UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MARCELO ROSA

EXPLORANDO O USO DE APROXIMAÇÕES NA SÍNTESE E NA IMPLEMENTAÇÃO DE CONTROLADORES PARA SISTEMAS A EVENTOS DISCRETOS

DISSERTAÇÃO

PATO BRANCO

2019

MARCELO ROSA

EXPLORANDO O USO DE APROXIMAÇÕES NA SÍNTESE E NA IMPLEMENTAÇÃO DE CONTROLADORES PARA SISTEMAS A EVENTOS DISCRETOS

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de "Mestre em Engenharia Elétrica" – Área de Concentração: Sistemas e processamento de energia.

Orientador: Prof. Dr. Marcelo Teixeira

Co-orientador: Prof. Dr. Robi Malik

PATO BRANCO

2019

R788e Rosa, Marcelo.
Explorando o uso de aproximações na síntese e na implementação de controladores para sistemas a eventos discretos / Marcelo Rosa. -- 2019.
79 f. : il. ; 30 cm

Orientador: Prof. Dr. Marcelo Teixeira
Coorientador: Prof. Dr. Robi Malik
Dissertação (Mestrado) - Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica. Pato Branco, PR, 2019.
Bibliografia: f. 73 - 74.

1. Sistema de tempo discreto. 2. Sistemas de controle supervisório. 3. Controladores programáveis. I. Teixeira, Marcelo, orient. II. Malik, Robi, coorient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica. IV. Título.

CDD 22. ed. 621.3

Ficha Catalográfica elaborada por
Suélem Belmudes Cardoso CRB9/1630
Biblioteca da UTFPR Campus Pato Branco

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Pato Branco
Diretoria de Pesquisa e Pós-Graduação
Programa de Pós-Graduação em Engenharia Elétrica

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

TERMO DE APROVAÇÃO

Título da Dissertação n.° 067

“Explorando o uso de aproximações na síntese e na implementação de controladores para sistemas a eventos discretos”

por

Marcelo Rosa

Dissertação apresentada às oito horas e vinte minutos, do dia quatorze de fevereiro de dois mil e dezenove, como requisito parcial para obtenção do título de MESTRE EM ENGENHARIA ELÉTRICA, do Programa de Pós-Graduação em Engenharia Elétrica – Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho APROVADO.

Banca examinadora:

Prof. Dr. Marcelo Teixeira (orientador)
UTFPR/PB

Profª. Drª. Patrícia Nascimento Pena
UFMG/BH

Prof. Dr. Marco Antônio de Castro Barbosa
UTFPR/PB

Prof. Dr. Cesar Rafael Claure Torrico
UTFPR/PB

Prof. Dr. Gustavo Weber Denardin
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica - PPGEE/UTFPR

A versão devidamente assinada desse termo, encontra-se em arquivo no PPGEE - UTFPR – Câmpus Pato Branco.

RESUMO

ROSA, Marcelo. EXPLORANDO O USO DE APROXIMAÇÕES NA SÍNTESE E NA IMPLEMENTAÇÃO DE CONTROLADORES PARA SISTEMAS A EVENTOS DISCRETOS. 79 f. Dissertação – Programa de Pós-graduação em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Pato Branco, 2019.

A *Teoria de Controle Supervisório* (TCS) define uma operação automática para a síntese de controladores para *Sistemas a Eventos Discretos* (SEDs). Apesar de suas inúmeras extensões e vantagens, a TCS ainda esbarra em alguns aspectos de complexidade que limitam a sua aplicação sobre problemas industriais reais. Em particular, essas limitações se permeiam entre as etapas de modelagem, síntese e implementação de controladores. Recentemente, foi proposta na literatura uma abordagem baseada no *refinamento de eventos* como forma de simplificar tarefas de modelagem, ampliando assim a aplicabilidade da TCS para uma gama maior de problemas. No entanto, essa abordagem não leva diretamente a ganhos computacionais no procedimento de síntese. Na verdade, pode ser mostrado que o custo computacional para calcular o controlador é o mesmo em relação à abordagem sem refinamentos. Também, pode ser mostrado que o custo em termos de *hardware*/memória para implementar a solução de controle com ou sem refinamentos é o mesmo. No sentido de estender vantagens do refinamento de eventos para a etapa de síntese, a literatura explora o uso de *aproximações*. Uma aproximação pode ser associada à ideia de abstração de modelos. Na prática, sua construção se materializa pela remoção de parte, ou de todo, o mecanismo que distingue eventos refinados no modelo do sistema, o deixando assim mais simples, no entanto menos preciso. Esse grau de imprecisão pode implicar na obtenção de controladores subótimos, casos em que a aproximação precisa ser reconstruída até que leve ao controlador ótimo. Essa reconstrução é um processo manual, que depende sobretudo da percepção do engenheiro e que, não raramente, é executado conforme uma política de tentativa e erro. Nesta dissertação, a construção de uma aproximação é automatizada e seus benefícios são estendidos para as etapas de síntese e implementação de controladores. Inicialmente, propõe-se um método para a construção de uma classe especial de aproximações que sempre leva ao controlador ótimo, sem a necessidade de qualquer verificação pós-síntese. Em seguida, apresenta-se uma arquitetura que estende para a etapa de implementação os benefícios do uso de refinamentos e de aproximações. Essa abordagem permite que o sistema de controle possa ser implementado de maneira descentralizada, por um sistema comunicante, que cumpre o mesmo papel de uma implementação centralizada, mas com ganhos no consumo de memória para representar a ação de controle em hardware. Por fim, a última contribuição proposta nesta dissertação estende o método de construção de aproximações e síntese, para o contexto do controle modular de SEDs. Essa abordagem potencializa a aplicação da TCS sobre problemas de grande porte, uma vez que, por definição, o controle modular independe do tamanho do sistema e pode ser conduzido incrementalmente.

Palavras-chave: Sistemas a Eventos Discretos, Controle Supervisório, Aproximações

ABSTRACT

ROSA, Marcelo. EXPLOITING THE USE OF APPROXIMATIONS IN THE SYNTHESIS AND IMPLEMENTATION OF CONTROLLERS FOR DISCRETE EVENT SYSTEMS . 79 f. Dissertação – Programa de Pós-graduação em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Pato Branco, 2019.

The *Supervisory Control Theory* (SCT) defines an automatic operation for the synthesis of controllers for *Discrete-Event Systems* (DESs). Despite its several extensions and improvements, the SCT still faces significant complexity issues that limit its practical application. In particular, these limitations permeate the stages of modeling, synthesis and implementation of controllers. Recently, an approach based on *event-refinements* has been proposed in the literature as a way of simplifying modeling tasks, thus extending the applicability of the SCT to a greater range of problems. However, this approach does not lead directly to computational gains in the synthesis procedure. Actually, it can be shown that the computational cost to calculate the controller is the same, with respect to the approach without refinements. Also, it can be shown that the cost in terms of hardware/memory to implement the control solution with or without refinements is the same. In order to extend the advantages of event refinements to the synthesis, the literature exploits the use of *approximations*. An approximation can be associated to the idea of model abstraction. In practice, its construction is materialized by the removal of part or all the mechanism that distinguishes refined events in the system model, thus making it simpler, but less precise. This level of imprecision may imply obtaining sub-optimal controllers, case in which the approach must be reconstructed until it leads to the optimal controller. This reconstruction is a manual process, which depends on the perception of the enginner and that, not rarely, is executed according to a trial-and-error strategy. In this dissertation, the construction of an approximation is automated and its benefits are extended to the stage of synthesis and implementation of controllers. Initially, a method for the construction of a special class of approximations that always leads to the optimal controller, without the need for any post-synthesis verification is proposed. Next, an architecture is proposed as a way to extend the benefits of refinements and approximations to the implementation phase. This approach allows the control system to be implemented in a decentralized manner, through a communicating system, which accomplishes the same role as a centralized implementation, but with gains in memory consumption to represent the control action in hardware. Finally, the last contribution of this dissertation extends the use of approximations to a modular case. This approach increases the applicability of the SCT to cover larger problems, since, by definition, the modular control is independent of the size of the system and can be derived incrementally.

Keywords: Discret-Event Systems, Control Supervisory, Approximations

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

SCT	*Supervisory Control Theory* - Teoria de Controle Supervisório
DESs	*Discrete Event Systems* - Sistemas a Eventos Discretos
SED	Sistema a Eventos Discretos
TCS	Teoria de Controle Supervisório
AF	Autômato Finito
TCS-D	Teoria de Controle Supervisório com Distinguidores
CML	Controle Modular Local
CMLD-A	Controle Modular Local com Distinguidores e Aproximações

LISTA DE SÍMBOLOS

Σ	Conjunto de eventos (alfabeto)
Σ^*	Conjunto de todas as cadeias de elementos de Σ
ε	Cadeia vazia
L	Linguagem
$\overline{L}$	Prefixo-fechamento de uma linguagem L
Q	Conjunto de estados de um AF
q°	Estado inicial de um AF
Q^ω	Conjunto de estados marcados de um AF
$\xrightarrow{\sigma}$	Transição entre dois estados de um AF com evento σ
$\Gamma(\mathsf{q})$	Conjunto de eventos possíveis em um estado q
$\|$	Composição síncrona entre AFs
$\mathcal{G}$	Conjunto de AFs
Σ_c	Conjunto de eventos controláveis
Σ_u	Conjunto de eventos não-controláveis
$\sup\mathcal{C}$	Máxima sub-linguagem controlável
Δ^σ	Conjunto de refinamentos do evento σ
Δ	Conjunto de eventos refinados
Δ_u	Conjunto de eventos refinados não-controláveis
Δ_c	Conjunto de eventos refinados controláveis
Π	Mapa mascarador
Π^{-1}	Mapa mascarador inverso
Δ^*	Conjunto de cadeias refinadas
D	Mapa distinguidor
L_d	Linguagem do distinguidor
G_d	Planta distinguida
G_a	Planta refinada
$\mathcal{S}^\mathrm{u}$	Conjunto de estados certamente não-controláveis
$\mathcal{S}^\mathrm{c}$	Conjunto de estados certamente controláveis
$\mathcal{S}^\mathrm{e}$	Conjunto de estados eventualmente controláveis
$\Delta_{\text{alvo}}(G_\mathrm{a})$	Conjunto de eventos alvos em G_a
$\Delta_{\text{indalvo}}(G_\mathrm{a})$	Conjunto de eventos indiretamente alvo em G_a
$\mathcal{G}_\mathrm{a}$	Conjunto de plantas refinadas
$\mathcal{E}_d$	Conjunto de especificações refinadas
$\mathcal{H}$	Conjunto de modelos distinguidores

SUMÁRIO

1 INTRODUÇÃO

Os sistemas industriais, em sua grande maioria, são formados por inúmeros componentes (equipamentos, sensores e atuadores) que interagem entre si de maneira coordenada, automatizando e otimizando o processo de produção. Esses componentes são, em geral, dirigidos por eventos que definem a ordem com que cada um se integra ao processo. Uma esteira, por exemplo, pode ter seu comportamento modelado por eventos de *liga* e *desliga*, cuja ocorrência pode desencadear outros eventos, oriundos de diferentes componentes, que passam então a atuar em função da primeira, e assim por diante. Dentro dessa ótica, esse tipo de sistema pode ser visto e classificado como um *Sistema a Eventos Discretos* (SED) (CASSANDRAS; LAFORTUNE, 2008).

Os SEDs são caracterizados por possuírem um conjunto enumerável de estados e um mecanismo de transição guiado a eventos assíncronos no tempo. Dessa forma, uma escolha natural para modelar um SED é por meio de diagramas de transições (máquinas de estados), e um formalismo que possibilita expressar tais diagramas são os autômatos finitos. Assim, os diversos componentes de um SED podem ser modelados por um conjunto de autômatos, de modo que a integração entre tais modelos resulta no comportamento do sistema a malha aberta, denominado de *planta*. Consequentemente, a coordenação/controle de tais componentes também é implementada por autômatos, denominados de *especificações*. A composição entre planta e especificações resulta em um modelo que pode ser visto como uma versão do sistema de controle, uma vez que as especificações implementam ações de controle sobre a planta por meio da desabilitação de eventos. No entanto, como alguns eventos não são passíveis de controle e essa versão do controlador não leva em conta isso, então situações que deveriam ser proibidas na planta pelo controlador são possíveis de ocorrem na prática.

Nesse sentido, uma alternativa formal que incorpora a ideia de distinção de eventos quanto à sua controlabilidade e também descreve a síntese de controladores ótimos para SEDs é a *Teoria de Controle Supervisório* (TCS) (RAMADGE; WONHAM, 1989). Na TCS, planta e especificações de controle do sistema são modeladas por um conjunto

de autômatos, que são dados como entrada a uma operação matemática que calcula o comportamento controlável minimamente restritivo da planta, implementado por um *supervisor*. O supervisor atua de modo a observar os eventos possíveis na planta e define dentre eles quais de fato devem ser habilitados.

Apesar de ser um mecanismo formal que proporciona robustez e segurança à prática de controle, a TCS ainda enfrenta limitações significativas quando aplicada a problemas reais de controle industrial. Uma dessas limitações, que é de interesse particular desse trabalho, está associada ao nível de informação observável no modelo de um SED, que define a forma como o problema pode ser representado e como a solução pode ser obtida (CURY et al., 2015; TEIXEIRA et al., 2014; CUNHA; CURY, 2007). Um obstáculo, em particular, surge quando tal nível de informação não é suficiente para expressar alguns requisitos de controle. Isso leva à necessidade de memorizar um conjunto de cadeias de eventos até que uma ação de controle possa ser tomada. Por exemplo, em um sistema de manufatura com retrabalho de peças (CURY et al., 2015), o controle da quantidade de retrabalhos requer uma estrutura que armazene o trajeto (cadeia de eventos) que cada peça percorre, a cada ciclo de trabalho, até que a decisão sobre um novo retrabalho possa ser tomada. A complexidade de se projetar tal modelo está ligada diretamente à quantidade de retrabalhos e ao número de peças processadas em paralelo. Características como memorização e dependências entre eventos, paralelismo, concorrência, etc., são bastante comuns em SEDs, e estão naturalmente suscetíveis a esse obstáculo que restringe a aplicação da TCS no controle de SEDs.

Uma opção para aumentar o nível de informações providas por um SED seria integrar mais sensores à planta do sistema. No entanto, essa abordagem pode ainda assim ser insuficiente para flexibilizar o problema de modelagem em alguns casos. No exemplo anterior, a simples inserção de mais sensores permitiria identificar peças (presença ou característica física), mas não o seu contexto. Ou seja, não permitiria, por exemplo, identificar a quantidade de vezes que cada peça passou pelo sistema, sendo ineficaz para o fim que se vislumbra sob o ponto de vista de controle.

Nesse sentido, a literatura (BOUZON et al., 2008; CURY et al., 2015) propõe o conceito de (sensores) *distinguidores*. Informalmente, um distinguidor pode ser pensando com um sensor inteligente que adiciona/embarca contexto em cima dos próprios eventos providos pela planta. Para isso, cada evento do modelo de um SED é associado a um conjunto de eventos mapeados de modo a identificar instâncias do evento original no sistema. Dispor desse conjunto de refinamentos tende a simplificar tarefas de modelagem

de especificações, pois um refinamento é mais rico em contexto do que o evento original e isso pode facilitar expressar uma regra de controle. A literatura tem mostrado (CURY et al., 2015) que o uso de distinguidores podem inclusive viabilizar a modelagem de problemas que seriam intratáveis usando apenas os eventos originais do sistema.

Em contrapartida, um modelo adicional, assim chamado de *distinguidor*, precisa ser associado à planta a fim de atribuir uma semântica particular a cada evento refinado. Assim, embora um distinguidor simplifique tarefas de modelagem, a complexidade computacional da síntese do controlador permanece essencialmente a mesma, com ou sem refinamentos. Ademais, espera-se que a complexidade de implementação também seja a mesma, pois o resultado de síntese é o mesmo, com ou sem refinamentos, com a mera distinção de serem definidos sobre dois domínios diferentes. Esse trabalho ataca essas duas limitações da TCS com distinguidores, complexidade de síntese (ROSA et al., 2018b, 2018a) e de implementação em hardware (ROSA et al., 2017).

No sentido de tornar essa abordagem computacionalmente mais atrativa, a literatura (BOUZON et al., 2009; TEIXEIRA et al., 2013; CURY et al., 2015) tem sugerido que um distinguidor possa ser *aproximado* de modo tal que a síntese pode ser conduzida com distinção parcial, ou nula, de eventos. Condições têm sido propostas para mostrar que, em alguns casos, essa abordagem ainda leva a um controlador ótimo. Entretanto, em outros casos, pode acontecer que uma aproximação leve a um controlador subótimo, i.e, mais restritivo, de modo que para obter o controlador ótimo é necessário especificar quais eventos devem ser distinguidos em tempo de síntese.

A questão que se coloca então é: como antecipar quais eventos de fato devem ser distinguidos na síntese de um controlador? Essa questão identifica um dos propósitos dessa dissertação. Até agora, o único resultado que tenta responder a esta questão é fornecido por (AGUIAR et al., 2013), onde duas meta-heurísticas são propostas para identificar as *distinções essenciais* baseada na evolução randômica do conjunto de soluções/configurações iniciais. No entanto, estima-se que essa abordagem de tentativa e erro é computacionalmente dispendiosa, uma vez que o grau de distinção de eventos na síntese é incrementalmente alterado e testado de acordo com uma condição estabelecida (CURY et al., 2015), para que se possa decidir quais eventos realmente devem ser distinguidos ou não. Além disso, isso não garante minimidade, ou seja, que somente as distinções essenciais serão adicionada a síntese.

Portanto, esta dissertação é assentada sobre essas lacunas e dela são derivados três resultados principais, conforme descritos a seguir:

(i) Como primeiro resultado, propõe-se um método para identificar os eventos que necessitam de distinção na síntese, de modo que tal informação é utilizada para construir *auto-aproximações* que, quando usadas na síntese, levam diretamente ao controlador menos restritivo e controlável, sem a necessidade da verificação pós-síntese como sugerida em (CURY et al., 2015; BOUZON et al., 2009; TEIXEIRA et al., 2013; AGUIAR et al., 2013). A abordagem é estruturada sobre os conceitos de *evento-alvo* e *estado-alvo*, os quais fundamentam a identificação de eventos particularmente importantes para a síntese e que, portanto, requerem um componente distinguidor associado (ROSA et al., 2018b, 2018a);

(ii) Derivada da própria ideia de auto-aproximação, combinada com a modularização de modelos distinguidores, nasce uma segunda contribuição desta dissertação, que versa sobre aspectos de implementação de controladores sintetizados a partir de auto-aproximações. Como na proposta de auto-aproximações o modelo distinguidor acaba sendo afastado do controlador em si, então seria intuitivo pensar em implementá-los também separados, como dois componentes comunicantes que cumprem o mesmo papel de uma versão do controlador que os combina via composição síncrona. Dessa dedução, propõe-se uma arquitetura de implementação descentralizada de controladores usando distinguidores e auto-aproximações que aproveita os ganhos obtidos nas etapas anteriores à implementação (ROSA et al., 2017);

(iii) Finalmente, como terceiro e último resultado, propõe-se combinar os resultados desse trabalho a um *framework* de síntese modular, de modo a favorecer a escalabilidade do sistema, de tal forma que adição de novos componentes/sub-sistemas não resulte na refatoração completa do sistema de controle.

2 CONCEITOS PRELIMINARES

Neste capítulo apresenta-se a classe de sistemas de interesse desta dissertação, os SEDs, bem como os conceitos básicos da teoria de linguagens, autômatos e algumas propriedades que fundamentam o desenvolvimento deste trabalho. Então, a partir desses fundamentos, apresentam-se de forma sucinta os principais conceitos da TCS e suas extensões envolvendo refinamentos de eventos e aproximações. Além disso, ao longo deste capítulo são fornecidos exemplos para ilustrar os conceitos apresentados.

2.1 SISTEMAS A EVENTOS DISCRETOS

A modelagem de sistema é, em geral, estruturada sobre dois fundamentos básicos: (i) conceito de *estado*, que identifica o status do sistema em determinada circunstância; e (ii) um mecanismo de *transição de estados*, que descreve a evolução do sistema. Assim, classificar um conjunto de sistemas que possuem alguma característica em comum, passa, sobretudo, por caracterizá-los de acordo com esses dois fundamentos (TEIXEIRA, 2013).

Por exemplo, alguns sistemas têm seus estados mapeados continuamente e suas transições são guiadas pelo tempo. Em outros casos, o espaço de estados pode não ser continuo, e sua dinâmica pode não depender do tempo, mas de eventos assíncronos. Sistemas que compartilham dessa dinâmica, i.e., que são caracterizados por um conjunto enumerável de estados, e cujos mecanismos de transição são guiados por eventos assíncronos, são denominados *Sistemas a Eventos Discretos* (SEDs) (CASSANDRAS; LAFORTUNE, 2008). Exemplos de SEDs incluem sistemas de manufatura, robótica, jogos, etc.

A modelagem de SEDs se dá, normalmente, por meio de diagramas de transição (máquinas de estados), pois permitem representar a evolução do sistema dentro do espaço de estados, de forma intuitiva, capturando apenas os comportamentos relevantes a serem modelados. Alguns formalismos que existem para se expressar SEDs são apresentados a seguir.

2.2 MODELAGEM DE SISTEMAS A EVENTOS DISCRETOS

O processo de se realizar experimentos sobre a estrutura real de um sistema é a maneira mais eficaz de analisá-lo, tendo como base o grau de precisão dos resultados obtidos (OGATA, 2010). Contudo, esse processo nem sempre é possível, seja devido a complexidade ou mesmo a indisponibilidade da estrutura real do sistema. Desse modo, uma alternativa é representá-lo por meio de um modelo, o que permite abstrair detalhes de seu comportamento, de modo a facilitar a sua compreensão e manipulação.

Dentre os vários formalismos existentes para se representar SEDs destacam-se, por exemplo, as *Redes de Petri*, a *Teoria de Filas* e a *Teoria de Autômatos e Linguagens*, etc. (CASSANDRAS; LAFORTUNE, 2008). No entanto, nenhum desses formalismos desponta como unânime na escolha para um projeto de SED, pois cada um possui características e finalidades distintas.

Este trabalho adota como formalismo de modelagem a *Teoria de Autômatos e Linguagens*. A escolha por essa abordagem se dá pelo fato de que as operações sobre autômatos e linguagens fundamentam, por construção, a síntese de controladores ótimos para SEDs. Nesse sentido, apresentam-se a seguir os principais conceitos de autômatos e linguagens.

2.2.1 TEORIA DE AUTÔMATOS E LINGUAGENS

Visto que o comportamento de um SED evolui conforme a ocorrência assíncrona de eventos, é intuitivo pensar que o seu modelo seja caracterizado por simplesmente mapear as sequências de eventos possíveis. Ao conjunto de todos os eventos que integram um SED, dá-se o nome de *alfabeto*, que é denotado por Σ. Já as sequências de eventos são denominadas *cadeias*, e o conjunto Σ^* contempla todas as cadeias finitas possíveis de serem construídas com eventos em Σ, incluindo a *cadeia vazia* ε.

A partir desses conceitos, pode-se introduzir a ideia de linguagens. Uma *linguagem* L sobre um alfabeto Σ, é um subconjunto de cadeias em Σ^*, i.e., $L \subseteq \Sigma^*$. O *prefixo-fechamento* ($\overline{L}$) de um linguagem L é definido por $\overline{L} = \{\, s \in \Sigma^* \mid st \in L \text{ para algum } t \in \Sigma^* \,\}$. L é dita ser *prefixo-fechada* se $L = \overline{L}$. Assim, uma linguagem L é prefixo-fechada se qualquer prefixo de qualquer cadeia de L, também é um elemento de L.

Uma linguagem é dita ser regular quando se pode expressá-la utilizando um *Autômato Finito* (AF) (CASSANDRAS; LAFORTUNE, 2008). Um AF é um diagrama

de transição de estados, que permite modelar diversos problemas, por meio de um método prático e intuitivo (CASSANDRAS; LAFORTUNE, 2008). Formalmente, um AF é representado por uma 5-tupla $G = \langle \Sigma, Q, Q^{\omega}, \mathsf{q}^{\circ}, \rightarrow \rangle$, em que:

- Σ é o alfabeto de eventos;
- Q é o conjunto finito de estados;
- $\mathsf{q}^{\circ} \in Q$ é o estado inicial;
- $Q^{\omega} \subseteq Q$ é o conjunto de estados marcados;
- $\rightarrow \subseteq Q \times \Sigma \times Q$ é a relação transição.

Nesta dissertação, considera-se que todos os estados são marcados, ou seja, $Q^{\omega} = Q$. Além disso, para quaisquer dois estados $\mathsf{q}, \mathsf{q}' \in Q$, a transição do estado q para o estado q' com o evento $\sigma \in \Sigma$ é representada por $\mathsf{q} \xrightarrow{\sigma} \mathsf{q}'$. Já a notação $\mathsf{q} \xrightarrow{\sigma}$ significa que $\mathsf{q} \xrightarrow{\sigma} \mathsf{q}'$ para algum estado q'. Por $G \xrightarrow{s}$ denota-se que a cadeia $s \in \Sigma^*$ é possível/aceita em G. Já $G \rightarrow \mathsf{q}$ expressa que $\mathsf{q}^{\circ} \xrightarrow{s} \mathsf{q}$ para alguma cadeia $s \in \Sigma^*$.

Em relação à ocorrência dos eventos a partir de um estado de um autômato pode-se identificar dois conjuntos. Por $\Gamma(\mathsf{q})$ denota-se o conjunto de todos os eventos possíveis de ocorrer a partir do estado q, ou seja, $\Gamma(\mathsf{q}) = \{\sigma \in \Sigma \mid \mathsf{q} \xrightarrow{\sigma}\}$. Diferentemente, o conjunto de eventos que não são possíveis de ocorrer a partir de q são todos os $\sigma \in \Sigma \backslash \Gamma(\mathsf{q})$, ou seja, são todos os eventos σ que pertencem ao alfabeto mas não são fisicamente possíveis em q (denotado por $\mathsf{q} \xrightarrow{\sigma}\!\!\!\!\!\!/\;\;$).

Uma alternativa para representar um AF graficamente é por meio de um grafo dirigido, de modo que os nós (vértices ou nodos) representam os estados e as arestas representam as transições entre os estados. A transição de estado é desencadeada perante a ocorrência de um determinado evento, associado a ela. O estado inicial é identificado por uma seta uniconectada apontada para si.

Exemplo 1 (Autômato Finito)**.** A Figura 1 ilustra um AF G descrito formalmente por:

- $\Sigma = \{a, b\}$;
- $Q = \{\mathsf{q}_0, \mathsf{q}_1\}$;
- $\mathsf{q}^{\circ} = \{\mathsf{q}_0\}$;
- $Q^{\omega} = Q$;

- $\rightarrow = \{\mathsf{q}_0 \xrightarrow{a} \mathsf{q}_1, \mathsf{q}_1 \xrightarrow{a} \mathsf{q}_1, \mathsf{q}_1 \xrightarrow{b} \mathsf{q}_0\}$.

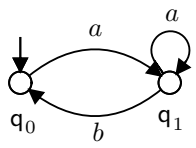


Figura 1: Autômato finito G

Nesse exemplo os conjuntos de eventos possíveis nos estados q_0 e q_1 são dados por $\Gamma(\mathsf{q}_0) = \{a\}$ e $\Gamma(\mathsf{q}_1) = \{a, b\}$, respectivamente. □

A *linguagem gerada* por um autômato G é definida como $L(G) = \{\, s \in \Sigma^* \mid \mathsf{q}^\circ \xrightarrow{s} \mathsf{q} \in Q \,\}$. Ou seja, $L(G)$ é o conjunto de todas as cadeias s em Σ^* mapeadas a partir do estado inicial que alcançam um estado em G, inclusive a cadeia ε.

A operação de *composição síncrona* ($\|$) de dois AFs, $A = \langle \Sigma_A, Q_A, Q_A^\omega, \mathsf{q}_A^\circ, \rightarrow_A \rangle$ *e* $\mathrm{B} = \langle \Sigma_B, Q_B, Q_B^\omega, \mathsf{q}_B^\circ, \rightarrow_B \rangle$, é um autômato

$$A \| B = (\Sigma_A \cup \Sigma_B, Q_A \times Q_B, (\mathsf{q}_A^\circ, \mathsf{q}_B^\circ), \rightarrow),$$

tal que a função de transição é dada por:

- $(\mathsf{q}_A, \mathsf{q}_B) \xrightarrow{\sigma} (\mathsf{q}'_A, \mathsf{q}'_B)$, se $\sigma \in \Sigma_A \cap \Sigma_B$, $\mathsf{q}_A \xrightarrow{\sigma} \mathsf{q}'_A$ e $\mathsf{q}_B \xrightarrow{\sigma} \mathsf{q}'_B$;
- $(\mathsf{q}_A, \mathsf{q}_B) \xrightarrow{\sigma} (\mathsf{q}'_A, \mathsf{q}_B)$, se $\sigma \in \Sigma_A \setminus \Sigma_B$ e $\mathsf{q}_A \xrightarrow{\sigma} \mathsf{q}'_A$;
- $(\mathsf{q}_A, \mathsf{q}_B) \xrightarrow{\sigma} (\mathsf{q}_A, \mathsf{q}'_B)$, se $\sigma \in \Sigma_B \setminus \Sigma_A$ e $\mathsf{q}_B \xrightarrow{\sigma} \mathsf{q}'_B$;
- *indefinida*, senão.

Isto é, dada a ocorrência de um evento habilitado em ambos autômatos, a evolução de estado ocorre de maneira síncrona em ambos os modelos. No caso, em que o evento é possível em somente um dos autômatos, ao passo que o outro autômato não reconheça tal evento em seu alfabeto, a evolução de estado ocorre de maneira assíncrona, ou seja, de maneira independente em cada autômato e somente aquele que reconhece o evento evolui de estado.

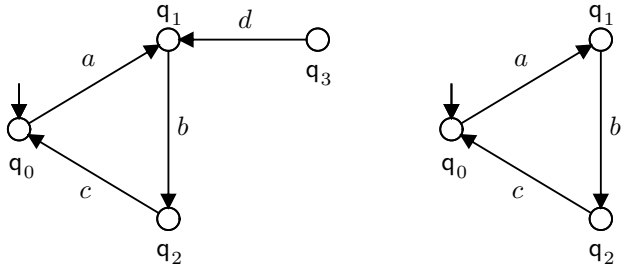
Para um conjunto de autômatos $\mathcal{G} = \{G^1, \cdots, G^n\}$, denota-se por $\|(\mathcal{G})$ a composição de todos os elementos em $\mathcal{G}$, i.e., $\|(\mathcal{G}) = G^1 \| \cdots \| G^n$.

Algumas propriedades importante sobre um autômato $G = \langle \Sigma, Q_G, \mathsf{q}^\circ_G, \rightarrow_G \rangle$, que são utilizadas na Seção 2.3, estão relacionadas ao conceito de acessibilidade. Essas propriedades são definidas como:

- *Estado acessível*: um estado $\mathsf{q} \in Q_G$ é dito ser *acessível* se existe ao menos uma cadeia $s \in \Sigma^*$ tal que $\mathsf{q}^\circ_G \xrightarrow{s} \mathsf{q}$;
- *Autômato acessível*: um AF G é dito ser *acessível* se todo estado $\mathsf{q} \in Q_G$ é acessível.

Em casos em que G é não acessível, um componente acessível G_{ac} pode ser obtido simplesmente eliminando os estados não acessíveis em G, bem como as transições conectadas a ele.

Exemplo 2 (Acessibilidade)**.** Considere o autômato G apresentado na Figura 2 (a).



(a) G (b) G_{ac}

Figura 2: Exemplo de acessibilidade

Note que o estado q_3 em G é não-acessível, pois a partir do estado inicial q_0 não existe nenhuma cadeia $s \in \Sigma^*$ capaz de alcançá-lo. Então, para obter a componente acessível G_{ac} basta remover o estado q_3 de G e a transição rotulada com o evento d que parte desse estado. O resultado disso é mostrado na Figura 2 (b). Note que, agora, todos os estados são alcançados a partir do estado inicial. □

As propriedades a seguir são importantes para o desenvolvimento desta dissertação (Seção 5.1) e são definidas em relação à possibilidade de eventos em autômatos.

Definição 1 (σ-habilitado)**.** *Dado um autômato $G = \langle \Sigma, Q, Q^\omega, q^\circ, \rightarrow \rangle$ e um evento $\sigma \in \Sigma$, G é dito ser* σ-habilitado*, se para todo estado $q \in Q$, é verdade que $\sigma \in \Gamma(q)$.*

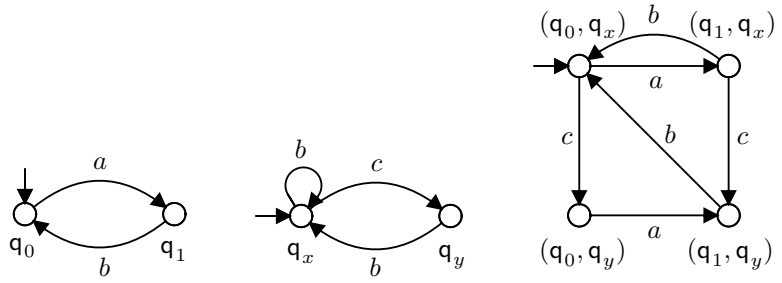
Essa definição pode ser estendida para um conjunto de eventos $\Sigma' \subseteq \Sigma$, se $\Sigma' \subseteq \Gamma(\mathsf{q})$.

A partir da definição de σ-habilitado e Σ'-habilitado pode-se caracterizar outras propriedades de interesse quanto à possibilidade de eventos em autômatos. Uma delas expressa a ideia de domínio de comportamento que um autômato exerce sobre um determinado evento e.r.a. outro autômato. Essa ideia é descrita formalmente como se segue.

Definição 2 ($\boldsymbol{\sigma}$-dominante)**.** *Sejam* $G^1 = \langle \Sigma_1, Q_1, Q_1^{\omega}, q_1^{\circ}, \rightarrow_1 \rangle$ *e* $G^2 = \langle \Sigma_2, Q_2, Q_2^{\omega}, q_2^{\circ}, \rightarrow_2 \rangle$ *dois autômatos, tal que* $\Sigma_1 \cap \Sigma_2 \neq \emptyset$*. Se* $G^1 \rightarrow q_1 \overset{\sigma}{\nrightarrow}$, $G^1 \rightarrow q_1{}' \overset{\sigma}{\rightarrow}$ *e* G^2 *é* σ*-habilitado (Def. 1), para* $\sigma \in \Sigma_1 \cap \Sigma_2$*, então* G^1 *é* σ*-dominante e.r.a.* G^2 *em* $G^1 \| G^2$*.*

Da definição de $\|$ pode-se concluir que se dois autômatos compartilham um evento $\sigma \in \Sigma_1 \cap \Sigma_2$ e esse evento é sempre habilitado em ambos, então σ será sempre habilitado na composição entre esses autômatos, i.e., $\sigma \in \Gamma((\mathsf{q}_1, \mathsf{q}_2))$ para todo $(\mathsf{q}_1, \mathsf{q}_2) \in Q_{G^1} \times Q_{G^2}$. No entanto, se σ é desabilitado por algum desses autômatos (no caso G^1) enquanto o outro autômato (G^2) sempre o habilita (Def. 1), então o comportamento daquele que o desabilita será preservado/dominante na composição, i.e., se $\sigma \notin \Gamma(\mathsf{q}_1)$ então $\sigma \notin \Gamma((\mathsf{q}_1, \mathsf{q}_2))$ e $\sigma \in \Gamma((\mathsf{q}_1{}', \mathsf{q}_2))$ se $\sigma \in \Gamma(\mathsf{q}_1{}')$. Sumarizando, a decisão da habilitação/desabilitação de um evento na composição de autômatos é análogo a uma porta lógica *and*, ou seja, se o evento compartilhado entre os autômatos é desabilitado por ao menos um deles, isso é uma condição suficiente para que esse evento seja desabilitado na composição e quem o desabilita é chamado de dominante.

Exemplo 3 ($\boldsymbol{\sigma}$-habilitado e $\boldsymbol{\sigma}$-dominante)**.** Considere os dois autômatos G^1 e G^2 das Figuras 3 (a) e (b) modelados com os alfabetos $\Sigma_1 = \{a, b\}$ e $\Sigma_2 = \{b, c\}$, respectivamente.



(a) G^1 (b) G^2 (c) $G^1 \| G^2_{\mathrm{a}}$

Figura 3: Modelos para G^1_{a}, G^2_{a} e E_{d}

Note que G^2 é b-habilitado visto que o evento b é possível em todos os estados q_i para $i \in \{x, y\}$. Diferentemente o evento b somente é possível em G^1 após o evento a, i.e., no estado q_1. Dessa forma, G^1 é b-dominante e.r.a. G^2 em $G^1 \| G^2$ da Figura 3 (c), tal fato resulta que b será elegível em $G^1 \| G^2$ somente após a ocorrência de a, i.e., nos estados $(\mathsf{q}_1, \mathsf{q}_i) \in Q_{G^1} \times Q_{G^2}$. □

Definição 3. *Sejam* $G^1 = \langle \Sigma_1, Q_1, Q_1^\omega, q_1^\circ, \rightarrow_1 \rangle$ *e* $G^2 = \langle \Sigma_2, Q_2, Q_2^\omega, q_2^\circ, \rightarrow_2 \rangle$ *dois autômatos, tal que* $\Sigma_1 = \Sigma_2$. G^1 *é dito ser uma* sobre-aproximação *de* G^2*, se após qualquer cadeia* $s \in \Sigma_1^*$*, tal que* $G^1 \xrightarrow{s} q_1$ *e* $G^2 \xrightarrow{s} q_2$*, é sempre verdade que* $\Gamma(q_2) \subseteq \Gamma(q_1)$*. Isso, implica que* $L(G^2) \subseteq L(G^1)$*.*

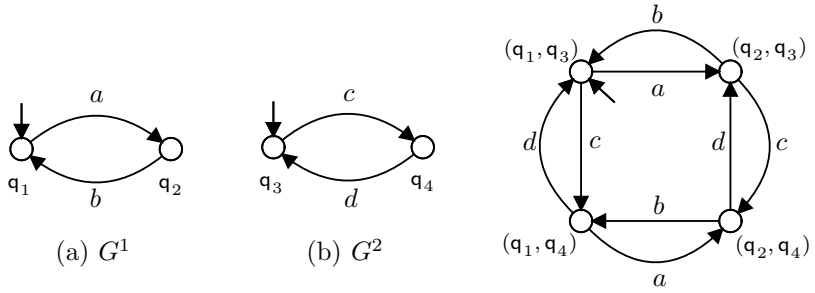
Se G^1 é uma sobre-aproximação de G^2 então toda cadeia reconhecida por G^2 também é reconhecida por G^1. No entanto, o contrário nem sempre é verdade.

Expostas as características principais de SEDs no contexto deste trabalho, a seguir será apresentado como tais características podem ser exploradas para a obtenção de lógicas de controle para SEDs.

2.3 TEORIA DE CONTROLE SUPERVISÓRIO

Os SEDs geralmente são formados por inúmeros subsistemas, de modo que cada subsistema pode ser modelado por um autômato G^i, tal que a composição $G = \prod_{i=1}^{n} G^i$ resulta no comportamento global do sistema, denominado de *planta*. A planta G expressa o comportamento do sistema em *malha aberta*, i.e., G não possui nenhuma ação de controle implementada sobre ela.

Exemplo 4 (Planta)**.** A Figura 4 ilustra um exemplo simples de um SED composto por duas máquinas M_1 e M_2, modeladas respectivamente por G^1 e G^2 (Figuras 4 (a) e (b)), tal que $G = G^1 \| G^2$ (Figura 4 (c)) expressa o comportamento da planta em malha aberta.



(a) G^1 (b) G^2 (c) $G^1 \| G^2$

Figura 4: Modelagem da planta de um SED

Assume-se que o comportamento das máquinas M_1 e M_2 é expresso em termos de eventos de início e fim de operação. Os eventos a e c modelam os respectivos inícios, enquanto que b e d os finais de operação. □

Na prática, os subsistemas operam de forma coordenada conforme um conjunto de regras pré-estabelecidas. Por exemplo, a precedência da máquina M_1 e.r.a. M_2. A

tarefa de impor essas regras sobre a planta G é atribuída a um autômato E, chamado de *especificação*. O modelo de E pode ser obtido de maneira análoga a G, i.e., $E = \|_{j=1}^{m} E^j$, em que cada E^j exerce uma ação proibitiva sobre G, de modo a observar os eventos possíveis na planta, e os desabilita quando são considerados como proibidos. Tal ação resulta no comportamento tido como esperado K implementado pelo autômato $G^K = G \| E$, ou seja, o comportamento desejado.

Exemplo 5 (Especificação)**.** Um exemplo de especificação associada à planta G (Figura 4 (c)), seria considerar que as máquinas M_1 e M_2 operam sequencialmente, i.e., M_2 inicia após o final de operação de M_1, e assim sucessivamente. O modelo E da Figura 5 expressa essa regra de controle, no qual as linhas tracejadas indicam quais eventos estão desabilitados em cada estado.

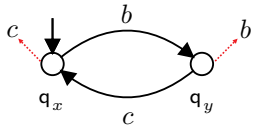


Figura 5: Modelagem de uma especificação

No estado inicial o evento c é desabilitado ($\mathsf{q}_x \overset{c}{\nrightarrow}$), de modo a evitar o início de operação de M_2, sem que antes M_1 tenha finalizado (evento b). Enquanto no estado q_y o evento b é desabilitado ($\mathsf{q}_y \overset{b}{\nrightarrow}$), evitando assim que M_1 finalize consecutivas vezes antes do início de M_2 (evento c). □

Em alguns casos, o comportamento modelado por K pode não ser condizente com o comportamento que se pode controlar de fato. Tal inconsistência ocorre pelo fato de E estar desabilitando eventos que não podem ser diretamente desabilitados em G, i.e., eventos que possuem a particularidade de serem espontâneos e não dependerem de qualquer política de controle. Por exemplo, os sinais de sensores que o sistema utiliza para obter informação são tipos de eventos que não podem ser diretamente desabilitados. Assim, para evitar essa inconsistência seria necessário estabelecer uma forma de distinguir a natureza da ocorrência dos eventos que integram a planta, de tal forma que a ação de controle tenha o conhecimento de quais eventos podem ser diretamente desabilitados ou não.

Essa ideia é incorporada pela TCS (RAMADGE; WONHAM, 1989), que define um método formal para a síntese de controladores, o qual é estruturado sobre autômatos e linguagens. Para isso, a TCS particiona o conjunto de eventos de um SED em $\Sigma = \Sigma_{\mathrm{c}} \cup \Sigma_{\mathrm{u}}$, em que Σ_{c} contém os eventos *controláveis* (que podem ser inibidos em G), enquanto Σ_{u}

os *não-controláveis* (não podem ser desabilitados diretamente em G), e então executa a síntese de modo a incorporar a análise de controlabilidade, definida como se segue.

Definição 4 (Controlabilidade)**.** *(RAMADGE; WONHAM, 1989) Sejam $K \subseteq \Sigma^*$ e $L(G) \subseteq \Sigma^*$ duas linguagens prefixo-fechadas. K é dita ser controlável e.r.a. $L(G)$ se $K\Sigma_{\mathrm{u}} \cap L(G) \subseteq K$.*

Ou seja, após qualquer prefixo em K, se um evento não-controlável é observado na planta ($L(G)$), a cadeia resultante continua sendo um prefixo em K. Dessa forma, garante-se que nenhum evento não-controlável, possível em $L(G)$, esteja sendo desabilitado pela ação de controle.

Em casos em que K é não controlável, pode-se reduzi-la à sua *máxima sub-linguagem controlável*, por meio da operação matemática sup$\mathcal{C}(K,G) = \cup\{\, K' \subseteq K \mid K'$ é controlável e.r.a. L$(G)\,\}$ (RAMADGE; WONHAM, 1989).

O cálculo de sup$\mathcal{C}(K,G)$ é um processo iterativo conhecido como síntese, que identifica e elimina os maus estados de um autômato $G^K = G\|E$ que implementa o comportamento de K. Maus estados podem ser definidos como segue.

Definição 5 (Mau Estado)**.** *Seja G uma planta e E um modelo de especificação para G. Um estado $(\mathsf{q}, \mathsf{w}) \in Q_G \times Q_E$ é dito ser mau estado se, existe $s \in \Sigma^*$ e $\mu \in \Sigma_{\mathrm{u}}$, tal que*

$$G \xrightarrow{s} \mathsf{q} \xrightarrow{\mu} \mathsf{q}' \qquad e \qquad G\|E \xrightarrow{s} (\mathsf{q}, \mathsf{w}) \not\xrightarrow{\mu} .$$

Ou seja, se após uma cadeia s, um evento não-controlável μ é possível em G, mas não em $G\|E$. Assim, o estado (q, w) alcançado pela cadeia s no autômato $G\|E$ é um mau estado.

O algoritmo que calcula sup$\mathcal{C}(K,G)$ pode ser definido basicamente pelos três passos:

Passo 1 - Identificar maus estados em G^K

Caso não existam, faça $V = G^K$;
fim.

Passo 2 - Remover maus estados em G^K

Caso existam, atualize G^K, eliminando os maus estados identificados;

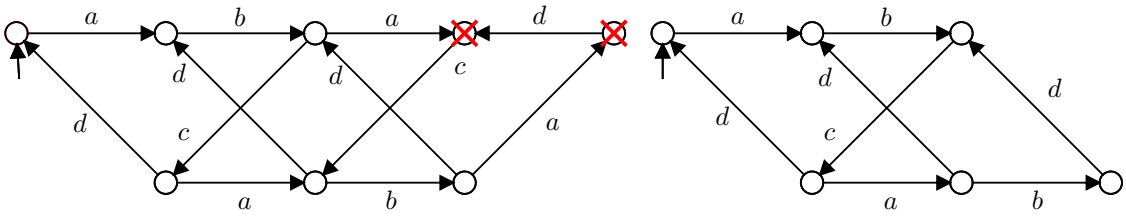
Passo 3 - Testar a acessibilidade em G^K

Calcule a componente acessível de G^K;

Volte ao passo 1.

Assim, sup$\mathcal{C}(K,G)$ pode ser associado ao comportamento controlável menos restritivo possível de ser implementado por um supervisor/controlador para G, de modo a respeitar um conjunto de especificações.

Exemplo 6 (Síntese)**.** A Figura 6 ilustra o processo de síntese para G e E das Figuras 4 (c) e 5, considerando que $\Sigma_c = \{a, c\}$ (controláveis) e $\Sigma_u = \{b, d\}$ (não-controláveis).



(a) $G^K = G \| E$ (b) V

Figura 6: Exemplo do processo de síntese

O comportamento desejado K implementado por G^K da Figura 6 (a), neste caso, é não controlável e.r.a. $L(G)$, uma vez que os estados marcados com um X (alcançados pelas cadeias *aba* e *abcaba*) desabilitam o evento b enquanto a planta o habilita. Dessa forma, esses estados são classificados como maus estados pela Def. 5 (passo 1), i.e., não-controláveis. Com o intuito de obter um comportamento que atenda a especificação e ao mesmo tempo seja controlável, a operação sup$\mathcal{C}(K,G)$ remove os maus estados (passo 2) de G^K, bem como as transições que incidem ou partem deles. Por fim, como $G^K = V$ (Figura 6 (b)) é acessível e não possui maus estados (passos 3 e 1), então esse é o comportamento do controlador minimamente restritivo e controlável para G. □

2.3.1 ESTUDO DE CASO

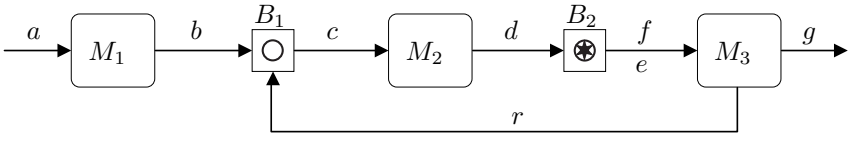
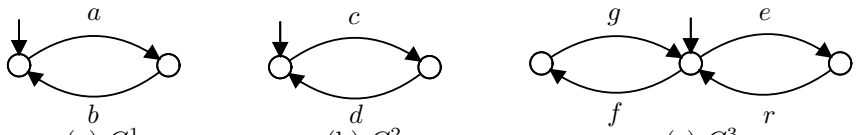


Figura 7: Sistema de manufatura com retrabalho

Considere o sistema de manufatura com retrabalho mostrado na Figura 7, adaptado de (TEIXEIRA et al., 2014), o qual é composto por três máquinas, M_1, M_2 e M_3, interconectadas por dois *buffers* unitários, B_1 e B_2. O *buffer* B_1 recebe novas peças

(evento b) ou peças a serem retrabalhadas (evento r). A máquina M_2 remove peças de B_1 (evento c), manufatura e as deposita em B_2 (evento d), onde um teste de qualidade determina qual operação será executada por M_3 (eventos e ou f), que resulta na liberação da peça (evento g) ou no retorno para retrabalho (evento r). Os AFs G^1, G^2 e G^3 na Figura 8 modelam M_1, M_2 e M_3, respectivamente.

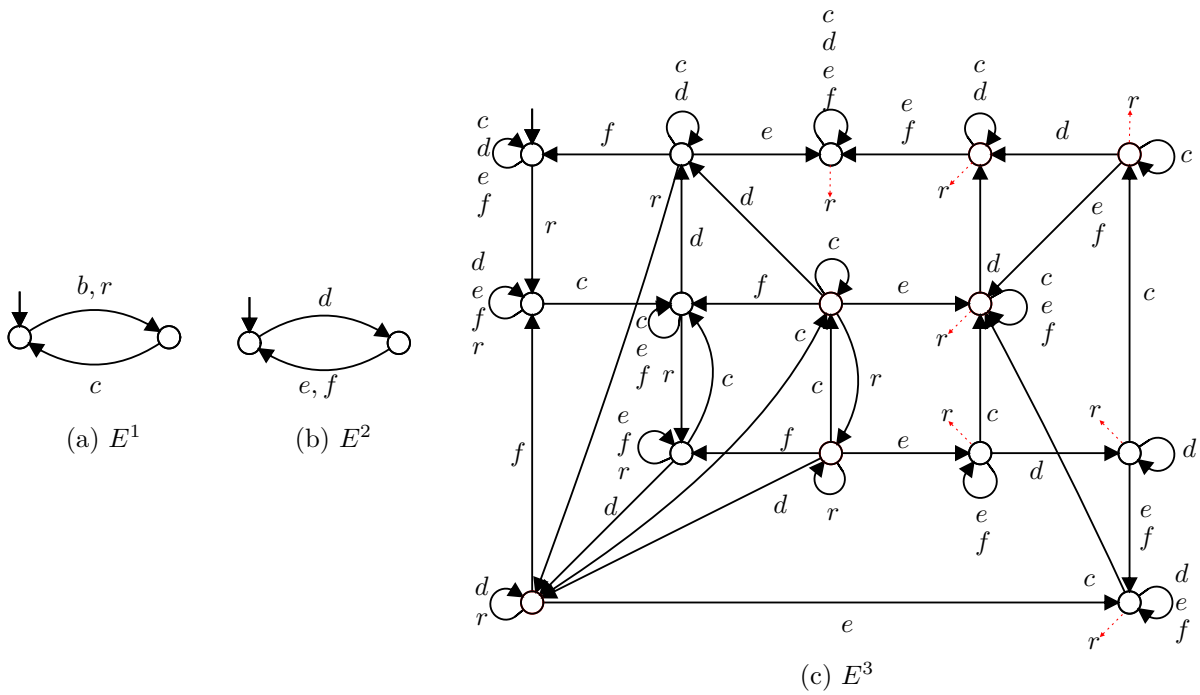


(a) G^1 (b) G^2 (c) G^3

Figura 8: Modelos para M_1, M_2 e M_3

A planta do sistema é dada pela composição $G = G^1 \| G^2 \| G^3$, com $\Sigma = \{a, b, c, d, e, f, g, r\}$, para $\Sigma_c = \{a, c, e, f\}$ e $\Sigma_u = \{b, d, g, r\}$.

Os objetivos de controle para este exemplo são: (i) evitar o *underflow* e *overflow* em B_j, $j = 1, 2$; (ii) limitar a um o número de ciclos de retrabalho de cada peça. O requisito (i) é implementado pelos modelos E^1 e E^2, enquanto que o requisito (ii) é implementado por E^3, os quais são representados conforme na Figura 9.



(a) E^1 (b) E^2 (c) E^3

Figura 9: Modelos para E^1, E^2 e E^3

Um fato importante a se observar nesse exemplo é que, como o modelo E^3 mapeia o percurso a partir de B_1 percorrido por cada uma das peças inseridas no sistema, de modo a ser possível identificar a quantidade de retrabalho realizado sobre cada uma delas.

Assim, caso o número máximo de retrabalhos permitidos fosse alterado para 5, estima-se que tal alteração exigiria que o engenheiro projetasse um novo modelo E^3 composto por 369 estados e 1179 transições. Como alternativa, a seguir é apresentada uma extensão da TCS que permite simplificar a tarefa de modelagem de especificações.

2.4 TCS COM DISTINGUIDORES

Embora, a TCS forneça, por construção, características/propriedades importantes para controladores como controlabilidade e máxima permissividade, ela esbarra em dois quesitos de complexidade quando aplicada a sistemas industriais reais, sendo eles: (i) modelagem de especificações; (ii) processamento computacional da operação de síntese.

Com intuito de tratar do quesito (i), criou-se a ideia de (sensores) distinguidores, o que deu origem a *Teoria de Controle Supervisório com Distinguidores* (TCS-D) (BOUZON et al., 2008; CURY et al., 2015). Um *distinguidor* pode ser pensado como um tipo especial de sensor que, ao ser associado ao modelo de um SED provê mais detalhes sobre determinados eventos do sistema, i.e., os eventos refinados pelos distinguidores passam a incorporar o contexto em que os eventos originais são habilitados no sistema. Assim, utilizando-se das informações contidas nos refinamentos, a tarefa de modelagem dos requisitos de controle pode ser potencialmente facilitada, uma vez que os eventos refinados possuem mais contexto que os eventos originais. No entanto, utilizar-se dos benefícios trazidos pelo refinamento de eventos requer um modelo extra, denominado de distinguidor, responsável por atribuir significado a cada refinamento adicionado à síntese de controladores para SEDs. Formalmente, essa ideia pode ser assim expressada.

Considere um SED modelado por um autômato G com eventos em Σ. Na abordagem com distinguidores, assume-se que cada evento $\sigma \in \Sigma$ passa a representar uma máscara para um conjunto de eventos $\Delta^\sigma \neq \emptyset$. Os eventos em Δ^σ são escolhidos de maneira a identificar diferentes instâncias em que σ pode ocorrer no sistema (CURY et al., 2015), tornando Σ um conjunto de máscaras para os eventos em $\Delta = \Delta_{\mathrm{c}} \cup \Delta_{\mathrm{u}}$, para $\Delta_{\mathrm{u}} = \bigcup_{\sigma \in \Sigma_{\mathrm{u}}} \Delta^\sigma$ e $\Delta_{\mathrm{c}} = \bigcup_{\sigma \in \Sigma_{\mathrm{c}}} \Delta^\sigma$.

A relação entre os alfabetos Σ e Δ pode ser definida por mapas, como segue.

Definição 6. *Um mapa mascarador* $\Pi : \Delta^* \to \Sigma^*$ *é definido recursivamente por*

$$\Pi(\varepsilon) = \varepsilon,$$
$$\Pi(t\delta) = \Pi(t)\sigma, \text{ para } t \in \Delta^*, \delta \in \Delta^\sigma \text{ e } \sigma \in \Sigma.$$

Ou seja, Π é um mapa que atribui cada cadeia refinada, à respectiva cadeia de máscaras em Σ. A definição de Π, pode ser estendida para qualquer linguagem $L^d \subseteq \Delta^*$ por

$$\Pi(L^d) = \{s \in \Sigma^* \mid \exists t \in L^d,\, \Pi(t) = s\}.$$

Definição 7. *Um mapa mascarador inverso* $\Pi^{-1}\colon \Sigma^* \to 2^{\Delta^*}$ *é definido para qualquer cadeia* $s \in \Sigma^*$ *como*

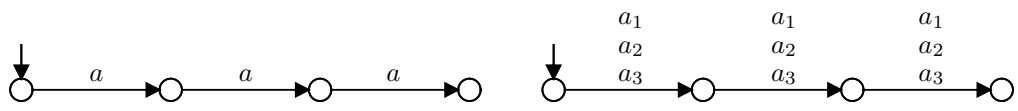
$$\Pi^{-1}(s) = \{t \in \Delta^* \mid \Pi(t) = s\}.$$

Análogo ao mapa Π, o Π^{-1} também pode ser estendido para qualquer linguagem $L \subseteq \Sigma^*$ por

$$\Pi^{-1}(L) = \{t \in \Delta^* \mid \Pi(t) \in L\},$$

e a mesma notação é assumida para AF de modo que $\Pi^{-1}(G)$ é um modelo que substitui os eventos de cada transição em G por seu respectivo conjunto de refinamentos.

Exemplo 7 (Efeito dos mapas $\mathbf{\Pi^{-1}}$ e $\mathbf{\Pi}$)**.** Considere a planta G da Figura 10 (a) com $\Sigma = \{a\}$, tal que $L(G) = \{\varepsilon, a, aa, aaa\}$.



(a) $G = \Pi(\Pi^{-1}(G))$ (b) $\Pi^{-1}(G)$

Figura 10: Efeito do mapa Π^{-1} sobre autômatos

Assumindo que $\Delta^a = \{a_1, a_2, a_3\}$, tal que os índices 1, 2 e 3 identificam as instâncias do evento a em G, o efeito de $\Pi^{-1}(L(G))$ é implementado pelo autômato $\Pi^{-1}(G)$ da Figura 10 (b). Note que Π^{-1} mapeia cada cadeia $s \in L(G)$ no conjunto de suas versões refinadas em Δ^*, i.e., $\Pi^{-1}(L(G)) = \{\varepsilon, \Delta^a, \Delta^a \times \Delta^a, \Delta^a \times \Delta^a \times \Delta^a\}$. Diferentemente, o mapa Π realiza o processo inverso que Π^{-1}, i.e., $\Pi(\Pi^{-1}(L(G))) = L(G)$. □

Definida a relação entre os alfabetos Σ e Δ, um distinguidor pode ser formalmente definido como segue.

Definição 8 (Distinguidor)**.** *(CURY et al., 2015) Dado um alfabeto* Σ*, e sua versão refinada* Δ*, um* distinguidor *é um mapa* $D : \Sigma^* \to 2^{\Delta^*}$ *definido, para todo* $s \in \Sigma^*, \sigma \in$

$\Sigma, r \in \Delta^*$ *e* $\delta \in \Delta$*, por:*

$$(i)\ D(\varepsilon) = \{\varepsilon\};$$
$$(ii)\ \textit{se}\ r\delta \in D(s\sigma)\ \textit{então}\ r \in D(s)\ \textit{e}\ \delta \in \Delta^\sigma;$$
$$(iii)\ \textit{se}\ r \in D(s)\ \textit{então}\ \exists \delta' \in \Delta^\sigma\ \textit{tal que}\ r\delta' \in D(s\sigma).$$

Ou seja, D é um mapa prefixo-preservante que sempre distingue e nunca desabilita totalmente uma máscara, i.e., sempre que uma máscara σ é possível em Σ ao menos um $\sigma_i \in \Delta^\sigma$ é possível em Δ. Assim, D simplesmente substitui cada cadeia em Σ^* por um conjunto não vazio de cadeias em Δ^*.

Análogo aos mapas Π e Π^{-1}, um distinguidor D também pode ser estendido para qualquer linguagem $L \subseteq \Sigma^*$ por

$$D(L) = \{r \in \Delta^* \mid \exists s \in L,\ r \in D(s)\}.$$

Em (CURY et al., 2015), os autores demonstram que $\Pi(D(L)) = L$, i.e., uma linguagem distinguida por D pode retornar a sua versão original em Σ por meio do mapa Π. Nesse contexto, a linguagem de um distinguidor pode ser definida como na Def. 9.

Definição 9 (Linguagem do distinguidor)**.** *(CURY et al., 2015) Para um distinguidor* D*, como na Def. 8, qualquer linguagem* $L_\mathrm{d} \subseteq \Delta^*$ *tal que* $L_\mathrm{d} = D(\Sigma^*)$ *e* $\Pi(L_\mathrm{d}) = \Sigma^*$ *pode ser definida como a linguagem de* D*.*

Note que D aproxima cadeias em Σ^*, i.e., $L_\mathrm{d} \subseteq \Delta^* = \Pi^{-1}(\Sigma^*)$. Então, o mapa Π^{-1} torna-se um caso particular do distinguidor quando $L_\mathrm{d} = \Delta^*$.

Assim, a partir da definição de Π^{-1} e L_d o efeito de um distinguidor pode ser reintroduzido por

$$D(L(G)) = \Pi^{-1}(L(G)) \cap L_\mathrm{d},$$

e a mesma notação é assumida para AF de modo que $D(G) = \Pi^{-1}(G) \| H_\mathrm{d}$, onde H_d é um autômato que implementa L_d. Os passos para a construção do autômato H_d são apresentados no Apêndice A.

O diagrama de blocos da Figura 11 contextualiza a interação entre o distinguidor D e a planta G.

Nesse diagrama, assume-se que a cada ocorrência de $\sigma \in \Sigma$ em G, esse evento é mapeado em um subconjunto Δ^σ de refinamentos, por meio do mapa Π^{-1}, que por sua vez

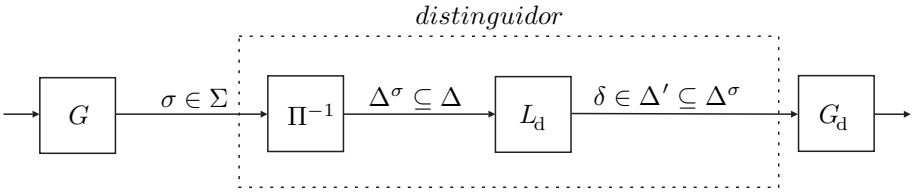


Figura 11: Diagrama de blocos interação de D e G

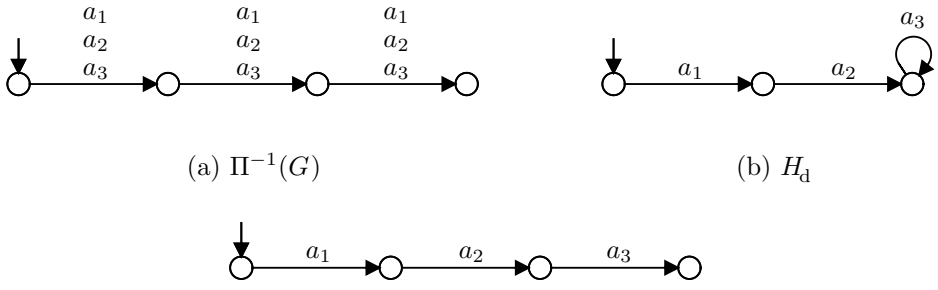
são filtrados pela linguagem $L_{\mathrm{d}} = L(H_{\mathrm{d}})$, a qual define de fato quais deles são habilitados na planta distinguida $G_{\mathrm{d}} = \Pi^{-1}(G) \| H_{\mathrm{d}}$.

Nesse contexto, outro caso particular do distinguidor D é quando, cada cadeia em Σ^* é associada a uma única cadeia em Δ^*, tal caso pode ser definido como segue.

Definição 10 (Preditibilidade)**.** *(CURY et al., 2015) Um distinguidor D (Def. 8) é dito ser preditível se, para todo $s \in \Sigma^*$, $|D(s)| = 1$. Do contrário, D é dito ser não-preditível.*

Ou seja, D é *preditível* se qualquer $s \in \Sigma^*$ é mapeada em exatamente uma cadeia $r \in \Delta^*$ ou, ainda, após qualquer cadeia $r \in L_{\mathrm{d}}$, existe exatamente um evento $\delta \in \Delta^\sigma$ elegível para cada $\sigma \in \Sigma$. Assim, a linguagem L_{d} permite sempre *predizer* a próxima distinção para qualquer máscara.

Exemplo 8 (Efeito do distinguidor $\boldsymbol{D}$ sobre uma planta $\boldsymbol{G}$)**.** Considerando a planta G com $\Sigma = \{a\}$ do Exemplo 7 e $\Delta^a = \{a_1, a_2, a_3\}$, o efeito D sobre G é ilustrado na Figura 12.



(a) $\Pi^{-1}(G)$ (b) H_{d}

(c) G_{d}

Figura 12: Efeito de D sobre G

O modelo do distinguidor H_{d} (Figura 12 (b)) é construído por meio do conjunto de passos apresentados no Apêndice A. Assim, o distinguidor D refina o evento a por meio do mapa Π^{-1} (Figura 12 (a)), e distingue os refinamentos em Δ^a por intermédio de $L_{\mathrm{d}} = L(H_{\mathrm{d}})$ (Figura 12 (b)) resultando na planta distinguida G_{d} (Figura 12 (c)). Além

disso, note que nesse caso D é predítivel (Def. 10), visto que cada cadeia em $L(G) = \{\varepsilon, a, aa, aaa\}$ é mapeada em uma única cadeia em $D(L(G)) = \{\varepsilon, a_1, a_1a_2, a_1a_2a_3\}$. □

Nesse contexto, seja G_{d} uma planta distinguida para G e E_{d} a versão da especificação E obtida utilizando os refinamentos em Δ, tal que $K_{\mathrm{d}} = L(G_{\mathrm{d}} \| E_{\mathrm{d}})$. Pode ser demonstrado (CURY et al., 2015) que, se D é preditível e se $K = \Pi(K_{\mathrm{d}})$, então

$$\Pi(\sup\mathcal{C}(K_{\mathrm{d}}, G_{\mathrm{d}})) = \sup\mathcal{C}(K, G) \text{ e}$$

$$\sup\mathcal{C}(K_{\mathrm{d}}, G_{\mathrm{d}}) = D(\sup\mathcal{C}(K, G)).$$

Ou seja, sob esses pressupostos, as duas versões do problema (com e sem distinguidor), levam à mesma solução de controle.

2.4.1 ESTUDO DE CASO COM DISTINGUIDORES

Com o intuito de ilustrar como distinguidores podem ser utilizados para simplificar a etapa de modelagem de especificações, revisita-se o exemplo do sistema de manufatura apresentado na subseção 2.3.1, reproduzido a seguir.

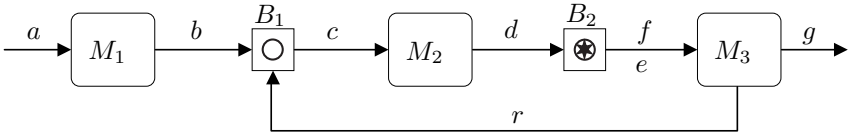
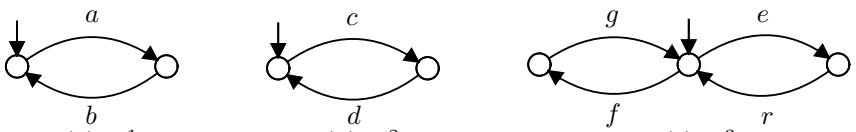


Figura 13: Sistema de manufatura com retrabalho

A planta do sistema é representada pelos autômatos G^1, G^2 e G^3 da Figura 14.

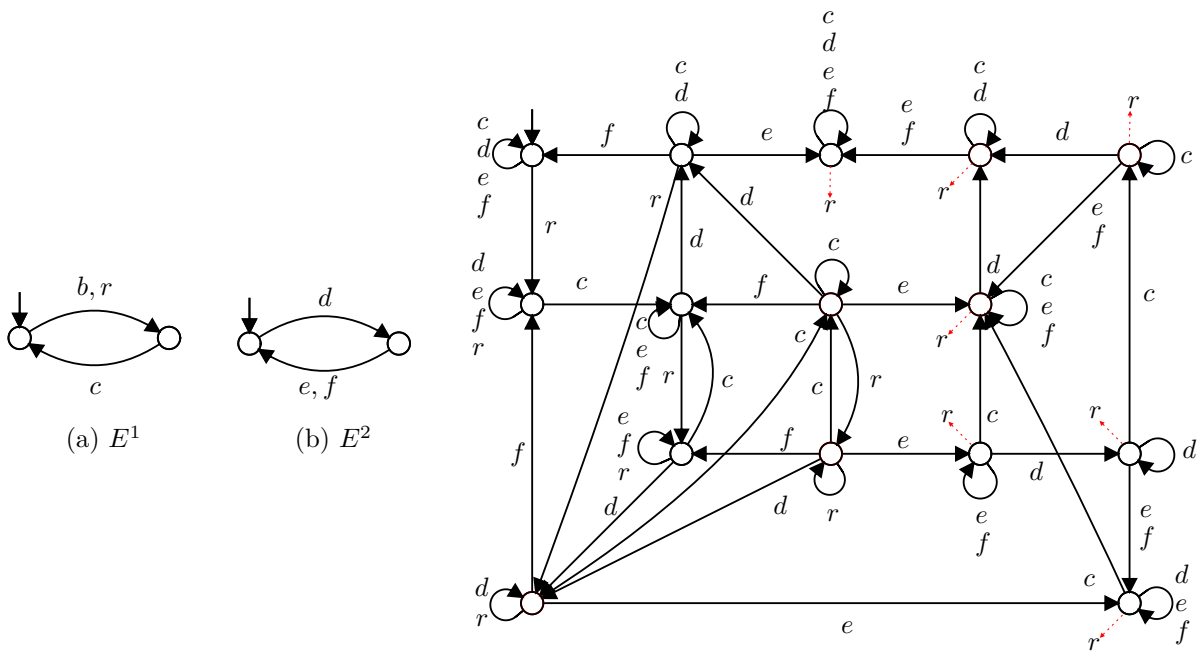


(a) G^1 (b) G^2 (c) G^3

Figura 14: Modelos para M_1, M_2 e M_3

Para esse sistema, foram definidas três especificações de controle, sendo E^1 e E^2 destinadas a evitar o *underflow* e o *overflow* nos *buffers* B_1 e B_2, e E^3 visando limitar a um o número máximo de ciclos de retrabalho para cada peça. Esse conjunto de especificações é implementado pelos autômatos da Figura 15.

Note que E^3 (Figura 15 (c)), com 15 estados, pode ser visto como o modelo mais "complexo"dentre as especificações deste exemplo, uma vez que ele mapeia o trajeto



(a) E^1 (b) E^2

(c) E^3

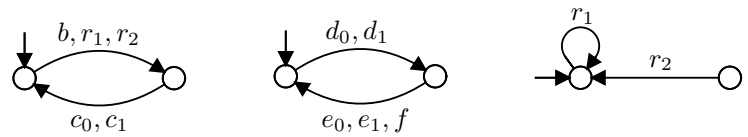
Figura 15: Modelos para E^1, E^2 e E^3

(cadeias de eventos) a partir de B_1 para cada uma das três possíveis peças em paralelo no sistema até que se possa determinar o segundo ciclo de trabalho, e assim evitá-lo (estados com uma seta tracejada apontando para r). Então, esse modelo pode ser simplificado por refinamentos que identifiquem quantas vezes cada peça foi retrabalhada. Para isso, utilizam-se os passos apresentados no Apêndice A.

Como E^3 toma uma ação de controle de acordo com o número de vezes que o evento r ocorreu para cada peça no sistema, seria natural pensar em refiná-lo em r_1 e r_2 (passo (i) e (ii)), de modo que r_1 representa a primeira instância de r, enquanto r_2 a segunda. No entanto, como o contexto das instâncias de r dependem do trajeto que cada peça percorre no sistema a partir de B_1, i.e., dos eventos c, d e e, tais eventos devem ser refinados também (passo (iii)), de modo a identificar cada ciclo de trabalho.

Assim, seja $\Delta^\sigma = \{\sigma_0, \sigma_1\}$, para $\sigma \in \{c, d, e\}$, e $\Delta^r = \{r_1, r_2\}$, tal que 0, 1 e 2 indicam a quantidade de retrabalhos efetuados. Então, a versão refinada de Σ é dada por $\Delta = \bigcup_{\sigma\in\Sigma} \Delta^\sigma$ e, utilizando-se das informações em Δ, E^3 pode ser modelado por E^3_{d} da Figura 9 (c), para $\Delta_{E^3_{\mathrm{d}}} = \Delta^r$. As versões de E^1 e E^2 passam a ser modeladas em Δ por $E^1_{\mathrm{d}} = \Pi^{-1}(E^1)$ e $E^2_{\mathrm{d}} = \Pi^{-1}(E^1)$ da Figura 16 (a) e (b) respectivamente, e a especificação global é dada pela composição $E_{\mathrm{d}} = E^1_{\mathrm{d}} \| E^2_{\mathrm{d}} \| E^3_{\mathrm{d}}$.

O modelo E^3_{d} desabilita o evento r_2 de modo que o segundo ciclo de retrabalho é evitado. Um ponto importante a se observar e.r.a. E^3_{d} é que o seu número de

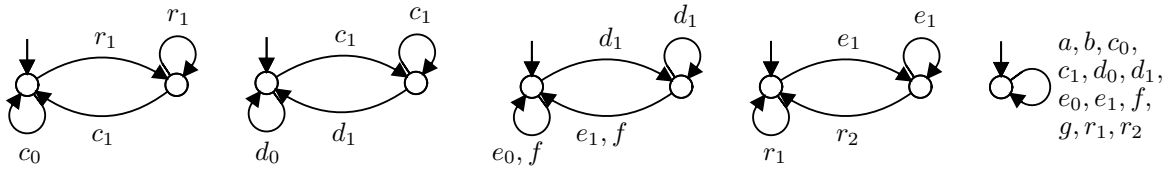


(a) $E^1_{\rm d} = \Pi^{-1}(E^1)$ (b) $E^2_{\rm d} = \Pi^{-1}(E^2)$ (c) $E^3_{\rm d}$

Figura 16: Modelos para $E^1_{\rm d}$, $E^2_{\rm d}$ e $E^3_{\rm d}$

estados independe da quantidade de retrabalhos e do número de peças processadas em paralelo, enquanto que a quantidade de estados de E^3 está diretamente ligado a essas duas restrições. Por exemplo, se o limite de retrabalho tivesse sido incrementado em dois, estima-se que o modelo de E^3 tivesse 369 estados, ao passo que $E^3_{\rm d}$ ainda permaneceria com dois estados.

Resta mostrar como os eventos em Δ podem ser distinguidos. Essa tarefa é atribuída aos distinguidores $H^c_{\rm d}$, $H^d_{\rm d}$, $H^e_{\rm d}$, $H^r_{\rm d}$ e $H^\Delta_{\rm d}$ da Figura 17. O distinguidor preditível, para este exemplo, é dado pela composição $H_{\rm d} = H^c_{\rm d} \| H^d_{\rm d} \| H^e_{\rm d} \| H^r_{\rm d} \| H^\Delta_{\rm d}$.



(a) $H^c_{\rm d}$ (b) $H^d_{\rm d}$ (c) $H^e_{\rm d}$ (d) $H^r_{\rm d}$ (e) $H^\Delta_{\rm d}$

Figura 17: Versão modular do distinguidor $H_{\rm d}$

Agora, para $G_{\rm d} = \Pi^{-1}(G) \| H_{\rm d}$ e $G^{K_{\rm d}} = E_{\rm d} \| G_{\rm d}$, tal que $K_{\rm d} = L(G^{K_{\rm d}})$, sintetiza-se um supervisor sup$\mathcal{C}(K_{\rm d},G_{\rm d})$, modelado por $V_{\rm d}$ na segunda linha da Tabela 1. Para fins comparativos com o caso em Σ, pode ser também calculado um supervisor sup$\mathcal{C}(K,G)$, para G e $K = L(G^K)$, em que $G^K = E^1 \| E^2 \| E^3 \| G$, o qual é modelado por V na primeira linha da Tabela 1.

Tabela 1: Síntese - estados (transições)

G	E	G^K	V
12 (40)	34 (96)	192 (364)	26 (45)
$G_{\rm d}$	$E_{\rm d}$	$G^{K_{\rm d}}$	$V_{\rm d}$
84 (276)	4 (18)	192 (364)	26 (45)

Note que, o modelo de $E_{\rm d}$ com 4 estados é mais simples que E, com 34 estados. No entanto, os modelo de G^K e $G^{K_{\rm d}}$ possuem a mesma quantidade de estados e transições, então a complexidade para se computar os controladores V e $V_{\rm d}$ é a mesma. Isso se deve

ao fato de que, na abordagem com refinamentos, os benefícios obtidos na simplificação da especificação são compensados pela adição do modelo do distinguidor a planta.

Sumarizando, pode-se dizer que as vantagens do uso de distinguidores, como apresentados até aqui, se resumem a simplificar tarefas de modelagem de especificações complexas sem, contudo, aumentar a complexidade de síntese, levando ainda assim ao controlador ótimo. É bem verdade, por outro lado, que essa abordagem também não simplifica diretamente a síntese, o que informalmente seria esperado como consequência de se trabalhar com modelos de especificações mais simples. Uma alternativa que permite explorar as vantagens do uso de refinamentos de eventos na síntese é apresentada a seguir e sustenta a proposta deste trabalho.

2.5 TCS COM APROXIMAÇÕES

A fim de estender os benefícios trazidos pelo refinamento de eventos também para a etapa de síntese, foi introduzido na literatura o conceito de *aproximações* (CURY et al., 2015). É mostrado (CURY et al., 2015) que, em determinados casos, uma aproximação permite calcular um controlador com menos ônus computacional, cujo comportamento é equivalente à versão obtida com distinguidores.

Essa abordagem consiste basicamente em fazer o uso das especificações modeladas em Δ, mas substituir, na síntese, a planta distinguida G_{d} por uma aproximação G_{a}, que em geral possui menos estados que G_{d}. Formalmente, uma aproximação pode ser definida como se segue.

Definição 11 (Aproximação)**.** *(CURY et al., 2015) Seja $D : \Sigma^* \to 2^{L_{\mathrm{d}}}$ um distinguidor preditível (Def. 10) para uma planta G e seja $D_{\mathrm{a}} : \Sigma^* \to 2^{L_{\mathrm{da}}}$ um distinguidor tal que $L_{\mathrm{d}} \subseteq L_{\mathrm{da}} \subseteq \Delta^*$. Então, qualquer AF $G_{\mathrm{a}} = D_{\mathrm{a}}(G)$ é uma* Aproximação *para $G_{\mathrm{d}} = D(G)$.*

Ou seja, D_{a} é um distinguidor que preserva as cadeias distinguidas por D preditível, que quando aplicado sobre uma planta G produz uma aproximação $G_{\mathrm{a}} = D_{\mathrm{a}}(G)$ para $G_{\mathrm{d}} = D(G)$, tal que $L(G_{\mathrm{d}}) \subseteq L(G_{\mathrm{a}})$. A principal diferença entre D_{a} e D é que, o distinguidor D_{a} pode talvez habilitar um conjunto maior de cadeias em Δ^* do que D preditível, i.e., para $s \in \Sigma^*$, $|D_{\mathrm{a}}(s)| \geq |D(s)| = 1$.

Resta definir, então, como modelar um distinguidor H_{da}, tal que $L_{\mathrm{da}} = L(H_{\mathrm{da}})$, de modo que quando associado à planta G produza uma aproximação para G_{d} e, consequentemente resulte no processo de síntese computacionalmente vantajoso, preservando-se a

otimidade do controlador. Nesse sentido, assumindo que H_{d}, tal que $L_{\mathrm{d}} = L(H_{\mathrm{d}})$, é modular por construção (Apêndice A), e que H_{d} distingue mais que H_{da} ($L_{\mathrm{d}} \subseteq L_{\mathrm{da}}$), então a construção de H_{da} pode se dar simplesmente desconsiderando partes do modelo H_{d}.

Assim, dois casos particulares que limitam as aproximações e.r.a. $L_{\mathrm{d}} \subseteq L_{\mathrm{da}} \subseteq \Delta^*$ são:

(i) o primeiro é quando $H_{\mathrm{da}} = H_{\mathrm{d}}$, i.e., $L_{\mathrm{da}} = L_{\mathrm{d}}$, nesse caso a aproximação G_{a} é igual a G_{d} e, portanto, não simplifica a síntese, ainda que preserve os benefícios da modelagem;

(ii) o segundo, e mais interessante computacionalmente, é quando $L_{\mathrm{da}} = \Delta^*$, i.e., quando todo o modelo de H_{d} é removido da síntese. Nesse caso, D_{a} é modelado por H_{d}^{Δ}, um autômato composto por um único estado com uma transição em auto-laço, rotulada com todo os eventos em Δ, i.e., $H_{\mathrm{d}}^{\Delta} \xrightarrow{\sigma_i} \mathsf{q}^{\circ}$ para todo $\sigma_i \in \Delta$. Assim, $G_{\mathrm{a}} = \Pi^{-1}(G)$ é um modelo com o mesmo número de estados de G.

Na prática, parte-se do caso (ii), visto que esse caso é o que mais beneficia a síntese $\sup\mathcal{C}(K_{\mathrm{a}},G_{\mathrm{a}})$ em termos computacionais. No entanto, pode ser mostrado (CURY et al., 2015) que, em certos casos, $\sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}}) \cap L_{\mathrm{d}}$ leva a um comportamento subótimo (mais restritivo) do controlador, i.e.,

$$\sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}}) \cap L_{\mathrm{d}} \subset \sup\mathcal{C}(K_{\mathrm{d}}, G_{\mathrm{d}}).$$

Para esses casos, aproximações intermediárias podem ser construídas por meio da composição incremental de qualquer módulo de H_{d}, até que resulte no controlador ótimo, i.e.,

$$\sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}}) \cap L_{\mathrm{d}} = \sup\mathcal{C}(K_{\mathrm{d}}, G_{\mathrm{d}}).$$

2.5.1 ESTUDO DE CASO COM APROXIMAÇÕES

Com objetivo de ilustrar os ganhos obtidos com a abordagem de aproximações, revisita-se o exemplo do sistema de manufatura com retrabalho da subseção 2.3.1.

Assim, para $G_{\mathrm{a}} = \Pi^{-1}(G) \| H_{\mathrm{da}}$ e $K_{\mathrm{a}} = L(G^{K_{\mathrm{a}}})$, tal que $H_{\mathrm{da}} = H_{\mathrm{d}}^{\Delta}$ (Figura 17 (e)) e $G^{K_{\mathrm{a}}} = G_{\mathrm{a}} \| E_{\mathrm{d}}$, sintetiza-se o controlador $\sup\mathcal{C}(K_{\mathrm{a}},G_{\mathrm{a}})$, modelado por V_{a} da segunda linha da Tabela 2. Para fins comparativos os resultados da Tabela 1 foram reproduzidos nas duas primeiras linhas da Tabela 2.

Note que, em termos de estados, G_{a} é equivalente a G, e são mais simples que G_{d}. Além disso, a síntese de V_{a}, a partir de um modelo com 48 estados, é mais simples

Tabela 2: Síntese - estados (transições)

G	E	G^K		V
12 (40)	34 (96)	192 (364)		26 (45)
G_{d}	E_{d}	$G^{K_{\mathrm{d}}}$		V_{d}
84 (276)	4 (18)	192 (364)		26 (45)
G_{a}	E_{d}	$G^{K_{\mathrm{a}}}$	V_{a}	$V_{\mathrm{a}} \| H_{\mathrm{d}}$
12 (60)	4 (18)	48 (132)	18 (40)	18 (32)

do que o caso preditível V_{d}, com 192 estados. No entanto, a versão implementável do controlador, $V_{\mathrm{a}} \| H_{\mathrm{d}}$, é mais restritiva do que V_{d}, pois possui apenas 18 estados enquanto V_{d} possui 26 estados. Essa restrição se deve ao fato de que, com a aproximação, o processo de síntese dispõe de menos informações sobre a planta, i.e., as informações contidas nos eventos podem em geral ser ambíguas, o que leva o controlador a se precaver de sequências que possam interferir na controlabilidade.

Com intuito de aproximar o supervisor V_{a}' de V_{d}, i.e., tornar V_{a}' tão permissivo quanto V_{d}, tal que $V_{\mathrm{a}}' \| H_{\mathrm{d}} = V_{\mathrm{d}}$, partes do distinguidor podem ser usadas para enriquecer a aproximação ($G_{\mathrm{a}}{}'$) mantendo, ao mesmo tempo, o processo de síntese a partir de $G_{\mathrm{a}}{}'$ mais simples do que o caso preditível com G_{d}. Resta descobrir quais partes de H_{d} podem ser úteis a este propósito, o que é investigado na Seção 3.1

3 SÍNTESE USANDO AUTO-APROXIMAÇÕES

Este capítulo apresenta os primeiros resultados obtidos no contexto dessa dissertação (ROSA et al., 2018b, 2018a). Basicamente, ele estende e melhora os resultados da literatura relacionados à síntese de controladores com refinamento de eventos. É desenvolvido um método que permite encontrar e construir uma classe particular de aproximação que sempre leva à síntese ótima. A sistemática do método, bem como as propriedades, demonstrações e provas que o sustentam, são descritos a seguir.

3.1 TCS COM AUTO-APROXIMAÇÕES

A combinação dos conceitos de *distinguidor* e *aproximação* leva a um mecanismo de síntese que mantém os benefícios de modelagem trazidos por refinamentos de eventos e, ao mesmo tempo, tende a simplificar o cálculo do controlador. No entanto, essa abordagem pode resultar em soluções mais restritivas, devido ao baixo nível de distinção, usual de uma planta aproximada (exemplo da subseção 2.5.1).

Considerando que um distinguidor possa ser exposto de forma modular, uma possível alternativa para enriquecer uma aproximação, i.e., aumentar o seu poder de distinção, é selecionar partes específicas para serem compostas à planta.

Resta descobrir quais módulos poderiam ser compostos à planta, o que pode não ser trivial e, em geral, é associado a um processo de tentativa e erro. Na sequência, será descrita uma sistemática que leva ao conceito de *auto-aproximações* (ROSA et al., 2018b, 2018a), ou seja, plantas que por construção garantem a síntese controlável e menos restritiva. A noção de *estados alvo* e *eventos alvo* são a chave para se definir auto-aproximações.

3.1.1 ESTADOS ALVO

Quando uma aproximação é usada em síntese, ao menos três classes de estados podem ser identificadas, de acordo com a maneira como as transições (rotuladas com eventos refinados) são habilitadas, tais classes são elencadas na Def. 12.

Definição 12 (Estados alvo)**.** *Para uma planta G, com eventos em Σ, e um distinguidor preditível D, seja G_{a} uma aproximação para $G_{\mathrm{d}} = D(G)$, e seja E_{d} um modelo de especificação. Defina os seguintes conjuntos de estados:*

$\boldsymbol{\mathcal{S}^{\mathrm{u}}}$ (Estados Certamente Não-Controláveis)

$$\mathcal{S}^{\mathrm{u}} = \{\, (\mathsf{x},\mathsf{y}) \in Q_{G_{\mathrm{a}}} \times Q_{E_{\mathrm{d}}} \mid \text{existe } s \in \Delta^* \text{ e } \mu \in \Sigma_{\mathrm{u}} \text{ tal que para todo } \mu_i \in \Delta^{\mu},$$
$$\text{se } G_{\mathrm{a}} \xrightarrow{s} \mathsf{x} \xrightarrow{\mu_i} \mathsf{x}' \text{ então } G_{\mathrm{a}} \| E_{\mathrm{d}} \xrightarrow{s} (\mathsf{x},\mathsf{y}) \overset{\mu_i}{\nrightarrow} \,\} \,.$$

$\boldsymbol{\mathcal{S}^{\mathrm{e}}}$ (Estados Eventualmente-Controláveis)*:*

$$\mathcal{S}^{\mathrm{e}} = \{\, (\mathsf{x},\mathsf{y}) \in Q_{G_{\mathrm{a}}} \times Q_{E_{\mathrm{d}}} \mid (\mathsf{x},\mathsf{y}) \notin \mathcal{S}^{\mathrm{u}} \text{ e existe } \mu \in \Sigma_{\mathrm{u}}, s \in \Delta^* \text{ e } \mu_i, \mu_j \in \Delta^{\mu},$$
$$\text{com } \mu_i \neq \mu_j, \text{ tal que } G_{\mathrm{a}} \xrightarrow{s} \mathsf{x} \xrightarrow{\mu_i,\mu_j} \mathsf{x}',\ G_{\mathrm{a}} \| E_{\mathrm{d}} \xrightarrow{s} (\mathsf{x},\mathsf{y}) \xrightarrow{\mu_i} (\mathsf{x}',\mathsf{y}'),$$
$$\text{e } G_{\mathrm{a}} \| E_{\mathrm{d}} \xrightarrow{s} (\mathsf{x},\mathsf{y}) \overset{\mu_j}{\nrightarrow} \,\} \,.$$

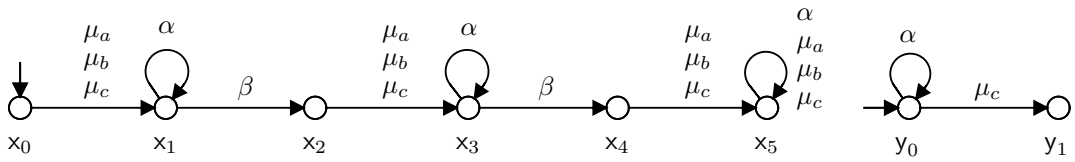
$\boldsymbol{\mathcal{S}^{\mathrm{c}}}$ (Estados Certamente-Controláveis)*:*

$$\mathcal{S}^{\mathrm{c}} = \{\, (\mathsf{x},\mathsf{y}) \in Q_{G_{\mathrm{a}}} \times Q_{E_{\mathrm{d}}} \mid (\mathsf{x},\mathsf{y}) \notin \{\mathcal{S}^{\mathrm{u}} \cup \mathcal{S}^{\mathrm{e}}\} \,\} \,.$$

Os estados de uma composição $G_{\mathrm{a}} \| E_{\mathrm{d}}$ são separados de acordo com o nível de restrições imposto sobre as transições que partem de cada estado. Quando uma restrição é não-ambígua e.r.a. eventos de um mesmo conjunto de refinamentos (que é o caso de $\mathcal{S}^{\mathrm{u}}$ e $\mathcal{S}^{\mathrm{c}}$), a interpretação do conceito de mau-estado no procedimento de síntese segue como usual. Do contrário (caso $\mathcal{S}^{\mathrm{e}}$), a síntese pode levar a diferentes desfechos, pois um estado pode ser classificado como proibido por efeito de um evento que pode nem mesmo ser possíveis na planta refinada. Assim, ao elevar o grau de distinção da planta, eleva-se a expectativa de que mais estados possam ser mantidos no controlador final.

Exemplo 9 (Estados alvo)**.** Seja $G_{\mathrm{a}} = \Pi^{-1}(G)$ e E_{d} como na Figura 18, com $\Delta^{\beta} = \{\beta\}$, $\Delta^{\mu} = \{\mu_a, \mu_b, \mu_c\}$ e $\Delta^{\alpha} = \{\alpha\}$, para $\beta \in \Sigma_{\mathrm{c}}$ e $\mu, \alpha \in \Sigma_{\mathrm{u}}$.

Denota-se por $(\mathsf{x}_i, \mathsf{y}_j)$, para $i = 0, \cdots, 5$ e $j = 0, 1$, os estados em $Q_{G_{\mathrm{a}}} \times Q_{E_{\mathrm{d}}}$. Note que E_{d} desabilita os eventos μ_c e α no estado y_1. Portanto, os estados $(\mathsf{x}_i, \mathsf{y}_1)$ são candidatos a maus estados. Como $(\mathsf{x}_m, \mathsf{y}_1)$, para $m \in \{1, 3, 5\}$, desabilitam todo o conjunto Δ^{α}, então eles são certamente maus estados, i.e., $(\mathsf{x}_m, \mathsf{y}_1) \in \mathcal{S}^{\mathrm{u}}$.



(a) G_{a} (b) E_{d}

Figura 18: Modelos para G_{a} e E_{d}

Diferentemente, os estados $(\mathsf{x}_n, \mathsf{y}_1)$, com $n \in \{2, 4\}$, desabilitam o evento μ_c após sua primeira ocorrência, enquanto que $\Delta^{\mu} \setminus \{\mu_c\}$ estão livres para ocorrer. Como a síntese é incapaz de reconhecer qual evento em Δ^{μ} é realmente elegível, então $(\mathsf{x}_n, \mathsf{y}_1)$ podem ser maus estados para μ_c, mas são certamente seguros para os demais refinamentos em Δ^{μ}. Assim, $(\mathsf{x}_n, \mathsf{y}_1) \in \mathcal{S}^{\mathrm{e}}$. □

3.1.2 EVENTOS ALVO

A noção de *eventos alvo* pode ser apresentada com uma função dos estados alvo.

Definição 13 (Eventos Alvo)**.** *Para uma planta G, com eventos em Σ, e um distinguidor preditível D, seja G_{a} uma aproximação para $G_{\mathrm{d}} = D(G)$ e seja E_{d} um modelo de especificação para G_{a}, tal que $K_{\mathrm{a}} = L(G_{\mathrm{a}} \| E_{\mathrm{d}})$. Um evento $\mu_j \in \Delta^{\mu} \subseteq \Delta_{\mathrm{u}}$, para $\mu \in \Sigma_{\mathrm{u}}$, é dito ser* alvo *em G_{a} se ele leva um estado de $G_{\mathrm{a}} \| E_{\mathrm{d}}$ a pertencer a $\mathcal{S}^{\mathrm{e}}$, i.e., se $G_{\mathrm{a}} \xrightarrow{s} x \xrightarrow{\mu_j} x'$ e $G_{\mathrm{a}} \| E_{\mathrm{d}} \xrightarrow{s} (x, y) \overset{\mu_j}{\nrightarrow}$, e $(x, y) \in \mathcal{S}^{\mathrm{e}}$.*

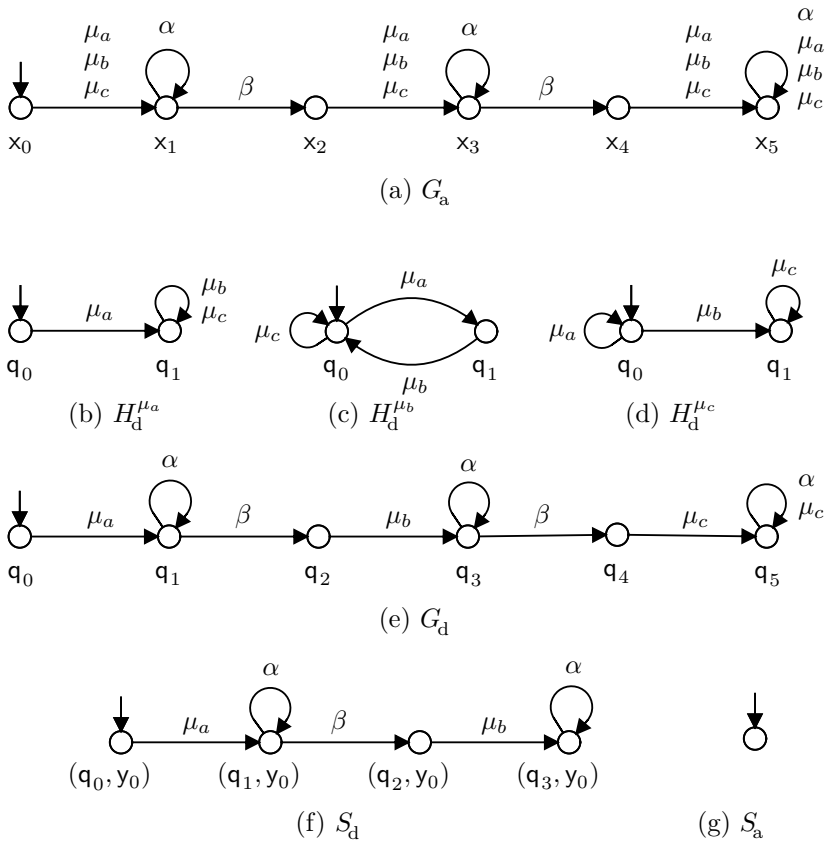
Denota-se por $\Delta_{\mathscr{U}}(G_{\mathrm{a}})$ o conjunto de todos os eventos alvos em G_{a}.

Exemplo 10 (Eventos alvo)**.** No exemplo da Figura 18, o evento μ_c é alvo em G_{a}, uma vez que esse evento leva os estados $(\mathsf{x}_n, \mathsf{y}_1)$, com $n \in \{2, 4\}$, a serem incluídos em $\mathcal{S}^{\mathrm{e}}$. De fato, $(\mathsf{x}_n, \mathsf{y}_1)$ são estados seguros para as transições rotuladas com μ_a e μ_b que partem desses estados, mas inseguros para o rótulo μ_c.

Tal ambiguidade não aconteceria no caso de G_{a} estar associado ao distinguidor preditível. Note que, ao utilizar uma planta G_{d} na síntese, o supervisor resultante é diferente de quando G_{a} é utilizado, como ilustrado na Figura 19.

Como G_{d} "reconhece" que μ_c é possível somente após μ_a e μ_b, então S_{d} evita μ_c (desabilitando β) somente quando ele seria realmente elegível. Diferentemente, G_{a} habilita μ_c já no estado inicial e um sistema sob o controle de S_{a} não inicia.

□



(a) G_{a}

(b) $H_{\mathrm{d}}^{\mu_a}$ (c) $H_{\mathrm{d}}^{\mu_b}$ (d) $H_{\mathrm{d}}^{\mu_c}$

(e) G_{d}

(f) S_{d} (g) S_{a}

Figura 19: Distinguidores $H_{\mathrm{d}}^{\mu_a}, H_{\mathrm{d}}^{\mu_b}$ e $H_{\mathrm{d}}^{\mu_c}$, plantas G_{a} e G_{d}, e supervisores S_{d} e S_{a}, com $\Delta_{S_{\mathrm{a}}} = \Delta$

3.1.3 AUTO-APROXIMAÇÃO

Quando $\Delta_{\not\mu}(G_{\mathrm{a}}) = \emptyset$, para uma dada aproximação G_{a}, é sempre possível determinar univocamente se um dado estado sobrevive ou não sob controle e nenhuma parte do distinguidor é necessária na síntese. Porém, quando $\Delta_{\not\mu}(G_{\mathrm{a}}) \neq \emptyset$, G_{a} tem que ser melhorada com alguma parte do distinguidor.

Neste trabalho, mostra-se que G_{a} pode ser melhorada de tal forma que possa distinguir univocamente somente os eventos alvo. Ou seja, G_{a} é construída de modo a se tornar preditível e.r.a. um evento em particular, ou a um conjunto de eventos, mesmo que ainda possa permanecer não preditível para outros. Essa ideia relaxa a propriedade de preditibilidade (total) da subseção 2.4, e introduz a *preditibilidade parcial* como se segue.

Definição 14 (Preditibilidade-parcial)**.** *Seja D_{a} um distinguidor (Def. 11) modelado por $L(H_{\mathrm{da}} = \|_{i=1}^{n} H_{\mathrm{da}}^{i}) = L_{\mathrm{da}}$, para $L_{\mathrm{da}} \subseteq \Delta^*$. Para $\sigma \in \Sigma$, $\delta^i \in \Delta^\sigma$, uma planta G e $G_{\mathrm{a}} = D_{\mathrm{a}}(G)$, D_{a} é dito ser δ^i-preditível se, para todo $t \in \Delta^*$,*

$$t\delta^i \in L(G_{\mathrm{a}}) \Rightarrow |t\Delta^\sigma \cap L(G_{\mathrm{a}})| = 1.$$

Ou seja, D_{a} é δ^i-preditível quando, após qualquer cadeia em Δ^*, se δ^i é elegível sob a distinção de D_{a}, ele é único entre os eventos em Δ^σ. Para qualquer conjunto $\Delta' \subseteq \Delta$, D_{a} é dito ser Δ'-preditível se é δ-preditível para todo $\delta \in \Delta'$. A mesma notação é generalizada para autômatos.

Lema 1. *Dados dois autômatos G^1 e G^2 modelados com eventos em Δ, seja G^1 δ-preditível, para $\delta \in \Delta$. Então, $G^1 \| G^2$ é δ-preditível.*

Nesse sentido, assume-se que, para qualquer $\sigma \in \Sigma$, cada evento $\delta \in \Delta^\sigma$ está implicitamente associado a um submodelo distinguidor H_{d}^{δ}, de modo que um distinguidor preditível D pode ser modelado por uma composição $H_{\mathrm{d}} = \| H_{\mathrm{d}}^{\delta_i}$, $i = 1, \cdots n$, para $L(H_{\mathrm{d}}) = L_{\mathrm{d}} \subseteq \Delta^*$.

Exemplo 11 (Preditibilidade-parcial)**.** Considere uma planta G e um distinguidor D_{a}, modelado por $L(H_{\mathrm{da}}) = L_{\mathrm{da}} = \Delta^*$, tal que $G_{\mathrm{a}} = D_{\mathrm{a}}(G)$ é como mostrado na Figura 18 (a).

Note, por exemplo, que G_{a} não é μ_c-preditível, pois após qualquer cadeia $t \in L(G_{\mathrm{a}})$, sempre que uma transição com eventos em Δ^μ é possível, $|t\Delta^\mu \cap L(G_{\mathrm{a}})| > 1$ e, portanto, essa versão de G_{a} não pode distinguir μ_c em Δ^μ. Diferentemente, G_{a} poderia ser melhorada por um distinguidor para μ_c a fim de torná-la μ_c-preditível. Considere que ${D_{\mathrm{a}}}'$ é modelado por ${L_{\mathrm{da}}}' = L_{\mathrm{da}} \| L(H_{\mathrm{d}}^{\mu_c})$, para $H_{\mathrm{d}}^{\mu_c}$ como na Figura 19 (d), tal que ${G_{\mathrm{a}}}' = G_{\mathrm{a}} \| H_{\mathrm{d}}^{\mu_c}$ é mostrada na Figura 20.

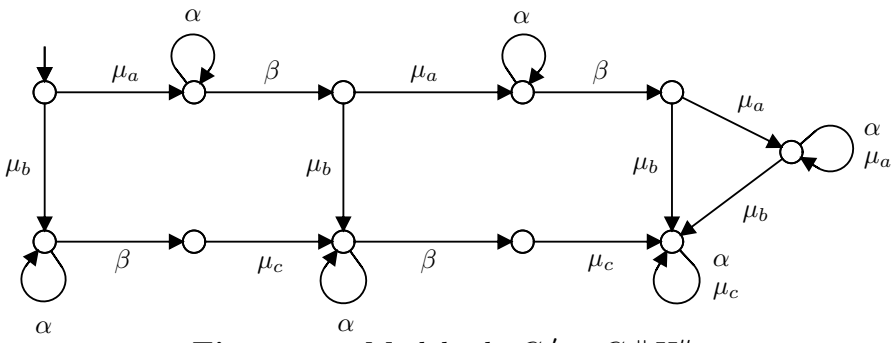


Figura 20: Modelo de ${G_{\mathrm{a}}}' = G_{\mathrm{a}} \| H_{\mathrm{d}}^{\mu_c}$

Observe que, agora após qualquer cadeia $t \in L({G_{\mathrm{a}}}')$, se μ_c é elegível (estados alcançados pelas cadeias $\mu_b\alpha^*\beta$ e $\mu_a\alpha^*\beta[\mu_b\alpha^*\beta + \mu_a\alpha^*\beta\mu_b]$), $|t\Delta^\mu \cap L({G_{\mathrm{a}}}')| = 1$. □

3.1.4 CONSTRUINDO AUTO-APROXIMAÇÕES

Intuitivamente, uma aproximação apropriada em síntese, aqui chamada de *aproximação-candidata*, é construída pela adição à planta de distinguidores que tornam os eventos alvo preditíveis, como formalizado a seguir.

Definição 15 (Aproximação-candidata)**.** *Para uma planta preditível G_{d} e uma aproximação $G_{\mathrm{a}} = \Pi^{-1}(G)$, seja $\Delta_{\mathcal{U}}(G_{\mathrm{a}})$ o conjunto de eventos alvo em G_{a} e seja H_{d}^{δ}, para todo $\delta \in \Delta$, um conjunto de distinguidores δ-preditíveis para G_{a}. Defina:*

$$\mathcal{G}_{\mathrm{c}} = G_{\mathrm{a}} \parallel [\Big\| _{\delta \in \Delta_{\mathcal{U}}(G_{\mathrm{a}})} H_{\mathrm{d}}^{\delta}]$$

como uma aproximação-candidata para G_{d}.

Como $\mathcal{G}_{\mathrm{c}}$ é $\Delta_{\mathcal{U}}(G_{\mathrm{a}})$-preditível, é intuitivo pensar em utilizá-la como auto-aproximação. No entanto, ao usar $\mathcal{G}_{\mathrm{c}}$ na síntese, pode ocorrer que eventos anteriormente não alvo tornem-se alvo indiretos. É o caso, por exemplo, de eventos não-controláveis que precederem os eventos tidos até então como alvo. Tais eventos devem ser igualmente distinguidos. Os conceitos de estado *local*, *global*, e *sub-estado* (ÅKESSON et al., 2002), são usados para identificar os eventos alvo indiretos.

Definição 16 (Estado global, local e sub-estado)**.** *Para um conjunto de autômatos $G = \{G^1, \cdots, G^n\}$, seja $G' = \{G^{1'}, \cdots, G^{n'}\}$ um subconjunto de G. Defina:*

- estado-local *como um estado $\mathsf{q}_l \in Q_{G^i}$, para um $G^i \in G$;*
- estado-global *como um estado $\mathsf{q}_g \in Q_{G^t}$, para $G^t = G^1 \| \cdots \| G^n$;*
- sub-estado *como um estado $\mathsf{q}_s \in Q_{G^p}$, para $G^p = G^{1'} \| \cdots \| G^{n'}$.*

Denote por $s\mathrm{S}(\mathsf{q}_g, G^p)$ a função que retorna um sub-estado $\mathsf{q}_s \in Q_{G^p}$ associado a um estado global $\mathsf{q}_g \in Q_{G^t}$.

Definição 17 (Eventos Indiretamente Alvos)**.** *Dada uma planta preditível G_{d} e uma especificação E_{d}, seja $G_{\mathrm{a}} = \Pi^{-1}(G)$ uma aproximação para G_{d}. Seja $\Delta_{\mathcal{U}}(G_{\mathrm{a}})$ o conjunto de eventos alvo em G_{a}, e $\mathcal{G}_{\mathrm{c}}$ uma aproximação $\Delta_{\mathcal{U}}(G_{\mathrm{a}})$-preditível como na Def. 15. Para $\delta \in \Delta_{\mathcal{U}}(G_{\mathrm{a}})$ e $t = t'c \in L(\mathcal{G}_{\mathrm{c}})$, com $t' \in \Delta^*$ e $c \in \Delta_{\mathrm{c}}$, os* Eventos Indiretamente Alvos *de $\mathcal{G}_{\mathrm{c}}$ e.r.a. $\Delta_{\mathcal{U}}(G_{\mathrm{a}})$ são todos os eventos que formam cadeias $\gamma \in {\Delta_{\mathrm{u}}}^*$, tal que*

$$\mathcal{G}_{\mathrm{c}} \xrightarrow{t} q \xrightarrow{\gamma} q' \xrightarrow{\delta} q'' \quad e \quad \mathcal{G}_{\mathrm{c}} \| E_{\mathrm{d}} \xrightarrow{t} (q, y) \xrightarrow{\gamma} (q', y') \not\xrightarrow{\delta} \quad e \quad q \neq q'$$
$$e \quad s\mathrm{S}((q', y'), G_{\mathrm{a}} \| E_{\mathrm{d}}) \in \mathcal{S}^{\mathrm{e}}.$$

Denote por $\mathcal{I}_{\mathcal{U}}(\mathcal{G}_{\mathrm{c}}, \Delta_{\mathcal{U}}(G_{\mathrm{a}}))$ a função que retorna o conjunto de eventos indiretamente alvos de $\mathcal{G}_{\mathrm{c}}$ e.r.a. $\Delta_{\mathcal{U}}(G_{\mathrm{a}})$, denotado por $\Delta_{i\mathcal{U}}(\mathcal{G}_{\mathrm{c}})$. Assim o seguinte resultado pode ser apresentado.

Lema 2. *Para uma dada aproximação G_{a} e uma especificação E_{d}, sejam $\Delta_{\not u}(G_{\mathrm{a}})$ e $\Delta_{\not i\not u}(G_{\mathrm{a}})$ respectivamente os conjuntos de eventos alvo e indiretamente alvo de G_{a}. Pode ser mostrado que, se $\Delta_{\not u}(G_{\mathrm{a}}) = \emptyset$, então $\Delta_{\not i\not u}(G_{\mathrm{a}}) = \emptyset$.*

Demonstração. Como $\Delta_{\not u}(G_{\mathrm{a}}) = \emptyset$, então $\mathcal{S}^{\mathrm{e}} = \emptyset$, pela Def. 13, o que implica que, para qualquer estado global q_g em $G_{\mathrm{a}} \| E_{\mathrm{d}}$, $s\mathrm{S}(\mathsf{q}_g, G_{\mathrm{a}} \| E_{\mathrm{d}}) \notin \mathcal{S}^{\mathrm{e}}$. □

Agora, utiliza-se esse raciocínio para mostrar que, se $\Delta_{\not u}(G_{\mathrm{a}})$ é vazio, então G_{a} e G_{d} levam ao supervisor controlável menos restritivo.

Teorema 1. *Dada uma planta preditível G_{d} e uma especificação E_{d}, com $K_{\mathrm{d}} = L(G_{\mathrm{d}} \| E_{\mathrm{d}})$, seja G_{a} uma aproximação para G_{d}, tal que $K_{\mathrm{a}} = L(G_{\mathrm{a}} \| E_{\mathrm{d}})$. Para os conjuntos $\Delta_{\not u}(G_{\mathrm{a}})$ e $\Delta_{\not i\not u}(G_{\mathrm{a}})$ de eventos alvo e indiretamente alvo em G_{a} (definições 13 e 17), se $\Delta_{\not u}(G_{\mathrm{a}}) \cup \Delta_{\not i\not u}(G_{\mathrm{a}}) = \emptyset$, então $\sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}}) \cap L_{\mathrm{d}} = \sup\mathcal{C}(K_{\mathrm{d}}, G_{\mathrm{d}})$.*

Demonstração. Para mostrar $\sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}}) \cap L_{\mathrm{d}} \subseteq \sup\mathcal{C}(K_{\mathrm{d}}, G_{\mathrm{d}})$, note que $L(G_{\mathrm{d}}) \subseteq L(G_{\mathrm{a}})$ por definição. Portanto, a controlabilidade de $\sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}})$ e.r.a. $L(G_{\mathrm{a}})$ implica na controlabilidade de $\sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}})$ e.r.a. $L(G_{\mathrm{d}})$. Além disso, a controlabilidade de L_{d} e.r.a. $L(G_{\mathrm{d}})$ segue de $L(G_{\mathrm{d}}) \subseteq L_{\mathrm{d}}$, e então $\sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}}) \cap L_{\mathrm{d}}$ é controlável e.r.a. $L(G_{\mathrm{d}})$. Assim, $\sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}}) \cap L_{\mathrm{d}} \subseteq K_{\mathrm{a}} \cap L_{\mathrm{d}} = K_{\mathrm{d}}$ implica

$$\sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}}) \cap L_{\mathrm{d}} \subseteq \sup\mathcal{C}(K_{\mathrm{d}}, G_{\mathrm{d}}). \tag{1}$$

Para mostrar $\sup\mathcal{C}(K_{\mathrm{d}}, G_{\mathrm{d}}) \subseteq \sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}}) \cap L_{\mathrm{d}}$, seja $s\mu_i \in \sup\mathcal{C}(K_{\mathrm{d}}, G_{\mathrm{d}})$ para $\mu_i \in \Delta^{\mu}$ e $\mu \in \Sigma_{\mathrm{u}}$. Como $\sup\mathcal{C}(K_{\mathrm{d}}, G_{\mathrm{d}})$ é controlável e.r.a. $L(G_{\mathrm{d}})$, então

$$G_{\mathrm{d}} \xrightarrow{s} \mathsf{q} \xrightarrow{\mu_i} \mathsf{q}' \qquad \text{e} \qquad G_{\mathrm{d}} \| E_{\mathrm{d}} \xrightarrow{s} (\mathsf{q}, \mathsf{y}) \xrightarrow{\mu_i} (\mathsf{q}, \mathsf{y}').$$

Pelo pressuposto $\Delta_{\not u}(G_{\mathrm{a}}) = \emptyset$, tem-se que para qualquer $\mu_j \in \Delta^{\mu}$, tal que

$$G_{\mathrm{a}} \xrightarrow{s} \mathsf{x} \xrightarrow{\mu_i, \mu_j} \mathsf{x}',$$

é verdade que

$$G_{\mathrm{a}} \| E_{\mathrm{d}} \xrightarrow{s} (\mathsf{x}, \mathsf{y}) \xrightarrow{\mu_i, \mu_j} (\mathsf{x}', \mathsf{y}')$$

pois, do contrário, a combinação de estados (x, y) pertencem a $\mathcal{S}^{\mathrm{e}}$ ou $\mathcal{S}^{\mathrm{u}}$, e $\Delta_{\not u}(G_{\mathrm{a}}) \neq \emptyset$. Agora, para $s = t\gamma$, tal que $t \in \Delta^*$ e $\gamma \in {\Delta_{\mathrm{u}}}^*$, $t\gamma \in L(G_{\mathrm{a}})$ e

$$G_{\mathrm{a}} \xrightarrow{t} \mathsf{x} \xrightarrow{\gamma} \mathsf{x}' \xrightarrow{\mu_i, \mu_j} \mathsf{x}'',$$

implica

$$G_{\mathrm{a}} \| E_{\mathrm{d}} \xrightarrow{t} (\mathsf{x}, \mathsf{y}) \xrightarrow{\gamma} (\mathsf{x}', \mathsf{y}') \xrightarrow{\mu_i, \mu_j} (\mathsf{x}'', \mathsf{y}''),$$

visto que $\Delta_{i\acute{u}}(G_{\mathrm{a}}) = \emptyset$, por suposição.

Logo, $s\mu_i \in \mathrm{sup}\mathcal{C}(K_{\mathrm{d}}, G_{\mathrm{d}})$ e $\Delta_{\acute{u}}(G_{\mathrm{a}}) = \emptyset$ implicam $s\mu_i \in \mathrm{sup}\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}})$. Como ainda $s\mu_i \in L(G_{\mathrm{d}}) \subseteq L_{\mathrm{d}}$, então $s\mu_i \in \mathrm{sup}\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}}) \cap L_{\mathrm{d}}$. □

Do Teorema 1, resta mostrar como se pode construir uma *auto-aproximação* $\mathcal{G}_{\mathrm{c}}$ que sempre leve aos conjuntos vazio de eventos alvo e indiretamente alvo, i.e, $\Delta_{\acute{u}}(\mathcal{G}_{\mathrm{c}}) = \Delta_{i\acute{u}}(\mathcal{G}_{\mathrm{c}}) = \emptyset$. O Algoritmo 1 sistematiza essa construção.

Algoritmo 1 Computação de $\mathcal{G}_{\mathrm{c}}$

Entrada: planta $G_{\mathrm{a}} = \Pi^{-1}(G)$; especificação E_{d}; eventos Σ, eventos Δ; $\mu \in \Sigma_{\mathrm{u}}$; conjunto de distinguidores δ-preditível H_{d}^{δ}, para todo $\delta \in \Delta$, tal que $H_{\mathrm{d}} = \| H_{\mathrm{d}}^{\delta}$ é preditível;

1: $\mathcal{G}_{\mathrm{c}} \leftarrow G_{\mathrm{a}}$; $\Delta_{\acute{u}}(G_{\mathrm{a}}) \leftarrow \emptyset$; $\Delta_{i\acute{u}}(\mathcal{G}_{\mathrm{c}}) \leftarrow \emptyset$;
2: **para todo** $(x, y) \in Q_{G_{\mathrm{a}}} \times Q_{E_{\mathrm{d}}}$ **faça**
3: **se** $(\mathsf{x}, \mathsf{y}) \in \mathcal{S}^{\mathrm{e}}$ **então**
4: $\Delta_{\acute{u}}(G_{\mathrm{a}}) \leftarrow \Delta_{\acute{u}}(G_{\mathrm{a}}) \cup \{\mu_j \in \Delta_{\mathrm{u}} \mid \mu_j \text{ é alvo}\}$;
5: **fim para**
6: **para todo** $\delta \in \Delta_{\acute{u}}(G_{\mathrm{a}})$ **faça**
7: $\mathcal{G}_{\mathrm{c}} \leftarrow \mathcal{G}_{\mathrm{c}} \parallel H_{\mathrm{d}}^{\delta}$;
8: **fim para**
9: $\Delta_{i\acute{u}}(\mathcal{G}_{\mathrm{c}}) \leftarrow \mathcal{I}_{\acute{u}}(\mathcal{G}_{\mathrm{c}}, \Delta_{\acute{u}}(G_{\mathrm{a}}))$;
10: **para todo** $\alpha \in \Delta_{i\acute{u}}(\mathcal{G}_{\mathrm{c}})$ **faça**
11: $\mathcal{G}_{\mathrm{c}} \leftarrow \mathcal{G}_{\mathrm{c}} \parallel H_{\mathrm{d}}^{\alpha}$;
12: **fim para**
13: **retornar** $\mathcal{G}_{\mathrm{c}}$;

Inicialmente, $\mathcal{G}_{\mathrm{c}}$ é considerado como $G_{\mathrm{a}} = \Pi^{-1}(G)$ (linha 1). Computacionalmente, esse é o melhor caso, pois nenhum distinguidor é incluído na síntese ($L_{\mathrm{da}} = \Delta^*$) e verifica-se se o produto é o supervisor controlável e menos restritivo.

Para isso, visita-se cada estado de $\mathcal{G}_{\mathrm{c}}$ verificando se é *eventualmente controlável* ($\mathcal{S}^{\mathrm{e}}$ na linha 3). Se não há estados em $\mathcal{S}^{\mathrm{e}}$, consequentemente não existem eventos alvo em G_{a} (pela Def. 13). Então, pelo Teorema 1, a síntese pode ser conduzida utilizando a planta $\mathcal{G}_{\mathrm{c}} = G_{\mathrm{a}} = \Pi^{-1}(G)$.

Diferentemente, a existência de um estado em $\mathcal{S}^{\mathrm{e}}$ sugere a reconstrução de $\mathcal{G}_{\mathrm{c}}$, de modo a melhorar sistematicamente seu grau de distinção por meio da composição de distinguidores específicos (linhas 7 e 11). Tais modelos são selecionados de acordo com: (i) eventos alvo (linha 4); (ii) eventos indiretamente alvo (linha 9). Então, o seguinte resultado pode ser anunciado.

Lema 3. *Seja uma planta G, uma aproximação $G_{\mathrm{a}} = \Pi^{-1}(G)$ e um conjunto de eventos alvo em G_{a} ($\Delta_{\text{alvo}}(G_{\mathrm{a}})$). Ao término do Algoritmo 1, $\mathcal{G}_{\mathrm{c}}$ é $(\Delta_{\text{alvo}}(G_{\mathrm{a}}) \cup \Delta_{\text{ind}}(\mathcal{G}_{\mathrm{c}}))$-preditível.*

Demonstração. Se segue do Lema 1, associado aos distinguidores δ-preditíveis parametrizados como entrada do Algoritmo 1. □

Exemplo 12 (Aproximação-alvo)**.** No exemplo da Figura 18, o evento μ_c é alvo em $G_{\mathrm{a}} = \Pi^{-1}(G)$, pela Def. 13. Assim, G_{a} é composta à $H_{\mathrm{d}}^{\mu_c}$ para formar a auto-aproximação $\mathcal{G}_{\mathrm{c}}$ da Figura 20. O conjunto de eventos indiretamente alvo para $\mathcal{G}_{\mathrm{c}}$ é $\Delta_{\text{ind}}(\mathcal{G}_{\mathrm{c}}) = \emptyset$, visto que o único estado $\mathsf{q}_i \in Q_{\mathcal{G}_{\mathrm{c}} \| E_{\mathrm{d}}}$, tal que $s\mathrm{S}(\mathsf{q}_i, G_{\mathrm{a}} \| E_{\mathrm{d}}) \in \mathcal{S}^{\mathrm{e}}$, é alcançado pela cadeia $\mu_b \alpha^* \beta \mu_c \beta$, de modo que o único evento que será desabilitado para evitar esse mau estado é $\beta \in \Delta_{\mathrm{c}}$.

Diferentemente de G_{a} (Figura 18 (a)), $\mathcal{G}_{\mathrm{c}}$ (Figura 20) somente permite μ_c após μ_b, exatamente do mesmo modo que em G_{d} da Figura 19 (e). Como E_{d} implementa uma ação de controle sob o evento μ_c, então a planta $\mathcal{G}_{\mathrm{c}}$ sabe exatamente quando μ_c pode realmente ocorrer, independente do caminho que o levaram a ocorrer. Dessa forma, se um evento $\sigma_i \in \Delta_{\mathrm{u}}$ é proibido por E_{d} em um dado estado $\mathsf{q} \in Q_{\mathcal{G}_{\mathrm{c}} \| E_{\mathrm{d}}}$, q é classificado como $\mathcal{S}^{\mathrm{u}}$ pela Def. 12, como ilustra a Figura 21.

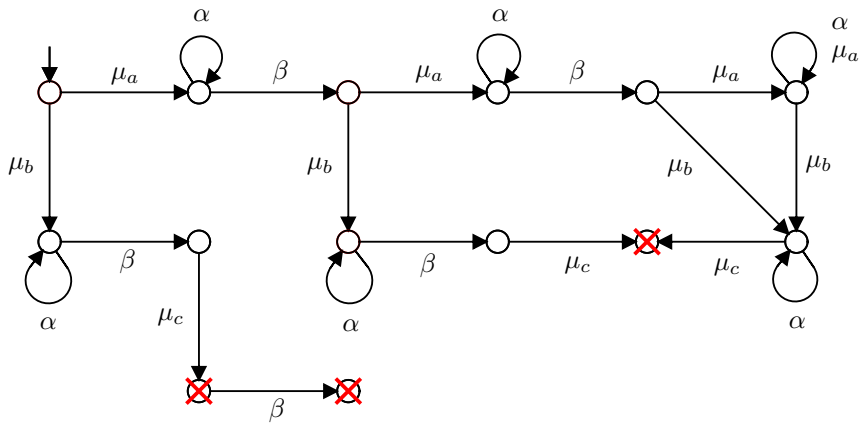


Figura 21: Modelo para $\mathcal{G}_{\mathrm{c}} \parallel E_{\mathrm{d}}$

□

Observe que, do Algoritmo 1, a inclusão

$$L(G_{\mathrm{d}}) \subseteq L(\mathcal{G}_{\mathrm{c}}) \subseteq L(G_{\mathrm{a}}) \subseteq \Delta^* \tag{2}$$

é verdadeira, i.e., $\mathcal{G}_{\mathrm{c}}$ é simplesmente uma aproximação particular que trata dos eventos alvo e indiretamente alvo. Então, a seguinte propriedade pode ser apresentada para $\mathcal{G}_{\mathrm{c}}$.

Lema 4. *Para uma planta G e uma aproximação $G_{\mathrm{a}} = \Pi^{-1}(G)$, seja $\mathcal{G}_{\mathrm{c}}$ uma aproximação computada como no Algoritmo 1. Se $\mathcal{G}_{\mathrm{c}}$ é $\Delta_{\text{alvo}}(G_{\mathrm{a}})$-preditível, então $\Delta_{\text{alvo}}(\mathcal{G}_{\mathrm{c}}) = \emptyset$.*

Demonstração. Pelo Lema 3, $\mathcal{G}_{\mathrm{c}}$ é $\Delta_{u}(G_{\mathrm{a}})$-preditível. Então para qualquer $\mu_i \in \Delta_{u}(G_{\mathrm{a}})$ e $s \in \Delta^*$, tal que

$$\mathcal{G}_{\mathrm{c}} \xrightarrow{s} \mathsf{q} \xrightarrow{\mu_i} \mathsf{q}' \qquad \text{e} \qquad \mathcal{G}_{\mathrm{c}} \| E_{\mathrm{d}} \xrightarrow{s} (\mathsf{q}, \mathsf{y}) \xrightarrow{\mu_i},$$

se segue que $|\Gamma(q) \cap \Delta^{\mu}| = 1$, pela Def. 14. Então, pela Def. 12, a combinação de estados $(q, \mathsf{y}) \in \mathcal{S}^{\mathrm{u}}$ e, generalizando, $\Delta_{u}(\mathcal{G}_{\mathrm{c}}) = \emptyset$, pela Def. 13. □

Finalmente, pode-se mostrar que uma *auto-aproximação* construída como no Algoritmo 1 sempre leva a síntese controlável menos restritiva.

Teorema 2. *Para uma planta G, uma especificação E, e um distinguidor preditível D, seja $G_{\mathrm{d}} = D(G)$ e E_{d} a versão de E modelada em Δ, com $K_{\mathrm{d}} = L(G_{\mathrm{d}} \| E_{\mathrm{d}})$. Para $\mathcal{G}_{\mathrm{c}}$ calculado como no Algoritmo 1, tal que $\mathcal{K}_{\mathrm{c}} = L(\mathcal{G}_{\mathrm{c}} \| E_{\mathrm{d}})$, é verdade que*

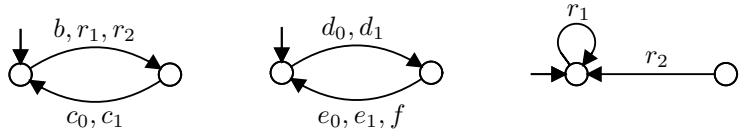
$$\sup\mathcal{C}(\mathcal{K}_{\mathrm{c}}, \mathcal{G}_{\mathrm{c}}) \cap L_{\mathrm{d}} = \sup\mathcal{C}(K_{\mathrm{d}}, G_{\mathrm{d}}).$$

Demonstração. Dos lemas 3 e 4, tem-se que $\Delta_{u}(\mathcal{G}_{\mathrm{c}}) = \emptyset$. Então, pelo Lema 2, $\Delta_{u}(\mathcal{G}_{\mathrm{c}}) = \emptyset$ e a prova para o Teorema 2 segue diretamente do Teorema 1. □

Note que, caso a construção de $\mathcal{G}_{\mathrm{c}}$ envolva todo o modelo do distinguidor, a abordagem pode realizar uma operação desnecessária ao tentar construir uma aproximação que não existe a priori. No entanto, esse caso é atípico na prática, pois é improvável que todos os eventos não-controláveis de um sistema sejam classificados como alvo ou indiretamente alvo.

3.1.5 ESTUDO DE CASO COM AUTO-APROXIMAÇÕES

Considere agora resolver o exemplo da subseção 2.5.1 usando uma auto-aproximação $\mathcal{G}_{\mathrm{c}}$ para G_{d}. Para isso, utiliza-se o Algoritmo 1 para classificar os estados de $G_{\mathrm{a}} \| E_{\mathrm{d}}$, com $G_{\mathrm{a}} = \Pi^{-1}(G)$, e então construir a auto-aproximação. Os autômatos que implementam as especificações de controle em Δ para esse exemplo são reproduzidos na Figura 22.



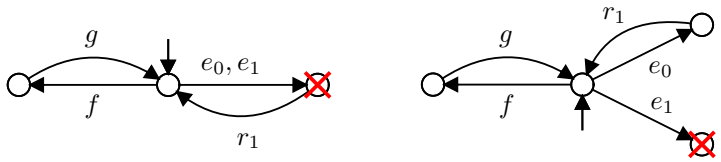
(a) $E_{\mathrm{d}}^1 = \Pi^{-1}(E^1)$ (b) $E_{\mathrm{d}}^2 = \Pi^{-1}(E^2)$ (c) E_{d}^3

Figura 22: Modelos para $E_{\mathrm{d}}^1, E_{\mathrm{d}}^2$ e E_{d}^3

Inicialmente, note que $E_{\mathrm{d}}^1 = \Pi^1(E^1)$, $E_{\mathrm{d}}^2 = \Pi^1(E^2)$ e E_{d}^3 da Figura 22 desabilitam eventos não-controláveis (Δ^b, Δ^d e Δ^r). Então é possível que existam maus estados em $G_{\mathrm{a}} \| E_{\mathrm{d}}$. Porém, as especificações E_{d}^1 e E_{d}^2, ao desabilitar um refinamento não-controlável, desabilitam todo o conjunto de eventos com a mesma máscara. Logo, os estados a partir dos quais esses eventos partem são incluídos em $\mathcal{S}^{\mathrm{u}}$ pela Def. 12.

Diferentemente, as combinações de estados para E_{d}^3 são classificadas como eventualmente controlável ($\mathcal{S}^{\mathrm{e}}$ na linha 3), uma vez que, E_{d}^3 desabilita o evento r_2 enquanto r_1 está habilitado. Então, pela Def. 13, o Algoritmo 1 identifica r_2 como alvo em G_{a} (linha 4).

Para fins de ilustração, considere a composição $G_{\mathrm{a}}^3 \| E_{\mathrm{d}}^3$ na Figura 23 (a). Após Δ^e, o estado alcançado é identificado como não-controlável e.r.a. r_2. No entanto, o mesmo estado pode ser visto como controlável e.r.a. r_1. Portanto, como a síntese com G_{a} não distingue r_2 em Δ^r, esse estado é removido, o que significa que o supervisor resultante irá restringir qualquer retrabalho de peças.



(a) $G_{\mathrm{a}}^3 \| E_{\mathrm{d}}^3$ (b) $G_{\mathrm{a}}^3 \| E_{\mathrm{d}}^3 \| H_{\mathrm{d}}^r$

Figura 23: Modelos para $G_{\mathrm{a}}^3 \| E_{\mathrm{d}}^3$ e $G_{\mathrm{a}}^3 \| E_{\mathrm{d}}^3 \| H_{\mathrm{d}}^r$

A fim de melhorar o supervisor, reconstrói-se G_{a} (linha 7) por meio da composição com H_{d}^r da Figura 17 (d), resultando em uma aproximação $\mathcal{G}_{\mathrm{c}}$ que é $\Delta_{\mathcal{U}}(G_{\mathrm{a}})$-preditível (pelo Lema 3). A escolha por H_{d}^r é baseada na identificação do conjunto de eventos alvo $\Delta_{\mathcal{U}}(G_{\mathrm{a}}) = \{r_2\}$ do Algoritmo 1.

Uma comparação entre G_{a} e sua versão $\Delta_{\mathcal{U}}(G_{\mathrm{a}})$-preditível pode ser vista respectivamente nas figuras 23 (a) e 23 (b). Observe que, em 23 (b), o estado alcançado por e_0 é seguro, diferentemente do caso da Figura 23 (a). Isso significa que cada nova peça poderá ser retrabalhada no máximo uma vez.

A função $\mathcal{I}_{\mathcal{U}}(\mathcal{G}_{\mathrm{c}}, \Delta_{\mathcal{U}}(G_{\mathrm{a}}))$ retorna $\Delta_{\mathcal{U}\cap}(\mathcal{G}_{\mathrm{c}}) = \{b\}$ (linha 9). Como o evento b não possui refinamento em Δ então, por construção, G_{a} é $\{b\}$-preditível. Assim, a auto-aproximação para G_{d} é simplesmente $\mathcal{G}_{\mathrm{c}} = G_{\mathrm{a}} \| H_{\mathrm{d}}^r$.

Agora, para $\mathcal{G}_{\mathrm{c}} = G_{\mathrm{a}} \| H_{\mathrm{d}}^r$ e $\mathcal{K}_{\mathrm{c}} = L(\mathcal{G}_{\mathrm{c}} \| E_{\mathrm{d}})$, sintetiza-se um supervisor $\sup\mathcal{C}(\mathcal{K}_{\mathrm{c}}, \mathcal{G}_{\mathrm{c}})$, modelado por V_{a}' na terceira linha da Tabela 3. As três primeiras linhas reproduzem a

Tabela 2.

Tabela 3: Síntese - estados (transições)

Sistema de manufatura com 2 retrabalhos				
G	E	G^K		V
12 (40)	34 (96)	192 (364)		26 (45)
G_{d}	E_{d}	$G^{K_{\mathrm{d}}}$		V_{d}
84 (276)	4 (18)	192 (364)		26 (45)
G_{a}	E_{d}	$G^{K_{\mathrm{a}}}$	V_{a}	$V_{\mathrm{a}} \| H_{\mathrm{d}}$
12 (60)	4 (18)	48 (132)	18 (40)	18 (32)
$\mathcal{G}_{\mathrm{c}}$	E_{d}	$G^{\mathcal{K}_{\mathrm{c}}}$	V_{a}'	$V_{\mathrm{a}}' \| H_{\mathrm{d}}$
16 (72)	4 (18)	64 (160)	19 (42)	26 (45)

Observe que, a auto-aproximação $\mathcal{G}_{\mathrm{c}}$ com 16 estados é mais simples que G_{d}, com 84 estados. No entanto, agora a versão parcial do controlador V_{a}', obtido de $G^{\mathcal{K}_{\mathrm{c}}}$ com 64 estados, leva a solução controlável e menos restritiva, i.e, $V_{\mathrm{a}}' \| H_{\mathrm{d}} = V_{\mathrm{d}}$. Além disso, $G^{\mathcal{K}_{\mathrm{c}}}$ com 64 estados é mais simples que $G^{K_{\mathrm{d}}}$ com 192 estados, e essa diferença pode ser substancialmente mais acentuada para casos com maior quantidade de retrabalhos, como ilustrado na Tabela 4.

Tabela 4: Síntese - estados (transições)

Sistema de manufatura com 5 retrabalhos				
${G_{\mathrm{d}}}^5$	${E_{\mathrm{d}}}^5$	${G^{K_{\mathrm{d}}}}^5$		${V_{\mathrm{d}}}^5$
1140 (3156)	4 (42)	1824 (3172)		50 (81)
${\mathcal{G}_{\mathrm{c}}}^5$	${E_{\mathrm{d}}}^5$	${G^{\mathcal{K}_{\mathrm{c}}}}^5$	${V_{\mathrm{a}}'}^5$	${V_{\mathrm{a}}'}^5 \| {H_{\mathrm{d}}}^5$
16 (144)	4 (42)	64 (304)	19 (72)	50 (81)

Como pode ser observado ${G^{\mathcal{K}_{\mathrm{c}}}}^5$ continua com 64 estados, enquanto que o número de estados de ${G^{K_{\mathrm{d}}}}^5$ é incrementado para 1824 estados. Além disso, note que, embora a síntese seja beneficiada via auto-aproximações, em ambos os casos com 2 e 5 retrabalhos, a versão implementável do controlador (última coluna) acaba sendo a mesma que o caso com distinguidores. Assim, o esforço para implementar computacionalmente a solução de controle com refinamentos de eventos é o mesmo que o caso obtido sem refinamentos. Nesse sentido, no próximo capítulo dessa dissertação propõe-se uma arquitetura de implementação de controladores com refinamento de eventos que aproveita os ganhos obtidos nas etapas que antecedem a implementação, e este consistirá no segundo resultado obtido nesta dissertação (ROSA et al., 2017).

4 IMPLEMENTAÇÃO DECENTRALIZADA DE CONTROLE

Na literatura, o uso de aproximações permite estender as vantagens de modelagem, trazidas por um distinguidor, também para a etapa de síntese. Nesta dissertação, no capítulo 3, foi mostrado que é possível sistematizar a construção de uma aproximação que sempre leva ao controlador ótimo.

Neste capítulo, será mostrado que, adicionalmente, é possível ainda estender tais vantagens para a etapa de implementação física do sistema de controle. Note que, independentemente das vantagens de modelagem e de síntese, a versão parcial do controlador obtido via aproximações precisa ser complementada/composta com o modelo que distingue a planta, para fins de implementação em hardware. Assim, embora o processo de obtenção do controlador possa ter sido simplificado decisivamente, a etapa de implementação manipula exatamente a mesma estrutura de um caso em que o controlador foi obtido utilizando todo o distinguidor, ou nenhum afinal.

Nesse sentido, será mostrado que é possível explorar a característica descentralizada do controlador em relação ao distinguidor, para viabilizar a implementação de um sistema de controle comunicante, que se beneficia de uma arquitetura descentralizada para executar efetivamente as ações sobre a planta.

4.1 ARQUITETURA PROPOSTA

No contexto da arquitetura proposta, controlador e distinguidor são implementados em estruturas separadas e se correspondem sistematicamente por trocas de mensagens. Na prática isso reduz o esforço de implementação, no sentido de consumo de *hardware*, ao passo que se obtém o mesmo efeito de um caso em que a implementação explora um modelo composto. Formalmente, essa ideia pode ser expressa como se segue.

Seja H_{d} um modelo para um distinguidor preditível D, tal que $L(H_{\mathrm{d}}) = L_{\mathrm{d}}$ e H_{da} um modelo para um distinguidor D_{a}, tal que $L(H_{\mathrm{da}}) = L_{\mathrm{da}}$, com $L_{\mathrm{d}} \subseteq L_{\mathrm{da}}$. Seja também S_{a} um modelo para um supervisor $\sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}})$ obtido utilizando uma apro-

ximação $G_{\mathrm{a}} = D_{\mathrm{a}}(G)$ para $G_{\mathrm{d}} = D(G)$. A fim de implementar o comportamento do controlador $\sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}}) \cap L_{\mathrm{d}}$ de forma descentralizada, i.e., S_{a} e L_{d} separados, propõe-se a arquitetura apresentada na Figura 24.

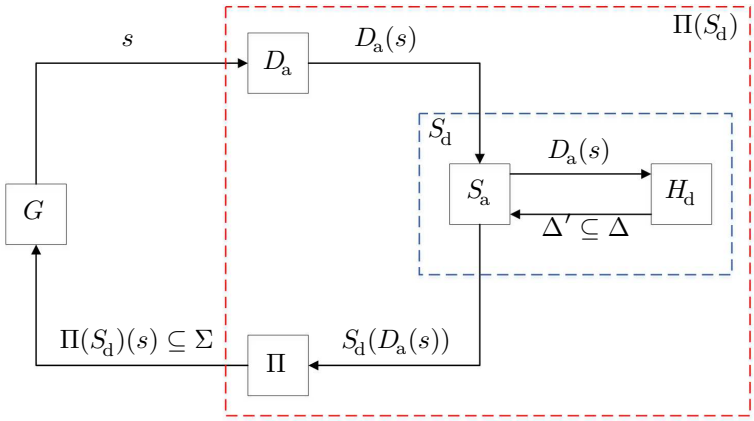


Figura 24: Implementação descentralizada de controladores com eventos refinados

Observe que, após uma cadeia $s \in L(G)$, a planta G espera o sistema de controle S_{d} responder com um conjunto de eventos habilitados, de modo que a planta possa se comportar de acordo com a ação de controle. Como o sistema de controle é tratado em Δ, e o modelo da planta reconhece eventos em Σ, então a planta espera receber uma resposta de $\Pi(S_{\mathrm{d}})$, em que $L(S_{\mathrm{d}}) = \sup\mathcal{C}(K_{\mathrm{d}}, G_{\mathrm{d}})$.

A questão agora é como reproduzir tal resposta para G, considerando que não se tem realmente o controlador S_{d}, e não se tem a intenção de implementar a composição equivalente a $\sup\mathcal{C}(K_{\mathrm{a}}, G_{\mathrm{a}}) \cap L_{\mathrm{d}}$, i.e., $S_{\mathrm{a}} \| H_{\mathrm{d}}$. Essa questão pode ser respondida pela comunicação entre as duas estruturas concorrentes, S_{a} e H_{d}.

A comunicação entre S_{a} e H_{d} é dirigida pelo grau de certeza que S_{a} possui sobre suas ações de controle, i.e., toda vez que S_{a} não for capaz de deliberar sobre a elegibilidade de um refinamento, ele irá requisitar H_{d}. Tome como base os seguintes exemplos:

Exemplo 13. Para $\Delta^{\alpha} = \{\alpha_1, \alpha_2\}$, $\Delta^{\beta} = \{\beta\}$ e $s \in L(G)$, após $D_{\mathrm{a}}(s)$ considere os seguintes casos em S_{a}:

(i) $S_{\mathrm{a}} \xrightarrow{D_{\mathrm{a}}(s)} \mathsf{q} \xrightarrow{\beta} \mathsf{q}'$: nesse caso S_{a} não precisa requisitar H_{d} para deliberar sobre a habilitação de β, uma vez que $|\Delta^{\beta}| = 1$;

(ii) $S_{\mathrm{a}} \xrightarrow{D_{\mathrm{a}}(s)} \mathsf{q} \xrightarrow{\alpha_1} \mathsf{q}'$ e $S_{\mathrm{a}} \xrightarrow{D_{\mathrm{a}}(s)} \mathsf{q} \xrightarrow{\alpha_2} \mathsf{q}''$: diferentemente, nesse caso S_{a} tem que requisitar H_{d}, pois S_{a} não distingue α_1 de α_2;

(iii) $S_{\mathrm{a}} \xrightarrow{D_{\mathrm{a}}(s)} \mathsf{q} \xrightarrow{\alpha_1} \mathsf{q}'$ e $S_{\mathrm{a}} \xrightarrow{D_{\mathrm{a}}(s)} \mathsf{q} \xrightarrow{\alpha_2} \not\rightarrow$: esse caso poderia levar à uma situação de inconsistência quanto a habilitação da máscara α em G caso não se consulte H_{d}, pois

para α_1 o evento α seria habilitado, enquanto que por α_2 não. Então, considerando que

$$H_{\mathrm{d}} \xrightarrow{D(D_{\mathrm{a}}(s))} \mathsf{q} \xrightarrow{\alpha_2} \mathsf{q}',$$

S_{a} levaria a uma ação de controle diferente de $S_{\mathrm{a}} \| H_{\mathrm{d}}$.

□

A caixa interna tracejada da Figura 24 implementa a interação entre S_{a} e H_{d}, e será detalhada na Seção 4.2. Essencialmente, ela reproduz o mesmo efeito que S_{d} devido à troca de informação entre S_{a} e H_{d}, de modo que a cada requisição do controlador S_{a}, o distinguidor H_{d} responde com um conjunto de eventos Δ' contendo escolhas preditíveis para cada evento habilitado por S_{a}, tal que $S_{\mathrm{d}}(D_{\mathrm{a}}(s)) = S_{\mathrm{a}}(D_{\mathrm{a}}(s)) \cap \Delta'$.

4.2 INTEGRANDO S_{a} E L_{d}

Nesta seção será apresentado como S_{a} e L_{d} podem ser integrados de tal modo que suas ações disjuntas levem ao mesmo comportamento que sua composição ($S_{\mathrm{a}} \| H_{\mathrm{d}}$). Dessa forma, espera-se que eles possam ser implementados separados. Os Algoritmos 2 e 3 sintetizam esta ideia.

Algoritmo 2 Implementação de S_{a}

Entrada: $\sigma \in \Sigma$; $s \in \Sigma^*$; distinguidor D_{a}; supervisor S_{a};
1: $\Delta' \leftarrow \emptyset$;
2: **para todo** $\sigma \in L(G)(s)$ **faça**
3: **se** $|S_{\mathrm{a}}(D_{\mathrm{a}}(s)) \cap \Delta^{\sigma}| \geq 1$ **então**
4: **se** $|\Delta^{\sigma}| > 1$ **então**
5: $\Delta' \leftarrow \Delta' \cup \{\mathrm{REQUEST}L_{\mathrm{d}}(D_{\mathrm{a}}(s)) \cap \Delta^{\sigma}\}$;
6: **senão**
7: $\Delta' \leftarrow \Delta' \cup \Delta^{\sigma}$;
8: **retornar** $S_{\mathrm{a}}(D_{\mathrm{a}}(s)) \cap \Delta'$;

Algoritmo 3 Implementação de H_{d}

Entrada: $L_{\mathrm{d}} = L(H_{\mathrm{d}})$; $D_{\mathrm{a}}(s) \subseteq \Delta^*$;
1: $\Delta_d \leftarrow \emptyset$;
2: **função** $\mathrm{REQUEST}L_{\mathrm{d}}((D_{\mathrm{a}}(s))$
3: $\Delta_d \leftarrow H_{\mathrm{d}}(D_{\mathrm{a}}(s)) \subseteq \Delta$;
4: **retornar** Δ_d;

Após cada cadeia $s \in L(G)$, o controlador S_{a} verifica se os refinamentos elegíveis para cada evento $\sigma \in \Sigma$, habilitados em $S_{\mathrm{a}}(D_{\mathrm{a}}(s))$, podem levar à uma situação de ambiguidade e.r.a. Δ^{σ} (linha 4). Caso exista ambiguidade ($|\Delta^{\sigma}| > 1$), então H_{d} é requisitado

(linha 5) por meio da função REQUESTL_{d} (Algoritmo 3), a fim de distinguir qual a instância exata $\sigma_i \in \Delta^{\sigma}$ deve ser habilitada ou não.

Caso contrário, i.e., quando S_{a} pode deliberar inequívocamente quanto a elegibilidade de um evento ($|\Delta^{\sigma}| = 1$) H_{d} não precisa ser consultado (linha 7). Assim, com conjunto de distinções preditíveis Δ', S_{a} pode decidir inequivocamente quanto a elegibilidade de um refinamento (linha 8).

Observe que, embora H_{d} esteja separado de S_{a}, eles ainda executam concorrentemente, com H_{d} a ser consultado sempre que S_{a} necessitar. Isso sugere que H_{d} e S_{a} poderiam ser implementados em hardwares diferentes e ainda desempenhariam a mesma função para o sistema de controle. Assim, o efeito da composição entre S_{a} e H_{d} seria produzido em tempo de execução a partir da troca de informações entre essas estruturas.

No entanto, vale lembrar que essa abordagem inevitavelmente impõe uma latência na ação de controle. A fim de estimar o *overhead* adicionado na malha de controle, e para inferir se isso deve ou não ser uma preocupação, foram realizados testes para dimensionar o tempo da comunicação.

4.3 ASPECTOS DE IMPLEMENTAÇÃO

Para estimar a latência introduzida à malha de controle pela arquitetura proposta neste trabalho, ela foi implementada fisicamente por de meio do uso de dois microcontroladores da *Texas Instruments* (INSTRUMENTS, 2018). O modelo do supervisor foi implementado em um *Tiva C Series TM4C1294NCPDT*, enquanto que o modelo do distinguidor em um *MSP430G2553*. Para a comunicação entre supervisor e distinguidor, foi utilizado o protocolo *Modbus*, amplamente utilizado em redes industrais, configurado no modo de transmissão *RTU* com um *baudrate* de 57600. O código fonte utilizado para implementar esse experimento está disponível em (ROSA, 2017). Os resultados são apresentados na Tabela 5.

Tabela 5: Atraso da comunicação

Teste	Clock (MHz)	Tempo(μs)
1	120	$206,0$
2	60	$206,0$
3	32	$208,0$
4	16	$210,0$
5	2	$299,1$

Os experimentos foram realizados em uma instância do exemplo da pequena fa-

brica com um buffer intermediário apresentado no Apêndice B e se referem à média de um conjunto representativo (fixo) de requisições. Para cada configuração de teste, o *clock* do microcontrolador *MSP430G2553* foi mantido fixo em $\approx 16MHz$, enquanto que no *Tiva C Series TM4C1294NCPDT* ele foi alterado de acordo com a Tabela 5. Como pode ser visto, nem o *overhead* se mostra suficientemente representativo para comprometer a malha de controle. Na prática, a dinâmica dos atuadores físicos presentes em sistemas de automação é na ordem dos segundos, o que é suficiente para que supervisor e distinguidor se comuniquem várias vezes e, portanto, estima-se que o *overhead* causado por essa comunicação seja irrelevante para o controle de sistemas reais.

5 CONTROLE MODULAR COM AUTO-APROXIMAÇÕES

Como vem sendo discutido ao logo deste trabalho, o uso de distinguidores, quando combinado com o uso de auto-aproximações, beneficia as etapas de desenvolvimento de controladores para SEDs, permeando-se por entre as fases de modelagem, síntese e implementação.

No entanto, tais benefícios ainda esbarram no fato de que, de maneira geral, esse *framework* de controle supervisório está inserido em um contexto monolítico. Ou seja, a operação de síntese de controle é processada sobre uma única estrutura (composição) de modelos. É possível mostrar que, computacionalmente, a complexidade de tal operação é exponencial em relação à quantidade de modelos envolvidos no problema de controle, o que remete, evidentemente, à quantidade de componentes presentes no sistema a ser controlado, bem como à quantidade de especificações de controle.

Por exemplo, seja um SED modelado por G e restrito por E, i.e.,

$$G = \Big|\Big|_{i=1}^{p} G^i \qquad \text{e} \qquad E = \Big|\Big|_{j=1}^{e} E^j.$$

Considerando-se que cada G^i tem no máximo m estados, então G possui m^p estados, no pior caso. Ainda, se cada E^j possui no máximo n estados, então E pode possuir até n^e estados. Logo, a operação monolítica de síntese possui complexidade de ordem $\mathcal{O}[(m^p.n^e)^2]$.

Na prática, esse fator de complexidade de síntese limita o uso da teoria de controle supervisório, e consequentemente de todas as abordagens propostas nesse trabalho, para tratar de problemas industriais de grande porte, caracterizados justamente por conterem inúmeros subsistemas, componentes e restrições.

Nesse sentido, a etapa de conclusão deste trabalho se dedica em prover uma forma de reduzir essa complexidade, explorando a ideia de auto-aproximações no contexto de síntese modular, de modo que cada parte de um sistema possa ser coordenada por

módulos descentralizados de controle, obtidos em separado do restante do sistema e, em tese, de maneira mais simples. Assim, estima-se que seja possível favorecer tanto o desenvolvimento quanto a escalabilidade do sistema de controle.

5.1 SÍNTESE MODULAR COM AUTO-APROXIMAÇÕES

Nesta seção, será apresentada uma breve revisão da literatura sobre síntese modular de controladores para SEDs (subseção 5.1.1). Em seguida, na subseção 5.1.2, será descrito um algoritmo que sistematiza a construção de auto-aproximações no contexto modular e algumas propriedades que estruturam o seu desenvolvimento. Logo após, na subseção 5.1.2, será apresentado um comparativo em relação a ordem de complexidade enfrentada pela síntese nas abordagens apresentadas ao longo deste trabalho. Por fim, na subseção 5.1.4, o exemplo da subseção 2.3.1 é revisitado e resolvido modularmente com auto-aproximações.

5.1.1 REVISÃO DA LITERATURA

Abordagens modulares vêm sendo utilizadas na TCS como alternativa para lidar com a complexidade na síntese de controladores para sistema de grande porte. Por exemplo, em (QUEIROZ; CURY, 2000), é explorada a ideia de compartilhamento de eventos entre modelos, planta e especificação, para selecionar somente as partes necessárias do sistema (sub-plantas) para computar a solução controlável ótima para cada especificação.

Em (ÅKESSON et al., 2002) essa ideia é estendida de modo que as sub-plantas são selecionadas, incrementalmente, pelo compartilhamento de eventos não-controláveis com a especificação ou com as sub-plantas já selecionadas. Isso tende a reduzir à quantidade de modelos no processo de síntese do controlador ótimo em relação a abordagem anterior. Já em (MALIK; TEIXEIRA, 2016), os autores melhoram os resultados de (ÅKESSON et al., 2002) pela seleção das sub-plantas que desabilitam algum evento não-controlável "causador"da não-controlabilidade da especificação em relação as sub-plantas já selecionadas.

Em (TEIXEIRA et al., 2018) os autores estende essa ideia para o contexto de refinamentos e aproximações de forma que parte do problema de controle pode ser resolvido modularmente no alfabeto original por meio da abordagem de (QUEIROZ; CURY, 2000), enquanto que a outra parte é resolvida modularmente utilizando refinamentos e aproximações. No entanto, a abordagem de síntese modular com refinamentos e aproximações

enfrenta a mesma limitação que sua a versão monolítica (Seção 2.5), i.e., quando necessário partes do distinguidor devem ser selecionadas e adicionadas a síntese.

Nesse contexto, esta dissertação apresenta uma versão de síntese modular que além de integrar a ideia de auto-aproximações, também melhora os resultados de (TEIXEIRA et al., 2018).

5.1.2 ALGORITMO DE SÍNTESE MODULAR COM AUTO-APROXIMAÇÕES

Esta subseção apresenta uma alternativa de síntese modular que, para cada especificação de controle seleciona, quando necessário, um conjunto apropriado de elementos da plantas e do mecanismo de distinção de eventos para síntese de um controlador. Será mostrado, então, que a ação conjunta dos controladores obtidos por meio deste processo é equivalente a de um controlador obtido a partir da abordagem monolítica. Para isso, considera-se que a planta refinada G_{a}, a especificação E_{d} e o modelo do distinguidor H_{d} são representados pelos seguintes conjuntos de autômatos:

(i) $\mathcal{G}_{\mathrm{a}} = \{G_{\mathrm{a}}^1, \cdots, G_{\mathrm{a}}^p\}$ é conjunto de todas as (sub)plantas, tal que $G_{\mathrm{a}} = ||(\mathcal{G}_{\mathrm{a}})$;

(ii) $\mathcal{E}_d = \{E_{\mathrm{d}}^1, \cdots, E_{\mathrm{d}}^e\}$ é conjunto de todas as especificações de controle, tal que $E_{\mathrm{d}} = ||(\mathcal{E}_d)$;

(iii) $\mathcal{H} = \{ \bigcup_{\delta\in\Delta} H_{\mathrm{d}}^{\delta} \} \cup \{H_{\mathrm{d}}^{\Delta}\}$, para H_{d}^{δ} δ-preditível (Def. 14) e $L(H_{\mathrm{d}}^{\Delta}) = \Delta^*$, é o conjunto de todas as partes de um distinguidor $H_{\mathrm{d}} = ||(\mathcal{H})$ Δ-preditível.

Além disso, denota-se por $\|\emptyset = \langle \Delta, \{\mathsf{x}^\circ\}, \{\mathsf{x}^\circ\}, \{\mathsf{x}^\circ\}, \{\mathsf{x}^\circ\}\times\Delta\times\{\mathsf{x}^\circ\}\rangle$ um autômato composto por um único estado que reconhece todas as cadeias $t \in \Delta^*$, i.e., $\|\emptyset \xrightarrow{\sigma_i} \mathsf{x}^\circ$ para todo $\sigma_i \in \Delta$. Esse autômato é importante, pois como partes da planta e do distinguidor serão selecionadas incrementalmente, algumas propriedades serão verificadas sobre $\|\emptyset$. Por exemplo, a identificação de eventos alvos que, ao invés de ser realizada sobre todo o modelo da planta refinada $||(\mathcal{G}_{\mathrm{a}})$ pode ser conduzida utilizando $\|\emptyset$, visto que $L(||(\mathcal{G}_{\mathrm{a}})) \subseteq L(\|\emptyset) = \Delta^*$.

No contexto modular, um fato importante a se ressaltar é que, embora as subplantas possam ser assíncronas ou não, quando compostas com qualquer versão do distinguidor, ele não somente as torna síncronas, pela complementação de seus alfabetos, mas também as sobre-aproxima (Def. 3) em relação a toda planta composta com o mesmo distinguidor, ou seja, o comportamento de cada sub-planta passa a ser maior do que o comportamento de toda a planta. Isso contradiz a noção convencional que se tem de

controle modular, ou seja, de que o comportamento de cada sub-planta está incluso no comportamento de toda a planta. Essa ideia pode ser exemplificada e formalizada conforme abaixo.

Exemplo 14. Seja $\mathcal{G}_{\mathrm{a}} = \{G_{\mathrm{a}}^1, G_{\mathrm{a}}^2\}$ e $\mathcal{H} = \{H_{\mathrm{d}}, H_{\mathrm{d}}^{\Delta}\}$ modelados como na Figura 25, tal que $\Delta^b = \{b_1, b_2\}$ e $\Delta = \{a, b_1, b_2, c\}$.

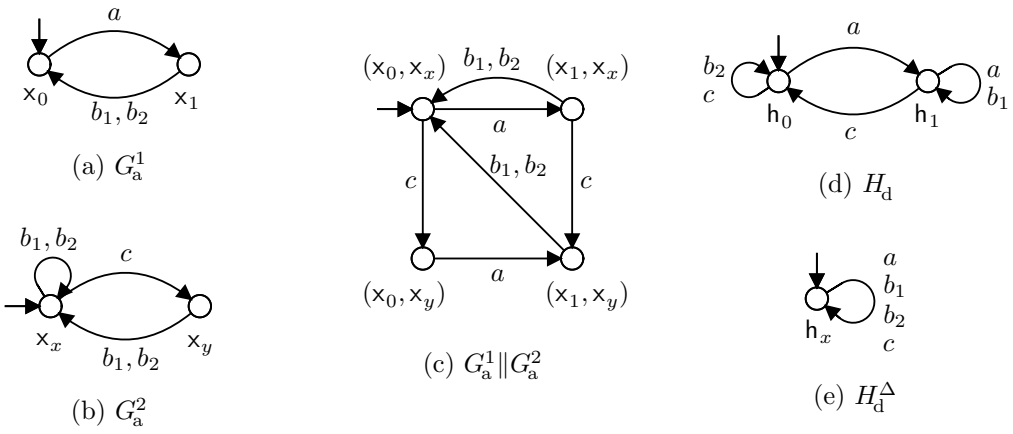


Figura 25: Modelos para G_{a}^1, G_{a}^2, $G_{\mathrm{a}}^1 \| G_{\mathrm{a}}^2$, H_{d} e H_{d}^{Δ}

Para demonstrar o efeito que distinguidores causam sobre sub-plantas, considere que G_{a}^1, G_{a}^2 e também a planta $G_{\mathrm{a}}^1 \| G_{\mathrm{a}}^2$ sejam compostas ao distinguidor H_{d} da Figura 25 (d). O resultado dessa operação é representado pelos autômatos $G_{\mathrm{a}}^1 \| H_{\mathrm{d}}$, $G_{\mathrm{a}}^2 \| H_{\mathrm{d}}$ e $G_{\mathrm{a}}^1 \| G_{\mathrm{a}}^2 \| H_{\mathrm{d}}$ da Figura 26 (a), (b) e (c), respectivamente.

Inicialmente, note que H_{d} complementa o alfabeto de G_{a}^1 pela habilitação do evento c em todos os estados de $G_{\mathrm{a}}^1 \| H_{\mathrm{d}}$ o mesmo acontece com evento a em $G_{\mathrm{a}}^2 \| H_{\mathrm{d}}$. Outro fato que pode ser observado é que, agora, tanto $L(G_{\mathrm{a}}^1 \| H_{\mathrm{d}})$ quanto $L(G_{\mathrm{a}}^2 \| H_{\mathrm{d}})$ não são sublinguagens de $L(G_{\mathrm{a}}^1 \| G_{\mathrm{a}}^2 \| H_{\mathrm{d}})$. Neste caso, $G_{\mathrm{a}}^1 \| H_{\mathrm{d}}$ e $G_{\mathrm{a}}^2 \| H_{\mathrm{d}}$ passam a ser sobre-aproximações de $G_{\mathrm{a}}^1 \| G_{\mathrm{a}}^2 \| H_{\mathrm{d}}$ (Def. 3), uma vez que $\Gamma((\mathsf{x}_i, \mathsf{x}_j, \mathsf{h}_k)) \subseteq \Gamma((\mathsf{x}_i, \mathsf{h}_k))$ e $\Gamma((\mathsf{x}_i, \mathsf{x}_j, \mathsf{h}_k)) \subseteq \Gamma((\mathsf{x}_j, \mathsf{h}_k))$ são sempre verdade, para $i, k \in \{0, 1\}$ e $j \in \{x, y\}$. O caso para o distinguidor H_{d}^{Δ} da Figura 25 (e) é análogo ao caso com H_{d}. □

Lema 5. *Para $\mathcal{G}_{\mathrm{a}}^i \subseteq \mathcal{G}_{\mathrm{a}}$ e $\mathcal{H}^i \subseteq \mathcal{H}$, tal que $H_{\mathrm{d}}^{\Delta} \in \mathcal{H}^i$, seja $\overline{\mathcal{G}_{\mathrm{a}}^i}$ o conjunto complementar de $\mathcal{G}_{\mathrm{a}}^i$. A relação de inclusão*

$$L(\Big|\Big|(\mathcal{G}_{\mathrm{a}}) \parallel \Big|\Big|(\mathcal{H}^i)) \subseteq L(\Big|\Big|(\mathcal{G}_{\mathrm{a}}^i) \parallel \Big|\Big|(\mathcal{H}^i))$$

é sempre verdade.

Demonstração. Será mostrado que $\Gamma((\mathsf{x}_i, \overline{\mathsf{x}_i}, \mathsf{h}_i)) \subseteq \Gamma((\mathsf{x}_i, \mathsf{h}_i))$, para $\mathsf{x}_i \in Q_{\|(\mathcal{G}_{\mathrm{a}}^i)}$, $\overline{\mathsf{x}_i} \in Q_{\|(\overline{\mathcal{G}_{\mathrm{a}}^i})}$

e $\mathsf{h}_i \in Q_{\|(\mathcal{H}^i)}$. Para isso, considere que, após uma cadeia qualquer $s \in L(\|(\mathcal{H}^i))$,

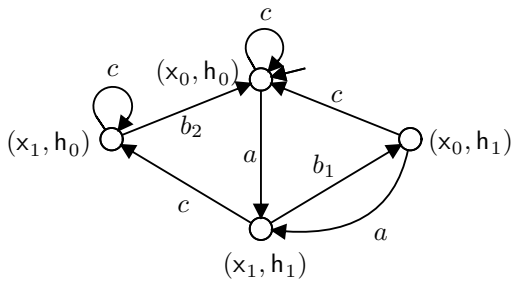
$$\Big\|(\mathcal{G}_{\mathrm{a}}) \parallel \Big\|(\mathcal{H}^i) \xrightarrow{s} (\mathsf{x}_i, \overline{\mathsf{x}_i}, \mathsf{h}_i) \quad \text{e} \quad \Big\|(\mathcal{G}_{\mathrm{a}}^i) \parallel \Big\|(\mathcal{H}^i) \xrightarrow{s} (\mathsf{x}_i, \mathsf{h}_i).$$

Então, para qualquer evento $\sigma_i \in \Gamma((\mathsf{x}_i, \overline{\mathsf{x}_i}, \mathsf{h}_i))$ é verdade que $\sigma_i \in \Gamma((\mathsf{x}_i, \mathsf{h}_i))$ e $\sigma_i \in \Gamma((\overline{\mathsf{x}_i}, \mathsf{h}_i))$, pois se $\sigma_i \notin \Gamma((\mathsf{x}_i, \mathsf{h}_i))$ ou $\sigma_i \notin \Gamma((\overline{\mathsf{x}_i}, \mathsf{h}_i))$ então $\sigma_i \notin \Gamma((\mathsf{x}_i, \overline{\mathsf{x}_i}, \mathsf{h}_i))$. Assim, $\|(\mathcal{G}_{\mathrm{a}}^i) \parallel \|(\mathcal{H}^i)$ é uma sobre-aproximação de $\|(\mathcal{G}_{\mathrm{a}}) \parallel \|(\mathcal{H}^i)$ pela Def. 3, e portanto

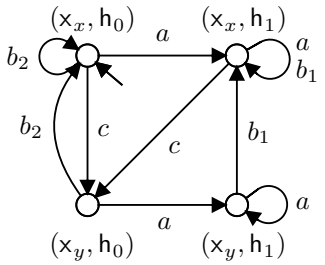
$$L(\Big\|(\mathcal{G}_{\mathrm{a}}) \parallel \Big\|(\mathcal{H}^i)) \subseteq L(\Big\|(\mathcal{G}_{\mathrm{a}}^i) \parallel \Big\|(\mathcal{H}^i)).$$

□

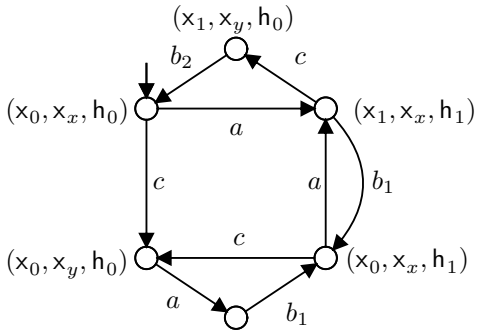
Ou seja, todas as cadeias reconhecidas por $L(\|(\mathcal{G}_{\mathrm{a}}) \parallel \|(\mathcal{H}^i))$ também são reconhecidas por $L(\|(\mathcal{G}_{\mathrm{a}}^i) \parallel \|(\mathcal{H}^i))$, uma vez que distinguidores sobre-aproximam o comportamento das sub-plantas.



(a) $G_{\mathrm{a}}^1 \| H_{\mathrm{d}}$



(b) $G_{\mathrm{a}}^2 \| H_{\mathrm{d}}$



(c) $G_{\mathrm{a}}^1 \| G_{\mathrm{a}}^2 \| H_{\mathrm{d}}$

Figura 26: Modelos para $G_{\mathrm{a}}^1 \| H_{\mathrm{d}}$, $G_{\mathrm{a}}^2 \| H_{\mathrm{d}}$ e $G_{\mathrm{a}}^1 \| G_{\mathrm{a}}^2 \| H_{\mathrm{d}}$

A seguir essas propriedades serão exploradas por um algoritmo para a síntese modular. A ideia básica é que partes da planta ($\mathcal{G}_{\mathrm{a}}' \in \mathcal{G}_{\mathrm{a}}$) e do distinguidor ($\mathcal{H}' \in \mathcal{H}$), quando necessárias, sejam selecionadas incrementalmente para a síntese. Como resultado espera-se que a complexidade da síntese seja reduzida e que o supervisor obtido a partir dessas partes seja controlável e.r.a. a planta distinguida G_{d}.

Inicialmente, o Algoritmo 4 verifica a existência de eventos alvo (Def. 13) em

Algoritmo 4 Síntese modular com auto-aproximações

Entrada: planta $\mathcal{G}_{\mathrm{a}} = \{G_{\mathrm{a}}^1, \cdots, G_{\mathrm{a}}^p\}$; especificação $E_{\mathrm{d}}{}' \in \mathcal{E}_d$; eventos não-controláveis Δ_{u}; distinguidor $\mathcal{H}$ como um conjunto de distinguidores H_{d}^{δ} δ-preditível, para todo $\delta \in \Delta$, e H_{d}^{Δ}, tal que $H_{\mathrm{d}} = \|(\mathcal{H})$ é Δ-preditível;

1: $\mathcal{G}_{\mathrm{a}}^0 \leftarrow \emptyset$; $\mathcal{H}^i \leftarrow \{H_{\mathrm{d}}^{\Delta}\}$; $S_{\mathrm{c}}^0 \leftarrow E_{\mathrm{d}}{}'$; $\Delta_{\mathrm{u}}^0 \leftarrow \emptyset$; $i \leftarrow 0$;

2: $\Delta_{\not\|}(\|(\mathcal{G}_{\mathrm{a}}^0)) \leftarrow \{\, \mu_j \in \Delta_{\mathrm{u}} \mid$ para $\mu_i \in \Delta$ tal que $\mu_i \neq \mu_j$, $(\|(\mathcal{G}_{\mathrm{a}}^i))\|S_{\mathrm{c}}^i \rightarrow (\mathsf{x},\mathsf{y})$ e $\mathsf{x} \xrightarrow{\mu_i,\mu_j}$ e $\mathsf{y} \xrightarrow{\mu_i}$ e $\mathsf{y} \not\xrightarrow{\mu_j}$ e $(\mathsf{x},\mathsf{y}) \notin \mathcal{S}^{\mathrm{u}} \,\}$;

3: $\mathcal{H}^i \leftarrow \mathcal{H}^i \cup \{\, H_{\mathrm{d}}^{\delta} \in \mathcal{H} \mid \delta \in \Delta_{\not\|}(\|(\mathcal{G}_{\mathrm{a}}^0)) \,\}$;

4: **Enquanto** $L(S_{\mathrm{c}}^i)$ é não Δ_{u}-controlável e.r.a. $L(\|(\mathcal{G}_{\mathrm{a}}^i) \parallel \|(\mathcal{H}^i))$ **faça**

5: $\quad \Delta_{\mathrm{u}}^{i+1} \leftarrow \Delta_{\mathrm{u}}^i \cup \{\, \mu_i \in \Delta_{\mathrm{u}} \mid (\|(\mathcal{G}_{\mathrm{a}}^i))\|S_{\mathrm{c}}^i \rightarrow (\mathsf{x},\mathsf{y})$ e $\mathsf{x} \xrightarrow{\mu_i}$ e $\mathsf{y} \not\xrightarrow{\mu_i} \,\}$;

6: $\quad \mathcal{G}_{\mathrm{a}}^{i+1} \leftarrow \{\, G_{\mathrm{a}}{}' \in \mathcal{G}_{\mathrm{a}} \mid G_{\mathrm{a}}{}' \rightarrow \mathsf{x} \not\xrightarrow{\mu_i}$ para algum $\mu_i \in \Delta_{\mathrm{u}}^{i+1} \,\}$;

7: $\quad \mathcal{H}^{i+1} \leftarrow \mathcal{H}^i \cup \{\, H_{\mathrm{d}}^{\delta} \in \mathcal{H} \mid \delta \in \mathcal{I}_{\not\|}(\, [\, \|(\mathcal{G}_{\mathrm{a}}^{i+1}) \parallel \|(\mathcal{H}^i)],\ \Delta_{\not\|}(\|(\mathcal{G}_{\mathrm{a}}^0)\,) \,\}$;

8: $\quad S_{\mathrm{c}}^{i+1} \leftarrow \sup\mathcal{C}(\|(\mathcal{G}_{\mathrm{a}}^{i+1} \parallel \|(\mathcal{H}^{i+1}), E_{\mathrm{d}}{}', \Delta_{\mathrm{u}}^{i+1})$;

9: $\quad i \leftarrow i+1$;

10: **retornar** S_{c}^i;

$\|(\mathcal{G}_{\mathrm{a}}^0)\|E_{\mathrm{d}}{}'$ (linha 2). Caso não existam, i.e., $\Delta_{\not\|}(\|(\mathcal{G}_{\mathrm{a}}^0)) = \emptyset$, a execução do Algoritmo 4 pode seguir sem adicionar nenhuma parte do distinguidor à síntese, já que, para qualquer $\mathcal{G}_{\mathrm{a}}^i \subseteq \mathcal{G}_{\mathrm{a}}$, a planta $\|(\mathcal{G}_{\mathrm{a}}^i)$ é uma auto-aproximação que leva à solução controlável ótima, pelo Teorema 1. Caso contrário, i.e., quando $\Delta_{\not\|}(\|(\mathcal{G}_{\mathrm{a}}^0)) \neq \emptyset$, os distinguidores $H_{\mathrm{d}}^{\delta} \in \mathcal{H}$, para $\delta \in \Delta_{\not\|}(\|(\mathcal{G}_{\mathrm{a}}^0))$, são adicionados a $\mathcal{H}^i$ (linha 3).

Em seguida, a controlabilidade de $S_{\mathrm{c}}^0 = E_{\mathrm{d}}{}'$ é testada (linha 4). Se S_{c}^0 é Δ_{u}-controlável, então o Algoritmo 4 finaliza sua execução retornando S_{c}^0, que é o próprio $E_{\mathrm{d}}{}'$. Mas, se S_{c}^0 não é Δ_{u}-controlável, inicia-se um processo iterativo para encontrar um S_{c}^i que seja Δ_{u}-controlável e.r.a. $\|(\mathcal{G}_{\mathrm{a}}^i) \parallel \|(\mathcal{H}^i)$.

Esse processo, inicialmente, identifica os eventos não-controláveis em Δ_{u} que são possíveis em $\|(\mathcal{G}_{\mathrm{a}}^i) \parallel \|(\mathcal{H}^i)$ mas não em S_{c}^i e os adiciona a $\Delta_{\mathrm{u}}^{i+1}$ (linha 5). Esses eventos são chamados de *causa da não-controlabilidade*. Note que os eventos não-controláveis que não estão em $\Delta_{\mathrm{u}}^{i+1}$ serão considerados pela síntese como controláveis.

Logo após, na linha 6, um conjunto de plantas $\mathcal{G}_{\mathrm{a}}^{i+1}$ é selecionado. Uma das principais vantagens da abordagem proposta em relação a (TEIXEIRA et al., 2018), e de maneira mais geral, ao próprio *Controle Modular Local* (CML) (QUEIROZ; CURY, 2000), é como as plantas em $\mathcal{G}_{\mathrm{a}}^{i+1}$ são selecionadas. Ao invés de selecionar as plantas pelo compartilhamento de eventos com a especificação, as plantas em $\mathcal{G}_{\mathrm{a}}$ são selecionadas por serem capaz de desabilitar algum evento não-controlável em $\Delta_{\mathrm{u}}^{i+1}$. Assim, as plantas ($\overline{\mathcal{G}_{\mathrm{a}}^{i+1}}$) que não são consideradas na síntese, pelo Algoritmo 4, mas possuem em seu alfabeto algum evento $\mu_i \in \Delta_{\mathrm{u}}^{i+1}$, são μ_i-habilitada (Def. 1), i.e., o evento μ_i é possível em

todos os estados dessas plantas. Isso garante que $||(\overline{\mathcal{G}_{\mathrm{a}}^{i+1}})$ não exerça nenhuma restrição comportamental sobre os eventos em $\Delta_{\mathrm{u}}^{i+1}$. Essa propriedade é formalmente descrita pelo Lema 6.

Lema 6. *Para $\Delta' \subseteq \Delta_{\mathrm{u}}$, sejam $\mathcal{G}_{\mathrm{a}}{}' = \{G_{\mathrm{a}}{}' \in \mathcal{G}_{\mathrm{a}} \mid G_{\mathrm{a}}{}' \to \mathsf{x} \overset{\mu_i}{\nrightarrow}$ para algum $\mu_i \in \Delta'\}$ e $\overline{\mathcal{G}_{\mathrm{a}}{}'} = \mathcal{G}_{\mathrm{a}} \setminus \mathcal{G}_{\mathrm{a}}{}'$ o conjunto complementar de $\mathcal{G}_{\mathrm{a}}{}'$. Se existe algum $G_{\mathrm{a}}{}'' \in \overline{\mathcal{G}_{\mathrm{a}}{}'}$ tal que $\mu_i \in \Delta^{G_{\mathrm{a}}{}''}$, então $G_{\mathrm{a}}{}''$ é μ_i-habilitado. Como efeito, pode-se generalizar que*

$$\Big|\Big|(\overline{\mathcal{G}_{\mathrm{a}}{}'}) \to \mathsf{q} \overset{\mu_i}{\nrightarrow} \text{ para todo } \mathsf{q} \in Q_{||(\overline{\mathcal{G}_{\mathrm{a}}{}'})}.$$

Demonstração. A prova desse lema se segue por construção pela linha 6 do Algoritmo 4 e da Def. 1. □

Essa propriedade será utilizada ao longo desta subseção para derivar e demonstrar algumas propriedades de controlabilidade e de equivalência entre soluções de controle.

Na sequência do algoritmo, na linha 7, é verificada a possível existência de eventos indiretamente alvo (Def. 17) em $||(\mathcal{G}_{\mathrm{a}}^{i+1}) \parallel ||(\mathcal{H}^i)$ e.r.a. eventos alvo $(\Delta_{\mathcal{U}}(\mathcal{G}_{\mathrm{a}}^0))$. Caso existam, o conjunto $\mathcal{H}^i$ é atualizado para $\mathcal{H}^{i+1}$ com os modelos $H_{\mathrm{d}}^{\delta} \in \mathcal{H}$ que tornam $||(\mathcal{G}_{\mathrm{a}}^{i+1})$ preditível e.r.a. eventos indiretamente alvo. Então, na linha 8, uma nova versão do supervisor S_{c}^i é computada. Esse processo iterativo se repete até que S_{c}^i seja Δ_{u}-controlável e.r.a. $||(\mathcal{G}_{\mathrm{a}}^i) \parallel ||(\mathcal{H}^i)$.

Exemplo 15. Para o Exemplo 14, considere que $\Delta_{\mathrm{c}} = \{a, c\}$, $\Delta_{\mathrm{u}} = \{b_1, b_2\}$ e a especificação E_{d} da Figura 27.

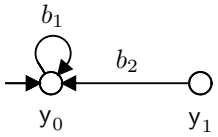


Figura 27: Modelo de E_{d} para o Exemplo 14

Utilizando o Algoritmo 4 para computar uma solução de controle para esse exemplo, o primeiro passo a ser executado é a verificação da possível existência de eventos alvos (linha 4). Para isso, o algoritmo utiliza a planta $||(\mathcal{G}_{\mathrm{a}}^0)$ (Figura 28 (a)) e $S_{\mathrm{c}}^0 = E_{\mathrm{d}}$, de modo que a verificação é realizada sobre o modelo $||(\mathcal{G}_{\mathrm{a}}^0)||S_{\mathrm{c}}^0$ (Figura 28 (b)). Note que no estado $(\mathsf{x}, \mathsf{y}_0)$ do autômato $||(\mathcal{G}_{\mathrm{a}}^0)||S_{\mathrm{c}}^0$, o evento b_2 é desabilitado enquanto que b_1 é habilitado, então b_2 é identificado com alvo e consequentemente adicionado a $\Delta_{\mathcal{U}}(||(\mathcal{G}_{\mathrm{a}}^0)) = \{b_2\}$. Então, o algoritmo seleciona o distinguidor para b_2 que nesse caso é H_{d} (Figura 25 (d)) e o adiciona a $\mathcal{H}^0 = \{H_{\mathrm{d}}^{\Delta}, H_{\mathrm{d}}\}$ (linha 3).

(a) $||(\mathcal{G}_{\mathrm{a}}^0)$ (b) $||(\mathcal{G}_{\mathrm{a}}^0)||S_{\mathrm{c}}^0$

Figura 28: Modelos para $||(\mathcal{G}_{\mathrm{a}}^0)$ e $||(\mathcal{G}_{\mathrm{a}}^0)||S_{\mathrm{c}}^0$

O próximo passo a ser realizado é o teste de controlabilidade (linha 4) do supervisor S_{c}^0 em relação a $||(\mathcal{G}_{\mathrm{a}}^0)||\,||(\mathcal{H}^0)$ (Figura 29), o qual resulta que S_{c}^0 não é Δ_{u}-controlável $||(\mathcal{G}_{\mathrm{a}}^0)||\,||(\mathcal{H}^0)$, visto que a cadeia b_2 é possível em $||(\mathcal{G}_{\mathrm{a}}^0)||\,||(\mathcal{H}^0)$ mas não em S_{c}^0. Assim, o

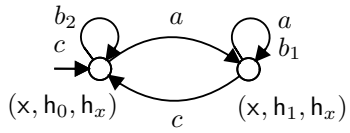


Figura 29: Modelo para $||(\mathcal{G}_{\mathrm{a}}^0)||\,||(\mathcal{H}^0)$

algoritmo entra na estrutura de repetição que executa os seguintes passos:

(i) *Identificação dos eventos causadores da não-controlabilidade* Δ_{u}^i (linha 5): Nesse passo o evento b_2 é identificado com causa da não-controlabilidade de S_{c}^0. Assim, $\Delta_{\mathrm{u}}^1 = \{b_2\}$;

(ii) *Seleção das plantas* Δ_{u}^i*-dominantes* (linha 6): A partir das informações em Δ_{u}^1, as plantas b_2-dominantes em $\mathcal{G}_{\mathrm{a}}$ são selecionadas e adicionadas a $\mathcal{G}_{\mathrm{a}}^1$. Nesse exemplo, a única planta que satisfaz a definição de dominância (Def. 2) e.r.a. evento b_2 é a planta G_{a}^1 (Figura 25 (a)). Dessa forma, $\mathcal{G}_{\mathrm{a}}^1 = \{G_{\mathrm{a}}^1\}$;

(iii) *Verificação dos eventos indiretamente alvo e seleção de distinguidor* (linha 7): Após a seleção das plantas, o algoritmo verifica a possível existência de eventos indiretamente alvo. Nesse exemplo, esse teste é realizado sobre o modelo $||(\mathcal{G}_{\mathrm{a}}^1)||\,||(\mathcal{H}^0)||E_{\mathrm{d}}$ (Figura 30).

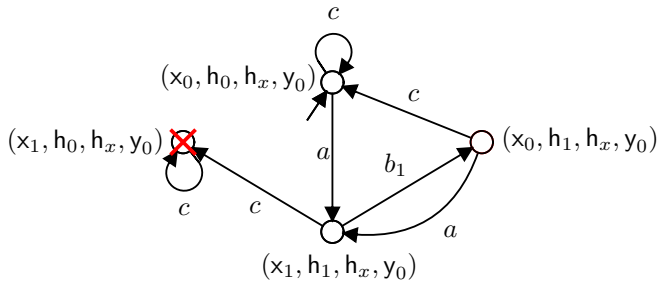
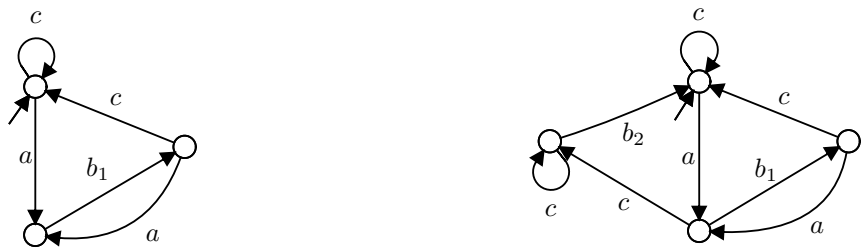


Figura 30: Modelo para $||(\mathcal{G}_{\mathrm{a}}^1)||\,||(\mathcal{H}^0)||E_{\mathrm{d}}$

Note que o estado $(\mathsf{x}_1, \mathsf{h}_0, \mathsf{h}_x, \mathsf{y}_0)$ em $\|(\mathcal{G}_{\mathrm{a}}^1)\| \, \|(\mathcal{H}^0)\| E_{\mathrm{d}}$ é visto como mau estado por desabilitar $b_2 \in \Delta_{\mathrm{u}}(\|(\mathcal{G}_{\mathrm{a}}^0))$, então o único evento que será desabilitado pela síntese a fim de evitar esse estado será o evento controlável c. Portanto, $\mathcal{H}^1 = \mathcal{H}^0 = \{H_{\mathrm{d}}^{\Delta}, H_{\mathrm{d}}\}$;

(iv) *Cálculo de um novo supervisor* S_{c}^i (linha 8): Por fim, um novo supervisor para a especificação E_{d} é calculado a partir das plantas, dos distinguidores e dos eventos causadores da não-controlabilidade selecionados pelos passos anteriores. O novo supervisor $S_{\mathrm{c}}^1 = \sup\mathcal{C}(\|(\mathcal{G}_{\mathrm{a}}^1) \parallel \|(\mathcal{H}^1), E_{\mathrm{d}}, \Delta_{\mathrm{u}}^1)$ calculado para esse exemplo é apresentado na Figura 31(a).



(a) S_{c}^1 (b) $\|(\mathcal{G}_{\mathrm{a}}^1) \parallel \|(\mathcal{H}^1)$

Figura 31: Modelos para S_{c}^1 e $\|(\mathcal{G}_{\mathrm{a}}^1) \parallel \|(\mathcal{H}^1)$

Na próxima iteração, a controlabilidade de S_{c}^1 e.r.a. $\|(\mathcal{G}_{\mathrm{a}}^1) \parallel \|(\mathcal{H}^1)$ (Figura 31(b)) utilizando todo o conjunto de eventos não-controláveis Δ_{u} é testada (linha 4). Esse teste resulta que S_{c}^1 é Δ_{u}-controlável e.r.a. $\|(\mathcal{G}_{\mathrm{a}}^1) \parallel \|(\mathcal{H}^1)$, então o algoritmo deixa a estrutura de repetição e finaliza sua execução retornando S_{c}^1 como a solução de controle para E_{d}. □

A partir da controlabilidade do supervisor S_{c}^i em relação às partes da planta e do distinguidor, selecionadas conforme o Algoritmo 4, pode-se derivar algumas propriedades de controlabilidade que auxiliam na demonstração da equivalência com a solução de controle monolítica (Seção 2.4). Essas propriedades são apresentadas a seguir.

Ao final da execução do Algoritmo 4 pode ser mostrado (Lema 7) que, além de S_{c}^i ser Δ_{u}-controlável e.r.a. $\|(\mathcal{G}_{\mathrm{a}}^i) \parallel \|(\mathcal{H}^i)$ (para especificação $E_{\mathrm{d}}{}' \in \mathcal{E}_d$), S_{c}^i também é Δ_{u}-controlável e.r.a. $\|(\mathcal{G}_{\mathrm{a}}) \parallel \|(\mathcal{H}^i)$.

Lema 7. *Para* Δ_{u}, $\mathcal{G}_{\mathrm{a}}^i \subseteq \mathcal{G}_{\mathrm{a}}$, $\mathcal{H}^i \subseteq \mathcal{H}$ *e* S_{c}^i, *computados pelo Algoritmo 4, seja* $\overline{\mathcal{G}_{\mathrm{a}}^i} = \mathcal{G}_{\mathrm{a}} \setminus \mathcal{G}_{\mathrm{a}}^i$ *o conjunto complementar de* $\mathcal{G}_{\mathrm{a}}^i$. *Se* $L(S_{\mathrm{c}}^i)$ *é* Δ_{u}*-controlável e.r.a.* $L(\|(\mathcal{G}_{\mathrm{a}}^i) \parallel \|(\mathcal{H}^i))$, *então* $L(S_{\mathrm{c}}^i)$ *também é* Δ_{u}*-controlável e.r.a.* $L(\|(\mathcal{G}_{\mathrm{a}}) \parallel \|(\mathcal{H}^i))$.

Demonstração. Sejam $s \in \Delta^*$, $\mu_i \in \Delta_{\mathrm{u}}$, $\mathsf{x}_i \in Q_{\|(\mathcal{G}_{\mathrm{a}}^i)}$, $\overline{\mathsf{x}_i} \in Q_{\|(\overline{\mathcal{G}_{\mathrm{a}}^i})}$, $\mathsf{h}_i \in Q_{\|(\mathcal{H}^i)}$ e $\mathsf{y} \in Q_{E_{\mathrm{d}}{}'}$. Considere que, após uma cadeia qualquer $s \in L(S_{\mathrm{c}}^i)$, um evento μ_i seja elegível em $\|(\mathcal{G}_{\mathrm{a}}^i) \parallel \|(\mathcal{H}^i)$, ou seja,

$$\Big\|(\mathcal{G}_{\mathrm{a}}^i) \parallel \Big\|(\mathcal{H}^i) \xrightarrow{s} (\mathsf{x}_i, \mathsf{h}_i) \xrightarrow{\mu_i} .$$

Então, pelo pressuposto de que S_{c}^i é Δ_{u}-controlável e.r.a. $\|(\mathcal{G}_{\mathrm{a}}^i) \parallel \|(\mathcal{H}^i)$, é verdade que

$$S_{\mathrm{c}}^i \xrightarrow{s} (\mathsf{x}_i, \mathsf{h}_i, \mathsf{y}) \xrightarrow{\mu_i} .$$

Pela linha 6 do Algoritmo 4, ou $\|(\overline{\mathcal{G}_{\mathrm{a}}^i})$ é μ_i-habilitado (Lema 6) ou $\mu_i \notin \Delta^{\|(\overline{\mathcal{G}_{\mathrm{a}}^i})}$. Caso contrário, o autômato $G_{\mathrm{a}}{}'' \in \overline{\mathcal{G}_{\mathrm{a}}^i}$, tal que $\mu_i \in \Delta^{G_{\mathrm{a}}{}''}$, seria adicionado a $\mathcal{G}_{\mathrm{a}}^i$ pela mesma linha do algoritmo. Então, pela definição de $\|$, é sempre verdade que

$$\Big\|(\mathcal{G}_{\mathrm{a}}^i) \parallel \Big\|(\overline{\mathcal{G}_{\mathrm{a}}^i}) \parallel \Big\|(\mathcal{H}^i) \xrightarrow{s} (\mathsf{x}_i, \overline{\mathsf{x}_i}, \mathsf{h}_i) \xrightarrow{\mu_i} .$$

Portanto, S_{c}^i também é Δ_{u}-controlável e.r.a. $\|(\mathcal{G}_{\mathrm{a}}) \parallel \|(\mathcal{H}^i))$, i.e.,

$$L(S_{\mathrm{c}}^i)\Delta_{\mathrm{u}} \cap L(\Big\|(\mathcal{G}_{\mathrm{a}}) \parallel \Big\|(\mathcal{H}^i)) \subseteq L(S_{\mathrm{c}}^i).$$

□

Pelo Lema 7, se um supervisor S_{c}^i é Δ_{u}-controlável e.r.a. uma parte $\|(\mathcal{G}_{\mathrm{a}}^i)$ da planta $\|(\mathcal{G}_{\mathrm{a}})$ e uma parte $\|(\mathcal{H}^i)$ do distinguidor $\|(\mathcal{H})$, ambas as partes selecionadas pelo Algoritmo 4, esse mesmo supervisor S_{c}^i é Δ_{u}-controlável e.r.a. toda planta composta a mesma parte do distinguidor. Além disso, pode ser demonstrado (Corolário 1) que, o supervisor S_{c}^i também é Δ_{u}-controlável e.r.a. $\|(\mathcal{G}_{\mathrm{a}}) \parallel \|(\mathcal{H})$.

Corolário 1. *Seja S_{c}^i um supervisor calculado pelo Algoritmo 4 para $E_{\mathrm{d}}{}'$ e.r.a. $\mathcal{G}_{\mathrm{a}}^i \subseteq \mathcal{G}_{\mathrm{a}}$ e $\mathcal{H}^i \subseteq \mathcal{H}$. Se S_{c}^i é Δ_{u}-controlável e.r.a. $L(\|(\mathcal{G}_{\mathrm{a}}) \parallel \|(\mathcal{H}^i))$, então S_{c}^i também é Δ_{u}-controlável e.r.a. $L(\|(\mathcal{G}_{\mathrm{a}}) \parallel \|(\mathcal{H}))$.*

Demonstração. Note que $L(\|(\mathcal{H})) \subseteq L(\|(\mathcal{H}^i)) \subseteq \Delta^*$ se segue da Def. 11. Portanto, a Δ_{u}-controlabilidade de S_{c}^i e.r.a. $L(\|(\mathcal{G}_{\mathrm{a}}) \parallel \|(\mathcal{H}))$ segue de $L(\|(\mathcal{G}_{\mathrm{a}}) \parallel \|(\mathcal{H})) \subseteq L(\|(\mathcal{G}_{\mathrm{a}}) \parallel \|(\mathcal{H}^i))$. □

Resta mostrar que o supervisor S_{c}^i, obtido a partir de partes da planta (linha 6) e do distinguidor (linha 3 e 7), leva a uma solução equivalente a quando se utiliza todas as partes da planta e do distinguidor na síntese.

Para isso, inicialmente, será mostrado (Lema 8) que, a síntese conduzida com as

plantas selecionadas, pelo Algoritmo 4 na linha 6, em relação aos eventos causa da não-controlabilidade (linha 5), leva a uma solução de controle equivalente à qual se considera toda a planta e todo o modelo do distinguidor. Então, em seguida esse resultado será estendido de modo a mostrar que, as partes do modelo do distinguidor, que não foram selecionadas pelo Algoritmo 4, podem ser removidas da síntese, e ainda assim a solução de controle permanece equivalente.

Lema 8. *Para $\mathcal{G}_{\mathrm{a}}^{i} \subseteq \mathcal{G}_{\mathrm{a}}$ e $\Delta_{\mathrm{u}}^{i} \subseteq \Delta_{\mathrm{u}}$, computado pelo Algoritmo 4 para $E_{\mathrm{d}}{}' \in \mathcal{E}_{d}$. Seja $||(\mathcal{H})$ o distinguidor Δ-preditível e $\overline{\mathcal{G}_{\mathrm{a}}^{i}}$ o conjunto complementar de $\mathcal{G}_{\mathrm{a}}^{i}$. Ao final do Algoritmo 4 é verdade que*

$$\mathrm{sup}\mathcal{C}(\Big|\Big|(\mathcal{G}_{\mathrm{a}}^{i}) \parallel \Big|\Big|(\mathcal{H}), E_{\mathrm{d}}{}', \Delta_{\mathrm{u}}^{i}) \parallel \Big|\Big|(\overline{\mathcal{G}_{\mathrm{a}}^{i}}) = \mathrm{sup}\mathcal{C}(\Big|\Big|(\mathcal{G}_{\mathrm{a}}) \parallel \Big|\Big|(\mathcal{H}), E_{\mathrm{d}}{}', \Delta_{\mathrm{u}}).$$

Demonstração. Sejam $s \in \Delta^{*}$, $\mu_i \in \Delta_{\mathrm{u}}$, $\mathsf{x}_i \in Q_{||(\mathcal{G}_{\mathrm{a}}^{i})}$, $\overline{\mathsf{x}_i} \in Q_{||(\overline{\mathcal{G}_{\mathrm{a}}^{i}})}$, $\mathsf{h} \in Q_{||(\mathcal{H})}$ e $\mathsf{y} \in Q_{E_{\mathrm{d}}{}'}$.

Como $S_{\mathrm{d}}^{i} = \mathrm{sup}\mathcal{C}(||(\mathcal{G}_{\mathrm{a}}^{i}) \parallel ||(\mathcal{H}), E_{\mathrm{d}}{}', \Delta_{\mathrm{u}}^{i})$ é Δ_{u}-controlável e.r.a. $||(\mathcal{G}_{\mathrm{a}}^{i}) \parallel ||(\mathcal{H})$, então S_{d}^{i} também é Δ_{u}-controlável e.r.a. $||(\mathcal{G}_{\mathrm{a}}) \parallel ||(\mathcal{H})$, pelo Lema 7. Isso resulta que, após uma cadeia qualquer $s \in L(S_{\mathrm{d}}^{i})$, se um evento $\mu_i \in \Delta_{\mathrm{u}}$ é observado em

$$\Big|\Big|(\mathcal{G}_{\mathrm{a}}^{i}) \parallel \Big|\Big|(\mathcal{H}) \xrightarrow{s} (\mathsf{x}_i, \mathsf{h}) \xrightarrow{\mu_i} \quad \text{e} \quad \Big|\Big|(\mathcal{G}_{\mathrm{a}}) \parallel \Big|\Big|(\mathcal{H}) \xrightarrow{s} (\mathsf{x}_i, \overline{\mathsf{x}_i}, \mathsf{h}) \xrightarrow{\mu_i} \tag{3}$$

então

$$S_{\mathrm{d}}^{i} \xrightarrow{s} (\mathsf{x}_i, \mathsf{h}, \mathsf{y}) \xrightarrow{\mu_i}.$$

Pela controlabilidade de $S_{\mathrm{d}} = \mathrm{sup}\mathcal{C}(||(\mathcal{G}_{\mathrm{a}}) \parallel ||(\mathcal{H}), E_{\mathrm{d}}{}', \Delta_{\mathrm{u}})$ e.r.a. $||(\mathcal{G}_{\mathrm{a}}) \parallel ||(\mathcal{H})$, segue que

$$\Big|\Big|(\mathcal{G}_{\mathrm{a}}) \parallel \Big|\Big|(\mathcal{H}^{i}) \xrightarrow{s} (\mathsf{x}_i, \overline{\mathsf{x}_i}, \mathsf{h}) \xrightarrow{\mu_i} \quad \text{e} \quad S_{\mathrm{d}} \xrightarrow{s} (\mathsf{x}_i, \overline{\mathsf{x}_i}, \mathsf{h}, \mathsf{y}) \xrightarrow{\mu_i}.$$

Além disso, visto que $L(||(\mathcal{G}_{\mathrm{a}}) \parallel ||(\mathcal{H}) \subseteq L(||(\mathcal{G}_{\mathrm{a}}^{i}) \parallel ||(\mathcal{H}))$ é verdade, pelo Lema 5, então se segue que $\Gamma((\mathsf{x}_i, \overline{\mathsf{x}_i}, \mathsf{h}, \mathsf{y})) \subseteq \Gamma((\mathsf{x}_i, \mathsf{h}, \mathsf{y}))$, e portanto $L(S_{\mathrm{d}}) \subseteq L(S_{\mathrm{d}}^{i})$.

Como, pela linha 6 do Algoritmo 4, ou $||(\overline{\mathcal{G}_{\mathrm{a}}^{i}})$ é μ_i-habilitado (Lema 6) ou $\mu_i \notin \Delta^{||(\overline{\mathcal{G}_{\mathrm{a}}^{i}})}$, então $S_{\mathrm{d}}^{i} \parallel ||(\overline{\mathcal{G}_{\mathrm{a}}^{i}})$ também é Δ-controlável e.r.a. $||(\mathcal{G}_{\mathrm{a}}^{i}) \parallel ||(\mathcal{H})$, e adicionalmente a $||(\mathcal{G}_{\mathrm{a}}) \parallel ||(\mathcal{H})$.

Assim, para uma cadeia qualquer $s \in L(S_{\mathrm{d}}^{i} \parallel ||(\overline{\mathcal{G}_{\mathrm{a}}^{i}}))$ e $\mu_i \in \Delta_{\mathrm{u}}$, a Equação (3) é sempre verdade, e $s\mu_i \in L(S_{\mathrm{d}}^{i} \parallel ||(\overline{\mathcal{G}_{\mathrm{a}}^{i}}))$. Isso implica que

$$\Gamma((\mathsf{x}_i, \overline{\mathsf{x}_i}, \mathsf{h}, \mathsf{y})) = \Gamma((\mathsf{x}_i, \mathsf{h}, \mathsf{y})) \setminus \{\Delta^{||(\overline{\mathcal{G}_{\mathrm{a}}^{i}})} \setminus \Gamma(\overline{\mathsf{x}_i})\}.$$

Portanto, $L(S_{\mathrm{d}}) = L(S_{\mathrm{d}}^{i} \parallel ||(\overline{\mathcal{G}_{\mathrm{a}}^{i}}))$. ∎

Teorema 3. *Sejam $\mathcal{G}_{\mathrm{a}}$, $E_{\mathrm{d}}{}' \in \mathcal{E}_d$ uma especificação, $\mathcal{H}$ um conjunto de distinguidores, tal que $H_{\mathrm{d}} = ||(\mathcal{H})$, e tal que $G_{\mathrm{d}} = ||(\mathcal{G}_{\mathrm{a}})\|H_{\mathrm{d}}$ é a versão Δ-preditível da planta $||(\mathcal{G}_{\mathrm{a}})$. Ainda, seja S_{c}^i um supervisor calculado pelo Algoritmo 4 para $E_{\mathrm{d}}{}'$ e.r.a. $\mathcal{G}_{\mathrm{a}}^i \subseteq \mathcal{G}_{\mathrm{a}}$ e $\mathcal{H}^i \subseteq \mathcal{H}$. Ao final do Algoritmo 4 é sempre verdade que*

$$L(S_{\mathrm{c}}^i \parallel \Big|\Big|(\mathcal{G}_{\mathrm{a}}) \parallel H_{\mathrm{d}}) = L(\mathrm{sup}\mathcal{C}(G_{\mathrm{d}}, E_{\mathrm{d}}{}', \Delta_{\mathrm{u}})).$$

Demonstração. Inicialmente, note que

$$\mathrm{sup}\mathcal{C}(\Big|\Big|(\mathcal{G}_{\mathrm{a}}^i) \parallel \Big|\Big|(\mathcal{H}), E_{\mathrm{d}}{}', \Delta_{\mathrm{u}}^i) \parallel \Big|\Big|(\overline{\mathcal{G}_{\mathrm{a}}^i}) = \mathrm{sup}\mathcal{C}(G_{\mathrm{d}}, E_{\mathrm{d}}{}', \Delta_{\mathrm{u}})$$

é verdade, pelo Lema 8. Como, ao final Algoritmo 4, pelas linhas 3 e 11, $||(\mathcal{G}_{\mathrm{a}}^i) \parallel ||(\mathcal{H}^i)$ é uma auto-aproximação, i.e., é uma aproximação livre de eventos alvo ($\Delta_{\not u}(||(\mathcal{G}_{\mathrm{a}}^i) \parallel ||(\mathcal{H}^i)) = \emptyset$) e indiretamente alvo ($\Delta_{\not\not u}(||(\mathcal{G}_{\mathrm{a}}^i) \parallel ||(\mathcal{H}^i)) = \emptyset$), então

$$\mathrm{sup}\mathcal{C}(\Big|\Big|(\mathcal{G}_{\mathrm{a}}^i) \parallel \Big|\Big|(\mathcal{H}^i), E_{\mathrm{d}}{}', \Delta_{\mathrm{u}}^i) \parallel \Big|\Big|(\overline{\mathcal{G}_{\mathrm{a}}^i}) \parallel H_{\mathrm{d}} = \mathrm{sup}\mathcal{C}(G_{\mathrm{d}}, E_{\mathrm{d}}{}', \Delta_{\mathrm{u}}),$$

pelo Teorema 2. □

O último passo a ser mostrado é que a ação conjunta de todos os supervisores S_{c}^i, computado pelo Algoritmo 4, para cada especificação $E_{\mathrm{d}}{}^j \in \mathcal{E}_d$, leva à mesma ação de controle coordenada por um supervisor monolítico, obtido pelo método convencional como em (CURY et al., 2015). Para isso, será inicialmente aplicado um resultado que mostra que, sob o ponto de vista de controlabilidade, a síntese envolvendo uma composição de especificações pode ser fragmentada e executada individualmente, para cada especificação.

Proposição 1. *Para uma planta $G_{\mathrm{d}} = ||(\mathcal{G}_{\mathrm{a}})\|H_{\mathrm{d}}$ e uma especificação $E_{\mathrm{d}} = ||(\mathcal{E}_d)$ é verdade que*

$$\Big|\Big|_{j=1}^{e} \mathrm{sup}\mathcal{C}(G_{\mathrm{d}}, E_{\mathrm{d}}{}^j, \Delta_{\mathrm{u}}) = \mathrm{sup}\mathcal{C}(G_{\mathrm{d}}, E_{\mathrm{d}}, \Delta_{\mathrm{u}}).$$

Demonstração. A prova dessa proposição se segue diretamente da proposição 7 em (BRANDIN et al., 2004). □

Teorema 4. *Seja $\mathcal{G}_{\mathrm{a}}$ um conjunto de plantas, $E_{\mathrm{d}} = ||(\mathcal{E}_d)$ uma especificação, H_{d} o distinguidor Δ-predítivel e $G_{\mathrm{d}} = ||(\mathcal{G}_{\mathrm{a}})\|H_{\mathrm{d}}$. Além disso, seja $\mathcal{S}_{\mathrm{c}}$ o conjunto de todos os supervisores S_{c}^i calculados para cada $E_{\mathrm{d}}{}' \in \mathcal{E}_d$ pelo Algoritmo 4. O supervisor $||(\mathcal{S}_{\mathrm{c}})$ leva à solução ótima, i.e.,*

$$\Big[\Big|\Big|(\mathcal{S}_{\mathrm{c}})\Big] \parallel \Big[\Big|\Big|(\mathcal{G}_{\mathrm{a}})\Big] \|H_{\mathrm{d}} = \mathrm{sup}\mathcal{C}(G_{\mathrm{d}}, E_{\mathrm{d}}, \Delta_{\mathrm{u}}).$$

Demonstração. Se segue diretamente do Teorema 3 e da Prop. 1. □

O Teorema 4 mostra que, a solução de controle para múltiplas especificações pode ser computada pelo Algoritmo 4 para cada especificação, de modo que a ação de controle é dada pela ação conjunta de todos os controladores obtidos modularmente.

Como na versão modular de controle parte, ou todo, o modelo do distinguidor também é removida da operação síntese, uma alternativa para implementar tal solução de controle seria utilizar a arquitetura proposta no Capítulo 4. Para isso, bastaria considerar $S_{\mathrm{a}} = ||(\mathcal{S}_{\mathrm{c}})||\,||(\mathcal{G}_{\mathrm{a}})$, de modo que o restante da implementação seguiria de maneira análoga.

5.1.3 ANÁLISE DE COMPLEXIDADE

Seja um SED composto por $|\mathcal{G}_{\mathrm{a}}| = p$ sub-plantas em Δ, modeladas com no máximo m estados cada. Então, m^p é o número de estados que modela a planta refinada $||(\mathcal{G}_{\mathrm{a}})$. Além disso, seja esse sistema restrito por $|\mathcal{E}_d| = e$ especificações, de modo que cada uma das especificações contenha no máximo n estados. Ainda, seja $|\mathcal{H}| = h$ a quantidade de distinguidores necessária para distinguir os eventos em Δ de forma que cada um deles seja modelado por r estados.

Assim, a complexidade da síntese monolítica a partir da versão Δ-preditível da planta ($G_{\mathrm{d}} = ||(\mathcal{G}_{\mathrm{a}})||\,||(\mathcal{H})$) é na ordem de

$$\mathcal{O}[(m^p n^e r^h)^2],$$

enquanto que com a versão da planta sem nenhuma distinção de eventos é de

$$\mathcal{O}[(m^p n^e r^0)^2].$$

Já a complexidade utilizando uma auto-aproximação (Seção 3.1) é na ordem de

$$\mathcal{O}[(m^p n^e r^{h'})^2],$$

para $0 \leq h' \leq h$. Note que, no pior caso a complexidade da síntese utilizando uma auto-aproximação é igual a da síntese utilizando todo o distinguidor, ao passo que no melhor caso é igual a versão sem nenhum distinguidor.

Por fim, a complexidade enfrentada para computar o controlador para cada especificação ${E_{\mathrm{d}}}' \in \mathcal{E}_d$ a partir da abordagem modular com auto-aproximação (Seção 5.1) é

na ordem de

$$\mathcal{O}[(m^{p'} n\ r^{h'})^2],$$

com $0 \leq p' \leq p$ e $0 \leq h' \leq h$.

A partir desta análise pode ser visto que a abordagem modular com auto-aproximações reduz a ordem de complexidade em relação ao número de plantas adicionadas à síntese, bem como fixa a 1 o expoente (e) do número de estados das especificações. Além disso, o expoente (h') associado à quantidade de modelos do distinguidor pode ser visto como mínimo, uma vez que na prática é improvável que eventos não-controláveis ocorram sequencialmente sem nenhuma interrupção causada pela ocorrência de ao menos um evento controlável.

Embora a ordem de complexidade enfrentada pela operação de síntese ainda seja exponencial em relação ao número de modelos, ela é bem menor se comparada às abordagens monolíticas apresentadas nesse trabalho.

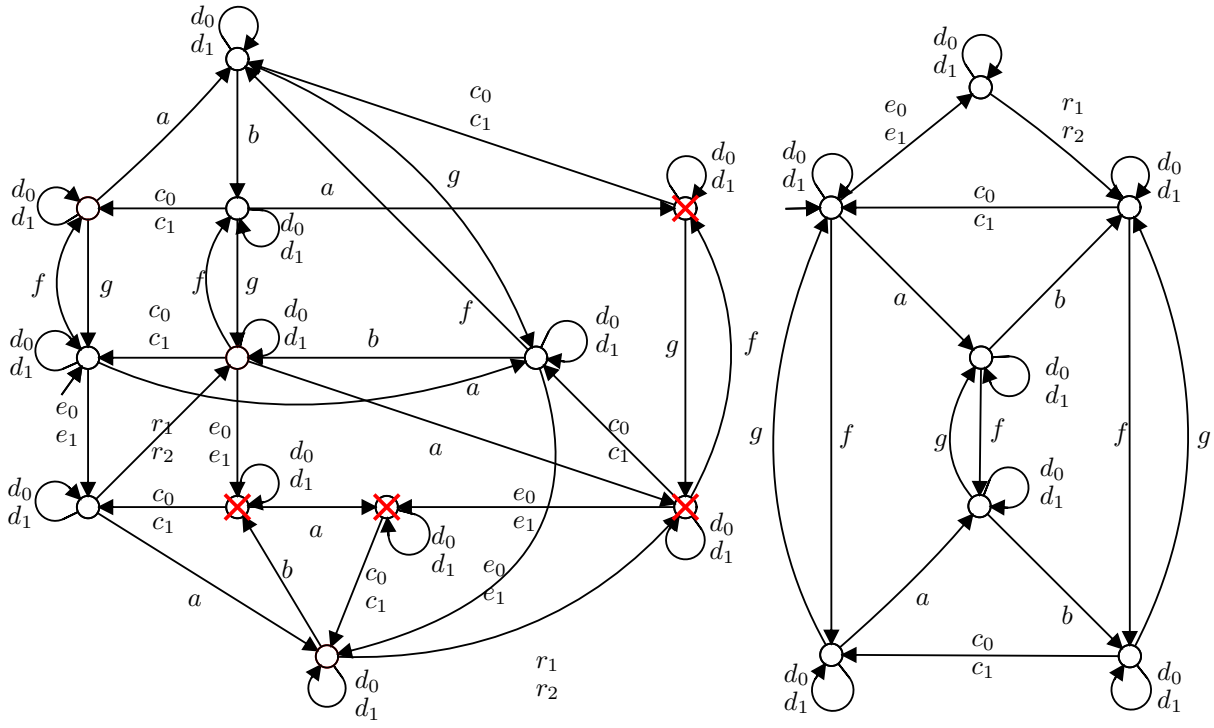
5.1.4 ESTUDO DE CASO COM SÍNTESE MODULAR

Considere obter a versão modular da solução de controle a partir de auto-aproximações para o exemplo da subseção 2.4.1 utilizando o Algoritmo 4. Como as especificações E^1_{d}, E^2_{d} e E^3_{d} desabilitam eventos em Δ_{u} as condições estabelecidas no Algoritmo 4 precisam ser verificadas.

- Como E^1_{d} desabilita Δ^b e Δ^r no segundo estado, então os estados em $\|(\mathcal{G}^0_{\mathrm{a}})\|E^1_{\mathrm{d}}$ nos quais esses eventos são proibidos são classificados como $\mathcal{S}^{\mathrm{u}}$ pela Def. 12. Assim, $\Delta_{\not\downarrow}(\|(\mathcal{G}^0_{\mathrm{a}})) = \emptyset$ (linha 2), e consequentemente a verificação dos eventos indiretamente alvos da linha 7 retorna que $\mathcal{I}_{\not\downarrow}([\|(\mathcal{G}^i_{\mathrm{a}}) \parallel \|(\mathcal{H}^i)], \Delta_{\not\downarrow}(\|(\mathcal{G}^0_{\mathrm{a}})) = \emptyset$, pelo Lema 2.

  No entanto, visto que E^1_{d} não é Δ_{u}-controlável por desabilitar Δ^b e Δ^r, então para $i = 0$ tais conjuntos de eventos são adicionados a $\Delta^1_{\mathrm{u}} = \{b, r_1, r_2\}$ (linha 5). Em seguida, as plantas que modelam a ocorrência/comportamento dos eventos em Δ^1_{u} são selecionadas (linha 6), como resultado tem-se que $\mathcal{G}^1_{\mathrm{a}} = \{G^1_{\mathrm{a}}, G^3_{\mathrm{a}}\}$.

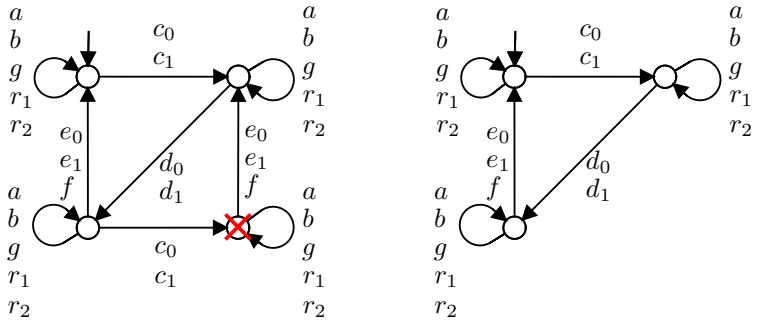
  Dessa forma, o supervisor $S^1_{\mathrm{c}} = \sup\mathcal{C}(\|(\mathcal{G}^1_{\mathrm{a}})\|H^{\Delta}_{\mathrm{d}}, E^1_{\mathrm{d}}, \{b, r_1, r_2\})$ (Figura 32 (b)) é obtido a partir do modelo $\|(\mathcal{G}^1_{\mathrm{a}})\|H^{\Delta}_{\mathrm{d}}\|E^1_{\mathrm{d}}$ (Figura 32 (a)) com 12 estados (linha 8) dos quais somente 7 estados sobrevivem após a síntese. Assim, a próxima iteração retorna que S^1_{c} é Δ_{u}-controlável e.r.a. $\|(\mathcal{G}^i_{\mathrm{a}})\|\mathcal{H}^i$ (linha 4), logo $S^{E^1_{\mathrm{d}}}_{\mathrm{c}} = S^1_{\mathrm{c}}$ é a solução ótima para E^1_{d}.



(a) $\|(\mathcal{G}_{\mathrm{a}}^1)\|H_{\mathrm{d}}^{\Delta}\|E_{\mathrm{d}}^1$ (b) $S_{\mathrm{c}}^{E_{\mathrm{d}}^1}$

Figura 32: Modelos para $\|(\mathcal{G}_{\mathrm{a}}^1)\|H_{\mathrm{d}}^{\Delta}\|E_{\mathrm{d}}^1$ e $S_{\mathrm{c}}^{E_{\mathrm{d}}^1}$

Analogamente, como E_{d}^2 desabilita Δ^d, a síntese do Algoritmo 4 explora o modelo $G_{\mathrm{a}}^2\|H_{\mathrm{d}}^{\Delta}\|E_{\mathrm{d}}^1$ (Figura 33 (a)) com 4 estados, de modo que o resultado é o supervisor $S_{\mathrm{c}}^{E_{\mathrm{d}}^2} = S_{\mathrm{c}}^1 = \sup\mathcal{C}(G_{\mathrm{a}}^2\|H_{\mathrm{d}}^{\Delta}, E_{\mathrm{d}}^2, \{d_1, d_2\})$ (Figura 33 (b)) com 3 estados.



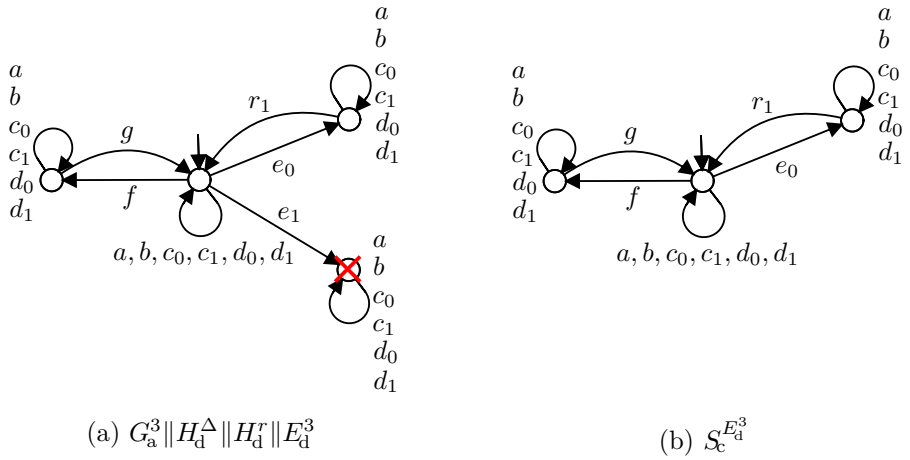
(a) $G_{\mathrm{a}}^2\|H_{\mathrm{d}}^{\Delta}\|E_{\mathrm{d}}^2$ (b) $S_{\mathrm{c}}^{E_{\mathrm{d}}^2}$

Figura 33: Modelos para $G_{\mathrm{a}}^2\|H_{\mathrm{d}}^{\Delta}\|E_{\mathrm{d}}^2$ e $S_{\mathrm{c}}^{E_{\mathrm{d}}^2}$

- Diferentemente, E_{d}^3 proíbe somente o evento r_2, enquanto que os demais eventos em Δ^r são livres para ocorrerem, então pela Def. 13 o evento r_2 é adicionado a $\Delta_u(\|(\mathcal{G}_{\mathrm{a}}^0)) = \{r_2\}$ (linha 2), e o modelo do distinguidor que torna qualquer $\mathcal{G}_{\mathrm{a}}^i \subseteq \mathcal{G}_{\mathrm{a}}$ em $\{r_2\}$-preditível é o H_{d}^r da Figura 17, então $\mathcal{H}^0 = \{H_{\mathrm{d}}^{\Delta}, H_{\mathrm{d}}^r\}$ (linha 3).

Como $S_{\mathrm{c}}^{0} = E_{\mathrm{d}}^{3}$ é não controlável por sempre proibir o evento r_2, tal evento é adicionado a $\Delta_{\mathrm{u}}^{1} = \{r_2\}$ (linha 5). Em seguida, na linha 6, a planta G_{a}^{3} é selecionada (por desabilitar r_2) fazendo $\mathcal{G}_{\mathrm{a}}^{1} = \{G_{\mathrm{a}}^{3}\}$.

A linha 7 resulta em $\mathcal{H}^{1} = \mathcal{H}^{0}$, porque o único estado em $\|(\mathcal{G}_{\mathrm{a}}^{1})\| \, \|(\mathcal{H}^{0})\| E_{\mathrm{d}}^{3}$ (Figura 34 (a)) classificado como $\mathcal{S}^{\mathrm{u}}$ por somente desabilitar eventos em $\Delta_{u}(\|(\mathcal{G}_{\mathrm{a}}^{0})) = \{r_2\}$ é alcançado pela cadeia te_1, para $t \in \Delta^{*}$, de modo que para evitar esse estado a síntese só desabilitaria o evento $e_1 \in \Delta_{\mathrm{c}}$.



(a) $G_{\mathrm{a}}^{3} \| H_{\mathrm{d}}^{\Delta} \| H_{\mathrm{d}}^{r} \| E_{\mathrm{d}}^{3}$ (b) $S_{\mathrm{c}}^{E_{\mathrm{d}}^{3}}$

Figura 34: Modelos para $G_{\mathrm{a}}^{3} \| H_{\mathrm{d}}^{\Delta} \| H_{\mathrm{d}}^{r} \| E_{\mathrm{d}}^{3}$ e $S_{\mathrm{c}}^{E_{\mathrm{d}}^{3}}$

Portanto, na linha 8, a síntese é processada a partir de uma auto-aproximação, e como resultado temos que o supervisor $S_{\mathrm{c}}^{E_{\mathrm{d}}^{3}} = S_{\mathrm{c}}^{1}$ (Figura 34 (b)) com 3 estados é a solução ótima para E_{d}^{3} pelo fim do Algoritmo 4.

A Tabela 6 contextualiza em número de estados o processo de síntese modular com auto-aproximações para cada especificação $E_{\mathrm{d}}{}^{j} \in \mathcal{E}_{d}$. Para fins comparativos os resultados da síntese monolítica obtidos nas seções anteriores utilizando todas ou alguma parte do distinguidor são apresentados nas duas últimas linhas da Tabela 6, respectivamente. Além disso, o mesmo problema de controle foi resolvido utilizando as abordagens de CML (QUEIROZ; CURY, 2000) e de *Controle Modular Local com Distinguidores e Aproximações* (CMLD-A) (TEIXEIRA et al., 2018) e os resultados são contextualizados na Tabela 7.

Observe que, em termos de complexidade, o pior caso enfrentado na síntese modular com auto-aproximações é para a especificação E_{d}^{1}. Nesse caso, o processo de síntese modular explora um modelo com 12 estados que, em comparação a versão monolítica com todo o modelo do distinguidor, com 192 estados (penúltima linha da Tabela 6), é

Tabela 6: Estatística da síntese para o sistema de manufatura com 2 retrabalhos

Síntese modular - estados (transições)					
${E_{\mathrm{d}}}^j \in \mathcal{E}_d$	$\mathcal{G}_{\mathrm{a}}^n$	$\mathcal{H}^n$	$\|\|(\mathcal{G}_{\mathrm{a}}^n)\| \|\|(\mathcal{H}^n)\| {E_{\mathrm{d}}}^j$	${S_{\mathrm{c}}^{E_{\mathrm{d}}}}^j$	${S_{\mathrm{c}}^{E_{\mathrm{d}}}}^j \| H_{\mathrm{d}}$
E_{d}^1	$G_{\mathrm{a}}^1, G_{\mathrm{a}}^3$	H_{d}^{Δ}	12 (65)	7 (32)	–
E_{d}^2	G_{a}^2	H_{d}^{Δ}	4 (32)	3 (22)	–
E_{d}^3	G_{a}^3	$H_{\mathrm{d}}^{\Delta}, H_{\mathrm{d}}^r$	4 (29)	3 (22)	–
$\|\|_{{E_{\mathrm{d}}}^j \in \mathcal{E}_d} {S_{\mathrm{c}}^{E_{\mathrm{d}}}}^j \| \|\|(\mathcal{G}_{\mathrm{a}})$				19 (42)	26 (45)
Síntese monolítica - estados (transições)					
$E_{\mathrm{d}}^1, E_{\mathrm{d}}^2, E_{\mathrm{d}}^3$	$G_{\mathrm{a}}^1, G_{\mathrm{a}}^2, G_{\mathrm{a}}^3$	$H_{\mathrm{d}}^c, H_{\mathrm{d}}^d, H_{\mathrm{d}}^e, H_{\mathrm{d}}^r$	192 (364)	–	26 (45)
$E_{\mathrm{d}}^1, E_{\mathrm{d}}^2, E_{\mathrm{d}}^3$	$G_{\mathrm{a}}^1, G_{\mathrm{a}}^2, G_{\mathrm{a}}^3$	H_{d}^r	64 (160)	19 (42)	26 (45)

Tabela 7: Estatística da síntese para o sistema de manufatura com 2 retrabalhos utilizando as abordagens de CML e CMLD-A

Síntese CML (QUEIROZ; CURY, 2000) - estados (transições)					
E^j	${\mathcal{G}^{\mathrm{j}}}_{loc}$		$\|\|({\mathcal{G}^{\mathrm{j}}}_{loc})\| E^j$		S^j_{loc}
E^1	G^1, G^2, G^3		24 (64)		14 (33)
E^2	G^2, G^3		12 (21)		9 (14)
E^3	G^2, G^3		42 (86)		32 (70)
$\|\|_{j=1}^3 S^j_{loc}$					26 (45)
Síntese CMLD-A (TEIXEIRA et al., 2018) - estados (transições)					
${E_{\mathrm{d}}}^j$	${\mathcal{G}_{\mathrm{a}}^j}_{loc}$	$\mathcal{H}'$	$\|\|({\mathcal{G}_{\mathrm{a}}^j}_{loc})\| \|\|(\mathcal{H}')\| {E_{\mathrm{d}}}^j$	${S_{\mathrm{c}}^j}_{loc}$	${S_{\mathrm{c}}^j}_{loc} \| H_{\mathrm{d}}$
E_{d}^3	$G_{\mathrm{a}}^2, G_{\mathrm{a}}^3$	H_{d}^r	8 (26)	6 (20)	32 (70)
$\|\|_{j=1}^2 S^j_{loc} \parallel \Pi({S_{\mathrm{c}}^3}_{loc} \| H_{\mathrm{d}})$					26 (45)

16 vezes menor. Já em relação à versão monolítica com auto-aproximação (última linha da Tabela 6) a redução é de 64 para 12 estados, o que corresponde a uma redução de aproximadamente 5 vezes.

Por outro lado, essa redução não é tão acentuada quando comparada aos resultados obtidos por meio das abordagens do CML e CMLD-A. No caso da síntese com as especificações E^1 e E_{d}^1 (terceira linha das Tabelas 6 e 7) a redução é de 24 para 12 estados, enquanto que para E^2 e E_{d}^2 (quarta linha das Tabelas 6 e 7) é de 12 para 4 estados. Já para o caso de E^3 e E_{d}^3 (quinta linha das Tabelas 6 e 7) é de 42 para 4 estados, uma redução de aproximadamente 10 vezes. Por fim, no caso em que E_{d}^3 é utilizada para a síntese em ambas as abordagens, CMLD-A (penúltima linha da Tabela 7) e com auto-aproximações (penúltima linha da Tabela 6), a redução é de 8 para 4 estados.

Note que, embora a diferença na complexidade enfrentada pela operação de síntese seja pequena em relação ao CMLD-A, esse trabalho possui vantagens adicionais de apresenta um algoritmo que seleciona automaticamente os distinguidores essenciais para síntese, ao passo que no CMLD-A essa tarefa é empírica que, adicionalmente, exige uma

verificação pós-síntese. Ademais, o mecanismo de seleção das sub-plantas utilizado neste trabalho se sobressai em relação ao do CMLD-A, pois seleciona somente os elementos necessários da planta para síntese independentemente do sincronismo entre eles, enquanto que no CMLD-A a seleção é conduzida com base no sincronismo de eventos, o que pode resultar na seleção de mais elementos do que o necessário.

Outro fato que vale ser destacado acerca deste exemplo é que, embora ele sirva para o propósito de ilustrar a abordagem proposta nesta dissertação, ele não reflete de fato o verdadeiro potencial de redução da complexidade na operação de síntese, uma vez que tal exemplo é uma versão parcial de um sistema industrial real onde podem existir outros subsistemas e especificações. Dessa forma, estima-se que a ordem de redução da complexidade na operação síntese provida pela abordagem modular desta dissertação tenda a ser cada vez maior, conforme o tamanho do sistema aumente.

6 CONCLUSÃO

Esse trabalho se propôs a desenvolver mecanismos que permitam explorar o uso de refinamento de eventos e aproximações para prover benefícios computacionais às etapas de síntese e implementação de controladores para SEDs. Para isso, foram desenvolvidos três métodos que aprimoram as fases de construção de aproximações, de processamento da operação de síntese e de implementação de controladores.

O primeiro método proposto, fundamentou-se no fato de que somente alguns eventos precisam ser distinguidos na síntese. Então, a partir disso foi desenvolvido um algoritmo para identificar tais eventos, e selecionar quais partes do mecanismos de distinção são necessárias para a construção de uma classe particular de aproximações, denominadas de auto-aproximações (ROSA et al., 2018b, 2018a). Foi demonstrado que as auto-aproximações detêm a propriedade de sempre levarem ao controlador ótimo, ao mesmo tempo em que mantêm importantes ganhos computacionais no processamento da operação de síntese. A principal vantagem desse método em relação aos existentes na literatura (AGUIAR et al., 2013; CURY et al., 2015) é que, além de sempre resultar na aproximação que utiliza o mínimo de partes do distinguidor na sua construção, ele também dispensa a necessidade da verificação pós-síntese. No entanto, ele está suscetível ao problema de explosão do número de estados, uma vez que ele visita cada estado do modelo que representa a composição entre planta e especificações.

Nesse sentido, foi apresentado um método de síntese que estende o mecanismo de construção de auto-aproximações para o contexto modular. Nesse método, as partes da planta, quando necessárias, são selecionadas incrementalmente de acordo com um critério baseado no conceito de controlabilidade, para cada uma das especificações. Assim, tanto o processo de seleção das partes do distinguidor quanto à operação de síntese do controlador são processados sobre um modelo que, em tese, é bem menor em termos de número de estados do que a versão monolítica do problema de controle. Além disso, pode-se demonstrar que a abordagem modular proposta, neste trabalho, se destaca em relação ao CMLD-A (TEIXEIRA et al., 2018) por dois motivos: i) possuir um mecanismo de seleção

de elementos da planta mais eficiente; ii) integrar o mecanismo de seleção para as partes do distinguidor.

O terceiro método apresentado neste trabalho descreveu uma arquitetura descentralizada que permite que distinguidor e controlador aproximado sejam implementados em duas estruturas que além de executarem paralelamente também se comunicam entre si (ROSA et al., 2017). Dessa forma, pode-se demonstrar que o mesmo efeito da ação de controle proveniente da composição entre essas estruturas, distinguidor e controlador, pode ser reproduzido em tempo de execução pela troca de informação. O principal benefício dessa arquitetura é que os ganhos obtidos na operação de síntese monolítica pelo uso das auto-aproximações podem ser inteiramente propagados para a fase de implementação. Porém, quando aplicada ao caso da solução de controle computada modularmente os ganhos obtidos são equivalentes ao caso da solução monolítica, uma vez que os supervisores modulares são compostos para o fim de implementação.

As perspectivas futuras deste trabalho apontam para a possibilidade de considerar a marcação de estados, bem como de estender a abordagem de implementação para que os ganhos obtidos na fase de síntese modular sejam utilizados integralmente.

REFERÊNCIAS

ÅKESSON, K.; FLORDAL, H.; FABIAN, M. Exploiting modularity for synthesis and verification of supervisors. **IFAC Proceedings Volumes**, v. 35, n. 1, p. 175 – 180, 2002.

AGUIAR, R. S. S. et al. Heuristic search of supervisors by approximated distinguishers. In: **Dependable Control of Discrete Systems (DCDS'13)**. York, UK: [s.n.], 2013. p. 121–126.

BOUZON, G.; QUEIROZ, M. H. de; CURY, J. E. R. Supervisory control of des with distinguishing sensors. In: **International Workshop on Discrete Event Systems, WODES'08**. Gothenburg, Sweden: [s.n.], 2008. p. 22–27.

BOUZON, G.; QUEIROZ, M. H. de; CURY, J. E. R. Exploiting distinguishing sensors in supervisory control of des. In: **Int. Conf. on Control and Automation, ICCA'09**. Christchurch, NZ: [s.n.], 2009. p. 442–447.

BRANDIN, B. A.; MALIK, R.; MALIK, P. Incremental verification and synthesis of discrete-event systems guided by counter examples. **IEEE Transactions on Control Systems Technology**, v. 12, n. 3, p. 387–401, May 2004. ISSN 1063-6536.

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

CUNHA, A. E. C. da; CURY, J. E. R. Hierarchical supervisory control based on discrete event systems with flexible marking. **IEEE Transactions Automatic Control**, v. 52, p. 2242–2253, Dec. 2007.

CURY, J. E. R. et al. Supervisory control of discrete event systems with distinguishers. **Automatica**, v. 56, p. 93 – 104, 2015.

INSTRUMENTS, T. 2018. Disponível em: <http://www.ti.com/>.

MALIK, R.; TEIXEIRA, M. Modular supervisor synthesis for extended finite-state machines subject to controllability. In: **Workshop on Discrete Event Systems (WODES'16)**. [S.l.: s.n.], 2016. p. 91–96.

OGATA, K. **Engenharia de controle moderno**. 5. ed. São Paulo: Pearson Prentice Hall, 2010.

QUEIROZ, M. H. D.; CURY, J. E. R. Modular supervisory control of large scale discrete event systems. In: **Discrete Event Systems: Analysis and Control. Proc. WODES'00**. [S.l.]: Kluwer Academic, 2000. p. 103–110.

RAMADGE, P. J.; WONHAM, W. M. The control of discrete event systems. **Discrete Event Dynamic Systems**, v. 77, p. 81–98, 1989.

ROSA, M. **Decentralized Implementation of Distinguished Controllers**. [S.l.], 2017. Disponível em: <https://github.com/marcellorosa/dcds2017>.

ROSA, M. et al. Efficient implementation of distinguished controllers for discrete-event systems. **IFAC-PapersOnLine**, Elsevier, v. 50, n. 1, p. 1187–1192, 2017.

ROSA, M.; TEIXEIRA, M.; MALIK, R. Controle supervisório de sistemas a eventos discretos com auto-aproximações. In: **Congresso Brasileiro de Automática**. João Pessoa, Brasil: [s.n.], 2018.

ROSA, M.; TEIXEIRA, M.; MALIK, R. Exploiting approximations in supervisory control with distinguishers. In: **International Workshop on Discrete Event Systems**. Sorrento, Italy: [s.n.], 2018.

TEIXEIRA, M. **Explorando o uso de distinguidores e de autômatos finitos estendidos na teoria do controle supervisório de sistemas a eventos discretos**. 137 p. Tese (Doutorado) — Universidade Federal de Santa Catarina, 2013.

TEIXEIRA, M.; CURY, J. E. R.; QUEIROZ, M. H. de. Local modular control with distinguishers applied to a manufacturing system. In: **Int. Conf. on Manufacturing Modelling, Management and Control**. [S.l.: s.n.], 2013.

TEIXEIRA, M.; CURY, J. E. R.; QUEIROZ, M. H. de. Exploiting Distinguishers in Local Modular Control of Discrete-Event Systems. **IEEE Transactions on Automation Science and Engineering**, p. 1–7, 2018. ISSN 1545-5955.

TEIXEIRA, M. et al. Supervisory control of des with extended finite-state machines and variable abstraction. **IEEE Transactions on Automatic Control**, v. 60, n. 1, p. 118–129, 2014.

APÊNDICE A – CONSTRUÇÃO DE UM DISTINGUIDOR

Uma característica importante de um distinguidor é a sua capacidade para identificar (predizer) sempre e exatamente um único refinamento (significado) para cada instância do evento no sistema. Essa exatidão permite preservar as características de segurança e máxima permissividade do sistema quando em operação controlada. No entanto, preditibilidade é um conveniente, mas forte, presunção em controle supervisório com distinguidores. Garantir tal propriedade, exige que o distinguidor filtre univocamente todas as cadeias do sistema, o que geralmente está associado a um grande modelo de espaço de estados.

Tornar D preditível é uma tarefa de modelagem que, em geral, pode ser feita de forma modular e depende da capacidade do engenheiro/projetista. Com intuito de auxiliar nessa tarefa em (CURY et al., 2015), uma vez que ainda não se pode automatizá-la completamente, é estabelecido um guia para a construção de um distinguidor preditível. Esse guia descreve os seguintes passos:

(i) *Identificação de um conjunto inicial de eventos $\sigma \in \Sigma$ a serem refinados*: identificar quais eventos em Σ, quando refinados, poderiam simplificar a modelagem de uma especificação de controle;

(ii) *Definição das instâncias de refinamentos $\delta \in \Delta^{\sigma}$*: a partir do passo anterior, deve-se determinar quais instâncias devem ser associadas a cada um dos eventos selecionados para o refinamento. Tais instâncias representam diferentes circunstâncias que o evento original pode ocorrer. Assim, cada instância também carrega uma semântica particular, a qual deve ser implementado pelo modelo do distinguidor;

(iii) *Complementação do conjunto Δ*: a partir de um dado conjunto de eventos a serem refinados (máscaras) e seus respectivos conjuntos de instâncias (refinamentos), a construção de um modelo que distingue a ocorrência de tais instâncias pode depender de outros refinamentos. De fato, o significado de uma certa instância de um evento pode se concretizar somente quando o sentido de outro evento é conhecido.

Então, tal evento deve ser refinado e suas instâncias serem distinguidas. Logo, é necessário revisitar os passos (i) e (ii), e assim definir corretamente o alfabeto Δ;

(iv) *Modelagem do distinguidor*: neste passo, se constrói um modelo que estabelece as interdependências entre as instâncias de refinamentos em Δ. Esse processo consiste na definição de um autômato H_{d}, tal que $L(H_{\mathrm{d}}) = L_{\mathrm{d}}$. O modelo H_{d}, é naturalmente simples no aspecto de modelagem e, em geral, pode ser construído de forma modular.

APÊNDICE B – SISTEMA DE MANUFATURA COM *BUFFER*

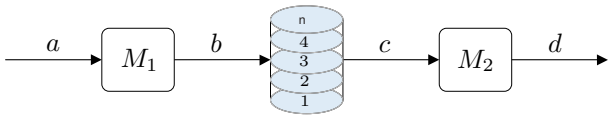
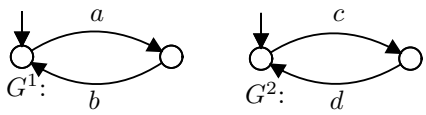


Figura 35: Exemplo de um SED

Considere como exemplo de um SED uma pequena fábrica (Figura 35) composta por duas máquinas, M_1 e M_2, operando de maneira sequencial, interligadas por um buffer B capaz de suportar a estocagem de até n peças empilhadas. Os autômatos G^1 e G^2 da Figura 8 modelam M_1 and M_2.



(a) G^1 (b) G^2

Figura 36: Modelos para M_1 e M_2

Os eventos a e c modelam o inicio de M_1 e M_2, respectivamente, enquanto b e d modelam sua conclusão. Foi assumido para $\Sigma = \{a, b, c, d\}$ que $\Sigma_{\mathrm{c}} = \{a, c\}$ e $\Sigma_{\mathrm{u}} = \{b, d\}$. O comportamento da planata em malha aberta é representado por $G = G^1 \| G^2$.

O objetivo de controle é evitar o *underflow* e *overflow* em B. Isso pode ser expressado pelo simples mapeamento de todas as combinações possíveis de b e c, que levaria a um autômato E com $n + 1$ estados, para um *buffer* de tamanho n, como apresentado na Figura 9.

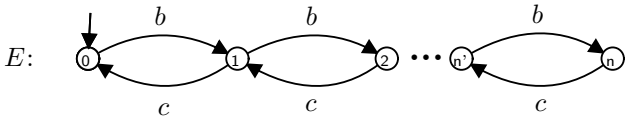


Figura 37: Modelo E

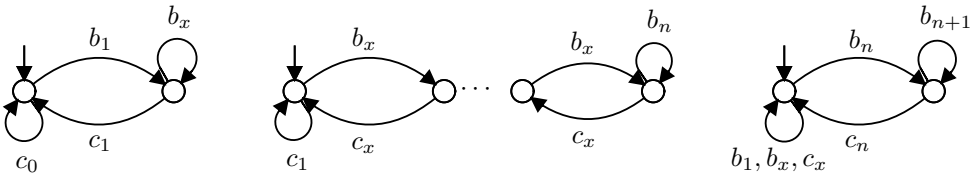
Apesar de E ser simples em termos de esforço de modelagem, sua implementação pode ser cara em termos de memória, pois depende do tamanho de B. Uma alternativa para projetar um modelo mais simples que expresse os mesmos requisitos de controle é dado pelo refinamentos dos eventos b e c de modo a identificar as posições do *buffer*.

Seja $\Delta^b = \{b_1, b_x, b_n, b_{n+1}\}$ e $\Delta^c = \{c_0, c_1, c_x, c_n\}$ tal que c_0 e b_{n+1} representam o *underflow* e *overflow*, respectivamente, enquanto os outros eventos representam as posições intermediárias. Seja também $\Delta^a = \{a\}$ e $\Delta^d = \{d\}$, tal que a versão refinada de Σ é dada por $\Delta = \Delta^a \cup \Delta^b \cup \Delta^c \cup \Delta^d$. Utilizando as informações em Δ, E pode ser representado pelos modelos E_{d}^U e E_{d}^O da Figura 38 que independem do tamanho de B.

(a) E_{d}^U (b) E_{d}^0

Figura 38: Modelos de E_{d}^U e E_{d}^O

Resta mostrar como os eventos em Δ poderiam ser distinguidos de tal forma que cada um expresse um significado particular. Este papel é desempenhado pelo modelo H_{d} apresentado na Figura 39, com $\beta = \{b_1, b_x, c_x\}$.



(a) H_{d}^1 (b) H_{d}^x (c) H_{d}^n

Figura 39: Modelo do distinguidor $H_{\mathrm{d}} = H_{\mathrm{d}}^1 \,||\, H_{\mathrm{d}}^x \,||\, H_{\mathrm{d}}^n$

Agora, a planta do sistema é modelada por um autômato G_{d}, tal que $G_{\mathrm{d}} = \Pi^{-1}(G) \| H_{\mathrm{d}}$. Enquanto a especificação em Δ é dada por $E_{\mathrm{d}} = E_{\mathrm{d}}^U \| E_{\mathrm{d}}^O$. Para $K_{\mathrm{d}} = L(G_{\mathrm{d}} \| E_{\mathrm{d}}) = L(G^{K_{\mathrm{d}}})$, o controlador ótimo é obtido por $\sup\mathcal{C}(K_d, G_d)$, o qual é implementado pelo autômato V_{d}. Além disso, o mesmo problema de controle foi resolvido utilizando uma auto-aproximação $\mathcal{G}_{\mathrm{c}} = \Pi^{-1}(G) \| H_{\mathrm{d}}^{\Delta} \| H_{\mathrm{d}}^n$. O controlador para o caso com auto-aproximação é dado por $\sup\mathcal{C}(\mathcal{K}_{\mathrm{c}}, \mathcal{G}_{\mathrm{c}}) = L(V_{\mathrm{a}})$, para $\mathcal{K}_{\mathrm{c}} = L(\mathcal{G}_{\mathrm{c}} \| E_{\mathrm{d}}) = L(G^{\mathcal{K}_{\mathrm{c}}})$. A Tabela 8 contextualiza os modelos utilizados na síntese do controladores.

Em termos de espaço de estados, $\mathcal{G}_{\mathrm{c}}$ é mais simples que G_{d}, visto que somente partes do distinguidor foram utilizadas na construção de $\mathcal{G}_{\mathrm{c}}$. Consequentemente, a síntese

Tabela 8: Resultados da síntese - estados (transições)

G_{d}	E_{d}	$G^{K_{\mathrm{d}}}$	V_{d}
$4n+4$ $(8n+8)$	4 (20)	$4n+4$ $(8n+4)$	$4n+2$ $(8n)$
$\mathcal{G}_{\mathrm{c}}$	E_{d}	$G^{K_{\mathrm{c}}}$	V_{a}
8 (40)	4 (20)	16 (72)	12 (47)

de V_{a}, usando $\mathcal{G}_{\mathrm{c}} \| E_{\mathrm{d}}$, é mais simples do que de V_{d}.

Em termos de números de transições, a síntese de V_{a} não depende do tamanho do *buffer*. Na verdade, o tamanho do *buffer* é associado ao modelo H_{d} que foi removido da síntese. Portanto, o supervisor V_{a} será o mesmo e pode ser computado com o mesmo esforço computacional para qualquer tamanho de *buffer* (CURY et al., 2015).