontrados na literatura, dentre os quais se podem destacar: (SAMPATH et al., 1995), (LAFORTUNE et al., 2001) and (GENC; LAFORTUNE, 2006, 2009).

Figura 3.7 – Exemplo de diagnosticador sem e com alcance não-observável. (a) Autômato s-diagnosticador G^s_{d4}; (b) Autômato c-diagnosticador G^c_{d4}.



(a) (b)

Fonte: (Autor.)

No cálculo do c-diagnosticador, reúnem-se num mesmo estado do diagnosticador todos os estados da planta que são alcançados a partir do estado em análise em decorrência de eventos não-observáveis ou sequências de eventos não-observáveis. Assim, como no estado (1) de G_4 existe uma transição com o evento não-observável e, que leva ao estado (2), então no diagnosticador obtém-se um estado formado pelos estados (1) e (2), conforme mostrado na Figura 3.7 (b). O c-diagnosticador também é utilizado em diversos trabalhos, dentre os quais se podem citar: (BASILIO; LAFORTUNE, 2009), (CASSANDRAS; LAFORTUNE, 2008), (PAOLI; LAFORTUNE, 2005; PAOLI; SARTINI; LAFORTUNE, 2011), (WATANABE et al., 2017a), (BASILIO; CARVALHO; MOREIRA, 2010) e (CARVALHO; BASILIO; MOREIRA, 2013).

Observação 1 *O algoritmo para a obtenção do autômato diagnosticador possui complexidade exponencial em relação a cardinalidade do espaço de estados do autônomo cuja linguagem gerada se deseja diagnosticar. Para solucionar esse problema foram propostos os chamados autômatos verificadores (MOREIRA; JESUS; BASILIO, 2011), cujos espaços de estados dos autômatos são polinomiais em relação à cardinalidade do espaço de estados de* G*. Porém, o diagnosticador apresenta a vantagem de permitir a diagnose do evento* f *durante o funcionamento do sistema (online), enquanto que o uso dos verificadores está restrito somente para fins de verificação acerca da possibilidade de diagnose de um evento, o que é feito offline. Recentemente, um estudo mostra que diagnosticadores e verificadores têm aproximadamente a mesma complexidade computacional média (CLAVIJO; BASILIO, 2017). Os autores Jiang et al. (2001), Yoo e Lafortune (2002) e Qiu e Kumar (2006) propuseram um método para verificar a diagnosticabilidade centralizada de SEDs e Wang, Yoo e Lafortune (2007), Basilio e Lafortune (2009) e Qiu e Kumar (2006) para a diagnose descentralizada.*

3.5 CONDIÇÕES PARA DIAGNOSTICABILIDADE DE UMA LINGUAGEM

Nesta seção são apresentadas condições necessárias e suficientes para que uma linguagem seja diagnosticável. Essas condições utilizam o conceito de ciclo indeterminado. Um ciclo indeterminado é formado por estados incertos de falhas em G_d e dois ciclos em G, sendo que um aparece sem a ocorrência de falha (rótulo N) e outro depois da ocorrência de falha (rótulo F), correspondentes ao ciclo de estados incertos em G_d.

As provas dessas condições são apresentadas em (SAMPATH et al., 1995).

Teorema 1 *(Condições para Diagnosticabilidade (SAMPATH et al., 1995)). Uma linguagem L gerada por um autômato G é diagnosticável em relação à projeção P_o e evento f se, e somente se, o seu diagnosticador G_d não possuir ciclos indeterminados.*

Vale destacar que as condições estabelecidas no Teorema 1 podem ser analisadas sobre qualquer um dos diagnosticadores, com ou sem alcance não-observável, pois a mudança na forma de obter o diagnosticador não altera a existência de ciclos indeterminados.

A linguagem gerada pelo autômato do Exemplo 5 da Figura 3.1 é diagnosticável, pois tanto no s-diagnosticador demonstrada da Figura 3.7 (a) como no c-diagnosticador da Figura 3.7 (b), não há ciclos indeterminados.

A seguir, é apresentado um exemplo para ilustrar uma linguagem que não é diagnosticável devido à noção de ciclo indeterminado.

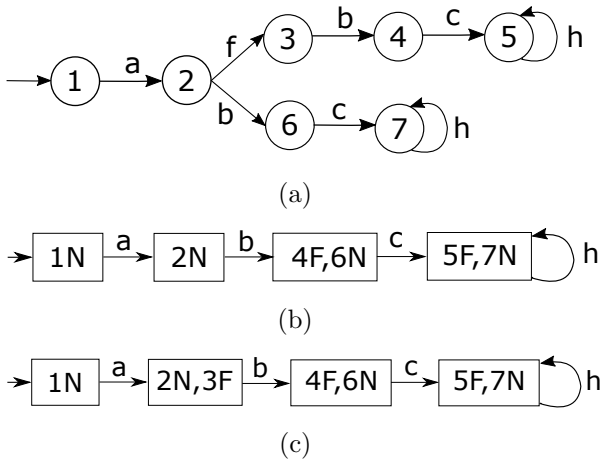
Exemplo 9 *Seja o autômato G_9 mostrado na Figura 3.8(a), cuja linguagem é dada por $L_9 = \overline{a(fbch^* + bch^*)}$, sendo $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,c,h\}$ e $\Sigma_f = \{f\}$.*

Observando as Figuras 3.8(b) e (c), verifica-se a existência de um ciclo de estado incerto no estado $(5F, 7N)$ tanto no s-diagnosticador G^s_{d9} como no c-diagnosticador G^c_{d9}. Já na Figura 3.8(a), observando os dois ciclos na planta G, um aparece sem a ocorrência da falha f (rótulo N) no estado (7) e o outro depois da ocorrência da falha f (rótulo F) no estado (5), correspondentes ao ciclo de estados incertos no estado $(5F, 7N)$ em G^s_{d9} ou G^c_{d9}.

A seguir, apresenta-se um exemplo de um SED cuja linguagem é diagnosticável.

Exemplo 10 *Seja o autômato G_{10} mostrado na Figura 3.9(a), cuja linguagem é dada por $L_{10} = \overline{a(fbdh^* + bch^*)}$, sendo $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,c,d,h\}$ e $\Sigma_f = \{f\}$.*

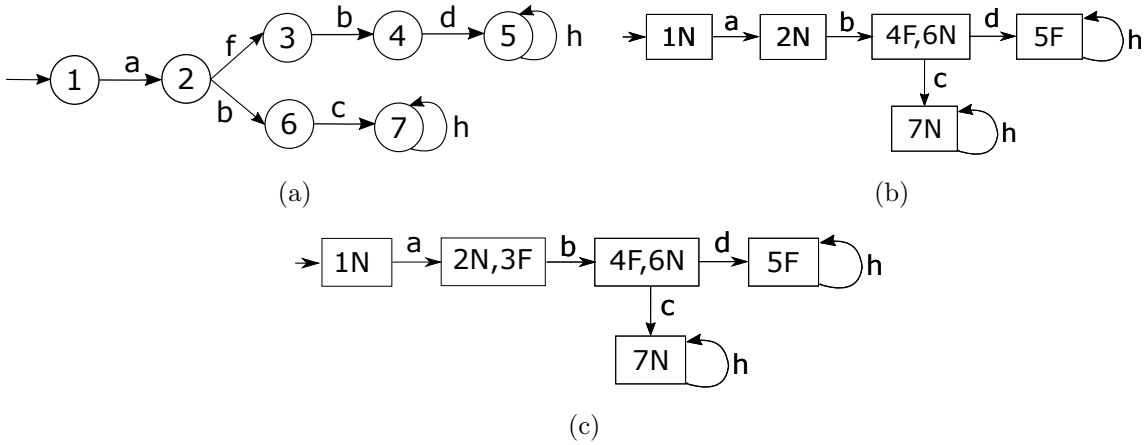
Figura 3.8 – Exemplo de linguagem não-diagnosticável. (a) Autômato G_9; (b) Autômato s-diagnosticador G^s_{d9}; (c) Autômato c-diagnosticador G^c_{d9}.



Fonte: (Autor.)

Observe que o s-diagnosticador G^s_{d10} da Figura 3.9(b) e o c-diagnosticador G^c_{d10} da Figura 3.9 (c) não apresentam ciclos de estados incertos, consequentemente podemos afirmar que a linguagem é diagnosticável.

Figura 3.9 – Exemplo de linguagem diagnosticável. (a) Autômato G_{10}; (b) Autômato s-diagnosticador G^s_{d10}; (c) Autômato c-diagnosticador G^c_{d10}.



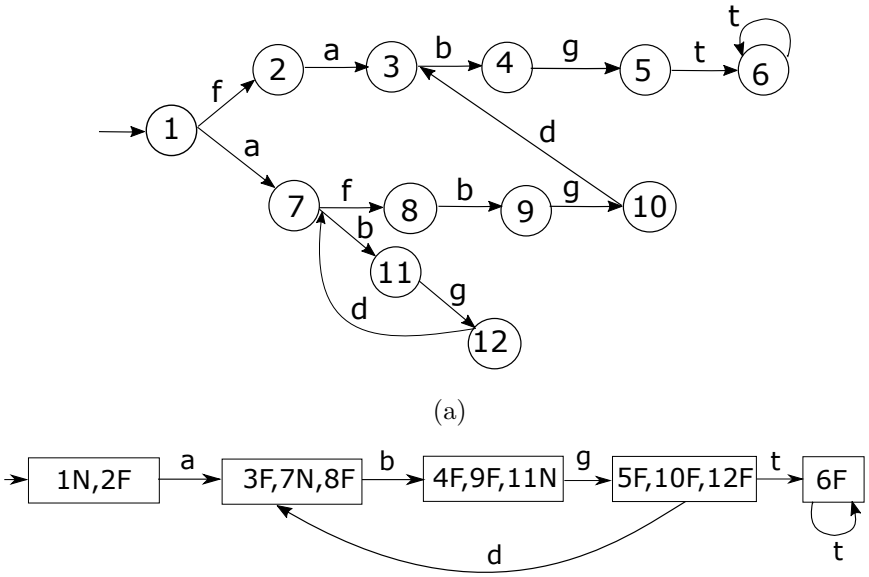
Fonte: (Autor.)

Observação 2 *É importante observar que a existência de um ciclo de estados incertos no diagnosticador não necessariamente implica na não diagnosticabilidade da ocorrência de uma falha. A linguagem L é diagnosticável em relação a P_o e evento f se esses ciclos de estados incertos não forem ciclos indeterminados. A seguir é apresentado um exemplo que caracteriza essa situação, conforme é explanado em Cassandras e Lafortune (2008) - pg. 114.*

Exemplo 11 *Seja o autômato G_{11} mostrado na Figura 3.10(a), cuja linguagem é dada por $L_{11} = \overline{(fa+a(bgd)^*fbgd)bgtt^*}$, sendo $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,d,g,t\}$ e $\Sigma_f = \{f\}$.*

Observe que o diagnosticador G_{11} da Figura 3.10(b) apresenta ciclo de estado incerto, porém no autômato G_{11} ilustrado na Figura 3.10(a) não apresenta ciclo de estados incertos após a falha, descaracterizando um ciclo indeterminado, assim pode-se afirmar que a linguagem L_{11} é diagnosticável.

Figura 3.10 – Exemplo de linguagem diagnosticável. (a) Autômato G_{11}; (b) Autômato diagnosticador G_{11}.



(a)

(b)

Fonte: (Baseado em Cassandras e Lafortune (2008).)

3.6 CONDIÇÕES PARA DIAGNOSTICABILIDADE DE UMA CADEIA

Nesta seção são apresentadas condições necessárias e suficientes para que a ocorrência de falha numa cadeia seja diagnosticável.

Conforme tratado anteriormente, a diagnosticabilidade de uma linguagem está associada à não existência de ciclos indeterminados. De forma análoga, para o estabelecimento de condições necessárias e suficientes para que uma cadeia s seja diagnosticável utiliza-se neste trabalho o conceito de ciclo indeterminado relativo a uma cadeia $s \in \Psi_L(f)$. A definição de ciclo indeterminado relativo a uma cadeia s foi baseada na Definição 5 do trabalho de Basilio, Carvalho e Moreira (2010). As condições apresentadas na definição a seguir valem tanto para o c-diagnosticador como para o s-diagnosticador.

Definição 3 *(Ciclo Indeterminado Relativo a uma Cadeia s).* Seja $G_d = (Q_d, \Sigma_o, \delta_d, q_{d,0})$ obtido a partir de $G = (Q, \Sigma_o, \delta, q_0)$. Um conjunto de estados incertos $\{q_{d1}, q_{d2}, ..., q_{dp}\} \subset Q_d$ forma um ciclo indeterminado relativo a uma cadeia $s \in \Psi_L(f)$ se as seguintes condições forem satisfeitas:
1) $q_{d1}, q_{d2}, ..., q_{dp}$ forma um ciclo em G_d, i.e., $\exists \mathrm{v}_o = \sigma_1\sigma_2...\sigma_p \in L(G, q_{d1})$, tal que $\delta_d(q_{dl}, \sigma_l) = q_{dl+1}, l = 1, ..., p-1$ e $\delta_d(q_{dp}, \sigma_p) = q_{d1}$.
2) $\exists (q_l^{k_l}, F), (\tilde{q}_l^{r_l}, N) \in q_{d_l}$, $q_l^{k_l}$ não necessariamente distinto de $\tilde{q}_l^{r_l}$, $l = 1, 2, ..., p$, $k_l = 1, 2, ..., m_l$ e $r_l = 1, 2, ..., \tilde{m}_l$ de tal sorte que as sequências de estados $\{q_l^{k_l}\}$, $l = 1, 2, ..., p$, $k_l = 1, 2, ..., m_l$, e $\{\tilde{q}_l^{r_l}\}$, $l = 1, 2, ..., p$, $r_l = 1, 2, ..., \tilde{m}_l$ podem ser rearranjadas para formar ciclos em G, cujas sequências correspondentes v e $\tilde{v}$, formadas com os eventos que definem a evolução dos ciclos, têm como projeção a cadeia $v_o = \sigma_1\sigma_2...\sigma_p$, a qual é conforme definido no item 1).
3) $\exists u \in \Sigma^* : \hat{\delta}(q_0, su) = q_l^{k_l}$ para $l \in \{1, 2, ..., p\}$ e $k_l \in \{1, 2, ..., m_l\}$, em que $q_l^{k_l}$ é conforme definido no item 2).

Em palavras, um conjunto de estados incertos forma um ciclo indeterminado relativo à uma cadeia $s \in \Psi_L(f)$ se existir um ciclo de estados incertos no diagnosticador G_d e dois ciclos em G, sendo que um aparece antes da ocorrência de falha e o outro depois da ocorrência de falha na cadeia s, correspondentes ao ciclo de estados incertos de G_d.

A seguir, apresenta-se um exemplo para ilustrar a noção de um ciclo indeterminado relativo a uma cadeia s.

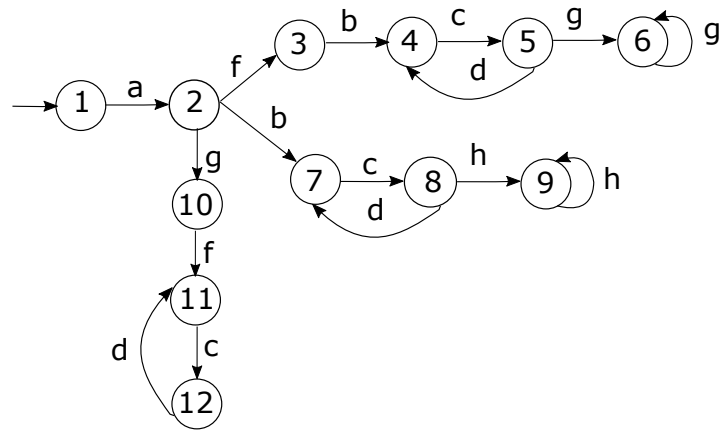
Exemplo 12 *Considere que o autômato G_{12} mostrado na Figura 3.11(a), cuja linguagem é dada por $L_{12} = \overline{a(fb(cd)^*gg^* + b(cd)^*hh^* + gf(cd)^*)}$, sendo $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a, b, c, d, g, h\}$ e $\Sigma_f = \{f\}$.*

O autômato G_{12} apresenta um ciclo indeterminado relativo a cadeia $s = af \in \Psi_L(f)$, pois existe um ciclo de estados incertos (4F,7N) e (5F,8N) no s-diagnosticador G_{d12}^s e esse ciclo possui dois ciclos correspondentes em G, sendo que um antes da ocorrência de falha (estados (7) e (8)) e outro depois da ocorrência da falha na cadeia s (estados (4) e (5)). Observa-se que o resultado é o mesmo considerando o c-diagnosticador G_{d12}^c.

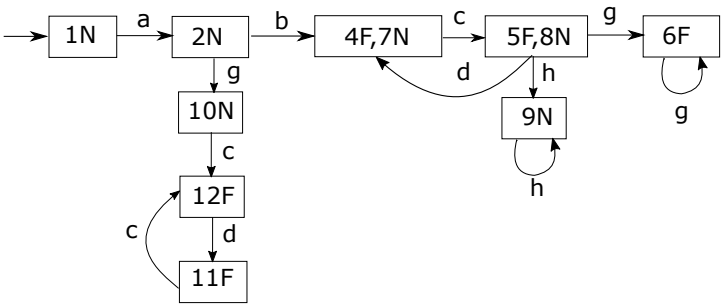
As condições apresentadas na proposição a seguir valem tanto para o c-diagnosticador como para o s-diagnosticador.

Proposição 1 *(Condições para Diagnosticabilidade de uma Cadeia).* Considere uma linguagem L e um autômato $G = (Q, \Sigma, \delta, q_0)$ que gera L. Seja $G_d = (Q_d, \Sigma_o, \delta_d, q_{d,0})$ o diagnosticador construído a partir de G. A ocorrência do evento f numa cadeia $s \in \Psi_L(f)$

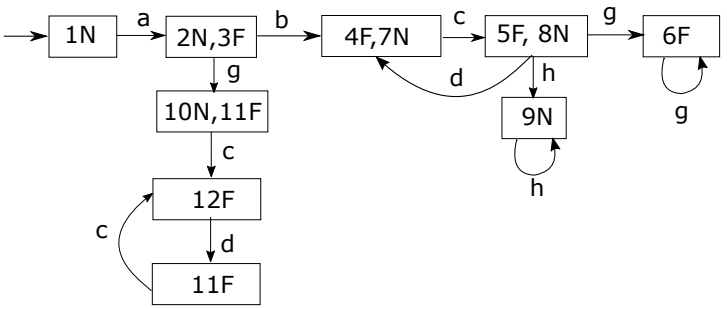
Figura 3.11 – Exemplo para ilustrar ciclo indeterminado relativo a cadeia s. (a) Autômato G_{12}; (b) Autômato s-diagnosticador G^s_{d12}; (c) Autômato c-diagnosticador G^c_{d12}.



(a)



(b)



(c)

Fonte: (Autor.)

é diagnosticável em relação a P_o se e somente se o diagnosticador G_d não tiver ciclos indeterminados relativos à cadeia s.

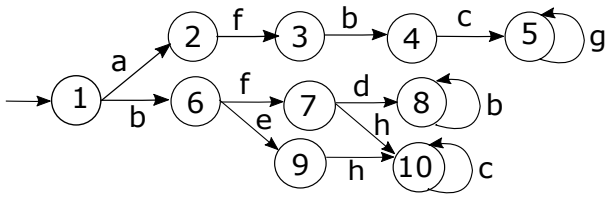
A prova dessa proposição é análoga a de condições para diagnosticabilidade para linguagem.

A seguir, apresenta-se um exemplo para ilustrar as condições estabelecidas na Proposição 1.

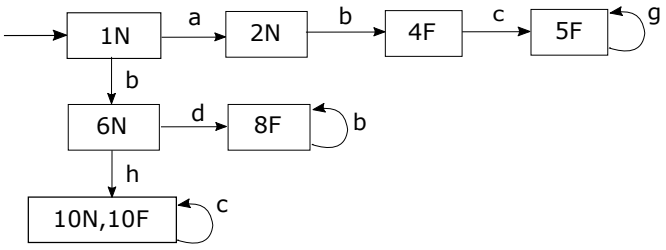
Exemplo 13 *Considere que o autômato* G_{13} *mostrado na Figura 3.12(a), cuja linguagem é dada por* $L_{13} = \overline{afbcg^* + bf(db^* + hc^*) + behc^*}$, *sendo* $\Sigma_{uo} = \{e, f\}$, $\Sigma_o = \{a, b, c, d, g, h\}$ *e* $\Sigma_f = \{f\}$.

Analisando o s-diagnosticador G^s_{d13} e c-diagnosticador G^c_{d13} apresentados nas Figuras 3.12b e 3.12c, respectivamente, pode-se observar que ambos levam às mesmas conclusões.

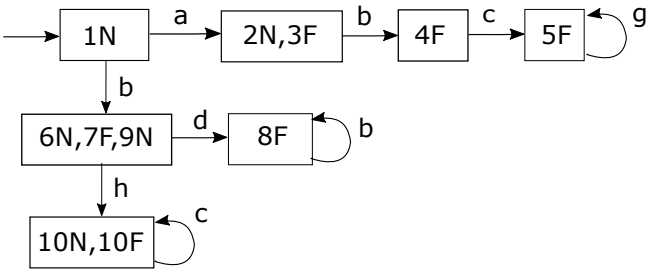
Figura 3.12 – Exemplo para ilustrar a análise da condição para cadeia diagnosticável. (a) Autômato G_{13}; (b) Autômato s-diagnosticador G^s_{d13}; (c) Autômato c-diagnosticador G^c_{d13}.



(a)



(b)



(c)

Fonte: (Autor.)

Conforme discutido anteriormente, existem duas cadeias $s \in \Psi_{L_{13}}(f)$ em L_{13}, ou seja, $s_1 = af$ and $s_2 = bf$. A cadeia s_1 é diagnosticável uma vez que não existe nenhum ciclo indeterminado relativo a ela. Por outro lado, a cadeia s_2 não é diagnosticável, uma vez que existe um ciclo indeterminado no estado $(10N, 10F)$ que é relativo à cadeia s_2, não satisfazendo a condição da Proposição 1.

As condições do Teorema 1 são válidas para verificação da diagnosticabilidade da linguagem através de cadeias.

3.7 DIAGNOSE SEGURA E DIAGNOSTICABILIDADE SEGURA DE FALHAS DE UMA LINGUAGEM

Nesta seção é tratado o problema da diagnose segura de falhas no contexto de SEDs. Mais precisamente, a partir da definição de diagnosticabilidade em SEDs dada na seção anterior, o problema da diagnose segura consiste em realizar a detecção das falhas antes do sistema executar uma cadeia proibida ou ilegal (neste trabalho estes termos serão usados indistintamente). Considere o caso em que se quer evitar que após ocorrer uma falha, o sistema execute uma cadeia proibida de um dado conjunto finito Φ, sendo:

$$\Phi = \{\xi \in \Sigma^* : \xi \text{ é uma cadeia proibida depois da falha}\}, \tag{3.2}$$

Os elementos do conjunto Φ capturam sequências de eventos que são ilegais após a ocorrência de uma falha. Essa situação pode ser formalizada definindo a linguagem ilegal $\mathscr{K}_f$, conforme Paoli e Lafortune (2005), como:

$$\mathscr{K}_f = \{y \in L/s : [s \in \Psi_L(f) \wedge \exists \xi \in \Phi : \xi \text{ é uma subcadeia de} y]\} \tag{3.3}$$

Em palavras, $\mathscr{K}_f$ contém todas as possíveis continuações após um evento f que possuem uma sequência proibida de Φ como subcadeia.

A seguir é apresentada a definição de diagnosticabilidade segura segundo Paoli e Lafortune (2005):

Definição 4 *(Diagnosticabilidade Segura (PAOLI; LAFORTUNE, 2005)).* Uma linguagem L prefixo-fechada, viva, e que não contém ciclos de eventos não-observáveis, é dita ser diagnosticável segura em relação a projeção P_o, evento f e a linguagem proibida $\mathscr{K}_f$, se as seguintes condições são atendidas:
($D4_1$) Condição de Diagnosticabilidade: L é diagnosticável em relação à P_o e f;
($D4_2$) Condição de segurança: $(\forall s \in \Psi_L(f))(\forall t \in L/s)$, tal que $\|t\| = n$, considerando que $t_c, \|t_c\| = n_{tc}$, seja o mais curto prefixo de t tal que $\mathscr{D}$ seja atendida; então $\overline{t_c} \cap \mathscr{K}_f = \emptyset$.

Em palavras, essa definição diz que uma linguagem é diagnosticável segura se for diagnosticável e, após um evento f, a continuação mais curta que assegura a diagnose não contém nenhuma cadeia ilegal.

A seguir, o Exemplo 5 da Figura 3.1 é retomado para exemplificar uma linguagem diagnosticável segura. Considere que o conjunto de cadeias proibidas após o evento f é $\Phi = \{b\}$, e a linguagem proibida é $\mathscr{K}_f = \{dbc^*\}$.

Observe que a ocorrência do evento b após a cadeia ea não é considerada proibida. Pode-se afirmar que L_5 é diagnosticável segura, pois além de diagnosticável, na continuação mais curta que assegura a diagnose ($t_c = d$) não contém a cadeia proibida, ou seja, $s = acf, t_c = d$ e $\overline{t_c} \cap \mathscr{K}_f = \emptyset$.

A seguir, essa definição é adaptada para o contexto de cadeias e introduz-se o conceito de cadeia diagnosticável segura.

3.8 DIAGNOSTICABILIDADE SEGURA DE UMA CADEIA

A diagnosticabilidade segura de uma cadeia $s \in \Psi_L(f)$ será usada como base para apresentar a noção de controlabilidade segura de uma cadeia pela diagnose, o que é feito no Capítulo 5.

Definição 5 *(Cadeia Diagnosticável Segura).* Considere uma linguagem L diagnosticável. A ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é diagnosticável segura em relação a P_o e $\mathscr{K}_f$ se ($\forall t \in L/s$, tal que $\mathscr{D}$ é satisfeita para st e não é satisfeita para nenhum sr, com $r < t$), $\bar{t} \cap \mathscr{K}_f = \emptyset$.

Em palavras, a ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é diagnosticável segura se a cadeia for diagnosticável sem incluir nenhuma cadeia ilegal.

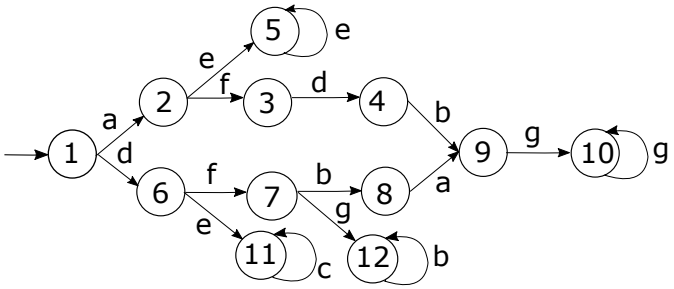
A seguir, apresenta-se um exemplo para ilustrar a noção de cadeia diagnosticável segura.

Exemplo 14 *Considere o autômato G_{14} mostrado na Fig. 3.13, cuja linguagem é dada por $L_{14} = \overline{aee^* + d(ec^* + fgb^*) + (afdb + dfba)gg^*}$, sendo que $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,c,d,e,g\}$ e $\Sigma_f = \{f\}$. O conjunto de cadeias proibidas após o evento f é $\Phi = \{g\}$, e a linguagem ilegal é $\mathscr{K}_f = \{dbgg^*, bagg^*, gb^*\}$.*

Existem duas cadeias $s \in \Psi_{L_{14}}(f)$ em L_{14}, isto é, $s_1 = af$ and $s_2 = df$. A cadeia s_1 é diagnosticável segura, pois a condição $\mathscr{D}$ é satisfeita para $s_1 t_1 = afd$ ($t_1 = d$) e $\overline{t_1} \cap \mathscr{K}_f = \emptyset$. A cadeia s_2 não é diagnosticável segura, pois $\exists t_2 = g \in L_{14}/s_2$ tal que a condição $\mathscr{D}$ é satisfeita para $s_2 t_2$, mas não é satisfeita para nenhum $s_2 r_2$ com $r_2 < t_2$, para qual $\overline{t_2} \cap \mathscr{K}_f \neq \emptyset$. Isto é, existe uma continuação de s_2 na qual a menor cadeia observável necessária para a diagnose é tal que inclui um elemento de Φ como subcadeia.

A partir da Definição 5 , pode-se reescrever a definição original de Paoli e Lafortune (2005), conforme segue.

Figura 3.13 – Exemplo de cadeia diagnosticável segura e outra não-diagnosticável segura. Autômato G_{14}.



Fonte: (Autor.)

Uma linguagem L prefixo-fechada, que é viva e não contém ciclos de eventos não-observáveis, é dita diagnosticável segura em relação à projeção P_o e ao evento f se a ocorrência do evento f é diagnosticável segura em todas as cadeias $s \in \Psi_L(f)$.

3.9 VERIFICAÇÃO DA DIAGNOSTICABILIDADE SEGURA

Para a análise da diagnosticabilidade segura de uma linguagem, Paoli e Lafortune (2005) introduziram o chamado diagnosticador seguro. A seguir apresentam-se os passos para a obtenção de tal diagnosticador.

1) Construir o rotulador A_s, o qual possui três ou mais estados. O estado inicial, rotulado com $(q_0 - NB)$, um estado alcançado com o evento de falha, rotulado com $(e - NB)$, e um mau estado, alcançado com a ocorrência de um evento (ou sequência de eventos) proibido após a falha. Esse último é rotulado com $(\sigma_b - B)$ e chamado de mau estado (*Bad state*).

2) Obter o autômato do diagnosticador seguro $G_{sd} = (Q_{sd}, \Sigma_o, \delta_{sd}, q_{sd,0})$, fazendo $Obs(G||A_s, \Sigma_o)$.

É importante ressaltar que em Paoli e Lafortune (2005), o cálculo do diagnosticador seguro é feito de forma a incluir o alcance não-observável nos estados do diagnosticador, o qual denomina-se c-diagnosticador seguro G^c_{sd}.

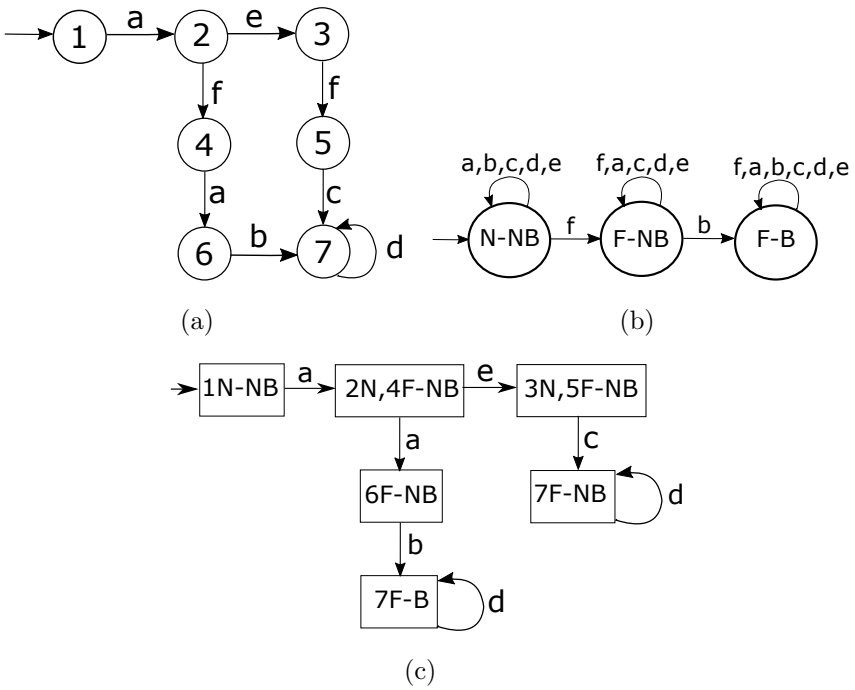
Portanto, denotamos por maus estados aqueles que correspondem a estados em G que são alcançados com a execução de uma cadeia pertencente a um conjunto Φ.

A seguir, apresenta-se um exemplo para ilustrar a construção de um diagnosticador seguro considerando um único evento ilegal.

Exemplo 15 *Considere que o autômato G_{15} mostrado na Figura 3.14(a), cuja linguagem é dada por $L_{15} = \overline{a(fab+efc)d^*}$, sendo $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,c,d,e\}$, $\Phi = \{b\}$ e $\Sigma_f = \{f\}$.*

Seguindo os passos para se obter o c-diagnosticador seguro G^c_{sd15}, é construído o rotulador A_{s15}, conforme ilustrado na Figura 3.14(b). Através do cálculo do observa-

Figura 3.14 – Exemplo de construção do diagnosticador seguro. (a) Autômato G_{15}; (b) Autômato rotulador A_{s15}; (c) Autômato c-diagnosticador Seguro G^c_{sd15}.



Fonte: (Autor.)

dor da composição síncrona da planta G_{15} com o rotulador A_{s15} é obtido o autômato c-diagnosticador seguro G^c_{sd15} representado na Figura 3.14 (c).

A Figura 3.14 (c) mostra o c-diagnosticador seguro G^c_{sd15} para G_{15} e considera que $\Phi = \{b\}$. Observe que no G^c_{sd15} aparecem dois rótulos denominados NB(*Not Bad*) e B(*Bad*), sendo que o rótulo B denota os estados alcançados com cadeias proibidas de Φ.

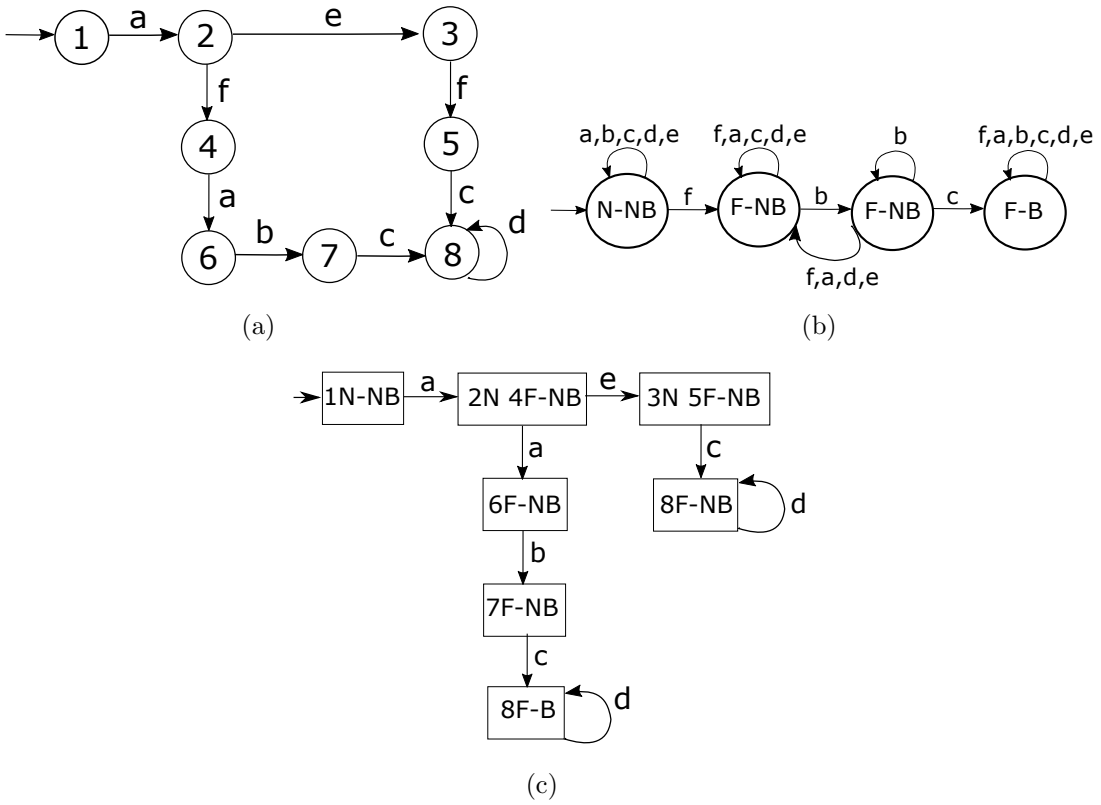
A seguir, apresenta-se outro exemplo de diagnosticador seguro com rotulador para o caso de uma cadeia ilegal.

Exemplo 16 *Considere o autômato G_{16} mostrado na Figura 3.15(a), cuja linguagem é dada por $L_{16} = \overline{a(fabc+efc)d^*}$, sendo $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,c,d,e\}$, $\Sigma_f = \{f\}$, e $\Phi = \{bc\}$.*

Seguindo os passos para se obter o c-diagnosticador seguro G^c_{sd16}, é construído o rotulador A_{s16}, conforme ilustrado na Figura 3.15(b). Através do cálculo do observador da composição síncrona da planta G_{16} com o rotulador A_{s16} é obtido o autômato c-diagnosticador seguro G^c_{sd16} representado na Figura 3.15 (c).

O diagnosticador seguro $G_{sd} = (Q_{sd}, \Sigma_o, \delta_{sd}, q_{sd,0})$ é um autômato construído do modelo de $G = (Q, \Sigma, \delta, q_0)$ e pode ser usado para testar a propriedade de diagnosticabilidade segura. Segundo Genc e Lafortune (2009), a escolha de incluir ou não o alcance não-observável pode afetar a estrutura do diagnosticador, mas não deveria afetar os resul-

Figura 3.15 – Exemplo de construção do diagnosticador seguro. (a) Autômato G_{16}; (b) Autômato rotulador A_{s16}; (c) Autômato c-diagnosticador Seguro G^c_{sd16}.



Fonte: (Autor.)

tados dele derivados. Entretanto, conforme será mostrado a seguir, condições necessárias e suficientes estabelecidas para diagnosticabilidade segura, quando expressas em termos de condições a serem verificadas sobre o diagnosticador, devem ser adaptadas conforme o tipo de diagnosticador usado.

3.10 CONDIÇÕES PARA DIAGNOSTICABILIDADE SEGURA DE UMA LINGUAGEM

O Teorema 2 determina condições necessárias e suficientes para a diagnosticabilidade segura de uma linguagem. As provas estão descritas em Paoli e Lafortune (2005).

Teorema 2 *(Condições para Diagnosticabilidade Segura (PAOLI; SARTINI; LAFORTUNE, 2011)). Considere uma linguagem diagnosticável L e um autômato $G = (Q, \Sigma, \delta, q_0)$ que gera L. L é diagnosticável segura em relação à projeção P_o, evento f e linguagem ilegal $\mathscr{K}_f$ se, e somente se, no c-diagnosticador seguro G^c_{sd} obtido a partir de G:*
($T2_1$) Não existir um estado $q_{sd} \in Q_{sd}$ que seja incerto de falha e que tenha um componente da forma (l,q) tal que $l = F$ e $q = B$;

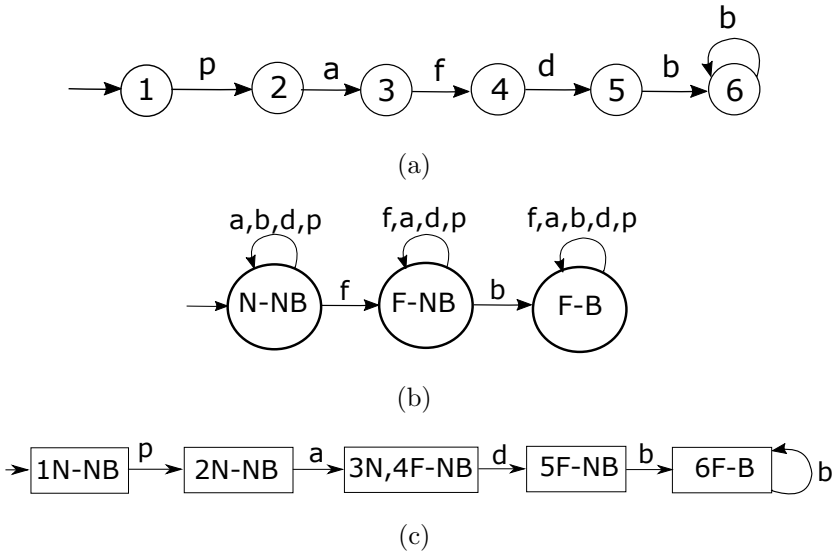
($T2_2$) *Não existir um par de estados* q_{sd}, q'_{sd} *tal que: (i)* q_{sd} *seja um estado certo de falha com um componente na forma* (l,q) *tal que* $l = F$ *e* $q = B$*; (ii)* q'_{sd} *seja um estado incerto de falha; e (iii)* q_{sd} *seja alcançável de* q'_{sd} *através de um evento* $e \in \Sigma_o$.

Em palavras, L é diagnosticável segura se, e somente se, no diagnosticador seguro todo mau estado for um estado certo de falha e seus antecessores imediatos não forem estados incertos.

A seguir, apresenta-se um exemplo para ilustrar condições para a diagnosticabilidade segura considerando apenas o c-diagnosticador, uma vez que as condições estabelecidas por Paoli e Lafortune (2005) foram baseadas nesse tipo de diagnosticador.

Exemplo 17 *Considere o autômato* G_{17} *mostrado na Figura 3.16(a), cuja linguagem é dada por* $L_{17} = \overline{pafdbb^*}$*, sendo* $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,d,p\}$, $\Sigma_f = \{f\}$, $\Phi = \{b\}$ *e a linguagem ilegal* $\mathscr{K}_f = \{dbb^*\}$.

Figura 3.16 – Exemplo de análise das condições para linguagem diagnosticável segura pelo Teorema 2. (a) Autômato G_{17}; (b) Autômato rotulador A_{s17}; (c) Autômato c-diagnosticador seguro G^c_{sd17}.



Fonte: (Autor.)

Seguindo os passos para se obter o c-diagnosticador seguro, é construído o rotulador A_{s17}, conforme ilustrado na Figura 3.16 (b). Adotando o procedimento anteriormente explicado, obtém-se o autômato c-diagnosticador seguro G^c_{sd17} representado na Figura 3.16 (c). Podemos concluir que L_{17} é diagnosticável segura, pois ela obedece as condições do Teorema 2, sendo que o mau estado $(6F-B)$ é um estado certo de falha e o seu antecessor não é um estado incerto de falha, mas certo de falha $(5F-NB)$.

Observação 3 *Os resultados da análise dos exemplos mostrados até o momento não são afetados pelo uso de c-diagnosticador ou s-diagnosticador. Porém, isso não ocorre em todos os casos. Portanto, a seguir são introduzidas novas condições para a prognosticabilidade estabelecidas sobre os ambos os tipos de diagnosticadores.*

A diferença das condições do Teorema 2 é que neste teorema os antecessores imediatos aos maus estados, além de não poderem ser estados incertos de falha, também não podem ser estados normais, ou seja, só podem ser estados certos de falha.

Teorema 3 *(Novas condições para Diagnosticabilidade Segura de uma Linguagem). Considere uma linguagem diagnosticável L e um autômato $G = (Q, \Sigma, \delta, q_0)$ que gera L. L é diagnosticável segura em relação à projeção P_o, evento f e linguagem ilegal $\mathscr{K}_f$ se, e somente se, no diagnosticador seguro G_{sd} obtido a partir de G:*
($T3_1$) Não existe um estado $q_{sd} \in Q_{sd}$ que seja incerto de falha e que tenha um componente da forma (q,l) tal que q seja um mau estado e $l = F$.
*($T3_2$) Não existe um par de estados $q_{sd}, q'_{sd} \in Q_{sd}$ tal que: (i) q_{sd} seja um estado certo de falha com um componente na forma (q,l) tal que q seja um mau estado e $l = F$; (ii) q'_{sd} seja um estado incerto de falha ou **estado normal**; e (iii) $q_{sd} = \delta(q'_{sd}, \sigma_o)$ com $\sigma_o \in \Sigma_o$.*

Em palavras, L é diagnosticável segura se, e somente se, no diagnosticador seguro os maus estados são certos de falha e seus antecessores imediatos são estados certos de falhas.

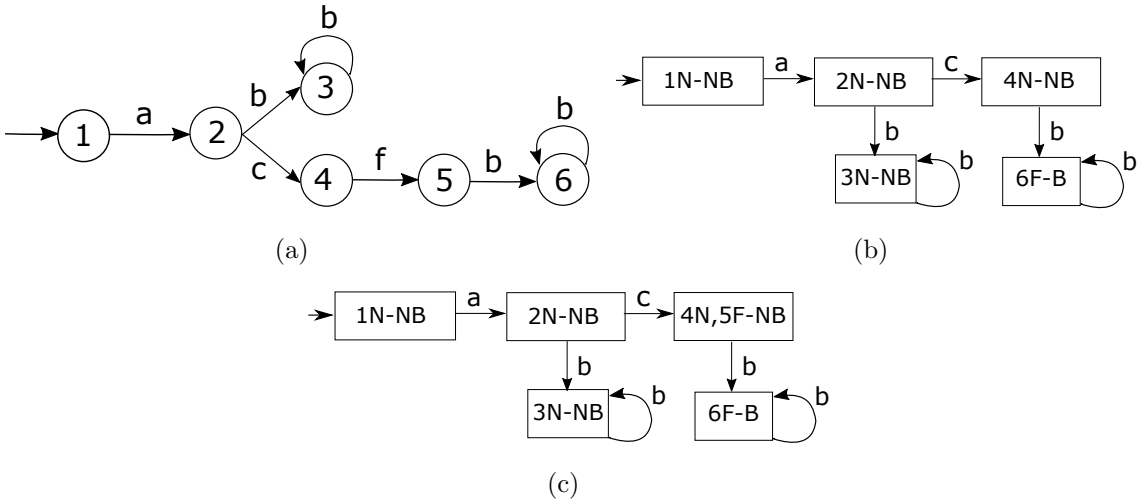
A prova do Teorema 3 é similar ao Teorema 2 apresentado em Paoli e Lafortune (2005), pois as modificações introduzidas nesse Teorema não afetam a prova.

A seguir, apresenta-se um exemplo para ilustrar condições para a diagnosticabilidade segura considerando ambos os tipos de diagnosticadores.

Exemplo 18 *Considere o Autômato G_{18} mostrado na Figura 3.17 (a), cuja linguagem é dada por $L_{18} = \overline{a(bb^* + cfbb^*)}$, sendo $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,c\}$ e $\Sigma_f = \{f\}$. O conjunto de cadeias proibidas após a falha é $\Phi = \{b\}$, e a linguagem proibida é $\mathscr{K}_f = \{bb^*\}$.*

Ao analisar o c-diagnosticador seguro G^c_{sd18} da Figura 3.17 (c), observa-se que existe um estado incerto de falha $(4N, 5F - NB)$ que é imediatamente sucedido pelo mau estado $(6F - B)$. Sendo assim, pode-se concluir que L_{18} não é diagnosticável segura. Ao analisar o s-diagnosticador (ver Figura 3.17 (b)), observa-se que existe um estado normal $(4N - NB)$ que é imediatamente sucedido pelo mau estado $(6F - B)$. Portanto, ambos os diagnosticadores naõ atendem as condições do Teorema 3 e, portanto, essa linguagem não é diagnosticável segura.

Figura 3.17 – Exemplo de análise das condições para linguagem diagnosticável segura pelo Teorema 3. (a) Autômato G_{18}; (b) Autômato s-diagnosticador seguro G^s_{sd18}; (c) Autômato c-diagnosticador seguro G^c_{sd18}.



Fonte: (Autor.)

A seguir é introduzida a condição necessária e suficiente que garante que a ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é diagnosticável segura.

3.11 CONDIÇÕES PARA DIAGNOSTICABILIDADE SEGURA DE UMA CADEIA

Nesta seção serão introduzidas condições necessárias e suficientes para que a ocorrência do evento f numa cadeia seja diagnosticável segura. Antes, porém, apresenta-se a função que mapeia uma cadeia $s \in \Psi_L(f)$ no conjunto dos primeiros estados certo de falha q_{sd} no diagnosticador seguro G_{sd}, que são alcançados com a menor cadeia observável que permite a diagnose da falha em s. Formalmente, considerando $Q^N_{sd} = \{q_{sd} \in Q_{sd} : q_{sd}$ é normal$\}$, $Q^C_{sd} = \{q_{sd} \in Q_{sd} : q_{sd}$ é certo de falha$\}$, e $Q^U_{sd} = \{q_{sd} \in Q_{sd} : q_{sd}$ é incerto de falha$\}$, define-se $FC(s) = \{q_{sd} \in Q^C_{sd} : [(q_{sd} = \hat{\delta}_{sd}(q_{sd,0}, v_o)$, com $v_o = P_o(st), t \in L/s) \wedge (\nexists q'_{sd} \in Q^C_{sd} : q'_{sd} = \hat{\delta}_{sd}(q_{sd,0}, v'_o)$ com $v'_o < v_o)]\}$.

Para essa proposição, é também necessário apresentar o conjunto Q^B. Seja Q^B um conjunto de maus estados (*Bad states*), definido formalmente como $Q^B = \{q_{sd} \in Q_{sd} : \exists (q,l) \in q_{sd}$ tal que q é um mau estado$\}$.

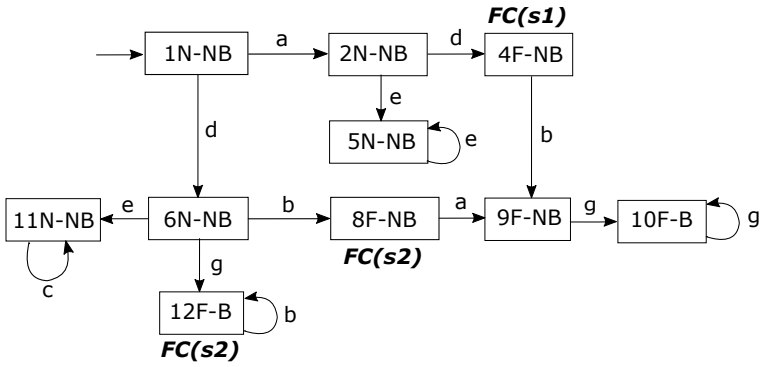
Proposição 2 *(Condições para Diagnosticabilidade Segura de uma Cadeia).* Considere uma linguagem diagnosticável L e um autômato $G = (Q, \Sigma, \delta, q_0)$ que gera L. Seja $G_{sd} = (Q_{sd}, \Sigma_o, \delta_{sd}, q_{sd,0})$ o diagnosticador seguro construído a partir de G. A ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é diagnosticável segura em relação à P_o e $\mathscr{K}_f$ se e somente se $\nexists q_{sd} \in FC(s)$, tal que $q_{sd} \in Q^B$.

Prova. A prova é em duas partes:
($\Rightarrow$) Primeiro, é provada por contradição que a ocorrência de f numa cadeia $s \in \Psi_L(f)$ é diagnosticável segura somente se $\nexists q_{sd} \in FC(s)$, tal que $q_{sd} \in Q^B$. Suponha que a ocorrência de f numa cadeia $s \in \Psi_L(f)$ é diagnosticável segura, mas $\exists q_{sd} \in FC(s)$ tal que $q_{sd} \in Q^B$. Pela definição de Q^B, sabe-se que se $q_{sd} \in Q^B$ então $\exists (q,l) \in q_{sd}$ tal que q é um mau estado. Suponha que este estado $q \in Q$ é alcançado em G por uma cadeia $st \in L$, tal que $s \in \Psi_L(f)$, ou seja, $q = \hat{\delta}(q_0, st)$. Uma vez que q é um mau estado, pode-se afirmar que t contém um elemento de Φ como uma subcadeia, e então $\bar{t} \cap \mathscr{K}_f \neq \emptyset$. Sem perda de generalidade, considere que $t = r\sigma$, com $r \in \Sigma^*$ e $\sigma \in \Sigma_o$. Considere também que $v_o = P_o(st)$. Então, $q_{sd} = \hat{\delta}_{sd}(q_{sd,0}, v_o)$. Uma vez que $q_{sd} \in FC(s)$, sabe-se que $\nexists q'_{sd} \in Q^C : q'_{sd} = \hat{\delta}_{sd}(q_{sd,0}, v'_o)$ com $v'_o < v_o$. Além disso, a condição $\mathscr{D}$ é atendida para st, enquanto não é atendida para nenhum sr, com $r < t$. Assim, $\exists t \in L/s$ tal que $\mathscr{D}$ é atendida para st, e não é atendida para nenhum sr, com $r < t$, para o qual $\bar{t} \cap \mathscr{K}_f \neq \emptyset$. Portanto, pela Definição 5 a ocorrência do evento f em $s \in \Psi_L(f)$ não é diagnosticável segura, violando a hipótese inicial.
($\Leftarrow$)Esta parte da prova também é feita por contradição. Suponha que a ocorrência de f numa cadeia $s \in \Psi_L(f)$ não é diagnosticável segura e que $\nexists q_{sd} \in FC(s)$ tal que $q_{sd} \in Q^B$. Por hipótese, a linguagem L é diagnosticável e então tem-se que a cadeia s é diagnosticável. Assim, se a ocorrência de f numa cadeia s não é diagnosticável segura, então, pela Definição 5, $\exists t \in L/s$ tal que $\mathscr{D}$ é satisfeita para st, mas não é satisfeita para nenhuma cadeia sr, com $r < t$, tal que $\bar{t} \cap \mathscr{K}_f \neq \emptyset$. Sem perda de generalidade, considere que $t = r\sigma$, com $t, r \in \Sigma^*$ e $\sigma \in \Sigma_o$. Considere ainda que $v_o = P_o(st)$ e $v'_o = P_o(sr)$, logo $v'_o < v_o$. Além disso, sejam $q_{sd} = \hat{\delta}_{sd}(q_{sd,0}, v_o)$ e $q'_{sd} = \hat{\delta}_{sd}(q_{sd,0}, v'_o)$ estados do diagnosticador seguro que correspondem a estados q e q' em G que são alcançados com as cadeias st e sr, respectivamente. Assim, $q = \hat{\delta}(q_0, st)$ e $q' = \hat{\delta}(q_0, sr)$. O fato da condição $\mathscr{D}$ser satisfeita pra st e não ser satisfeita para sr implica que $q_{sd} \in Q^C_{sd}$ e $q'_{sd} \notin Q^C_{sd}$. Sendo assim, pela definição de $FC(s)$, pode-se afirmar que $q_{sd} \in FC(s)$. Tendo em vista que $\bar{t} \cap \mathscr{K}_f \neq \emptyset$, sabe-se que a cadeia t possui um elemento de Φ como subcadeia e, portanto, o estado $q \in Q$ alcançado por essa cadeia é um mau estado. Dessa forma, $(q,l) \in q_{sd}$ e $q_{sd} \in Q^B$. Sendo assim, $\exists q_{sd} \in FC(s)$ tal que $q_{sd} \in Q^B$, contrariando a hipótese inicial. □

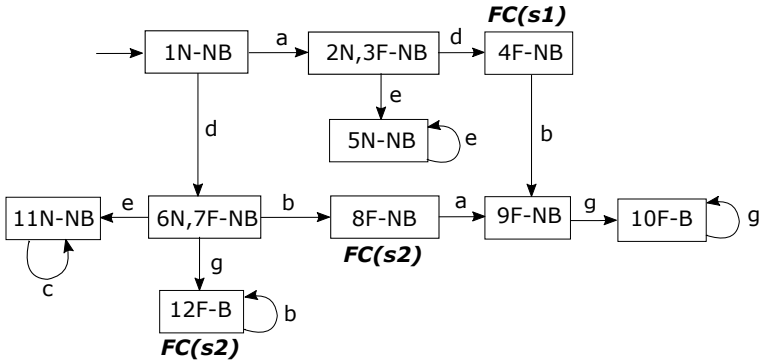
Em palavras, a ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é diagnosticável segura se e somente se nenhum dos estados do conjunto $FC(s)$ for um mau estado.

A seguir, retoma-se o Exemplo 14 da Figura 3.13 para ilustrar a análise de condições para diagnosticabilidade segura numa cadeia. A Figura 3.18 mostra o s-diagnosticador seguro G^s_{Csd14} e o c-diagnosticador seguro G^c_{Csd14} para análise.

Figura 3.18 – Exemplo para ilustrar a análise da condição para cadeia diagnosticável segura. (a) Autômato s-diagnosticador G^s_{sd14}; (b) Autômato c-diagnosticador G^c_{sd14}.



(a)



(b)

Fonte: (Autor.)

Conforme discutido anteriormente, há duas cadeias $s \in \Psi_{L_{14}}(f)$ em L_{14}, ou seja, $s_1 = af$ and $s_2 = df$. Para essas cadeias, tanto para o s-diagnosticador G^s_{sd14} como para o c-diagnosticador G^c_{sd14}, têm-se $FC(s_1) = \{(4F-NB)\}$ e $FC(s_2) = \{(8F-NB),(12F-B)\}$. A cadeia s_1 é diagnosticável segura uma vez que o estado $(4F-NB)$ não é um mau estado. Por outro lado, a cadeia s_2 não é diagnosticável segura uma vez que o estado $(12F-B) \in FC(s2)$ é um mau estado, não satisfazendo a condição da Proposição 2.

A seguir é introduzido um teorema que apresenta condições para linguagem diagnosticável através da abordagem por cadeias.

Teorema 4 *(Condições para Diagnosticabilidade Segura de uma Linguagem na abordagem por cadeias.). Considere uma linguagem diagnosticável L e um autômato $G = (Q, \Sigma, \delta, q_0)$ que gera L. Seja $G_{sd} = (Q_{sd}, \Sigma_o, \delta_{sd}, q_{sd,0})$ o diagnosticador seguro construído a partir de G. A linguagem L é diagnosticável segura se e somente se para toda cadeia $s \in \Psi_L(f)$, $\nexists q_{sd} \in FC(s)$ tal que $q_{sd} \in Q^B$.*

Ese teorema substitui o Teorema 3 e sua prova é similar a da Proposição 2.

3.12 CONSIDERAÇÕES FINAIS

Neste capítulo foram revisados os problemas da diagnose de falhas e diagnose segura em SEDs. Após, introduzimos os novos conceitos de diagnose de falha e diagnose segura de falhas no contexto de cadeias. Também foram introduzidas condições necessárias e suficientes para diagnosticabilidade e diagnosticabilidade segura de uma cadeia. Foram apresentados exemplos para cada definição e proposição para fins didáticos.

Foram mostradas as duas formas de obtenção dos autômatos diagnosticadores, a saber, o autômato diagnosticador sem incluir o alcance não-observável no estado do diagnosticador e o que inclui o alcance não-observável no estado do diagnosticador. A Tabela 3.1 apresenta um resumo das condições para diagnosticabilidade e diagnosticabilidade segura para uma linguagem utilizando s-diagnosticador e c-diagnosticador. A Tabela 3.2 apresenta um resumo de condições para diagnosticabilidade e diagnosticabilidade segura para uma cadeia utilizando s-diagnosticador e c-diagnosticador.

Tabela 3.1 – Comparativo entre condições para a diagnose e diagnose segura de uma linguagem estabelecidas a partir dos diferentes diagnosticadores.

Mecanismo (Linguagem)	Tipo do diagnosticador	Condição	Obra
Diagnose	s-diagnosticador	Se, e somente se, o diagnosticador não tem ciclos indeterminados.	(SAMPATH et al., 1995)
	c-diagnosticador	Se, e somente se, o diagnosticador não tem ciclos indeterminados.	(BASILIO; LAFORTUNE, 2009)
Diagnose Segura	s-diagnosticador c-diagnosticador	Se, e somente se, no diagnosticador seguro, os maus estados e seus respectivos antecessores imediatos são estados certos de falha.	(WATANABE et al., 2017b)
	c-diagnosticador	Se, e somente se, no diagnosticador seguro, os maus estados são estados certos de falha e os antecessores imediatos não são estados incertos.	(PAOLI, 2003; PAOLI; LAFORTUNE, 2005)

Fonte: (Autor.)

Os conceitos da diagnosticabilidade segura estudados neste capítulo serão importantes no contexto do capítulo 5 no qual será abordada a controlabilidade segura pela diagnose.

Foi verificada a necessidade da introdução de um novo teorema referente à diagnosticabilidade segura para que condições estabelecidas neste fossem válidas para ambos os tipos de diagnosticadores, com e sem o alcance não-observável. Por fim, destaca-se que

Tabela 3.2 – Comparativo entre condições para a diagnose e diagnose segura de uma cadeia estabelecidas a partir dos diferentes diagnosticadores.

Mecanismo (Cadeia)	Tipo do diagnosticador	Condição	Obra
Diagnose	s-diagnosticador - - - - - - - - - - c-diagnosticador	Se, e somente se, o diagnosticador não tem ciclos indeterminados relativos a cadeia.	Este trabalho
Diagnose segura	s-diagnosticador - - - - - - - - - - c-diagnosticador	Se, e somente se, $\nexists q_{sd} \in FC(s)$, tal que $q_{sd} \in Q^B$.	Este trabalho

Fonte: (Autor.)

a maior contribuição deste capítulo foi a introdução dos conceitos de diagnose e diagnose segura de cadeias, bem como o estabelecimento de condições que garantem a diagnose e a diagnose segura numa cadeia. Esses conceitos serão usados na abordagem da Controlabilidade Segura pela diagnose numa cadeia no Capítulo 5.

4 PROGNOSE DE FALHAS EM SEDS

O problema de prognose de eventos em SEDs trata de inferir sobre a futura ocorrência de um determinado evento. A prognose pode ser realizada tanto de um evento observável quanto de um evento não-observável, porém, neste trabalho será dado foco na prognose de eventos não-observáveis, especificamente de eventos de falhas (evento f). É de interesse especial realizar a prognose de falhas em SEDs modelados por autômatos. Essa capacidade de predizer a falha permite a tomada de medidas preventivas tais como reconfigurar ou até desligar o sistema antes da ocorrência da mesma. A prognose difere da diagnose de falhas que se obtém a detecção de uma falha somente após sua ocorrência.

Este capítulo apresenta a definição de prognosticabilidade introduzida por Genc e Lafortune (2009) e condições necessárias e suficientes para que uma linguagem seja prognosticável. Além disso, como uma importante contribuição desta tese, neste capítulo introduz-se o conceito de prognose de falhas numa cadeia e estabelecem-se condições necessárias e suficientes para que a ocorrência de falha numa cadeia seja prognosticável.

Este capítulo está organizado da seguinte forma. Na seção 4.1 é apresentada uma revisão bibliográfica sobre prognose de falhas. É apresentada também uma classificação estrutural: centralizada, decentralizada e distribuída. A formulação do problema da prognose de falhas e a apresentação da definição de prognosticabilidade de falha de uma linguagem são feitas na seção 4.2. A proposição que estabelece que a diagnosticabilidade é uma condição necessária para prognosticabilidade é apresentada na seção 4.3. Na seção 4.4 é introduzida a noção de cadeia prognosticável. A proposição que estabelece que a diagnosticabilidade de uma cadeia é uma condição necessária para prognosticabilidade de uma cadeia é apresentada na seção 4.5. Na seção 4.6 é apresentado o diagnosticador para análise das condições para prognosticabilidade da linguagem. Na seção 4.7 são apresentadas condições necessárias e suficientes para prognosticabilidade. Também é apresentado o teorema que estabelece as condições para a prognosticabilidade de uma linguagem. Condições necessárias e suficientes para se obter cadeia prognosticável são introduzidas na seção 4.8. Finalmente, na seção 4.9 são apresentadas as considerações finais.

4.1 REVISÃO BIBLIOGRÁFICA SOBRE PROGNOSE DE FALHAS

Esta seção apresenta uma breve revisão dos artigos publicados na área de prognose de falhas. Portanto, o leitor que tiver conhecimento dos trabalhos relacionados poderá dirigir à seção seguinte, sem prejuízos para o entendimento do trabalho.

A classificação da forma estrutural na prognose é, como na diagnose, realizada da seguinte forma: centralizada, decentralizada e distribuída. Este trabalho utiliza a abordagem centralizada.

Prognose Centralizada: Na estrutura centralizada, a prognose se dá por intermédio de um diagnosticador global, como na diagnose. A sua principal vantagem é a simplicidade conceitual, porém modelos grandes podem resultar em grandes complexidades computacionais. A proposta descrita por Ye, Dague e Nouioua (2013) é mostrar um novo algoritmo com complexidade polinomial inspirado em um método de planta gêmea adaptável a uma estrutura distribuída.

Prognose Decentralizada: A estrutura decentralizada é empregada para sistemas de grande porte e fisicamente distribuídos, em que múltiplos prognosticadores locais que dependem de seus próprios subconjuntos de sensores acessíveis tomam decisões de prognose local tal que se fundem para decidir uma decisão global. Kumar e Takai (2010) introduziram a noção de coprognosticabilidade, que serve como uma necessária e suficiente condição para a existência de prognosticador decentralizado.

Prognose Distribuída: Na estrutura distribuída os prognosticadores locais trocam suas observações de eventos executadas pela planta a fim de chegar a uma decisão de prognose (TAKAI; KUMAR, 2009), (TAKAI; KUMAR, 2012). Segundo Takai e Kumar (2012), as observações são trocadas sobre canais de comunicações que introduzem atrasos limitados. Os autores introduziram uma propriedade denominada *joint-prognosability* para capturar as condições sob as quais qualquer falha possa ser prognosticada antes de sua ocorrência.

Os primeiros trabalhos nesta área surgiram com Cao (1989), Buss, Papadimitriou e Tsitsiklis (1991) e Fadel e Holloway (1999). Cao (1989) introduziu a noção de preditibilidade em relação as propriedades de um tipo especial de projeção entre duas linguagens. A análise da predição do estado de sistemas de autômatos idênticos e o cálculo através de um controle global simétrico é realizado por Buss, Papadimitriou e Tsitsiklis (1991). A predição de falhas apresentada por Fadel e Holloway (1999) é baseada em análise estatística. Um aviso é emitido quando há a possibilidade de ocorrer uma falha na evolução do sistema. Porém, não é garantida que haverá emissão de ocorrência de falha com total segurança. O primeiro trabalho sobre predição de eventos em SEDs modelados por linguagens regulares foi desenvolvido por Genc e Lafortune (2006). Esse trabalho foi inspirado no problema de diagnose de falhas em SEDs, assim a noção de diagnosticabilidade de um sistema é usada para obter a necessária e suficiente condição para preditibilidade. De lá para cá, a área de prognose em SEDs tem recebido considerável atenção.

A noção de preditibilidade apresentada por Jeron et al. (2008) é diferente das apresentadas anteriormente por Buss, Papadimitriou e Tsitsiklis (1991) e Fadel e Holloway (1999). Esse trabalho tem uma forte relação com o trabalho de Shengbing e Kumar (2004) que considera a noção de inevitabilidade da diagnose. Jeron et al. (2008) definem a preditibilidade como sendo a detecção da inevitabilidade estritamente antes de sua ocorrência. Além disso, foi estendida para predição da ocorrência de uma certa sequência padrão. Enquanto Fadel e Holloway (1999) consideram que é possível que falsas predições de falhas possam ser avisadas, já Genc e Lafortune (2009) consideram que isso não ocorre. A noção de preditibilidade introduzida por Genc e Lafortune (2009) além de apresentar que a preditibilidade é uma condição mais forte do que a diagnosticabilidade de uma linguagem, mostra que para qualquer evento iminente, é inequivocamente conhecido que ocorrerá dentro de um número uniformemente limitado de passos. Briones e Madalinski (2011) dizem que Genc e Lafortune (2009) não colocam nenhum limite, por conseguinte, uma falha pode ocorrer no instante seguinte depois de ter sido previsto. Em contraste, a abordagem de preditibilidade mencionada por Briones e Madalinski (2011) introduz limites para a ocorrência da falha, um limite inferior e superior, assegurando que uma falha ocorra após um limite inferior de eventos observáveis e antes de um limite superior de eventos observáveis. A proposta dos autores é usar o limite como forma de garantir que haverá tempo suficiente para emitir um aviso ou algumas outras medidas de prevenção antes da ocorrência da falha. Briones e Madalinski (2013) explicam que preditibilidade com limites oferece informações que podem ser explorados pelo sistema para adotar o melhor plano contingencial. Um limite inferior garante que uma falha ocorra após determinado evento de execução, enquanto um limite superior garante que uma falha ocorra, no futuro, mas antes de algum evento. Foi estudado como inferir a propriedade de preditibilidade com limites de um sistema complexo (distribuído e com falhas múltiplas) a partir de uma verificação paralela da preditibilidade com limites de cada um de seus componentes, sincronizando com versões sem falhas do outro. A ocorrência de eventos prognosticáveis para autômatos temporizados parcialmente observáveis foi investigada por Cassez e Grastien (2013). Os autores argumentam que a consideração explícita do tempo é crucial para a prognose de falhas em SEDs. Nessa abordagem é fornecido o tempo restante antes da ocorrência de um evento de falha para poder interromper ou reconfigurar o sistema. A prognosticabilidade estocástica de falhas, usando autômato probabilístico é estudada por Nouioua, Dague e Ye (2014). Chen e Kumar (2014, 2015) estudaram a prognose de falhas em sistemas a eventos discretos estocásticos, sendo introduzida a noção de predição da falha de m-passos antecipados. Enquanto Lefebvre (2014) abordou predição de falhas para sistemas a eventos discretos estocásticos usando redes Petri parcialmente ob-

serváveis, Ammour et al. (2017) investigou o problema da prognose de falhas em SEDs temporizados e estocásticos modelados por redes Petri e YIN (2018) verificou a capacidade de prognosticabilidade utilizando redes de Petri rotuladas. Yokotani e Takai (2014) realizaram o complemento de estudos de propriedades como observabilidade, normalidade, capacidade de diagnose e capacidade de prognose para o controle/diagnose de sistemas a eventos discretos parcialmente observáveis. O problema da preditibilidade em sistemas a eventos discretos parcialmente observáveis, nos quais é possível prever um intervalo de tempo a qual a ocorrência da falha pode ocorrer no sistema (GRASTIEN, 2015). A noção de prognosticabilidade robusta é introduzida por Takai (2015). A prognosticabilidade robusta é uma condição necessária e suficiente para a existência de um prognosticador, de tal forma que, para todos os possíveis modelos, cada qual com sua especificação, ele consegue predizer a falha antes de sua ocorrência. Tal prognosticador é chamado de prognosticador robusto. No cenário da prognose robusta, um único prognosticador prognostica múltiplos modelos, enquanto no cenário da prognose decentralizada/distribuída múltiplos prognosticadores prognosticam um único modelo. O problema da prognose robusta é diferente dos problemas de prognose decentralizada/distribuída considerados por Khoumsi e Chakib (2012), Kumar e Takai (2010) e Takai e Kumar (2012). O agente local pode enviar informações binários, com regra disjuntiva (KHOUMSI; CHAKIB, 2012) e disjuntiva mais conjuntiva (YIN; LI, 2016) para o coordenador. Takai e Kumar (2017) introduziram uma estrutura generalizada para prognose decentralizada baseada em inferência que suporta tanto os esquemas disjuntivos como conjuntivos na tomada de decisão. Yin e Li (2016) propuseram dois novos protocolos decentralizados para o prognose de falhas; ou seja, o protocolo baseado em estimativa de estado positivo (*Positive State Estimate* - PSE) e o protocolo baseado em estimativa de estado negativo (*Negative State Estimate* - NSE). Em ambos protocolos, cada agente local usa um subconjunto de sua estimativa de estado conforme a informação que envia ao coordenador. Mais especificamente, no protocolo baseado em PSE, cada agente local envia o conjunto de estados que possui os motivos pelos quais um alarme de falha deve ser emitido; enquanto no protocolo baseado em NSE, cada agente local envia o conjunto de estados que possui os motivos pelos quais um alarme de falha não deve ser emitido. Em seguida, o coordenador faz a interseção das estimativas locais do estado para calcular uma decisão prognóstica global. Benmessahel, Touahria e Nouioua (2017) estudaram o problema da preditibilidade em sistemas a eventos discretos *fuzzy* (SEDFs). SEDFs combinam teoria de conjunto *fuzzy* com SEDs. Esse trabalho mostra também que preditibilidade é mais forte do que diagnosticabilidade. Um autômato chamado verificador é proposto para verificação da preditibilidade em SEDFs.

4.2 PROGNOSE E PROGNOSTICABILIDADE DE FALHAS DE UMA LINGUAGEM

Nesta seção é tratado o problema da prognose da ocorrência de eventos em SEDs parcialmente observáveis modelados por autômatos de estados finitos. Genc e Lafortune (2009) modelam o sistema como uma linguagem L sobre um conjunto de eventos observáveis e apresentam a seguinte definição.

Definição 6 *(Prognosticabilidade (GENC; LAFORTUNE, 2009)).* Dada uma linguagem L prefixo-fechada e viva sobre Σ, as ocorrências do evento $f \in \Sigma$ são prognosticáveis em L com respeito a P_o se $(\exists n \in \mathbb{N})(\forall s \in \Psi(f))(\exists t < s)[(f \notin t) \wedge \mathscr{P}]$, em que a condição de prognosticabilidade $\mathscr{P}$ é expressa como: $(\forall u \in L)(\forall v \in L/u)[(P_o(u) = P_o(t)) \wedge (f \notin u) \wedge (\|v\| \geq n) \Rightarrow (f \in v)]$.

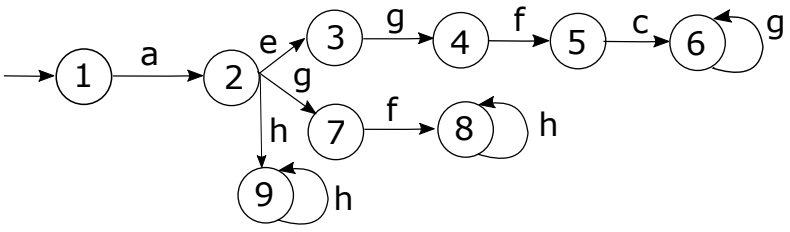
Em palavras, a ocorrência do evento f em L é prognosticável se para cada cadeia s que termina com o evento f, existe um prefixo t de s tal que t não contém o evento f, e todas as continuações suficientemente longas em L das cadeias com a mesma projeção de t contém o evento f.

A seguir, apresenta-se um exemplo para ilustrar uma linguagem prognosticável.

Exemplo 19 *Considere que o autômato G_{19} mostrado na Figura 4.1, cuja linguagem é dada por $L_{19} = \overline{a(hh^* + egfcg^* + gfh^*)}$, sendo $\Sigma_{uo} = \{e, f\}$, $\Sigma_o = \{a, c, g, h\}$ e $\Sigma_f = \{f\}$.*

Segundo a Definição 6, a análise deve ser feita para todas as cadeias que terminam com a falha. Nesse exemplo temos $s_1 = aegf$ e $s_2 = agf$. Na cadeia s_1, existe $t_1 = aeg \in \overline{s_1}$, com $P_o(t_1) = ag$, existe uma cadeia $u_1 = ag \in L$ tal que $P_o(u_1) = P_o(t_1)$ e cuja sua continuação $v_1 = fh^*$ contém o evento f. A análise para a cadeia s_2 é semelhante e resulta na mesma conclusão. Portanto, de acordo com a Definição 6, a linguagem L_{19} é prognosticável.

Figura 4.1 – Exemplo de linguagem prognosticável. Autômato G_{19}.

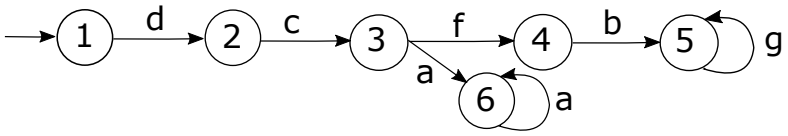


Fonte: (Autor.)

A seguir, apresenta-se um exemplo que ilustra uma linguagem que não é prognosticável.

Exemplo 20 *Considere o autômato G_{20} mostrado na Figura 4.2, cuja linguagem é dada por $L_{20} = \overline{dc(fbg^* + aa^*)}$, sendo que $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,c,d,g\}$ e $\Sigma_f = \{f\}$.*

Figura 4.2 – Exemplo de linguagem diagnosticável, porém não-prognosticável. Autômato G_{20}.



Fonte: (Autor.)

4.3 DIAGNOSTICABILIDADE X PROGNOSTICABILIDADE

Segundo Genc e Lafortune (2009), a prognosticabilidade é uma condição mais forte do que a diagnosticabilidade de uma linguagem L com respeito à f. De acordo com a Proposição 3 (GENC; LAFORTUNE, 2009), a diagnosticabilidade é uma condição necessária para a prognosticabilidade.

Proposição 3 *(Prognosticabilidade × Diagnosticabilidade (GENC; LAFORTUNE, 2009)).* Dada uma linguagem $L \subseteq \Sigma^*$ viva e prefixo-fechada, se a ocorrência de $f \in \Sigma$ for prognosticável em L com respeito a Projeção P_o, então L é diagnosticável em relação a P_o e f.

A prova da Proposição 3 não é apresentada nesta tese, mas pode ser encontrada em Genc e Lafortune (2009).

Em palavras, essa proposição estabelece que se a ocorrência de um evento f não for diagnosticável numa dada linguagem, então a ocorrência deste evento necessariamente não será prognosticável nessa mesma linguagem.

O Exemplo 9 da Figura 3.8 é retomado para ilustrar a condição necessária da diagnose para prognose numa linguagem. A partir do autômato G_9 da Figura 3.8, a linguagem L_9 não é diagnosticável e também não é prognosticável, pois $t = a$, a sua continuação $u = bch^*$, não contém o evento f não importando o quanto a cadeia u seja longa.

Observação 4 *A diagnose segura numa linguagem não é uma condição necessária para a prognose numa linguagem.*

4.4 PROGNOSTICABILIDADE DE UMA CADEIA

De forma análoga ao conceito de diagnosticabilidade de uma cadeia, a seguir apresenta-se o conceito de prognosticabilidade de uma cadeia cadeia $s \in \Psi_L(f)$.

A prognosticabilidade de uma cadeia $s \in \Psi_L(f)$ será usada como uma condição para a controlabilidade segura de uma cadeia pela prognose, a qual será introduzida na Seção 5.9.

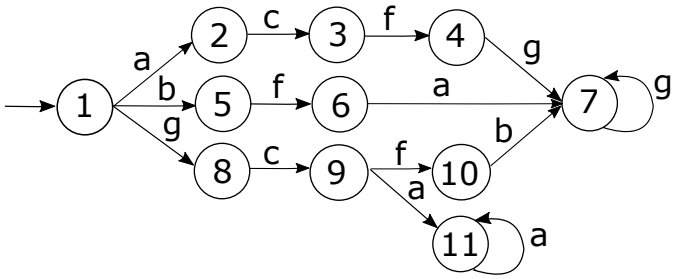
Definição 7 *(Cadeia Prognosticável).* Dada uma linguagem L prefixo-fechada e viva sobre Σ, a ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é prognosticável em relação a P_o se $(\exists t \in \overline{s})[(f \notin t) \wedge \mathscr{P}]$, tal que a condição de prognosticabilidade $\mathscr{P}$ é expressa como: $(\forall t' \in L)(\forall u' \in L/t')[(P_o(t') = P_o(t)) \wedge (f \notin t') \wedge (\exists n \in \mathbb{N})(\|u'\| \geq n) \Rightarrow (f \in u')]$.

Em palavras, a ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é prognosticável se for possível inferir sobre a sua futura ocorrência nessa específica cadeia baseado nos prefixos observáveis dessa cadeia.

A seguir, apresenta-se um exemplo para ilustrar o conceito de cadeia prognosticável.

Exemplo 21 *Considere o autômato G_{21} mostrado na Figura 4.3, cuja linguagem é dada por $L_{21} = \overline{(acfg + bfa + gcfb)g^* + gcaa^*}$, sendo que $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,c,g\}$ e $\Sigma_f = \{f\}$.*

Figura 4.3 – Exemplo de cadeia prognosticável e não-prognosticável. Autômato G_{21}.



Fonte: (Autor.)

Existem três cadeias $s_1, s_2, s_3 \in \Psi_{L_{21}}(f)$ em L_{21}, ou seja, $s_1 = acf$, $s_2 = bf$ e $s_3 = gcf$. A cadeias s_1 é prognosticável, pois existe $t_1 = ac \in \overline{s_1}$, sendo $f \notin t_1$, que atende a condição $\mathscr{P}$, uma vez que observando $P_o(t_1) = ac$, tem-se certeza sobre a futura ocorrência do evento f. A cadeia s_2 é também prognosticável, pois existe $t_2 = b \in \overline{s_2}$, sendo $f \notin t_2$, que atende a condição $\mathscr{P}$, uma vez que observando $P_o(t_2) = b$, tem-se certeza sobre a futura ocorrência do evento f. Por outro lado, s_3 não é prognosticável, pois mesmo para $t_3 = gc$,

que é o mais longo prefixo de s_3, a sua continuação $u_3 = aa^* \in L_{21}/t_3$, não contém o evento f não importando o quanto a cadeia u_3 seja longa.

A partir da Definição 7, pode-se reescrever a definição de prognosticabilidade de (GENC; LAFORTUNE, 2009), conforme segue.

Uma linguagem L prefixo-fechada, que é viva e não contém ciclos de eventos não-observáveis, é dita prognosticável em relação à projeção P_o e ao evento f se a ocorrência do evento f é prognosticável em todas as cadeias $s \in \Psi_L(f)$.

4.5 DIAGNOSTICABILIDADE X PROGNOSTICABILIDADE DE CADEIAS

Assim como no contexto de linguagens, a prognosticabilidade de uma cadeia s é uma condição mais forte do que a diagnosticabilidade de uma cadeia s com respeito à f. Assim, na Proposição 4 estabelece-se que a diagnosticabilidade na cadeia s é uma condição necessária para a prognosticabilidade de uma cadeia.

Proposição 4 *(Prognosticabilidade × Diagnosticabilidade em Cadeias).* Dada uma cadeia $s \in \Sigma^*$ viva e prefixo-fechada, se a ocorrência de $f \in \Sigma$ é prognosticável em s com respeito à Projeção P_o, então L é diagnosticável em relação a P_o e f.

A prova dessa proposição é igual a prova da Proposição 3 de (GENC; LAFORTUNE, 2009).

Em palavras, essa proposição estabelece que se a ocorrência de um evento f não é diagnosticável em uma determinada cadeia, então a ocorrência deste evento necessariamente não será prognosticável nessa mesma cadeia.

Observação 5 *A diagnose segura numa cadeia não é uma condição para a prognose numa cadeia.*

O Exemplo 13 da Figura 3.12 é retomado para ilustrar a necessidade da diagnose, porém não da diagnose segura como uma condição para a prognose no contexto de cadeias. Considera-se que $\Phi = \{b\}$ para a análise da diagnose segura numa cadeia.

Conforme visto anteriormente, existem duas cadeias $s \in \Psi_{L_{13}}(f)$ em L_{13}, ou seja, $s_1 = af$ e $s_2 = bf$. A cadeia s_1 é diagnosticável, porém não é diagnosticável segura, pois $t_1 \in L/s_1 = b$ e $\overline{t_1} \cap \mathscr{K}_f \neq \emptyset$. A cadeia s_1 embora não seja diagnosticável segura, é prognosticável. A cadeia s_2 não é diagnosticável conforme já anteriormente analisado, e

percebe-se que também não é prognosticável, pois $t_2 = b$, a sua continuação $u_2 = ehc^*$, não contém o evento f não importando o quanto a cadeia u_2 seja longa.

4.6 VERIFICAÇÃO DA PROGNOSTICABILIDADE DE UMA LINGUAGEM

De acordo com Genc e Lafortune (2009), a verificação da prognosticabilidade de um evento pode ser realizada de duas formas, uma utilizando diagnosticadores e outra usando verificadores. Tendo em vista que o foco deste trabalho está no uso da prognose *online* para fins de controle, foi utilizado diagnosticadores para verificar condições de prognosticabilidade. Esse autômato diagnosticador é o mesmo utilizado para verificação da diagnosticabilidade citado na Seção 3.4.

4.7 CONDIÇÕES PARA A PROGNOSTICABILIDADE

No teorema a seguir, Genc e Lafortune (2009) estabelecem as condições necessárias e suficientes para a prognosticabilidade da ocorrência de um evento. Essas condições são baseadas na análise dos ciclos do diagnosticador. Antes de apresentar o teorema é necessário introduzir algumas explanações apresentadas no trabalho de Genc e Lafortune (2009). Para prover condições necessárias para a prognosticabilidade foi utilizado o conjunto F_D. Seja F_D o conjunto de estados normais do diagnosticador que possuem um sucessor imediato que não é normal. Formalmente, $F_D = \{q_d \in Q_d^N : (\exists q'_d = \delta_d(q_d, \sigma_o))[(\sigma_o \in \Sigma_o) \wedge (q'_d \notin Q_d^N)]\}$.

Teorema 5 *(Condições para Prognosticabilidade de uma Linguagem (GENC; LAFORTUNE, 2009)). Seja $G = (Q, \Sigma, \delta, q_0)$ o autômato que gera uma linguagem L prefixo-fechada e viva. Seja $G_d^s = (Q_d, \Sigma_o, \delta_d, q_{d,0})$ o s-diagnosticador para G. As ocorrências do evento f são prognosticáveis em L em relação a P_o se e somente se para todo $q_d \in F_D$, a condição $\mathscr{C}$ é satisfeita, sendo que $\mathscr{C}$: todos os ciclos em $A_c(G_d, q_d)$ são ciclos de estados certos no diagnosticador.*

Em palavras, esse teorema estabelece que uma linguagem é prognosticável se e somente se todos os ciclos existentes nos estados alcançados a partir dos estados que pertencem a F_D, são ciclos de estados certos no diagnosticador G_d.

A prova desse teorema é apresentada por Genc e Lafortune (2009). A prova do mesmo é baseada no diagnosticador sem alcance não-observável e a condição estabelecida

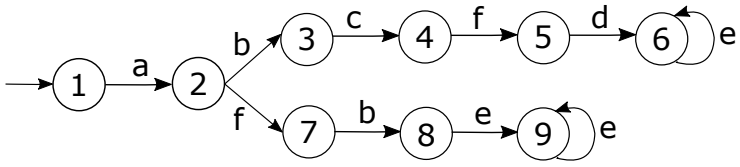
para prognosticabilidade é válida somente para esse tipo de diagnosticador, conforme será ilustrado posteriormente.

A seguir, são apresentados alguns exemplos a fim de ilustrar a análise das condições estabelecidas no Teorema 5. Isso será feito para ambos os tipos de diagnosticadores, ou seja, diagnosticador sem alcance não-observável (s-diagnosticador) nos estados do diagnosticador (GENC; LAFORTUNE, 2009) e o diagnosticador com alcance não observável (c-diagnosticador).

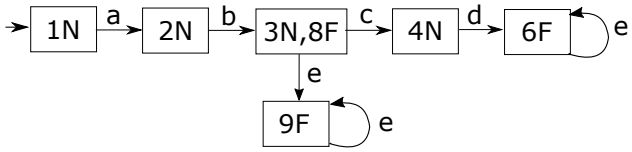
O primeiro exemplo ilustra as condições de uma linguagem prognosticável.

Exemplo 22 *Considere o autômato G_{22} mostrado na Figura 4.4 (a), cuja linguagem é dada por $L_{22} = \overline{a(bcfde^* + fbee^*)}$, sendo que $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,c,d,e\}$ e $\Sigma_f = \{f\}$.*

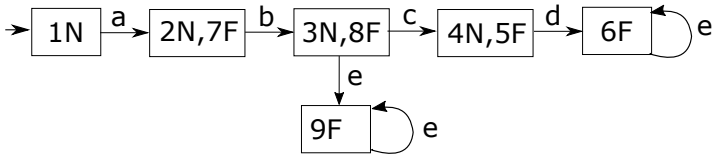
Figura 4.4 – Exemplo para ilustrar a análise das condições de prognosticabilidade de uma linguagem. (a) Autômato G_{22}; (b) Autômato s-diagnosticador G^s_{d22}; (c) Autômato c-diagnosticador G^c_{d22}.



(a)



(b)



(c)

Fonte: (Autor.)

Analisando o s-diagnosticador G^s_{d22} da Figura 4.4 (b), pode-se verificar que o estado normal $(2N)$ possui um sucessor imediato $(3N,8F)$ que não é normal, e que o estado normal $(4N)$ possui um sucessor imediato $(6F)$ que não é normal, portanto $F_D = \{(2N),(4N)\}$. Assim, de acordo com o Teorema 5, a linguagem L_{22} é prognosticável, pois todos os ciclos existentes nos estados alcançados a partir dos estados que pertencem a F_D são ciclos de estados certos em $(6F)$ e $(9F)$ no s-diagnosticador G^s_{d22}.

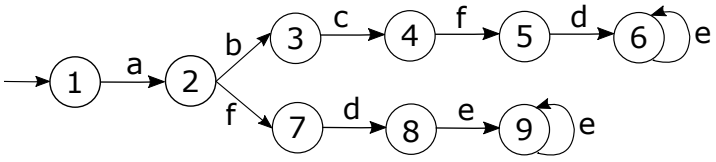
Analisando o c-diagnosticador G^c_{d22} da Figura 4.4 (c), pode-se verificar o estado normal $(1N)$ é o único que possui um sucessor imediato $(2N,7F)$ que não é normal, portanto

$F_D = \{(1N)\}$. Assim, de acordo com o Teorema 5, a linguagem L_{22} é prognosticável, pois todos os ciclos existentes nos estados alcançados a partir dos estados que pertencem a F_D, são ciclos de estados certos em $(6F)$ e $(9F)$ no c-diagnosticador G^c_{d22}.

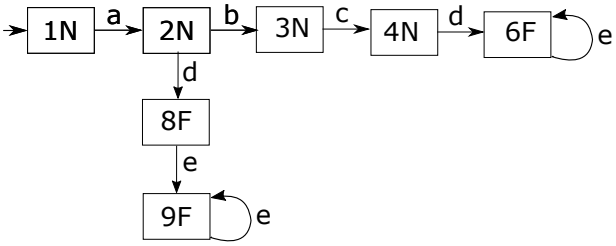
De acordo com Genc e Lafortune (2009), nem sempre é necessário analisar todos os estados normais do conjunto F_D, conforme é ilustrado no exemplo a seguir.

Exemplo 23 *Considere o autômato G_{23} mostrado na Figura 4.5 (a), cuja linguagem é dada por $L_{23} = \overline{a(bcfde^* + fdee^*)}$, sendo que $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,c,d,e\}$ e $\Sigma_f = \{f\}$.*

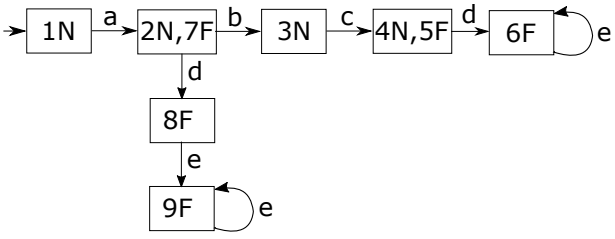
Figura 4.5 – Exemplo para ilustrar a análise das condições de prognosticabilidade de uma linguagem. (a) Autômato G_{23}; (b) Autômato s-diagnosticador G^s_{d23}; (c) Autômato c-diagnosticador G^c_{d23}.



(a)



(b)



(c)

Fonte: (Autor.)

Ao analisar o s-diagnosticador G^s_{d23} da Figura 4.5 (b), pode-se verificar que o estado normal $(2N)$ possui um sucessor imediato $(8F)$ que não é normal, e que o estado normal $(4N)$ também possui um sucessor imediato $(6F)$ que não é normal, portanto $F_D = \{(2N),(4N)\}$. A linguagem L_{23} é prognosticável, pois todos os ciclos que sucedem os estados que pertencem a F_D, são ciclos de estados certos em $(6F)$ e $(9F)$ no diagnosticador G^s_{d23}. De acordo com Genc e Lafortune (2009), nesse caso é preciso testar as condições de

prognosticabilidade nos estados $(2N)$ e $(4N)$, mas somente no estado $(2N)$, pois o estado $(4N)$ é alcançado a partir do estado $(2N)$.

Ao analisar o c-diagnosticador G^c_{d23} da Figura 4.5 (c), pode-se verificar que o estado normal $(1N)$ possui um sucessor imediato $(2N,7F)$ que não é normal, e que o estado normal $(3N)$ também possui um sucessor imediato $(4N,5F)$ que não é normal, portanto $F_D = \{(1N),(3N)\}$. A linguagem L_{23} é prognosticável, pois todos os ciclos que sucedem os estados que pertencem a F_D, são ciclos de estados certos em $(6F)$ e $(9F)$ no c-diagnosticador G^c_{d23}. De acordo com Genc e Lafortune (2009), não é preciso testar as condições de prognosticabilidade do estado $(3N)$, somente o estado $(1N)$, pois o estado $(3N)$ é alcançado a partir do estado $(1N)$.

Observação 6 *Pode-se observar que os exemplos até o momento ao utilizar o autômato s-diagnosticador ou c-diagnosticador para verificação da prognosticabilidade não houve problemas, ou seja, os resultados foram iguais. Porém, isso não ocorre em todos os casos. Portanto, a seguir são introduzidas novas condições para a prognosticabilidade estabelecidas sobre os c-diagnosticadores (WATANABE et al., 2017a).*

Para essas novas condições será necessária a introdução do conjunto $\mathcal{FU}$ *(First Uncertain).*

Seja $\mathcal{FU}$ o conjunto dos primeiros estados incertos no diagnosticador alcançados a partir do estado inicial, considerando todos os caminhos existentes. Formalmente, $\mathcal{FU} = \{q_d \in Q^U_d : (\exists s_o \in \Sigma^*_o)$ tal que $(\hat{\delta}(q_{d,0},s_o)=q_d)$ e $(\forall t_o < s_o)(\hat{\delta}(q_{d,0},t_o) \notin Q^U_d)\}$.

Teorema 6 *(Condições para Prognosticabilidade de uma Linguagem (WATANABE et al., 2017a)). Seja $G=(Q,\Sigma,\delta,q_0)$ um autômato que gera uma linguagem L prefixo-fechada e viva. Seja $G^c_d=(Q_d,\Sigma_o,\delta_d,q_{d,0})$ o c-diagnosticador construído a partir de G. As ocorrências do evento f são prognosticáveis em L com respeito à P_o se e somente se para todo $q_d \in \mathcal{FU}$, a condição $\mathscr{C}$ é satisfeita, sendo que $\mathscr{C}$: todos os ciclos em $A_c(G^c_d,q_d)$ são ciclos de estados certos no diagnosticador.*

A prova do Teorema 6 não é apresentada aqui uma vez que é análoga à prova do Teorema 8, a qual foi apresentada em Genc e Lafortune (2009). A diferença consiste na mudança no conjunto usado para a análise das condições (de F_D para $\mathcal{FU}$). Essa troca, reside no fato de, ao incluir o alcance não observável no cálculo do diagnosticador, aqueles estados normais que eram sucedidos por um estado certo de falha no s-diagnosticador passam a ser estados incertos seguidos de um estado certo no c-diagnosticador. Assim, a análise da prognosticabilidade no c-diagnosticador deve ser feita sobre os estados incertos,

o que foi feito com a mudança no conjunto $\mathcal{FU}$. Um estado incerto no c-diagnosticador corresponde a estados na planta que são separados por um evento de falha (possivelmente com outros eventos não-observáveis). Assim, como a prognose tem que se dar antes da falha, condições foram estabelecidas sobre os primeiros estados nos quais a falha é iminente, ou seja, os primeiros estados incertos.

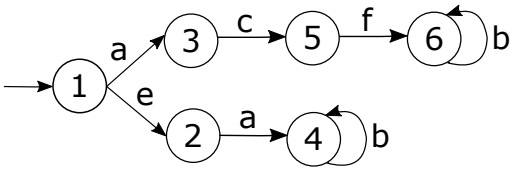
Em palavras, esse teorema estabelece que uma linguagem é prognosticável se e somente se todos os ciclos existentes nos estados alcançados a partir dos estados que pertencem a $\mathcal{FU}$, são ciclos de estados certos no c-diagnosticador G_d^c.

Observação 7 *Assim como no caso da prognosticabilidade de linguagens, a análise da prognosticabilidade de uma cadeia poderia ser feita sobre o diagnosticador G_d^c. Entretanto, adotou-se o c-diagnosticador seguro com alcance a eventos não-observáveis G_{sd}^c para fins de padronização. Além disso, para padronização, deste ponto será adotado o diagnosticador seguro G_{sd}^c para a análise de todas as modalidades (diagnose segura, prognose e controlabilidade segura pela diagnose e pela prognose).*

A seguir, apresenta-se um exemplo para ilustrar a condição da prognosticabilidade de uma linguagem sob o amparo do Teorema 6.

Exemplo 24 *Considere o autômato G_{24} mostrado na Figura 4.6, cuja linguagem é dada por $L_{24} = \overline{acfb^* + eab^*}$, sendo que $\Sigma_{uo} = \{e, f\}$, $\Sigma_o = \{a, b, c\}$ e $\Sigma_f = \{f\}$.*

Figura 4.6 – Exemplo para ilustrar a análise da condição de prognosticabilidade de uma linguagem (Exemplo G_1 da Genc e Lafortune (2009)). Autômato G_{24};

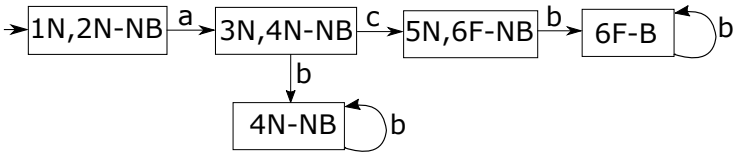


(a)

Fonte: (Autor.)

Analisando o c-diagnosticador seguro G_{sd24}^c da Figura 4.7, considerando $\phi = \{b\}$, pode-se verificar que $\mathcal{FU} = \{(5N, 6F - NB)\}$. Pelo Teorema 6, a linguagem L_{24} é prognosticável, uma vez que a partir do estado $(5N, 6F - NB)$ o único ciclo de estados alcançados é o estado certo de falha em $(6F - B)$.

Figura 4.7 – Exemplo para ilustrar a análise da condição de prognosticabilidade de uma linguagem. Autômato c-diagnosticador Seguro G^c_{sd24}.

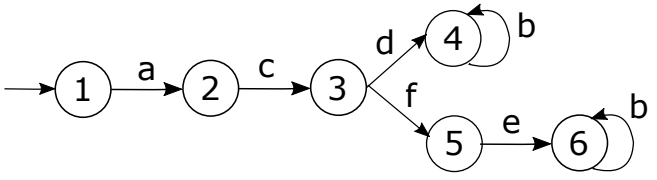


Fonte: (Autor.)

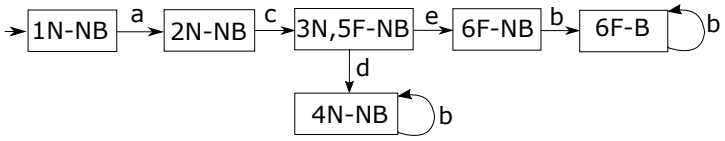
A seguir, apresenta-se um exemplo para ilustrar uma linguagem não-prognosticável.

Exemplo 25 *Considere o Autômato G_{25} mostrado na Figura 4.8 (a), cuja linguagem é dada por $L_{25} = \overline{ac(db^* + feb^*)}$, sendo que $\Sigma_{uo} = \{f\}$, $\phi = \{b\}$, $\Sigma_o = \{a,b,c,d,e\}$ e $\Sigma_f = \{f\}$.*

Figura 4.8 – Exemplo para ilustrar a análise da condição de prognosticabilidade de uma linguagem. (a) Autômato G_{25}; (b) Autômato c-diagnosticador seguro G^c_{sd25}.



(a)



(b)

Fonte: (Autor.)

Analisando o c-diagnosticador G^c_{sd25} da Figura 4.8 (b), conclui-se que $\mathcal{FU} = \{(3N, 5F - NB)\}$, e a partir do estado $(3N, 5F - NB) = \mathcal{FU}$ alcança-se o estado $(4N - NB)$ que possui um ciclo de estado normal. Assim, pelo Teorema 6, pode-se concluir que L_{25} não é prognosticável.

Na próxima seção são introduzidas condições sob as quais se pode garantir que a ocorrência do evento f na cadeia $s \in \Psi_L(f)$ é prognosticável.

4.8 CONDIÇÕES PARA PROGNOSTICABILIDADE DE UMA CADEIA

As prognosticabilidade de uma cadeia $s \in \Psi_L(f)$ será uma condição para controlabilidade segura de uma cadeia pela prognose. Para essa nova condição será necessária a introdução da função $FU(s)$.

Seja $FU(s)$ uma função que mapeia uma cadeia $s \in \Psi_L(f)$ no primeiro estado incerto no G^c_{sd} alcançado a partir do estado inicial $q_{sd,0}$ por uma cadeia $s_o = P_o(s)$. Formalmente, $FU(s) = q_{sd} \in Q^U_{sd} : [q_{sd} = \hat{\delta}_{sd}(q_{sd,0}, P_o(s))]$.

Proposição 5 *(Condições para Prognosticabilidade de uma Cadeia).* Considere uma linguagem diagnosticável L e um autômato $G = (Q, \Sigma, \delta, q_0)$ que gera L. Seja $G^c_{sd} = (Q_{sd}, \Sigma_o, \delta_{sd}, q_{sd,0})$ o c-diagnosticador construído a partir de G. A ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é prognosticável em relação a P_o se e somente se a condição $\mathscr{C}$ é satisfeita para o estado $q_{sd} = FU(s)$, sendo $\mathscr{C}$: todos os ciclos em $A_c(G^c_{sd}, q_{sd})$ são ciclos de estados certos no diagnosticador seguro.

Prova. A prova é em duas partes:
($\Rightarrow$) Primeiro, é provado por contradição que a ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é prognosticável em relação a P_o somente se a condição $\mathscr{C}$ é satisfeita para o estado $q_{sd} = FU(s)$. Suponha que a ocorrência de f numa cadeia $s \in \Psi_L(f)$ é prognosticável, mas a condição $\mathscr{C}$ não é satisfeita para $q_{sd} = FU(s)$. Assim, $A_c(G^c_{sd}, q_{sd})$ contém um ciclo que não é composto exclusivamente por estados certos no diagnosticador. Considere que este ciclo é formado por $\{q_{sd,1}, ..., q_{sd,m}\}$ e $\sigma_{o,1}...\sigma_{o,m} \in \Sigma^*_o$, sendo que $q_{sd,i} \notin Q^C_{sd}$ para algum $i \in \{1, 2, ..., m\}$, sendo $m \in \mathbb{N}$. Considere que um estado deste ciclo é alcançado de $q_{sd} = FU(s)$ por uma cadeia $v_o \in \Sigma^*_o$. Sem perda de generalidade, suponha que $\hat{\delta}_{sd}(q_{sd}, v_o) = q_{sd,1}$. Conforme a definição, $q_{sd} = FU(s) = \hat{\delta}_{sd}(q_{sd,0}, P_o(s))$, então $q_{sd,1} = \hat{\delta}_{sd}(q_{sd,0}, s_o v_o)$, sendo que $s_o = P_o(s)$ e $v_o \in \Sigma^*_o$. De acordo com o Lema 5, apresentado em Genc e Lafortune (2009), se num ciclo de estados no diagnosticador existe um estado $q_{sd,i}$ que não é um estado certo de falha, então nenhum dos outros estados no ciclo são certos. Assim, $q_{sd,i} \notin Q^C_{sd}$ para todo $i \in \{1, 2, ..., m\}$. De acordo com o Lema 6 em Genc e Lafortune (2009), se existe um ciclo no G^c_{sd} que é composto pelos estados normais ou incertos, então existe um ciclo correspondente em G tal que todos seus estados possuem rótulos com N no ciclo do diagnosticador G^c_{sd}. Suponha que o ciclo em G é formado por $\{q_1, ..., q_m\}$ e $y_1...y_m \in \Sigma^*$, tal que $(q_i, N) \in q_{sd,i}$ e $P_o(y_i) = \sigma_{o,i}$ para $i = 1, 2, ..., m$. Assim, $q_1 = \hat{\delta}(q_1, (y_1...y_m)^k)$ para $k \in \mathbb{N}$ e $f \notin y_1...y_m$. Além disso, pode-se afirmar que $q_1 = \hat{\delta}(q_0, t'v')$ para $t', v' \in \Sigma^*$, tal que $f \notin t'v', P_o(t') = s_o$ e $P_o(v') = v_o$. Assim, como $q_1 = \hat{\delta}(q_1, (y_1...y_m)^k)$ para $k \in \mathbb{N}$ e $f \notin y_1...y_m$, pode-se concluir que $\exists t' \in L$ e $\exists u' = v'(y_1...y_m)^k \in L/t'$ com $P_o(t') = P_o(s)$ e $f \notin t'$ para o qual $f \notin u' = v'(y_1...y_m)^k$ para $k \in \mathbb{N}$, o qual viola a condição $\mathscr{P}$ para t'. Finalmente, sabe-se que se a condição $\mathscr{P}$ é violada para t', com $P_o(t') = P_o(s)$, então também é violada para qualquer $t \in \bar{s}$. Assim, $\forall t \in \bar{s}$ com $f \notin t$, a condição $\mathscr{P}$ é violada, e portanto, a ocorrência de f em $s \in \Psi_L(f)$ não é prognosticável, contrariando a hipótese inicial.

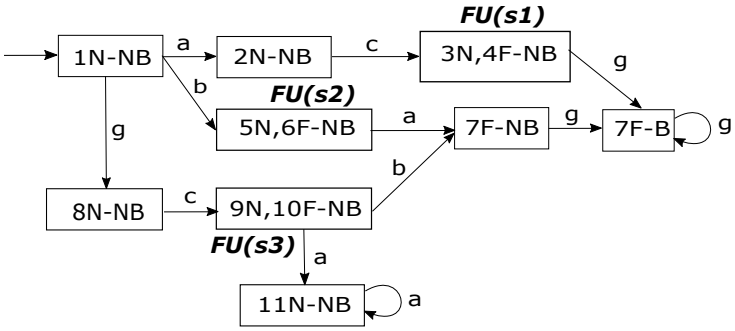
($\Leftarrow$) Agora, prova-se que se a condição $\mathscr{C}$ é satisfeita para o estado $q_{sd} = FU(s)$, então a ocorrência de f em $s \in \Psi_L(f)$ é prognosticável em relação a P_o. A cadeia s pode ser escrita como $s = tf$, tal que $t \in \Sigma^*$ e $f \notin t$. Assim, $t \in \overline{s}$ e $P_o(t) = P_o(s)$. Considere qualquer cadeia $t' \in L$ tal que $f \notin t'$ e $P_o(t') = P_o(s)$. Suponha que $q = \hat{\delta}(q_0, t)$ e $q' = \hat{\delta}(q_0, t')$. Uma vez que $P_o(t) = P_o(t') = P_o(s)$, sabe-se que $(q, N), (q', N) \in q_{sd} = FU(s)$. Se a condição $\mathscr{C}$ é satisfeita para $q_{sd} = FU(s)$, então todos os ciclos em $A_c(G^c_{sd}, q_{sd})$ são ciclos de estados certos no diagnosticador, o que significa que todas as continuações suficientemente longas em L de cadeias com a mesma projeção de t contêm f. Assim, $\exists t \in \overline{s}$, com $f \notin t$ e $(\forall t' \in L)(\forall u' \in L/t')[(P_o(t') = P_o(t)) \wedge (f \notin t') \wedge (\exists n \in \mathbb{N})(\|u'\| \geq n) \Rightarrow (f \in u')]$. Portanto, a condição $\mathscr{P}$ é satisfeita e assim a ocorrência do evento f em $s \in \Psi_L(f)$ é prognosticável em relação a P_o. □

Em palavras, a ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é prognosticável se e somente se todos os estados alcançados a partir do estado $q_{sd} = FU(s)$ possui somente ciclos de estados certos.

Observação 8 . *Lembrando que o c-diagnosticador G^c_{sd} não precisaria ser utilizado para verificar condições de prognosticabilidade de cadeias, bastaria o c-diagnosticador G^c_d, porém por motivos de padronização, foi adotado o mesmo.*

O **Exemplo** 21 da Figura 4.3 foi retomado para ilustrar a condição da cadeia prognosticável. O c-diagnosticador seguro G^c_{sd21} para este SED é mostrado na Figura 4.9.

Figura 4.9 – Exemplo para ilustrar a análise da condição de prognosticabilidade de uma cadeia. Autômato c-diagnosticador G^c_{sd21} ilustrando $FU(s)$.



Fonte: (Autor.)

Conforme discutido anteriormente, existem 3 cadeias $s_1, s_2, s_3 \in \Psi_{L_{21}}(f)$ em L_{21}, ou seja, $s_1 = acf$, $s_2 = bf$ e $s_3 = gcf$. Considerando $\Phi = \{g\}$, para as cadeias s_1, s_2 e s_3, tem-se $FU(s_1) = (3N, 4F - NB), FU(s_2) = (5N, 6F - NB)$ e $FU(s_3) = (9N, 10F - NB)$. As cadeias s_1 e s_2 são prognosticáveis, pois os estados alcançados pelos estados $(3N, 4F - NB)$

e $(5N, 6F-NB)$, respectivamente, têm somente ciclos de estados certos, satisfazendo as condições estabelecidas na Proposição 5. Entretanto, a cadeia s_3 não é prognosticável, pois o estado $(11N-NB)$ alcançado a partir do estado $(9N, 10F-NB) = FU(s_3)$ tem um ciclo de estado normal, violando a condição estabelecida na Proposição 5.

A seguir é introduzida um teorema que apresenta condições para linguagem prognosticável através da abordagem por cadeias.

Teorema 7 *(Condições para Prognosticabilidade de uma linguagem na abordagem por cadeias). Considere uma linguagem diagnosticável L e um autômato $G = (Q, \Sigma, \delta, q_0)$ que gera L. Seja $G^c_{sd} = (Q_{sd}, \Sigma_o, \delta_{sd}, q_{sd,0})$ o c-diagnosticador seguro construído a partir de G. A linguagem L é prognosticável se e somente se para toda cadeia $s \in \Psi_L(f)$, condição $\mathscr{C}$ é satisfeita para o estado $q_{sd} = FU(s)$, sendo $\mathscr{C}$: todos os ciclos em $A_c(G^c_{sd}, q_{sd})$ são ciclos de estados certos no c-diagnosticador seguro.*

Esse teorema substitui o Teorema 6 e a sua prova é similar a da Proposição 5.

4.9 CONSIDERAÇÕES FINAIS

Neste capítulo foram revisados o problema da prognose de falhas em SEDs em linguagens. Então, foi introduzido um novo conceito de prognose de falhas no contexto de cadeias. Foram apresentadas condições necessárias e suficientes para prognosticabilidade em linguagem e foram introduzidas condições e suficientes para cadeias. Nesta seção foram analisadas as condições estabelecidas no trabalho da Genc e Lafortune (2009) e foram apresentadas novas condições para a prognosticabilidade estabelecidas sobre c-diagnosticadores em linguagem (WATANABE et al., 2017a) e cadeias. Essas novas condições foram necessárias neste trabalho, pois optou-se por adotar o c-diagnosticador como sendo o padrão para a análise de todas as propriedades (diagnose, diagnose segura e prognose). Foram apresentados exemplos para cada definição e proposição para fins didáticos.

Foram apresentados exemplos para mostrar a necessidade de se utilizar o diagnosticador com alcance não-observável para a análise da prognosticabilidade. Portanto, foi proposto o Teorema 6 para definir novas condições para prognosticabilidade em função do conjunto $\mathcal{FU}$ ao invés do F_D. A Tabela 4.1 apresenta um resumo das condições para prognosticabilidade em linguagem utilizando s-diagnosticador e c-diagnosticador.

A contribuição mais relevante deste capítulo foi a introdução do conceito da prognosticabilidade numa cadeia e o estabelecimento de condição necessária e suficiente para

Tabela 4.1 – Comparativo entre condições para a prognose em linguagem estabelecidas a partir dos diferentes diagnosticadores.

Mecanismo (Linguagem)	Tipo do diagnosticador	Condição	Obra
Prognose	s-diagnosticador	Se, e somente se, todos os ciclos existentes nos estados alcançados a partir dos estados, que pertencem a F_D, são ciclos de estados certos de falha.	(GENC; LAFORTUNE, 2006, 2009)
	c-diagnosticador	Se, e somente se, todos os ciclos existentes nos estados alcançados a partir dos estados que pertencem a $\mathcal{FU}$, são ciclos de estados certos de falha.	(WATANABE et al., 2017a)

Fonte: (Autor.)

assegurar que a ocorrência do evento f na cadeia $s \in \Psi_L(f)$ seja prognosticável. Esse conceito será utilizado no Capítulo 5 para tratar da Controlabilidade Segura de SEDs.

A seguir, a Tabela 4.2 apresenta condições para a prognose em cadeia a partir do c-diagnosticador seguro.

Tabela 4.2 – Condições para a prognose em cadeia estabelecidas a partir do c-diagnosticador seguro.

Conceito	Tipo do diagnosticador	Condição	Obra
Prognose em cadeia	c-diagnosticador seguro	Se e somente se a condição $\mathscr{C}$ é satisfeita para o estado $q_{sd} = FU(s)$, sendo $\mathscr{C}$: todos os ciclos em $A_c(G^c_{sd}, q_{sd})$ são ciclos de estados certos no diagnosticador seguro.	Este trabalho

Fonte: (Autor.)

5 CONTROLABILIDADE SEGURA EM SEDS

Conforme Paoli, Sartini e Lafortune (2011), um sistema supervisório tolerante a falhas apresenta os seguintes objetivos: diagnosticar a ocorrência do evento de falha f antes do sistema executar uma sequência ilegal que possa criar danos ao sistema; forçar o sistema a parar a evolução antes de executar as sequências proibidas; e mudar a direção para uma nova especificação pós-falha, que pode levar a um comportamento degradado, mas funcional e seguro. Para alcançar tais objetivos, os autores introduziram o conceito de controlabilidade segura no contexto de linguagens regulares. Basicamente, controlabilidade segura representa a capacidade de impedir que o sistema realize ações indesejadas (ou proibidas). De acordo com Paoli, Sartini e Lafortune (2011), um SED é controlável seguro (pela diagnose) se sua linguagem é diagnosticável segura e após a diagnose da falha existe um evento controlável que possa ser desabilitado para impedir que o sistema execute uma ação proibida. Uma vez que essa noção é baseada na diagnose de falhas, neste trabalho nós denotamos como "controlabilidade segura pela diagnose".

Baseada nessa noção, em um trabalho anterior (WATANABE et al., 2017a) foi introduzido o conceito de controlabilidade segura pela prognose. Nesse, caso, ao invés de usar diagnose de falhas, foi empregada a prognose de falhas para fins de controlabilidade segura. Basicamente, uma linguagem é controlável segura pela prognose se toda a ocorrência de falha é prognosticável e após a prognose da falha existe um evento controlável que pode ser desabilitado para impedir a execução das cadeias ilegais.

Na literatura, ambas as definições de controlabilidade segura (pela diagnose ou pela prognose) têm em comum o requisito que todas as ocorrências de falhas na linguagem têm que ser controláveis pelo mesmo mecanismo de detecção de falha, ou seja, usando somente diagnose (PAOLI; SARTINI; LAFORTUNE, 2011) ou usando somente prognose (WATANABE et al., 2017a).

Neste capítulo apresenta-se uma nova abordagem de controlabilidade segura que combina a diagnose e a prognose de falhas online. Para alcançar este objetivo foram introduzidas as noções de diagnosticabilidade, diagnosticabilidade segura, prognosticabilidade, controlabilidade segura pela diagnose e controlabilidade segura pela prognose sobre cadeias. Foram introduzidas também necessárias e suficientes condições para garantir essas propriedades.

A seguir, apresentam-se as principais contribuições deste capítulo:
i) Pequena correção na definição de controlabilidade segura apresentada por Paoli, Sartini e Lafortune (2011) e explicitação de que a mesma se dá no contexto da diagnose;

ii) Introdução da definição de controlabilidade segura de uma cadeia pela diagnose;
iii) Estabelecimento de condições para a controlabilidade segura de uma cadeia pela diagnose;
iv) Introdução da definição de controlabilidade segura numa linguagem pela prognose;
v) Introdução da definição de controlabilidade segura numa cadeia pela prognose;
vi) Estabelecimento de condições para a controlabilidade segura numa linguagem pela prognose;
vii) Estabelecimento de condições para a controlabilidade segura numa cadeia pela prognose;
viii) Generalização da definição de controlabilidade segura, a qual engloba os conceitos de controlabilidade segura pela diagnose e controlabilidade segura pela prognose;
ix) Estabelecimento de condições necessárias e suficientes para a controlabilidade segura definida anteriormente.

A seguir será apresentada a organização deste capítulo.

Tendo em vista que este capítulo trata da controlabilidade de SEDs, na seção 5.1 são revisados conceitos da Teoria de Controle Supervisório de SEDs. Na seção 5.2 é apresentada a definição de controlabilidade segura. Ainda nessa seção, é feita uma pequena alteração na definição original de controlabilidade segura apresentada por Paoli, Sartini e Lafortune (2011) e a nomenclatura é alterada para controlabilidade segura pela diagnose. Na seção 5.3 é introduzida a definição de controlabilidade segura de uma cadeia pela diagnose. Condições de controlabilidade segura de uma linguagem pela diagnose estão apresentadas na seção 5.4. Na seção 5.5 são introduzidas condições de controlabilidade segura numa cadeia pela diagnose. Na seção 5.6 é introduzida a definição de controlabilidade segura de uma linguagem pela prognose. A definição de controlabilidade segura de uma cadeia pela prognose é introduzida na seção 5.7. Nas seções 5.8 e 5.9 são introduzidas condições de controlabilidade segura de uma linguagem pela prognose e condições de controlabilidade segura de uma cadeia pela prognose, respectivamente. Na seção 5.10 apresenta-se uma generalização da definição de controlabilidade segura de uma linguagem, a qual engloba os conceitos de controlabilidade segura de uma cadeia pela diagnose ou pela prognose. Além disso, nesta seção são estabelecidas condições necessárias e suficientes para a controlabilidade segura definida anteriormente. Na seção 5.11 é apresentada uma discussão sobre o uso da controlabilidade segura para fins de Controle Tolerante a Falhas (CTF). Finalmente, na seção 5.12 são apresentadas as considerações finais.

5.1 TEORIA DE CONTROLE SUPERVISÓRIO DE SEDS

A Teoria do Controle Supervisório (TCS) (RAMADGE; WONHAN, 1987b) (RAMADGE; WONHAN, 1989) estabelece um método formal para cálculo de supervisores ótimos para SEDs. Esses supervisores vão atuar sobre o sistema de forma minimamente restritiva, mas sem permitir comportamentos indesejados, que poderiam causar eventuais danos operacionais ao processo produtivo. Essa teoria apoia-se, essencialmente, na Teoria dos Autômatos e Linguagens, sendo aplicável a toda classe dos SEDs.

O controle supervisório sem falhas parte do modelo do sistema não-controlável denominado G e um conjunto de especificações. Em geral, o G é expresso como uma composição paralela dos vários componentes do sistema chamado de $G_1, ..., G_n$. O comportamento de G é descrito pela linguagem $L(G)$ e deve ser restrito pelo controle para satisfazer o conjunto de especificações. Com este objetivo, é projetado o supervisor S. Portanto, obtém-se o sistema controlado $G_{sup} = G||S$ com sua linguagem associada $L(G_{sup})$, satisfazendo o conjunto de especificações da linguagem $\mathscr{K}$. O Supervisor interage com a planta de forma a fechar a malha de controle.

5.1.1 Controle Supervisório de SED com Falhas

Ao considerar a ocorrência de falha num sistema, seu modelo nominal G passará a ser denotado por G^{n+f}. Assim, G^{n+f} representa tanto o comportamento nominal quanto o comportamento faltoso do sistema não controlado. que $L(G^{n+f}) \supset L(G)$ com correspondentes conjunto de eventos $\Sigma^{n+f} = \Sigma \cup \{f\}$, sendo $f \in \Sigma_{uo}^{n+f} \cap \Sigma_{uc}^{n+f}$, ou seja, f é não-observável e não-controlável. Se um dado supervisor S foi projetado sem levar em conta a ocorrência de falhas, ou seja, com base no modelo G, então a sua atuação sobre um sistema com falhas G^{n+f} será dado por $G_{sup}^{n+f} = G^{n+f}||S$. A estrutura de G^{n+f} contém a parte nominal e o conjunto de falhas.

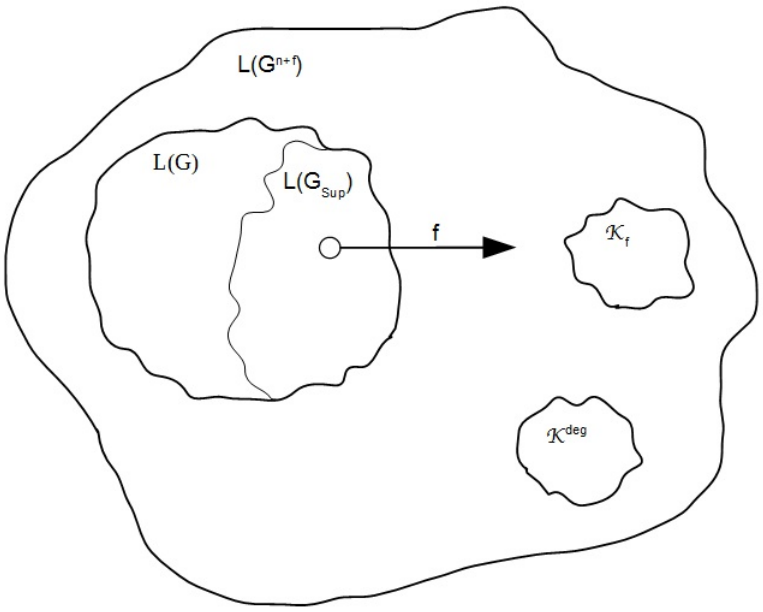
Pela construção de S, não há ações indesejáveis na parte nominal. Entretanto, sequências de ações indesejáveis podem surgir nos modelos pós-falhas devido a ação do supervisor nominal nos componentes de falhas, conforme obtido em G_{sup}^{n+f}. Portanto, deve-se impedir que falhas locais se desenvolvam em falhas que possam causar riscos de segurança. Esta condição é estritamente ligada a definição de diagnosticabilidade segura, porque se houver uma cadeia do conjunto Φ, sendo Φ o conjunto finito de cadeias proibidas após a falha, então é necessário detectar a falha antes da cadeia proibida ser executada, conforme Paoli e Lafortune (2005). Consequentemente, deve-se proibir que após a falha f, o sistema execute uma subcadeia proibida de um dado conjunto finito Φ, sendo $\Phi \subseteq \Sigma^*$.

Essencialmente, os elementos do conjunto Φ vêm a ser ilegais após a ocorrência da falha f. A linguagem do sistema G^{n+f} é dividida em duas partes: a parte nominal (correspondente a G) e a parte de falha. A parte de falha inclui a linguagem ilegal $\mathscr{K}_f$, a qual possui todas as possíveis continuações após o evento f que contêm uma cadeia proibida do conjunto Φ como sub-cadeia. Então, o comportamento do sistema nominal $L(G)$ será de acordo com a especificação $\mathscr{K}$ sob a supervisão de S, porém adicionando falhas que podem incluir cadeias que estão na linguagem ilegal pós-falha.

Pode-se portanto, segundo Paoli, Sartini e Lafortune (2011), designar os objetivos do sistema supervisório tolerante a falha conforme segue:
(A) Diagnose da ocorrência do evento f antes do sistema executar alguma sequência ilegal do conjunto Φ;
(B) Forçar o sistema parar sua evolução antes da execução da sequência proibida;
(C) garantir que o comportamento do sistema com falha atenda uma nova (eventualmente degradada) especificação de controle pós-falha, que é expressa por $\mathscr{K}^{deg}$.

A Figura 5.1 ilustra o cenário descrito sob o ponto de vista de uma especificação de uma linguagem. Para melhor compreensão foi ilustrado que o $\mathscr{K}$ é controlável e observável em relação a G, sob a hipótese de que $\mathscr{K} = L(G_{sup})$. Além disso, com abuso de representação gráfica, por uma questão de clareza, as partes de falhas isoladas representam cadeias de pós-falha que não estão totalmente contidas no modelo nominal supervisionado.

Figura 5.1 – Especificações de tolerância a falhas para um SED supervisionado.



Fonte: Adaptado de (PAOLI; SARTINI; LAFORTUNE, 2011).

Paoli, Sartini e Lafortune (2011) afirmam que o objetivo (A) é alcançado se a propriedade de diagnosticabilidade segura é satisfeita pelo sistema. Na sequência, o objetivo (B) é estudado em termos de controlabilidade segura pela diagnose, e o objetivo (C) é ligado a tolerância de falha ativa. É importante enfatizar que as especificações pós-falha $\mathscr{K}^{deg}$ são em geral diferentes de $L(G_{sup}^{n+f})$; portanto para satisfazê-lo, é necessário chavear de um supervisor nominal S para um novo supervisor chamado S^{deg}, seguindo a abordagem de tolerância a falha de Blanke M. Kinnaert e Staroswiecki (2003).

Este trabalho tomou como base os objetivos traçados por Paoli, Sartini e Lafortune (2011), porém, além de diagnosticar, prognosticar o evento f antes de executar alguma sequência ilegal, tendo como foco principal apresentar uma nova abordagem de controlabilidade segura numa linguagem usando diagnose e prognose.

5.2 CONTROLABILIDADE SEGURA DE SEDS

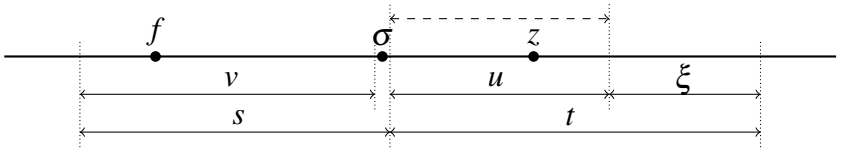
Paoli, Sartini e Lafortune (2011) introduziram a definição de SED controlável seguro, a qual, nas palavras dos autores, está relacionada com a capacidade de, após a ocorrência de uma falha, manter o sistema longe de regiões proibidas. Assim, os comportamentos ditos inseguros são evitados via ação de controle após a diagnose da falha. Portanto, a diagnose segura é uma condição necessária para a controlabilidade segura. Para apresentar a definição de SED controlável seguro de Paoli, Sartini e Lafortune (2011)(Seção 3 - Definição 3), considere que o conjunto de eventos Σ de uma linguagem L é particionado como $\Sigma = \Sigma_c \dot{\cup} \Sigma_{uc}$, sendo que Σ_c representa os eventos controláveis e Σ_{uc} o conjunto de eventos não-controláveis. O conjunto de eventos controláveis é aquele que pode ser controlado como comando de desligar uma bomba, e os eventos não-controláveis são os que não podem ser controlados, como por exemplo uma falha de sistema.

Definição 8 *(SED Controlável Seguro (PAOLI; SARTINI; LAFORTUNE, 2011)).* Uma linguagem L prefixo-fechada que é viva e que não contém ciclos de eventos não-observáveis é dita ser controlável segura em relação à projeção P_o, evento f e linguagem proibida $\mathscr{K}_f$ se atender às seguintes condições:
(i) Condição de diagnosticabilidade segura: L é diagnosticável segura em relação à P_o, f e $\mathscr{K}_f$.
(ii) Condição de controlabilidade segura: Considere uma cadeia qualquer $s \in L$ tal que $f \in s$ e $s = v\sigma$, com $\sigma \in \Sigma_o$. Suponha que a condição de diagnosticabilidade $\mathscr{D}$ não é atendida para a cadeia v, mas é atendida para a cadeia s. Então, $(\forall t \in L/s)$ tal que $t = u\xi$ com $\xi \in \Phi, \exists z \in \Sigma_c$ tal que $z \in u$.

Em palavras, uma linguagem é controlável segura se para qualquer cadeia que contém uma falha f e uma cadeia proibida ou ilegal ξ, existir (i) um evento observável σ que assegure a detecção de falha antes do sistema executar uma subcadeia proibida, e (ii) um evento controlável z após o evento observável σ, mas antes de uma subcadeia proibida $\xi \in \Phi$. Portanto, o evento controlável z poderá estar em qualquer região da subcadeia u, conforme ilustrado através de uma linha tracejada na Figura 5.2. Desta maneira, após a detecção da falha, é sempre possível desabilitar o evento controlável z e evitar o comportamento proibido.

Observação 9 *As letras representadas nos próximos gráficos são referentes às respectivas definições.*

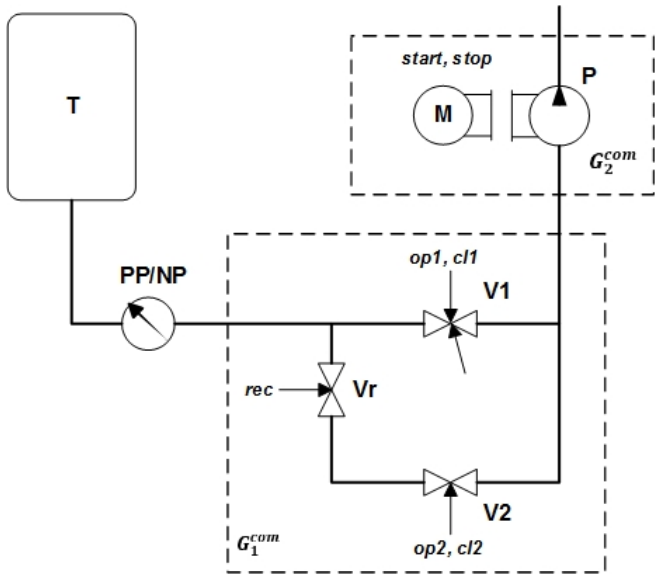
Figura 5.2 – Gráfico para ilustrar a controlabilidade segura segundo Paoli, Sartini e Lafortune (2011).



Fonte: (Autor.)

Em Paoli, Sartini e Lafortune (2011), é apresentado um exemplo ilustrativo de um sistema hidráulico composto por um tanque T, uma bomba P e um conjunto de válvulas ($V1$,$V2$ e Vr) e tubulações associadas, conforme Figura 5.3.

Figura 5.3 – Exemplo da planta do sistema hidráulico.



Fonte: Adaptado de (PAOLI; SARTINI; LAFORTUNE, 2011).

Tabela 5.1 – Mapeamento de sensores da planta do sistema hidráulico.

Nome do estado	Leitura do sensor
x1N,x5F	NP
x2N	NP
x4N	NP
x3N	PP
x6F	PP
x8F	PP
x7F	PP
x5N	PP

Fonte: (Autor.)

A bomba P, através dos comandos de liga (*start*) e desliga (*stop*), permite o fluido mover do tanque à tubulação. Seu funcionamento deve ser sincronizado com o conjunto de válvulas redundantes. Admite-se que apenas a válvula $V1$ pode falhar e somente quando esta estiver fechada, ou seja, a falha consiste no travamento desta válvula fechada, a qual passa a não responder ao comando de abertura. O sistema é equipado por um sensor de pressão que é utilizado para sinalizar se há ou não sobrepressão. Se a válvula $V1$ é fechada ($cl1$), o sensor lê sobrepressão na tubulação (PP), e se é aberta ($op1$), o sensor não acusa sobrepressão (NP). Em funcionamento normal, a bomba P somente é ligada (*start*) após a válvula $V1$ ser aberta ($op1$). Se ocorrer dessa válvula ficar travada fechada ($cl1$) por alguma falha, não se deve permitir que a bomba P seja ligada por motivos de segurança. Com exceção do evento f, que é não-controlável e não-observável, todos os demais eventos são considerados observáveis e controláveis. Mais detalhes podem ser encontrados em Paoli, Sartini e Lafortune (2011).

Foi construído um mapeamento de sensores onde cada estado do diagnosticador é relacionado com o estado do sensor, conforme Tabela 5.1.

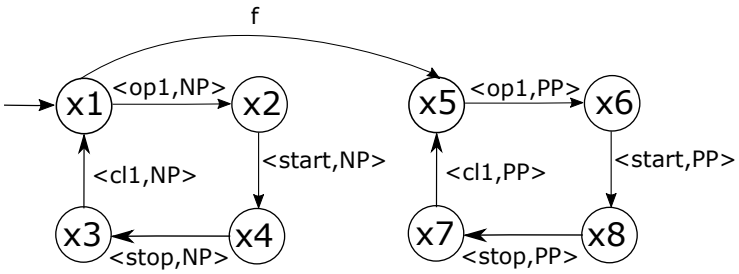
Mapeamento de sensores: O mapeamento de sensores é uma estratégia utilizada quando a falha não é diagnosticável a partir do modelo da planta. Por intermédio desse mapeamento, aumenta-se a quantidade de informação no modelo sem a necessidade de efetivamente modelar o referido sensor por intermédio de um autômato, e, com isso, a falha pode ser tornar diagnosticável. Basicamente consiste em, a partir de uma tabela que relaciona os estados dos sensores com cada estado do autômato da planta, acrescentar no diagnosticador as informações sobre a situação deste sensor sem que o mesmo seja modelado por um autômato. Maiores detalhes sobre mapeamento de sensores ver Cassandras e Lafortune (2008), página 106.

O autômato G_{sup}^{n+f} representado na Figura 5.4 apresenta as leituras do sensores juntamente com os eventos. Conforme descrito anteriormente, quando a válvula $V1$ é fechada,

o sensor lê sobrepressão (PP), e se é aberta, o sensor não acusa sobrepressão (NP). Portanto, a situação proibida que deve ser impedida é a bomba P estar trabalhando com a válvula $V1$ fechada.

A Figura 5.5 ilustra o diagnosticador do autômato G_{sup}^{n+f}. Nesse diagnosticador são mostrados os eventos de comandos de ligar (*start*) e desligar (*stop*) a bomba e de abrir (*op1*) e fechar (*cl1*) a válvula $V1$. Esses comandos estão relacionados com a leitura do sensor de pressão, ou seja sobrepressão (PP) ou não-sobrepressão (NP).

Figura 5.4 – Exemplo do sistema hidráulico. Autômato G_{sup}^{n+f}.



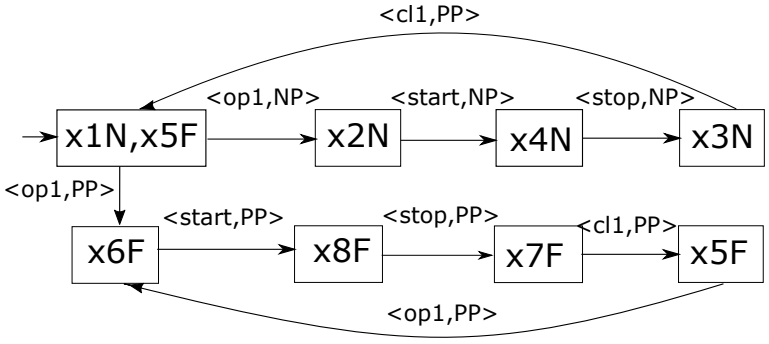
Fonte: Adaptado de (PAOLI; SARTINI; LAFORTUNE, 2011).

Se ocorrer uma falha f na válvula $V1$, travando-a fechada, a mesma não responderá ao comando de abrir e o sensor indicará sobrepressão PP. Por outro lado, se a válvula não apresentar falha, ao ser aberta haverá a indicação de não-sobrepressão NP. Desta forma, o diagnóstico da falha se dá pelo evento $< op1, PP >$. Conforme a Figura 5.5, o evento $< start, PP >$ leva ao estado proibido ($x8F$), sendo $\Phi = \{start\}$. Observe que o evento $< start, PP >$ pode ocorrer imediatamente, levando ao estado proibido, entretanto, como esse evento é controlável, o mesmo pode ser desabilitado pela ação de controle. O motivo de apresentar esse exemplo é mostrar que a condição de que o evento controlável tenha que existir antes do evento proibido pode ser relaxada, podendo ser o próprio evento proibido.

A rigor, de acordo com a Definição 8, o evento controlável z deve ocorrer antes da cadeia proibida $\xi \in \Phi$. Entretanto, de acordo com o exemplo do sistema hidráulico citado anteriormente, o evento controlável poderia ser um evento da própria cadeia proibida.

A Definição 9, introduzida por Watanabe et al. (2017a), apresenta uma releitura da Definição 8 de forma a contemplar a possibilidade de o evento controlável poder ser um elemento da própria cadeia proibida. Além disso, explicita-se nesta que a controlabilidade segura se dá a partir da diagnose. Tal informação é importante no contexto desta tese para permitir a diferenciação do conceito de controlabilidade segura pela prognose, que será introduzido no capítulo 4.

Figura 5.5 – Exemplo do sistema hidráulico. Autômato diagnosticador de G_{sup}^{n+f}.



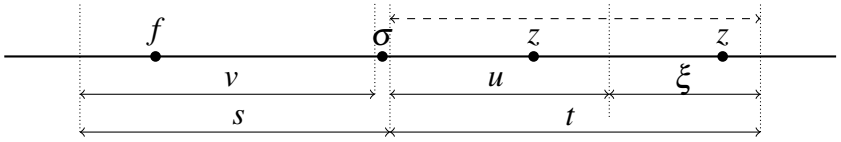
Fonte: Adaptado de (PAOLI; SARTINI; LAFORTUNE, 2011).

Definição 9 *(Linguagem Controlável Segura pela Diagnose (WATANABE et al., 2017a)).* Uma linguagem L prefixo-fechada que é viva e que não contém ciclos de eventos não-observáveis é dita ser controlável segura pela diagnose em relação à projeção P_o, evento f e linguagem proibida $\mathscr{K}_f$ se atender às seguintes condições:
($D9_1$) Condição de diagnosticabilidade segura: L é diagnosticável segura em relação à P_o, f e $\mathscr{K}_f$.
($D9_2$) Condição de controlabilidade segura: Considere uma cadeia qualquer $s \in L$ tal que $f \in s$ e $s = v\sigma$, com $\sigma \in \Sigma_o$. Suponha que a condição de diagnosticabilidade $\mathscr{D}$ não é atendida para a cadeia v, mas é atendida para a cadeia s. Então, $(\forall t \in L/s)$ tal que $t = u\xi$, com $\xi \in \Phi$, $\exists z \in \Sigma_c$ tal que $z \in t$.

Em palavras, uma linguagem é controlável segura pela diagnose se para qualquer cadeia que contém um evento f e uma cadeia proibida ou ilegal ξ, existir um evento observável σ que assegure a detecção de falha antes do sistema executar uma subcadeia proibida (condição ($D9_1$)) e um evento controlável z após o evento observável σ tal que z possa ser usado para impedir a execução de qualquer sequência ilegal (condição ($D9_2$)). Portanto, o evento controlável z pode estar antes do início da sequência ilegal (na subcadeia u) ou é um evento da própria cadeia proibida (na subcadeia $\xi \in \Phi$), conforme ilustrado através de uma linha tracejada na Figura 5.6. No caso particular em que a cadeia proibida é composta somente por um evento σ e a diagnose ocorra antes desse evento, então é suficiente que σ seja um evento controlável para se ter controlabilidade segura pela diagnose.

A propriedade da controlabilidade segura pela diagnose é ilustrada a seguir através de dois exemplos.

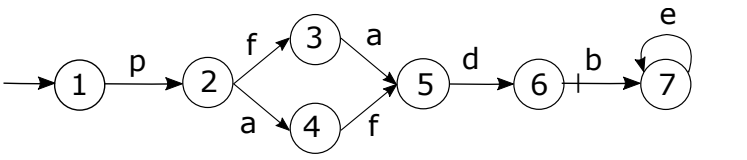
Figura 5.6 – Gráfico para ilustrar linguagem controlável mais abrangente.



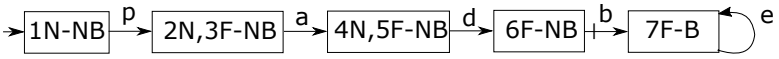
Fonte: (Autor.)

Exemplo 26 *Considere que o autômato G_{26} mostrado na Figura 5.7(a), cuja linguagem é dada por $L_{26} = \overline{p(fa+af)dbe^*}$, sendo $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,d,e,p\}$, $\Sigma_c = \{b\}$ e $\Sigma_f = \{f\}$. A cadeia proibida após a falha f é $\Phi = \{b\}$ e a linguagem ilegal é $\mathscr{K}_f = \{(adbe^* + dbe^*)\}$.*

Figura 5.7 – Exemplo de linguagem controlável segura pela diagnose. (a) Autômato G_{26}; (b) Autômato c-diagnosticador seguro G^c_{sd26}.



(a)



(b)

Fonte: (Autor.)

Em análise ao c-diagnosticador seguro G^c_{sd26} obtido a partir do autômato G_{26} apresentado na Figura 5.7(b), percebe-se que o único mau estado é o $(7F-B)$, que é um estado certo de falha. O estado $(6F-NB)$, que o antecede, é um estado certo de falha, estando portanto de acordo com a condição $(D9_1)$. Além disso, a linguagem é controlável segura pela diagnose, pois existe um evento controlável (evento b) após a diagnose da falha mesmo sendo o primeiro evento da cadeia ilegal $\xi = b$ (condição $(D9_2)$).

A seguir, é retomado o exemplo anterior (Exemplo 26) para ilustrar uma linguagem não-controlável segura pela diagnose. Porém, considere a retirada do evento b do conjunto de eventos controláveis. Note que com essa alteração, embora a linguagem seja diagnosticável segura, a mesma não é controlável segura pela diagnose, pois não existe um evento controlável após a diagnose (evento d) e antes do evento proibido (evento b).

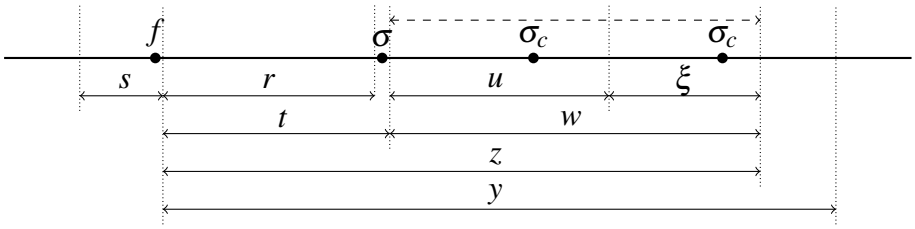
5.3 CONTROLABILIDADE SEGURA DE UMA CADEIA PELA DIAGNOSE

Baseado no conceito de controlabilidade segura de uma linguagem pela diagnose, é introduzido o conceito de cadeia controlável segura pela diagnose.

Definição 10 *(Cadeia Controlável Segura pela Diagnose).* Considere uma linguagem diagnosticável L. A ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é controlável segura pela diagnose em relação à P_o e $\mathscr{K}_f$ se as seguintes condições são atendidas:
($D10_1$) *Condição de Diagnosticabilidade Segura*: a ocorrência do evento f em $s \in \Psi_L(f)$ é diagnosticável segura em relação à P_o e $\mathscr{K}_f$.
($D10_2$) *Condição de Controlabilidade Segura pós-Diagnose*: Considere qualquer cadeia $t \in L/s$ tal que $t = r\sigma$, com $r \in \Sigma^*$ e $\sigma \in \Sigma_o$, e tal que a condição de diagnosticabilidade $\mathscr{D}$ não é satisfeita para sr, enquanto é satisfeita para st. Então, $\forall w \in L/st$ tal que $w = u\xi$, com $\xi \in \Phi$, $\exists \sigma_c \in \Sigma_c$ tal que $\sigma_c \in w$.

Em palavras, a ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é controlável segura pela diagnose se essa ocorrência é diagnosticável segura e existe um evento controlável σ_c para impedir a execução de qualquer evento ilegal após a falha na cadeia s. Assim, o evento controlável σ_c pode estar antes do início da sequência ilegal (na subcadeia u) ou é um evento da própria cadeia proibida (na subcadeia $\xi \in \Phi$), conforme ilustrada pela linha tracejada na Figura 5.8.

Figura 5.8 – Gráfico para ilustrar o conceito de cadeia controlável segura pela diagnose.

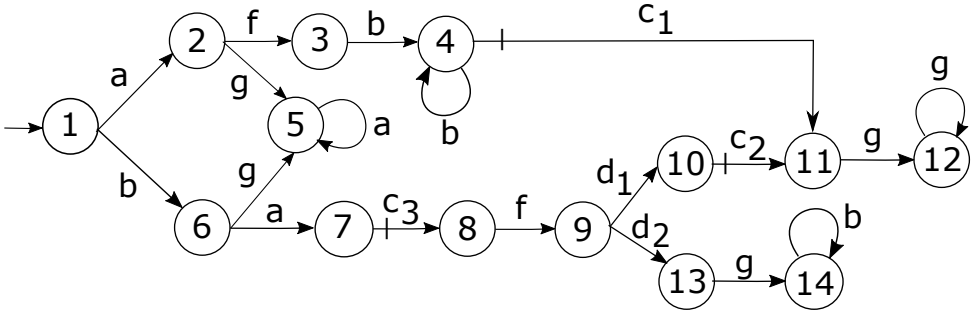


Fonte: (Autor.)

A seguir, apresenta-se um exemplo para ilustrar uma cadeia controlável segura pela diagnose e uma cadeia não-controlável segura pela diagnose.

Exemplo 27 *Considere o autômato G_{27} mostrado da Fig. 5.9, cuja linguagem é dada por $L_{27} = \overline{a(ga^* + fbb^*c_1gg^*) + b(ga^* + ac_3f(d_1c_2gg^* + d_2gb^*))}$, tal que $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,c_1,c_2,c_3,d_1,d_2,g\}$, $\Sigma_c = \{c_1,c_2,c_3\}$ e $\Sigma_f = \{f\}$. O conjunto de cadeias proibidas após a falha f é $\Phi = \{g\}$, e a linguagem ilegal é $\mathscr{K}_f = \{bb^*c_1gg^*, d_1c_2gg^*, d_2gb^*\}$.*

Figura 5.9 – Exemplo para ilustrar uma cadeia controlável segura pela diagnose e uma cadeia não-controlável segura pela diagnose. 43 Autômato G_{27}.



Fonte: (Autor.)

Existem duas cadeias $s \in \Psi_{L_{27}}(f)$, a saber, $s_1 = af$ e $s_2 = bac_3f$. A cadeia s_1 é diagnosticável segura, pois a condição $\mathscr{D}$ é satisfeita para $s_1t_{c_1} = afb$ $(t_{c_1} = b)$ e $\overline{t_{c_1}} \cap \mathscr{K}_f = \emptyset$, atendendo a condição $(D10_1)$. A cadeia s_1 é controlável segura pela diagnose, pois atende também a segunda condição, ou seja, existe o evento controlável c_1 que pode evitar a ocorrência do evento ilegal (condição $(D10_2)$). A cadeia s_2 também é diagnosticável segura, pois a condição $\mathscr{D}$ é satisfeita para $s_2t_{c_2} = bac_3fd_1$ $(t_{c_2} = d_1)$ e $\overline{t_{c_2}} \cap \mathscr{K}_f = \emptyset$ e satisfaz para $s_2t'_{c_2} = bac_3fd_2$ $(t'_{c_2} = d_2)$ e $\overline{t'_{c_2}} \cap \mathscr{K}_f = \emptyset$, atendendo a condição $(D10_1)$. Porém, a cadeia s_2 não é controlável segura pela diagnose, pois não possui um evento controlável após a evento (d_2) no estado (9), não sendo possível impedir a ocorrência do evento g no estado (13) (condição $(D10_2)$).

A partir da Definição 10, pode-se reescrever a definição de cadeia controlável segura pela diagnose, conforme segue.

Uma linguagem diagnosticável L é dita ser controlável segura pela diagnose em relação a projeção P_o, evento f e conjunto Φ se a ocorrência do evento f em todas as cadeias $s \in \Psi_L(f)$ é controlável segura pela diagnose.

5.4 CONDIÇÕES PARA CONTROLABILIDADE SEGURA DE UMA LINGUAGEM PELA DIAGNOSE

Antes de apresentar a proposição que estabelece condições necessárias e suficientes para a controlabilidade segura de uma linguagem pela diagnose, são apresentados alguns conceitos importantes usados em Paoli, Sartini e Lafortune (2011).

Seja $\mathcal{FC}$ o conjunto dos primeiros estados certos de falha alcançados no c-diagnosticador seguro G^c_{sd} obtido a partir de G, ou seja, $\mathcal{FC}$ é o conjunto de todos estados q

do diagnosticador seguro, tal que q é certo de falha e existe um estado q' no diagnosticador seguro que é incerto de falha, tal que q é alcançado a partir de q' através de um evento $\sigma_o \in E_o$. O conjunto $\mathcal{FC}$ tem um número finito de elementos: $\mathcal{FC} = \{q_i\}, (i = 1,...,m)$. Para cada $q_i = \{(q_j,F);(q_k,F),...,(q_l,F)\} \in \mathcal{FC}(i = 1,...,m)$ constrói-se um novo modelo não-controlável pós-diagnose de falha G_i^{deg}, tomando a parte acessível de G^{n+f} de todos estados distintos $q_j, q_k, ..., q_l$ de G^{n+f} que aparecem no i-ésimo estado do diagnosticador seguro. Denota-se por $\downarrow C$ a operação padrão relativa à operação de cálculo da ínfima superlinguagem prefixo-fechada e controlável de uma dada linguagem, e considerando um dado conjunto de eventos não-controláveis (CASSANDRAS; LAFORTUNE, 2008) (Seção 3.4, Pag. 155).

A Proposição 6, a seguir, apresenta condição necessária e suficiente para a controlabilidade segura de uma linguagem pela diagnose. Esta proposição é a mesma apresentada por Paoli, Sartini e Lafortune (2011), mas alterou-se o nome a fim de explicitar que se trata da controlabilidade pela diagnose da falha. A prova desta proposição se encontra em (PAOLI; SARTINI; LAFORTUNE, 2011).

Proposição 6 *(Condições para Controlabilidade Segura de uma Linguagem pela Diagnose (PAOLI; SARTINI; LAFORTUNE, 2011)).* Considere uma linguagem L gerada por um autômato G e suponha que L é diagnosticável segura em relação à P_o, f e $\mathscr{K}_f$. Seja $\mathcal{FC}$ o conjunto dos primeiros estados certos de falha alcançados no c-diagnosticador seguro G_{sd}^c obtido a partir de G. A linguagem L é controlável segura pela diagnose se e somente se $\forall q_i \in \mathcal{FC}$, a linguagem $\{\varepsilon\}^{\downarrow C}$, calculada a partir do modelo não controlado pós-diagnose de falha G_i^{deg}, não contém nenhum elemento de Φ como subcadeia.

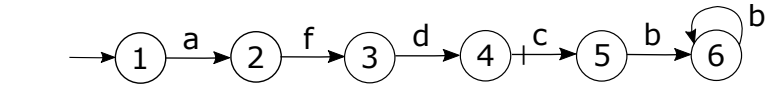
A ínfima superlinguagem controlável $\{\varepsilon\}^{\downarrow C}$ calculada em relação a G_i^{deg} contém todas as concatenações de eventos não-controláveis possíveis em G_i^{deg}.

Em palavras, a linguagem L é controlável segura em uma linguagem pela diagnose se e somente se for diagnosticável segura e, após a diagnose da falha, o sistema possa ser interrompido, via ação de controle, antes que execute uma ação proibida.

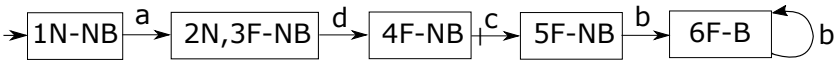
A seguir, dois exemplos são usados para ilustrar condições para controlabilidade segura pela diagnose seguindo a Proposição 6.

Exemplo 28 *Considere o autômato G_{28} mostrado na Figura 5.10(a), cuja linguagem é dada por $L_{28} = \overline{afdcbb^*}$, sendo $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,c,d\}$, $\Sigma_c = \{c\}$ e $\Sigma_f = \{f\}$. A cadeia proibida após a falha f é $\Phi = \{b\}$ e a linguagem ilegal é $\mathscr{K}_f = \{dcbb^*\}$.*

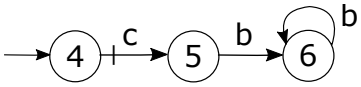
Figura 5.10 – Exemplo de linguagem controlável segura pela diagnose. (a) Autômato G_{28}; (b) Autômato c-diagnosticador seguro G^c_{sd28}; (c) Autômato da planta degradada pós-diagnose de falha $G_1^{deg,d}$.



(a)



(b)



(c)

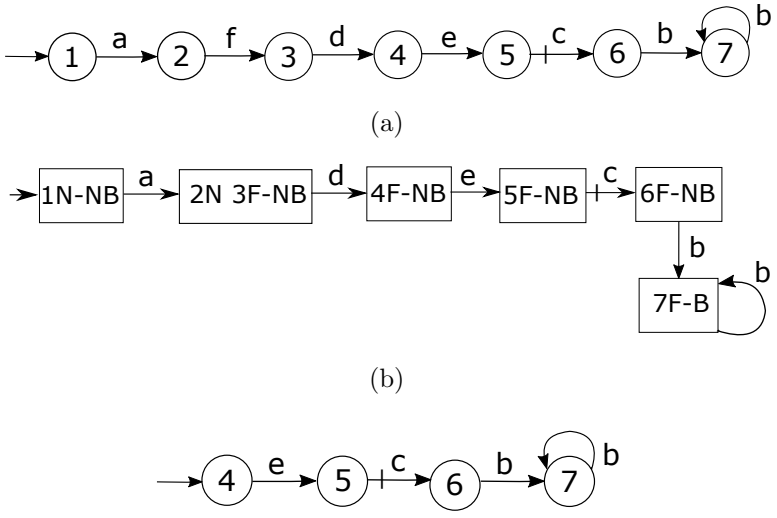
Fonte: (Autor.)

Em análise ao c-diagnosticador seguro G^c_{sd28} obtido a partir do autômato G_{28} apresentado na Figura 5.10(b), percebe-se que o conjunto dos primeiros estados certos é dado por $\mathcal{FC} = \{(4F-NB)\}$. O autômato $G_1^{deg,d}$, ilustrado, na Figura 5.10(c), é obtido a partir do estado (4) do autômato G_{28} apresentado na Figura 5.10(a), que corresponde ao estado $(4F-NB) \in \mathcal{FC}$. Ao calcular a ínfima superlinguagem de ε a partir de $G_1^{deg,d}$, que é controlável em relação à $L(G_1^{deg,d})$, obtém-se $\{\varepsilon\}^{\downarrow C} = \{\varepsilon\}$, pois pode-se desabilitar o evento controlável c existente no estado inicial de $G_1^{deg,d}$. A linguagem L_{28} é controlável segura pela diagnose, conforme a Proposição 6, pois a linguagem $\{\varepsilon\}^{\downarrow C}$ não possui nenhuma subcadeia proibida.

Exemplo 29 *Considere o autômato G_{29} mostrado na Figura 5.11(a), cuja linguagem é dada por $L_{29} = \overline{afdecbb^*}$, sendo $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,c,d,e\}$, $\Sigma_c = \{c\}$ e $\Sigma_f = \{f\}$. A cadeia proibida após a falha f é $\Phi = \{b\}$, e a linguagem ilegal é $\mathscr{K}_f = \{decbb^*\}$.*

Em análise ao c-diagnosticador seguro G^c_{sd29} obtido a partir do autômato G_{29} apresentado na Figura 5.11(b), percebe-se que o conjunto dos primeiros estados certos $\mathcal{FC} = (4F,NB)\}$. O autômato $G_1^{deg,d}$, ilustrado, na Figura 5.11(c), é obtido a partir do estado (4) do autômato G_{29} apresentado na Figura 5.11(a), que corresponde ao estado $(4F-NB) \in \mathcal{FC}$. Ao calcular a ínfima superlinguagem de ε a partir de $G_1^{deg,d}$, que é controlável em relação à $L(G_1^{deg,d})$, obtém-se $\{\varepsilon\}^{\downarrow C} = \{e\}$, pois a partir do estado inicial de $G_1^{deg,d}$ o primeiro evento controlável que pode ser desabilitado é o evento c, que se encontra no estado 5, ou seja, tal desabilitação se dá após a ocorrência do evento não controlável

Figura 5.11 – Exemplo de linguagem controlável segura pela diagnose. (a) Autômato G_{29}; (b) Autômato c-diagnosticador Seguro G^{c}_{sd29}; (c) Autômato da planta degradada pós-diagnose $G_1^{deg,d}$.



Fonte: (Autor.)

e. A linguagem L_{29} é controlável segura pela diagnose, conforme a Proposição 6, pois a linguagem $\{\varepsilon\}^{\downarrow C}$ não possui nenhuma subcadeia proibida.

5.5 CONDIÇÕES PARA CONTROLABILIDADE SEGURA DE UMA CADEIA PELA DIAGNOSE

Na sequência são introduzidas condições necessárias e suficientes para controlabilidade segura de uma cadeia pela diagnose. Antes, será introduzida a definição de $FB(s)$. Seja $FB(s)$ uma função que mapeia uma cadeia $s \in \Psi_L(f)$ no primeiro mau estado $q_{sd,B} \in Q^B$ no diagnosticador seguro G_{sd}, tal que $q_{sd,B}$ é alcançado do estado inicial $q_{sd,0}$ por uma continuação de $s_o = P_o(s)$. Formalmente, $FB(s) = \{q_{sd,B} \in Q^B : [(q_{sd,B} = \hat{\delta}_{sd}(q_{sd,0}, s_o z_o)$, com $s_o = P_o(s)$ e $z_o = P_o(z)$, tal que $z = v\xi \in L/s) \wedge (\nexists q'_{sd,B} \in Q^B : q'_{sd,B} = \hat{\delta}(q_{sd,0}, s_o z'_o)$ com $s_o z'_o < s_o z_o)]\}$.

Nesse momento, é importante destacar que na abordagem original de Paoli, Sartini e Lafortune (2011) para analisar a controlabilidade segura de uma linguagem pela diagnose é necessário calcular a ínfima superlinguagem controlável $\{\varepsilon\}^{\downarrow C}$ para cada ocorrência de falha. Entretanto, no intuito de simplificar a análise da controlabilidade segura de cadeias pela diagnose, nesta tese considera-se que os eventos controláveis são observáveis, i.e., $E_c \subseteq E_o$. Entende-se que essa consideração não é demasiadamente restritiva, pois normalmente nos sistemas de controle centralizado as ações de controle estão relacionadas

a saídas do dispositivo de controle (CLP ou microcontrolador), as quais são naturalmente observáveis por parte do referido dispositivo. Além disso, num trabalho futuro pretende-se explorar a propriedade da observabilidade relativa no contexto da controlabilidade segura a fim de flexibilizar tal consideração.

Proposição 7 *(Condições para Controlabilidade Segura de uma Cadeia pela Diagnose).* Considere um autômato G que gera linguagem L e considere que $\Sigma_c \subseteq \Sigma_o$. Seja $G_{sd} = (Q_{sd}, \Sigma_o, \delta_{sd}, q_{sd,0})$ o diagnosticador seguro construído a partir de G. A ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é controlável segura pela diagnose em relação à P_o e $\mathscr{K}_f$ se e somente se $\forall q_{sd} \in FC(s)$, as seguintes condições são atendidas:
$(P7_1)$ $q_{sd} \notin Q^B$;
$(P7_2)$ $\nexists w_o \in \Sigma_{uc}^* : \hat{\delta}_{sd}(q_{sd}, w_o) = q_{sd,B}$, sendo $q_{sd,B} \in FB(s)$.

Prova. A prova é em duas partes:
($\Rightarrow$) A condição necessária é provada em duas partes por contradição. Primeiro, suponha que a ocorrência de f numa cadeia $s \in \Psi_L(f)$ é controlável segura pela diagnose, mas a condição $(P7_1)$ não é satisfeita. Então, $\exists q_{sd} \in FC(s)$ tal que $q_{sd} \in Q^B$, e, dessa forma, pela Proposição 2, pode-se afirmar que a ocorrência de f em s não é diagnosticável segura. Assim, pela Definição 10, conclui-se que a ocorrência de f em s não é controlável segura pela diagnose, o que contraria a hipótese inicial. Suponha agora que a ocorrência de f em s é controlável segura pela diagnose, a condição $(P7_1)$ é atendida, mas a condição $(P7_2)$ não é satisfeita. Assim, pela Proposição 2 sabe-se que a ocorrência de f em s é diagnosticável segura e, portanto, $\forall q_{sd} \in FC(s)$ é tal que $q_{sd} \notin Q^B$. Se a condição $(P7_2)$ é violada, então $\exists q_{sd} \in FC(s)$ para o qual $\exists w_o \in \Sigma_{uc}^* : \hat{\delta}_{sd}(q_{sd}, w_o) = q_{sd,B}$, tal que $q_{sd,B} \in FB(s)$. Pela definição de $FB(s)$, o estado $q_{sd,B} \in Q^B$ é alcançado no diagnosticador por uma cadeia $s_o z_o$, para a qual $s_o = P_o(s), z_o = P_o(z)$, sendo $z = v\xi \in L/s$. Sem perda de generalidade, considere que $z = tw$, sendo que $t = r\sigma$, com $r \in \Sigma^*$ and $\sigma \in \Sigma_o$. Considere que a condição $\mathscr{D}$ não é atendida para sr, enquanto é atendida para st. Assim, $q_{sd} = \hat{\delta}_{sd}(q_{sd,0}, P_o(st))$ é tal que $q_{sd} \in FC(s)$. Uma vez que a ocorrência de f em s é diagnosticável segura, $q_{sd} \notin FB(s)$, e t não contém nenhum elemento de Φ como subcadeia. Por outro lado, como $q_{sd,B} \in Q^B$, a cadeia $z = tw$ contém um elemento de Φ como subcadeia, de onde se pode concluir que w contém um elemento de Φ como subcadeia. Uma vez $w_o \in \Sigma_{uc}^*$ e $\Sigma_c \subseteq \Sigma_o$, então para $w_o = P_o(w)$ pode-se afirmar que $w \in \Sigma_{uc}^*$. Assim, $\exists t \in L/s$ tal que $t = r\sigma$, com $r \in \Sigma^*$ e $\sigma \in \Sigma_o$, e tal que a condição de diagnosticabilidade $\mathscr{D}$ não é satisfeita para sr, enquanto é satisfeita para st para a qual $\exists w \in L/st$ tal que $w = u\xi$, com $\xi \in \Phi$, para a qual $\nexists \sigma_c \in \Sigma_c$ tal que $\sigma_c \in w$. Dessa forma, a condição $(D10_2)$ da Definição 10 não é satisfeita e, com

isso, pode-se afirmar que a ocorrência do evento f na cadeia $s \in \Psi_L(f)$ não é controlável segura pela diagnose, o que viola a hipótese inicial.

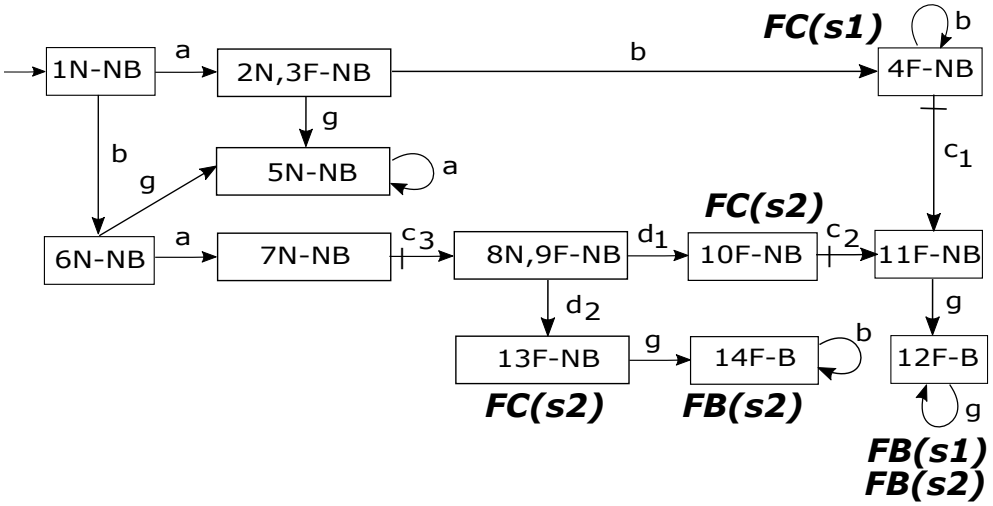
($\Leftarrow$) A condição suficiente é também provada por contradição. Suponha que a ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ não é controlável segura pela diagnose, mas as condições $(P7_1)$ e $(P7_2)$ são válidas para qualquer $q_{sd} \in FC(s)$. Sabe-se que se a ocorrência do evento f em s não é controlável segura pela diagnose, pelo menos uma das condições da Definição 10 não é válida. Se a condição $(D10_1)$ não é satisfeita, então pela Proposição 2 sabe-se que $\exists q_{sd} \in FC(s)$ tal que $q_{sd} \in Q^B$, que viola a condição $(P7_1)$, contrariando a hipótese inicial. Considere agora que a condição $(D10_1)$ é atendida (i.e., s é diagnosticável segura), mas a condição $(D10_2)$ não é satisfeita. Considere qualquer cadeia $t \in L/s$ tal que $t = r\sigma$, com $r \in \Sigma^*$ e $\sigma \in \Sigma_o$. Suponha que a condição de diagnosticabilidade $\mathscr{D}$ não é atendida para nenhum sr enquanto é atendida para st. Assim, $q_{sd} = \hat{\delta}_{sd}(q_{sd,0}, P_o(st))$ é tal que $q_{sd} \in FC(s)$. Além disso, como a ocorrência do evento f em s é diagnosticável segura (condição $(D10_1)$), $q_{sd} \notin Q^B$ e então a cadeia $t \in L/s$ não contém nenhum elemento de Φ como subcadeia. Se a condição $(D10_2)$ não é satisfeita para $s \in \Psi_L(f)$, então $\exists w \in L/st$ tal que $w = u\xi$, com $\xi \in \Phi$, para o qual $\nexists \sigma_c \in \Sigma_c$ tal que $\sigma_c \in w$. Como $w \in \Sigma^*_{uc}$, então $w_o = P_o(w) \in \Sigma^*_{uc}$. Para $q_{sd} = \hat{\delta}_{sd}(q_{sd,0}, P_o(st)) \in FC(s)$ tem-se que $\hat{\delta}_{sd}(q_{sd}, w_o) = q_{sd,B} \in Q^B$, tal que $w_o = P_o(w)$ e $w_o \in \Sigma^*_{uc}$. Assim, $\exists w_o \in \Sigma^*_{uc} : \hat{\delta}_{sd}(q_{sd}, w_o) = q_{sd,B}$, tal que $q_{sd,B} \in Q^B$. Sem perda de generalidade, considere que $z = tw$, sendo que $w = u\xi$. Então, para $s_o z_o = P_o(sz) = P_o(stw), \hat{\delta}_{sd}(q_{sd,0}, s_o z_o) = q_{sd,B} \in Q^B$ e $\nexists s_o z'_o < s_o z_o$ tal que $\hat{\delta}_{sd}(q_{sd,0}, s_o z'_o) \in Q^B$. Então, pela definição de $FB(s)$ pode-se dizer que $\hat{\delta}_{sd}(q_{sd}, w_o) = q_{sd,B} \in FB(s)$. Finalmente, pode-se concluir que $\exists q_{sd} \in FC(s)$ para o qual $\exists w_o \in \Sigma^*_{uc}$ tal que $\hat{\delta}_{sd}(q_{sd}, w_o) = q_{sd,B}$, sendo que $q_{sd,B} \in FB(s)$, violando a condição $(P7_2)$ e contrariando a hipótese inicial. □

Em palavras, a ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é controlável segura pela diagnose se e somente se nenhum estado $q_{sd} \in FC(s)$ é um mau estado e existir um evento controlável em cada subcadeia que parte de um estado de $FC(s)$ e alcança um estado de $FB(s)$.

A seguir, o Exemplo 27 da Figura 5.9 é retomado para ilustrar a condição para controlabilidade segura de uma cadeia pela diagnose. A Figura 5.12 mostra o c-diagnosticador seguro G^c_{sd27} para análise.

A linguagem L_{27} do autômato G_{27} apresenta duas cadeias $s \in \Psi_{L_{27}}(f)$, isto é, $s_1 = af$ e $s_2 = bac_3f$. Em análise ao c-diagnosticador seguro G^c_{sd27} obtido a partir do autômato G_{27}, percebe-se que para essas cadeias têm-se $FC(s_1) = \{(4F - NB)\}$ e $FC(s_2) = \{(10F - NB), (13F - NB)\}$. As funções $FB(s_1)$ e $FB(s_2)$, que representam os conjuntos de todos os primeiros maus estados, ou seja, primeiros estados que possuem rótulo B na

Figura 5.12 – Exemplo para ilustrar a análise de condição de controlabilidade segura de uma cadeia pela diagnose. Autômato c-diagnosticador seguro G^c_{sd27} ilustrando $FC(s)$ e $FB(s)$.



Fonte: (Autor.)

cadeia, são $\{(12F-B)\}$ e $\{(12F-B),(14F-B)\}$, respectivamente. As cadeias s_1 e s_2 são diagnosticáveis seguras, pois nenhum dos estados de $FC(s_1)$ e $FC(s_2)$ não é um mau estado, satisfazendo a condição $(P7_1)$ da Proposição 7. A cadeia s_1 é controlável segura pela diagnose, pois existe o evento controlável c_1 entre o estado $(4F-NB) \in FC(s_1)$ e o estado $(12F-B) \in FB(s_1)$, satisfazendo a condição $(P7_2)$ da Proposição 7. Embora, a cadeia s_2 seja diagnosticável segura, ela não é controlável segura pela diagnose, pois não existe um evento controlável na subcadeia do estado $(13F-NB) \in FC(s_2)$ ao estado $(14F-B) \in FB(s_2)$, não satisfazendo a condição $(P7_2)$ da Proposição 7.

A seguir é introduzida uma proposição que apresenta condições para linguagem controlável segura pela diagnose através da abordagem por cadeias.

Proposição 8 *(Condições para Controlabilidade Segura de uma Linguagem pela Diagnose).* Considere uma linguagem diagnosticável L e um autômato $G = (Q, \Sigma, \delta, q_0)$ que gera L. Seja $G^c_{sd} = (Q_{sd}, \Sigma_o, \delta_{sd}, q_{sd,0})$ o c-diagnosticador seguro construído a partir de G. A linguagem L é controlável segura pela diagnose em relação a projeção P_o, evento f e conjunto Φ se e somente se para toda cadeia $s \in \Psi_L(f)$, seguintes condições são atendidas para todo $q_{sd} \in FC(s)$:

(i) $q_{sd} \notin Q^B$; e

(ii) $\nexists w_o \in \Sigma^*_{uc} : \hat{\delta}_{sd}(q_{sd}, w_o) = q_{sd,B}$, sendo $q_{sd,B} \in FB(s)$.

Essa proposição substitui a Proposição 6 e a sua prova é similar a da Proposição 7.

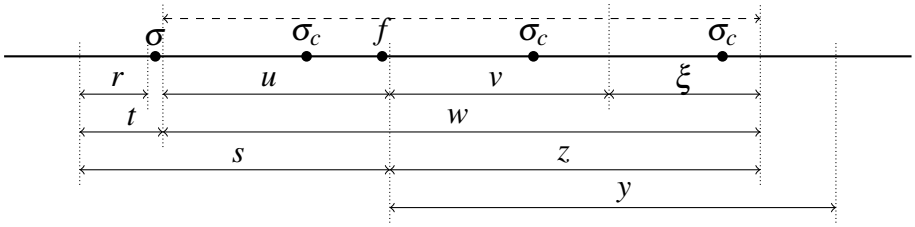
5.6 CONTROLABILIDADE SEGURA DE UMA LINGUAGEM PELA PROGNOSE

Introduz-se nesta seção o conceito de controlabilidade segura de uma linguagem pela prognose. Antes, porém, é importante lembrar que para fins de prognose de falhas, o diagnosticador utilizado será o c-diagnosticador seguro com alcance de eventos não-observáveis, conforme ressaltado na Observação 7.

Definição 11 *(Linguagem Controlável Segura pela Prognose (WATANABE et al., 2017a)).* Uma linguagem prefixo-fechada L que é viva e que não contém ciclos de eventos não-observáveis é dita ser controlável segura pela prognose em relação à projeção P_o, evento f e conjunto de cadeias proibidas Φ se atender às seguintes condições:
($D11_1$) Condição de prognosticabilidade: As ocorrências do evento f são prognosticáveis em L com relação à P_o.
($D11_2$) Condição de controlabilidade segura: Considere para qualquer cadeia $s \in L$ tal que $s = tu$ com $f \in u$ e $t = r\sigma$ tal que $\sigma \in \Sigma_o$. Suponha que a condição $\mathscr{P}$da definição de prognosticabilidade não é atendida para a cadeia r, mas é atendida para a cadeia t. Então, $(\forall w \in L/t)$ tal que $w = uv\xi$, com $\xi \in \Phi$, $\exists \sigma_c \in \Sigma_c$ tal que $\sigma_c \in w$.

Em palavras, uma linguagem é controlável segura pela prognose se para qualquer cadeia que contém uma falha f e uma cadeia proibida ou ilegal ξ, existir um evento observável σ que assegura a prognose da falha (condição ($D11_1$)), e um evento controlável σ_c, após o evento observável σ (condição ($D11_2$)), que possa ser usado para impedir a execução de qualquer sequência ilegal. Portanto, o evento controlável σ_c poderá estar antes ou depois da falha f (na subcadeia u) ou é um evento da própria cadeia proibida (na subcadeia $\xi \in \Phi$), conforme ilustrado através de uma linha tracejada na Figura 5.13.

Figura 5.13 – Gráfico para ilustrar conceito de cadeia controlável segura pela prognose.



Fonte: (Autor).

O Exemplo 28 da Figura 5.10 foi retomado para ilustrar a controlabilidade segura pela prognose. Porém, considere $\Sigma_c = \{d\}$ e $\Phi = \{b\}$.

A linguagem do autômato G_{28} é controlável segura pela prognose, pois as condições de prognosticabilidade são atendidas. A condição ($D11_1$) é satisfeita, pois existe um evento observável que garante a prognose (evento a). A condição ($D11_2$) é atendida, pois $s = tu = af$ e $t = r\sigma = a$, sendo que para $w \in L/t = fdcb$, tal que $w = uv\xi = fdcb$, com $\xi = b \in \Phi$, existe um evento controlável $\sigma_c = d \in \Sigma_c$, tal que $\sigma_c \in w$ que pode ser usado para evitar o evento ilegal b.

5.7 CONTROLABILIDADE SEGURA DE UMA CADEIA PELA PROGNOSE

Baseado no conceito de controlabilidade segura (de uma linguagem) pela prognose, apresentado por Watanabe et al. (2017a), é introduzido o conceito de Cadeia Controlável Segura pela Prognose.

Definição 12 *(Cadeia Controlável Segura pela Prognose).* Considere uma linguagem diagnosticável L. A ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é controlável segura pela prognose em relação à P_o e Φ se as seguintes condições são atendidas:
($D12_1$) *Condição de Prognosticabilidade*: a ocorrência do evento f é prognosticável em s em relação à P_o.
($D12_2$) *Condição de controlabilidade Segura pós-prognose*: considere $s = tu$ e $t = r\sigma$, com $r \in \Sigma^*$ e $\sigma \in \Sigma_o$, e tal que a condição de prognosticabilidade $\mathscr{P}$ não é atendida para r, enquanto é atendida para t. Então, ($\forall w \in L/t$ tal que $w = uv\xi$, com $\xi \in \Phi$), $\exists \sigma_c \in \Sigma_c$ tal que $\sigma_c \in w$.

Em palavras, a ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é controlável segura pela prognose se tal ocorrência é prognosticável e, após a ocorrência da menor cadeia que garante a prognose, existe um evento controlável σ_c para impedir a execução de qualquer cadeia ilegal após a ocorrência da falha nessa mesma cadeia. Assim, o evento controlável σ_c pode estar antes de f (na subcadeia u), depois de f (na subcadeia v) ou é um evento da própria cadeia proibida (na subcadeia ξ), conforme ilustrado pela linha tracejada na Figura 5.13.

O Exemplo 21 da Figura 4.3 é retomado para ilustrar condições para a controlabilidade segura de uma cadeia pela prognose. Porém, considera-se $\Sigma_c = \{c\}$ e $\Phi = \{g\}$.

Conforme discutido anteriormente, existem três cadeias em L_{21} do autômato G_{21}, sendo s_1 e s_2 prognosticáveis e s_3 não é prognosticável. A cadeia s_1 é controlável se-

gura pela prognose, pois sendo prognosticável, a condição ($D12_1$) é satisfeita e a condição ($D12_2$) também é satisfeita, pois $s_1 = t_1u_1 = acf$ e $t_1 = r_1\sigma_1 = a$, tal que a condição de prognosticabilidade $\mathscr{P}$ é atendida para a. Então, para $w_1 = u_1v_1\xi = cfg \in L/t_1, \exists\sigma_c = c \in \Sigma_c$ tal que $c \in w_1$. A cadeia s_2, embora seja prognosticável, satisfazendo a condição ($D12_1$), não é controlável segura pela prognose, pois a condição ($D12_2$) não é satisfeita, pois $s_2 = t_2u_2 = bf$ e $t_2 = r_2\sigma_2 = b$, tal que a condição de prognosticabilidade $\mathscr{P}$ é atendida para b. Porém, para $w_2 = u_2v_2\xi = fag \in L/t_2, \nexists\sigma_c \in \Sigma_c$ tal que $\sigma_c \in w_2$. A cadeia s_3 não é controlável segura pela prognose, pois não sendo prognosticável a condição ($D12_1$) não é satisfeita.

A partir da Definição 11, pode-se reescrever a definição de cadeia controlável segura pela prognose, conforme segue.

Uma linguagem diagnosticável L é dita ser controlável segura pela prognose em relação a projeção P_o, evento f e conjunto Φ se a ocorrência do evento f em todas as cadeias $s \in \Psi_L(f)$ é controlável segura pela prognose.

5.8 CONDIÇÕES PARA CONTROLABILIDADE SEGURA DE UMA LINGUAGEM PELA PROGNOSE

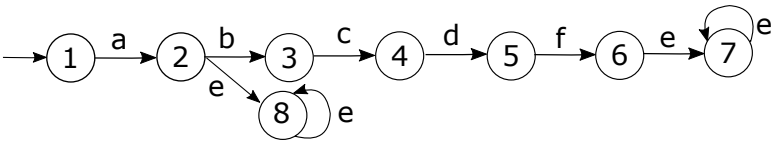
A Proposição 9 introduz as condições necessárias e suficientes para a controlabilidade segura pela prognose. Antes, porém, são apresentados alguns conceitos que serão importantes no contexto da Proposição 9. A diagnose segura não é uma condição necessária para a controlabilidade segura pela prognose.

Em sistemas de controle, o melhor é obter a prognose o quanto antes para possibilitar a tomada de decisão o mais cedo possível. O conjunto $\mathcal{FU}$ foi estabelecido para prover novas condições para a prognosticabilidade de falhas numa linguagem, porém, não é o mais adequado para tratar da controlabilidade segura pela prognose. Portanto, foi proposto por Watanabe et al. (2017a) o conjunto dos primeiros estados que asseguram a prognose (*First-entered states that assure the Prognosis*), denominado $\mathcal{FP}$. Esses estados são os estados do $\mathcal{FU}$ ou dos estados que precedem este conjunto e podem ser estados normais ou incertos. Formalmente, $\mathcal{FP} = \{q_{sd} \in Q_{sd}^N \cup Q_{sd}^U : [(\text{condição } \mathscr{C} \text{ é satisfeita para } q_{sd}) \wedge (\exists q'_{sd} \in BS(q_{sd}) \text{ para o qual a condição } \mathscr{C} \text{ não é satisfeita})]\}$, sendo $BS(q_{sd})$ o alcance observável um passo para trás de um estado $q_{sd} \in Q_{sd}$, formalmente definido por $BS(q_{sd}) = \{q'_{sd} \in Q_{sd} : (\exists\sigma \in \Sigma_o)(\delta_{sd}(q'_{sd}, \sigma) = q_{sd})\}$.

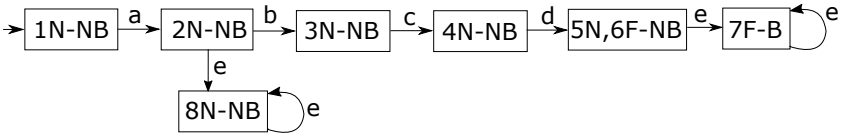
A seguir, apresenta-se um exemplo para mostrar a aplicação do $\mathcal{FP}$.

Exemplo 30 *Considere o autômato G_{30} mostrado na Figura 5.14 (a), cuja linguagem é dada por $L_{30} = \overline{a(bcdfee^* + ee^*)}$, sendo que $\Sigma_{uo} = \{f\}$, $\Sigma_o = \{a,b,c,d,e\}$ e $\Sigma_f = \{f\}$. O conjunto de cadeias proibidas após a falha f é dado por $\Phi = \{e\}$ e a linguagem ilegal é $\mathscr{K}_f = \{ee^*\}$.*

Figura 5.14 – Exemplo para ilustrar a obtenção de $\mathcal{FP}$. (a) Autômato G_{30}; (b) Autômato c-diagnosticador G^c_{sd30}.



(a)



(b)

Fonte: (Autor.)

Analisando o c-diagnosticador G^c_{sd30} mostrado na Figura 5.14 (b), obtido a partir do autômato G_{30}, pode-se observar que o conjunto de estados incertos é $\mathcal{FU} = \{(5N, 6F - NB)\}$. A linguagem L_{30} é prognosticável, pois o único ciclo de estados alcançado a partir do estado $(5N, 6F - NB)$, é um ciclo de estado certo $(7F - NB)$ no c-diagnosticador G^c_{sd30}.

Analisando o c-diagnosticador G^c_{sd30} da Figura 5.14 (b), verifica-se que o estado $(3N - NB)$ é o primeiro estado (nesse caso, normal) que atende a condição $\mathscr{C}$ e, portanto, $\mathcal{FP} = \{(3N - NB)\}$. Observe que, de fato, a prognose da falha se dá após a observação da cadeia ab, de modo que o conjunto $\mathcal{FP}$ captura os estados alcançados com as menores cadeias que levam à prognose da falha.

A seguir é apresentado o algoritmo introduzido por Watanabe et al. (2017a) que calcula $\mathcal{FP}$ para um SED prognosticável. A seguir, são introduzidas mais algumas definições usadas no algoritmo.

Seja $BR(q_{sd})$ o alcance observável para trás a partir de um estado $q_{sd} \in Q_{sd}$, definido formalmente por $BR(q_{sd}) = \{q'_{sd} \in Q_{sd} : (\exists t_o \in \Sigma_o^*)(\hat{\delta}_{sd}(q'_{sd}, t_o) = q_{sd})\}$.

Seja $BR(Q)$ o alcance observável para trás a partir de um conjunto de estados $Q \subseteq Q_{sd}$, definido formalmente por $BR(Q) = \bigcup_{q_{sd} \in Q_{sd}} BR(q_{sd})$.

Seja $FS(q'_{sd})$ o alcance observável um passo para frente de um estado $q_{sd} \in Q_{sd}$, definido formalmente por $FS(q'_{sd}) = \{q_{sd} \in Q_{sd} : (\exists \sigma \in \Sigma_o)(\delta_{sd}(q'_{sd}, \sigma) = q_{sd})\}$.

Algoritmo 1: Primeiros estados que asseguram a prognose

Entrada: $\mathcal{FU}$

Saída: $\mathcal{FP}$

1 **início**
2 $\quad \mathcal{FP} \leftarrow \mathcal{FU}$;
3 $\quad$ **para** *cada estado* $q_{sd} \in \mathcal{FP}$ *nao analisado* **faça**
4 $\quad\quad$ calcule $BS(q_{sd})$;
5 $\quad\quad$ **para** *cada estado* $q'_{sd} \in BS(q_{sd})$ **faça**
6 $\quad\quad\quad$ **se** $FS(q'_{sd}) \subseteq \mathcal{FP}$ **então**
7 $\quad\quad\quad\quad$ $\mathcal{FP} = \mathcal{FP} \cup \{q'_{sd}\}$;
8 $\quad\quad\quad$ **fim**
9 $\quad\quad$ **fim**
10 $\quad$ **fim**
11 $\quad$ $R \leftarrow \mathcal{FP}$;
12 $\quad$ **para** *cada estado* $q_{sd} \in R$ **faça**
13 $\quad\quad$ **se** $BS(q_{sd}) \subseteq R$ **então**
14 $\quad\quad\quad$ remove q_{sd} de $\mathcal{FP}$;
15 $\quad\quad$ **fim**
16 $\quad$ **fim**
17 **fim**
18 **retorna** $\mathcal{FP}$

Fonte: (Autor)

O Algoritmo 1 é dividido em duas principais partes, sendo que a primeira parte é relacionada com a adição de estados em $\mathcal{FP}$ (passo 3 ao 10). Basicamente, são acrescentados os estados de G^c_{sd} que precedem estados de $\mathcal{FP}$, desde que estes alcancem (um passo à frente) somente estados que já estão em $\mathcal{FP}$. A segunda parte (passo 12 ao 16) diz respeito à exclusão dos estados de $\mathcal{FP}$ que já não são mais considerados como o primeiro estado que assegura a prognose. No passo 11 são salvos no conjunto auxiliar R as informações sobre todos os estados adicionados durante a primeira parte. Isso é importante para garantir que todos os estados alcançados somente pelos estados em R serão removidos de $\mathcal{FP}$.

Antes de introduzir a condição necessária e suficiente para controlabilidade segura pela prognose, são apresentadas mais algumas noções importantes. Seja $G^{deg,p}_j$ o j-nésimo modelo que representa o comportamento não controlado pós-prognose, ou seja, o modelo para o comportamento degradado da planta que segue a cadeia responsável pela prognose.

A ínfima superlinguagem controlável de ε, $\{\varepsilon\}^{\downarrow C}$, calculada em relação à $G_j^{deg,p}$ contém todas as concatenações de eventos não-controláveis possíveis em $L(G_j^{deg,p})$.

Proposição 9 *(Condições para Controlabilidade Segura de uma Linguagem pela Prognose (WATANABE et al., 2017a)).* Considere uma linguagem L gerada por um Autômato G e assuma que as ocorrências do evento f são prognosticáveis em relação à P_o. Seja $G_{sd}^c = (Q_{sd}, \Sigma_o, \delta_{sd}, q_{sd,0})$ o c-diagnosticador seguro construído a partir de G. Seja $\mathcal{FP}$ o conjunto dos primeiros estados em G_{sd}^c que asseguram a prognose. A linguagem L é controlável segura pela prognose se e somente se $\forall q_j \in \mathcal{FP}$, a linguagem $\{\varepsilon\}^{\downarrow C}$, calculada em relação a $G_j^{deg,p}$ é tal que $P^{-1}(\{\varepsilon\}^{\downarrow C}) \cap L(G_j^{deg,p})$, não contém nenhum elemento de Φ como subcadeia após o evento f.

Prova. A prova é em duas partes:
($\Rightarrow$) Primeiro, é provado por contradição que L é controlável segura pela prognose se $\forall q_{sd,j} \in \mathcal{FP}$, a linguagem $\{\varepsilon\}^{\downarrow C}$ calculada em relação à $L(G_j^{deg,p})$ é tal que $P^{-1}(\{\varepsilon\}^{\downarrow C}) \cap L(G_j^{deg,p})$ não contém nenhum elemento de Φ como subcadeia após um evento f. Suponha que L é controlável segura pela prognose, mas $\exists q_{sd,j} \in \mathcal{FP}$ tal que $P^{-1}(\{\varepsilon\}^{\downarrow C}) \cap L(G_j^{deg,p})$ contém um elemento de Φ como subcadeia após um evento f. Assuma que em G_{sd}^c o estado $q_{sd,j}$ é alcançado do estado inicial com uma cadeia $t_o \in \Sigma_o^*$. Assim, de acordo com a definição de $\mathcal{FP}$, a condição $\mathscr{C}$ é atendida para $q_{sd,j}$ e $\exists q'_{sd} \in BS(q_{sd,j})$ para o qual a condição $\mathscr{C}$ não é atendida. Então, a condição $\mathscr{P}$ é atendida para t tal que $t_o = P_o(t)$ enquanto não é atendida para nenhum prefixo de t. Conforme pela hipótese $P^{-1}(\{\varepsilon\}^{\downarrow C}) \cap L(G_j^{deg,p})$ contém um elemento de Φ como subcadeia após um evento f, não existe um evento controlável em $L(G_j^{deg,p})$ que possa ser usado para evitar a ocorrência da subcadeia ilegal após f. Assim, pode-se concluir que $\exists w \in L/t$ tal que $w = uv\xi$, com $\xi \in \Phi$ para o qual $\nexists \sigma_c \in \Sigma_c$ tal que $\sigma_c \in w$. Sem perda de generalidade, considere que $s \in \Psi(f)$ e deixe $s = tu$. Assim, pode-se concluir que a condição $(D11_2)$ da Definição 11 é violada e portanto, a ocorrência de f em $s \in \Psi(f)$ não é controlável segura pela prognose. Assim, de acordo com a Definição 11, L não é controlável segura pela prognose, contrariando a hipótese inicial.
($\Leftarrow$) Agora, pode-se provar por contradição que L é controlável segura pela prognose se $\forall q_{sd,j} \in \mathcal{FP}$, $P^{-1}(\{\varepsilon\}^{\downarrow C}) \cap L(G_j^{deg,p})$ não contém um elemento de Φ como subcadeia após um evento de falha f. Suponha que L não é controlável segura pela prognose, mas $\forall q_{sd} \in \mathcal{FP}, P^{-1}(\{\varepsilon\}^{\downarrow C}) \cap L(G_j^{deg,p})$ não contém um elemento de Φ como subcadeia após um evento f. Uma vez supondo que L é prognosticável, pela Definição 6 a ocorrência de f em qualquer cadeia $s \in \Psi(f)$ é prognosticável, o qual satisfaz a condição $(D11_1)$ da Definição 11. Assim,

conclui-se que a condição ($D11_2$) da Definição 11 não é atendida para $s \in \Psi(f)$. Portanto, sem perda de generalidade, deixe $s = tu$ com $s \in \Psi(f)$ e $t = r\sigma$, com $r \in \Sigma^*$ e $\sigma \in \Sigma_o$, ser tal que a condição de prognosticabilidade $\mathscr{P}$ não é atendida para r, enquanto é atendida para t. Então, pode-se concluir que $\exists w \in L/t$ tal que $w = uv\xi$, com $\xi \in \Phi$ para qual $\nexists \sigma_c \in \Sigma_c$ tal que $\sigma_c \in w$. Ou seja, não existe nenhum evento controlável após a prognose de falha e antes da execução da sequência proibida ξ após f. Assim, uma vez que a cadeia $t_o = P_o(t)$ conduz para um estado $q_{sd,j} \in \mathcal{FP}$, pode-se concluir que para o correspondente $G_j^{deg,p}$ a linguagem $P^{-1}(\{\varepsilon\}^{\downarrow C}) \cap L(G_j^{deg,p})$ contém $\xi \in \Phi$ como subcadeia após f, contrariando a hipótese inicial. □

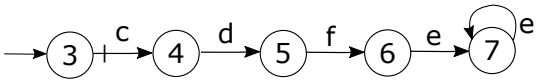
Em palavras, a linguagem L é controlável segura pela prognose se e somente se for prognosticável e para cada estado alcançado com um evento de prognose de falha, a evolução da planta possa ser interrompida, via ação de controle, antes da ocorrência de uma ação proibida. Vale destacar a possibilidade de tal interrupção se dar antes mesmo da ocorrência da falha, e que, neste caso, não haveria uma cadeia a ser impedida após a falha.

A seguir, o Exemplo 30 da Figura 5.14 é retomado, porém, considerando $\Sigma_c = \{c\}$ para ilustrar as condições de controlabilidade segura de uma linguagem pela prognose.

É interessante observar que o evento $e \in \Phi$ (ou a cadeia $ee^* \in \mathscr{K}_f$) pode ocorrer tanto no estado (2) quanto no estado (6) de G_{30} mostrado na Figura 5.14 (a), mas que ele só se torna um evento proibido após a ocorrência da falha, de modo que o mesmo não é proibido no estado (2) de G_{30}.

A Figura 5.14 (b) ilustra o c-diagnosticador G_{sd30}^c, do qual se pode determinar que o $\mathcal{FU} = \{(5N, 6F - NB)\}$. Pode-se verificar que a linguagem é prognosticável, pois o ciclo alcançado a partir do estado $(5N, 6F - NB)$ é um ciclo de estado certo. Também se pode encontrar o conjunto dos primeiros estados de G_{sd30}^c nos quais a condição $\mathscr{C}$ é atendida, dado por $\mathcal{FP} = \{(3N - NB)\}$. O autômato para o comportamento degradado pós-prognose $G_1^{deg,p}$ ilustrado na Figura 5.15 é obtido a partir do estado (3) do G_{30}. Ao calcular a ínfima superlinguagem de ε que é controlável em relação a $L(G_1^{deg,p})$, obtém-se $\{\varepsilon\}^{\downarrow C} = \{\varepsilon\}$, pois não há nenhum evento não-controlável antes do evento controlável c.

Figura 5.15 – Exemplo para ilustrar a análise da condição de controlabilidade segura de uma linguagem pela prognose. Autômato da planta degradada pós-prognose $G_1^{deg,p}$.



Fonte: (Autor.)

Portanto, a linguagem L_{30} é controlável segura pela prognose, conforme a Proposição 9, pois a linguagem $\{\varepsilon\}^{\downarrow C}$ não possui nenhuma subcadeia proibida.

Observação 10 *É importante destacar neste exemplo a importância do $\mathcal{FP}$ em relação ao F_D para fins de controle, pois analisando o G^c_{sd30}, não teríamos a controlabilidade segura pela prognose se utilizássemos o $\mathcal{FU}$, já que o evento e não é controlável. No caso do $\mathcal{FP}$, após o estado $(3N)$ temos um evento controlável garantindo assim a controlabilidade segura pela prognose.*

5.9 CONDIÇÕES PARA CONTROLABILIDADE SEGURA DE UMA CADEIA PELA PROGNOSE

Antes de introduzir condições para a controlabilidade segura de uma cadeia pela prognose, define-se $FP(s)$.

Seja $FP(s)$ uma função que mapeia uma cadeia $s \in \Psi_L(f)$ em um estado $q_{sd} \in G^c_{sd}$ que é alcançado do estado inicial $q_{sd,0}$ com o menor prefixo de $P_o(s)$ que assegura a prognose de falha na cadeia s. Formalmente, $FP(s) = q_{sd} \in Q_{sd} : [(q_{sd} = \hat{\delta}_{sd}(q_{sd,0}, t_o)$, com $t_o \in \overline{P_o(s)}$, para a qual a condição $\mathscr{C}$ é atendida) $\wedge$ $(\nexists q'_{sd} \in Q_{sd} : q'_{sd} = \hat{\delta}_{sd}(q_{sd,0}, r_o)$, com $r_o < t_o$, para a qual a condição $\mathscr{C}$ é atendida)].

Pelo mesmo motivo que foi apresentado na introdução das condições para controlabilidade segura de uma cadeia pela diagnose, considera-se aqui que os os eventos controláveis (Σ_c) são observáveis (Σ_o), i.e., $\Sigma_c \subseteq \Sigma_o$.

Proposição 10 *(Condições para Controlabilidade Segura de uma Cadeia pela Prognose).* Considere um autômato G que gera a linguagem L, assuma que $\Sigma_c \subseteq \Sigma_o$, e considere que L é diagnosticável em relação à f e P_o. Seja $G^c_{sd} = (Q_{sd}, \Sigma_o, \delta_{sd}, q_{sd,0})$ o c-diagnosticador seguro construído a partir de G. A ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é controlável segura pela prognose em relação à P_o e $\mathscr{K}_f$ se e somente se as seguintes condições são atendidas:
$(P10_1)$ a condição $\mathscr{C}$ é atendida para o estado $q_{sd} = FU(s)$;
$(P10_2)$ para $q'_{sd} = FP(s)$, $\nexists w_o \in \Sigma^*_{uc} : \hat{\delta}_{sd}(q'_{sd}, w_o) = q_{sd,B}$, sendo que $q_{sd,B} \in FB(s)$.

Prova. A prova é em duas partes:
$(\Rightarrow)$ Condição necessária é provada em duas partes por contradição. Primeiro, suponha que a ocorrência de f numa cadeia $s \in \Psi_L(f)$ é controlável segura pela prognose, mas a condição $(P10_1)$ não é satisfeita. De acordo com a Proposição 5, se a condição $\mathscr{C}$ não é atendida para $q_{sd} = FU(s)$ então a ocorrência de f em s não é prognosticável, o qual viola

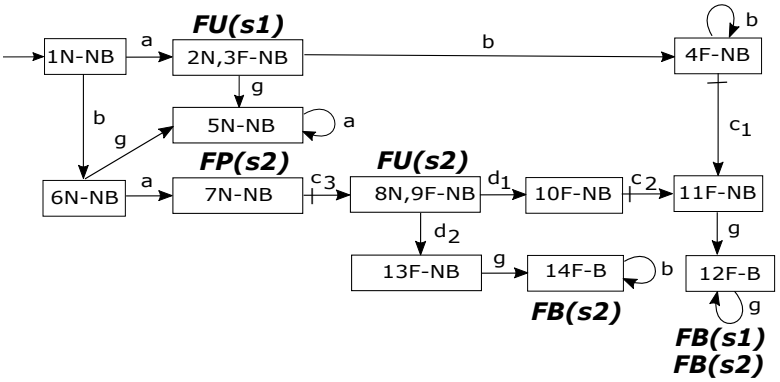
a condição ($D12_1$) da Definição 12 e, portanto, a ocorrência de f em s não é controlável segura pela prognose. A seguir, suponha que a ocorrência de f em $s \in \Psi_L(f)$ é controlável segura pela prognose, a condição ($P10_1$) é atendida, mas a condição ($P10_2$) não é satisfeita. Assim, se a condição $\mathscr{C}$ é atendida para $q_{sd} = FU(s)$, sabe-se que a ocorrência de f em s é prognosticável. Suponha que $s = tu$ e $t = r\sigma$, com $r \in \Sigma^*$ e $\sigma \in \Sigma_o$ e tal que a condição de prognosticabilidade $\mathscr{P}$ não é atendida para r enquanto é atendida para t. Então, $t_o = P_o(t)$ é o menor prefixo de $P_o(s)$ que assegura prognose de falha em s. Assim, pela definição de $FP(s), \hat{\delta}_{sd}(q_{sd,o}, t_o) = q'_{sd} = FP(s)$, ou seja, no diagnosticador seguro, a cadeia t_o leva do estado inicial até o primeiro estado que assegura a prognose de falhas em s. Se a condição ($P10_2$) é violada, então para $q'_{sd} = FP(s)$, $\exists w_o \in \Sigma^*_{uc} : \hat{\delta}_{sd}(q'_{sd}, w_o) = q_{sd,B}$, tal que $q_{sd,B} \in FB(s)$. Pela definição de $FB(s)$, sabe-se que para $w_o = P_o(w)$, a cadeia w é da forma $w = v\xi$, com $\xi \in \Phi$. Sem perda de generalidade, considere que $w = uv\xi$. Uma vez que $\Sigma_c \subseteq \Sigma_o$, então $w_o = P_o(w)$ é tal que $w \in \Sigma^*_{uc}$. Assim pode-se concluir que $\exists w \in L/t$ tal que $w = uv\xi$, com $\xi \in \Phi$, para a qual $\nexists \sigma_c \in \Sigma_c$ tal que $\sigma_c \in w$. Portanto, a condição ($D12_2$) da Definição 12 é violada e assim a ocorrência de f em $s \in \Psi_L(f)$ não é controlável segura pela prognose, contrariando a hipótese inicial.

($\Leftarrow$) Nesta parte da prova considera-se que ambas condições ($P10_1$) e ($P10_2$) são atendidas e prova-se que a ocorrência do evento f em $s \in \Psi_L(f)$ é controlável segura pela prognose. Primeiro, suponha que a condição ($P10_1$) é atendida. Então, a condição $\mathscr{C}$ é atendida para o estado $q_{sd} = FU(s)$ e pela, Proposição 5, a ocorrência do evento f na cadeia s é prognosticável, satisfazendo a condição ($D12_1$) da Definição 12. A seguir, considere $s = tu$ com $t = r\sigma$, $r \in \Sigma^*$ e $\sigma \in \Sigma_o$, tal que a condição da prognosticabilidade $\mathscr{P}$ não é atendida para r, enquanto é atendida para t. Assim, $t_o = P_o(t)$ é o menor prefixo de $P_o(s)$ que assegura a prognose de falha em s. Então, $\hat{\delta}_{sd}(q_{sd,0}, t_o) = q'_{sd} = FP(s)$. Agora, suponha que a condição ($P10_2$) é atendida e então para $q'_{sd} = FP(s)$, $\nexists w_o \in \Sigma^*_{uo}$ tal que $\hat{\delta}_{sd}(q'_{sd}, w_o) = q_{sd,B}$, sendo $q_{sd,B} \in FB(s)$. Em outras palavras, $\exists w_o \in \Sigma^*_o$ tal que $\hat{\delta}_{sd}(q'_{sd}, w_o) = q_{sd,B} \in FB(s)$, $\exists \sigma_c \in \Sigma_c \cap \Sigma_o$ tal que $\sigma_c \in w_o$. Uma vez que $q_{sd,B} \in FB(s)$, então sabe-se que w é da forma $w = uv\xi$, com $\xi \in \Phi$. Também sabe-se que para $w_o = P_o(w), \sigma_c \in w_o \Rightarrow \sigma_c \in w$. Assim, $\forall w \in L/t$ tal que $w = uv\xi$, com $\xi \in \Phi$, $\exists \sigma_c \in \Sigma_c$ tal que $\sigma_c \in w$, o qual satisfaz a condição ($D12_2$) da Definição 12. Finalmente, uma vez que ($D12_1$) e ($D12_2$) são atendidas, conclui-se que a ocorrência do evento f em $s \in \Psi_L(f)$ é controlável segura pela prognose. □

Em palavras, a ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ é controlável segura pela prognose se e somente se todos os estados alcançados do estado $q_{sd} = FU(s)$ atendem a condição $\mathscr{C}$ e existe pelo menos um evento controlável que possa ser usado para impedir o alcance de qualquer mau estado de $FB(s)$ a partir do estado $q'_{sd} = FP(s)$.

A seguir, retoma-se o Exemplo 27 da Figura 5.9 a fim de ilustrar as condições estabelecidas na Proposição 10. Na Figura 5.16 apresenta-se o c-diagnosticador seguro G^c_{sd27} obtido a partir de G_{27}. Para facilitar a análise, nesta figura são indicados os estados de FU(s), FP(s) e FB(s) para as duas cadeias que terminam com falha em L_{27}.

Figura 5.16 – Exemplo para ilustrar a análise de condição de controlabilidade segura de uma cadeia pela prognose. Autômato c-diagnosticador seguro G^c_{sd27} ilustrando $FU(s)$, $FP(s)$ e $FB(s)$.



Fonte: (Autor.)

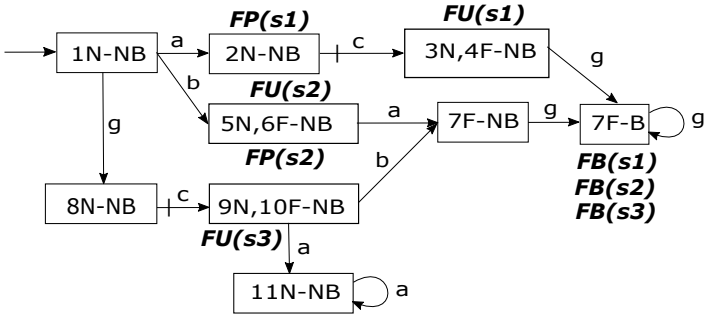
Conforme discutido anteriormente, existem duas cadeias $s \in \Psi_{L_{27}}(f)$ em L_{27}, isto é, $s_1 = af$ e $s_2 = bac_3f$. Para essas cadeias, têm-se $FU(s_1) = (2N, 3F - NB)$, $FU(s_2) = (8N, 9F - NB)$, $FP(s_2) = (7N - NB)$, $FB(s_1) = \{(12F - B)\}$ e $FB(s_2) = \{(12F, B), (14F, B)\}$. A cadeia s_1 não é controlável segura pela prognose, pois a condição $\mathscr{C}$ não é atendida para o estado $(2N, 3F - NB) = FU(s_1)$, assim não satisfazendo a condição $(P10_1)$ da Proposição 10. Entretanto, a cadeia s_2 é prognosticáve, pois a condição $\mathscr{C}$ é atendida para o estado $(8N, 9F - NB) = FU(s_2)$ satisfazendo a condição $(P10_1)$. A condição $(P10_2)$ é analisada através do primeiro estado que assegura a prognose para a cadeia s_2 que é dado por $FP(s_2)$. A cadeia s_2 é controlável segura pela prognose, pois existe um evento controlável c_3 entre o estado $(7N - NB) = FP(s_2)$ e os estados $(12F - B) \in FB(s_2)$ e $(14F - B) \in FB(s_2)$, satisfazendo a condição $(P10_2)$ da Proposição 10.

A seguir, retoma-se o Exemplo 21 da Figura 4.3 para ilustrar a condição para controlabilidade segura de uma cadeia pela prognose.

A análise será realizada no c-diagnosticador seguro G^c_{sd21} mostrado na Figura 5.17. Vale lembrar que $\Phi = \{g\}$ e que $\Sigma_c = \{c\}$. Conforme discutido anteriormente, existem três cadeias $s \in \Psi_{L_{21}}(f)$ em L_{21}, ou seja, $s_1 = acf$, $s_2 = bf$ e $s_3 = gcf$. Para essas cadeias, têm-se as funções que mapeiam os primeiros estados certos $FU(s_1) = (3N, 4F - NB), FU(s_2) = (5N, 6F - NB)$ e $FU(s_3) = (9N, 10F - NB)$, as funções que mapeiam o primeiro estado que assegura prognose $FP(s_1) = (2N - NB)$, $FP(s_2) = (5N, 6F - NB)$ e as funções que mapeiam o primeiro mau estado $FB(s_1) = FB(s_2) = FB(s_3) = \{(7F - B)\}$. As cadeias s_1

e s_2 são prognosticáveis, pois os estados alcançados a partir dos estados $(3N, 4F-NB) = FU(s_1)$ e $(5N, 6F-NB) = FU(s_2)$, respectivamente, têm somente ciclos de estados certos, satisfazendo as condições $(P10_1)$. A cadeia s_1 é controlável segura pela prognose, pois tem um evento controlável c entre o estado $(2N-NB) = FP(s_1)$ e o estado $(7F-B) \in FB(s_1)$, satisfazendo a condição $(P10_2)$. A cadeia s_2 não é controlável segura pela prognose, apesar da condição $(P10_1)$ ser satisfeita. A cadeia s_2 não é controlável segura pela prognose, pois não tem um evento controlável entre o estado $(5N, 6F-NB) = FP(s_2)$ e o estado $(7F-B) \in FB(s_2)$, não satisfazendo a condição $(P10_2)$. A cadeia s_3 não é prognosticável, pois o estado $(11N-NB)$ alcançado a partir de $(9N, 10F-NB) = FU(s_3)$ tem um ciclo de estado normal, violando as condições estabelecidas $(P10_1)$. Portanto, a cadeia s_3 não é controlável segura pela prognose, pois as condições da Proposição 10 não são satisfeitas.

Figura 5.17 – Exemplo para ilustrar a análise de condição de controlabilidade segura pela prognose uma cadeia. Autômato c-diagnosticador seguro G^c_{sd21} ilustrando $FU(s)$, $FP(s)$ e $FB(s)$.



Fonte: (Autor.)

A seguir é introduzida uma proposição que apresenta condições para linguagem controlável segura pela prognose através da abordagem por cadeias.

Proposição 11 *(Condições para Controlabilidade Segura de uma Linguagem pela Prognose).* Considere uma linguagem diagnosticável L e um autômato $G = (Q, \Sigma, \delta, q_0)$ que gera L. Seja $G^c_{sd} = (Q_{sd}, \Sigma_o, \delta_{sd}, q_{sd,0})$ o c-diagnosticador seguro construído a partir de G. A linguagem L é controlável segura pela prognose em relação a projeção P_o, evento f e conjunto Φ se e somente se para toda cadeia $s \in \Psi_L(f)$, seguintes condições são atendidas:
(i) condição $\mathscr{C}$ é atendida para o estado $q_{sd} = FU(s)$; e
(ii) para $q'_{sd} = FP(s)$, $\nexists w_o \in \Sigma^*_{uc} : \hat{\delta}_{sd}(q'_{sd}, w_o) = q_{sd,B}$, sendo que $q_{sd,B} \in FB(s)$.

Essa proposição substitui a Proposição 9 e sua prova é similar a da Proposição 10.

5.10 CONTROLABILIDADE SEGURA DE UMA LINGUAGEM PELA DIAGNOSE OU PROGNOSE

A seguir introduz-se uma definição mais abrangente para controlabilidade segura de SEDs que contempla tanto a diagnose quanto a prognose. Nessa definição é usada a abordagem pela controlabilidade segura de uma cadeia pela diagnose ou prognose.

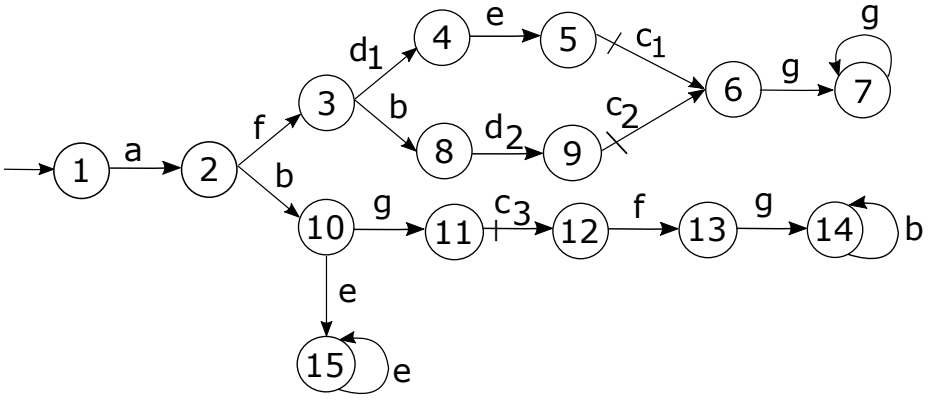
Definição 13 *(Linguagem Controlável Segura pela Diagnose ou Prognose).* Uma linguagem diagnosticável L é dita ser controlável segura pela diagnose ou prognose *(DP-Controlável Segura)* em relação à projeção P_o, evento f e Φ, se a ocorrência do evento f em toda cadeia $s \in \Psi_L(f)$ é controlável segura pela diagnose ou controlável segura pela prognose.

Em palavras, uma linguagem é controlável segura pela diagnose ou pela prognose se todas as cadeias $s \in \Psi_L(f)$ são controláveis seguras pela diagnose ou pela prognose. Dessa forma, parte das cadeias pode ser controlável segura pela diagnose e outra parte pode ser controlável segura pela prognose.

A seguir, será apresentado um exemplo para ilustrar esse conceito.

Exemplo 31 *Considere o autômato G_{31} mostrado na Figura 5.18, cuja linguagem é dada por $L_{31} = \overline{a[f(d_1ec_1 + bd_2c_2)gg^* + b(ee^* + ga^*c_3fgb^*)]}$, sendo $\Sigma_o = \{a,b,c_1,c_2,c_3,d_1, d_2,e,g\}$, $\Sigma_{uo} = \{f\}$, $\Sigma_c = \{c_1,c_2,c_3\}$ e $\Sigma_f = \{f\}$. O conjunto de cadeias proibidas após a falha f é dada por $\Phi = \{g\}$ e a linguagem ilegal é $\mathscr{K}_f = \{(d_1ec_1gg^*, bd_2c_2gg^*, gb^*\}$.*

Figura 5.18 – Exemplo de controlabilidade segura pela diagnose ou prognose. Autômato G_{31}.



Fonte: (Autor.)

Para analisar a controlabilidade segura deve-se verificar se a ocorrência do evento de falha f em qualquer cadeia $s \in \Psi_{L_{31}}(f)$ é controlável segura pela diagnose ou prognose.

Existem duas cadeias $s \in \Psi_{L_{31}}(f)$, isto é, $s_1 = af$ e $s_2 = abga^*c_3f$. Iniciando a análise pela controlabilidade segura pela diagnose na cadeia s_1, tem-se que s_1 é diagnosticável segura para $s_1t'_{c_1} = afd_1(t'_{c_1} = d_1)$ e $\overline{t'_{c_1}} \cap \mathscr{K}_f = \emptyset$ e para $s_1t''_{c_1} = afb(t''_{c_1} = b)$ e $\overline{t''_{c_1}} \cap \mathscr{K}_f = \emptyset$, atendendo a condição $(D10_1)$. A cadeia s_1 é controlável segura pela diagnose, pois atende também a segunda condição, ou seja, existem os eventos controláveis c_1 e c_2 que podem evitar a ocorrência do evento ilegal g, satisfazendo assim a condição $(D10_2)$. A cadeia s_2 não é diagnosticável segura para $s_2t_{c2} = abga^*c_3fg(t_{c2} = g)$, uma vez que $t_{c2} = g$ é a menor cadeia que assegura a diagnose e $\overline{t_{c2}} \cap \mathscr{K}_f \neq \emptyset$. Portanto, s_2 não é controlável segura pela diagnose. A cadeia s_1 não é controlável segura pela prognose, pois não sendo prognosticável não satisfaz a condição $(D12_1)$. Já a cadeia s_2 é controlável segura pela prognose, pois após a observação da cadeia abg tem-se certeza da futura ocorrência da falha (prognose), atendendo a condição $(D12_1)$ e após essa cadeia existe um evento controlável c_3 que pode ser desabilitado para evitar que ocorra o evento proibido g, satisfazendo também a condição $(D12_2)$. Portanto a linguagem L_{31} é DP-Controlável Segura, ou seja, controlável segura pela diagnose através da cadeia s_1 e controlável segura pela prognose pela cadeia s_2.

A seguir, introduz-se condições necessárias e suficientes para a controlabilidade segura de um SED.

Teorema 8 *(Condições para Controlabilidade Segura de uma Linguagem pela Diagnose ou Prognose). Considere uma linguagem diagnosticável L e um autômato $G = (Q, \Sigma, \delta, q_0)$ que gera L. Seja $G^c_{sd} = (Q_{sd}, \Sigma_o, \delta_{sd}, q_{sd,0})$ o c-diagnosticador seguro construído a partir de G. A linguagem L é DP-Controlável Segura em relação à P_o, f, e Φ se e somente se para toda a cadeia $s \in \Psi_L(f)$ pelo menos uma das seguintes condições é atendida:*
$(T8_1)$ *Condições para Controlabilidade Segura de uma Cadeia pela Diagnose: $\forall q_{sd} \in FC(s)$ (i) $q_{sd} \notin Q^B$ e (ii) $\nexists w_o \in \Sigma^*_{uc} : \hat{\delta}_{sd}(q_{sd}, w_o) = q_{sd,B}$, sendo $q_{sd,B} \in FB(s)$;*
$(T8_2)$ *Condições para Controlabilidade Segura de uma Cadeia pela Prognose: (i) A condição $\mathscr{C}$ é atendida para o estado $q_{sd} = FU(s)$ e (ii) para $q'_{sd} = FP(s)$, $\nexists w_o \in \Sigma^*_{uc} : \hat{\delta}_{sd}(q'_{sd}, w_o) = q_{sd,B}$, sendo que $q_{sd,B} \in FB(s)$.*

Prova. A prova é em duas partes.
($\Rightarrow$) A condição necessária é provada por contradição. Suponha que a linguagem L é *DP-Controlável Segura* em relação a P_o, f e Φ, mas existe uma cadeia $s \in \Psi_L(f)$ para a qual ambas condições $(T8_1)$ e $(T8_2)$ não são atendidas. Se a condição $(T8_1)$ não é satisfeita para s, pela Proposição 7 pode-se concluir que a ocorrência de f em s não é controlável segura pela diagnose. Além disso, se a condição $(T8_2)$ não é atendida para essa mesma

cadeia $s \in \Psi_L(f)$, pela Proposição 10 sabe-se que a ocorrência de f em s não é controlável segura pela prognose. Assim, existe uma cadeia $s \in \Psi_L(f)$ cuja ocorrência de f não é controlável segura nem pela diagnose e nem pela prognose. Portanto, pela Definição 13 a linguagem L não é *DP-Controlável Segura*, o que viola a hipótese inicial.
($\Leftarrow$) Agora, prova-se que se $(T8_1)$ ou $(T8_2)$ são atendidos para todas as cadeias $s \in \Psi_L(f)$, então L é *DP-Controlável Segura*. Esta parte da prova também é feita por contradição. Suponha que L não é *DP-Controlável Segura*, mas $(T8_1)$ ou $(T8_2)$ é atendida para toda cadeia $s \in \Psi_L(f)$. Pela Definição 13, sabe-se que se L não é *DP-Controlável Segura*, então a ocorrência do evento f numa cadeia $s \in \Psi_L(f)$ não é nem controlável segura pela diagnose nem controlável segura pela prognose. Pela Proposição 7, se a ocorrência de f em s não é controlável segura pela diagnose, então a condição estabelecida em $(T8_1)$ não é atendida. Além disso, se a ocorrência de f na mesma cadeia $s \in \Psi_L(f)$ não é controlável segura pela prognose, pela Proposição 10 sabe-se que condições estabelecidas em $(T8_2)$ não são satisfeitas. Assim, existe uma cadeia $s \in \Psi_L(f)$ para qual nem $(T8_1)$ nem $(T8_2)$ é atendida, contrariando a hipótese inicial. □

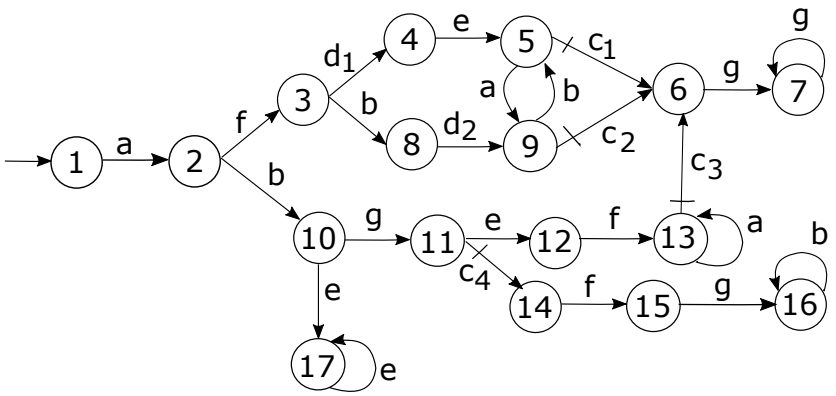
Em palavras, uma linguagem L é controlável segura pela diagnose ou prognose se e somente se cada cadeia $s \in \Psi_L(f)$ atender condições para controlabilidade segura numa cadeia pela diagnose ou condições para controlabilidade segura numa cadeia pela prognose.

A seguir, são apresentados dois exemplos para ilustrar as condições estabelecidas no Teorema 8.

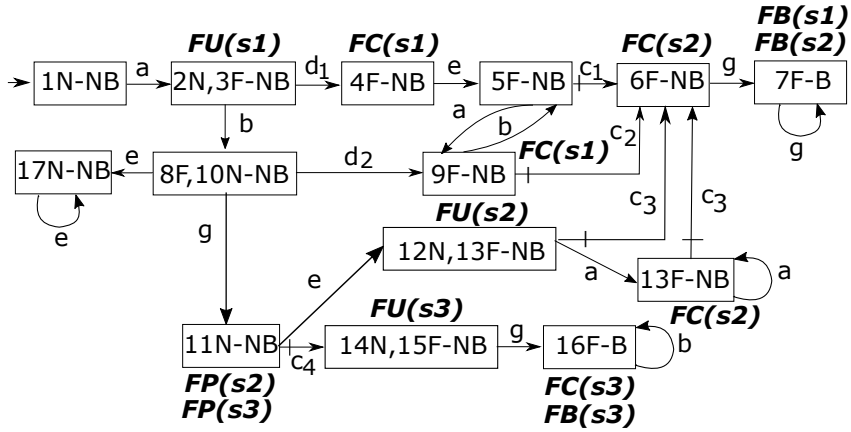
Exemplo 32 *Considere o autômato G_{32} mostrado na Figura 5.19 (a), cuja linguagem é dada por $L_{32} = \overline{a[f(d_1e(ab)^*(c_1+ac_2)+bd_2(ba)^*(c_2+bc_1))gg^*+b(ee^*+g(efa^*c_3gg^*+\overline{c_4fgb^*))]}}$, sendo $\Sigma_{uo}=\{f\}$, $\Sigma_o=\{a,b,c_1,c_2,c_3,c_4,d_1,d_2,e,g\}$, $\Sigma_c=\{c_1,c_2,c_3,c_4\}$ e $\Sigma_f=\{f\}$. O conjunto de cadeias proibidas após a falha f é dado por $\Phi=\{g\}$ e a linguagem ilegal é $\mathscr{K}_f=\{d_1e(ab)^*c_1gg^*, d_1e(ab)^*ac_2gg*, bd_2(ba)^*c_2gg^*, bd_2(ba)^*bc_1gg^*, gb^*, a^*c_3gg^*\}$.*

Existem três cadeias $s \in \Psi_{L_{32}}(f)$ em L_{32}, ou seja, $s_1 = af$, $s_2 = abgef$ e $s_3 = abgc_4f$. Em análise ao autômato da planta, pode-se perceber que, a cadeia s_1 é controlável segura pela diagnose, uma vez que ao observar as cadeias ad_1 ou abd_2 pode-se concluir sobre a ocorrência da falha e que após estas observações pode-se desabilitar, respectivamente, os eventos controláveis c_1 e c_2, impedindo a ocorrência do evento proibido g. Por outro lado, essa mesma cadeia não é prognosticável e, dessa forma, também não é controlável segura pela prognose. A cadeia s_2 é diagnosticável, mas não é controlável segura pela diagnose uma vez que a diagnose se dá após a observação da cadeia $abgec_3$, a partir da qual não se pode mais impedir a ocorrência do evento g. Já a cadeia s_3 não é diagnosticável

Figura 5.19 – Exemplo para ilustrar a análise de condição de controlabilidade segura de uma linguagem pela diagnose ou prognose. (a) Autômato G_{32}; (b) Autômato c-diagnosticador seguro G^c_{sd32} ilustrando $FC(s)$, $FU(s)$, $FP(s)$ e $FB(s)$.



(a)



(b)

Fonte: (Autor.)

segura, pois para que se tenha a diagnose é preciso observar a cadeia $abgc_4g$, de modo que o evento proibido g precisa ocorrer depois da falha para que se tenha a diagnose. Por outro lado, as cadeias s_2 e s_3 são controláveis seguras pela prognose, uma vez que após observar a sequência abg já se pode garantir a futura ocorrência de falha numa dessas cadeias e que em s_2 pode-se desabilitar o evento controlável c_3 e em s_3 pode-se desabilitar o evento controlável c_4 de modo a evitar a ocorrência do evento g após a falha. Assim, embora a linguagem L_{32} não seja, toda ela, controlável segura pela diagnose (não satisfaz as condições da Proposição 7) nem tampouco seja (toda ela) controlável segura pela prognose (não satisfaz as condições da Proposição 9), é controlável segura pela diagnose ou prognose, de acordo com o Teorema 8.

Agora, vamos fazer a análise através do c-diagnosticador seguro G^c_{sd32} mostrado na Figura 5.19 (b). Conforme discutido anteriormente, existem três cadeias $s \in \Psi_{L_{32}}(f)$ em L_{32}. Para essas cadeias, têm-se as funções que mapeiam os primeiros estados certos

$FC(s_1) = \{(4F - NB), (9F - NB)\}$, $FC(s_2) = \{(6F - NB), (13F - NB)\}$, $FC(s_3) = \{(16F - B)\}$, as funções que mapeiam os primeiros estados incertos $FU(s_1) = (2N, 3F - NB)$, $FU(s_2) = (12N, 13F - NB)$, $FU(s_3) = (14N, 15F - NB)$, as funções que mapeiam o primeiro estado que assegura prognose $FP(s_2) = FP(s_3) = (11N - NB)$, e finalmente as funções que mapeiam o primeiro mau estado $FB(s_1) = FB(s_2) = \{(7F - B)\}$ e $FB(s_3) = \{(16F - B)\}$.

A cadeia s_1 é controlável segura pela diagnose, pois todos os estados pertencentes ao conjunto $FC(s_1)$ não sejam maus estados, ou seja, $qsd \notin Q^B$ e não existem cadeias formadas unicamente por eventos não controláveis separando um estado qualquer de $FC(s_1)$ de um estado de $FB(s_1)$, satisfazendo a condição $(T8_1)$. Embora, a cadeia s_2 seja diagnosticável segura, ela não é controlável segura pela diagnose, pois não existe um evento controlável entre o estado $(6F - NB) \in FC(s_2)$ e $(7F - B) \in FB(s_2)$, portanto, não satisfazendo a condição $(T8_1)$. A cadeia s_3 não é controlável segura pela diagnose, pois ela não é diagnosticável segura, pois o estado $(16F - B) \in FC(s_3)$ é um mau estado, contrariando a condição de $(T8_1)$.

A cadeia s_1 não é controlável segura pela prognose, pois a condição $\mathscr{C}$ não é atendida para o estado $(2N, 3F - NB) = FU(s_1)$, assim não satisfazendo a condição $(T8_2)$. A cadeia s_2 é controlável segura pela prognose, uma vez que: (i) A condição $\mathscr{C}$ é atendida para o estado $(12N, 13F - NB) = FU(s_2)$; e (ii) não existe uma cadeia de eventos não controláveis que interligue o estado $(11N - NB) \in FP(s_2)$ ao estado $(7F - B) \in FB(s_2)$, satisfazendo a condição $(T8_2)$. De forma análoga, a cadeia s_3 é controlável segura pela prognose, uma vez que: (i) A condição $\mathscr{C}$ é atendida para o estado $(14N, 15F - NB) = FU(s_3)$; e (ii) não existe uma cadeia de eventos não controláveis que interligue o estado $(11N - NB) \in FP(s_3)$ ao estado $(16F - B) \in FB(s_3)$, satisfazendo a condição $(T8_2)$. Portanto, pode-se concluir que a linguagem L_{32} é DP-Controlável Segura, de acordo com o Teorema 8.

A seguir, revisita-se o Exemplo 27 da Figura 5.9 a fim de ilustrar condições para a DP-Controlabilidade Segura.

Existem duas cadeias $s \in \Psi_{L_{27}}(f)$ em L_{27}, ou seja, $s_1 = af$, $s_2 = bac_3f$. Esse exemplo já foi apresentado para análise da controlabilidade segura em cadeia pela diagnose e pela prognose, separadamente. A conclusão foi que a cadeia s_1 é controlável segura pela diagnose e não é controlável segura pela prognose. A cadeia s_2 não é controlável segura pela diagnose, enquanto é controlável segura pela prognose. Assim, a linguagem L_{27} é controlável segura pela diagnose através da cadeia s_1 e controlável segura pela prognose através da cadeia s_2. Portanto, pode-se concluir que a linguagem L_{27} é DP-Controlável Segura, de acordo com o Teorema 8.

5.11 DISCUSSÃO SOBRE O USO DA CONTROLABILIDADE SEGURA PARA FINS DE CONTROLE TOLERANTE A FALHAS

De acordo com Jiang e Yu (2012), o principal propósito de um Sistema de Controle Tolerante a Falhas (SCTF) é preservar a estabilidade de um sistema como um todo para manter um nivel aceitável do sistema com falhas. O problema de CTF tem várias abordagens (DARABI; JAFARI; BUCZAK, 2003), (ROHLOFF, 2005), (DUMITRESCU et al., 2007), (YANG; JIANG; COCQUEMPOT, 2010) e (RADEL; MULAHUWAISH; LEDUC, 2015). O trabalho desenvolvido por Paoli, Sartini e Lafortune (2011) trata do problema de CTF em SEDs através de duas abordagens: passiva (CTFP) e ativa (CTFA). A abordagem passiva objetiva encontrar um único controlador geral que satisfaça as especificações de controle tanto na operação nominal como após a ocorrência de falhas. Este controlador assegura que o sistema de malha fechada permaneça insensível a certas falhas usando técnicas de controle robusto sem a intervenção do sistema de diagnóstico de falhas. Por outro lado, na abordagem ativa adapta-se a lei de controle ao comportamento faltoso do sistema para atingir os objetivos de controle. Nesse trabalho, os autores mostram como este problema pode ser resolvido usando uma arquitetura de controle que se utiliza de um tipo especial de diagnosticador, denominado de *Diagnosing-Controller*, o qual é composto de um diagnosticador G_d e um conjunto de supervisores. Este diagnosticador é usado para detectar falhas e chavear entre o supervisor de controle nominal e um banco de supervisores de controle reconfigurados utilizando as informações *online* do sistema de diagnóstico de falhas. Com base na arquitetura de CTF proposta por Paoli, Sartini e Lafortune (2011), apresentou-se uma solução para o caso de linguagens controláveis seguras pela prognose (WATANABE et al., 2017a). Nesse caso, o elemento responsável pela prognose e pelas reconfigurações de controle é chamado de Prognosticador-Controlador (ou simplesmente *P-Controller*). Nesse caso, analogamente ao que foi proposto por (PAOLI; SARTINI; LAFORTUNE, 2011), ao entrar em um estado $q_j \in \mathcal{FP}$, o *P-Controller* deve chavear do Supervisor nominal para o Supervisor degradado, a fim de atender a especificação pós-prognose $\mathscr{K}^{deg}$ ao apropriado Supervisor degradado S^{deg}. As vantagens de usar o *P-Controller* são consequências diretas da inferência sobre a ocorrência futura de uma falha, em contraste com a diagnose que ocorre após a ocorrência da falha. Para os casos em que a linguagem não é (toda ela) controlável segura pela diagnose e também não é controlável segura pela prognose, mas é DP-Controlável Segura, propõe-se a utilização de um $''DiagnosticadorPrognosticador-Controlador''$ ou simplesmente *DP-Controller*.

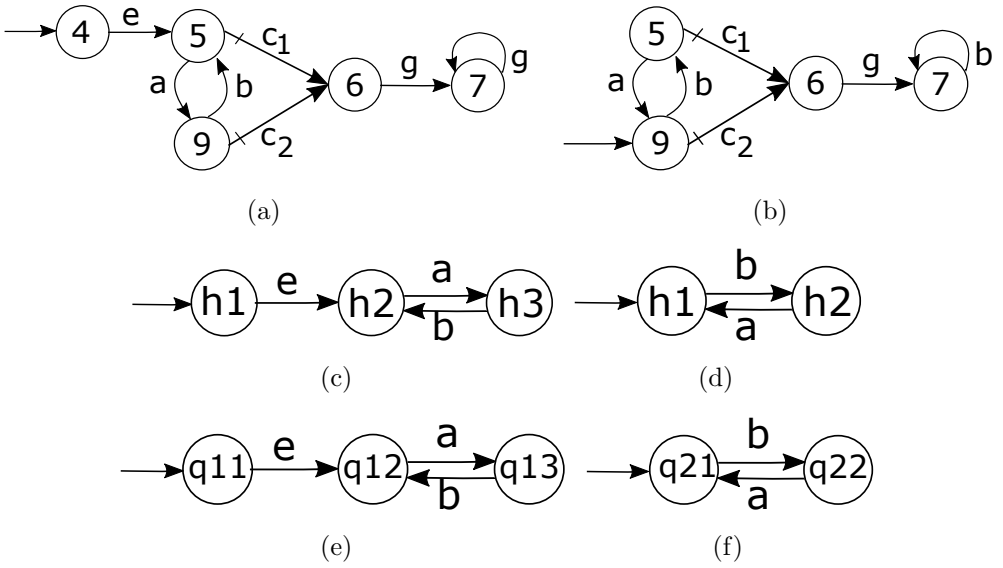
Ilustra-se a seguir, a partir do Exemplo 32 da Figura 5.19, como se pode obter o *DP-Controller*. Vale destacar que para tal obtenção seguem-se os passos descritos em (PAOLI; SARTINI; LAFORTUNE, 2011) e em (WATANABE et al., 2017a). Assim, para informações mais detalhadas o leitor pode consultar esses trabalhos. Para cada cadeia s que é controlável segura pela diagnose, monta-se um autômato $G_1^{deg,d}$ para cada estado $q_{sd,i} \in FC(s)$. Na Figura 5.20(a) ilustra-se o autômato para $G_1^{deg,d}$, o qual consiste na componente alcançável de G_{32}^{n+f} a partir do estado (4), sendo este relativo ao estado $(4F-NB) \in FC(s1)$. Na Figura 5.20(b) ilustra-se o autômato para $G_2^{deg,d}$, o qual consiste na componente alcançável de G_{32}^{n+f} a partir do estado (9), o qual é relativo ao estado $(9F-NB) \in FC(s1)$. A partir de $G_1^{deg,d}$, pode-se estabelecer uma especificação para o comportamento degradado, o que é feito pela linguagem $\mathscr{K}_1^{deg}$ e representado pelo autômato $H_1^{deg,d}$ mostrado na Figura 5.20 (c). O produto $H_1^{deg,d}$ x $G_1^{deg,d}$ resulta no autômato $R_1^{deg,d}$, que é uma especificação que tem por objetivo proibir alguns eventos da linguagem $\mathscr{K}_1^{deg}$. Nesse exemplo, o autômato $R_1^{deg,d}$ é igual ao autômato de $H_1^{deg,d}$, que por sua vez é igual ao Supervisor pós-falha $S_1^{deg,d}$ mostrado na Figura 5.20(e).

Analogamente, a partir de $G_2^{deg,d}$, pode-se estabelecer uma especificação para o comportamento degradado, o que é feito pela linguagem $\mathscr{K}_2^{deg}$ e representada pelo autômato $H_2^{deg,d}$ mostrado na Figura 5.20 (d). O produto $H_2^{deg,d}$ x $G_2^{deg,d}$ resulta no autômato $R_2^{deg,d}$ que é uma especificação que tem por objetivo proibir alguns eventos da linguagem $\mathscr{K}_2^{deg}$. Nesse exemplo, o autômato $R_2^{deg,d}$ é igual ao autômato de $H_2^{deg,d}$, que por sua vez é igual ao Supervisor pós-falha $S_2^{deg,d}$ mostrado na Figura 5.20(f).

De forma, análoga ao que foi feito para as cadeias controláveis seguras pela diagnose, para cada cadeia s que é controlável segura pela prognose, monta-se um autômato $G_j^{deg,p}$ para o estado $q_{sd,j} = FP(s)$. Assim, para a cadeia s_3, o autômato $G_3^{deg,p}$ é obtido levando em consideração a parte acessível do autômato G_{32}^{n+f} apresentado na Figura 5.21(a), a partir do estado (11), que corresponde ao estado $(11N-NB) = FP(s_2)$. Vale destacar que como $FP(s_2) = FP(s_3)$, não é necessário construir um modelo degradado a partir de s_3. A partir de $G_3^{deg,p}$, pode-se estabelecer uma especificação para o comportamento degradado, o que é feito pela linguagem $\mathscr{K}_3^{deg}$ e representado pelo autômato $H_3^{deg,p}$ ilustrado na Figura 5.21 (b). O produto $H_3^{deg,p}$ x $G_3^{deg,p}$ resulta no autômato $R_3^{deg,p}$ que é uma especificação que tem por objetivo proibir alguns eventos da linguagem $\mathscr{K}_3^{deg}$. Nesse exemplo, o autômato $R_3^{deg,p}$ é igual ao autômato de $H_3^{deg,p}$, que por sua vez é igual ao Supervisor pós-falha $S_3^{deg,p}$ ilustrado na Figura 5.21(c).

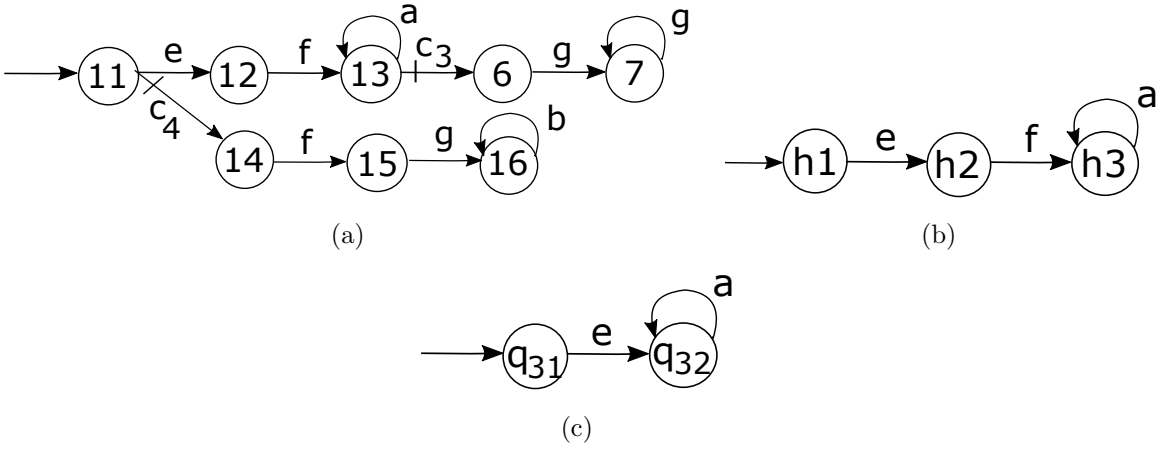
A parte controlável segura pela diagnose do *DP-Controller* ilustrado na Figura 5.22 pode ser construída a partir do diagnosticador da Figura 5.19 (b), sendo que a sua construção é interrompida no estado $(4F-NB) \in FC(s_1)$ e também no estado $(9F-NB) \in FC(s_2)$

Figura 5.20 – Exemplo 32 - CTFA utilizando controlabilidade Segura (parte da diagnose). (a) Autômato da planta degradada pós-falha $G_1^{deg,d}$; (b) Autômato da planta degradada pós-falha $G_2^{deg,d}$; (c) Autômato da especificação nominal $H_1^{deg,d}$; (d) Autômato da especificação nominal $H_2^{deg,d}$; (e) Autômato do supervisor degradado pós-falha $S_1^{deg,d}$; (f) Autômato do supervisor degradado pós-falha $S_2^{deg,d}$.



Fonte: (Autor.)

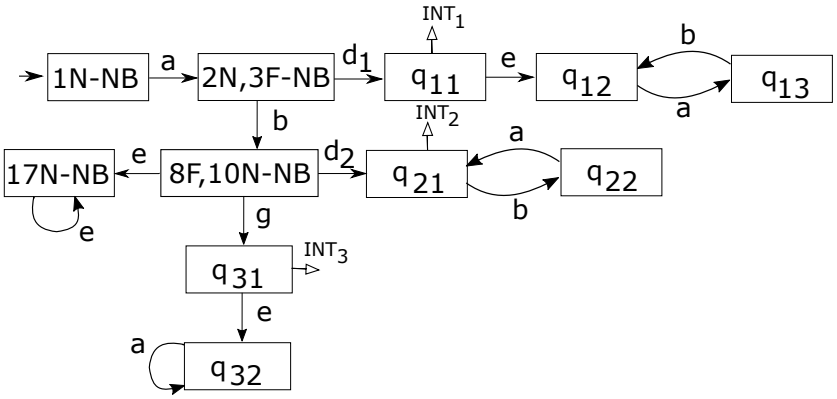
Figura 5.21 – Exemplo 32 - CTFA utilizando controlabilidade Segura (parte da prognose). (a) Autômato da planta degradada pós-falha $G_3^{deg,p}$; (b) Autômato da especificação nominal $H_3^{deg,p}$; (c) Autômato do supervisor degradado pós-falha $S_3^{deg,p}$.



Fonte: (Autor.)

convenientemente sobrepostos a estes estados os supervisores pós-falha $S_1^{deg,d}$ e $S_2^{deg,d}$, respectivamente. É importante ressaltar que nestes estados $(4F-NB)$ e $(9F-NB)$ também os sinais INT_1 e INT_2 são habilitados, pelos quais os supervisores nominais são desabilitados e as ocorrências das ações proibidas (evento g) são evitadas.

Figura 5.22 – Exemplo 32 - *DP-Controller*.



Fonte: (Autor.)

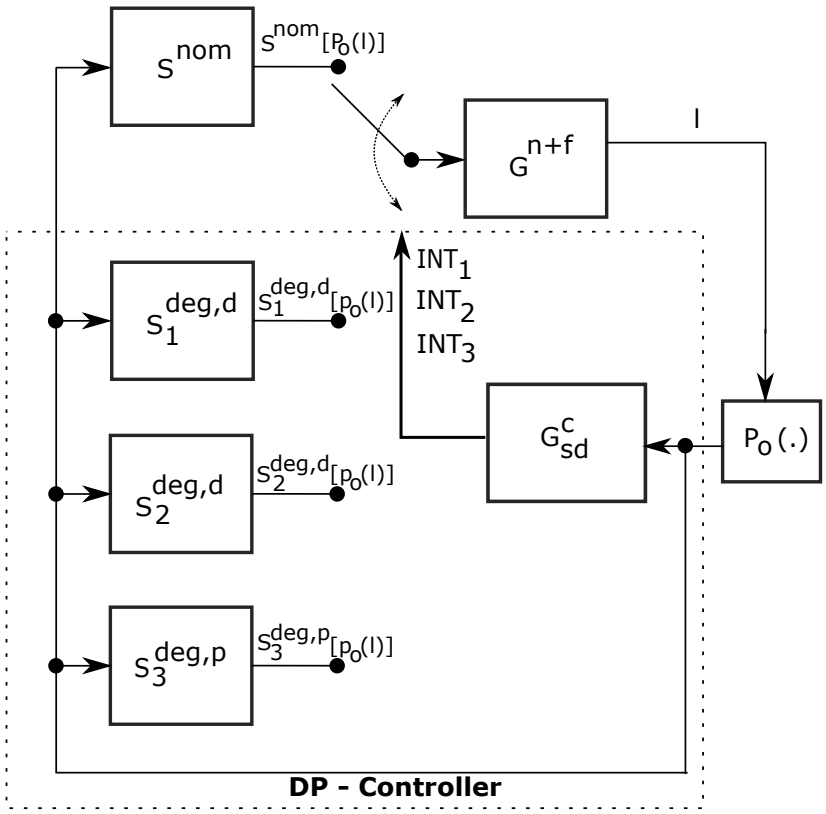
A parte controlável segura pela prognose do *DP-Controller*, pode ser construída a partir do diagnosticador da Figura 5.21 (c). A sua construção é interrompida no estado $(11N-NB)$ o qual é comum a $FP(s_2)$ e $FP(s_3)$ convenientemente sobreposto a este estado o supervisor pós-falha $S_3^{deg,p}$. É importante ressaltar que neste estado $(11N-NB)$ também o sinal INT_3 é habilitado, pelo qual o supervisor nominal é desabilitado e a ocorrência da ação proibida é evitada.

Na Figura 5.23 ilustra-se, para o exemplo em análise, a estrutura de controle ativo tolerante a falhas usando o *DP-Controller*. Nessa estrutura, se uma falha é diagnosticada pelo *DP-Controller*, ele interrompe o funcionamento normal da planta e troca o supervisor nominal pelo supervisor degradado ($S_1^{deg,d}$ ou $S_2^{deg,d}$, de acordo com o sinal INT_i). Da mesma forma, se a ocorrência da falha é prognosticada em s_2 ou s_3, então o *DP-Controller* chaveia para o supervisor $S_3^{deg,p}$, após interromper a evolução normal da planta.

5.12 CONSIDERAÇÕES FINAIS

Baseados nos conceitos de controlabilidade segura de uma linguagem para diagnose, conforme Paoli, Sartini e Lafortune (2011) e controlabilidade segura de uma linguagem pela prognose apresentado em um trabalho anterior por Watanabe et al. (2017a), nesta seção introduzimos o conceito de controlabilidade segura de uma cadeia pela diagnose e de controlabilidade segura de uma cadeia pela prognose. Finalmente, esses conceitos foram utilizados para alcançar um objetivo maior que é a introdução de uma nova abordagem de controlabilidade segura que combina a diagnose e a prognose de falhas online. Pode-se observar que uma linguagem pode não ser controlável segura pela diagnose e pode também não ser controlável segura pela prognose, mas ainda assim essa linguagem pode ser controlável segura pela diagnose ou prognose (DP-Controlável). Desta forma, foi possível

Figura 5.23 – Exemplo 32 - Planta G^{n+f} e controlador *DP-Controller*.



Fonte: (Autor.)

trazer contribuições neste capítulo, sendo as seguintes as consideradas mais relevantes: introdução da definição de controlabilidade segura de uma cadeia pela diagnose e o estabelecimento de suas condições; introdução da definição de controlabilidade segura numa cadeia pela prognose e o estabelecimento de suas condições; e a generalização da definição de controlabilidade segura, a qual engloba os conceitos de controlabilidade segura pela diagnose e controlabilidade segura pela prognose e o estabelecimento de suas condições. A Tabela 5.2 apresenta um resumo de condições para a controlabilidade segura de uma cadeia pela diagnose e pela prognose e controlabilidade segura de uma linguagem pela diagnose ou prognose.

Tabela 5.2 – Comparativo entre as condições para a controlabilidade segura de uma cadeia pela diagnose e pela prognose e controlabilidade segura de uma linguagem pela diagnose ou prognose.

Mecanismo	Tipo do diagnosticador	Condição	Obra
Controlabilidade Segura de uma Cadeia pela Diagnose	c-diagnosticador seguro	Se e somente se $\forall q_{sd} \in FC(s)$, seguintes condições são atendidas: 1) $q_{sd} \notin Q^B$; 2) $\nexists w_o \in \Sigma_{uc}^* : \hat{\delta}_{sd}(q_{sd}, w_o) = q_{sd,B}$, sendo $q_{sd,B} \in FB(s)$.	Este trabalho
Controlabilidade Segura de uma Cadeia pela Prognose	c-diagnosticador seguro	Se e somente se as seguintes condições são atendidas: 1) A condição $\mathscr{C}$ é atendida para o estado $q_{sd} = FU(s)$; 2) para $q'_{sd} = FP(s)$, $\nexists w_o \in \Sigma_{uc}^* : \hat{\delta}_{sd}(q'_{sd}, w_o) = q_{sd,B}$, sendo que $q_{sd,B} \in FB(s)$.	Este trabalho
Controlabilidade Segura de uma Linguagem pela Diagnose ou Prognose	c-diagnosticador seguro	Se e somente se para toda a cadeia $s \in \Psi_L(f)$ pelo menos uma das seguintes condições é atendida: (1) $\forall q_{sd} \in FC(s)$ (i) $q_{sd} \notin Q^B$ e (ii) $\nexists w_o \in \Sigma_{uc}^* : \hat{\delta}_{sd}(q_{sd}, w_o) = q_{sd,B}$, sendo $q_{sd,B} \in FB(s)$; (2) (i) A condição $\mathscr{C}$ é atendida para o estado $q_{sd} = FU(s)$ e (ii) para $q'_{sd} = FP(s)$, $\nexists w_o \in \Sigma_{uc}^* : \hat{\delta}_{sd}(q'_{sd}, w_o) = q_{sd,B}$, sendo que $q_{sd,B} \in FB(s)$.	Este trabalho

Fonte: (Autor.)

6 CONCLUSÃO E TRABALHOS FUTUROS

Objetivou-se neste trabalho resolver o problema da controlabilidade segura em SEDs. Basicamente, o problema consiste em evitar a ocorrência de eventos que podem levar o sistema a um comportamento proibido após a ocorrência de uma falha. Através da revisão bibliográfica sobre a diagnose, prognose de falhas e controlabilidade segura foi verificada uma lacuna na literatura na área de controlabilidade segura pela prognose. Após, introduzir a controlabilidade segura pela prognose, foi proposto neste trabalho usar a abordagem da diagnose em conjunto com a prognose para contemplar uma contribuição inédita na área. Essas duas abordagens foram concebidas no âmbito de linguagens, de forma que uma linguagem precisa ser, toda ela, controlável segura pela diagnose ou controlável segura pela prognose para que se tenha a controlabilidade segura da linguagem. Porém, se alguma ocorrência de falha não é controlável segura por uma ou outra abordagem, ou seja pela diagnose ou prognose, então a linguagem não é considerada controlável segura. Portanto, para poder ampliar a noção de controlabilidade segura, foi proposto nesta tese mudar o âmbito de linguagens para cadeias. Nesse caso, as propriedades são analisadas para cada cadeia que contém o evento de falha e não sobre o conjunto de cadeias. Para alcançar os objetivos propostos são introduzidos os conceitos de cadeia diagnosticável, cadeia diagnosticável segura, cadeia prognosticável, cadeia controlável segura pela diagnose, cadeia controlável segura pela prognose. São apresentadas também condições necessárias e suficientes para garantir tais propriedades. A partir desses conceitos, define-se a controlabilidade segura de uma linguagem pela diagnose ou prognose, denominada de DP-Controlabilidade Segura. Nesse caso, uma linguagem é DP-Controlável Segura se cada uma das cadeias que contém a falha é controlável segura pela diagnose ou é controlável segura pela prognose. Dessa forma, uma linguagem que não é, toda ela, controlável segura por uma mesma abordagem de detecção de falha (diagnose ou prognose), pode ser DP-Controlável Segura. E então, condições para uma linguagem DP-Controlável Segura são apresentadas.

A seguir, são apresentadas as contribuições ao longo dos capítulos para alcançar os objetivos propostos: No capítulo 3 foram introduzidos os conceitos de diagnose e diagnose segura em cadeias, bem como o estabelecimento de condições que garantem a diagnose e a diagnose segura numa cadeia. Já no Capítulo 4 foi introduzido o conceito da prognosticabilidade numa cadeia e o estabelecimento de condições que asseguram que a ocorrência do evento de falha numa cadeia seja prognosticável. Após, foram estabelecidas condições necessárias e suficientes para cadeia controlável segura pela prognose. No Capítulo 5 fo-

ram introduzidos os conceitos de controlabilidade segura de uma cadeia pela diagnose e o estabelecimento de suas condições; introdução da definição de controlabilidade segura numa cadeia pela prognose e o estabelecimento de suas condições; e finalmente a generalização da definição de controlabilidade segura, ou seja, uma linguagem controlável segura pela diagnose ou prognose, e o estabelecimento de suas condições.

A decisão a ser tomada após obter a informação que uma linguagem é controlável segura pela diagnose ou pela prognose será de acordo com a necessidade do sistema. Muitas vezes, é importante para propósitos de controle ter controlabilidade segura pela prognose o mais rápido possível, a fim de se obter mais opções de controle para evitar áreas proibidas.

Nesta tese, como ponto de partida, considerou-se a controlabilidade segura no âmbito da abordagem monolítica. A partir dos resultados obtidos, acredita-se que um campo fértil para pesquisas seja a extensão dessa abordagem para sistemas distribuídos. Nesse caso, sugere-se a investigação de uma estratégia de controle modular local tolerante a falhas que se utilize da diagnose e prognose online também distribuídas. Outro possível trabalho seria a verificação das condições para prognosticabilidade utilizando verificadores. Além disso, as contribuições para trabalhos futuros nesta área poderiam ser o estabelecimento de um procedimento formal para obtenção do *DP-Controller* para arquitetura de controle tolerante a falha, o desenvolvimento de algoritmos para automatizar a obtenção das funções FP(s), FC(s), FU(s) e FB(s), e a implementação desses numa ferramenta computacional como o Nazdoru (PINHEIRO et al., 2015). Um grande desafio também é buscar uma aplicação real.

REFERÊNCIAS BIBLIOGRÁFICAS

AMMOUR, R.; LECLERCQ, E.; SANLAVILLE, E.; LEFEBVRE, D. Fault prognosis of timed stochastic discrete event systems with bounded estimation error. **Automatica**, 2017. v. 82, p. 35 – 41, 2017. ISSN 0005-1098.

ATHANASOPOULOU, E.; LINGXI, L.; HADJICOSTIS, C. Maximum likelihood failure diagnosis in finite state machines under unreliable observations. **IEEE Transactions on Automatic Control**, 2010. v. 55, n. 3, p. 579–593, March 2010. ISSN 0018-9286.

BASILIO, J. C.; CARVALHO, L. K.; MOREIRA, M. V. Diagnose de falhas em sistemas a eventos discretos modelados por autômatos finitos. **Revista Controle e Automação (Impresso)**, 2010. v. 5, p. 510–533, setembro 2010.

BASILIO, J. C.; LAFORTUNE, S. Robust codiagnosability of discrete event systems. **2009 American Control Conference**, 2009. p. 2202–2209, June 2009. ISSN 0743-1619.

BASILIO, J. C.; LIMA, S. T. S.; LAFORTUNE, S.; MOREIRA, M. Computation of minimal event bases that ensure diagnosability. **Discrete Event Dynamic Systems**, 2012. p. 249–292, 2012.

BENMESSAHEL, B.; TOUAHRIA, M.; NOUIOUA, F. Predictability of fuzzy discrete event systems. **Discrete Event Dynamic Systems**, 2017. v. 27, n. 4, p. 641–673, Dec 2017. ISSN 1573-7594.

BLANKE M. KINNAERT, J. L. M.; STAROSWIECKI, M. **Diagnosis and fault-tolerant control**. Berlin: Springer-Verlag, 2003.

BRIONES, L. B.; MADALINSKI, A. Bounded predictability for faulty discrete event systems. **30nd International Conference of the Chilean Computer Science Society (SCCC)**, 2011. p. 142–146, Nov 2011. ISSN 1522-4902.

BRIONES, L. B.; MADALINSKI, A. Distributed bounded predictability. **32nd International Conference of the Chilean Computer Science Society (SCCC)**, 2013. p. 95–99, Nov 2013. ISSN 1522-4902.

BUSS, S.; PAPADIMITRIOU, C.; TSITSIKLIS, J. On the predictability of coupled automata: An allegory about chaos. **Complex Systems Publications**, 1991. p. 525 – 539, 1991.

CABRAL, F. G. **Diagnóstico online de falhas em Sistemas a Eventos Discretos modelados por autômatos finitos: Uma abordagem utilizando Redes de Petri.** Dissertação (Mestrado) — Programa de Pós-Graduação da Universidade Federal do Rio de Janeiro, 2014.

CAO, X. R. The predictability of discrete event systems. **IEEE Transactions on Automatic Control**, 1989. v. 34, n. 11, p. 1168–1171, Nov 1989. ISSN 0018-9286.

CARVALHO, L. K. **Diagnose robusta de sistemas a eventos discretos**. Dissertação (Tese de Doutorado) — Programa de Pós-Graduação da Universidade Federal do Rio de Janeiro, 2011.

CARVALHO, L. K.; BASILIO, J. C.; MOREIRA, M. V. Robust diagnosis of discrete event systems against permanent loss of observations. **Automatica**, 2013. v. 1, p. 223–231, 2013.

CARVALHO, L. K.; WU, Y. C.; KWONG, R.; LAFORTUNE, S. Detection and prevention of actuator enablement attacks in supervisory control systems. **13th International Workshop on Discrete Event Systems (WODES)**, 2016. p. 298–305, May 2016.

CASSANDRAS, C. G.; LAFORTUNE, S. **Introduction to Discrete Event Systems**. 2. ed. New York: Springer, 2008.

CASSEZ, F.; GRASTIEN, A. Predictability of event occurrences in timed systems. **Formal Modeling and Analysis of Timed Systems**, 2013. p. 62–76, 2013.

CHEN, J.; KUMAR, R. Failure prognosability of stochastic discrete event systems. **American Control Conference**, 2014. June 2014.

CHEN, J.; KUMAR, R. Stochastic failure prognosability of discrete event systems. **IEEE Transactions on Automatic Control**, 2015. v. 60, n. 6, p. 1570–1581, june 2015.

CLAVIJO, L. B.; BASILIO, J. C. Empirical studies in the size of diagnosers and verifiers for diagnosability analysis. **Discrete Event Dynamic Systems**, 2017. v. 27, n. 4, p. 701–739, 2017.

CONTANT, O.; LAFORTUNE, S.; TENEKETZIS, D. Diagnosability of discrete event systems with modular structure. **Discrete Event Dynamic Systems**, 2006. Kluwer Academic Publishers, v. 16, n. 1, p. 9–37, 2006. ISSN 0924-6703.

DARABI, H.; JAFARI, M.; BUCZAK, A. A control switching theory for supervisory control of discrete event systems. **IEEE Transactions on Robotics and Automation**, 2003. Institute of Electrical and Electronics Engineers Inc., v. 19, n. 1, p. 131–137, 2 2003. ISSN 1042-296X.

DEBOUK, R.; LAFORTUNE, S.; TENEKETZIS, D. Coordinated decentralized protocols for failure diagnosis of discrete event systems. **Discrete Event Dynamic Systems**, 2000. v. 10, p. 33–86, 2000.

DEBOUK, R.; MALIK, R.; BRANDIN, B. A modular architecture for diagnosis of discrete event systems. **41st IEEE Conference on Decision and Control**, 2002. p. 417–422, 2002.

DUMITRESCU, E.; GIRAULT, A.; MARCHAND, H.; RUTTEN, E. Optimal discrete controller synthesis for the modeling of fault-tolerant distributed systems. **Research Report, RR-6137, INRIA**, 2007. p. 01–35, 2007.

FADEL, H. K.; HOLLOWAY, L. E. Using spc and template monitoring method for fault detection and prediction in discrete event manufacturing systems. 1999. p. 150–155, Sept 1999. ISSN 2158-9860.

FANTI, M. P.; MANGINI, A. M.; UKOVICH, W.; LESAGE, J. J.; VIARD, K. A petri net model of an integrated system for the health care at home management. **IEEE International Conference on Automation Science and Engineering (CASE**, 2014. p. 582–587, 2014.

GENC, S.; LAFORTUNE, S. Predictability in discrete-event systems under partial observation. **IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes**, 2006. 2006.

GENC, S.; LAFORTUNE, S. Predictability of event occurrences in partially-observed discrete-event systems. **Automatica**, 2009. Pergamon Press, Inc., Tarrytown, NY, USA, v. 45, n. 2, p. 301–311, fev. 2009. ISSN 0005-1098.

GRASTIEN, A. Interval predictability in discrete event systems. **CoRR**, 2015. abs/1508.00683, 2015.

JERON, T.; MARCHAND, H.; GENC, S.; LAFORTUNE, S. Predictability of sequence patterns in discrete event systems. **Proceedings of the 17th World Congress The International Federation of Automatic Control Seoul**, 2008. 2008.

JIANG, J.; YU, X. Fault-tolerant control systems: A comparative study between active and passive approaches. **Annual Reviews in Control**, 2012. v. 36, n. 1, p. 60 – 72, 2012. ISSN 1367-5788.

JIANG, S.; HUANG, Z.; CHANDRA, V.; KUMAR, R. A polynomial algorithm for testing diagnosability of discrete-event systems. **IEEE Transactions on Automatic Control**, 2001. p. 1318–1321, 2001.

KHOUMSI, A.; CHAKIB, H. Conjunctive and disjunctive architectures for decentralized prognosis of failures in discrete-event systems. **IEEE Trans. Autom. Sci. Eng.**, 2012. p. 412–417, 2012.

KUMAR, R.; TAKAI, S. Decentralized prognosis of failures in discrete event systems. **IEEE Transactions on Automatic Control**, 2010. v. 55, n. 1, p. 48–59, Jan 2010. ISSN 0018-9286.

LAFORTUNE, S. Discrete event systems: Modeling, observation, and control. **Annual Review of Control, Robotics, and Autonomous Systems**, 2019. v. 2, n. 1, p. 141–159, 2019.

LAFORTUNE, S.; TENEKETZIS, D.; SAMPATH, M.; SENGUPTA, R.; SINNAMOHIDEEN, K. Failure diagnosis of dynamic systems: an approach based on discrete event systems. **Proceedings of the American Control Conference**, 2001. p. 2058–2071, 2001.

LEFEBVRE, D. Probability of current state and future faults with partially observed stochastic petri nets. **2014 European Control Conference (ECC)**, 2014. p. 258–263, June 2014.

LIMA, S. T.; BASILIO, J. C.; LAFORTUNE, S.; MOREIRA, M. V. Bases mínimas para a diagnose de falhas de sistemas a eventos discretos. parte 1: Eventos essenciais para a diagnose e trajetórias primas. **XVIII Congresso Brasileiro de Automática**, 2010. 2010.

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

LIU, F.; WU, L. Decentralized safe diagnosis of fuzzy discrete-event systems. **37th Chinese Control Conference (CCC)**, 2018. p. 1970–1975, July 2018. ISSN 1934-1768.

MIYAGI, P.; RIASCOS, L. Modeling and analysis of fault-tolerant systems for machining operations based on petri nets. **Control Engineering Practice**, 2006. v. 14, n. 4, p. 397 – 408, 2006. ISSN 0967-0661.

MOREIRA, M. V.; JESUS, T. C.; BASILIO, J. C. Polynomial time verification of decentralized diagnosability of discrete event systems. **IEEE Transactions on Automatic Control**, 2011. v. 56, n. 7, p. 1679–1684, July 2011. ISSN 0018-9286.

NOUIOUA, F.; DAGUE, P.; YE, L. Probabilistic analysis of predictability in discrete event systems. **Proceedings of the 25th Edition of the International Workshop on Principles of Diagnosis**, 2014. 2014.

PAOLI, A. **Fault Detection and Fault Tolerant Control for Distributed Systems. A general framework**. Dissertação (Ph.D. Thesis) — University of Bologna, 2003.

PAOLI, A.; LAFORTUNE, S. Safe diagnosability for fault-tolerant supervision of discrete-event systems. **Automatica**, 2005. v. 41, n. 8, p. 1335 – 1347, 2005. ISSN 0005-1098.

PAOLI, A.; SARTINI, M.; LAFORTUNE, S. Active fault tolerant control of discrete event systems using online diagnostics. **Automatica**, 2011. v. 47, n. 4, p. 639 – 649, 2011. ISSN 0005-1098.

PENCOLÉ, Y. Diagnosability analysis of distributed discrete event systems. **In Proc. international workshop on principles of diagnosis (DX'04)**, 2004. p. 173–178, 2004.

PINHEIRO, L. P.; LOPES, Y. K.; LEAL, A. B.; ROSSO, R. S. U. Nadzoru: A software tool for supervisory control of discrete event systems. **5th IFAC International Workshop On Dependable Control of Discrete Systems**, 2015. p. 182–187, 2015.

QIU, W.; KUMAR. Decentralized failure diagnosis of discrete event systems. **IEE Transactions on Systems,Man and Cybernetics Part A: Systems and Humans**, 2006. p. 384–395, 2006.

RADEL, S.; MULAHUWAISH, A.; LEDUC, R. J. Fault tolerant controllability. **2015 American Control Conference (ACC)**, 2015. p. 1603–1610, July 2015. ISSN 0743-1619.

ROHLOFF, K. R. Sensor failure tolerant supervisory control. **44th IEEE Conference on Decision and Control, and the European Control Conference**, 2005. 2005.

SAMPATH, M.; LAFORTUNE, S.; TENEKETZIS, D. Active diagnosis of discrete-event systems. **IEEE Transactions on Automatic Control**, 1998. v. 43, p. 908–929, september 1998.

SAMPATH, M.; SENGUPTA, R.; LAFORTUNE, S.; SINNAMOHIDEEN, K.; TENEKETZIS, D. C. Diagnosticability of discrete-event models. **IEEE Transactions on Automatic Control**, 1995. v. 40, n. 9, p. 1555–1575, september 1995.

SAMPATH, M.; SENGUPTA, R.; LAFORTUNE, S.; SINNAMOHIDEEN, K.; TENEKETZIS, D. C. Failure diagnosis using discrete-event models. **IEEE Transactions on Control Systems Technology**, 1996. v. 4, n. 2, p. 105–124, march 1996.

SHENGBING, J.; KUMAR, R. Failure diagnosis of discrete event systems with linear-time temporal logic specifications. **IEEE Transactions on Automatic Control**, 2004. v. 49, p. 934–945, june 2004.

SU, R.; WONHAM, W. A model of component consistency in distributed diagnosis. **In Proc. 7th international workshop on discrete event systems (WODES'04)**, 2004. p. 427–432, 2004.

TAKAI, S. Robust failure diagnosis of partially observed discrete event systems. **10th IFAC Workshop on Discrete Event Systems**, 2010. v. 43, n. 12, p. 205 – 210, 2010. ISSN 1474-6670.

TAKAI, S. Robust prognosability for a set of partially observed discrete event systems. **Automatica**, 2015. v. 51, p. 123 – 130, 2015. ISSN 0005-1098.

TAKAI, S.; KUMAR, R. Distributed prognosis of discrete event systems under bounded-delay communications. **Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference**, 2009. p. 1235–1240, Dec 2009. ISSN 0191-2216.

TAKAI, S.; KUMAR, R. Distributed failure prognosis of discrete event systems with bounded-delay communications. **IEEE Transactions on Automatic Control**, 2012. v. 57, n. 5, p. 1259–1265, May 2012. ISSN 0018-9286.

TAKAI, S.; KUMAR, R. A generalized inference-based prognosis framework for discrete event systems. **IFAC-PapersOnLine**, 2017. v. 50, n. 1, p. 6819 – 6824, 2017. ISSN 2405-8963. 20th IFAC World Congress.

THORSLEY, D.; YOO, T.; GARCIA, H. Diagnosability of stochastic discrete event systems under unreliable observations. **American control conference**, 2008. p. 1158–1165, 2008.

VIANA, G. S.; BASILIO, J. C. Codiagnosability of discrete event systems revisited: A new necessary and sufficient condition and its applications. **Automatica**, 2019. v. 101, p. 354 – 364, 2019. ISSN 0005-1098.

WANG, Y.; YOO, T.; LAFORTUNE, S. Diagnosis of discrete event systems using decentralized architectures. **Discrete Event Dynamic Systems**, 2007. p. 233–263, June 2007.

WATANABE, A. T.; LEAL, A. B.; CURY, J. E.; QUEIROZ, M. H. de. Safe controllability using online prognosis. **IFAC-PapersOnLine**, 2017. v. 50, n. 1, p. 12359 – 12365, 2017. ISSN 2405-8963. 20th IFAC World Congress.

WATANABE, A. T. Y.; LEAL, A. B.; MOREIRA, B. G.; CURY, J. E. R.; QUEIROZ, M. H. Análise das condições para diagnosticabilidade e prognosticabilidade de falhas. **XIII Simposio Brasileiro de Automaçao Inteligente Porto Alegre – RS**, 2017. n. 577, p. 1968–1975, Outubro 2017.

YANG, H.; JIANG, B.; COCQUEMPOT, V. **Fault Tolerant Control Design for Hybrid Systems**. Berlin: Springer-Verlag, 2010.

YE, L.; DAGUE, P.; NOUIOUA, F. Predictability analysis of distributed discrete event systems. **52nd IEEE Conference on Decision and Control**, 2013. p. 5009–5015, Dec 2013. ISSN 0191-2216.

YIN, X. Verification of prognosability for labeled petri nets. **IEEE Transactions on Automatic Control**, 2018. v. 63, n. 6, p. 1828–1834, June 2018. ISSN 0018-9286.

YIN, X.; CHEN, J.; LI, Z.; Li, S. Robust fault diagnosis of stochastic discrete event systems. **IEEE Transactions on Automatic Control**, 2019. p. 1–1, 2019. ISSN 0018-9286.

YIN, X.; LI, Z. Decentralized fault prognosis of discrete event systems with guaranteed performance bound. **Automatica**, 2016. v. 69, p. 375 – 379, 2016. ISSN 0005-1098.

YOKOTANI, M.; TAKAI, S. Abstraction-based verification for partially observed discrete event systems. **IFAC Proceedings Volumes**, 2014. v. 47, n. 2, p. 356 – 361, 2014. ISSN 1474-6670. 12th IFAC International Workshop on Discrete Event Systems (2014).

YOO, T.; LAFORTUNE, S. Polynomial-time verification of diagnosability of partially observed discrete event systems. **IEEE Transactions on Automatic Control**, 2002. p. 1491–1495, 2002.

ZAYTOON, J.; LAFORTUNE, S. Overview of fault diagnosis methods for discrete event systems. **Elsevier Annual Reviews in Control**, 2013. p. 308–320, 2013.

ZHAO, R.; LIU, F.; LIU, Z. Relative diagnosability of discrete-event systems and its opacity-based test algorithm. **International Journal of Control, Automation and Systems**, 2017. v. 15, n. 4, p. 1693–1700, Aug 2017. ISSN 2005-4092.

ZHOU, C.; KUMAR, R.; SREENIVAS, R. S. Decentralized modular diagnosis of concurrent discrete event systems. **9th Workshop on Discrete Event Systems**, 2008. p. 388–393, 2008.