Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Geração Automática de Modelos em Lógicas Modais: Implementação

Daniella Albuquerque dos Angelos

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientadora
Prof.ª Dr.ª Cláudia Nalon

Brasília
2016

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Coordenador: Prof. Dr. Rodrigo Bonifácio de Almeida

Banca examinadora composta por:

Prof.ª Dr.ª Cláudia Nalon (Orientadora) — CIC/UnB
Prof. Dr. Professor I — CIC/UnB
Prof. Dr. Professor II — CIC/UnB

CIP — Catalogação Internacional na Publicação

dos Angelos, Daniella Albuquerque.

Geração Automática de Modelos em Lógicas Modais: Implementação / Daniella Albuquerque dos Angelos. Brasília : UnB, 2016.
41 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2016.

1. lógica modal, 2. método de resolução, 3. geração automática

CDU 004.4

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília–DF — Brasil

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Geração Automática de Modelos em Lógicas Modais: Implementação

Daniella Albuquerque dos Angelos

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof.ª Dr.ª Cláudia Nalon (Orientadora)
CIC/UnB

Prof. Dr. Professor I
CIC/UnB

Prof. Dr. Professor II
CIC/UnB

Prof. Dr. Rodrigo Bonifácio de Almeida
Coordenador do Bacharelado em Ciência da Computação

Brasília, 6 de junho de 2016

Dedicatória

Dedico a....

Agradecimentos

Agradeço a....

Abstract

A geração automática de modelos é uma técnica complementar àquela da prova de teoremas e realizada em paralelo com a avaliação experimental. Se o provador de teoremas falha em encontrar uma prova, o modelo automaticamente extraído serve como testemunha da impossibilidade de se encontrar tal prova. Além disso, com a possibilidade de uso combinado de estratégias, a não obtenção de um modelo serve como testemunha da incompletude de tal combinação, sendo portanto ferramenta de suporte ao entendimento teórico. Este trabalho tem como objetivo a implementação de um gerador automático de modelos para a lógica modal proposicional K, bem como a realização de testes e avaliação dos algoritmos implementados.

Palavras-chave: lógica modal, método de resolução, geração automática

Abstract

Automatic generation of models is a complementary technique to that of proof theorems and held in parallel with the experimental evaluation. If the program fails to find the proof, the automatically extracted model serves as witness of the impossibility of finding such proof. Furthermore, with the possibility of combining some strategies, not finding a model serves as witness of the incompleteness of such combination, thus being a tool of theoretical support. This work aims to implement an automatic generator models for propositional modal logic K, as well as testing and evaluation of the implemented algorithms.

Keywords: modal logics, resolution method, automated generation

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Linguagens lógicas podem ser utilizadas para verificar propriedades de um sistema complexo, expressas através de possibilidade, crença, probabilidade entre outras modalidades [4, 9, 10, 15]. Diferentes modalidades definem diferentes lógicas modais.

Um teorema é uma fórmula que pode ser provada com base nos axiomas e no conjunto de regras de inferência que definem um determinado cálculo. Uma vez que um sistema tenha sido especificado na linguagem lógica por meio de fórmulas, é possível utilizar métodos de provas para verificá-los.

Em [12] são apresentados cálculos baseados em resolução para quinze famílias de lógicas modais. As regras de inferência baseiam-se nas propriedades dos modelos subjacentes, ao invés de se fixar na forma dos axiomas. Deste modo, obtém-se um procedimento uniforme para se lidar com várias lógicas. Uma das intenções de tal proposta é justamente prover técnicas que facilitem o projeto de cálculos combinados tanto para fusões de lógicas quanto para lógicas em que interações fossem permitidas. Interações são, em geral, caracterizadas por axiomas contendo operadores das diferentes lógicas componentes.

Grande parte dos provadores para lógicas modais são, porém, baseados em tradução, o que acaba, por vezes, se tornando inconveniente ao usuário. Além disso, dada a natureza das aplicações descritas com o auxílio destas lógicas, é normal o uso da combinação de diferentes linguagens modais. A combinação de linguagens, todavia, pode acarretar no aumento da complexidade ou mesmo na indecidibilidade do problema de satisfatibilidade na lógica resultante [7]. Portanto, é importante o desenvolvimento de técnicas que possam ser utilizadas de modo uniforme na combinação de métodos de prova para lógicas obtidas a partir de fusões e/ou em linguagens que permitam interações.

Os métodos apresentados em [12] e em trabalhos anteriores têm esta característica de uniformidade, mas carecem de refinamentos a fim de permitir a construção de ferramentas que possam ser, de fato, utilizadas na verificação formal de sistemas complexos.

O problema básico de satisfatibilidade da lógica modal K é PSPACE [11]. Além disso, sabe-se que métodos de prova para lógica proposicional são intratáveis [3]. Em [13], uma estratégia de busca por provas, baseada em níveis modais foi apresentada. A implementação e a avaliação desta estratégia, comparando resultados de eficiência com provadores no estado da arte, mostram que a separação em níveis modais pode auxiliar na melhoria do tempo para a obtenção de provas [14].

A geração automática de modelos é complementar àquela da prova de teoremas e realizada em paralelo com a avaliação experimental. Se o provador de teoremas falha em

encontrar uma prova, o modelo automaticamente extraído serve como testemunha da impossibilidade de se encontrar tal prova. Além disso, com a possibilidade de uso combinado de estratégias, a não obtenção de um modelo serve como testemunha da incompletude de tal combinação, sendo portanto ferramenta de suporte ao entendimento teórico.

O objetivo específico deste trabalho consiste na implementação de um gerador automático de modelos para a lógica modal proposicional K. A entrada será o conjunto de cláusulas fornecido pelo provador implementado em [14]. A saída será a declaração da inexistência de um modelo, no caso do conjunto de cláusulas ser insatisfatível, ou a apresentação formal de um modelo que testemunhe a satisfatibilidade do conjunto de cláusulas.

Os dados obtidos na experimentação prática serão analisados, em conformidade com o enfoque corrente, de forma comparativa a partir de *benchmarks* estabelecidos e especialmente preparados para tal fim. Tais resultados são apresentados no Capítulo 4.

Capítulo 2

Revisão Teórica

Neste capítulo, introduziremos brevemente a base do pensamento modal.

A lógica modal é determinada semanticamente pela avaliação de necessidade e possibilidade de uma proposição, levando em consideração diferentes possíveis contextos (ou *mundos*). Na literatura, representações análogas à de necessidade e possibilidade são utilizadas para a avaliação de estados epistemológicos, como por exemplo, conhecimento e crença, respectivamente [5]. Basicamente, uma proposição é *necessária* se ocorre em todos os mundos possíveis e *possível* se ocorre em algum mundo [2]. A ideia é que uma ação ou objeto pode possuir valorações distintas em mundos diferentes, mas qualquer objeto que possuir valoração verdadeira em todos os mundos possíveis é necessária, enquanto a que ocorre em pelo menos um mundo, é possível.

Os operadores utilizados são os mesmos já conhecidos da lógica proposicional ($\wedge$, $\vee$, $\rightarrow$ etc) com a inclusão de dois novos: $\boxed{a}$ e $\diamondsuit_a$. Uma sentença da forma $\boxed{a}\varphi$ (*necessariamente* φ) é verdadeira se, e somente se, φ é uma proposição verdadeira em todos os mundos possíveis, de acordo com o agente a; já uma sentença da forma $\diamondsuit_a\varphi$ (*possivelmente* φ) é verdadeira no caso em que φ é uma proposição verdadeira em algum mundo existente, também de acordo com o agente a.

Uma ilustração válida para a lógica modal, é pensar em uma coleção de possíveis mundos, incluindo nosso próprio, o mundo real, onde sentenças da linguagem são possivelmente verdadeiras ou falsas. O propósito principal da lógica modal é modelar a ocorrência verdadeira destas sentenças. Esta lógica pode ser expressa de diferentes formas e contar com diferentes axiomas. sistemas, isto será explorado com mais detalhes

2.1 Linguagens Lógicas

2.2 Formalização da Linguagem Modal K_n

2.2.1 Sintaxe

Esta seção é dedicada a trazer o básico dos conceitos de sintaxe da linguagem da lógica modal. As definições formais apresentadas podem ser úteis ao entendimento.

A linguagem proposicional clássica é formada por um conjunto enumerável de sentenças *atômicas* (ou símbolos proposicionais):

$$\mathcal{P} = \{p, q, r, \ldots\} \tag{2.1}$$

Estas são as sentenças mais simples possível.

Já as *sentenças moleculares* (não-atômicas) são formadas por meio das *operações sintáticas*, ou *operadores lógicos* de negação ($\neg$), de disjunção ($\vee$) e de necessidade ($\boxed{a}$, para todo $a \in \mathcal{A} = \{1, \ldots, n\}$, $n \in \mathbb{N}$). Observe que $\neg$ e $\boxed{a}$ são operadores unários e $\vee$ é um operador binário. Além disso, um conjunto de sinais de pontuação $\{(,)\}$ é usado para evitar ambiguidade na leitura de sentenças.

Podemos então definir uma fórmula como uma combinação finita de sentenças moleculares. A *Linguagem Lógica Modal* é, portanto, equivalente ao seu conjunto de *fórmulas bem formadas*, definido a seguir.

Definição 1 O conjunto de *fórmulas bem formadas da linguagem modal*, denotado por FBF_K é definido indutivamente como segue:

1. Os símbolos proposicionais estão em FBF_K;
2. Se $\varphi \in FBF_K$, então $\neg\varphi$ e $\boxed{a}\varphi$, $a \in \mathcal{A}$, estão em FBF_K; e
3. Se φ e ψ estão em FBF_K, então $(\varphi \vee \psi)$ está em FBF_K.

Outros operadores lógicos podem ser introduzidos como abreviações para fórmulas construídas a partir dos operadores já definidos, como usual. Em particular, neste trabalho as seguintes abreviações serão utilizadas: $(\varphi \wedge \psi) = \neg(\neg\varphi \vee \neg\psi)$ (conjunção), $(\varphi \Rightarrow \psi) = (\neg\varphi \vee \psi)$ (implicação), $\diamondsuit_a\varphi = \neg\boxed{a}\neg\varphi$ (possibilidade), $\mathbf{false} = (\varphi \wedge \neg\varphi)$ (*falsum*) e $\mathbf{true} = \neg\mathbf{false}$ (*verum*). A precedência dos operadores é dada na seguinte ordem (onde $a \in \mathcal{A}$):

1. $\{\neg, \boxed{a}, \diamondsuit_a\}$
2. $\{\wedge\}$
3. $\{\vee\}$
4. $\{\rightarrow\}$

Quando não houver ambiguidade na leitura de uma fórmula, parênteses podem ser omitidos.

Lógicas que envolvem vários agentes na lógica modal utilizam n operadores, com $n \in \mathbb{N}$, como descrito acima, são chamadas de *lógicas multimodais*. A lógica modal com um único agente, onde apenas temos os operadores $\diamondsuit_1$ e $\boxed{1}$ (ou simplesmente $\diamondsuit$ e $\Box$), isto é, onde $\mathcal{A} = \{1\}$, bem como a lógica proposicional clássica, onde $\mathcal{A} = \{\}$, são, portanto, casos especiais da lógica multimodal.

O *nível modal* de uma fórmula é o número máximo de ocorrências de operadores modais sob o qual a fórmula ocorre. Por exemplo, em $\boxed{a}\diamondsuit_a p$, o nível modal de p é 2.

É importante definir algumas convenções para minimizar dúvidas que poderiam surgir em algumas expressões. Expressões da forma $A_1 \wedge \ldots \wedge A_n$ e $A_1 \vee \ldots \vee A_n$ representam conjunções e disjunções arbitrárias mas não especificadas das sentenças $A_1, \ldots, A_n$. O objetivo é deixar claro que tanto $\wedge$ como $\vee$ obedecem as regras de associatividade lógica.

2.2.2 Semântica

A ideia geral do significado de fórmulas na lógica proposicional modal pode ser entendida, de forma simples, em termos de uma coleção de mundos possíveis, um conjunto de atribuições de valores booleanos para cada símbolo proposicional em cada mundo e relações de acessibilidade, que são definidas com base em um conjunto de agentes.

Uma das estruturas mais estudadas em lógica modal corresponde à estrutura de Kripke.

Definição 2 Uma **estrutura de Kripke**, para o conjunto de símbolos proposicionais $\mathcal{P}$ e o conjunto de agentes $\mathcal{A} = \{1, \ldots, n\}$, é uma tupla $\mathcal{M} = (W, w_0, R_1, \ldots, R_n, \pi)$, onde: W é um conjunto não-vazio de mundos possíveis; $w_0 \in W$ é a raiz do modelo; $\forall a \in \mathcal{A}, R_a \subseteq W \times W$ e $\pi : W \times \mathcal{P} \longrightarrow \{false, true\}$

A partir da definição de estrutura de Kripke, podemos definir a satisfatibilidade e a validade de uma fórmula.

Definição 3 Seja $\mathcal{M} = (W, w_0, R_1, \ldots, R_n, \pi)$ uma estrutura de Kripke sobre $\mathcal{P}$ e $\mathcal{A}$, e considere $w \in W$, $p \in \mathcal{P}$ e φ e ψ fórmulas bem formadas. A **relação de satisfatibilidade**, denotada por $\models^{\mathcal{M}}_{w} \varphi$, entre o mundo w e uma fórmula φ no modelo $\mathcal{M}$, é definida indutivamente como se segue:

- $\models^{\mathcal{M}}_{w} p$ se e somente se, $\pi(w, p) = true$;
- $\models^{\mathcal{M}}_{w} \neg\varphi$ se e somente se, $\not\models^{\mathcal{M}}_{w} \varphi$
- $\models^{\mathcal{M}}_{w} \varphi \vee \psi$ se e somente se, $\models^{\mathcal{M}}_{w} \varphi$ ou $\models^{\mathcal{M}}_{w} \psi$
- $\models^{\mathcal{M}}_{w} \boxed{a}\varphi$ se e somente se, $\forall\, t \in W$ com $a \in \mathcal{A}$ e $(w, t) \in R_a$, tem-se que $\models^{\mathcal{M}}_{t} \varphi$.

Definição 4 Seja $\varphi \in FBF_K$, dizemos que φ é **satisfatível** se existe um modelo $\mathcal{M}$ tal que $\models^{\mathcal{M}}_{w_0} \varphi$.

Definição 5 Seja $\varphi \in FBF_K$, dizemos que φ é **válida** se para todo modelo $\mathcal{M}$ temos que $\models^{\mathcal{M}}_{w_0} \varphi$.

Nós denotamos por $\mathsf{profundidade}(\mathsf{w})$ o tamanho do caminho único entre w_0 e w através da união das relações de acessibilidade em $\mathcal{M}$. Nós chamamos de *camada modal* a classe de equivalência entre os mundos de mesma profundidade em um modelo.

Podemos notar que o problema de verificar a satisfatibilidade de uma fórmula φ em w_0 pode ser reduzido ao problema de se checar a satisfatibilidade de todas as suas subfórmulas que ocorrem na camada modal de um modelo correspondente ao nível modal onde estas subfórmulas ocorrem (ver [1]).

Observe que, como dito anteriormente, a Lógica Modal K_n inclui a Lógica Proposicional e a Lógica Modal com um único agente. Para tal, basta tomar $W = \{w_0\}$, com $\mathcal{A} = \{\}$, no caso da Lógica Proposicional, e, no caso da Lógica Modal com um único agente, W como um conjunto não-vazio de mundos possíveis, com $\mathcal{A} = \{1\}$.

2.3 Métodos de Prova

Informalmente, uma prova é um argumento que convence. Formalmente, uma prova é uma fórmula, é um objeto finito construído de acordo com regras de sintaxe fixadas que fazem referência unicamente à estrutura das fómulas e não ao seu significado pretendido. As regras sintáticas que definem as provas especificam o que chamamos de *procedimento de prova*. Um procedimento de prova é *sólido* para uma lógica em particular se para qualquer fórmula que possuir uma prova, esta fórmula se trata de uma fórmula válida nesta lógica. Um procedimento de prova é *completo* para uma lógica se toda fórmula válida daquela lógica possui uma prova. Dessa forma, um procesimento sólido e completo de provas nos permite constuir "testemunhas", chamadas de provas, de que determinada fórmula é válida.

Existem inúmeros tipos de procedimentos de provas. Genericamente, podemos dividi-los em duas categorias: *sintéticos* e *analíticos*. Os termos são sugestivos, mas não extremamente precisos. Uma procedimento de prova analítico decompõe uma fórmula em partes mais simples. Um procedimento de prova sintético, por outro lado, constrói uma prova até a fórmula que se deseja obter a demonstração. O primeiro tende a ser mais facilmente aplicado, já que o espaço em que se trabalha é limitado: nunca olha-se para muito longe do escopo onde se encontra a fórmula que se está tentando prova. Já o segundo costuma ser mais utilizado por produzir provas especialmente elegantes.

O exemplo mais comum de procedimento sintético é um *sistema de axiomas*. Algumas fórmulas são tomadas como axiomas, uma prova começa com esses axiomas e, usando regras de inferência definidas, que produzem novas fórmulas, é construído uma sequência de fórmulas que finalmente termina na fórmula que se está tentando provar.

Sistemas Tableaux são um dos mais comuns procedimentos de prova analítica. Para provar uma fórmula, precisamos inicialmente negá-la, analisando as consequências de ser tomada tal ação, utilizando uma estrutura em forma de árvore. Este tipo de sistema é conhecido como *sistemas de refutação*. Se, as consequências resultarem em algo impossível, pode-se concluir que a fórmula original está provada. Falaremos mais sobre este sistema na Seção 2.3.2.

2.3.1 Resolução Clausal

Este método de prova é conhecido como um procedimento refutacional: para podermos provar a fórmula φ, nós transformamos a sua negação ($\neg\varphi$) na Forma Normal, para, então, aplicarmos as regras de inferência ao conjunto de cláusulas resultante com a finalidade de gerar a cláusula vazia, que indicará uma contradição no conjunto de cláusulas, provando a fórmula original.

Existe uma forma normal específica para a lógica modal K, a Forma Normal Separada para Sistemas Normais baseada em Níveis Modais $\mathsf{SNF}_{\mathrm{K}}$, que faz com que os contextos referentes aos diferentes agentes sejam separados. As regras de transformação e a prova de correção do método de tradução para $\mathsf{SNF}_{\mathrm{K}}$ podem ser encontrados em [13]. Uma fórmula em $\mathsf{SNF}_{\mathrm{K}}$ é representada por um conjunto de cláusulas, que são verdadeiras em todos os mundos alcançáveis. Uma fórmula em $\mathsf{SNF}_{\mathrm{K}}$ possui o seguinte formato:

$$\bigwedge_i ml : C_i \tag{2.2}$$

onde cada C_i é uma cláusula e ml é o nível modal onde a cláusula ocorre.

As seguintes definições também são necessárias:

Definição 6 *Literal* é um símbolo proposicional $\mathbb{P}$ ou sua negação $\neg\mathbb{P}$. Denotamos por $\mathcal{L}$ o conjunto de todos os literais.

Definição 7 Um *literal modal* é uma fórmula na forma $\boxed{a}l$ ou sua negação $\neg\boxed{a}l$, sendo l um literal e com $a \in \mathcal{A}$.

Uma cláusula pode assumir um dos seguintes formatos:

- Uma cláusula de literais:

$$ml : \bigvee_{b=1}^{r} l_b$$

- Uma cláusula a-positiva:

$$ml : l' \Rightarrow \boxed{a}l$$

- Uma cláusula a-negativa:

$$ml : l' \Rightarrow \neg\boxed{a}l$$

onde l, l' e $l_b \in \mathcal{L}$, $a \in \mathcal{A}$, $r \in \mathbb{N}$ e $ml \in \mathbb{N}$.

Uma vez que as fórmulas estejam em sua forma normal, o método de resolução pode ser aplicado, ou seja, as regras de inferência podem ser aplicadas ao conjunto de cláusulas resultante. Nós omitiremos a descrição do método, que pode ser encontrada em [13], uma vez que para os objetivos deste trabalho precisamos tão somente da definição da forma normal apresentada acima.

2.3.2 Tableaux

Métodos de demonstração utilizando tableaux são frequentemente utilizados, com sucesso, em lógica modal para prover procedimentos de decisão. O tableaux consiste em uma prova representada graficamente na forma de árvore. É um método analítico baseado na prova por contradição [6].

Definição 8 Uma **regra** σ no método tableaux consiste em um numerador N, acima da linha, e uma lista (finita) de denominadores $D_1, D_2, \ldots, D_k$ (abaixo da linha), separados por barras verticais. O numerador e cada denominador são um conjunto finito de fórmulas [16].

Cada regra é lida de cima para baixo, e se o numerador é satisfatível, então um dos denominadores também o é. O numerador de cada regra do tableaux contém uma ou mais fórmulas distintas chamadas de *fórmula(s) principal(is)*. Um *cálculo baseado em tableaux* para uma lógica L ($\mathcal{C}_L$) é um conjunto finito de regras [8].

Dado um $\mathcal{C}_L$, um tableaux para uma fórmula φ é uma árvore tal que sua raiz contém φ e os outros nós contém conjuntos finitos de fórmulas obtidas a partir de seus nós-pais, através da ativação de uma regra de $\mathcal{C}_L$. Um *ramo*, em um tableaux, representa um caminho entre a raiz da árvore e um de seus nós. K; o artigo que você leu, do Goré, é para combinações envolvendo 4.

Definição 9 Seja Δ um conjunto de regras de tableaux. Dizemos que ψ é **obtível a partir de** φ por aplicações de regras em Δ se existe um tableaux para φ, que utiliza somente as regras contidas em Δ, e possui um nó que contém ψ.

Definição 10 Um ramo em um tableaux é dito **fechado** se seu nó final contém apenas $\bot$. Um tableaux é dito **fechado** se cada um de seus *ramos* está fechado. Um tableaux é chamado de **aberto** se não estiver fechado.

Definição 11 Um conjunto finito Γ de fórmulas é dito **$\mathcal{C}_L$-consistente** se cada $\mathcal{C}_L$-tableaux para Γ é aberto. Se existir pelo menos um $\mathcal{C}_L$-tableaux fechado para Γ então Γ é **$\mathcal{C}_L$-inconsistente**.

Definição 12 Um Cálculo baseado em Tableaux $\mathcal{C}_L$ é **correto** se para todos os conjuntos finitos Γ de fórmulas, se Γ é L-satisfatível então Γ é $\mathcal{C}_L$-consistente. É **completo** se para todos os conjuntos finitos Γ de fórmulas, se Γ é $\mathcal{C}_L$-consistente então Γ é L-satisfatível.

Definição 13 Seja σ uma regra de inferência pertencente a $\mathcal{C}_L$. Dizemos que σ é

correta com respeito a L se para qualquer instância σ' de σ, se o numerador de σ' for L-satisfatível então o denominador de σ' também o é.

Qualquer cálculo baseado em tableaux $\mathcal{C}_L$ contendo somente regras corretas com respeito a L, é correto.

O cálculo utilizado para implementar o gerador automático de modelos compreende um conjunto de regras de inferência para lidar com raciocínio tanto proposicional quando modal. A seguir, denotaremos por δ o resultado de unificar os rótulos das premissas de cada regra. Formalmente, a unificação é dada por uma função $\delta : \mathscr{P}(\mathbb{N}) \longrightarrow \mathbb{N}$, onde $\delta(\{ml\}) = ml$, para qualquer outra entrada, δ não está definida. As regras de inferência na Figura 2.1 só podem ser aplicadas se a unificação dos rótulos está definida[13].

[NEG]
$$\frac{\begin{array}{l} ml_1 : l'_1 \Rightarrow \langle a \rangle\, l_1 \\ \vdots \\ ml_m : l'_m \Rightarrow \langle a \rangle\, l_m \end{array}}{\begin{array}{l} ml+1 : l_1 \vee \ldots \vee l_m \\ \textit{onde } ml = \delta(\{ml_1, \ldots, ml_m\}) \end{array}}$$

[POS]
$$\frac{\begin{array}{l} ml_1 : l'_1 \Rightarrow [a]\, l_1 \\ \vdots \\ ml_m : l'_m \Rightarrow [a]\, l_m \\ ml_{m+1} : l' \Rightarrow \langle a \rangle\, l \end{array}}{\begin{array}{l} ml+1 : l_1 \wedge \ldots \wedge l_m \wedge l \\ \textit{onde } ml = \delta(\{ml_1, \ldots, ml_m, ml_{m+1}\}) \end{array}}$$

Figura 2.1: Regras de inferência do tableaux

Capítulo 3

Construção

Capítulo 4

Resultados

Referências

[1] Carlos Areces, Rosella Gennari, Juan Heguiabehere, and Maarten De Rijke. Tree-based heuristics in modal theorem proving. In *Proc. ECAI 2000*, pages 199–203. IOS Press, 2000. 6

[2] B. F. Chellas. *Modal logic — an introduction.* Press Syndicate of the University of Cambridge, London, 1980. 3

[3] S. A. Cook. The complexity of theorem-proving procedures. In *In Proceedings of the 3rd annual ACM STOC*, pages 151–158, 1971. 1

[4] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge.* MIT Press, Cambridge, MA, USA, 1995. 1

[5] M. Fitting. Destructive modal resolution, 1989. 3

[6] M. Fitting and R. L. Mendelsohn. *First order modal logic.* Kluwer Academic Plubishers, New York, 1998. 8

[7] D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. Many-dimensional modal logics: theory and applications. *Studies in logic and the foundations of Mathematics*, 148, 2003. 1

[8] R. Goré and L. A. Nguyen. Clausal tableaux for multimodal logics of belief. *Fundamenta Informaticae*, 94:21–40, 2009. 8

[9] B. T. Hailpern. *Verifying Concurrent Processes Using Temporal Logic*, volume 129 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin/New York, 1982. 1

[10] J. Halpern, Z. Manna, and B. Moszkowski. A Hardware Semantics Based on Temporal Intervals. *Lecture Notes in Computer Science*, 154:278–291, 1983. 1

[11] R. E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM J. Comput*, 6, 1977. 1

[12] C. Nalon and C. Dixon. Clausal resolution for normal modal logics. *Journal of Algorithms*, 62:117–134, 2007. 1

[13] Cláudia Nalon, Ullrich Hustadt, and Clare Dixon. A modal-layered resolution calculus for K. In Hans De Nivelle, editor, *Automated Reasoning with Analytic Tableaux and Related Methods - 24th International Conference, TABLEAUX 2015, Wrocław, Poland, September 21-24, 2015. Proceedings*, volume 9323 of *Lecture Notes in Computer Science*, pages 185–200. Springer, 2015. 1, 7, 9

[14] Cláudia Nalon, Ullrich Hustadt, and Clare Dixon. Ksp: A resolution-based prover for multimodal k. In Nicola Olivetti and Ashish Tiwari, editors, *Automated Reasoning: 8th International Joint Conference, IJCAR 2016, Coimbra, Portugal, June 27 – July 2, 2016, Proceedings*, pages 406–415, Cham, 2016. Springer International Publishing. 1, 2

[15] A. S. Rao and M. P. Georgeff. Modeling Rational Agents within a BDI-Architecture. In R. Fikes and E. Sandewall, editors, *Proceedings of Knowledge Representation and Reasoning (KR&R-91)*, pages 473–484, Cambridge, MA, USA, April 1991. Morgan-Kaufmann. 1

[16] Wolfgang Rautenberg. Modal tableau calculi and interpolation. *Journal of Philosophical Logic*, 12(4):403–423, 1983. 8