

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS**

Pedro Afonso Costa Ferraz Leite

**CONTROLE SUPERVISÓRIO DE REDES DE
REGULAÇÃO GÊNICA COM RESTRIÇÕES PARA
REALIZAÇÃO DO SUPERVISOR**

Florianópolis

2018

Pedro Afonso Costa Ferraz Leite

**CONTROLE SUPERVISÓRIO DE REDES DE
REGULAÇÃO GÊNICA COM RESTRIÇÕES PARA
REALIZAÇÃO DO SUPERVISOR**

Dissertação submetida ao Programa
de Pós-Graduação em Engenharia de
Automação e Sistemas para a obtenção
do Grau de Mestre em Engenharia de
Automação e Sistemas.
Orientador: Prof. Dr. Fabio Luis Baldissera
Coorientador: Prof. Dr. José Eduardo
Ribeiro Cury

Florianópolis

2018

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

Leite, Pedro Afonso Costa Ferraz
Controle Supervisório de Redes de Regulação Gênica
com Restrições para Realização do Supervisor / Pedro
Afonso Costa Ferraz Leite ; orientador, Fabio Luis
Baldissera, coorientador, José Eduardo Ribeiro
Cury, 2018.
70 p.

Dissertação (mestrado profissional) -
Universidade Federal de Santa Catarina, Centro
Tecnológico, Programa de Pós-Graduação em Engenharia
de Automação e Sistemas, Florianópolis, 2018.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2.
Engenharia de Automação e Sistemas. 3. Sistemas a
Eventos Discretos. 4. Teoria de Controle
Supervisório. 5. Biologia Sintética. I. Baldissera,
Fabio Luis. II. Cury, José Eduardo Ribeiro. III.
Universidade Federal de Santa Catarina. Programa de
Pós-Graduação em Engenharia de Automação e Sistemas.
IV. Título.

Pedro Afonso Costa Ferraz Leite

**CONTROLE SUPERVISÓRIO DE REDES DE
REGULAÇÃO GÊNICA COM RESTRIÇÕES PARA
REALIZAÇÃO DO SUPERVISOR**

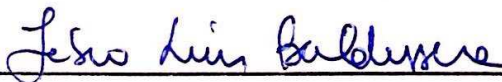
Esta Dissertação foi julgada aprovada para a obtenção do Título de “Mestre em Engenharia de Automação e Sistemas” e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Automação e Sistemas.

Florianópolis, 14 de Setembro de 2018.



Prof. Dr. Werner Kraus Junior
DAS/UFSC
Coordenador do Curso

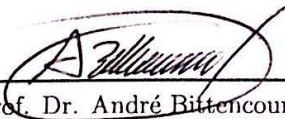
Banca Examinadora:



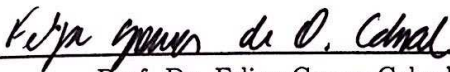
Prof. Dr. Fabio Luis Baldissera
DAS/UFSC



Prof. Dr. Daniel Santos Mansur
CCB/MIP/UFSC



Prof. Dr. André Bittencourt Leal
FEJ/UDESC



Prof. Dr. Felipe Gomes Cabral
DAS/UFSC

Dedico este trabalho às pessoas que amo.

AGRADECIMENTOS

Agradeço aos meus pais, Afonso e Heda, e à minha irmã Walkyria, por todo suporte durante essa jornada. Boa parte do homem que sou hoje é fruto da educação, carinho e amor infindáveis que recebi deles durante toda a minha vida.

Também agradeço ao suporte e amor de todo o resto da minha família, a qual felizmente é muito grande para citar nome por nome aqui.

Por fim, agradeço aos meus orientadores, professores Cury e Fabio, por desempenharem com maestria a função de orientador, sempre me guiando pelos melhores caminhos. Sem vocês eu não chegaria de x_0 a X_m .

It is far better to grasp the universe as
it really is than to persist in delusion,
however satisfying and reassuring.

(Carl Sagan, 1995)

RESUMO

Esta dissertação trata da síntese de controladores capazes de alterar o comportamento de uma classe de redes biológicas intracelulares, as redes de regulação gênica, quando há restrições para realização biológica das ações de controle. Este trabalho é uma extensão de uma abordagem existente, em que a realização dos controladores de redes de regulação se dá por meio de genes sintéticos inseridos nas células cujos comportamentos se desejam alterar. Como já discutido na literatura, no entanto, não é possível projetar genes sintéticos capazes de implementar qualquer estratégia de controle. Certas funções matemáticas são difíceis ou mesmo impossíveis de serem obtidas biologicamente. Assim, é necessário lidar com um problema de síntese de controladores em que a representação matemática do controlador é restrita, de modo a refletir as restrições biológicas. Nesta dissertação, o problema de controle de redes biológicas sob restrições de implementação é formalizado e resolvido, ampliando, assim, a possibilidade de emprego de ideias existentes na literatura para controle de redes reais.

Palavras-chave: Biologia Sistêmica. Teoria de Controle Supervisório. Redes de Regulação Gênica. Biologia Sintética.

ABSTRACT

This dissertation deals with the synthesis of controllers capable of altering the behavior of a class of intracellular biological networks, i.e. gene regulatory networks, when there are restrictions for the biological realization of the control actions. This work is an extension of an existing approach, in which the realization of the regulatory network controllers occurs through synthetic genes inserted into cells whose behaviors are desired to be altered. As already discussed in the literature, however, it is not possible to design synthetic genes capable of implementing any control strategy. Certain mathematical functions are difficult or even impossible to obtain biologically. Thus, it is necessary to deal with a controller synthesis problem in which the mathematical representation of the controller is restricted, so as to reflect the biological constraints. In this dissertation, the problem of controlling biological networks under implementation restrictions is formalized and solved, thus expanding the possibility of using existing ideas in the literature to control real networks.

Keywords: Systems Biology. Supervisory Control Theory. Gene Regulatory Networks. Synthetic Biology.

LISTA DE FIGURAS

Figura 1	Esquema de como as áreas do conhecimento se relacionam para culminar no problema desta dissertação.....	24
Figura 2	Esquema do processo de síntese proteica.	26
Figura 3	Gráfico de uma RRG hipotética. Arcos terminados em setas indicam interação de ativação, e terminados em barra indicam interação de repressão. Por exemplo, o arco direcionado de B para C indica que a proteína codificado por B tem um efeito inibidor em C, e o arco direcionado de B para A indica que a proteína codificada por B tem um efeito ativador em A.....	28
Figura 4	Esquema de uma simples linha de manufatura como exemplo de um SED. A peça entra na máquina 1 (M_1), segue para o <i>buffer</i> (B), entra na máquina 2 (M_2) e sai dela como peça pronta.	30
Figura 5	Estrutura em malha fechada do controle supervísório...	32
Figura 6	Gráfico de uma RRG hipotética formada por três genes, v_1 , v_2 e v_3 . Arcos terminados em setas indicam interação de ativação, e terminados em barra indicam interação de repressão...	36
Figura 7	Representação do diagrama de transição de estados assíncrono para a rede de genes da Figura 6. Cada estado é representado por uma cadeia binária que indica quais genes nativos v_i estão sendo expressados, i.e., quais fatores de transcrição u_i estão presentes na célula. Os estados hachurados correspondem aos atratores da rede, que são os estados que não possuem arcos saindo deles. Imagem extraída de (BALDISSERA; CURY; RAISCH, 2016).....	36
Figura 8	Autômato G associado ao diagrama da figura 7. O estado inicial é indicado por uma seta ($q_0 = 010$) e o conjunto de estados marcados são os estados com duplo círculo ($Q_m = \{101\}$). Imagem extraída de (BALDISSERA; CURY; RAISCH, 2016).....	38
Figura 9	RRG expandida com os genes sintéticos v_1^s e v_3^s . Imagem extraída de (BALDISSERA; CURY; RAISCH, 2016).....	39
Figura 10	Autômato G^{hi} com novas interações em cinza. Imagem extraída de (BALDISSERA; CURY; RAISCH, 2016).....	40
Figura 11	Representação das regiões reguladoras dos genes v_1^s e v_3^s . Imagem extraída de (BALDISSERA; CURY; RAISCH, 2016).....	44
Figura 12	Gráfico de uma RRG hipotética utilizada no exemplo da Seção 4.2.	48

Figura 13 Autômato G associado à RRG hipotética da Figura 12. Uma barra no arco significa evento controlável e um círculo no arco significa evento forçável.	49
Figura 14 Primeiros passos na construção da árvore T usada para resolver o exemplo da Seção 4.2. A segunda linha de cada caixa mostra o padrão de ativação dos genes sintéticos necessário para executar a entrada de controle na terceira linha (1 para ativar o gene, 0 para desativar e * para “não importa”).	52
Figura 15 Conjuntos de nodos C_0 e C_1 para o exemplo da Seção 4.2.	53
Figura 16 Árvore associada à solução do exemplo da Seção 4.2. A cada iteração, o conjunto C_k é computado a partir de C_{k-1} , como explicado no texto. O critério de parada foi alcançado quando o nodo n_8 de C_5 foi calculado, pois sua projeção no espaço de estado é o estado inicial x_0 , significando que existe uma solução para este problema.	54
Figura 17 Parte acessível, a partir de n_8 , da árvore de nodos da Figura 16.	57
Figura 18 Exemplo hipotético dos testes realizados pelo Algoritmo 1 para definir novos nodos.	61

SUMÁRIO

1	INTRODUÇÃO	19
1.1	MOTIVAÇÃO	19
1.2	JUSTIFICATIVA	19
1.3	OBJETIVOS	20
1.4	ESCOPO E MÉTODO.....	21
1.5	ORGANIZAÇÃO DO DOCUMENTO	23
2	CONCEITOS PRELIMINARES	25
2.1	CONCEITOS DE BIOLOGIA	25
2.2	REDES BOOLEANAS.....	29
2.3	SISTEMAS A EVENTOS DISCRETOS.....	30
2.4	TEORIA DE CONTROLE SUPERVISÓRIO	31
3	O PROBLEMA DE ATRAÇÃO DE ESTADOS PARA REDES DE REGULAÇÃO GÊNICA.....	35
4	O PROBLEMA DE ATRAÇÃO DE ESTADOS SOB RESTRIÇÕES BIOLÓGICAS	47
4.1	DECLARAÇÃO DO PROBLEMA	47
4.2	EXEMPLO DIDÁTICO.....	48
4.3	SOLUÇÃO GERAL	55
4.4	DESCRIÇÃO DO ALGORITMO	55
4.5	PROPRIEDADES DO ALGORITMO PROPOSTO	58
5	CONCLUSÕES	65
	REFERÊNCIAS	67

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Os genes, e as interações entre eles, são responsáveis por grande parte do comportamento de uma célula, como será descrito em mais detalhes no próximo capítulo. Isto é verdade tanto para seres unicelulares mais simples, como bactérias, quanto para seres multicelulares complexos, como os seres humanos (ALBERTS et al., 2013). Assim, a intervenção sobre a dinâmica dos genes e das redes que eles formam encontra inúmeras aplicações em Medicina e Biotecnologia, como produção de biocombustíveis, tratamento de doenças e desenvolvimento de alimentos geneticamente modificados (BENENSON, 2012; KARLSSON; WEBER, 2012; PURNICK; WEISS, 2009).

A intervenção do homem sobre processos biológicos em nível celular, ou mesmo mais sistêmico, não é algo novo. Medicamentos diversos e vacinas são exemplos corriqueiros. Contudo, o poder de intervenção sobre sistemas celulares aumentou significativamente nas últimas décadas, em grande parte devido a três fatores: a) o avanço tecnológico, que permite mapear, a um custo cada vez menor, uma quantidade cada vez maior de biomoléculas e suas interações (o leitor interessado pode consultar (HORGAN; KENNY, 2011), que retrata avanços nas áreas de genômica, proteômica e metabolômica); b) o emprego de modelos matemáticos para compreensão da dinâmica de processos celulares, uma área conhecida como Biologia Sistêmica (MACHADO et al., 2011), facilitando a análise de complexos processos intracelulares; e c) o advento e desenvolvimento de um campo da ciência denominado Biologia Sintética (ANDRIANANTOANDRO et al., 2006), dedicado ao projeto e síntese de biomoléculas outrora inexistentes, possibilitando a construção de moléculas capazes de medir variadas grandezas de interesse celular, processar informações e disparar ações em contextos específicos.

Este trabalho se insere neste contexto de intervenção do homem sobre processos biológicos intracelulares por meio de moléculas sintéticas.

1.2 JUSTIFICATIVA

Moléculas são responsáveis por processar informações no ambiente intracelular, ou seja, elas possuem a habilidade de computar. O

mecanismo pelo qual as moléculas processam informação foi construído gradualmente, através de bilhões de anos, pelo processo de evolução dos seres vivos. A característica de computar das moléculas possibilita o seu uso como um dos meios de implementar um sistema de controle celular. Neste contexto, novas moléculas são criadas em laboratório para agir como sensores, atuadores e controladores dentro de células (ANDRIANANTOANDRO et al., 2006).

A área que estuda essa síntese de novas moléculas, ou modificações de moléculas existentes, é a *Biologia Sintética*. Por mais que este campo do conhecimento esteja em pleno progresso, a síntese de moléculas não é uma tarefa trivial e tem suas limitações, a despeito dos recentes avanços em *Biologia Sintética* (ANDRIANANTOANDRO et al., 2006). Portanto, é necessário levar em consideração as restrições biológicas inerentes ao processo de síntese de moléculas para implementar controle celular através de moléculas sintéticas. Nesse sentido, um estudo que se mostra importante é desenvolver uma abordagem de controle para derivar uma lógica que satisfaça as especificações desejadas que ainda assim não exija moléculas que não podem ser sintetizadas.

1.3 OBJETIVOS

O objetivo principal deste trabalho é apresentar um algoritmo que recebe como entradas: a) um modelo matemático de uma rede de regulação gênica a ser controlada; b) uma especificação de controle para o comportamento em regime permanente da rede, i.e., um estado alvo que deve ser atingido; e c) um conjunto de restrições relacionadas às funções matemáticas que podem ser implementadas por moléculas sintéticas; e retorna como saída especificações de genes sintéticos que satisfazem as restrições em (c) e que, em contato com a rede de regulação gênica a ser controlada, garantem que a especificação dada em (b) seja atendida.

A formalização do problema mencionado no parágrafo anterior resultou em um problema de controle ainda não tratado na literatura. Nele há restrições bem particulares sobre a realização do controlador, como será detalhado adiante neste trabalho. Como acreditamos ser possível que esta formulação apareça em outras aplicações não relacionadas ao domínio biológico que originou esta dissertação, elencamos também como objetivo secundário deste manuscrito compartilhar com a comunidade de teoria de controle a natureza matemática deste problema não ortodoxo, independentemente do problema biológico subja-

cente.

1.4 ESCOPO E MÉTODO

Este trabalho foca no controle do comportamento em regime permanente de redes de regulação gênica (RRGs). Elas são importantes redes de interação entre genes que coordenam uma variedade de processos vitais dentro das células (ALBERTS et al., 2010), como descreveremos em mais detalhes no próximo capítulo. Frequentemente, essas redes possuem um grande número de interações entre os genes, o que as tornam estruturas muitas vezes complexas. Porém, para controlá-las, não é necessário compreendê-las em toda sua complexidade. Basta abstraí-las com um bom modelo matemático. Nesta dissertação, optamos por modelar o comportamento das redes usando formalismos matemáticos advindos da área de sistemas a eventos discretos (CASSANDRAS; LAFORTUNE, 2009). Uma vez modeladas, podem ser aplicadas técnicas de controle às RRGs. A Teoria de Controle Supervisório é uma delas.

A Teoria de Controle Supervisório (TCS) é um corpo de conhecimento com o propósito de derivar controladores (chamados de supervisores) para sistemas a eventos discretos. Sistemas a eventos discretos (SEDs) são sistemas dinâmicos que evoluem de acordo com ocorrências abruptas de eventos físicos a intervalos não regulares. Um exemplo de evento em sistemas biológicos é a ativação de um gene por um fator de transcrição. O supervisor (controlador) S , dentro da estrutura da TCS, interage com o sistema a ser controlado (planta P), de acordo com o conceito usual de controle realimentado. Ou seja, S observa alguns eventos que P executa, então S diz a P quais eventos são permitidos em seguida ou mesmo força certos eventos (preemptabilidade), com o intuito de atender a um conjunto de especificações E . Assim, o controlador pode induzir, habilitar ou desabilitar eventos na planta. A TCS se baseia na Teoria de Linguagens e Autômatos de Estados Finitos. A planta, o supervisor e as especificações podem ser modelados por autômatos.

Duas abordagens principais podem ser reconhecidas na literatura para controlar a dinâmica das redes intracelulares: uma que utiliza dispositivos técnicos como microscopia, microfluídica e computadores para implementar as funções de sensoriamento, controle e atuação (MILIARIS; ARGENTIS et al., 2011; MENOLASCINA; BERNARDO; BERNARDO, 2011); e uma que utiliza moléculas sintéticas inseridas nas células a serem

controladas (BAGH et al., 2011). O presente trabalho segue a segunda estratégia e é uma extensão das ideias propostas por (BALDISSERA; CURY; RAISCH, 2016).

Esta abordagem baseia-se na união de três vertentes do conhecimento: Biologia Sistêmica, Biologia Sintética e Teoria de Controle Supervisório. Nela os autores abordaram o problema de controlar um determinado tipo de rede intracelular, as redes de regulação gênica, por meio de técnicas baseadas na Teoria de Controle Supervisório. Os autores argumentam que o supervisor derivado para uma RRG poderia ser implementado por genes sintetizados em laboratório capazes de computar funções Booleanas. Essa abordagem pode ser resumida nas seguintes etapas:

1. Abstrair uma RRG (planta P) como um autômato de estados finitos.
2. Derivar um supervisor S através da Teoria de Controle Supervisório.
3. Sintetizar moléculas que computam as funções Booleanas de S e inseri-las na célula para fechar o ciclo de controle.

Esta abordagem, apesar de já ter se mostrado promissora, possui alguns problemas a serem resolvidos. Um deles é o das restrições biológicas que surgem no processo da síntese de genes que implementam o controlador. A abordagem desenvolvida até o momento não considera essas restrições durante o cálculo do supervisor S . Portanto, existe a necessidade de desenvolver um método formal para derivar um supervisor que atenda estas restrições e seja implementável através de genes sintéticos.

A realização do supervisor através de genes sintéticos, embora viável de acordo com os avanços recentes observados na área de Biologia Sintética (ANDRIANANTOANDRO et al., 2006), esbarra em restrições biológicas que foram negligenciadas em (BALDISSERA; CURY; RAISCH, 2016). Parte desse problema já foi tratado em (BALDISSERA; CURY, 2014), no qual os autores apresentam maneiras de simplificar as funções Booleanas que os genes sintéticos devem computar, levando a funções com menos variáveis e operações lógicas. No entanto, as dificuldades para implementar funções Booleanas arbitrárias com genes sintéticos não estão necessariamente associadas a medidas matemáticas da complexidade dessas funções (por exemplo, número de entradas e número de conectivos lógicos). Embora existam genes na natureza que

implementam funções complexas com 20 proteínas diferentes como entradas (ver a regulação transcricional do gene *Eve* na mosca-da-fruta *Drosophila* (ALBERTS et al., 2013)), pode haver limitações severas na implementação de funções simples, como por exemplo a operação Booleana AND entre duas entradas, dependendo das proteínas sendo consideradas. Essas dificuldades ficarão mais claras para o leitor não versado em Biologia Molecular no próximo capítulo.

1.5 ORGANIZAÇÃO DO DOCUMENTO

Esta dissertação é organizada da seguinte maneira. No Capítulo 2, os conceitos básicos, brevemente explicados na Introdução, são aprofundados. Neste capítulo serão apresentados os Sistemas a Eventos Discretos, a Teoria de Controle Supervisório, Fundamentos de Biologia, Biologia Sistêmica, Biologia Sintética, Redes de Regulação Gênica, Redes Booleanas e o Problema de Atração de Estados. A Figura 1 mostra uma representação em fluxograma de como essas áreas se relacionam. No Capítulo 3, o problema a ser resolvido neste trabalho é apresentado e explicado, chamado de Problema de Atração de Estados Sob Restrições Biológicas na Realização do Supervisor. Esse capítulo também contém um exemplo resolvido, um método detalhado de resolução e um pseudo algoritmo proposto para implementação do método. Finalmente, o Capítulo 5 é constituído das conclusões, em que os resultados e ideias para trabalhos futuros são discutidos.

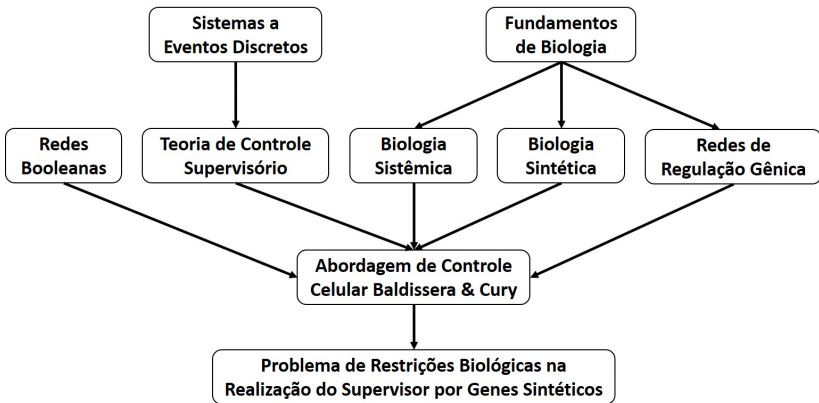


Figura 1 – Esquema de como as áreas do conhecimento se relacionam para culminar no problema desta dissertação.

2 CONCEITOS PRELIMINARES

Este capítulo é dedicado a introduzir os conceitos básicos nos quais a abordagem de controle desenvolvida é fundamentada. O capítulo começa com uma seção de biologia básica e redes de regulação gênica, passa por uma breve explicação sobre redes Booleanas e sistemas a eventos discretos nas seções seguintes e termina com uma seção sobre a Teoria de Controle Supervisório.

2.1 CONCEITOS DE BIOLOGIA

Nesta seção, abordaremos conceitos básicos de Biologia Celular necessários para compreensão da dissertação. Para mais informações sugere-se a leitura de (ALBERTS et al., 2010).

As células são consideradas unidades da vida, pois elas são as menores entidades que possuem todas as características necessárias para serem consideradas vivas, que são absorver nutrientes do ambiente à sua volta, multiplicar-se através da informação genética contida no DNA (ácido desoxirribonucleico) e excrementar material residual da geração de energia (ALBERTS et al., 2010). O processo de reprodução das células é importante para o contexto desse trabalho e merece uma explicação mais detalhada. Dentro de cada célula está armazenada toda a informação hereditária do organismo, não importa se unicelular ou multicelular. Essa informação é armazenada através de uma codificação que utiliza quatro caracteres. As moléculas de fita dupla do DNA, longas cadeias poliméricas pareadas, são constituídas de monômeros denominados nucleotídeos. Cada nucleotídeo é formado por um açúcar, a desoxirribose, ligado a um grupo fosfato e a uma base. Essa base, por sua vez, pode ser de quatro tipos apenas, adenina (A), timina (T), citosina (C) e guanina (G). Então, o “alfabeto” utilizado para codificar informação genética é o mesmo para todas as formas de vida na Terra, A, T, C e G.

A leitura da informação gênica contida no DNA, também chamada de expressão gênica, é dividida em dois processos, a *transcrição* e a *tradução*, ambas esquematizadas na Figura 2. Na primeira etapa, moléculas de RNA (ácido ribonucleico) são produzidas pelo processo de transcrição a partir do código contido em um dado trecho de DNA (sequência contida nos segmentos codantes). Essas moléculas de RNA, assim como as de DNA, formam estruturas em fitas com quatro bases

nitrogenadas, com a diferença de possuir uracila (U) ao invés da timina (T) encontrada no DNA e ser formado por somente uma fita. Os genes podem ser entendidos como moldes para a construção de fitas de RNA. Na segunda etapa, o RNA é traduzido em moléculas de proteína. Portanto, cada proteína é codificada por um único gene. Então, é possível associar o número de moléculas de proteína presentes no meio celular com o que é chamado de expressão gênica.

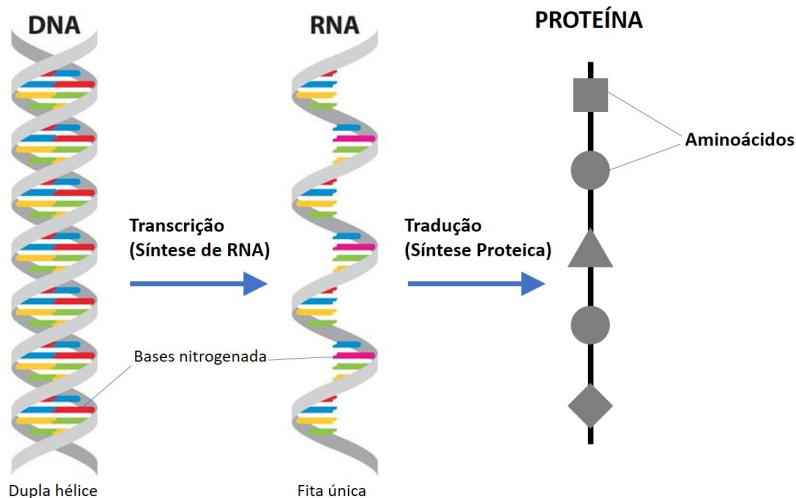


Figura 2 – Esquema do processo de síntese proteica.

Além de serem responsáveis por outras funções dentro de células (como a catalização de reações químicas), as moléculas de RNA também desempenham a função de cópias descartáveis de informações. O gene, que pode ser visto como o plano original para a produção de uma proteína, não precisa ser acessado e lido cada vez que uma proteína necessita ser produzida. O mecanismo intermediário da transcrição permite que diversas moléculas de RNA sejam criadas quando o gene estiver disponível para leitura, e mais tarde cada RNA pode ser traduzido várias vezes em proteínas e então descartado.

Um fato importante para a compreensão das redes de regulação gênica, assunto dos próximos parágrafos, é que quem coordena a leitura dos genes, e por consequência seus níveis de expressão, são os próprios genes. Essa coordenação é realizada por meio de proteínas denominadas *fatores de transcrição*, as quais se acoplam às regiões de regulação de

genes e regula o seu nível de expressão, aumentando ou reduzindo a taxa ou probabilidade de transcrição desses genes. Também é importante esclarecer que a regulação gênica muitas vezes é resultado de diferentes mecanismos biológicos (e.g. redes metabólicas e de sinalização), porém o escopo deste trabalho são as redes gênicas reguladas por fatores de transcrição, conforme descrito neste parágrafo.

Redes de regulação gênica são redes biológicas intracelulares que coordenam a maior parte dos processos celulares (ALBERT, 2004). Como as RRGs coordenam quais proteínas são produzidas na célula, elas são responsáveis por processos vitais como diferenciação celular, ciclo celular, apoptose (morte celular programada), adaptação a diferentes ambientes, entre outros. Na abstração Booleana destas redes, cada nodo equivale a um gene, que assume dois diferentes estados, ativado ou desativado. Se um gene está sendo transcrito, diz-se que o gene está ativo. Ao contrário, diz-se que o gene está inativo. Pode-se entender o estado de um gene como uma medida do quanto ele está expresso, i.e., a taxa da qual o maquinário celular lê informações deste gene para gerar proteínas. Na abstração Booleana, essa taxa assume somente dois valores. Ou a proteína está sendo produzida em uma taxa alta o bastante para influenciar outros genes (valor Booleano 1), ou não (valor Booleano 0).

Como já explicado, os fatores de transcrição se acoplam a partes específicas do DNA, o que pode resultar em um efeito inibidor ou ativador na expressão dos genes. Dessa maneira, um gene pode afetar vários outros, incluindo a si próprio. Além disso, cada gene pode ser afetado por diversos outros genes, quando vários fatores de transcrição diferentes “cooperam” para influenciar a expressão de um certo gene. Para completar, todo esse impacto dos fatores de transcrição sobre genes pode ocorrer em diferentes níveis de intensidade, conferindo a essas redes um relativo alto grau de complexidade.

Além da região codante, explicada na Seção 2.1, os genes também possuem uma região de regulação. Essa última regula em quais condições o gene estará expresso, ou seja, quando sua proteína correspondente será produzida e em qual taxa. Então, a região reguladora funciona como um sensor que verifica quais fatores de transcrição estão presentes no ambiente intracelular e regula a expressão do gene de acordo com essa informação.

A síntese de moléculas de RNA da informação codificada no gene é catalizada por uma enzima chamada RNA polimerase (RNAP), a qual deve se acoplar a uma região específica do gene chamada *região promotora* para que seja iniciada a síntese do mRNA. Então a probabilidade

do acoplamento da RNAP à região de regulação de um gene é um fator crucial para o início da transcrição. A modulação da expressão genética através de fatores de transcrição é um resultado do impacto que estes últimos têm nesta probabilidade de acoplamento da RNAP. Por exemplo, é possível, através de Biologia Sintética, projetar um *sítio de ligação* (região do DNA onde um fator de transcrição se acopla) sobreposto à região promotora de um gene. Então, sempre que este fator de transcrição estiver presente, ele vai se acoplar ao seu sítio de ligação, que sobrepõe a região reguladora onde a RNAP deve se acoplar para iniciar a transcrição, assim inibindo o gene. Por outro lado, é possível projetar genes nos quais a presença de certos fatores de transcrição facilitam o acoplamento da RNAP, ou seja, ativam o gene. Além dessas, existem diversas outras maneiras pelas quais é possível induzir certos comportamentos em genes sintéticos, vide (ALBERTS et al., 2013).

Para o entendimento do presente trabalho, este conceito básico de redes de regulação é suficiente, porém, caso haja interesse, é possível se aprofundar no assunto em (KARLEBACH; SHAMIR, 2008).

Na representação gráfica de RRGs, um arco direcionado do gene v_i para um gene v_j indica o efeito do fator de transcrição produzido por v_i , denotado por u_i , sobre a expressão (ativação ou inibição) do gene v_j . A Figura 3 mostra um exemplo de RRG.

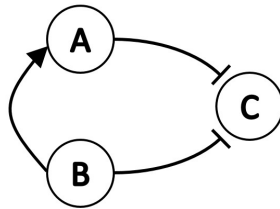


Figura 3 – Gráfico de uma RRG hipotética. Arcos terminados em setas indicam interação de ativação, e terminados em barra indicam interação de repressão. Por exemplo, o arco direcionado de B para C indica que a proteína codificado por B tem um efeito inibidor em C, e o arco direcionado de B para A indica que a proteína codificada por B tem um efeito ativador em A.

A RRG da Figura 3 é constituída por três genes, A, B e C. Os genes A e B produzem fatores de transcrição que influenciam a expressão do gene C, no sentido de reprimi-lo. O fator de transcrição produzido por B tem o efeito de ativar A. Então, C é expressado na ausência

das proteínas codificadas pelos outros dois genes e A é expressado na presença da proteína codificada por B .

Dado a grande quantidade de genes e interações entre eles em RRGs, se faz necessário a utilização de modelos matemáticos formais para tratar o problema de controle dessas redes. Ao abstrair o funcionamento dessas redes, é possível diminuir sua complexidade sem perder as informações pertinentes para seu controle. Um dos modelos possíveis faz uso de redes Booleanas e será apresentado a seguir.

2.2 REDES BOOLEANAS

Redes Booleanas são caracterizadas por um par (V, F) , em que $V = \{v_1, \dots, v_N\}$ é um conjunto de variáveis Booleanas $v_i \in \{0, 1\}$ e $F = \{f_1, \dots, f_N\}$ é um conjunto de funções Booleanas $f_i : \{0, 1\}^N \rightarrow \{0, 1\}$ usadas para atualizar o estado de v_i , baseado no presente valor de $\mathbf{v} = [v_1, \dots, v_N]$. A política de atualização dos estados pode ser síncrona, em que dois ou mais estados podem ser atualizados simultaneamente, ou assíncrona, em que somente um estado é atualizado por vez.

Redes Booleanas podem ser usadas para modelar o comportamento de redes de regulação gênica (ALBERT; OTHMER, 2003; SAMAL; JAIN, 2008). As variáveis v_i representam estados de genes (ativos ou inativos) e as funções Booleanas f_i (denominadas *funções de atualização*) descrevem as interações entre os genes, causada pelos fatores de transcrição. Em redes Booleanas, a expressão do gene é abstraída como valores Booleanos: 1 para gene ativo e 0 para gene inativo. Por exemplo, possíveis funções de atualização para os genes A e C da Figura 3 são $f_A = B$ e $f_C = \overline{A} \wedge \overline{B}$, respectivamente.

Note que não é possível chegar nas funções de atualização para os genes somente a partir da representação gráfica da rede, neste caso, Figura 3. São necessários testar esses genes em laboratório e determinar como eles se comportam na ausência ou presença dos fatores de transcrição. Por exemplo, a partir da Figura 3 sabemos que os genes A e B influenciam C no sentido de inibi-lo, porém não sabemos se é necessários ambos os fatores de transcrição de A e B estarem ausentes para expressar o gene C ou se somente um deles ausente já é suficiente. Nos exemplos deste trabalho sempre usaremos a operação Booleana AND para estes casos, como pode ser visto na função de atualização para o gene C : $f_C = \overline{A} \wedge \overline{B}$.

Diversos autores já utilizaram abstrações Booleanas para com-

prender fenômenos biológicos intracelulares. Em (KAUFFMAN; STROHMAN, 1994), foram introduzidas as redes Booleanas para modelagem de redes de regulação gênica. Já em (DATTA et al., 2007), é apresentada uma abordagem de controle de redes de regulação gênica modeladas como redes Booleanas, na qual o objetivo é conduzi-las de um estado inicial a um estado final em um dado intervalo de tempo e com o menor custo possível. E como um último exemplo, os autores de (BERNOT et al., 2004) trabalharam com extensões de redes Booleanas aplicadas em redes biológicas de regulação.

2.3 SISTEMAS A EVENTOS DISCRETOS

Sistemas a eventos discretos são sistemas dinâmicos que possuem um conjunto discreto de estados e cuja transição entre esses estados se dá pela ocorrência de eventos (WONHAM, 2015). Como exemplos, eventos podem ser o início e o término da execução de uma tarefa, um sinal de trânsito mudando de cor ou a ativação de um gene por um fator de transcrição. Nos SEDs, essas mudanças ocorrem de maneira abrupta, ou seja, após a ocorrência de um evento o sistema se acomoda instantaneamente, e permanece nesse estado até a ocorrência de um novo evento.

Sistemas a eventos discretos podem ser usados para representar vários sistemas físicos, como linhas de manufatura, sistemas de logística e até mesmo alguns sistemas biológicos. Um simples exemplo para ilustrar o conceito de SEDs seria uma linha de manufatura com duas máquinas e um *buffer* entre elas, conforme a Figura 4.

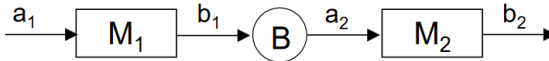


Figura 4 – Esquema de uma simples linha de manufatura como exemplo de um SED. A peça entra na máquina 1 (M_1), segue para o *buffer* (B), entra na máquina 2 (M_2) e sai dela como peça pronta.

Nesse exemplo, é fácil perceber que, independente da passagem de tempo, o sistema evolui conforme a ocorrência dos eventos a_1 (peça entra na máquina 1), b_1 (fim de operação da máquina 1, com o seu descarregamento de peça no *buffer*), a_2 (peça sai do *buffer* e entra

na máquina 2 para início de sua operação) e b_2 (fim de operação da máquina 2).

Podemos resumir a definição de sistema a eventos discretos como: sistema dinâmico que evolui de acordo com a ocorrência abrupta de eventos físicos, em intervalos de tempo em geral irregulares e desconhecidos.

A Teoria de Controle Supervisório é um formalismo para calcular controladores para os sistemas a eventos discretos. Ela será explicada na próxima seção.

2.4 TEORIA DE CONTROLE SUPERVISÓRIO

A Teoria de Controle Supervisório, proposta por Ramadge e Wonham (RAMADGE; WONHAM, 1987), é uma abordagem formal para síntese de controladores de sistemas a eventos discretos. A TCS é baseada na Teoria de Linguagens e Autômatos e originalmente, ela foi concebida para alterar a *linguagem gerada* por um dado sistema a ser controlado, também chamado de planta. Entende-se por linguagem gerada o conjunto de todas as possíveis sequências de eventos possíveis na planta. A intervenção no comportamento do sistema sob controle visa garantir que ele atenda a um conjunto de especificações, cada uma delas associada a uma linguagem contida na linguagem da planta. As especificações representam os comportamentos desejáveis (admissíveis) para o sistema. Para restringir o comportamento do sistema não controlado, utiliza-se um *supervisor*, uma entidade que observa os eventos gerados pela planta e, em função das sequências observadas, desabilita a ocorrência de certos eventos. Nem todos os eventos podem ser desabilitados pelo supervisor, fato que leva à partição dos eventos da planta em eventos controláveis e não controláveis.

Em nossa abordagem, o supervisor observa o estado em que a rede se encontra e restringe os eventos que podem acontecer ou, ainda, força certos eventos, para que as especificações de controle sejam atingidas. Os eventos forçáveis então são aqueles que o supervisor pode impor sua ocorrência, podendo preemptar outros eventos. A rede evolui segundo as determinações impostas pelo supervisor, que observa o novo estado alcançado por ela e reinicia mais um ciclo de medição e atuação, prosseguindo dessa maneira até que os estados finais sejam atingidos.

Na estrutura da TCS, a planta reflete o comportamento do sistema sem restrições. Todas as ações fisicamente possíveis estão descri-

tas na planta, inclusive as indesejadas ou problemáticas. Retomando o exemplo da Figura 4, pode-se supor que a capacidade do *buffer* seja de três peças, porém o modelo da planta possibilita que uma quantidade qualquer de peças seja depositada nele. O papel do supervisor é então exercer uma ação de controle restritiva sobre a planta, de modo que certas especificações sejam satisfeitas, uma vez que a capacidade do *buffer* não é uma limitação da planta.

A planta do sistema é modelada por um gerador, ou autômato, formado por uma quintupla $G = (Q, \Sigma, \delta, q_0, Q_m)$, sendo Q o conjunto de estados, Σ o conjunto de eventos, $\delta : Q \times \Sigma \rightarrow Q$ a função de transição parcial definida em cada estado de Q para um subconjunto de Σ , $q_0 \in Q$ o estado inicial e $Q_m \subseteq Q$ o subconjunto de estados marcados.

O supervisor interage com a planta G em uma estrutura em malha fechada, em que o supervisor observa o estado atual da planta e define quais eventos fisicamente possíveis são habilitados, definindo uma entrada de controle que é atualizada a cada nova ocorrência de evento observada em G . A Figura 5 ilustra a interação do supervisor com a planta em malha fechada.

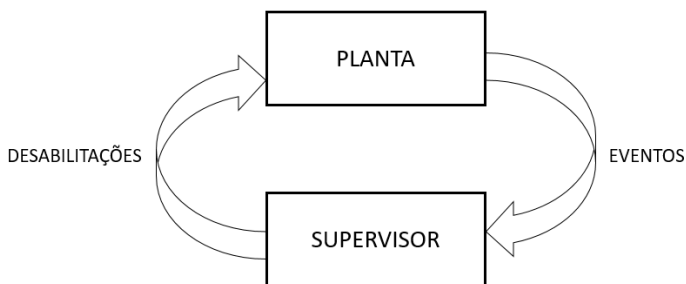


Figura 5 – Estrutura em malha fechada do controle supervisorio.

A Teoria de Controle Supervisorio entra no contexto desse trabalho como sendo o formalismo que nós utilizamos para derivar a lógica do supervisor que vai intervir no comportamento da rede de genes. Isso é possível pois redes de regulação gênica podem ser modeladas como sistemas a eventos discretos, e como já vimos, através da TCS é possível calcular supervisores para esses sistemas.

Já a realização do supervisor dentro de nossa abordagem, baseia-se na possibilidade de construir moléculas sintéticas capazes de efetuar

a operação lógica “se *estado*, então *ação*”, operação que, como vimos no capítulo anterior, já foi construída a partir de diferentes circuitos moleculares, dentre eles, circuitos compostos por genes sintéticos.

3 O PROBLEMA DE ATRAÇÃO DE ESTADOS PARA REDES DE REGULAÇÃO GÊNICA

Este capítulo é dedicado a explicar a abordagem de controle celular na qual o presente trabalho se baseia. Esta abordagem foi apresentada em (BALDISSERA; CURY; RAISCH, 2016) e é sustentada por três pilares: 1) é uma abordagem para controle de redes de regulação gênica intracelulares; 2) utiliza a Teoria de Controle Supervisório para derivar um controlador e 3) visa a implementação do controlador através da inserção de genes sintéticos nas células.

Para uma melhor compreensão de como o controle de RRGs através da TCS funciona, segue um exemplo hipotético proposto em (CURY; BALDISSERA, 2013), que é usado até o fim deste capítulo.

Inicialmente, os autores assumiram uma abstração Booleana para caracterizar os estados dos genes, isto é, foi considerado que eles podem estar ativados (sendo transcritos) ou desativados (transcrição não ocorrendo), também chamado de gene silenciado. Essa abstração é necessária para a construção da rede Booleana que modela o sistema.

Considere uma RRG constituída por três genes nativos v_1 , v_2 e v_3 , que codificam suas respectivas proteínas u_1 , u_2 e u_3 (que são fatores de transcrição, i.e., proteínas especiais que regulam a expressão de genes), e suas respectivas funções de atualização f_1 , f_2 e f_3 , conforme a Figura 6. Os genes interagem entre si através dos fatores de transcrição que eles codificam, u_1 , u_2 e u_3 . Observe que no exemplo de rede de regulação gênica da Seção 2.1, escrevemos as funções de atualização em função dos próprios genes v_i , porém a partir de agora iremos escrevê-las em função dos fatores de transcrição u_i codificados pelos genes v_i . A interação entre fatores de transcrição e genes é representada por arcos na Figura 6.

A partir desta representação de RRG em rede Booleana da Figura 6, pode-se construir um diagrama de transição de estados assíncrono, mostrado na Figura 7. A política de atualização assíncrona aqui considerada significa que a cada transição somente o estado de um gene é atualizado por vez. Cada estado representa qual fator de transcrição está presente na célula. Por exemplo, o estado 001 representa o estado celular no qual somente o fator de transcrição u_3 está presente, e assim por diante.

Para ilustrar como o diagrama de estados assíncronos é obtido a partir da RRG e das funções de atualização, considere o estado 000 na Figura 7. O estado 000 representa o estado celular no qual nenhum dos

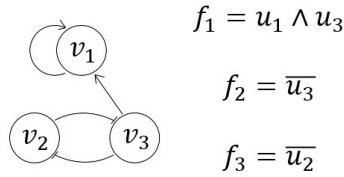


Figura 6 – Gráfico de uma RRG hipotética formada por três genes, v_1 , v_2 e v_3 . Arcos terminados em setas indicam interação de ativação, e terminados em barra indicam interação de repressão.

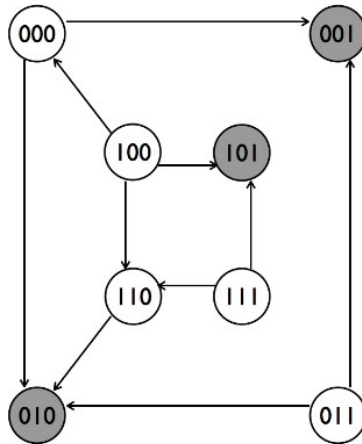


Figura 7 – Representação do diagrama de transição de estados assíncrono para a rede de genes da Figura 6. Cada estado é representado por uma cadeia binária que indica quais genes nativos v_i estão sendo expressados, i.e., quais fatores de transcrição u_i estão presentes na célula. Os estados hachurados correspondem aos atratores da rede, que são os estados que não possuem arcos saindo deles. Imagem extraída de (BALDISSERA; CURY; RAISCH, 2016).

três fatores de transcrição, u_1 , u_2 e u_3 , estão presentes, ou seja, nenhum dos três genes está expressado. Para saber como o sistema pode evoluir a partir de 000 temos que verificar as funções de atualização. Neste

estado, f_1 resulta em 0, então o gene v_1 permanece inativo e podemos ver no diagrama da Figura 7 que não existe um arco indo do estado 000 para 100. Já as funções f_2 e f_3 resultam em 1 no estado 000, e isso é refletido no diagrama pelos arcos que vão do estado 000 para 010 e de 000 para 001, respectivamente. Aplicando esse raciocínio para todos os oito possíveis estados, chegamos no diagrama da Figura 7. Repare que por causa da política de atualização assíncrona, as transições só alteram um *bit* do estado por vez. Então, não é possível ir do estado 000 direto para 011, por mais que este seja o resultado das três funções de atualização no estado 000. Inclusive, neste caso, 011 nem é um estado acessível.

Escolhendo um estado inicial q_0 , pode-se associar este diagrama a um autômato de estados finitos $G = (Q, \Sigma, \delta, q_0, Q_m)$, representado na Figura 8, em que:

- $Q = \{000, 001, 010, 011, 100, 101, 110, 111\}$ é o conjunto de estados;
- $\Sigma = \{+u_i, -u_i\}$, para $i = \{1, 2, 3\}$, é o conjunto de eventos que indicam, respectivamente, o início ou fim da transcrição da proteína u_i ;
- $\delta : Q \times \Sigma \rightarrow Q$ é a função de transição parcial;
- $q_0 \in Q$ é o estado inicial;
- Q_m é um conjunto de estados marcados e corresponde à especificação de estados a serem alcançados. A escolha de Q_m depende dos objetivos do controle.

Para intervir no comportamento da rede de genes da Figura 6, considere a inserção de dois genes sintéticos dentro da célula, v_1^s e v_3^s . Esses genes codificam as mesmas proteínas que v_1 e v_3 , respectivamente, ou seja, os fatores de transcrição u_1 e u_3 . Suas regiões reguladoras serão projetadas antes de inseri-los na célula, determinando em que estados os genes deverão ser ativados. A síntese do supervisor vai definir qual função Booleana deve ser computada por cada gene. A Figura 9 mostra a RRG da Figura 6 expandida, contendo os genes sintéticos. Nesta figura, a seta com interrogação que vai da camada nativa para a camada sintética indica que ainda não sabemos como a rede irá influenciar os genes sintéticos, ou seja, em quais estados eles serão expressado. Precisamos derivar o supervisor para saber quais funções os genes sintéticos devem ser projetados para computar, i.e., quais serão suas funções de

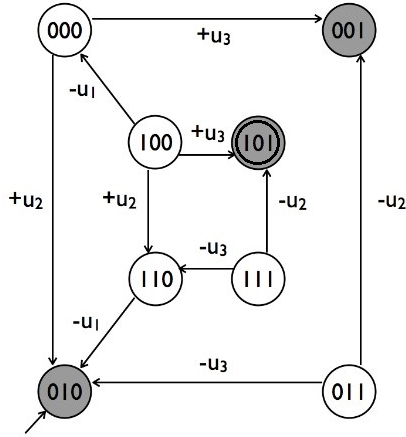


Figura 8 – Autômato G associado ao diagrama da figura 7. O estado inicial é indicado por uma seta ($q_0 = 010$) e o conjunto de estados marcados são os estados com duplo círculo ($Q_m = \{101\}$). Imagem extraída de (BALDISSERA; CURY; RAISCH, 2016).

atualização. Porém, note que já sabemos como os genes sintéticos influenciam a rede, que será sempre com os mesmos fatores de transcrição de seus associados nativos, indicados pelos subíndices. Então neste caso, v_1^s e v_3^s estão associados a v_1 e v_3 , respectivamente, de modo que eles codificam os mesmos fatores de transcrição.

A escolha das proteínas codificadas por cada gene sintético foi arbitrária, mas poderia ter sido consequência de limitações biológicas reais. Por exemplo, se a síntese da proteína u_2 possuir um custo energético elevado, inserir o gene sintético v_2^s poderia sobrecarregar a célula.

Nós supomos que a eficiência de transcrição e tradução dos genes sintéticos é superior à dos genes nativos, conferindo uma dinâmica mais rápida para a produção de proteínas por parte dos genes sintéticos. Isso quer dizer que um evento causado pela ativação de um gene sintético pode preemptar a ocorrência de eventos associados a genes nativos. Esta hipótese é suportada por (BLAZECK et al., 2012; GINGOLD; PILPEL, 2011) e ela será exemplificada mais a frente.

Agora pode-se modificar o autômato G para englobar as possibilidades conferidas pelos genes sintéticos, gerando um novo autômato

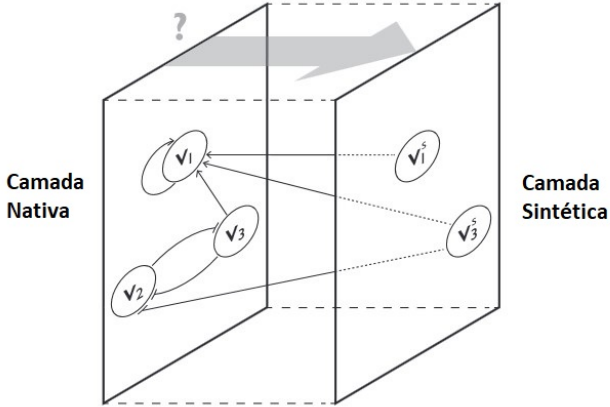


Figura 9 – RRG expandida com os genes sintéticos v_1^s e v_3^s . Imagem extraída de (BALDISSERA; CURY; RAISCH, 2016).

$G^{hi} = (X, \Sigma^{hi}, \delta^{hi}, x_0, X_m)$, representado na Figura 10. Este novo autômato G^{hi} somente difere de G pelas funções de transição e o conjunto de eventos, pois x_0 , X e X_m são iguais a q_0 , Q e Q_m , respectivamente.

Para controlar o comportamento de G , supomos então que estão disponíveis m genes sintéticos v_1^s, \dots, v_m^s , $m \leq n$ (n é o número de genes nativos), em que cada v_i^s pode: a) medir o estado x em que o sistema se encontra; e b) ser ativado ou desativado, dependendo do estado x . O gene sintético possui essas características pois podemos projetar eles para serem expressados (ativados) ou inibidos (desativados) em determinados estados da rede de genes, bastando projetar suas regiões de regulação de acordo. Lembrar que a região de regulação define na presença de quais fatores de transcrição (i.e., estado da rede) o gene será expressado. Se ele estiver ativado em x , então v_i^s : i) desabilita a ocorrência do evento $-u_i \in \Sigma$ se $\delta(x, -u_i)$ está definido em x ; ou ii) força a ocorrência do evento $+u_i$ se u_i não estiver sendo transcrita em x , ou seja, $v_i = 0$ (gene sintético i desativado). No caso do nosso exemplo, os m genes sintéticos são v_1 e v_3 .

Repare que na Figura 10 surgiram novos arcos e novas notações nos arcos, em relação a Figura 8, devido a inclusão dos genes sintéticos. Tudo que foi incluído devido aos genes sintéticos está destacado em cinza. Essas novas transições surgem porque não sabemos o que vai causar a ativação ou desativação dos genes sintéticos ainda, então G^{hi}

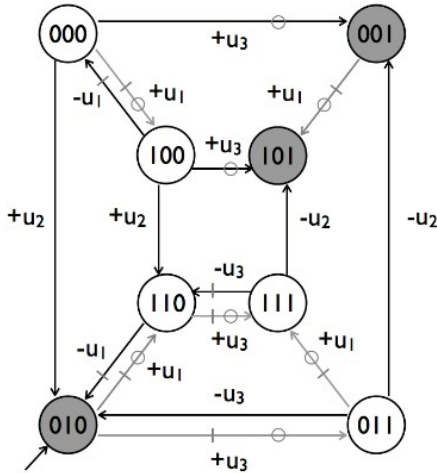


Figura 10 – Autômato G^{hi} com novas interações em cinza. Imagem extraída de (BALDISSERA; CURY; RAISCH, 2016).

engloba todas as novas possibilidades. Por exemplo, não existia uma transição do estado 010 para 011 no autômato da rede nativa (Figura 8), pois o gene nativo v_3 não codifica o fator de transcrição u_3 no estado 010 (presença somente de u_2). Com a adição do gene sintético v_3^s essa transição se torna possível, pois podemos sintetizar esse gene de maneira que ele implemente uma função que o ative no estado 010, o que resulta na produção do fator de transcrição u_3 .

Um traço em um arco da Figura 10 significa evento controlável e um círculo significa evento forçável. Os eventos forçáveis são somente do tipo $+u_i$. Um exemplo de evento forçável na Figura 10 é o evento $+u_1$ que vai de 000 para 100. Este evento pode ser forçado a acontecer pelo supervisor, e se ele for forçado ele vai preemptar todos os outros eventos que saem de 000, no caso $+u_2$ e $+u_3$. Isso acontece devido a hipótese de que a dinâmica dos genes sintéticos é mais rápida do que a dos genes nativos explicada na Seção 2.1.

Já um evento controlável é aquele que pode ser desabilitado pelo supervisor, pode ser um evento do tipo $+u_i$ ou $-u_i$. Por exemplo, a transição $-u_3$ que vai de 111 para 110 na Figura 10 faz parte da rede nativa, porém, com a adição dos genes sintéticos, este evento agora se torna um evento controlável. Isso acontece porque agora é possível projetar o gene sintético v_3^s para ser ativado no estado 111, o que impede

a ocorrência de $-u_3$, levando em conta a hipótese de que a produção de proteínas u_3 por parte do gene sintético v_3^s tem uma dinâmica mais rápida do que a degradação da proteína u_3 .

Se um evento $+u_i$ é forçado, ele preempta a ocorrência de todos os outros eventos (preempção forte). Note então que nesta abordagem o conjunto de eventos Σ é particionado em subconjuntos Σ_{cf} , $\Sigma_{c\bar{f}}$, $\Sigma_{\bar{c}f}$ e $\Sigma_{\bar{c}\bar{f}}$, nos quais o subscrito “c” (“c”) significa “controlável” (“não controlável”), e “f” (“f”) significa “forçável” (“não forçável”).

Cada evento em Σ é classificado como controlável (na Figura 10, representado pelos arcos marcados com traço perpendicular), i.e. podem ser desabilitados pela ação de um supervisor, ou não controláveis. Esses últimos não podem ser desabilitados, porém podem ser preemptados pelos chamados eventos forçáveis (representados pelos arcos com círculo na Figura 10). O raciocínio por trás da classificação de cada evento em controlável/não controlável e forçável/não forçável é discutido em detalhes em (BALDISSERA; CURY; RAISCH, 2016).

O problema é então formulado dentro da estrutura de controle supervisorio de Sistemas a Eventos Discretos. Sendo assim, o supervisor (controlador) para o autômato G^{hi} é definido como uma função $S : X \rightarrow 2^\Sigma$, ou seja, para cada estado $x \in X$ o supervisor associa um subconjunto de eventos em Σ^{hi} , que denotam os eventos que são permitidos em seguida, i.e., eles podem ser habilitados ou forçados. O problema de atração de estados consiste em conduzir o sistema do estado inicial x_0 para o conjunto de estados marcados X_m e mantê-lo em X_m indefinidamente. Para alcançar essa especificação, é definido um conjunto de *entradas de controle admissíveis* em x , denotado por $A(x)$, em que cada membro de $A(x)$ assume uma das seguintes formas:

1. $\phi(x) = \{\sigma_f\}$, com $\sigma_f \in \Sigma_f(x)$;
2. $\phi(x) = \zeta \cup \Sigma_{\bar{c}}(x)$, com $\zeta \in 2^{\Sigma_c(x)}$,

em que $\Sigma_{\bar{c}}(x) \doteq \Sigma_{\bar{c}f}(x) \cup \Sigma_{\bar{c}\bar{f}}(x)$, $\Sigma_c(x) \doteq \Sigma_{cf}(x) \cup \Sigma_{c\bar{f}}(x)$, e $\Sigma_f(x) \doteq \Sigma_{cf}(x) \cup \Sigma_{\bar{c}f}(x)$. Repare que $\Sigma(x)$ é o conjunto de eventos possíveis de acontecerem em x , e essa função pode ser delimitada dependendo do subíndice.

Note que denotamos uma entrada de controle específica para um estado x por $\phi(x)$, o que difere do conjunto $A(x)$ que denota todas as entradas de controle admissíveis para o estado x . Matematicamente, $\phi(x) = \{\sigma \mid \sigma \in \Sigma(x)\}$ e $A(x) = \{\phi(x) \mid \phi(x) = \{\sigma_f\} \text{ ou } \phi(x) = \zeta \cup \Sigma_{\bar{c}}(x)\}$.

O primeiro tipo de entrada de controle é usado para preempitar certos eventos ao forçar a ocorrência de um evento σ_f . Por exemplo, na Figura 10, a entrada de controle admissível $\phi(011) = \{+u_1\}$ preempita a ocorrência de ambos $-u_2$ e $-u_3$. O segundo tipo é uma intervenção de controle mais passiva, em que o controlador meramente desabilita alguns eventos para levar o sistema à direção desejada. Novamente tomando a Figura 10 como exemplo, repare que desabilitar $-u_3$ em 111, deixa o sistema com somente a opção do evento $-u_2$, ou seja, a entrada de controle admissível $\phi(111) = \{-u_2\}$.

Um caso interessante que podemos analisar é a entrada de controle admissível $\phi(011) = \{+u_1, -u_2, -u_3\}$ para o autômato da Figura 10. Como o evento $+u_1$ só pode ocorrer devido a ativação do gene sintético v_1^s , e como isso implica que o evento $+u_1$ irá preempitar os eventos $-u_2$ e $-u_3$, então essa entrada de controle é equivalente a $\phi(011) = \{+u_1\}$. Porém, se o evento $+u_1$ já existisse na rede nativa, então seria possível realizar a entrada de controle $\phi(011) = \{+u_1, -u_2, -u_3\}$, bastando não ativar v_1^s no estado 011.

É denotado por $H(x, \phi(x))$ o conjunto de todos os estados y que podem ser alcançados de x sob a entrada de controle admissível $\phi(x)$. Este conjunto também será chamado de imagem de x sob $\phi(x)$.

Uma proposição declarando as condições necessárias e suficientes para a existência de um supervisor que resolve o problema de atração de estados apresentado aqui pode ser encontrada em (BALDISSERA; CURY; RAISCH, 2016).

Note que as funções de atualização de cada gene sintético podem ser representadas por disjunções de conjunções, onde cada cláusula é associada a um estado específico em que o gene deve ser ativado. Além disso, para implementar uma função na forma normal disjuntiva com k cláusulas, pode-se fazer uso de k cópias do gene sintético, cada um implementando uma cláusula. Depois que eles são devidamente “programados”, a proposta é inserir os genes sintéticos dentro de células para fechar o ciclo de controle que conduz a rede para os estados desejados, i.e. induzem o comportamento celular desejado. De um ponto de vista biológico, a realização de um supervisor consiste em projetar genes sintéticos com regiões de regulação que computam as funções Booleanas dadas por f_i^s .

Também note que todas as ações de controle (forçar, habilitar ou desabilitar eventos) executadas por um supervisor $S : X \rightarrow 2^\Sigma$ em um dado estado x são realizadas somente pela *ativação* e/ou *repressão* de genes sintéticos nesses estados. Essas são as únicas duas ações que o supervisor pode exercer. Então para forçar um evento, por exemplo, o

supervisor irá ativar um gene sintético, e para habilitar ou desabilitar eventos, o supervisor reprime um ou mais genes sintéticos.

Resumindo, com relação a G , G^{hi} acrescenta eventos controláveis e forçáveis, muda a natureza da controlabilidade de alguns eventos e possibilita forçar eventos que antes não eram forçáveis. Agora só resta derivar um supervisor.

Primeiro, é necessário estabelecer uma especificação arbitrária. Suponha que deseja-se levar a rede de genes do estado atrator $x_0 = 010$ para $X_m = \{101\}$. Ou seja, é necessário alcançar um estado fisiológico em que somente os fatores de transcrição u_1 e u_3 estão presentes na célula. Os estados atratores podem corresponder, por exemplo, a diferentes tipos celulares do processo de diferenciação da célula.

Pela Figura 10, é possível verificar que o caminho 010-110-111-101 é uma possível solução para o problema proposto. Para percorrer este caminho deve-se forçar o evento $+u_1$ em 010, forçar $+u_3$ em 110 e desabilitar $-u_3$ em 111. Forçar $+u_1$ em 010 requer a ativação do gene sintético v_1^s neste mesmo estado. Assim como forçar $+u_3$ em 110 requer a ativação do gene sintético v_3^s nesse estado. Em 111, v_3^s deve ser ativado para desabilitar a transição $-u_3$, i.e., o gene sintético irá codificar o fator de transcrição u_3 , impedindo que a rede retorne para o estado 110 e restando somente a opção de seguir para o estado 101. Em 101 nenhum gene sintético precisa ser ativado, pois não existe nenhum evento saindo deste estado (estado marcado), então não há necessidade de nenhuma ação de controle para manter a rede neste estado. Resumindo, v_1^s deve ser expressado somente em 010 e v_3^s em 110 e 111. Portanto, as funções de atualização de v_1^s e v_3^s , em termos dos fatores de transcrição, podem ser escritas como:

$$\begin{aligned} f_1^s &= \overline{u_1} \wedge u_2 \wedge \overline{u_3} \\ f_3^s &= u_1 \wedge u_2 \end{aligned}$$

Uma maneira de implementar as funções f_1^s e f_3^s na região reguladora dos genes v_1^s e v_3^s , respectivamente, pode ser vista na Figura 11. O parágrafo seguinte explica como essa implementação funciona biologicamente, de uma maneira simplificada porém de fácil compreensão.

Na Figura 11, repare que a região de regulação do gene sintético v_1^s foi projetada de maneira a sobrepor o sítio de ligação das proteínas u_1 e u_3 (O_{u_1} e O_{u_3} , respectivamente) à região promotora do gene v_1^s . Desta forma, a ligação de qualquer uma das duas proteínas a v_1^s reprime a ativação de v_1^s . Além disso, pode-se empregar uma região promotora com baixa afinidade pela RNAP, de tal maneira que seja necessário o suporte de uma outra proteína em um sítio de ligação para estabele-

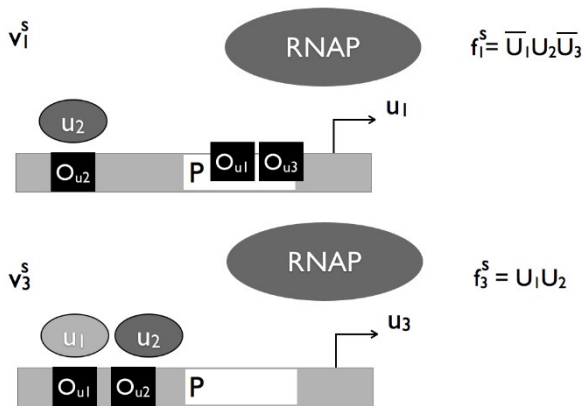


Figura 11 – Representação das regiões reguladoras dos genes v_1^s e v_3^s . Imagem extraída de (BALDISSERA; CURY; RAISCH, 2016).

zar, neste caso a proteína u_2 no sítio O_{u_2} . Assim, este gene sintético computa a função Booleana f_1^s desejada, sendo que ele só vai estar expresso na presença de u_2 e na ausência de ambos u_1 e u_3 , ou matematicamente, $f_1^s = \bar{u}_1 \wedge u_2 \wedge \bar{u}_3$. A mesma lógica pode ser usada para o gene sintético v_3^s da Figura 11, só que neste caso não existem sítios de ligação sobrepondo a região promotora, e sim dois sítios que suportam o acoplamento da RNAP, O_{u_1} e O_{u_2} , ou seja, v_3^s só é expressado na presença de ambos u_1 e u_2 . Note que para obter a operação de negação das variáveis Booleanas, basta projetar sítios de ligação sobrepostos à região promotora do gene, e para obter o contrário, pode-se projetar sítios de ligação na região reguladora dos genes.

O passo descrito no parágrafo anterior corresponde à “programação” das expressões Booleanas de f_1^s e f_3^s nas regiões reguladoras dos genes sintéticos v_1^s e v_3^s , respectivamente; estes genes devem então ser inseridos na célula para fechar o ciclo de controle. O problema é que não é tão simples assim “programar” genes para computar qualquer expressão Booleana. Existem diversas restrições biológicas nesse processo, como já explicado na introdução. No exemplo resolvido neste capítulo, suponha que exista uma restrição biológica determinada que só é possível implementar regiões reguladoras que computam as seguintes funções Booleanas: \bar{u}_3 e $u_1 \wedge u_2$.

Dada esta restrição, o supervisor S que foi derivado para o exemplo deste capítulo não seria implementável. O objetivo deste trabalho é justamente criar um formalismo para derivar supervisores implementáveis sujeitos a esse tipo de restrição.

4 O PROBLEMA DE ATRAÇÃO DE ESTADOS SOB RESTRIÇÕES BIOLÓGICAS

Neste capítulo, apresentaremos o método desenvolvido para resolução do problema proposto nesta dissertação. Primeiro, vamos formalizar o problema e em seguida expor um exemplo simples e didático passo a passo para facilitar a compreensão do método de resolução. Finalmente, o capítulo é encerrado com a formalização do método e a apresentação de um possível algoritmo para sua implementação.

4.1 DECLARAÇÃO DO PROBLEMA

Problema 1: Seja $G = (X, \Sigma, \delta, x_0, x_m)$ um autômato obtido de uma RRG (como explicado no Capítulo 3) e R o conjunto de todas as funções Booleanas que podem ser implementadas por genes sintéticos. Encontrar um supervisor $S : X \rightarrow 2^\Sigma$ tal que: a) é garantido que o sistema em malha fechada alcança e se mantém no estado x_m , partindo de x_0 (veja o problema de atração de estados no Capítulo 3); b) a realização do supervisor em termos de genes sintéticos é feita exclusivamente por funções em R .

É suposto que as funções Booleanas em R estão na forma normal disjuntiva, i.e. elas são disjunções de conjunções. Além disso, para o funcionamento correto do algoritmo desenvolvido, sem perda de generalidade, é suposto que os genes sintéticos disponíveis são $v_1^s, v_2^s, \dots, v_m^s$, com índices em sequência e iniciado por 1. Por exemplo, se para o caso da RRG da Figura 3 são escolhidos os genes sintéticos v_1^s e v_3^s para a realização do seu controle, então deve-se trocar o nome dos genes v_2 e v_3 para que o índice dos genes sintéticos fiquem em sequência e iniciem por 1, ou seja v_1^s e v_2^s .

Também note que assumimos que há somente um estado marcado x_m , para fins de simplificação. Porém, é possível estender o método de resolução apresentado aqui para um conjunto de estados marcados X_m .

Por fim, repare que o conjunto R é considerado informação dada para o problema. Ele não possui estrutura e nem relação com os genes. Supostamente ele é o conjunto de todas as funções Booleanas que podem ser implementadas por genes sintéticos, e é dado pelas limitações em Biologia Sintética. O conjunto R pode variar conforme necessidade. Por exemplo, suponha que sintetizar genes sintéticos que são ativados

pela proteína u_2 seja muito custoso, então pode-se definir R como o conjunto das funções Booleanas que não envolvam u_2 , mas somente $\overline{u_2}$.

4.2 EXEMPLO DIDÁTICO

Nesta seção vamos apresentar e resolver passo a passo um exemplo didático de controle de RRG sob restrições biológicas.

Considere a RRG da Figura 12. Nesta RRG, a expressão do gene v_1 influencia o gene v_2 de forma a ativá-lo. Além disso, v_1 é ativado pelo gene v_3 e reprimido pelo gene v_2 . Já o gene v_2 é ativado pelo gene v_1 , reprime o gene v_1 e ativa v_3 . Por fim, v_3 ativa v_1 e é ativado por v_2 . Suponha que as funções de atualização para os genes v_1 , v_2 e v_3 são $f_1 = \overline{v_2} \wedge v_3$, $f_2 = v_1$ e $f_3 = v_2$, respectivamente.

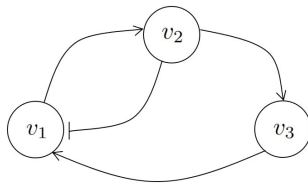


Figura 12 – Gráfico de uma RRG hipotética utilizada no exemplo da Seção 4.2.

Agora considere o autômato da Figura 13. Ele contém os estados e eventos associados com a dinâmica da rede de genes da Figura 12 junto com os eventos que podem ser induzidos pela ação dos genes sintéticos v_1^s e v_2^s , escolhidos arbitrariamente para este exemplo. Seus estados $x \in X$ são nomeados por cadeias binárias de comprimento n , os quais representam os estados individuais dos n genes de G . Os genes sintéticos escolhidos, v_1^s e v_2^s , sintetizam as proteínas u_1 e u_2 , respectivamente. Note que os genes nativos v_1 e v_2 também codificam as proteínas u_1 e u_2 , respectivamente. Observe que o que difere os genes sintéticos dos seus correspondentes nativos é qual combinação de fatores de transcrição presentes no ambiente celular ativa ou reprime suas expressões, pois a proteína que eles codificam sempre será a mesma. Ou seja, somente as suas regiões de regulação são distintas. Então, por exemplo, ambos v_1^s e v_1 codificam o fator de transcrição u_1 , porém suas funções de atualização são diferentes, i.e., $f_1^s \neq f_1$.

O propósito do método desenvolvido neste trabalho é justamente

calcular quais funções Booleanas as regiões de regulação dos genes sintéticos devem computar, considerando somente funções definidas em um subconjunto R de funções.

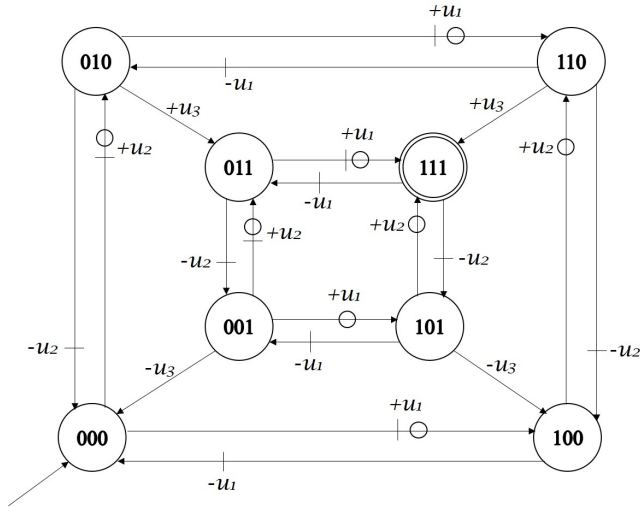


Figura 13 – Autômato G associado à RRG hipotética da Figura 12. Uma barra no arco significa evento controlável e um círculo no arco significa evento forçável.

O autômato da Figura 13 é dado por $G = (X, \Sigma, \delta, x_0, x_m)$, em que:

- X é o conjunto de estados, em que cada estado x é uma cadeia de variáveis Booleanas que possui 1 (0) na i -ésima posição se a proteína u_i está (não está) sendo sintetizada no estado x (independente de se u_i está sendo sintetizado pelo gene nativo v_i ou pelo gene sintético v_i^s);
- $\Sigma = \{+u_i, -u_i\}$ é o conjunto de eventos (produção ou degradação de fatores de transcrição);
- $\delta : X \times \Sigma \rightarrow X$ é a função de transição parcial;
- $x_0 \in X$ é o estado inicial;
- $x_m \in X$ é o estado marcado, estado alvo a que a rede deve ser levada.

É fácil verificar no autômato da Figura 13 que o caminho 000-100-110-111 é uma possível solução para o problema proposto. Para percorrer esse caminho, o evento $+u_1$ deve ser forçado no estado 000, $+u_2$ deve ser forçado ou $-u_1$ desabilitado em 100, e ambos $-u_1$ e $-u_2$ devem ser desabilitados em 110 e 111.

Forçar $+u_1$ em 000 requer a ativação do gene sintético v_1^s e a desativação de v_2^s neste estado. Em 100, o único caso que não deve acontecer é a desativação simultânea de v_1^s e v_2^s . Em 110, ambos genes sintéticos v_1^s e v_2^s devem ser ativados para que os eventos $-u_1$ e $-u_2$ sejam desabilitados e o evento $+u_3$ ocorra. Finalmente, no estado alvo 111, ambos genes sintéticos devem ser ativados novamente, para garantir que o sistema irá permanecer no estado 111.

Em resumo, v_1^s deve ser ativado somente em 000, 100, 110 e 111, e v_2^s somente em 100, 110 e 111. Portanto, as funções de atualização de v_1^s e v_2^s , em termos dos fatores de transcrição, podem ser escritas como:

$$f_1^s = (\overline{u_1} \wedge \overline{u_2} \wedge \overline{u_3}) \vee (u_1 \wedge \overline{u_2} \wedge \overline{u_3}) \vee (u_1 \wedge u_2 \wedge \overline{u_3}) \vee (u_1 \wedge u_2 \wedge u_3) \text{ e}$$

$$f_2^s = (u_1 \wedge \overline{u_2} \wedge \overline{u_3}) \vee (u_1 \wedge u_2 \wedge \overline{u_3}) \vee (u_1 \wedge u_2 \wedge u_3).$$

Dado um conjunto de fatores de transcrição, há restrições com relação a quais funções Booleanas podem ser computadas por uma região de regulação gênica (ver Seção 2.1). No âmbito da abordagem de (BALDISSERA; CURY; RAISCH, 2016), isso se traduz em funções de atualização f_i^s que com frequência não podem ser qualquer função Booleana. Portanto, para realizar o supervisor como genes sintéticos, é necessário chegar em funções de atualização viáveis para os genes sintéticos.

Seja G o autômato da Figura 13 e $R = A^\dagger$, sendo $A = \{u_1 \wedge u_3, \overline{u_2} \wedge u_3, \overline{u_3}\}$ e A^\dagger denota o fechamento do conjunto A para a operação de disjunção. Ou seja, estamos definindo o conjunto de restrições R para esse exemplo como sendo o conjunto formado pelas três funções Booleanas de A e todas as combinações de disjunções entre elas. Então, por exemplo, a função $u_1 \wedge u_3 \vee \overline{u_3}$ pertence a R . O conjunto A é somente um conjunto auxiliar para facilitar a definição de R .

Para resolver o problema proposto no início do Capítulo 4, uma árvore de nodos $T = (N, E)$ será construída, em que N é o conjunto de nodos e E é o conjunto de arcos direcionados. A seguir, serão descritos a semântica de cada nodo em N e o processo de construção da árvore T para o exemplo enunciado no parágrafo anterior.

O primeiro passo consiste em verificar se existe uma entrada de controle para o estado marcado tal que $H(x_m, \phi(x_m)) = \{x_m\}$, ou seja, $\phi(x_m) = \emptyset$. Se essa entrada de controle existe, então é possível manter o sistema no estado marcado após alcançá-lo. Para o exemplo, essa

entrada de controle existe e pode ser implementada pela ativação dos dois genes sintéticos v_1^s e v_2^s em 111. As seguintes funções Booleanas de R podem ser usadas para esse propósito:

$$B = \{u_1 \wedge u_3, (u_1 \wedge u_3) \vee (\overline{u_2} \wedge u_3), (u_1 \wedge u_3) \vee \overline{u_3}, (u_1 \wedge u_3) \vee \overline{u_3} \vee (\overline{u_2} \wedge u_3)\}.$$

O primeiro nodo de T , n_0 , é então definido como $(x_m, \phi(x_m), Q_{n_0})$, no qual $Q_{n_0} = (Q_{n_0}^1, Q_{n_0}^2)$ com $Q_{n_0}^1 = B$ e $Q_{n_0}^2 = B$ (iguais no caso desse exemplo, mas poderiam ser diferentes). $Q_{n_0}^1$ e $Q_{n_0}^2$, portanto, contêm o conjunto de funções Booleanas em R que implementam $\phi(x_m)$ para os genes sintéticos v_1^s e v_2^s , respectivamente. Note que um nodo da árvore T é definido como uma tripla formada por um estado $x \in X$, uma entrada de controle $\phi(x) \in A(x)$ para este mesmo estado x e uma tupla Q_n . A tupla Q_n por sua vez é formada por m conjuntos de funções Booleanas, um para cada gene sintético.

Seja $C_0 = \{n_0\}$, um conjunto formado por todos os nodos da primeira iteração, que sempre será unitário, contendo somente n_0 , por construção. Os próximos passos consistem em computar, iterativamente, uma sequência de conjuntos de nodos C_k , começando por C_0 , até alcançar o critério de parada que será exposto mais a frente. Para obter o conjunto C_1 , primeiro os estados adjacentes a 111 são identificados, isto é, os estados que podem alcançar 111 em somente um passo. Para cada um desses estados adjacentes, é verificado se existe uma entrada de controle admissível que conduz o sistema para 111, o qual é o estado do nodo n_0 . A figura 14 é uma representação gráfica dos passos descritos até o momento e mostra as entradas de controle que satisfazem essa condição e os valores de v_i^s necessários para implementá-las.

O próximo passo é verificar se a ação de controle que leva x a 111 pode ser implementada por funções Booleanas que são compatíveis com $Q_{n_0}^1$ e $Q_{n_0}^2$. Denominamos este procedimento *herança de restrições*. A herança de restrições para $\phi_1(101)$ da Figura 14 ocorre da seguinte maneira. São verificados quais elementos de $Q_{n_0}^2$ são avaliados em 1 no estado 101 (pois deseja-se que $v_2^s = 1$ em 101 para que $\phi_1(101)$ seja implementado). Note que o índice superior dos conjuntos Q_n^i correspondem aos índices dos genes sintéticos v_i^s , então $Q_{n_0}^2$ está associado a v_2^s e assim por diante. Todos os quatro elementos de $Q_{n_0}^2$ satisfazem essa condição, portanto todas elas são herdadas para $Q_{n_1}^2$, então $Q_{n_1}^2 = Q_{n_0}^2$. Já o conjunto $Q_{n_1}^1$ é igual a $Q_{n_0}^1$, pois o gene v_1^s não tem influência na realização de $\phi_1(101)$. Como todos os Q_{n_1} são diferentes de vazio, i.e. $Q_{n_1}^1 \neq \emptyset$ e $Q_{n_1}^2 \neq \emptyset$, então $n_1 = (101, \phi_1(101), Q_{n_1})$ é um

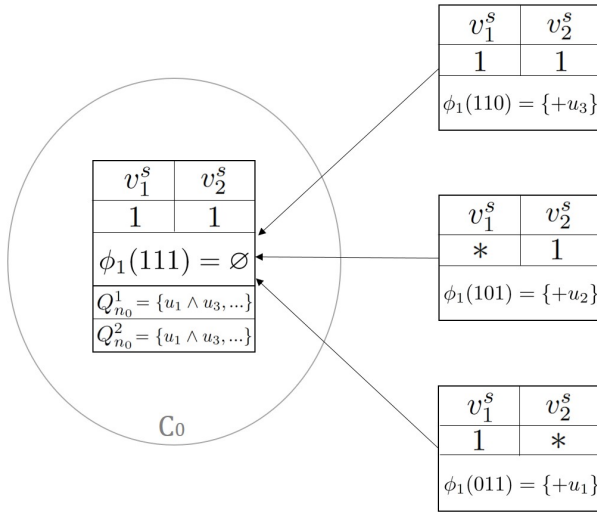


Figura 14 – Primeiros passos na construção da árvore T usada para resolver o exemplo da Seção 4.2. A segunda linha de cada caixa mostra o padrão de ativação dos genes sintéticos necessário para executar a entrada de controle na terceira linha (1 para ativar o gene, 0 para desativar e * para “não importa”).

nodo que pertence ao conjunto C_1 . Note que se ao menos um conjunto Q_n for igual a vazio, isso significa que para ao menos um gene sintético, não há funções Booleanas que possam implementá-lo sem que haja conflito “ao longo do caminho” até o estado alvo. A adição do nodo n_1 ao conjunto C_1 significa que é possível conduzir o sistema de 101 até o estado marcado usando qualquer uma das funções Booleanas de $Q_{n_1}^i$ para implementar o gene sintético v_i^s .

O mesmo procedimento descrito no último parágrafo é aplicado para todas as outras entradas de controle dos estados adjacentes a 111 e que levam a 111. No caso do exemplo, elas são $\phi_1(110)$ e $\phi_1(011)$. O resultado dessa iteração é $C_1 = \{(101, \phi_1(101), Q_{n_1}), (110, \phi_1(110), Q_{n_2})\}$. O conjunto C_1 possui dois nodos. Note que não existe nodo em C_1 associado ao estado 011. Isso ocorre porque o processo de herança de restrições aplicado a $\phi_1(011)$ resulta em pelo menos um conjunto Q vazio, ou seja, não existe função Booleana que possa implementar os genes sintéticos para que $\phi_1(011)$ seja executado e que induza o sistema

a chegar ao estado marcado. A Figura 15 resume os passos executados até o momento.

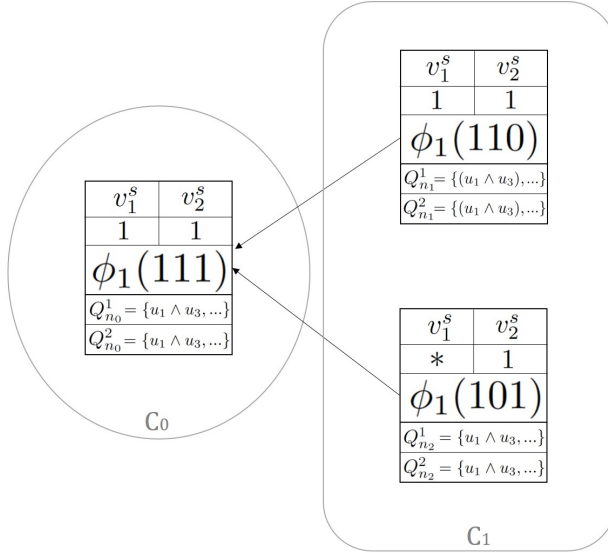


Figura 15 – Conjuntos de nodos C_0 e C_1 para o exemplo da Seção 4.2.

O mesmo raciocínio é aplicado para o cálculo da seqüência C_k para todo k até que seja alcançado um C_n ($n \geq 1$) e aconteça qualquer um dos critérios de parada abaixo:

1. $n_j = (x_0, \phi(x_0), Q_{n_j}) \in C_n$, o que indica que existe um caminho do estado inicial ao estado final que resolve o problema; ou
2. $C_n = \emptyset$, o que significa que não existe solução para o problema.

Se um nodo associado ao estado inicial foi adicionado a um conjunto C_n , é trivial que existe um caminho que liga o estado inicial ao final e que pode ser implementado por genes sintéticos usando somente funções Booleanas de R . Por outro lado, se a iteração para porque algum C_n é vazio e um nodo associado a x_0 não pertence a $\bigcup_{k=0}^{k=n} C_k$, então é evidente que um caminho que soluciona o problema não existe. Repare que este método tem como objetivo encontrar somente uma solução, a solução com o caminho de menor número de passos entre estados inicial e final, pois o algoritmo para assim que ela é encontrada. Porém, isso não quer dizer que não existam outras soluções.

A Figura 16 consiste na árvore que resulta da aplicação do raciocínio descrito nos últimos parágrafos ao exemplo apresentado nesta seção. Observe que a condição (1) é alcançada para o nodo n_8 em C_5 . Na Tabela 1 consta o supervisor que é derivado da árvore encontrada. A implementação desse supervisor é dada pelos genes sintéticos v_1^s e v_2^s com funções de atualização f_1^s e f_2^s que podem ser iguais a qualquer função de $Q_{n_8}^1 = \{(u_1 \wedge u_3) \vee (\bar{u}_2 \wedge u_3)\}$ e $Q_{n_8}^2 = \{(u_1 \wedge u_3) \vee \bar{u}_3\}$, respectivamente. Note que essas funções satisfazem as restrições apresentadas em R , o que significa que existe um caminho viável de 000 para 111. Repare também que no caso deste exemplo, cada nodo está ligado somente a um outro nodo, mas isso não é regra. Podem haver caminhos soluções com ramificações não determinísticas, contanto que o sistema saia do estado inicial, alcance o estado final e se mantenha nele.

Algo interessante de notar na solução deste exemplo é que não foi encontrado um caminho solução de três passos, mas sim um de cinco passos. Isso acontece porque todas as soluções de três passos são eliminadas no processo de herança de restrições, ou seja, não é possível implementar um caminho de três passos usando somente funções do conjunto R . Então na construção do conjunto C_3 , todos os candidatos a nodos que possuem como estado o estado inicial 000, são eliminados. Também poderia acontecer a situação em que o único caminho viável, dado as restrições, ser um caminho de sete passos, que percorre todos os estados da rede.

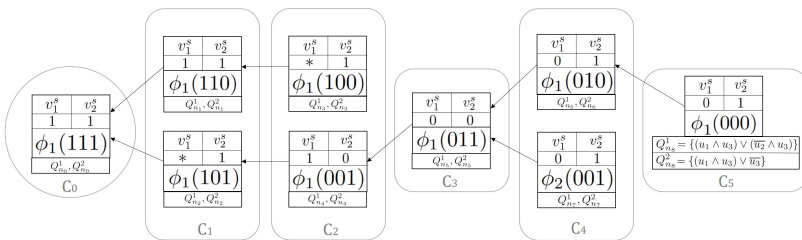


Figura 16 – Árvore associada à solução do exemplo da Seção 4.2. A cada iteração, o conjunto C_k é computado a partir de C_{k-1} , como explicado no texto. O critério de parada foi alcançado quando o nodo n_8 de C_5 foi calculado, pois sua projeção no espaço de estado é o estado inicial x_0 , significando que existe uma solução para este problema.

Tabela 1 – Supervisor $S : X \rightarrow 2^\Sigma$ encontrado para o exemplo da Seção 4.2 e suas implicações.

Estados	S(x)	Eventos preemptados ou desabilitados	Ativação requerida
000	$\{+u_2\}$	$\{+u_1\}$	v_2^s
010	$\{+u_3\}$	$\{+u_1, -u_2\}$	v_2^s
011	$\{-u_2\}$	$\{+u_1\}$	–
001	$\{+u_1\}$	$\{+u_2, -u_3\}$	v_1^s
101	$\{+u_2\}$	$\{-u_1, -u_3\}$	v_2^s
111	$\{\}$	$\{-u_1, -u_2\}$	v_1^s, v_2^s

4.3 SOLUÇÃO GERAL

Neste capítulo apresentamos um procedimento que resolve o problema declarado no Capítulo 4, o Problema 1, para uma rede de regulação gênica e conjunto de restrições R arbitrários. Este algoritmo recebe como entradas:

- $G = (X, \Sigma, \delta, x_0, x_m)$: autômato que modela a planta junto as possíveis entradas de controle.
- R : conjunto de todas as funções Booleanas que podem ser biologicamente implementadas.

Como saída, o algoritmo retorna, se existir, um supervisor $S(x)$ que conduz o sistema de x_0 para x_m e pode ser realizado por genes sintéticos v_i^s dos quais as funções de ativação f_i^s consistem apenas de funções em R .

4.4 DESCRIÇÃO DO ALGORITMO

O algoritmo desenvolvido foi dividido em duas partes para ficar mais organizado, o algoritmo principal (Algoritmo 1) e sua sub-rotina (Algoritmo 2). Antes de apresentar os algoritmos em si, é necessário introduzir algumas definições que serão usadas nesta seção. Nestas definições, X denota um conjunto de estados e N um conjunto de nodos na forma $n = (x, \phi(x), Q_n)$, com Q_n sendo uma tupla com m conjuntos Booleanos Q_n^k , $1 \leq k \leq m$.

Definição 1 $P : N \rightarrow X$ é uma função de projeção de N para X definida como:

$$P(n) = x.$$

A função P projeta um nodo $n \in N$ em seu estado associado $x \in X$. Além disso, P pode ser naturalmente estendida para agir sobre conjuntos de nodos, da seguinte maneira:

$$P(\{n_0, n_1, \dots, n_k\}) \doteq \bigcup_{i=0}^k P(n_i).$$

Ou seja, se a projeção P for aplicada a um conjunto de nodos, ela retorna o conjunto de todos os estados associados a esses nodos.

Definição 2 Sejam $x \in X$ e $L \subseteq J \subseteq N$, sendo N o conjunto de nodos da árvore $T = (N, E)$. Então:

$$\begin{aligned} & \text{Entradas_Controle}(x, J, L) \\ & \doteq \{\phi \in A(x) \mid H(x, \phi(x)) \subseteq L \wedge J \cap H(x, \phi(x)) \neq \emptyset\}. \end{aligned}$$

A função $\text{Entradas_Controle}(x, J, L)$ retorna todas as entradas de controle admissíveis $\phi(x) \in A(x)$ para um estado $x \in X$ tal que $\phi(x)$:

- i) conduz o sistema de x para ao menos um estado de J ;
- ii) conduz o sistema de x para somente estados do conjunto L .

Definição 3 Seja $V \subseteq X$ e $\text{Ham}(x, y)$ a distância Hamming entre duas cadeias binárias x e y , i.e. quantos *bits* variam entre x e y . Então:

$$\text{Estados_Adjacentes}(V) \doteq \{y \in X \mid \text{Ham}(y, x) = 1 \text{ para algum } x \in V\}.$$

Definição 4 Seja $Z \subseteq N$. Então:

$$\begin{aligned} & \text{Conjuntos_Nodos_Imagem}(x, \phi(x), Z) \doteq \{\Omega \in 2^Z \mid P(\Omega) = \\ & H(x, \phi(x)) \wedge \forall \omega_i, \omega_j \in \Omega, \omega_i \neq \omega_j \rightarrow P(\omega_i) \neq P(\omega_j)\}. \end{aligned}$$

A função $\text{Conjuntos_Nodos_Imagem}$ retorna todos os subconjuntos de Z que possuem como projeção P o conjunto $H(x, \phi(x))$ e que, além disso, quaisquer dois elementos do conjunto não podem possuir a mesma projeção y .

Definição 5 Seja A um conjunto de funções Booleanas $g : \{0, 1\}^n \rightarrow \{0, 1\}$. Então:

$$\text{Herança_Restrições}(x, \phi(x), k, A) \doteq A \cap B_k,$$

em que B_k é o conjunto de todas as funções Booleanas b que realizam $\phi(x)$ para a entrada de controle u_k .

A função Herança_Restrições retorna as funções Booleanas permitidas para a implementação de uma dada entrada de controle $\phi(x)$ associada a um conjunto de nodos Z , i.e. $P(Z) = H(x, \phi(x))$. Para cada gene sintético m , a herança de restrições consiste em calcular a intersecção entre todos os Q_m dos nodos de Z e filtrar somente as funções Booleanas dessa intersecção resultante que podem implementar v_i^s para $\phi(x)$.

Definição 6 Seja $T = (N, E)$ uma árvore com N sendo seu conjunto de nodos e $E \subseteq N \times N$ seus arcos. Usamos $e(1)$ e $e(2)$ para denotar o primeiro e segundo elemento, respectivamente, do par $e \in E$. Seja também $c : N \times N \rightarrow \{0, 1\}$ uma função tal que $c(n, m) = 1$ se e somente se existir uma sequência $e_0 \dots e_k$ de arcos em E tal que: a) $e_0(1) = n$; b) $e_k(2) = m$; e c) $e_{i-1}(2) = e_i(1)$, $1 \leq i \leq k$. A *parte acessível* da árvore T partindo de um nodo $p \in N$ é definida como:

$$Ac(T, p) = (N' \subseteq N, E' \subseteq E), \text{ com}$$

$$N' = \{p\} \cup \{m \in N \mid c(p, m) = 1\} \text{ e}$$

$$E' = \{(n, m) \in E \mid n, m \in N'\}.$$

A função $Ac(T, p)$ retorna a parte acessível de uma árvore, de um nodo p até o nodo final, i.e., ela “apara” a árvore. Como um exemplo, a Figura 17 consiste em $Ac(T, n_8)$, sendo T a árvore da Figura 16. Finalmente, nós usaremos **Extraia_S(T)** para nos referirmos a função que extrai todas as ações de controle $\phi(x)$ de uma árvore T .

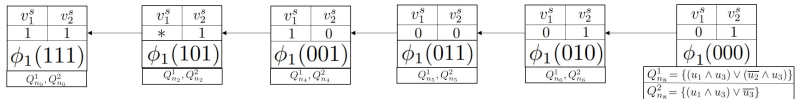


Figura 17 – Parte acessível, a partir de n_8 , da árvore de nodos da Figura 16.

Como considerações finais antes da apresentação dos algoritmos, o Algoritmo 1 é constituído da função Computa_Árvore, a qual

computa iterativamente a árvore explicada no exemplo da Seção 4.2 (veja Figura 16). Computa_Árvore retorna o supervisor $S(x)$ e a tupla $Q = (Q^1, \dots, Q^m)$, que contém os conjuntos com possíveis realizações para u_1, \dots, u_m , correspondente a S . O Algoritmo 2 é constituído pela função Atualiza_Árvore, a qual é invocada pelo Algoritmo 1 e, para um dado par $(x, \phi(x))$ na iteração k , ela verifica se a realização de $\phi(x)$ é consistente com os nodos já computados para os conjuntos C_0, \dots, C_{k-1} . No caso de resposta afirmativa, Atualiza_Árvore cria o nodo $(x, \phi(x), Q)$, adicionando ele à árvore solução.

Antes de encerrar este capítulo, é interessante explicar com mais clareza como acontecem os testes para a adição de um novo nodo em C_n . Para este fim, considere o exemplo hipotético da Figura 18. Suponha que o estado x , o qual é a projeção de um nodo qualquer em C_{n-1} , está sendo testado na iteração atual do Algoritmo 1. O primeiro passo é calcular todos os seus estados adjacentes, que na Figura 18 são x_a , x_b e x_c . Então, para cada um destes estados adjacentes a x , procede-se a verificar quais possuem entradas de controle que satisfazem duas condições: a) deve levar para ao menos um estado de $P(C_{n-1})$, e b) deve levar somente a estados de $P(UC)$. No exemplo da Figura 18, somente os estados x_b e x_c possuem entradas de controle que satisfazem essas condições.

Como podem existir dois ou mais nodos em um mesmo C_k que possuem a mesma projeção, o próximo passo é verificar quantos conjuntos diferentes de nodos que satisfazem essas condições existem, para cada entrada de controle. Neste passo não há eliminação de candidatos, pois cada entrada de controle deve possuir ao menos um conjunto de nodos K associado. Por fim vem o teste da herança de restrições, no qual serão verificadas se as funções de atualização são compatíveis com as dos nodos subsequentes. No exemplo da Figura 18, dos cinco potenciais nodos, somente dois possuem conjunto Q diferente de vazio, ou seja, somente os nodos n_b e n_c serão adicionados a C_n por parte do estado $x \in P(C_{n-1})$.

4.5 PROPRIEDADES DO ALGORITMO PROPOSTO

Nesta seção demonstraremos que toda solução encontrada pelo Algoritmo 1 é de fato uma solução para o Problema 1, e que se o Algoritmo 1 não encontra nenhuma solução, então não existe solução para o Problema 1. Ou seja, só existe uma solução para o Problema 1, se e somente se o Algoritmo 1 encontra uma solução.

Algorithm 1 Pseudo código para a rotina *Computa_Árvore*, que resolve o problema de atração de estados sob restrições biológicas na realização.

```

1: main Computa_Árvore
2: Inicializando e declarando índices e variáveis
3:  $sc_1, sc_2 \leftarrow \text{false}$ ; ▷ Flags para os critérios de parada.
4:  $V \leftarrow \{x_m\}$ ; ▷ Conjunto de estados vizinhos.
5:  $C \leftarrow \emptyset$ ; ▷ Conjunto de nodos construído a cada iteração.
6:  $C' \leftarrow \{x_m\}$ ; ▷ Projeção no espaço de estados de  $C$ .
7:  $j \leftarrow 0$ ; ▷ Índice dos conjuntos  $C_j$ .
8:  $E \leftarrow \emptyset$ ; ▷ Conjunto de arcos direcionados da árvore.
9:  $N \leftarrow \emptyset$ ; ▷ Conjunto de nodos da árvore.
10:  $N' \leftarrow \{x_m\}$ ; ▷ Projeção no espaço de estados de  $N$ .
11: Loop principal
12: while  $(sc_1 = \text{false}) \wedge (sc_2 = \text{false})$  do
13:    $C_j \leftarrow \emptyset$ ;
14:   for all  $(x \in V)$  do
15:      $\Phi \leftarrow \text{Entradas\_Controle}(x, C', N')$ ;
16:     for all  $(\phi(x) \in \Phi)$  do
17:        $(C, E) \leftarrow \text{Atualiza\_Árvore}(x, \phi(x), N, E)$ ;
18:        $C_j \leftarrow C_j \cup C$ ;
19:     end for
20:   end for
21: Teste dos critérios de parada
22:   if  $(x_0 \in P(C_j))$  then
23:      $sc_1 \leftarrow \text{true}$ ; ▷ Solução encontrada.
24:   else if  $(C_j = \emptyset)$  then
25:      $sc_2 \leftarrow \text{true}$ ; ▷ Não existe solução.
26:   else
27: Critérios de parada não satisfeitos
28:      $C' \leftarrow P(C_j)$ ;
29:      $N \leftarrow N \cup C_j$ ;
30:      $N' \leftarrow P(N)$ ;
31:      $j \leftarrow j + 1$ ;
32:      $V \leftarrow \text{Estados\_Adjacentes}(C')$ ;
33:   end if
34: end while
35: if  $sc_1 = \text{true}$  then
36:   return  $\text{Extraia\_S}(\text{Ac}(T, n))$ ; para algum  $n$  t.q.  $P(n) = x_0$ 
37: else
38:   return “não há solução”;
39: end if
40: end main;

```

Algorithm 2 Pseudo código para a sub-rotina *Atualiza_Árvore*, que atualiza o conjunto de arcos E e computa um conjunto intermediário de nodos C que mais tarde irá compor o conjunto C_j , dado um estado x , uma entrada de controle $\phi(x)$ e um conjunto de nodos N .

```

1: subroutine Atualiza_Árvore( $x, \phi(x), N, E$ )
2:  $C \leftarrow \emptyset$ ;
3:  $W \leftarrow \text{Conjuntos\_Nodos\_Imagem}(x, \phi(x), N)$ 
4: Caso exclusivo para a primeira iteração, ou seja, construção de  $n_0$ .
5: if  $W = \emptyset$  then
6:    $Q_n^i \leftarrow \text{Herança\_Restrições}(x, \phi(x), R), i \in \{1, \dots, m\}$ ;
7:   if ( $Q_n^i \neq \emptyset, \forall i \in \{1, \dots, m\}$ ) then
8:      $C \leftarrow \{(x, \phi(x), Q_n)\}$ ;  $\triangleright Q_n = (Q_1, \dots, Q_m)$ 
9:      $E \leftarrow \emptyset$ ;
10:  end if
11: end if
12: Para todas as outras iterações, exceto a primeira.
13: for all ( $K \in W$ ) do
14:    $Q_n^i \leftarrow \text{Herança\_Restrições}(x, \phi(x), i, \bigcap_{\forall k \in K} Q_k^i), i \in \{1, \dots, m\}$ ;
15:   if ( $Q_n^i \neq \emptyset, \forall i \in \{1, \dots, m\}$ ) then
16:      $n \leftarrow (x, \phi(x), Q_n)$ ;
17:      $C \leftarrow C \cup \{n\}$ ;
18:     for all ( $k \in K$ ) do
19:        $E \leftarrow E \cup \{(n, k)\}$ ;
20:     end for
21:   end if
22: end for
23: return ( $C, E$ );  $\triangleright$  Retorna os conjuntos  $C$  e  $E$  atualizados.
24: end subroutine

```

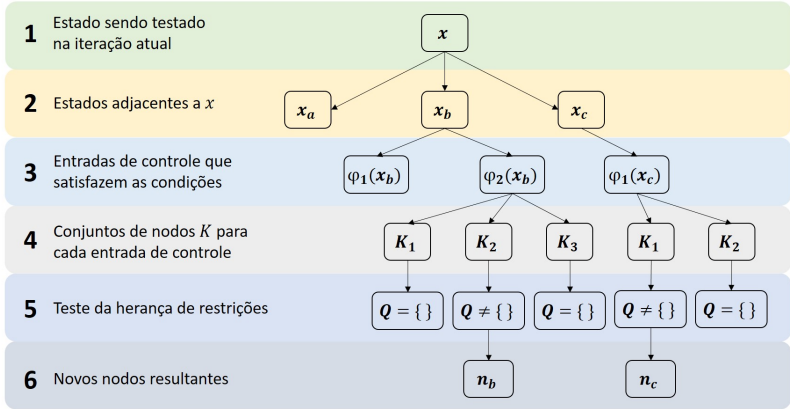


Figura 18 – Exemplo hipotético dos testes realizados pelo Algoritmo 1 para definir novos nodos.

Teorema 1. Existe uma solução para o problema de atração de estados sob restrições biológicas (Problema 1, declarado na Seção 4.3) se, e somente se, o procedimento descrito no Algoritmo 1 (Seção 4.4) encontra uma solução.

Prova:

Para provar a afirmação de que se o Algoritmo 1 acha uma solução, ela realmente é uma solução, nós vamos mostrar que o supervisor retornado pelo Algoritmo 1 induz uma sequência de conjuntos de estados $D_0 D_1 \dots D_k$, com $D_i \in 2^X$, de tal forma que:

1. Para todo $x \in D_i$, $0 \leq i \leq k - 1$, $H(x, S(x)) \subseteq D_{i+1}$;
2. $x_{in} \in D_0$, sendo x_{in} o estado inicial;
3. $x_f \in D_k$ e $H(x_f, S(x_f)) = \{x_f\}$, sendo x_f o estado final;
4. As funções Booleanas h_i 's usadas para implementar S pertencem ao conjunto R .

Seja $\{C_i\}_{i=0}^k$ a sequência de conjuntos de nodos obtida iterativamente pelo Algoritmo 1. Então, definimos para cada C_i um conjunto de estados $D_{k-i} = P(C_i)$. Agora vamos provar que tal sequência $\{D_j\}_{j=0}^k$ satisfaz as condições (1), (2) e (3).

Por construção $D_0 = P(C_k)$ contém x_{in} (critério de parada sc_1 , linha 23 do Algoritmo 1). Portanto, a condição (2) é satisfeita. Além disso, as linhas 17 e 18 do Algoritmo 1 mostram que $(C_0, \emptyset) = \text{Atualiza_Árvore}(x_f, \phi(x_f), \emptyset, \emptyset)$, o que, de acordo com o Algoritmo 2, resulta em um C_0 que satisfaz $P(C_0) = D_k = \{x_f\}$. Ainda mais, a sub-rotina Atualiza_Árvore somente é invocada no caso de $\Phi \neq \emptyset$, o que assegura que $H(x_f, \phi(x_f)) = \{x_f\}$. Portanto, a condição (3) é satisfeita.

Como pode ser visto nas linhas 14-20 do Algoritmo 1, na j -ésima iteração, por construção, somente os estados x que satisfazem $H(x, \phi(x)) \subseteq P(C_{j-1})$ para algum ϕ são estados candidatos para serem parte dos nodos em C_j (ver Definição 2 para a função Entradas_Controle), i.e., se tal ϕ não existe, o ciclo das linhas 16-19 do Algoritmo 1 não será executado. Portanto, $\forall j \in \{1, \dots, k\}$, todos os estados $x \in P(C_j) = D_{k-j}$ satisfazem $H(x, S(x)) \subseteq P(C_{j-1}) = D_{k-(j-1)} = D_{(k-j)+1}$. Isto é equivalente a dizer que para todo $x \in D_i$, $0 \leq i \leq k-1$, $S(x)$ é tal que $H(x, S(x)) \subseteq D_{i+1}$. Condição (1) é então satisfeita.

Finalmente, nós procedemos com a prova de que a condição (4) também é verificada por S . Para isso, primeiro lembre que para cada novo nodo n inserido em C_j na iteração j do Algoritmo 1, Q_n^i é necessariamente não vazio para todo $i \in \{1, \dots, m\}$ (linhas 7 e 15 do Algoritmo 2) e que, pela Definição 5 (herança de restrições), todo Q_n^i satisfaz a condição $Q_n^i \subseteq \bigcap_{n_k \in K} Q_{n_k}^i$ (ver linha 14 do Algoritmo 2), em que $K \subseteq \bigcup_{m=0}^{j-1} C_m$. Isto é, para quaisquer dois nodos n, m , com $(n, m) \in E$, $n \in C_j$ e $m \in C_0 \cup \dots \cup C_{j-1}$, $Q_n^i \subseteq Q_m^i$. Como o nodo inicial é n_0 , do qual $Q_{n_0}^i \subseteq R$ (ver linha 6 do Algoritmo 2), segue que qualquer nodo n na árvore solução satisfaz $Q_n^i \subseteq R$ em qualquer iteração. Em particular, o nodo $n_s \in C_k$, para o qual $P(n_s) = x_{in}$. Qualquer função Booleana em $Q_{n_s}^i$ realiza o supervisor S para a entrada de controle u_i . Como $Q_{n_s}^i \subseteq R$, para todo $i \in \{1, \dots, m\}$ a condição (4) também é verificada.

Agora nós iremos provar que o Algoritmo 1 é completo, ou seja, se ele não encontra uma solução para o problema, é porque não existe uma solução para o problema. Para o Algoritmo 1 parar sem uma solução, deve ser atribuído o valor *true* para o *flag* sc_2 na última iteração (ver linha 25 do Algoritmo 1), a qual nos referimos por iteração j . Isso só acontece quando C_j é um conjunto vazio (linha 24 do Algoritmo 1). Se o conjunto C_j é vazio, então uma das condições abaixo é verdadeira (ver linhas 14-20 do Algoritmo 1):

1. $V = \emptyset$;
2. $V \neq \emptyset$ e $\Phi = \emptyset$ (linha 15 do Algoritmo 1);

3. $V, \Phi \neq \emptyset$ e $\forall \phi(x) \in \Phi, C = \emptyset$ (linha 17 do Algoritmo 1).

A Condição (1) nunca é satisfeita, dado que todo estado $x \in X$ possui vizinhos. Então procedemos com a análise das condições (2) e (3).

A Condição (2) só é verdadeira se para todo $x \in V$ nenhuma entrada de controle admissível $\phi(x)$ é capaz de trazer x para C' e, portanto, para x_f . Então, nenhum controle é capaz de impor uma trajetória de x_{in} para x_f e o problema realmente não possui uma solução.

Finalmente, para a condição (3) ser verdadeira, *Atualiza_Árvore* deve retornar um conjunto C vazio para todas as tuplas $(x, \phi(x), N, E)$ na iteração j , em que $x \in V$, $\phi(x) \in \Phi$ e (N, E) é a árvore computada até a iteração $j - 1$. Isto, de acordo com o Algoritmo 2, somente acontece se, para todas as tuplas $(x, \phi(x), N, E)$:

- a) $W = \emptyset$ e $x \neq x_{in}$; ou
- b) $W \neq \emptyset$ e $\forall K \in W$, é verdade que $\exists i \in \{1, \dots, m\}$ tal que $Q_n^i = \emptyset$.

O Caso (a) não pode ocorrer, porque se o procedimento *Atualiza_Árvore* é invocado na linha 17 do Algoritmo 1, então $\Phi \neq \emptyset$ na linha 15 e, portanto, existe algum $x \in V$ e algum $M \subseteq N$ tal que $H(x, \phi(x)) = P(M)$. W então possui pelo menos um elemento K . Resta analisar o Caso (b). Se a sentença em (b) é verdadeira, significa que para todo $x \in V$, a implementação da entrada de controle $\phi(x)$ que faz $H(x, \phi(x)) = P(K)$, $K \in W$, é inconsistente com a implementação de $\phi(y)$ para algum estado y que “está no caminho” de x e x_f . Portanto, nenhuma entrada de controle $\phi(x)$ que leva o sistema “em frente” pode ser implementada sem causar efeitos colaterais prejudiciais nas entradas de controle previamente computadas. Ou seja, o problema então não possui solução.

Aqui encerramos a prova do Teorema 1, ao analisar todos os casos possíveis de resultados do Algoritmo 1. Isso conclui o capítulo e em seguida vamos apresentar as conclusões e sugestões para trabalhos futuros.

5 CONCLUSÕES

Nesta dissertação, nós introduzimos, formalizamos e resolvemos um problema de atração de estados com restrições na realização das entradas de controle. O problema que apresentamos aqui foi inicialmente motivado por ideias previamente desenvolvidas pelos autores que aplicaram a Teoria de Controle Supervisório para intervir no comportamento de redes biológicas intracelulares. Neste contexto, existem restrições biológicas que impactam na realização do supervisor, restrições que podem ser formuladas como limitações de funções Booleanas que implementam entradas de controle.

Tratar das restrições biológicas na realização por genes sintéticos de um supervisor é um passo importante para aplicar na prática, em casos reais, a abordagem de controle celular apresentada em (BALDISSERA; CURY; RAISCH, 2016). Com os rápidos avanços da Biologia Sintética, a tendência é que haja menos restrições biológicas na síntese de moléculas, entretanto é bem improvável que essas restrições sejam extintas por completo.

A estratégia do método desenvolvido é basicamente computar todos os caminhos de estados que são soluções em potencial, de trás para frente (i.e. começando pelo estado final), enquanto são verificados possíveis conflitos entre as funções de atualização que podem ser usada para implementar o supervisor S . A cada iteração, novos nodos são analisados e possivelmente adicionadas à árvore solução. Após atingir um critério de parada, é possível extrair desta árvore, usando a função de acessibilidade, um supervisor que atende as especificações desejadas e que pode ser implementado através de genes sintéticos que computam somente funções Booleanas do conjunto R , o qual reflete as restrições biológicas inerentes da síntese de genes.

Apesar de solucionarmos de maneira correta e completa o problema de atração de estados com restrições na realização, não foi o principal objetivo deste trabalho apresentar um algoritmo eficiente para computar o supervisor para o problema declarado, mas sim formalizar e compartilhar este problema com a comunidade científica. Melhorias no uso de memória e tempo de execução do algoritmo são objetivos de trabalhos futuros. Outros aspectos a serem investigados são: a exploração da estrutura do conjunto de funções Booleanas R que podem ser implementadas (um conjunto fechado sob disjunção por suposição); ou o uso de uma representação simbólica de estados Booleanos com diagramas de decisão binária.

Nós também pretendemos generalizar a formalização dos subsistemas da planta P_i , com o intuito de permitir eles a: a) serem definidos por funções Booleanas $f_i(x, u)$, com $u = [u_1 \cdots u_m]$, ao invés do conjunto mais restrito $f_i(x, u_i)$ como aqui considerado; e b) serem multidimensionais, i.e., relaxando as restrições de que $x_i \in \{0, 1\}$ e permitindo $x_i \in \{0, 1\}^p$, $p \geq 1$. Este relaxamento no domínio de x_i pode também ser usado para tratar do caso em que os estados x_i de P_i assumem valores em domínios finitos e discretos, mas não Booleanos, e.g., codificando cada x_i não Booleano como um vetor Booleano $x_i = [x_i^1 \cdots x_i^p]$ com $p = \lceil \log(\max(x_i)) \rceil$, em que $\lceil \cdot \rceil$ denota a função teto.

Os resultados deste trabalho geraram a submissão de um artigo ao periódico *Discrete Event Dynamic Systems* (LEITE; BALDISSERA; CURY, 2018), cuja avaliação Qualis da Capes na área é A2.

REFERÊNCIAS

ALBERT, R. Boolean modeling of genetic regulatory networks. In: **Complex networks**. [S.l.]: Springer, 2004. p. 459–481.

ALBERT, R.; OTHMER, H. G. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *drosophila melanogaster*. **Journal of theoretical biology**, Elsevier, v. 223, n. 1, p. 1–18, 2003.

ALBERTS, B. et al. **Essential cell biology**. [S.l.]: Garland Science, 2013.

ALBERTS, B. et al. **Biologia molecular da célula**. [S.l.]: Artmed Editora, 2010.

ANDRIANANTOANDRO, E. et al. Synthetic biology: new engineering rules for an emerging discipline. **Molecular systems biology**, EMBO Press, v. 2, n. 1, 2006.

BAGH, S. et al. An active intracellular device to prevent lethal disease outcomes in virus-infected bacterial cells. **Biotechnology and bioengineering**, Wiley Online Library, v. 108, n. 3, p. 645–654, 2011.

BALDISSERA, F. L.; CURY, J. E. Dealing with biological constraints in the synthesis of controllers for gene regulatory networks. **IFAC Proceedings Volumes**, Elsevier, v. 47, n. 3, p. 2435–2441, 2014.

BALDISSERA, F. L.; CURY, J. E.; RAISCH, J. A supervisory control theory approach to control gene regulatory networks. **IEEE Transactions on Automatic Control**, IEEE, v. 61, n. 1, p. 18–33, 2016.

BENENSON, Y. Biomolecular computing systems: principles, progress and potential. **Nature Reviews Genetics**, Nature Publishing Group, v. 13, n. 7, p. 455–468, 2012.

BERNOT, G. et al. Application of formal methods to biological regulatory networks: extending thomas asynchronous logical approach with temporal logic. **Journal of theoretical biology**, Elsevier, v. 229, n. 3, p. 339–347, 2004.

BLAZECK, J. et al. Controlling promoter strength and regulation in *saccharomyces cerevisiae* using synthetic hybrid promoters.

Biotechnology and bioengineering, Wiley Online Library, v. 109, n. 11, p. 2884–2895, 2012.

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

CURY, J. E.; BALDISSERA, F. L. Systems biology, synthetic biology and control theory: a promising golden braid. **Annual Reviews in Control**, Elsevier, v. 37, n. 1, p. 57–67, 2013.

DATTA, A. et al. Control approaches for probabilistic gene regulatory networks-what approaches have been developed for addressing the issue of intervention? **IEEE Signal Processing Magazine**, IEEE, v. 24, n. 1, p. 54–63, 2007.

GINGOLD, H.; PILPEL, Y. Determinants of translation efficiency and accuracy. **Molecular systems biology**, EMBO Press, v. 7, n. 1, p. 481, 2011.

HORGAN, R. P.; KENNY, L. C. Omic technologies: genomics, transcriptomics, proteomics and metabolomics. **The Obstetrician & Gynaecologist**, Wiley Online Library, v. 13, n. 3, p. 189–195, 2011.

KARLEBACH, G.; SHAMIR, R. Modelling and analysis of gene regulatory networks. **Nature Reviews Molecular Cell Biology**, Nature Publishing Group, v. 9, n. 10, p. 770–780, 2008.

KARLSSON, M.; WEBER, W. Therapeutic synthetic gene networks. **Current opinion in biotechnology**, Elsevier, v. 23, n. 5, p. 703–711, 2012.

KAUFFMAN, S. A.; STROHMAN, R. C. **The Origins of Order: self organization and selection in evolution**. [S.l.]: Oxford university press New York, 1994.

LEITE, P. A.; BALDISSERA, F. L.; CURY, J. E. State-based supervisory control with restrictions on the supervisor realization. **Discrete Event Dynamic Systems**, Springer US, v. 29, n. 1, 2018.

MACHADO, D. et al. Modeling formalisms in systems biology. **AMB express**, SpringerOpen, v. 1, n. 1, p. 45, 2011.

- MENOLASCINA, F.; BERNARDO, M. D.; BERNARDO, D. D. Analysis, design and implementation of a novel scheme for in-vivo control of synthetic gene regulatory networks. **Automatica**, Elsevier, v. 47, n. 6, p. 1265–1270, 2011.
- MILIAS-ARGEITIS, A. et al. In silico feedback for in vivo regulation of a gene expression circuit. **Nature biotechnology**, Nature Research, v. 29, n. 12, p. 1114–1116, 2011.
- PURNICK, P. E.; WEISS, R. The second wave of synthetic biology: from modules to systems. **Nature reviews Molecular cell biology**, Nature Publishing Group, v. 10, n. 6, p. 410–422, 2009.
- RAMADGE, P. J.; WONHAM, W. M. Supervisory control of a class of discrete event processes. **SIAM journal on control and optimization**, SIAM, v. 25, n. 1, p. 206–230, 1987.
- SAMAL, A.; JAIN, S. The regulatory network of e. coli metabolism as a boolean dynamical system exhibits both homeostasis and flexibility of response. **BMC systems biology**, BioMed Central, v. 2, n. 1, p. 21, 2008.
- WONHAM, W. M. **Supervisory control of discrete-event systems**. [S.l.]: Springer, 2015.