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.
59 p. : il. ; 29,5 cm.

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

1. palvrachave1, 2. palvrachave2, 3. palvrachave3

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 ciência...

Palavras-chave: palvrachave1, palvrachave2, palvrachave3

Abstract

The science...

Keywords: keyword1, keyword2, keyword3

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Em [**?**] 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 [**?**]. 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 [**?**] 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 [**?**]. Entretanto, as complexidades dos algoritmos propostos em [**?**] ainda não foram determinadas, sendo um dos objetos de investigação do atual projeto. Sabe-se, porém, que métodos de prova para lógica proposicional são intratáveis [**?**]. Em geral, métodos baseados em resolução, se ingenuamente implementados, levam também à utilização exponencial de espaço; entretanto, a utilização de estratégias garante a linearidade de espaço do método de resolução para lógicas proposicionais [**?**]. É, portanto, nosso intuito conduzir investigação da extensão e implementação de estratégias conhecidas (e.g. resolução linear, deleção de unidade e subsunção) que permitam a implementação eficiente dos algoritmos propostos em [**?**].

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 [**?**]. 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.

Capítulo 2

Revisão Teórica

2.1 Introdução a Lógica Modal

Neste capítulo, introduziremos brevemente a base do pensamento modal.

A lógica modal é determinada semanticamente por uma avaliação de necessidade e possibilidade — na literatura, encontramos representações análogas, como por exemplo, de conhecimento e crença, respectivamente [**?**] — isto é, em resumo , a base da estrutura modal que data desde discursos do filósofo Leibniz: uma proposição é *necessária* se ocorre em todos os possíveis mundos e *possível* se ocorre em algum mundo [**?**]. A ideia é que diferentes ações ou objetos podem ser verdadeiros em mundos diferentes, mas qualquer coisa que ocorra em todos os possíveis mundos é necessário, enquanto o 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: $\Box$ e $\Diamond$. Uma sentença da forma $\Box A$ — *necessariamente* A — é verdadeira se, e somente se, A é uma proposição verdadeira em todos os possíveis mundos; já uma sentença da forma $\Diamond A$ — *possivelmente* A — é verdadeira no caso em que A é uma proposição verdadeira em algum possível mundo.

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, e isto é feito listando uma sequência, podendo ser esta lista infinita, de conjuntos destes possíveis mundos,

$$P_0,\ P_1,\ P_2, \ldots \tag{2.1}$$

A intuição por trás deste modelo é que, para cada número natural n, o conjunto P_n contém somente os possíveis mundos onde a sentença $\mathbb{P}_n$ é verdadeira. Em outras palavras, a sequência $P_0,\ P_1,\ P_2, \ldots$ representa as sentenças atômicas estipulando em quais possíveis mundos elas ocorrem, ou seja, onde são verdadeiras, (e, por omissão, em quais mundos elas são falsas) da seguinte maneira:

$$\mathbb{P}_n \textit{ e verdadeiro em um possivel mundo } \alpha \textit{ se e somente se } \alpha \in P_n \tag{2.2}$$

A lógica modal pode ser expressa de diferentes formas e contar com diferentes axiomas através da ideia de sistemas, isto será explorado com mais detalhes na seção 2.3.1. Um

dos sistemas mais simples conhecidos é o $S5$, e este será utilizado para introdução, e, posteriormente, sua estrutura será expandido para uma forma mais geral.

Um modelo no sistema $S5$ é, portanto, uma tupla: $<W, P>$ onde W é um conjunto de possíveis mundos e P uma abreviação para a sequência infinita $P_0,\ P_1,\ P_2, \ldots$ de subconjuntos de W. Note que W pode conter mundos que não estão presentes em nenhum dos conjuntos P_n; de fato, qualquer um desses conjuntos pode ser vazio.

Definimos o valor da sentença (verdadeiro ou falso) de acordo com sua forma e em termos de um possível mundo em um modelo. Usamos a notação:

$$\models_{\alpha}^{\mathcal{M}} A \tag{2.3}$$

onde A é uma sentença e α é um possível mundo em um modelo $\mathcal{M} =< W, P >$. A lei 2.3 é um resumo para: A *verdadeiro em* α *nomodelo* $\mathcal{M}$.

As condições de verdade estão expressas na Tabela **??**.

Tabela 2.1: Truth conditions

(1) $\models_{\alpha}^{\mathcal{M}} \mathbb{P}_n$ se e somente se $\alpha \in P_n$ com $n = 0, 1, 2, \ldots$

(2) $\models_{\alpha}^{\mathcal{M}}$

(3) $Not\ \models_{\alpha}^{\mathcal{M}}$

(4) $\models_{\alpha}^{\mathcal{M}} \neg A$ se e somente se $not\ \models_{\alpha}^{\mathcal{M}} A$

(5) $\models_{\alpha}^{\mathcal{M}}$

(6) $\models_{\alpha}^{\mathcal{M}}$

(7) $\models_{\alpha}^{\mathcal{M}}$

(8) $\models_{\alpha}^{\mathcal{M}}$

(9) $\models_{\alpha}^{\mathcal{M}} \Box A$

(10) $\models_{\alpha}^{\mathcal{M}} \Diamond A$

Alguns esclarecimentos sobre essas definições podem ser úteis:

- A cláusula (1) reflete a premissa sobre os conjuntos $P_0, P_1, P_2, \ldots$ em um modelo: uma sentença atômica $\mathbb{P}_n$ é verdadeira em um possível mundo α somente no caso em que α é um elemento do conjunto P_n.
- De acordo com a cláusula (2), a constante verdadeira $\top$ é sempre válida em α.
- Por (3), a constante falsa $\bot$ é sempre falsa em α.
- A cláusula (4) afirma que a negação $\neg A$ é verdadeira em α se, e somente se, sua negação A é falsa em α.
- A afirmação da cláusula (5) diz que uma conjunção $A \wedge B$ é verdadeira em α somente no caso em que ambas sentenças, A e B o são.

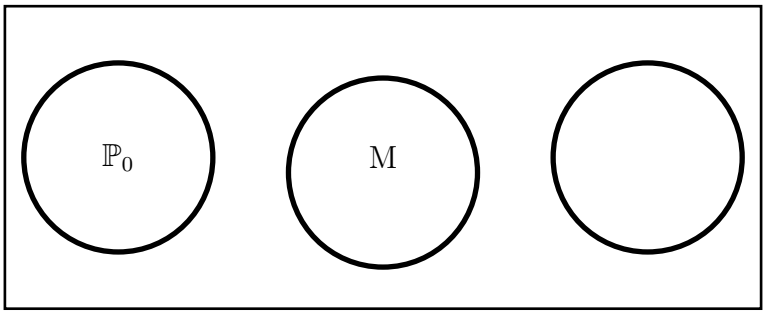

Figura 2.1: Exemplo lógica modal

- Já de acordo com a cláusula (6), uma disjunção $A \vee B$ é verdadeira em α quando pelo menos uma das sentenções, A ou B o é.
- A inteção com a cláusula (7) é compreender que uma implicação $A \rightarrow B$ é verdadeira em α desde que nunca ocorra que a sentença antecedente, A, seja verdadeira ao mesmo tempo em que a consequente, B, seja falsa.
- Similarmente, na cláusula (8) a intenção se repete de ambos os lados, ou seja, a condição $A \leftrightarrow B$ é verdadeira quando ambas sentenças, A e B, são verdadeiras, ou ambas são falsas.
- A cláusua (9) formula a interpretação leibniziana de necessidade: $\Box A$
- Finalmente, de acordo com a cláusula (10), $\Diamond A$

Observe que as constantes $\top$ e $\bot$ são, na realidade, operadores de aridade zero. Abstrações necessárias para algumas simplificações que veremos mais a frente.

Na Figura 2.1 está ilustrado um exemplo de modelo para a lógica modal. No exemplo, o modelo $\mathcal{M}$ possui três mundos, α, β e ω. Os conjuntos de mundos onde as sentenças atômicas são verdadeiras será expressado da seguinte forma:

$$P_0 = \{\alpha, \beta, \omega\} \tag{2.4}$$
$$P_1 = \{\alpha, \beta\} \tag{2.5}$$
$$P_2 = \{\alpha, \omega\} \tag{2.6}$$
$$P_n = \emptyset, \forall n > 2 \tag{2.7}$$

Podemos, portanto expressar as seguintes sentenças para o modelo $\mathcal{M}$, sendo todas verdadeiras:

$$P_n = \emptyset, \forall n > 2 \tag{2.8}$$

Observe que não é verdade que $\Box\mathbb{P}_1$, pois $\mathbb{P}_1$ não é ocorre em ω, ou seja, não satisfaz a condição de ocorrer em todos os possíveis mundos.

Uma sentença verdadeira em cada possível mundo em qualquer modelo é chamada *sentença válida*. Usamos o símbolo $\models$ novamente, mas desta vez sem as marcações superior

e inferior (para enfatizar que a sentença é verdadeira, sem importar o mundo ou o modelo), e escrevemos $\models A$ quando A é uma sentença válida. Mais formalmente, então, definimos a validade de uma sentença da seguinte forma:

$$\models A \textit{ se, e somente se, para todo modelo } \mathcal{M} \textit{ e para qualquer mundo } \alpha \textit{ em } \mathcal{M}\textit{, temos que } \models_{\alpha}^{\mathcal{M}} \tag{2.9}$$

No estudo da lógica da necessidade e possibilidade, queremos saber quais são as sentenças válidas — verdadeiras, sem se importar com interpretações, em qualquer mundo possível — e quais não são. Por exemplo, podemos demonstrar que toda sentença da forma $\Box A \to A$ é válida, enquanto que nem toda sentença da forma $A \to \Box A$ o é.

2.2 Preliminares lógicas

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.

Sentenças

A linguagem é baseada em um conjunto enumerável de sentenças *atômicas*:

$$\mathbb{P}_0, \mathbb{P}_1, \mathbb{P}_2, \ldots \tag{2.10}$$

Estas são as sentenças mais simples possíveis.

As *sentenças moleculares* (não-atômicas) são formadas por meio das nove *operações sintáticas*, ou *operadores lógicos*:

$$\top, \bot, \neg, \wedge, \vee, \to, \leftrightarrow, \Box, \Diamond \tag{2.11}$$

Como mencionado na seção anterior, $\top$ e $\bot$ são operadores de aridade zero, ou constantes; $\neg$, $\Box$ e $\Diamond$ são operadores de aridade um; e $\wedge$, $\vee$, $\to$ e $\leftrightarrow$ são operadores de aridade dois.

O conjunto de sentenças pode, então, ser definido formalmente como mostrado na Tabela 2.2.

As sentenças atômicas são todas distintas entre si, e a intenção é que sentenças de formas diferentes também sejam distintas, como, por exemplo, nenhuma sentença condicional é uma necessidade. Isto é garantido pela suposição de que o alcance dos operadores sintáticos é disjunto dois a dois e ainda, disjunto do conjunto de sentenças atômicas. A falta de ambiguidade das sentenças é garantida pela suposição de que as operações são todas *um-pra-um*, ou seja, duas condicionais são, por exemplo, iguais se, e somente se, ambos antecedentes e consequentes o são. Também é válido lembrar que o conjunto de sentenças é enumerável.

Tabela 2.2: Truth conditions

(1) a

(2) a

(3) a

(4) a

(5) a

(6) a

(7) a

(8) a

(9) a

(10) a

Convenções

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.

Fórmulas

Agora que expressamos claramente a sintaxe que será utilizada na escrita de sentenças e provas, podemos definir alguns conceitos que serão importantes mais a frente.

A definição de *fórmula* segue diretamente da sintaxe, e dizemos que é qualquer sequência finita de símbolos lógicos.

Definição 1 [Fórmula bem-formada (FBF)] A lingagem lógica proposicional, denotada por L_p, é equivalente ao seu conjunto de fórmulas bem-formadas, denotado por FBF_{L_P}, que é definito recursivamente como segue:

- se $p \in P$, então $p \in FBF_{L_P}$
- se $\phi \in FBF_{L_P}$ e $\varphi \in FBF_{L_P}$, então: $\neg\phi,\ (\wedge), (\vee), (\rightarrow), (\leftrightarrow) \in FBF_{L_P}$,

Definição 2 [Forma Normal Negada (FNN)] Seja $\varphi \in FBF_{L_P}$, dizemos que φ está na Forma Normal Negada (FNN) se contém apenas os conectivos $\neg$, $\wedge$ e $\vee$ e o conectivo de negação é aplicado apenas a símbolos proposicionais.

A transformação em FNN é dada pelo seguinte procedimento:

1. Substitua

2. Substitua
3. Aplique as leis de De Morgan
4. Elimine

A simplificação de fórmulas aplica as seguintes regras de reescrita:

•

Definição 3 [Forma Normal Conjuntiva (FNC)] Seja $\varphi \in FBF_{L_P}$, dizemos que φ está na Forma Normal Conjuntiva se é uma conjunção de cláusulas.

A transformação de uma fórmula φ em uma fórmula semanticamente equivalente, φ', na FNC, é dada pelo seguinte procedimento:

Definição 4 [Forma Normal Disjuntiva (FND)] Seja $\varphi \in FBF_{L_P}$, dizemos que φ está na Forma Normal Disjuntiva se é uma disjunção de conjunções de literais.

A transformação de uma fórmula φ em uma fórmula semanticamente equivalente, φ', na FND, é dada pelo seguinte procedimento:

2.2.2 Semântica

2.3 Modelos padrões para a lógica modal

A ideia geral dada na seção 2.1 é bastante simples, sendo modelada basicamente em termos de uma coleção de possíveis mundos junto com uma atribuição de valores booleanos, em cada mundo, a cada uma das sentenças atômicas. Vimos que isso dá uma noção de validade um tanto restrita. Nesta seção, iremos estender a definição leibniziana de necessidade e possibilidade introduzindo o conceito de relações. O resultado é uma noção de validade muito mais flexível.

Um modelo padrão será estruturado da seguinte forma:

$$\mathcal{M} = < W, R, P > \tag{2.12}$$

onde W representa o conjunto de possíveis mundos, P representa a atribuição dos subconjuntos de possíveis mundos a cada sentença atômica e o novo elemento R, é uma relação entre possíveis mundos, ou seja, R é uma relação binária em W ($R \subseteq W \times W$).

Escreveremos

$$\alpha R \beta \tag{2.13}$$

para dizer que o mundo β está relacionado com, ou é relevante para, o mundo α. É importante ressaltar que R pode ser qualquer tipo de relação binária em W, nenhuma suposição é feita sobre sua estrutura.

As condições de verdade de sentenças não-modais, dadas na Tabela **??**, permanecem inalteradas. Já as relacionadas às sentenças modais sofre uma pequena alteração, levando em conta agora, a relação R da seguinte forma:

Tabela 2.3: Condições de verdade: sentenças modais

Seja α e

(1) $\models^{\mathcal{M}}_{\alpha} \mathbb{P}_n$ se e somente se $\alpha \in P_n$ com $n = 0, 1, 2, \ldots$

(2) $\models^{\mathcal{M}}_{\alpha}$

2.3.1 Sistemas de lógica modal

Sistema $S5$

Nesta seção vamos examinar a necessidade e a possibilidade em $S5$ de um ponto de vista axiomático. Começamos com uma axiomatização baseada nos princípios das seções anteriores, ou seja, adotaremos como axiomas, ou teoremas básicos, quaisquer sentenças das seguintes formas:

$$T. \tag{2.14}$$
$$5. \tag{2.15}$$
$$K. \tag{2.16}$$
$$Df\Diamond. \tag{2.17}$$
$$PL. \tag{2.18}$$

E assumimos as seguintes regras de inferência:

$$(RN)\ \frac{A}{\Box\, A}$$

$$(MP)\ \frac{A \rightarrow\ B, A}{B}$$

Por teorema, geralmente, queremos dizer qualquer sentença que possa ser provada com base nos axiomas e nas regras de inferência e denotamos como $\vdash A$ para dizer que a sentença A é também um teorema.

Podemos observar que $S5$, aqui formulado como um sistema dedutivo, inclui a lógica proposicional, ou seja, a seguinte regra de inferência pode ser derivada de dentro do sistema:

$$(RPL)\ \frac{A_1, \ldots, A_n}{A}\ (n \geq\ 0)$$

Para provarmos tal regra, mostramos que se a inferência de $A_1, \ldots, A_n$ para A é proposicionalmente correta, e cada um dos $A_1, \ldots, A_n$ é um teorema, então A também é um teorema. A suposição de que a inferência é proposicionalmente correta implica que

A é verdadeiro em qualquer validação onde cada um dos $A_1, \ldots, A_n$ é verdadeiro, o que basicamente expressa que a sentença

$$A_1 \to (\ldots (A_n \to A) \ldots) \tag{2.19}$$

é uma tautologia (PL), ou seja, um teorema. Aplicamos a regra (MP) n vezes, e concluímos que A também é um teorema.

Utilizamos o sistema $S5$ para ilustrar alguns conceitos iniciais de prova e regras de inferência. A seguir, iremos mostrar como aplicá-los a sistemas mais complexos da lógica modal.

Outros sistemas

Os diferentes sistemas de lógica modal surgem em parte pela necessidade em diferentes tipos de aplicações e em parte porque nossas intuições sobre o que é necessariamente verdade ainda não estão completamente desenvolvidas. Originalmente, lógicas modais diferentes eram caracterizadas por conjuntos de axiomas distintos.

Definimos sistemas de lógica modal em termos do fecho sobre a regra de inferência (RPL), onde A é uma consequência tautológica das sentenças $A_1, \ldots, A_n$.

Definição 5 [Sistema de lógica modal] Um conjunto de sentenças é um *sistema de lógica modal* se, e somente se, é fechado sobre (RPL).

Assim, um sistema de lógica modal é qualquer conjunto de sentenças fechado com respeito a todos os modos proposicionalmente corretos de inferência. Reservaremos a notação Σ como uma variável para conjuntos de sequências que são sistemas de lógica modal, e, para simplicidade, constantemente iremos nos referir aos mesmos simplesmente como sistemas.

Os teoremas em um sistema são simplesmente as sentenças que o mesmo contém. Escrevemos $\vdash_\Sigma A$ quando A é um teorema de Σ.

Definição 6 [Teorema em sistemas modais] $\vdash_\Sigma A$ se, e somente se $A \in \Sigma$.

Como sistemas são simplesmente conjuntos de sentenças, a força de um sistema é medido em termos de inclusão: um sistema é pelo menos tão forte quanto um sistema Σ — e denotaremos como um Σ-sistema — somente no cso em que contém todo e qualquer teorema de Σ (note que esta relação é reflexiva).

Theorem 2.3.1. *O conjunto de tautologias é definido a partir do axioma PL, definido na seção anterior.*
(1) PL é um sistema de lógica modal.
(2) Todo sistema de lógica modal é um PL-sistema.
(3) PL é o menor sistema de lógica modal.

Demonstração. □

Podemos definir noções de *dedutibilidade* e *consistência*, em termos de teoremicidade.

Uma sentença A é dedutível de um conjunto de sentenças Γ e um sistema Σ — e escrevemos $\Gamma \vdash_\Sigma A$ — se, e somente se, Σ contém um teorema da forma:

$$(A_1 \wedge \ldots \wedge A_n) \rightarrow A \tag{2.20}$$

onde cada A_i $(i = 1, \ldots, n)$ dos antecedentes é uma sentença em Γ.

Um conjunto de sentenças Γ é consistente em Σ — e escrevemos $Con_\Sigma\Gamma$ — somente no caso em que a sentença $\perp$ não é Σ-dedutível a partir de Γ.

Mais formalmente:

Definição 7 [Dedutibilidade]

Definição 8 [Consistência]

Teoremicidade, dedutibilidade e consistência, como definitos, possuem todas as propriedades esperadas, algunas das quais estão enumeradas no seguinte teorema.

Theorem 2.3.2.

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

12.

13.

14.

15.

Demonstração. □

Tabela 2.4: Alguns sistemas padrões de lógica modal

Sistema	condições para classe de quadros
K	sem condições
D	serial
T	reflexivo
B	reflexivo, simétrico
$K4$	transitivo
$S4$	reflexivo, transitivo
$S5$	reflexivo, simétrico, transitivo

Iremos agora introduzir um conjunto dos sistemas mais conhecidos da lógica modal: K, D, T, B, $K4$, $S4$ além de retomarmos brevemente sobre o sistema $S5$ mencionado anteriormente. A definição destes sistemas utiliza a semântica de possíveis mundos.

Definição 9 [L-Válida] Dizemos que o modelo $\mathcal{M} =< W, R, P >$ é baseado no quadro $< W, R >$. Umafórmula X é válida no modelo $\mathcal{M}$ se é verdadeira em todo mundo de W. Uma fórmula X é válida em um quadro se é válida em cada modelo baseado naquele quadro. Finalmente, se L é uma coleção de quadros, X é L-válida se X é válida em cada quadro em L.

Diferentes sistemas de lógica modal são caracterizados semanticamente como as fórmulas L-válidas, para L classes em particular de quadros. Para exemplificar, o sistema chamado T é caracterizado pela classe de quadros que contenham a propriedade de que todo mundo é acessível por ele mesmo. Este tipo de propriedade pode vir a ser útil posteriormente e está definida a seguir, junto com outras propriedades de valor similar.

Definição 10 Seja $Q =< W, R >$ um quadro. Dizemos que Q é:

1. *reflexivo*
2. *simétrico*
3. *transitivo*
4. *serial*
5. *euclidiano*

O sistema T, mencionado acima, é caracterizado pela classe de quadros reflexivos. A Tabela 2.5 informa as classes de quadros aos quais os demais sistemas são caracterizados.

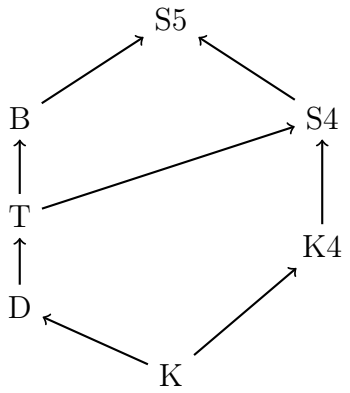

Figura 2.2: Subsistemas

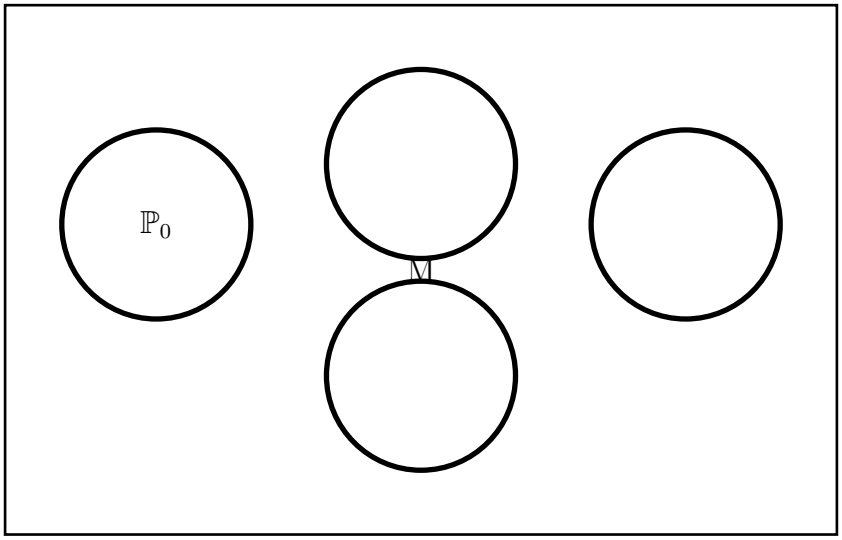

Figura 2.3: Exemplo estrutura de Kripke

2.3.2 Lógica Multimodal

Estrutura de Kripke

Uma das estruturas mais estudadas em lógica modal corresponde à estrutura de Kripke. Uma estrutura de Kripke para P e $A_n = \{1, \ldots, n\}$ é dada da seguinte forma:

$$\mathbb{M} =< W,\ R_1, \ldots,\ R_n, \pi > \tag{2.21}$$

onde:

- W é um conjunto não-vazio de possíveis mundos
- para todo $a \in A_n, R_a \subseteq W \times W$
- $\pi : W \times P \longrightarrow \{Falso, Verdadeiro\}$

2.4 Resolução

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 2.4.2.

2.4.1 Resolução Clausal

Resolução clausal é um procecimento refutacional: para podermos provar a sentença φ, nós transformamos a negação de φ na Forma Normal Conjuntiva antes de aplicar a única regra de inferência definida para este tipo de prova.

Definição 11 [RES]

$$\frac{(\varphi \vee \gamma) \qquad (\psi \vee \neg\gamma)}{(\varphi \vee \psi)}$$

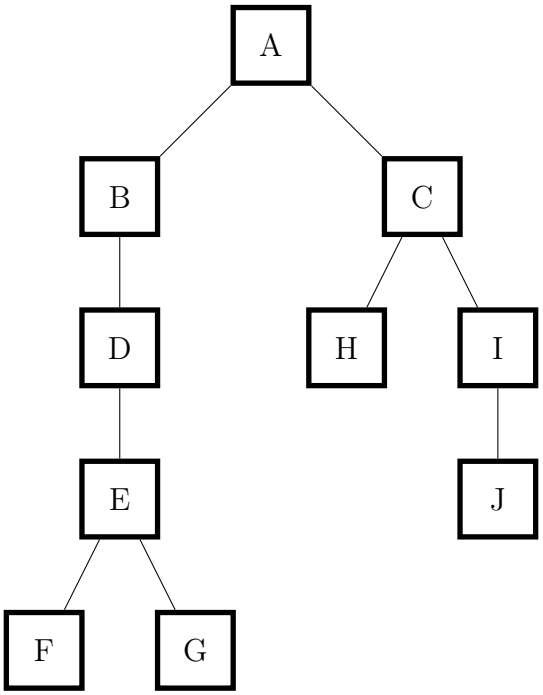

Figura 2.4: Exemplo lógica modal

2.4.2 Tableaux

Métodos de demonstração utilizando tableaux são frequentemente utilizados, com sucesso, em lógica modal para prover decisões de procedimento.

O tableaux consiste em uma prova representada graficamente na forma de árvore. É um método analítico baseado na prova por contradição.

Na Figura **??**, os quadrados são chamados *nós*, que, no *tableaux*, representam as fórmulas pré-fixadas, definidas a seguir. O nó `A` é o nó *raiz* e os nós `F, F, H` e `J` são o que chamamos de *folhas*. O nó `A` possui dois *filhos*, `B` e `C`, com `B` sendo o filho da esquerda e `C` o filho da direita. Se um nó é filho de outro nó, o segundo é chamado *pai* do primeiro — por exemplo, `C` é o pai de `H` e de `I`. A sequência A, B, D, E é um exemplo de *caminho*. Outro exemplo de caminho é a sequência A, B, D, E, F, que também chamado de *ramo*, que corresponde a um caminho máximo.

Definição 12 [Prefixo] Um prefixo é uma sequência finita de inteiros positivos. Uma *fórmula prefixada* é uma expressão da forma σX, onde σ é um prefixo e X é uma fórmula.

Escrevemos prefixos usando pontos para separar os inteiros, por exemplo: 1.2.3.2.1. Também, se σ é um prefixo e n é um inteiro positivo, $\sigma.n$ é um prefixo.

A ideia intuitiva é que um prefixo σ nomeia um possível mundo em algum modelo, e σX nos diz que X é verdadeiro no mundo que σ representa. Nossa intenção é que $\sigma.n$ deve sempre representar um mundo que é acessível daquele que σ representa. Outros

fatos sobre os prefixos dependem em qual lógica modal se está considerando. Estes serão abordados mais a frente.

Uma tentativa de construção de prova por tableaux para uma fórmula Z começa inicializando a árvore de prova com $1\neg Z$ como raiz, ou seja, afirmamos que Z não ocorre (não é verdadeiro) em um possível modelo no mundo representado por 1. Em seguida, ramos são construídos de acordo com certas *Regras de Extensão*, definidas em breve. Finalmente, é necessário que cada ramo esteja *fechado*, definição que será dada após as regras de construção dos ramos.

Definição 13 [Regras Conjuntivas] Para qualquer prefixo σ:

$$\frac{\sigma\ \neg\neg X}{\sigma\ X} \qquad \frac{\sigma\ \neg\neg X}{\sigma\ X}$$

$$\frac{\sigma\ \neg\neg X}{\sigma\ X} \qquad \frac{\sigma\ \neg\neg X}{\sigma\ X}$$

Definição 14 [Regras Disjuntivas] Para qualquer prefixo σ:

$$\frac{\sigma\ \neg\neg X}{\sigma\ X} \qquad \frac{\sigma\ \neg\neg X}{\sigma\ X}$$

$$\frac{\sigma\ \neg\neg X}{\sigma\ X} \qquad \frac{\sigma\ \neg\neg X}{\sigma\ X}$$

Definição 15 [Regra da Dupla Negação] Para qualquer prefixo σ:

$$\frac{\sigma\ \neg\neg X}{\sigma\ X}$$

Definição 16 [Regras de Possibilidade] Para qualquer prefixo σ:

$$\frac{\sigma\ \neg\neg X}{\sigma\ X} \qquad \frac{\sigma\ \neg\neg X}{\sigma\ X}$$

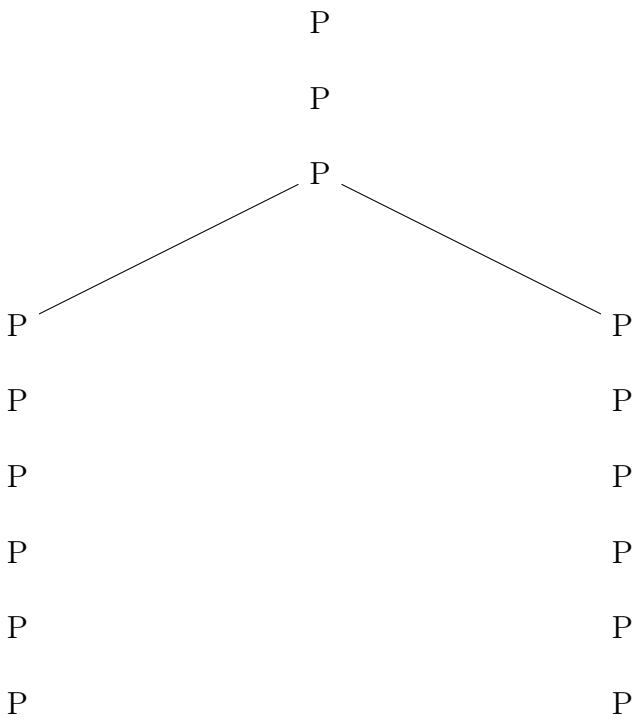

Figura 2.5: Exemplo de prova por tableaux

Definição 17 [Regras de Necessidade Básicas] Para qualquer prefixo σ:

$$\frac{\sigma\ \neg\neg X}{\sigma\ X} \qquad \frac{\sigma\ \neg\neg X}{\sigma\ X}$$

Definição 18 [Fecho] Um ramo do tableaux é dito *fechado* se contém ambos σX e $\sigma\neg X$ para alguma fórmula X. Um ramo que não está fechado, é dito *aberto*. Um tableaux é dito fechado se todos os seus ramos estão fechados.

Definição 19 [Prova por Tableaux] Um tableaux fechado para $1\neg Z$ é uma *prova por tableaux* de Z, e Z é um *teorema* se possui uma prova por tableaux.

Tabela 2.5: Alguns sistemas padrões de lógica modal

Sistema	condições para classe de quadros
D	serial
T	reflexivo
B	reflexivo, simétrico
$K4$	transitivo
$S4$	reflexivo, transitivo
$S5$	reflexivo, simétrico, transitivo

Definição 20 [Regras especiais de necessidade]

$$T \quad \frac{\sigma\ \neg\neg X}{\sigma\ X} \qquad \frac{\sigma\ \neg\neg X}{\sigma\ X}$$

$$D \quad \frac{\sigma\ \neg\neg X}{\sigma\ X} \qquad \frac{\sigma\ \neg\neg X}{\sigma\ X}$$

$$B \quad \frac{\sigma\ \neg\neg X}{\sigma\ X} \qquad \frac{\sigma\ \neg\neg X}{\sigma\ X}$$

$$4 \quad \frac{\sigma\ \neg\neg X}{\sigma\ X} \qquad \frac{\sigma\ \neg\neg X}{\sigma\ X}$$

$$4r \quad \frac{\sigma\ \neg\neg X}{\sigma\ X} \qquad \frac{\sigma\ \neg\neg X}{\sigma\ X}$$

2.4.3 Tableaux Clausal

Capítulo 3

Implementação

3.1 Pré-Processamento

3.1.1 Níveis de *Hash*

3.2 Detalhes da Geração de Modelos

3.2.1 Estruturas

```
typedef struct model_t{
    unsigned int id;
    struct world_l* worldslist;
}model_t;
```

```
struct world_t {
    unsigned int id;
    unsigned int modal_level;
    struct literalslist* literals;
    struct pair_t* positives;
    struct pair_t* negatives;
    relation_t* relation;
};
```

```
struct pair_t {
    unsigned int agent;
    struct literalslist* literals;
    struct pair_t* next;
};
```

```
struct relation_t {
    unsigned int agent;
    unsigned int modal_level;
    world_l* worldslist;
    relation_t* next;
```

```
};
```

```
struct world_l {
    world_t* world;
    world_l* next;
};
```

```
typedef struct branch_t {
    unsigned int id;
    int dead;
    struct world_t* world;
    struct branch_t* next;
}branch_t;
```

3.3 Estudo da Complexidade do Algoritmo

3.3.1 Pontos Críticos de Processamento

Capítulo 4

Resultados

Referências