UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE TECNOLÓGICO
PROGRAMA DE PÓS-GRADUA-ÇÃO EM ENGENHARIA DE AUTOMAÇÃO E SISTEMAS

Guilherme Teixeira Araújo

Diagnóstico Síncrono Descentralizado de Sistemas a Eventos Discretos Sujeito a Atrasos de Comunicação de Eventos

Florianópolis
2021

Guilherme Teixeira Araújo

Diagnóstico Síncrono Descentralizado de Sistemas a Eventos Discretos Sujeito a Atrasos de Comunicação de Eventos

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina para a obtenção do título de Mestre em Engenharia de Automação e Sistemas.
Orientador: Prof. Felipe Gomes de Oliveira Cabral, Dr.

Florianópolis
2021

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Teixeira Araújo, Guilherme
Diagnóstico Síncrono Descentralizado de Sistemas a Eventos Discretos Sujeito a Atrasos de Comunicação de Eventos / Guilherme Teixeira Araújo ; orientador, Felipe Gomes de Oliveira Cabral Gomes de Oliveira Cabral, 2021.
61 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Florianópolis, 2021.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2. Sistemas a eventos discretos; Diagnóstico de falhas. 3. Arquitetura decentralizada; Atraso de comunicação. 4. Verificação de diagnosticabilidade. I. Gomes de Oliveira Cabral, Felipe Gomes de Oliveira Cabral. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Automação e Sistemas. III. Título.

Guilherme Teixeira Araújo

Diagnóstico Síncrono Descentralizado de Sistemas a Eventos Discretos Sujeito a Atrasos de Comunicação de Eventos

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

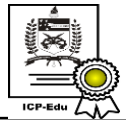
Prof. Felipe Gomes de Oliveira Cabral, Dr.
Universidade Federal de Santa Catarina

Prof. Max Hering de Queiroz, Dr.
Universidade Federal de Santa Catarina

Prof. Gustavo da Silva Viana, Dr.
Universidade Federal do Rio de Janeiro

Prof. Carlos Eduardo Viana Nunes, Dr.
Universidade Federal da Bahia

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de Mestre em Engenharia de Automação e Sistemas.



Documento assinado digitalmente
Werner Kraus Junior
Data: 13/10/2021 19:24:37-0300
CPF: 531.085.239-53
Verifique as assinaturas em https://v.ufsc.br

Prof. Werner Kraus Junior, Dr.
Coordenador do Programa

Documento assinado digitalmente
Felipe Gomes de Oliveira Cabral
Data: 13/10/2021 17:58:21-0300
CPF: 124.807.927-22
Verifique as assinaturas em https://v.ufsc.br

Prof. Felipe Gomes de Oliveira Cabral, Dr.
Orientador

Florianópolis, 21 de Julho de 2021.

Este trabalho é dedicado aos meus pais, irmãos,
namorada e amigos.

AGRADECIMENTOS

Agradeço aos meu pais José Lucas e Maria Odete, meus irmãos Lucas e Gustavo por me darem todo apoio e motivação para essa conquista.

Agradeço à minha namorada Clarissa pela paciência e companheirismo tanto na execução deste trabalho, como nas mudanças que estamos passando.

Agradeço imensamente ao meu orientador, Felipe G. Cabral, pelas horas dedicadas a mim e pela determinação na conclusão deste trabalho.

Agradeço aos amigos que fiz nas disciplinas do PGEAS e ao meu amigo Vitor Igor que me acompanhou nessa jornada em Florianópolis.

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo suporte financeiro.

“Não creio que haja uma emoção
mais intensa para um inventor do que
ver suas criações funcionando.
Essa emoção faz você esquecer de comer,
de dormir, de tudo.”
(Tesla, Nikola)

RESUMO

Recentemente, arquiteturas para o diagnóstico síncrono de sistemas a eventos discretos foram propostas com o objetivo de reduzir o custo computacional de implementação da técnica de diagnóstico quando comparada a métodos tradicionais. Nessa abordagem, diagnosticadores locais, construídos a partir do comportamento sem falha dos componentes do sistema, são implementados separadamente e inicializados ao mesmo tempo, funcionando em paralelo para identificar a ocorrência de falhas. Dentre as arquiteturas de diagnóstico síncrono, destaca-se a descentralizada, em que os diagnosticadores locais podem ser implementados de forma espacialmente distribuída, incluindo em diferentes equipamentos, permitindo uma flexibilização maior para sua implementação. Entretanto, problemas de comunicação entre equipamentos podem ocorrer, gerando possíveis atrasos entre a detecção do evento por um sensor e seu registro com sucesso pelo diagnosticador. Esse problema foi endereçado recentemente no contexto de diagnóstico descentralizado, mas não em arquiteturas síncronas. É importante destacar que uma adaptação direta do método de diagnóstico descentralizado sujeito a atraso de eventos implicaria em uma abordagem de diagnóstico síncrono menos eficiente, podendo levar, inclusive, à não possibilidade de diagnosticar determinadas falhas que poderiam ser diagnosticáveis. Nesta dissertação, um método de diagnóstico síncrono descentralizado robusto a atrasos de comunicação de eventos é proposto. O método é baseado em uma modificação nos modelos dos componentes do sistema, de tal forma que possíveis atrasos de comunicação sejam considerados e, assim, diagnosticadores locais que sejam calculados a partir dos modelos modificados sejam robustos a possíveis atrasos de comunicação de eventos. Essa modificação primeiro leva em consideração uma atualização de referencial para o atraso dos eventos do sistema completo para seus componentes. Nesse contexto, também é proposta uma definição de diagnosticabilidade síncrona robusta a atrasos de eventos e uma discussão acerca de sua verificação é apresentada.

Palavras-chave: Sistemas a eventos discretos; Diagnóstico de falhas; Arquitetura descentralizada; Atraso de comunicação; Verificação de diagnosticabilidade.

ABSTRACT

Recently, synchronous diagnosis architectures for discrete event systems have been proposed in the literature with the view to reduce the computational cost for the implementation of diagnosis techniques when compared to traditional methods. In this approach, local diagnosers, computed from the fault-free behavior models of the system components, are implemented separately and are initialized at the same time, running in parallel to identify fault event occurrences. Among the synchronous diagnosis architectures, the decentralized one is interesting since the local diagnosers can be implemented in a spatial distributed manner, in different equipments, which allows its use in different configurations. However, communication problems between equipments can occur, which can generate possible delays between the event detection by a sensor and its registration by the diagnoser. This problem has been recently addressed in the context of decentralized diagnosis, but a contribution to a synchronous diagnosis scheme has not been explored. It is important to remark that a direct application of the decentralized diagnosis robust to event communication delays method to the synchronous diagnosis scheme would lead to an inefficient diagnosis approach, that could cause a diagnosable fault to not be diagnosable. In this master thesis, a decentralized synchronous diagnosis method robust to event communication delays is proposed. The method is based on the modification of the system component models such that possible communication delays are considered in the modified models, which lead to local diagnosers robust to event communication delays. This modification takes into account a change in the referential of the event communication delay from the system model to the local component models. In this regard, in this work, the definition of a synchronous codiagnosability robust to event communication delays and a discussion on the verification of this property are also presented.

Key-words: Discrete event systems; Fault diagnosis; Decentralized architecture; Communication delays; Diagnosability verification.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

A quarta revolução industrial chegou oferecendo oportunidades espalhadas por todas as áreas da produtividade, dentre elas logística, aviação, transporte, saúde, produção de energia, produção de petróleo e gás e manufatura. Diversos proprietários e coordenadores de grandes empresas consideram cada vez mais o uso dessa tecnologia (GILCHRIST, 2016). Conhecida também como Indústria 4.0, essa mudança de paradigma gerou um aumento do uso de dispositivos interconectados tornando os sistemas de engenharia mais conectados e descentralizados, com capacidade de processamento local e troca de informações (GILCHRIST, 2016). Esses sistemas são chamados de Sistemas Ciber-Físicos (SCF). Essa tecnologia tem permitido que sistemas de manufatura sejam desenvolvidos em arquiteturas descentralizadas em que, embora fisicamente distribuídos, seus componentes podem trocar informações entre si. A figura 1 ilustra o exemplo de uma aplicação em uma fábrica real em que diversos componentes estão implementados fisicamente distribuídos, podendo trocar informações entre si por meio de redes e protocolos industriais.

Nesse contexto, o diagnóstico de falhas tem um papel importante, já que as consequências da ocorrência de uma falha em um componente de um SCF pode se espalhar para outros módulos do sistema, potencialmente causando danos a equipamentos e operadores. Um dos primeiros trabalhos que endereçam o problema de diagnóstico de falhas em SCFs modelados por Sistemas a Eventos Discretos (SEDs) é apresentado em Sampath *et al.* (1995). Nesse trabalho, um diagnosticador construído a partir de um observador do modelo completo do sistema em autômato é proposto cujo esquema de diagnóstico é apresentado na Figura 2(a). O cálculo do diagnostica-

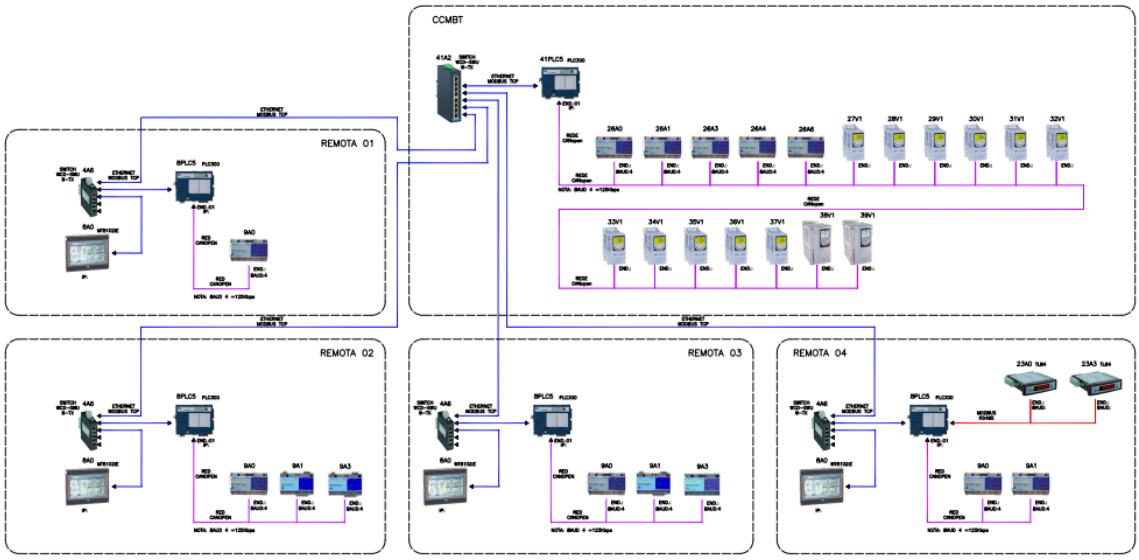
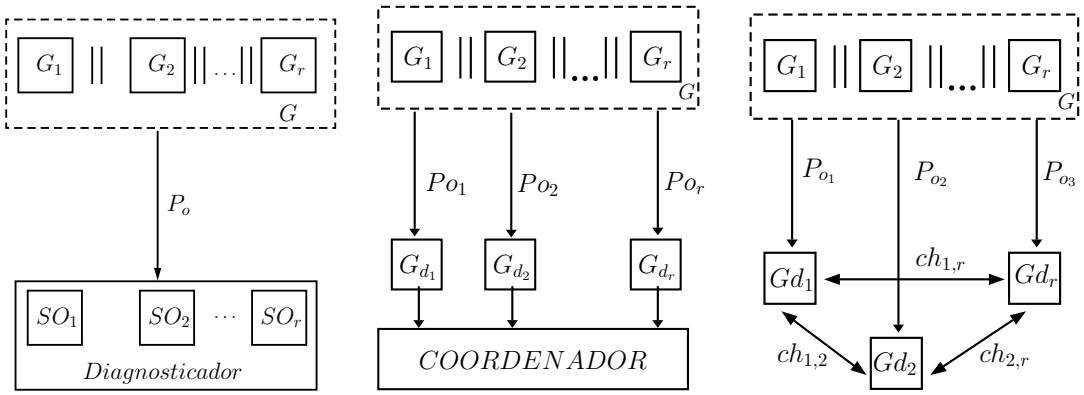


Figura 1 – Redes implementadas em uma fábrica real.



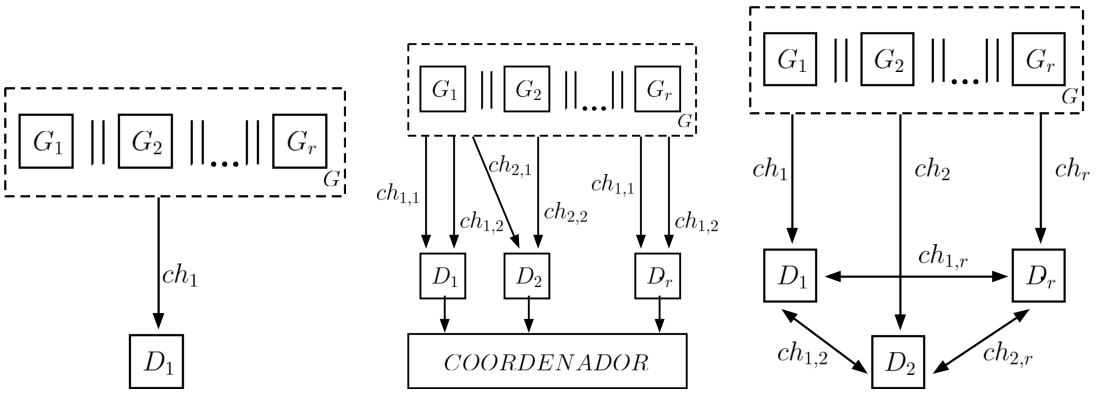
(a) Centralizado (b) Decentralizado (c) Distribuído

Figura 2 – Comparação entre as principais técnicas de diagnóstico.

dor proposto em Sampath *et al.* (1995) geralmente é evitado, uma vez que seu espaço de estados cresce exponencialmente com o número de estados da planta (que por sua vez, cresce exponencialmente com o número de componentes do sistema) na análise de pior caso. A abordagem de se construir um único diagnosticador baseado no modelo completo do sistema é chamada de abordagem centralizada, ou monolítica.

Os Sistemas Ciber-Físicos são constituídos de diversos componentes que trabalham em conjunto com o objetivo de completar uma tarefa pré determinada. Sendo assim, uma arquitetura para diagnóstico que falhas que se favoreça da natureza fisicamente distribuída de SCFs pode ser mais eficaz do que a arquitetura monolítica. Partindo desse princípio, abordagens descentralizadas e distribuídas de diagnóstico de falhas de SEDs têm sido propostas na literatura. Em Debouk *et al.* (2000), W. Qiu e R. Kumar (2006) e Wang *et al.* (2007), uma arquitetura de diagnóstico de falhas descentralizada, conforme ilustrado na Figura 2(b), foi proposta. Nessa abordagem, diagnosticadores locais baseados na planta completa do sistema, porém com acesso apenas a observações de eventos locais são construídos. Se pelo menos um desses diagnosticadores locais identificar a ocorrência de falha em seu diagnóstico, ele encaminha essa informação a um coordenador geral, que por sua vez encaminha essa mensagem ao operador. Em Wenbin Qiu e Ratnesh Kumar (2008) e Keroglou e Hadjicostis (2014, 2018) uma arquitetura distribuída foi desenvolvida com diagnosticadores locais calculados a partir do modelo completo da planta do sistema, conforme ilustra a Figura 2(c). Nesse esquema, os diagnosticadores locais trocam informações entre si sobre a observação de eventos e estimativas de estados com o objetivo de aumentar a eficiência do diagnóstico.

Em F. G. Cabral e Moreira (2020), o diagnóstico de falhas síncrono em SEDs foi proposto. Nesse trabalho, o cálculo de diagnosticadores locais para cada componente do sistema, levando em consideração apenas seus modelos sem falhas, é feito. As arquiteturas centralizada e descentralizada foram consideradas, como mos-



(a) Centralizado (b) Decentralizado (c) Distribuído

Figura 3 – Comparação entre as técnicas de diagnóstico síncrono.

tram as Figuras 3(a) e 3(b), respectivamente. Note que, nessa proposta o número de diagnosticadores locais corresponde ao número de componentes do sistema. Esse método foi explorado em Veras *et al.* (2018) para considerar uma arquitetura distribuída como ilustra a Figura 3(c), em que diagnosticadores locais são conectados em rede e comunicam entre si observações de eventos e estimativas de estado para refinar o diagnóstico.

Todos os trabalhos citados até este ponto consideram que os canais de comunicação entre a planta e os diagnosticadores são ideais, ou seja, não há perdas ou atrasos de comunicação. Esses canais de comunicação são responsáveis pelo envio dos eventos, que são detectados pelos sensores na planta, para os diagnosticadores locais. A principal consequência de problemas de comunicação nesses canais é a observação de eventos em uma ordem não prevista pelo diagnosticador ou a observação de um evento não esperado na estimativa de estados atual. Um esquema de diagnóstico que não seja projetado levando em consideração esses casos pode emitir falsos positivos acerca da ocorrência de eventos de falha. Com o objetivo de evitar isso, em Nunes *et al.* (2018) é proposto uma abordagem de diagnóstico descentralizada robusta a atrasos de comunicação e perdas de eventos para o diagnóstico descentralizado. Apesar do trabalho desenvolvido em Nunes *et al.* (2018) ter sido bem sucedido em implementar diagnosticadores locais robustos a problemas de comunicação de eventos, a arquitetura explorada é a descentralizada prevista no Protocolo 3 de Debouk *et al.* (2000) e, portanto, os diagnosticadores são construídos a partir da planta global do sistema.

O diagnóstico robusto a problemas de comunicação de eventos, como atrasos ou perdas de pacote, ainda não foi explorado em arquiteturas síncronas, ou seja, em F. G. Cabral e Moreira (2020) e Veras *et al.* (2018) é suposto que todos os canais de comunicação são ideais. Porém, em aplicações reais, possível que esses canais apresentem problemas de comunicação, geralmente relacionados a interferên-

cias eletromagnéticas ou mecânicas. Esses problemas podem acarretar atrasos na comunicação de eventos observáveis, o que pode provocar um diagnóstico impreciso, gerando falsos positivos. Assim, neste trabalho, um método de diagnóstico síncrono descentralizado robusto a atrasos de comunicação de eventos é proposto. Para tanto, o método apresentado em Nunes *et al.* (2018) é explorado no contexto do diagnóstico síncrono descentralizado. Como contribuições deste trabalho, uma noção de codiagnosticabilidade síncrona sujeita a atrasos de comunicação de eventos é proposta, além de um método para verificação dessa propriedade. Um método para construção de diagnosticadores locais robustos a atrasos de comunicação de eventos também é apresentado.

1.1 OBJETIVOS

O Objetivo deste trabalho é propor um novo protocolo para o Diagnóstico Síncrono Descentralizado de SEDs que seja robusto ao atraso de comunicação de eventos. Com base nisso, objetivos secundários são listados a seguir.

- Propor a construção de diagnosticadores locais robustos a atrasos de comunicação de eventos.
- Definir a noção de Codiagnosticabilidade Síncrona Sujeita a Atrasos de Comunicação Eventos (CSSA).
- Propor um método de verificação da CSSA.

1.2 ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação está organizada da seguinte forma. No capítulo seguinte são apresentados os fundamentos teóricos necessários para o entendimento deste trabalho, desde o conceito de SEDs, uma revisão de autômatos, até os métodos de diagnóstico apresentados por outros autores. O Capítulo 3 apresenta o diagnóstico descentralizado sujeito a atrasos de comunicação proposto em Nunes *et al.* (2018). Posteriormente, no Capítulo 4 método desenvolvido de Diagnóstico Síncrono Descentralizado Sujeito a Atrasos de Comunicação é apresentado. Um exemplo é utilizado ao longo do texto para ilustrar os resultados obtidos. No último capítulo as conclusões do trabalho são apresentadas juntamente com sugestões de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os fundamentos teóricos de sistemas a eventos discretos e protocolos de diagnóstico de falhas que foram utilizados para estudo, desenvolvimento e conclusão deste trabalho.

2.1 SISTEMAS A EVENTOS DISCRETOS

Define-se como sistema determinados componentes, ou parte limitada do universo, que trabalham em conjunto para o alcance de um objetivo comum que não seria possível realizar individualmente (CASSANDRAS; LAFORTUNE, 2008). São exemplos de sistemas: circuitos elétricos, sistemas respiratórios e vascular no corpo humano, entre outros. Dentre os variados tipos de sistemas, destacam-se os Sistemas a Eventos Discretos (SEDs) em que o espaço de estados é formado por um conjunto discreto e as transições entre esses elementos são dadas pela ocorrência de eventos. Eventos podem ser, por exemplo, o inicio de uma rotina, uma falha de equipamento ou uma balança que atinge determinado valor desejado (CASSANDRAS; LAFORTUNE, 2008). A definição formal de um SED é apresentada a seguir.

Definição 1 (Sistema a Eventos Discreto) *Um Sistema a Eventos Discreto é um sistema dinâmico cujo espaço de estados é discreto e que evolui com a ocorrência repentina de eventos.*

Devido à grande aplicação desse tipo de sistema surgiu a necessidade de desenvolver um formalismo matemático adequado para descrever o funcionamento de SEDs. Esse formalismo deve ser capaz de descrever o estado inicial do sistema, o estado atual e as regras de transição de estados de acordo com a ocorrência, em geral assíncrona no tempo, de eventos.

O conjunto de eventos de um SED pode ser considerado um alfabeto do sistema, uma sequência de eventos formam palavras e o conjunto formado por todas as sequências formam a linguagem do sistema. As linguagens determinam a evolução dos SEDs a partir da ocorrência de eventos, possuindo uma função análoga as equações diferenciais de sistemas dinâmicos contínuos no tempo (CASSANDRAS; LAFORTUNE, 2008). A linguagem de um SED e suas propriedades são apresentados a seguir.

2.1.1 Linguagens

Uma linguagem pode ser descrita como um conjunto de sequências finitas de eventos que podem ser geradas por um sistema. Assim, a linguagem descreve todos os possíveis comportamentos que um SED pode apresentar. A linguagem L é definida

no alfabeto Σ, como o conjunto de sequências de eventos construídas a partir de elementos de Σ. Para o alfabeto $\Sigma = \{a, b, c\}$, por exemplo, a linguagem $L_1 = \{c, b, acb\}$ ou $L_2 = \{$*Todas as possíveis sequências que começam com b*$\}$ podem ser definidas (CASSANDRAS; LAFORTUNE, 2008).

O conjunto de todas as sequências de eventos finitos que podem ser formadas por elementos de Σ é denotado por $\Sigma^\star$, incluindo a sequência vazia ε. $\Sigma^\star$ é chamado de Fecho de Kleene de Σ. O prefixo-fechamento de uma linguagem $L \subseteq \Sigma^\star$ é definido por $\bar{L} = \{s \in \Sigma^\star : (\exists t \in \Sigma^\star)[st \in L]\}$. Uma linguagem L é considerada prefixo-fechada se qualquer prefixo de qualquer sequência de L também é uma sequência de L. Como as linguagens são conjuntos, todas as operações aplicáveis a conjuntos podem também ser utilizadas para manipular linguagens. Além disso, algumas operações exclusivas a linguagens são definidas a seguir.

2.1.2 Operações com linguagens

Diversas operações em sequências e linguagens são essenciais para modificar e analisar SEDs. As operações de concatenação, Fecho de Kleene, Fecho de prefixo, Pós linguagem e Projeção são apresentadas como segue.

Definição 2 (Concatenação) *Sejam $L_1, L_2 \subseteq \Sigma^\star$, a concatenação $L_1 L_2$ é definida como*

$$L_1 L_2 = \{s = s_1 s_2 : (s_1 \in L_1) \wedge (s_2 \in L_2)\}.$$

A operação de concatenação agrupa cada sequência de uma linguagem com cada sequência de outra linguagem, sendo assim, a sequência s pertence a $L_1 L_2$ se ela puder ser construída como uma concatenação de uma sequência de L_1 com uma sequência de L_2.

Definição 3 (Fecho de Kleene) *Seja $L \subseteq \Sigma^\star$, então*

$$L^\star = \{\varepsilon\} \cup L \cup LL \cup \ldots$$

Um elemento de $L^\star$ é formado pela concatenação de elementos de L. Por definição, a sequência vazia ε é também um elemento de $L^\star$.

Definição 4 (Fecho de Prefixo) *Seja $L \subseteq \Sigma^\star$, então*

$$\bar{L} = \{s \in \Sigma^\star : (\exists t \in \Sigma^\star)[st \in L]\}.$$

$\bar{L}$ é constituído por todos os prefixos de todas as sequências de L. $L \subseteq \bar{L}$ é dita ser prefixo-fechada quando $L = \bar{L}$.

Definição 5 (Pós linguagem) *Se $L \subseteq \Sigma^\star$ e $s \in L$. A pós linguagem de L após s é denotada como L/s e definida da seguinte forma:*

$$L/s = \{t \in \Sigma^\star : st \in L\}.$$

Por definição, $L/s = \emptyset$ se $s \notin L$.

Definição 6 (Projeção) *A projeção $P_s^l : {\Sigma_l}^\star \rightarrow {\Sigma_s}^\star$, em que $\Sigma_s \subset \Sigma_l$, é definida como:*

$$P_s^l(\varepsilon) = \varepsilon$$

$$P_s^l(\sigma) = \begin{cases} \sigma, & se\ \sigma \in \Sigma_s \\ \varepsilon, & se\ \sigma \in \Sigma_l \setminus \Sigma_s, \end{cases}$$

sendo que $\setminus$ é a diferença entre os conjuntos.

Supondo os conjuntos $\Sigma_l = \{a, b, c\}$ e $\Sigma_s = \{c\}$ e as sequências de eventos $s_1 = ac$ e $s_2 = ab$, em que $s_1, s_2 \in {\Sigma_l}^\star$. A projeção P_s^l de s_1 é igual a $P_s^l(s_1) = c$ e a projeção $P_s^l(s_2) = \varepsilon$.

Definição 7 (Projeção inversa) *A projeção inversa $P_s^{l^{-1}} : \Sigma_s^\star \rightarrow 2^{\Sigma_l^\star}$ é definida como:*

$$P_s^{l^{-1}}(t) = \{s \in \Sigma_l^\star : P_s^l(s) = t\}.$$

Supondo que exista uma sequência s formada a partir de um conjunto Σ_s, a projeção inversa $P_s^{l^{-1}}$ sobre s resulta em um conjunto formado por todas as sequências de Σ_l, cuja projeção P_s^l resulta em s.

Tanto a operação de projeção, como a de projeção inversa podem ser aplicadas em linguagens. Para isso, basta aplicá-las a todas as sequências pertencentes à linguagem considerada. As operações de projeção em linguagens são definidas como (CASSANDRAS; LAFORTUNE, 2008):

Definição 8 (Projeção de linguagem) *Dados os conjuntos de eventos $\Sigma_s \subset \Sigma_l$ e a linguagem $L \subseteq \Sigma_l^\star$, então a projeção de $P_s^l(L)$ é definida como:*

$$P_s^l(L) = \{t \in \Sigma_s^\star : (\exists s \in L)[P_s^l(s) = t]\}.$$

Definição 9 (Projeção inversa de linguagem) *Dada a linguagem $L_s \subseteq \Sigma_s^\star$, então a projeção inversa $P_s^{l^{-1}}(L_s)$ é definida como:*

$$P_s^{l^{-1}}(L_s) = \{s \in \Sigma_l^\star : (\exists t \in L_s)[P_s^l(s) = t]\}.$$

As operações de projeção são muito usadas para representar as linguagens observadas por um controlador ou diagnosticador a partir dos eventos que são registrados por sensores. Eventos que não possuem sensores associados para detectar sua ocorrência são conhecidos como eventos não observáveis e são retirados da linguagem observada a partir da linguagem gerada do sistema utilizando-se uma operação de projeção.

Apesar se linguagens serem úteis para representar SEDs, sua utilização não é prática. A seguir, uma ferramenta capaz de representar linguagens, conhecida como autômato, é apresentada.

2.2 AUTÔMATOS

O Autômato é uma ferramenta capaz de representar linguagens com regras bem definidas, se tornando bem útil para visualizar SEDs. A definição formal de autômatos é apresentada a seguir (CASSANDRAS; LAFORTUNE, 2008).

Definição 10 (Autômatos) *Um autômato G é uma quádrupla:*

$$G = (Q, \Sigma, f, q_0),$$

sendo Q o conjunto de estados, Σ *o conjunto de eventos,* $f : Q \times \Sigma^\star \to Q$ *a função de transição, em que* $\Sigma^\star$ *é o Fecho de Kleene de* Σ*, e* q_0 *o estado inicial do sistema (CASSANDRAS; LAFORTUNE, 2008).*

A função $\Gamma_G : Q \to 2^\Sigma$ é definida como a função de eventos ativos de um estado de G. A função de eventos ativos $\Gamma_G(q)$ é o conjunto de todos os eventos σ tais que $f(q, \sigma)$ é definida. Note que a função de eventos ativos Γ_G pode ser completamente descrita a partir da função de transição f.

O autômato pode ser representado graficamente através de um grafo orientado, denominado diagrama de transição de estados. Nesse grafo, estados são representados por nós e as transições são representadas por flechas rotuladas pelos eventos que provocam as transições, como é ilustrado no exemplo a seguir.

Exemplo 1 *A Figura 4 representa o diagrama de transição de estados de um autômato G, em que* $Q = \{0, 1, 2\}$ *é o conjunto de estados e* $\Sigma = \{a, d, g, \sigma_1\}$ *é o conjunto de eventos. A função de transição é dada por* $f(0, a) = 1$*,* $f(1, g) = 1$*,* $f(1, \sigma_1) = 2$ *e* $f(2, d) = 0$*. Sendo o estado inicial* q_0 *igual* 0*.*

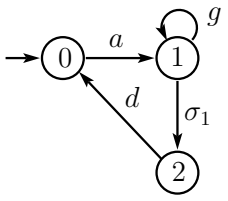


Figura 4 – Representação gráfica de um autômato.

2.2.1 Operações com autômatos

As operações realizadas em autômatos são capazes de modificar seu diagrama de transição de estados de acordo com alguma operação correspondente da linguagem gerada. Essas operações são usadas tanto para analisar SEDs quanto para combinar dois ou mais autômatos fazendo com que sistemas complexos possam ser obtidos a partir de vários sistemas menores e mais simples.

Definição 11 (Parte acessível de G) *Denotada por Ac(G), é obtida apagando de G todos os estados que não são alcançáveis por nenhuma sequência a partir do estado inicial (CASSANDRAS; LAFORTUNE, 2008), formalmente:*

$$Ac(G) := (Q_{ac}, \Sigma, f_{ac}, q_0) \text{ em que}$$
$$Qac = \{q \in Q : (\exists s \in \Sigma^\star)[f(q_0, s) = q]$$
$$f_{ac} = f|_{Q_{ac} \times \Sigma \to Q_{ac}}$$

Ao realizar a operação de parte acessível, a função de transição é restringida a um domínio menor dos estados acessíveis.

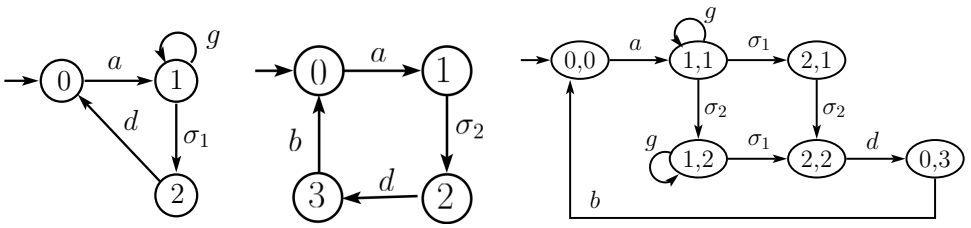
Definição 12 (Composição paralela) *Sejam $G_1 = (Q_1, \Sigma_1, f_1, q_{01})$ e $G_2 = (Q_2, \Sigma_2, f_2, q_{02})$ dois autômatos. $G_1 \| G_2$ denota a composição paralela de G_1 e G_2 (CASSANDRAS; LAFORTUNE, 2008) e é dado por:*

$$G_1 || G_2 := Ac(Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, f_{1||2}, (q_{01}, q_{02}), \text{ em que:}$$

$$f_{1||2}((q_1, q_2), \sigma) := \begin{cases} f_1(q_1, \sigma), f_2(q_2, \sigma) & \text{se } \sigma \in \Gamma_{G_1}(q_1) \cap \Gamma_{G_2}(q_2) \\ f_1(q_1, \sigma), q_2 & \text{se } \sigma \in \Gamma_{G_1}(q_1) \setminus \Sigma_2 \\ q_1, f_2(q_2, \sigma) & \text{se } \sigma \in \Gamma_{G_2}(q_2) \setminus \Sigma_1 \\ \text{Indefinido} & \text{Caso contrário} \end{cases}$$

Um evento comum em Σ_1 e Σ_2 só pode ser executado de forma síncrona nos dois autômatos, os outros ocorrem assincronicamente, um exemplo de composição paralela é apresentado a seguir.

Exemplo 2 *Considere os autômatos G_1 e G_2 apresentados na Figura 5(a) e 5(b), respectivamente. O autômato $G_1 \| G_2$ resultante da composição paralela entre G_1 e G_2 é apresentado na Figura 5(c).*



(a) G_1 (b) G_2 (c) $G_1 \| G_2$

Figura 5 – Composição Paralela

2.3 DIAGNOSTICABILIDADE DE SISTEMAS A EVENTOS DISCRETOS

2.3.1 Falha e Sequência sem falhas

O conjunto de eventos de G pode ser particionado em

$$\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo},$$

em que Σ_o e Σ_{uo} são, respectivamente, o conjunto de eventos observáveis e não observáveis. O conjunto $\Sigma_f \subseteq \Sigma_{uo}$ denota o conjunto de eventos de falha. Sem perda de generalidade, neste trabalho é suposto que existe apenas um evento de falha, ou seja, $\Sigma_f = \{\sigma_f\}$.

Definição 13 *(Sequência de falha e sequência sem falhas) Uma sequência de falha é a sequência de eventos s em que σ_f é um dos elementos que compõe essa sequência s. Analogamente, uma sequência s que não contém o elemento σ_f é chamada de sequência sem falhas.*

A linguagem $L_N \subset L$ é o conjunto de todas sequências sem falhas contidas em L. O autômato G_N, obtido a partir de G (MOREIRA *et al.*, 2011), é o autômato que gera a linguagem L_N. Portanto, o conjunto de todas as sequências de falhas é dado por

$$L_F = L \setminus L_N.$$

O autômato G_F gera a linguagem $\overline{L_F}$.

O alcance não observável de um estado $q \in Q$ em relação ao conjunto Σ_{uo} é denominado por

$$UR(q) = \{y \in Q : (\exists t \in \Sigma_{uo}^{\star})[f(q,t) = y]\}.$$

Essa definição é estendida a um subconjunto de estados B $\subseteq Q$ como

$$UR(B) = \cup_{q \in B} UR(q).$$

2.3.2 Diagnóstico Centralizado de SEDs

Definição 14 *(Diagnosticabilidade) Sejam L e L_N as linguagens vivas e prefixo-fechadas geradas por G e G_N, respectivamente. $L_F = L \setminus L_N$. L é dita ser diagnosticável em relação à projeção $P_o : \Sigma^\star \to \Sigma_o^\star$ e Σ_f se*

$$(\exists z \in \mathbb{N})(\forall s \in L_f)(\forall st \in L_F)(||t|| \geq z \Rightarrow P_o(st) \notin P_o(L_N)). \quad (1)$$

Se a linguagem L é diagnosticável, sempre será possível identificar a ocorrência do evento de falha após um número limitado de observações de eventos.

Em Sampath *et al.* (1995) e Sampath *et al.* (1996) é proposto um autômato diagnosticador para verificar a diagnosticabilidade de L e também para o diagnóstico de falhas. Esse diagnosticador é construindo baseado no autômato G_l, obtido rotulando os estados de G de acordo com os rastreamentos gerados pelo sistema. Se um estado de G é alcançado por uma sequência de falha, o estado recebe o rótulo F, caso contrário, o estado recebe o rótulo N. O autômato diagnosticador G_d é obtido calculando-se o observador de G_l em relação aos seus eventos observáveis, $G_d = Obs(G_l, \Sigma_o)$. Esses passos são resumidos no Algoritmo 1, apresentado a seguir.

Algoritmo 1 Autômato diagnosticador G_d.

Entrada: Autômato $G = (Q, \Sigma, f, q_0)$.
Saída: Autômato diagnosticador $G_d = (Q_d, \Sigma_o, f_d, q_{0,d})$.

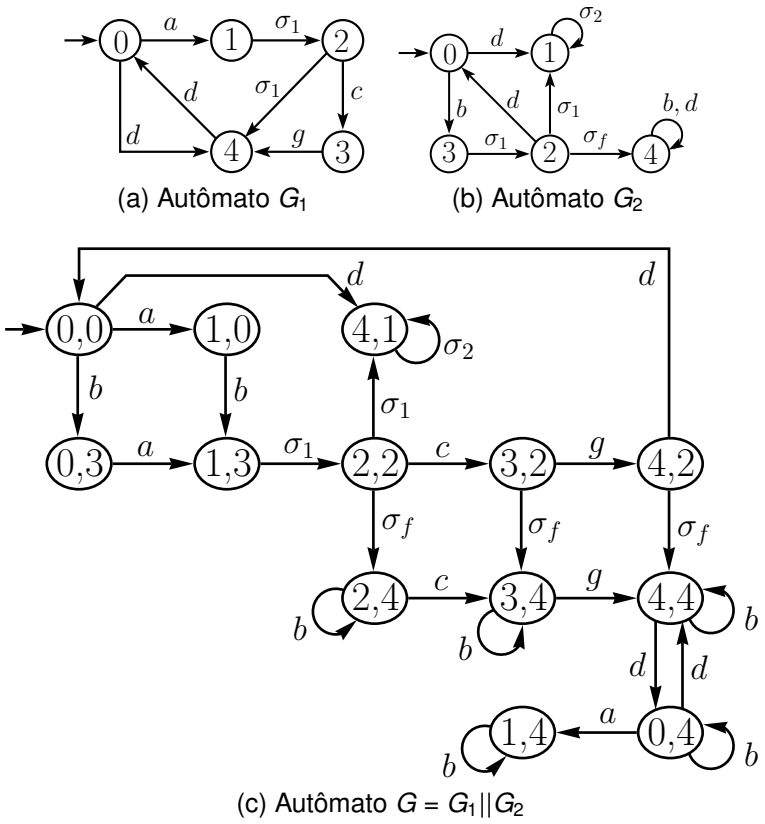
1: Define autômato $A_l = (Q_l, \Sigma_f, f_l, q_{0,t}$ em que $Q_l = \{N, F\}$, $f_l(N, \sigma_f) = F$, $f_l(F, \sigma_f) = F$ e $q_{0,l} = N$.

2: Calcular autômato $G_l = G||A_l$.

3: Calcular o autômato do diagnosticador $G_d = Obs(G_l, \Sigma_o)$.

Exemplo 3 *Considere o sistema G composto pelos módulos G_1 e G_2, $G = G_1 \| G_2$ na Figura 6. O conjunto de eventos de G é $\Sigma_1 \cup \Sigma_2 = \{a, b, c, d, g, \sigma_1, \sigma_2, \sigma_f\}$, em que $\Sigma_o = \{a, b, c, d, g\}$ e $\Sigma_{uo} = \{\sigma_1, \sigma_2, \sigma_f\}$. O conjunto de eventos de falha é $\Sigma_f = \{\sigma_f\}$. De acordo com o Algoritmo 1, o primeiro passo para construir o autômato G_d é calcular o autômato A_l, em que o diagrama de transição de estados está representado na Figura 7(a). Posteriormente, o autômato $G_l = G||A_l$ é obtido, apresentado na Figura 7(b). Por último, G_d é obtido ao calcular-se o autômato observador $Obs(G_l, \Sigma_o)$.*

Analisando a Figura 7(c), após a observação das sequências de eventos ab ou ba, G_d alcança os estados $\{(1, 3N); (2, 2N); (4, 1N); (2, 4F)\}$. Os rótulos N e F existem no estado, portanto, o diagnosticador G_d não tem a certeza de que o evento de falha ocorreu. Contudo, se o sistema gerar a sequência observada $b^\star$, não há sequências



(a) Autômato G_1 (b) Autômato G_2

(c) Autômato $G = G_1||G_2$

Figura 6 – G_1 e G_2 para Exemplo 3.

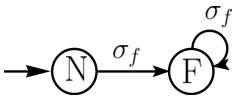
sem falhas que se confundam com a sequência de falha e a ocorrência do evento de falha é diagnosticada.

Apesar do diagnosticador G_d poder ser usado tanto para a verificação da diagnosticabilidade do sistema quanto para o diagnóstico online, sua construção geralmente é evitada. Isso se deve pelo fato de que, para obter-se G_d, o cálculo de um observador é feito, cuja complexidade computacional é exponencial com o número de estados da planta.

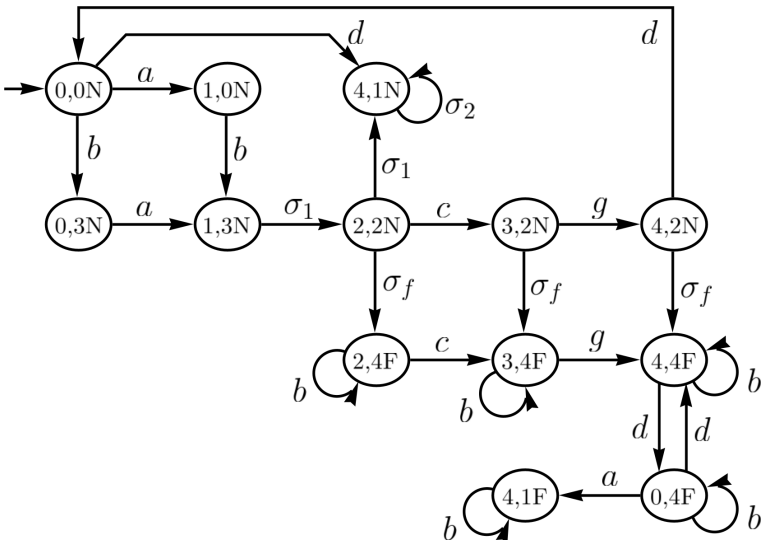
2.4 CODIAGNOSTICABILIDADE DE SEDS

Considere o diagnosticador descentralizado descrito no Protocolo 3 de (DEBOUK *et al.*, 2000). Esse protocolo consiste em l diagnosticadores locais que não se comunicam entre si, em que cada diagnosticador local tem seu próprio conjunto de eventos observáveis. Em seguida, para cada diagnosticador local, o conjunto de eventos pode ser dividido em partes como $\Sigma = \Sigma_{o_i} \dot{\cup} \Sigma_{uo_i}$, para $i = 1, ..., l$. No esquema de diagnóstico descentralizado proposto em (DEBOUK *et al.*, 2000) dois conjuntos de eventos observáveis diferentes podem ter eventos em comum, ou seja, $\Sigma_{o_i} \cap \Sigma_{o_j}$ não é necessariamente igual ao conjunto vazio, para $i \neq j, i, j \in \{1, ..., l\}$.

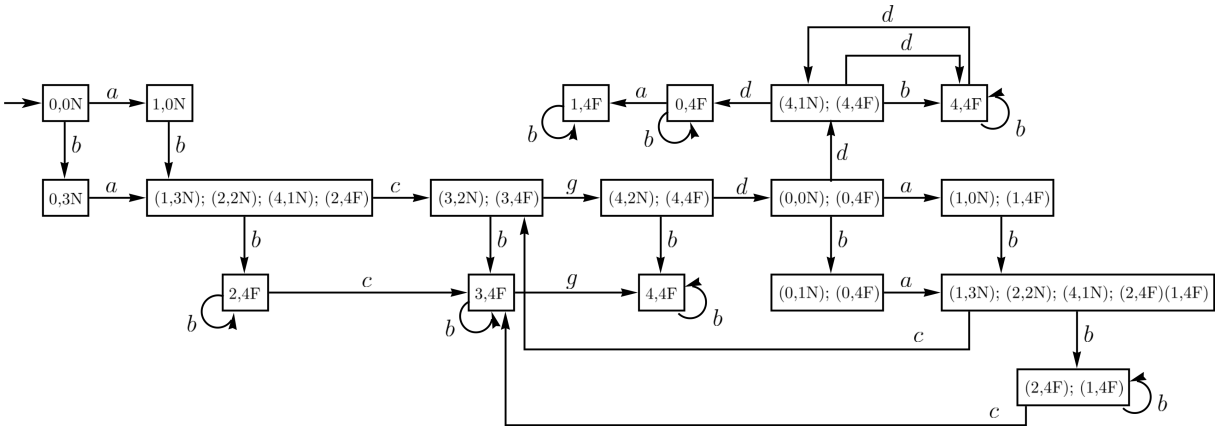
A definição de codiagnosticabilidade de uma linguagem L é dada por:



(a) autômato A_l



(b) autômato $G_l = G||A_l$



(c) Autômato $G_d = Obs(G_l, \Sigma_o)$

Figura 7 – A_l, G_l e G_d para Exemplo 3.

Definição 15 *(Codiagnosticabilidade) Seja L a linguagem viva gerada por G. L é dita ser codiagnosticável em relação à projeção* $P_{o_i} : \Sigma^\star \to \Sigma_{o_i}^\star$*, para* $i = 1, ..., l$ *e* σ_f *se:*

$$(\exists z \in \mathbb{N})(\forall s \in L \setminus L_N)(\forall st \in L \setminus L_N)(||t|| \geq z \Rightarrow (\exists i \in \{1, ..., l\})[P_{o_i}(st) \notin P_{o_i}(L_N)]. \quad (2)$$

2.5 DIAGNÓSTICO SÍNCRONO DE SEDS

Um método de diagnóstico foi proposto em (CABRAL, Felipe G *et al.*, 2015), que fornece a estimativa de estados da parte sem falhas do sistema, G_N, após a ocorrência de uma sequência observável.

Esse método foi usado como inspiração para o chamado diagnóstico síncrono

de SEDs em F. G. Cabral e Moreira (2020) que considera que um sistema G é obtido a partir de r componentes, ou seja, $G = ||_{k=1}^{r} G_k$, em que $G_k = (Q_k, \Sigma_k, f_k, q_{0,k})$, para $k = 1, 2, \ldots, r$, e $\Sigma_{k,o} \subseteq \Sigma_k$, denota o conjunto de eventos observáveis de G_k. $Obs(G_k, \Sigma_{k,o})$ denota o observador de G_k.

A ideia do diagnóstico síncrono apresentado em F. G. Cabral e Moreira (2020) é implementar observadores de estado das partes sem falha de G_k, G_{N_k}, simultaneamente. Para tanto, a seguinte noção de diagnosticabilidade síncrona é definida.

Definição 16 *(Diagnosticabilidade Síncrona) Sejam L e $L_N \subset L$ linguagens geradas por G e G_N, $L_F = L \setminus L_N$. Considerando que o sistema é composto por r módulos, de tal modo que $G_N = ||_{k=1}^{r} G_{N_k}$, em que G_{N_k} é o autômato que modela o comportamento sem falhas de G_k, L_{N_k} denota a linguagem gerada por G_{N_k},para $k = 1, ..., r$. L é dita ser diagnosticável de forma síncrona em relação a L_{N_k}, $P_{k,o}^{o} : \Sigma_o^\star \to \Sigma_{k,o}^\star$, para $k = 1, ..., r$, $P_o : \Sigma^\star \to \Sigma_o^\star$ e Σ_f se*

$$(\exists z \in \mathbb{N})(\forall s \in L_F)(\forall st \in L_F)(||t|| \geq z \Rightarrow P_o(st) \notin \cap_{k=1}^{r} P_{k,o}^{o^{-1}}(P_{k,o}(L_{N_k}))). \quad (3)$$

Em F. G. Cabral e Moreira (2020), um método de verificação da diagnosticabilidade síncrona de SEDs é apresentado. Essa definição pode ser generalizada para o caso descentralizado conforme é apresentado em F. G. Cabral e Moreira (2020).

No próximo capítulo, o método de diagnóstico descentralizado robusto a atrasos de comunicação de eventos proposto em Nunes *et al.* (2018) é apresentado.

3 DIAGNÓSTICO DESCENTRALIZADO SUJEITO A ATRASOS DE COMUNICAÇÃO DE EVENTOS

Atualmente, diversos sistemas de engenharia são implementados de forma distribuída no espaço. Isso exige que sensores, controladores e atuadores sejam conectados em redes de comunicação para trocar informações acerca de observações de eventos e envio de comandos de controle. Nesse contexto, o diagnóstico de falhas descentralizado pode ser uma boa opção, dado que é possível implementar diagnosticadores locais com observações distintas de eventos. Entretanto, essa implementação está sujeita a problemas de comunicação, como atrasos ou mesmo perdas de pacote. Em Nunes *et al.* (2018), um método para o diagnóstico descentralizado robusto a problemas de comunicação é proposto. Nesta dissertação, o problema de diagnóstico síncrono descentralizado sujeito a atrasos de comunicação de eventos é considerado como uma extensão do trabalho desenvolvido por Nunes *et al.* (2018). Portanto, neste capítulo, o método de diagnóstico descentralizado sujeito a atrasos de comunicação de eventos proposto em Nunes *et al.* (2018) é revisado.

Em Nunes *et al.* (2018), um método de diagnóstico descentralizado para plantas com múltiplos locais de medição, MS_j, $j = 1, \ldots, m$, é proposto. Nesse método, cada local de medição MS_j lê os sinais associados a um subconjunto de eventos $\Sigma_{MS_j} \subset \Sigma_o$ dos eventos observáveis do sistema. Nessa configuração, eventos de Σ_{MS_j} são comunicados para um diagnosticador local LD_i, $i = 1, \ldots, n$, por um canal de comunicação exclusivo denominado $ch_{i,j}$ que envia apenas os eventos detectados pelo local de medição MS_j. O conjunto de eventos que são comunicados ao diagnosticador local LD_i através do canal de comunicação $ch_{i,j}$ é denotado como $\Sigma_{o_{i,j}} \subseteq \Sigma_{MS_j}$. Caso não exista um canal de comunicação $ch_{x,y}$ entre um local MS_x e um diagnosticador local LD_y, $\Sigma_{o_{x,y}} = \emptyset$. Portanto, o conjunto de eventos observáveis de LD_i, Σ_{o_i}, é dado por

$$\Sigma_{o_i} = \bigcup_{j=1}^{m} \Sigma_{o_{i,j}}. \tag{4}$$

E o conjunto de eventos observáveis do sistema, Σ_o, é dado por $\Sigma_o = \cup_{i=1}^{n} \Sigma_{o_i}$.

Na Figura 8, a arquitetura de diagnóstico descentralizada utilizada em Nunes *et al.* (2018), com três diferentes locais de medição e dois diagnosticadores é apresentada. Note que o local de medição MS_1 é capaz de comunicar para o diagnosticador local LD_1 pelo canal $ch_{1,1}$ apenas os eventos que fazem parte do conjunto $\Sigma_{o_{1,1}} \subseteq \Sigma_{MS_1}$, assim como o local de medição MS_3 comunica para o diagnosticador local LD_2 apenas os eventos presentem em $\Sigma_{o_{2,3}} \subseteq \Sigma_{MS_3}$. Por outro lado, o local de medição MS_2 consegue comunicar os eventos que do conjunto $\Sigma_{o_{1,2}} \subseteq \Sigma_{MS_2}$ para o diagnosticador local LD_1 e os eventos do conjunto $\Sigma_{o_{2,2}} \subseteq \Sigma_{MS_2}$ para o diagnosticador LD_2. É importante ressaltar que para a arquitetura proposta em Nunes *et al.* (2018), um local de medição pode transmitir conjuntos diferentes de eventos observáveis para

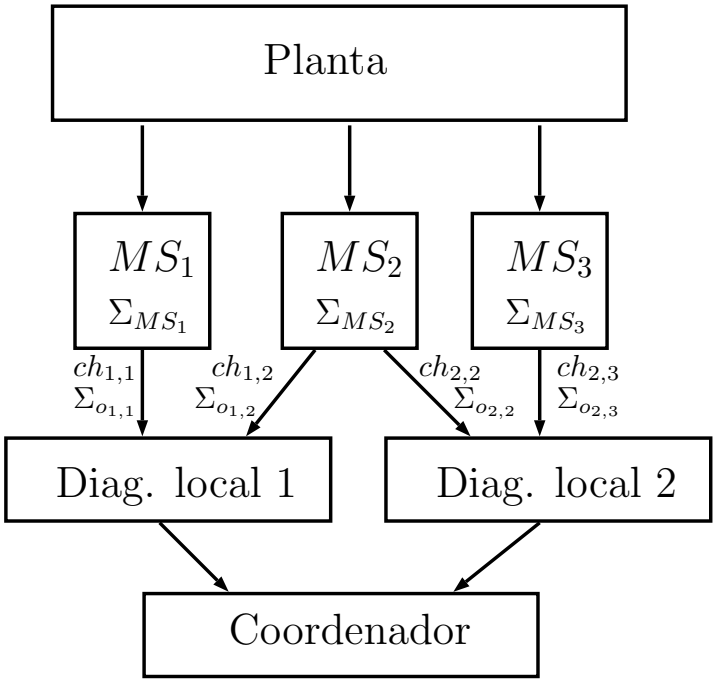


Figura 8 – Arquitetura do diagnóstico síncrono descentralizado.

diferentes diagnosticadores locais, sendo assim na Figura 8, o conjunto $\Sigma_{o_{1,2}}$ pode ser diferente do conjunto $\Sigma_{o_{2,2}}$.

A comunicação entre locais de medição e diagnosticadores locais realizada por redes de comunicação podem introduzir dois problemas para o diagnóstico: (*i*) a perda de dados transmitido pelo canal; e (*ii*) atrasos na comunicação de uma ocorrência de evento para um diagnosticador local. Independente de qual situação ocorra, o diagnosticador pode enviar uma decisão errada ao coordenador e, portanto, o diagnóstico perde a credibilidade.

Neste trabalho, apenas problemas relacionados a atrasos de comunicação de eventos são considerados. Nesse contexto, as seguintes hipóteses são feitas em Nunes *et al.* (2018):

H1. O atraso na comunicação de um evento $\sigma \in \Sigma_o$ é contado em passos (TRIPAKIS, 2004), sendo que um passo corresponde à ocorrência de um evento em *G*. Assim, o atraso é medido pelo número de eventos gerados em *G* após a ocorrência de σ e antes de sua efetiva comunicação ao diagnosticador local.

H2. Os atrasos na comunicação de eventos são limitados.

H3. Os canais de comunicação se comportam como uma fila FIFO (*first-in first-out*).

H4. Há apenas um canal $ch_{i,j}$ entre o local de medição MS_j e o diagnosticador LD_i, e o atraso máximo de comunicação do canal $ch_{i,j}$ é conhecido e denotado por $k_{i,j}$.

H5. Os conjuntos de eventos Σ_{MS_i} e Σ_{MS_j} são disjuntos para todo $i, j \in \{1, 2, \ldots, m\}$, $i \neq j$.

3.1 MODELO DA PLANTA SUJEITO A ATRASOS DE COMUNICAÇÃO

Na estrutura do sistema mostrada pela Figura 8, um dado transmitido por um canal $ch_{i,p}$, pode atrasar em relação a outro canal de comunicação $ch_{i,q}$, em que $p \neq q$ e $p, q \in \{1, 2, \ldots, j\}$. Como consequência, eventos podem ser observados em uma ordem diferente de sua real ocorrência no sistema pelo diagnosticador local LD_i, levando a um diagnóstico falso em relação à ocorrência de uma falha. A fim de resolver esse problema em sistemas a eventos discretos com atraso na comunicação, surge a necessidade de construir os autômatos G_i, $i = 1, 2, \ldots, n$, que representam todas as possíveis observações de sequências executadas pela planta pelos diagnosticadores LD_i.

Para distinguir um evento $\sigma \in \Sigma_{o_{i,j}}$ que ocorre na planta a partir de sua observação pelo diagnosticador LD_i, o evento σ_{s_i} é criado para representar a observação com sucesso do evento σ pelo diagnosticador local LD_i. Assim

$$\Sigma^{s}_{o_{i,j}} = \{\sigma_{s_i} : \sigma \in \Sigma_{o_i}\} \tag{5}$$

denota o conjunto de eventos que são observados no diagnosticador local LD_i e cuja a ocorrência foi registrada em MS_j, e

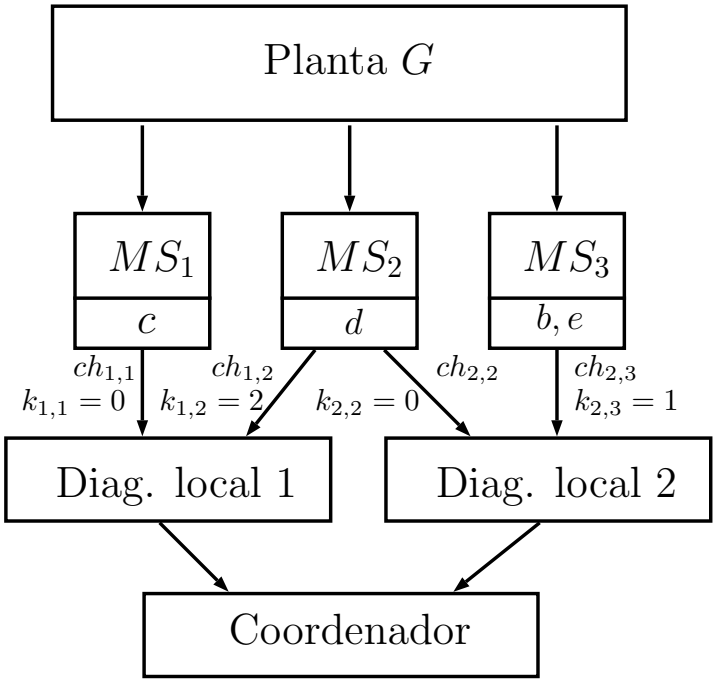
$$\Sigma^{s}_{o_i} = \bigcup_{j=1}^{m} \Sigma^{s}_{o_{i,j}} \tag{6}$$

denota o conjunto de eventos observáveis que são comunicados com sucesso para o diagnosticador local LD_i. Os seguintes conjuntos de eventos podem ser definidos

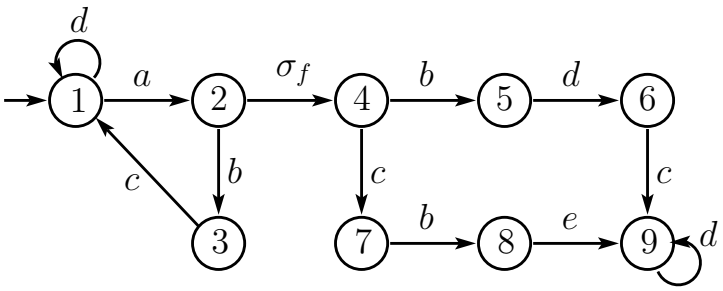
$$\Sigma_i = \Sigma \cup \Sigma^{s}_{o_i}, i = 1, \ldots, n, \tag{7}$$

em que os eventos de Σ são não observáveis para todos diagnosticadores LD_i, $i = 1, \ldots, n$, e os eventos $\Sigma^{s}_{o_i}$ são observáveis para o diagnosticador local LD_i.

Exemplo 4 *Considere novamente a arquitetura de diagnóstico descentralizado representado na Figura 8. Suponha que o sistema de diagnóstico seja composto por dois diagnosticadores locais LD_1 e LD_2 e três locais de medição. Na Figura 8, o modelo da planta em autômato, G, é apresentada, em que $\Sigma = \{a, b, c, d, e, \sigma_f\}$. Os conjuntos $\Sigma_{MS_1} = \{c\}$, $\Sigma_{MS_2} = \{d\}$ e $\Sigma_{MS_3} = \{b, e\}$ são os eventos detectados pelos locais de medição MS_1, MS_2 e MS_3, respectivamente. Sendo assim, o conjunto de eventos observáveis do diagnosticador local LD_1 é $\Sigma_{o_1} = \{c, d\}$. O conjunto de eventos observados com sucesso pelo diagnosticador local LD_1 referente aos eventos $\{c\}$ e $\{d\}$ é*



(a)



(b) G

Figura 9 – Esquema de diagnóstico e autômato G para o Exemplo 4.

formado por $\Sigma^s_{o_1} = \{c_{s_1}, d_{s_1}\}$. A ocorrência de um evento do conjunto Σ_{o_1}, é transmitida pelos canais de comunicação $ch_{1,1}$ e $ch_{1,2}$, portanto, $\Sigma_{o_{1,1}} = \{c\}$ e $\Sigma_{o_{1,2}} = \{d\}$. Analogamente, o conjunto de eventos observáveis de LD_2 é $\Sigma_{o_2} = \{b, d, e\}$. O conjunto de eventos observados com sucesso pelo diagnosticador localLD_2 é $\Sigma^s_{o_2} = \{b_{s_2}, d_{s_2}, e_{s_2}\}$. A ocorrência de um evento do conjunto $\Sigma^s_{o_2}$ é comunicado pelos canais $ch_{2,2}$ e $ch_{2,3}$, portanto, $\Sigma_{o_{2,2}} = \{d\}$ e $\Sigma_{o_{1,2}} = \{b, e\}$. Neste exemplo, σ_f representa o evento de falha. Os atrasos máximos de cada canal de comunicação são $k_{1,2} = 2$,$k_{2,3} = 1$ e $k_{1,1} = k_{2,2} = 0$.

Note que o autômato G pode gerar a sequência livre de falha $s_N = d^p abcd^p$ e as sequências com falha $s_{F_1} = d^p a\sigma_f bdcd^p$ e $s_{F_2} = d^p a\sigma_f cbed^{p-1}$, sendo $p \in \{1, 2, \ldots\}$. O conjunto de eventos observáveis do diagnosticador LD_1 é $\Sigma_{o_1} = \{c, d\}$ e os eventos observáveis de LD_2 é $\Sigma_{o_1} = \{b, d, e\}$. Supondo que não haja problemas de comunicação, as sequências observadas por LD_1 são $P_{o_1}(s_N) = P_{o_1}(s_{F_2}) = d^p cd^p$ e $P_{o_1}(s_{F_1}) = d^p dcd^{p-1}$, demonstrado na Figura 10, as sequências observadas por LD_2 são $P_{o_2}(s_N) = P_{o_2}(s_{F_1}) = d^p bd^p$ e $P_{o_2}(s_{F_2}) = d^p bed^p$, demonstrado na Figura 11. Isso

implica que nenhum diagnosticador local consegue diferenciar a linguagem de falha da linguagem sem falha sozinho. Porém, LD_1 consegue diagnosticar a sequência de falha s_{f_1}, e o diagnosticador LD_2 diagnostica a sequência de falha s_{f_2} . Portanto, a linguagem desse sistema é codiagnosticável.

Suponha agora que tenhamos uma situação em que haja atrasos de comunicação de $k_{2,3} = 1$ e $k_{2,2} = 0$. A planta gera a sequência $s = \sigma_f bdc$, o diagnosticador local LD_2, observa a ocorrência do evento b atrasada de um passo e observa a ocorrência do evento d sem atrasos. Note na 12 que , b e d são transmitidos por canais deferentes de comunicação e LD_2 pode observar o evento d antes de observar o evento b. Como consequência, as seguintes sequências representam todas as possíveis observações da sequência $s = \sigma_f bdc$ pelo diagnosticador local LD_2:

- *A sequência $b_{s_2} d_{s_2}$ que modela o caso de não atraso na observação do evento b, ou o caso do atraso na observação de b ser igual a um passo, porém LD_2 ainda recebe a informação da ocorrência de b antes de receber a informação da ocorrência de d;*
- *A sequência $d_{s_2} b_{s_2}$ que modela o caso em que o atraso de observação de b é de um passo e LD_2 recebe a informação da ocorrência do evento d antes de receber a informação da ocorrência de b.*

Para que todas as possíveis observações da sequência $s \in L(G)$ pelo diagnosticador local LD_i sejam obtidas, uma função de inserção de eventos que pertencem a $\Sigma^s_{o_i}$ baseada nos atrasos máximos de comunicação $k_{i,j}$ e nos conjuntos de eventos $\Sigma_{o_{i,j}}$ deve ser definida. Para tanto, a seguir são apresentadas algumas funções de projeção.

$$P_i : \Sigma_i^\star \to \Sigma^\star, \tag{8}$$

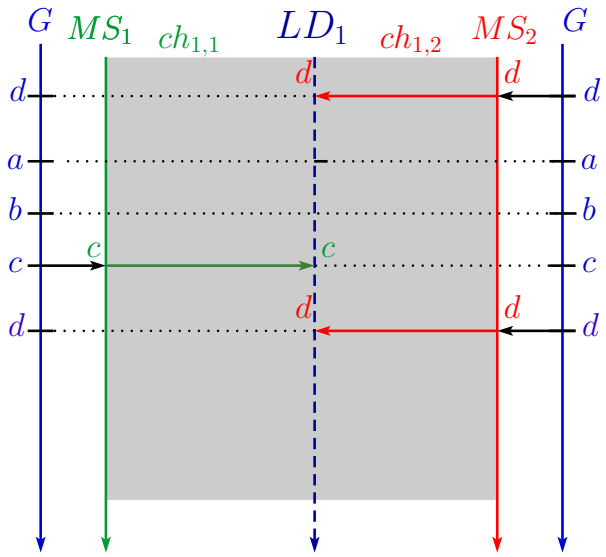
$$P_{i,o_{i,j}} : \Sigma_i^\star \to \Sigma_{o_{i,j}}^\star, \tag{9}$$

$$P_{i,s_{i,j}} : \Sigma_i^\star \to \Sigma_{o_{i,j}}^{s\star}, \tag{10}$$

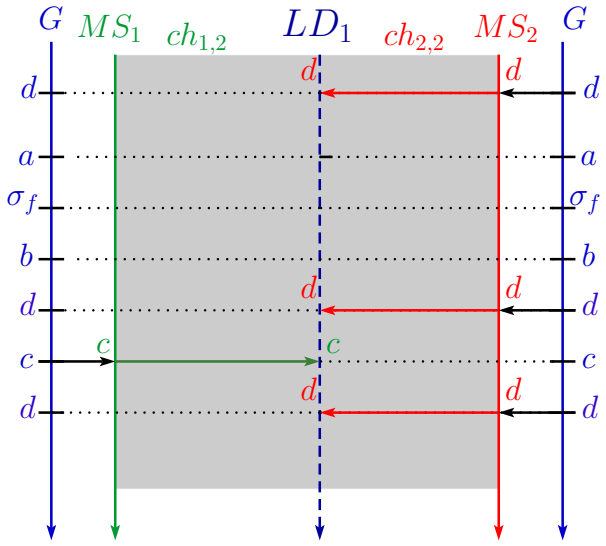
Além disso, seja $\omega_{\sigma^{(l)}}$ o prefixo da sequência $\omega \in \Sigma_i^\star$ cujo último evento é a l-ésima ocorrência de σ, e seja $\omega_{\sigma_s^{(l)}}$ o prefixo de ω cujo último evento é a l-ésima ocorrência de σ_{s_i}, se $\sigma_{s_i}^{(l)} \in \omega$, ou ω, se $\sigma_{s_i}^{(l)} \notin \omega$.

Definição 17 *A função de inserção associada ao diagnosticador local LD_i e ao conjunto de eventos observáveis em $\Sigma_{o_{i,j}}$, transmitidos através dos canais de comunicação $ch_{i,j}$ que possuem atrasos máximos de comunicação $k_{i,j}$, $j = 1, \ldots, m$ é o mapeamento*

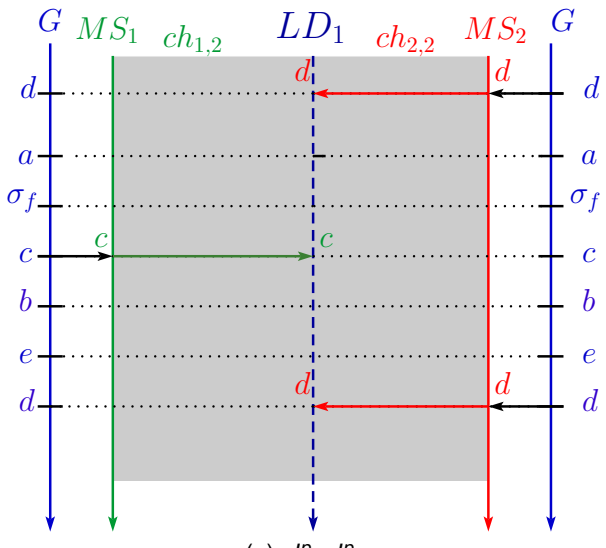
$$\chi_i : \Sigma^\star \to 2^{\Sigma_i^\star},$$
$$s \mapsto \chi_i(s),$$



(a) $d^p cd^p$

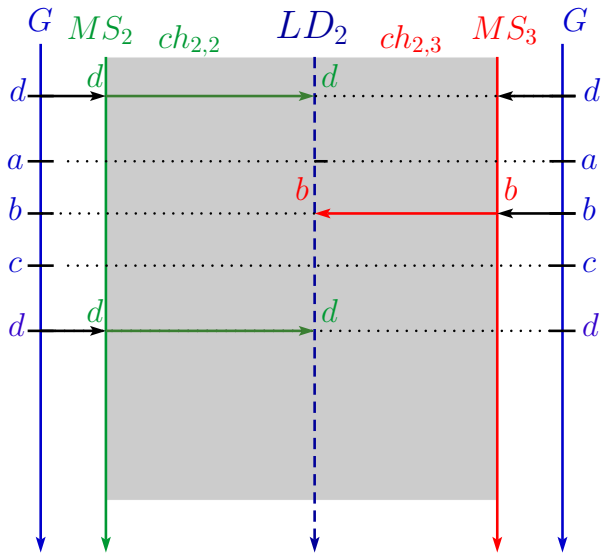


(b) $d^p dcd^{p-1}$



(c) $d^p cd^p$

Figura 10 – Linha do tempo do sistema sem atrasos para o LD_1.



(a) $d^p c d^p$

(b) $d^p d c d^{p-1}$

(c) $d^p c d^p$

Figura 11 – Linha do tempo do sistema sem atrasos para o LD_2.

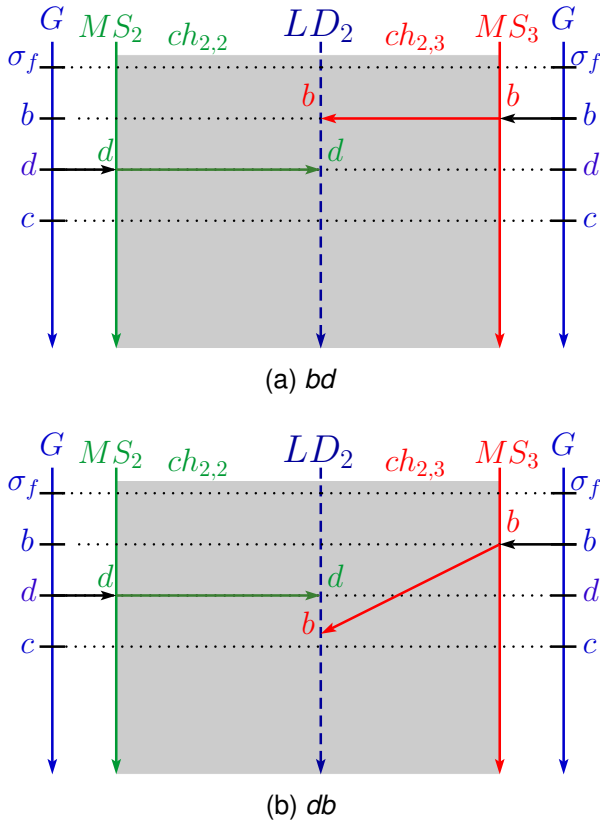


Figura 12 – Linha do tempo do sistema com atrasos para o LD_2.

em que $\omega \in \chi_i(s)$ *se* ω *satisfaz as seguintes condições:*

1. $P_i(\omega) = s$;
2. *Para todo* $\sigma \in \Sigma_{o_{i,j}}$, *e* $\sigma^{(l)} \in \omega$:

$$\|P_i(\omega_{\sigma_s^{(l)}})\| - \|P_i(\omega_{\sigma^{(l)}})\| \leq k_{i,j}$$

3. *Para todo* $\sigma_{s_i} \in \Sigma^s_{o_{i,j}}$ *e* $\sigma^{(l)}_{s_i} \in \omega$:

$$\sigma^{(l)} \in \omega_{\sigma_s^{(l)}},$$

e

$$\|P_{i,o_{i,j}}(\omega_{\sigma^{(l)}})\| = \|P_{i,s_{i,j}}(\omega_{\sigma_s^{(l)}})\|.$$

A extensão de χ_i *para o domínio* $2^{\Sigma^\star}$ *é definida como* $\chi_i(L) := \cup_{t \in L}\chi_i(t)$.

A seguir, algumas operações e funções são definidas com o objetivo de construir os modelos em autômato da planta sob atrasos de comunicação de eventos para o diagnóstico descentralizado.

Definição 18 *Seja* $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$. *Defina* $\Sigma_{o\nu} = \Sigma_o \cup \{\nu\}$ *e o conjunto de estados Q, em que cada estado* $q \in Q$ *é rotulado com uma sequência* $s \in \Sigma_{o\nu}^{\star}$. *Então, as seguintes funções podem ser definidas:*

a. *A função de reposição rep é definida como:*

$$rep : Q\ x\ \mathbb{N} \rightarrow Q$$

em que para todo $q = q_1 q_2 \dots q_l \in Q$,

$$rep(q, i) = \begin{cases} q_1 q_2 \dots q_{i-1} \nu q_{i+1} \dots q_l, \textit{se } i \leq l \\ \textit{indefinido, caso contrário.} \end{cases}$$

b. *A função de eliminação cut é definida como:*

$$cut : Q \rightarrow Q$$

em que para todo $q = q_1 q_2 \dots q_l in Q$,

$$cut(q) = \begin{cases} q_i q_{i+1} \dots q_l, \ \textit{se } (\exists i \leq l)[(q_i \neq \nu) \wedge (q_k = \nu, \forall k \in \{1, 2, \dots, i-1\})] \\ \nu, \ \textit{se } q_k = \nu, \forall \in \{1, 2, \dots, l\}. \end{cases}$$

c. *A função de índice do local de medição ms é definida como:*

$$ms : \Sigma_{o\nu} \rightarrow \{1, 2, \dots, m\}$$

em que para todo $\sigma \in \Sigma_{o_\nu}$,

$$ms(\sigma) = \begin{cases} j : \ \textit{se } \sigma \in \Sigma_{o_{i,j}} \textit{ para algum } i \in \{1, 2, \dots, n\} \\ \textit{indefinido, caso contrário.} \end{cases}$$

d. *A função bijetiva* ϕ_i *é definida, para* $i = 1, \dots, n$, *como:*

$$\begin{aligned} \phi_i : & \Sigma_{o_i}^s \rightarrow \Sigma_{o_i}, \\ & \sigma_{s_i} \mapsto \phi_i(\sigma_{s_i}) = \sigma, \end{aligned}$$

e ϕ_i *pode ser estendida para conjuntos de eventos como*

$$\phi_i(\Sigma_{o_i}^s) = \bigcup_{\sigma_{s_i} \in \Sigma_{o_i}^s} \phi_i(\sigma_{s_i}).$$

De acordo com a definição 18, $rep(q, i)$ substitui o i-ésimo elemento do estado q por ν. Essa função é introduzida para representar que um evento ocorreu, mas o conhecimento de qual evento ocorreu não é importante. A função $cut(q)$ elimina o maior prefixo do estado q formado por apenas elementos ν e a função $ms(\sigma)$ retorna o índice j que corresponde ao local de medição (MS_j) que detecta a ocorrência do evento

σ. A função $\phi_i(\sigma_{s_i})$ retorna o evento σ cuja observação bem sucedida é representada por σ_{s_i}.

O Algoritmo 2 descreve a construção do autômato D_i, associado ao diagnosticador local LD_i, que modela todos os possíveis atrasos na comunicação de eventos para LD_i, do local de medição MS_j, $j = 1, 2, \ldots, m$. O autômato D_i é o modelo de comunicação de atraso de rede.

Algoritmo 2 Construção do automato D_i (Modelo com atraso de comunicação)

Entradas: $m, n, \Sigma_{o_{i,j}}, k_{i,j}$, para $i = 1, \ldots, n, j = 1, \ldots, m$.
Saídas: $D_i = (Q_i, \Sigma_i, \delta_i, \Lambda_i, q_{0_i})$, $i = 1, \ldots, n$.

1: **para** $i = 1, \ldots, n$ **faça**
2: Defina $q_{0_i} = \nu$ e $Q_i = \emptyset$.
3: Calcule $\Sigma^s_{o_i}$ de acordo com as equações (5) e (6) , e defina $\Sigma_i = \Sigma \cup \Sigma^s_{o_i}$.
4: $F \leftarrow (q_{0_i})$, em que F denota uma fila FIFO.
5: **enquanto** $F \neq \emptyset$ **faça**
6: $u \leftarrow head[F]$.
7: **se** $u = q_{0_i}$ **então**
8: **para todo** $\sigma \in \Sigma$ **faça**
9: Calcule $\tilde{q} = \delta_i(u, \sigma) = \begin{cases} \sigma, \text{ se } \sigma \in \Sigma_{o_i} \\ u, \text{ se } \sigma \in \Sigma_{uo_i} \end{cases}$
10: **se** $\tilde{q} \neq u$ **então**
11: $Enqueue(F, \tilde{q})$
12: $Q_i \leftarrow Q_i \cup \{u\}$
13: $Dequeue(F)$
14: **senão se** $u \neq q_{0_i}$ **então**
15: Faça $l = \|u\|$ e forme o conjunto $I_l = \{1, 2, \ldots, l\}$.
16: Defina $u = \sigma_1\sigma_2 \ldots \sigma_l$ e calcule $I_\nu = \{y \in I_l : (\exists \sigma_y \in u)[\sigma_y = \nu]\}$.
17: Calcule $I_{/\nu} = I_l \setminus I_\nu$.
18: **para todo** $\sigma \in \Sigma o_i$ **faça**
19: $\tilde{q} = \delta_i(u, \sigma) = \begin{cases} u\sigma, \text{ se } \|\sigma_y\sigma_{y+1} \ldots \sigma_l\| \leq k_{i,ms(\sigma_y)}, \forall y \in I_{/\nu} \\ \text{indefinido, caso contrário.} \end{cases}$
20: Se $\tilde{q}$ é indefinido, $Enqueue(F, \tilde{q})$.
21: **para todo** $\sigma \in \Sigma_{uo_i}$ **faça**
22: $\tilde{q} = \delta_i(u, \sigma) = \begin{cases} u\nu, \text{ se } \|\sigma_y\sigma_{y+1} \ldots \sigma_l\| \leq k_{i,ms(\sigma_y)}, \forall y \in I_{/\nu} \\ \text{indefinido, caso contrário} \end{cases}$
23: Se $\tilde{q} \notin F$, $Enqueue(F, \tilde{q})$.
24: **para todo** $\Sigma^s_{o_{i,j}}$, em que $j = 1, 2, \ldots, m$ **faça**
25: Crie o conjunto $Y = \{y : (\sigma_y \in u) \wedge (\sigma_y \in \phi_i(\Sigma^s_{o_{i,j}}))\}$.
26: Se $Y \neq \emptyset$, então calcule $\hat{y} = min(Y)$ e $\tilde{q} = \delta_i(u, \phi_i^{-1}(\sigma_{\hat{y}})) = cut(rep(u, \hat{y}))$.
27: Se $(\tilde{q} \notin Q_i) \wedge (\tilde{q} \notin F)$, $Enqueue(F, \tilde{q})$.
28: Defina $Q_i \leftarrow Q_i \cup \{u\}$.
29: $Dequeue(F)$
30: **para todo** $q_i \in Q$ **faça**
31: $\Lambda_i(q_i) = \{\sigma \in \Sigma_i : \delta_i(q_i, \sigma)!\}$

O exemplo a seguir ilustra a construção do autômato D_i de acordo com Algoritmo 2.

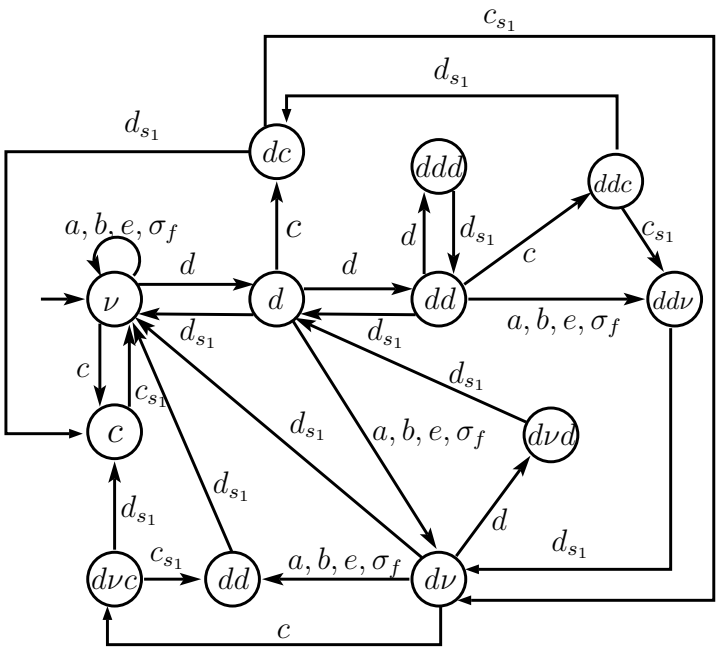
Exemplo 5 *Considere a arquitetura descentralizada da Figura 9a, e o autômato G da Figura 9b, em que* $\Sigma = \{a, b, c, d, e.\sigma_f\}$*. Assim como no Exemplo 4. o conjunto de eventos observáveis de* LD_1 *e* LD_2 *são respectivamente,* $\Sigma_{o_1} = \{c, d\}$ *e* $\Sigma_{o_2} = \{b, d, e\}$*. Para o diagnosticador local* LD_1*,* $\Sigma_{o_{11}} = \{c\}$ *e* $\Sigma_{o_{12}} = \{d\}$*,para o diagnosticador local* LD_2*,* $\Sigma_{o_{22}} = \{d\}$ *e* $\Sigma_{o_{23}} = \{b, e\}$ *o sistema está sujeito a atrasos de comunicação,em que,* $k_{11} = k_{22} = 0$*,*$k_{12} = 1$ *e* $k_{23} = 1$*. Assim, para o diagnosticador local* LD_1*, a observação da ocorrência do evento d pode se atrasar em no máximo de dois passos,e para o diagnosticador local* LD_2*, a observação da ocorrência do evento b e e pode atrasar no máximo em um passo.*

A fim de modelar a observação dos atrasos associados a LD_1*, foi construído o autômato* D_1*, mostrado na Figura 13a, Seguindo os passos do Algorítimo 2. Em que o estado inicial* $q_{0_i} = \nu$ *é adicionado a F. Enquanto F não for vazio , o primeiro elemento de F é atribuído a variável u, e uma vez que* $u = \nu$*, as transições de* ν *serão definidas para todo* $\sigma \in \Sigma$*, como:* $\delta_1(\nu, c) = c$*,* $\delta_1(\nu, d) = d$ *e* $\delta_1(\nu, \sigma_f) = \delta_1(\nu, a) = \delta_1(\nu, b) = \delta_1(\nu, e) = \nu$*. Posteriormente, os eventos {c} e {d} são adicionados no final da fila F, assim* $f = (\nu, c, d)$*, o estado* ν *é adicionado no conjunto* Q_1*, ou seja,* $Q_1 = \{\nu\}$*. Finalmente o primeiro elemento de F é removido , e a fila se tornará* $F = (c, d)$*.*

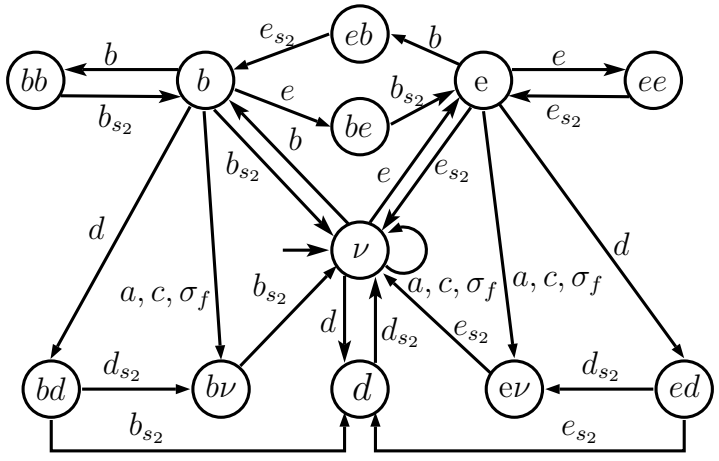
Na segunda iteração, o primeiro elemento da fila é então atribuído à variável u, ou seja,$u = a$*, e uma vez que u é diferente de* ν*, calcula-se o comprimento de u e atribuído a variável l e o conjunto* $I_l = \{1\}$ *é formado. Então , os conjuntos* $I_\nu = \emptyset$ *e* $I_{\setminus\nu} = I_l$ *são computadas. Nota-se que as Linhas* 18 *a* 20 *se o comprimento do sufixo de* $u = \sigma_1\sigma_2 \ldots \sigma_l$ *é menor ou igual ao atraso do canal de comunicação que transmite o evento* σ_y *para todo y em* I_ν*. Desse modo, na linhas* 18 *a* 20 *nenhuma transição do estado c é definida, uma vez que o canal* ch_{11}*, que transmite c, não esta sujeito a atrasos de comunicação. Por outro lado, de acordo com a Linhas* 21 *a* 23*, a transição do estado c para o estado* ν *rotulado como* c_{s_1} *é criada, que representa a observação com sucesso de c no* LD_1*. Ao terminar esta interação o estado c é adicionado ao conjunto* Q_1*, ou seja,* $Q_1 = \{\nu, a\}$ *e removido de F, tornando* $F = (c)$*.*

Na terceira interação, $u = d$*. O comprimento de u é calculado e atribuído a variável l e o conjunto* $I_l = \{1\}$ *é formado. Então, os conjuntos* $I_\nu = \emptyset$ *e* $I_{I\setminus\nu} = i_l$ *são computados. Sabendo que o canal* ch_{12} *transmite d, esta sujeito a um atraso de comunicação de dois passos, na Linha* 18 *a* 20*, as transições do estado c são definidas para todos* $\sigma \in \Sigma_{o_1}$ *como:* $\delta_1(d, c) = dc$ *e* $\delta_1(c, c) = cc$ *e nas Linhas* 21 *e* 22*, os estados dc e dd são adicionados no final da fila F e* $F = (dc, dd)$*. Feito isso, as transições do estado c são definidas para todos* $\sigma \in \Sigma_{uo_1}$ *como:* $\delta_1(d, b) = \delta_1(d, c) = \delta(d, \sigma_f)$*. Por último, retira todos os elementos da fila F até que* $F = \emptyset$*.*

Para modelar os atrasos de observações do LD_2*, necessita construir o autômato*



(a) D_1



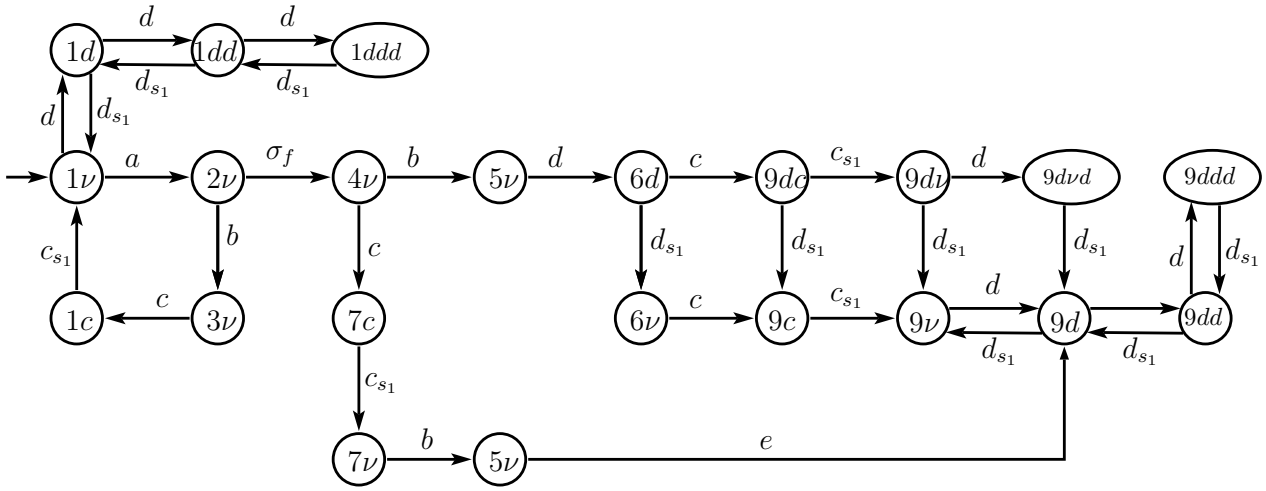
(b) D_2

Figura 13 – Autômatos D_1 e D_2 para o atraso de comunicação para o Exemplo 5

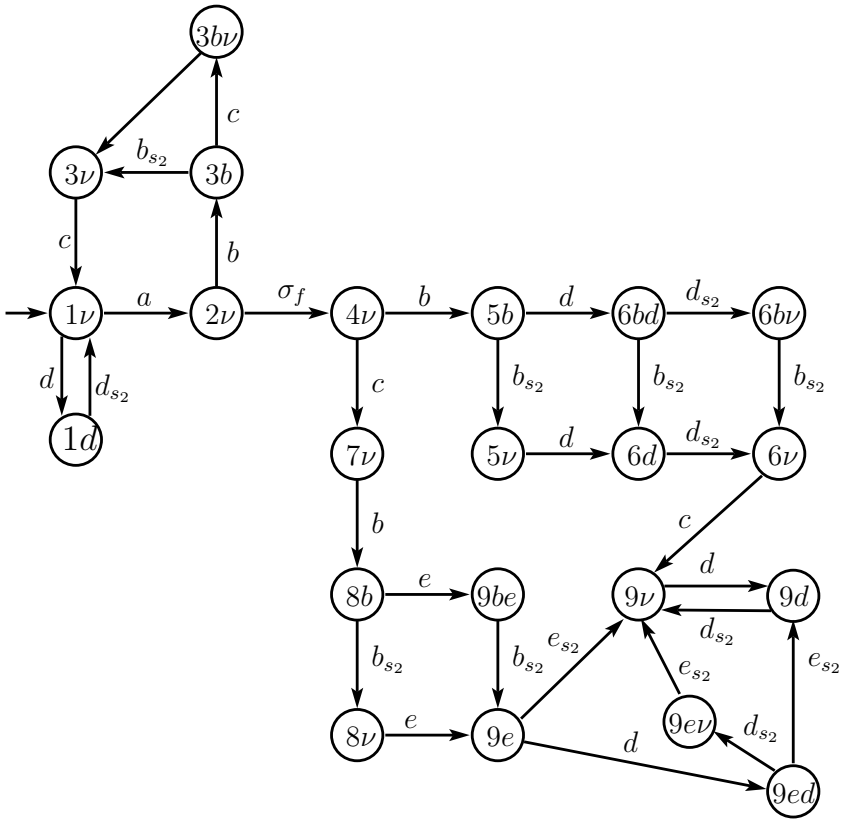
D_2 mostrado na Figura 13b, seguindo o mesmos passos da construção de D_1. É importante relembrar que os eventos b e e são comunicados pelo mesmo canal ch_{23}, a ordem de observação desses eventos não pode ser trocada.

Exemplo 6 *Considere a mesma planta e a arquitetura descentralizada presente no Exemplo 5. Os autômatos G_1 e G_2 mostrados nas Figuras 14a e b, foram construídos de acordo com a composição paralela $G_i = G||D_i$, para $i = 1, 2$, O conjunto de eventos observáveis e não observáveis de G_1 são $\Sigma_{1_o} = \{c_{s_1}, d_{s_1}\}$, e $\Sigma_{1_{uo}} = \{a, b, c, d, e, \sigma_f\}$, e o conjunto de eventos observáveis e não observáveis de G_2 são $\Sigma_{2_o} = \{b_{s_2}, d_{s_2}, e_{s_2}\}$ e $\Sigma_{2_{uo}} = \{a, b, c, d, e, \sigma_f\}$, respectivamente.*

As linguagens $L(G_1)$ e $L_(G_2)$ representa todas as ordens possíveis de observação das sequências $s \in L_G$, respeitando $\Sigma_{o_{i,j}}$ e $k_{i,j}$. para $j \in \{1, 2, 3\}$.



(a) G_1



(b) G_2

Figura 14 – Autômatos G_1 e G_2 para o atraso de comunicação para o Exemplo 6.

Com base nas definições apresentadas neste capítulos, é possível definir a codiagnosticabilidade sujeita a atrasos de comunicação de eventos como segue.

Definição 19 *Sejam L a linguagem gerada por G e L_N a linguagem formada por todas as sequências livres de falha de L. L é dita ser codiagnosticável sujeita a atrasos de comunicação de eventos em relação a $\chi_i : \Sigma^\star \to 2^{\Sigma_i^\star}$, $P_{s_i} : \Sigma_i^\star \to \Sigma_{o_i}^{s^\star}$, para $i = 1, \ldots, n$, e Σ_f se*

$$(\exists z \in \mathbb{N})(\forall s \in L \setminus L_M)(\forall st \in L \setminus L_N, \|t\| \geq z) \Rightarrow$$
$$(\exists i \in \{1, \ldots, n\})[(P_{s_i}(\chi_i(st)) \cap (P_{s_i}(\chi_i(\omega)) = \emptyset, \forall \omega \in L_N].$$

É importante destacar que em Nunes *et al.* (2018) é mostrado que $L(G_i) = \chi_i(L)$, em que L é a linguagem gerada pela planta G. Isso significa que os autômatos G_i representam todas as possíveis trocas de observação de eventos que podem ser observadas pelos diagnosticadores locais LD_i. Portanto, o método apresentado neste capítulo propõe novos modelos para cada diagnosticador local LD_i com base na planta, no conjunto de eventos observáveis local e nos valores de atraso máximo de cada canal de comunicação de eventos.

Além disso, como o método propõe a construção de modelos que representam todas as possíveis observações locais, esses modelos podem ser utilizados não apenas para o diagnóstico local, como também para a verificação da definição 19 de codiagnosticabilidade sujeita a atrasos de comunicação de eventos. Isso pode ser feito, como mostrado em Nunes *et al.* (2018) considerando também possíveis perdas de observação, adaptando-se o método de verificação apresentado em Moreira *et al.* (2011).

No próximo capítulo desta dissertação, o problema de diagnóstico síncrono descentralizado sujeito a atrasos de comunicação de eventos é explorado. O objetivo é adaptar a metodologia apresentada neste capítulo para uma arquitetura de diagnóstico em que os diagnosticadores locais são calculados a partir dos componentes do sistema e estudar as implicações dessa implementação.

4 DIAGNÓSTICO SÍNCRONO DESCENTRALIZADO SUJEITO A ATRASOS DE COMUNICAÇÃO DE EVENTOS

4.1 FORMULAÇÃO DO PROBLEMA

Neste trabalho, o problema do diagnóstico síncrono descentralizado sujeito a atrasos de comunicação de eventos entre locais de medição (tipicamente sensores) e os diagnosticadores locais é considerado. Para tanto, considere que o modelo do sistema G é composto por r componentes, ou seja, $G = \|_{i=1}^{r} G_i$ é associado a cada componente G_i, $i \in \{1, \ldots, r\}$, existe um diagnosticador local D_i, calculado a partir do comportamento livre de falha G_{N_i}, conforme apresentado em (CABRAL, F. G.; MOREIRA, 2020). Nessa configuração, diferentes locais de medição enviam informações acerca da ocorrência de eventos observados para os diagnosticadores locais D_i por meio de canais de comunicação $ch_{i,j}, j \in \{1, \ldots, \eta_i\}$, em que η_i é o número total de canais de comunicação conectados ao diagnosticador D_i.

A Figura 15 ilustra a arquitetura considerada neste trabalho. Na configuração ilustrada na Figura: 15, existem dois canais, $ch_{1,1}$ $ch_{1,2}$, que enviam informações acerca de observação de eventos na planta para o diagnosticador D_1. Já os canais $ch_{2,1}$, $ch_{2,2}$, e $ch_{2,3}$ enviam informações de observação de eventos para o diagnosticador local D_2 e os canais $ch_{r,1}$ e $ch_{r,2}$ enviam informações ao diagnosticador local D_r. É importante notar que o número de canais de comunicação que estão conectados a cada diagnosticador local não está associado ao número de componentes do sistema.

O conjunto de eventos do componente G_i é denotado por $\Sigma_i = \Sigma_{i,o} \dot{\cup} \Sigma_{i,u}$, em que $\Sigma_{i,o}$ e $\Sigma_{i,u}$ denotam os conjuntos de eventos observáveis e não observáveis de G_i, respectivamente, e $\dot{\cup}$ denota a união de conjuntos disjuntos. O conjunto de eventos observáveis de G é definido como $\Sigma_o = \cup_{i=1}^{r} \Sigma_{i,o}$. Os conjuntos de eventos observáveis

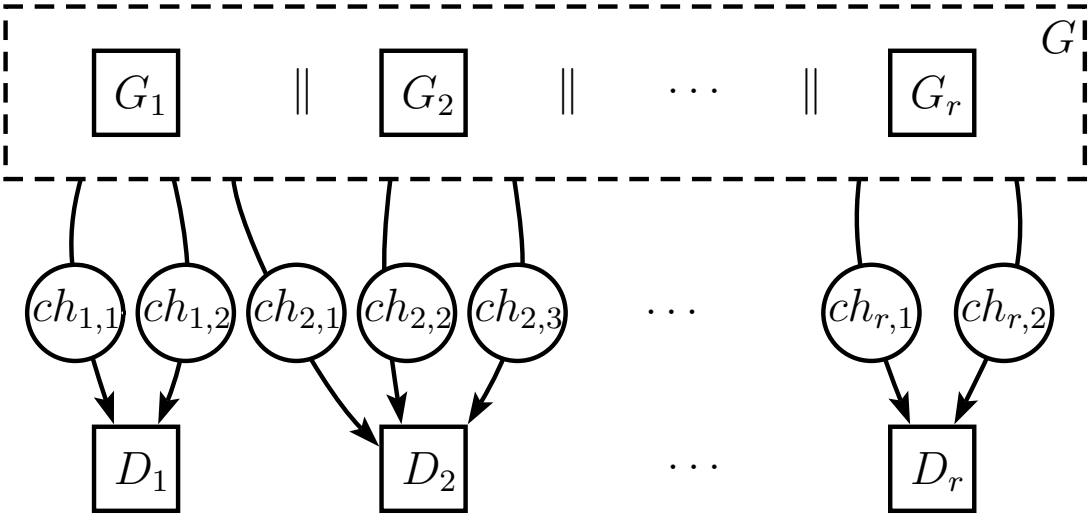


Figura 15 – Arquitetura do diagnóstico síncrono descentralizado.

locais $\Sigma_{i,o}$, para $i = 1, \ldots, r$, são definidos como $\Sigma_{i,o} = \cup_{j=1}^{\eta_i} \Sigma_o^{i,j}$, em que $\Sigma_o^{i,j}$ é o conjunto de eventos observáveis cuja observação é comunicada ao diagnosticador local D_i por meio do canal de comunicação $ch_{i,j}$.

A comunicação de eventos pelos canais de comunicação $ch_{i,j}$ nem sempre pode ser considerada como instantânea, podendo sofrer atrasos decorrentes da distância física entre a planta e diagnosticadores ou mesmo por causa de interferência eletromagnética. Como consequência desses possíveis atrasos de comunicação, a ordem com que os eventos são observados pelo diagnosticador local D_i pode ser trocada quando comparada com a ordem dos eventos gerados pela planta. Se esse fenômeno não for levado em consideração no projeto dos diagnosticadores locais, a observação em ordem trocada de eventos pode gerar falsos positivos, ou seja, o diagnosticador pode informar a ocorrência de uma falha que não ocorreu no sistema.

Com o objetivo de tornar o diagnóstico descentralizado robusto a atrasos de comunicação, como apresentado no capítulo anterior, Nunes *et al.* (2018) apresentaram uma alteração no modelo da planta para cada diagnosticador local. Essa alteração de modelo leva em consideração todas as possíveis trocas de observação de eventos pelos diagnosticadores locais por causa dos atrasos de comunicação. A técnica apresentada em Nunes *et al.* (2018) pode ser aplicada no contexto do diagnóstico descentralizado clássico, em que diagnosticadores locais são calculados com base no comportamento global da planta, considerando-se observação local de eventos. Entretanto, a extensão do método apresentado em Nunes *et al.* (2018) para arquiteturas síncronas de diagnóstico não foi explorada, sendo que se esse método fosse aplicado diretamente para o caso síncrono, levaria a um diagnóstico muito conservador, ou seja, levaria a um aumento da quantidade máxima de eventos gerados pela planta necessária para o diagnóstico, ou mesmo tornaria o sistema não diagnosticável de forma síncrona.

Assim, neste capítulo, o problema de atraso de comunicação de eventos para o diagnóstico síncrono descentralizado é considerado. Para tanto, assim como é feito em Nunes *et al.* (2018), as hipóteses **H1**-**H5** também são consideradas. Porém, a consideração da hipótese **H1** para o caso do diagnóstico síncrono descentralizado deve ser analisada com cuidado. Para tanto, **H1** é reapresentada a seguir.

H1. O atraso na comunicação de um evento $\sigma \in \Sigma_o^{i,j}$ é contado em passos (TRIPAKIS, 2004; NUNES *et al.*, 2018), sendo que um passo corresponde à ocorrência de um evento na planta G. Assim, o atraso é medido pelo número de eventos gerados em G após a ocorrência de σ e antes de sua efetiva comunicação ao diagnosticador D_i.

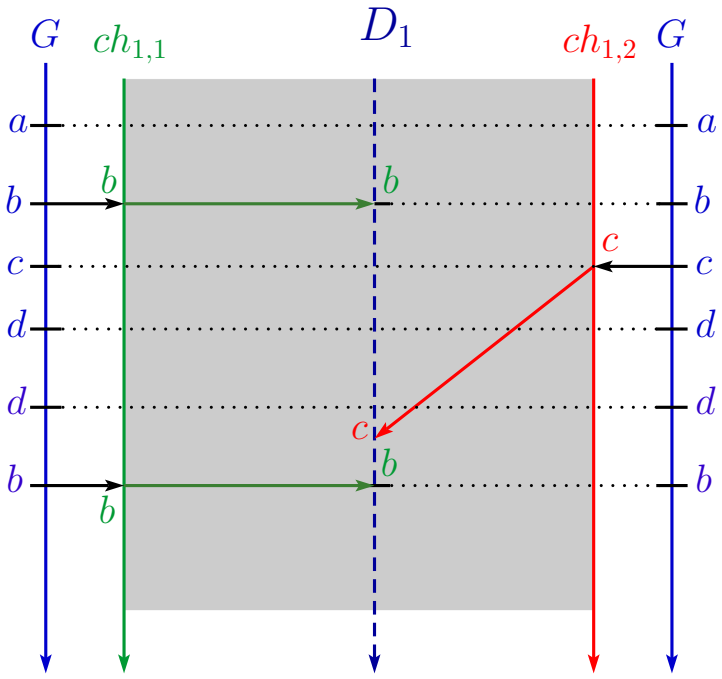
Com base na hipótese **H1**, o atraso máximo de cada canal de comunicação é medido de acordo com o número de eventos que são gerados na planta até que

o sinal (observação do evento) é efetivamente recebido pelo diagnosticador local D_i. Portanto, a referência para a medida do atraso máximo dos canais de comunicação é centralizada e dada pelo sistema global, como é considerado em Nunes *et al.* (2018). Entretanto, ao considerarmos diagnosticadores locais que são calculados a partir do comportamento livre de falha dos componentes do sistema, nem sempre o atraso de comunicação de um canal pode resultar em uma observação equivocada por determinado diagnosticador local. Para ilustrar esse fenômeno, considere o exemplo a seguir.

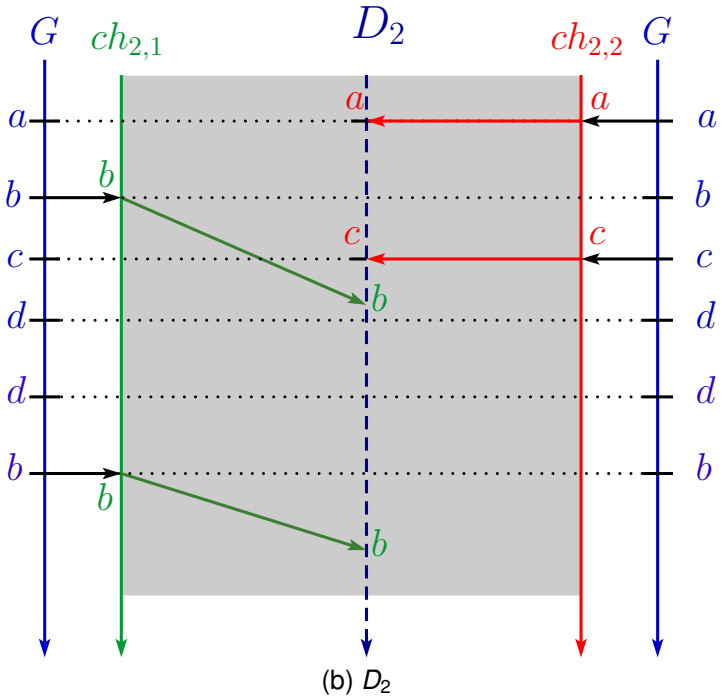
Exemplo 7 *Considere novamente a arquitetura de diagnóstico síncrono descentralizado apresentada na figura 15. Suponha que o sistema tenha três componentes tais que* $\Sigma_o = \Sigma_{1,o} \cup \Sigma_{2,o} \cup \Sigma_{3,o} = \{a, b, c, d\}$*, em que* $\Sigma_{1,o} = \{b, c\}$*,* $\Sigma_{2,o} = \{a, b, c\}$ *e* $\Sigma_{3,o} = \{c, d\}$*. Neste exemplo, consideramos apenas os diagnosticadores locais* D_1 *e* D_2 *que possuem dois canais de comunicação conectados a cada diagnosticador local,* $ch_{1,1}$*,* $ch_{1,2}$*,* $ch_{2,1}$ *e* $ch_{2,2}$*. Os eventos comunicados a* D_1 *e* D_2 *por meio dos canais* $ch_{1,1}$*,* $ch_{1,2}$*,* $ch_{2,1}$ *e* $ch_{2,2}$ *são* $\Sigma_o^{1,1} = \{b\}$*,* $\Sigma_o^{1,2} = \{c\}$*,* $\Sigma_o^{2,1} = \{b\}$ *e* $\Sigma_o^{2,2} = \{a, c\}$*, respectivamente. Suponha que o sistema tenha gerado a sequência* $s = abcddb$ *e que os canais de comunicação não possuem atrasos. Portanto, os diagnosticadores locais* D_1 *e* D_2 *observam as sequências* $s_{o_1} = P^o_{1,o}(s) = bcb$ *e* $s_{o_2} = P^o_{2,o}(s) = abcb$*, respectivamente, em que* $P^o_{i,o} : \Sigma_o^\star \to \Sigma_{i,o}^\star$*,* $i = 1, 2$*, são projeções.*

Suponha agora que os canais de comunicação $ch_{1,2}$ *and* $ch_{2,1}$ *possuem atraso máximo igual a dois, ou seja,* $k_{1,2} = k_{2,1} = 2$*. Portanto, um evento comunicado a* D_1 *e* D_2 *por meio dos canais* $ch_{1,2}$ *e* $ch_{2,1}$ *pode ser observado por* D_1 *e* D_2 *após duas ocorrências de eventos em G. Nesse cenário, após a ocorrência do primeiro evento b, o diagnosticador local* D_2 *pode observar o evento c antes do evento b por causa do atraso de comunicação do canal e, portanto, a observação da sequência s em* D_2 *se torna* $s'_{o_2} = acbb$*, que é diferente de* s_{o_2}*. Essa situação é ilustrada na figura 16 que mostra, por meio de uma linha do tempo, a diferença entre a ocorrência e a efetiva observação dos eventos por* D_2*. Por outro lado, no caso do diagnosticador local* D_1*, após a segunda ocorrência do evento c, o sistema pode gerar duas ocorrências do evento d, que não é definido para o componente* G_1 *e o diagnosticador* D_1 *observa corretamente a sequência gerada pelo sistema. Este exemplo mostra que embora exista um atraso de comunicação em um determinado canal, esse atraso pode não ocasionar nenhum efeito prático para o diagnosticador local, uma vez que sua medida é feita a partir de um referencial central, baseado nas ocorrências de eventos da planta global.*

O exemplo 7 mostra que é necessário fazer uma adaptação do método proposto em Nunes *et al.* (2018) para considerar a arquitetura síncrona descentralizada. Uma vez que, neste trabalho, os diagnosticadores locais são calculados a partir dos com-



(a) D_1



(b) D_2

Figura 16 – Linha do tempo para exemplo 7.

ponentes do sistema, é necessário converter a referência de atraso de eventos em relação ao sistema global para uma referência local, em relação aos diagnosticadores locais. A proposta de mudança de referencial para o atraso é apresentada na seção seguinte.

4.2 CONVERSÃO DO ATRASO MÁXIMO PARA O DIAGNÓSTICO SÍNCRONO DESCENTRALIZADO

Para realizar o diagnóstico síncrono descentralizado sujeito a atrasos de comunicação, é necessário modelar a consequência do atraso de comunicação nos modelos dos componentes que são usados para gerar os diagnosticadores locais. Para tanto, nesta seção, propomos um método para converter o referencial do atraso de comunicação de eventos da planta global para os diagnosticadores locais.

Uma vez que os eventos gerados pela planta são usados como referência para os atrasos de cada canal de comunicação, é possível que o atraso máximo de comunicação $k_{i,j}$ definido para o canal $ch_{i,j}$ não seja observado para o diagnosticador local D_i. Nesse caso, um valor menor do que $k_{i,j}$ deve ser considerado. Isso pode ser feito analisando-se o comportamento global do sistema para identificar as situações em que o valor de $k_{i,j}$ pode ser menor, tornando o diagnóstico síncrono mais preciso.

Antes de apresentar o método de conversão do referencial de atraso máximo para o diagnóstico síncrono, é necessário introduzir as seguintes funções.

Definição 20 *As seguintes funções podem ser definidas:*

1. *Projeção* $P_{i,j} : \Sigma^\star \to (\Sigma_i \setminus \Sigma_o^{i,j})^\star$*;*

2. *Função de pós-linguagem limitada a* ν*,* $\ell : Q \times \Sigma \times \mathbb{N} \to \Sigma^\star$ *definida como:*

$$\ell(q, \sigma, \nu) = \{s \in \Sigma^\star : f(q, \sigma s)! \wedge \|s\| = \nu\}; \tag{11}$$

3. *Função de comprimento máximo de sequências* $\mu : \Sigma^\star \to \mathbb{N}$ *definida como:*

$$\mu(A) = max(\|s\| : s \in A), \tag{12}$$

   em que A é uma linguagem;

4. *Função de renomeação de eventos* $R_i^d : \Sigma_o^{i,j} \times k \times i \times j \to \Sigma_{i,o}^R$*,* $k \in \mathbb{N}$*,* $k \leq k_{i,j}$*,* $i \in \{1, \ldots, r\}$*,* $j \in \{1, \ldots, \eta_i\}$*.* $R_i^d(\sigma)$ *registra o valor de atraso máximo correto em que* $\sigma \in \Sigma_{i,o}$ *pode ser observado em cada evento* σ*:*

$$R_i^d(\sigma, k, i, j) = \sigma_k^{i,j}. \tag{13}$$

A partir da definição 20, o algoritmo 3 que calcula os modelos G_{d_i}, $i \in \{1, \ldots, r\}$, a partir de G_i, é apresentado. Os eventos observáveis de G_{d_i} são modificados para

Algoritmo 3 Modelos dos componentes do sistema com registro de atraso máximo de eventos.

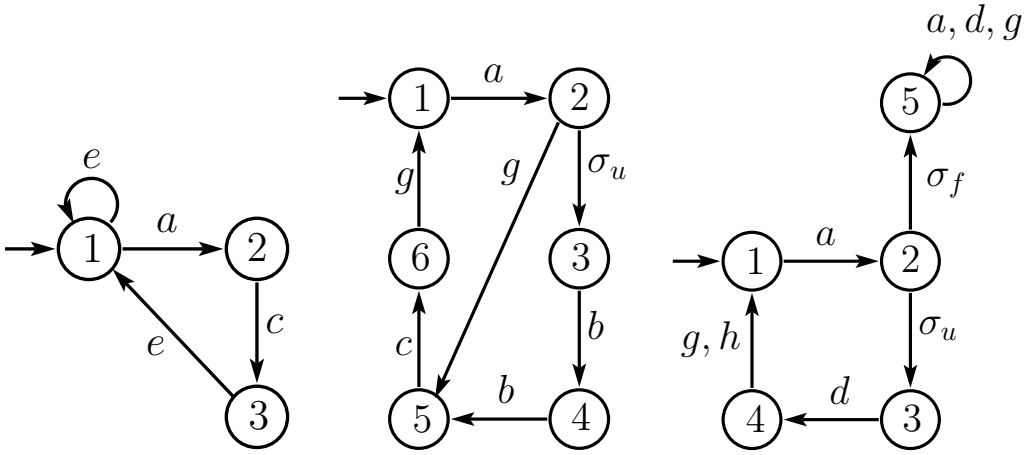
Entradas: $G = (Q, \Sigma, f, q_0)$, $G_i = (Q_i, \Sigma_i, f_i, q_{0,i})$, $i = 1, \ldots, r$, $\Sigma_o^{i,j}$, $k_{i,j}$, e $j \in \{1, \ldots, \eta_i\}$.
Saídas: $G_{d_i} = (Q_i, \Sigma_{d_i} = \Sigma_{i,u} \dot{\cup} \Sigma_{i,o}^R, f_{d_i}, q_{0,i})$, $i = 1, \ldots, r$.

1: $\Sigma_{i,o} \leftarrow \emptyset$, $\Sigma_{i,o}^R \leftarrow \emptyset$, $k \leftarrow 0$, $B \leftarrow \emptyset$.
2: **para todo** $i \in \{1, \ldots, r\}$ **faça**
3: **para todo** $j \in \{1, \ldots, \eta_i\}$ **faça**
4: **para todo** (q_i, σ, q_i') de G_i tal que $\sigma \in \Sigma_o^{i,j}$ **faça**
5: **se** $k_{i,j} = 0$ **então**
6: $\Sigma_{i,o}^R \leftarrow \Sigma_{i,o}^R \cup R_i^d(\sigma, k_{i,j}, i, j)$.
7: Defina $f_{d_i}(q_i, R_i^d(\sigma, k_{i,j}, i, j)) = f_i(q_i, \sigma)$.
8: **senão se** $k_{i,j} \neq 0$ **então**
9: **para todo** (q, σ, q') de G, em que q_i é a $i$ésima coordenada de q **faça**
10: $B \leftarrow B \cup \ell(q, \sigma, k_{i,j})$.
11: $k \leftarrow \mu(P_{i,j}(B))$.
12: $\Sigma_{i,o}^R \leftarrow \Sigma_{i,o}^R \cup R_i^d(\sigma, k, i, j)$.
13: Defina $f_{d_i}(q_i, R_i^d(\sigma, k, i, j)) = f_i(q_i, \sigma)$.
14: $B \leftarrow \emptyset$, $k \leftarrow 0$.
15: **para todo** $f_i(q_i, \sigma)!$, em que $\sigma \in \Sigma_{i,u}$ e $q_i \in Q_i$ **faça**
16: Defina $f_{d_i}(q_i, \sigma) = f_i(q_i, \sigma)$.

registrar o atraso máximo possível em que podem ser observados pelos diagnosticadores locais D_i.

A ideia do algoritmo 3 é mapear as transições dos componentes locais verificando o atraso máximo possível que de fato pode ocorrer para os diagnosticadores locais. Para tanto, verifica-se na planta G quais situações permitem ou não uma troca de observação em um diagnosticador local D_i construído a partir do componente G_i. Uma vez detectado o real atraso máximo que pode ocorrer por transição observável, o evento σ que rotula essa transição em G_i é renomeado para registrar o canal por onde σ é transmitido e o valor de atraso máximo convertido para o diagnosticador local D_i. O exemplo 8 ilustra a construção dos autômatos G_{d_i}, $i = 1, \ldots, r$, utilizando-se o algoritmo 3.

Exemplo 8 *Considere uma planta G formada por três componentes,* G_1, G_2 *e* G_3, *apresentados na figura 17. O sistema completo* $G = G_1 \| G_2 \| G_3$ *pode ser visto na figura 18. Neste exemplo,* $\Sigma_1 = \Sigma_{1,o} \dot{\cup} \Sigma_{1,u} = \{a, c, e\}$, $\Sigma_2 = \Sigma_{2,o} \dot{\cup} \Sigma_{2,u} = \{a, b, c, g, \sigma_u\}$ *e* $\Sigma_3 = \Sigma_{3,o} \dot{\cup} \Sigma_{3,u} = \{a, d, g, h, \sigma_u, \sigma_f\}$, *em que* $\Sigma_{1,o} = \{a, c\}$, $\Sigma_{1,u} = \{e\}$, $\Sigma_{2,o} = \{a, b, c, g\}$, $\Sigma_{2,u} = \{\sigma_u\}$, $\Sigma_{3,o} = \{a, d, g, h\}$ *e* $\Sigma_{3,u} = \{\sigma_u, \sigma_f\}$. *Nesse cenário, os canais de comunicação são* $ch_{1,1}$, $ch_{1,2}$, $ch_{2,1}$, $ch_{2,2}$, $ch_{2,3}$, $ch_{3,1}$ *e* $ch_{3,2}$ *tais que* $\Sigma_o^{1,1} = \{c\}$, $\Sigma_o^{1,2} = \{a\}$, $\Sigma_o^{2,1} = \{c\}$, $\Sigma_o^{2,2} = \{a, b\}$, $\Sigma_o^{2,3} = \{g\}$, $\Sigma_o^{3,1} = \{d, g, h\}$ *e* $\Sigma_o^{3,2} = \{a\}$. *Os atrasos máximos dos canais*



(a) G_1 (b) G_2 (c) G_3

Figura 17 – Modelos dos componentes G_1, G_2, e G_3 do exemplo 8.

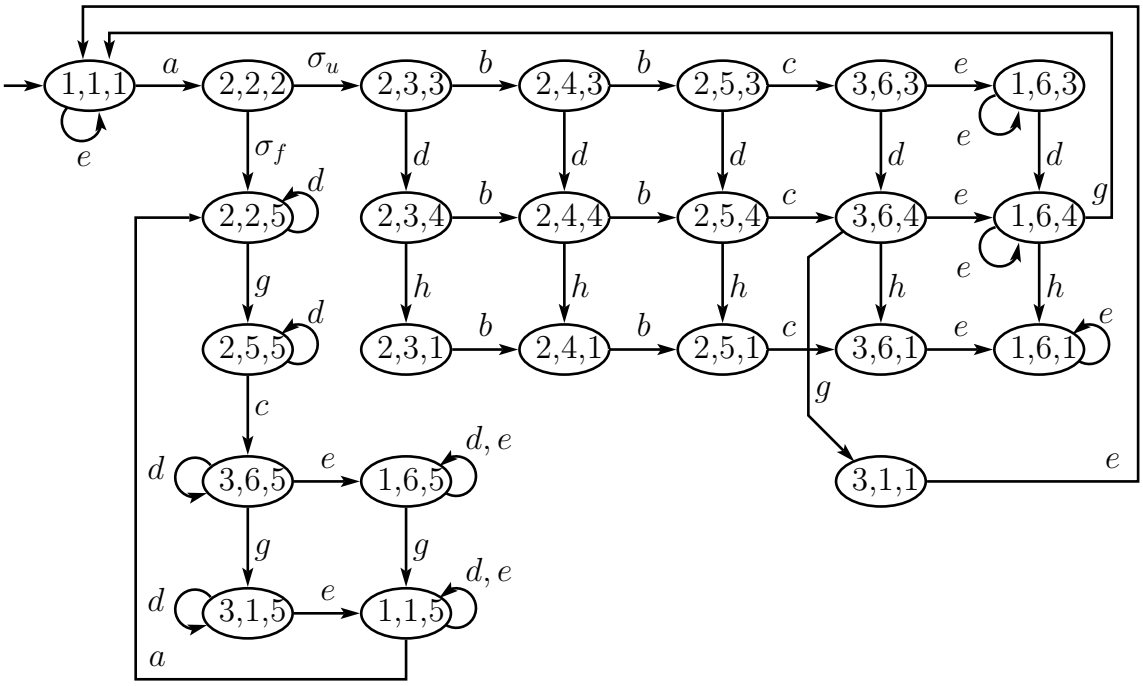


Figura 18 – Modelo da planta $G = G_1 \| G_2 \| G_3$ do exemplo 8.

de comunicação são $k_{1,1} = k_{2,2} = k_{3,1} = 0$, $k_{2,3} = k_{3,2} = 1$, *e* $k_{1,2} = k_{2,1} = 2$.

Aplicando-se o algoritmo 3 aos autômatos G_1, G_2 *e* G_3, *obtém-se os autômatos* G_{d_1}, G_{d_2} *e* G_{d_3} *apresentados na figura 20. Para ilustrar o uso do algoritmo 3, considere a transição* $(1, a, 2)$ *do autômato* G_1 *da figura 17. Como o canal de comunicação* $ch_{1,2}$, *por onde o evento a é comunicado, tem atraso máximo* $k_{1,2} = 2$, *é necessário verificar todas as sequências s de G com comprimento igual a* $k_{1,2} = 2$ *após a ocorrência do evento a nos estados* $q = (1, q_2, q_3)$, *em que* $q_2 \in Q_2$ *e* $q_3 \in Q_3$, *representado na Figura 19. Os estados de G cuja primeira coordenada é 1 e o evento a é viável são* $(1, 1, 1)$ *e* $(1, 1, 5)$ *e as possíveis sequências com comprimento igual a* $k_{1,2} = 2$ *após*

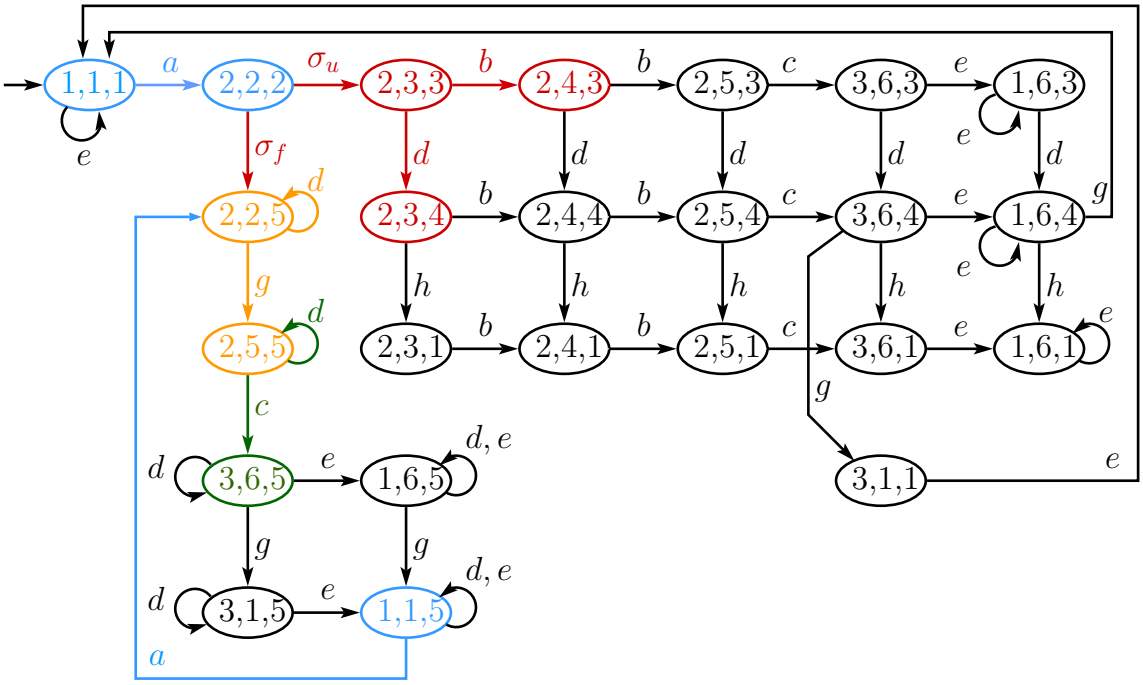


Figura 19 – Possíveis sequências com comprimento igual a $k_{1,2} = 2$.

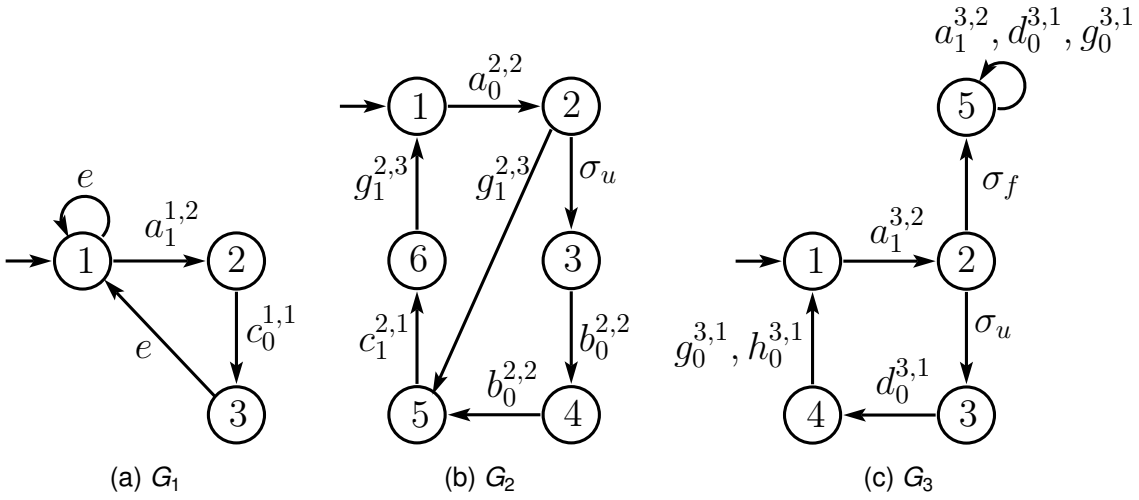


Figura 20 – Modelos dos componentes com registro do máximo atraso de observação de eventos G_{d_1}, G_{d_2} e G_{d_3} do exemplo 8.

a ocorrência do evento a a partir dos estados $(1,1,1)$ *e* $(1,1,5)$ *são* $s_1 = \sigma_u b, s_2 = \sigma_u d, s_3 = \sigma_f d, s_4 = \sigma_f g, s_5 = dd, s_6 = dg, s_7 = gc$, *e* $s_8 = gd$. *Uma vez que essas sequências são calculadas, na linha 11 do algoritmo 3, o valor de k é atualizado com o comprimento máximo de* $\|P_{1,2}(s_l)\|$, *para* $l = 1, \ldots, 8$, *em que* $P_{1,2} : \Sigma^\star \to (\Sigma_1 \setminus \Sigma_o^{1,2})^\star$, *que é, nesse caso, igual a* $\|P_{1,2}(s_7)\| = \|P_{1,2}(gc)\| = \|c\| = 1$. *Portanto, o evento a que rotula a transição* $(1, a, 2)$ *em* G_1 *é renomeado para* $a_k^{1,2} = a_1^{1,2}$, *resultando na transição* $(1, a_1^{1,2}, 2)$ *em* G_{d_1}. *Isso significa que o atraso máximo para observação do evento a no diagnosticador local* D_1 *nesse caso é igual a* $1 < k_{1,2} = 2$.

Observação 1 *É importante observar que ao aplicar o algoritmo 3 em G_i, o conjunto $\Sigma^R_{i,o}$ pode ter $\sum_{j=1}^{\eta_i}(k_{i,j}+1)\times|\Sigma_o^{i,j}|$ eventos no pior caso em que todos os eventos comunicáveis são renomeados com um valor diferente de atraso, em que $|.|$ denota cardinalidade.*

4.3 MODELOS DOS COMPONENTES DO SISTEMA SUJEITOS A ATRASOS NA OBSERVAÇÃO DE EVENTOS

Uma vez obtido os autômatos G_{d_i} utilizando-se o algoritmo 3, os modelos em autômato dos componentes do sistema sujeitos a atrasos na observação de eventos para o diagnóstico síncrono descentralizado, denotados por G_{δ_i}, podem ser obtidos. Para isso, como em Nunes *et al.* (2018), é necessário distinguir um evento $\sigma \in \Sigma_{i,o}$ que pode ocorrer na planta de sua observação pelo diagnosticador local D_i. Isso é feito criando-se um evento σ_s que representa a observação com sucesso de σ. Como os autômatos G_{d_i}, $i = 1,\ldots,r$, são usados neste trabalho para representar o atraso máximo de comunicação correto, o evento $\sigma^{i,j}_{k,s}$ é criado a partir do evento $\sigma^{i,j}_k \in \Sigma^R_{i,o}$ em G_{d_i}. Nesse contexto, os seguintes conjuntos de eventos são considerados.

$$\Sigma^{R,s}_{i,o} = \{\sigma^{i,j}_{k,s} : \sigma^{i,j}_k \in \Sigma^R_{i,o}\}, i = 1,\ldots,r. \tag{14}$$

Quando um evento observável $\sigma \in \Sigma_{i,o}$ ocorre na planta G, σ é comunicado ao diagnosticador D_i, com um possível atraso máximo $k_{i,j}$, pelo canal $ch_{i,j}$. O algoritmo 3 converte a referência de atraso máximo da planta para o diagnosticador local e registra o valor de atraso convertido $k \leq k_{i,j}$ ao criar o evento $\sigma^{i,j}_k \in \Sigma^R_{i,o}$. Para diferenciar a observação de fato do evento σ em D_i da ocorrência de σ com registro de atraso, $\sigma^{i,j}_k$, o evento $\sigma^{i,j}_{k,s}$ é criado, sendo $\Sigma^s_{i,o}$, obtido a partir da equação (14), o conjunto dos eventos que são observados pelo diagnosticador local D_i com o registro tanto do canal por onde foram comunicados quanto do valor de atraso máximo com referência local. Com base nisso, os seguintes conjuntos de eventos podem ser criados.

$$\Sigma_{\delta_i} = \Sigma_{d_i} \cup \Sigma^s_{i,o}, i = 1,\ldots,r. \tag{15}$$

Nesse caso, os eventos Σ_{d_i} são considerados não observáveis para o diagnosticador local D_i e os eventos de $\Sigma^s_{i,o}$ são considerados observáveis para o diagnosticador local D_i.

Para se obter os modelos dos componentes do sistema sujeitos a atrasos de comunicação de eventos G_{δ_i}, primeiro é necessário calcular os modelos em autômato do atraso de comunicação de eventos para cada diagnosticador local, denotados por Δ_i. Para isso, neste trabalho, o método proposto em Nunes *et al.* (2018), apresentado no capítulo anterior, é adaptado para considerar o efeito da mudança de referencial de atraso implementada no algoritmo 3.

Antes de apresentar o algoritmo de construção dos autômatos Δ_i, é necessário definir as seguintes funções, adaptadas de (NUNES *et al.*, 2018) (apresentadas no capítulo anterior) para a arquitetura e notação consideradas neste trabalho.

Definição 21 *Seja $\Sigma_i = \Sigma_{i,o} \dot{\cup} \Sigma_{i,u}$ e considere o conjunto $\Sigma_{i,o\gamma} = \Sigma_{i,o} \cup \{\gamma\}$. Considere também o conjunto de estados Q_{Δ_i}, em que cada estado $q \in Q_{\Delta_i}$ é rotulado com uma sequência $s \in \Sigma^{\star}_{i,o\gamma}$. As seguintes funções são definidas:*

1. *A função de substituição sub é definida como:*

$$sub : Q_{\Delta_i} \times \mathbb{N} \rightarrow Q_{\Delta_i}$$

em que para todo $q = q_1 q_2 \dots q_l \in Q_{\Delta_i}$,

$$sub(q, n) = \begin{cases} q_1 q_2 \dots q_{n-1} \gamma q_{n+1} \dots q_l, \text{ se } n \leq l \\ \text{indefinido, caso contrário.} \end{cases}$$

2. *A função de eliminação elim é definida como:*

$$elim : Q_{\Delta_i} \rightarrow Q_{\Delta_i}$$

em que para todo $q = q_1 q_2 \dots q_l \in Q_{\Delta_i}$,

$$elim(q) = \begin{cases} q_n q_{n+1} \dots q_l, \text{ se } (\exists n \leq l)[(q_n \neq \gamma) \wedge (q_m = \gamma, \forall m \in \{1, 2, \dots, n-1\})] \\ \gamma, \text{ se } q_m = \gamma, \forall m \in \{1, 2, \dots, l\}. \end{cases}$$

3. *A função de registro de máximo atraso de comunicação dr é definida como:*

$$dr : \Sigma^{R}_{i,o} \rightarrow \{0, 1, \dots, k_{i,max}\},$$

em que $k_{i,max} = max(k_{i,j})$, para $j \in \{1, \dots, \eta_i\}$, e para todo $\sigma^{i,j}_k \in \Sigma^{R}_{i,o}$,

$$dr(\sigma^{i,j}_k) = k$$

4. *A função bijetora ω_i é definida, para $i = 1, \dots, r$, como*

$$\begin{aligned} \omega_i : & \Sigma^{s}_{i,o} \rightarrow \Sigma^{R}_{i,o}, \\ & \sigma^{i,j}_{k,s} \mapsto \omega_i(\sigma^{i,j}_{k,s}) = \sigma^{i,j}_k, \end{aligned}$$

e ω_i pode ser estendida para conjuntos de eventos da seguinte forma:

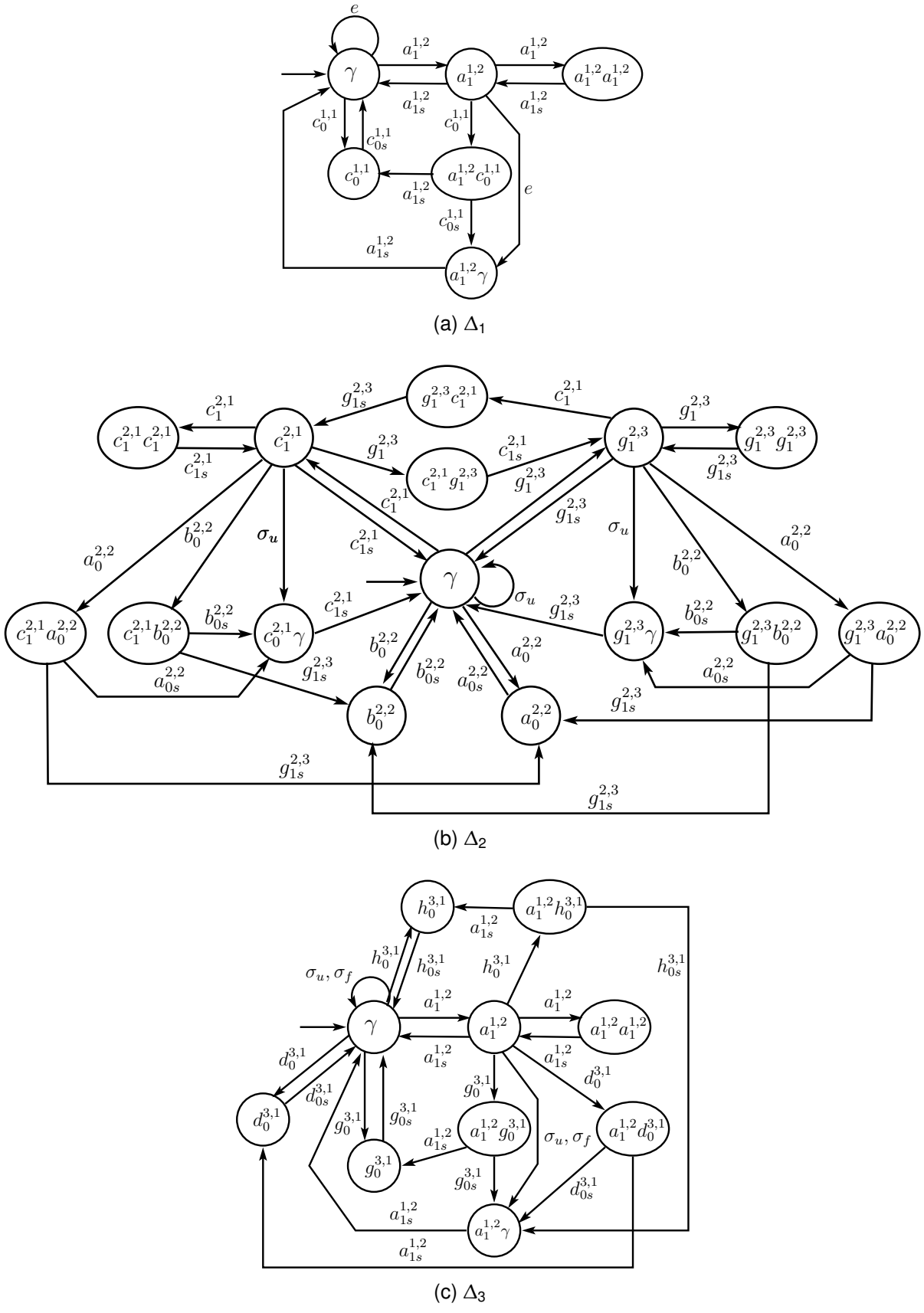
$$\omega_i(\Sigma^{s}_{i,o}) = \bigcup_{\sigma^{i,j}_{k,s} \in \Sigma^{s}_{i,o}} \omega_i(\sigma^{i,j}_{k,s})$$

A definição 21 estabelece algumas funções necessárias para a construção dos autômatos que modelam o atraso de comunicação de eventos Δ_i. A função $sub(q, i)$ substitui o i-ésimo elemento do estado q por γ para representar que um evento ocorreu, embora o conhecimento de qual evento tenha sido gerado não seja relevante. A função $elim(q)$ elimina o maior prefixo de q formado por elementos γ e a função $dr(\sigma_k^{i,j})$ retorna o atraso máximo que deve ser considerado para o evento σ. A função $\omega_i(\sigma_{k,s}^{i,j})$ retorna o evento $\sigma_k^{i,j}$ cuja observação bem sucedida é representada por $\sigma_{k,s}^{i,j}$.

O algoritmo 4 é proposto como um modelo do atraso de comunicação dos canais de comunicação de eventos da planta para os diagnosticadores locais D_i. O algoritmo 4 calcula os autômatos que modelam o comportamento dos atrasos de eventos para cada diagnosticador local D_i, Δ_i. No algoritmo 4 as operações $Enqueue(F, q)$ e $Dequeue(F)$ denotam a adição do elemento q à fila F e a retirada do primeiro elemento da fila F, respectivamente. Note que os estados de Δ_i registram as ocorrências de eventos na planta de tal forma que é possível acompanhar todas as possíveis trocas de observação por conta dos atrasos de comunicação. O exemplo a seguir mostra a construção dos autômatos Δ_i de acordo com o algoritmo 4.

Exemplo 9 *Considere novamente o sistema apresentado no exemplo 8. O algoritmo 4 é usado para se obter os modelos Δ_1, Δ_2 e Δ_3, ilustrados na figura 21. Para isso, o algoritmo é inicializado nas linhas 1 e 2 e, na linha 3, os conjuntos de eventos $\Sigma_{\Delta_1} = \Sigma_{d_1} \cup \Sigma_{1,o}^{R,s}$, $\Sigma_{\Delta_2} = \Sigma_{d_2} \cup \Sigma_{2,o}^{R,s}$ e $\Sigma_{\Delta_3} = \Sigma_{d_3} \cup \Sigma_{3,o}^{R,s}$ são formados, em que $\Sigma_{d_1} = \{a_1^{1,2}, c_0^{1,1}, e\}$, $\Sigma_{1,o}^{R,s} = \{a_{1,s}^{1,2}, c_{0,s}^{1,1}\}$, $\Sigma_{d_2} = \{c_1^{2,1}, a_0^{2,2}, b_0^{2,2}, g_1^{2,3}, \sigma_u\}$, $\Sigma_{2,o}^{R,s} = \{c_{1,s}^{2,1}, a_{0,s}^{2,2}, b_{0,s}^{2,2}, g_{1,s}^{2,3}\}$, $\Sigma_{d_3} = \{d_0^{3,1}, g_0^{3,1}, h_0^{3,1}, a_1^{3,2}, \sigma_u, \sigma_f\}$ e $\Sigma_{3,o}^{R,s} = \{d_{0,s}^{3,1}, g_{0,s}^{3,1}, h_{0,s}^{3,1}, a_{1,s}^{3,2}\}$. Após isso, os autômatos Δ_1, Δ_2 e Δ_3 são construídos levando-se em consideração todas as possibilidades de observação dos eventos $\Sigma_{i,o}^{R,s}$, para $i = 1, 2, 3$. Por exemplo, note que em Δ_1, é possível que as sequências $s_1 = a_1^{1,2} c_0^{1,1} a_{1,s}^{1,2} c_{0,s}^{1,1}$ e $s_2 = a_1^{1,2} c_0^{1,1} c_{0,s}^{1,1} a_{1,s}^{1,2}$ sejam geradas a partir do estado inicial γ. Isso mostra que as observações possíveis pelo diagnosticador local D_1 são $a_{1,s}^{1,2} c_{0,s}^{1,1}$ e $c_{0,s}^{1,1} a_{1,s}^{1,2}$, que representam a observação de ac ou ca. Esse caso ilustra que, apesar da ordem correta de observação ser a sequência ac, é possível que, por causa do atraso máximo de comunicação do evento a ser 1, o diagnosticador local poderá observar ac, se não houver atraso de comunicação, ou ca, caso haja atraso na comunicação do evento a.*

Em Nunes *et al.* (2018) é mostrado que os autômatos Δ_i de fato representam todas as possibilidades de observação considerando-se os possíveis atrasos de comunicação de eventos para a arquitetura descentralizada clássica apresentada no Protocolo 3 de Debouk *et al.* (2002). Neste trabalho, o método é adaptado para considerar, além da observação local dos eventos, a possibilidade do mesmo evento poder sofrer atrasos de observação distintos. Para tanto, os eventos são renomeados e en-



(a) Δ_1

(b) Δ_2

(c) Δ_3

Figura 21 – Autômatos Δ_1, Δ_2 e Δ_3 do exemplo 9.

Algoritmo 4 Modelos dos atrasos de comunicação de eventos para o diagnóstico síncrono descentralizado

Entradas: $\Sigma_{d_i} = \Sigma_{i,u} \dot{\cup} \Sigma_{i,o}^{R}$, $i = 1, \ldots, r$.
Saídas: $\Delta_i = (Q_{\Delta_i}, \Sigma_{\Delta_i}, f_{\Delta_i}, q_{0,\Delta_i})$, $i = 1, \ldots, r$.

para $i = 1, \ldots, r$ **faça**
  Defina $q_{0,\Delta_i} = \gamma$ e $Q_{\Delta_i} = \emptyset$.
  Calcule $\Sigma_{i,o}^{R,s}$ de acordo com a equação (14), e defina $\Sigma_{\Delta_i} = \Sigma_{d_i} \cup \Sigma_{i,o}^{R,s}$.
  $F \leftarrow (q_{0,\Delta_i})$, em que F denota uma fila FIFO.
  enquanto $F \neq \emptyset$ **faça**
    $u \leftarrow head[F]$.
    se $u = q_{0,\Delta_i}$ **então**
      para todo $\sigma \in \Sigma_{d_i}$ **faça**
        Calcule $\tilde{q} = f_{\Delta_i}(u, \sigma) = \begin{cases} \sigma, \text{ se } \sigma \in \Sigma_{i,o}^{R} \\ u, \text{ se } \sigma \in \Sigma_{i,u} \end{cases}$
        se $\tilde{q} \neq u$ **então**
          $Enqueue(F, \tilde{q})$
      $Q_{\Delta_i} \leftarrow Q_{\Delta_i} \cup \{u\}$
      $Dequeue(F)$
    senão se $u \neq q_{0,\Delta_i}$ **então**
      Faça $l = \|u\|$ e forme o conjunto $I_l = \{1, 2, \ldots, l\}$.
      Defina $u = \sigma_1 \sigma_2 \ldots \sigma_l$ e calcule $I_\gamma = \{x \in I_l : (\exists \sigma_x \in u)[\sigma_u = \gamma]\}$.
      Calcule $I_{l \setminus \gamma} = I_l \setminus I_\gamma$.
      para todo $\sigma \in \Sigma_{i,o}^{R}$ **faça**
        $\tilde{q} = f_{\Delta_i}(u, \sigma) = \begin{cases} u\sigma, \text{ if } \|\sigma_x \sigma_{x+1} \ldots \sigma_l\| \leq dr(\sigma_x), \forall x \in I_{l \setminus \gamma} \\ \text{indefinido, caso contrário} \end{cases}$
        Se $\tilde{q}$ é indefinido, $Enqueue(F, \tilde{q})$.
      para todo $\sigma \in \Sigma_{i,u}$ **faça**
        $\tilde{q} = f_{\Delta_i}(u, \sigma) = \begin{cases} u\gamma, \text{ if } \|\sigma_x \sigma_{x+1} \ldots \sigma_l\| \leq dr(\sigma_x), \forall x \in I_{l \setminus \gamma} \\ \text{indefinido, caso contrário} \end{cases}$
        Se $\tilde{q} \notin F$, $Enqueue(F, \tilde{q})$.
      para todo $j = 1, 2, \ldots, \eta_i$ **faça**
        Crie o conjunto $X = \{x : (\sigma_x \in u) \wedge (\sigma_x \in \omega_i(\Sigma_{i,o}^{R,s}))\}$.
        Se $X = \emptyset$, então calcule $\hat{x} = min(X)$ e $\tilde{q} = f_{\Delta_i}(u, \omega_i^{-1}(\sigma_{\hat{x}})) = elim(sub(u, \hat{x}))$.
        Se $(\tilde{q} \notin Q_{\Delta_i}) \wedge (\tilde{q} \notin F)$, $Enqueue(F, \tilde{q})$.
      Defina $Q_{\Delta_i} \leftarrow Q_{\Delta_i} \cup \{u\}$.
      $Dequeue(F)$

carados como eventos distintos, preservando-se o resultado apresentado em Nunes *et al.* (2018).

Após o cálculo dos modelos dos atrasos de comunicação de eventos Δ_i, esses modelos precisam ser considerados junto com as plantas locais com registro do atraso máximo convertido obtidas no algoritmo 3. Para isso, realizamos a composição

paralela entre G_{d_i}, obtidos a partir do algoritmo 3, e Δ_i, obtidos a partir do algoritmo 4, obtendo-se, assim, os modelos dos componentes locais considerando-se todas as possibilidades de reordenamento de eventos ocasionadas pelos atrasos de comunicação, G'_{d_i}:

$$G'_{d_i} = G_{d_i} \| \Delta_i = (Q'_{d_i}, \Sigma_{d_i} \cup \Sigma^{R,s}_{i,o}, f'_{d_i}, q'_{0,d_i}).$$

Uma vez que G'_{d_i} é obtido, os rótulos relacionados ao canal de comunicação $ch_{i,j}$ e ao atraso máximo com referência convertida k dos eventos $\sigma^{i,j}_k$ e $\sigma^{i,j}_{k,s}$ podem ser retirados, levando aos autômatos que modelam o comportamento dos componentes do sistema sujeitos a atrasos de comunicação de eventos G_{δ_i}. Antes de apresentar o algoritmo que calcula os autômatos G_{δ_i} a partir de G'_{d_i}, os conjuntos $\Sigma^s_{i,o}$ são definidos da seguinte forma:

$$\Sigma^s_{i,o} = \{\sigma_s : \sigma^{i,j}_{k,s} \in \Sigma^{R,s}_{i,o}\}, i = 1, \ldots, r. \tag{16}$$

Além disso, a função R_i é definida como segue:

$$R_i : \Sigma^R_{i,o} \cup \Sigma^{R,s}_{i,o} \rightarrow \Sigma_{i,o} \cup \Sigma^s_{i,o}, \tag{17}$$

em que, para todo $\sigma^{i,j}_k \in \Sigma^R_{i,o}$:

$$R_i(\sigma^{i,j}_k) = \sigma,$$

e para todo $\sigma^{i,j}_{k,s} \in \Sigma^{R,s}_{i,o}$:

$$R_i(\sigma^{i,j}_{k,s}) = \sigma_s.$$

O algoritmo 5 mostra como autômatos G_{δ_i} são calculados a partir de G'_{d_i} para $i = 1, \ldots, r$.

Algoritmo 5 Modelos dos componentes sujeitos a atrasos de comunicação de eventos

Entradas: $G'_{d_i} = (Q'_{d_i}, \Sigma_{d_i} \cup \Sigma^{R,s}_{i,o}, f'_{d_i}, q'_{0,d_i})$, $i = 1, \ldots, r$.
Saídas: $G_{\delta_i} = (Q'_{d_i}, \Sigma_i \cup \Sigma^s_{i,o}, f_{\delta_i}, q'_{0,d_i})$, $i = 1, \ldots, r$.

1: **para** $i = 1, \ldots, r$ **faça**
2: **para todo** (q_i, σ, q'_i) de G'_{d_i} tal que $\sigma \in \Sigma^R_{i,o} \cup \Sigma^{R,s}_{i,o}$ **faça**
3: Defina $f_{\delta_i}(q, R_i(\sigma)) = f'_{d_i}(q, \sigma)$
4: **para todo** (q_i, σ, q'_i) de G'_{d_i} tal que $\sigma \in \Sigma_{i,u}$ **faça**
5: Defina $f_{\delta_i}(q, \sigma) = f'_{d_i}(q, \sigma)$

Note que o conjunto de eventos observáveis de G_{δ_i} é $\Sigma^s_{i,o}$ e o conjunto de eventos não observáveis é Σ_i já que $\Sigma^s_{i,o}$ é o conjunto formado pelos eventos cuja observação é realizada pelo diagnosticador local D_i.

O exemplo a seguir mostra como os autômatos G_{δ_i} são obtidos.

Exemplo 10 *Considere novamente os componentes do sistema G, G_1, G_2 e G_3, apresentados no exemplo 8 e os autômatos Δ_1, Δ_2 e Δ_3 obtidos no exemplo 9. Para se obter os modelos dos componentes do sistema sujeitos a atrasos de comunicação de eventos, é necessário primeiro calcular os autômatos $G'_{d_1} = G_{d_1} \| \Delta_1$, $G'_{d_2} = G_{d_2} \| \Delta_2$ e $G'_{d_3} = G_{d_3} \| \Delta_3$ que são usados como entrada no algoritmo 5. Note que o algoritmo 5 apenas altera o rótulo dos eventos de G'_{d_1}, G'_{d_2} e G'_{d_3} para retirar as informações relacionadas ao canal em que os eventos são comunicados e ao atraso máximo de comunicação convertido para as referências locais. Uma vez que o algoritmo 5 seja aplicado aos autômatos G'_{d_1}, G'_{d_2} e G'_{d_3}, os autômatos G_{δ_1}, G_{δ_2} e G_{δ_3}, que modelam o comportamento dos componentes sujeito a atrasos de comunicação de eventos, são calculados e apresentados na figura 22.*

Teorema 1 *Os autômatos G_{δ_i}, $i = 1, \ldots, r$, obtidos aplicando-se os algoritmos 3, 4 e 5 modelam o comportamento observado dos componentes do sistema para diagnosticadores locais D_i considerando-se os conjuntos de eventos Σ_i como não observáveis e os conjuntos de eventos $\Sigma^s_{i,o}$ como observáveis.*

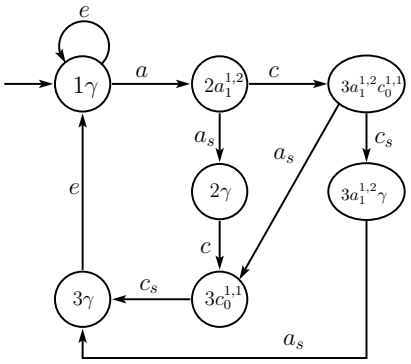
Prova. Os algoritmos 3, 4 e 5 implementam o método apresentado em Nunes *et al.* (2018) atualizando o atraso máximo para cada local e para cada componente do sistema. Dessa forma, os comportamentos observados dos autômatos G_{δ_i} correspondem ao comportamento dos componentes do sistema sob a possibilidade de atrasos de comunicação de eventos considerando-se Σ_i como conjunto de eventos não observáveis e $\Sigma^s_{i,o}$ como conjunto de eventos observáveis.

4.4 CODIAGNOSTICABILIDADE SÍNCRONA SUJEITA A ATRASOS DE COMUNICAÇÃO DE EVENTOS

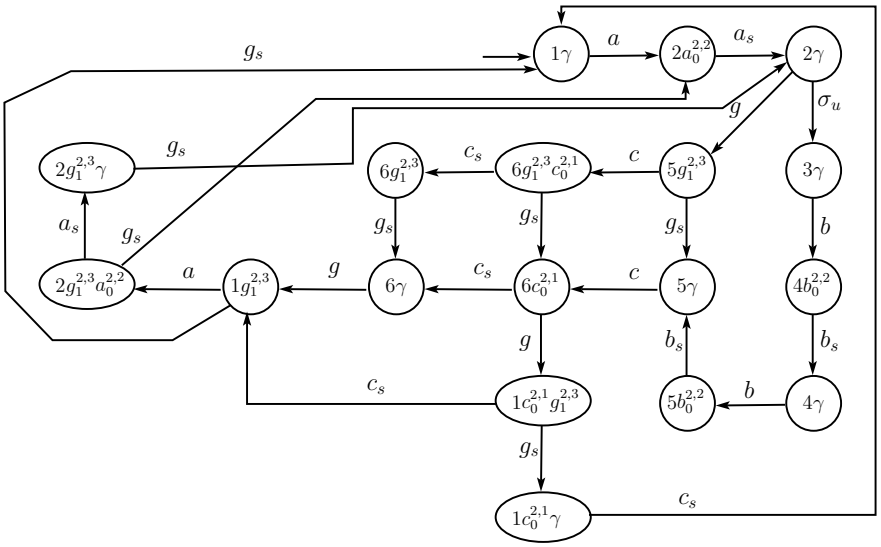
De acordo com o teorema 1, os autômatos G_{δ_i} modelam corretamente o efeito de possíveis atrasos de comunicação de eventos entre os locais de medição e os diagnosticadores locais. Assim sendo, a codiagnosticabilidade síncrona sujeita a atrasos de comunicação de eventos (CSSA) pode ser enunciada com base na linguagem gerada pelos modelos G_{δ_i} considerando-se os eventos $\Sigma^s_{i,o}$ como observáveis. Como os autômatos G_{δ_i} modelam os componentes do sistema sujeitos a atraso de comunicação de eventos, é possível obter o autômato $G_\delta = \|^r_{i=1} G_{\delta_i}$ que modela o comportamento completo da planta como segue.

$$G_\delta = \|^r_{i=1} G_{\delta_i} = (Q_\delta, \Sigma_\delta, f_\delta, q_{0,\delta}).$$

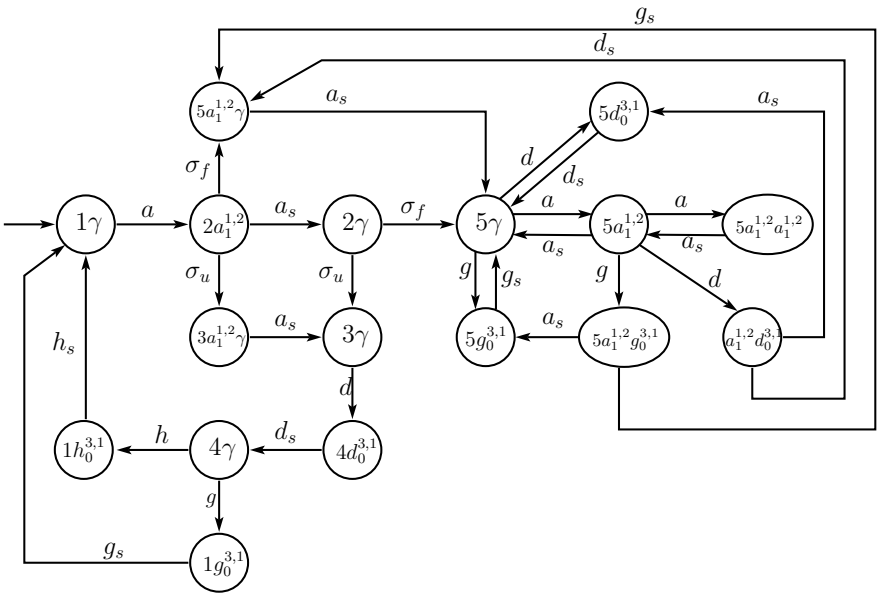
Note que o conjunto de eventos de G_δ é igual a $\Sigma_\delta = \Sigma_{\delta,u} \cup \Sigma^s_{\delta,o}$, em que $\Sigma_{\delta,u}$ é o conjunto de eventos não observáveis e $\Sigma^s_{\delta,o}$ é o conjunto de eventos observáveis de G_δ. Em resumo, o autômato G_δ representa a planta sob possibilidades de atraso



(a) G_{δ_1}



(b) G_{δ_2}



(c) G_{δ_3}

Figura 22 – Modelos dos componentes sujeitos a atrasos de comunicação de eventos G_{δ_1}, G_{δ_2} e G_{δ_3} do exemplo 10.

de comunicação de eventos para os diagnosticadores locais D_i e, portanto, pode ser usado para mapear os comportamentos de falha e livre de falha, necessários para definir a CSSA, como segue.

Definição 22 (CSSA) *Seja $G_{N_\delta} = \|G_{N,\delta_i}$, em que G_{N,δ_i} são os autômatos que modelam o comportamento livre de falha de G_{δ_i}, sendo G_{δ_i} obtido a partir do algoritmo 5. $L_{F_\delta} = L_\delta \setminus L_{N_\delta}$ representa a linguagem formada por todas as sequências geradas por G_δ tais que σ_f é um dos eventos dessas sequências, em que L_δ é a linguagem gerada por G_δ e L_{N_δ} é a linguagem livre de falha de G_δ. Sejam L_{N,δ_i} as linguagens livres de falha geradas por G_{N,δ_i}, para $i = 1, \ldots, r$. Suponha que existem r diagnosticadores locais tais que os conjuntos de eventos observáveis locais são dados por $\Sigma^s_{i,o}$. Assim, a linguagem gerada pelo sistema G, L, é dita ser codiagnosticável de forma síncrona sujeita a atrasos de comunicação de eventos em relação a Σ_f, L_{N,δ_i} e $P^s_{\delta_{i,o}} : \Sigma^\star_\delta \to \Sigma^{s^\star}_{i,o}$ se*

$$(\exists z \in \mathbb{N})(\forall s \in L_{F_\delta})(\forall st \in L_{F_\delta}, \|t\| \geq z) \Rightarrow$$
$$(\exists i \in \{1, 2, \ldots, r\})(P^s_{\delta_{i,o}}(st) \notin P^s_{\delta_{i,o}}(L_{N,\delta_i})).$$

De acordo com a definição 22, a linguagem L é codiagnosticável de forma síncrona sujeita a atrasos de comunicação de eventos se, para ao menos um diagnosticador local, toda a sequência $st \in L_{F_\delta}$ de falha de comprimento arbitrariamente longo após o evento de falha não tiver a mesma projeção do que sequências livres de falha em L_{N,δ_i}. Note que a definição 22 é adaptada de F. G. Cabral e Moreira (2020) para o caso síncrono descentralizado. Isso é feito substituindo-se a nova planta obtida por meio do algoritmo 5 e considerando-se os conjuntos $\Sigma^s_{i,o}$, para $i = 1, \ldots, r$, como conjuntos de eventos observáveis locais.

A CSSA pode ser verificada adaptando-se o método apresentado em Moreira *et al.* (2011), como feito em F. G. Cabral e Moreira (2020), considerando-se as modificações propostas neste trabalho. Resumidamente, o método de verificação apresentado neste trabalho baseia-se em mapear as projeções das sequências de falha com projeções das sequências livres de falha dos componentes nos quais os diagnosticadores locais são baseados. Isso é feito comparando-se a linguagem de falha do sistema, L_{F_δ}, com as linguagens livre de falha de cada componente tornando privados todos os eventos não observáveis de cada local. Para tanto, considere a seguinte função de renomeação de eventos apresentada a seguir.

$$\rho_i(\sigma) = \begin{cases} \sigma, & \text{se } \sigma \in \Sigma^s_{i,o} \\ \sigma_{\rho_i}, & \text{se } \sigma \in \Sigma_i \setminus \Sigma_f \end{cases} \tag{18}$$

O método de verificação da CSSA é apresentado no algoritmo 6. A seguir, apresentamos um teorema que garante que o algoritmo 6 de fato pode ser usado para verificar a CSSA.

Algoritmo 6 Método de verificação da CSSA

Entradas: G_δ, $G_{\delta_i} = (Q'_{d_i}, \Sigma_i \cup \Sigma^s_{i,o}, f_{\delta_i}, q'_{0,d_i})$, $i = 1, \ldots, r$.

Saídas: Verificador $V_\delta = (Q_{V_\delta}, \Sigma_{V_\delta}, f_{V_\delta}, q_{0,V_\delta})$ e status da CSSA.

1: Calcule o autômato G_{N_δ} que modela o comportamento livre de falha de G_δ (MOREIRA *et al.*, 2011).
2: **para todo** $i = 1, \ldots, r$ **faça**
3: Calcule $G_{N,\delta_i} = (Q_{N,\delta_i}, \Sigma_{N,\delta_i}, f_{N,\delta_i}, q'_{0,d_i})$, que modela o comportamento livre de falha de G_{δ_i}, conforme apresentado em F. G. Cabral e Moreira (2020), em que $\Sigma_{N,\delta_i} = (\Sigma_i \cup \Sigma^s_{i,o}) \setminus \Sigma_f$.
4: Calcule $\tilde{G}_{N,\delta_i} = (Q_{N,\delta_i}, \tilde{\Sigma}_{N,\delta_i}, \tilde{f}_{N,\delta_i}, q'_{0,d_i})$, tal que $\tilde{f}_{N,\delta_i}(q_i, \rho_i(\sigma)) = f_{N,\delta_i}(q_i, \sigma)$, $\forall q_i \in Q_{N,\delta_i}$ e $\sigma \in \Sigma_{N,\delta_i}$.
5: Calcule $G_{NF,\delta} = \|^r_{i=1} \tilde{G}_{N,\delta_i}$.
6: Calcule o autômato que modela o comportamento de falha $G_{F_\delta} = (Q_{F_\delta}, \Sigma_\delta, f_{F_\delta}, q_{0,F_\delta})$, tal que $L(G_{F_\delta}) = \overline{L_{F_\delta}}$ (MOREIRA *et al.*, 2011). Note que a última coordenada dos estados de G_{F_δ} que são alcançados por sequências que contenham σ_f é F.
7: Calcule o autômato verificador $V_\delta = (Q_{V_\delta}, \Sigma_{V_\delta}, f_{V_\delta}, q_{0,V_\delta}) = G_{NF,\delta} \| G_{F_\delta}$.
8: Verifique se existe pelo menos um caminho cíclico $cl = (q^u_V, \sigma_u, q^{u+1}_V, \sigma_{u+1}, \ldots, \sigma_v, q^u_V)$ em V_δ, $v \geq u > 0$, que satisfaz a seguinte condição:

$$
\begin{aligned}
&q^w_V \in Q_{V_\delta}, \forall w \in \{u, u+1, \ldots, v\}, \text{ tal que} \\
&\text{a última coordenada de } q^w_V \text{ é igual a } F \text{ e} \\
&\sigma_w \in \Sigma_\delta \text{ para algum } w \in \{u, u+1, \ldots, v\}.
\end{aligned}
$$

Se a resposta for *sim*, então L não é codiagnosticável de forma síncrona sujeita a atrasos de comunicação de eventos em relação a Σ_f, L_{N,δ_i} e $P^s_{\delta_{i,o}} : \Sigma^\star_\delta \to \Sigma^{s^\star}_{i,o}$. Caso contrário, L é codiagnosticável de forma síncrona sujeita a atrasos de comunicação de eventos.

Teorema 2 *A linguagem L é codiagnosticável de forma síncrona sujeita a atrasos de comunicação de eventos em relação a Σ_f, L_{N,δ_i} e $P^s_{\delta_{i,o}} : \Sigma^\star_\delta \to \Sigma^{s^\star}_{i,o}$ se e somente se não existe um caminho cíclico $cl = (q^u_V, \sigma_u, q^{u+1}_V, \sigma_{u+1}, \ldots, \sigma_v, q^u_V)$ em V_δ, $v \geq u > 0$, que satisfaz a seguinte condição:*

$$
\begin{aligned}
&q^w_V \in Q_{V_\delta}, \forall w \in \{u, u+1, \ldots, v\}, \textit{ tal que} \\
&\textit{a última coordenada de } q^w_V \textit{ é igual a } F \textit{ e} \\
&\sigma_w \in \Sigma_\delta \textit{ para algum } w \in \{u, u+1, \ldots, v\}.
\end{aligned}
$$

prova: Considere um sistema $G = \|^r_{i=1} G_i$, formado por r componentes, cuja linguagem gerada é L, sendo que a linguagem gerada de cada componente G_i é L_i. Considere

também que o conjunto de eventos observáveis e não observáveis de G_i sejam denotados por $\Sigma_{i,o}$ e $\Sigma_{i,u}$. Ao tornar particulares os eventos não observáveis de G_i utilizando-se uma função de renomeação, obtém-se G_i^R, cuja linguagem gerada é L_i^R e cujo conjunto de eventos não observáveis se torna $\Sigma_{i,u}^R$, tal que $\Sigma_{i,u}^R \cap \Sigma_{j,u}^R = \emptyset$, para $i \neq j$ e $i, j \in \{1, \ldots, r\}$. Seja $G^R = \|_{i=1}^r G_i^R$, que possui Σ_o e Σ_u^R como conjuntos de eventos observáveis e não observáveis, respectivamente, sendo $\Sigma^R = \Sigma_o \cup \Sigma_u^R$. A linguagem gerada por G^R é denotada por L_R. Em F. G. Cabral e Moreira (2020) é mostrado que $P_o^R(L_R) = \bigcap_{i=1}^r P_{i,o}^{o^{-1}}\left(P_{i,o}^i(L_i)\right)$. Ou seja, a projeção nos eventos observáveis da linguagem L_R corresponde à linguagem gerada pela composição paralela dos observadores dos componentes G_i locais.

Assim sendo, como os eventos não observáveis de G_{N,δ_i} são renomeados para obter-se $\tilde{G}_{N,\delta_i}$ e, consequentemente, $G_{NF,\delta}$, nos passos 4 e 5 do algoritmo 6, não existem eventos não observáveis em comum entre $G_{NF,\delta}$ e G_{F_δ}, que gera a linguagem $\overline{L_{F_\delta}}$. Portanto, a comparação entre observações de sequências em $G_{NF,\delta}$ e observações de sequências em G_{F_δ} pode ser feita realizando-se a composição paralela entre esses autômatos, o que é realizado no passo 7 do algoritmo 6 para se obter o autômato verificador V_δ.

Como é mostrado em Moreira *et al.* (2011), os estados de V_δ que possuem rótulo F são alcançados por sequências livres de falha e sequências de falha que possuem a mesma observação. Portanto, se um caminho cíclico $cl = (q_V^u, \sigma_u, q_V^{u+1}, \sigma_{u+1}, \ldots, \sigma_v, q_V^u)$ em V_δ, $v \geq u > 0$, é encontrado em V_δ que satisfaz a condição:

$$
\begin{aligned}
&q_V^w \in Q_{V_\delta}, \forall w \in \{u, u+1, \ldots, v\}, \text{ tal que} \\
&\text{a última coordenada de } q_V^w \text{ é igual a } F \text{ e} \\
&\sigma_w \in \Sigma_\delta \text{ para algum } w \in \{u, u+1, \ldots, v\}.
\end{aligned}
$$

Então, existe ao menos uma sequência de falha de comprimento arbitrariamente longo após a falha cuja projeção se confunde com ao menos uma sequência livre de falha. Como V_δ mapeia todas as possíveis observações locais, isso significa que a existência do ciclo cl mostra que nenhum diagnosticador local pode detectar a ocorrência da falha para ao menos uma sequência de falha de comprimento arbitrariamente longo após a falha, o que torna a linguagem L não codiagnosticável de forma síncrona sujeita a atrasos de comunicação de eventos em relação a Σ_f, L_{N,δ_i} e $P_{\delta_{i,o}}^s : \Sigma_\delta^\star \to \Sigma_{i,o}^{s^\star}$.

Assim, o método de verificação da CSSA proposto neste trabalho corresponde a aplicar o método proposto em Moreira *et al.* (2011) a um sistema cujo autômato de falha gera $\overline{L_{F_\delta}}$ e cujo autômato que modela todas as sequências livres de falha gera $L(G_{NF,\delta})$. Portanto, a mesma condição necessária e suficiente para determinar a CSSA seria obtida ao utilizar o método apresentado em Moreira *et al.* (2011), o que conclui a prova. ■

Observação 2 *É importante comentar que o verificador V_δ obtido utilizando-se o al-*

goritmo 6 tem crescimento exponencial com o número de componentes do sistema. Entretanto, a análise da CSSA pode ser feita offline e, uma vez que o sistema seja codiagnosticável de forma síncrona sujeito a atrasos de comunicação de eventos, os diagnosticadores locais, que são baseados no comportamento livre de falha dos módulos do sistema, podem ser implementados.

5 CONCLUSÃO

Neste trabalho, o diagnóstico síncrono descentralizado robusto a atrasos de comunicação de eventos é proposto. O método consiste em modificar os modelos dos componentes do sistema de tal forma a incorporar o efeito do atraso de comunicação de eventos aos diagnosticadores locais. Por causa do atraso de comunicação de eventos entre locais de medição e diagnosticadores locais, é necessário definir a codiagnosticabilidade síncrona sujeita a atrasos de comunicação de eventos. Além disso, um método para verificar essa propriedade também é proposto neste trabalho.

Alguns resultados deste trabalho foram apresentados no XXIII Congresso Brasileiro de Automática (CBA 2020) no artigo Araújo *et al.* (2020). Em Araújo *et al.* (2020) o método para o cálculo dos modelos dos componentes locais sujeitos a atrasos de comunicação de eventos é proposto, entretanto, a noção de codiagnosticabilidade síncrona sujeita a atrasos de comunicação de eventos e o método de verificação não são apresentados. Todos os resultados propostos nesta dissertação serão submetidos para publicação em periódico internacional.

Como sugestão de trabalhos futuros visa-se o estudo do atraso de comunicação no contexto explorado em Veras *et al.* (2018). Em Veras *et al.* (2018), a arquitetura de diagnóstico síncrono distribuída é proposta, em que diagnosticadores locais podem trocar informação, como observação de eventos e estimativas de estado, entre si. Entretanto, em Veras *et al.* (2018) é suposto que toda a comunicação (entre planta e diagnosticadores locais e a comunicação entre diagnosticadores locais) é ideal, ou seja, não há perdas ou atrasos de comunicação. Uma vez que esse cenário nem sempre pode ser garantido, é importante desenvolver diagnosticadores locais robustos a esses problemas de comunicação.

REFERÊNCIAS

ARAÚJO, Guilherme T; CABRAL, Felipe G; MOREIRA, Marcos V. Diagnosticabilidade Síncrona Distribuída de Sistemas a Eventos Discretos Sujeita a Atrasos de Comunicação de Eventos. *In*: CONGRESSO Brasileiro de Automática - CBA. [*S.l.*: *s.n.*], 2020.

CABRAL, F. G.; MOREIRA, M. V. Synchronous Diagnosis of Discrete-Event Systems. **IEEE Transactions on Automation Science and Engineering**, v. 17, n. 2, p. 921–932, 2020.

CABRAL, Felipe G; MOREIRA, Marcos V; DIENE, Oumar. Online fault diagnosis of modular discrete-event systems. *In*: 54TH Conference on Decision and Control (CDC). Osaka, Japan: [*s.n.*], 2015. P. 4450–4455.

CASSANDRAS, C.G.; LAFORTUNE, S. **Introduction to Discrete Event System**. Secaucus, NJ: Springer-Verlag New York, Inc., 2008.

DEBOUK, R.; LAFORTUNE, S.; TENEKETZIS, D. Coordinated decentralized protocols for failure diagnosis of discrete event systems. **Discrete Event Dynamic Systems: Theory and Applications**, v. 10, n. 1, p. 33–86, 2000.

DEBOUK, R.; MALIK, R.; BRANDIN, B. A modular architecture for diagnosis of discrete event systems. *In*: 41ST IEEE Conference on Decision and Control. Las Vegas, Nevada USA: [*s.n.*], 2002. P. 417–422.

GILCHRIST, A. **Industry 4.0: The Industrial Internet of Things**. 1st ed. [*S.l.*]: Berkeley, CA, USA: Apress, 2016, set. 2016.

KEROGLOU, C.; HADJICOSTIS, C. N. Distributed diagnosis using predetermined synchronization strategies. *In*: PROCEEDINGS of 53rd IEEE Conference on Decision and Control (CDC). Los Angeles, CA, USA: [*s.n.*], 2014. P. 5955–5960.

KEROGLOU, C.; HADJICOSTIS, C. N. Distributed Fault Diagnosis in Discrete Event Systems via Set Intersection Refinements. **IEEE Transactions on Automation Science and Engineering**, 63, 10, 3601–3607, 2018.

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

NUNES, Carlos EV; MOREIRA, Marcos V; ALVES, Marcos VS; CARVALHO, Lilian K; BASILIO, João Carlos. Codiagnosability of networked discrete event systems subject to communication delays and intermittent loss of observation. **Discrete Event Dynamic Systems**, v. 28, n. 2, p. 215–246, 2018.

QIU, W.; KUMAR, R. Decentralized failure diagnosis of discrete event systems. **IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans**, v. 36, n. 2, p. 384–395, 2006.

QIU, Wenbin; KUMAR, Ratnesh. Distributed diagnosis under bounded-delay communication of immediately forwarded local observations. **IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans**, v. 38, n. 3, p. 628–643, 2008.

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

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

TRIPAKIS, Stavros. Decentralized control of discrete-event systems with bounded or unbounded delay communication. **IEEE Transactions on Automatic Control**, v. 49, n. 9, p. 1489–1501, 2004.

VERAS, M Z M; CABRAL, F G; MOREIRA, M V. Distributed Synchronous Diagnosability of Discrete-Event Systems. **14th IFAC Workshop on Discrete Event Systems**, Elsevier Ltd, v. 51, n. 7, p. 88–93, 2018.

WANG, Y.; YOO, T.-S; LAFORTUNE, S. Diagnosis of discrete event systems using decentralized architectures. **Discrete Event Dynamic Systems: Theory And Applications**, v. 17, p. 233–263, 2007.