



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS UNIVERSITÁRIO REITOR JOÃO DAVID FERREIRA LIMA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE AUTOMAÇÃO E
SISTEMAS

Jhonatan Alves

Síntese Automática de Normas Eficientes Para Sistemas Multiagente

Florianópolis-SC
2023

Jhonatan Alves

Síntese Automática de Normas Eficientes Para Sistemas Multiagente

Tese submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina para a obtenção do título de Doutor em Engenharia de Automação e Sistemas.

Orientador: Prof. Jomi Fred Hübner, Dr.

Coorientadora: Profa. Jerusa Marchi, Dra.

Florianópolis-SC

2023

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Alves, Jhonatan

Síntese Automática de Normas Eficientes Para Sistemas Multiagente / Jhonatan Alves ; orientador, Jomi Fred Hübner, coorientadora, Jerusa Marchi, 2023.

171 p.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Florianópolis, 2023.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2. Sistemas Multiagente Normativos. 3. Síntese de Normas Eficientes. 4. Alcançabilidade dos Objetivos do Sistema. 5. Ocorrência de Conflitos. I. Hübner, Jomi Fred. II. Marchi, Jerusa. III. Universidade Federal de Santa Catarina. Programa de Pós Graduação em Engenharia de Automação e Sistemas. IV. Título.

Jhonatan Alves

Síntese Automática de Normas Eficientes Para Sistemas Multiagente

O presente trabalho em nível de doutorado foi avaliado e aprovado, em 27 de novembro de 2023, pela banca examinadora composta pelos seguintes membros:

Prof. Jaime Simão Sichman, Dr.
USP

Prof. Ricardo Azambuja Silveira, Dr.
UFSC

Profa. Viviane Torres da Silva, Dra.
IBM Research

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

Coordenação do Programa de
Pós-Graduação

Prof. Jomi Fred Hübner, Dr.
Orientador

Profa. Jerusa Marchi, Dra.
Coorientadora

Florianópolis-SC, 2023.

Aos meus queridos pais, Leomar e Marlei, e à minha irmã, Jhenifer.

AGRADECIMENTOS

Quando eu era criança, eu estudava com minha mãe todos os dias antes de ir para a escola. Revisávamos coisas simples como a tabuada e a gramática. Quando meu pai chegava à noite em casa do trabalho, íamos brincar e, nessas brincadeiras, jogávamos coisas sobre o que eu tinha aprendido na escola. Me lembro de um dia, já adolescente, meu pai me dizer que a única herança que ele poderia deixar para mim e a minha irmã e que jamais alguém nos tomaria seria os estudos. Olhando para o passado, eu vejo que desde criança os meus pais me ensinaram a gostar de estudar. Talvez, não pelo acaso do destino, mas, principalmente, pela influência deles, eu segui carreira acadêmica me tornando professor universitário e, agora, Doutor em Engenharia.

Meus pais, Leomar Argentil Alves e Marlei Angelita Borges Alves, vieram de famílias humildes e desde crianças tiveram que trabalhar para ajudar no sustento da casa. Infelizmente, não tiveram boas oportunidades e incentivos para os estudos. Porém, tendo ciência de que o ensino é chave para a formação e transformação do cidadão, eles buscaram proporcionar a mim e minha irmã estudo de qualidade. Aliado a isso, o amor, o carinho e a dedicação deles me trouxeram até aqui. Pai e mãe, obrigado por confiarem em mim, por apostarem na minha carreira, por torcerem por mim e por não medirem esforços na minha educação. Sou grato por tudo. Eu os amo. Muito!

Agradeço a minha irmã, Jhenifer Alves, pelo seu constante apoio nessa trajetória. Durante os anos de doutorado, você se preocupou comigo, com a minha saúde e felicidade. Foram várias as conversas onde discutimos sobre as dificuldades e obstáculos do andamento da tese e, no que pôde, você me ajudou e me confortou para que eu pudesse seguir em frente e terminar o meu trabalho com sucesso. Obrigado, minha irmã querida, pois você foi fundamental nesse processo.

Agradeço aos meu orientadores, professor Jomi Fred Hübner e professora Jersusa Marchi, por me conduzirem na minha caminhada com maestria. Obrigado pelo auxílio no correto entendimento do tema de pesquisa, por me guiarem nos melhores caminhos e me ajudarem a tomar decisões assertivas. Obrigado pela disponibilidade, apoio na revisão de artigos, apresentação de trabalhos e escrita da tese. Também gostaria de agradecer-los pela compreensão. Durante estes anos de doutorado, alguns percalços ocorreram, ocasionando atrasos no andamento da tese. Ambos os professores me incentivaram a continuar e estenderam a mão para me ajudar, dando-me a oportunidade de finalizá-la. Eu sou imensamente grato por terem acreditado em mim e na minha pesquisa. Vocês foram fundamentais para que eu chegasse aqui. Muito obrigado por tudo! Eu os admiro muito!

Gostaria de agradecer às minhas duas grandes amigas, Emanuelle Castro Bitencourt e Scheila Valéria Colaço, por todo o apoio ao longo destes anos. A amizade de vocês foi fundamental para eu crescer como pessoa e como profissional. Sinto

que me transformei nestes últimos anos e vocês são parte dessas mudanças. Embora estejamos longe, a nossa amizade é forte e genuína. O carinho de vocês está refletido nessa conquista para a minha vida e profissão. Vocês são parte disso tudo e eu as amo.

Por fim, gostaria de agradecer aos professores membros da banca avaliadora, Prof. Jaime Simão Sichman, Prof. Ricardo Azambuja Silveira e Profa. Viviane Torres da Silva, por aceitarem o convite em avaliar o meu trabalho, pelo apoio e sugestões que contribuíram para que um trabalho de qualidade fosse produzido.

*“Mas os que esperam no Senhor renovarão as forças,
subirão com asas como águias; correrão, e não se
cansarão; caminharão, e não se fatigarão.”
(Isaías 40:31)*

RESUMO

A síntese de normas eficientes, um problema em aberto e de solução não trivial, consiste em determinar normas que evitam situações de conflitos ao passo que mantêm os objetivos do sistema alcançáveis. Contudo, sintetizar normas eficientes traz à tona uma série de desafios como identificar corretamente as normas que mantêm um equilíbrio entre o controle que exercem e a autonomia dos agentes assim como lidar com um tempo potencialmente alto para que a síntese seja bem-sucedida. Para situar esta problemática, considere dois tipos de normas: normas muito específicas e muito gerais. Enquanto normas muito específicas tendem a ser incapazes de evitar conflitos, dado que dão uma grande liberdade aos agentes para executarem as suas ações, normas muito gerais tendem a inviabilizar o alcance dos objetivos do sistema, dado que regulam os agentes em excesso, o que pode impedi-los de executar ações necessárias para que os objetivos do sistema sejam alcançados. Neste sentido, para serem eficientes, as normas necessitam ter uma generalidade balanceada não sendo, portanto, nem muito específicas nem muito gerais. Além disso, como o número de normas candidatas para regular os agentes pode ser grande, determinar quais normas possuem a generalidade balanceada pode ser uma tarefa custosa e que demanda tempo.

Para lidar com estes problemas, nós propomos: i) uma conceitualização formal para a síntese automática de normas eficientes; ii) uma integração entre a síntese de normas e a análise da alcançabilidade dos objetivos do sistema; iii) uma forma de considerar os conflitos do sistema na síntese de normas; iv) estratégias para estimar a generalidade das normas; v) estratégias para ordenar as normas; vi) estratégias para explorar espaços de busca; e vii) estratégias para reduzir a complexidade da síntese de normas eficientes. Isso nos permite organizar as normas, aproximadamente, das mais específicas para as mais gerais e obter espaços de busca ordenados. Sobre tais espaços, aplicamos estratégias de exploração que se beneficiam da estrutura das ordenações para encontrar normas eficientes. Assumindo que tais normas não estão localizadas nos extremos das ordenações, podemos podar parte do espaço de busca e economizar tempo e esforço computacional na síntese de normas eficientes.

Nós também propomos um conjunto de algoritmos para a síntese de normas eficientes que são avaliados em cenários distintos. Para cada cenário, os algoritmos organizam e exploram os espaços de busca de diferentes formas. Como principais resultados dos experimentos, nós verificamos que: i) certas estratégias para organizar e explorar o espaço de busca são mais apropriadas para a síntese de normas eficientes ao aprimorarem o desempenho da síntese; ii) a análise da alcançabilidade dos objetivos do sistema auxilia a construir espaços de busca com normas melhor alinhadas a comportamentos que necessitam de regulação e que permitem aos agentes executarem, pelo menos, uma sequência de ações que os levem aos objetivos do sistema; e iii) a análise da alcançabilidade dos objetivos do sistema auxilia a construir espaços de busca menores fazendo com que o tempo de busca por normas eficientes diminua.

Palavras-chave: Sistemas Multiagente Normativos; Síntese de Normas Eficientes; Alcançabilidade dos Objetivos do Sistema; Ocorrência de Conflitos; Generalidade das Normas.

ABSTRACT

The synthesis of efficient norms, an open problem with a non-trivial solution, consists of determining norms that avoid conflict situations while maintaining the system goals reachable. However, synthesizing efficient norms brings up several challenges, such as correctly identifying norms that maintain a balance between the control they exert and the autonomy of agents, as well as dealing with a potentially high time to the synthesis be successful. To contextualize this problem, consider two types of norms: very specific and very general norms. While very specific norms tend to be incapable of avoiding conflicts, as they grant agents significant freedom to execute their actions, very general norms may impede the system goals to be reached, as they overregulate agents, which can prevent them from performing necessary actions for satisfying such goals. In this sense, to be efficient, norms need to have a balanced generality, neither being too specific nor too general. In addition, as the number of candidate norms for regulating agents can be large, determining which norms have a balanced generality can be a costly and time-consuming task.

To address these problems, in this thesis we propose: i) a formal conceptualization for the automatic efficient norm synthesis; ii) an integration between norm synthesis and system goal reachability analysis; iii) a method to consider system conflicts in norm synthesis; iv) strategies for estimating the norms generality; v) strategies for ordering norms; vi) strategies for exploring the search space; and vii) strategies for reducing the complexity of efficient norm synthesis. Through this approach, we can organize the norms approximately from the most specific to the most general and obtain ordered search spaces. In these spaces, we can apply exploration strategies that benefit from the structure of the orderings to find efficient norms. Assuming that such norms are not located at the extremes of the orderings, we can prune part of the search space, saving time and computational effort in the synthesis of efficient norms.

We also propose a set of algorithms for synthesizing efficient norms, which are evaluated in various scenarios. For each scenario, the algorithms organize and explore the search spaces in different ways. The main results of the experiments indicate that: i) certain strategies for organizing and exploring the search space enhance the synthesis performance of efficient norms; ii) the analysis of the system goals reachability aids in constructing search spaces with norms aligned to behaviors that need regulation and enable agents to execute, at least, a sequence of actions leading to the reachability of system goals; and iii) the analysis of the system goals reachability helps to build smaller search spaces, which may reduce the search time for efficient norms.

Keywords: Normative Multi-Agent Systems; Efficient Norm Synthesis; System Goals Reachability; Conflict Occurrence; Norms Generality.

LISTA DE FIGURAS

Figura 1 – Grid 3×3 do cenário do tráfego de carros.	19
Figura 2 – Classificação das normas em SMAn.	28
Figura 3 – Classificação da síntese de normas em SMAn.	32
Figura 4 – Situação inicial para o cenário do tráfego de carros simplificado. . .	57
Figura 5 – Árvore com três histórias do cenário do tráfego de carros simplificado.	57
Figura 6 – Árvore com dez histórias do cenário do tráfego de carros simplificado.	60
Figura 7 – Diagrama de Venn das histórias de n_i, n_j, n_k, n_l, n_m e n_p	62
Figura 8 – Diagrama de Venn para as histórias das normas do exemplo 17. . .	63
Figura 9 – Uma árvore de histórias hipotéticas.	65
Figura 10 – Síntese com backtracking sobre a história h_1	66
Figura 11 – Síntese ingênua sobre a história h_3	69
Figura 12 – Síntese com backtracking sobre a história h_3	69
Figura 13 – Síntese ingênua sobre a história h_4	70
Figura 14 – Síntese com backtracking sobre a história h_4	70
Figura 15 – Árvore do exemplo 17 com áreas de poda.	72
Figura 16 – Exemplo de duas sequências de conjuntos de generalidade.	78
Figura 17 – Exemplo de uma sequência semântica.	80
Figura 18 – Sequência aproximada O_1 obtida pela função NGI.	81
Figura 19 – Sequência aproximada O_2 obtida pela função NGJ.	82
Figura 20 – Sequência aproximada O_3 obtida pela função NGK.	83
Figura 21 – Sequência aproximada O_4 obtida pela função NGF.	84
Figura 22 – Grid 3×3 do cenário do tráfego de carros com túneis.	88
Figura 23 – Resultados do cenário do tráfego de carros com túnel.	89
Figura 24 – Resultados do cenário da segurança de acesso.	91
Figura 25 – Resultados do cenário do atendimento médico.	94
Figura 26 – Tendências no crescimento e decrescimento das taxas de sucesso, conflito e timeout.	96
Figura 27 – Exemplo de aplicação do algoritmo 1.	102
Figura 28 – Exemplo de aplicação dos algoritmos 3 e 4.	109
Figura 29 – Exemplos de aplicação das estratégias de visita.	111
Figura 30 – Sequências aproximadas com normas mínimas.	115
Figura 31 – Síntese com backtracking realizada pelo algoritmo 10 sobre a história h_4	123
Figura 32 – Síntese com backtracking realizada pelo algoritmo 10 sobre a história h_8	124

LISTA DE TABELAS

Tabela 1 – Comparativo entre os trabalhos correlatos.	44
Tabela 2 – Comparativo entre as funções de generalidade.	85
Tabela 3 – Comparativo dos resultados dos experimentos.	97
Tabela 4 – Prós, Contras e Aprimoramentos dos Algoritmos de Síntese de Normas Eficientes.	128
Tabela 5 – Tamanho do espaço das normas.	130
Tabela 6 – Percentual de redução do tamanho do espaço das normas.	130
Tabela 7 – Médias de normas analisadas pelo algoritmo 8 no cenário do tráfego de carros.	131
Tabela 8 – Médias de normas analisadas pelo algoritmo 10 no cenário do tráfego de carros.	132
Tabela 9 – Médias de normas analisadas pelo algoritmo 11 no cenário do tráfego de carros.	132
Tabela 10 – Médias do tempo de busca do algoritmo 8 no cenário do tráfego de carros.	132
Tabela 11 – Médias do tempo de busca do algoritmo 10 no cenário do tráfego de carros.	133
Tabela 12 – Médias do tempo de busca do algoritmo 11 no cenário do tráfego de carros.	133
Tabela 13 – Médias de normas analisadas pelo algoritmo 8 no cenário da segurança de acesso.	134
Tabela 14 – Médias de normas analisadas pelo algoritmo 10 no cenário da segurança de acesso.	134
Tabela 15 – Médias de normas analisadas pelo algoritmo 11 no cenário da segurança de acesso.	134
Tabela 16 – Médias do tempo de busca do algoritmo 8 no cenário da segurança de acesso.	135
Tabela 17 – Médias do tempo de busca do algoritmo 10 no cenário da segurança de acesso.	135
Tabela 18 – Médias do tempo de busca do algoritmo 11 no cenário da segurança de acesso.	135
Tabela 19 – Médias de normas analisadas pelo algoritmo 8 no cenário do atendimento médico.	137
Tabela 20 – Médias de normas analisadas pelo algoritmo 10 no cenário do atendimento médico.	137
Tabela 21 – Médias de normas analisadas pelo algoritmo 11 no cenário do atendimento médico.	137

Tabela 22 – Médias do tempo de busca do algoritmo 8 no cenário do atendimento médico.	138
Tabela 23 – Médias do tempo de busca do algoritmo 10 no cenário do atendimento médico.	138
Tabela 24 – Médias do tempo de busca do algoritmo 11 no cenário do atendimento médico.	138
Tabela 25 – Configurações igualmente boas para o algoritmo 8.	140
Tabela 26 – Configurações igualmente boas para o algoritmo 10.	140
Tabela 27 – Configurações igualmente boas para o algoritmo 11.	141

LISTA DE ABREVIATURAS E SIGLAS

AAOS	Análise da Alcançabilidade dos Objetivos do Sistema
ATL	Alternating-time Temporal Logic
IRON	Intelligent Robust On-line Norm Synthesis Machine
JaCaMo	Jason, Cartago, and Moise
JADE	Java Agent DEvelopment Framework
LION	LIberal On-line Norm synthesis
LTL	Linear Temporal Logic
SIMON	SImple Minimal On-line Norm Synthesis
SMA	Sistema Multiagente
SMA _n	Sistema Multiagente Normativo

LISTA DE SÍMBOLOS

\mathcal{L}	Linguagem lógica de 1ª ordem
Pred	Conjunto de símbolos de predicados de \mathcal{L}
Var	Conjunto de símbolos de variáveis de \mathcal{L}
Const	Conjunto de símbolos de constantes de \mathcal{L}
Conec	Conjunto de conectivos lógicos de \mathcal{L}
\mathcal{L}^{uaf}	Fórmulas atômicas estritamente unground de \mathcal{L}
\mathcal{L}'	Fórmulas lógicas bem formadas de \mathcal{L}
\mathcal{L}^{gaf}	Fórmulas atômicas ground de \mathcal{L}
s	Estado do sistema
S	Conjunto dos estados do sistema
S_0	Conjunto de todos os estados iniciais do sistema
a	Ação
Nome	Identificador de uma ação
Par	Parâmetros de uma ação
Pre	Pré-condições de execução de uma ação
Add	Efeitos verdadeiros de uma ação
Del	Efeitos falsos de uma ação
\mathcal{A}	Conjunto de ações
$\mathcal{I}^{\mathcal{A}}$	Conjunto das instâncias das ações de \mathcal{A}
\mathcal{G}	Objetivos do sistema
\mathcal{G}^*	Conjunto de todos os objetivos do sistema
$\mathcal{I}^{\mathcal{G}}$	Conjunto das instâncias dos objetivos \mathcal{G}
$S^{\mathcal{G}}$	Conjunto dos estados de objetivos que satisfazem \mathcal{G}
c	Conflito do sistema
\mathcal{C}	Conjunto dos conflitos do sistema
S^c	Conjunto dos estados de conflito que satisfazem c
n	Norma proibitiva
φ	Contexto de uma norma proibitiva
\mathcal{I}^n	Conjunto das instâncias de uma norma n
$n \triangleright s$	Norma n é aplicável no estado s
\mathcal{N}	Espaço das normas
\mathcal{N}^{equiv}	Conjunto de normas semanticamente equivalentes
Δ	Timeout do sistema
\mathbb{N}^*	Naturais não nulos
t	Traço
\mathcal{T}	Conjunto de traços
h	História

\mathcal{H}	Conjunto de histórias do sistema
Ag	Conjunto de agentes
\mathcal{H}^{suc}	Conjunto de histórias de sucesso
\mathcal{H}^{insuc}	Conjunto de histórias de insucesso
\mathcal{H}^{conf}	Conjunto de histórias de conflito
\mathcal{H}^{time}	Conjunto de histórias de timeout
\mathcal{H}_n	Histórias da norma n
\mathcal{H}_n^{suc}	Histórias de sucesso da norma n
\mathcal{H}_n^{insuc}	Histórias de insucesso da norma n
\mathcal{H}_n^{conf}	Histórias de conflito da norma n
\mathcal{H}_n^{time}	Histórias de timeout da norma n
\mathcal{N}^{efic}	Conjunto de normas eficientes
t_n^{suc}	Taxa de sucesso da norma n
t_n^{conf}	Taxa de conflito da norma n
ω	Função de análise de alcançabilidade
Sa^n	Conjunto de estados nos quais a norma n é aplicável
\mathcal{R}	Ordem parcial irreflexiva, assimétrica e transitiva
\mathbb{R}_+	Reais não negativos
f	Função de generalidade
$formulas$	Função formulas
\mathbb{Z}_+	Inteiros não negativos
$variaveis$	Função variaveis
G_k	Conjunto de generalidade
G^G	Conjunto de conjuntos de generalidade
O	Sequência crescente de conjuntos de generalidade
G^{min}	Conjunto de generalidade do limite inferior de uma sequência O
G^{cen}	Conjunto de generalidade do meio de uma sequência O
G^{max}	Conjunto de generalidade do limite superior de uma sequência O
$G_i * G_j$	Conjuntos de generalidade contíguos
$n_i * n_j$	Normas contíguas
\mathbb{R}_+^*	Reais positivos

SUMÁRIO

1	INTRODUÇÃO	18
1.1	FORMULAÇÃO DO PROBLEMA	18
1.2	PERGUNTAS DE PESQUISA E HIPÓTESES	21
1.3	OBJETIVOS	22
1.4	ESCOPO DO TRABALHO	23
1.5	ORGANIZAÇÃO DA TESE	25
2	FUNDAMENTAÇÃO TEÓRICA E ESTADO DA ARTE	26
2.1	SISTEMAS MULTIAGENTE NORMATIVOS	27
2.1.1	Classificação das Normas	27
2.1.1.1	Normas Regulativas	28
2.1.1.2	Normas Constitutivas	29
2.1.1.3	Normas Substantivas e Processuais	29
2.1.1.4	Convenções	29
2.2	SÍNTESE DE NORMAS EM SMAN	30
2.2.1	Classificação da Síntese de Normas	30
2.2.1.1	O Escopo da Síntese	31
2.2.1.2	A Generalidade das Normas	32
2.2.1.3	A Ocorrência de conflitos	33
2.2.1.4	Os Objetivos do Sistema	33
2.2.1.5	A Ordenação do Espaço das Normas	34
2.2.1.6	As Propriedade das Normas Sintetizadas	34
2.3	TRABALHOS CORRELATOS	35
2.3.1	Síntese Off-line	36
2.3.2	Síntese On-line	39
2.3.2.1	Emergência	39
2.3.2.2	Observação	41
2.3.3	Análise dos Trabalhos Correlatos	42
2.4	CONCLUSÃO	46
3	ALCANÇABILIDADE DOS OBJETIVOS DO SISTEMA	48
3.1	DEFINIÇÕES BÁSICAS	48
3.2	HISTÓRIAS DE UM SMAn	55
3.3	ALCANÇABILIDADE DOS OBJETIVOS DO SISTEMA	64
4	EXPLORANDO A GENERALIDADE DAS NORMAS	74
4.1	CONCEITOS BÁSICOS	74
4.2	EXPERIMENTOS	86
4.2.1	Cenário do Tráfego de Carros	88
4.2.2	Cenário da Segurança de Acesso	90

4.2.3	Cenário do Atendimento Médico	92
4.3	DISCUSSÃO	95
5	DESENVOLVIMENTO INCREMENTAL DE ALGORITMOS PARA A SÍNTESE DE NORMAS EFICIENTES	100
5.1	ALGORITMOS PARA A SÍNTESE DE NORMAS	100
5.2	EXPERIMENTOS E RESULTADOS	129
5.2.1	Tamanho do Espaço das Normas	129
5.2.2	Médias de Normas Analisadas e Tempos de Busca	130
5.2.2.1	Tráfego de Carros	131
5.2.2.2	Segurança de Acesso	133
5.2.2.3	Atendimento Médico	136
5.3	DISCUSSÃO	139
5.3.1	Como as configurações impactam a síntese de normas?	139
5.3.2	Como a análise da alcançabilidade impacta a síntese de normas eficientes?	145
5.3.3	Qual é o algoritmo mais apropriado para a síntese de normas eficientes?	145
5.3.4	Considerações Finais	147
6	CONCLUSÃO E TRABALHOS FUTUROS	149
6.1	RESPOSTAS PARA AS PERGUNTAS DE PESQUISA	151
6.2	CONTRIBUIÇÕES DA TESE	152
6.2.1	Contribuições Secundárias da Tese	154
6.3	TRABALHOS FUTUROS	155
	REFERÊNCIAS	159

1 INTRODUÇÃO

O ser humano, para que pudesse se organizar em sociedade e conviver de forma pacífica, adotou ao longo da sua história regras que estabelecem a conduta e regulam a vida e as relações dos indivíduos a serviço do bem comum. Estas regras, denominadas de *normas*, definem padrões de comportamento esperados dentro das sociedades fazendo com que os indivíduos ajam de forma ordenada e organizada (HECHTER, 2018). Do nascimento à morte, a nossa vida se desenvolve em meio a uma teia de normas oriundas dos diferentes grupos sociais dos quais fazemos parte. A Igreja, o Estado, a família, dentre outras, são instituições que constituem-se de um conjunto de normas que, embora seja diferente no conteúdo ou sujeitos a quem se dirige, converge para uma característica comum: definir as ações dos indivíduos em termos do que é *permitido, proibido ou obrigatório* (BOBBIO, 2003).

Normas são um tema de interesse em áreas como Sociologia (OPP, 2015), Direito (ELICKSON, 1999; POSNER, 2009), Filosofia (LEWIS, 1969), Economia (AKERLOF, 1976; EGGERTSSON, T.; EGGERTSSON, R., 1999), Ciências Políticas (FINEMORE; SIKKINK, 1998; RAYMOND, 1997), Antropologia (GEERTZ, 1973), dentre outras. Desde o trabalho precursor de Shoham e Tennenholtz (1992), as normas têm sido objeto de estudo também no campo dos Sistema Multiagente (SMA) como um mecanismo para regular a sociedade dos agentes na forma de regras que, em certos contextos, prescrevem comportamentos desejáveis (BOELLA; LESMO, 2001). Neste contexto, um Sistema Multiagente Normativo (SMAN) é um SMA no qual os agentes são regulados através de normas (VERHAGEN; NEUMANN; SINGH, M., 2018).

O estudo das normas no campo dos SMAN é extenso e variado. Dentre os temas de interesse tem-se a *síntese de normas* (CHRISTELIS, G.; ROVATSOS, 2009; MORALES *et al.*, 2013, 2014), a elaboração de linguagens formais para a representação das normas (FIGUEIREDO; SILVA, V. T. d., 2011; HÜBNER; BOISSIER; BORDINI, 2011), o desenvolvimento de agentes com capacidade de raciocínio normativo (ALECHINA; DASTANI; LOGAN, 2012), o desenvolvimento de mecanismos capazes de identificar violações e aplicar punições aos agentes infratores (DERAKHSHAN, 2010; PERREAU DE PINNINCK; SIERRA; SCHORLEMMER, 2008), dentre outros. Em especial, o foco desta tese se concentra sobre *o problema da síntese de normas*.

1.1 FORMULAÇÃO DO PROBLEMA

A síntese de normas consiste na criação de um conjunto de normas para regular o comportamento dos agentes (FRANTZ, C. K.; PIGOZZ, 2018). Todavia, sintetizar normas apropriadas para tal fim é um problema complexo (SHOHAM; TENNENHOLTZ, 1995) que requer a análise de um conjunto de normas possivelmente grande. Chamaremos este conjunto de *espaço das normas*. Neste processo, algumas questões

importantes necessitam ser consideradas para que as normas sintetizadas regulem os agentes mantendo um equilíbrio entre o *controle* que exercem e a *autonomia* dos agentes.

Visando fornecer um contexto para a formulação do problema desta tese, considere um cenário de tráfego de carros constituído por uma grid 3×3 e 3 agentes móveis, ag_1 , ag_2 e ag_3 , conforme ilustra a figura 1.

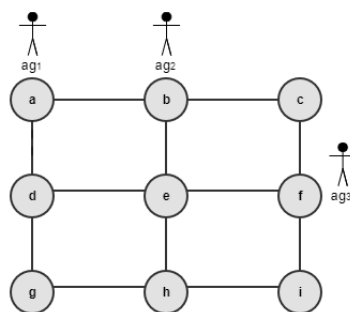


Figura 1 – Grid 3×3 do cenário do tráfego de carros.

Os vértices da grid representam locais e as arestas representam caminhos que conectam tais locais. Os agentes ag_1 , ag_2 e ag_3 estão inicialmente sobre os vértices a , b e f , respectivamente, e o objetivo do sistema é que os agentes alcancem um destino sem conflitos. Um conflito, por sua vez, caracteriza-se como uma colisão que ocorre quando um agente se desloca para um local ocupado por outro agente. Os agentes podem optar por se moverem entre locais adjacentes ou permanecerem parados em seus locais atuais. Considere três normas para este cenário, n_1 , que proíbe apenas o agente ag_1 de se mover para locais ocupados, n_2 , que proíbe qualquer agente de se mover para locais ocupados e n_3 , que proíbe qualquer agente de se mover.

Ao compararmos as normas n_1 e n_2 , percebemos que a norma n_2 regula comportamentos que a norma n_1 também regula, porém, o contrário não ocorre. A norma n_2 é, portanto, *mais geral* do que n_1 , ou seja, n_2 se aplica a mais circunstâncias do que n_1 . Da mesma forma, ao compararmos as normas n_2 e n_3 , podemos perceber que a norma n_3 é *mais geral* que n_2 . Se escolhermos a norma n_1 para regular os agentes, diferentes colisões podem ocorrer, pois, se assim desejarem, os agentes ag_2 e ag_3 podem se deslocar para locais ocupados. Este problema ocorre porque a norma n_1 é *muito específica*, o que restringe um número pequeno de comportamentos que, por sua vez, é insuficiente para evitar todas as colisões. Por outro lado, se escolhermos a norma n_3 , nenhuma colisão ocorre, contudo, os agentes ficam presos em seus locais de partida e, conseqüentemente, nenhum destino é alcançado. Isto ocorre devido à norma n_3 ser *muito geral*, o que restringe um número grande de comportamentos e impede¹ os agentes de executarem ações necessárias para o alcance dos objetivos do

¹ As normas não retiram do *roll* de ações dos agentes as ações que regulam. Todavia, ao optarem por obedecer as normas, os agentes podem se tornar incapazes de concluir as suas tarefas.

sistema. Portanto, neste cenário, ambas as normas n_1 e n_3 não são boas opções para regular os agentes. Todavia, se escolhermos a norma n_2 , os agentes deslocam-se apenas entre locais desocupados e, conseqüentemente, colisões não ocorrem. Podemos observar que, dentre as três normas, n_2 é uma norma com *generalidade balanceada*, isto é, a norma n_2 não é muito específica nem muito geral.

Considere agora a situação inicial ilustrada pela grid. Ao serem regulados pela norma n_1 ou n_2 , os agentes podem encontrar diferentes caminhos que os levam aos seus destinos. Entretanto, devido aos problemas previamente expostos, é possível que os agentes colidam ao serem regulados pela norma n_1 . Por outro lado, não é possível encontrar uma sequência de ações que alcance os objetivos do sistema quando a norma n_3 regula os agentes, pois a ação *mover* não pode ser executada em nenhuma circunstância. Embora o exemplo da norma n_3 seja extremo, diferentes normas aplicadas em contextos menos restritivos podem regular os agentes em excesso tornando-os inaptos a alcançarem os objetivos do sistema. Portanto, a *análise da alcançabilidade dos objetivos do sistema* pode ser útil na síntese de normas para determinar se os agentes ao obedecerem as normas são capazes de executar, pelo menos, uma sequência de ações que os levem aos objetivos do sistema.

A síntese de normas *muito específicas*, como n_1 , pode dar aos agentes uma grande liberdade para agirem como quiserem. Desta forma, os agentes podem executar ações indesejadas levando o sistema a situações de conflito. Por outro lado, a síntese de normas *muito gerais*, como n_3 , pode regular os agentes excessivamente dando-lhes pouca liberdade para realizarem as suas ações. Neste contexto, nós introduzimos o conceito de *norma eficiente*. Uma norma é eficiente se ela for capaz de evitar que os agentes ocasionem conflitos ao passo que mantém os objetivos do sistema alcançáveis. Para isso, consideramos que as normas eficientes devem possuir uma generalidade balanceada e para sintetizá-las é necessário explorar a generalidade das normas assim como analisar a alcançabilidade dos objetivos do sistema. Como a norma n_2 evita conflitos e mantém os objetivos do sistema alcançáveis, n_2 é, portanto, uma norma eficiente.

As normas podem ser sintetizadas de forma *manual* ou *automática*. Na síntese manual, o projetista determina um conjunto de normas para regular os agentes enquanto na síntese automática, um mecanismo computacional realiza esta tarefa (FRANTZ, C. K.; PIGOZZ, 2018). Embora a síntese manual conte com a expertise do projetista, esta abordagem tende a ser trabalhosa e sujeita a erros, principalmente em SMAn grandes e complexos. Por outro lado, a automatização da síntese pode trazer uma série de benefícios como a redução do esforço para explorar espaços de normas grandes e economia de tempo para encontrar normas eficientes. Como os SMAn são dinâmicos e necessitam se adaptar a variações de objetivos ou no ambiente em que os agentes estão situados (JENNINGS, 2000), a síntese automática pode criar ou adaptar

as normas dinamicamente em resposta às mudanças no sistema (BOELLA; TORRE; VERHAGEN, 2008). Além do mais, à medida que o desenvolvimento de SMAn cresce e os sistemas tornam-se mais complexos, o emprego de mecanismos capazes de sintetizar normas de forma automática torna-se cada vez mais importante.

Neste contexto, o problema desta tese é *como sintetizar automaticamente normas eficientes para SMAn*.

1.2 PERGUNTAS DE PESQUISA E HIPÓTESES

Consideramos a síntese de normas eficientes um problema de busca em um espaço de normas. Contudo, em um espaço de normas, certas normas podem ser mais adequadas que outras para regular os agentes evitando conflitos e mantendo os objetivos do sistema alcançáveis. Embora possam existir diferentes critérios para sintetizar as normas apropriadas para tal fim, uma possibilidade é considerar a generalidade das normas. Todavia, de acordo com a generalidade das normas, é possível que elas regulem os agentes pouco ou em excesso. Conforme discutido anteriormente, isto pode configurar diferentes problemas para o SMAn. Portanto, determinar a generalidade que as normas devem possuir é um fator importante para a síntese de normas eficientes. Neste contexto, consideramos a seguinte pergunta:

Pergunta 1 *Dado um espaço de normas, como determinar as normas que possuem a generalidade balanceada?*

Para responder esta pergunta, uma possibilidade é ordenar o espaço das normas, das mais específicas para as mais gerais. Isso nos possibilita obter um espaço de busca ordenado sobre o qual podemos aplicar estratégias de exploração que levam em conta a generalidade das normas para encontrar normas eficientes. Considerando que as normas com generalidade balanceada não são muito específicas nem muito gerais, eles devem estar posicionadas em uma região intermediária da ordenação. Nesta direção, a hipótese 1 postula a localização das normas balanceadas em espaços de normas ordenados.

Hipótese 1 *Dado um espaço cujas normas estão ordenadas das mais específicas para as mais gerais, as normas com generalidade balanceada encontram-se em torno do meio da ordenação.*

As normas eficientes devem ocupar um lugar de equilíbrio entre as normas mais específicas e gerais oferecendo um nível adequado entre especificidade e generalidade para regular os agentes de forma apropriada. Ao considerarmos a hipótese 1 podemos aplicar uma busca que explora inicialmente o meio de uma ordenação. Caso uma norma eficiente seja encontrada em tal ponto, parte do espaço de busca é desconsiderado. Ao ordenarmos as normas do cenário do tráfego pela sua generalidade,

podemos verificar que a norma n_2 encontra-se no meio da ordenação. Assim, uma busca que inicia pelo meio da ordenação analisa apenas uma norma. No entanto, ordenar o espaço das normas é uma tarefa que tende a ser custosa, portanto torna-se relevante investigar estratégias para simplificar o processo de ordenação.

Além do mais, considerando que o espaço das normas é um conjunto potencialmente grande, a busca por normas eficientes pode demandar tempo. Portanto, medidas para reduzir o tamanho do espaço das normas podem ser ponderadas para que o tempo de busca por normas eficientes seja viável. Neste contexto, consideramos a seguinte pergunta:

Pergunta 2 *Como reduzir o tamanho do espaço das normas sem desconsiderar normas possivelmente eficientes?*

Para isso, mecanismos para descartar normas podem ser considerados. Como a análise da alcançabilidade auxilia a determinar se os agentes são capazes de alcançar os objetivos do sistema ao serem regulados por uma norma, a hipótese 2 postula sobre a redução do tamanho do espaço das normas através da análise da alcançabilidade.

Hipótese 2 *O espaço das normas pode reduzir de tamanho ao desconsiderarmos normas que não mantêm os objetivos do sistema alcançáveis. Para isso, a análise da alcançabilidade dos objetivos do sistema pode ser considerada.*

Como somente as normas n_1 e n_2 do cenário do tráfego mantêm os objetivos do sistema alcançáveis, ao desconsiderarmos a norma n_3 , reduzimos o tamanho do espaço das normas.

1.3 OBJETIVOS

A síntese de normas eficientes é um problema pouco explorado na literatura. Logo, a escassez de fontes sobre o assunto dificulta a compreensão do problema e a identificação de possíveis soluções. Isso ressalta a importância de investigações e do desenvolvimento de novas metodologias para abordar essa questão. Nesse sentido, esta tese tem como principal objetivo propor uma *conceitualização* para a *síntese automática de normas eficientes*. Para isto, desenvolvemos um ferramental teórico e formal que aborda os problemas inerentes à síntese de normas eficientes. Os objetivos específicos considerados incluem:

1. Investigar o estado da arte da síntese de normas eficientes em Sistemas Multiagente.
2. Definir como os objetivos do sistema podem ser considerados na síntese de normas eficientes.

3. Definir como os conflitos causados pelas ações dos agentes podem ser considerados na síntese de normas eficientes.
4. Definir estratégias para estimar a generalidade das normas e ordenar o espaço das normas.
5. Definir estratégias para explorar espaços de normas ordenados visando a busca por normas eficientes.
6. Definir estratégias para reduzir a complexidade da síntese de normas eficientes.
7. Construir algoritmos para a síntese de normas eficientes.

1.4 ESCOPO DO TRABALHO

A seguir, são apresentadas as delimitações do escopo deste trabalho. As delimitações 1, 2 e 3 foram estabelecidas para reduzir o problema da síntese de normas eficientes, tornando-o passível de investigação. A delimitação 4 aborda a adoção das normas pelos agentes, enquanto as delimitações 5 e 6 dizem respeito aos objetivos e à execução dos Sistemas Multiagente Normativos. Por fim, a delimitação 7 concentra-se nos algoritmos utilizados na síntese de normas eficientes.

1. **Consideramos apenas proibições.** Sistemas Multiagente Normativos empregam diferentes tipos de normas como as *proibições*, *obrigações* e *permissões*. Embora seja possível evitar que os agentes causem conflitos e alcancem os objetivos do sistema obrigando-os a executarem certas ações, as obrigações representam uma forma mais restritiva de regular os agentes. Isto ocorre porque os agentes só podem executar as ações que as normas impõem. Por outro lado, ao regular os agentes com proibições, eles têm a opção de executar outras ações, uma vez que, de acordo com a lógica deôntica, o que não é proibido é permitido (DIMISHKOVSKA, 2017; GOVERNATORI; ROTOLO, 2019). Neste trabalho, seguindo Christelis e Rovatsos (2009) e Morales et al. (2013), nós optamos por sintetizar unicamente normas proibitivas, o que nos possibilita trabalhar com espaços de normas reduzidos.
2. **Buscamos por uma norma eficiente.** Em muitos domínios do mundo real, diferentes normas podem ser necessárias para que os agentes sejam regulados adequadamente. No entanto, na nossa abordagem, assumimos que, para cada tipo de conflito a ser evitado e para manter os objetivos do sistema alcançáveis, uma única norma é suficiente. Desta forma, podemos nos concentrar em definir meios eficazes para encontrar uma norma eficiente, não necessariamente a melhor, com a perspectiva de estendê-los no futuro para encontrar múltiplas normas eficientes.

3. **Aplicamos uma abordagem simbólica restritiva.** Para representarmos as definições e conceitos que caracterizam a síntese de normas eficientes, nós adotamos uma abordagem simbólica que introduz uma linguagem de primeira ordem restrita. Em tal linguagem, somente certos conectivos lógicos são representados. Desta forma, delimitamos a quantidade de normas que podem ser representadas através da linguagem lógica e, conseqüentemente, podemos obter espaços de normas menores.
4. **Os agentes optam por obedecer as normas.** Em essência, normas são violáveis. Embora existam diferentes razões pelas quais os agentes optem por obedecer as normas, uma forma de motivá-los a isto é através de mecanismos que aplicam sanções ou recompensas. Devido à complexidade do desenvolvimento destes mecanismos, existe uma área em Sistemas Multiagentes Normativos denominada *Norm Enforcement* (HORNE, 2007; ALECHINA *et al.*, 2015) que se preocupa em estudar meios de criar e implantar tais mecanismos. Todavia, este é um tópico que não faz parte do problema deste trabalho. Portanto, consideramos que os agentes optam naturalmente por estarem em conformidade com as normas que lhes são impostas.
5. **Os objetivos dos Sistemas Multiagente são declarativos.** Sistemas Multiagente empregam diferentes tipos de *objetivos globais* (RIEMSDIJK; DASTANI; WINIKOFF, 2008) como os *objetivos declarativos* (HINDRIKS *et al.*, 2001; GIRET; BOTTI, Vicente, 2004), por exemplo. Um objetivo declarativo descreve uma situação desejada (*state of affairs*) que os agentes devem satisfazer. Conforme apresenta o capítulo 3, a conceitualização do problema da síntese de normas eficientes emprega uma abordagem simbólica para representar os conceitos deste trabalho. Logo, os objetivos declarativos estão melhor alinhados a nossa proposta, pois podem ser representados como uma fórmula lógica que descreve o que os agentes necessitam satisfazer.
6. **Os Sistemas Multiagente Normativos são simulados.** Embora existam diferentes frameworks para o desenvolvimento de SMA como o JADE (BELLIFEMINE *et al.*, 2005) e JaCaMo (BOISSIER *et al.*, 2013), por exemplo, nós optamos por desenvolver um simulador próprio de SMAn para avaliar o impacto das normas no comportamento dos agentes. Desta forma, nós nos concentramos no desenvolvimento dos algoritmos de síntese, na experimentação e prospecção de resultados sem a necessidade de explorar detalhes técnicos dos frameworks, elencar o framework que melhor atende as nossas necessidades e estendê-lo para suportar a nossa abordagem.
7. **As propriedades dos algoritmos de síntese não são verificadas.** Nós desenvolvemos um conjunto de algoritmos para a síntese de normas eficientes.

Com isto, o nosso objetivo é validar a nossa proposta através de experimentos realizados em diferentes domínios de aplicação em Sistemas Multiagente. Para isto, nós focamos apenas no processo de síntese apresentando diferentes formas de aprimorar o tempo e o desempenho dos algoritmos para sintetizarem as normas eficientes. Contudo, não nos atentando a questões sobre a corretude, completude e complexidade dos algoritmos.

1.5 ORGANIZAÇÃO DA TESE

O restante desta tese está organizado da seguinte forma. O capítulo 2 apresenta os conceitos básicos da área de Sistemas Multiagente Normativos e contextualiza este trabalho dentro do estado da arte discutindo diferentes abordagens existentes para a síntese de normas. O capítulo 3 apresenta os conceitos e definições iniciais do nosso ferramental teórico e formal para a síntese de normas eficientes. Este capítulo apresenta também uma investigação sobre a aplicabilidade da análise da alcançabilidade dos objetivos do sistema na síntese de normas eficientes. O capítulo 4 apresenta conceitos e definições atreladas à generalidade das normas e uma investigação sobre como determiná-la. Como fruto desta investigação, o capítulo 4 apresenta estratégias para ordenar o espaço das normas. Ao final do capítulo 4, um conjunto de experimentos e uma discussão sobre o impacto de tais estratégias no posicionamento das normas eficientes nas ordenações do espaços das normas são apresentados. O capítulo 5 apresenta a construção incremental de um conjuntos de algoritmos para sintetizar normas. Para cada algoritmo, nós expomos as suas deficiências na busca por normas eficientes. Tais deficiências tentam ser superadas através de um conjunto de aprimoramentos empregado pelo algoritmo seguinte que auxilia a diminuir a complexidade da síntese de normas. Ao final, nós obtemos três algoritmos que são capazes de sintetizar normas eficientes ao aplicarem a análise da alcançabilidade dos objetivos do sistema aliada a uma busca em espaços de normas ordenados de acordo com a generalidade das normas. Tal capítulo apresenta também um conjunto de estratégias de visita para explorar os espaços das normas. Ao final, o capítulo 5 apresenta um conjunto de experimentos que compara os resultados dos três algoritmos na busca por normas eficientes variando a estratégia utilizada para ordenar os espaços das normas e a estratégia que os algoritmos empregam para visitar tais espaços. Por fim, o capítulo 6 resume as contribuições desta tese e aponta perspectivas de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA E ESTADO DA ARTE

Normas são um aspecto fundamental da vida em sociedade e desempenham um papel importante na regulação do comportamento humano (DURKHEIM, 1982). Elas formam um vasto e complexo conjunto de regras e expectativas que exerce uma poderosa influência sobre as nossas ações. Quando dirigimos um veículo, realizamos uma reunião, disputamos um jogo ou exercemos um papel, por exemplo, as normas estão presentes (BRENNAN *et al.*, 2013). As normas diferem-se em conteúdo e sujeitos a quem se dirigem, podem variar de uma sociedade para outra, e são influenciadas por diferentes fatores culturais, religiosos e históricos (WEBER, 1978).

Segundo Durkheim (1982), um dos fundadores da sociologia moderna, as normas são regras de conduta que descrevem como os indivíduos devem agir. Para Searle (1995), um dos principais filósofos contemporâneos, as normas são imperativos que representam uma demanda dirigida que pode ser seguida ou violada. Por outro lado, para Bobbio (2003), um dos mais importantes pensadores políticos do século XX, as normas são regras ditadas por autoridades. Embora existam distintas definições para as normas, elas convergem para algo em comum: definir como os indivíduos devem idealmente se comportar, estabelecendo parâmetros para o que é considerado aceitável em diferentes contextos sociais (SINGH, A.; SINGH, M., 2023).

As normas impactam o comportamento dos indivíduos de diferentes formas. Elas promovem a organização e a ordem social (LEWIS, 1969), evitam e solucionam conflitos (AMSLER, 2014), auxiliam na cooperação entre os indivíduos (BICCHIERI, 1990) e na coordenação de atividades (ELSTER, 1989), reduzem o esforço no raciocínio para a tomada de decisões (BINMORE, 2005), dentre outros. Logo, as nossas vidas e interações sociais são impregnadas por diversas normas e sem elas, possivelmente, as sociedades humanas não seriam da forma como as conhecemos hoje (BERGER; LUCKMANN, 1966).

As normas são objeto de estudo em diversos campos do conhecimento, envolvendo áreas como a Sociologia (DURKHEIM, 1982; HECHTER; OPP, 2001), Filosofia (LEWIS, 1969), Direito (ELLICKSON, 1999; POSNER, 2009), Antropologia (GEERTZ, 1973), Economia (AKERLOF, 1976; EGGERTSSON, T.; EGGERTSSON, R., 1999), Ciências Políticas (FINNEMORE; SIKKINK, 1998; RAYMOND, 1997), dentre outras. Em especial, a comunidade de SMA tem estudado diferentes problemas relacionados às normas e ao seu emprego para regular a sociedade dos agentes autônomos (BOELLA; TORRE; VERHAGEN, 2006). Embora exista uma vasta teoria relacionada às normas em diferentes aspectos em SMA, neste capítulo nós focamos nos conceitos e definições relevantes a esta tese. Neste sentido, na seção seção 2.1, nós fornecemos uma visão geral sobre as normas correlacionando-as a sua exploração no campo dos Sistemas Multiagente. Posteriormente, na seção 2.2, nós fundamentamos

a síntese de normas em Sistemas Multiagente. Na seção 2.3, nós apresentamos e discutimos os trabalhos correlatos frente a proposta desta tese. Por fim, na seção 2.4, nós apresentamos as nossas conclusões a cerca da temática deste capítulo.

2.1 SISTEMAS MULTIAGENTE NORMATIVOS

A fim de lidar com a heterogeneidade, autonomia, diversidade de interesses e desejos dos agentes, as normas têm sido empregadas em SMA como um mecanismo de controle social para regular o seu comportamento (SINGH, M. P. *et al.*, 2013; MAHMOUD *et al.*, 2014; AKINKUNMI; BABALOLA, 2020). Neste contexto, um SMA no qual as normas regem o comportamento dos agentes e suas interações é denominado de *Sistema Multiagente Normativo* (SMAN) (BOELLA; TORRE; VERHAGEN, 2006; VERHAGEN; NEUMANN; SINGH, M., 2018; MORRIS-MARTIN; BOS-DE VOS; PADGET, 2020). Em essência, as normas de um SMAN determinam, para certos contextos, como os agentes devem e não devem idealmente se comportar e o que são permitidos a fazer (GABBAY, D. *et al.*, 2013).

O emprego das normas em SMAN fornece meios para a solução de diferentes problemas (VERHAGEN; NEUMANN; SINGH, M., 2018) como a cooperação e coordenação dos agentes (LÓPEZ; LUCK; D'INVERNO, 2002), manutenção da segurança de sistemas (USZOK *et al.*, 2003), modelagem de organizações (DIGNUM, V.; VÁZQUEZ-SALCEDA; DIGNUM, F., 2004) e instituições eletrônicas (GARCIA-CAMINO; NORIEGA; RODRIGUEZ-AGUILAR, 2005), prevenção de conflitos (MORALES *et al.*, 2013), dentre outros. Além disto, as normas fornecem diferentes benefícios à sociedade dos agentes, como a redução de comunicação e esforço necessário para os agentes tomarem decisões (CHRISTELIS, G.; ROVATSOS, 2009).

Diferentes tipos de normas são empregados em SMAN. A fim de definir e caracterizar as normas que são de interesse neste trabalho, nós apresentamos a seguir definições e classificações para as normas em SMAN.

2.1.1 Classificação das Normas

A natureza multidisciplinar das normas traz à tona um amplo conjunto de classificações (HOLLANDER; WU, 2011; HECHTER; OPP, 2001). Não obstante, a área de SMAN também conta com diferentes classificações para as normas. Neste tese, nós seguimos o trabalho de Boella e van der Torre (2004), onde as normas são classificadas como *regulativas* ou *constitutivas*, o trabalho de Boella e van der Torre (2008), onde as normas são classificadas como *substantivas* ou *processuais* e o trabalho de Mahmoud et al. (2014) que, adicionalmente, classifica as normas como *convenções*.

A figura 2 ilustra uma taxonomia para tais tipos de normas.

A seguir, nós descrevemos a referente taxonomia com maiores detalhes.

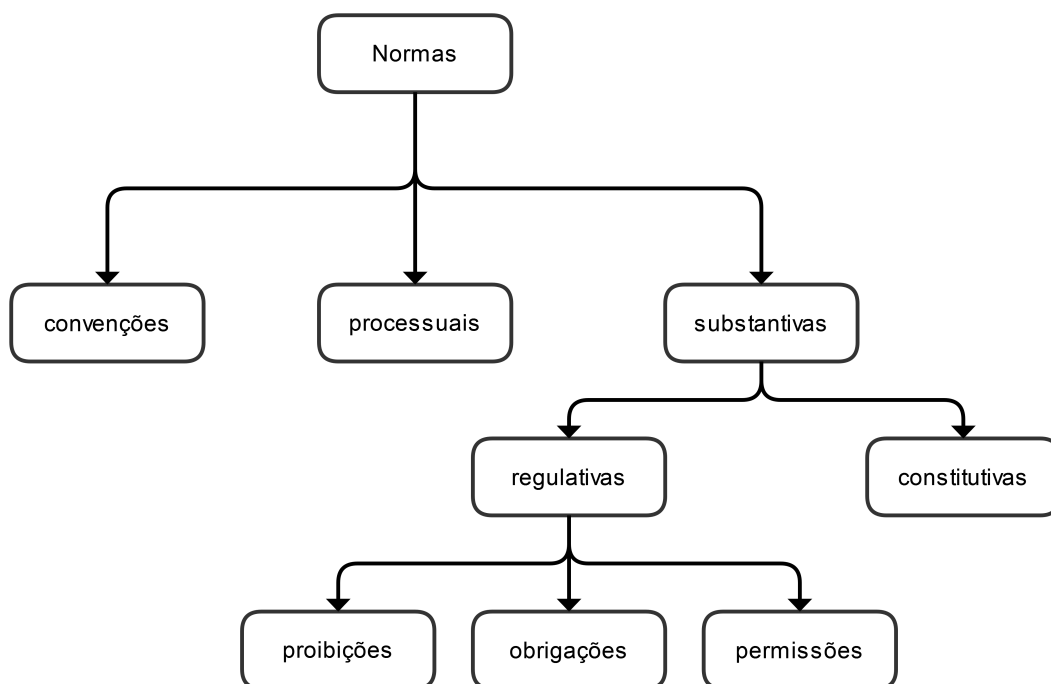


Figura 2 – Classificação das normas em SMAn.

2.1.1.1 Normas Regulativas

As normas regulativas determinam os comportamentos esperados dos agentes em certos contextos em termos operadores deônticos, isto é, *obrigações*, *proibições* ou *permissões*. Segundo Searle (1969; 1995), este tipo de norma regula comportamentos que podem existir independentemente das normas. Um exemplo de norma regulativa pode ser “*é obrigatório dar passagem a veículos de emergência*”. No entanto, não há um consenso na área de SMAn a respeito dos componentes que devem formar as normas regulativas (OLIVEIRA; SILVESTRE; SILVA, V. T. da, 2017). Em linhas gerais, as normas regulativas são constituídas por um operador deôntico, uma ação e uma précondição (ou contexto) da norma. Adicionalmente, pode-se considerar especificar uma entidade executora da ação (um agente, um papel ou uma organização, por exemplo), uma sanção e/ou recompensa (CRIADO; ARGENTE; BOTTI, Vicent, 2011). Sanções, por sua vez, descrevem punições para os agentes infratores enquanto as recompensas descrevem bonificações para os agentes que optam por estarem em conformidade com as normas.

Para Morales et al. (2013), por exemplo, uma norma regulativa é descrita como uma tupla $\langle \varphi, \theta(\alpha) \rangle$, onde φ , θ e α referem-se, respectivamente, a uma précondição, um operador deôntico e uma ação. Por outro lado, Vasconcelos et al. (2009), caracterizam as normas regulativas como um tupla $\langle hv, t_d, t_a, t_e \rangle$, onde v é da forma $\theta_{ag:\rho}\alpha \circ \Gamma$, tal que, θ refere-se a um operador deôntico, ag refere-se a um agente que possui um papel ρ e α refere-se a uma ação sobre a qual se aplica um conjunto de restrições Γ . Os demais elementos da norma, t_d , t_a e t_e , são valores inteiros discretos que representam

o momento no qual a norma é criada, ativada e desativada, respectivamente, com $t_d \leq t_a \leq t_e$.

2.1.1.2 Normas Constitutivas

As normas constitutivas são responsáveis por criar realidades institucionais através da definição de ações e papéis que fazem sentido apenas dentro da instituição à qual pertencem (BOELLA; TORRE, 2004; BRITO; HÜBNER; BOISSIER, 2016). Segundo Searle (SEARLE, 1995), as normas constitutivas regulam comportamentos existencialmente dependentes das normas e podem ser definidas por meio de expressões na forma “*X conta como Y no contexto C*”, onde um fato bruto *X* é redefinido como um fato institucional *Y* no contexto *C*. Um exemplo de norma constitutiva pode ser “*a peça que se move em L conta como um cavalo em um jogo de xadrez*”. Neste caso, a norma cria uma relação existencial entre a peça e o conceito de cavalo, uma vez que o conceito de cavalo em um jogo de xadrez é definido pela norma que descreve como a peça se move. Assim, uma peça cujo comportamento é movimentar-se em *L* em um jogo de xadrez é ressignificada como sendo um cavalo.

2.1.1.3 Normas Substantivas e Processuais

Normas substantivas determinam condutas e direitos enquanto as normas processuais estabelecem os mecanismos para que as normas substantivas sejam cumpridas (FONSECA, 2015). Neste sentido, Boella e van der Torre (2008) argumentam que as normas substantivas são o conjunto formado pelas normas regulativas e constitutivas, ao passo que as normas processuais definem uma conexão entre normas regulativas e mecanismos de sanção e recompensa. Sob esta ótica, as normas regulativas prescrevem apenas os comportamentos esperados dos agentes. Um exemplo de norma processual pode ser “*os motoristas que não dão passagem aos veículos de emergência cometem uma infração grave e são punidos com uma multa altíssima*”.

2.1.1.4 Convenções

Normas que emergem de forma espontânea da interação entre indivíduos de uma sociedade são denominadas de *convenções* (BICCHIERI, 2016). Em SMAn, as convenções emergem quando uma parte significativa dos agentes concorda sobre quais normas devem regular os seus comportamentos. Basicamente, durante a execução do sistema, cada agente determina as normas com as quais deseja estar em conformidade. Esta escolha pode ocorrer por diferentes razões como quando um agente verifica que seguir certas normas é benéfico para si ou deseja ser cooperativo, por exemplo. Ao longo das interações, certas normas espalham-se mais entre os membros da sociedade, ganhando a preferência dos agentes. Quando a maioria dos agentes

passa a seguir uma dada norma, diz-se que a norma emergiu na sociedade (VIL-LATORO; SEN; SABATER-MIR, 2010). Os trabalhos em emergência podem ainda subdividir-se entre aqueles cuja sociedade dos agentes possui ou não uma estrutura topológica. Nos trabalhos com topologia, os agentes interagem com aqueles que estão relacionados em uma rede. Desta forma, tais trabalhos estão interessados em determinar como as estruturas topológicas impactam na síntese de normas.

A maioria dos trabalhos sobre convenções baseiam-se em princípios alinhados a teoria dos jogos (MORRIS-MARTIN; DE VOS; PADGET, 2020). Geralmente, estes trabalhos consideram modelar SMAn onde os agentes estão envolvidos em situações de conflito em um jogo e necessitam tomar decisões estratégicas que maximizam os seus ganhos e minimizam as suas perdas. Estratégias que são vistas como bem-sucedidas, prosperam e se espalham entre os agentes. Uma vez que um número suficiente de agentes adota uma determinada estratégia, essa estratégia se torna dominante e os outros agentes têm incentivos para adotá-la também. Quando nenhum agente tem incentivo para mudar sua estratégia unilateralmente, o jogo alcança o equilíbrio de Nash¹, e a estratégia mais adotada é dada com uma norma da sociedade dos agentes (MORALES *et al.*, 2017).

2.2 SÍNTESE DE NORMAS EM SMAN

A síntese de normas é o processo que consiste na criação de um conjunto de normas para regular agentes autônomos apropriadamente (FRANTZ, C. K.; PIGOZZ, 2018). Desde o trabalho precursor de Shoham e Tennenholtz (1992; 1995), diferentes abordagens têm sido propostas para que normas sejam devidamente sintetizadas para SMAn. Contudo, a síntese de normas eficientes é um problema que ainda permanece em aberto. Neste contexto, a fim de classificar as diferentes contribuições da literatura sobre síntese de normas e compará-las a nossa proposta, nós apresentamos a seguir uma revisão sobre a síntese de normas na área dos SMAn. Mais especificamente, na seção 2.2.1, nós estabelecemos um conjunto de critérios para que na seção 2.3.3 nós possamos analisar e classificar os trabalhos correlatos assim como discuti-los frente a esta tese.

2.2.1 Classificação da Síntese de Normas

A síntese de normas pode ser classificada sob diferentes perspectivas (FRANTZ, C. K.; PIGOZZ, 2018). Uma das formas de classificá-la é de acordo com o *momento*

¹ O equilíbrio de Nash é um conceito da teoria dos jogos que descreve um estado em que nenhum jogador pode melhorar sua situação individual mudando sua estratégia, desde que todos os outros jogadores continuem a jogar da mesma maneira (CARMONA, 2013). No caso dos SMAn, alcançar o equilíbrio de Nash significa que nenhum agente tem como se beneficiar ao desviar da norma que emerge. Portanto, estar em conformidade com a norma implica em uma escolha racional.

em que a síntese é realizada. Neste sentido, a síntese *off-line* ocorre antes do SMAN ser executado enquanto a síntese *on-line* ocorre durante a execução do SMAN. Assim, na síntese *off-line*, os agentes são regulados desde o início da execução do SMAN, enquanto na síntese *on-line*, o SMAN inicia sem regulações e os agentes agem livremente até que as normas sejam sintetizadas. A síntese de normas também pode ser classificada quanto ao *modo de operação*. Sob esta perspectiva, a síntese pode ser *manual* ou *automática*. Enquanto a síntese manual é realizada pelo projetista do sistema, a síntese automática é realizada por mecanismos computacionais.

Mais especificamente, a síntese automática se distingue entre as abordagens por *emergência*, *observação* e *dirigida a conflitos*. Basicamente, na emergência, as normas são sintetizadas pelos agentes que, em comum acordo, decidem sobre as normas que devem regulá-los (as *convenções*). Em contrapartida, na observação, um mecanismo monitora os agentes durante a execução do SMAN e, ao detectar que certos comportamentos necessitam ser regulados, sintetiza normas para que os agentes se comportem de forma adequada. Logo, a emergência e a observação são modalidades de síntese *on-line*. Por outro lado, a síntese dirigida a conflitos é uma modalidade *off-line* onde normas são criadas automaticamente considerando situações onde os conflitos ocorrem.²

Por fim, uma terceira forma de classificar a síntese é de acordo com o *autor* da síntese (SAVARIMUTHU; CRANFIELD, 2009). Neste sentido, quando as normas são criadas por agentes, a síntese é classificada como *bottom-up*, e quando as normas são criadas por outras entidades, a síntese é classificada como *top-down*.

A figura 3 fornece uma visão geral esquemática da classificação da síntese de normas em SMAN conforme os tipos de síntese apresentados nesta seção.

A fim de contextualizar esta tese dentro de um quadro geral, nós fornecemos uma classificação para caracterizar a síntese de normas em diferentes dimensões, o que nos permite avaliar a proposta desta tese frente as diferentes contribuições da literatura. Para isto, além da classificação da síntese de normas ilustrada pela figura 3, nós consideramos adicionalmente: i) *o escopo da síntese*; ii) *a generalidade das normas*; iii) *a ocorrência de conflitos*; iv) *os objetivos do sistema*; v) *a ordenação do espaço das normas*; e vi) *as propriedades das normas sintetizadas*.

A seguir, nós descrevemos estas seis classificações com maiores detalhes.

2.2.1.1 O Escopo da Síntese

O espaço das normas é um conjunto potencialmente grande. Logo, explorar o espaço das normas a fim de determinar quais normas regulam os agentes apropria-

² Embora diferentes trabalhos da modalidade *on-line* considerem a ocorrência de conflitos para que normas sejam sintetizadas, mantivemos o nome *síntese dirigida a conflitos* - originalmente chamada de *conflict rooted synthesis* no trabalho de Christelis e Rovatsos (2009) - para o modo *off-line* como foi originalmente idealizada.

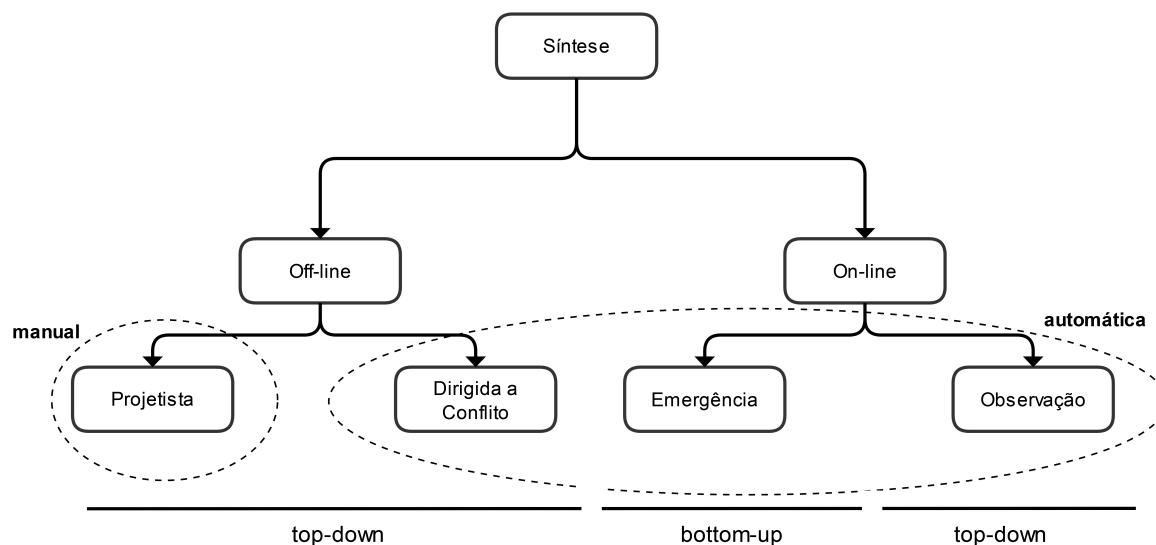


Figura 3 – Classificação da síntese de normas em SMAn.

damente pode ser uma tarefa custosa e que demanda tempo. Neste sentido, realizar a síntese sobre espaços reduzidos pode ser desejável. Assim, dado um espaço de normas, estamos interessados em saber se a síntese aplica alguma redução sobre o espaço a fim de diminuir o tempo e a complexidade da síntese de normas.

Em suma, consideramos que o escopo de síntese pode ser *fixo* ou *variável*. Neste sentido, a síntese de normas com escopo fixo considera explorar um espaço constituído por todas as possíveis de normas de um domínio. Logo, o espaço não sofre alteração de tamanho ao longo da síntese. Alternativamente, na síntese de normas com escopo variável, o espaço sofre alteração de tamanho ao longo da síntese. Para isto, emprega-se um método ou abordagem para que o espaço das normas seja reduzido a um conjunto de potenciais normas apropriadas para regular os agentes.

2.2.1.2 A Generalidade das Normas

A generalidade das normas refere-se à extensão de comportamentos que as normas são capazes de regular. Neste sentido, normas muito específicas regulam poucos comportamentos, o que pode ocasionar uma grande ocorrência de conflitos, enquanto normas muito gerais regulam comportamentos em excesso, o que pode inviabilizar a execução de ações necessárias para o alcance dos objetivos do sistema. Logo, ambos os tipos de normas tendem a não ser boas opções para regular os agentes. Conforme discutido no capítulo 1, consideramos que explorar a generalidade das normas é fundamental para sintetizar normas que sejam capazes de manter um equilíbrio entre o controle que exercem e a autonomia dos agentes. Assim, explorar a generalidade das normas possibilita determinarmos se as normas são suficientemente abrangentes para regularem uma ampla gama de comportamentos evitando conflitos

sem dirimir o alcance dos objetivos do sistema, isto é, se as normas são *eficientes*.

2.2.1.3 A Ocorrência de conflitos

Os conflitos são tema de interesse em diversas áreas, possuem diferentes tipos e geralmente são definidos de acordo com as características particulares das áreas nas quais são estudados. Embora possam haver distintas definições para conceitualizar os conflitos, a maioria das definições compartilham que eles são situações indesejáveis que podem levar a consequências negativas e, portanto, necessitam ser evitados (EGGERT; FALZON, 2004).

Em SMA, existem diferentes tipos de conflitos que resultam de diferentes fontes como escassez de recursos, interdependências entre tarefas, interesses próprios, metas conflitantes ou estratégias divergentes, por exemplo. Como os conflitos podem afetar o desempenho, a eficiência e a cooperação entre os agentes e assim levar a impasses, resultados sub-ótimos e até mesmo à insatisfação dos objetivos do sistema, resolver os conflitos em SMA é uma tarefa importante (TESSIER; CHAUDRON; MÜLLER, 2001). Embora nem todos os conflitos sejam prejudiciais, nós consideramos que os conflitos são situações negativas resultantes da execução dos agentes que podem gerar implicações adversas para o correto e bom funcionamento dos SMA.

2.2.1.4 Os Objetivos do Sistema

Em SMA, existem os objetivos do sistema e os objetivos dos agentes (GIRET; BOTTI, Vicente, 2004; LAMSWEERDE, 2001). Os objetivos do sistema referem-se a um conjunto de metas gerais ou resultados desejados do sistema como um todo. Eles estão relacionados ao propósito global do sistema e podem envolver critérios como desempenho, eficiência, otimização de recursos, etc. Estes objetivos são estabelecidos para garantir o funcionamento adequado e efetivo do sistema sendo *compartilhados* entre os agentes que devem alcançá-los através da execução de suas ações. Por outro lado, os objetivos dos agentes são metas individuais (HINDRIKS *et al.*, 2001). Cada agente pode ter seus próprios objetivos específicos, que podem ser influenciados por suas preferências, limitações, tarefas atribuídas ou recompensas esperadas. Os objetivos dos agentes podem variar de acordo com a natureza do agente, seu papel no sistema e suas interações com outros agentes. Embora os objetivos do sistema e dos agentes não precisem ser mutuamente exclusivos, em diferentes situações, os objetivos dos agentes podem estar desalinhados com os objetivos do sistema.

Os objetivos em SMA possuem diferentes tipos (RIEMSDIJK; DASTANI; WINIKOFF, 2008). Neste trabalho, nós consideramos os objetivos do sistema como *declarativos*, i.e., descrições de situações desejáveis (*state of affairs*) que devem ser satisfeitas pelos agentes (WINIKOFF *et al.*, 2002). Os objetivos declarativos variam de acordo com o domínio de aplicação e as características específicas do SMA e podem

ser expressos por meio de linguagens de representação do conhecimento, como as linguagens lógicas. Essas linguagens fornecem uma forma estruturada adequada para representar os objetivos, permitindo que os agentes compreendam e compartilhem informações sobre as metas globais a serem atingidas. Em outras palavras, os objetivos declarativos fornecem uma base para o planejamento, a tomada de decisões e a coordenação dos agentes.

2.2.1.5 A Ordenação do Espaço das Normas

A ordenação do espaço das normas pode trazer uma série de benefícios à síntese de normas eficientes. Primeiramente, um espaço organizado pode ajudar a encontrar as normas desejadas mais facilmente. Embora existam distintas formas de ordenar as normas, uma possibilidade é considerar a sua generalidade. Desta forma, podemos arranjar o espaço aproximadamente das normas mais específicas para as mais gerais. Em segundo lugar, a ordenação viabiliza priorizar regiões do espaço onde deseja-se iniciar uma busca por normas eficientes. Neste sentido, ao considerarmos a hipótese 1, podemos iniciar a busca em torno do meio do espaço organizado. Considerando que as normas eficientes estejam localizadas em tal região, podemos economizar tempo e esforço computacional para sintetizar as normas desejadas.

Em terceiro lugar, a ordenação facilita a comparação entre as normas e viabiliza a aplicação de diferentes estratégias para explorar o espaço das normas. Neste sentido, certas estratégias podem focar em explorar o espaço à medida que as normas tornam-se mais gerais enquanto outras podem focar na exploração do espaço à medida que as normas tornam-se mais específicas, por exemplo.

2.2.1.6 As Propriedade das Normas Sintetizadas

Normas podem possuir diferentes propriedades. Segundo Shoham e Tennenholtz (1992), uma norma é *útil* se os agentes podem alcançar as suas metas individuais quando a norma os regula. Por outro lado, para Fitoussi e Tennenholtz (1998), uma norma é *mínima*³ se for *útil* e aplicar a menor quantidade de restrições para regular os agentes. Para Hoek et al. (2007) uma norma é *eficaz* se garantir que o propósito para o qual a norma está regulando os agentes é satisfeito. Neste contexto, os autores consideram que o *propósito* da norma⁴, representa o que pretende-se alcançar com a introdução da norma no SMAn (e.g. em um cenário de tráfego de carros, o objetivo das normas pode ser regular os agentes para evitar acidentes). Para Morales et al. (2013), as normas podem ser *eficazes* e *necessárias* para evitar conflitos. Basicamente, a propriedade *eficaz* que Morales et al. (2013) mencionam é equivalente a propriedade

³ Javier et al. (2015) referenciam as normas mínimas como *normas liberais*.

⁴ Christelis e Rovatsos (2009) posteriormente referenciam o propósito das normas como o *propósito social* das normas.

eficaz mencionada por Hoek et al. (2007), contudo, o propósito das normas nos trabalhos de Morales et al. (2013) é fixado em evitar conflitos. Para tais autores, quando as normas são obedecidas e os conflitos não ocorrem, as normas são *eficazes*. Por outro lado, quando as normas são violadas e os conflitos ocorrem, elas são *necessárias*.

Uma propriedade de particular interesse nesta tese e que tem sido pouco explorada pela comunidade de SMAn é se as normas são *eficientes*. Conforme introduzido anteriormente, uma norma é eficiente se ela for capaz de evitar conflitos e manter os objetivos do sistemas alcançáveis.

2.3 TRABALHOS CORRELATOS

Esta seção apresenta uma revisão bibliográfica sistemática sobre a síntese de normas em SMAn. O objetivo é fornecer uma visão abrangente das abordagens existentes e subsidiar a identificação de lacunas e tendências emergentes nesta área de pesquisa. Para realizar a revisão bibliográfica, adotamos uma abordagem estruturada, onde realizamos uma busca em bases de dados como *ACM Digital Library*, *CiteSeerXm*, *IEEE Xplore*, *ScienceDirect*, *Scopus* e *SpringerLink*.

A estratégia de busca foi concebida com base na identificação de palavras-chave e na utilização de operadores booleanos para refinar os resultados. As palavras-chave selecionadas, tanto em português quanto em inglês, foram: síntese/criação de normas/normas eficientes; síntese/criação de sistemas normativos/regras normativas/leis sociais; emergência/convenção de normas em Sistemas Multiagente; Sistemas Multiagente Normativos/baseados em normas; objetivos do Sistema Multiagente na síntese/criação de normas.

Inicialmente, o recorte bibliográfico considerou os últimos 10 anos de produção, de 2013 a 2023. Para selecionar os trabalhos melhor alinhados ao tema desta tese, foi realizada uma filtragem inicial considerando os títulos e resumos dos trabalhos. Em seguida, sobre os trabalhos previamente selecionados, foi realizada uma leitura completa, onde trabalhos com embasamento teórico consistente, qualidade metodológica, resultados claros, fator de impacto relevante e/ou autores reconhecidos, foram considerados. Este processo resultou em apenas 10 artigos selecionados.

Com o intuito de enriquecer a nossa revisão bibliográfica, iniciamos uma segunda etapa, onde consultamos o referencial bibliográfico dos trabalhos selecionados. Nesta etapa, identificamos que Shoham e Tennenholtz (1992) são os pioneiros no emprego das normas em Sistemas Multiagente. Considerando este fato, aplicamos uma nova revisão bibliográfica sistemática mais abrangente, dentre o período de 1992 a 2023, a fim de identificar novos trabalhos relevantes a esta tese. Ao repetirmos os passos anteriores, selecionamos, ao total, 23 artigos que cobrem as diferentes modalidades de síntese apresentadas na seção anterior.

Seguindo a classificação ilustrada pela figura 3, nós agrupamos os trabalhos

selecionados entre as abordagens off-line (seção 2.3.1) e on-line (seção 2.3.2) para então agrupar os trabalhos on-line entre as abordagens por emergência (seção 2.3.2.1) e observação (seção 2.3.2.2). Nós finalizamos esta seção apresentando uma discussão sobre os trabalhos correlatos frente a proposta desta tese (seção 2.3.3).

2.3.1 Síntese Off-line

Shoham e Tennenholtz (1992) apresentam a primeira abordagem para a síntese de normas em SMAn. Em sua proposta, os autores estão interessados em sintetizar normas que evitam conflitos e viabilizam aos agentes alcançarem os seus objetivos individuais sem interferir na execução dos planos uns dos outros. Neste sentido, os autores discutem sobre o emprego de normas úteis para regular o comportamento dos agentes. Os autores também demonstram que sintetizar normas úteis é um problema NP-Completo. Posteriormente, em (SHOHAM; TENNENHOLTZ, 1995), os autores aprofundam a discussão sobre a síntese automática das normas úteis.

Fitoussi e Tennenholtz (1998) introduzem o conceito de *minimalidade* para a síntese de normas menos restritivas. Neste contexto, dentre um conjunto de normas úteis, a norma *mínima* é aquela que impõe a menor quantidade de restrições sobre o comportamento dos agentes. Desta forma, a norma mínima maximiza a autonomia dos agentes para atingirem os seus objetivos individuais. Os autores também apresentam um método para sintetizar normas mínimas e demonstram que este é um problema NP-Hard. Posteriormente, em (FITOUSSI; TENNENHOLTZ, 2000), os autores introduzem o conceito de *simplicidade* para a síntese de normas *simples* de serem cumpridas. Os autores alegam que seguir certas normas pode requerer diferentes capacidades sensoriais e falhas ou a falta de sensores apropriados podem levar certos agentes a infringirem as normas. Neste sentido, as normas simples são menos dependentes das capacidades sensoriais dos agentes, o que torna possível que elas sejam seguidas por uma parcela maior de agentes.

Hoek et al. (2007) propõem tratar a síntese de normas como um problema de verificação de modelos em ATL (Alternating-time Temporal Logic)⁵. Para isto, os autores definem um SMAn como um sistema de transições alternadas (estrutura semântica que sustenta o ATL) cujos estados e transições representam, respectivamente, estados do sistema e ações dos agentes. Por outro lado, uma norma é definida por um propósito, que representa o que se pretende alcançar com a aplicação da norma, e restrições comportamentais, que mapeiam ações a estados onde as ações são proibidas. Assim, empregar uma norma sobre um sistema de transições alternadas resulta em um novo sistema onde as transições proibidas pela norma são descartadas. Neste contexto, uma norma é *eficaz* se assegura que o seu propósito pode ser satisfeito no sistema

⁵ ATL é uma generalização da lógica CTL para o raciocínio e a representação de alianças em SMA (ALUR; HENZINGER; KUPFERMAN, 2002).

reduzido. Dado que os propósitos das normas são descritos como fórmulas em ATL, a síntese de normas eficazes consiste em averiguar através de uso de verificadores de modelos se o propósito das normas ocorre no sistema de transições alternadas (i.e., se as fórmulas que o descrevem são verdadeiras no sistema). Adicionalmente, os autores demonstram que a síntese de normas eficazes em ATL é um problema NP-Hard.

Christelis e Rovatsos (2009) propõem um método para a síntese de normas eficazes para evitar conflitos considerando planejamento proposicional geral (NEBEL, 2000). Nesta modalidade de planejamento, a busca por planos ocorre em espaços de estados representados através de especificações (i.e., descrições incompletas do mundo). Assim, uma especificação representa diferentes estados do sistema e a busca por planos ocorre em espaços de estados menores. A partir da especificação de um estado de conflito, o método determina o conjunto de todas as sequências de especificações e ações que alcançam e partem de tal estado, tal que, a primeira e a última especificação de uma sequência é livre de conflito. Então, para cada sequência, uma norma é sintetizada, onde a especificação inicial constitui a precondição da norma e a ação que parte da especificação constitui a ação proibida pela norma. Por fim, para cada sequência, é verificado se o SMAn pode transitar da especificação inicial para a final através de um caminho alternativo sem conflito considerando as restrições impostas pelas normas sintetizadas. Se for possível, as normas são consideradas eficazes para regular os agentes. Posteriormente, alegando que esta abordagem leva a uma solução verbosa constituída por múltiplas normas específicas, em (CHRISTELIS, G. D.; ROVATSOS; PETRICK, 2010), os autores estendem o seu primeiro trabalho para suportar lógica de primeira ordem, o que beneficia a síntese para a obtenção de normas mais expressivas e gerais.

Boela et al. (2012) propõem um framework neuro-simbólico para a síntese, raciocínio e aprendizado de normas em SMAn. Para isto, os agentes são equipados com uma base de conhecimento que descreve um conjunto de normas em lógica de entrada e saída (MAKINSON; TORRE, 2000) e uma rede neural lógica (GARCEZ, A. S. d.; BRODA; GABBAY, D. M., 2002) que incorpora as normas em sua estrutura. Através de uma extensão do algoritmo CLIP (GARCEZ, A.; ZAVERUCHA, 1999), os agentes constroem uma rede neural considerando as suas bases de conhecimento. As redes neurais são, então, usadas para processar os dados recebidos do ambiente em que os agentes atuam e as saídas das redes denotam as ações que os agentes são obrigados ou permitidos realizar⁶. As redes neurais são utilizadas também para aprimorar a base de conhecimento dos agentes, onde novas normas são adicionadas às bases conforme verifica-se a necessidade de regular comportamentos para os quais não existem normas apropriadas. As bases aprimoradas são utilizadas para construir

⁶ Os autores simulam um agente recebendo entradas hipotéticas do ambiente em seus experimentos. Por ser uma simulação, consideramos que a síntese de normas ocorre em modo off-line.

novas redes neurais que, por sua vez, são melhores para auxiliar os agentes a se comportarem de forma coerente no ambiente.

Bulling e Dastani (2016) propõem um framework baseado em teoria dos jogos para sintetizar normas e determinar se comportamentos específicos desejáveis pelo projetista podem ser impostos pelas normas quando os agentes seguem as suas preferências subjetivas. Para isto, o projetista deve definir, primeiramente, um conjunto de normas para evitar conflitos, modelar em LTL (*Linear Temporal Logic*) possíveis sequências de execução do SMAN que consideram as restrições impostas pelas normas para, então, identificar uma sequência onde o sistema alcança o *equilíbrio de Nash*⁷. Neste sentido, as normas que guiam o sistema a tal estado são consideradas eficazes para regular os agentes.

Morales et al. (2017) apresentam um framework para a síntese de normas baseada na teoria dos jogos evolutivos (SMITH, 1982). Para isto, os autores consideram simular um SMAN onde as situações de conflito ao serem detectadas são modeladas como jogos nos quais os agentes competem pelas normas que os regularão. Assim, para cada situação de conflito, normas são sintetizadas e delegadas aos agentes participantes para resolverem o jogo correspondente. Durante a simulação do SMAN, os jogos ocorrem múltiplas vezes e os agentes podem adotar novas normas como as suas escolhas para regularem a sociedade baseado-se em percepções sobre o quão eficazes as normas são em evitar conflitos. Desta forma, normas consideradas apropriadas pela maioria dos agentes em cada jogo prosperam e, em conjunto, elas são capazes de regular as diferentes situações de conflito emuladas. Os autores argumentam que ao simularem a evolução das estratégias em um SMAN, eles podem antecipar as normas que levam o sistema a uma regulação bem-sucedida.

Alechina et al. (2022) propõem um framework formal para a síntese de normas dinâmicas, isto é, normas que especificam estados do sistema que são ilegais de serem alcançados. Para isso, as normas são definidas como máquinas de Mealy que contêm pares de estados e agentes ilegais. Nesse sentido, a síntese é tratada como um problema em ATL* (LAROUSSINIE; MARKEY, 2015), onde busca-se identificar uma norma que quando implementada no sistema assegura que um dado propósito

⁷ Em um jogo, o equilíbrio de Nash simboliza uma situação na qual nenhum jogador pode ganhar se mudar sua estratégia unilateralmente. Isto significa que, se cada participante escolhe a sua melhor estratégia dentro de um jogo, então, entende-se que os mesmos jogadores não têm como melhorar suas estratégias sozinhos (pois já escolheram as melhores e os adversários também). Com isto, a estabilidade entre as estratégias escolhidas pelos jogadores é o que define um Equilíbrio de Nash. Apesar de os participantes não cooperarem, é possível que a busca individual da melhor solução conduza o jogo a um resultado em que se verifique estabilidade, não havendo incentivo para que nenhum deles altere o seu comportamento. Esta possibilidade deriva das previsões que os jogadores possam fazer relativamente ao comportamento do seu oponente. Assim, o equilíbrio de Nash corresponde a uma combinação de estratégias em que a estratégia de cada jogador é a melhor possível para ele próprio, tendo em consideração a estratégia escolhida pelo outro jogador. Neste contexto, o resultado do jogo agrada a ambos os jogadores, pelo que nenhum deles tem incentivo para alterar a estratégia que seguiu.

(análogo ao propósito definido em (2007)) é satisfeito. Os autores também demonstram que a síntese de normas dinâmicas é $(k + 1)$ -EXPTIME, onde k é a profundidade de alternância dos quantificadores na especificação ATL*.

Dell'Ana et al. (DELL'ANNA *et al.*, 2022) propõem um framework formal para a síntese de normas considerando traços de comportamentos dos agentes. Os traços são particionados em dois conjuntos, os bons, que levam os agentes a satisfazer os objetivos do sistema, e os ruins, que não levam os agentes a satisfazer os objetivos do sistema. Neste contexto, os autores estão interessados na síntese de normas que são obedecidas em todos os traços bons e violadas em todos os traços ruins. Os autores também consideram o problema de revisar normas que consiste em realizar atualizações nas especificações das normas. Os autores alegam que este problema é relevante quando uma norma é aceitável, porém, não auxilia no alcance dos objetivos do sistema. Assim, uma modificação mínima na especificação da norma pode fazer com que ela passe a auxiliar no alcance dos objetivos do sistema. Os autores demonstram que a revisão das normas é um problema NP-completo.

Considerando a importância do desenvolvimento de SMA nos quais os agentes se comportem de forma ética, Nieves e Sierra (2022) propõem um modelo para a síntese de normas baseada na promoção de valores. Os autores alegam que as normas são mecanismos com o potencial de garantir que valores morais sejam respeitados no sistema. Para isso, os valores morais são definidos como funções que avaliam os estados do sistema em termos do grau de conformidade com os valores em questão. Dado que as normas restringem as histórias que podem ocorrer (sequências de estados e ações), a análise de tais histórias permite calcular o alinhamento das normas aos valores a serem incorporados no sistema. Nesse sentido, o objetivo é sintetizar um conjunto de normas que maximizam o alcance de estados finais que sejam compatíveis com os valores em questão o máximo possível. Uma vez estabelecido o alinhamento das normas com os valores em termos absolutos, é possível comparar diferentes conjuntos de normas entre si. Para tanto, os autores definem o alinhamento relativo entre dois conjuntos de normas e, para encontrar o conjunto de normas ótimo, os autores propõem o emprego de métodos de otimização como os algoritmos genéticos (KATOCH; CHAUHAN; KUMAR, 2021).

2.3.2 Síntese On-line

2.3.2.1 Emergência

Walker e Wooldridge (1995) propõem um conjunto de estratégias para auxiliar os agentes à convergirem a um acordo global sobre uma norma para regulá-los. A cada iteração, os agentes, que usam apenas informações locais (estado interno), decidem, segundo uma estratégia, qual norma adotar. Quando os agentes adotam a mesma

norma, então tal norma passa a ser a norma da sociedade dos agentes. Os autores realizam uma série de experimentos a fim de averiguar qual estratégia auxilia os agentes a convergirem a uma norma mais rapidamente.

Epstein (2001) emprega um mecanismo de imitação para a emergência de normas em redes em anel. Cada agente ocupa uma posição fixa no anel e é um objeto caracterizado por dois atributos, uma norma e um raio de amostragem. A norma e o raio são definidos aleatoriamente para cada agente. Então, os agentes atualizam as suas normas com base nas observações que realizam sobre os demais agentes. Basicamente, um agente adota como norma, a norma com maior frequência dentro do seu raio. Por sua vez, o raio de um agente é incrementado quando a frequência da norma adotada é diferente em um raio maior ou decrementado quando a frequência de tal norma é igual em um raio menor. Os agentes encerram a atualização das suas normas quando os seus raios decrescem para zero. Neste momento, cada agente possui uma norma para seguir.

Boela e Van der Torre (2007) empregam um modelo de negociação para a emergência de normas. O modelo, denominado de ciclo de delegação social, estabelece que um conjunto de agentes representativos realiza propostas de objetivos a serem alcançados e obrigações a serem cumpridas. Para isto, o modelo faz uso de um protocolo de negociação que estabelece as regras para propor um acordo (i.e., quais são os objetivos e obrigações dos agentes), aceitar ou rejeitar propostas e quebrar negociações. O processo de negociação, por sua vez, pode terminar sucesso, fracasso ou sem resultado. Se um agente propõe um acordo e os outros agentes o aceitam, então a negociação termina com sucesso e uma convenção se estabelece. Se os agentes não chegam a nenhum acordo, a negociação termina em fracasso. Por fim, se um agente decidir interromper a negociação, ela termina sem nenhum resultado. Para os dois últimos casos, nenhuma norma emerge na sociedade.

Savarimuthu et al. (2008) analisam o efeito das recomendações na emergência de normas em redes aleatórias e livres de escala. Para isto, os agentes seguem o *jogo do ultimato* (PAVLOVIĆ; ĐURAKOVIĆ, 2021), onde certos agentes possuem o papel de líder e outros possuem o papel de seguidores. No jogo, os agentes devem entrar em consenso sobre como compartilhar um dado valor e para isto necessitam de uma norma que estabeleça a regra de partilha. Neste contexto, os líderes propagam na rede as suas recomendações de normas, ao passo que os seguidores determinam se aceitam as recomendações. À medida que a recomendação de um líder passa a ser aceita, a sua influência dentro da rede aumenta fazendo com que as suas recomendações sejam mais facilmente aceitas. Ao longo das interações, emergi como norma a recomendação do líder mais influente. Posteriormente, em (SAVARIMUTHU; CRANFIELD, 2009), os autores estendem o mecanismo para suportar a emergência de normas onde as redes sociais são dinâmicas.

2.3.2.2 Observação

Morales et al. (2013) propõem um mecanismo para a síntese de normas através de raciocínio baseado em casos. O mecanismo, o qual é denominado de Intelligent Robust On-line Norm Synthesis Machine (IRON), monitora o comportamento dos agentes a fim de detectar conflitos para então sintetizar normas que sejam capazes de evitá-los em situações futuras. Para isto, o Intelligent Robust On-line Norm Synthesis Machine (IRON) busca em uma base de conhecimento casos similares ao contexto no qual os conflitos ocorrem tomando-os como base a criação de novas normas. As normas são adicionadas a uma rede, onde são organizadas hierarquicamente de acordo com a sua generalidade (normas mais gerais apontam para normas específicas através de uma verificação sintática). Durante a execução do Sistema Multiagente Normativo (SMAN), os agentes são regulados pelas normas ativas na rede. A rede passa por operações de refinamento que consistem em avaliar a *necessidade* e a *eficácia* das normas em evitar conflitos. Normas cuja necessidade e eficácia são menores do que um certo limiar são desconsideradas para regular os agentes e desativadas da rede. Quando não se observa a ocorrência de conflitos por um longo período de tempo, as normas ativas na rede são consideradas necessárias e eficazes para regular os agentes.

Posteriormente, Morales et al. (2014) apresentam o Simple Minimal On-line Norm Synthesis (SIMON), um mecanismo para a síntese de conjuntos de normas compactos. O SIMON estende as funcionalidades do IRON incorporando uma técnica alternativa para a generalização das normas. O SIMON faz uso de uma taxonomia de termos que estrutura o conhecimento sobre o domínio de aplicação como uma árvore para calcular a generalização mais específica dos termos que compõem um conjunto de normas generalizáveis. Ao detectar que duas ou mais normas na rede são generalizáveis, necessárias e eficazes em evitar conflitos, o SIMON cria a norma mais específica que as generaliza. Diferentemente do IRON que sintetiza todas as normas mais específicas para então sintetizar uma norma mais geral, o SIMON é capaz de obter redes de normas menores. Conseqüentemente, o conjunto de normas ativas que regulam apropriadamente os agentes tende também a ser menor.

Posteriormente, Morales et al. (2015) apresentam o Liberal On-line Norm synthesis (LION), um mecanismo que estende as funcionalidades do IRON e do SIMON para construir conjuntos de normas compactos e liberais. Para isto, LION explora a *substitutibilidade* e *complementaridade* das normas, dois conceitos advindos da economia (MCKENZIE, 1977). Enquanto duas normas são substituíveis se satisfazem os mesmos requisitos regulatórios e, portanto, podem ser usadas para substituir uma à outra, duas normas são complementares caso sejam melhores quando usadas em conjunto para regular os agentes do que separadamente. Neste contexto, o LION estabelece as relações de substitutibilidade e complementaridade entre todas as normas em sua rede obtendo diferentes conjuntos de normas que regulam os agentes apropria-

damente. Considerando tais conjuntos, o LION pode determinar o conjunto de normas mais compacto e com normas liberais (normas mínimas).

Kadir et al. (2020) propõem um método para a síntese de normas baseada em aprendizagem com reforço (SUTTON; BARTO, 2018). Para isto, o método faz uso do algoritmo *Q-learning* (WATKINS; DAYAN, 1992) para que os agentes estimem pares de estado-ação proibitivos e obrigatórios. Posteriormente, os resultados do Q-learning são extraídos, abstraídos e generalizados para reduzir o número de regras extraídas pelos agentes e, assim, construir um conjunto de normas mais compacto. Para resolver este problema, o método de síntese faz uso de uma abordagem baseada em Conhecimento Hierárquico (APELDOORN; KERN-ISBERNER, 2017) e Deontica Dinâmica (FRANTZ, C. et al., 2014)⁸.

Riad e Golpayegani (2022) propõem um mecanismo para a síntese de normas que estende as funcionalidades do IRON (MORALES et al., 2013). O mecanismo detecta conflitos durante a execução do SMAn, cria normas para evitá-los futuramente através de raciocínio baseado em casos e organiza as normas em uma rede de generalização e especialização. Todavia, o mecanismo calcula a utilidade ganha pelo sistema quando os agentes optam por estar em conformidade com as normas sintetizadas. Para isto, faz-se uso de uma função de utilidade que adiciona os objetivos do sistema com uma função de maximização e subtrai os objetivos do sistema com uma função de minimização. Assim, para cada situação que pode levar a ocorrência de conflitos, o mecanismo determina a norma da rede com a maior utilidade para regular os agentes. Como a utilidade das normas pode variar ao longo da execução do sistema, a norma considerada mais apropriada para regular os agentes pode divergir entre as situações que necessitam de regulação.

2.3.3 Análise dos Trabalhos Correlatos

A tabela 1 fornece uma visão geral das contribuições sobre síntese de normas em SMAn apresentadas na seção 2.3 e as classifica com base nas diferentes dimensões discutidas na seção 2.2.1. Logo, a tabela 1 classifica os trabalhos quanto⁹: ao **momento** da síntese (*off-line* ou *on-line*); a **operação** da síntese (*manual* ou *automática*); o **escopo** da síntese (*fixo* ou *variável*); a **ordenação** do espaço das normas (*ordenado* ou *desordenado*); a exploração da **generalidade** das normas (*considera* ou *desconsidera*); a exploração de **conflito** (*considera* ou *desconsidera*); a exploração dos **objetivos** do sistema (*considera* ou *desconsidera*); a **propriedade** das normas sintetizadas (*útil, mínima, eficaz, necessária* e/ou *eficiente*). Além do mais, usamos o símbolo “-” (traço) para indicar que um dado trabalho não fornece informações suficientes para que o respectivo item seja devidamente preenchido.

⁸ Tradução livre para *Dynamic Deontic*.

⁹ Destacamos em negrito as palavras usadas para formar os nomes das colunas da tabela 1.

Para fins de comparação, ao final da tabela 1, adicionamos o nosso trabalho, que está representado pelo símbolo ✱.

Trabalho	Momento	Operação	Escopo	Espaço	Generalidade	Conflito	Objetivos	Propriedade
Shoham e Tennenholz (1992)	off-line	automática	fixo	desordenado	considerada	considerado	desconsiderado	útil
Shoham e Tennenholz (1995)	off-line	automática	fixo	desordenado	considerada	considerado	desconsiderado	útil
Fitoussi e Tennenholz (1998)	off-line	automática	fixo	desordenado	desconsiderada	considerado	desconsiderado	mínima
Fitoussi e Tennenholz (2000)	off-line	automática	fixo	desordenado	desconsidera	considera	desconsidera	mínima ou simples
Hoek et al. (2007)	off-line	automática	fixo	desordenado	considera	considera	desconsidera	eficaz
Christelis e Rovatsos (2009)	off-line	automática	fixo	desordenado	considera	considera	desconsidera	eficaz
Christelis et al. (2010)	off-line	automática	fixo	desordenado	considera	considera	desconsidera	eficaz
Boella et al. (2012)	off-line	automática	fixo	desordenado	considera	considera	desconsidera	eficaz
Bulling e Dastani (2016)	off-line	manual	fixo	desordenado	considera	considera	desconsidera	eficaz
Morales et al. (2017)	off-line	automática	fixo	desordenado	considera	considera	desconsidera	eficaz
Dell'Anna et al. (2022)	off-line	automática	fixo	desordenado	considera	desconsidera	desconsidera	-
Alechina et al. (2022)	off-line	automática	fixo	desordenado	desconsidera	desconsidera	desconsidera	-
Nieves e Sierra (2022)	off-line	automática	fixo	desordenado	desconsidera	desconsidera	desconsidera	eficaz
Walker e Wooldridge (1995)	on-line	automática	fixo	desordenado	desconsidera	desconsidera	desconsidera	-
Epstein (2001)	on-line	automática	fixo	desordenado	desconsidera	desconsidera	desconsidera	-
Boella e van der Torre (2007)	on-line	automática	fixo	desordenado	desconsidera	desconsidera	desconsidera	-
Savarimuthu et al. (2008)	on-line	automática	fixo	desordenado	desconsidera	desconsidera	desconsidera	-
Savarimuthu e Cranefield (2009)	on-line	automática	fixo	desordenado	desconsidera	desconsidera	desconsidera	-
Morales et al. (2013)	on-line	automática	fixo	desordenado	considera	considera	desconsidera	necessária e eficaz
Morales et al. (2014)	on-line	automática	fixo	desordenado	considera	considera	desconsidera	necessária e eficaz
Morales et al. (2015)	on-line	automática	fixo	desordenado	considera	considera	desconsidera	necessária, eficaz e mínima
Kadir et al. (2020)	on-line	automática	fixo	desordenado	desconsidera	considera	desconsidera	útil
Riad e Golpayegani (2022)	on-line	automática	fixo	desordenado	desconsidera	considera	considera	eficiente
*	of-line	automática	variável	ordenado	considera	considera	considera	eficiente

Tabela 1 – Comparativo entre os trabalhos correlatos.

Com base nesta tabela, concluímos que:

1. A maioria dos trabalhos não considera os objetivos do sistema. Com exceção de Dell'Anna et al. (2022) e Riad e Golpayegani (2022), os trabalhos ignoram alcançabilidade dos objetivos do sistema na síntese de normas. Contudo, para que os objetivos do sistema sejam de fato alcançáveis, é necessário que os agentes possam executar, pelo menos, uma sequência de ações que satisfaça os objetivos do sistema. Portanto, integrar a análise da alcançabilidade dos objetivos do sistema à síntese de normas é fundamental para que as normas sintetizadas garantam que os agentes possam alcançar os objetivos do sistema. No trabalho de Riad e Golpayegani (2022), os objetivos do sistema necessitam ser quantificáveis para que a função de utilidade faça sentido. Logo, para diferentes domínios, o trabalho de Riad e Golpayegani (2022) não pode ser aplicado ou não é apropriado para a síntese de normas. Por outro lado, no trabalho de Dell'Anna et al. (2022), diferentes sequências de ações que levam a satisfação dos objetivos do sistema necessitam ser obtidas para que as normas sejam sintetizadas. Contudo, é possível que não exista uma norma que auxilie os agentes a satisfazerem os objetivos do sistema através de todas as sequências selecionadas. Assim, quanto mais sequências são consideradas, mais difícil é sintetizar normas apropriadas para regular os agentes. Além do mais, obter as diferentes sequências pode ter um custo alto tornando a síntese um processo ainda mais complexo.

2. A maioria dos trabalhos não considera a generalidade das normas. A generalidade das normas é uma característica pouco explorada na síntese de normas. Ela permite analisar a extensão de comportamentos que as normas regulam e entender quais generalidades conferem às normas um equilíbrio entre o controle que exercem e a autonomia dos agentes. Logo, explorar a generalidade das normas é fundamental para que normas eficientes sejam devidamente sintetizadas. Dos trabalhos analisados, somente os trabalhos de Fitoussi e Tennenholtz (1998; 2000) e Morales et al. (2013; 2014; 2015) exploram a generalidade das normas. Nos trabalhos de Fitoussi e Tennenholtz (1998; 2000), tem-se interesse em sintetizar normas menos restritivas, logo, considerando um conjunto de normas apropriadas para regular os agentes, há necessidade de realizar uma comparação entre todas as normas a fim de determinar a norma mais flexível (mínima). Por outro lado, nos trabalhos de Morales et al. (2013; 2014; 2015), calcula-se a relação de generalidade entre todos os pares de normas sintetizadas com o objetivo de construir uma rede hierárquica de normas. Como o número de normas sintetizadas tende a crescer à medida que o SMA executa, determinar a relação de generalidade entre as normas tende a ser um processo custoso.

3. Os trabalhos não consideram um escopo variável para a síntese. Conforme dito anteriormente, o espaço das normas é um conjunto potencialmente grande, logo, consideramos que medidas para torná-lo menor são fundamentais para diminuir

o tempo pela busca por normas eficientes. Dentre os trabalhos analisados, nenhum trabalho emprega uma técnica para tornar o espaço das normas menor.

4. Os trabalhos não consideram ordenar o espaço das normas. Um espaço ordenado possibilita a síntese de normas ser tratada como um problema de busca. Desta forma, podemos aplicar diferentes estratégias de visita que exploram o espaço de busca de diferentes formas priorizando regiões onde as normas eficientes são mais prováveis de serem encontradas. Seguindo a hipótese 1, caso ela se confirme, podemos economizar tempo e esforço computacional ao iniciarmos a busca pelo meio da ordenação podendo, então, parte do espaço de busca. Dentre os trabalhos analisados, nenhum trabalho emprega alguma técnica para organizar o espaço das normas.

5. A maioria dos trabalhos não considera sintetizar normas eficientes. Conforme dito anteriormente, uma norma é eficiente caso evite conflitos ao passo que mantém os objetivos do sistema alcançáveis. Embora os trabalhos de van der Hoek et al. (2007), Bulling e Dastani (2016), Morales et al. (2013; 2014; 2015), Kadir et al. (2020) e Riad e Golpayegani (2022) estejam interessados em sintetizar normas para evitar conflitos, a síntese não considera nenhum tipo de objetivo para criar normas apropriadas para regular os agentes. Os trabalhos de Shoham e Tennenholtz (1992; 1995), Fitoussi e Tennenholtz (1998; 2000), Christelis e Rovatsos (2009; 2010) e Morales et al. (2017) estão interessados na síntese de normas que evitam conflitos, contudo, a síntese considera criar normas que viabilizem aos agentes satisfazerem os seus objetivos individuais. Por outro lado, o trabalho de Riad e Golpayegani (2022) é o único que considera tanto evitar conflitos quanto manter os objetivos do sistema alcançáveis na síntese de normas. Desta forma, consideramos que o trabalho de Riad e Golpayegani (2022) é o único a sintetizar normas eficientes. Contudo, dentro do contexto do respectivo trabalho, busca-se sintetizar a norma que maximiza a utilidade para o alcance dos objetivos do sistema. Conforme dito no item 1, para isto, os objetivos do sistema necessitam ser mensuráveis, logo, a síntese que os autores propõem se aplica a uma parcela restrita de domínios.

6. A maioria dos trabalhos considera a ocorrência de conflitos. Com exceção Walker e Wooldridge (1995), Epstein (2001), Boella e van der Torre (2007), Savarimuthu et al. (2008), Savarimuthu e Cranefield (2009) e Dell'Anna et al. (2022), os trabalhos consideram a ocorrência de conflitos na síntese de normas. Isto mostra que a comunidade de SMAn tem foco na síntese de normas com o objetivo de evitar conflitos na execução do sistema.

2.4 CONCLUSÃO

De acordo com o exposto neste capítulo, conclui-se que o campo da síntese de normas carece de mecanismos capazes de sintetizar normas eficientes. Para este fim,

apenas o trabalho de Riad e Golpayegani (2022) considera explorar os conflitos e os objetivos do sistema. Embora vários trabalhos considerem os conflitos na síntese de normas, julgamos que evitá-los não é suficiente para o bom funcionamento dos SMAn. Nesse sentido, além de evitá-los, as normas devem levar o sistema para alguma direção a partir da qual os agentes possam alcançar os objetivos do sistema.

Observamos também que os mecanismos correntes não estão aptos a explorar em conjunto: i) os objetivos do sistema; ii) os conflitos causados pelas ações dos agentes; iii) a generalidade das normas; iv) a ordenação do espaço das normas; e v) a redução do escopo da síntese. Neste contexto, nós focamos no restante desta tese no desenvolvimento de uma proposta capaz de suprir estas carências. Para isto, no próximo capítulo, nós iniciamos a construção do nosso ferramental teórico e formal apresentando os conceitos e definições fundamentais para a síntese de normas eficientes através de uma abordagem simbólica. Isto nos permite caracterizar uma vasta gama de Sistemas Multiagente viabilizando a síntese de normas eficientes em diferentes domínios.

Ao final do desenvolvimento da nossa proposta, nós apresentamos um conjunto de algoritmos capazes de sintetizar normas eficientes. Como parte do processo de síntese, tais algoritmos simulam a execução dos SMAn para atestar a capacidade das normas em evitar conflitos e manter os objetivos dos sistemas alcançáveis. Por este motivo, na tabela 1, classificamos o nosso trabalho como *off-line* e *automático*. Além do mais, ao longo do texto, nós também buscamos responder às perguntas de pesquisa levantadas no capítulo 1.

3 ALCANÇABILIDADE DOS OBJETIVOS DO SISTEMA

Conforme introduzido no capítulo 1, a análise da alcançabilidade dos objetivos do sistema refere-se ao problema de determinar a existência de uma sequência de ações que, ao ser executada pelos agentes, satisfaz os objetivos do sistema. No contexto deste trabalho, este problema é de fundamental importância para que possamos determinar como considerar os objetivos do sistema na síntese de normas eficientes. Para isto, nós construímos um ferramental teórico e formal que nos permite caracterizar e representar os conceitos e definições relacionadas à síntese de normas eficientes. Este capítulo apresenta o nosso arcabouço conceitual para a alcançabilidade dos objetivos do sistema na síntese de normas.

Este capítulo está organizado da seguinte forma. Na seção 3.1 apresentamos os conceitos básicos do nosso ferramental. Na seção 3.2 nós avançamos na construção do ferramental apresentando conceitos relacionados à capacidade das normas em regular os agentes apropriadamente para, então, definirmos formalmente as normas eficientes. Por fim, na seção 3.3, nós apresentamos os conceitos relacionados à alcançabilidade dos objetivos do sistema correlacionando-os com a síntese de normas eficientes. Além do mais, nós recorremos ao cenário do tráfego de carros, apresentado no capítulo 1 para exemplificar os conceitos apresentados ao longo deste capítulo.

3.1 DEFINIÇÕES BÁSICAS

O ferramental teórico e formal para a síntese de normas eficientes apresentado neste trabalho tem como base a seguinte linguagem de primeira ordem restrita.

Definição 1 (Linguagem Restrita) *A linguagem lógica $\mathcal{L} = \langle \text{Pred}, \text{Var}, \text{Const}, \text{Conec} \rangle$ é uma linguagem de primeira ordem restrita, onde $\text{Pred} = \{\text{pred}_1, \dots, \text{pred}_j\}$ é um conjunto de símbolos de predicados, $\text{Var} = \{v_1, \dots, v_j\}$ é um conjunto de símbolos de variáveis, $\text{Const} = \{c_1, \dots, c_k\}$ é um conjunto de símbolos de constantes e $\text{Conec} = \{\wedge\}$ é um conjunto de conectivos lógicos.*

Uma fórmula atômica p é um símbolo de predicado associado a um conjunto finito de termos. Um termo, por sua vez, é uma constante ou variável. Assumimos que termos descritos por símbolos distintos representam objetos distintos, símbolos de variáveis iniciam com letras maiúsculas e símbolos de constantes iniciam com letras minúsculas. Por sua vez, uma fórmula atômica *ground* é uma fórmula atômica cujos termos são constantes enquanto uma fórmula atômica *unground* é uma fórmula atômica que tem, pelo menos, uma variável. Por conveniência, chamamos as fórmulas atômicas cujos termos são *exclusivamente* variáveis de fórmulas atômicas *estritamente unground* e são representadas pelo conjunto $\mathcal{L}^{uaf} \subset \mathcal{L}'$, onde \mathcal{L}' é o conjunto das fórmulas lógicas bem formadas de \mathcal{L} obtidas através das regras de produção da

gramática da lógica de primeira ordem (RUSSELL; NORVIG, 2010) considerando os conectivos em Conec. Por outro lado, representamos através do conjunto $\mathcal{L}^{gaf} \subset \mathcal{L}'$ o conjunto das fórmulas atômicas ground.

Exemplo 1 Considere que $\{next, at\} \subset Pred$, $\{L_1, L_2\} \subset Var$ e $\{ag_1, a\} \subset Const$. Através das regras de produção da gramática de \mathcal{L} , podemos obter uma fórmula estritamente unground $next(L_1, L_2) \in \mathcal{L}^{uaf}$ para denotar que um local arbitrário L_1 é adjacente a outro lugar arbitrário L_2 . Além do mais, podemos obter uma fórmula atômica ground $at(ag_1, a) \in \mathcal{L}^{gaf}$ para denotar que um agente ag_1 está no local a .

Definição 2 (Estado do Sistema) Um estado do sistema é um conjunto $s = \{p_1, \dots, p_i\}$ de fórmulas atômicas ground interpretados como uma conjunção, tal que, $s \subseteq \mathcal{L}^{gaf}$. Além do mais, $S = \{s_1, \dots, s_j\}$ é o conjunto dos estados do sistema, tal que, $S \subseteq 2^{\mathcal{L}^{gaf}}$, enquanto $S_0 \subseteq S$ é o conjunto de todos os possíveis estados iniciais do sistema.

Exemplo 2 Considere a figura 1 da seção 1.1 que ilustra a situação inicial dos agentes na grid. Tal situação pode ser descrita como um estado $s = \{at(ag_1, a), at(ag_2, b), at(ag_3, f), next(a, b), next(a, d), \dots, next(h, i), next(f, i)\}^1$, com $s \in S$.

Consideramos a hipótese do mundo fechado (PRATT, 1994) para representar as informações sobre os estados do sistema. Desta forma, uma fórmula atômica ausente em um estado é considerada *falsa*.

Definição 3 (Ação) Uma ação é uma tupla $a = \langle Nome, Par, Pre, Add, Del \rangle$, tal que, Nome é um identificador unívoco, $Par \subseteq Var$ é um conjunto de variáveis como parâmetros da ação, $Pre \subseteq \mathcal{L}^{uaf}$ é um conjunto de pré-condições para a ação ser executada, $Add \subseteq \mathcal{L}^{uaf}$ é um conjunto de efeitos que passam a ser verdadeiros após a execução da ação e $Del \subseteq \mathcal{L}^{uaf}$ é um conjunto de efeitos que passam a ser falsos após a execução da ação, com Const e Var introduzidos na definição 1. Além disso, $\mathcal{A} = \{a_1, \dots, a_j\}$ é um conjunto das ações de um cenário de aplicação.

Exemplo 3 Considere que a ação “mover entre locais adjacentes da grid” seja descrita como $\langle move, \{Ag_1, L_1, L_2\}, \{at(Ag_1, L_1), next(L_1, L_2)\}, \{at(Ag_1, L_2)\}, \{at(Ag_1, L_1)\} \rangle$. De acordo com tal ação, para que um agente Ag_1 se mova de um local L_1 para um local L_2 , o agente deve estar em L_1 e L_1 deve ser adjacente a L_2 . Após a execução da ação, o agente Ag_1 deixa de estar no local L_1 e passa a estar no local L_2 .

Quando for necessário, a fim de distinguirmos as propriedades de diferentes elementos de um mesmo conceito, descreveremos os símbolos dos elementos sobrescritos aos símbolos das propriedades. Por exemplo, dado duas ações distintas a_i e a_j , representaremos as suas pré-condições como Pre^{a_i} e Pre^{a_j} , respectivamente.

¹ Por simplicidade, nos exemplos deste capítulo, omitimos da descrição dos estados algumas fórmulas atômicas que representam locais adjacentes na grid.

Definição 4 (Instância de Ação) *Dada uma ação $a \in \mathcal{A}$, uma instância a_i de a é uma representação específica de a , tal que, $\text{Par}^{a_i} \subseteq \text{Const}$ é um conjunto de constantes e $\text{Pre}^{a_i}, \text{Add}^{a_i}, \text{Del}^{a_i} \subseteq \mathcal{L}^{gaf}$ são conjuntos de fórmulas atômicas ground, onde $\text{Nome}^{a_i} = \text{Nome}^a$, com Const e \mathcal{A} introduzidos nas definições 1 e 3, respectivamente. Além do mais, $\mathcal{I}^{\mathcal{A}} = \{a_1, \dots, a_j\}$ é um conjunto das instâncias das ações de \mathcal{A} .*

Para cumprirem as suas tarefas, os agentes executam instâncias de ações de $\mathcal{I}^{\mathcal{A}}$. Assim, uma ação $a_i \in \mathcal{I}^{\mathcal{A}}$ é possível de ser executada em um estado $s \in \mathcal{S}$ se $\text{Pre}^{a_i} \subseteq s$. Se as pré-condições da ação estão satisfeitas no estado e a ação é realizada, então o sistema transita de s para um estado $s_j \in \mathcal{S}$, tal que, $s_j = (s - \text{Del}^{a_i}) \cup \text{Add}^{a_i}$.

Chamamos de *assinatura* de uma ação a concatenação entre o seu nome, um parêntese aberto, os seus parâmetros separados por vírgula e um parêntese fechado. Por exemplo, a assinatura da ação do exemplo 3 é $\text{move}(\text{Ag}_1, L_1, L_2)$. Nos próximos exemplos, nos referimos às ações e suas instâncias através das suas assinaturas.

Exemplo 4 *Considere o estado s introduzido no exemplo 2 e a instância $\text{move}(\text{ag}_1, a, d)$ da ação mover introduzida no exemplo 3. As pré-condições, efeitos verdadeiros e falsos de tal instâncias são, respectivamente, $\text{Pre} = \{\text{at}(\text{ag}_1, a), \text{next}(a, d)\}$, $\text{Add} = \{\text{at}(\text{ag}_1, d)\}$ e $\text{Del} = \{\text{at}(\text{ag}_1, a)\}$. Dado que $\text{Pre} \subseteq s$, se o agente ag_1 executar $\text{move}(\text{ag}_1, a, d)$ no estado s , então o sistema transita para o estado $s_j = \{\text{at}(\text{ag}_1, d), \text{at}(\text{ag}_2, b), \text{at}(\text{ag}_3, f), \text{next}(a, b), \text{next}(a, d), \dots, \text{next}(h, i), \text{next}(f, i)\}$, com $s_j \in \mathcal{S}$.*

Conforme discutido no capítulo 2, em SMAn empregam-se diferentes tipos de objetivos. Neste trabalho, os objetivos do sistema são definidos como *declarativos*, i.e., descrições de situações desejáveis (*state of affairs*) compartilhadas entre os agentes que devem alcançá-las através da execução de suas ações (RIEMSDIJK; DASTANI; WINIKOFF, 2008; HINDRIKS *et al.*, 2001).

Definição 5 (Objetivos do Sistema) *Os objetivos do sistema, denotado por $\mathcal{G} = \{p_1, \dots, p_j\}$, tal que, $\mathcal{G} \subseteq \mathcal{L}'$, são um conjunto de fórmulas atômicas interpretados como uma conjunção. Além disso, $\mathcal{G}^* = \{\mathcal{G}_1, \dots, \mathcal{G}_j\}$ é o conjunto de todos os objetivos possíveis do sistema.*

Definição 6 (Instância dos Objetivos do Sistema) *Dado um conjunto de objetivos do sistema $\mathcal{G} \in \mathcal{G}^*$, uma instância \mathcal{G}_j de \mathcal{G} é uma representação específica de \mathcal{G} , tal que, $\mathcal{G}_j \subseteq \mathcal{L}^{gaf}$ é um conjunto de fórmulas atômicas ground, com \mathcal{G} e \mathcal{G}^* introduzidos na definição 5. Além do mais, $\mathcal{I}^{\mathcal{G}} \subseteq 2^{\mathcal{L}^{gaf}}$ é o conjunto das instâncias de \mathcal{G} .*

Definição 7 (Estado de Objetivos) *Dado um conjunto de objetivos do sistema $\mathcal{G} \in \mathcal{G}^*$, um estado $s \in \mathcal{S}$ é dito ser um estado de objetivos de \mathcal{G} , denotado por $s \models \mathcal{G}$, se $\exists \mathcal{G}_j \in \mathcal{I}^{\mathcal{G}}: \mathcal{G}_j \subseteq s$, com \mathcal{S} , \mathcal{G} , \mathcal{G}^* e $\mathcal{I}^{\mathcal{G}}$ introduzidos nas definições 2, 5, 5 e 6,*

respectivamente. Além do mais, $S^{\mathcal{G}} = \{s \mid s \in \mathcal{S} \wedge s \models \mathcal{G}\}$ é o conjunto dos estados de objetivos do conjunto de objetivos do sistema \mathcal{G} .

Assim, dado um conjunto de objetivos do sistema $\mathcal{G} \in \mathcal{G}^*$, dizemos que os agentes satisfazem \mathcal{G} ao alcançarem um de seus estados de objetivos em $S^{\mathcal{G}}$.

Exemplo 5 Considere $\mathcal{G}^* = \{\mathcal{G}_1, \mathcal{G}_2\}$, com $\mathcal{G}_1 = \{at(ag_1, d), at(ag_2, b), at(ag_3, f)\}$ e $\mathcal{G}_2 = \{at(Ag_1, a)\}$, e os estados s e s_i introduzidos nos exemplos 2 e 4, respectivamente. Dado que $\mathcal{I}^{\mathcal{G}_1} = \{\mathcal{G}_1\}$ (\mathcal{G}_1 é uma instância de si próprio) e $\mathcal{I}^{\mathcal{G}_2} = \{\{at(ag_1, a)\}, \{at(ag_2, a)\}, \{at(ag_3, a)\}\}$, então $s \in S^{\mathcal{G}_1}$, pois a instância $\mathcal{G}_1 \subseteq s$. Por outro lado, s_i não é um estado de objetivos, pois não existem instâncias de \mathcal{G}_1 nem de \mathcal{G}_2 que estão contidas em s_i .

Um outro conceito relevante para esta tese é o *conflito*. Conforme discutido no capítulo 2, um conflito é geralmente visto como uma situação indesejável que pode gerar consequências negativas e, portanto, necessita ser evitado (EGGERT; FALZON, 2004). Embora nem todo conflito seja prejudicial, neste trabalho, consideramos que os conflitos são situações potencialmente negativas resultantes da execução das ações dos agentes que podem gerar implicações adversas para o correto e bom funcionamento dos SMAn.

Definição 8 (Conflito) Um conflito $c = \{p_1, \dots, p_i\}$, tal que, $c \subseteq \mathcal{L}'$, é um conjunto de fórmulas atômicas interpretados como uma conjunção. Além do mais, o conjunto $\mathcal{C} = \{c_1, \dots, c_j\}$, tal que, $\mathcal{C} \subseteq 2^{\mathcal{L}'}$, é o conjunto dos conflitos de um cenário de aplicação, com \mathcal{L} introduzida na definição 1.

Exemplo 6 Uma colisão entre dois agentes na grid pode ser descrita como $c = \{at(Ag_1, L_1), at(Ag_2, L_1)\}$. Portanto, uma colisão ocorre quando dois agentes distintos, Ag_1 e Ag_2 , estão sobre um mesmo local L_1 .

Definição 9 (Instância de Conflito) Dado um conflito $c \in \mathcal{C}$, uma instância c_i de c é uma representação específica de c , tal que, $c_i \subseteq \mathcal{L}^{gaf}$ é um conjunto de fórmulas atômicas ground, com \mathcal{C} introduzido na definição 8. Além do mais, $\mathcal{I}^c \subseteq 2^{\mathcal{L}^{gaf}}$ é o conjunto das instâncias de c .

Exemplo 7 Considere o conflito c introduzido no exemplo 6. Uma instância de c é $c_i = \{at(ag_1, a), at(ag_2, a)\}$ que denota que os agentes ag_1 e ag_2 colidem no local a da grid.

Definição 10 (Estado de Conflito) Dado um conflito $c \in \mathcal{C}$, um estado $s \in \mathcal{S}$ é dito ser um estado de conflito de c , denotado por $s \models c$, se $\exists c_i \in \mathcal{I}^c: c_i \subseteq s$, com \mathcal{S} , \mathcal{C} e \mathcal{I}^c introduzidos nas definições 2, 8 e 9, respectivamente. Além do mais, $S^c = \{s \mid s \in \mathcal{S} \wedge s \models c\}$ é o conjunto dos estados de conflito do conflito c .

Exemplo 8 Considere o estado s , o conflito c e a instância c_i de c introduzidos nos exemplos 2, 6 e 7, respectivamente. Considere também que o agente ag_2 executa sobre s a ação $move(ag_2, b, a)$, o que faz com que sistema transite de s para o estado $s_j = \{at(ag_1, a), at(ag_2, a), at(ag_3, f), next(a, b), next(a, d), \dots, next(h, i), next(f, i)\}$, com $s_j \in \mathcal{S}$. Como ambos os agentes ag_1 e ag_2 ocupam o local a da grid, temos, então, uma colisão entre tais agentes. Como $c_i \in \mathcal{I}^c$ e $c_i \subseteq s_j$, então $s_j \models c$.

Diferentes mecanismos foram propostos para auxiliar os agentes a resolverem conflitos (EKENBERG, 1997; RESMERITA; HEYMANN, 2003). Um mecanismo bem empregado para este fim são as normas (CRIADO; ARGENTE; BOTTI, Vicent, 2011). Embora existam diferentes tipos de normas em SMAN, neste trabalho, utilizamos as normas proibitivas (BOELLA; TORRE, 2004) como um mecanismo para evitar conflitos e manter os objetivos do sistema alcançáveis. No contexto deste trabalho, isto significa evitar estados de conflito e manter, pelo menos, um estado de objetivos alcançável.

Definição 11 (Norma Proibitiva) Uma norma proibitiva (ou proibição) $n = \varphi \xrightarrow{p} a$ é uma norma regulativa que proíbe a execução de uma ação $a \in \mathcal{A}$ em um contexto $\varphi \subseteq \mathcal{L}^{uaf}$ caso as pré-condições de execução da ação sejam satisfeitas, onde $\varphi = \{p_1, \dots, p_i\}$ é um conjunto de fórmulas atômicas e \mathcal{A} está introduzido na definição 3.

Exemplo 9 Considere a ação $move$ introduzida no exemplo 3 e os estados de objetivos e conflito, s e s_j , introduzidos nos exemplos 5 e 8, respectivamente. Uma norma que ao ser obedecida é capaz de evitar o estado s_j e manter o estado s alcançável² é $n = \{at(Ag_2, L_2), at(Ag_1, L_1), next(L_1, L_2)\} \xrightarrow{p} move(Ag_1, L_1, L_2)$ que, em linguagem natural, pode ser lida como “se um agente Ag_2 está em um local L_2 , um agente Ag_1 está em um local L_1 e L_1 é adjacente a L_2 , então Ag_1 está proibido de se mover de L_1 para L_2 ”.

Definição 12 (Instância de Norma) Dada uma norma n , uma instância n_i de n é uma representação específica de n , tal que, $\varphi^{n_i} \subseteq \mathcal{L}^{gaf}$ é um conjunto de fórmulas atômicas ground, $a^{n_i} \in \mathcal{A}'$ é uma instância da ação $a^n \in \mathcal{A}$, com \mathcal{A} , \mathcal{A}' e n introduzidos nas definições 3, 4 e 11, respectivamente. Além do mais, $\mathcal{I}^n = \{inst_1, \dots, inst_k\}$ é o conjunto das instâncias de n .

Exemplo 10 Considere a norma n introduzida no exemplo 9. Uma instância de n é $n_i = \{at(ag_2, b), at(ag_1, a), next(a, b)\} \xrightarrow{p} move(ag_1, a, b)$ que proíbe o agente ag_1 de se mover do local a para o local b se o agente ag_2 estiver em b , ag_1 estiver em a e a e b forem adjacentes.

² Como os agentes podem se deslocar entre locais desocupados livremente, a norma n não impede que os agentes alcancem os seus locais de destino. Supondo que exista, pelo menos, um caminho que os agentes possam percorrer na grid para se deslocarem para os seus destinos, então, os objetivos do sistema são alcançáveis.

Definição 13 (Norma Aplicável) *Uma norma n é aplicável em um estado $s \in S$, denotado por $n \triangleright s$, se $\exists n_i \in \mathcal{I}^n : \varphi^{n_i} \cup \text{Pre}^{a^{n_i}} \subseteq s$, com S , Pre , n e \mathcal{I}^n introduzidos nas definições 2, 3, 11 e 12, respectivamente.*

Exemplo 11 *Considere o estado s , a norma n e a instância n_i de n introduzidos nos exemplos 2, 9 e 10, respectivamente. Dado que $\varphi^{n_i} \cup \text{Pre}^{a^{n_i}} \subseteq s$, então $n \triangleright s$. Como a instância n_i regula o agente ag_1 , tem-se que ag_1 está proibido de executar a ação $\text{move}(ag_1, a, b)$ no estado s .*

Por meio do seu ciclo de raciocínio, os agentes podem determinar as ações que são proibidos de executar em um estado ao analisarem as instâncias das normas aplicáveis naquele estado. Assim, as ações proibidas pelas instâncias são, então, ações que não devem ser executadas pelos agentes no estado em questão.

Definição 14 (Regulação) *Dado um estado $s \in S$ e uma ação $a \in \mathcal{I}^A$ executável em s , uma norma n regula um agente no estado s se: (i) o agente tem a intenção de executar a ; (ii) $n \triangleright s$; (iii) $\exists n_i \in \mathcal{I}^n : a^{n_i} = a$; e (iv) o agente muda a sua intenção e obedece à norma, com S , \mathcal{I}^A e \mathcal{I}^n introduzidos nas definições 2, 4 e 12, respectivamente.*

Dizemos também que uma norma aplicável em um estado não regula um agente quando: não há instância da norma que incide sobre o agente; há instância que incide sobre o agente, contudo, o agente não tem intenção de executar a ação que a instância proíbe; o agente desobedece à norma. Todavia, saber as intenções dos agentes requer ter acesso ao seu maquinário interno e estados mentais, o que geralmente não é possível (SERRANO; ROVATSOS; BOTÍA, 2012; GURNEY *et al.*, 2022). Logo, neste trabalho, assumimos que não é possível ter este tipo acesso. Como neste trabalho os agentes optam por obedecer às normas, nós assumimos também que quando uma norma é aplicável em um estado, ela está efetivamente regulando os agentes para os quais as suas instâncias incidem, independentemente de alterações nas intenções prévias dos agentes.

Exemplo 12 *Considere o estado s , a norma n e a sua instância n_i introduzidos nos exemplos 2, 9 e 10, respectivamente. Considere também duas situações que ocorrem em s : na primeira, o agente ag_1 pretende ocupar o local b onde encontra-se o agente ag_2 ao executar a ação $\text{move}(ag_1, a, b)$; na segunda, o agente ag_1 pretende ocupar o local vazio d ao executar a ação $\text{move}(ag_1, a, d)$. Conforme o exemplo 11, tem-se que $n \triangleright s$. Assim, na primeira situação, o agente ag_1 identifica que a instância n_i , que proíbe a execução de $\text{move}(ag_1, a, b)$, incide sobre ele e, ao segui-la, executa alternativamente a ação $\text{move}(ag_1, a, d)$. Neste caso, como a norma alterou a intenção do agente, podemos dizer que a norma regulou o agente de fato. Embora na segunda situação uma colisão entre os agentes também seja evitada, isso não ocorre por causa*

da norma. Como o comportamento esperado já seria apresentado por ag_1 em tal situação, ele se desloca para o local desejado independentemente da instância n_j incidir sobre ele. Desta forma, a intenção inicial de ag_1 permanece inalterada. Logo, a norma não regulou o agente de fato. Contudo, para verificarmos isto, necessitamos comparar o estado mental do agente antes e depois dele considerar a norma em seu raciocínio. Como assumimos que isto é inviável e os agentes seguem as normas, consideramos então que em ambas as situações a norma regulou os agentes efetivamente.

Definição 15 (Espaço das Normas) O espaço das normas $\mathcal{N} = \{\varphi \xrightarrow{p} a \mid \varphi \in 2^{\mathcal{L}^{uaf}} \wedge a \in \mathcal{A}\}$ é o conjunto de todas as normas de um cenário de aplicação, com \mathcal{A} e n introduzidos nas definições 3 e 11, respectivamente.

O espaço das normas é formado por normas que variam as ações que proíbem e as fórmulas atômicas que formam os seus contextos. Tais contextos, por sua vez, variam o tamanho de 0 a $|\mathcal{L}^{uaf}|$. Contudo, conforme introduzido no capítulo 1, o espaço das normas é um conjunto potencialmente grande, pois pode explodir combinatoriamente com o crescimento do número de fórmulas atômicas e ações.

Exemplo 13 Considerando que no cenário do tráfego de carros $|\mathcal{L}^{uaf}| = 4$ e $|\mathcal{A}| = 2$, então, existem $|\mathcal{N}| = 32$ normas candidatas para regular os agentes. No entanto, caso o cenário do tráfego de carros seja estendido de modo que $|\mathcal{L}^{uaf}| = 10$ e $|\mathcal{A}| = 10$, então, o número de normas candidatas aumenta para $|\mathcal{N}| = 10240$.

Além do mais, o espaço das normas pode ser constituído por normas equivalentes, todavia, sintaticamente distintas.

Definição 16 (Normas Equivalentes) Duas normas n_i e n_j são equivalentes se $\varphi^{n_i} \cup Pre^{a^{n_i}} = \varphi^{n_j} \cup Pre^{a^{n_j}}$ e $a^{n_i} = a^{n_j}$.

Duas normas equivalentes são aplicáveis no mesmo conjunto de estados do sistema e possuem o mesmo conjunto de instâncias. Portanto, elas regulam os mesmos comportamentos dos agentes.

Exemplo 14 Considere a norma $n = \{at(Ag_2, L_2), at(Ag_1, L_1), next(L_1, L_2)\} \xrightarrow{p} move(Ag_1, L_1, L_2)$ do exemplo 9 e as normas $n_j = \{at(Ag_2, L_2), next(L_1, L_2)\} \xrightarrow{p} move(Ag_1, L_1, L_2)$ e $n_k = \{at(Ag_2, L_2)\} \xrightarrow{p} move(Ag_1, L_1, L_2)$. Dado que $\varphi^n \cup Pre^{a^n} = \varphi^{n_j} \cup Pre^{a^{n_j}} = \varphi^{n_k} \cup Pre^{a^{n_k}} = \{at(Ag_1, L_1), at(Ag_2, L_2), next(L_1, L_2)\}$, então n , n_j e n_k são normas equivalentes.

Definição 17 (Norma Mínima) Dado um conjunto de normas equivalentes, conforme a definição 16, denotado por $\mathcal{N}^{equiv} = \{n_1, \dots, n_i\}$, a norma mínima $n_j \in \mathcal{N}^{equiv}$ é a norma para qual $\varphi^{n_j} \cap Pre^{a^{n_j}} = \emptyset$.

Portanto, a norma mínima é a norma que possui o menor contexto do conjunto \mathcal{N}^{equiv} a qual pertence.

Exemplo 15 Considere o conjunto $\mathcal{N}^{equiv} = \{n, n_j, n_k\}$, com as normas n , n_j e n_k introduzidas nos exemplos 9, 14 e 14, respectivamente. Como $\varphi^{n_k} \cap Pre^{a^{n_k}} = \emptyset$, $\varphi^{n_j} \cap Pre^{a^{n_j}} = \{next(L_1, L_2)\}$ e $\varphi^n \cap Pre^{a^n} = \{at(Ag_1, L_1), next(L_1, L_2)\}$, então, n_k é mínima.

O conceito de norma mínima é útil para tornar o espaço das normas menor ao considerarmos uma única norma, a mínima, de um conjunto de normas equivalentes.

Outro tipo de estado importante para este trabalho é o *estado de timeout*.

Definição 18 (Estado de Timeout) Considerando um timeout $\Delta \in \mathbb{N}^*$ (naturais diferentes de zero) para os agentes alcançarem um conjunto de objetivos do sistema $\mathcal{G} \in \mathcal{G}^*$, um estado $s \in \mathcal{S}$ é dito ser um estado de timeout se s for o estado do sistema no instante de tempo Δ e s não for um estado de objetivos nem de conflitos.

3.2 HISTÓRIAS DE UM SMAN

Na seção anterior, nós apresentamos os conceitos básicos para a construção do nosso ferramental teórico e formal para a síntese de normas eficientes. Nesta seção, nós avançamos na construção deste arcabouço apresentando um conjunto de conceitos que estão relacionados à capacidade das normas em regular os agentes apropriadamente para, então, definirmos formalmente o que são as normas eficientes. Para isto, nós apresentamos inicialmente as definições de *traço* e *história*, ambas sequências baseadas em árvores.

Definição 19 (Traço) Um traço $t = ((null, s_i, a_r), (s_i, s_j, a_b), (s_j, s_k, a_u), \dots, (s_l, s_m, null))$ é uma sequência finita de triplas, onde $\{s_i, s_j, s_k, \dots, s_l, s_m\} \subset \mathcal{S}$ e $\{a_r, a_b, \dots, a_u\} \subset \mathcal{I}^A$, com \mathcal{S} e \mathcal{I}^A introduzidos nas definições 2 e 4, respectivamente. Além do mais, $\mathcal{T} = \{t_1, t_2, \dots, t_n\}$ é o conjunto de todos os traços.

Em um traço, uma tripla (s_i, s_j, a_b) denota que o estado s_i é pai do estado s_j e sobre o estado s_j é executada a ação a_b . A execução da ação a_b resulta em uma transição do estado s_j para um estado filho s_k , conforme a tripla (s_j, s_k, a_u) . Por sua vez, a tripla $(null, s_i, a_r)$ denota que o estado s_i não possui pai. Logo, s_i é o estado inicial do traço. Por outro lado, a tripla $(s_l, s_m, null)$ denota que nenhuma ação é executada sobre o estado s_m . Logo, s_m não possui estado filho e é o estado final do traço. Adicionalmente, de acordo com a seção 3.1, podemos classificar o estado s_m como um estado de *objetivos*, *conflito* ou *timeout*.

Definição 20 (História) Uma história $h = ((null, s_0, a_r), (s_0, s_j, a_b), (s_j, s_k, a_u), \dots, (s_l, s_m, null))$ é um traço cujo estado inicial é um estado inicial do sistema $s_0 \in \mathcal{S}_0$, com \mathcal{S}_0

introduzido na definição 2. Além disto, $\mathcal{H} = \{h_1, h_2, \dots, h_p\}$, com $\mathcal{H} \subset \mathcal{T}$, é o conjunto de todas as histórias do sistema, com \mathcal{T} introduzido na definição 19.

Por fim, nós apresentamos as definições essenciais para conceitualizarmos, de acordo com o contexto deste trabalho, os *Sistemas Multiagente Normativos*.

Definição 21 (SMA_n) *Um Sistema Multiagente Normativo SMA_n = $\langle S, S_0, \mathcal{I}^A, \mathcal{G}^*, \mathcal{C}, \mathcal{N}, \Delta, \mathcal{H}, Ag \rangle$ é um SMA, onde $S, S_0, \mathcal{I}^A, \mathcal{G}^*, \mathcal{C}, \mathcal{N}$ e \mathcal{H} são conjuntos de estados do sistema, estados iniciais, instâncias de ações, objetivos do sistema, conflitos, normas proibitivas e histórias, conforme as definições 2, 2, 4, 5, 8, 15 e 20, respectivamente, Δ é um timeout, conforme a definição 18, e $Ag = \{ag_1, \dots, ag_m\}$ é um conjunto de agentes.*

Definição 22 (Histórias de Sucesso e Insucesso) *As histórias de sucesso do sistema, denotadas por $\mathcal{H}^{suc} \subseteq \mathcal{H}$, são o conjunto das histórias que findam em um estado de objetivos $s_{\mathcal{G}} \in S^{\mathcal{G}}$ de um objetivo do sistema $\mathcal{G} \in \mathcal{G}^*$, com \mathcal{G} e \mathcal{G}^* introduzidos na definição 5 e $S^{\mathcal{G}}$ introduzido na definição 7. Por outro lado, as histórias de insucesso do sistema, denotadas por $\mathcal{H}^{insuc} \subseteq \mathcal{H}$, são o conjunto das histórias que não findam em um estado de objetivos. As histórias de insucesso, por sua vez, dividem-se entre as histórias de conflito e as histórias de timeout. As histórias de conflito, denotadas por $\mathcal{H}^{conf} \subseteq \mathcal{H}^{insuc}$, são constituídas pelas histórias que findam em um estado de conflito, enquanto as histórias de timeout, denotadas por $\mathcal{H}^{time} \subseteq \mathcal{H}^{insuc}$, são constituídas pelas histórias que findam em um estado de timeout. Portanto, $\mathcal{H} = \{h \mid h \in \mathcal{H}^{suc} \vee h \in \mathcal{H}^{insuc}\}$ e $\mathcal{H}^{insuc} = \{h \mid h \in \mathcal{H}^{conf} \vee h \in \mathcal{H}^{time}\}$.*

Exemplo 16 *Considere um cenário do tráfego de carros simplificado ilustrada pela figura 4, onde dois agentes móveis, ag_1 e ag_2 , devem, a partir do estado inicial $s_0 = \{at(ag_1, g), at(ag_2, f), next(a, b), \dots, next(h, i)\}$ alcançar o conjunto de objetivos do sistema $\mathcal{G} = \{at(ag_1, f), at(ag_2, i)\}$. Considere também que, para um timeout $\Delta = 5$, as histórias do sistema são constituídas pelos conjuntos $\mathcal{H}^{suc} = \{h_1\}$, $\mathcal{H}^{conf} = \{h_2\}$ e $\mathcal{H}^{time} = \{h_3\}$. A figura 5 ilustra tais histórias como uma árvore, onde os vértices $\{s_0, \dots, s_{12}\}$ representam estados do sistema, com s_0 sendo a raiz, e as arestas representam instâncias das ações $move(Ag_1, L_1, L_2)$ e $idle(Ag_1, L_1)$ ³. Um caminho entre a raiz e uma folha é uma história^{4,5} (identificada à direita da folha), onde as arestas partem de um estado pai para um estado filho. As folhas, por sua vez, estão destacadas em azul, vermelho e laranja para representar estados de sucesso, conflito e timeout, respectivamente.*

³ Conforme dito no capítulo 1, além de moverem-se entre locais adjacentes, os agentes podem permanecer parados nos locais que ocupam na grid. Esta ação é descrita por $idle(Ag_1, L_1)$, onde um agente Ag_1 fica parado em um local L_1 caso esteja ocupando tal local.

⁴ Por simplicidade, nós omitimos dos vértices das árvores nos exemplos deste capítulo todas as fórmulas atômicas cujos predicados são *next*.

⁵ Ao analisarmos a árvore, podemos reconstruir as histórias de acordo com a definição 20. Con-

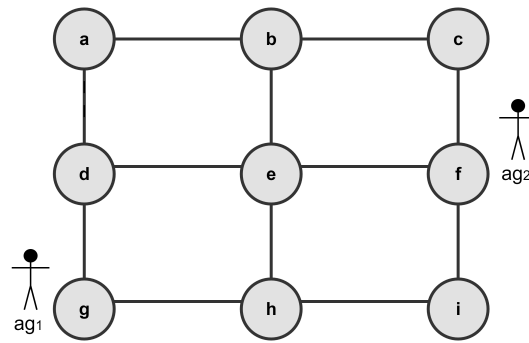


Figura 4 – Situação inicial para o cenário do tráfego de carros simplificado.

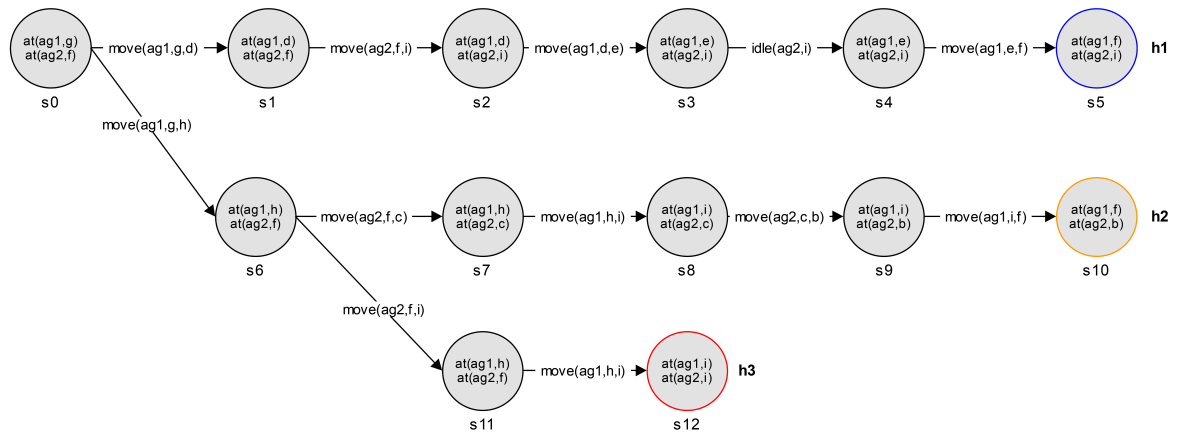


Figura 5 – Árvore com três histórias do cenário do tráfego de carros simplificado.

Ao analisarmos a figura 5, podemos observar que na história h_1 , os agentes alcançam os objetivos do sistema em 5 passos. Por outro lado, na história h_2 , os agentes percorrem um caminho na grid através do qual não é possível alcançar os objetivos do sistema de acordo com o timeout estabelecido. Logo, a história h_2 finda em um estado de timeout. Por fim, na história h_3 , os agentes colidem no local i, o que faz com que a história finde em um estado de conflito.

Ao regularem os agentes, as normas restringem os seus comportamentos a um conjunto de comportamentos esperados ou aceitáveis (BALKE *et al.*, 2013). No contexto deste trabalho, isto significa restringir as histórias de \mathcal{H} a um conjunto de histórias que possam ocorrer⁶ quando os agentes são regulados pelas normas.

Definição 23 (Histórias das Normas) Dada uma norma $n \in \mathcal{N}$, denotamos por $\mathcal{H}_n \subseteq \mathcal{H}$ as histórias de \mathcal{H} que podem ocorrer quando os agentes são regulados pela norma

siderando a história h_3 podemos verificar que $h_3 = ((null, s_0, move(ag_1, g, h)), (s_0, s_6, move(ag_2, f, i)), (s_6, s_{11}, move(ag_1, h, i)), (s_{11}, s_{12}, null))$.

⁶ Embora as normas não retirem do *roll* de ações dos agentes as ações que proíbem, elas podem inviabilizar a ocorrência de certas histórias quando as ações proibidas são necessárias para que tais histórias ocorram. Como, neste trabalho, os agentes obedecem às normas, eles se tornam inaptos a concluir tais histórias.

n , com \mathcal{N} e \mathcal{H} introduzidos nas definições 15 e 20, respectivamente. Tais histórias são categorizadas em histórias de sucesso, denotadas por $\mathcal{H}_n^{suc} \subseteq \mathcal{H}^{suc}$, e histórias de insucesso, denotadas por $\mathcal{H}_n^{insuc} \subseteq \mathcal{H}^{insuc}$. As histórias de insucesso, por sua vez, dividem-se entre as histórias de conflito, denotadas por $\mathcal{H}_n^{conf} \subseteq \mathcal{H}^{conf}$ e histórias de timeout, denotadas por $\mathcal{H}_n^{time} \subseteq \mathcal{H}^{time}$. Portanto, temos que $\mathcal{H}_n = \mathcal{H}_n^{suc} \cup \mathcal{H}_n^{insuc}$ e $\mathcal{H}_n^{insuc} = \mathcal{H}_n^{conf} \cup \mathcal{H}_n^{time}$, com \mathcal{H}^{suc} , \mathcal{H}^{insuc} , \mathcal{H}^{conf} e \mathcal{H}^{time} introduzidos na definição 22.

A história do SMAn que ocorre quando ele executa com uma norma pode ser de sucesso sem a contribuição da norma. Isto ocorre quando as ações que os agentes executam não geram conflitos, o que faz com que inexistam estados onde a norma necessite ser aplicável para que conflitos sejam evitados. Nestas circunstâncias, consideramos também que para uma história de sucesso ser de uma norma é necessário que a história possua, pelo menos, um estado onde a norma é aplicável. Desta forma, a norma auxilia os agentes a apresentarem um comportamento desejável quando há necessidade de regulação contribuindo, então, para que a história de sucesso ocorra.

Definição 24 (Histórias de Sucesso das Normas) *Dada uma norma $n \in \mathcal{N}$, uma história de sucesso $h \in \mathcal{H}^{suc}$ é uma história de sucesso de n se: i) $\exists s \in \mathcal{S} \exists tp \in h : n \triangleright s \wedge s \in tp$ (com tp sendo uma tripla de h); e ii) a história pode ocorrer, com \mathcal{S} , \mathcal{N} e \mathcal{H}^{suc} introduzidos nas definições 2, 15 e 22, respectivamente.*

Certas histórias de timeout ocorrem por causa das normas. Por exemplo, uma norma que proíbe os agentes de se locomoverem pela grid faz com que as ações que compõem a história que ocorre sejam instâncias da ação $idle(Ag_1, L_1)$. Neste contexto, o timeout é atingido e os agentes não alcançam um estado de objetivos. Por outro lado, certas histórias de timeout ocorrem por motivos não relacionados as normas como quando os agentes entram em loop ou os seus planos não podem ser finalizados dentro do timeout estabelecido. Neste sentido, assim como ocorre com as histórias de sucesso, consideramos também que para uma história de timeout ser de uma norma é necessário que a história possua, pelo menos, um estado onde a norma é aplicável. Desta forma, a norma contribui para que a história de timeout ocorra.

Definição 25 (Histórias de Timeout das Normas) *Dada uma norma $n \in \mathcal{N}$, uma história de timeout $h \in \mathcal{H}^{time}$ é uma história de timeout de n se: i) $\exists s \in \mathcal{S} \exists tp \in h : n \triangleright s \wedge s \in tp$ (com tp sendo uma tripla de h); e ii) a história pode ocorrer, com \mathcal{S} , \mathcal{N} e \mathcal{H}^{time} introduzidos nas definições 2, 15 e 22, respectivamente.*

Considerando que os agentes obedecem às normas, os conflitos ocorrem meramente pela incapacidade das normas em regular os agentes de forma adequada. Por exemplo, é possível que existam estados onde os agentes necessitem ser regulados para evitar conflitos, contudo, certas normas não são aplicáveis, o que possibilita aos

agentes agirem livremente e os conflitos ocorrerem. Por outro lado, embora certas normas sejam aplicáveis em tais estados, os conflitos ainda ocorrem, pois elas não descrevem corretamente o comportamento que os agentes devem seguir para que os conflitos sejam evitados. Neste sentido, consideramos, mais especificamente, que as histórias de conflito de uma norma são aquelas que podem ocorrer independentemente de existir um estado onde a norma é aplicável.

Definição 26 (Histórias de Conflito das Normas) *Dada uma norma $n \in \mathcal{N}$, uma história de conflito $h \in \mathcal{H}^{conf}$ é uma história de conflito de n se: i) a norma não é aplicável em nenhum estado de nenhuma tripla da história; ou ii) a norma é aplicável em algum estado de alguma tripla da história, contudo, a história pode ocorrer normalmente.*

Exemplo 17 *Considere o cenário do tráfego de carros simplificado apresentado no exemplo 16. Considere também as normas $n_1 = \{at(ag_2, i)\} \xrightarrow{p} move(ag_1, h, i)$, $n_2 = \{at(Ag_2, L_2)\} \xrightarrow{p} move(Ag_1, L_1, L_2)$ e $n_3 = \{ \} \xrightarrow{p} move(Ag_1, L_1, L_2)$, e os conjuntos de histórias $\mathcal{H}^{suc} = \{h_1, h_2, h_9\}$, $\mathcal{H}^{conf} = \{h_3, h_4, h_8\}$ e $\mathcal{H}^{time} = \{h_5, h_6, h_7, h_{10}\}$, com $\mathcal{H}^{suc} \subset \mathcal{H}^{suc}$, $\mathcal{H}^{conf} \subset \mathcal{H}^{conf}$ e $\mathcal{H}^{time} \subset \mathcal{H}^{time}$. A figura 6 ilustra as histórias de tais conjuntos como uma árvore construída de modo similar à árvore do exemplo 16, onde os vértices são estados do conjunto $\{s_0, \dots, s_{29}\}$.*

Ao analisarmos a figura 6, podemos observar que não existe estado da história de sucesso h_1 onde as normas n_1 e n_2 são aplicáveis⁷. Logo, a história h_1 pode ocorrer sem a contribuição de tais normas. Por outro lado, ambas as normas n_1 e n_2 são aplicáveis no estado s_7 proibindo os agentes de alcançarem o estado de conflito s_{11} . Logo, tais normas evitam que a história de conflito h_3 ocorra e contribuem para que a história de sucesso h_2 ocorra. Entretanto, a norma n_1 não é capaz de evitar que a história de conflito h_4 ocorra, uma vez que a norma não é aplicável no estado s_{12} (onde há necessidade de os agentes serem regulados). Contudo, a norma n_2 é aplicável no estado s_{12} proibindo os agentes de alcançarem o estado de conflito s_{13} . Assim, a norma n_2 evita que a história de conflito h_4 ocorra. Contudo, a partir do estado s_{12} , ao obedecerem a norma n_2 , os agentes podem alcançar o estado de timeout s_{14} fazendo com que a história de timeout h_5 ocorra. Como o estado s_7 também faz parte da história h_5 , ambas as normas n_1 e n_2 contribuem para que tal história ocorra.

A história de timeout h_6 não possui estados onde as normas anteriores são aplicáveis. Portanto, elas não contribuem para que a história h_6 ocorra. A norma n_1 não é aplicável nos estados das demais histórias. Entretanto, como h_8 é uma história de conflito e a norma n_1 não é capaz de evitá-la, então a história h_8 é uma história de conflito de n_1 . Por outro lado, a norma n_2 evita a história de conflito h_8 e contribui

⁷ Enquanto a norma n_1 é aplicável nos estados que possuem as fórmulas atômicas $at(ag_1, h)$ e $at(ag_2, i)$, a norma n_2 é aplicável nos estados que possuem fórmulas atômicas que denotam que dois agentes distintos ocupam locais adjacentes na grid, como o estado s_7 , por exemplo.

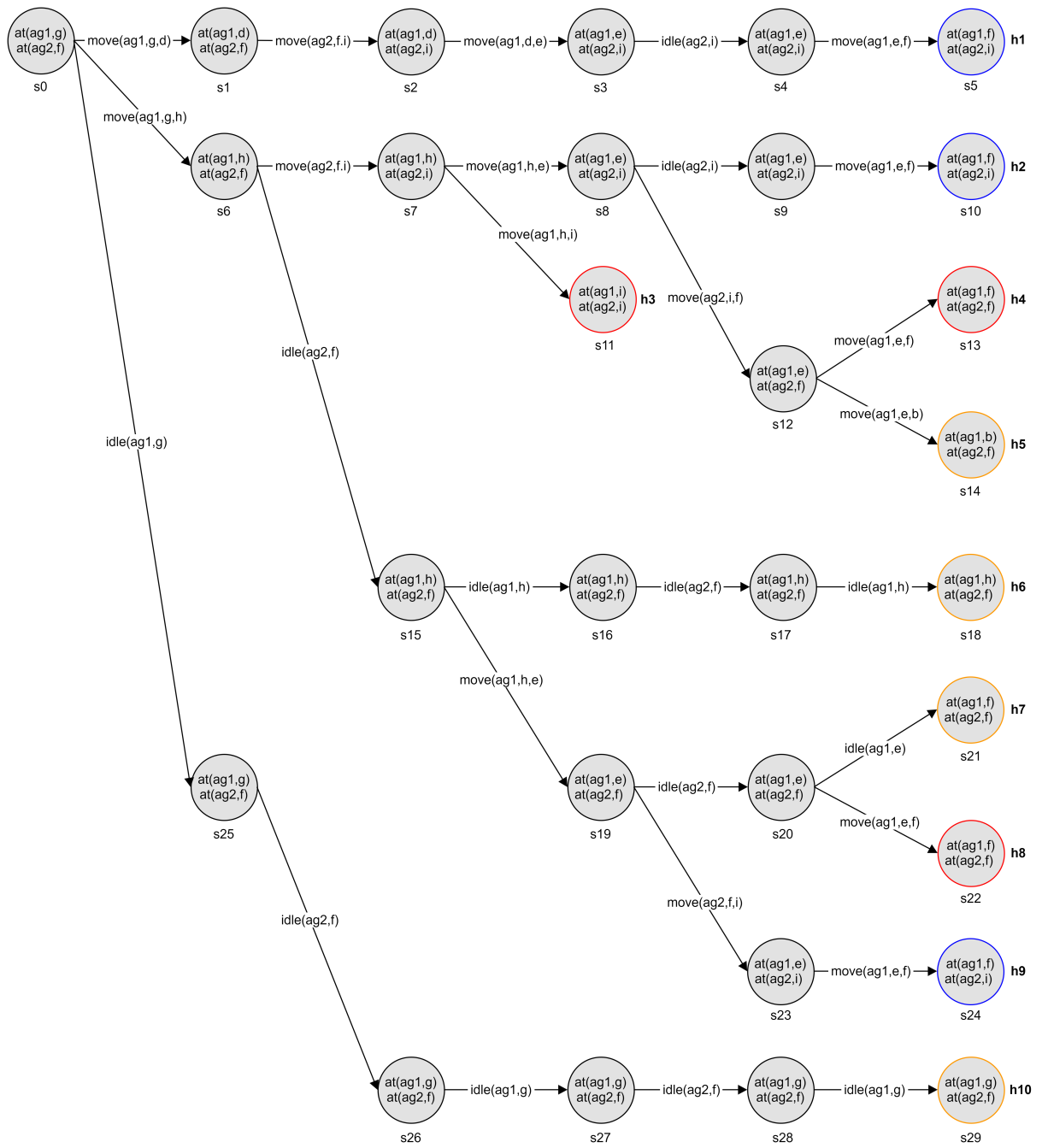


Figura 6 – Árvore com dez histórias do cenário do tráfego de carros simplificado.

para que a história de sucesso h_9 ocorra, uma vez que é aplicável nos estados s_{19} e s_{20} . Como tais estados também fazem parte da história de timeout h_7 , tem-se que h_7 é uma história de timeout de n_2 . Ainda, n_2 não é aplicável nos estados da história de timeout h_{10} , logo h_{10} não é uma história de timeout de n_2 .

Em contrapartida, a norma n_3 é aplicável em todos os estados do sistema⁸.

⁸ As instâncias da norma n_3 são constituídas por fórmulas atômicas que denotam que um agente ocupa um local da grid que é adjacente a um outro local. Como a grid é um grafo onde todos os vértices possuem vizinhos e, em todos os estados do sistema, os agentes estão, obrigatoriamente,

Logo, os agentes podem executar somente a ação idle. Consequentemente, por não se moverem, os agentes não colidem nem alcançam os objetivos do sistema. Portanto, a única história que pode ocorrer quando o sistema executa com a norma n_3 é a história de timeout h_{10} . Assim, os conjuntos das histórias de sucesso, conflito e timeout das normas n_1 , n_2 e n_3 são, respectivamente, $\mathcal{H}_{n_1}^{suc} = \{h_2\}$, $\mathcal{H}_{n_1}^{conf} = \{h_4, h_8\}$, $\mathcal{H}_{n_1}^{time} = \{h_5\}$, $\mathcal{H}_{n_2}^{suc} = \{h_2, h_9\}$, $\mathcal{H}_{n_2}^{conf} = \{\}$, $\mathcal{H}_{n_2}^{time} = \{h_5, h_7\}$, $\mathcal{H}_{n_3}^{suc} = \{\}$, $\mathcal{H}_{n_3}^{conf} = \{\}$ e $\mathcal{H}_{n_3}^{time} = \{h_{10}\}$.

Neste ponto, nós introduzimos os conceitos essenciais para fundamentarmos, conforme a definição 27, as *normas eficientes*.

Definição 27 (Norma Eficiente) Dados um estado inicial $s_0 \in S_0$, um conjunto de objetivos do sistema $\mathcal{G} \in \mathcal{G}^*$ e um estado de objetivos $s_{\mathcal{G}} \in S^{\mathcal{G}}$, dizemos que uma norma $n \in \mathcal{N}$ é eficiente se contribui para que o sistema transite de s_0 para $s_{\mathcal{G}}$ sem permitir o alcance de estados de conflitos. Para isto: i) a norma deve selecionar, pelo menos, uma história de sucesso, tal que, $\mathcal{H}_n^{suc} \neq \{\}$; e ii) a norma não deve selecionar histórias de conflito, tal que, $\mathcal{H}_n^{conf} = \{\}$. Por sua vez, S_0 , \mathcal{G}^* , \mathcal{G} , $S^{\mathcal{G}}$ e \mathcal{N} estão introduzidos nas definições 2, 5, 5, 7 e 15, respectivamente.

Exemplo 18 Considere novamente o exemplo 17. Supondo que $\mathcal{H} = \{h_1, \dots, h_{10}\}$, então a norma n_1 é eficiente dado que $\mathcal{H}_{n_2}^{suc} \neq \{\}$, $\mathcal{H}_{n_2}^{conf} = \{\}$. Por outro lado, como $\mathcal{H}_{n_1}^{conf} \neq \{\}$ e $\mathcal{H}_{n_3}^{suc} = \{\}$, então as normas n_1 e n_3 não são eficientes.

Por outro lado, podemos dizer que uma norma é *ineficiente* caso o seu conjunto de histórias de sucesso seja vazio. Portanto, a norma n_3 do exemplo 18 é ineficiente. Além do mais, embora seja intuitivo pensar que normas eficientes também evitam histórias de timeout, conforme exposto anteriormente, o propósito das normas eficientes é evitar histórias de conflito e contribuir para a ocorrência de, pelo menos, uma história de sucesso. Embora seja possível, as normas eficientes não necessariamente evitam histórias de timeout.

Definição 28 (Norma Mais Flexível) Dados um estado inicial $s_0 \in S_0$, um conjunto de objetivos do sistema $\mathcal{G} \in \mathcal{G}^*$ e duas normas n_i e n_j , com $\{n_i, n_j\} \subseteq \mathcal{N}$, dizemos que n_i é mais flexível que n_j , denotado por $n_i \succ_f n_j$, se $|\mathcal{H}_{n_i}^{suc}| > |\mathcal{H}_{n_j}^{suc}|$. Neste contexto, dado um conjunto de normas eficientes $\mathcal{N}^{efic} = \{n_1, \dots, n_m\}$, tal que, $\mathcal{N}^{efic} \subseteq \mathcal{N}$, uma norma $n_k \in \mathcal{N}^{efic}$ é a norma eficiente mais flexível de \mathcal{N}^{efic} caso $\nexists n_{k'} \in \mathcal{N}^{efic} : n_{k'} \succ_f n_k$, com S_0 , \mathcal{G}^* , \mathcal{G} e \mathcal{N} introduzidos nas definições 2, 5, 5 e 15, respectivamente.

Ao serem regulados pela norma eficiente mais flexível, os agentes possuem mais autonomia para executarem as suas ações, pois tal norma dá aos agentes mais opções de histórias de sucesso para seguirem.

ocupando um local, então a norma n_3 é aplicável em todos os estados do sistema.

Exemplo 19 Considere um espaço de normas $\mathcal{N} = \{n_i, n_j, n_k, n_m, n_l, n_p\}$, onde n_i e n_j são normas eficientes, n_k e n_m são normas ineficientes e n_l e n_p são normas que não são eficientes nem ineficientes. O diagrama de Venn ilustrado pela figura 7 apresenta o conjunto das histórias das normas de \mathcal{N} para um estado inicial $s_0 \in S_0$, um conjunto de objetivos do sistema $\mathcal{G} \in \mathcal{G}^*$ e um timeout Δ .

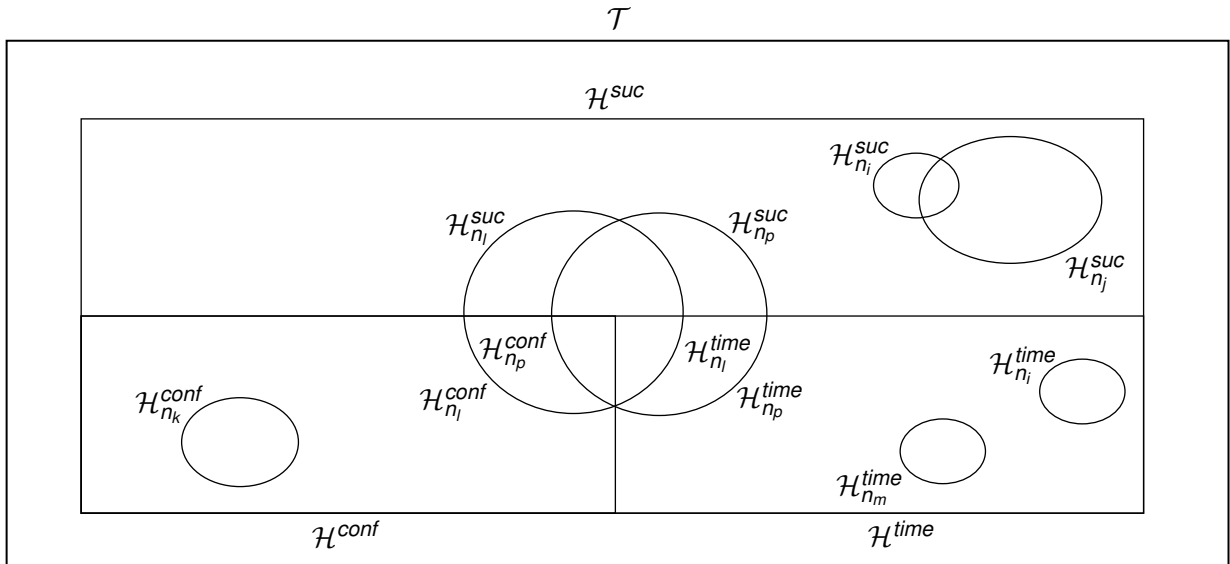


Figura 7 – Diagrama de Venn das histórias de n_i, n_j, n_k, n_l, n_m e n_p .

No diagrama, não há representação para o conjunto das histórias de conflito das normas n_i e n_j já que, de acordo com a definição 27, tais conjuntos são vazios. Entretanto, a norma n_j possui uma representação não vazia para o seu conjunto de histórias de timeout, o que denota que quando os agentes são regulados por n_j é possível que o sistema finde a sua execução em um estado de timeout. Além do mais, podemos concluir que $n_j > n_i$, já que, visualmente, $|\mathcal{H}_{n_j}^{SUC}| > |\mathcal{H}_{n_i}^{SUC}|$.

Por outro lado, as normas n_k e n_m por serem ineficientes não possuem representações para os conjuntos das suas histórias de sucesso já que tais conjuntos são vazios. A norma n_k , por sua vez, possui representação apenas para o seu conjunto de histórias de conflito, o que denota que o sistema finda a sua execução em estados de conflito quando os agentes são regulados pela norma n_k . Contudo, a norma n_m possui representação apenas para o seu conjunto de histórias de timeout, logo, quando o sistema executa com tal norma, apenas histórias de timeout ocorrem.

Por fim, as normas n_l e n_p não são eficientes nem ineficientes, logo, possuem todos os tipos de histórias. Ao executar com tais normas, dependendo da intenção dos agentes, o sistema pode obter histórias de sucesso ou insucesso. Visualmente, podemos concluir que $|\mathcal{H}_{n_l}^{SUC}| \cong |\mathcal{H}_{n_p}^{SUC}|$, $|\mathcal{H}_{n_l}^{CONF}| > |\mathcal{H}_{n_p}^{CONF}|$ e $|\mathcal{H}_{n_p}^{TIME}| > |\mathcal{H}_{n_l}^{TIME}|$.

Adicionalmente, a figura 8 apresenta o diagrama de Venn para as histórias de sucesso, conflito e timeout das normas n_1, n_2 e n_3 do exemplo 17.

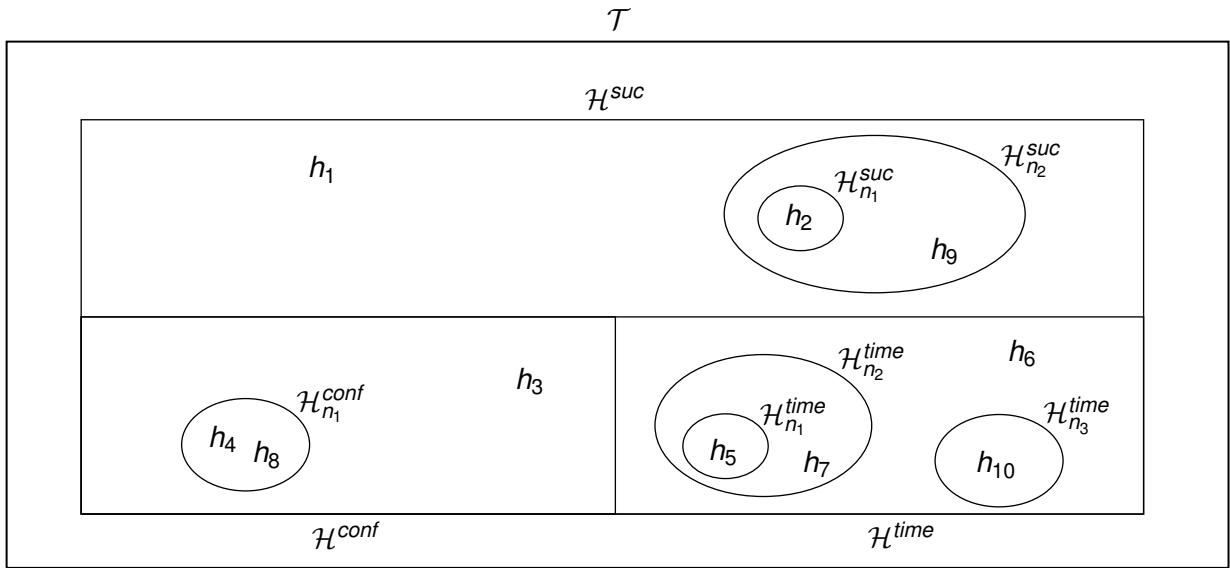


Figura 8 – Diagrama de Venn para as histórias das normas do exemplo 17.

O diagrama mostra que existem três histórias de \mathcal{H} , h_1 , h_3 e h_6 , que não fazem parte das histórias das normas n_1 , n_2 e n_3 . Também temos que $\mathcal{H}_{n_1}^{suc} \subset \mathcal{H}_{n_2}^{suc}$, $\mathcal{H}_{n_1}^{time} \subset \mathcal{H}_{n_2}^{time}$ e que $\mathcal{H}_{n_1}^{conf}$ e $\mathcal{H}_{n_3}^{time}$ não possuem interseção com outros conjuntos. Podemos ressaltar ainda que ao modificarmos o timeout, estado inicial ou objetivos do sistema obteremos, possivelmente, diferentes diagramas de Venn, onde os conjuntos das histórias das normas possuem mais ou menos histórias do sistema.

De acordo com o exemplo 19, uma pergunta que surge é: qual norma é melhor para o sistema, n_l ou n_p ? A resposta para esta pergunta pode ser particularmente útil em cenários onde normas eficientes não existem ou encontrá-las é inviável (dado que para determinar se uma norma é eficiente é necessário analisar todo o conjunto \mathcal{H}^{conf}). Alternativamente, pode-se buscar por normas quase eficientes onde a necessidade de que os conjuntos das histórias de conflitos das normas sejam vazios é relaxada.

Definição 29 (Norma Quase Eficiente) Dado um estado inicial $s_0 \in S_0$ e um conjunto de objetivos do sistema $\mathcal{G} \in \mathcal{G}^*$, dizemos que uma norma $n \in \mathcal{N}$ é quase eficiente se contribui para que o sistema transite de s_0 para um estado de objetivos $s_g \in S_g$ e selecione, pelo menos, uma história de sucesso, tal que, $\mathcal{H}_n^{suc} \neq \{\}$.

De acordo com a definição 29, diferentes normas de um espaço podem ser consideradas quase eficientes. Todavia, é desejável que o conjunto de histórias de conflito de uma norma quase eficiente seja pequeno ou vazio. Embora as normas n_l e n_p do exemplo 19 possam ser consideradas normas quase eficientes, podemos intuitivamente dizer que n_p é mais apropriada para regular os agentes do que n_l , pois o seu conjunto de histórias de conflito é significativamente menor. Todavia, para aferirmos qual das duas normas quase eficientes é melhor, avaliações quantitativas podem ser

empregadas. Para isto, podemos comparar as taxas das histórias de sucesso das normas conforme apresenta a definição 30.

Definição 30 (Taxas de Histórias de Sucesso e Conflito) *Dada uma norma $n \in \mathcal{N}$ e um subconjunto finito de histórias $\mathcal{H}' \subset \mathcal{H}$, a taxa das histórias de sucesso de n é dada por $t_n^{suc} = |\mathcal{H}'^{suc_n}|/|\mathcal{H}'_n|$, enquanto a taxa das histórias de conflito de n é dada por $t_n^{conf} = |\mathcal{H}'^{conf}|/|\mathcal{H}'_n|$.*

Como o mecanismo de comparação fornecido pela definição 30 depende de que as histórias das normas sejam totalmente conhecidas, podemos simplificá-lo considerando apenas um subconjunto de tais histórias. A partir de tal subconjunto, podemos determinar as taxas de histórias de sucesso e conflito e compará-las, por exemplo, a certos limiares de modo que a primeira taxa seja maior ou igual a um valor α enquanto a segunda taxa seja menor ou igual a um valor β , com $\{\alpha, \beta\} \subset \mathbb{R}_+^*$ (reais positivos). Assim, podemos determinar o quão apropriadas as normas quase eficientes são para regular os agentes. Diferentes estratégias podem ser empregadas para que subconjuntos de histórias de normas sejam obtidos, como amostragem, geração aleatória de histórias ou identificação de histórias em tempo de execução, por exemplo.

Exemplo 20 *Considere os conjuntos \mathcal{H}'^{suc} e \mathcal{H}'^{conf} de histórias de sucesso e conflito, respectivamente, das normas n_1 , n_2 e n_3 do exemplo 17. Podemos verificar que as taxas de sucesso das normas são $t_{n_1}^{suc} = 0.25$, $t_{n_2}^{suc} = 0.5$, $t_{n_3}^{suc} = 0$, ao passo que as taxas de conflito são $t_{n_1}^{conf} = 0.25$, $t_{n_2}^{conf} = 0$, $t_{n_3}^{conf} = 0$. Se considerarmos os seguintes limiares $\alpha = 0.5$ e $\beta = 0.5$, podemos concluir que a norma n_2 é a melhor opção para regular os agentes no cenário do tráfego de carros simplificado.*

3.3 ALCANÇABILIDADE DOS OBJETIVOS DO SISTEMA

Embora normas sejam mecanismos usados para evitar conflitos, certas normas os evitam em detrimento do alcance dos objetivos do sistema. Neste contexto, uma pergunta que surge é: *como sintetizar normas eficientes que, ao evitarem conflitos, mantêm os objetivos do sistema alcançáveis?* Uma abordagem ingênua, porém, intuitiva para responder esta pergunta é sintetizar normas que proíbem a execução da ação que faz com que o sistema transite de um estado $s_j \in \mathcal{S}$ para um estado de conflito $s_j \in \mathcal{S}$. Chamaremos esta ação de *ação proibitiva* e esta abordagem de *síntese ingênua*. Embora, ao serem obedecidas, tais normas façam com que os agentes evitem estados de conflito, é possível que não existam transições do estado s_j para um estado $s_k \in \mathcal{S}$ por onde os agentes possam satisfazer os objetivos do sistema. Para exemplificar este problema, a figura 9 apresenta uma árvore construída de forma análoga às árvores dos exemplos 16 e 17 considerando os conjuntos de histórias $\mathcal{H}^{conf} = \{h_1\}$, $\mathcal{H}^{time} = \{h_2, h_4\}$ e $\mathcal{H}^{suc} = \{h_3\}$.

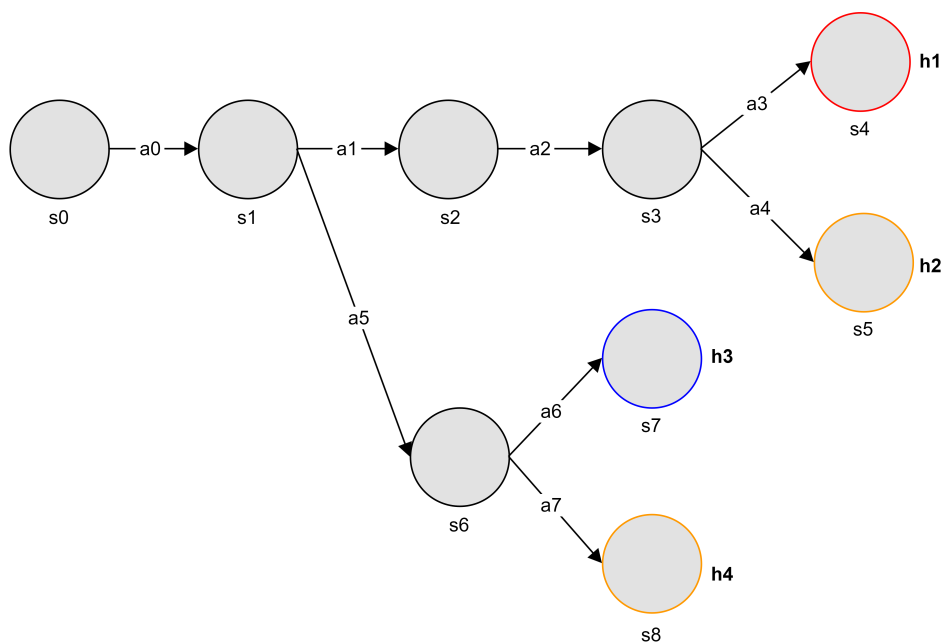


Figura 9 – Uma árvore de histórias hipotéticas.

De acordo com a síntese ingênua, podemos considerar a_3 uma ação proibitiva e sintetizar uma norma $n_1 = \varphi \xrightarrow{p} a_3$, com $\varphi \subseteq s_3$. Portanto, ao alcançarem o estado s_3 , a norma n_1 proíbe os agentes de partirem para o estado de conflito s_4 . Entretanto, ao obedecerem a norma n_1 , os agentes não conseguem alcançar os objetivos do sistema, visto que a_4 é a única ação permitida no estado s_3 e a sua execução leva os agentes a alcançarem o estado de timeout s_5 . Considerando que para alcançarem os objetivos do sistema os agentes necessitam transitar do estado s_1 para o estado s_6 , então, idealmente, a norma sintetizada deveria proibir a execução da ação a_1 , o que contribuiria para a ocorrência da história de sucesso h_3 .

Desta forma, podemos dizer que a síntese ingênua é falha, pois carece de uma análise sobre as histórias de conflito a fim de determinar ações que, em certos contextos, ao serem proibidas, evitam estados de conflitos e mantêm os objetivos do sistema alcançáveis. Para isso, uma abordagem alternativa é integrar à síntese de normas a *análise da alcançabilidade dos objetivos do sistema* a fim de auxiliar na identificação das ações que devem ser proibidas devidamente. Nesta abordagem, pode-se buscar por um estado a partir do qual exista um traço que finde em um estado de objetivos. Chamaremos tal traço e estado de *traço e estado viabilizadores*. Ao realizarmos *backtracking* a partir do estado final de uma história de conflito $h \in \mathcal{H}^{conf}$ em direção ao estado inicial podemos verificar para cada estado visitado, através da análise de alcançabilidade, a existência de um traço viabilizador. Caso um estado viabilizador s_v seja encontrado, então a ação a_p que parte de s_v na história de conflito pode ser considerada uma *ação proibitiva* mais adequada do que a ação obtida na abordagem ingênua. Ao ser proibida, a ação a_p evita que a história de conflito aconteça e, caso

optem, os agentes podem alcançar os objetivos do sistema ao executarem as ações do traço viabilizador. Chamaremos esta abordagem de *síntese com backtracking*.

Portanto, ao considerarmos um estado viabilizador s_v e uma ação proibitiva a_p , podemos sintetizar uma norma $n_2 = \varphi' \xrightarrow{p} a_p$, com $\varphi' \subseteq s_v$. Por outro lado, caso um estado viabilizador não seja encontrado, então não é possível sintetizar uma norma eficiente baseando-se na história analisada. A figura 10 ilustra a aplicação da síntese com backtracking sobre a história de conflito h_1 , onde as arestas tracejadas em vermelho representam a ordem de visita dos estados na busca por um estado viabilizador, enquanto a parte tracejada em lilás e azul na história h_3 representa um traço viabilizador identificado na análise de alcançabilidade dos objetivos do sistema.

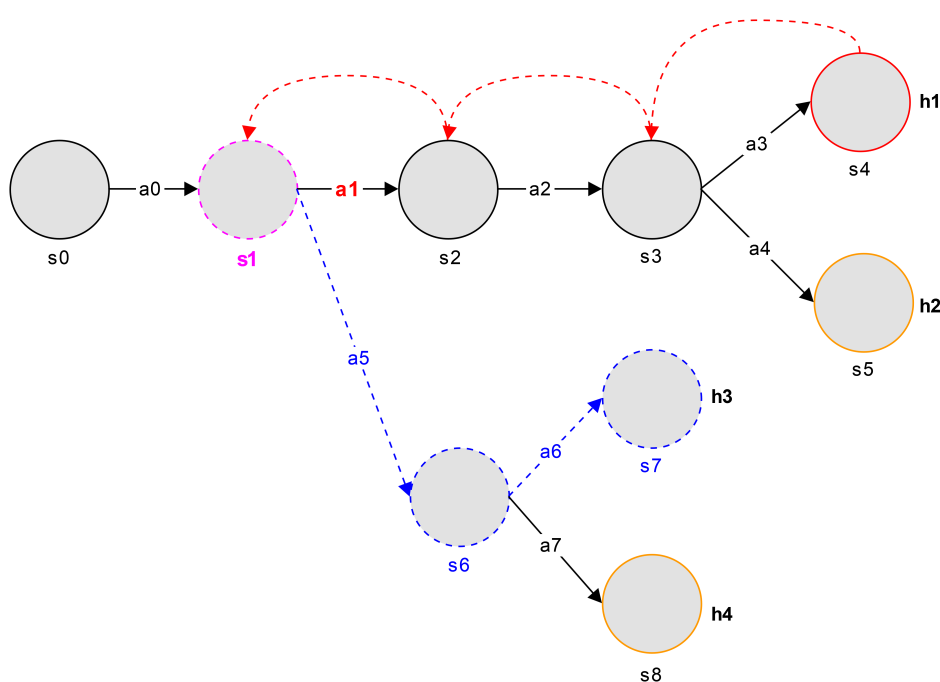


Figura 10 – Síntese com backtracking sobre a história h_1 .

Durante a busca, a análise da alcançabilidade identifica que a partir do estado s_1 existe um traço viabilizador $((s_0, s_1, a_5), (s_1, s_6, a_6), (s_6, s_7, null))$. Deste modo, s_1 pode ser considerado um estado viabilizador e a ação a_1 que parte de s_1 na história de conflito h_1 pode ser considerada uma ação proibitiva. A partir de tais elementos, uma norma $n_3 = \varphi'' \xrightarrow{p} a_1$, com $\varphi'' \subseteq s_1$, pode ser sintetizada. Diferentemente da norma n_1 , a norma n_3 é capaz de evitar conflitos e manter os objetivos do sistema alcançáveis. Como a síntese com backtracking indica um ponto apropriado onde podar as histórias de conflito, ao avançarmos a busca para outros estados e escolhermos outras ações para proibir podemos acabar podando mais histórias do que o desejável (diminuindo a autonomia dos agentes) ou inviabilizar o alcance dos objetivos do sistema. Na figura 10, por exemplo, ambas as situações podem se concretizar ao avançarmos o backtracking para o estado s_0 e escolhermos incorretamente a ação a_0 para ser proibida.

Ao que se pode observar, a síntese com backtracking é um caminho para a *síntese de normas eficientes*. Neste sentido, com o intuito de apoiar o desenvolvimento de uma proposta para a síntese de normas eficientes, apresentamos a seguir as definições formais de *alcançabilidade*, *estado* e *traço viabilizadores* e *análise da alcançabilidade de objetivos do sistema* no contexto desta tese.

Definição 31 (Alcançabilidade) *Dados dois estados s_i e s_j , com $\{s_i, s_j\} \subset \mathcal{S}$, dizemos que o estado s_i alcança s_j , denotado por $s_i \rightsquigarrow_t s_j$, se existir um traço $t \in \mathcal{T}$ a partir do qual é possível transitar de s_i para s_j , tal que, $t = ((s_k, s_i, a_r), \dots, (s_l, s_j, a_u))$, onde $s_k \in \mathcal{S} \cup \text{null}$, $a_u \in \mathcal{I}^A \cup \text{null}$, e \mathcal{S} , \mathcal{I}^A e \mathcal{T} estão introduzidos nas definições 2, 4 e 19, respectivamente.*

Exemplo 21 *Conforme a árvore ilustrada pela figura 10, tem-se que $s_0 \rightsquigarrow_t s_2$, pois a partir do traço $((\text{null}, s_0, a_0), (s_0, s_1, a_1), (s_1, s_2, a_2))$ é possível transitar de s_0 para s_2 .*

Definição 32 (Estado e Traço Viabilizadores) *Dado um conjunto de objetivos do sistema $\mathcal{G} \in \mathcal{G}^*$, um estado de objetivos $s_{\mathcal{G}} \in \mathcal{S}^{\mathcal{G}}$, uma história de conflito $h \in \mathcal{H}^{\text{conf}}$ e uma tripla $(s_j, s_v, a_p) \in h$, dizemos que s_v é um estado viabilizador se s_v for um estado livre de conflito e $s_v \rightsquigarrow_t s_{\mathcal{G}}$, com \mathcal{G} , \mathcal{G}^* , $\mathcal{S}^{\mathcal{G}}$ e \rightsquigarrow_t introduzidos nas definições 5, 5, 7 e 31, respectivamente. Nestas condições, o traço $t_v \in \mathcal{T}$ a partir do qual é possível transitar de s_v para $s_{\mathcal{G}}$ é dito ser um traço viabilizador.*

Definição 33 (Ação Proibitiva) *Dada uma história de conflito $h \in \mathcal{H}^{\text{conf}}$ e uma tripla $(s_j, s_v, a_p) \in h$, caso s_v seja um estado viabilizador, então a ação a_p é uma ação proibitiva, com s_v introduzido na definição 32.*

Uma norma que proíbe a execução de uma ação proibitiva em um estado de uma história de conflito, evita que tal história ocorra. Contudo, a partir do estado viabilizador, os agentes podem executar as ações do traço viabilizador e, assim, alcançar um estado de objetivos, o que faz com que uma história de sucesso ocorra.

Definição 34 (Análise de Alcançabilidade) *Dado um conjunto de objetivos do sistema $\mathcal{G} \in \mathcal{G}^*$, um estado de objetivos $s_{\mathcal{G}} \in \mathcal{S}^{\mathcal{G}}$ e uma história de conflito $h \in \mathcal{H}^{\text{conf}}$, a análise da alcançabilidade dos objetivos do sistema é dada pela função recursiva $\omega : (s_v, s_j, a_b) \rightarrow (s_j, s_v, a_p) \cup \mathcal{E}$ que determina um estado viabilizador s_v em h , com (s_v, s_j, a_b) e (s_j, s_v, a_p) triplas de h , \mathcal{E} um símbolo para representar a inexistência de solução e \mathcal{G} , \mathcal{G}^* , $\mathcal{S}^{\mathcal{G}}$ e \mathcal{H} introduzidos nas definições 5, 5, 7 e 20, respectivamente.*

$$\omega((s_v, s_j, a_b)) = \begin{cases} s_v, & \text{se } s_v \rightsquigarrow_t s_{\mathcal{G}} \\ \mathcal{E}, & \text{se } s_v = \text{null} \\ \omega((s_j, s_v, a_p)), & \text{caso contrário} \end{cases} \quad (1)$$

A função ω realiza backtracking sobre a história h . Assim, dada uma tripla $(s_v, s_j, a_b) \in h$, caso o estado pai s_v seja um estado viabilizador, então $s_v \rightsquigarrow_t s_G$ ⁹. Caso contrário, a função retrocede para a tripla onde s_v é um estado filho, isto é, a tripla antecessora $(s_j, s_v, a_p) \in h$. A recursão ocorre até que seja encontrado um estado viabilizador ou seja concluído que não há solução. Nesta situação, o backtracking alcança uma tripla $(null, s_0, a_b)$ e retorna \mathcal{E} .

Isto posto, considerando as definições introduzidas nesta seção, apresentamos a seguir a definição de síntese com backtracking.

Definição 35 (Síntese com Backtracking) *Dada uma história de conflito $h \in \mathcal{H}^{conf}$ e uma função de recursão ω , a síntese com backtracking consiste em: i) determinar um estado viabilizador $s_v = \omega((s_v, s_j, a_b))$, com $(s_v, s_j, a_b) \in h$; e ii) considerando $s_v \neq \mathcal{E}$ e $(s_j, s_v, a_p) \in h$, obter a ação proibitiva a_p , obter um contexto $\varphi \subseteq s_v$ e sintetizar uma norma $\varphi \xrightarrow{p} a_p$. Por sua vez, a função está apresentada na definição 34.*

Se $s_v = \mathcal{E}$, então não é possível sintetizar uma norma considerando a história h .

A fim de ilustrar os conceitos apresentados nesta seção de forma mais concreta, apresentamos a seguir os exemplos 22 e 23, onde normas são sintetizadas para evitar colisões entre os agentes e manter os objetivos do sistema alcançáveis para o cenário do tráfego de carros simplificado do exemplo 16. Para isto, os exemplos consideram as histórias do exemplo 17 da seção 3.2, onde as histórias formam árvores construídas de forma análoga às árvores ilustradas pelas figuras 5 e 6. Para cada exemplo, a primeira norma é sintetizada de acordo com a abordagem da *síntese ingênua*, enquanto a segunda norma é sintetizada de acordo com a abordagem da *síntese com backtracking*. Além do mais, assumimos que os contextos de tais normas são iguais aos estados nos quais as normas são aplicáveis.

Exemplo 22 *Considere a árvore ilustrada pela figura 11 constituída pelas histórias de sucesso h_2 e conflito h_3 .*

De acordo com a abordagem de síntese ingênua, a fim de que os agentes evitem o estado de conflito s_{11} , podemos sintetizar uma norma $n_4 = s_7 \xrightarrow{p} \text{move}(ag_1, h, i)$. Ao obedecerem a norma n_4 , os agentes evitam a história de conflito h_3 e podem fazer a história de sucesso h_2 acontecer.

Agora, considere a árvore ilustrada pela figura 12 constituída unicamente pela história de conflito h_3 , onde as arestas e estados tracejados em lilás e azul correspondem a um traço viabilizador.

⁹ Em uma história de conflito, um estado viabilizador s_v é, obrigatoriamente, um estado pai. Logo, a história de conflito é constituída pelas triplas (s_j, s_v, a_p) e (s_v, s_j, a_b) , com a primeira precedendo a segunda. Como no backtracking visitamos a história recursivamente do fim para o início, a tripla corrente pode conter um estado pai como um estado viabilizador. Neste sentido, a função ω verifica para cada tripla visitada, se o estado pai é um estado viabilizador.

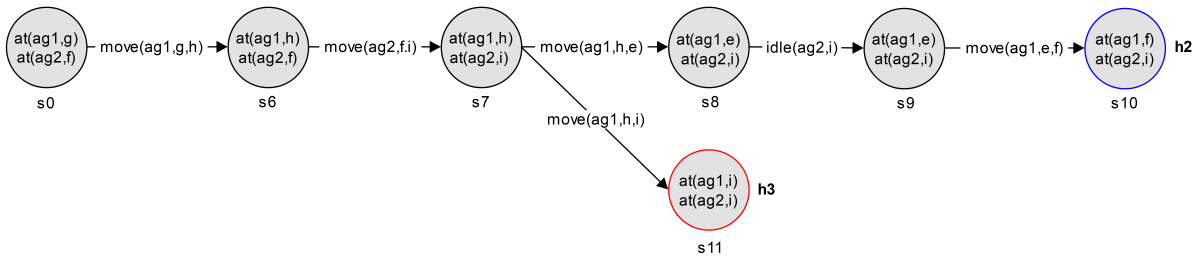


Figura 11 – Síntese ingênua sobre a história h_3 .

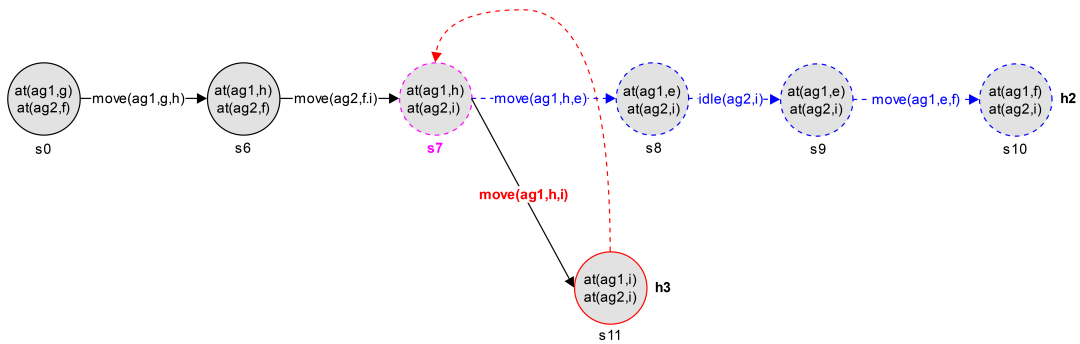


Figura 12 – Síntese com backtracking sobre a história h_3 .

De acordo com a síntese com backtracking, ao aplicarmos a função ω sobre a história h_3 a partir da tripla $(s_7, s_{11}, null)$, estaremos realizando backtracking do estado s_{11} para o s_7 e ao realizarmos a análise da alcançabilidade a partir do estado s_7 , que está destacado em lilás, identificaremos o traço viabilizador $((s_6, s_7, move(ag_1, h, e)), (s_7, s_8, idle(ag_2, i)), (s_8, s_9, move(ag_1, e, f)), (s_9, s_{10}, null))$. Assim, s_7 e $move(ag_1, h, i)$, que está destacada em vermelho, são considerados um estado viabilizador e uma ação proibitiva, respectivamente. Portanto, neste caso, a norma sintetizada pela síntese com backtracking é igual a norma sintetizada pela síntese ingênua. Isto ocorre porque o estado viabilizador s_7 alcança o estado de conflito s_{11} diretamente, sem passar por nenhum outro estado. Assim, ambas as normas possuem o mesmo contexto e a mesma ação proibitiva.

Exemplo 23 Considere a árvore ilustrada pela figura 13 constituída pelas histórias de sucesso h_2 , conflito h_4 e timeout h_5 .

De acordo com a abordagem da síntese ingênua, a fim de que os agentes evitem o estado de conflito s_{13} , podemos sintetizar uma norma $n_5 = s_{12} \xrightarrow{p} move(ag_1, e, f)$. Embora ao obedecerem a norma n_5 os agentes evitam a história de conflito h_4 , eles não podem alcançar os objetivos do sistema, pois a partir do estado s_{12} , os agentes conseguem concluir somente a história de timeout h_5 . Portanto, a norma sintetizada pela síntese ingênua não contribui para que ocorra uma história de sucesso.

Agora, considere a árvore ilustrada pela figura 14 constituída unicamente pela história de conflito h_4 , onde as arestas e estados tracejados em lilás e azul correspon-

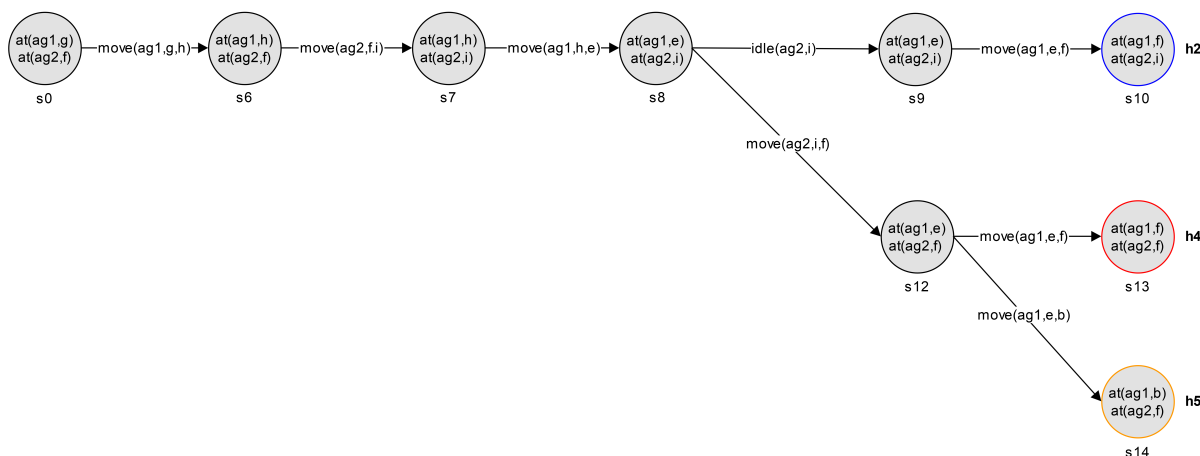


Figura 13 – Síntese ingênua sobre a história h_4 .

dem a um traço viabilizador.

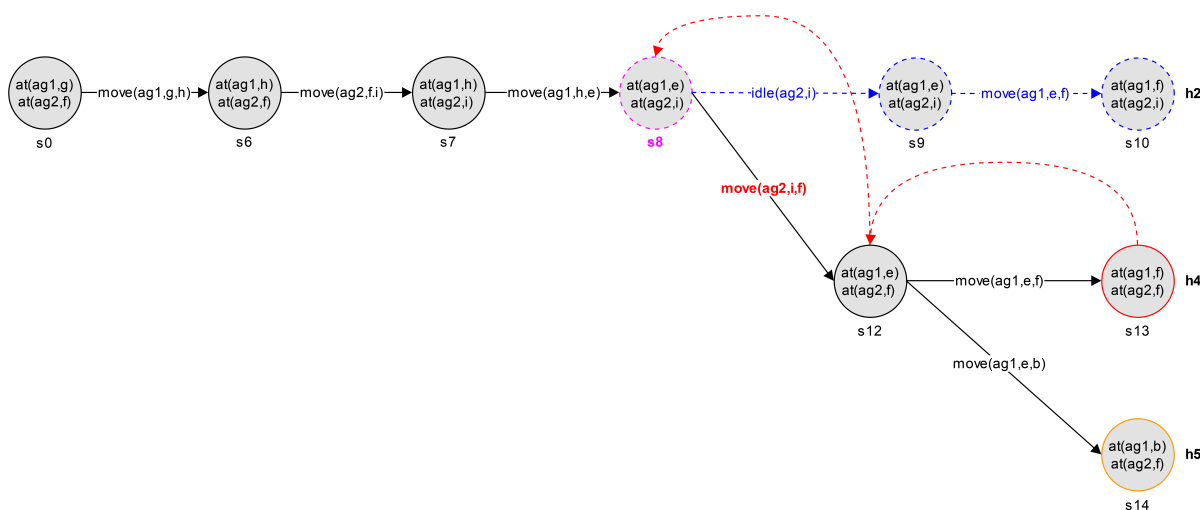


Figura 14 – Síntese com backtracking sobre a história h_4 .

De acordo com a síntese com backtracking, ao aplicarmos a função ω sobre a história h_4 a partir da tripla $(s_{12}, s_{13}, \text{null})$, estaremos realizando backtracking do estado s_{13} para o estado s_{12} e ao realizarmos a análise da alcançabilidade a partir do estado s_{12} , verificaremos que não existe um traço viabilizador. Contudo, ao partirmos do estado s_{12} para o estado s_8 , que está destacado em lilás, podemos identificar que a partir de s_8 existe o traço viabilizador $((s_7, s_8, \text{idle}(ag_2, i)), (s_8, s_9, \text{move}(ag_1, e, f)), (s_9, s_{10}, \text{null}))$. Assim, s_8 e $\text{move}(ag_2, i, f)$, que está destacada em vermelho, são considerados um estado viabilizador e uma ação proibitiva, respectivamente. Portanto, a norma sintetizada pela síntese com backtracking é $n_6 = s_8 \xrightarrow{p} \text{move}(ag_2, i, f)$. Diferentemente da norma n_5 , a norma n_6 evita a história de conflito h_4 e, dado que o traço viabilizador faz parte da história de sucesso h_2 , a norma n_6 contribui para que uma história de

sucesso ocorra.

De acordo com os exemplos 22 e 23, podemos observar que as normas obtidas pela síntese ingênua evitam apenas uma história de conflito, enquanto as normas obtidas pela síntese com backtracking tendem a evitar diferentes histórias de insucesso. Isso ocorre porque quanto mais estados existem entre um estado viabilizador e um estado de conflito, mais histórias podem existir, pois a partir de tais estados diferentes traços podem se ramificar. Embora evitar histórias de timeout não faça parte dos objetivos deste trabalho, podemos dizer que a não seleção de tais histórias é um efeito positivo, porém, não planejado da síntese com backtracking. A figura 15 ilustra a árvore do exemplo 17 da seção 3.2 com áreas de poda (destacadas por retângulos em lilás) ocasionadas pelas normas n_4 , do exemplo 22, n_5 e n_6 , do exemplo 23, e, adicionalmente, $n_7 = s_{19} \xrightarrow{p} \text{idle}(ag_2, f)$.

Em conjunto, tais normas podam da árvore as histórias de insucesso h_3 , h_4 , h_5 , h_7 e h_8 . Entretanto, à medida que o espaço das histórias cresce, novas normas podem ser necessárias para que histórias de insucesso sejam evitadas. Basicamente, isso ocorre porque tanto a síntese ingênua quanto a síntese com backtracking obtêm normas muito específicas que são aplicáveis em poucos estados do sistema. Este problema pode se agravar em cenários onde, para cada história de conflito, uma norma específica e distinta capaz de evitá-la e contribuir com a ocorrência de uma história de sucesso é necessária.

Neste contexto, uma pergunta que surge é: *como evitar o aumento excessivo do número de normas à medida que o espaço das histórias do sistema cresce?* Uma forma de responder à esta pergunta é através da *exploração da generalidade das normas*. Normas mais gerais subsumem normas mais específicas, o que as tornam capazes de serem aplicáveis nos estados onde as normas mais específicas são e evitarem as histórias que as normas mais específicas evitam. Assim, ao sintetizarmos normas mais gerais, abrangemos normas mais específicas e podemos diminuir a quantidade de normas que regulam os agentes. Além do mais, ao expandirmos o conjunto das histórias, é possível que as normas mais gerais sejam capazes de regular os agentes em novas histórias, pois a sua generalidade pode fazer com que sejam aplicáveis em estados de tais histórias. Neste sentido, podemos buscar sintetizar normas com generalidade balanceada, pois possivelmente são gerais o suficiente para subsumir normas mais específicas capazes de evitar histórias de conflito, contudo não muito gerais a ponto de subsumir uma quantidade de normas que impeça os agentes de alcançarem os objetivos do sistema.

Considere a norma $n_2 = \{at(Ag_2, L_2)\} \xrightarrow{p} \text{move}(Ag_1, L_1, L_2)$ do exemplo 17. Como n_2 generaliza as normas n_4 , n_5 , n_6 e n_8 , ela é aplicável nos estados onde dois agentes ocupam locais adjacentes. Logo, ela evita histórias de conflito onde os agentes colidem.

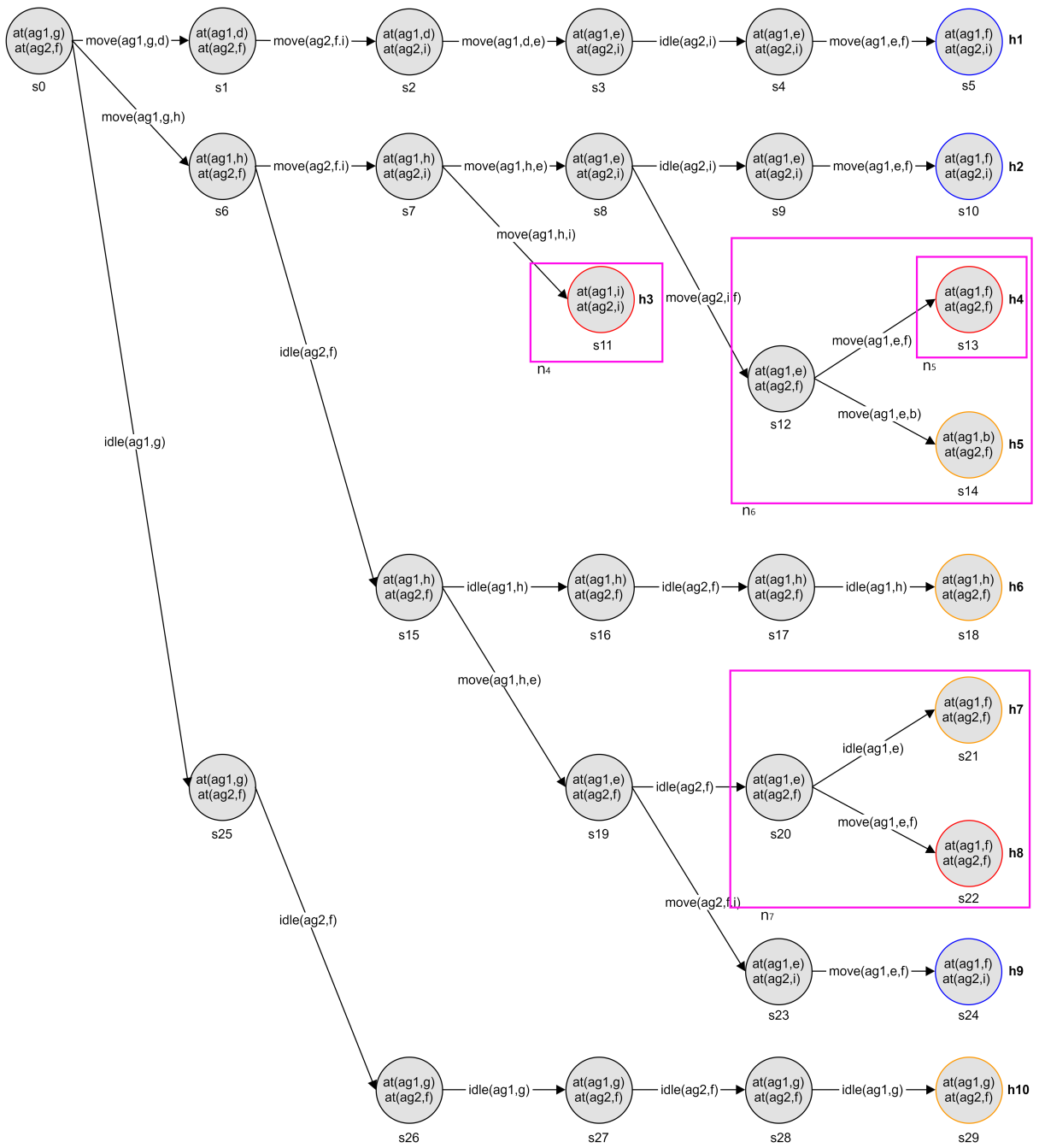


Figura 15 – Árvore do exemplo 17 com áreas de poda.

Portanto, a norma mais geral evita as histórias de conflito que as normas n_4 , n_5 , n_6 e n_8 evitam. Além disso, dependendo da generalidade, as normas podem ser aplicáveis em muitos estados proibindo os agentes de executarem ações necessárias para que os objetivos do sistema sejam alcançados. Considere agora a norma $n_3 = \{ \} \xrightarrow{p} move(Ag_1, L_1, L_2)$ também do exemplo 17. Por ser muito geral, n_3 é aplicável em todos os estados do sistema no cenário do tráfego de carros. Quando os agentes são regulados por tal norma, eles não se movem e, apesar de não ocorrerem conflitos, os objetivos do sistema não podem ser alcançados. Portanto, uma outra questão a ser

ponderada é: *qual deve ser a generalidade das normas para que sejam eficientes?*

A fim de investigarmos formas de responder a estas perguntas e aprimorarmos a abordagem da síntese com backtracking, nós provemos um estudo sobre a generalidade das normas no capítulo seguinte.

4 EXPLORANDO A GENERALIDADE DAS NORMAS

A generalidade das normas refere-se à extensão de comportamentos dos agentes que as normas são capazes de regular. Neste sentido, determinar a generalidade das normas é fundamental para que possamos sintetizar normas que mantêm um equilíbrio entre o controle que exercem e a autonomia dos agentes. Conforme discutido no capítulo 1, enquanto normas muito específicas regulam pouco os agentes dando-lhes uma grande liberdade para executarem as suas ações, o que tende a gerar conflitos, normas muito gerais regulam os agentes em excesso podendo reduzir significativamente a sua autonomia e inviabilizar o alcance dos objetivos do sistema. Sob estas circunstâncias, nós argumentamos que para serem eficientes, as normas necessitam ter uma generalidade balanceada, o que significa que as normas não devem ser muito específicas nem muito gerais. Neste contexto, nós consideramos que explorar a generalidade das normas é fundamental para a síntese de normas eficientes.

Neste capítulo, nós damos continuidade à construção do nosso ferramental teórico e formal para a síntese de normas eficientes, onde consideramos conceitualizar e formalizar a generalidade das normas. Para isto, na seção 4.1, nós discutimos que determinar a generalidade das normas é um problema inviável. Para contornarmos este problema, nós propomos uma forma alternativa de determinar a generalidade das normas através de uma abordagem sintática que nos permite obter sequências de normas ordenadas, aproximadamente, das mais específicas para as mais gerais. Na seção 4.2, nós realizamos um conjunto de experimentos com o objetivo de identificar a existência de regiões nas sequências de normas onde as normas eficientes possam estar localizadas. Por fim, na seção 4.3, nós discutimos o impacto da nossa abordagem sintática no posicionamento das normas eficientes nas sequências e como este conhecimento é particularmente útil para a construção de algoritmos para a síntese de normas eficientes.

4.1 CONCEITOS BÁSICOS

Conforme a definição 13, as normas regulam os agentes através dos estados em que são aplicáveis. Nesse sentido, a definição 36 conceitualiza a generalidade das normas no contexto desta tese.

Definição 36 (Generalidade das Normas) *Dada uma norma $n \in \mathcal{N}$, $Sa^n = \{s \in S \mid n \triangleright s\}$ é o conjunto de estados nos quais a norma n é aplicável, com S , n , \triangleright e \mathcal{N} introduzidos nas definições 2, 11, 13 e 15, respectivamente.*

Um outro conceito importante atrelado à generalidade das normas é o conceito de *norma mais geral*.

Definição 37 (Norma Mais Geral) Dadas duas normas distintas n_i e n_j , com $\{n_i, n_j\} \subset \mathcal{N}$, a norma n_i é mais geral que n_j , denotado por $n_i \succ_g n_j$, se $Sa^{n_i} \subset Sa^{n_j}$, $Sa^{n_i} \not\subset Sa^{n_j}$ e $a^{n_i} = a^{n_j}$, onde a refere-se a ação que as normas n_i e n_j proíbem e Sa refere-se ao conjunto de estados do sistema onde, conforme a definição 36, n_i e n_j são aplicáveis.

A relação *mais geral* que define uma ordem parcial irreflexiva, assimétrica e transitiva $\mathcal{R} = \{(n_j, n_i) \mid (n_j, n_i) \in \mathcal{N} \times \mathcal{N} \wedge n_i \succ_g n_j\}$ sobre o conjunto \mathcal{N} . Obter esta ordem é um problema que requer: i) construir o conjunto \mathcal{S} ; ii) construir o conjunto Sa^n para cada norma $n \in \mathcal{N}$; e iii) obter a relação \mathcal{R} . No entanto, esta é uma tarefa complexa e inviável, pois: \mathcal{S} é um conjunto possivelmente grande; obter um conjunto Sa^n necessita analisar todos os estados de \mathcal{S} ; e, obter a relação \mathcal{R} necessita comparar todas as normas de \mathcal{N} entre si, onde \mathcal{N} também é um conjunto potencialmente grande.

Para lidarmos com estes problemas, nós propomos determinar a generalidade das normas através de uma abordagem sintática, onde os elementos utilizados para descrever as normas são considerados. Isto nos permite estimar uma generalidade aproximada das normas de forma rápida, embora potencialmente falha, mas que viabiliza uma investigação sobre as diferentes generalidades das normas e como elas afetam a regulação dos agentes. Desta forma, nós almejamos identificar as generalidades das normas que possam ser consideradas balanceadas e com este conhecimento sintetizar normas eficientes. Para isto, nós apresentamos a seguir as definições que fundamentam a nossa abordagem sintática para explorar a generalidade das normas.

Definição 38 (Valor de Generalidade) O valor de generalidade de uma norma $n \in \mathcal{N}$ é um número real $k \in \mathbb{R}_+$ (reais não negativos) que representa a generalidade de n em termos absolutos, com \mathcal{N} introduzido na definição 15.

Para calcular o valor de generalidade das normas, nós propomos o conceito de *função de generalidade*.

Definição 39 (Função de Generalidade) Uma função de generalidade $f : \mathcal{N} \rightarrow \mathbb{R}_+$ é uma função que atribui um valor de generalidade $f(n)$ a uma norma $n \in \mathcal{N}$ de acordo com as características sintáticas da norma, com \mathcal{N} introduzido na definição 15.

De acordo com a definição 11, as características sintáticas de uma norma são os *predicados*, *variáveis*, *fórmulas atômicas*, *operador deôntico* e *ação*. Enquanto algumas características não impactam na generalidade das normas, como as ações e o operador deôntico, a ocorrência de certas características, como as *fórmulas atômicas* e *variáveis*, podem influenciar na extensão de estados onde as normas são aplicáveis. Considere, por exemplo, as normas $n_1 = \{at(Ag_2, L_2)\} \xrightarrow{p} move(Ag_1, L_1, L_2)$ e $n_2 = \{\} \xrightarrow{p} move(Ag_1, L_1, L_2)$, tal que, $n_2 \succ_g n_1$. A norma n_1 tem 1 fórmula atômica e 4 variáveis, enquanto a norma n_2 tem 0 fórmulas atômicas e 3 variáveis. Embora este

seja um exemplo simples, ele mostra que quanto mais específica uma norma é, mais detalhada ela necessita ser, o que resulta em uma ocorrência maior das características sintáticas em sua especificação. Por outro lado, quanto mais geral uma norma é, menos detalhada ela necessita ser, o que resulta em uma ocorrência menor das características sintáticas que a descreve. Logo, podemos observar que a generalidade das normas é *inversamente proporcional* à ocorrência de tais características sintáticas. Assim, à medida que a ocorrência das *fórmulas atômicas* e *variáveis* diminui, a generalidade das normas aumenta. Neste contexto, para calcular o valor de generalidade das normas, uma função de generalidade pode considerar a ocorrência das fórmulas atômicas e variáveis e, para isto, definimos adicionalmente o conceito de *função auxiliar*.

Uma função auxiliar fornece a ocorrência de elementos distintos de uma certa característica sintática. Como o número de ações e operadores deônticos é sempre 1, não definimos uma função específica para contá-los. Nós definimos duas funções auxiliares, *formulas* e *variaveis*. A função *formulas* conta a ocorrência de fórmulas atômicas das normas em seus contextos de ativação, enquanto a função *variaveis* conta a ocorrência de variáveis na especificação das normas.

Definição 40 (Função formulas) A função *formulas*: $\mathcal{N} \rightarrow \mathbb{Z}_+$ (inteiros não negativos) fornece a ocorrência de fórmulas atômicas de uma norma $n \in \mathcal{N}$, tal que, $formulas(n) = |\varphi^n|$, com \mathcal{N} introduzido na definição 15.

Definição 41 (Função variaveis) A função *variaveis*: $\mathcal{N} \rightarrow \mathbb{Z}_+$ (inteiros não negativos) fornece a ocorrência de variáveis de uma norma $n \in \mathcal{N}$, tal que, $variaveis(n) = |(\bigcup_{p \in \varphi^n \wedge t \in \text{Termos}^p} t) \cup \text{Par}^{a^n}|$, onde Termos^p refere-se ao conjunto de termos de uma fórmula atômica $p \in \mathcal{L}'$, com Par^{a^n} e \mathcal{N} introduzidos nas definições 3 e 15, respectivamente.

As funções auxiliares podem ser combinadas de diferentes formas para construir diferentes funções de generalidade. Portanto, o valor de generalidade das normas pode variar de acordo com as funções de generalidade. Além disso, certas normas podem ser mapeadas ao mesmo valor de generalidade quando possuem a mesma ocorrência das características sintáticas exploradas. Considerando que uma norma mais geral deve ter um valor de generalidade maior do que uma norma mais específica, observamos a seguinte propriedade desejável das funções de generalidade.

Propriedade 1 (Função de Generalidade Crescente) Uma função de generalidade f é dita ser crescente se $\forall n_i \in \mathcal{N} \forall n_j \in \mathcal{N} : n_i \underset{g}{>} n_j \rightarrow f(n_i) > f(n_j)$, com \mathcal{N} e f introduzidos nas definições 15 e 39, respectivamente.

Definição 42 (Conjunto de Generalidade) Dada uma função de generalidade f , um conjunto de generalidade $G_k = \{n \in \mathcal{N} \mid f(n) = k\}$ é um conjunto constituído pelas

normas cujos valores de generalidade são k , com k e f introduzidos nas definições 38 e 39, respectivamente.

Definição 43 (Sequência de Conjuntos de Generalidade) *Dado um conjunto de conjuntos de generalidade $G^G = \{G_i, G_j, \dots, G_m\}$, dizemos que $O = (G_i, G_j, \dots, G_m)$ é uma sequência crescente de conjuntos de generalidade do conjunto G^G se $i < j < \dots < m$. Dizemos também que G^{min} , G^{cen} e G^{max} são, respectivamente, os conjuntos de generalidade de limite inferior, meio e superior de O se $G^{min} = G_i$, $G^{cen} = G_r$ e $G^{max} = G_m$ com r sendo a mediana de (i, j, \dots, m) . Além do mais, chamados de metade inferior e metade superior de uma sequência O os intervalos $[G^{min}, G^{cen}]$ e $[G^{cen}, G^{max}]$, respectivamente, com $G \in G^G$ introduzido na definição 42.*

A atribuição de valores de generalidade às normas resulta em uma sequência de conjuntos de generalidades crescente conforme a generalidade das normas. Por meio de tais conjuntos, as normas estão ordenadas, aproximadamente, das mais específicas para as mais gerais de modo que, à medida que as normas se tornam mais gerais, a ocorrência de características sintáticas exploradas pelas funções diminui, enquanto o valor de generalidade das normas aumenta. Além disto, o número de conjuntos de generalidade nas sequências resultantes, o tamanho de tais conjuntos e os seus valores de generalidade k variam de acordo com as funções de generalidade.

Definição 44 (Conjuntos de Generalidade Contíguos) *Dados um conjunto de conjuntos de generalidade G^G e uma sequência crescente O de G^G , dizemos que dois conjuntos de generalidade G_i e G_j , com $\{G_i, G_j\} \subset G^G$, são contíguos em O , representado por $G_i * G_j$, se $i < j$ e não existe outro conjunto de generalidades $G_k \in G^G$, tal que, $i < k < j$, com G^G e O introduzidos na definição 43.*

Definição 45 (Normas Contíguas) *Dadas duas normas, n_i e n_j , com $\{n_i, n_j\} \subset \mathcal{N}$, dizemos que n_i e n_j são normas contíguas, representado por $n_i * n_j$, se $n_i \underset{g}{>} n_j$ e não existe outra norma $n_k \in \mathcal{N}$, tal que, $n_i \underset{g}{>} n_k \underset{g}{>} n_j$.*

Dependendo da função de generalidade, normas contíguas são mapeadas para conjuntos de generalidade não contíguos. Tais funções distribuem as normas em um número maior de conjuntos de generalidade, obtendo, então, sequências maiores. Neste contexto, observamos a seguinte propriedade das funções de generalidade sobre o mapeamento de normas contíguas em conjuntos de generalidade.

Propriedade 2 (Função de Generalidade Contígua) *Uma função de generalidade f é dita ser contígua se $\forall n_i \in \mathcal{N} \forall n_j \in \mathcal{N} : n_i * n_j \rightarrow G_{k_1} * G_{k_2}$, onde $k_1 = f(n_i)$, $k_2 = f(n_j)$, $n_i \in G_{k_1}$ e $n_j \in G_{k_2}$, com \mathcal{N} e f introduzidos nas definições 15 e 39, respectivamente.*

Exemplo 24 Considere dois conjuntos de conjuntos de generalidade, $G_1^G = \{G_{0.22}, G_{0.50}, G_{0.80}\}$ e $G_2^G = \{G_{0.18}, G_{0.33}, G_{0.61}, G_{0.97}\}$, obtidos, respectivamente, pela aplicação de duas funções de generalidade, f_1 , que explora as fórmulas atômicas das normas, e f_2 , que explora as variáveis das normas, sobre um espaço de normas $\mathcal{N} = \{n_1, \dots, n_7\}$. Considere também duas seqüências de conjuntos de generalidade, O_1 e O_2 , obtidas, respectivamente, a partir dos conjuntos G_1^G e G_2^G . A figura 16 ilustra tais seqüências, onde os conjuntos de generalidade estão representados por elipses verticais cujos identificadores estão em azul e a ocorrência das características sintáticas exploradas pelas funções de generalidade estão descritas no topo das elipses. Além disso, a relação mais geral entre as normas está representada como um grafo, onde os vértices são normas e as arestas conectam normas contíguas partindo das normas mais específicas para as normas mais gerais.

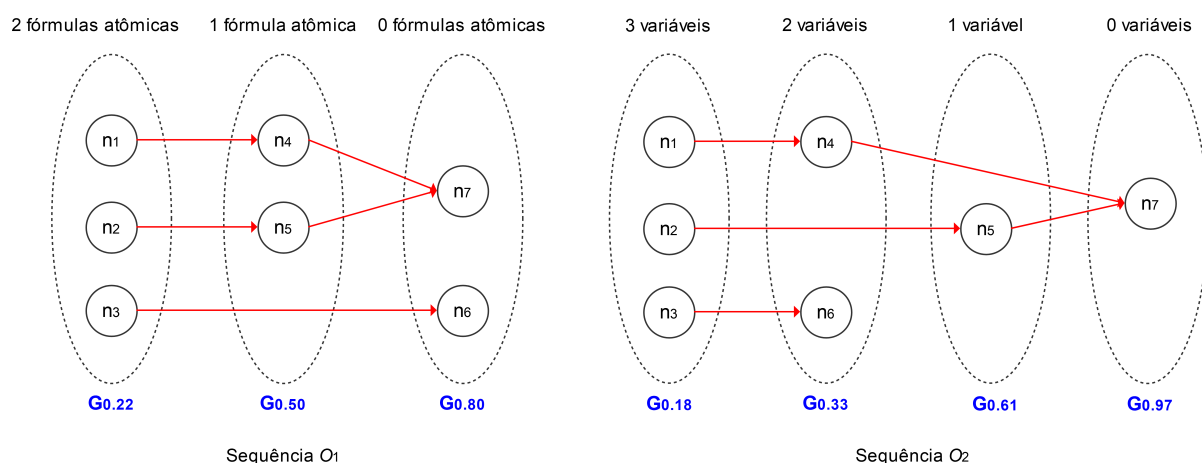


Figura 16 – Exemplo de duas seqüências de conjuntos de generalidade.

Embora as relações de generalidade representadas nas seqüências O_1 e O_2 sejam as mesmas, tais seqüências são constituídas por conjuntos distintos. Enquanto a seqüência O_1 é menor e composta por 3 conjuntos, a seqüência O_2 é composta por 4 conjuntos. Contudo, em ambas seqüências, à medida que a generalidade das normas cresce, a ocorrência das características sintáticas exploradas diminui. Considerando a seqüência O_1 , tem-se $G^{min} = G_{0.22}$, $G_{cen} = G_{0.50}$ e $G_{max} = G_{0.80}$, onde $G_{0.22} * G_{0.50}$, $G_{0.50} * G_{0.80}$ e $\neg(G_{0.22} * G_{0.80})$. Com exceção das normas n_3 e n_6 , as demais normas contíguas são mapeadas para conjuntos de generalidades contíguas. Por outro lado, considerando a seqüência O_2 , tem-se $G^{min} = G_{0.18}$, $G_{cen} = G_{0.47}$ e $G_{max} = G_{0.97}$, onde o valor de generalidade do conjunto G_{cen} é a mediana dos valores de generalidade dos conjuntos de G_2^G . Com exceção das normas n_2 , n_4 , n_5 e n_7 , as demais normas contíguas são mapeadas para conjuntos de generalidade contíguas.

Para fins de discussão, faremos distinção entre dois tipos de seqüências de normas: *aproximadas* e *semânticas*. Seqüências aproximadas seguem a definição 43,

onde as normas estão ordenadas através dos conjuntos de generalidade. Por outro lado, em uma sequência semântica, as normas estão ordenadas segundo o conjunto \mathcal{R} (derivado a partir da definição 37) que estabelece, através de seus pares, relações de ordem entre as normas. Portanto, ao seguirmos as relações em \mathcal{R} podemos obter uma sequência de normas, onde a ordenação segue a generalidade das normas. Para visualizarmos uma sequência semântica, podemos considerar construir o diagrama de Hasse (ROSEN, 2012) de \mathcal{R} . O diagrama de Hasse é um tipo de grafo onde podemos considerar os vértices como as normas e as arestas como as relações de ordem entre as normas, tal que, à medida que os níveis de profundidade do grafo aumentam, as normas tornam-se mais gerais. Podemos, então, agrupar as normas em conjuntos de acordo com o nível de profundidade em que estão situadas e construir a sequência semântica como uma sequência crescente destes conjuntos, do conjunto que agrupa as normas do nível 0 ao conjunto que agrupa as normas do último nível.

Desta forma, obtém-se uma sequência onde as normas também estão ordenadas através de conjuntos, contudo, cada norma está corretamente posicionada na sequência. Assim, normas contíguas fazem parte de conjuntos contíguos na sequência semântica enquanto em uma sequência aproximada tais normas podem fazer parte de conjuntos não contíguos estando, então, incorretamente posicionadas. Logo, podemos dizer que as sequências aproximadas são aproximações das sequências semânticas. Além disso, podemos dizer que a hipótese 1 está formulada sobre sequências semânticas, pois postula sobre as normas eficientes estarem posicionadas em torno do meio de espaços cujas normas estão ordenadas conforme a sua generalidade. Portanto, é possível que a posição de tais normas em sequências aproximadas destoem.

Exemplo 25 *A figura 17 ilustra a sequência semântica O_3 para as normas do exemplo 24, onde o diagrama de Hasse representa, assim como no exemplo anterior, a relação mais geral entre normas contíguas. A sequência é formada por três conjuntos, G_1 , G_2 e G_3 , tal que, se $n_i * n_j$, então as normas n_i e n_j estão posicionadas em conjuntos contíguos G_p e G_q , com $n_i \in G_p$, $n_j \in G_q$ e $p > q$.*

Como as normas estão agrupadas de acordo com o nível de profundidade em que estão situadas na grafo de Hasse, a posição das normas nos conjuntos da sequência semântica O_3 diverge um pouco da posição que ocupam nos conjuntos das sequências aproximadas O_1 e O_2 do exemplo 24.

Neste ponto, nós introduzimos as definições essenciais para criarmos funções de generalidade. Uma função de generalidade combina as funções auxiliares para explorar uma ou duas características sintáticas das normas para obter sequências aproximadas. Como existem diferentes formas de combinar tais funções, identificar uma combinação que viabilize a construção de boas aproximações é um desafio. Com o intuito de explorar diferentes possibilidades para tal fim, nós apresentamos as funções

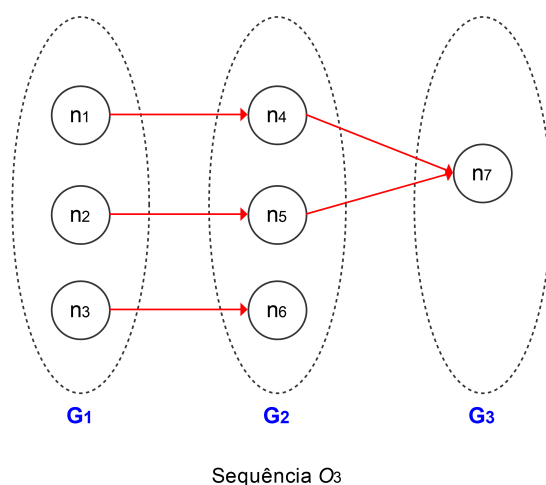


Figura 17 – Exemplo de uma sequência semântica.

de generalidade NGI , NGJ , NGK e NGF introduzidas nas definições 46, 47, 48 e 49, respectivamente. Para deixar mais claro como o espaço das normas é organizado através dessas funções, nós apresentamos primeiramente um espaço de normas, conforme o exemplo 26, e à medida que as funções são introduzidas, nós mostramos as sequências aproximadas resultantes. Além disto, o esquema da representação das sequências é semelhante ao esquema apresentado no exemplo 23. Contudo, seguimos Morales et al. (2013) para, de uma forma sintática, determinar quando uma norma generaliza outra. Embora falha, nesta forma, uma norma $n_i \in \mathcal{N}$ generaliza uma norma $n_j \in \mathcal{N}$ se $\varphi^{n_i} \subset \varphi^{n_j}$ e $a^{n_i} = a^{n_j}$.

Além disto, o esquema da representação de tais sequências é semelhante ao esquema apresentado no exemplo 24.

Exemplo 26 Para os próximos quatro exemplos, considere que $\mathcal{N} = \{n_1, \dots, n_{10}\}$ é o espaço das normas para o cenário do tráfego de carros simplificado, tal que:

Definição 46 (Função NGI) A função de generalidade $NGI: \mathcal{N} \rightarrow]0,1]$ explora o número de fórmulas atômicas de uma norma, de acordo com a definição 40, para atribuir-lhe um valor de generalidade como segue :

$$NGI(n) = \frac{1}{1 + \text{formulas}(n)} \quad (2)$$

Na sequência aproximada, à medida que os valores de generalidade aumentam, a ocorrência de fórmulas atômicas diminui. Portanto, quanto menos fórmulas atômicas uma norma tiver, maior será seu valor de generalidade. A sequência começa com as normas que possuem o maior número de fórmulas atômicas e termina com as normas que possuem o menor número desta característica sintática.

$$\begin{aligned}
 n_1 &= \{at(Ag_2, L_2), next(L_1, L_2)\} \xrightarrow{p} move(Ag_1, L_1, L_2) \\
 n_2 &= \{at(Ag_2, L_2), waiting(Ag_1, L_2)\} \xrightarrow{p} move(Ag_1, L_1, L_2) \\
 n_3 &= \{next(L_1, L_2), waiting(Ag_1, L_2)\} \xrightarrow{p} move(Ag_1, L_1, L_2) \\
 n_4 &= \{at(Ag_2, L_2)\} \xrightarrow{p} move(Ag_1, L_1, L_2) \\
 n_5 &= \{next(L_1, L_2)\} \xrightarrow{p} move(Ag_1, L_1, L_2) \\
 n_6 &= \{waiting(Ag_1)\} \xrightarrow{p} move(Ag_1, L_1, L_2) \\
 n_7 &= \{\} \xrightarrow{p} move(Ag_1, L_1, L_2) \\
 n_8 &= \{at(Ag_2, L_2), next(L_1, L_2)\} \xrightarrow{p} idle(Ag_1, L_1) \\
 n_9 &= \{at(Ag_1, L_1)\} \xrightarrow{p} idle(Ag_1, L_1) \\
 n_{10} &= \{\} \xrightarrow{p} idle(Ag_1, L_1)
 \end{aligned}$$

Exemplo 27 A figura 18 ilustra a seqüência aproximada O_1 obtida pela aplicação da função de generalidade NGI sobre o espaço \mathcal{N} .

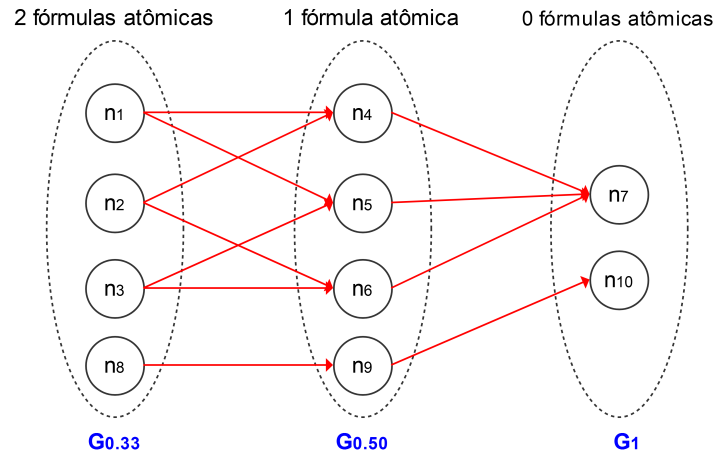


Figura 18 – Seqüência aproximada O_1 obtida pela função NGI.

A seqüência O_1 é constituída por três conjuntos de generalidades, $G_{0.33}$, $G_{0.50}$ e G_1 , cujas normas possuem 2, 1 e 0 fórmulas atômicas, respectivamente. Além disso, tem-se $G^{min} = G_{0.33}$, $G^{cen} = G_{0.50}$ e $G^{max} = G_1$. De acordo com a função NGI, as normas mais específicas de \mathcal{N} são aquelas que formam o conjunto $G_{0.33}$, enquanto as normas mais gerais são aquelas que formam o conjunto G_1 .

Definição 47 (Função NGJ) A função de generalidade NGJ: $\mathcal{N} \rightarrow]0,1]$ explora o número de variáveis de uma norma, de acordo com a definição 41, para atribuir-lhe um valor de generalidade da seguinte forma:

$$NGJ(n) = \left\{ \frac{1}{1 + \text{variaveis}(n)} \right. \quad (3)$$

Na sequência aproximada, à medida que os valores de generalidade aumentam, a ocorrência de variáveis diminui. Portanto, quanto menos variáveis uma norma tiver, maior será seu valor de generalidade. Assim, a sequência começa com as normas que possuem o maior número de variáveis e termina com as normas que possuem o menor número desta característica sintática.

Exemplo 28 A figura 19 ilustra a sequência aproximada O_2 obtida pela aplicação da função de generalidade NGJ sobre o espaço \mathcal{N} .

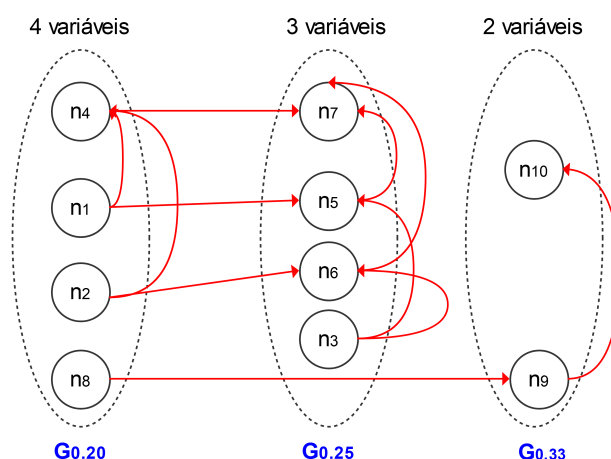


Figura 19 – Sequência aproximada O_2 obtida pela função NGJ.

A sequência O_2 é constituída por 3 conjuntos de generalidades, $G_{0.20}$, $G_{0.25}$ e $G_{0.33}$, cujas normas possuem 4, 3 e 2 variáveis, respectivamente. Além disto, tem-se $G^{min} = G_{0.20}$, $G^{cen} = G_{0.25}$ e $G^{max} = G_{0.33}$. Observa-se que normas contíguas que possuem o mesmo número de variáveis são mapeadas ao mesmo conjunto de generalidade. Isto ocorre com as normas das relações $n_1 * n_4$ e $n_2 * n_4$, que fazem parte do conjunto $G_{0.20}$, e $n_3 * n_5$ e $n_3 * n_6$, que fazem parte do conjunto $G_{0.25}$. Como existem relações de generalidade onde a norma mais específica e geral são atribuídas a um mesmo valor de generalidade, a função NGJ não apresenta a propriedade desejável 1. Além disto, de acordo com a função NGJ, as normas mais específicas de \mathcal{N} são aquelas mapeadas para o conjunto $G_{0.20}$, enquanto as normas mais gerais são aquelas mapeadas para o conjunto $G_{0.33}$.

Definição 48 (Função NGK) A função de generalidade NGK: $\mathcal{N} \times \{w\} \rightarrow]0, w]$, onde $w \in \mathbb{R}_+^*$ (reais positivos), explora o número de fórmulas atômicas e variáveis de uma norma, de acordo com as definições 40 e 41, respectivamente, para atribuir-lhe um valor de generalidade da seguinte forma:

$$NGK(n, w) = \frac{1}{\text{formulas}(n) + \frac{1 + \text{variaveis}(n)}{w}} \quad (4)$$

Na sequência aproximada, à medida que os valores de generalidade aumentam, tanto a ocorrência de fórmulas atômicas quanto de variáveis diminui. Portanto, quanto menos fórmulas atômicas e variáveis uma norma tiver, maior será seu valor de generalidade. Assim, a sequência começa com as normas que possuem o maior número de fórmulas atômicas e variáveis e termina com as normas que possuem o menor número destas características sintáticas. Além disto, em relação às normas com a mesma ocorrência de fórmulas atômicas, o parâmetro w auxilia a ordenarmos tais normas de forma decrescente de acordo com o seu número de variáveis.

Exemplo 29 A figura 20 ilustra a sequência aproximada O_3 obtida pela aplicação da função de generalidade NGK sobre o espaço \mathcal{N} , considerando $w = 10$.

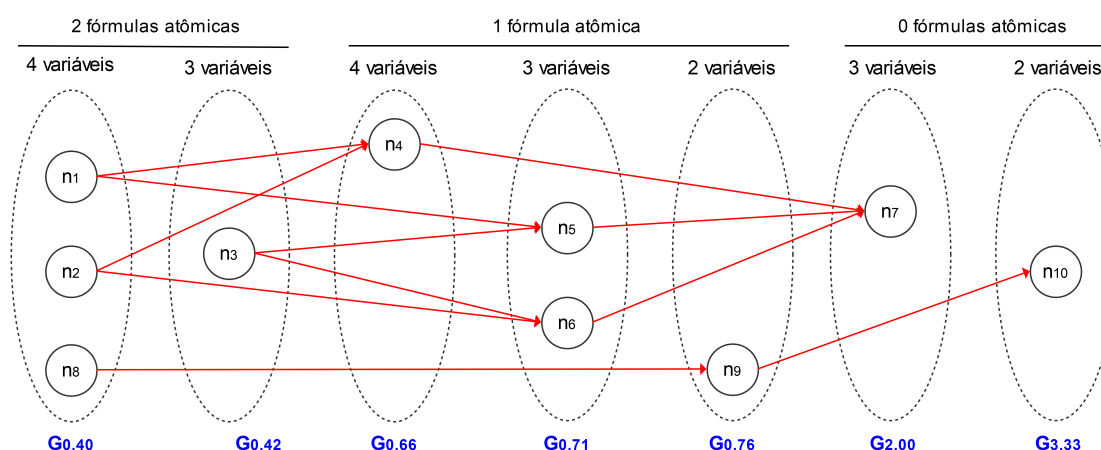


Figura 20 – Sequência aproximada O_3 obtida pela função NGK.

A sequência O_3 é maior que as sequências anteriores sendo constituída por 7 conjuntos de generalidade, $G_{0.40}$, $G_{0.42}$, $G_{0.66}$, $G_{0.71}$, $G_{0.76}$, $G_{2.00}$, e $G_{3.33}$. Tais conjuntos, por sua vez, são em média menores do que os conjuntos das sequências O_1 e O_2 . Além disto, tem-se $G^{min} = G_{0.40}$, $G^{cen} = G_{0.71}$ e $G^{max} = G_{3.33}$. Na sequência O_3 , o número de fórmulas atômicas diminui de 2 até 0, enquanto o número de variáveis das normas com o mesmo número de fórmulas atômicas pode diminuir de 4 até 2. De acordo com a função NGK, as normas mais específicas de \mathcal{N} são aquelas do conjunto $G_{0.40}$, enquanto as normas mais gerais são aquelas do conjunto $G_{3.33}$.

Definição 49 (Função NGF) A função de generalidade NGF: $\mathcal{N} \rightarrow]0,1 + |\text{Var}]$, onde Var corresponde ao conjunto de variáveis da linguagem \mathcal{L} (conforme a definição 1), também explora o número de fórmulas atômicas e variáveis de uma norma, de acordo com as definições 40 e 41, respectivamente, para atribuir-lhe um valor de generalidade, todavia, da seguinte forma:

$$NGF(n) = \frac{1}{\text{formulas}(n) + \frac{1}{1 + \text{variaveis}(n)}} \tag{5}$$

Esta função funciona de forma semelhante à função NGK. No entanto, à medida que os valores de generalidade aumentam, a ocorrência de fórmulas atômicas diminui enquanto a ocorrência de variáveis aumenta. Isto resulta da inversão da segunda fração no denominador da fórmula da função NGK, onde o peso w é fixado em 1, o que deriva a função NGF. Assim, a sequência aproximada inicia com as normas que possuem o maior número de fórmulas atômicas e o menor número de variáveis e termina com as normas que possuem o menor número de fórmulas atômicas e o maior número de variáveis. Dado que o maior número de variáveis que uma norma pode ter é $|Vars|$, o maior valor de generalidade que pode ser atribuído a uma norma é $1 + |Vars|$ que ocorre quando $variaveis(n)$ é igual a $|Vars|$.

Exemplo 30 A figura 21 ilustra a sequência aproximada O_4 obtida pela aplicação da função de generalidade NGF sobre o espaço \mathcal{N} .

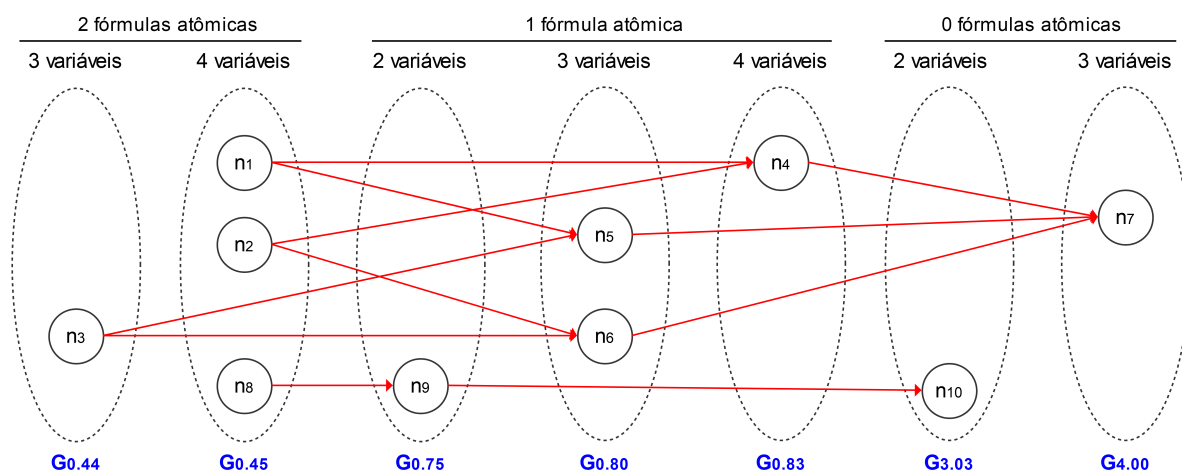


Figura 21 – Sequência aproximada O_4 obtida pela função NGF.

A diferença entre as sequências O_3 e O_4 é que na sequência O_4 as normas que possuem a mesma ocorrência de fórmulas atômicas são ordenadas crescentemente de acordo com o seu número de variáveis. A sequência O_4 também é constituída por 7 conjuntos de generalidade, $G_{0.44}$, $G_{0.45}$, $G_{0.75}$, $G_{0.80}$, $G_{0.83}$, $G_{3.03}$ e $G_{4.00}$, que também são em média menores do que os conjuntos das sequências obtidas pelas funções NGI e NGJ. Além disto, tem-se $G^{min} = G_{0.44}$, $G^{cen} = G_{0.80}$ e $G^{max} = G_{4.00}$. De acordo com a função NGF, as normas mais específicas de \mathcal{N} são as normas do conjunto $G_{0.44}$, enquanto as normas mais gerais são as normas do conjunto $G_{4.00}$.

A tabela 2 compara as funções de generalidade nos exemplos 27, 28, 29 e 30, conforme três critérios: i) função de generalidade crescente (sim ou não - conforme a propriedade 1); ii) função de generalidade contígua (sim ou não - conforme a propriedade 2); e iii) número de conjuntos de generalidade (3 ou 7).

Critério	NGI	NGJ	NGK	NGF
Função de generalidade crescente	sim	não	sim	sim
Função de generalidade contígua	sim	não	não	não
Número de conjuntos de generalidade	3	3	7	7

Tabela 2 – Comparativo entre as funções de generalidade.

As funções NGI, NGK e NGF mapeiam normas contíguas a conjuntos de generalidade distintos, onde a norma mais geral possui um valor de generalidade maior do que o valor de generalidade da norma mais específica. Desta forma, observa-se que tais funções apresentam a propriedade 1. Como a função NGI ordena as normas de acordo com o seu número de fórmulas atômicas e as normas contíguas possuem uma fórmula atômica de diferença, então a função NGI mapeia normas contíguas a conjuntos contíguos. Portanto, observamos que a função NGI apresenta a propriedade 2. Por outro lado, as funções NGK e NGF mapeiam diferentes normas contíguas a conjuntos de generalidade não contíguos. Como tais funções discriminam as normas pelo número de fórmulas atômicas e variáveis, elas criam sequências maiores distribuindo as normas em mais conjuntos de generalidade. Desta forma, normas contíguas são mapeadas a conjuntos que podem estar afastados por outros conjuntos de generalidade. Logo, as funções NGK e NGF não apresentam a propriedade 2.

Por sua vez, a função NGJ mapeia certas normas contíguas a um mesmo conjunto de generalidade. Como tal função ordena as normas conforme o número de variáveis e existem normas contíguas que possuem a mesma ocorrência dessa característica sintática, para tais normas, a função NGJ lhes atribui o mesmo valor de generalidade. Logo, a função NGJ não apresenta as propriedades 1 e 2. Podemos observar também que as funções que exploram duas características sintáticas das normas permitem mais formas de agrupar as normas em conjuntos de generalidade do que as funções que exploram uma de tais características. Enquanto as sequências obtidas a partir das funções NGK e NGF possuem 7 conjuntos, as sequências obtidas a partir das funções NGJ e NGI possuem 3 conjuntos.

Por fim, apresentamos a definição de *conjunto de generalidade eficiente*.

Definição 50 (Conjunto de Generalidade Eficiente) *Dado um conjunto de generalidades G_k e uma norma eficiente $n \in \mathcal{N}$, chamamos G_k de conjunto de generalidade eficiente, denotado por G^{apex} , se n for uma norma eficiente e $n \in G_k$, com \mathcal{N} , n e G_k introduzidos nas definições 15, 27 e 43, respectivamente.*

Exemplo 31 *Considere a norma $n_4 = \{at(Ag_2, L_2)\} \xrightarrow{\rho} move(Ag_1, L_1, L_2)$ do exemplo 26. Conforme visto anteriormente no exemplo 16, n_4 é uma norma eficiente. Portanto, os conjuntos G^{apex} das sequências aproximadas O_1 , O_2 , O_3 e O_4 são, respectivamente, os conjuntos de generalidade $G_{0,50}$, $G_{0,20}$, $G_{0,66}$ e $G_{0,83}$, dado que eles contêm n_4 .*

4.2 EXPERIMENTOS

Nesta seção, nós estamos interessados em identificar a existência de regiões nas sequências aproximadas onde as normas eficientes possam estar localizadas e com este conhecimento construir algoritmos para a síntese de normas eficientes. Todavia, para determinarmos se uma norma é eficiente, é necessário verificar se, pelo menos, uma história de sucesso e nenhuma história de conflito possam ocorrer quando a norma regula os agentes. Contudo, a quantidade de histórias do sistema é um conjunto que pode explodir combinatoriamente, o que pode tornar a análise das normas eficientes inviável. Para contornarmos este problema, podemos fazer o uso do conceito das normas *quase eficientes* (conforme a definição 29), o que relaxa a necessidade de que nenhuma história de conflito possa ocorrer quando o sistema executa com tais normas, embora seja desejável que esta possibilidade seja pequena ou nula. Assim, ao considerarmos um subconjunto das histórias do sistema, a norma que obtém a *maior taxa de sucesso e a menor taxa de conflito* (conforme a definição 30) pode ser considerada um ótimo local (a melhor norma quase eficiente). Portanto, buscamos identificar a posição dos conjuntos G^{apex} nas sequências aproximadas através do encontro de normas quase eficientes.

Nesta direção, nós pretendemos verificar se a hipótese 1, que postula sobre a localização das normas eficientes em espaços de normas ordenados, é válida sobre as sequências aproximadas. Todavia, conforme discutido na seção anterior, esta hipótese está formulada sobre as sequências semânticas, logo, a localização das normas eficientes nas sequências aproximadas pode destoar. Como fazemos uso das funções de generalidade para ordenar as normas ao explorarem as suas características sintáticas, uma pergunta que surge é: *como as diferentes atribuições sintáticas impactam no posicionamento das normas eficientes nas sequências aproximadas?* A resposta para estas perguntas é importante, pois ela pode impactar na construção dos algoritmos para a síntese de normas eficientes.

Para contemplarmos os objetivos desta seção e responder à questão anterior, nós desenvolvemos um simulador de SMAn em Java para avaliar o desempenho das normas em evitar conflitos e manter os objetivos do sistema alcançáveis. Para isso, o simulador recebe como entrada um modelo de SMAn que define o SMAn para um determinado domínio. Este modelo inclui informações sobre os agentes, seus objetivos, os objetivos do sistema, o ambiente em que os agentes operam, etc. Inicialmente, o simulador constrói um espaço de normas proibitivas \mathcal{N} para, então, simular o sistema para cada tupla $(s_0, \mathcal{G}, c, n) \in (S_0 \times \mathcal{G}^* \times \mathcal{C} \times \mathcal{N})$, com S_0 , \mathcal{G}^* , \mathcal{N} e \mathcal{C} introduzidos nas definições 2, 7, 8 e 15, respectivamente. Desta forma, o subconjunto de histórias que consideramos para avaliar as normas é constituído por histórias que ocorrem em tempo de execução para diferentes estados iniciais e objetivos do sistema.

Durante a execução do simulador, à medida que a tupla corrente muda, o com-

portamento dos agentes tende a variar e histórias distintas ocorrer. Isso nos permite avaliar as normas com mais propriedade, pois a variabilidade de histórias que acontece pode ser maior¹. A simulação, por sua vez, pode resultar em: (i) história de sucesso; (ii) história de conflito; ou (iii) história de timeout.

Posteriormente, o simulador organiza o espaço das normas obtendo sequências aproximadas através das funções de generalidade NGI, NGJ, NGK e NGF, introduzidas nas definições 46, 47, 48 e 49, respectivamente. Para fins de comparação, nós estabelecemos três regiões das sequências aproximadas, a saber, região G^{min} , G^{cen} e G^{max} . Enquanto a primeira região inclui o conjunto G^{min} e o conjunto sucessor, a segunda inclui o conjunto G^{cen} e os conjuntos em seu entorno e a terceira inclui o conjunto G^{max} e o conjunto predecessor. Por fim, o simulador sumariza os resultados das suas execuções. Para isso, para cada conjunto de generalidade de uma sequência aproximada, o simulador analisa as histórias ocorridas através da regulação das normas que compõem o conjunto para, então, obter a taxa (porcentagem) de histórias de sucesso, histórias de conflito e histórias de timeout para aquele conjunto.

Considerando que a quantidade de estados iniciais e objetivos do SMAn pode ser grande, nós empregamos sobre os conjuntos \mathcal{S}_0 e \mathcal{G}^* uma amostragem sistemática (KHAN, 2020) com nível de confiança de 95% e margem de erro de 5%. Como o espaço das normas também é um conjunto potencialmente grande, nós empregamos sobre \mathcal{N} um filtro que mantém no espaço apenas as normas mínimas (conforme a definição 15). Além disso, definimos que o parâmetro w da função NGK é 10.

Os experimentos são realizados sobre três cenários e, para cada cenário, apresentamos quatro gráficos, onde cada gráfico é construído considerando uma sequência aproximada. Assim, dada uma sequência aproximada (G_i, G_j, \dots, G_m) , o respectivo gráfico tem como domínio (i, j, \dots, m) , contradomínio o intervalo $[0, 100]$ e imagem as taxas de histórias de sucesso (curvas azuis), conflito (curvas vermelhas) e timeout (curvas amarelas)². Nós inserimos uma linha vertical pontilhada no meio do eixo-x de cada gráfico para evidenciar a *distância* (em número de conjuntos) entre os conjuntos G^{cen} e G^{apex} . Além disso, identificamos à esquerda do eixo y a função de generalidade que obtém a sequência aproximada do respectivo gráfico. Por fim, os experimentos foram realizados em um computador com processador Intel Core i7-1165G7 de 16 GB de RAM, rodando a 2,8 GHz e com sistema operacional Ubuntu 22.10 de 64 bits.

¹ Embora as normas possam selecionar diferentes histórias, ao simularmos o SMAn distintas vezes com a mesma tupla, é possível que ocorra sempre a mesma história. Devido à escolha racional dos agentes, é possível que eles escolham a mesma sequência de ações para a tupla em questão.

² Como a taxa (porcentagem) de um tipo de história é a proporção de histórias daquele tipo em relação a quantidade total de histórias, então, o contradomínio é o intervalo $[0, 100]$.

4.2.1 Cenário do Tráfego de Carros

O cenário do tráfego de carros que implementamos, ilustrado pela figura 22, é uma adaptação de (CHRISTELIS, G.; ROVATSOS, 2009). Na grid, representada como um grafo direcionado, os vértices denotam *locais* ou *túneis*, enquanto as arestas denotam *caminhos* que conectam tais vértices. A grid possui 2 túneis, t_1 e t_2 , que são representados por quadrados azuis (cujas entradas e saídas são os lados dos quadrados onde chegam e partem as arestas, respectivamente), e 7 locais, de a a g , que são representados por círculos cinzas. Um conflito, por sua vez, é caracterizado por uma colisão que ocorre entre diferentes agentes em um dado vértice da grid (conforme exemplo 6). Além disto, existem 5 agentes móveis na grid, de ag_1 a ag_5 , que pretendem chegar aos seus destinos executando seus trajetos sem colidirem. Os agentes podem escolher *moverem-se* entre vértices adjacentes ou *manterem-se parados* nos locais onde estão. Ao entrarem em um túnel, é requerido aos agentes saírem dele no instante seguinte e não é possível aos agentes permanecerem parados em túneis nem haver mais de um agente em um túnel.

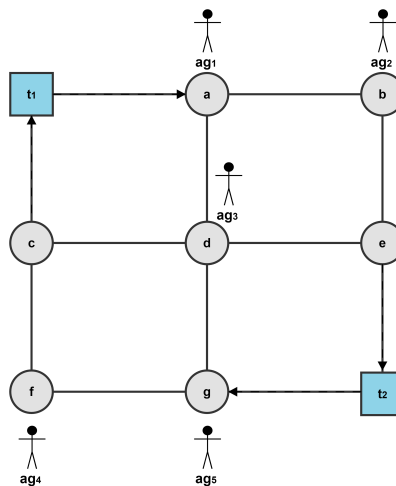


Figura 22 – Grid 3 × 3 do cenário do tráfego de carros com túneis.

O conjunto S_0 é constituído por estados que denotam os locais de partida dos agentes, enquanto o conjunto \mathcal{G}^* é constituído por objetivos do sistema que denotam os destinos dos agentes (túneis não são considerados pontos de partida nem de chegada). Os agentes, por sua vez, fazem uso da *distância de Manhattan* para estimar o melhor plano e alcançar os objetivos do sistema. Considerando *Vertices* o conjunto dos vértices da grid e *Agentes* o conjunto de agentes móveis sobre a grid, estipulamos que cada agente tem $2^{|Vertices|}$ etapas para concluir o seu trajeto, logo, o timeout do sistema é $|Agentes| * 2^{|Vertices|}$ passos. Além do mais, tem-se que $|S_0| = 225$, $|\mathcal{G}^*| = 225$ e $|\mathcal{N}| = 14$. Assim, o simulador de SMA foi executado 708.705 vezes.

Os resultados deste cenário estão ilustrados pela figura 23.

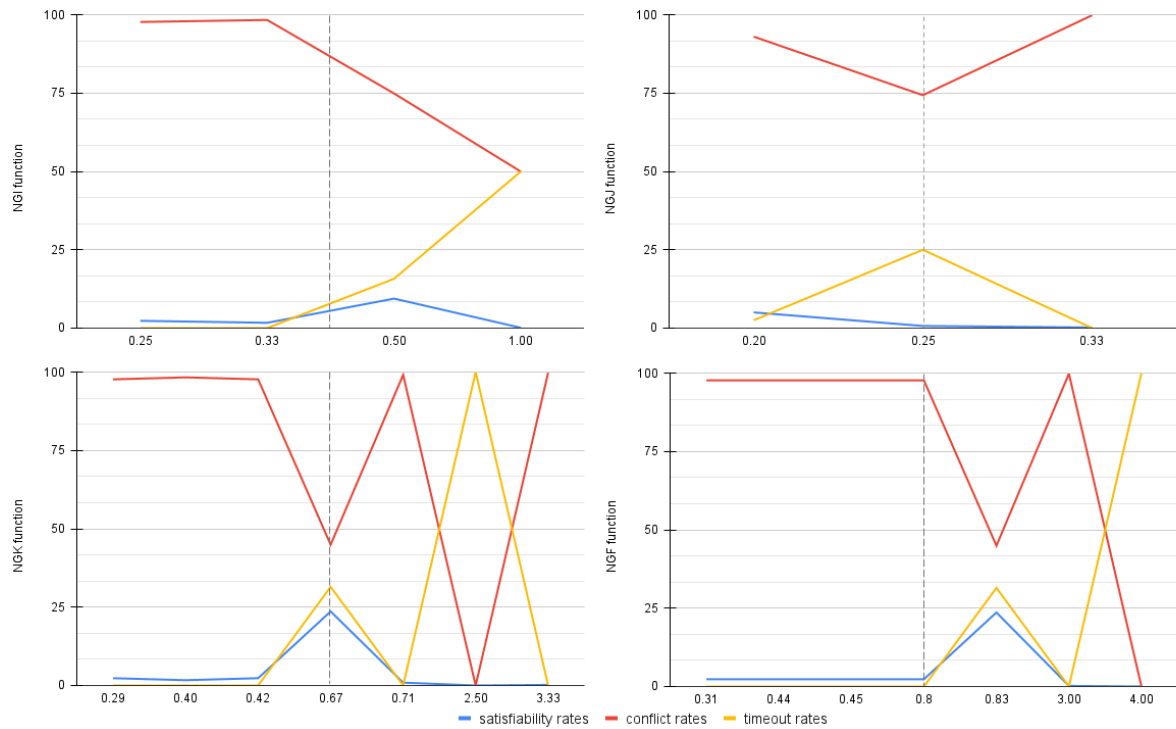


Figura 23 – Resultados do cenário do tráfego de carros com túnel.

De modo geral, as taxas de histórias de conflito são muito maiores do que as taxas de histórias de sucesso para a maioria dos conjuntos de generalidade devido à incapacidade da maioria das normas em evitar colisões. No entanto, a diferença entre estas taxas diminui nos conjuntos de generalidade $G_{0.50}$, $G_{0.20}$, $G_{0.67}$ e $G_{0.83}$ das funções NGI, NGJ, NGK e NGF, respectivamente. Basicamente, isto ocorre porque tais conjuntos são constituídos pela norma $n = \{at(Ag_2, L_2)\} \xrightarrow{p} move(Ag_1, L_1, L_2)$. Como os agentes possuem tempo suficiente para finalizar os seus planos e, conforme descreve o exemplo 9, colisões não ocorrem quando os agentes são regulados por tal norma, as taxas de histórias de sucesso dos conjuntos anteriores são impactadas positivamente pelo número de execuções do simulador que findam em histórias de sucesso. A norma n apresenta taxas de histórias de sucesso, conflito e timeout de 90%, 0% e 10%³, respectivamente. Como não observamos outras normas com taxas de histórias de sucesso e conflito, respectivamente, maiores ou iguais às taxas da norma n , consideramos n a melhor norma quase eficiente deste cenário. Como histórias de conflitos não ocorrem quando o simulador executa com a norma n e apenas 10% das execuções findam em histórias de timeout, consideramos n uma norma eficiente. Neste contexto, os conjuntos $G_{0.50}$, $G_{0.20}$, $G_{0.67}$ e $G_{0.83}$ correspondem, conforme a definição 50, aos conjuntos G^{apex} das sequências a quais pertencem. No entanto, os conjuntos G^{apex}

³ Em certas execuções do simulador, ocorre de um agente ficar trancado em um vértice de um canto da grid quando outros agentes estão ocupando os vértices adjacentes e não lhe dão passagem. Logo, nestas situações, ocorrem histórias de timeout. Contudo, dado o estado inicial e os objetivos do sistema considerados em tais execuções, é provável que os agentes possam satisfazer os objetivos do sistema ao escolherem ações alternativas percorrendo, então, outros caminhos na grid.

são constituídos também por normas não eficientes e, conseqüentemente, as suas taxas de histórias de sucesso são afetadas negativamente pelo mal desempenho do simulador quando os agentes são regulados por elas. Logo, as taxas de histórias de sucesso dos conjuntos G^{apex} não alcançam 100%.

Além disto, como as funções NGK e NGF obtêm as maiores seqüências aproximadas, os seus conjuntos de generalidade são em média menores. Assim, os seus conjuntos G^{apex} apresentam taxas de histórias de sucesso maiores do que os conjuntos G^{apex} das demais seqüências, pois menos histórias de insucesso são contabilizadas no cálculo das taxas. Por outro lado, a função NGJ obtém a seqüência cujo conjunto G^{apex} possui a menor taxa de histórias de sucesso. Isto ocorre porque a maioria das normas estão em G^{apex} , o que contribui para que a taxa seja afetada negativamente pelas grande quantidade de normas não eficientes que o constitui. As taxas de histórias de timeout crescem a partir da região G^{cen} alcançando os maiores valores na região G^{max} . Isto ocorre porque, à medida que as normas tornam-se mais gerais, certas normas passam a proibir a execução de ações necessárias para o alcance dos objetivos do sistema.

Ao analisarmos as posições dos conjuntos G^{apex} nos eixos-x, percebemos que existe uma região onde tais conjuntos são encontrados: a região G^{cen} . A fim de verificarmos se os conjuntos G^{apex} tendem a ser encontrados nesta região e, assim, validarmos a hipótese 1, nós aprofundamos a nossa investigação nas próximas seções explorando cenários maiores e mais complexos.

4.2.2 Cenário da Segurança de Acesso

Uma organização compartilha com os seus funcionários um conjunto de documentos que são classificados como *públicos* ou *confidenciais*. Os documentos estão disponíveis em um repositório de um servidor local, acessível a partir de qualquer computador da rede interna da organização. Então, para abrir um determinado documento, os funcionários devem fornecer as suas credenciais, que podem ser *baixas* ou *altas* (conforme as funções que desempenham na organização), para acessarem o repositório. No entanto, devido a questões de proteção de dados e privacidade (como roubo e vazamento de informações), espera-se que apenas funcionários com credenciais altas abram documentos confidenciais (exceto as normas, outros mecanismos para evitar violações de dados não são considerados). Assim, quando um funcionário com credencial baixa abre um documento confidencial, um conflito ocorre. Embora a abertura de documentos confidenciais por funcionários com credencial baixa não faça parte dos objetivos do sistema, isto pode fazer parte dos objetivos individuais dos agentes.

Essencialmente, o comportamento dos funcionários consiste em entrar no repositório, tentar abrir os documentos que lhes são atribuídos pelos objetivos do sistema ou aqueles relacionados aos seus objetivos individuais (quando existem) e, então, sair

do repositório. Nós definimos que 30% dos funcionários que possuem credencial baixa possuem objetivos individuais não coerentes com os objetivos do sistema (i.e., eles pretendem abrir documentos confidenciais). Por sua vez, a cada turno de execução do sistema, tais funcionários decidem com 50% de chance qual tipo de objetivo (individual ou global) pretendem satisfazer. O conjunto S_0 é constituído por estados iniciais que variam as classificações dos funcionários e dos documentos. Além do mais, em todos os estados iniciais, os funcionários estão desconectados do repositório. Por outro lado, o conjunto \mathcal{G}^* é constituído por objetivos do sistema que variam os documentos que os funcionários devem abrir. Além disto, estes objetivos especificam que, após abrir os documentos determinados, os funcionários devem efetuar a saída do repositório.

Cada funcionário tem $2 + (Docs \times Chance)$ passos para alcançar os objetivos do sistema, onde *Docs* denota o número de documentos que um funcionário deve acessar, *Chance* denota um incremento percentual no número de tentativas de abertura dos documentos e 2 representa as ações de entrada e saída do repositório. Definimos também que $Docs = 3$, $Chance = 1,5$ (aumento de 50%) e que existem 30 funcionários na organização e 30 documentos no repositório. Considerando que *Funcionarios* refere-se ao conjunto dos funcionários, então o timeout do sistema é $|Funcionarios| \times [2 + (Docs \times Chance)]$. Além do mais, tem-se que $|S_0| = 245$, $|\mathcal{G}^*| = 245$ e $|\mathcal{N}| = 388$. Assim, o simulador foi executado $2,3 \times 10^7$ vezes.

Os resultados deste cenário estão ilustrados pela figura 24.

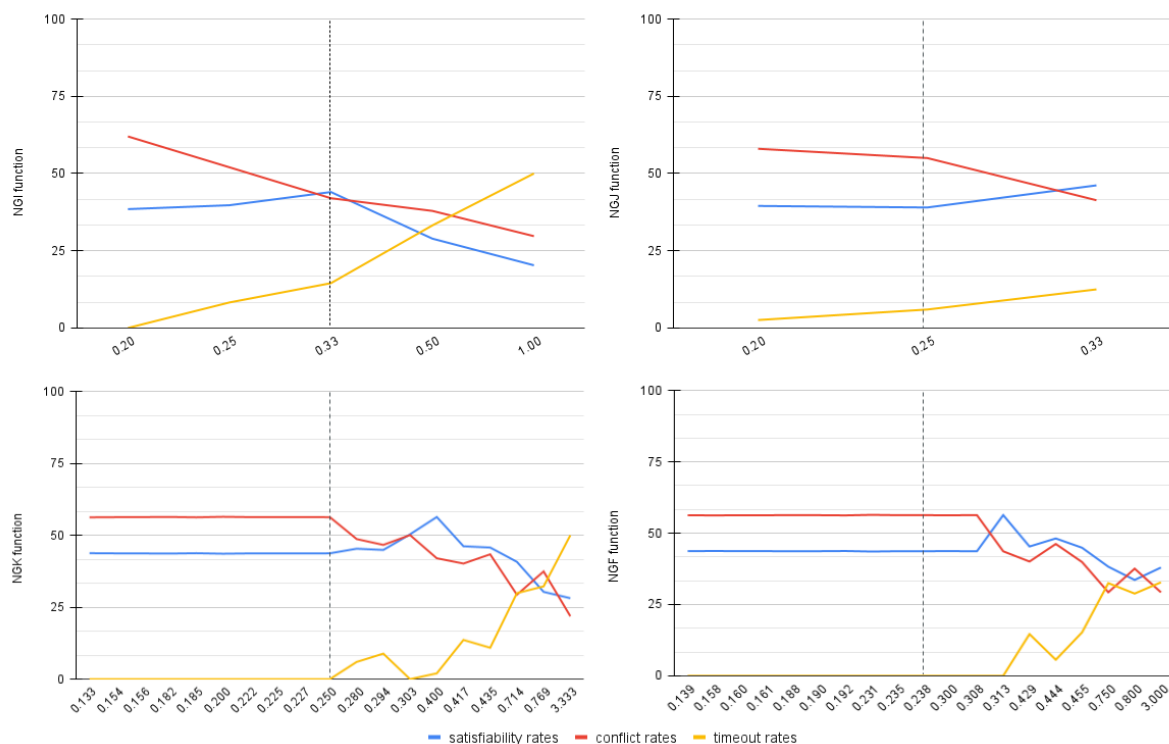


Figura 24 – Resultados do cenário da segurança de acesso.

Neste cenário, podemos verificar que os conjuntos $G_{0.33}$, $G_{0.33}$, $G_{0.400}$ e $G_{0.429}$

das funções NGI, NGJ, NGK e NGF, respectivamente, possuem as maiores taxas de sucesso das sequências a quais pertencem. Isto ocorre porque tais conjuntos são constituídos pela norma $n' = \{lowCredential(Ag_1), confidencial(Doc_1)\} \xrightarrow{p} open(Ag_1, Doc_1)$, que proíbe um funcionário Ag_1 abrir um documento Doc_1 se a sua credencial for baixa e Doc_1 for confidencial. Ao obedecerem a norma n' , os agentes com credencial baixa não ocasionam conflitos e como há tempo hábil para que os objetivos do sistema sejam alcançados, os conjuntos anteriores possuem as suas taxas de sucesso impactadas positivamente pelo bom desempenho do simulador ao executar com tal norma. A norma n' apresenta taxas de histórias de sucesso, conflito e timeout de 100%, 0% e 0%, respectivamente. Como não observamos outras normas com as mesmas taxas de histórias de sucesso e conflito de n' , consideramos n' a melhor norma quase eficiente deste cenário. Como histórias de conflitos não ocorrem quando o simulador executa com tal norma e 100% das execuções do simulador findam em histórias de sucesso, consideramos n' uma norma eficiente. Neste contexto, os conjuntos $G_{0.33}$, $G_{0.33}$, $G_{0.400}$ e $G_{0.429}$ correspondem aos conjuntos G^{apex} das sequências que pertencem. No entanto, da mesma forma como ocorre no cenário anterior, os conjuntos G^{apex} são constituídos também por normas não eficientes e, conseqüentemente, as suas taxas de sucesso são afetadas negativamente pelo mal desempenho do sistema quando os agentes são regulados por elas. Logo, as taxas de sucesso dos conjuntos G^{apex} também não alcançam 100%.

Em todas as sequências, as taxas de histórias de conflito são maiores que as taxas de histórias de sucesso para a maioria dos conjuntos de generalidade. As taxas de histórias de sucesso alcançam os maiores valores nos conjuntos G^{apex} sendo eles maiores nas sequências obtidas pelas funções NGK e NGF do que nas sequências obtidas pelas funções NGI e NGJ. Além disto, a sequência obtida pela função NGJ apresenta o conjunto G^{apex} com a menor taxa de histórias de sucesso. Nas sequências obtidas pela função NGI, a taxa de histórias de timeout cresce a partir da região G^{min} , enquanto nas sequências obtidas pelas demais funções, ela cresce a partir da região G^{cen} e, para todas as sequências, ela alcança o maior valor na região G^{max} .

Ao analisarmos a posição dos conjuntos G^{apex} nos eixos-x, podemos visualizar que existem, basicamente, duas regiões onde tais conjuntos são encontrados. Nas sequências obtidas pelas funções NGI e NGJ, o conjunto G^{apex} faz parte da região G^{cen} (embora também faça parte da região G^{max} no sequência obtida pela função NGJ). Por outro lado, nas sequências obtidas pelas funções NGK e NGF, os conjuntos G^{apex} estão localizados entre as regiões G^{cen} e G^{max} .

4.2.3 Cenário do Atendimento Médico

Um atendimento ambulatorial é oferecido a um conjunto de pacientes que procuram atendimento médico. O atendimento é prestado em uma clínica médica constituída

de uma sala de espera e um consultório. A sala de espera contém um painel de senha e três fileiras de quatro cadeiras conectadas. Devido à COVID-19 (VINDEGAARD; BENROS, 2020) e políticas de distanciamento social, quando dois pacientes sentam-se um ao lado do outro, uma situação de conflito ocorre (consideramos que os pacientes estão usando máscaras faciais e respeitando os demais protocolos sanitários). O comportamento dos pacientes consiste em entrar na sala de espera (onde recebem uma senha de atendimento automaticamente), escolher uma cadeira para sentar e aguardar a consulta médica. A escolha das cadeiras é aleatória e, caso a norma proíba o paciente de sentar, ele pode optar por aguardar a consulta em pé dentro da sala de espera ou fora da clínica. Devido ao medo de contaminação pela COVID-19 em ambientes pequenos e fechados, os pacientes têm preferência de 80% pela segunda opção.

O painel mostra uma nova senha em um intervalo t e o paciente que possui a senha corrente entra no consultório médico e o deixa no instante seguinte. Considerando que *Pacientes* é o conjunto de pacientes, definimos que o timeout do sistema é $|Pacientes| \times t + 4$, onde o valor 4 representa o número de ações necessárias para que os objetivos do sistema possam ser alcançados quando o último paciente está sentado em uma cadeira (i.e., ele necessita levantar da cadeira, entrar e sair do consultório médico e sair da clínica). Definimos também que $t = 2 \times |Pacientes|$ e $|Pacientes| = 12$. O conjunto S_0 é constituído por um estado inicial que especifica que os pacientes estão fora da clínica, enquanto o conjunto \mathcal{G}^* é constituído por um conjunto de objetivos do sistema que especifica que os pacientes devem receber atendimento médico e partir da clínica (para evitar aglomerações no seu interior). Além disto, este cenário possui 6150 normas. Portanto, o sistema foi executado 6150 vezes. A figura 25 ilustra os resultados deste cenário.

Neste cenário⁴, os conjuntos $G_{0.333}$, $G_{0.200}$, $G_{0.400}$ e $G_{0.455}$ das funções NGI, NGJ, NGK e NGF, respectivamente, possuem as maiores taxas de histórias de sucesso das sequências a quais pertencem. Isto ocorre porque tais conjuntos são constituídos pela norma $n'' = \{seated(Ag_2, C_2), next(C_2, C_1)\} \xrightarrow{p} sit(Ag_1, C_1)$, que proíbe um paciente Ag_1 de sentar em uma cadeira C_1 se, além de estar desocupada, ela for conectada a uma cadeira C_2 onde está sentado um paciente Ag_2 . Logo, ao obedecerem a norma n'' , os pacientes não ocasionam conflitos, pois não sentam em cadeiras adjacentes quando uma delas está ocupada e, como há tempo hábil para que os objetivos do sistema sejam alcançados, os conjuntos anteriores possuem as suas taxas de histórias de sucesso impactadas positivamente pelo bom desempenho do sistema ao executar com a norma n'' . Por sua vez, a norma n'' apresenta taxas de histórias de sucesso, conflito e timeout de 100%, 0% e 0%, respectivamente.

Como não observamos outras normas com as mesmas taxas de histórias de

⁴ As sequências obtidas pelas funções NGK e NGF possuem 39 conjuntos de generalidade, porém, por falta de espaço, parte dos valores do domínio - os valores de generalidade que formam os identificadores dos conjuntos de generalidade - são omitidos.

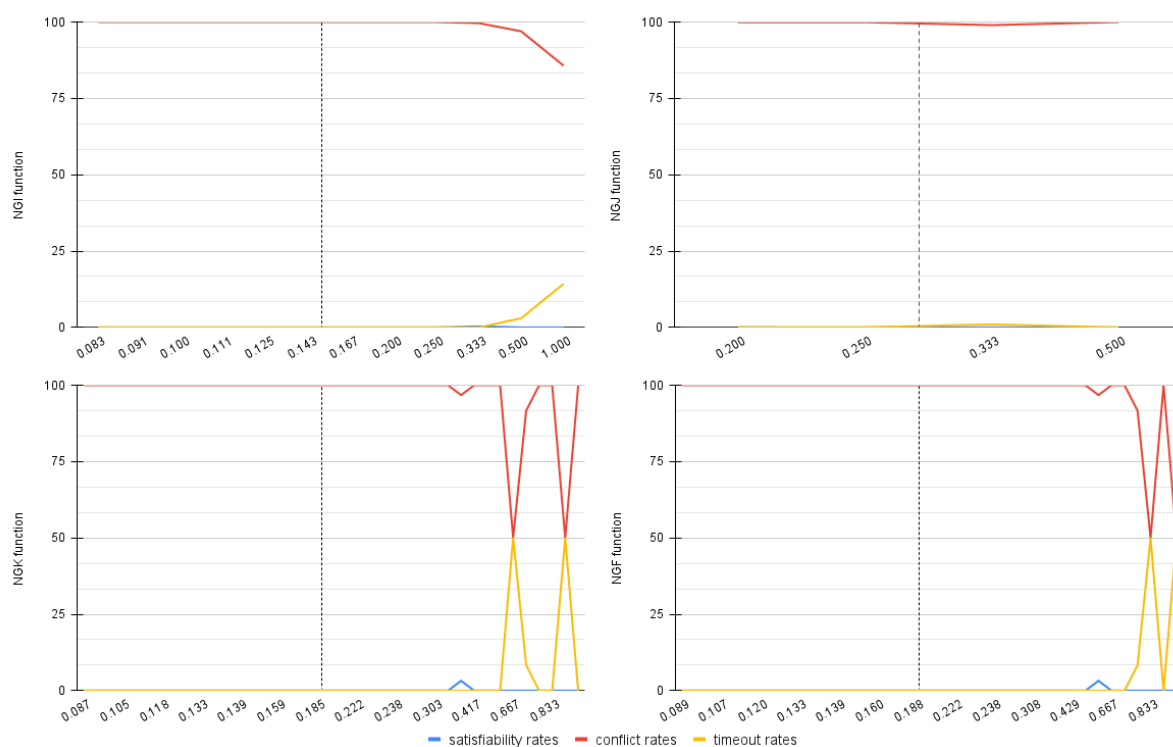


Figura 25 – Resultados do cenário do atendimento médico.

sucesso e conflito da norma n'' , consideramos n'' a melhor norma quase eficiente deste cenário. Como histórias de conflitos não ocorrem quando o simulador executa com a norma n'' e 100% das execuções findam em histórias de sucesso, consideramos n'' uma norma eficiente. Neste contexto, os conjuntos $G_{0.333}$, $G_{0.200}$, $G_{0.400}$ e $G_{0.455}$ correspondem aos conjuntos G^{apex} das sequências a quais pertencem. No entanto, da mesma forma como ocorre nos dois cenários anteriores, os conjuntos G^{apex} são constituídos também por normas não eficientes e, conseqüentemente, as suas taxas de sucesso são afetadas negativamente pelo mal desempenho do simulador quando os pacientes são regulados por elas. Portanto, a taxa de histórias de sucesso dos conjuntos G^{apex} também não alcançam 100%.

Como o espaço das normas deste cenário é muito maior do que os espaços das normas dos cenários anteriores, os conjuntos de generalidade das sequências deste cenário são, em média, muito maiores do que os conjuntos de generalidade das sequências dos cenários do tráfego de carros e segurança de acesso. Portanto, os conjuntos de generalidade das sequências deste cenário são afetados negativamente pelas normas não eficientes que os constituem mais do que nos cenários anteriores, o que faz com que as taxas de histórias de sucesso sejam próximas de zero. Isto contribui para que as taxas de histórias de sucesso dos conjuntos G^{apex} das sequências obtidas pelas funções NGI e NGJ não sejam visíveis nos gráficos correspondentes e para que as taxas de histórias de sucesso dos conjuntos G^{apex} das sequências obtidas pelas funções NGK e NGF sejam muito menores do que nos cenários anteriores.

Por outro lado, as taxas de histórias de conflito e timeout deste cenário são semelhantes nos cenários anteriores. No entanto, para as funções NGI, NGK e NGF, as taxas de histórias de timeout começam a crescer próximas à região G^{max} , enquanto as taxas de histórias de conflito começam a decrescer próximas a tal região. Além do mais, para a função NGJ, a taxa de histórias de timeout alcança o maior valor na região G^{max} para, então, decrescer, enquanto a taxa de histórias de conflito alcança o menor valor nesta mesma região para, então, crescer.

Ao analisarmos as posições dos conjuntos G^{apex} nos eixos- x , podemos visualizar que existem, basicamente, duas regiões onde tais conjuntos são encontrados. Nas sequências obtidas pelas funções NGI, NGK e NGF, o conjunto G^{apex} está entre as regiões G^{cen} e G^{max} . Por outro lado, em relação à sequência obtida pela função NGJ, o conjunto G^{apex} encontra-se na região G^{min} .

4.3 DISCUSSÃO

Embora a hipótese 1 postule que as normas eficientes estejam localizadas em torno do meio das sequências semânticas, por obtermos sequências aproximadas, a posição das normas eficientes nas aproximações pode destoar das suas posições reais. Além disso, conforme verificado em cada cenário, as posições onde as normas eficientes são encontradas nas aproximações muda de acordo com as funções de generalidade. Embora exista uma divergência no posicionamento de tais normas, nós destacamos *dois resultados principais* para os experimentos. Primeiramente, nós observamos que existe uma tendência no crescimento e decrescimento das taxas de sucesso, conflito e timeout associadas às sequências aproximadas obtidas a partir das funções NGI, NGK e NGF. Segundo, nós observamos que, em tais sequências, os conjuntos G^{apex} (definição 50) encontram-se na metade superior. Essas observações estão ilustradas pela figura 26, onde as curvas em azul, vermelho e laranja representam, respectivamente, as taxas de sucesso, conflito e timeout. Com linhas verticais tracejadas, nós marcamos os pontos do gráfico onde os elementos do eixo- x correspondem aos conjuntos G^{min} , G^{cen} , G^{apex} e G^{max} . Por sua vez, a área destacada em lilás corresponde aos locais na metade superior onde encontramos as normas eficientes. Devido à hipótese 1, representamos no gráfico $G^{cen} = G^{apex}$.

A taxa de conflito inicia alta e, em torno do meio das sequências, passa a diminuir gradualmente. Por outro lado, ambas as taxas de sucesso e timeout iniciam pequenas. Contudo, a taxa de sucesso começa a crescer em torno do meio alcançando o maior valor em G^{cen} . A partir de tal ponto, a taxa de sucesso passa a diminuir. Por sua vez, a taxa de timeout cresce gradualmente a partir do meio das sequências e alcança o maior valor em G^{max} .

Estas tendências são observadas por algumas razões. À medida que as normas se afastam do conjunto G^{cen} para G^{min} , as normas passam a ter uma maior ocorrência

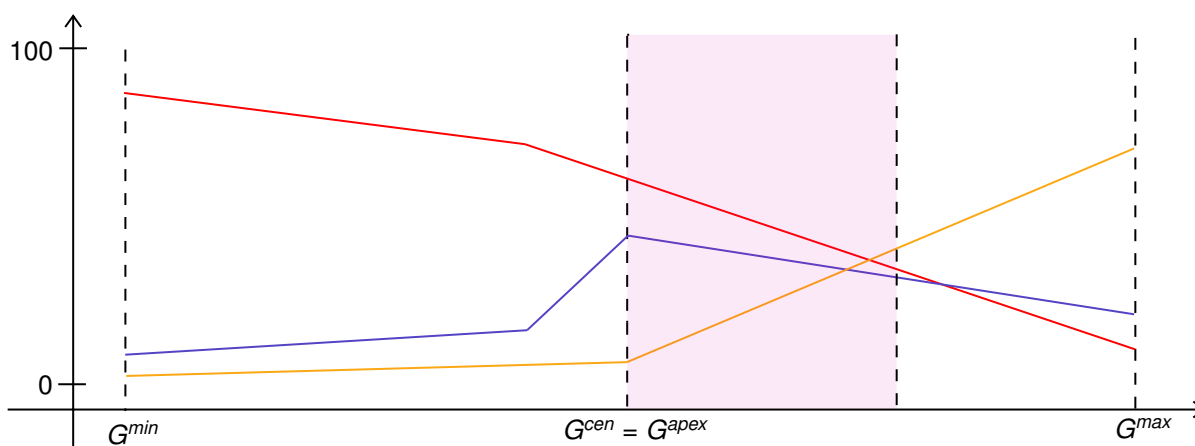


Figura 26 – Tendências no crescimento e decréscimo das taxas de sucesso, conflito e timeout.

das características sintáticas exploradas pelas funções de generalidade. Isso torna as normas mais específicas, o que dificulta elas serem aplicáveis em diferentes estados do sistema. Conseqüentemente, as normas passam a dar mais liberdade aos agentes para executarem as suas ações, o que tende a gerar diferentes comportamentos indesejados e aumentar a taxa de conflito. Em contraste, à medida que as normas se afastam do conjunto G^{cen} para G^{max} , elas passam a ter uma menor ocorrência das características sintáticas exploradas pelas funções de generalidade. Isso torna as normas mais gerais, o que facilita elas serem aplicáveis em diferentes estados do sistema. Conseqüentemente, as normas passam a dar menos liberdade aos agentes para executarem as suas ações, o que tende a inviabilizar o alcance dos objetivos do sistema e aumentar a taxa de timeout. Neste contexto, é intuitivo pensar que, por possuírem uma ocorrência média das características sintáticas exploradas pelas funções de generalidade, as normas que estão no meio das sequências são capazes de manter um equilíbrio entre o controle que exercem e a autonomia dos agentes. Por não serem muito específicas, as normas devem evitar conflitos e, por não serem muito gerais, elas devem manter os objetivos do sistema alcançáveis. Diante disto, as normas do meio são mais propensas a possuírem uma generalidade *balanceada*, onde parte delas é, possivelmente, eficiente. Por estes motivos, a taxa de sucesso, que inicia pequena, cresce mais acentuadamente em torno do meio, alcança o maior valor em G^{cen} e, então, passa a decrescer. Contudo, se G^{cen} não for igual a G^{apex} , então o ponto na área lilás onde o conjunto G^{apex} está localizado é onde o comportamento das taxas em torno de G^{cen} descritas anteriormente ocorrem.

Além do mais, a existência da área lilás também pode ser justificada por outras duas razões. Conforme a definição 27, as normas eficientes não permitem a ocorrência de conflitos e necessitam selecionar apenas uma história de sucesso, logo, para certos domínios, é possível que tais normas necessitem exercer mais controle sobre os agentes do que dar-lhes autonomia. Neste sentido, para serem balanceadas, as

normas eficientes podem necessitar ser mais gerais do que postula a hipótese 1. Além disso, a generalidade que confere às normas serem eficientes pode ser dependente de domínio. Portanto, para certos cenários de aplicação, a generalidade balanceada é menos restritiva, o que pode fazer com que as normas eficientes estejam localizadas na metade inferior das sequências aproximadas, enquanto para outros cenários, ela necessita ser mais restritiva, fazendo com que as normas eficientes estejam localizadas na metade superior de tais sequências.

Para nos auxiliar a responder à pergunta realizada no início deste capítulo, *como as diferentes atribuições sintáticas impactam no posicionamento das normas eficientes nas sequências aproximadas*, a tabela 3 resume os resultados dos experimentos, por função de generalidade, de acordo com quatro critérios de avaliação: *Conj* - o número de conjuntos de generalidade; *Dist* - a distância entre os conjuntos G^{cen} e G^{apex} (dada em número de conjuntos de generalidade); *Pos* - a posição dos conjunto G^{apex} nas sequências aproximadas (à *esquerda* ou à *direita* dos conjuntos G^{cen} ou *meio* quando $G^{cen} = G^{apex}$); e *Suc* - a taxa de sucesso dos conjuntos G^{apex} .

Cenário	NGI				NGJ				NGK				NGF			
	Conj	Dist	Suc	Pos	Conj	Dist	Suc	Pos	Conj	Dist	Suc	Pos	Conj	Dist	Suc	Pos
Tráfego	4	1	9.4%	<i>direita</i>	3	1	5%	<i>esquerda</i>	7	0	23.6%	<i>meio</i>	7	1	23.6%	<i>direita</i>
Segurança	5	0	45%	<i>meio</i>	3	1	41.1%	<i>direita</i>	19	4	56.3%	<i>direita</i>	19	4	56.3%	<i>direita</i>
Médico	12	4	0.35%	<i>direita</i>	4	2	0.03%	<i>esquerda</i>	39	11	3.2%	<i>direita</i>	39	14	3.2%	<i>direita</i>

Tabela 3 – Comparativo dos resultados dos experimentos.

Podemos observar que *as funções de generalidade impactam na posição das normas eficientes de diferentes formas*. Ao compararmos as funções NGK e NGF com a função NGI, podemos verificar que tais funções obtêm sequências maiores. Ao contrário de NGI que discrimina as normas unicamente pela ocorrência de fórmulas atômicas, NGK e NGF discriminam as normas pela ocorrência de fórmulas atômicas e variáveis, o que faz com que elas mapeiem as normas a mais conjuntos de generalidade. Enquanto a *média de conjuntos*⁵ das sequências obtidas por NGK e NGF é 21,66, a média de conjuntos das sequências obtida pela função NGI é 7.

A capacidade das funções NGK e NGF distribuírem as normas a mais conjuntos de generalidade auxilia elas a mapearem as normas eficientes a conjuntos mais distantes de G^{cen} do que a função NGI. Enquanto a *distância média* entre G^{cen} e G^{apex} nas sequências obtidas por NGI é 1,66 conjuntos de generalidade⁶, a distância média nas sequências obtidas por NGK e NGF é 5 e 6,33 conjuntos, respectivamente. Por outro lado, a função NGJ obtém as menores sequências aproximadas. Em média, as

⁵ Considerando uma função de generalidade, a média de conjuntos em suas sequências aproximadas é calculada somando os valores da respectiva coluna Conj e dividindo o resultado pelo número total de cenários.

⁶ Considerando uma função de generalidade, a distância média entre G^{cen} e G^{apex} em suas sequências aproximadas é calculada somando os valores da respectiva coluna Dist e dividindo o resultado pelo número total de cenários.

sequências possuem 3,33 conjuntos de generalidade. Isso auxilia NGJ a possuir a menor distância média entre G^{cen} e G^{apex} em suas sequências, isto é, 1,33 conjuntos.

Como as funções NGK e NGF obtêm as maiores sequências, os seus conjuntos de generalidade possuem a menor média de normas, o que auxilia aos conjuntos G^{apex} possuírem as maiores taxas de sucesso. Enquanto tais conjuntos possuem em média⁷ uma taxa de sucesso de 27,7%, os conjuntos G^{apex} das sequências obtidas a partir de NGI e NGJ possuem em média uma taxa de sucesso de 18,25% e 15,35%, respectivamente.

Além disso, em todos os cenários, observamos que as funções NGI, NGK e NGF apresentam a propriedade 1 e as suas sequências aproximadas possuem as normas eficientes na metade superior. Por outro lado, observamos que a função NGJ não apresenta a propriedade 1 em nenhum cenário. Em virtude disso, ela mapeia diferentes normas contíguas a um mesmo conjunto de generalidade, o que faz com que não exista uma região das sequências aproximadas onde as normas eficientes tendem a ser encontradas. Em certos cenários, as sequências aproximadas obtidas pela função NGJ possuem as normas eficientes na metade inferior, enquanto em outros cenários, elas possuem tais normas na metade superior. Mediante o exposto, julgamos que as funções NGI, NGK e NGF fomentam a criação de sequências aproximadas mais próximas das sequências semânticas do que as sequências da função NGJ e que as sequências semânticas devem apresentar uma tendência em suas taxas de sucesso, conflito e timeout similar a tendência observada para NGI, NGK e NGF.

Embora para obtermos conclusões mais assertivas seja necessário realizarmos experimentos em outros cenários, considerando que os resultados discutidos são observados à medida que os cenários dos experimentos tornam-se mais complexos e o tamanho do espaço das normas e das sequências aproximadas aumenta, nós concluímos também que: i) explorar *duas características sintáticas das normas* torna as funções de generalidade mais propensas a obterem sequências aproximadas maiores, com conjuntos de generalidade menores e conjuntos G^{apex} com taxas de sucesso maiores do que funções que exploram apenas uma de tais características; ii) explorar *uma característica sintática das normas* torna as funções de generalidade mais propensas a obterem sequências aproximadas onde a distância entre os conjuntos G^{cen} e G^{apex} é menor do que em sequências obtidas a partir de funções que exploram duas de tais características; iii) De acordo com as características sintáticas exploradas em comum entre as funções NGI, NFK e NGF, julgamos que explorar, pelo menos, *as fórmulas atômicas das normas* torna as funções de generalidade mais propensas a obterem sequências mais próximas das sequências semânticas do que explorar unicamente as variáveis, o que auxilia as funções a mapearem as normas eficientes na parte superior

⁷ Considerando uma função de generalidade, a média da taxa de sucesso do conjunto G^{apex} em suas sequências aproximadas é calculada somando os valores da respectiva coluna Suc e dividindo o resultado pelo número total de cenários.

das sequências aproximadas.

Por fim, nós pretendemos utilizar o conhecimento sobre a tendência anterior e as regiões das sequências aproximadas onde identificamos as normas eficientes como uma heurística para orientar a busca por normas eficientes. Neste sentido, no próximo capítulo, nós apresentamos um conjunto de algoritmos para a síntese de normas que fazem uso dos conceitos e conhecimentos adquiridos nos capítulos 3 e 4.

5 DESENVOLVIMENTO INCREMENTAL DE ALGORITMOS PARA A SÍNTESE DE NORMAS EFICIENTES

Anteriormente, no capítulo 3, nós discutimos que para uma norma ser eficiente é necessário que a norma: i) selecione, pelo menos, uma história de sucesso; e ii) evite histórias de conflito. Complementarmente, no capítulo 4, nós discutimos que para isto a norma deve ter uma generalidade balanceada, o que implica em a norma não ser nem muito específica nem muito geral. Embora, conforme a hipótese 1, para serem balanceadas, as normas eficientes devem estar localizadas em torno do meio das sequências semânticas, encontrar tais normas traz à tona uma série de desafios. Neste contexto, o objetivo deste capítulo é apresentar os problemas envolvidos na síntese de normas eficientes e avaliar algoritmos capazes de sintetizá-las considerando tais problemas. Para isto, na seção 5.1, nós apresentamos o desenvolvimento incremental de um conjunto de algoritmos através de um conjunto de aprimoramentos que se aplica, inicialmente, sobre um algoritmo base que descreve os passos da abordagem ingênua para a síntese de normas introduzida no capítulo 3. À medida que os aprimoramentos são aplicados, novos algoritmos para a síntese de normas são criados e, para cada algoritmo obtido, nós discutimos as vantagens e as desvantagens do seu uso. Ao final, nós obtemos três algoritmos que tratam a síntese de normas como um problema de busca em espaços ordenados. Nós também apresentamos exemplos sobre o funcionamento dos algoritmos considerando o cenário do tráfego de carros simplificado introduzido na seção 3.2. Por sua vez, na seção 5.2, nós realizamos experimentos sobre os cenários do capítulo anterior, onde certos algoritmos de síntese da seção 5.1 são explorados e comparados. Por fim, na seção 5.3, nós apresentamos uma discussão sobre os resultados obtidos.

5.1 ALGORITMOS PARA A SÍNTESE DE NORMAS

Conforme discutido no capítulo 3, uma abordagem ingênua, porém, intuitiva para sintetizar normas eficientes é sintetizar normas que, sob certos contextos de ativação, proíbem a execução da ação que faz com que o sistema transite de um estado $s_i \in \mathcal{S}$ para um estado de conflito $s_j \in \mathcal{S}$. Neste contexto, o algoritmo 1 descreve os passos da abordagem ingênua para a síntese de normas eficientes. Para isto, o algoritmo recebe como entrada um conflito $c \in \mathcal{C}$, um conjunto de instâncias de ações \mathcal{I}^A e um conjunto de fórmulas atômicas ground $\mathcal{L}^{gaf} \subset \mathcal{L}'$.

O algoritmo 1 inicia criando um conjunto de normas \mathcal{N} vazio para armazenar normas proibitivas (linha 1). Em seguida, verifica para cada estado s do sistema (linha 2), onde um estado corresponde a um conjunto do conjunto potência $2^{\mathcal{L}^{gaf}}$ (conforme a definição 2), se s é um estado de conflito (linha 3 - conforme a definição 10). Se s for um estado de conflito, então o algoritmo determina as ações de \mathcal{I}^A que necessitam

Algoritmo 1: Abordagem ingênua para a síntese de normas

```

Data:  $c, \mathcal{I}^A, \mathcal{L}^{gaf}$ 
1  $\mathcal{N} \leftarrow \emptyset$ 
2 for  $s \in 2^{\mathcal{L}^{gaf}}$  do
3   if  $s \models c$  then // definição 10
4     for  $a_p \in \mathcal{I}^A$  do
5       if  $add^{a_p} \subset s \wedge \nexists p \in del^{a_p} : p \in s$  then
6          $S^{a_p} \leftarrow getEstadosProibitivos(a_p, s)$ 
7         for  $s_p \in S^{a_p}$  do
8            $\varphi = \gamma(s_p)$ 
9            $n \leftarrow \varphi \xrightarrow[p]{}$   $a_p$ 
10           $\mathcal{N} \leftarrow \mathcal{N} \cup n$ 
11 return  $\mathcal{N}$ 

```

ser proibidas para que o estado s seja evitado. Para isto, é verificado para cada ação $a_p \in \mathcal{I}^A$ (linha 4), se o estado s é alcançado quando a_p é executada. Se os efeitos de adição add^{a_p} estão contidos em s e não existem efeitos de deleção de del^{a_p} em tal estado (linha 5), então a_p é uma ação a ser proibida. Neste caso, é necessário identificar os estados onde a_p , ao ser executada, alcança o estado s . Para isto, o algoritmo obtém, através do método *getEstadosProibitivos* (linha 6), o conjunto S^{a_p} dos estados onde as pré-condições de execução da ação a_p são satisfeitas e há uma transição de tais estados para s a partir da execução de a_p . Para cada estado proibitivo $s_p \in S^{a_p}$ (linha 7), um contexto φ é determinado pela função γ^1 (linha 8) considerando as fórmulas atômicas ground de s_p . Então, uma norma n é sintetizada (linha 9) e adicionada ao conjunto \mathcal{N} (linha 10). Quando todos os estados do sistema são analisados, o conjunto \mathcal{N} é retornado (linha 11).

Exemplo 32 Considere o grafo do exemplo 17 que apresenta um conjunto de histórias do sistema para o cenário do tráfego de carros. A figura 27 ilustra um subgrafo de tal grafo. De acordo com este recorte, considere também que $2^{\mathcal{L}^{gaf}} = \{s_0, \dots, s_{14}\}$, $\mathcal{H} = \{h_1, h_2, h_3, h_4, h_5\}$, $\mathcal{I}^A = \{move(ag1, g, d), \dots, move(ag1, e, b)\}^2$ e $c = \{at(Ag_1, L_2), at(Ag_2, L_2)\}$, tal que, s_5 e s_{10} são estados de sucesso, s_{11} e s_{13} são estados de conflito, s_{14} é um estado de timeout, h_1 e h_2 são histórias de sucesso, h_3 e h_4 são histórias de conflito, h_5 é uma história de timeout e $move(ag1, e, f)$ e $move(ag1, h, i)$ são ações proibitivas.

Ao gerar os estados de $2^{\mathcal{L}^{gaf}}$, o algoritmo 1 identifica que $s_{11} \models c$ e $s_{13} \models c$. Então, a partir de tais estados, o algoritmo 1 identifica, respectivamente, que as ações

¹ γ é uma função hipotética que determina o contexto de uma norma. O contexto é um conjunto de fórmulas atômicas ground $\varphi \subseteq 2^{S_p}$.
² Nós omitimos parte das ações que constituem o conjunto \mathcal{I}^A para simplificar a sua representação. Conforme a figura 27, tem-se que $|\mathcal{I}^A| = 10$.

$move(ag1,h,i)$ e $move(ag1,e,f)$ geram conflito. Como existe uma transição do estado s_7 para s_{11} quando $move(ag1,h,i)$ é executada e, como tal ação é executável somente em s_7 , o algoritmo 1 constrói $S^{move(ag1,h,i)} = \{s_7\}$ (ver linha 6 do algoritmo 1). Por outro lado, a ação $move(ag1,e,f)$ é executável nos estados s_4, s_9 e s_{12} . Contudo, somente a partir de s_{12} existe uma transição para s_{13} quando $move(ag1,e,f)$ é executada. Logo, o algoritmo 1 constrói $S^{move(ag1,e,f)} = \{s_{12}\}$. A partir de $S^{move(ag1,h,i)}$, o algoritmo 1 sintetiza a norma $n_1 = \varphi' \xrightarrow{p} move(ag1,h,i)$ e, a partir de $S^{move(ag1,e,f)}$, é sintetizada a norma $n_2 = \varphi'' \xrightarrow{p} move(ag1,e,f)$, com $\varphi' \subseteq s_7$ e $\varphi'' \subseteq s_{12}$. Por fim, o algoritmo 1 retorna o espaço de normas $\mathcal{N} = \{n_1, n_2\}$.

Embora a norma n_2 evite o estado de conflito s_{13} e, assim, contribua para que a história de conflito h_4 não ocorra, ela não é capaz de manter os objetivos do sistema alcançáveis, pois não seleciona história de sucesso alguma. Por outro lado, embora a norma n_1 mantenha os objetivos do sistema alcançáveis ao selecionar a história de sucesso h_2 e evite o estado de conflito s_{11} contribuindo para que a história de conflito h_3 não ocorra, ela não é capaz de evitar a ocorrência da história de conflito h_4 . Logo, para que os estados de conflito s_{11} e s_{13} sejam evitados, é necessário que ambas as normas n_1 e n_2 regulem os agentes em conjunto.

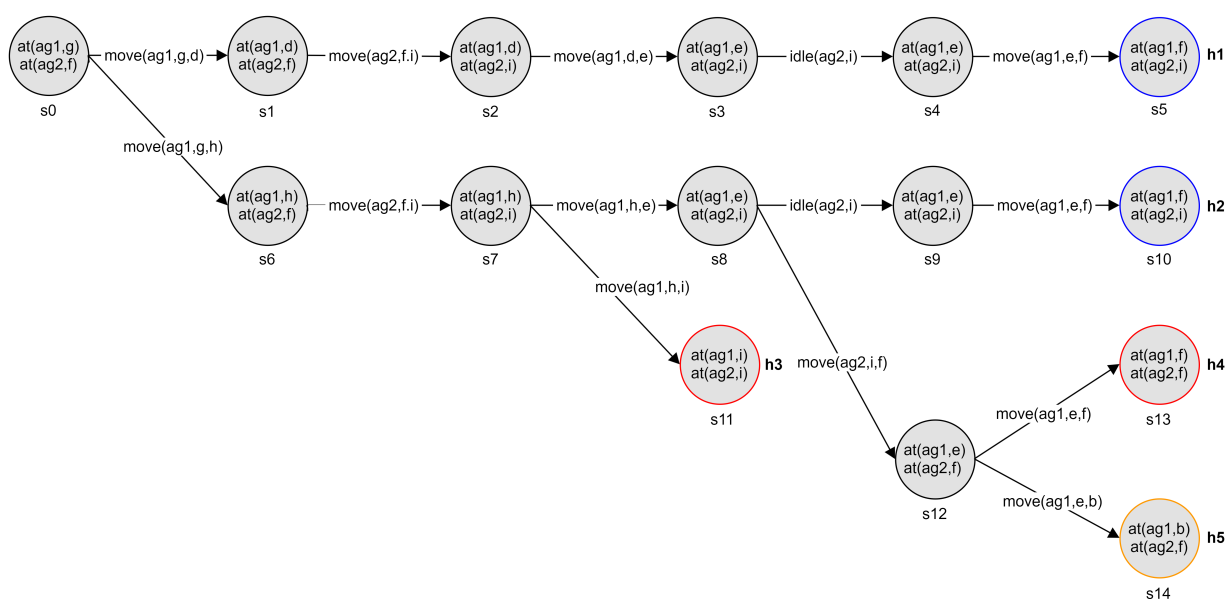


Figura 27 – Exemplo de aplicação do algoritmo 1.

Embora o algoritmo 1 seja capaz de construir um conjunto de normas que evita conflitos sem uma análise prévia da eficiência das normas em regular os agentes, dependendo do domínio, obtê-lo pode ser inviável, pois requer analisar todos os estados do sistema que, por sua vez, tende a ser um conjunto grande. Além do mais, o conjunto das normas resultante também pode ser grande dado que, para cada estado de

conflito, uma norma é sintetizada para que o estado seja evitado. Consequentemente, decidir sobre quais ações os agentes devem executar considerando muitas normas pode demandar tempo de raciocínio. Como as normas obtidas na síntese ingênua são muito específicas, é possível que certas normas não tenham utilidade ao sistema, pois os comportamentos que elas regulam podem não ser apresentados pelos agentes em tempo de execução. Somando-se a isto, conforme discutido no capítulo 3, não há garantia de que as normas obtidas pela síntese ingênua selecionem, pelo menos, uma história de sucesso (como a norma n_2 que possui um conjunto de histórias de sucesso vazio) e, conforme discutido no capítulo 4, por regularem poucos comportamentos, as normas muito específicas tendem a selecionar mais histórias de conflito. Logo, para que os estados de conflito não sejam alcançados, possivelmente, é necessário que os agentes sejam regulados por diferentes normas específicas de um domínio.

A fim de evitar determinar todos os estados do sistema para identificar os estados de conflito e assim construir um conjunto de normas proibitivas, um aprimoramento a ser aplicado sobre o algoritmo 1 é construir um conjunto de normas combinando diretamente os elementos da linguagem \mathcal{L} (definição 1). Além disto, para que as normas sejam eficientes, um segundo aprimoramento a ser aplicado sobre o algoritmo 1 é *considerar normas que satisfaçam os critérios de eficiência da definição 27*. Desta forma, as normas devem selecionar, pelo menos, uma história de sucesso e não selecionar nenhuma história de conflito, considerando um dado estado inicial s_0 e um conjunto de objetivos do sistema \mathcal{G} . Tais aprimoramentos são contemplados pelo algoritmo 2 que recebe como entrada um conjunto de ações \mathcal{A} , um conjunto de instâncias de ações $\mathcal{I}^{\mathcal{A}}$, um conjunto de fórmulas atômicas unground \mathcal{L}^{uaf} , uma função de substituição de variáveis por constantes $Map : Var \rightarrow Const$, um estado inicial $s_0 \in S_0$, um conjunto de objetivos do sistema $\mathcal{G} \in \mathcal{G}^*$ e um timeout $\Delta \in \mathbb{Z}_+^*$ (inteiros positivos).

Algoritmo 2: Aprimoramento 1

```

Data:  $\mathcal{A}, \mathcal{I}^{\mathcal{A}}, \mathcal{L}^{uaf}, Map, s_0, \mathcal{G}, \Delta$ 
1  $\mathcal{H} \leftarrow getHistorias(s_0, \mathcal{I}^{\mathcal{A}}, \Delta)$ 
2 for  $(\varphi, a) \in (2^{\mathcal{L}^{uaf}} \times \mathcal{A})$  do
3    $n \leftarrow \varphi \xrightarrow{p} a$ 
4    $\mathcal{N} \leftarrow n \cup getSubstituicoes(n, Map)$ 
5   for  $n_i \in \mathcal{N}$  do
6      $\mathcal{H}_{n_i}^{suc}, \mathcal{H}_{n_i}^{conf} \leftarrow getHistoriasNorma(n_i, c, \mathcal{G}, \mathcal{H})$ 
7     if  $\mathcal{H}_{n_i}^{suc} \neq \emptyset \wedge \mathcal{H}_{n_i}^{conf} = \emptyset$  then // definição 27
8       return  $n_i$ 
9 null;

```

O algoritmo 2 inicia criando, através do método *getHistorias* (linha 1), um conjunto de histórias \mathcal{H} (conforme a definição 20) para auxiliar na busca de normas efi-

cientes. Mais especificamente, o método *getHistorias* constrói todas as histórias do sistema que iniciam em s_0 de todos os tamanhos possíveis considerando as instâncias de ações de \mathcal{I}^A e o timeout Δ . Em seguida, o algoritmo cria, para cada par de fórmulas atômicas estritamente unground e ações do produto cartesiano $2^{\mathcal{L}^{uaf}} \times \mathcal{A}$ (linha 3 - conforme a definição 15), uma norma n . Então, através do método *getSubstituicoes* (linha 4), o algoritmo constrói um conjunto de normas \mathcal{N} constituído pela norma n e por normas ground e unground obtidas substituindo as variáveis de n por constantes, conforme a função *Map*. Tais normas são obtidas substituindo p variáveis de n por p constantes, onde $p \in [1, \text{variaveis}(n)]$ (com a função *variaveis* introduzida na definição 41), o que viabiliza a obtenção de normas mais específicas. Posteriormente, o algoritmo determina, através do método *getHistoriasNorma* (linha 6), as histórias de sucesso $\mathcal{H}_{n_i}^{suc}$ e conflito $\mathcal{H}_{n_i}^{conf}$ de cada norma $n_i \in \mathcal{N}$. Para isto, o método identifica as histórias de sucesso e conflito de \mathcal{H} , considerando, respectivamente, o conflito c e o conjunto de objetivos do sistema \mathcal{G} , que possuem estados nos quais a norma n_i é aplicável. Por fim, o algoritmo determina se a norma n_i corrente é eficiente (linha 7). Se for o caso, n_i é retornada (linha 8). Caso nenhuma norma eficiente seja sintetizada, então o algoritmo retorna *null*.

Exemplo 33 Considere que o algoritmo 2 constrói, inicialmente, o conjunto de histórias \mathcal{H} do exemplo 32. Também considere que o algoritmo 2 cria a norma $n = \{at(Ag_2, L_2)\} \xrightarrow{p} \text{move}(Ag_1, L_1, L_2)$ e, a partir de tal norma, ele cria as normas n_1 e n_2 do exemplo 32, tal que, $n \underset{g}{>} n_1 \underset{g}{>} n_2$. Desta forma, é obtido um conjunto de normas $\mathcal{N} = \{n, n_1, n_2\}$. Seguindo as definições 24 e 26, como a norma n_1 é aplicável no estado s_7 da história sucesso h_2 e conflito h_3 , tem-se que $h_2 \in \mathcal{H}_{n_1}^{suc}$ e $h_3 \notin \mathcal{H}_{n_1}^{conf}$. Contudo, como n_1 não impede a história de conflito h_4 ocorrer e não contribui para que a história de sucesso h_1 ocorra, tem-se que $h_4 \in \mathcal{H}_{n_1}^{conf}$ e $h_2 \notin \mathcal{H}_{n_1}^{suc}$. Logo, a norma n_1 não satisfaz a definição 27 para ser uma norma eficiente, pois embora selecione, pelo menos, uma história de sucesso, o seu conjunto de histórias de conflito não é vazio. Assim, o algoritmo 2 não retorna a norma n_1 . Por outro lado, como a norma n_2 é aplicável no estado s_{12} da história de conflito h_4 , tem-se que $h_4 \notin \mathcal{H}_{n_2}^{conf}$. Porém, como tal norma não é aplicável em nenhum estado das histórias h_1 , h_2 e h_3 , tem-se que $h_1 \notin \mathcal{H}_{n_2}^{suc}$, $h_2 \notin \mathcal{H}_{n_2}^{suc}$ e $h_3 \in \mathcal{H}_{n_2}^{conf}$. Logo, a norma n_2 também não é uma norma eficiente, pois o seu conjunto de histórias de sucesso é vazio e o seu conjunto de histórias de conflito não é vazio. Conseqüentemente, o algoritmo 2 também não retorna a norma n_2 . Por fim, como a norma n generaliza as normas n_1 e n_2 , ela é aplicável nos mesmos estados que as normas generalizadas são. Portanto, a norma n evita as histórias de conflito h_3 e h_4 e contribui para que a história de sucesso h_2 ocorra. Deste modo, tem-se que $\mathcal{H}_n^{suc} = \{h_2\}$ e $\mathcal{H}_n^{conf} = \{\}$. Portanto, como a norma n atende a definição 27, n é uma norma eficiente e é, portanto, retornada pelo algoritmo 2.

Embora o algoritmo 2 seja capaz de sintetizar normas eficientes independentemente de uma análise prévia do conjunto dos estados do sistema, ele não necessariamente o faz de forma mais rápida e menos custosa do que o algoritmo 1. Dado que $|\mathcal{L}^{uaf}| \leq |\mathcal{L}^{gaf}|$, então obter o conjunto potência $2^{\mathcal{L}^{uaf}}$ tende a ser menos custoso do que obter $2^{\mathcal{L}^{gaf}}$. Entretanto, o produto cartesiano entre os conjuntos $2^{\mathcal{L}^{uaf}}$ e \mathcal{A} , calculado pelo algoritmo 2 (linha 2), pode ser maior do que o conjunto potência $2^{\mathcal{L}^{gaf}}$, calculado pelo algoritmo 1 (linha 2). Dado que ambos os conjuntos $2^{\mathcal{L}^{uaf}}$ e $2^{\mathcal{L}^{gaf}}$ tendem a ser grandes, sintetizar normas considerando $2^{\mathcal{L}^{uaf}} \times \mathcal{A}$ pode demandar mais tempo do que considerar unicamente $2^{\mathcal{L}^{gaf}}$. Além do mais, o conjunto de normas considerado pelo algoritmo 2 tende a ser maior do que o conjunto considerado pelo algoritmo 1, pois ele pode ser constituído por todas as normas da linguagem \mathcal{L} . Assim, o tempo para sintetizar normas eficientes do algoritmo 2 pode ser consideravelmente maior do que o tempo para sintetizar todas as normas específicas do algoritmo 1. Somando-se a isto, é possível que o algoritmo 2 sintetize normas de todas as generalidades, o que faz com que parte da busca por normas eficientes seja gasta com normas que trazem pouco ou nenhum benefício ao sistema, como as normas *muito específicas*, que regulam poucos comportamentos dos agentes e tendem a permitir muitos conflitos.

Conforme discutido no capítulo 4, normas generalizadoras regulam os comportamentos que as normas generalizadas regulam. Assim, as normas generalizadoras são aplicáveis nos estados do sistema onde as normas que elas generalizam também são. Partindo deste pressuposto, um primeiro aprimoramento a ser aplicado sobre o algoritmo 2, a fim de contornar as limitações previamente expostas, é *sintetizar as normas de acordo com a definição 11*. Desta forma, as normas são especificadas com termos sendo unicamente variáveis. Para fins de comparação, neste capítulo, chamaremos as normas da definição 11 de *normas estritamente unground*, as normas que possuem, pelo menos, uma variável de *normas unground* e as normas cujos termos são unicamente constantes de *normas ground*. Como as normas estritamente unground generalizam normas unground e ground, podemos dizer que as normas desconsideradas na busca estão contidas no conjunto das normas implicitamente através das normas estritamente unground. Conseqüentemente, as normas estritamente unground são aplicáveis em mais estados do sistema do que as normas que elas generalizam, o que pode fazer com que as normas estritamente unground evitem mais histórias de conflito e contribuam para a ocorrência de mais histórias de sucesso. Neste sentido, assumimos para o próximo algoritmo que para serem eficientes, as normas necessitam ser estritamente unground. Assim, através do primeiro aprimoramento para o algoritmo 2, o conjunto das normas torna-se menor, o que pode auxiliar a diminuir o tempo de busca por normas eficientes. Além do mais, deixa-se de gastar tempo com normas que, por serem muito específicas, contribuem pouco ou nada para que conflitos sejam evitados e os objetivos do sistema sejam alcançados. Conforme o exemplo 33, a norma n , que é

estritamente unground, generaliza as normas n_1 e n_2 , que são ground. Logo, a norma n regula os comportamentos que n_1 e n_2 regulam e é aplicável nos estados onde n_1 e n_2 são. Portanto, ao considerarmos $\mathcal{N} = \{n\}$, as normas n_1 e n_2 estão consideradas no conjunto das normas através de n e, como n é eficiente, o tempo para encontrar uma norma eficiente pode ser menor ao considerarmos este espaço reduzido.

Além do mais, o algoritmo 2 analisa as normas de acordo com a ordem na qual elas são criadas. Logo, provavelmente, as normas não são analisadas segundo a sua generalidade, o que pode impossibilitar explorarmos inicialmente normas de uma generalidade desejada. Neste sentido, um segundo aprimoramento a ser aplicado sobre o algoritmo 2 é *organizar o espaço das normas ordenando-as segundo a sua generalidade*. Este aprimoramento permite obtermos uma sequência semântica, conforme discutido no capítulo 4, sobre a qual podemos aplicar diferentes estratégias de visita que priorizam a análise de normas com uma dada generalidade. Por fim, o algoritmo 2 constrói todas as histórias que iniciam no estado s_0 . Contudo, a combinação entre estados e ações pode explodir combinatoriamente gerando um conjunto de histórias grande. Entretanto, este é um custo a ser pago para assegurar que as normas sintetizadas sejam efetivamente eficientes.

Os aprimoramentos destacados para o algoritmo 2 estão contemplados pelo algoritmo 3 que recebe como entrada um conjunto de ações \mathcal{A} , um conjunto de instâncias de ações $\mathcal{I}^{\mathcal{A}}$, um conjunto de fórmulas atômicas ground \mathcal{L}^{gaf} , um estado inicial $s_0 \in \mathcal{S}_0$, um conjunto de objetivos do sistema $\mathcal{G} \in \mathcal{G}^*$, um timeout $\Delta \in \mathbb{Z}_+^*$ e uma estratégia de visita $v : \mathcal{O} \rightarrow \mathcal{O} \cup null$ que estabelece a ordem de visita dos conjuntos de generalidade de uma sequência semântica \mathcal{O} .

O algoritmo 3 realiza a síntese de normas eficientes como uma busca em um espaço de normas ordenado e funciona em 4 etapas. Primeiramente, constrói um espaço de normas. Em seguida, determina a generalidade das normas. Posteriormente, constrói uma sequência semântica. Por fim, realiza uma busca por normas eficientes sobre tal sequência. Para isto, o algoritmo 3 cria os conjuntos vazios \mathcal{N} , \mathcal{S}_a e \mathcal{R} (linha 1) para armazenar, respectivamente, as normas, os conjuntos de estados onde as normas são aplicáveis e as relações de generalidade entre as normas. Em seguida, o algoritmo 3 cria o conjunto de estados do sistema \mathcal{S} (linha 2) para auxiliar, posteriormente, na determinação da generalidade das normas (conforme a definição 36, a generalidade das normas é dada em função do conjunto de estados onde as normas são aplicáveis). Então, o algoritmo 3 obtém uma norma n de forma análoga ao algoritmo 2 (da linha 3 e 4), insere a norma no espaço \mathcal{N} (linha 5) e cria um conjunto de estados \mathcal{S}_a^n para armazenar os estados onde a norma n é aplicável (linha 6). Em seguida, para cada norma $n_i \in \mathcal{N}$ (linha 7) e $n_j \in \mathcal{N}$ (linha 8), é verificado se $n_i \succ_g n_j$ (linha 9 - conforme definição 37). Se for o caso, o par de normas (n_i, n_j) é inserido no conjunto \mathcal{R} (linha 10) que consiste em uma relação de ordem parcial, irreflexiva, assimétrica e transitiva que,

Algoritmo 3: Aprimoramento 2

```

Data:  $\mathcal{A}, \mathcal{I}^{\mathcal{A}}, s_0, \mathcal{G}, c, \Delta, v$ 
1  $\mathcal{N}, \mathcal{R} \leftarrow \emptyset$ 
2  $\mathcal{S} \leftarrow 2^{\mathcal{L}^{gaf}}$ 
3 for  $(\varphi, a) \in (2^{\mathcal{L}^{uaf}} \times \mathcal{A})$  do
4    $n \leftarrow \varphi \xrightarrow{p} a$ 
5    $\mathcal{N} \leftarrow \mathcal{N} \cup n$ 
6    $Sa^n \leftarrow \{s \mid s \in \mathcal{S} \wedge n \triangleright s\}$ 
7 for  $n_i \in \mathcal{N}$  do
8   for  $n_j \in \mathcal{N}$  do
9     if  $Sa^{n_j} \subset Sa^{n_i} \wedge Sa^{n_i} \not\subset Sa^{n_j} \wedge a^{n_i} = a^{n_j}$  then // se  $n_i > n_j$ 
10     $\mathcal{R} \leftarrow \mathcal{R} \cup (n_i, n_j)$ 
11  $G^{min} \leftarrow getConjuntoGmin(\mathcal{R})$ 
12  $O \leftarrow getSequenciaSemantica(G^{min}, \mathcal{R}, \mathcal{N})$ 
13  $first \leftarrow getIndice(O, v)$ 
14  $\mathcal{H} \leftarrow getHistorias(s_0, \mathcal{I}^{\mathcal{A}}, \Delta)$ 
15  $n_{eff} \leftarrow visitar(O, v, first, \mathcal{G}, \mathcal{H}, c, s_0)$ 
16 return  $n_{eff}$ 

```

conforme capítulo 4, é a base para a construção das sequências semânticas.

Após obter a relação \mathcal{R} , o algoritmo 3 inicia a construção de uma sequência semântica de normas O . Para isto, determina primeiramente o conjunto G^{min} da sequência semântica através do método *getConjuntoGmin* (linha 11) identificando as normas que não aparecem no lado esquerdo de nenhum par do conjunto \mathcal{R} . Consequentemente, tais normas não generalizam outras normas e são, então, as normas mais específicas do espaço de normas \mathcal{N} . Em seguida, a sequência O é construída por completo através do método *getSequenciaSemantica* (linha 12), descrito pelo algoritmo 4, que recebe como entrada os conjuntos G^{min} , \mathcal{R} e \mathcal{N} . Em seguida, através do método *getIndice* (linha 13), o algoritmo 3 obtém o índice do conjunto de generalidade por onde a busca por normas eficientes deve iniciar (i.e, as normas cuja generalidade é priorizada na busca). Para isto, o método *getIndice* recebe como entrada a sequência O e a estratégia de visita v . Então, o algoritmo obtém, através do método *getHistorias* (linha 14), o conjunto das histórias do sistema que iniciam no estado s_0 considerando as ações em $\mathcal{I}^{\mathcal{A}}$ e o timeout Δ . Por fim, através do método *visitar* (linha 15), a estratégia de visita v é aplicada sobre a sequência semântica O a partir da qual uma norma eficiente n_{eff} é obtida, caso exista. Por sua vez, a norma n_{eff} é retornada (linha 16).

O algoritmo 4, por sua vez, cria inicialmente uma sequência semântica de normas O como um vetor vazio (linha 1) para que seja possível usufruir de uma estrutura na qual a ordem dos conjuntos de generalidade é preservada. Em seguida, o algoritmo

Algoritmo 4: *getSequenciaSemantica*

```

Data:  $G^{min}, \mathcal{R}, \mathcal{N}$ 
1  $O \leftarrow []$ 
2  $k \leftarrow 0$ 
3  $G_k \leftarrow G^{min}$ 
4 while  $G_k \neq \emptyset$  do
5    $O_k \leftarrow G_k$ 
6    $\mathcal{N}_g \leftarrow \emptyset$ 
7   for  $n_j \in G_k$  do
8     for  $n_i \in \mathcal{N}$  do
9       if  $(n_i, n_j) \in \mathcal{R}$  then
10        if  $\nexists n_k \in \mathcal{N} : (n_i, n_k) \in \mathcal{R} \wedge (n_k, n_j) \in \mathcal{R}$  then // definição 45
11           $\mathcal{N}_g \leftarrow \mathcal{N}_g \cup n_i$ 
12    $k \leftarrow k + 1$ 
13    $G_k \leftarrow \mathcal{N}_g$ 
14 return  $O$ 

```

4 cria um inteiro k igual zero (linha 2) para auxiliar na identificação unívoca dos conjuntos de generalidade e a determinar a posição na qual tais conjuntos são inseridos em O . Posteriormente, o primeiro conjunto de generalidade de O , o conjunto G_0 , é criado recebendo as normas do conjunto G^{min} (linha 3). Então, a partir de G_0 , o algoritmo 4 constrói, de forma incremental, os demais conjuntos de generalidade de O (da linha 4 a 13). Para isto, a partir de um conjunto G_k , um conjunto de generalidade G_{k+1} é construído, tal que, para cada norma $n_i \in G_{k+1}$ existe uma norma $n_j \in G_k$, com $n_i * n_j$ (conforme a definição 45). Precisamente, o conjunto G_k corrente é armazenado na posição k da sequência O (linha 5) e um conjunto \mathcal{N}_g , inicialmente vazio, é criado (linha 6) para auxiliar na construção do conjunto G_{k+1} ao armazenar as normas de G_{k+1} temporariamente. Em seguida, para cada norma $n_j \in G_k$ (linha 7) e $n_i \in \mathcal{N}$ (linha 8) é determinado se o par (n_i, n_j) está em \mathcal{R} (linha 9). Se o par estiver, então $n_i \underset{g}{>} n_j$. Todavia, para que n_i e n_j sejam contíguas (i.e, $n_i * n_j$), não deve haver uma norma $n_k \in \mathcal{N}$, tal que, $n_i \underset{g}{>} n_k \underset{g}{>} n_j$. A fim de averiguar se n_i e n_j são contíguas, o algoritmo 4 analisa se não existe um norma $n_k \in \mathcal{N}$, tal que, os pares (n_i, n_k) e (n_k, n_j) pertencem a \mathcal{R} (linha 10). Se tal norma não existir, tem-se que $n_i * n_j$ e a norma n_i é adicionada ao conjunto \mathcal{N}_g (linha 11). Caso contrário, tem-se que n_i e n_j não são contíguas. Quando todas as normas do conjunto G_k são analisadas, todas as normas contíguas às normas de G_k são obtidas. Então, o inteiro k é incrementado (linha 12), um conjunto de generalidade G_{k+1} é criado recebendo as normas de \mathcal{N}_g (linha 13). Por sua vez, a construção da sequência O ocorre até que o conjunto G^{max} seja obtido (conforme a definição 43). Por fim, o algoritmo 4 retorna a sequência obtida (linha 14).

Exemplo 34 Considere que desejamos obter a sequência semântica O_3 do exemplo 25. Para isto, considere também que o algoritmo 3 considera o espaço das normas $\mathcal{N} = \{n_1, \dots, n_7\}$ do exemplo 24 e que, a partir dos conjuntos Sa^n para cada norma $n \in \mathcal{N}$, o algoritmo 3 obtém a relação $\mathcal{R} = \{(n_7, n_4), (n_7, n_5), (n_4, n_1), (n_5, n_2), (n_6, n_3)\}$ que, por sua vez, está ilustrada pelos grafos do exemplo 24. Então, ao analisar a relação \mathcal{R} , o algoritmo 3 identifica que as normas n_1, n_2 e n_3 não aparecem no lado esquerdo de nenhum par de \mathcal{R} . Consequentemente, tais normas são as mais específicas do espaço das normas. Desta forma, o algoritmo 3 obtém o conjunto $G^{min} = \{n_1, n_2, n_3\}$. Em seguida, o algoritmo 4 inicia a construção da sequência semântica O_3 obtendo, para $k = 0$, o conjunto $G_0 = G^{min}$. Posteriormente, ao analisar as normas de G_0 , o algoritmo 4 identifica que os pares (n_4, n_1) , (n_5, n_2) e (n_6, n_3) da relação \mathcal{R} são constituídos por normas contíguas. Considerando tais pares, o algoritmo 4 obtém as normas generalizadoras e constrói, para $k = 1$, o conjunto $G_1 = \{n_4, n_5, n_6\}$. Então, a partir de G_1 , o algoritmo 4 identifica que os pares (n_7, n_4) e (n_7, n_5) da relação \mathcal{R} são, também, constituídos por normas contíguas. Ao obter as normas generalizadoras de tais pares, o algoritmo 4 constrói, para $k = 2$, o conjunto $G_2 = \{n_7\}$. Com todas as normas foram analisadas, tem-se que $G^{max} = G_2$ e $O_3 = \{G_0, G_1, G_2\}$.

A figura 28 ilustra a sequência semântica O_3 , onde as relações de generalidade entre as normas são representadas nos mesmos moldes do exemplo 24.

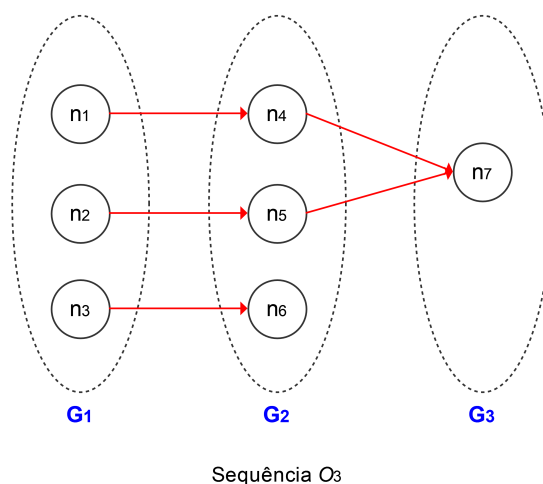


Figura 28 – Exemplo de aplicação dos algoritmos 3 e 4.

Por sua vez, o método *visitar*, descrito pelo algoritmo 5, recebe como entrada uma sequência O de conjuntos de generalidade, uma estratégia de visita v , um índice *first* de um conjunto de generalidade da sequência O por onde a visita inicia, um conjunto de objetivos do sistema \mathcal{G} , um conjunto de histórias do sistema \mathcal{H} , um conflito c e um estado inicial s_0 .

Algoritmo 5: visitar

Data: $O, v, first, \mathcal{G}, \mathcal{H}, c, s_0$

```

1  $k \leftarrow first$ 
2 while  $k \neq null$  do
3    $G_k \leftarrow O[k]$ 
4   for  $n_i \in G_k$  do
5      $\mathcal{H}_{n_i}^{suc}, \mathcal{H}_{n_i}^{conf} \leftarrow getHistoriasNorma(n_i, c, \mathcal{G}, \mathcal{H})$ 
6     if  $\mathcal{H}_{n_i}^{suc} \neq \emptyset \wedge \mathcal{H}_{n_i}^{conf} = \emptyset$  then // definição 27
7       return  $n_i$ 
8    $k \leftarrow v(k)$ 
9 return  $null$ 

```

O algoritmo 5 funciona de forma similar ao algoritmo 2. Entretanto, em vez de analisar as normas na ordem na qual elas são criadas, analisa de acordo com a ordem na qual os conjuntos de generalidade da sequência O são visitados. Para isto, o algoritmo obtém o índice k do primeiro conjunto de generalidade a ser visitado (linha 1). Então, enquanto o valor de k for diferente de $null$ (linha 2), isto é, existem conjuntos de generalidade a serem visitados, a busca por normas eficientes prossegue. Durante a busca, o conjunto G_k que ocupa a posição k da sequência O é obtido (linha 3) e as normas que o constituem são analisadas de forma análoga a como ocorre no algoritmo 2 (linhas 4 a 6). Caso uma norma eficiente n_i seja encontrada, então o algoritmo retorna-a (linha 7). Caso contrário, o próximo conjunto de generalidade a ser visitado é obtido. Para isto, ao receber o identificador k do conjunto G_k , a estratégia de visita v retorna o identificador do próximo conjunto de generalidade (linha 8). Caso todos os conjuntos de generalidade tenham sido visitados e nenhuma norma eficiente tenha sido encontrada, então a estratégia v retorna $null$ e o algoritmo encerra.

Embora diferentes estratégias de visita possam ser empregadas, nós destacamos quatro delas, onde as duas primeiras consideram a hipótese 1 e as regiões das sequências onde as normas eficientes são observadas nos experimentos do capítulo 4 como heurísticas para guiar a busca. A primeira estratégia, a qual denominamos de $G^{cen} G^{max}$, especifica que a visita inicia pelo conjunto G^{cen} de uma sequência de conjuntos de generalidade para, então, visitar os conjuntos do lado direito e, posteriormente, visitar os conjuntos do lado esquerdo. Por outro lado, a segunda estratégia, a qual denominamos de *Zigzag*, especifica que a visita também inicia pelo conjunto G^{cen} para, então, visitar alternativamente (em zigzag) os conjuntos do lado direito e esquerdo. Para isto, visita-se primeiramente o conjunto à direita de G^{cen} e, em seguida, o conjunto à esquerda de G^{cen} até que os conjuntos G^{max} e G^{min} sejam ambos visitados, respectivamente. Para fins de comparação, as outras duas estratégias de visita iniciam em locais distintos do meio das sequências. A terceira estratégia, a qual denominamos de $G^{min} G^{max}$, especifica que a visita inicia pelo conjunto G^{min} para então

visitar os demais conjuntos sucessivamente. Por fim, a quarta estratégia, a qual denominamos de *Aleatória*, visita os conjuntos de generalidade aleatoriamente permitindo que normas eficientes sejam encontradas ao acaso.

Exemplo 35 A figura 29 ilustra a aplicação das estratégias de visita sobre uma sequência semântica hipotética, onde as elipses representam os conjuntos de generalidade, com G^{min} , G^{cen} e G^{max} destacados com rótulos em azul, a aresta em vermelho sólida representa o primeiro conjunto a ser visitado e as arestas em vermelho tracejadas representam a ordem na qual os demais conjuntos são visitados. Para auxiliar na visualização da ordem de visita dos conjuntos de generalidade, nós adicionamos nas pontas das arestas números ordinais.

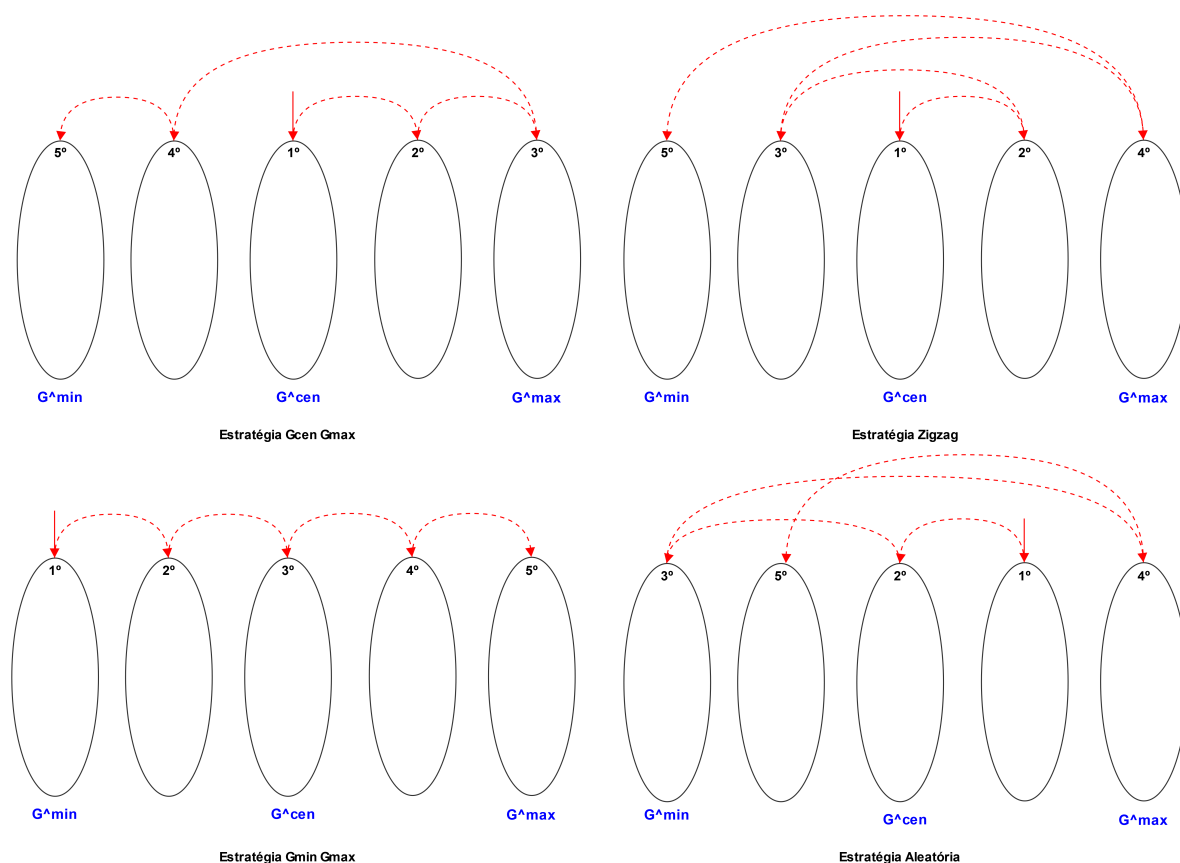


Figura 29 – Exemplos de aplicação das estratégias de visita.

Embora o algoritmo 3, que faz uso dos métodos *getSequencia* (algoritmo 4) e *visitar* (algoritmo 5), organize o espaço das normas de forma a viabilizar a busca por normas eficientes em regiões onde julgamos ser mais provável encontrar tais normas, o seu custo computacional é alto. Primeiramente, o algoritmo 3 requer construir o conjunto dos estados do sistema S para poder determinar o conjunto Sa^n dos estados

onde cada norma $n \in \mathcal{N}$ é aplicável e, então, a generalidade das normas. Além disto, o algoritmo 3 necessita realizar o produto cartesiano $\mathcal{N} \times \mathcal{N}$ (da linha 7 a 8) para construir a relação de ordem parcial \mathcal{R} e uma sequência semântica O . Neste contexto, um aprimoramento a ser aplicado sobre o algoritmo 3 é *determinar a generalidade das normas através de uma aproximação*. Desta forma, podemos evitar construir a relação \mathcal{R} e, conseqüentemente, os conjuntos \mathcal{S} e \mathcal{S}_a . Embora as aproximações da generalidade das normas possam ser determinadas de diferentes formas, uma possibilidade é aplicar as funções de generalidade (conforme a definição 39) que exploram certas características sintáticas da especificação das normas para atribuir-lhes uma generalidade como um valor real (conforme a definição 38).

Embora o algoritmo 3 construa espaços cujas normas são estritamente unground, o que auxilia a obter espaço menores, o espaço das normas ainda pode ser potencialmente grande, portanto, medidas para construir espaços ainda menores podem ser necessárias para economizar tempo e esforço computacional na síntese de normas eficientes. Dado que o espaço das normas pode ser constituído por diferentes normas que são semanticamente equivalentes, um segundo aprimoramento a ser aplicado sobre o algoritmo 3 é *criar espaços cujas normas são mínimas* (conforme a definição 17). Desta forma, o espaço das normas pode tornar-se menor, pois, para cada conjunto de normas semanticamente equivalentes, ele contém apenas a respectiva norma mínima. Conseqüentemente, se uma norma mínima não for útil ao sistema, economizamos tempo e esforço ao evitarmos a análise das outras normas semanticamente equivalentes que, por sua vez, também não serão consideradas úteis.

Os aprimoramentos destacados para o algoritmo 3 estão contemplados pelo algoritmo 6 que realiza uma busca por normas eficientes sobre uma sequência aproximada. Para isso, ele recebe como entrada uma estratégia de visita v , uma função de generalidade f , um conjunto potência $2^{\mathcal{L}^{uaf}}$, um conjunto de ações \mathcal{A} , um conjunto de instâncias de ações $\mathcal{I}^{\mathcal{A}}$, um estado inicial s_0 , um conjunto de objetivos do sistema \mathcal{G} , um conflito c e um timeout $\Delta \in \mathbb{Z}_+^*$ (inteiros positivos).

Algoritmo 6: Aprimoramento 3

Data: $v, f, 2^{\mathcal{L}^{uaf}}, \mathcal{A}, \mathcal{I}^{\mathcal{A}}, s_0, \mathcal{G}, c, \Delta$

- 1 $O \leftarrow \text{getSequenciaAproximada}(2^{\mathcal{L}^{uaf}}, \mathcal{A}, f)$
- 2 $first \leftarrow \text{getIndice}(O)$
- 3 $\mathcal{H} \leftarrow \text{getHistorias}(s_0, \mathcal{I}^{\mathcal{A}}, \Delta)$
- 4 $n_{eff} \leftarrow \text{visitar}(O, v, first, \mathcal{G}, c, \mathcal{H})$
- 5 **return** n_{eff}

Primeiramente, através do método *getSequenciaAproximada* (linha 1), o algoritmo 6 obtém uma sequência de conjuntos de generalidade aproximada O (conforme a definição 43). Em seguida, através do método *getIndice* (linha 2), o algoritmo 6

Algoritmo 7: getSequenciaAproximada

```

Data:  $2^{\mathcal{L}^{uaf}}$ ,  $\mathcal{A}$ ,  $f$ 
1  $\mathcal{X} \leftarrow \emptyset$ 
2 for  $(\varphi, a) \in (2^{\mathcal{L}^{uaf}} \times \mathcal{A})$  do
3    $n \leftarrow \varphi \xrightarrow{p} a$ 
4   if  $\varphi^n \cap Pre^{a^n} = \emptyset$  then // definição 17
5      $k \leftarrow f(n)$ 
6      $G_k \leftarrow getConjuntoGeneralidade(\mathcal{X}, k)$ 
7     if  $G_k = \emptyset$  then
8        $G_k \leftarrow \{n\}$ 
9        $\mathcal{X} \leftarrow \mathcal{X} \cup G_k$ 
10    else
11       $G_k \leftarrow G_k \cup \{n\}$ 
12  $O \leftarrow getOrdenacao(\mathcal{X})$ 
13 return  $O$ 

```

obtem o identificador *first* do primeiro conjunto de generalidade da sequência O a ser visitado pela estratégia v , obtém, através do método *getHistorias* (linha 3), o conjunto de histórias do sistema \mathcal{H} para, então, aplicar o método *visitar* (linha 4) sobre a sequência O a fim de encontrar uma norma eficiente. Por fim, caso tal norma exista, o algoritmo 6 retorna a norma eficiente encontrada (linha 5). Por sua vez, o método *getSequenciaAproximada*, descrito pelo algoritmo 7, não constrói uma relação \mathcal{R} para armazenar as relações de generalidade entre as normas, logo, não é possível obter antecipadamente o conjunto G^{min} para que, a partir dele, uma sequência de conjuntos de generalidade seja construída incrementalmente assim como ocorre no algoritmo 4.

Alternativamente, o método *getSequenciaAproximada* (algoritmo 7) constrói os conjuntos de generalidade de uma sequência aproximada à medida que as normas são sintetizadas, de modo que, normas com o mesmo valor de generalidade (conforme a definição 38) são inseridas no mesmo conjunto. Para isto, o algoritmo 7 defini, inicialmente, um conjunto vazio \mathcal{X} (linha 1) para armazenar os conjuntos de generalidade. Em seguida, o algoritmo 7 inicia a criação das normas estritamente unground (da linha 2 a 3), tal que, para cada norma n é determinado se a norma é mínima (linha 4). Isto é feito verificando se a interseção entre Pre^{a^n} (as pré-condições de execução da ação a^n proibida por n) e φ^n (o contexto de n) é vazia³. Se a norma for mínima, então é determinado o valor de generalidade k da norma através da função de generalidade f (linha 5). Embora possam existir distintas funções de generalidade, neste trabalho

³ De acordo como o espaço das normas é criado (definição 15), para uma dada ação $a \in \mathcal{A}$, existem normas que possuem de 0 a $|Pre^a|$ fórmulas atômicas de Pre^a . Dado um conjunto de normas semanticamente equivalentes, a norma mínima é aquela que possui 0 fórmulas atômicas em comum com as pré-condições de execução da ação que as normas do conjunto proibem.

nós empregamos as funções NGI, NGJ, NGK e NGF apresentadas no capítulo 4 (conforme as definições 46, 47, 48 e 49, respectivamente). De acordo com k , o método *getConjuntoGeneralidade* busca pelo conjunto de generalidade G_k no conjunto \mathcal{X} (linha 6)⁴. Caso G_k não exista em \mathcal{X} , então o método retorna um conjunto G_k vazio, a norma n é adicionada a G_k (linha 8) e G_k é adicionado a \mathcal{X} (linha 9). Caso contrário, a norma n é simplesmente adicionada ao conjunto G_k previamente existente (linha 11). Quando todas as normas mínimas tiverem sido inseridas em seus respectivos conjuntos de generalidade, através do método *getOrdenacao* (linha 12), a sequência aproximada O é ordenada crescentemente de acordo com os identificadores k dos seus conjuntos de generalidade. Desta forma, através de O , as normas estão ordenadas, aproximadamente, das mais específicas para as mais gerais. Por fim, a sequência O é retornada (linha 13).

Exemplo 36 *Considere que o método *getSequenciaAproximada* sintetiza as normas apresentadas no exemplo 26, $\mathcal{N} = \{n_1, \dots, n_{10}\}$. Ao determinar quais normas são mínimas, o método verifica que as normas n_2, n_4, n_6, n_7, n_8 e n_{10} satisfazem as condições da definição 17. Desta forma, em vez de construir uma sequência aproximada considerando um espaço com 10 normas, o método a constrói considerando unicamente 6 normas. Por sua vez, a sequência resultante depende da função de generalidade utilizada. A figura 30 ilustra as sequências aproximadas dos exemplos 27, 28, 29 e 30 considerando o espaço de normas mínimas $\mathcal{N}' \subset \mathcal{N}$, tal que, $\mathcal{N}' = \{n_2, n_4, n_6, n_7, n_8, n_{10}\}$.*

Ao evitar comparar as normas para determinar a generalidade delas, o algoritmo 6 pode economizar tempo para obter uma sequência de conjuntos de generalidade. Contudo, para isto, o algoritmo 6 necessita arcar com o custo de ordenar o conjunto \mathcal{X} . Além do mais, como as sequências obtidas são aproximações, é possível que elas divirjam das sequências semânticas em número de conjuntos de generalidade, número de normas por conjunto e localização das normas eficientes. No entanto, é possível que tais diferenças não constituam problemas na busca pelas normas eficientes, pois conforme discutido no capítulo 4, certas funções são capazes de mapear as normas eficientes em torno do meio para a direita nas sequências aproximadas, o que viabiliza aplicarmos estratégias de visita que exploram inicialmente tais regiões. Além disso, certas sequências podem possuir uma média menor de normas por conjunto de generalidade. Desta forma, é possível que dentre G^{cen} e G^{apex} , o total de normas a ser analisado seja menor do que em sequências onde a média de normas nos conjuntos de generalidade é maior. Logo, este é um fator que pode aumentar o

⁴ Basicamente, através de uma iteração sobre \mathcal{X} , pode-se, ao obter um conjunto G_k , verificar se o valor de generalidade k associado ao identificador do conjunto é igual ao valor de k passado como argumento ao método *getConjuntoGeneralidade*.

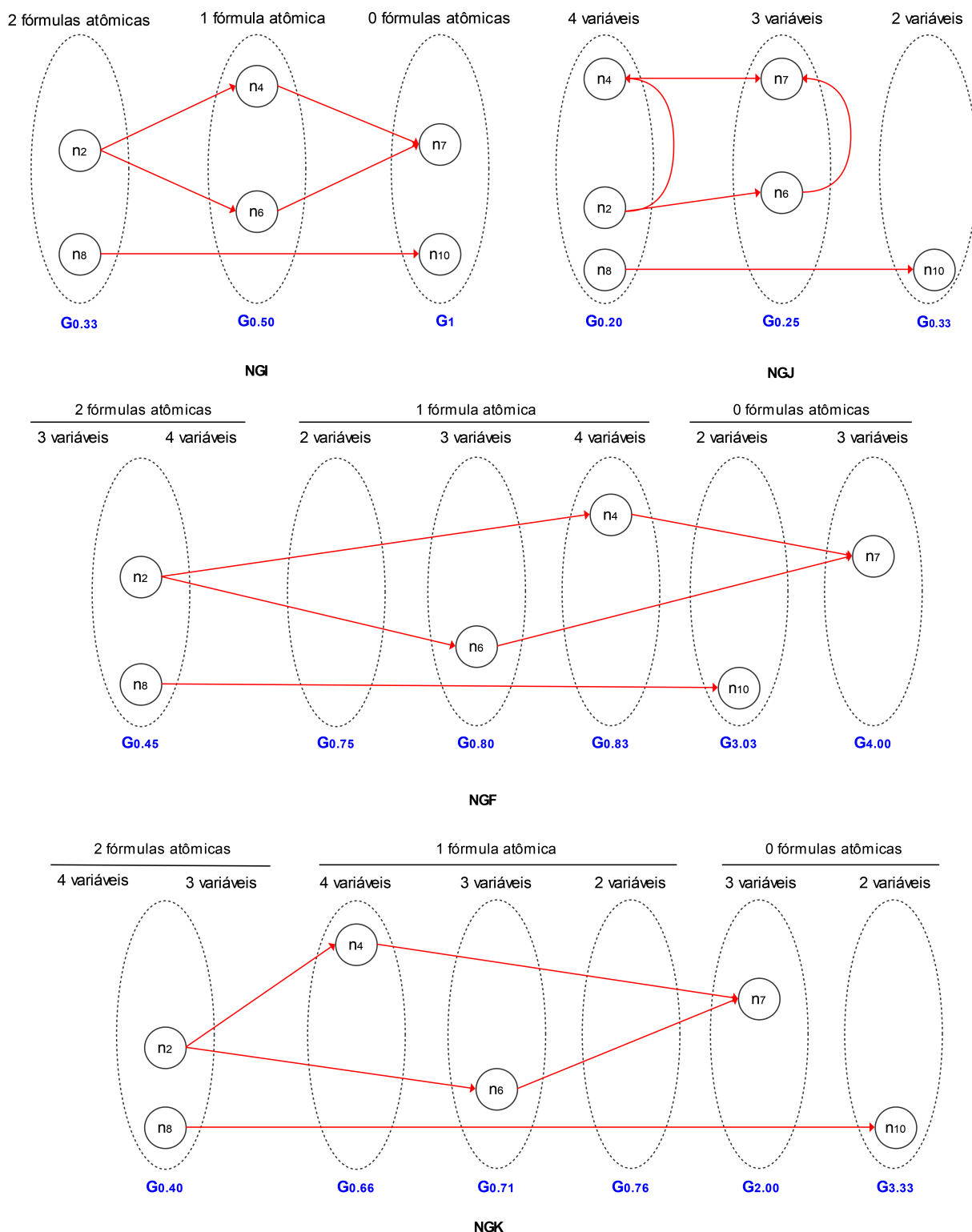


Figura 30 – Sequências aproximadas com normas mínimas.

desempenho da busca pelas normas eficientes. Todavia, estes são pontos a serem investigados com maior profundidade nos experimentos da próxima seção.

Além do mais, o algoritmo 6 é um dos algoritmos que arca com o custo de obter o conjunto de histórias do sistema \mathcal{H} para encontrar as normas eficientes. A fim

de evitar que o conjunto \mathcal{H} seja determinado completamente, um aprimoramento que podemos aplicar sobre o algoritmo 6 é *realizar a busca por normas eficientes através das normas quase eficientes*. As normas quase eficientes (conforme a definição 29) relaxam a necessidade de que nenhuma história de conflito possa ocorrer quando o sistema executa com tais normas. Embora possam existir diferentes formas de encontrar normas quase eficientes, uma possibilidade para isto é, conforme discutido no capítulo 3, calcular as taxas de histórias de sucesso e conflito das normas considerando um subconjunto de histórias $\mathcal{H}' \subset \mathcal{H}$ (conforme a definição 30). Neste contexto, as normas quase eficientes que satisfazem certos limiares podem ser consideradas apropriadas para regularem os agentes. Diferentes estratégias podem ser empregadas para que os subconjuntos das histórias das normas sejam obtidos como amostragem, geração aleatória de histórias ou ocorrência de histórias em tempo de execução, por exemplo. Em especial, consideramos a *terceira* opção mais adequada, pois através dela podemos verificar quais histórias são mais propensas a acontecer e calcular as taxas das histórias de sucesso e conflito das normas de uma forma mais realística.

Todavia, como diferentes normas de um espaço de normas podem ser quase eficientes, dado que para isto basta que as normas selecionem, pelo menos, uma história de sucesso, a região escolhida para iniciar a busca pode impactar no quão capazes as normas são em evitar conflitos e manter os objetivos do sistema alcançáveis. Conforme discutido no capítulo 4, considerando que à medida que as normas se afastam de G^{cen} em direção a G^{min} elas se tornam mais específicas e a taxa de conflito aumenta, à medida que as normas se afastam de G^{cen} em direção a G^{max} as normas se tornam mais gerais e a taxa de timeout aumenta, e em torno de G^{cen} as normas apresentam a maior taxa de sucesso, o que corrobora para que tais normas possuam uma generalidade balanceada, podemos seguir a hipótese 1 e aplicar estratégias de visita que iniciam a busca por normas quase eficientes a partir do meio das sequências aproximadas. Desta forma, julgamos plausível encontrar normas quase eficientes que sejam eficientes. Assim, dado um subconjunto de histórias $\mathcal{H}' \subset \mathcal{H}$, caso a taxa de sucesso de uma norma quase eficiente $n \in \mathcal{N}$ seja maior que zero e a sua taxa de conflito seja zero, tem-se que $\mathcal{H}'^{suc_n} \neq \{ \}$ e $\mathcal{H}'^{conf_n} = \{ \}$ o que, por sua vez, satisfaz as condições para uma norma ser eficiente (conforme a definição 27).

Contudo, devido à escolha racional e às preferências dos agentes, é possível que a ocorrência de histórias em tempo de execução resulte na mesma história ao fixarmos o estado inicial e os objetivos do sistema. Todavia, conforme o estado inicial e os objetivos mudam, o comportamento dos agentes tende a variar e, assim, distintas histórias ocorrer. Desta forma, é possível que distintas situações que tornem o alcance dos objetivos mais complexo ou levam a ocorrência de conflitos surjam, o que nos permite avaliar com mais propriedade o desempenho das normas em regular os agentes apropriadamente em diferentes configurações de um SMAn. Para isto, podemos

executar o SMAn para um conjunto de estados iniciais S_0 e um conjunto de objetivos do sistema \mathcal{G}^* , conforme as definições 2 e definição 5, respectivamente.

O aprimoramento destacado para o algoritmo 6 está contemplado pelo algoritmo 8 que realiza uma busca por normas eficientes através do encontro de normas quase eficientes em sequências aproximadas. Para isso, ele recebe como entrada uma estratégia de visita v , uma função de generalidade f , um conjunto potência $2^{\mathcal{L}^{uaf}}$, um conjunto de ações \mathcal{A} , um conjunto de estados iniciais S_0 , um conjunto de conjuntos de objetivos do sistema \mathcal{G}^* , um conflito c , um limiar $tsuc \in [0,100]$ que estabelece o valor mínimo de taxa de sucesso, um limiar $tconf \in [0,100]$ que estabelece o valor máximo de taxa de conflito, um modelo de SMAn⁵ e um timeout $\Delta \in \mathbb{Z}_+^*$.

Algoritmo 8: Aprimoramento 4

Data: $v, f, 2^{\mathcal{L}^{uaf}}, \mathcal{A}, S_0, \mathcal{G}^*, c, tsuc, tconf, SMAn, \Delta$

- 1 $O \leftarrow getSequenciaAproximada(2^{\mathcal{L}^{uaf}}, \mathcal{A}, f)$
- 2 $first \leftarrow getIndice(O)$
- 3 $n_{qefic} \leftarrow visitarComTaxas(O, first, v, S_0, \mathcal{G}^*, c, tsuc, tconf, SMAn, \Delta)$
- 4 **return** n_{qeff}

O algoritmo 8 obtém, primeiramente, uma sequência de conjuntos de generalidade aproximada O através do método *getSequenciaAproximada* (linha 1) para em seguida obter, através do método *getIndice* (linha 2), o índice *first* do primeiro conjunto de generalidade da sequência O a ser visitado pela estratégia v . Posteriormente, o algoritmo executa o método *visitarComTaxas*, descrito através do algoritmo 9, sobre a sequência O (linha 3) para encontrar uma norma quase eficiente n_{qefic} cujas taxas de histórias de sucesso e conflito satisfaçam os limiares $tsuc$ e $tconf$, respectivamente. Por fim, a norma quase eficiente encontrada é retornada (linha 4).

Por sua vez, o método *visitarComTaxas* é responsável por realizar a busca por uma norma quase eficiente que satisfaça um conjunto de limiares ao explorar uma sequência aproximada. Para isto, o método recebe como entrada do algoritmo 8 a estratégia de visita v , a sequência aproximada O , o índice *first*, os conjuntos S_0 e \mathcal{G}^* , os limiares $tsuc$ e $tconf$ e o timeout Δ . Primeiramente, o método *visitarComTaxas* cria uma variável nc (linha 1) para armazenar a norma quase eficiente candidata a ser retornada. Como inicialmente tal norma não existe, a variável nc recebe *null*. Em seguida, o método cria as variáveis $taxasuc_{nc}$ (linha 2) e $taxaconf_{nc}$ (linha 3) para armazenar, respectivamente, as taxas das histórias de sucesso e conflito da norma nc , onde assume inicialmente que os seus valores são, respectivamente, 0 e 100. Desta forma, a primeira norma a ser analisada é considerada uma norma candidata

⁵ Conforme dito no capítulo 4, um modelo de SMAn define o SMAn para um determinado domínio. Este modelo inclui informações sobre os agentes, seus objetivos, os objetivos do sistema, o ambiente em que os agentes operam, etc.

Algoritmo 9: visitarComTaxas

Data: O , $first$, v , S_0 , \mathcal{G}^* , c , $tsuc$, $tconf$, SMA_n , Δ

```

1  $nc \leftarrow null$ 
2  $taxasuc_{nc} \leftarrow 0$ 
3  $taxaconf_{nc} \leftarrow 100$ 
4  $k \leftarrow first$ 
5  $total \leftarrow |S_0| \times |\mathcal{G}^*|$ 
6 while  $k \neq null$  do
7    $G_k \leftarrow O[k]$ 
8   for  $n \in G_k$  do
9      $numsuc_n, numconf_n, taxasuc_n, taxaconf_n \leftarrow 0$ 
10    for  $s_0 \in S_0$  do
11      for  $\mathcal{G} \in S_{\mathcal{G}}$  do
12         $h \leftarrow SMA_n.executar(s_0, \mathcal{G}, n, c, \Delta)$ 
13         $s \leftarrow h.getEstadoFinal()$ 
14        if  $\mathcal{G} \subseteq s$  then
15           $numsuc_n \leftarrow numsuc_n + 1$ 
16        else
17          if  $c \triangleright s$  then
18             $numconf_n \leftarrow numconf_n + 1$ 
19         $taxasuc_n \leftarrow numsuc_n / total$ 
20         $taxaconf_n \leftarrow numconf_n / total$ 
21        if  $taxasuc_n \geq tsuc \wedge taxaconf_n \leq tconf$  then
22          return  $n$ 
23        else
24          if  $taxasuc_n \geq taxasuc_{nc} \wedge taxaconf_n \leq taxaconf_{nc}$  then
25             $taxasuc_{nc} \leftarrow taxasuc_n$ 
26             $taxaconf_{nc} \leftarrow taxaconf_n$ 
27             $nc \leftarrow n$ 
28     $k \leftarrow v(k)$ 
29 return  $nc$ 

```

a ser retornada. Contudo, à medida que as normas são analisadas, a variável nc é atualizada quando identifica-se que existe uma norma mais apropriada para regular os agentes. Então, é criada a variável k (linha 4), inicialmente com $first$, para armazenar o identificador dos conjuntos de generalidade a serem visitados no processo de busca. Por fim, é criada a variável $total$ (linha 5) para armazenar o número de vezes que o SMA_n deve ser simulado na análise de cada norma.

Então, o método dá início à busca de uma norma quase eficiente que satisfaça os limiares $tsuc$ e $tconf$. Para isto, enquanto k for diferente de $null$ (linha 6), a sequência O é percorrida, o conjunto de generalidade G_k corrente é obtido (linha 7) e, para cada norma $n \in G_k$ (linha 8), são criadas as variáveis $numsuc_n$, $numconf_n$, $taxasuc_n$ e $taxaconf_n$, inicialmente com o valor 0 (linha 9), onde $numsuc_n$ e $numconf_n$ são para armazenar, respectivamente, o número de simulações do SMA_n em que ocorrem histórias de sucesso e conflito ao executar com a norma n , enquanto $taxasuc_n$ e $taxaconf_n$ são para armazenar, respectivamente, as taxas de histórias de sucesso e conflito de n . Posteriormente, para cada estado inicial $s_0 \in S_0$ (linha 10) e conjunto de objetivos do sistema $\mathcal{G} \in \mathcal{G}^*$ (linha 11), o SMA_n é, através do método *executar* (linha 12), simulado considerando, além de tais conjuntos, o conflito c e o timeout Δ . Na simulação do SMA_n , os objetivos do sistema são partilhados entre os agentes. Nesse sentido, os agentes têm como objetivos, os objetivos provenientes do sistema e, de acordo com o cenário de aplicação, objetivos individuais. A simulação do SMA_n resulta em uma história $h \in \mathcal{H}$ cujo estado final s é obtido através do método *getEstadoFinal* (linha 13). Então, é verificado se s é um estado de objetivos (linha 14) ou de conflito (linha 17). Se a primeira situação ocorrer, então h é uma história de sucesso e a variável $numsuc_n$ é incrementada (linha 15). Senão, h é uma história de conflito e a variável $numconf_n$ é incrementada (linha 18)⁶.

Quando o SMA_n é simulado para todos os estados iniciais e objetivos considerando a norma n , as taxas de histórias de sucesso e conflito da norma são calculadas (linhas 19 e 20). Em seguida, é verificado se a norma n satisfaz os limiares $tsuc$ e $tconf$ (linha 21). Se for o caso, a norma n é retornada (linha 22). Caso contrário, é determinado se o desempenho de n em regular os agentes é melhor ou igual ao desempenho da norma nc (linha 24). Para isto, a norma n deve possuir uma taxa de sucesso maior ou igual a taxa de sucesso de nc e uma taxa de conflito menor ou igual a taxa de conflito de nc . Assim, a norma n passa a ser a norma quase eficiente candidata a retornar, tal que, os as variáveis atreladas a norma nc são atualizadas para armazenar os valores das variáveis atreladas a norma n (linhas 25 a 27). Por fim, quando todas as normas do conjunto de generalidade corrente G_k são analisadas, a variável k é, através da aplicação da estratégia de visita v (linha 28), atualizada para o índice do

⁶ Embora possam ocorrer histórias de timeout, conforme a definição 29, elas não são contabilizadas na análise das normas quase eficientes.

próximo conjunto de generalidade a ser visitado. Se nenhuma norma quase eficiente que satisfaça os limiares seja encontrada, então é retornada a norma candidata que possui a maior taxa de sucesso e a menor taxa de conflito (linha 29).

Exemplo 37 *A fim de sintetizarmos normas eficientes para a simplificação do cenário do tráfego de carros através do algoritmo 8, podemos considerar, primeiramente, um conjunto potência $2^{\mathcal{L}^{uaf}}$, um conjunto de ações A e uma função de generalidade f para a construção de uma sequência aproximada O e, então, aplicar uma estratégia de visita v sobre O , tal que, para cada norma analisada, podemos simular o SMAn considerando um conjunto S_0 de estados iniciais que variam os locais de partida dos agentes, um conjunto \mathcal{G}^* de objetivos do sistema que variam os locais de destino dos agentes, um conflito $c = \{at(Ag_1, L_2), at(Ag_2, L_2)\}$ para indicar que dois agentes distintos Ag_1 e Ag_2 estão em colisão ao ocuparem um mesmo local L_2 , um timeout longo o suficiente para que os agentes possam satisfazer os objetivos do sistema como $|Agentes| * 2^{|Vertices|}$ ⁷, uma taxa de sucesso $tsuc$ e uma taxa de conflito $tconf$. De acordo com os experimentos da seção 4.2.1, para uma grid 3×3 com 5 agentes, podemos considerar $tsuc = 90\%$ e $tconf = 0\%$. Desta forma, o algoritmo 8 pode sintetizar a norma eficiente $n = \{at(Ag_2, L_2)\} \xrightarrow{p} move(Ag_1, L_1, L_2)$.*

Contudo, como o espaço das normas é composto por normas que combinam todas as fórmulas atômicas estritamente unground e ações do sistema (conforme a definição 15), então, é possível que diferentes normas do espaço não sejam úteis, pois combinam tais elementos sem critério. Dessa forma, enquanto diferentes normas não são aplicáveis em estados onde os agentes necessitam ser regulados para evitar conflitos, outras normas são aplicáveis em tais estados, contudo, os conflitos ainda ocorrem, pois elas não especificam corretamente os comportamentos que necessitam ser regulados para que os conflitos sejam evitados. Além disso, existem normas que proíbem comportamentos que os agentes não tendem a apresentar em tempo de execução e normas que inviabilizam o alcance dos objetivos do sistema. Logo, o algoritmo 8 pode gastar um tempo considerável da busca por normas eficientes na análise de normas que não possuem utilidade. A fim de reduzir o espaço das normas a um conjunto constituído por normas melhor alinhadas aos comportamentos dos agentes que necessitam de regulação, um aprimoramento a ser aplicado ao algoritmo 8 é *sintetizar normas considerando os contextos que levam os agentes a situações de conflito*. Para isto, uma possibilidade é analisar as histórias de conflito do sistema. Todavia, como podem existir diversas histórias de conflitos e considerando que certas histórias são mais prováveis de acontecer do que outras (devido à escolha racional e

⁷ Conforme mencionado na seção 4.2.1, *Agentes* e *Vertices* referem-se, respectivamente, ao conjunto dos agentes e vértices da grid, assim, cada agente tem $2^{|Vertices|}$ passos para alcançar o seu local de destino.

preferência dos agentes, conforme dito anteriormente), podemos simular o *SMAN* e analisar a história que ocorre. Caso a história seja de conflito, conforme discutido no capítulo 3, podemos realizar *backtracking* sobre ela (conforme a definição 35) e, para cada estado visitado, *analisar a alcançabilidade dos objetivos do sistema*.

O aprimoramento destacado para o algoritmo 8 está contemplado pelo algoritmo 10 que realiza uma busca por normas eficientes através do encontro de normas quase eficientes em sequências aproximadas formadas por normas melhor alinhadas aos comportamentos dos agentes que necessitam de regulação. Para isso, ele recebe como entrada uma estratégia de visita v , uma função de generalidade f , um conjunto potência $2^{\mathcal{L}^{uaf}}$, um conjunto de ações \mathcal{A} , um conjunto de instâncias de ações $\mathcal{I}^{\mathcal{A}}$, um conjunto de estados iniciais \mathcal{S}_0 , um conjunto de conjuntos de objetivos do sistema \mathcal{G}^* , um conflito c , um limiar $tsuc \in [0,100]$ que estabelece o valor mínimo de taxa de sucesso, um limiar $tconf \in [0,100]$ que estabelece o valor máximo de taxa de conflito, um modelo de *SMAN*, um timeout $\Delta \in \mathbb{Z}_+^*$, um estado inicial $s_0 \in \mathcal{S}_0$ e um conjunto de objetivos $\mathcal{G} \in \mathcal{G}^*$ para simular o *SMAN* a fim de obter uma história sobre a qual será feita a análise prévia para a obtenção de um espaço de normas.

Algoritmo 10: Síntese com backtracking - Aprimoramento 5

```

Data:  $v, f, \mathcal{L}^{gaf}, \mathcal{A}, \mathcal{I}^{\mathcal{A}}, \mathcal{S}_0, \mathcal{G}^*, c, tsuc, tconf, SMAN, \Delta, s_0, \mathcal{G}$ 
1  $h \leftarrow SMAN.executar(s_0, \mathcal{G}, null, c, \Delta)$ 
2  $s \leftarrow s.getEstadoFinal()$ 
3 if  $c \triangleright s$  then
4    $m \leftarrow |h| - 1$ 
5   while  $m \geq 0$  do
6      $s_v \leftarrow h.getTupla(m).getEstadoPai()$ 
7     if  $s_v \neq null$  then
8        $t_v \leftarrow analisar(s_v, \mathcal{G}, \mathcal{I}^{\mathcal{A}}, c, \Delta - m - 1)$ 
9       if  $t_v \neq []$  then
10         $a_p \leftarrow h.getTupla(m - 1).getAcaoProibitiva()$ 
11         $a_p \leftarrow getAcao(a_p)$ 
12         $O \leftarrow getSequenciaAproximada(\mathcal{L}^{uaf}, \{a_p\}, f)$ 
13         $first \leftarrow getIndex(O)$ 
14         $n \leftarrow visitarComTaxas(v, O, first, \mathcal{S}_0, \mathcal{G}^*, tsuc, tconf)$ 
15        return  $n$ 
16      $m \leftarrow m - 1$ 
17 return  $null$ 

```

Primeiramente, o algoritmo 10 simula, através do método *executar*, o *SMAN* e obtém uma história $h \in \mathcal{H}$ (linha 1). Como nesta etapa não existem normas, o método *executar* recebe como entrada, além dos parâmetros s_0 , \mathcal{G} e Δ , o valor *null* no lugar de uma norma. Logo, os agentes determinam as ações que desejam executar livremente. Em seguida, é obtido o último estado s da história h através do método *getEstadoFinal*

(linha 2) e é verificado se s é um estado de conflito (linha 3). Se for o caso, o algoritmo 10 dá início à síntese com backtracking (conforme a definição 35). Para isto, cria uma variável m (linha 4) inicializada com o índice da última tupla de h (onde a primeira tupla possui índice 0) para auxiliar o algoritmo 10 a percorrer todas as tuplas e assim obter os estados pais a partir dos quais a análise da alcançabilidade é realizada. Assim, enquanto m for maior ou igual a 0 (linha 5) o backtracking é executado. Neste processo, determina-se o estado s_v a ser visitado (linha 6) obtendo, primeiramente, a tupla corrente m e, em seguida, o estado pai na tupla através dos métodos *getTupla* e *getEstadoPai*, respectivamente. Caso o estado pai seja diferente de *null* (linha 7), o método *analisar* (linha 8) realiza a análise da alcançabilidade dos objetivos do sistema. Para isso, o método busca identificar a existência de um traço viabilizador t_v (conforme a definição 32) que inicia em s_v e finda em algum estado de objetivos $s_g \in \mathcal{G}$. Para isto, o método *analisar* necessita do conjunto das fórmulas atômicas ground \mathcal{L}^{gaf} e instâncias de ações \mathcal{I}^A (conforme a definição 4) para construir os estados e ações dos traços que são possíveis de serem expandidos a partir do estado s_v . Todavia, para isto, é importante que o timeout Δ seja considerado. Assim, ao subtrairmos $m - 1$ de Δ , podemos identificar o número de ações que pode ser considerado na análise da alcançabilidade de modo a não exceder o timeout. Neste sentido, ao executar o método *analisar*, o algoritmo 10 passa além de s_v , \mathcal{G} , \mathcal{I}^A e c , $\Delta - m - 1$ como parâmetro.

Posteriormente, é verificado se o traço t_v , que é retornado como um vetor, não é vazio (linha 9). Se t_v for vazio, então a variável m é decrementada (linha 16). Caso contrário, existe um traço que satisfaz os parâmetros passados ao método *analisar* e uma ação proibitiva a_p (conforme a definição 33) da história h deve ser obtida. Para isso, obtém-se, através do método *getTupla*, a tupla $m - 1$ onde encontra-se a ação proibitiva. Em seguida, obtém-se tal ação através do método *getAcaoProibitiva*. Dado que a_p é uma instância de ação (cujos termos são constantes, conforme a definição 4) e as normas são criadas para proibir ações (cujos termos são variáveis, conforme a definição 3), a ação a partir da qual a_p é instanciada é obtida através do método *getAcao* (linha 11). Então, o algoritmo 10 executa o método *getSequenciaAproximada* (linha 12) passando como parâmetro, além do conjunto \mathcal{L}^{uaf} e função f , um conjunto unitário constituído pela ação a_p para obter uma sequência aproximada O cujas normas proíbem unicamente a execução de a_p . Então, o algoritmo 10 obtém, através do método *getIndice* (linha 13), o índice *first* do primeiro conjunto de generalidade da sequência O a ser visitado na busca por normas quase eficientes. Por fim, através do método *visitarComTaxas* (linha 14), busca-se uma norma quase eficiente n que satisfaça os limiares desejados. A norma encontra é então retornada (linha 15). Se h não for uma história de conflito ou ela for visitada por completo e nenhuma norma quase eficiente que satisfaça os limiares seja encontrada, o algoritmo 10 retorna *null* (linha 17).

Exemplo 38 Considere novamente as histórias de sucesso h_2 e conflito h_4 do exem-

plo 17. A figura 31 ilustra a aplicação da síntese com backtracking do algoritmo 10 sobre a história h_4 , onde as arestas tracejadas em vermelho denotam a ordem de visita dos estados de h_4 e os estados e as arestas tracejadas em azul e lilás denotam um traço viabilizador.

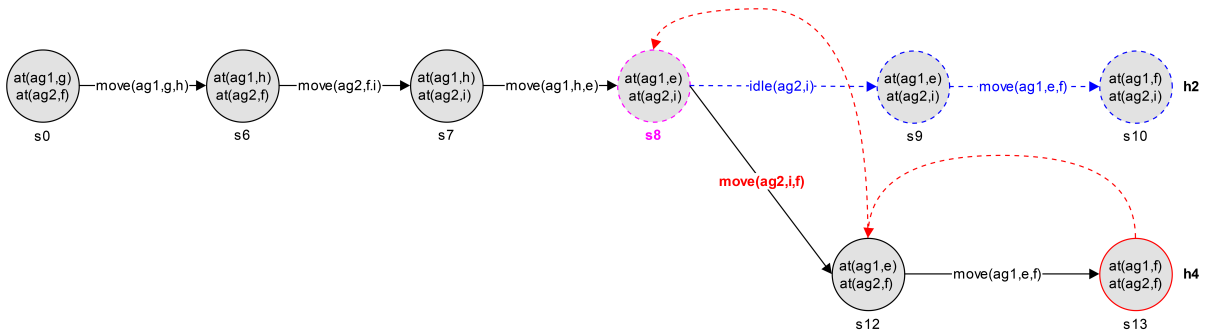


Figura 31 – Síntese com backtracking realizada pelo algoritmo 10 sobre a história h_4 .

Ao percorrer a história h_4 , o algoritmo 10 não identifica a partir do estado s_{12} um traço viabilizador através do qual os objetivos do sistema possam ser satisfeitos dentro do timeout especificado. Logo, o algoritmo 10 avança para o estado s_8 , a partir do qual, ele identifica o traço viabilizador $t_v = ((s_7, s_8, \text{idle}(ag_2, i)), (s_8, s_9, \text{move}(ag_1, e, f)), (s_9, s_{10}, \text{null}))$. Desta forma, tem-se que $s_v = s_8$ (destacado em lilás) e $a_p = \text{move}(ag_2, i, f)$ (destacada em vermelho). Ao aplicar o método `getAcao` sobre a_p , o algoritmo 10 obtém $a_p = \text{move}(Ag_1, L_1, L_2)$ e, ao executar o método `getSequenciaAproximada`, o algoritmo 10 obtém um espaço de normas menor do que o algoritmo 8, pois considera unicamente a ação $\text{move}(Ag_1, L_1, L_2)$ na criação das normas enquanto o algoritmo 8 considera, além de tal ação, a ação $\text{idle}(Ag_1, L_1)$. Assim como no exemplo 37, ao considerar uma taxa de sucesso de 90% e conflito de 0%, o algoritmo 10 pode sintetizar a norma eficiente $\{at(Ag_2, L_2)\} \xrightarrow{p} \text{move}(Ag_1, L_1, L_2)$.

Contudo, caso a história obtida na simulação do SMAn seja de sucesso ou uma história de conflito que não fomenta a síntese de normas que satisfaçam os limiares desejados, o algoritmo 10 retorna null. Considere que a simulação resulta na história de sucesso h_2 . Ao analisar o último estado de h_2 , o estado s_{10} , o algoritmo 10 verifica que s_{10} não é um estado de conflito e, portanto, retorna null. Dado que histórias de conflito podem não ocorrer, para aumentarmos a possibilidade de que a simulação do SMAn resulte em uma história de conflito e assim a síntese de normas possa ocorrer, pode-se realizar a simulação do SMAn com estados iniciais e/ou objetivos do sistema que induzam a situações indesejadas. Para o cenário do tráfego de carros podemos considerar uma grid parcialmente lotada. Como a grid tem dimensão 3×3 , em um estado inicial com 5 agentes, a grid tem a maior parte dos seus locais ocupada, o que torna provável que ocorra colisão entre os agentes. Contudo, como a grid possui locais livres, é provável que durante a análise da alcançabilidade, um traço viabilizador pelo

qual os agentes alcançam os locais de destino sem colidirem seja encontrado. Portanto, há necessidade de que o estado inicial e os objetivos do sistema sejam escolhidos de forma prudente para que uma história de conflito ocorra.

Considere agora as histórias de conflito h_8 e sucesso h_9 do exemplo 17. Considere também que a simulação do SMAn resulta na história h_8 . A figura 32 ilustra a aplicação da síntese com backtracking do algoritmo 10 sobre tal história, onde os estados e arestas tracejados em vermelho, azul e lilás possuem as mesmas conotações descritas anteriormente para as histórias h_2 e h_4 . Como não é possível encontrar um traço viabilizador a partir do estado s_{20} , o algoritmo 10 avança para o estado s_{19} a partir do qual encontra o traço viabilizador $t_v = ((s_{15}, s_{19}, \text{move}(\text{ag}_2, f, i)), (s_{19}, s_{23}, \text{move}(\text{ag}_1, e, f)), (s_{23}, s_{24}, \text{null}))$. Desta forma, tem-se que $s_v = s_{19}$ (destacado em lilás) e $a_p = \text{idle}(\text{ag}_2, f)$ (destacada em vermelho). Então, ao aplicar o método `getAcao` sobre a_p , o algoritmo 10 obtém $a_p = \text{idle}(\text{Ag}_1, L_1)$ e, posteriormente, cria a norma quase eficiente $\varphi \xrightarrow{p} \text{idle}(\text{Ag}_1, L_1)$, com $\varphi \subseteq s_{19}$. Contudo, ao proibir os agentes de ficarem parados, esta norma faz com que os agentes estejam sempre se movimentando, o que dificulta a satisfação dos objetivos do sistema e torna provável a ocorrência de colisões. Portanto, esta norma possui, possivelmente, taxas de conflito e/ou timeout altas e uma taxa de satisfação pequena ou igual a zero. Logo, é improvável que tal norma seja eficiente e traga algum benefício ao sistema.

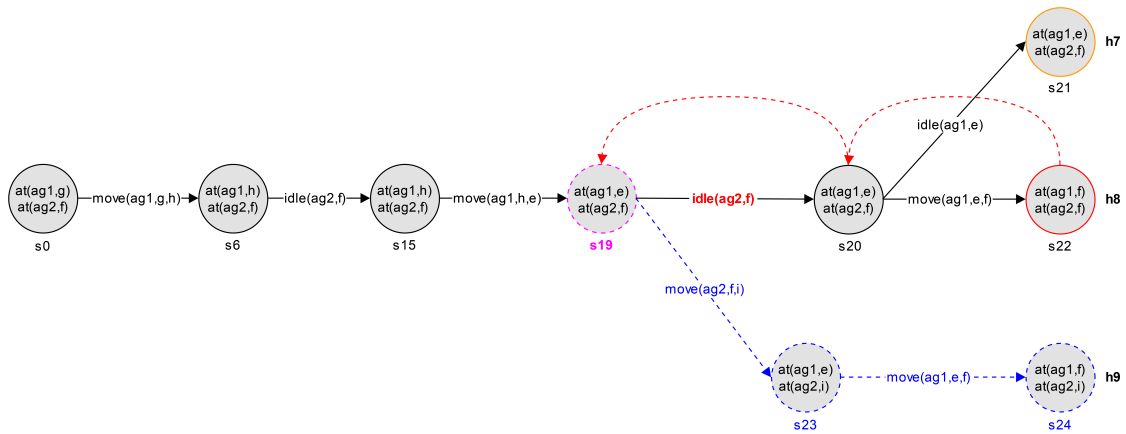


Figura 32 – Síntese com backtracking realizada pelo algoritmo 10 sobre a história h_8 .

Quando o número de ações do conjunto \mathcal{A} é maior que 1, a síntese com backtracking viabiliza ao algoritmo 10 construir espaços de normas menores do que o algoritmo 8, pois em vez de considerar todas as ações de \mathcal{A} na construção de tais espaços, conforme faz o algoritmo 8, o algoritmo 10 considera apenas uma das ações de \mathcal{A} , a ação que identifica como proibitiva. Todavia, o algoritmo 10 considera utilizar todas as fórmulas atômicas estritamente unground da linguagem \mathcal{L} , o conjunto \mathcal{L}^{uaf} , para a construção dos espaços de normas. A fim de obtermos espaços de normas

ainda menores, um aprimoramento a ser realizado sobre o algoritmo 10 é *considerar o estado viabilizar identificado na análise da alcançabilidade dos objetivos do sistema na construção dos espaços das normas*.

Como os estados do sistema variam de tamanho⁸, é possível que ao generalizarmos as fórmulas atômicas de um estado viabilizador, obtenhamos um número de fórmulas atômicas estritamente unground menor do que $|\mathcal{L}^{Uaf}|$. Neste sentido, podemos considerar um subconjunto próprio⁹ de \mathcal{L}^{Uaf} na construção do espaço das normas que, por sua vez, tende a ser menor do que o espaço obtido pelo algoritmo 10. Além do mais, como a história da simulação do *SMA*n pode ser grande, o processo de backtracking pode ser custoso, pois pode requerer realizar a análise da alcançabilidade para todos os estados da história. Assim, um segundo aprimoramento que podemos realizar sobre o algoritmo 10 é *realizar a análise da alcançabilidade somente sobre os estados onde há possibilidade de executar duas ou mais ações*. Desta forma, pode-se evitar que a análise da alcançabilidade ocorra a partir de estados onde não é possível expandir novos traços (i.e., estados onde somente uma ação é executável). Assim, o método *analisar* pode ser invocado um número menor de vezes, o que pode gerar uma economia de tempo na busca pelas normas quase eficientes.

Os aprimoramentos destacados para o algoritmo 10 estão contemplados pelo algoritmo 11 que realiza uma busca por normas eficientes semelhante ao algoritmo 10, contudo, a faz em espaços de normas que tendem a ser menores aplicando uma análise da alcançabilidade aprimorada. Para isso, ele recebe como entrada os mesmos parâmetros do algoritmo 10.

O algoritmo 11 apresenta duas diferenças de execução em relação ao algoritmo 10. Primeiramente, durante o backtracking sobre uma história h , o algoritmo 11 determina, através do método *getNumeroAcoesExecutaveis* (linha 8), o número de ações executáveis no estado corrente s_v . Se o número obtido for maior ou igual a 2, a análise da alcançabilidade dos objetivos do sistema é realizada. Em segundo lugar, através do método *getFormulas* (linha 13), o algoritmo 11 determina as fórmulas estritamente unground a serem consideradas na construção do espaço das normas ao generalizar as fórmulas atômicas ground do estado viabilizador s_v . O estado viabilizador, por sua vez, recebe as fórmulas generalizadas e é passado como argumento ao método *getSequenciaAproximada* (linha 14). Desta forma, o algoritmo 11 constrói espaços cujas normas combinam as fórmulas atômicas estritamente unground obtidas de s_v com a ação proibitiva a_p , isto é, $2^{S_v} \times a_p$. Além do mais, assim como ocorre com o algoritmo 10, o algoritmo 11 também pode obter histórias que não fomentam a síntese de normas eficientes. Portanto, a escolha prudente do estado inicial e dos

⁸ Conforme a definição 2, tem-se que $S \subseteq 2^{gaf}$. Logo, o número de fórmulas atômicas que forma um estado pode ser distinto de outro.

⁹ Um subconjunto próprio possui, pelo menos, um elemento a menos do que o conjunto do qual é subconjunto.

Algoritmo 11: Síntese com backtracking - Aprimoramento 6

```

Data:  $v, f, \mathcal{L}^{gaf}, \mathcal{A}, \mathcal{I}^{\mathcal{A}}, S_0, \mathcal{G}^*, c, tsuc, tconf, SMAn, \Delta, s_0, \mathcal{G}$ 
1  $h \leftarrow SMAn.executar(s_0, \mathcal{G}, null, c, \Delta)$ 
2  $s \leftarrow s.getEstadoFinal()$ 
3 if  $c \triangleright s$  then
4    $m \leftarrow |h| - 1$ 
5   while  $m \geq 0$  do
6      $s_v \leftarrow h.getTupla(m).getEstadoPai()$ 
7     if  $s_v \neq null$  then
8       if  $getNumeroAcoesExecutaveis(s_v) \geq 2$  then
9          $t_v \leftarrow analisar(s_v, \mathcal{G}, \mathcal{I}^{\mathcal{A}}, c, \Delta - m - 1)$ 
10        if  $t_v \neq []$  then
11           $a_p \leftarrow h.getTupla(m - 1).getAcaoProibitiva()$ 
12           $a_p \leftarrow getAcao(a_p)$ 
13           $s_v \leftarrow getFormulas(a_p)$ 
14           $O \leftarrow getSequenciaAproximada(s_v, \{a_p\}, f)$ 
15           $first \leftarrow getIndex(O)$ 
16           $n \leftarrow visitarComTaxas(v, O, first, S_0, \mathcal{G}^*, tsuc, tconf)$ 
17          return  $n$ 
18         $m \leftarrow m - 1$ 
19 return  $null$ 

```

objetivos do sistema é necessária para que a simulação do *SMAn* resulte em história de conflito que contribua para a ocorrência da síntese de normas.

Exemplo 39 Considere novamente as histórias de sucesso h_2 e conflito h_4 ilustradas pela figura 31 do exemplo 38. Ao percorrer a história h_4 , o algoritmo 11 não realiza a análise da alcançabilidade dos objetivos do sistema a partir do estado s_{12} , pois verifica que o número de ações executáveis a partir de s_{12} é 1. Por outro lado, ao avançar para o estado s_8 , o algoritmo 11 verifica que o número de ações executáveis a partir de tal estado é 2. Portanto, o algoritmo 11 economiza tempo e esforço ao realizar a análise da alcançabilidade dos objetivos do sistema somente a partir do estado s_8 . Assim como ocorre no exemplo anterior, o algoritmo 11 obtém o traço viabilizador $t_v = ((s_7, s_8, idle(ag_2, i)), (s_8, s_9, move(ag_1, e, f)), (s_9, s_{10}, null))$, o estado viabilizador $s_v = s_8$ e a ação proibitiva $a_p = move(ag_2, i, f)$. Através do método *getAcao*, ao generalizar a ação proibitiva, o algoritmo 11 obtém $a_p = move(Ag_1, L_1, L_2)$. Dado que $s_8 = \{at(ag_1, e), at(ag_2, i), next(a, b), \dots, next(h, i)\}^{10}$, através do método *getFormulas*, ao generalizar as fórmulas atômicas de s_v , o algoritmo 11 obtém $s_v = \{at(Ag_1, L_1), at(Ag_2, L_2), next(L_1, L_2)\}^{11}$. Considerando $\mathcal{L}^{uaf} = \{at(Ag_1, L_1), at(Ag_1, L_2), at(Ag_2, L_2), next(L_1, L_2)\}$,

¹⁰ Assim como nos capítulos anteriores, algumas fórmulas atômicas que denotam locais adjacentes na grid estão omitidas na descrição dos estados por questões de simplificação.

¹¹ O algoritmo 11 pode generalizar $at(ag_1, e)$ para $at(Ag_1, L_1)$, $at(ag_2, i)$ para $at(Ag_2, L_2)$ e as fórmulas

como $|s_V| < |\mathcal{L}^{uaf}|$, desconsiderando a exclusão das normas não-mínimas, o algoritmo 11 obtém um espaço de normas menor do que os algoritmos 8 e 10. Enquanto o espaço obtido pelo algoritmo 11 possui 8 normas, os espaços obtidos pelos algoritmos 8 e 10 possuem 32 e 16 normas, respectivamente.

A tabela 4 sumariza as vantagens, desvantagens e aprimoramentos apontados para os algoritmos 1, 2, 3, 6, 8, 10 e 11. Como os algoritmos 4, 5, 7 e 9 são utilizados como métodos pelos algoritmos anteriores, eles foram omitidos da tabela 4. Contudo, os seus pontos positivos e negativos são considerados pelos algoritmos que os utilizam. Ainda, a coluna Alg refere-se aos algoritmos, a sigla AAOS é uma abreviação para a análise da alcançabilidade dos objetivos do sistema e o símbolo “-” indica que não existem aprimoramentos sugeridos para o algoritmo 11.

Alg	Prós	Contras	Aprimoramentos
1	<p>i) sintetiza normas que evitam os estados de conflito do sistema.</p>	<p>i) requer analisar todos os estados do sistema; ii) pode sintetizar muitas normas específicas; iii) pode sintetizar normas sem utilidade para o sistema; iv) não garante a alcançabilidade dos objetivos do sistema.</p>	<p>i) construir um conjunto de normas combinando diretamente os elementos da linguagem \mathcal{L}; ii) considerar normas que satisfaçam a definição 27.</p>
2	<p>i) sintetiza normas eficientes; ii) não requer analisar os estados do sistema.</p>	<p>i) o custo da síntese pode ser maior do que o algoritmo 1; ii) pode sintetizar mais normas do que o algoritmo 1; iii) sintetiza normas de todas as generalidades; iv) analisa as normas sem uma ordenação prévia; v) requer construir o conjunto \mathcal{H}.</p>	<p>i) sintetizar normas estritamente unground; ii) ordenar as normas segundo a sua generalidade.</p>
3	<p>i) constrói conjuntos de normas menores do que o algoritmo 2; ii) constrói sequências semânticas; iii) viabiliza a aplicação de diferentes estratégias de visita.</p>	<p>i) requer analisar todos os estados do sistema para determinar a generalidade das normas; ii) requer realizar o produto cartesiano $\mathcal{N} \times \mathcal{N}$ para construir a relação de ordem parcial \mathcal{R}; iii) sintetiza normas replicadas (semanticamente equivalentes). iv) requer construir o conjunto \mathcal{H}.</p>	<p>i) determinar a generalidade das normas através de uma aproximação; ii) sintetizar unicamente normas mínimas.</p>
6	<p>i) constrói conjuntos de normas menores do que o algoritmo 3. ii) constrói sequências aproximadas.</p>	<p>i) requer ordenar as normas; ii) requer construir o conjunto \mathcal{H}.</p>	<p>i) realizar a busca por normas eficientes através das normas quase eficientes.</p>
8	<p>i) sintetiza normas eficientes através das normas quase eficientes; ii) a síntese é independente da construção do conjunto \mathcal{H}.</p>	<p>i) a norma encontrada pode não ser necessariamente eficiente; ii) o conjunto das normas pode ser constituído por diferentes normas não alinhadas a comportamentos que necessitam de regulação; iii) requer simular o SMAN múltiplas vezes.</p>	<p>i) sintetizar normas considerando contextos que levam os agentes a situações de conflito.</p>
10	<p>i) realiza a AAOS; ii) constrói conjuntos de normas melhor alinhadas aos comportamentos dos agentes que necessitam de regulação ao considerar uma ação proibitiva; iii) pode construir conjuntos de normas menores do que o algoritmo 8.</p>	<p>i) requer simular o SMAN e obter uma história de conflito; ii) pode realizar a AAOS para cada estado da história de conflito; iii) requer que um traço viabilizador seja encontrado; iv) requer que a história de conflito fomente a síntese de normas.</p>	<p>i) Considerar o estado viabilizar identificado na AAOS na construção dos espaços das normas; ii) realizar a AAOS somente sobre os estados onde há possibilidade de executar duas ou mais ações.</p>
11	<p>i) realiza a AAOS; ii) constrói conjuntos de normas melhor alinhadas aos comportamentos dos agentes que necessitam de regulação ao considerar uma ação proibitiva e as fórmulas atômicas do estado viabilizador; iii) pode realizar a AAOS menos vezes que o algoritmo 10; iv) pode construir conjuntos de normas menores do que o algoritmo 10.</p>	<p>i) requer simular o SMAN e obter uma história de conflito; ii) requer que um traço viabilizador seja encontrado; iii) requer que a história de conflito fomente a síntese de normas.</p>	<p>-</p>

Tabela 4 – Prós, Contras e Aprimoramentos dos Algoritmos de Síntese de Normas Eficientes.

5.2 EXPERIMENTOS E RESULTADOS

Conforme discutido anteriormente, a síntese de normas eficientes é um problema de solução não trivial que requer determinar a generalidade das normas e o conjunto de histórias do sistema. Visando obter uma forma factível de sintetizar normas eficientes, nós apresentamos na seção anterior um conjunto de algoritmos para a síntese de normas construídos incrementalmente através de um conjunto de aprimoramentos e apontamos que uma forma de sintetizar normas eficientes de forma possivelmente menos custosa é através da síntese de normas quase eficientes. Dentre os algoritmos apresentados, nós consideramos que somente os algoritmos 8, 10 e 11 são viáveis para tal fim, pois independem de determinar a generalidade das normas e o conjunto das histórias do sistema para sintetizar normas eficientes. Além do mais, tais algoritmos são capazes de construir espaços de normas menores, o que pode aumentar o desempenho da síntese de normas.

Nós estendemos o nosso simulador de SMAn para suportar os algoritmos 8, 10 e 11. Assim, o simulador roda um dado algoritmo de síntese e executa o modelo de SMAn. Para comparar o desempenho dos algoritmos 8, 10 e 11 na síntese de normas eficientes, nós realizamos experimentos sobre os cenários do tráfego de carros, segurança de acesso e atendimento médico das seções 4.2.1, 4.2.2 e 4.2.3, respectivamente, onde avaliamos: *i*) o impacto dos algoritmos no tamanho do espaço das normas (apresentado na seção 5.2.1); e *ii*) a média de normas analisadas para encontrar normas eficientes e a média do tempo de busca por tais normas (apresentadas na seção 5.2.2). Em particular, no experimento *ii*, de acordo com os limiares desejados (taxa de sucesso, t_{suc} , e taxa de conflito, t_{conf}), os algoritmos buscam no primeiro cenário pela norma $n = \{at(Ag_2, L_2)\} \xrightarrow{\rho} move(Ag_1, L_1, L_2)$, no segundo cenário pela norma $n' = \{lowCredential(Ag_1), confidencial(Doc_1)\} \xrightarrow{\rho} open(Ag_1, Doc_1)$ e no terceiro cenário pela norma $n'' = \{seated(Ag_2, C_2), next(C_2, C_1)\} \xrightarrow{\rho} sit(Ag_1, C_1)$.

Assim como ocorreu no capítulo anterior, os experimentos foram realizados em um computador com processador Intel Core i7-1165G7 de 16 GB de RAM, rodando a 2,8 GHz e com sistema operacional Ubuntu 22.10 de 64 bits. Além do mais, os algoritmos 10 e 11 fazem uso do planejador automático PDDL4J (PELLIER; FIORINO, 2018) para realizar a análise da alcançabilidade dos objetivos do sistema.

5.2.1 Tamanho do Espaço das Normas

A tabela 5 apresenta o tamanho do espaço das normas obtidos pelos algoritmos 8, 10 e 11 para os cenários do tráfego de carros, segurança de acesso e atendimento médico. Executamos tais algoritmos uma única vez, sem a simulação do SMAn, para que o espaço das normas fosse construído e, assim, verificarmos o seu tamanho. Para fins de comparação, a linha *sem algoritmo* apresenta o tamanho do espaço das normas

obtido conforme a definição 15, isto é, $|\{\varphi \xrightarrow[p]{\rightarrow} a \mid \varphi \in 2^{\mathcal{L}^{uaf}} \wedge a \in \mathcal{A}\}|$. Neste caso, o espaço é constituído por todas as normas estritamente unground da linguagem \mathcal{L} .

Algoritmo	Tráfego de Carros	Segurança de Acesso	Atendimento Médico
sem algoritmo	256	1280	14343
algoritmo 8	94	896	6144
algoritmo 10	12	128	256
algoritmo 11	2	32	16

Tabela 5 – Tamanho do espaço das normas.

Para todos os cenários, podemos observar que à medida que os algoritmos 8, 10 e 11 são aplicados, o tamanho do espaço das normas diminui. Podemos observar também que a redução do tamanho do espaço em termos de valores absolutos para o cenário do tráfego de carros é significativamente menor do que nos demais cenários. Todavia, ao analisarmos a redução em termos percentuais, podemos verificar que a redução é, na verdade, um valor consideravelmente alto. A tabela 6 apresenta, para cada cenário, o percentual de redução entre o tamanho do espaço das normas obtido *sem algoritmo* e o tamanho do espaço obtido pelos algoritmos 8, 10 e 11.

Algoritmo	Tráfego de Carros	Segurança de Acesso	Atendimento Médico
algoritmo 8	63,28%	30,00%	57,16%
algoritmo 10	95,31%	90,00%	98,21%
algoritmo 11	99,21%	97,50%	99,88%

Tabela 6 – Percentual de redução do tamanho do espaço das normas.

Para todos os cenários, o algoritmo 8 obtém espaços de normas que são, pelo menos, 30% menores do que os espaços obtidos *sem algoritmo* (considerando o cenário da segurança de acesso como base para as comparações), enquanto que os algoritmos 10 e 11 obtêm espaços de normas que são, respectivamente, 90,00% e 97,50%, menores (considerando o cenário do tráfego de carros como base). Além do mais, observa-se que o percentual de redução do algoritmo 11 é maior do que os percentuais de redução dos demais algoritmos, onde obtém uma redução de quase 100% nos cenários do tráfego de carros e atendimento médico.

5.2.2 Médias de Normas Analisadas e Tempos de Busca

Considere que uma *configuração* é um par ordenado (f, v) formado por uma função de generalidade $f \in \{\text{NGK, NGF, NGI, NGJ}\}$ (definições 46 a 49) e uma estratégia de visita $v \in \{\text{Gmin Gmax, Gcen Gmax, Zigzag, Aleatória}\}$ (introduzidas na seção 5.1). Para cada configuração, os algoritmos 8, 10 e 11 são executados 20 vezes e, a partir dos resultados obtidos, nós calculamos a média e o desvio padrão do *número de normas analisadas* na busca por normas quase eficientes e a média e o desvio padrão

do tempo de busca por tais normas. Como existem 16 configurações, cada algoritmo é executado 320 vezes por cenário. Logo, em um cenário, os algoritmos são executados 960 vezes em conjunto, o que contabiliza 2880 execuções nos três cenários. Além do mais, para cada execução de um algoritmo, os conjuntos de generalidade das sequências aproximadas são embaralhados para captar o impacto da ordem das normas nos conjuntos na busca pelas normas eficientes.

Os resultados de cada algoritmo estão agrupados em duas tabelas, uma para as médias de normas analisadas e outra para as médias de tempo de busca (dada em minutos), onde as linhas e colunas referem-se às estratégias de visita e às funções de generalidade, respectivamente. Por sua vez, os resultados das configurações são descritos no formato *média ± desvio padrão*. Consideramos que as melhores configurações para um algoritmo em um cenário para obter o menor número de normas analisadas e/ou tempo de busca são aquelas cujas *somas das médias e desvios-padrão* são as *menores somas* das suas respectivas tabelas. Tais configurações são destacadas em negrito azul. Entretanto, é possível que diferentes configurações sejam aproximadamente iguais as melhores configurações. Isto ocorre quando as médias de tais configurações fazem parte do intervalo [média - desvio padrão, média + desvio padrão] das melhores configurações. Tais configurações são destacadas em negrito laranja. Portanto, consideramos que as configurações destacadas em negrito azul e negrito laranja em uma tabela são *igualmente boas* para o respectivo algoritmo.

5.2.2.1 Tráfego de Carros

As tabelas 7, 8 e 9 apresentam, respectivamente, as médias e os desvios-padrão do número de normas analisadas pelos algoritmos 8, 10 e 11 na busca pela norma eficiente $n = \{at(Ag_2, L_2)\} \xrightarrow{p} move(Ag_1, L_1, L_2)$. Considerando os resultados dos experimentos da seção 4.2.1, nós definimos $t_{suc} = 90$ e $t_{conf} = 0$.

Estratégia	NGK	NGF	NGI	NGJ
Gmin Gmax	60,60 ± 5,50	70,65 ± 5,49	80,45 ± 11,55	25,00 ± 14,81
Aleatória	35,40 ± 14,63	35,41 ± 14,58	35,71 ± 15,33	34,00 ± 18,01
Zigzag	12,01 ± 5,52	28,00 ± 5,50	30,60 ± 10,51	70,61 ± 12,43
Gcen Gmax	10,45 ± 5,50	27,40 ± 5,51	30,50 ± 10,51	70,55 ± 12,41

Tabela 7 – Médias de normas analisadas pelo algoritmo 8 no cenário do tráfego de carros.

Para todas as configurações, a média de normas analisadas diminui, respectivamente, do algoritmo 8 para os algoritmos 10 e 11. Conforme apresentam as tabelas 7 e 8, para cada função, exceto NGJ, a estratégia Gmin Gmax possui as maiores médias de normas analisadas seguida, respectivamente, pelas estratégias Aleatória, Zigzag e Gcen Gmax. Como o par (NGK, Gcen Gmax) possui a menor soma da média com o

Estratégia	NGK	NGF	NGI	NGJ
Gmin Gmax	7,50 ± 0,51	8,45 ± 0,50	9,40 ± 0,53	6,01 ± 1,00
Aleatória	7,41 ± 2,51	7,55 ± 2,42	7,59 ± 2,43	7,49 ± 2,62
Zigzag	3,60 ± 0,48	5,50 ± 0,52	6,51 ± 0,49	7,00 ± 1,00
Gcen Gmax	3,45 ± 0,51	5,40 ± 0,50	6,40 ± 0,50	7,01 ± 1,00

Tabela 8 – Médias de normas analisadas pelo algoritmo 10 no cenário do tráfego de carros.

Estratégia	NGK	NGF	NGI	NGJ
Gmin Gmax	1,00 ± 0	1,00 ± 0	1,00 ± 0	1,00 ± 0
Aleatória	1,50 ± 0,50	1,45 ± 0,50	1,45 ± 0,50	1,50 ± 0,49
Zigzag	2,00 ± 0	2,00 ± 0	2,00 ± 0	2,00 ± 0
Gcen Gmax	2,00 ± 0	2,00 ± 0	2,00 ± 0	2,00 ± 0

Tabela 9 – Médias de normas analisadas pelo algoritmo 11 no cenário do tráfego de carros.

desvio padrão, ele é considerado a melhor configuração para os algoritmos 8 e 10 analisarem o menor número de normas. Além do mais, como as médias da configuração (NGK,Zigzag) fazem parte dos intervalos formados pelas médias e desvios-padrão da configuração (NGK,Gcen Gmax), o par (NGK,Zigzag) é considerado também uma configuração igualmente boa para os algoritmos 8 e 10. Por outro lado, como o algoritmo 11 analisa um espaço muito pequeno constituído por apenas 2 normas, conforme apresenta a tabela 9, diferentes configurações possuem as mesmas médias e os mesmos desvios-padrão. Desta forma, diferentes configurações apresentam média e desvio padrão mínimos (i.e., 1 e 0, respectivamente). Portanto, o algoritmo 11 possui mais de uma melhor configuração para analisar o menor número de normas, a saber, os pares (NGK, Gmin Gmax), (NGF, Gmin Gmax), (NGI, Gmin Gmax) e (NGJ, Gmin Gmax). Diferentemente dos resultados das tabelas anteriores, as estratégias Zigzag e Gcen Gmax, que possuem resultados iguais, possuem as maiores médias de normas analisadas seguidas pelas estratégias Aleatória e Gmin Gmax.

Por sua vez, as tabelas 10, 11 e 12 apresentam as médias e os desvios-padrão do tempo de busca dos algoritmos 8, 10 e 11, respectivamente, pela norma n .

Estratégia	NGK	NGF	NGI	NGJ
Gmin Gmax	5,12 ± 0,51	5,98 ± 0,51	6,64 ± 1,08	2,27 ± 1,44
Aleatória	2,59 ± 1,49	2,59 ± 1,49	2,59 ± 1,55	2,55 ± 1,65
Zigzag	1,02 ± 0,51	2,37 ± 0,51	2,55 ± 0,99	5,99 ± 1,14
Gcen Gmax	0,88 ± 0,51	2,32 ± 0,51	2,52 ± 0,99	5,97 ± 1,14

Tabela 10 – Médias do tempo de busca do algoritmo 8 no cenário do tráfego de carros.

Estratégia	NGK	NGF	NGI	NGJ
Gmin Gmax	0,63 ± 0,14	0,70 ± 0,14	0,77 ± 0,14	0,48 ± 0,21
Aleatória	0,62 ± 0,51	0,63 ± 0,49	0,62 ± 0,49	0,62 ± 0,55
Zigzag	0,31 ± 0,14	0,45 ± 0,14	0,53 ± 0,13	0,58 ± 0,21
Gcen Gmax	0,28 ± 0,14	0,44 ± 0,14	0,52 ± 0,14	0,58 ± 0,21

Tabela 11 – Médias do tempo de busca do algoritmo 10 no cenário do tráfego de carros.

Estratégia	NGK	NGF	NGI	NGJ
Gmin Gmax	0,10 ± 0,02	0,10 ± 0,02	0,10 ± 0,02	0,10 ± 0,02
Aleatória	0,15 ± 0,14	0,15 ± 0,14	0,15 ± 0,14	0,15 ± 0,14
Zigzag	0,20 ± 0,02	0,20 ± 0,02	0,20 ± 0,02	0,20 ± 0,02
Gcen Gmax	0,20 ± 0,02	0,20 ± 0,02	0,20 ± 0,02	0,20 ± 0,02

Tabela 12 – Médias do tempo de busca do algoritmo 11 no cenário do tráfego de carros.

Conforme apresentam tais tabelas, para cada configuração, a média do tempo de busca diminui do algoritmo 8 para os algoritmos 10 e 11, respectivamente. Mais especificamente, para cada função, exceto NGJ, as médias do tempo de busca dos algoritmos 8 e 10 decrescem ao executarem com estratégia Gmin Gmax para as estratégias Aleatória, Zigzag e Gcen Gmax. De acordo com as tabelas 10 e 11, o par (NGK, Gcen Gmax) é a melhor configuração para os algoritmos 8 e 10 obterem o menor tempo de busca enquanto que o par (NGK, Zigzag) é uma configuração igualmente boa para tal fim. Conforme apresenta a tabela 12, o algoritmo 11 obtém as menores médias e desvios-padrão de tempo de busca ao executar com as configurações (NGK, Gmin Gmax), (NGF, Gmin Gmax), (NGI, Gmin Gmax) e (NGJ, Gmin Gmax). Contudo, para todas as funções, as médias do tempo de busca do algoritmo 11 crescem quando ele executa, respectivamente, com a estratégia Aleatória e com as estratégias Zigzag e Gcen Gmax (estas duas últimas estratégias apresentam as mesmas médias). Observa-se ainda nas tabelas 7, 8, 10 e 11 que, para cada estratégia, o resultado com o maior desvio padrão ocorre com a função NGJ enquanto que, para cada função, o resultado com o maior desvio padrão ocorre com a estratégia Aleatória. Embora o algoritmo 11 obtenha também, para cada função, o maior desvio padrão ao executar com a estratégia Aleatória, não é possível observar nas tabelas 7 e 10 que, para cada estratégia, o maior desvio padrão ocorre com uma única função.

5.2.2.2 Segurança de Acesso

As tabelas 13, 14 e 15 apresentam, respectivamente, as médias e os desvios-padrão de normas analisadas pelos algoritmos 8, 10 e 11 na busca pela norma efi-

ciente $n' = \{lowCredential(Ag_1), confidencial(Doc_1)\} \xrightarrow{\rho} open(Ag_1, Doc_1)$. Dados os resultados dos experimentos da seção 4.2.2, nós definimos $t_{suc} = 100$ e $t_{conf} = 0$.

Estratégia	NGK	NGF	NGI	NGJ
Gmin Gmax	723,40 ± 19,05	685,55 ± 17,22	757,25 ± 41,34	758,55 ± 51,92
Aleatória	528,35 ± 112,89	523,00 ± 117,16	481,55 ± 198,41	432,20 ± 259,78
Zigzag	683,50 ± 22,75	642,00 ± 17,04	670,85 ± 38,50	600,17 ± 65,84
Gcen Gmax	428,50 ± 21,60	405,25 ± 17,32	456,55 ± 45,15	597,85 ± 63,16

Tabela 13 – Médias de normas analisadas pelo algoritmo 8 no cenário da segurança de acesso.

Estratégia	NGK	NGF	NGI	NGJ
Gmin Gmax	82,70 ± 6,05	82,40 ± 6,19	93,55 ± 12,44	101,35 ± 15,61
Aleatória	65,80 ± 13,07	66,85 ± 13,65	58,55 ± 20,41	47,55 ± 35,50
Zigzag	41,01 ± 6,10	40,50 ± 6,05	50,65 ± 9,42	71,10 ± 15,88
Gcen Gmax	40,60 ± 6,11	40,55 ± 5,97	49,70 ± 10,33	69,65 ± 16,42

Tabela 14 – Médias de normas analisadas pelo algoritmo 10 no cenário da segurança de acesso.

Estratégia	NGK	NGF	NGI	NGJ
Gmin Gmax	14,50 ± 3,10	14,55 ± 3,12	15,01 ± 2,99	20,50 ± 5,51
Aleatória	13,95 ± 8,27	14,20 ± 8,02	14,90 ± 7,97	9,50 ± 14,43
Zigzag	4,20 ± 2,97	4,15 ± 2,90	4,20 ± 2,95	10,45 ± 6,15
Gcen Gmax	4,15 ± 2,91	4,15 ± 2,85	4,19 ± 2,94	10,50 ± 6,05

Tabela 15 – Médias de normas analisadas pelo algoritmo 11 no cenário da segurança de acesso.

De forma análoga à seção anterior, para todas as configurações, a média de normas analisadas decresce do algoritmo 8 para os algoritmos 10 e 11, respectivamente. Conforme apresenta a tabela 13, para cada função, exceto NGJ, a estratégia Gmin Gmax possui a maior média de normas analisadas seguida pelas estratégias Zigzag, Aleatória e Gcen Gmax. Por outro lado, de acordo com a tabela 14, embora a estratégia Gmin Gmax apresente a maior média de normas analisadas, a sua média é seguida pela estratégia Aleatória e, então, pelas estratégias Zigzag e Gcen Gmax que, por sua vez, apresentam resultados aproximadamente iguais. De acordo com tais tabelas, o par (NGF,Gcen Gmax) é a melhor configuração para os algoritmos 8 e 10 analisarem o menor número de normas. Contudo, de acordo com a tabela 14, os pares (NGK,Zigzag), (NGK,Gcen Gmax) e (NGF,Zigzag) são também configurações igualmente boas para o algoritmo 10 para tal fim dado que apresentam resultados próximos aos resultados da melhor configuração.

Conforme apresenta a tabela 15, para cada função, exceto NGJ, as estratégias Gmin Gmax e Aleatória apresentam as maiores médias de normas analisadas, sendo elas valores aproximadamente iguais. Por outro lado, assim como ocorre com a tabela 14, as estratégias Gcen Gmax e Zigzag apresentam as menores médias, sendo elas valores também aproximadamente iguais. Por sua vez, o par (NGF,Gcen Gmax) é a melhor configuração para o algoritmo 11 analisar o menor número de normas enquanto os pares (NGK,Zigzag), (NGK,Gcen Gmax), (NGF,Zigzag), (NGI,Zigzag) e (NGI,Gcen Gmax) são configurações igualmente boas para o algoritmo 11 para tal fim.

As tabelas 16, 17 e 18 apresentam as médias do tempo de busca pela norma n' dos algoritmos 8, 10 e 11, respectivamente.

Estratégia	NGK	NGF	NGI	NGJ
Gmin Gmax	21,70 ± 0,57	20,56 ± 0,51	22,71 ± 1,24	22,75 ± 1,55
Aleatória	15,85 ± 3,24	15,69 ± 3,38	14,44 ± 5,95	12,96 ± 7,79
Zigzag	20,50 ± 0,68	19,26 ± 0,51	20,12 ± 1,15	18,00 ± 1,97
Gcen Gmax	12,85 ± 0,64	12,15 ± 0,51	13,69 ± 1,35	17,93 ± 1,89

Tabela 16 – Médias do tempo de busca do algoritmo 8 no cenário da segurança de acesso.

Estratégia	NGK	NGF	NGI	NGJ
Gmin Gmax	2,41 ± 0,18	2,47 ± 0,18	2,80 ± 0,37	3,04 ± 0,46
Aleatória	1,97 ± 0,39	2,00 ± 0,40	1,75 ± 0,61	1,42 ± 1,06
Zigzag	1,23 ± 0,18	1,21 ± 0,18	1,51 ± 0,28	2,13 ± 0,47
Gcen Gmax	1,21 ± 0,18	1,21 ± 0,17	1,49 ± 0,30	2,08 ± 0,49

Tabela 17 – Médias do tempo de busca do algoritmo 10 no cenário da segurança de acesso.

Estratégia	NGK	NGF	NGI	NGJ
Gmin Gmax	0,43 ± 0,09	0,43 ± 0,09	0,45 ± 0,08	0,61 ± 0,16
Aleatória	0,41 ± 0,24	0,42 ± 0,24	0,41 ± 0,23	0,28 ± 0,43
Zigzag	0,12 ± 0,08	0,12 ± 0,08	0,12 ± 0,08	0,31 ± 0,18
Gcen Gmax	0,12 ± 0,08	0,12 ± 0,08	0,12 ± 0,08	0,31 ± 0,18

Tabela 18 – Médias do tempo de busca do algoritmo 11 no cenário da segurança de acesso.

Como os algoritmos 8, 10 e 11 obtêm espaços de normas maiores neste cenário, as médias de tempo de busca que eles apresentam correntemente são maiores do que no cenário do tráfego de carros. Assim como no cenário anterior, para todas as configurações, a média do tempo de busca decresce do algoritmo 8 para os algoritmos 10 e

11, respectivamente. Para toda função, exceto NGJ, de acordo com a tabela 16, Gmin Gmax é a estratégia que demanda em média mais tempo de busca para encontrar a norma desejada seguida pelas estratégias Zigzag, Aleatória e Gcen Gmax. Por outro lado, a estratégia Gmin Gmax apresenta, de acordo com a tabela 17, a maior média de tempo de busca seguida pela estratégia Aleatória e, então, pelas estratégias Zigzag e Gcen Gmax que, por sua vez, apresentam médias de tempo aproximadamente iguais. De acordo com tais tabelas, o par (NGF,Gcen Gmax) é a melhor configuração para os algoritmos 8 e 10 encontrarem a norma eficiente com o menor tempo de busca e de acordo com a tabela 17, os pares (NGK,Zigzag), (NGK,Gcen Gmax) e (NGF,Zigzag) são também configurações igualmente boas para o algoritmo 10 para tal fim.

Conforme apresenta a tabela 18, para cada função, exceto NGJ, as estratégias Gmin Gmax e Aleatória apresentam médias de tempo de busca aproximadamente iguais sendo elas as maiores médias de tempo obtidas pelo algoritmo 11. Por outro lado, as estratégias Gcen Gmax e Zigzag possuem as menores médias de tempo de busca, onde os pares (NGK,Zigzag), (NGK,Gcen Gmax), (NGF,Zigzag), (NGF,Gcen Gmax), (NGI,Zigzag) e (NGI,Gcen Gmax) possuem a mesma média e o mesmo desvio padrão. Como a diferença entre as médias de normas analisadas por tais configurações é muito pequena, não há diferença nos resultados do tempo de busca quando o algoritmo 11 executa com tais configurações (considerando dois dígitos de precisão). Logo, assim como ocorre no cenário anterior, o algoritmo 11 possui mais de uma melhor configuração para obter o menor tempo de busca.

Para cada função, exceto NGJ, ao compararmos as estratégias Gcen Gmax e Gmin Gmax podemos observar que Gcen Gmax possui, conforme a tabela 16, médias de tempo, no mínimo, 39,71% menores do que as médias da estratégia Gmin Gmax e, conforme a tabela 17, Gcen Gmax possui, no mínimo, médias de tempo 46,78% menores. Por outro lado, ao compararmos as estratégias Gcen Gmax e Zigzag com a estratégia Gmin Gmax podemos observar que as duas primeiras estratégias possuem, conforme a tabela 18, médias de tempo, no mínimo, 72,09% menores do que as médias da terceira estratégia. Além do mais, de acordo com todas as tabelas deste cenário, para todas as estratégias, o maior desvio padrão ocorre com a função NGJ enquanto que, para cada função, o maior desvio padrão ocorre com a estratégia Aleatória.

5.2.2.3 Atendimento Médico

As tabelas 19, 20 e 21 apresentam, respectivamente, as médias e os desvios-padrão de normas analisadas pelos algoritmos 8, 10 e 11 na busca pela norma quase eficiente $n'' = \{seated(Ag_2, C_2), next(C_2, C_1)\} \xrightarrow{p} sit(Ag_1, C_1)$. Considerando os resultados dos experimentos da seção 4.2.3, nós definimos $t_{suc} = 100$ e $t_{conf} = 0$.

Assim como ocorre nos dois cenários anteriores, para todas as configurações, a média de normas analisadas diminui, respectivamente, do algoritmo 8 para os algo-

Estratégia	NGK	NGF	NGI	NGJ
Gmin Gmax	4592,40 ± 81,52	5142,40 ± 122,87	5638,10 ± 181,55	2436,10 ± 321,01
Aleatória	3125,50 ± 1924,72	3234,10 ± 2201,74	3458,80 ± 2589,56	2506,95 ± 2996,10
ZigZag	3802,70 ± 105,46	4442,55 ± 150,87	5038,10 ± 192,55	4956,45 ± 1041,10
Gcen Gmax	2289,90 ± 80,87	2840,85 ± 120,33	2801,15 ± 180,98	4979,05 ± 1018,30

Tabela 19 – Médias de normas analisadas pelo algoritmo 8 no cenário do atendimento médico.

Estratégia	NGK	NGF	NGI	NGJ
Gmin Gmax	151,50 ± 8,58	173,50 ± 12,92	208,05 ± 19,00	42,05 ± 33,01
Aleatória	80,15 ± 41,74	106,10 ± 45,39	141,95 ± 46,91	153,05 ± 88,63
ZigZag	120,50 ± 15,98	145,80 ± 20,21	175,15 ± 25,36	180,50 ± 35,60
Gcen Gmax	64,05 ± 8,55	78,50 ± 12,51	96,45 ± 18,15	191,15 ± 35,64

Tabela 20 – Médias de normas analisadas pelo algoritmo 10 no cenário do atendimento médico.

Estratégia	NGK	NGF	NGI	NGJ
Gmin Gmax	7,50 ± 0,52	7,51 ± 0,53	10,51 ± 1,53	3,98 ± 2,13
Aleatória	5,55 ± 3,51	5,49 ± 3,40	4,27 ± 6,67	4,35 ± 7,61
ZigZag	3,55 ± 0,50	3,50 ± 0,52	4,55 ± 1,47	11,25 ± 2,71
Gcen Gmax	3,50 ± 0,50	3,51 ± 0,50	4,50 ± 1,52	11,50 ± 2,47

Tabela 21 – Médias de normas analisadas pelo algoritmo 11 no cenário do atendimento médico.

ritmos 10 e 11. Conforme apresentam as tabelas 19 e 20, para cada função, exceto NGJ, a média de normas analisadas pelos algoritmos 8 e 10 é maior ao executarem com a estratégia Gmin Gmax decrescendo, respectivamente, ao executarem com as estratégias Zigzag, Aleatória e Gcen Gmax. Além do mais, de acordo com tais tabelas, o par (NGK,Gcen Gmax) é a melhor configuração para tais algoritmos analisarem o menor número de normas. Por outro lado, conforme apresenta a tabela 21, para cada função, exceto NGJ, a média de normas analisadas pelo algoritmo 11 decresce da estratégia Gmin Gmax para as estratégias Aleatória, Zigzag e Gcen Gmax. Assim como ocorre com os demais algoritmos, o par (NGK,Gcen Gmax) é também a melhor configuração para o algoritmo 11 analisar o menor número de normas. Todavia, o algoritmo 11 possui outras quatro configurações igualmente boas, a saber (NGK,Zigzag), (NGF,Zigzag), (NGF,Gcen Gmax) e (NGJ,Gmin Gmax).

Por sua vez, as tabelas 22, 23 e 24 apresentam as médias do tempo de busca dos algoritmos 8, 10 e 11 pela norma n'' , respectivamente.

Como nos experimentos deste cenário são criados os maiores espaços de busca, as médias do tempo de busca das tabelas 22, 23 e 24 alcançam os maiores valores dentre os três cenários. De acordo com as tabelas 22 e 23, para cada função, exceto NGJ, a média do tempo de busca dos algoritmos 8 e 10 é maior ao executa-

Estratégia	NGK	NGF	NGI	NGJ
Gmin Gmax	183,69 ± 5,26	205,69 ± 4,91	225,52 ± 7,26	96,44 ± 12,84
Aleatória	125,02 ± 76,98	129,36 ± 88,06	142,35 ± 111,58	100,27 ± 119,84
Zigzag	152,10 ± 4,21	177,70 ± 6,03	201,52 ± 7,70	198,25 ± 41,64
Gcen Gmax	91,59 ± 3,23	113,63 ± 4,81	138,46 ± 3,11	199,16 ± 40,73

Tabela 22 – Médias do tempo de busca do algoritmo 8 no cenário do atendimento médico.

Estratégia	NGK	NGF	NGI	NGJ
Gmin Gmax	6,06 ± 0,34	6,94 ± 0,51	8,32 ± 0,76	1,68 ± 1,80
Aleatória	3,20 ± 1,66	4,24 ± 1,81	5,67 ± 1,87	6,12 ± 3,54
Zigzag	4,82 ± 0,63	5,83 ± 0,80	7,00 ± 1,01	7,22 ± 1,42
Gcen Gmax	2,56 ± 0,34	3,14 ± 0,50	3,85 ± 0,73	7,64 ± 1,42

Tabela 23 – Médias do tempo de busca do algoritmo 10 no cenário do atendimento médico.

Estratégia	NGK	NGF	NGI	NGJ
Gmin Gmax	0,30 ± 0,02	0,30 ± 0,02	0,42 ± 0,06	0,15 ± 0,08
Aleatória	0,22 ± 0,14	0,21 ± 0,13	0,17 ± 0,26	0,17 ± 0,30
Zigzag	0,14 ± 0,02	0,14 ± 0,02	0,18 ± 0,06	0,45 ± 0,10
Gcen Gmax	0,14 ± 0,02	0,14 ± 0,02	0,18 ± 0,06	0,46 ± 0,09

Tabela 24 – Médias do tempo de busca do algoritmo 11 no cenário do atendimento médico.

rem com a estratégia Gmin Gmax e diminui, respectivamente, ao executarem com as estratégias Zigzag, Aleatória e Gcen Gmax. Além do mais, conforme apresentam tais tabelas, o par (NGK,Gcen Gmax) é a melhor configuração para os algoritmos 8 e 10 obterem o menor tempo de busca para encontrar a norma eficiente.

Conforme a tabela 24, para cada função, exceto NGJ, algoritmo 11 obtém médias de tempo de busca maiores ao executar com a estratégia Gmin Gmax. Todavia, as médias diminuem quando o algoritmo 11 executa, respectivamente, com a estratégia Aleatória e com as estratégias Gcen Gmax e Zigzag que, por sua vez, apresentam resultados iguais. Logo, neste cenário, o algoritmo 11 também possui mais de uma melhor configuração para obter o menor tempo de busca, a saber os pares (NGK,Zigzag), (NGK,Gcen Gmax), (NGF,Zigzag) e (NGF,Gcen Gmax). Além disto, o par (NGJ,Gmin Gmax) é também uma configuração igualmente boa para o algoritmo 11 para tal fim.

Para cada função, exceto NGJ, ao compararmos as estratégias Gcen Gmax e Gmin Gmax, podemos observar que Gcen Gmax possui, conforme a tabela 22, médias de tempo, no mínimo, 36,87% menores do que as médias da estratégia Gmin Gmax e, conforme a tabela 23, Gcen Gmax possui médias de tempo, no mínimo, 53,72% menores. Por outro lado, ao compararmos as estratégias Gcen Gmax e Zigzag com a

estratégia Gmin Gmax, podemos observar que as duas primeiras estratégias possuem, conforme a tabela 24, médias de tempo, no mínimo, 53,33% menores do que as médias da terceira estratégia. Além do mais, de acordo com todas as tabelas deste cenário, para todas as estratégias, o maior desvio padrão ocorre com a função NGJ, enquanto que, para cada função, o maior desvio padrão ocorre com a estratégia Aleatória.

5.3 DISCUSSÃO

Conforme os experimentos deste capítulo, nós observamos três principais resultados. Primeiramente, através da busca por normas quase eficientes, nós constatamos que é possível encontrar normas eficientes. Conforme discutido nos experimentos do capítulo 4, as normas $n = \{at(Ag_2, L_2)\} \xrightarrow{p} move(Ag_1, L_1, L_2)$, $n' = \{lowCredential(Ag_1), confidencial(Doc_1)\} \xrightarrow{p} open(Ag_1, Doc_1)$ e $n'' = \{seated(Ag_2, C_2), next(C_2, C_1)\} \xrightarrow{p} sit(Ag_1, C_1)$ sintetizadas pelos algoritmos 8, 10 e 11, são eficientes. Contudo, a síntese de tais normas é realizada de uma forma mais simples sem a necessidade de determinar a generalidade das normas e as histórias do sistema. Segundo, a análise da alcançabilidade dos objetivos do sistema auxilia a construir espaços de busca com normas alinhadas a comportamentos que necessitam de regulação e que permitem aos agentes executarem, pelo menos, uma sequência de ações que os levem aos objetivos do sistema. Terceiro, os aprimoramentos empregados pelos algoritmos 8, 10 e 11 podem reduzir consideravelmente o número de normas analisadas e o tempo de busca por normas eficientes ao obterem espaços de normas menores.

Embora os aprimoramentos empregados pelos algoritmos 8, 10 e 11 sejam capazes de melhorar o desempenho da síntese de normas, algumas perguntas sobre os resultados obtidos necessitam ser ponderadas: i) como as configurações impactam a síntese de normas? ii) como a análise da alcançabilidade impacta os resultados dos algoritmos? e, iii) qual é o algoritmo mais apropriado para a síntese de normas? Nas seções seguintes, nós discutimos as respostas para estas perguntas.

5.3.1 Como as configurações impactam a síntese de normas?

Em suma, as configurações impactam o *número de normas analisadas e o tempo de busca*. Contudo, certas combinações de funções de generalidade e estratégias de visita mostram ser mais apropriadas para a síntese de normas eficientes nos cenários avaliados. Diferentes razões estão relacionadas a isto. Primeiramente, conforme discutido no capítulo 4, as funções de generalidade impactam na organização do espaço das normas ao determinarem: i) o número de conjuntos de generalidade que constituem as sequências aproximadas; ii) o tamanho de tais conjuntos; e iii) a posição das normas eficientes nas sequências. Consequentemente, as funções impactam no desempenho da busca, pois certas sequências podem auxiliar as estratégias de visita

a encontrar as normas eficientes de forma mais eficaz permitindo economizar tempo e esforço computacional, enquanto outras sequências podem favorecer pouco ou nada o encontro de tais normas. Ainda, de acordo com a localização das normas eficientes nas sequências aproximadas, as estratégias de visita possibilitam explorar uma parte reduzida do espaço das normas ao determinarem o ponto de partida da busca e a ordem na qual os conjuntos de generalidade são percorridos, o que é especialmente útil em espaços de normas grandes. Desta forma, certas estratégias de visita podem ser mais apropriadas para a síntese de normas do que outras, pois assim como certas sequências, elas viabilizam encontrar as normas eficientes mais rapidamente.

Para auxiliar a responder a pergunta desta seção com mais propriedade, as tabelas 25, 26 e 27 sumarizam as *configurações igualmente boas*¹² para os algoritmos 8, 10 e 11, respectivamente, nos cenários dos experimentos da seção anterior. Observamos que as melhores configurações para um algoritmo analisar o menor número de normas são também as melhores configurações para o algoritmo obter o menor tempo de busca. Estas configurações estão representadas nas tabelas pelo símbolo ▼. Observamos também que certas configurações são *aproximadamente iguais* às melhores configurações para um algoritmo analisar o menor número de normas, todavia, elas fazem parte das melhores configurações para o algoritmo obter o menor tempo de busca. Isso ocorre quando a diferença entre as médias de normas analisadas é pequena, o que não resulta em uma diferença significativa no tempo de busca. Estas configurações estão representadas nas tabelas pelo símbolo ▽. Por outro lado, observamos que certas configurações são *aproximadamente iguais* às melhores configurações tanto para o número de normas analisadas quanto para o tempo de busca pelas normas eficientes. Estas configurações estão representadas nas tabelas pelo símbolo ▾.

EstratégiaFunção	Tráfego de Carros				Segurança de Acesso				Atendimento Médico			
	NGK	NGF	NGI	NGJ	NGK	NGF	NGI	NGJ	NGK	NGF	NGI	NGJ
Gmin Gmax												
Gcen Gmax	▼					▼			▼			
Zigzag	▽											
Aleatória												

Tabela 25 – Configurações igualmente boas para o algoritmo 8.

EstratégiaFunção	Tráfego de Carros				Segurança de Acesso				Atendimento Médico			
	NGK	NGF	NGI	NGJ	NGK	NGF	NGI	NGJ	NGK	NGF	NGI	NGJ
Gmin Gmax												▽
Gcen Gmax	▼				▽	▼			▼			
Zigzag	▽				▽	▽						
Aleatória												

Tabela 26 – Configurações igualmente boas para o algoritmo 10.

¹² Conforme discutido na seção anterior, as configurações igualmente boas dividem-se entre as melhores configurações e as configurações aproximadamente iguais.

EstratégiaFunção	Tráfego de Carros				Segurança de Acesso				Atendimento Médico			
	NGK	NGF	NGI	NGJ	NGK	NGF	NGI	NGJ	NGK	NGF	NGI	NGJ
Gmin Gmax	▼	▼	▼	▼								▼
Gcen Gmax					▼	▼	▼		▼	▼		
Zigzag					▼	▼	▼		▼	▼		
Aleatória												

Tabela 27 – Configurações igualmente boas para o algoritmo 11.

Observa-se ainda que: a) para todos os cenários, à medida que o tamanho do espaço das normas diminui, o número de configurações igualmente boas aumenta; e b) exceto para o algoritmo 11 no cenário do tráfego de carros, o conjunto das configurações para os algoritmos obterem tanto o menor número de normas analisadas quanto o menor tempo de busca é composto, pelo menos, por (NGK,Gcen Gmax) ou (NGF,Gcen Gmax). Para a observação a, podemos dizer que o impacto das configurações na síntese de normas tem relação com o tamanho do espaço das normas. Espaço grandes viabilizam as funções de generalidade obterem sequências heterogêneas (i.e., sequências que divergem significativamente entre si possuindo distintos tamanhos, médias de normas nos conjuntos de generalidade e/ou distribuição das normas em tais conjuntos). Desta forma, os resultados dos algoritmos tendem a divergir e poucas configurações mostram-se boas opções para a síntese, pois a maioria delas apresenta resultados que distanciam-se significativamente dos resultados das melhores configurações.

Por outro lado, conforme o espaço das normas torna-se menor, as sequências que as funções de generalidade obtêm tendem a ser homogêneas (i.e, sequências que tendem a se aproximar significativamente em tamanho, média de normas nos conjuntos de generalidade e/ou distribuição das normas em tais conjuntos). Desta forma, os resultados dos algoritmos tendem a ser mais próximos. Assim, mais configurações são boas opções para a síntese de normas, pois mais configurações apresentam resultados que se aproximam dos resultados das melhores configurações. De acordo com as tabelas 23, 24 e 25, o número de configurações igualmente boas para os algoritmos 8, 10 e 11 no cenário do Tráfegos de Carros é, respectivamente, 2, 2 e 4, no cenário da Segurança de Acesso é, respectivamente, 1, 4 e 6 e no cenário do atendimento médico é 1, 2 e 5 configurações.

Para a observação b, podemos considerar dois pontos da discussão do capítulo 4: primeiramente, a posição das normas eficientes nas sequências aproximadas, e segundo, o tamanho médio dos conjuntos de generalidade de tais sequências. Sobre o primeiro ponto, indagamos que embora a posição das normas eficientes nas sequências aproximadas possa destoar da posição que ocupam nas sequências semânticas, por satisfazerem a propriedade 1 nos cenários dos experimentos, as funções NGK e NGF mapeiam as normas eficientes a partir do meio para a direita nas sequências aproximadas. Sobre o segundo ponto, indagamos que as funções NGK e NGF tendem a mapear as normas a um número maior de conjuntos de generalidade, o que faz com

que a média de normas nos conjuntos das sequências que elas obtêm seja menor do que nas demais sequências. Logo, o total de normas a ser analisado entre G^{cen} e G^{apex} pode ser menor nas sequências obtidas pelas funções NGK e NGF, e como a estratégia Gcen Gmax parte do meio das sequências para a direita, as configurações (NGF,Gcen Gmax) e (NGK,Gcen Gmax) tendem a apresentar resultados melhores do que as demais configurações. Embora a estratégia Zigzag também parta do meio das sequências para a direita, caso as normas eficientes não estejam em torno de G^{cen} , por visitar *ambos os lados das sequências alternativamente*, a estratégia Zigzag tende a analisar mais normas e demandar mais tempo de busca do que a estratégia Gcen Gmax. Para certos cenários, a estratégia Zigzag apresenta resultados aproximadamente iguais a estratégia Gcen Gmax sobre as sequências obtidas pelas funções NGK e NGF, contudo, as configurações (NGF,Zigzag) e (NGK,Zigzag) não fazem parte do conjunto das melhores configurações de nenhum algoritmo. Portanto, para os cenários dos experimentos, as configurações (NGK,Gcen Gmax) e (NGF,Gcen Gmax) são as mais apropriadas para a síntese de normas eficientes.

Por outro lado, as configurações (NGF,Gmin Gmax) e (NGK,Gmin Gmax) tendem a apresentar números de normas analisadas e tempos de busca maiores do que as configurações anteriores. Em suma, isto ocorre devido à estratégia Gmin Gmax iniciar a busca pelo conjunto G^{min} que é, possivelmente, dentre os conjuntos de generalidade por onde as estratégias iniciam a busca, o conjunto mais distante de G^{apex} . Por sua vez, as configurações (NGF,Aleatória) e (NGK,Aleatória) também tendem a apresentar números de normas analisadas e tempos de busca altos. De acordo com os experimentos, os resultados de tais configurações são próximos aos resultados das configurações (NGF,Gmin Gmax) e (NGK,Gmin Gmax). Contudo, as configurações constituídas pela estratégia Aleatória apresentam os maiores desvios-padrão. Embora seja possível encontrar as normas eficientes visitando aleatoriamente poucos conjuntos de generalidade, a probabilidade de que isto ocorra pode ser pequena¹³. Além do mais, a busca com a estratégia Aleatória pode requerer analisar um número de normas consideravelmente maior do que a média obtida nos experimentos e fazer com que o tempo de busca seja muito maior do que o esperado. Como o custo para realizar a busca ao acaso é provavelmente alto, concluímos que explorar o espaço com a estratégia Aleatória não é uma boa opção para a síntese de normas eficientes.

Por sua vez, a função NGJ, que não satisfaz a propriedade 1, mapeia as normas eficientes a ambos os lados das sequências. Mesmo que o espaço das normas seja grande, as sequências obtidas por tal função tendem a ser as menores, pois ao explorar unicamente as variáveis das normas, a função NGJ mapeia certas normas contíguas a um mesmo conjunto. Isto faz com que as sequências que ela obtêm sejam formadas

¹³ Podemos dizer que a probabilidade de que o conjunto G^{apex} seja visitado inicialmente é $\frac{1}{k}$, onde k refere-se ao número de conjuntos de generalidade da sequência aproximada.

por um número menor de conjuntos de generalidade que, por sua vez, são formados por uma média maior de normas. Embora poucos conjuntos sejam visitados em tais sequências, conforme os experimentos, as configurações constituídas pela função NGJ apresentam um desvio padrão alto, pois a grande quantidade de normas em G^{apex} faz com que o número de normas analisadas nas buscas realizadas sejam significativamente distantes entre si. Embora possamos verificar que a configuração (NGJ, Gmin Gmax) faz parte das melhores configurações do algoritmo 11 no cenário do Atendimento Médico, devido ao desvio padrão alto, realizar a busca em sequências obtidas pela função NGJ pode, assim como ocorre com as configurações constituídas pela estratégia Aleatória, requerer analisar um número de normas muito maior do que os resultados dos experimentos e demandar mais tempo para encontrar as normas eficientes. Portanto, concluímos que usar a função NGJ para organizar o espaço das normas não é uma boa opção para a síntese de normas eficientes.

Por fim, a função NGI obtém sequências cujos tamanhos são intermediários aos tamanhos das sequências obtidas pelas demais funções. Os conjuntos de generalidade das sequências obtidas pela função NGI são em média maiores do que os conjuntos de generalidade das sequências obtidas pelas funções NGK e NGF e menores do que os conjuntos de generalidade das sequências obtidas pela função NGJ. Isto faz com que os resultados das configurações constituídas pela função NGI tendam a ser maiores do que as configurações constituídas pelas funções NGK e NGF, contudo, menores do que os resultados das configurações constituídas pela função NGJ.

De acordo com o exposto, duas perguntas secundárias necessitam ser respondidas:

1. *O número de características sintáticas das normas exploradas pelas funções de generalidade impacta a síntese de normas eficientes?*

De acordo com os resultados, podemos observar que o número de características sintáticas das normas exploradas pelas funções de generalidade impacta a estrutura das sequências aproximadas, o que, por sua vez, impacta o desempenho da busca pelas normas eficientes. As funções NGF e NGK exploram *duas características sintáticas* das normas, as fórmulas atômicas e as variáveis, o que lhes permite obter sequências aproximadas com conjunto de generalidade em média menores. Ao mapearem as normas eficientes na metade superior das sequências aproximadas, as funções NGF e NGK viabilizam as estratégias Gcen Gmax e Zigzag desconsiderar parte do espaço de busca (i.e., a metade inferior, no caso de Gcen Gmax, ou parte dele, no caso de Zigzag) e assim economizar tempo e esforço computacional.

A função NGI explora apenas *uma característica sintática* das normas, as fórmulas atômicas, o que faz com que ela obtenha sequências aproximadas com conjuntos de generalidade em média maiores do que as sequências obtidas pelas funções NGK e NGF. Embora a função NGI também mapeie as normas eficientes na metade su-

perior das sequências aproximadas, por terem mais normas em seus conjuntos, tais sequências auxiliam as estratégias de visita na busca pelas normas eficientes de uma forma mais sutil. Por outro lado, a função NGJ, que também explora *uma característica sintática* das normas, as variáveis, não apresenta um padrão de região onde mapear as normas eficientes nas sequências aproximadas. Logo, não há um indicativo de qual estratégia pode se beneficiar da estrutura de uma sequência obtida por NGJ.

Dado que considerar *duas características sintáticas* para organizar o espaço das normas traz *mais benefícios* à busca nos cenários dos experimentos do que considerar apenas uma de tais características, as funções NGK e NGF sobressaem-se às funções NGI e NGJ. Embora neste momento não tenhamos como saber *a priori* se para um dado domínio o ideal é explorar uma ou duas características sintáticas das normas para organizar o espaço, os resultados obtidos sugerem que explorar duas características sintáticas deve resultar em ordenações do espaço das normas mais apropriadas para a síntese de normas eficientes. Como as funções NGK e NGF exploram as mesmas características sintáticas, a diferença entre as sequências que elas obtêm está, basicamente, na ordenação das normas em relação ao número de variáveis (enquanto NGK a faz de forma crescente, NGF a faz de forma decrescente). Nesta direção, nós consideramos ambas as funções igualmente apropriadas para organizar o espaço das normas.

2. Qual estratégia de visita é mais apropriada para a síntese de normas eficientes?

Conforme discutido anteriormente, a busca pelas normas eficientes ao acaso pode resultar em um número alto de normas analisadas. Como isto pode levar a estratégia Aleatória a um desempenho ruim, nós julgamos as configurações constituídas por tal estratégia opções pouco vantajosas para a síntese de normas eficientes. Considerando que as normas eficientes estão na metade superior das sequências aproximadas, por iniciar a busca pelo conjunto G^{min} , a estratégia Gmin Gmax não é uma boa opção para a síntese de normas eficientes, pois parte de um conjunto possivelmente distante de G^{apex} . Além do mais, como a estratégia Zigzag requer visitar ambos os lados das sequências alternativamente, um número excessivo de normas pode ser analisado caso o conjunto G^{apex} não esteja em torno de G^{cen} . Por outro lado, como a estratégia Gcen Gmax visita primeiramente a metade superior das sequências aproximadas, ela tende a visitar o conjunto G^{apex} em menos passos do que as demais estratégias. De acordo com essas observações, nós consideramos Gcen Gmax a estratégia mais apropriada para a síntese de normas eficientes.

5.3.2 Como a análise da alcançabilidade impacta a síntese de normas eficientes?

A análise da alcançabilidade tem como intuito sintetizar normas considerando o alcance dos objetivos do sistema. Ela permite antever se as normas que compõem um espaço selecionam, pelo menos, uma história de sucesso. Para isto, a análise da alcançabilidade determina, primeiramente, a existência de um traço viabilizador, conforme a definição 32, para auxiliar os algoritmos 10 e 11 na identificação de uma ação proibitiva. Desta forma, a análise da alcançabilidade direciona os algoritmos 10 e 11 a criarem espaços cujas normas proíbem unicamente ações que podem levar os agentes a estados de conflito e que viabilizam o alcance dos objetivos do sistema. Logo, tais espaços são formados por normas melhor alinhadas aos comportamentos dos agentes que necessitam de regulação, diferentemente do algoritmo 8 que, sem uma análise prévia de quais ações podem ocasionar situações indesejáveis, considera todas as ações do conjunto \mathcal{A} para construir um espaço de normas. Desta forma, a análise da alcançabilidade fomenta aos algoritmos 10 e 11 construírem espaços de normas menores do que o algoritmo 8, o que pode diminuir o tempo de busca pelas normas eficientes, conforme mostram os experimentos deste capítulo.

Todavia, é possível que a análise da alcançabilidade descarte normas eficientes que proíbam ações distintas da ação proibitiva considerada na construção do espaço das normas. Além disso, embora parte das normas descartadas possa manter os objetivos do sistema alcançáveis, a análise da alcançabilidade empregada pelos algoritmos 10 e 11 desconsidera do espaço todas as normas que não mantêm os objetivos do sistema alcançáveis. Portanto, podemos verificar que a hipótese 2 é válida.

Além disso, a análise da alcançabilidade é uma tarefa que pode ter um custo alto e a complexidade dos algoritmos 10 e 11 é dependente dela. Como reduzimos a análise da alcançabilidade a um problema de *planejamento clássico*, a sua complexidade é NP-hard (GHALLAB; NAU; TRAVERSO, 2004). Portanto, em outros cenários de aplicação, o tempo de busca dos algoritmos 10 e 11 pode ser maior do que o tempo de busca do algoritmo 8 devido à análise da alcançabilidade poder demandar um tempo maior para determinar a existência de um traço viabilizador. Além disto, em outras situações, é possível que as simulações do SMAn resultem em histórias de conflito grandes. Logo, a etapa de backtracking pode visitar muitos estados fazendo com que a análise da alcançabilidade seja executada mais vezes do que o observado nos experimentos. Consequentemente, o tempo de busca pelas normas eficientes pode ser muito maior.

5.3.3 Qual é o algoritmo mais apropriado para a síntese de normas eficientes?

Conforme os experimentos deste capítulo, não podemos concluir qual dentre os algoritmos 8, 10 e 11 é o mais apropriado para a síntese de normas eficientes. Embora

o algoritmo 8 tenda a explorar espaços de normas maiores, o que pode demandar um tempo de busca alto, tais espaços podem ser constituídos por mais normas eficientes do que os espaços construídos pelos algoritmos 10 e 11. Contudo, os espaços obtidos pelo algoritmo 8 podem ser constituídos também por uma grande quantidade de normas que não estão alinhadas a comportamentos que necessitam de regulação, o que pode resultar em uma grande parte da busca sendo gasta com normas que não possuem utilidade ao sistema. Por outro lado, análise da alcançabilidade pode fazer com que os algoritmos 10 e 11 explorem espaços de normas consideravelmente menores resultando em ganho de tempo para encontrar normas eficientes. Não obstante, de acordo com a seção 5.3.2, a análise da alcançabilidade é um processo custoso que e, ao contrário da afirmação anterior, pode fazer com que o tempo de busca dos algoritmos 10 e 11 seja consideravelmente maior do que o tempo do algoritmo 8.

A fim de apresentar resultados mais assertivos e uma discussão mais detalhada sobre os benefícios e limitações dos algoritmos 8, 10 e 11, ainda há necessidade de uma investigação mais abrangente e profunda em diferentes domínios para concluirmos sobre o algoritmo mais apropriado para a síntese de normas eficientes. Contudo, para certas circunstâncias, é possível que façamos sugestões sobre o algoritmo a ser utilizado. Neste contexto, a seguir, nós listamos cinco situações a serem ponderadas para a síntese de normas eficientes.

1. O espaço das normas é grande. O espaço das normas é um conjunto potencialmente grande. Somando-se a isto, ao simular o SMAN múltiplas vezes para cada norma do espaço, encontrar normas eficientes pode demandar um tempo alto e um grande esforço computacional. A fim de tornar a busca um processo menos custoso, pode-se considerar diminuir o tamanho do espaço das normas. Neste contexto, nós sugerimos aplicar o algoritmo 11, pois a síntese de normas que ele emprega pode considerar um número menor de fórmulas atômicas estritamente unground e ações do que os algoritmos 8 e 10, o que, por sua vez, pode resultar em espaços de normas menores. Além do mais, para determinar se o espaço das normas para um dado cenário é grande, pode-se calcular o número de normas através da definição 15.

2. A alcançabilidade dos objetivos do sistema não é relevante. Em domínios onde deseja-se apenas evitar conflitos, não é necessário verificar se as normas selecionam comprovadamente, pelo menos, uma história de sucesso. Portanto, a análise da alcançabilidade dos objetivos do sistema pode ser desconsiderada na síntese de normas. Neste contexto, o algoritmo 8 é mais adequado, pois a busca que ele aplica se concentra unicamente em encontrar normas que satisfaçam os limiares desejados.

3. A alcançabilidade dos objetivos do sistema é relevante. Em domínios onde sintetizar normas que garantam a alcançabilidade dos objetivos do sistema é relevante, pode-se empregar o algoritmo 10 ou 11. Como o algoritmo 11 pode construir

espaço menores do que o algoritmo 10, nós sugerimos considerar o algoritmo 11.

4. O tempo de busca é um recurso escasso. Embora os experimentos deste capítulo mostrem que o algoritmo 8 obtém para os cenários considerados os maiores tempos de busca pelas normas eficientes, conforme discutido anteriormente, os algoritmos 10 e 11 podem apresentar tempos de busca superiores ao algoritmo 8 caso a análise da alcançabilidade demande um tempo grande. Logo, não podemos indicar com assertividade qual algoritmo é o ideal para a síntese de normas em cenários onde o tempo de busca é um recurso escasso. Todavia, certas características do SMAn podem ser úteis para auxiliar na escolha de um algoritmo. Em sistemas onde o número de agentes e ações são pequenos, por exemplo, o tempo para encontrar traços viabilizadores é possivelmente menor do que em sistemas onde o número de agentes e ações é grande. Basicamente, isto ocorre porque menos ações podem ser executadas por estados, o que diminui as ramificações de traços a serem exploradas na análise da alcançabilidade, e menos agentes necessitam cumprir as suas tarefas, o que pode resultar em traços viabilizadores menores. Neste contexto, nós sugerimos o algoritmo 11. Caso contrário, nós sugerimos que seja ponderado se garantir a alcançabilidade dos objetivos do sistema é relevante, conforme os itens 2 e 3 desta seção.

5. O tempo de busca é um recurso abundante. Caso o tempo necessário para sintetizar normas não seja uma preocupação, nós sugerimos que o algoritmo 11 seja considerado. Deste modo, mesmo que a análise da alcançabilidade possa ser custosa, as normas sintetizadas garantem que os agentes possam satisfazer os objetivos do sistema através de, pelo menos, uma história de sucesso.

Em suma, de acordo com a presente discussão, nós consideramos as funções NGK e NGF ambas igualmente apropriadas para organizar o espaço das normas e Gcen Gmax a estratégia mais apropriada para visitar espaços de normas ordenados. Além disso, nós sugerimos: i) aplicar o algoritmo 8 quando a alcançabilidade dos objetivos do sistema não é relevante; ii) aplicar o algoritmo 10 ou 11 quando a alcançabilidade dos objetivos do sistema é relevante; iii) aplicar o algoritmo 11 quando o espaço das normas é grande ou tempo de busca é um recurso abundante. iv) analisar características do domínio e ponderar se a alcançabilidade dos objetivos do sistema é relevante para decidir qual algoritmo usar quando o tempo de busca é um recurso escasso.

5.3.4 Considerações Finais

A síntese de normas eficientes é um problema de solução não-trivial. Contudo, nesta tese, nós mostramos resultados promissores para este campo dos SMAn. Primeiramente, pudemos verificar que a hipótese 1, embora formulada sobre as sequências semânticas, pôde ser verificada, conforme os experimentos do capítulo 4, em sequên-

cias aproximadas. Contudo, como a posição das normas eficientes nas sequências semânticas e aproximadas pode divergir, nós também observamos normas eficientes em outras regiões da metade superior das sequências aproximadas.

Considerando que, para diferentes domínios, isto deve ocorrer em sequências aproximadas obtidas a partir de funções de generalidade crescentes (propriedade 1), então nós estabelecemos um meio de podar parte do espaço de busca ao aplicarmos estratégias de visita que iniciam pelo meio das sequências aproximadas. Portanto, conclui-se que ordenar as normas conforme a sua generalidade possui um impacto positivo para a síntese de normas eficientes.

Pudemos verificar também que a análise da alcançabilidade é uma ferramenta valiosa para a síntese de normas eficientes, não somente por assegurar que as normas de um espaço selecionam, pelo menos, uma história de sucesso, mas também viabilizam a construção de espaços menores, o que pode impactar positivamente no tempo e esforço para que tais normas sejam sintetizadas.

Contudo, o problema da síntese de normas eficientes ainda não foi definitivamente solucionado. Devido a complexidade inerente do problema, o problema continua sendo desafiador e requer mais investigação. Embora não tenhamos atingido uma resolução definitiva, lançamos luz sobre diversas nuances do problema da síntese de normas eficientes e abrimos portas para futuras investigações.

Adicionalmente, consideramos que a nossa abordagem é um mecanismo versátil para a síntese de normas, pois pode ser facilmente estendida para sintetizar normas de diferentes generalidades e aplicar diferentes estratégias de visita (além das exploradas). Isso é particularmente útil em cenários onde é desejável sintetizar normas mais flexíveis e tolerantes a conflito ou normas mais rígidas, por exemplo. Para isto, pode-se aplicar estratégias de visita que iniciam a busca na região onde as normas com a generalidade desejada são mais prováveis de serem localizadas e avançam para os demais conjuntos de generalidade de acordo com critérios estabelecidos.

6 CONCLUSÃO E TRABALHOS FUTUROS

Esta tese investigou a problemática da *síntese automática de normas eficientes para SMAn*. Enquanto normas muito específicas regulam pouco os agentes dando-lhes uma grande liberdade para executarem as suas ações, o que tende a gerar situações de conflitos, normas muito gerais regulam os agentes em excesso podendo reduzir significativamente a sua autonomia e inviabilizar o alcance dos objetivos do sistema. Sob estas circunstâncias, nós argumentamos que para ser eficiente, uma norma deve possuir uma generalidade balanceada, isto é, a norma não deve ser muito específica nem muito geral e deve ser capaz de manter um equilíbrio entre o controle que exerce e a autonomia dos agentes. Para isto, nós consideramos que explorar a generalidade das normas e a análise da alcançabilidade dos objetivos do sistema na síntese de normas é fundamental para que normas eficientes possam ser devidamente sintetizadas.

Determinar a generalidade das normas é um problema que requer, no contexto deste trabalho, analisar todos os estados do SMAn. Como esta tarefa pode ser inviável, nós propomos uma alternativa para estimar a generalidade das normas através de uma perspectiva sintática. Isto nos possibilitou ordenar as normas de diferentes formas, aproximadamente, das mais específicas para as mais gerais e aplicar diferentes estratégias de busca por normas eficientes que se beneficiam da organização do espaço das normas. Aliada às estratégias, nós aplicamos a análise da alcançabilidade dos objetivos do sistema a fim de determinar se uma norma candidata a regular os agentes garante que, ao obedecerem, os agentes possam satisfazer os objetivos do sistema através de, pelo menos, uma sequência de ações. Embora a análise da alcançabilidade também possa ser um problema complexo, nós assumimos este custo para que as normas garantam o alcance dos objetivos do sistema.

Contudo, para certificar que uma norma é eficiente, é necessário analisar todas as sequências de ações dos agentes para aferir se a norma não permite a ocorrência de conflitos. Porém, a quantidade de sequências de ações dos agentes pode explodir combinatoriamente. Para contornarmos este problema, nós conceitualizamos as normas *quase eficientes*. Uma norma é quase eficiente caso satisfaça o quesito das normas eficientes de manter os objetivos do sistema alcançáveis, entretanto, a necessidade de que o conjunto de sequências de ações dos agentes que geram conflitos seja vazio é relaxada, embora seja desejável que tal conjunto seja pequeno ou vazio. Então, nós definimos um método para determinar normas quase eficientes, onde analisamos o desempenho do SMAn ao executar com uma norma múltiplas vezes em termos de taxa de sucesso (a norma regula os agentes apropriadamente de modo que nenhum conflito ocorre e os objetivos do sistema são alcançados) e conflito (a norma não regula os agentes apropriadamente, pois há ocorrência de conflito). Caso o desempenho do sistema satisfaça certos limiares, então a norma é considerada quase eficiente.

Neste sentido, considerando situações onde as taxas de sucesso e conflito de uma norma quase eficiente sejam, respectivamente, maior que zero e zero, então a norma é provavelmente eficiente, pois como o SMAN executa múltiplas vezes, múltiplas situações que levam a conflitos são evitadas e os objetivos do sistema são alcançados (embora isto não signifique que todas as situações de conflito existentes sejam verificadas ou que a norma seja capaz de evitar todas elas). Além disso, considerando a hipótese 1, que postula sobre as normas do meio das sequências possuírem uma generalidade balanceada, as normas quase eficientes localizadas em torno de tal região são mais prováveis de serem eficientes do que as normas quase eficientes localizadas em outras regiões das sequências que, por sua vez, podem regular pouco ou em excesso os agentes. Considerando isto, as estratégias de visita podem iniciar no meio das sequências e economizar tempo e esforço computacional caso normas eficientes sejam encontradas em tal região. Portanto, a síntese de normas eficientes ocorre como uma busca por normas quase eficientes em espaços de normas ordenados segundo a generalidade das normas combinando estratégias de visita com a análise da alcançabilidade dos objetivos do sistema. Para isso, a síntese considera encontrar normas quase eficientes que satisfaçam determinadas taxas de sucesso e conflito que uma norma eficiente supostamente possui.

Por fim, nós apresentamos um conjunto de algoritmos para a síntese de normas eficientes construídos de forma incremental através de uma sucessão de aprimoramentos. Durante a construção, nós apontamos as vantagens e desvantagens de cada algoritmo e realizamos a implementação de alguns deles que, por sua vez, usam como base a nossa abordagem para a síntese das normas quase eficientes. Nós realizamos um conjunto de experimentos, onde comparamos o desempenho de tais algoritmos em termos do número de normas analisadas e tempo de busca para encontrar as normas eficientes. Em nossos experimentos, nós variamos as estratégias sintáticas para organizar o espaço das normas, assim como as estratégias de busca para encontrar as normas eficientes.

Como resultados nós constatamos que: i) através da busca por normas quase eficientes é possível encontrar normas eficientes; ii) certas estratégias para organizar e visitar o espaço de busca são mais apropriadas para a síntese de normas eficientes, pois aprimoram o desempenho da síntese; iii) a análise da alcançabilidade dos objetivos do sistema auxilia a construir espaços de busca com normas melhor alinhadas a comportamentos que necessitam de regulação e que permitem aos agentes executarem, pelo menos, uma sequência de ações que os levem aos objetivos do sistema; iv) a análise da alcançabilidade dos objetivos do sistema pode auxiliar a construir espaços de busca menores e diminuir o tempo de busca pelas normas eficientes.

A seguir, na seção 6.1 nós retomamos as perguntas de pesquisa deste trabalho, na seção 6.2 nós apresentamos as contribuições da tese e na seção 6.3 nós

apresentamos perspectivas de trabalhos futuros.

6.1 RESPOSTAS PARA AS PERGUNTAS DE PESQUISA

Na introdução desta tese, nós realizamos duas perguntas de pesquisa. Nesta seção, nós apresentamos as respostas das perguntas 1 e 2.

Pergunta 1. Dado um espaço de normas, como determinar as normas que possuem generalidade balanceada?

Conforme a hipótese 1, em um espaço cujas normas estão ordenadas crescentemente de acordo com a sua generalidade, as normas com generalidade balanceada encontram-se em torno do meio da sequência. Todavia, para todos os cenários dos nossos experimentos, na maioria das sequências resultantes, as normas eficientes foram localizadas na metade superior. Basicamente, isto ocorreu por três motivos. Primeiramente, as sequências que construímos são aproximações. Portanto, as posições das normas eficientes nas aproximações podem destoar das suas posições reais. Segundo, como as normas eficientes não permitem a ocorrência de conflitos, é possível que, para certos domínios, tais normas necessitem exercer mais controle sobre os agentes do que dar-lhes autonomia. Neste sentido, as normas podem necessitar ser mais gerais do que postula a hipótese 1. Terceiro, a generalidade que confere às normas a característica de serem eficientes é dependente de domínio. Enquanto para certos domínios a generalidade balanceada é menos restritiva, o que pode fazer com que as normas eficientes estejam localizadas na metade inferior das sequências, para outros domínios, ela necessita ser mais restritiva, fazendo com que as normas eficientes estejam localizadas na metade superior de tais sequências.

Pergunta 2. Como reduzir o tamanho do espaço das normas sem desconsiderar normas possivelmente eficientes?

O espaço das normas é um conjunto potencialmente grande. Logo, formas de diminuir o seu tamanho tornam-se essenciais para otimizar o tempo de busca pelas normas eficientes. Conforme a hipótese 2, o espaço das normas pode ser reduzido ao desconsiderarmos as normas que não mantêm os objetivos do sistema alcançáveis. Uma forma de fazer isto é considerar apenas normas para as quais existe, pelo menos, uma sequência de ações que alcança os objetivos do sistema. A fim de tornar esta possibilidade concreta, nós realizamos a síntese de normas eficientes durante a simulação do SMan buscando identificar ações que desencadeiam situações de conflito e que, mesmo sendo proibidas, os objetivos do sistema permanecem alcançáveis. Desta forma, sintetizamos normas considerando unicamente tais ações, o que pode auxiliar

a diminuir o tamanho do espaço das normas caso o número de ações considerado na síntese seja menor do que o número de ações do sistema. Logo, construímos espaços formados apenas por normas que mantêm os objetivos do sistema alcançáveis e que proíbem a execução de ações que levam a situações de conflito. Consequentemente, isto também auxilia a diminuir o tempo de busca por normas eficientes. Contudo, é possível que existam outras normas eficientes que proíbam ações diferentes daquelas consideradas na construção do espaço. Embora tais normas não façam parte da busca pelas normas eficientes, considerando que muitas normas não são boas opções para regular os agentes, concentrar a busca em espaços menores, mesmo que com um número de normas eficientes reduzido, pode ter vantagens sobre a busca em espaços grandes, onde, embora possam possuir um número maior de normas eficientes, são formados por uma grande quantidade de normas que não regulam os agentes adequadamente, o que pode demandar mais tempo e esforço computacional para que a síntese de normas eficientes ocorra de forma bem-sucedida.

6.2 CONTRIBUIÇÕES DA TESE

Como principais contribuições desta tese podemos citar:

1. Uma abordagem que considera os objetivos do sistema na síntese de normas.

Na literatura de Sistemas Multiagentes Normativos, a maioria dos trabalhos dão atenção aos conflitos e ignoram a alcançabilidade dos objetivos do sistema. Todavia, não há garantia de que os agentes ao terem certos comportamentos regulados por uma norma poderão realizar as suas tarefas devidamente e, assim, alcançarem os objetivos do sistema. Esta garantia ocorre se a norma viabiliza a execução de, pelo menos, uma sequência de ações que os agentes possam executar para alcançarem tais objetivos. Neste contexto, nós concebemos uma abordagem simbólica que integra a análise da alcançabilidade dos objetivos do sistema à síntese de normas eficientes, o que viabiliza sintetizarmos normas que, ao serem obedecidas, permitem aos agentes alcançar os objetivos do sistema.

2. Um ferramental teórico e formal para caracterizar a síntese de normas eficientes.

A área da síntese de normas em Sistemas Multiagentes Normativos carece de um ferramental teórico e formal para conceitualizar o problema da síntese de normas eficientes. Frente a esta limitação, nós apresentamos ao longo desta tese um conjunto de conceitos e definições que caracterizam formalmente a síntese de normas eficientes.

Neste contexto, nós identificamos e conceitualizamos também diferentes problemas inerentes à síntese de normas eficientes.

3. Estratégias para ordenar e visitar o espaço das normas.

Ao investigarmos formas de determinar a generalidade das normas, nós estipulamos diferentes estratégias para ordenar as normas de acordo com a sua generalidade. Também definimos diferentes estratégias de visita que se beneficiam da estrutura do espaço de busca ordenado para encontrar normas eficientes. A ordenação das normas também possibilita sintetizar normas de diferentes generalidades. Isso é particularmente útil em cenários onde é desejável sintetizar normas mais flexíveis e tolerantes a conflito ou normas mais rígidas, por exemplo. Para isto, pode-se aplicar estratégias de visita que iniciam a busca na região das sequências onde as normas da generalidade desejada são mais prováveis de serem localizadas e avançam para as demais regiões conforme critérios estabelecidos.

4. Estratégias para reduzir a complexidade da busca por normas eficientes.

A síntese de normas é um problema complexo. Todavia, a síntese de normas eficientes pode ser um problema mais custoso, pois estimar a generalidade das normas e analisar a alcançabilidade dos objetivos do sistema são problemas intrinsecamente difíceis. Com o intuito de tornar a síntese de normas eficientes um problema viável, nós propomos estratégias que simplificam este problema e tornam a sua investigação possível. Essas estratégias foram discutidas ao longo da tese e refletidas na construção dos algoritmos de síntese.

5. Uma abordagem automática para sintetizar normas.

Embora as normas sejam um tópico de grande interesse em Sistemas Multiagentes e exista uma vasta gama de trabalhos nesta área, o tópico da síntese de normas foi pouco explorada. A maioria dos trabalhos assume que as normas são conhecidas *a priori* ou sintetizadas manualmente pelo projetista do sistema. Por outro lado, a maioria dos trabalhos que propõe formas automáticas de sintetizar normas o fazem envolvendo os agentes em um processo conhecido como *convenção*. Neste contexto, é necessário que os agentes possuam conhecimento específico sobre como as normas impactam sobre as suas ações e entrem em um acordo sobre quais normas devem regulá-los, o que nem sempre é possível por diferentes razões como o prevalecimento de objetivos individuais divergentes e/ou egoístas, por exemplo. Portanto, um maquinário à parte da sociedade dos agentes é interessante para que a síntese de normas seja realizada acima dos objetivos individuais e a serviço do bem comum visando a satisfação dos

objetivos do sistema. Nesse sentido, propomos um conjunto de algoritmos para sintetizar normas eficientes de forma automática como um mecanismo externo que simula a execução do SMAn, observa os agentes, identifica situações de conflito e sintetiza normas para evitá-los e manter os objetivos do sistema alcançáveis.

6.2.1 Contribuições Secundárias da Tese

Além das contribuições elencadas anteriormente, de acordo com os resultados apresentados no capítulo 5, podemos indicar os seguintes itens como contribuições secundárias desta tese.

1. A análise da alcançabilidade pode auxiliar a diminuir o tempo de busca por normas eficientes.

Embora o intuito de considerar a análise da alcançabilidade dos objetivos do sistema na síntese de normas seja sintetizar normas que mantêm os objetivos do sistema alcançáveis, conforme discutido na resposta da pergunta de pesquisa 2, análise da alcançabilidade pode auxiliar a diminuir o tamanho do espaço das normas. Consequentemente, o tempo de busca por normas eficientes pode ser menor quando comparado ao tempo de uma busca realizada em espaços onde todas as normas de um domínio são consideradas. Em nossos experimentos, nós verificamos que em todos os cenários o tempo de busca pelas normas eficientes diminui quando a análise da alcançabilidade faz parte da síntese de normas. Todavia, é possível que para certos domínios, o tempo de busca por tais normas aumente quando a análise da alcançabilidade é considerada, pois este é um problema complexo que pode demandar tempo e esforço computacional.

2. O número de características sintáticas consideradas para determinar a generalidade das normas auxilia a diminuir o tempo de busca por normas eficientes.

A nossa abordagem para determinar a generalidade das normas é constituída por diferentes estratégias que consideram 1 ou 2 características sintáticas das normas. A generalidade atribuída às normas, por sua vez, é um valor real que denota o quão geral uma norma é de modo que, quanto mais geral, maior é o valor de generalidade da norma. A generalidade é utilizada para ordenar as normas obtendo uma sequência crescente. Contudo, o número de características sintáticas utilizado para determinar o valor de generalidade das normas influencia no tamanho das sequências, onde diferentes normas (possivelmente não comparáveis) podem ocupar uma mesma posição em uma dada sequência quando tais normas possuem o mesmo valor de generalidade. Estratégias que consideram 2 características sintáticas das normas tendem a

gerar sequências maiores do que as estratégias que consideram apenas 1 de tais características. Todavia, as sequências maiores tendem a possuir um número menor de normas que ocupam uma mesma posição. Considerando que um número menor de normas são analisadas por posição em sequências maiores, o total de normas a ser analisado entre a posição de partida da busca (como o meio das sequências) e a posição onde estão as normas eficientes pode ser menor em sequências maiores. Conseqüentemente, o tempo de busca em tais sequências pode ser menor. De acordo com os resultados apresentados no capítulo 5, para a maioria dos experimentos, obtivemos tempos menores quando a busca foi realizada em sequências obtidas a partir da exploração de 2 características sintáticas das normas.

3. Certas estratégias de visita são mais adequadas para a síntese de normas eficientes.

Nós apresentamos diferentes estratégias de visita para explorar as sequências de normas na busca por normas eficientes. Basicamente, tais estratégias consideram posições distintas para iniciar a busca assim como a ordem na qual as posições das sequências são visitadas. Considerando a resposta da pergunta de pesquisa 1, iniciar a busca por normas eficientes a partir do meio das sequências de normas e avançar para a metade superior tende a ser uma estratégia promissora. Nós verificamos isto através dos resultados apresentados no capítulo 5, onde tal estratégia obtém, para todos os experimentos, o menor número de normas analisadas e tempo de busca. Contrastando com isso, estratégias que visitam, primeiramente, a metade inferior para, então, visitar a metade superior, e estratégias que visitam as sequências aleatoriamente apresentam ambas os piores resultados para a síntese de normas eficientes.

6.3 TRABALHOS FUTUROS

As direções que seguem representam caminhos de trabalhos futuros que podem ser considerados para o avanço da síntese automática de normas eficientes.

1. Tornar a linguagem lógica mais geral.

Conforme a definição 1, a linguagem lógica utilizada como base para a construção e formalização dos conceitos e definições deste trabalho é uma linguagem de 1^o ordem restrita que, dentre os conectivos lógicos, considera unicamente a conjunção. Embora isto nos ajude a delimitar o espaço das normas, isto também restringe as formas como podemos descrevê-las. A fim de que a especificação de normas mais complexas seja possível, há necessidade de aumentarmos a expressividade da nossa linguagem lógica de forma a torná-la uma linguagem de 1^o ordem completa. Desta

forma, é possível sintetizarmos normas que combinam, por exemplo, conjunções e disjunções de fórmulas atômicas positivas ou negativas. Conseqüentemente, esta modificação nos permitirá regular mais comportamentos distintos dos agentes. Todavia, com isto há necessidade de verificar a validade da hipótese 1 sobre os espaços de normas ordenados que consideram as novas normas.

2. Considerar outros tipos de normas.

Conforme apresentado no capítulo 2, além das proibições, os SMAn consideram diferentes tipos de normas para regular os agentes como as *obrigações* e *permissões*. Para certos domínios, proibições eficientes podem não existir ou o emprego de obrigações ou permissões pode ser mais adequado. Neste sentido, para que possamos sintetizar normas eficientes para uma gama maior de domínios, é necessário investigarmos como o nosso ferramental teórico e formal pode ser adequado para que as diferentes normas regulativas sejam contempladas. Para isto, há também a necessidade de investigar se a hipótese 1 tem aplicabilidade em sequências de normas que consideram proibições, obrigações e permissões. Além disso, há possivelmente a necessidade de um método para determinar qual tipo de norma regulativa é mais adequada para o domínio de aplicação e a adaptação dos algoritmos do capítulo 5 para que o espaço de busca seja constituído pelas diferentes normas regulativas.

3. Considerar sintetizar múltiplas normas.

Conforme discutido no capítulo 1, para diferentes domínios do mundo real, não é possível encontrar uma norma que isoladamente regula os agentes apropriadamente. Para tais domínios, é necessário considerar diferentes normas que, ao regularem os agentes em conjunto, evitam conflitos e mantêm os objetivos do sistema alcançáveis. Para trabalharmos com tais domínios, há necessidade de adaptarmos os algoritmos do capítulo 5 para que explorem o espaço de busca procurando por múltiplas normas. Todavia, considerar múltiplas normas traz a tona novos problemas como os *conflitos entre normas*, por exemplo. É possível que uma norma contradiga outra ou o cumprimento de uma norma faça com que os agentes violem outras (SANTOS *et al.*, 2017). Nesse contexto, mecanismos para a detecção de conflitos entre normas são necessários (SILVESTRE *et al.*, 2019), o que significa termos que investigar como podemos considerá-los em nosso trabalho.

4. Considerar múltiplos conflitos.

Nesse trabalho, nós consideramos cenários com apenas um único conflito. Todavia, diferentes domínios do mundo real possuem mais de um tipo de conflito. Nesse

sentido, para contemplar tais problemas, podemos investigar como estender o nosso ferramental teórico e formal e algoritmos para que a síntese de normas eficientes possa considerar múltiplos conflitos.

5. Elencar a melhor norma eficiente.

O espaço das normas pode ser constituído por diferentes normas eficientes. Todavia, durante a busca por normas eficientes, retornamos a primeira norma quase eficiente encontrada. Um aprimoramento que podemos adicionar aos algoritmos do capítulo 5 é buscar pelas diferentes normas eficientes de um domínio e elencar a melhor norma. Conforme apresenta o capítulo 3, introduzimos a definição de norma mais flexível que, dentre um conjunto de normas com as quais é comparável, seleciona o maior número de histórias de sucesso. Desta forma, ao serem regulados por tal norma, os agente têm mais opções de sequências de ações que alcançam os objetivos do sistema. Logo, a norma mais flexível dá aos agentes mais liberdade para escolherem as suas ações. Consequentemente, a síntese da norma eficiente mais flexível é um problema fundamental para aumentar a autonomia dos agentes. Todavia, considerar sintetizar tal norma pode resultar em uma busca que explora todo o espaço das normas. Portanto, este tipo de síntese tende a ser mais custosa que a síntese de normas considerada neste trabalho.

6. Considerar mecanismos de sanção na síntese.

Em nosso trabalho, consideramos que os agentes optam por obedecer às normas. Todavia, as normas são mecanismos violáveis, o que faz com que seja necessário o emprego de mecanismos que detectam violações de normas e aplicam punições aos agentes infratores. Como o nosso foco se deteve sobre a síntese de normas eficientes, tais mecanismos não foram consideramos nos objetivos desta tese. Contudo, estes mecanismos podem vir a ser considerados como parte do nosso simulador de SMAn aumentando o uso prático das normas e a autonomia dos agentes. Todavia, isto significa dotar os agentes (ou parte deles) de um raciocínio normativo, isto é, os agentes devem analisar durante o seu ciclo de raciocínio os benefícios de obedecerem às normas e as consequências de violá-las para decidirem com propriedade sobre estarem ou não em conformidade com as normas.

7. Considerar a síntese de normas em SMAn abertos.

Em nossos experimentos, consideramos que os agentes que fazem parte da simulação do SMAn são somente aqueles que existem desde o início da sua execução. Nesse contexto, podemos considerar que a síntese de normas eficientes ocorre em

uma simulação de um SMAn fechado. Todavia, consideramos importante que a síntese de normas eficientes possa ocorrer em qualquer sistema independentemente de ser aberto ou fechado. Para isto, podemos simular a entrada e a saída de agentes no simulador de SMAn ou empregar Sistemas Multiagente reais da literatura, por exemplo. No segundo caso, existe a necessidade de sintetizarmos as normas de acordo com a sintaxe que os agentes do framework escolhido compreendem, além de analisarmos como adaptar os algoritmos implementados do capítulo 5 para que a síntese de normas ocorra em modo on-line.

8. Considerar uma integração com frameworks da literatura.

Uma possibilidade para evitar a simulação do SMAn nos algoritmos implementados do capítulo 5 é integrá-los ao ciclo global de frameworks da literatura. Para isto, existe a necessidade de investigarmos como tais algoritmos podem ser adaptados ao funcionamento de tais frameworks de modo que as normas possam ser devidamente sintetizadas e analisadas quanto a sua capacidade de evitar conflitos e manter os objetivos do sistema alcançáveis durante a própria execução do sistema. Além do mais, tal integração nos possibilita superar a limitação de que os agentes executam em turno.

REFERÊNCIAS

ABDUL KADIR, Mohd Rashdan; SELAMAT, Ali; KREJCAR, Ondrej. Normative Rule Extraction from Implicit Learning into Explicit Representation. *In: KNOWLEDGE Innovation Through Intelligent Software Methodologies, Tools and Techniques: Proceedings of the 19th International Conference on New Trends in Artificial Intelligence and Applications*. Netherlands: IOS Press, set. 2020. Chapter 1, p. 88–101. ISBN 9781643681146.

AKERLOF, George. The Economics of Caste and of the Rat Race and Other Woeful Tales. **The Quarterly Journal of Economics**, v. 90, n. 4, p. 599–617, 1976.

AKINKUNMI, Babatunde; BABALOLA, Florence. A norm enforcement mechanism for a time-constrained conditional normative framework. **Autonomous Agents and Multi-Agent Systems**, v. 34, jan. 2020.

ALECHINA, Natasha; BULLING, Nils; DASTANI, Mehdi; LOGAN, Brian. Practical Run-Time Norm Enforcement with Bounded Lookahead. *In: PROCEEDINGS of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*. [S.l.: s.n.], 2015. P. 443–451.

ALECHINA, Natasha; DASTANI, Mehdi; LOGAN, Brian. Programming Norm-Aware Agents. *In: (AAMAS '12)*, p. 1057–1064.

ALECHINA, Natasha; DE GIACOMO, Giuseppe; LOGAN, Brian; PERELLI, Giuseppe. Automatic Synthesis of Dynamic Norms for Multi-Agent Systems. *In: PROCEEDINGS OF THE 19TH INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING*. [S.l.: s.n.], ago. 2022. P. 12–21.

ALUR, Rajeev; HENZINGER, Thomas A.; KUPFERMAN, Orna. Alternating-Time Temporal Logic. **Journal of the ACM**, Association for Computing Machinery, New York, NY, USA, v. 49, n. 5, p. 672–713, set. 2002. ISSN 0004-5411.

AMSLER, Lisa. The evolution of social norms in conflict resolution. **Journal of Natural Resources Policy Research**, v. 6, p. 285–290, out. 2014.

APELDOORN, Daan; KERN-ISBERNER, Gabriele. Towards An Understanding of What is Learned: Extracting Multi-Abstraction-Level Knowledge from Learning Agents. *In: RUS, Vasile; MARKOV, Zdravko (Ed.)*. **Proceedings of the Thirtieth International**

Florida Artificial Intelligence Research Society Conference, FLAIRS 2017, Marco Island, Florida, USA, May 22-24, 2017. [S.l.]: AAI Press, 2017. P. 764–767.

BALKE, Tina; PEREIRA, Célia da Costa; DIGNUM, Frank; LORINI, Emiliano; ROTOLO, Antonino; VASCONCELOS, Wamberto; VILLATA, Serena. Norms in MAS: Definitions and Related Concepts. *In*: ANDRIGHETTO, Giulia; GOVERNATORI, Guido; NORIEGA, Pablo; TORRE, Leendert W. N. van der (Ed.). **Normative Multi-Agent Systems**. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2013. v. 4. (Dagstuhl Follow-Ups). P. 1–31. ISBN 978-3-939897-51-4.

BELLIFEMINE, Fabio; BERGENTI, Federico; CAIRE, Giovanni; POGGI, Agostino. Jade — A Java Agent Development Framework. *In*: **Multi-Agent Programming: Languages, Platforms and Applications**. Edição: Rafael H. Bordini, Mehdi Dastani, Jürgen Dix e Amal El Fallah Seghrouchni. Boston, MA: Springer US, 2005. P. 125–147. ISBN 978-0-387-26350-2.

BERGER, Peter L.; LUCKMANN, Thomas. **The Social Construction of Reality: A Treatise in the Sociology of Knowledge**. [S.l.]: Doubleday, 1966. (Anchor books). ISBN 9780385058988.

BICCHIERI, Cristina. **Norms in the Wild: How to Diagnose, Measure, and Change Social Norms**. [S.l.]: New York, US: Oxford University Press USA, 2016.

BICCHIERI, Cristina. Norms of Cooperation. **Ethics**, The University of Chicago Press, v. 100, n. 4, p. 838–861, 1990. ISSN 00141704, 1539297X.

BINMORE, Ken G. **Natural Justice**. [S.l.]: Oxford University Press, 2005. (Oxford scholarship online. Economics and finance module). ISBN 9780195178111.

BOBBIO, Norberto. **Teoria da norma jurídica**. Tradução: Ariani Bueno Sudatti e Fernando Pavan Baptista. 2. ed. São Paulo: EdiPRO, 2003.

BOELLA, Guido; LESMO, Leonardo. Deliberate Normative Agents. *In*: **Social Order in Multiagent Systems**. Edição: Rosaria Conte e Chrysanthos Dellarocas. Boston, MA: Springer US, mai. 2001. P. 85–110. ISBN 978-1-4613-5612-7.

BOELLA, Guido; TORRE, Leendert van der. Norm negotiation in multiagent systems. **International Journal of Cooperative Information Systems (IJCIS)**, v. 16, n. 2, 2007.

BOELLA, Guido; TORRE, Leendert van der. Regulative and Constitutive Norms in Normative Multiagent Systems. *In: PROCEEDINGS of the Ninth International Conference on Principles of Knowledge Representation and Reasoning*. Whistler, British Columbia, Canada: AAAI Press, 2004. (KR'04), p. 255–265.

BOELLA, Guido; TORRE, Leendert van der. Substantive and procedural norms in normative multiagent systems. **Journal of Applied Logic**, v. 6, n. 2, p. 152–171, 2008. Selected papers from the 8th International Workshop on Deontic Logic in Computer Science. ISSN 1570-8683.

BOELLA, Guido; TORRE, Leendert van der; VERHAGEN, Harko. Introduction to Normative Multiagent Systems. **Computation and Mathematical Organizational Theory, Special issue on Normative Multiagent Systems**, Kluwer Academic Publishers, USA, v. 12, n. 2-3, p. 71–79, out. 2006. ISSN 1381-298X.

BOELLA, Guido; TORRE, Leendert van der; VERHAGEN, Harko. Ten Challenges for Normative Multiagent Systems. *In: INTERNATIONAL Workshop on Programming Multi-Agent Systems*. [S.l.: s.n.], 2008.

BOELLA, Guido; TOSATTO, Silvano Colombo; GARCEZ, Artur D'Avila; GENOVESE, Valerio; PEROTTI, Alan; TORRE, Leendert van der. Learning and Reasoning about Norms Using Neural-Symbolic Systems. *In: PROCEEDINGS of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*. Valencia, Spain: International Foundation for Autonomous Agents e Multiagent Systems, 2012. (AAMAS '12), p. 1023–1030.

BOISSIER, Olivier; BORDINI, Rafael H.; HÜBNER, Jomi F.; RICCI, Alessandro; SANTI, Andrea. Multi-agent oriented programming with JaCaMo. **Science of Computer Programming**, v. 78, n. 6, p. 747–761, 2013. ISSN 0167-6423.

BRENNAN, Geoffrey; ERIKSSON, Lina; GOODIN, Robert E.; SOUTHWOOD, Nicholas. Explaining Norms. Oxford University Press, v. 31, n. 1, 2013.

BRITO, Maiquel de; HÜBNER, Jomi F.; BOISSIER, Olivier. Coupling Regulative and Constitutive Dimensions in Situated Artificial Institutions. *In: ROVATSOS, Michael; VOUIROS, George; JULIAN, Vicente (Ed.). Multi-Agent Systems and Agreement Technologies*. Cham: Springer International Publishing, 2016. P. 318–334.

BULLING, Nils; DASTANI, Mehdi. Norm-based mechanism design. **Artificial Intelligence**, v. 239, p. 97–142, 2016. ISSN 0004-3702.

CARMONA, Guilherme. **Existence and Stability of Nash Equilibrium**. [S.l.]: World Scientific, 2013. ISBN 9789814390651.

CHRISTELIS, George; ROVATSOS, Michael. Automated Norm Synthesis in an Agent-based Planning Environment. *In*: PROCEEDINGS of The 8th International Conference on Autonomous Agents and Multiagent Systems. Budapest, Hungary: International Foundation for Autonomous Agents e Multiagent Systems, 2009. v. 1. (AAMAS '09), p. 161–168.

CHRISTELIS, George Dimitri; ROVATSOS, Michael; PETRICK, Ronald P. A. Exploiting domain knowledge to improve norm synthesis. *In*: HOEK, Wiebe van der; KAMINKA, Gal A.; LESPÉRANCE, Yves; LUCK, Michael; SEN, Sandip (Ed.). **9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, May 10-14, 2010, Volume 1-3**. [S.l.]: IFAAMAS, 2010. P. 831–838.

CRIADO, Natalia; ARGENTE, Estefania; BOTTI, Vicent. Open issues for normative multi-agent systems. **AI Commun.**, v. 24, p. 233–264, jan. 2011.

DELL'ANNA, Davide; ALECHINA, Natasha; DALPIAZ, Fabiano; DASTANI, Mehdi; LÖFFLER, Maarten; LOGAN, Brian. The Complexity of Norm Synthesis and Revision. *In*: AJMERI, Nirav; MORRIS MARTIN, Andrea; SAVARIMUTHU, Bastin Tony Roy (Ed.). **Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XV**. Cham: Springer International Publishing, 2022. P. 38–53.

DERAKHSHAN, Farnaz. **Norm Enforcement in Normative Multiagent Systems: Design, Implementation and Development**. [S.l.: s.n.], mar. 2010. ISBN 9783639246117.

DIGNUM, Virginia; VÁZQUEZ-SALCEDA, Javier; DIGNUM, Frank. A Model of Almost Everything: Norms, Structure and Ontologies in Agent Organizations. **Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2004**, v. 3, jul. 2004.

DIMISHKOVSKA, Ana. Deontic Logic and Legal Rules. *In: Encyclopedia of the Philosophy of Law and Social Philosophy*. Edição: Mortimer Sellers e Stephan Kirste. Dordrecht: Springer Netherlands, 2017. P. 1–7. ISBN 978-94-007-6730-0.

DURKHEIM, Emile. **Rules of Sociological Method And selected texts on sociology and its method**. Edição: Steven Lukes. [S.l.]: Free Press, 1982. (Contemporary social theory). ISBN 9780029079409.

EGGERT, Max A.; FALZON, Wendy. **Resolving Conflict Pocketbook**. [S.l.]: Management Pocketbooks, 2004. (Management Pocketbooks Series). ISBN 9781903776063.

EGGERTSSON, Thrainn; EGGERTSSON, Rainn. **Norms in economics: With special reference to economic development**. [S.l.]: Max Planck Inst. for Research into Economic Systems, jan. 1999. P. 76–104. (Diskussionsbeiträge).

EKENBERG, Love. Detecting conflicts in multi-agent systems. *In: ZHANG, Chengqi; LUKOSE, Dickson (Ed.). Multi-Agent Systems: Methodologies and Applications*. Springer-Verlag: Lecture Notes in Artificial Intelligence 1286, 1997.

ELLICKSON, Robert C. The Evolution of Social Norms: A Perspective from the Legal Academy. **Program for Studies in Law, Economics and Public Policy**, 230 jul. 1999.

ELSTER, Jon. Rationality and Social Norms. *In: FENSTAD, Jens Erik; FROLOV, Ivan T.; HILPINEN, Risto (Ed.). Logic, Methodology and Philosophy of Science VIII*. [S.l.]: Elsevier, 1989. v. 126. (Studies in Logic and the Foundations of Mathematics). P. 531–552.

EPSTEIN, Joshua M. Learning to Be Thoughtless: Social Norms and Individual Computation. **Comput. Econ.**, Kluwer Academic Publishers, USA, v. 18, n. 1, p. 9–24, ago. 2001. ISSN 0927-7099.

FIGUEIREDO, Karen da Silva; SILVA, Viviane Torres da. Norm-ML - A Modeling Language to Model Norms. *In: PROCEEDINGS of the 3rd International Conference on Agents and Artificial Intelligence*. [S.l.: s.n.], jan. 2011. P. 232–237.

FINNEMORE, Martha; SIKKINK, Kathryn. International Norm Dynamics and Political Change. **International Organization**, [MIT Press, University of Wisconsin Press,

Cambridge University Press, International Organization Foundation], v. 52, n. 4, p. 887–917, 1998. ISSN 00208183, 15315088.

FITOUSSI, David; TENNENHOLTZ, Moshe. Choosing social laws for multi-agent systems: Minimality and simplicity. **Artificial Intelligence**, v. 119, n. 1, p. 61–101, 2000. ISSN 0004-3702.

FITOUSSI, David; TENNENHOLTZ, Moshe. Minimal Social Laws. *In*: PROCEEDINGS of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence. [S.l.: s.n.], 1998. P. 26–31.

FONSECA, Elisa. Normas Processuais e Normas Substantivas: a Primazia das Normas de Jus Cogens e o Entendimento da Corte Internacional de Justiça. **Revista de Teorias da Justiça, da Decisão e da Argumentação Jurídica**, v. 1, p. 95, dez. 2015.

FRANTZ, Christopher; PURVIS, Martin K.; NOWOSTAWSKI, Mariusz; SAVARIMUTHU, Bastin Tony Roy. Modelling Institutions Using Dynamic Deontics. *In*: BALKE, Tina; DIGNUM, Frank; RIEMSDIJK, M. Birna van; CHOPRA, Amit K. (Ed.). **Coordination, Organizations, Institutions, and Norms in Agent Systems IX**. Cham: Springer International Publishing, 2014. P. 211–233.

FRANTZ, Christopher K.; PIGOZZ, Gabriella. Modeling Norm Dynamics in Multi-Agent Systems. [S.l.], ago. 2018.

GABBAY, Dov; HORTY, John; PARENT, Xavier; RON VAN DER, Meyden; LEENDERT VAN DER, Torre. **Handbook of Deontic Logic and Normative Systems**. [S.l.]: College Publications, 2013. ISBN 9781848901322.

GARCEZ, Artur; ZAVERUCHA, Gerson. The Connectionist Inductive Learning and Logic Programming System. **Applied Intelligence**, v. 11, p. 59–77, jul. 1999.

GARCEZ, Artur S. d'Avila; BRODA, Krysia B.; GABBAY, Dov M. **Neural-Symbolic Learning Systems: Foundations and Applications**. [S.l.]: Springer London, 2002. (Perspectives in Neural Computing). ISBN 9781852335120.

GARCIA-CAMINO, Andres; NORIEGA, Pablo; RODRIGUEZ-AGUILAR, Juan A. Implementing Norms in Electronic Institutions. *In*: PROCEEDINGS of the Fourth

International Joint Conference on Autonomous Agents and Multiagent Systems. The Netherlands: Association for Computing Machinery, 2005. (AAMAS '05), p. 667–673.

GEERTZ, Clifford. Thick Description: Towards an Interpretive Theory of Culture. *In*: THE Interpretation of Cultures. New York: Basic Books, 1973. cap. 1, p. 3–32.

GHALLAB, Malik; NAU, Dana; TRAVERSO, Paolo. **Automated Planning: Theory and Practice**. Amsterdam: Morgan Kaufmann, 2004. (The Morgan Kaufmann Series in Artificial Intelligence). ISBN 978-1-55860-856-6.

GIRET, Adriana; BOTTI, Vicente. MULTI AGENT SYSTEM GOALS. Citeseer, 2004.

GOVERNATORI, Guido; ROTOLO, Antonino. **Is Free Choice Permission Admissible in Classical Deontic Logic?** [*S.l.: s.n.*], 2019. arXiv: 1905.07696 [cs.LG].

GURNEY, Nikolos; MARSELLA, Stacy; USTUN, Volkan; PYNADATH, David V. Operationalizing Theories of Theory of Mind: A Survey. *In*: GURNEY, Nikolos; SUKTHANKAR, Gita (Ed.). **Computational Theory of Mind for Human-Machine Teams**. Cham: Springer Nature Switzerland, 2022. P. 3–20.

HECHTER, Michael. Norms in the evolution of social order. **Social Research**, New School University, v. 85, n. 1, p. 23–51, 2018. ISSN 0037-783X.

HECHTER, Michael; OPP, Karl-Dieter. **Social norms**. New York, US: Russell Sage Foundation, 2001. ISBN 0-87154-354-0.

HINDRIKS, Koen V.; BOER, Frank S. de; HOEK, Wiebe van der; MEYER, John-Jules Ch. Agent Programming with Declarative Goals. *In*: CASTELFRANCHI, Cristiano; LESPÉRANCE, Yves (Ed.). **Intelligent Agents VII Agent Theories Architectures and Languages**. Berlin, Heidelberg: Springer Berlin Heidelberg, jan. 2001. P. 228–243.

HOEK, Wiebe van der; ROBERTS, Mark; MICHAEL, Wooldridge. Social laws in alternating time: effectiveness, feasibility, and synthesis. **Synthese**, v. 156, n. 1, p. 1–19, 2007.

HOLLANDER, Christopher D.; WU, Annie S. The Current State of Normative Agent-Based Systems. **Journal of Artificial Societies and Social Simulation**, v. 14, n. 2, 2011.

- HORNE, Christine. Explaining Norm Enforcement. **Rationality and Society**, v. 19, n. 2, p. 139–170, 2007.
- HÜBNER, Jomi F.; BOISSIER, Olivier; BORDINI, Rafael H. A Normative Programming Language for Multi-Agent Organisations. Kluwer Academic Publishers, USA, v. 62, 1–2, p. 27–53, jun. 2011. ISSN 1012-2443.
- JENNINGS, Nicholas R. On agent-based software engineering. **Artificial Intelligence**, v. 117, n. 2, p. 277–296, 2000. ISSN 0004-3702.
- KATOCH, Sourabh; CHAUHAN, Sumit Singh; KUMAR, Vijay. A review on genetic algorithm: past, present, and future. **Multimedia Tools Appl.**, Kluwer Academic Publishers, USA, v. 80, n. 5, p. 8091–8126, fev. 2021. ISSN 1380-7501.
- KHAN, Naushad. Critical Review of Sampling Techniques in the Research Process in the World. SSRN, 2020.
- LAMSWEERDE, Axel van. Goal-oriented requirements engineering: a guided tour. *In*: PROCEEDINGS Fifth IEEE International Symposium on Requirements Engineering. [S.l.: s.n.], 2001. P. 249–262.
- LAROUSSINIE, François; MARKEY, Nicolas. Augmenting ATL with strategy contexts. **Information and Computation**, v. 245, p. 98–123, 2015. ISSN 0890-5401.
- LEWIS, David Kellogg. **Convention: A Philosophical Study**. Cambridge: Harvard University Press, 1969. ISBN 9780631232575.
- LÓPEZ, Fabiola López y; LUCK, Michael; D'INVERNO, Mark. Constraining autonomy through norms. *In*: (AAMAS '02), p. 674–681.
- MAHMOUD, Moamin; AHMAD, Mohd; YUSOFF, Mohd; MUSTAPHA, Aida. A Review of Norms and Normative Multiagent Systems. **TheScientificWorldJournal**, v. 2014, p. 684587, jul. 2014.
- MAKINSON, David; TORRE, Leendert van der. Input/Output Logics. **Journal of Philosophical Logic**, Springer Netherlands, v. 29, n. 4, p. 383–408, 2000.

MCKENZIE, George. Complementarity, Substitutability, and Independence. **Oxford Economic Papers**, Oxford University Press, v. 29, n. 3, p. 430–441, 1977. ISSN 00307653, 14643812.

MONTES, Nieves; SIERRA, Carles. Synthesis and Properties of Optimally Value-Aligned Normative Systems. **Journal of Artificial Intelligence Research**, AI Access Foundation, El Segundo, CA, USA, v. 74, set. 2022. ISSN 1076-9757.

MORALES, Javier; LOPEZ-SANCHEZ, Maire; RODRIGUEZ-AGUILAR, Juan A.; WOOLDRIDGE, Michael; VASCONCELOS, Wamberto. Automated Synthesis of Normative Systems. *In*: PROC. of AAMAS 2013. St. Paul, MN, USA: [s.n.], 2013. P. 483–490.

MORALES, Javier; LÓPEZ-SÁNCHEZ, Maite; RODRÍGUEZ-AGUILAR, Juan A.; WOOLDRIDGE, Michael; VASCONCELOS, Wamberto. Minimality and simplicity in the on-line automated synthesis of normative systems. *In*: AAMAS. [S.l.: s.n.], 2014.

MORALES, Javier; LÓPEZ-SÁNCHEZ, Maite; RODRIGUEZ-AGUILAR, Juan A.; WOOLDRIDGE, Michael; VASCONCELOS, Wamberto. Synthesising Liberal Normative Systems. *In*: (AAMAS '15), p. 433–441.

MORALES, Javier; WOOLDRIDGE, Michael; RODRÍGUEZ-AGUILAR, Juan; LÓPEZ-SÁNCHEZ, Maite. Synthesising Evolutionarily Stable Normative Systems. **arXiv**, v. 32, set. 2017. eprint: 1710.00709 (cs.MA).

MORRIS-MARTIN, Andreea; BOS-DE VOS, Marina; PADGET, Julian. **A Norm Emergence Framework for Normative MAS - Position Paper**. [S.l.: s.n.], 2020. eprint: 2004.02575.

MORRIS-MARTIN, Andreea; DE VOS, Marina; PADGET, Julian. Norm Emergence in Multiagent Systems: A Viewpoint Paper. *In*: PROCEEDINGS of the 19th International Conference on Autonomous Agents and MultiAgent Systems. Auckland, New Zealand: International Foundation for Autonomous Agents e Multiagent Systems, 2020. (AAMAS '20), p. 2152–2154.

NEBEL, Bernhard. On the Compilability and Expressive Power of Propositional Planning Formalisms. **Journal of Artificial Intelligence Research**, AI Access Foundation, v. 12, p. 271–315, mai. 2000.

OLIVEIRA, Luccas Felipe; SILVESTRE, Eduardo Augusto; SILVA, Viviane Torres da. Um Compilador Para Definição e Geração de Normas em Sistemas Multiagentes. **Revista Inova Ciência e Tecnologia / Innovative Science and Technology Journal**, v. 3, n. 2, p. 56–63, dez. 2017.

OPP, Karl-Dieter. Norms. *In: International Encyclopedia of the Social Behavioral Sciences*. Edição: James D. Wright. 2. ed. [S.l.: s.n.], dez. 2015. v. 17, p. 5–10. ISBN 9780080430768.

PAVLOVIĆ, Duan; ĐURAKOVIĆ, Stevo. The Three Versions of the Ultimatum Game. **Anali Hrvatskog politolokog drutva**, 2021.

PELLIER, Damien; FIORINO, Humbert. PDDL4J: a planning domain description library for java. **Journal of Experimental and Theoretical Artificial Intelligence**, Taylor & Francis, 30(1), p. 143–176, 2018.

PERREAU DE PINNINCK, Adrian; SIERRA, Carles; SCHORLEMMER, Marco. Distributed Norm Enforcement: Ostracism in Open Multi-Agent Systems. *In: CASANOVAS, Pompeu; SARTOR, Giovanni; CASELLAS, Núria; RUBINO, Rossella (Ed.). Computable Models of the Law*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. P. 275–290.

POSNER, Eric A. **Law and Social Norms**. [S.l.]: Harvard University Press, 2009. ISBN 9780674042308.

PRATT, Ian. Closed World Assumptions. *In: ARTIFICIAL Intelligence*. London: Macmillan Education UK, 1994. P. 65–84. ISBN 978-1-349-13277-5.

RAYMOND, Gregory A. Problems and Prospects in the Study of International Norms. **Mershon International Studies Review**, International Studies Association, v. 41, n. 2, p. 205–245, 1997. ISSN 10791760.

RESMERITA, Stefan; HEYMANN, Michael. Conflict resolution in multi-agent systems. *In: 42ND IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*. [S.l.: s.n.], jan. 2003. v. 3, 2537–2542 vol.3.

RIAD, Maha; GOLPAYEGANI, Fatemeh. Run-Time Norms Synthesis in Multi-objective Multi-agent Systems. *In: THEODOROU, Andreas; NIEVES, Juan Carlos; DE VOS, Marina (Ed.). Coordination, Organizations, Institutions, Norms, and*

Ethics for Governance of Multi-Agent Systems XIV. Cham: Springer International Publishing, 2022. P. 78–93.

RIEMSDIJK, M. Birna van; DASTANI, Mehdi; WINIKOFF, Michael. Goals in Agent Systems: A Unifying Framework. *In: (AAMAS '08)*, p. 713–720.

ROSEN, Kenneth H. **Discrete Mathematics and Its Applications**. 7th ed. [S.l.]: McGraw-Hill, 2012. (McGraw-Hill higher education). ISBN 9780072930337.

RUSSELL, Stuart; NORVIG, Peter. **Artificial Intelligence: A Modern Approach**. 3. ed. [S.l.]: Prentice Hall, 2010.

SANTOS, Jéssica S.; ZAHN, Jean O.; SILVESTRE, Eduardo A.; SILVA, Viviane T.; VASCONCELOS, Wamberto W. Detection and Resolution of Normative Conflicts in Multi-Agent Systems: A Literature Survey. **Autonomous Agents and Multi-Agent Systems**, Kluwer Academic Publishers, USA, v. 31, n. 6, p. 1236–1282, nov. 2017. ISSN 1387-2532.

SAVARIMUTHU, Bastin Tony Roy; CRANFIELD, Stephen. A categorization of simulation works on norms. *In: BOELLA, Guido; NORIEGA, Pablo; PIGOZZI, Gabriella; VERHAGEN, Harko (Ed.). Normative Multi-Agent Systems*. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2009. v. 9121. (Dagstuhl Seminar Proceedings (DagSemProc)), p. 1–20.

SAVARIMUTHU, Bastin Tony Roy; CRANFIELD, Stephen; PURVIS, Maryam; PURVIS, Martin. Role Model Based Mechanism for Norm Emergence in Artificial Agent Societies. *In: PROC. of COIN 2008*. [S.l.: s.n.], 2008. P. 203–217.

SEARLE, John R. **Speech Acts: An Essay in the Philosophy of Language**. Cambridge, London: Cambridge University Press, 1969.

SEARLE, John R. **The Construction of Social Reality**. [S.l.]: Free Press, 1995. ISBN 9780029280454.

SERRANO, Emilio; ROVATSOS, Michael; BOTÍA, Juan A. An Approach for the Qualitative Analysis of Open Agent Conversations. English. *In: PROCEEDINGS of The Third International Workshop on INFRASTRUCTURES AND TOOLS FOR MULTIAGENT SYSTEMS ITMAS 2012*. [S.l.]: Editorial Universitat Politècnica de València, 2012. P. 79–92.

SHOHAM, Yoav; TENNENHOLTZ, Moshe. On social laws for artificial agent societies: off-line design. **Artificial Intelligence**, v. 73, n. 1, p. 231–252, 1995. ISSN 0004-3702.

SHOHAM, Yoav; TENNENHOLTZ, Moshe. On the Synthesis of Useful Social Laws for Artificial Agent Societies. *In*: PROCEEDINGS of the Tenth National Conference on Artificial Intelligence. San Jose, California: AAAI Press, 1992. (AAAI'92), p. 276–281.

SILVESTRE, Eduardo Augusto; SILVA, Viviane Torres da; SILVA, Oscar da Silva da; OLIVEIRA, Luccas Felipe; RAMOS, Wendler Souza. A Platform for Detection and Resolution of Conflicts Among Multiple Norms in Mas (Multi-agent Systems). **Revista Inova Ciência amp; Tecnologia / Innovative Science amp; Technology Journal**, v. 5, n. 2, p. 74–83, abr. 2019.

SINGH, Amika; SINGH, Munindar. Norm Deviation in Multiagent Systems: A Foundation for Responsible Autonomy. *In*: PROCEEDINGS of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China. [S.l.: s.n.], ago. 2023. P. 289–297.

SINGH, Munindar P. *et al.* The Uses of Norms. *In*: ANDRIGHETTO, Giulia; GOVERNATORI, Guido; NORIEGA, Pablo; TORRE, Leendert W. N. van der (Ed.). **Normative Multi-Agent Systems**. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2013. v. 4. (Dagstuhl Follow-Ups). P. 191–229. ISBN 978-3-939897-51-4.

SMITH, John Maynard. **Evolution and the Theory of Games**. [S.l.]: Cambridge University Press, 1982.

SUTTON, Richard S.; BARTO, Andrew G. **Reinforcement Learning: An Introduction**. Second Edition. [S.l.]: The MIT Press, 2018.

TESSIER, Catherine; CHAUDRON, Laurent; MÜLLER, Heinz-Jürgen (Ed.). **Conflicting Agents: Conflict Management in Multi-Agent Systems**. USA: Kluwer Academic Publishers, 2001. ISBN 0792372107.

USZOK, Andrzej *et al.* KAoS policy and domain services: toward a description-logic approach to policy representation, deconfliction, and enforcement. *In*: PROCEEDINGS POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks. [S.l.: s.n.], jul. 2003. P. 93–96.

VASCONCELOS, Wamberto; KOLLINGBAUM, Martin; NORMAN, Timothy. Normative conflict resolution in multi-agent systems. **Autonomous Agents and Multi-Agent Systems**, v. 19, p. 124–152, out. 2009.

VERHAGEN, Harko; NEUMANN, Martin; SINGH, Munindar. Normative Multiagent Systems: Foundations and History. *In: Handbook of Normative Multiagent Systems*. Edição: Amit Chopra, Leendert van der Torre, Harko Verhagen e Serena eds Villata. [S.l.]: College Publications, ago. 2018. P. 3–25. ISBN 978-1-84890-285-5.

VILLATORO, Daniel; SEN, Sandip; SABATER-MIR, Jordi. Of Social Norms and Sanctioning: A Game Theoretical Overview. **Int. J. Agent Technol. Syst.**, IGI Global, USA, v. 2, n. 1, p. 1–15, jan. 2010. ISSN 1943-0744.

VINDEGAARD, Nina; BENROS, Michael Eriksen. COVID-19 pandemic and mental health consequences: Systematic review of the current evidence. **Brain, Behavior, and Immunity**, v. 89, p. 531–542, 2020.

WALKER, Adam; WOOLDRIDGE, Michael. Understanding the Emergence of Conventions in Multi-Agent Systems. *In: PROCEEDINGS of the First International Conference on Multiagent Systems*. [S.l.]: MIT Press, 1995. P. 384–389.

WATKINS, Christopher; DAYAN, Peter. Q-learning. **Machine Learning**, v. 8, p. 279–292, 1992.

WEBER, Max. **Economy and Society: An Outline of Interpretive Sociology**. Edição: Guenther Roth e Claus Wittich. [S.l.]: University of California Press, 1978. (Economy and Society: An Outline of Interpretive Sociology). ISBN 9780520035003.

WINIKOFF, Michael; PADGHAM, Lin; HARLAND, James; THANGARAJAH, John. Declarative & Procedural Goals in Intelligent Agent Systems, mar. 2002.