UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

LUIZ FERNANDO PUTTOW SOUTHIER

CONVERSÃO ENTRE MODELOS DE SISTEMAS DE AUTOMAÇÃO A EVENTOS DISCRETOS PARA SIMPLIFICAÇÃO DA SÍNTESE DE CONTROLADORES

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO
2017

LUIZ FERNANDO PUTTOW SOUTHIER

CONVERSÃO ENTRE MODELOS DE SISTEMAS DE AUTOMAÇÃO A EVENTOS DISCRETOS PARA SIMPLIFICAÇÃO DA SÍNTESE DE CONTROLADORES

Trabalho de Conclusão de Curso 1, apresentado à disciplina de Trabalho de Conclusão de Curso 1, do Curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná - UTFPR, Câmpus Pato Branco, como requisito parcial para obtenção do título de Engenheiro da Computação.

Orientador: Prof. Dr. Marcelo Teixeira

PATO BRANCO

2017

TERMO DE APROVAÇÃO

O Trabalho de Conclusão de Curso intitulado **CONVERSÃO ENTRE MODELOS DE SISTEMAS DE AUTOMAÇÃO A EVENTOS DISCRETOS PARA SIMPLIFICAÇÃO DA SÍNTESE DE CONTROLADORES** do acadêmico **Luiz Fernando Puttow Southier** foi considerado **APROVADO** de acordo com a ata da banca examinadora **N°** de **2017**.

Fizeram parte da banca examinadora os professores:

Prof. Dr. Marcelo Teixeira

Prof. Dr. César Rafael Claure Torrico

Prof Dr. Marco Antônio de Castro Barbosa

LISTA DE FIGURAS

LISTA DE ACRÔNIMOS E SIGLAS

AF	Autômato Finito.
AFD	Autômato Finito Determinístico.
AFEs	Autômatos Finitos Estendidos.
CML	Controle Modular Local.
SED	Sistema a Evento Discreto.
TCS	Teoria do Controle Supervisório.
TLF	Teoria das Linguagens Formais.

LISTA DE SÍMBOLOS

Σ	Alfabeto
w	Cadeia
$\|w\|$	Comprimento de uma cadeia w
ϵ	Cadeia Vazia
Σ^*	Conjunto de todas as cadeias sobre um alfabeto Σ
L	Linguagem
$\overline{L}$	Prefixo-fechamento da linguagem *L*
Q	Conjunto de estados
q°	Estado inicial
Q^ω	Conjunto de estados marcados
δ	Função de transição
$\mathcal{L}(A)$	Linguagem Gerada por um autômato A
$\mathcal{L}^\omega(A)$	Linguagem Marcada por um autômato A
$A_1 \,\|\, A_2$	Composição Síncrona
G	Modelo da planta
E	Modelo das especificações
S	Supervisor
$\sup\mathcal{C}$	Elemento supremo para controle supervisório
Δ	Alfabeto refinado
Π	Mapa mascarador
D	Distinguidor
H_d	Modelo para distinção da planta refinada
V	Conjunto de variáveis
Q°	Conjunto de estados iniciais

SUMÁRIO

1 INTRODUÇÃO

Apresenta-se neste capítulo uma perspectiva geral sobre o estado da arte no qual este trabalho está contextualizado. Descreve-se, também, a problematização do assunto abordado, bem como os objetivos e sua correspondente justificativa de pesquisa destacando a importância deste tema. Expõe-se, ainda, a estrutura adotada no trabalho.

1.1 CONSIDERAÇÕES INICIAIS

Sistemas a Eventos Discretos (SED) são sistemas que se caracterizam por estarem sujeitos a eventos que ocorrem em intervalos de tempo irregulares ou desconhecidos. Exemplos de SEDs são a transmissão de um pacote em um sistema de comunicação, a finalização de uma tarefa ou a falha de uma máquina em um sistema de manufatura, por exemplo (CASSANDRAS; LAFORTUNE, 2009).

Uma das formas de modelar SEDs é por meio de Linguagens Formais e por Autômatos Finitos (AF) (HOPCROFT *et al.*, 1939). Lógicas de controle para SEDs podem, em geral, ser implementadas usando-se os mesmos formalismos utilizados na sua modelagem. A Teoria do Controle Supervisório (TCS), por exemplo, define um método formal para a síntese automática de supervisores ou controladores ótimos para SEDs (RAMADGE; WONHAM, 1987). A TCS, assim como a maioria das teorias de controle, tem por objetivo a síntese de controladores partindo de um modelo formal do sistema a ser controlado, chamado de planta, e os requisitos de comportamento do sistema controlado, chamados de especificações (THISTLE, 2004).

Embora bem fundamentada, a TCS enfrenta alguns problemas na prática que acabam limitando sua aplicabilidade na indústria. Por exemplo, para alguns problemas em particular, uma única especificação do sistema pode exigir monoliticamente combinar milhares de estados, tornando a modelagem inviável (TEIXEIRA, 2013). O conceito de distinguidor, proposto por Bouzon *et al.* (2008), Bouzon *et al.* (2009), tem por objetivo simplificar o processo de modelagem de especificações refinando o conjunto de eventos do modelo dos SEDs.

Outra abordagem de simplificação da complexidade de modelagem é o uso de Autômatos Finitos Estendidos (AFEs) investigados por Chen e Lin (2000), Chen e Lin (2001), Skoldstam *et al.* (2007) , Fei *et al.* (2012) e Cheng e Krishnakumar (1993). Os AFEs se caracterizam por monitorar um conjunto de elementos dos sistemas por meio de variáveis. Essas variáveis são atualizadas por fórmulas e expressões lógicas associadas aos eventos do modelo, fazendo com que a ação de controle seja direcionada pelo valor atual das variáveis.

1.2 PROBLEMÁTICA E JUSTIFICATIVA

Informalmente, Distinguidores e AFEs apresentam características semelhantes e têm por objetivo a resolução de problemas de natureza similar, ou seja, problemas para os quais a tarefa de construir um modelo é trabalhosa ou, em alguns casos, é inviável de ser conduzida. Então, a adoção dessas abordagens tende a tornar a tarefa de modelagem mais simples. O fundamento chave por trás dessa simplificação é o conceito de refinamento de informações sobre o sistema. Um refinamento é um mecanismo associado ou aos estados do modelo (no caso dos AFEs) ou às suas transições (no caso dos Distinguidores), que provê ao *designer* maiores detalhes sobre como o sistema se comporta, o que pode contribuir decisivamente na hora de expressar um requisito para o sistema.

Além dos benefícios de modelagem apresentados, as abordagens por Distinguidores e AFEs são ainda associados na literatura a alternativas avançadas de síntese, capazes de levar ao cálculo de controladores ótimos, de maneira mais simples do que a convencional, com menor dispêndio computacional. Tais alternativas caracterizam-se pelo uso de abstrações no caso dos AFEs (TEIXEIRA *et al.*, 2015) e aproximações no caso dos Distinguidores (CURY *et al.*, 2015). Contudo, a revisão de literatura realizada indica que não existe uma relação formal e explícita entre as duas abordagens. Isso impacta negativamente no processo de modelagem e síntese, no sentido de que isso impede que as vantagens de ambas as abordagens possam ser combinadas na resolução de um mesmo problema. Ou, ainda, que possa ser feita uma comparação explícita de qual abordagem é mais adequada para cada tipo de problema de controle.

Os resultados existentes até o momento sugerem que a síntese de controle de SEDs utilizando Distinguidores é relativamente mais vantajosa do que a síntese com AFEs. De fato, ela facilita o processo de modelagem (BOUZON *et al.*, 2008), quando associada a aproximações reduz a complexidade de síntese (CURY *et al.*, 2015), e ela ainda possui suporte ao Controle Modular Local (CML) (TEIXEIRA *et al.*, 2013), fundamental para lidar com problemas industriais de grande porte (QUEIROZ; CURY, 2000). Como principal desvantagem em relação à síntese convencional, a abordagem com

Distinguidores impõe um *overhead* no sentido de que, nesse caso, se tem que construir um modelo adicional para distinguir a planta refinada do sistema, tarefa esta que não compõe o método convencional de síntese, é manual e pode ser complexa em alguns casos (FISCHER; LEAL, 2014).

Por outro lado, AFEs possuem a vantagem de serem integrados com uma estrutura de variáveis, o que gera benefícios de modelagem similares aos Distinguidores, além de ser mais natural sob o ponto de vista computacional. Além disso, a literatura associa abstrações ao controle supervisório com AFEs (TEIXEIRA *et al.*, 2015), o que gera benefícios computacionais similares ao uso de aproximações na síntese com Distinguidores. Como principal desvantagem, a abordagem com AFEs não possui suporte ao CML. Um método de modularização de síntese com AFEs foi recentemente proposto por Malik e Teixeira (2016). No entanto, a abordagem ainda é essencialmente teórica, não conta com ferramental de síntese, além de que, nessa abordagem, AFEs são explorados sem a ideia de abstrações e, portanto, não geram vantagens comparáveis à aproximações no CML. Ademais, a estrutura de variáveis não é diretamente associável às ferramentas de síntese e geração de código, requerendo um passo adicional para converter AFEs em autômatos ordinários que podem, então, ser tratados pelos algoritmos convencionais de síntese. Por fim, existe ainda a desvantagem de se ter que construir a estrutura lógica de atualização da planta que, embora pareça mais simples do que a modelagem de Distinguidores, não deixa de ser um esforço extra no processo de síntese.

Neste trabalho, argumenta-se que, dependendo da aplicação ou problema em que está sendo utilizado, o refinamento de eventos acaba por si só se tornando um problema de modelagem. Isso se deve ao fato de que a construção de uma estrutura de distinção precisa incorporar a ocorrência de cada instância de refinamento de acordo com o comportamento do sistema. No entanto, entende-se que, caso o mesmo problema seja modelado utilizando AFEs, o esforço de modelagem poderia ser resumido à implementação de uma base de fórmulas e expressões lógicas, ao invés da construção de autômatos. Fórmulas atualizam variáveis associando valores a instâncias do comportamento de um SED. Em tese, estima-se que essa construção seja mais simples e passível de automatização, ao passo que um modelo distinguidor continua dependendo puramente da modelagem de uma máquina de estados, tarefa que recai inevitavelmente sobre o engenheiro.

Em face ao exposto, o presente trabalho visa construir um método de conversão que possibilite combinar vantagens de ambas abordagens, por Distinguidores e por AFEs. A característica atrativa dos AFEs, de facilitar a modelagem de problemas complexos, será explorada durante a fase de *design* e, após obtidos os modelos, planeja-se convertê-los para modelos refinados para fins de síntese. O processo inverso de conversão também será realizado a fim de que, a partir de um modelo ordinário refinado, se possa obter a sua versão equivalente em AFE.

1.3 OBJETIVOS

A proposta tem seus objetivos divididos em objetivo geral, aludindo ao resultado final obtido depois do desenvolvimento do trabalho, e objetivos específicos, que são as etapas intermediárias para alcançar o objetivo geral descritos como se segue.

1.3.1 OBJETIVO GERAL

Desenvolver um método formal de conversão entre modelos de SEDs construídos utilizando AFEs e Distinguidores que expressem comportamento equivalente.

1.3.2 OBJETIVOS ESPECÍFICOS

- Desenvolver um método de conversão entre máquinas de estados;
- Modelar um exemplo de um SED através de Distinguidores e AFEs;
- Aplicar método de conversão sobre o exemplo;
- Sintetizar um controlador para um exemplo de um SED utilizando Distinguidores;
- Sintetizar um controlador para um exemplo de um SEDs utilizando AFEs;
- Sintetizar um controlador para um exemplo de um SED utilizando o método de conversão;
- Analisar e comparar os resultados obtidos com a abordagem inicial.

1.4 ESTRUTURA

O presente trabalho apresenta inicialmente as definições teóricas inerentes à modelagem de SEDs, tais como Teoria das Linguagens Formais e seu reconhecimento por Autômatos Finitos Determinísticos. Posteriormente, retrata os conceitos referentes ao controle de SEDs, por meio da Teoria do Controle Supervisório, sob as abordagens de Distinguidores e de AFEs. Prossegue-se indicando a proposta do presente trabalho, justificando sua importância.

2 REFERENCIAL TEÓRICO

Apresentam-se neste capítulo os fundamentos teóricos. Descreve-se a Teoria dos Autômatos, reforçando os conceitos centrais de Alfabetos, Cadeias, Linguagens e Operações. Conceitua-se, ainda, a teoria dos Autômatos Finitos Determinísticos e sua aplicação como reconhecedores de linguagens. Por fim, é apresentada a Teoria de Controle Supervisório, usada na síntese de controladores para SEDs, sob a abordagem clássica, por Distinguidores e por AFEs.

2.1 MODELAGEM DE SED

Ramadge e Wonham (1989) definem um SED como sendo um sistema dinâmico com um espaço de estados discreto e com trajetórias entre estados constantes. Os instantes de tempo em que as transições ocorrem, assim como a transição em si, em geral são imprevisíveis, contrastando-os com sistemas de dinâmica contínua no tempo. A Figura 1 apresenta um exemplo de transições (e_i) que ocorrem em determinados instantes de tempo (t_i) entre os estados (s_i) de um SED.

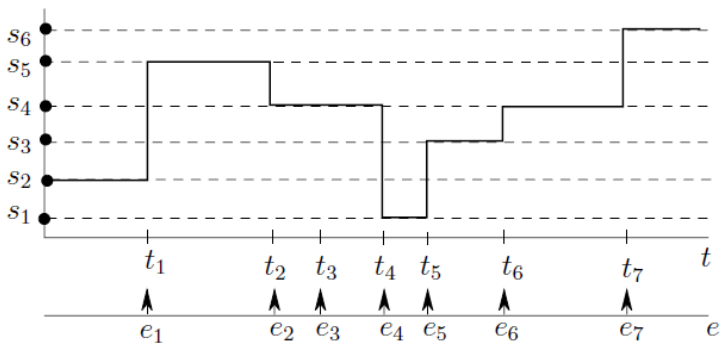

Figura 1: Exemplo de transições de estados de um SED

Fonte: Cassandras e Lafortune (2009).

Enquanto um sistema dirigido pelo tempo pode ser descrito por equações diferenciais, para Cassandras e Lafortune (2009) os SEDs são mais naturalmente modelados por estruturas de diagramação, como AFDs por exemplo. As próximas duas seções trazem conceitos referentes à *Teoria da Linguagens Formais* e aos *Autômatos Finitos Determinísticos* utilizados amplamente na modelagem de SEDs.

2.1.1 TEORIA DAS LINGUAGENS FORMAIS

De acordo com Hopcroft *et al.* (1939) as definições mais importantes que permeiam a Teoria das Linguagens Formais (TLF) consistem em Alfabetos, Cadeias e Linguagens e são definidas da seguinte maneira.

Um *alfabeto* pode ser definido como um conjunto finito e não vazio de símbolos, denotado por Σ. Uma *cadeia* w é uma sequência finita de símbolos pertencentes a um alfabeto, cujo comprimento é denotado por $|w|$. Uma *cadeia vazia* é uma cadeia com nenhuma ocorrência de símbolos, por isso pode ser formada a partir de qualquer alfabeto, e é representada por ϵ, tal que $|\epsilon| = 0$. O conjunto de todas as cadeias sobre um alfabeto Σ é convencionalmente denotado Σ^*.

Duas cadeias $\alpha = a_1a_2...a_i$ e $\beta = b_1b_2...b_j$ podem ser concatenadas em uma única cadeia $\gamma = a_1a_2...a_ib_1b_2...b_j$. Se a cadeia $s = xyz$, com $s, x, y, z \in \Sigma^*$, então x é um *prefixo* de s, y uma *subcadeia* de s e z é um *sufixo* de s (HOPCROFT *et al.*, 1939).

A partir de Σ^*, podem ser definidos subconjuntos, denominados *linguagem* (L), que incorporam cadeias reconhecidas em determinado contexto. Ou seja, para um dado alfabeto Σ, $L \subseteq \Sigma^*$, caracteriza L como uma linguagem sobre Σ cujas cadeias apresentam um significado em determinado contexto. Enquanto que as cadeias $s \in L$ são finitas, L pode ser um conjunto infinito de cadeias finitas.

Cassandras e Lafortune (2009) definem algumas operações sobre Linguagens, complementando as operações usuais de conjuntos, tais como união, interseção e diferença. Dentre elas destaca-se o *prefixo-fechamento* ($\overline{L}$) de uma linguagem L. Este pode ser definido como

$$\overline{L} = \{s \in \Sigma^* : \exists t \in \Sigma^* \text{ tal que } st \in L\} \tag{1}$$

Em palavras, o prefixo-fechamento de L é a linguagem que consiste de todos os prefixos de todas as cadeias pertencentes à L. L é dita prefixo-fechada se $L = \overline{L}$, ou seja, uma linguagem L é prefixo-fechada se todos os prefixos das cadeias de L também são elementos de L.

2.1.2 AF DETERMINÍSTICOS COMO RECONHECEDORES DE LINGUAGENS

Hopcroft *et al.* (1939) define um Autômato Finito Determinístico (AFD) como sendo a quíntupla $A = (\Sigma, Q, q^\circ, Q^\omega, \delta)$, onde:

- Σ é o alfabeto de *símbolos de entrada*;
- Q é um *conjunto finito de estados*;
- $q^{\circ} \in Q$ é o *estado inicial*;
- $Q^{\omega} \subseteq Q$ é o *conjunto de estados finais ou de aceitação*;
- δ: $Q \times \Sigma \rightarrow Q$ é uma *função de transição*, normalmente parcial em relação ao seu domínio, que tem como argumentos um estado e um símbolo de entrada e, como retorno, um estado. Na representação de um autômato por meio de um grafo direcionado, δ é representada por nós e transições identificadas. Considerando estados q e p, e um símbolo de entrada a, e que $q \xrightarrow{a} p$, então a representação por grafo de δ possui uma transição identificada por a de q para p (CASSANDRAS; LAFORTUNE, 2009).

Para reconhecimento de linguagens, Hopcroft *et al.* (1939) sugere a utilização de AFD, isto é, AFD são estruturas que podem ser utilizadas para reconhecer uma linguagem. Um AFD pode ser utilizado para decidir se uma cadeia é aceita ou não (pertence ou não a uma linguagem). A linguagem de um AFD é o conjunto de todas as cadeias reconhecidas por ele (HOPCROFT *et al.*, 1939).

A *Linguagem Gerada* $\mathcal{L}(A)$ por um autômato $A = (\Sigma, Q, q^{\circ}, Q^{\omega}, \delta)$ é definida por Cassandras e Lafortune (2009) como:

$$\mathcal{L}(A) = \{s \in \Sigma^{*} \text{ tal que } \delta\,(q^{\circ}, s) \text{ é definida } \} \tag{2}$$

A *Linguagem Marcada* $\mathcal{L}^{\omega}(A)$ por um autômato $A = (\Sigma, Q, q^{\circ}, Q^{\omega}, \delta)$ é definida por Cassandras e Lafortune (2009) como:

$$\mathcal{L}^{\omega}(A) = \{s \in \mathcal{L}\,(A) \text{ tal que } \delta\,(q^{\circ}, s) \in Q^{\omega}\} \tag{3}$$

À classe de linguagens que podem ser reconhecidas por um Autômato Finito (LEWIS; PAPADIMITRIOU, 1997), dá-se o nome de *linguagens regulares*. Na indústria, a regularidade é importante porque define as linguagens passíveis de processamento computacional, isto é, elas podem ser representadas por autômatos que ocupam memória finita quando armazenados em um computador (CASSANDRAS; LAFORTUNE, 2009).

No caso de um SED ser modelado por um autômato G, os eventos do sistema podem ser representados por um alfabeto Σ_G, tal que as sequências de eventos possíveis no sistema podem ser identificadas por $\mathcal{L}(G)$ e suas cadeias marcadas por $\mathcal{L}^{\omega}(G)$. Sendo assim, AFDs são uma ferramenta notória para expressar e modelar o comportamento de SEDs (RAMADGE; WONHAM, 1989).

2.1.2.1 COMPOSIÇÃO SÍNCRONA DE AFD

A *Composição Síncrona* é um método de junção dos comportamentos de um conjunto de AFDs que ocorrem simultaneamente. Cassandras e Lafortune (2009) definem a composição síncrona $A_1 \,||\, A_2$ entre os autômatos $A_1 = (\Sigma_1, Q_1, q_1^{\circ}, Q_1^{\omega}, \delta_1)$ e $A_2 = (\Sigma_2, Q_2, q_2^{\circ}, Q_2^{\omega}, \delta_2)$ como:

$$
\begin{aligned}
A_1 \,\big|\big|\, A_2 &= \left(\Sigma_1 \cup \Sigma_2, Q_1 \times Q_2, \left(q_1^{\circ}, q_2^{\circ}\right), Q_1^{\omega} \times Q_2^{\omega}, \delta_{1\|2}\right), \text{ tal que} \\
\delta_{1\|2}\left(\left(q_1, q_2\right), s\right) &= \begin{cases} (\delta_1(q_1, s), \delta_2(q_2, s)) \text{ se } s \in \Sigma_1 \cap \Sigma_2 \\ (\delta_1(q_1, s), q_2) \text{ se } s \in \Sigma_1 \setminus \Sigma_2 \\ (q_1, \delta_2(q_2, s)) \text{ se } s \in \Sigma_2 \setminus \Sigma_1 \\ \text{indefinido para outros casos} \end{cases}
\end{aligned} \tag{4}
$$

Ao analisar dois autômatos que representam os comportamentos de dois processos de um SED é possível utilizar essa operação para combinar esses comportamentos em um novo autômato, de tal modo que os eventos compartilhados ocorram simultaneamente e os demais sejam intercalados em qualquer ordem (TEIXEIRA, 2013).

2.2 CONTROLE DE SED

A partir da operação de composição, se torna possível que os componentes de um SED, também chamados de subsistemas, sejam modelados individualmente e, em tese, de forma mais simples. Espera-se que o resultado da composição das partes leve a uma estrutura equivalente à obtida caso o sistema fosse modelado como um todo. O modelo que reflete a execução dos elementos do sistema em conjunto é conhecido como *planta* (CASSANDRAS; LAFORTUNE, 2009), que é dita estar em *malha aberta* quando nenhuma interferência, ação de controle ou coordenação externa atua em seu comportamento.

Na prática, uma planta em malha aberta deve ser restrita a determinado comportamento esperado para quando estiver em funcionamento. Dessa necessidade nasce o conceito de *especificação*. Uma especificação pode ser entendida como a representação de um ato proibitivo que, quando associada à planta, interfere em determinados eventos possíveis em malha aberta, de modo que o comportamento resultante atenda as condições de controle propostas. Ao comportamento de uma planta restrito por um conjunto de especificações, dá-se o nome de *malha fechada*.

Se a planta de um SED é modelada por um conjunto de autômatos $G^j,\ j = 1, \cdots, n$, então a composição síncrona $G = \|_{j=1}^{n} G^j$ representa o modelo do sistema em malha aberta. Da mesma forma, podem ser combinados os m autômatos E^i que representam a especificação $E = \|_{i=1}^{m} E^i$. Combinando planta e especificações obtêm-se a estrutura $K = G \| E$, em que $\mathcal{L}^\omega(K) \subseteq \mathcal{L}^\omega(G)$ correspondendo ao comportamento desejado da planta sob a ação das especificações.

Pela própria definição de composição síncrona, se um evento possível na planta G é desabilitado pela especificação E, então o resultado é uma ação que desabilita esse evento em G. Logo, o próprio modelo K pode ser utilizado para realizar o controle da planta G de um SED, desde que se assuma que todos os eventos em Σ são passíveis de desabilitação externa.

Na prática, porém, existem eventos que não podem ser diretamente controlados como, por exemplo, o evento de quebra ou parada de um equipamento, o evento de recepção de um pacote em uma rede, etc. A natureza da ocorrência desse tipo de evento é não-controlável e, portanto, a estrutura que regula a habilitação ou não desses eventos na planta precisa incorporar essa característica. Uma alternativa formal para o controle adequado de SEDs é apresentada na sequência.

2.2.1 TEORIA DE CONTROLE SUPERVISÓRIO

A *Teoria de Controle Supervisório* (TCS) é um formalismo que define a síntese de supervisores (ou controladores) ótimos para SEDs. Um supervisor é ótimo se ele é controlável, minimamente restritivo e não-bloqueante. Para permitir o cálculo de tal supervisor, considerando o aspecto de controlabilidade, a TCS particiona o conjunto de eventos $\Sigma = \Sigma_{\mathrm{c}} \cup \Sigma_{\mathrm{u}}$, onde Σ_{c} é o conjunto de eventos controláveis, i.e., aqueles que podem ser desabilitados, e Σ_{u} é o conjunto de eventos não controláveis, i.e., que não podem ser desabilitados diretamente (RAMADGE; WONHAM, 1987; RAMADGE; WONHAM, 1989).

A partir desse particionamento, a TCS define a operação que calcula o *supervisor*, a entidade que efetivamente implementa a ação de controle na planta. Formalmente, um supervisor S é um mapa $S : \mathcal{L}(G) \rightarrow 2^{\Sigma_c}$ associado a uma linguagem $M \subseteq \mathcal{L}^\omega(G)$ que, após qualquer cadeia s em G, desabilita eventos controláveis $S(s) \subseteq \Sigma_c$ e marca cadeias de acordo com $\mathcal{L}^\omega(S/G) = \mathcal{L}(S/G) \cap M$ (BOUZON *et al.*, 2008; CASSANDRAS; LAFORTUNE, 2009; RAMADGE; WONHAM, 1989).

O cálculo de S passa pelo conceito de *controlabilidade*. Uma linguagem $K \subseteq \Sigma^*$ é controlável em relação à L se $\overline{K}\Sigma_u \cap L \subseteq \overline{K}$, ou seja, após qualquer prefixo de K, se um evento não-controlável é possível em L, a cadeia resultante continua em $\overline{K}$. Um supervisor marcador não-bloqueante S possui $\mathcal{L}^\omega(S/G) = K$ e pode ser implementado por qualquer autômato V em que $K = \mathcal{L}^\omega(V) \cap \mathcal{L}^\omega(G)$ e $\overline{K} = \mathcal{L}(V) \cap \mathcal{L}(G)$. O controle ocorre por meio de S que desabilita eventos possíveis em $\mathcal{L}(G)$ que não são possíveis em $\mathcal{L}(V)$ após uma cadeia $s \in \mathcal{L}(S/G)$ (BOUZON *et al.*, 2009).

Se K não é controlável, então se torna necessário calcular a sua máxima linguagem controlável, ou seja, o subcomportamento que mais se aproxima de K e, ao mesmo tempo, é controlável. Considerando $\mathcal{C}(K, G)$ como o conjunto das linguagens controláveis de K em relação a G em que $L \subseteq K$ tal que L é controlável em relação à $\mathcal{L}(G)$, $\mathcal{C}(K, G)$ possui um único elemento supremo, denotado por $\sup\mathcal{C}(K, G)$, que expressa o comportamento menos restritivo possível a ser implementado por controle supervisório sobre a planta, respeitando o conjunto de especificações.

2.2.1.1 EXEMPLO: TCS

Com o objetivo de exemplificar a utilização da TCS, e suas limitações sob a abordagem clássica, é apresentado a seguir um problema simples, proposto por Teixeira (2013), que posteriormente será discutido sob as abordagens de Distinguidores e AFEs.

Considera-se para este exemplo uma máquina M que está inicialmente desligada e entra em funcionamento após um evento a, que representa a chegada de uma peça ou a disponibilidade de um recurso, por exemplo. Supõe-se que essa máquina precisa ser colocada em manutenção após a ocorrência de três eventos a.

É possível modelar M utilizando os conceitos anteriormente apresentados. Dessa modelagem obtêm-se os autômatos G e E apresentados na Figura 2. G modela o comportamento da planta, sempre está habilitada a receber um evento a, enquanto que E modela o comportamento da especificação ou restrição apresentada, a máquina precisa entrar em manutenção após a ocorrência de três eventos a.

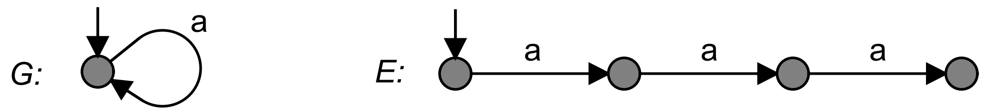

Figura 2: Modelagem do SED sob a abordagem clássica

A TCS enfrenta problemas na prática que limitam sua aplicação diretamente na indústria. Ao considerar não mais três, mas sim 100000 ocorrências do evento a no problema acima, por exemplo, é interessante observar que a especificação E alcançaria um número de estados consideravelmente grande, forçando o desenvolvedor a combinar monoliticamente esses milhares de estados. Se ainda, em conjunto com esses eventos, o desenvolvedor tiver que considerar eventos concorrentes, a complexidade aumenta de uma maneira a tornar o problema intratável, como mostrado em Teixeira (2013), Cury *et al.* (2015), Bouzon *et al.* (2009), Bouzon *et al.* (2008).

Considerando que este exemplo retrata apenas uma fração singela do SEDs em problemas práticos, e que ele possui apenas uma especificação, é possível inferir que para um problema real a modelagem utilizando essa abordagem muitas vezes se torna inviável. Com isso, na literatura propõem-se abordagens para lidar com esse tipo de característica da TCS. Dentre as abordagens, as próximas duas seções tratam da abordagem por Distinguidores e da abordagem por AFEs.

2.2.2 TCS COM DISTINGUIDORES

Esta seção apresenta os fundamentos do conceito de Distinguidores, proposto por Bouzon *et al.* (2008), Bouzon *et al.* (2009), que visa a simplificação do processo de modelagem de especificações identificando e distinguindo ocorrências de um evento no sistema. Uma característica interessante a se destacar é que a abordagem por Distinguidores proporciona uma solução de controle equivalente à obtida pela TCS clássica.

Seja G um autômato que modela um SED com eventos em Σ. Assume-se que cada elemento $\sigma \in \Sigma$ corresponde a uma máscara para eventos em um conjunto $\Delta^\sigma \neq \emptyset$ que refina σ (TEIXEIRA, 2013). Cada máscara está associada a um conjunto próprio de refinamentos, diferente dos outros e não vazio. Dessa forma, Σ passa a ser um conjunto de máscaras para eventos de um alfabeto refinado $\Delta = \bigcup_{\sigma\in\Sigma} \Delta^\sigma$. A estrutura de controle é definida no alfabeto refinado por $\Delta = \Delta_\mathrm{c} \cup \Delta_\mathrm{u}$, em que $\Delta_\mathrm{u} = \bigcup_{\sigma\in\Sigma_\mathrm{u}} \Delta^\sigma$ e $\Delta_\mathrm{c} = \bigcup_{\sigma\in\Sigma_\mathrm{c}} \Delta^\sigma$.

A relação entre Σ e Δ pode ser definida por um mapa mascarador Π: $\Delta^* \to \Sigma^*$ definido por Bouzon *et al.* (2008) a seguir:

$$\begin{aligned} &\Pi(\epsilon) = \epsilon \\ &\Pi(t\delta) = \Pi(t)\sigma \text{ para } t \in \Sigma^*, \delta \in \Delta^\sigma \text{ e } \sigma \in \Sigma \end{aligned} \tag{5}$$

O mapa mascarador Π pode ser estendido para qualquer linguagem $L_\Delta \subseteq \Delta^*$ por $\Pi(L_\Delta) = \{s \in \Sigma^* | \exists t \in L_\Delta, \Pi(t) = s\}$. Da mesma forma, por ser definido também o mapa mascarador inverso $\Pi^{-1} : \Sigma^* \to 2^{\Delta^*}$ como sendo $\Pi^{-1}(s) = \{t \in \Delta^* | \Pi(t) = s\}$ e que pode ser estendido sobre uma linguagem $\Pi^{-1}(L) = \{t \in \Delta^* | \Pi(t) \in L\}$.

Por fim, dada uma linguagem $L \subseteq \Sigma^*$, um distinguidor para L pode ser definido como um mapa D: $\Sigma^* \to 2^{\Delta^*}$ tal que

$$D(L) = \Pi^{-1}(L) \cap L_{\mathrm{d}} \tag{6}$$

em que L_{d} é uma linguagem distinguidora. Dessa forma, o efeito de D sobre L é tal que ele mapeia cada cadeia de L em todas as possíveis cadeias refinadas ($\Pi^{-1}(L)$), e filtra quais delas devem de fato ocorrer (L_{d}). Pode-se notar que, no caso dos Distinguidores, resta definir um autômato complementar que reconheça L_{d}.

Considerando um SED, cujo comportamento é modelado por autômato G, o efeito de um distinguidor D sobre G é dado por:

$$\begin{aligned} D(\mathcal{L}(G)) &= \Pi^{-1}(\mathcal{L}(G)) \cap L_{\mathrm{d}} \\ D(\mathcal{L}^\omega(G)) &= \Pi^{-1}(\mathcal{L}^\omega(G)) \cap L_{\mathrm{d}} \end{aligned} \tag{7}$$

Nesse contexto, seja G_{d} um autômato tal que $\mathcal{L}(G_{\mathrm{d}}) = D(\mathcal{L}(G))$ e $\mathcal{L}^\omega(G_{\mathrm{d}}) = D(\mathcal{L}^\omega(G))$, e seja E_{d} uma especificação definida em Δ. Os requisitos expressos por E_{d} usando as informações adicionais providas pelo distinguidor devem se referir aos mesmos requisitos expressos por E em Σ, tal que $E_{\mathrm{d}} \cap \mathcal{L}^\omega(G_{\mathrm{d}}) = K_{\mathrm{d}} = D(K)$, onde $K = E \cap \mathcal{L}^\omega(G)$. Nesse caso, a especificação $K \subseteq \mathcal{L}^\omega(G)$ seria dada por $K = \Pi(E_{\mathrm{d}} \cap \Pi^{-1}(\mathcal{L}^\omega(G)) \cap L_{\mathrm{d}})$.

Então, o problema de controle com distinguidor consiste em encontrar um supervisor não-bloqueante $S_{\mathrm{d}} : \Delta^* \to 2^\Delta$, tal que $\mathcal{L}^\omega(S_{\mathrm{d}}/G_{\mathrm{d}}) \subseteq K_{\mathrm{d}}$. Cury *et al.* (2015) demonstra que:

$$\begin{aligned}\Pi(\sup\mathcal{C}(K_{\mathrm{d}}, G_{\mathrm{d}})) &= \sup\mathcal{C}(K, G) \\ \sup\mathcal{C}(K_{\mathrm{d}}, G_{\mathrm{d}}) &= D(\sup\mathcal{C}(K, G))\end{aligned} \tag{8}$$

Ou seja, a abordagem por Distinguidores proporciona uma solução de controle equivalente à obtida pela TCS clássica.

2.2.2.1 EXEMPLO: TCS COM DISTINGUIDORES

Baseado no exemplo proposto da seção da TCS e considerando suas limitações apresentadas, é possível adaptar a modelagem do problema utilizando Distinguidores. Nesse problema, é possível observar que o evento a se caracteriza por uma condição de modelagem que justifica sua exploração. No caso de Distinguidores, cada ocorrência do evento a, pode ser identificada por um novo evento a_i, no qual i está relacionado ao contexto em que o evento a ocorre, sendo este contexto incorporado semanticamente por um distinguidor.

Para o exemplo proposto, ao evento a podem ser atribuídos três contextos diferentes (primeira, segunda e terceira ocorrência de a na máquina M). Nesse caso o conjunto de refinamentos do evento a é $\Delta^a = \{a_1, a_2, a_3\}$. Dessa forma, a planta G, modelada com eventos em Σ, pode ser substituída por uma versão em Δ, dada por $G_a = \Pi^{-1}(G)$. Também a especificação E, anteriormente definida, pode ser substituída pela especificação E_d, que desabilita a ocorrência de $a_i \in \Delta^a$ após a ocorrência de a_3, apresentada na Figura 3.

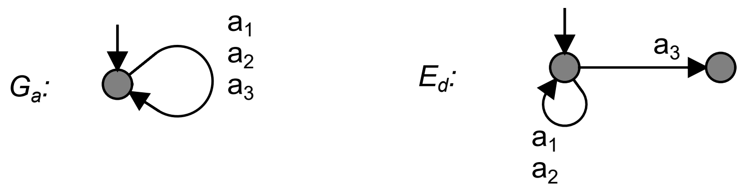

Figura 3: Modelagem do SED sob a abordagem por Distinguidores

É interessante notar que o processo de modelagem é claramente facilitado pela abordagem por Distinguidores, pois a mesma especificação pode ser usada para qualquer quantidade n de eventos a a serem memorizados, enquanto um modelo equivalente dependeria de n caso fosse modelado em Σ. Como demonstrado por Teixeira (2013), os resultados dessa abordagem e pela TCS convencional são equivalentes, i.e., $K = G \parallel E = \Pi(G_a \parallel H_d \parallel E_d) = \Pi(K_d)$, o que por conseguinte leva também a um controlador equivalente.

A desvantagem dessa abordagem é que, aparentemente, ela introduz um passo a mais no processo de modelagem, pois requer a construção de um modelo para distinguir a planta refinada do sistema, o qual, por sua vez, corresponde a uma estrutura que volta a depender de n. Para esse exemplo, tal modelo é mostrado na Figura 4.

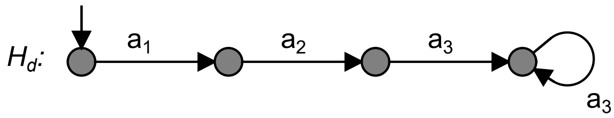

Figura 4: Modelo para distinção da planta refinada

Ao ser associado à planta G_a, o modelo H_d distingue a ocorrência de cada instância do evento a no sistema. Isso leva a uma planta distinguida $G_d = G_a \| H_d$, que pode então ser usada na síntese de controle. Note que, embora a modelagem em Δ não mostre vantagens aparentes em relação à modelagem em Σ (já que H_d e E possuem o mesmo número de estados) H_d é, geralmente, por natureza modular (devido ao refinamento dos eventos), ao contrário de E. H_d pode, na verdade, ser modelado por um conjunto de autômatos de 2 estados (CURY *et al.*, 2015).

A próxima sessão apresenta os conceitos teóricos referentes à segunda abordagem de otimização da TCS, por meio de AFEs, que implementa ideia similar aos Distinguidores, mas usando outra estrutura de modelos.

2.2.3 TCS COM AUTÔMATOS FINITOS ESTENDIDOS

AFEs compõem uma abordagem de modelagem que pode ser caracterizada por AFs que incorporam variáveis definidas sobre um determinado domínio. As transições de um AFE possuem fórmulas de guarda, que são regras definidas sobre as variáveis, e funções de atualização dos valores dessas variáveis (CHEN; LIN, 2000; CHEN; LIN, 2001). AFEs são formalmente definidos como a sétupla $A = (\Sigma, V, Q, Q^{\circ}, Q^{\omega}, \delta)$, (TEIXEIRA, 2013) onde:

- Σ é o *alfabeto* de eventos;
- $V = \{v_1, ..., v_n\}$ é o *conjunto de variáveis*;
- Q é um *conjunto finito de estados*;
- $Q^{\circ} \in Q$ é o conjunto de *estados iniciais*;
- $Q^{\omega} \subseteq Q$ é o *conjunto de estados finais ou de aceitação*;

- $\delta \subseteq Q \times \Sigma \times \Pi_V \rightarrow Q$ é uma *função de transição*, normalmente parcial em relação ao seu domínio, na qual Π_V é o conjunto de fórmulas sobre V. O efeito de uma fórmula altera os valores das variáveis no estado corrente para novos valores no estado alcançado. Para representar formalmente esse efeito, pode-se definir um conjunto auxiliar, denotado por V', que contém as mesmas variáveis de V, com o mesmo domínio. Ou seja, todos os valores de variáveis possivelmente assumidos em V também podem ser assumidos em V'. Assim, Π_V passa a ser um conjunto de fórmulas sobre $V \cup V'$, tal que cada $p \in \Pi_V$ implementa uma condição *booleana* $p(v, v') = true$ ou $false$, em que $v' \in V'$ é a variável v após atualização de seu valor por meio de uma fórmula lógica. Considerando estados q e p e um evento σ, e que $q \overset{\sigma:\pi}{\rightarrow} p$, então a representação por grafo de δ possui uma transição identificada q para p com o evento $\sigma \in \Sigma$ e com a fórmula $\pi \in \Pi_V$.

Um AFE também pode ser definido por meio da *definição explícita*, que nada mais é do que a versão materializada dos *updates* sobre as variáveis que, então, passam a compor a semântica dos estados. Segundo Teixeira (2013) a versão explícita é dada pelo autômato $A_V = (\Sigma, Q_A, Q_A^\circ, Q_A^\omega, \delta_A)$ onde:

- $Q_A = Q \times dom(V)$;
- $Q_A^\circ = Q^\circ \times \{(v_1^o, ..., v_n^o)\}$;
- $Q_A^\omega = Q^\omega \times dom(V)$;
- δ_A é tal que $(x, \overline{v}) \overset{\sigma}{\rightarrow} (y, \overline{v}')$, para $\overline{v}, \overline{v}' \in dom(V)$, se existe $x \overset{\sigma:p}{\rightarrow} y$ com $p(\overline{v}, \overline{v}') = true$.

A mesma relação de transição pode ser aplicada a Σ^* por $(x, \overline{v}) \overset{\epsilon}{\rightarrow} (x, \overline{v})$, para todo $(x, \overline{v}) \in Q_A$ e $(x, \overline{v}) \overset{s\sigma}{\rightarrow} (x'', \overline{v}'')$, se $(x, \overline{v}) \overset{s}{\rightarrow} (x', \overline{v}') \overset{\sigma}{\rightarrow} (x'', \overline{v}'')$, para algum elemento $(x', \overline{v}') \in Q_A$. Teixeira (2013) denota por $A_V \overset{s}{\rightarrow} (x, \overline{v})$ a existência de um estado $(q^\circ, \overline{v}^o) \in Q_A^\circ$, tal que $(q^\circ, \overline{v}^o) \overset{s}{\rightarrow} (x, \overline{v})$ e define as linguagens gerada e marcada por A_V, respectivamente, como:

$$\begin{aligned} \mathcal{L}(A_V) &= \left\{ s \in \Sigma^* \text{ tal que } A_V \overset{s}{\rightarrow} (x, \overline{v}) \in Q_A \right\} \\ \mathcal{L}^\omega(A_V) &= \left\{ s \in \Sigma^* \text{ tal que } A_V \overset{s}{\rightarrow} (x, \overline{v}) \in Q_A^\omega \right\} \end{aligned} \tag{9}$$

A composição síncrona $A_1||A_2$ também pode ser definida para dois AFEs $A_1 = (\Sigma, V, Q_1, Q_1^\circ, Q_1^\omega, \delta_1)$ e $A_2 = (\Sigma, V, Q_2, Q_2^\circ, Q_2^\omega, \delta_2)$ (OUEDRAOGO *et al.*, 2011; TEIXEIRA, 2013) como:

$$A_1 \Big|\Big| A_2 = \big(\Sigma_1 \cup \Sigma_2, V_1 \cup V_2, Q_1 \times Q_2, Q_1^\circ \times Q_2^\circ, Q_1^\omega \times Q_2^\omega, \delta_{1\|2}\big)\text{, tal que}$$

$$\delta_{1\|2} = \begin{cases} (x_1, x_2) \xrightarrow{\sigma:p_1 \wedge p_2} (y_1, y_2) \text{ se } \sigma \in \Sigma_1 \cap \Sigma_2,\ x_1 \xrightarrow{\sigma:p_1}_1 y_1 \text{ e } x_2 \xrightarrow{\sigma:p_2}_2 y_2 \\ (x_1, x_2) \xrightarrow{\sigma:p_1} (y_1, x_2) \text{ se } \sigma \in \Sigma_1 \setminus \Sigma_2 \text{ e } x_1 \xrightarrow{\sigma:p_1}_1 y_1 \\ (x_1, x_2) \xrightarrow{\sigma:p_2} (x_1, y_2) \text{ se } \sigma \in \Sigma_2 \setminus \Sigma_1 \text{ e } x_2 \xrightarrow{\sigma:p_2}_2 y_2 \\ \text{indefinido para outros casos} \end{cases} \tag{10}$$

A controlabilidade é outra propriedade da TCS que pode ser estendida para AFEs. Sejam $E_\mathrm{v} = (\Sigma, V, Q_E, Q_E^\circ, Q_E^\omega, \delta_E)$ e $G_\mathrm{v} = (\Sigma, V, Q_G, Q_G^\circ, Q_G^\omega, \delta_G)$ dois AFEs, Teixeira (2013) diz que E_v é V-controlável em relação a G_v, se o seguinte for verdadeiro para todo $s \in \Sigma^*$, $\mu \in \Sigma_u$, $x_E \in Q_E$, x_G e $x'_G \in Q_G$, $\overline{v}$ e $\overline{v}' \in dom(V)$:

$$\begin{aligned} E_\mathrm{v} &\xrightarrow{s} (x_E, \overline{v}) \\ \wedge G_\mathrm{v} &\xrightarrow{s} (x_G, \overline{v}) \xrightarrow{\mu} (x'_G, \overline{v}') \\ \Rightarrow \exists\, x1_E \in Q_E \mid E_\mathrm{v} &\xrightarrow{s} (x_E, \overline{v}) \xrightarrow{\mu} (x'_E, \overline{v}') \end{aligned} \tag{11}$$

Sendo assim, uma especificação E_v é V-controlável se não proíbe eventos não-controláveis em G_v e se atualiza as variáveis da mesma forma como na planta. Considerando $E_\mathrm{v} \parallel G_\mathrm{v}$ como o modelo do comportamento esperado sob controle, define-se o conjunto:

$$C_v = \{K_v \subseteq E_\mathrm{v} \parallel G_\mathrm{v} | K_v \text{ é V-controlável em relação a } G_\mathrm{v}\} \tag{12}$$

C_v possui um AFE supremo, denotado por $\sup\mathcal{C}_V(E_\mathrm{v}, G_\mathrm{v})$, representando o comportamento menos restritivo a ser implementado sobre G_v, de modo a garantir E_v. Pode ser mostrado (TEIXEIRA *et al.*, 2015) que, nessas condições, $\mathcal{L}(\sup\mathcal{C}_V(E_\mathrm{v}, G_\mathrm{v})) = \sup\mathcal{C}(\mathcal{L}(E_\mathrm{v} \| G_\mathrm{v}), \mathcal{L}(G_\mathrm{v}))$, ou seja, a linguagem de $\sup\mathcal{C}_V$ equivale à síntese da linguagem de um determinado AFE, ou de sua versão ordinária.

2.2.3.1 EXEMPLO: TCS COM AFE

O mesmo exemplo proposto anteriormente é reproduzido a seguir, dessa vez com a modelagem do problema utilizando AFEs. Nesse caso, após cada ocorrência do evento a, o estado alcançado pode ser identificado por um valor discreto dentro de um domínio de uma variável x, onde $dom(x) = \{0, 1, 2, 3, 4\}$ e $x^o = 0$. Sendo assim, o objetivo da planta é atribuir à variável x um valor que identifica o contexto em que o estado alcançado por a possui. Já a especificação objetiva restringir a planta de modo que a condição de controle seja satisfeita, ou seja, apenas 3 ocorrências de a devem ser possíveis. A Figura 5 mostra como a planta G_v e a especificação E_v do exemplo podem ser modeladas por AFEs.

Figura 5: Modelagem do SED sob abordagem por AFEs

A cada ocorrência do evento a, a variável x é incrementada dentro de seu domínio. O domínio 0 é usado para representar a condição do estado inicial, enquanto que o domínio 4 sustenta o fato de ser possível ocorrer algo proibido na planta, justamente o que é desabilitado via controle. Ao construir uma condição de guarda que considera o estado alcançado com base no valor atual de x, é possível identificar o contexto desse estado e impedir a ocorrência de a não desejada.

É interessante notar que os modelos, tanto da planta, quanto das especificações, possuem apenas um estado, assim como a composição entre eles. Isso indica que o esforço e complexidade de modelagem recai sobre as fórmulas lógicas implementadas para atualização e guarda das variáveis. Porém, sob o ponto de vista computacional, a versão explícita do autômato K_v possui a mesma quantidade de estados que os autômatos K e K_d, o que será exposto na sequência.

3 PROPOSTA

A motivação principal para o uso de AFEs e de Distinguidores como alternativas de extensão da TCS, é que essas abordagens facilitam expressar requisitos de controle. Uma breve comparação dos exemplos providos nas figuras 2, 3 e 5 dimensiona esse argumento. Neles, um modelo com n estados é substituído por estruturas de 1 e 2 estados fixos, para qualquer n. Esse fator, por si só, pode tornar viável o tratamento de problemas de controle cuja solução envolvendo apenas as políticas convencionais da TCS seria intratável.

Adicionalmente, Distinguidores e AFEs têm ainda a vantagem de facilitar tarefas de modelagem sem complexificar a síntese de controle como efeito. Na verdade, a literatura tem provado que o esforço de síntese, com ou sem o uso dessas abordagens é exatamente o mesmo. Isso pode ser facilmente ilustrado para o exemplo abordado anteriormente no contexto do comportamento desejado. Os modelos a seguir mostram como seria o modelo K e suas respectivas versões K_d e K_v (explícito).

$$K: \qquad q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \xrightarrow{a} q_3 \tag{13}$$

$$K_d: \qquad q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \xrightarrow{a_3} q_3 \tag{14}$$

$$K_v: \qquad (q_0,0) \xrightarrow{a} (q_1,1) \xrightarrow{a} (q_2,2) \xrightarrow{a} (q_3,3) \tag{15}$$

Portanto, é esperado que $L(K) = L(K_v) = \Pi(L(K_d))$ e, assim, também é esperada uma solução de controle equivalente.

Um aspecto que poderia ser visto como uma desvantagem é que a simplificação da modelagem cobra como contrapartida a construção de uma estrutura adicional de modelagem: o modelo distinguidor, no caso da abordagem com refinamentos; e a estrutura de fórmulas, no caso dos AFEs. Ainda assim, essas são atividades naturalmente modulares, que podem ser conduzidas incrementalmente e, em tese, de maneira bastante simples.

Esse trabalho se desenvolve sobre problemas de controle para os quais essas estruturas de tratamento de modelos é determinante para a obtenção de uma solução. Para tais, é necessário definir, antes de mais nada, qual abordagem será usada: Distinguidores ou AFEs. Sabe-se que os resultados teóricos existentes até o momento não suportam a combinação direta dessas abordagens. Sabe-se, ainda, que para alguns casos, uma abordagem pode se mostrar mais vantajosa do que a outra. A Tabela 1 estabelece um comparativo entre os principais indicadores que levariam Distinguidores e AFEs a serem adotados no projeto de desenvolvimento de um controlador.

Tabela 1: Comparação de características entre as três abordagens

Característica	TCS	Distinguidores	AFEs
Otimização de modelagem	Nenhuma	Por meio de refinamentos de eventos	Por inserção de variáveis num domínio específico
Foco do refinamento	Não se aplica	Nos eventos	Nos estados
Modularização	Sim	Sim	Sim
Modularização com vantagens computacionais	Não	Sim	Não
Modelagem extra	Nenhuma	Distinguidor H_d	Fórmulas lógicas de atualização e guarda
Modelagem de problemas complexos	Inviável	Possível	Possível
Aderência a ferramental	Sim	Sim	Restrita
Suporte a geração de código	Sim	Sim	Restrita
Equivalência em relação à TCS convencional	Não se aplica	Sim	Restrita

Sendo assim, entende-se que seria de utilidade prática dispor de um método que permitisse a conversão entre modelos por Distinguidores e AFEs, de modo que as versões adotadas na modelagem e na síntese pudessem ser distintas e, assim, pudessem combinar as vantagens de cada método, conforme a conveniência. O presente trabalho propõe a formulação desse método de conversão. O objetivo é basicamente estabelecer as condições teóricas para mapear o elemento que materializa o refinamento da informação, de uma abordagem para a outra, e vice-versa, provendo o respectivo ferramental de conversão.

Alguns indícios de que esse método de conversão é alcançável, podem ser observados empiricamente. Se, por exemplo, identifica-se um evento b em um SED que pode ser refinado por Distinguidores, com o propósito de simplificação, em b_1, b_2, b_3 e b_4, é possível afirmar que a esse evento é atribuído quatro contextos diferentes. Sendo assim, a mesma identificação de contextos pode ser reproduzida sobre um AFEs, onde existiria uma variável y inicialmente valorada em 0, por exemplo, que tem seu valor incrementado a cada ocorrência de b e possui um domínio de 0 a 4. Essa variável então assumiria a cada incremento de seu valor o mesmo contexto e significado atribuído aos eventos refinados citados anteriormente.

REFERÊNCIAS

BOUZON, Gustavo; QUEIROZ, Max H de; CURY, José ER. Supervisory control of des with distinguishing sensors. In: IEEE. **Discrete Event Systems, 2008. WODES 2008. 9th International Workshop on**. [S.l.], 2008. p. 22–27.

BOUZON, Gustavo; QUEIROZ, Max H de; CURY, José ER. Exploiting distinguishing sensors in supervisory control of des. In: **IEEE International Conference on Control and Automation, ICCA**. [S.l.: s.n.], 2009. v. 9, p. 442–447.

CASSANDRAS, Christos G; LAFORTUNE, Stephane. **Introduction to discrete event systems**. [S.l.]: Springer Science & Business Media, 2009.

CHEN, Yi-Liang; LIN, Feng. Modeling of discrete event systems using finite state machines with parameters. In: IEEE. **Control Applications, 2000. Proceedings of the 2000 IEEE International Conference on**. [S.l.], 2000. p. 941–946.

CHEN, Yi-Liang; LIN, Feng. Safety control of discrete event systems using finite state machines with parameters. In: IEEE. **American Control Conference, 2001. Proceedings of the 2001**. [S.l.], 2001. v. 2, p. 975–980.

CHENG, Kwang Ting; KRISHNAKUMAR, Avinash S. Automatic functional test generation using the extended finite state machine model. In: ACM. **Proceedings of the 30th international Design Automation Conference**. [S.l.], 1993. p. 86–91.

CURY, José ER; QUEIROZ, Max Hering de; BOUZON, Gustavo; TEIXEIRA, Marcelo. Supervisory control of discrete event systems with distinguishers. **Automatica**, Elsevier, v. 56, p. 93–104, 2015.

FEI, Zhennan; MIREMADI, Sajed; ÅKESSON, Knut; LENNARTSON, Bengt. A symbolic approach to large-scale discrete event systems modeled as finite automata with variables. In: IEEE. **2012 IEEE International Conference on Automation Science and Engineering (CASE)**. [S.l.], 2012. p. 502–507.

FISCHER, G.; LEAL, A. InvestigaÃ§Ã£o do uso de distinguidores na sÃntese de supervisores em funÃ§Ã£o de mudanÃ§as na planta. In: **Congresso Brasileiro de AutomÃ¡tica**. [S.l.: s.n.], 2014.

HOPCROFT, John E.; MOTWANI, Rajeev; ULLMAN, Jeffrey D. **Introduction to Automata Theory, Languages, and Computation**. 2. ed. United States of America: Pearson Education, 1939.

LEWIS, Harry R; PAPADIMITRIOU, Christos H. **Elements of the Theory of Computation**. [S.l.]: Prentice Hall PTR, 1997.

MALIK, Robi; TEIXEIRA, Marcelo. Modular supervisor synthesis for extended finite-state machines subject to controllability. In: IEEE. **Discrete Event Systems (WODES), 2016 13th International Workshop on**. [S.l.], 2016. p. 91–96.

OUEDRAOGO, Lucien; KUMAR, Ratnesh; MALIK, Robi; AKESSON, Knut. Nonblocking and safe control of discrete-event systems modeled as extended finite automata. **IEEE Transactions on Automation Science and Engineering**, IEEE, v. 8, n. 3, p. 560–569, 2011.

QUEIROZ, Max H De; CURY, José ER. Modular control of composed systems. In: IEEE. **American Control Conference, 2000. Proceedings of the 2000**. [S.l.], 2000. v. 6, p. 4051–4055.

RAMADGE, Peter J; WONHAM, W Murray. Supervisory control of a class of discrete event processes. **SIAM journal on control and optimization**, SIAM, v. 25, n. 1, p. 206–230, 1987.

RAMADGE, Peter JG; WONHAM, W Murray. The control of discrete event systems. **Proceedings of the IEEE**, IEEE, v. 77, n. 1, p. 81–98, 1989.

SKOLDSTAM, Markus; AKESSON, Knut; FABIAN, Martin. Modeling of discrete event systems using finite automata with variables. In: IEEE. **Decision and Control, 2007 46th IEEE Conference on**. [S.l.], 2007. p. 3387–3392.

TEIXEIRA, Marcelo. Explorando o uso de distinguidores e de autômatos finitos estendidos na teoria do controle supervisório de sistemas a eventos discretos. Tese de Doutorado - Doutorado em Engenharia de Automação e Sistemas - Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Universidade Federal de Santa Catarina - UFSC, Florianópolis, 2013.

TEIXEIRA, Marcelo; CURY, José ER; QUEIROZ, Max H de. Local modular control with distinguishers applied to a manufacturing system. **IFAC Proceedings Volumes**, Elsevier, v. 46, n. 9, p. 263–268, 2013.

TEIXEIRA, Marcelo; MALIK, Robi; CURY, José ER; QUEIROZ, Max H de. Supervisory control of des with extended finite-state machines and variable abstraction. **IEEE Transactions on Automatic Control**, IEEE, v. 60, n. 1, p. 118–129, 2015.

THISTLE, John G. Synthesis of supervisory controls for discrete event systems. In: IEEE. **Application of Concurrency to System Design, 2004. ACSD 2004. Proceedings. Fourth International Conference on**. [S.l.], 2004. p. 151.