Ariel Agne da Silveira

Implementação de uma biblioteca de lógica modal em Coq

Joinville

2020

UNIVERSIDADE DO ESTADO DE SANTA CATARINA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Ariel Agne da Silveira

IMPLEMENTAÇÃO DE UMA BIBLIOTECA DE LÓGICA MODAL EM COQ

Trabalho de conclusão de curso submetido à Universidade do Estado de Santa Catarina como parte dos requisitos para a obtenção do grau de Bacharel em Ciência da Computação

Karina Girardi Roggia
Orientadora

Paulo Henrique Torrens
Coorientador

Joinville, Setembro de 2020

IMPLEMENTAÇÃO DE UMA BIBLIOTECA DE LÓGICA MODAL EM COQ

Ariel Agne da Silveira

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção do título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo Curso de Ciência da Computação Integral do CCT/UDESC.

Banca Examinadora

Karina Girardi Roggia - Doutora (orientadora)

Paulo Henrique Torrens - Mestre (coorientador)

Cristiano Damiani Vasconcellos - Doutor

Rafael Castro Gonçalves Silva - Mestre

"Lógica de Programação é a Terra plana da computação."

Gabriela Moreira Mafra

Resumo

A modelagem de determinados tipos de sistemas computacionais com a lógica clássica possui fatores limitantes. Neste contexto, a apresentação de outros sistemas lógicos, como a lógica modal, e a construção de uma biblioteca para o assistente de provas Coq tem o intuito de auxiliar na modelagem e facilitar o uso para a verificação de propriedades de sistemas. A semântica da lógica modal é representada pela semântica dos mundos possíveis, onde existe uma relação de acessibilidade que conecta os mundos de um modelo. Diferentes restrições impostas na relação de acessibilidade constroem sistemas da lógica modal que auxiliam na representação de propriedades nas mais diversas áreas de estudo. O desenvolvimento da biblioteca tem como objetivo sustentar a formalização de propriedades de *softwares* e prová-los em Coq.

Palavras-chaves: Lógica Modal, Coq, Biblioteca

Abstract

The modelling of certain types of computational systems with classical logic includes limiting factors. In this context, the presentation of other logical systems, such as modal logic, and the construction of a library for the proof assistant Coq intents to help in the modelling and facilitate usage on the verification of systems' properties. The semantics of modal logic is represented by the semantics of possible worlds, where there is an accessibility relationship that connects the worlds of a model. Different restrictions imposed on the accessibility relation build modal logic systems that help representation of properties on a wide array of research areas. The libary development aims to held the formalization of properties in softwares and prove them on Coq.

Keywords: Modal Logic, Coq, Library

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

MP	*Modus Ponens*
CIC	*Calculus of Inductive Constructions*
CoC	*Calculus of Constructions*

1 Introdução

A lógica matemática é de fundamental importância para as linguagens de programação na construção de programas computacionais (BERTOLINI; CUNHA; FORTES, 2017). Durante o século XX houve um estudo avançado para o desenvolvimento de vários sistemas ou lógicas (GRUPO DE LÓGICA E FUNDAMENTOS DA FÍSICA, 2008) onde não se era permitido modelar algumas sentenças em lógica clássica. Com a construção de lógicas não clássicas viabilizou-se a formalização de aspectos temporais, do conhecimento, quânticos, entre outros. A definição de lógicas não clássicas abrange aspectos nos quais a lógica clássica não satisfaz. De acordo com Haack (1978) em uma lógica não clássica há uma extensão da lógica clássica ou quebra de pelo menos um possível paradigma existente, seja o princípio do terceiro excluído, referente a uma proposição possuir somente dois valores como verdadeiro ou falso, a não contradição, em que uma proposição e sua forma negada não se contradizem, ou a identidade, no qual uma proposição é sempre igual a si mesma.

Diferentes lógicas foram construídas a partir da lógica proposicional clássica com o intuito de facilitar a modelagem em diferentes áreas. Como, por exemplo, a lógica multivalorada, que não satisfaz o princípio do terceiro excluído e possui como foco a atribuição de múltiplos valores-verdades e não somente verdadeiro e falso como proposto por Aristóteles. Neste caso, a lógica se ramifica para outras sub-lógicas, tal como a *fuzzy*, probabilística ou ternária. Outro exemplo é a lógica quântica, no qual o princípio da identidade não se aplica, seu foco é apontado para o raciocínio de proposições para a física quântica, de forma que possa modelar sub-espaços fechados de partículas e posições vetoriais, chamados de espaço de Hilbert (AGUDELO, 2009).

A lógica modal se enquadra no caso de extensão da lógica clássica, já que possui o acréscimo de novos operadores e não fere nenhum dos princípios da lógica proposicional. Para Garson (2018), a utilização de seus símbolos significa o tratamento da qualificação de uma afirmação analisada, ou seja, sentenças determinadas como necessárias ou possíveis. Com a construção de novos operadores, a lógica modal além de alcançar argumentos tratados de formas alética, pode tratar também de argumentos temporais, de obrigatoriedade e permissão, ou a respeito de conhecimento e crença. Não somente vista na ciência da computação, a lógica modal é utilizada em diferentes ramos, como

por exemplo no direito, em que se aplica a lógica da obrigatoriedade e permissão, conhecida como lógica deôntica, e na área da filosofia, empregando a lógica do conhecimento, intitulada por epistêmica.

Dentro da ciência da computação a lógica modal é empregada em diferentes ramos, como por exemplo teoria de criptografia, inteligência artificial, teoria de banco de dados, sistemas distribuídos e verificação de programas (GOLINSKA-PILAREK; MUNOZ-VELASCO; MORA, 2010). Conforme a modalidade tratada na lógica, pode-se realizar verificações para ocorrência de um programa, de acordo com Blackburn, Rijke e Venema (2001), essas verificações se enquadram na observação de *dead ends*, *loops* e *forkings*, de tal forma que facilita a programação do sistema e evita tais problemas.

A formalização de lógica modal em assistentes de provas de teoremas auxilia na especificação e verificação de propriedades de *software*, como a análise de concorrência entre sistemas para um ambiente compartilhado ou a verificação de conhecimento das gerações passadas para as gerações futuras em algoritmo cultural. A utilização destes assistentes de provas, como por exemplo, Coq, Isabelle, HOL-Light e Lean, faz com que provas formais e verificações de programas sejam desenvolvidas com a assistência de um usuário. O assistente de provas Coq é um ambiente para o desenvolvimento de fatos matemáticos (PAULIN-MOHRING, 2012), em que se utiliza táticas em premissas e hipóteses para chegar em uma conclusão válida. A linguagem utilizada no Coq é uma variedade de tipos, chamada de *Calculus of Inductive Constructions* (CIC) (TEAM, 2019). Segundo Paulin-Mohring (2015), este formalismo representa programas funcionais como a linguagem ML e consegue caracterizar estruturas de dados, como listas e árvores binárias, onde essas árvores podem ter ramificações infinitas.

A disponibilidade online de materiais didáticos e a ampla pesquisa em sistemas de tipos e assistentes de provas (SILVA, 2019) faz com que o Coq seja conhecido em diversos meios, tanto ambientes acadêmicos quanto em áreas de pesquisa. Segundo Alves (2018), o Coq baseia-se em programação de ordem superior, no qual se permite a extração de código para outras linguagens, como Scheme e Haskell, visto que o uso da ferramenta pode ser usufruído para o desenvolvimento de programas comuns.

1.1 Objetivo Geral

Este trabalho tem como objetivo implementar uma biblioteca de lógica modal para aplicação em um assistente de provas de teoremas denominado Coq.

1.2 Objetivos Específicos

Com base no objetivo geral, são definidos os seguintes objetivos específicos:

- Apresentar de forma clara e objetiva a lógica modal.
- Representar fórmulas de lógica modal em Coq.
- Modelar a semântica da lógica modal em Coq.
- Implementar os diferentes sistemas de lógica modal em Coq.
- Provar a correção da lógica modal com base na biblioteca criada.

1.3 Metodologia

O procedimento metodológico apresentado neste trabalho iniciou por um levantamento de conceitos a respeito da lógica modal e os diferentes sistemas que estão contidos na mesma, conceitualização sobre assistentes de provas e aprofundamento sobre a implementação de táticas em Coq. Em paralelo, foi realizada uma revisão bibliográfica nos trabalhos relacionados para a identificação de temas semelhantes que poderiam auxiliar no desenvolvimento das tarefas propostas.

Finalizada a etapa anterior, foi realizada a implementação da biblioteca de lógica modal no assistente de provas Coq e a demonstração que o desenvolvimento proposto é robusto através da prova de correção. O acesso ao código desenvolvido pode ser encontrado no GitHub do autor, endereçado por:
`https://github.com/arielsilveira/ModalLibrary`

1.4 Estrutura do Trabalho

O presente trabalho está organizado no seguinte formato: o Capítulo 2 apresenta a lógica modal, tal como conceitos, exemplos, provas de correção e completude, e extensões a partir do sistema K. O Capítulo 3 demonstra o que são assistentes de provas, como por exemplo Isabelle, HOL-Light, Lean e retrata de forma aprofundada o Coq. O Capítulo 4 apresenta os trabalhos relacionados ao objetivos deste. O Capítulo 5 apresenta as principais características implementadas da biblioteca modal e algumas bibliotecas do Coq que auxiliaram no desenvolvimento da mesma. O Capítulo 6 aborda as conclusões finais do trabalho e apresenta algumas sugestões como trabalhos futuros.

2 Lógica Modal

A lógica modal é um desenvolvimento sistemático das diferentes noções lógicas expressas na linguagem natural por palavras e frases modais (COCCHIARELLA; FREUND, 2008). Diversas áreas da computação utilizam a lógica modal para modelar sistemas, tal como inteligência artificial (MASTOP, 2012), sistemas distribuídos (ALLWEIN; HARRISON, 2016), verificação de programas (MALANOVICZ, 2001), representação do conhecimento, semântica formal e linguística computacional (BLACKBURN; RIJKE; VENEMA, 2001). Esta lógica é estendida a partir da lógica proposicional clássica, com a adição de dois novos conectivos unários, $\Box$ e $\Diamond$, denominados respectivamente por necessidade e possibilidade (WIND, 2001).

O objetivo da lógica modal é a caracterização de validade ou invalidade dos argumentos propostos (GARSON, 2018), diferentemente da lógica proposicional clássica, não é possível fornecer tabela-verdade para a lógica modal, pois não há colunas que definem a utilização dos conectivos de necessidade ($\Box$) e possibilidade ($\Diamond$) (GARSON, 2018). Segundo MALANOVICZ (2002), as lógicas modais tratam argumentos de necessidade, possibilidade e contingência, na sua forma verdadeira ou falsa. Para Zalta (1995), os operadores modais analisam a validade das proposições nos mundos em que se encontram, ou seja, uma fórmula pode ser necessariamente verdadeira ou possivelmente verdadeira.

A análise semântica, vista no tratamento de formas necessárias através da modalidade alética, é dada pela interpretação de uma fórmula $\Box p$, em que p seja verdade em todos os mundos possíveis, como por exemplo, "É necessário que todas as bolas sejam redondas", ou seja, em todos os mundos possíveis as bolas têm que ser redondas. Entretanto, a possibilidade é vista de forma que, uma fórmula é possivelmente verdadeira se, e somente se, pelo menos em um mundo possível esta fórmula é verdadeira (ROCHA, 2010). Uma propriedade vista em modalidade contingente ocorre pela verdade de uma proposição no mundo atual e falsa em alguns mundos possíveis (MELO, 2018), como por exemplo, "O fogo é fonte de calor", onde a frase é verdadeira no mundo atual, porém, podem existir alguns mundos possíveis em que o fogo não seja fonte de calor. A diferença de uma proposição ser contingente e possível é vista, respectivamente, de forma local e global. Uma proposição de contingência se dá pela existência dela no mundo atual e

contraditória em alguns outros mundos, já uma possivelmente verdadeira, independe do seu valor verdade no mundo atual, entretanto, pode ser existente em algum outro mundo acessível.

Neste Capítulo serão definidos conceitos da lógica modal, apresentando suas definições, linguagem, semântica, sistema dedutivo, meta propriedades, os diferentes sistemas da lógica e algumas versões além da lógica modal alética. A Sessão 2.1 representa a linguagem estendida da lógica modal sobre a lógica proposicional clássica. A Sessão 2.2 expõe sobre a semântica e exemplo demonstrativo da funcionalidade da relação de acessibilidade. A Sessão 2.3 mostra a axiomatização, apresenta o sistema K, regra da substituição e exemplo de prova através da axiomatização. A Sessão 2.4 retrata as provas de correção e completude da lógica modal. A Sessão 2.5 apresenta os diferentes sistemas de lógica modal e as suas relações de acessibilidade. A Sessão 2.6 mostra algumas extensões da lógica modal.

2.1 Linguagem

A linguagem de uma lógica é definida pelos símbolos que a compõe e a forma de construir fórmulas. Uma fórmula modal é uma fórmula constituída de símbolos proposicionais, operadores clássicos e operadores modais (MALANOVICZ, 2001). A simbologia presente neste trabalho denota-se por $\mathcal{P} = \{p,\ q,\ r,\ \dots\}$ para a representação do conjunto de símbolos proposicionais, e $\Gamma = \{\alpha, \beta, \gamma, \dots\}$ denota o conjunto de fórmulas bem formadas da lógica. Os símbolos $\top$ e $\bot$ são símbolos da lógica matemática que representam a tautologia e falsidade, respectivamente. A linguagem da lógica proposicional clássica é definida a seguir:

Definição 1 (Linguagem da Lógica Proposicional)**.** *O conjunto das fórmulas da lógica proposicional clássica, denotado L_{LPC}, é estabelecido indutivamente por:*

- $\top, \bot \in L_{LPC}$
- $\mathcal{P} \subseteq L_{LPC}$
- *Se $\varphi \in L_{LPC}$, então $\neg\varphi \in L_{LPC}$*
- *Se $\varphi, \psi \in L_{LPC}$, então $\varphi \circ \psi \in L_{LPC}$ tal que $\circ \in \{\wedge, \vee, \rightarrow\}$*

- *Nada mais pertence a* L_{LPC} ■

A linguagem da lógica modal (L_{LM}) é composta pela L_{LPC} com o acréscimo dos operadores unários de necessidade e possibilidade segundo a definição abaixo.

Definição 2 (Linguagem da Lógica Modal)**.** *O conjunto das fórmulas da lógica modal, denotado* L_{LM}*, é estabelecido indutivamente por:*

- $L_{LPC} \subseteq L_{LM}$
- *Se* $\varphi \in L_{LM}$*, então* $\Box\varphi$ *e* $\Diamond\varphi \in L_{LM}$*.*
- *Se* φ *e* $\psi \in L_{LM}$*, então* $\varphi \rightarrow \psi$*,* $\varphi \wedge \psi$*,* $\varphi \vee \psi$ *e* $\neg\varphi \in L_{LM}$
- *Nada mais pertence a* L_{LM}*.* ■

O conjunto de subfórmulas de φ, denotado por Subf(φ), define todos os componentes para a construção de uma fórmula φ. O caso básico analisa as fórmulas de estruturas mais simples (SILVA; FINGER; MELO, 2006), nos demais casos indutivos, verifica uma ou mais fórmulas juntamente com um operador unário ou binário.

Definição 3 (Subfórmulas)**.** *O conjunto de subfórmulas de uma fórmula* φ*, denotado* $Subf(\varphi)$*, é definido indutivamente por:*

- *Se* $\varphi = p$*, então Subf(*φ*)* $= \{p\}$*.*
- *Se* $\varphi = \circ\ \psi$*, então Subf(*φ*)* $= \{\circ\ \psi\ \} \cup$ *Subf(*ψ*), sendo* $\circ \in \{\neg, \Box, \Diamond\}$
- *Se* $\varphi = \psi \circ \rho$*, então Subf(*$\varphi$*)* $= \{\ \psi \circ \rho\ \} \cup$ *Subf(*ψ*)* $\cup$ *Subf(*ρ*), sendo* $\circ \in \{\wedge, \vee, \rightarrow\}$*.*

■

Os conectivos $\Box$ e $\Diamond$ são lidos como "φ é necessário" e "φ é possível", conhecidos como operadores de necessidade e possibilidade, respectivamente (GORANKO, 1999). Em uma representação abrangente da lógica modal, os operadores ganham novas definições, como por exemplo na lógica epistêmica em que $\Box\varphi$ é representado por "sabe-se que φ". Na lógica deôntica $\Box\varphi$ corresponde à "é obrigatório que φ", o símbolo de possibilidade é equivalente à "é permitido que φ". A Tabela 2.1 representa outras variedades de leituras.

Tabela 2.1: Significado correspondente para os símbolos modais $\Box$ e $\Diamond$

$\Box\varphi$	$\Diamond\varphi$
É necessariamente verdade que φ	É possível que φ
Sempre será verdade que φ	Alguma hora no futuro φ
Deve ser verdade que φ	É permitido que φ
O agente Q acredita que φ	φ é consistente com o que o agente Q acredita
O agente Q sabe que φ	Pelo que o agente Q sabe, φ
Depois de qualquer execução do problema P, φ é verdade	Depois de alguma execução do problema P, φ é verdadeira

Fonte: Huth e Ryan (2008, p. 240)

2.2 Semântica

A semântica de uma lógica fornece uma definição de validade caracterizando o comportamento verdade das sentenças do sistema (GARSON, 2018). Segundo Wind (2001), Kripke desenvolveu pesquisas sobre a semântica da lógica modal durante os anos de 1950 e 1960 e apresentou o termo de mundos possíveis. A noção de mundos possíveis é uma ilustração, obediente às regras da lógica, de como as coisas são ou podem ser (PRIMO, 2009), este conceito utiliza a aplicação da não contradição de fórmulas nos mundos do sistema analisado. Um mundo possível possui um conjunto de proposições válidas nele e uma relação de acessibilidade para próximos mundos, desta forma a relação de acessibilidade define como os mundos se comunicam. Seja $\mathcal{W}$ um conjunto de mundos, $w, y \in \mathcal{W}$ e $\mathrm{R} \subseteq \mathcal{W} \times \mathcal{W}$ uma relação de acessibilidade então, wRy, é lido por "o mundo y é acessível a partir do mundo w". Segundo Kripke (1963), diferentes sistemas modais variam com diferentes propriedades da relação de acessibilidade, como reflexividade, simetria, transitividade, etc.

Garson (2018) descreve uma relação de equivalência na utilização dos quantificadores universais e existenciais da lógica proposicional clássica com os operadores de necessidade e possibilidade da lógica modal, respectivamente. O operador $\Box$ pode ser comparado paralelamente com o quantificador universal $\forall$, pois é descrito como uma fórmula válida para todos os mundos acessíveis a partir do atual. Similarmente o operador $\Diamond$ tem relação com o quantificador existencial $\exists$, (GARSON, 2018), onde existe pelo menos

um mundo acessível que uma fórmula seja satisfazível. Segundo Vardi (1997), o dual do operador modal $\Box\varphi$, é dado por $\neg\Diamond\neg\varphi$, o mesmo se aplica ao trocar entre si $\Box$ por $\Diamond$.

Definição 4 (*Frame*)**.** *Um Frame é um par ordenado que especifica o conjunto de mundos e sua relação de acessibilidade, desta forma tem-se que* $\mathcal{F} = \langle \mathcal{W}, R \rangle$*, onde* $\mathcal{W}$ *é um conjunto não vazio de mundos e* $R \subseteq \mathcal{W} \times \mathcal{W}$*, é uma relação.* ■

Definição 5 (Modelo)**.** *Um Modelo,* $\mathcal{M} = \langle \mathcal{F}, \mathcal{V} \rangle$ *, é especificado a partir de um frame* $\mathcal{F} = \langle \mathcal{W}, R \rangle$ *e uma função total binária, denominada por função de valoração,* $\mathcal{V} : \mathcal{P} \times \mathcal{W} \rightarrow$ *{0,1}.* ■

Hilpinen (2006) afirma que a verificação da validade de uma sentença não depende da tabela verdade como visto na lógica proposicional clássica, mas sim da verificação de mundos acessíveis, visto que não é possível construir linhas na tabela para os operadores modais. Uma valoração é dita como valor verdade para cada variável proposicional presente em um mundo possível de $\mathcal{W}$ (GARSON, 2018). Uma fórmula $\Box\varphi$ é verdadeira em um mundo se todos os mundos acessíveis a partir do mundo atual analisado obtiverem φ como verdade, já para $\Diamond\varphi$ ser considerado verdade em um mundo, precisa-se analisar se pelo menos algum mundo acessível possui φ como verdade. A definição de satisfação é estabelecida por:

Definição 6 (Satisfação de fórmula em um mundo)**.** *A relação de satisfação em um certo mundo indica que a fórmula* φ *é válida em um mundo* $w \in \mathcal{W}$ *de um modelo* $\mathcal{M} = \langle \mathcal{F}, \mathcal{V} \rangle$*, denotada por.*

$$\mathcal{M}, w \vDash \varphi \text{ } ou \text{ } w \vDash_{\mathcal{M}} \varphi$$

e satisfaz as seguintes condições.

- $w \nvDash_{\mathcal{M}} \bot$
- $w \vDash_{\mathcal{M}} \top$
- $w \vDash_{\mathcal{M}} p \text{ } sse \text{ } \mathcal{V}(p, w) = 1$
- $w \nvDash_{\mathcal{M}} \varphi \text{ } sse \text{ } Não \text{ } w \vDash_{\mathcal{M}} \varphi$
- $w \vDash_{\mathcal{M}} \neg\varphi \text{ } sse \text{ } w \nvDash_{\mathcal{M}} \varphi$

- $w \vDash_{\mathcal{M}} \varphi \wedge \psi$ *sse* $w \vDash_{\mathcal{M}} \varphi$ *e* $w \vDash_{\mathcal{M}} \psi$
- $w \vDash_{\mathcal{M}} \varphi \vee \psi$ *sse* $w \vDash_{\mathcal{M}} \varphi$ *ou* $w \vDash_{\mathcal{M}} \psi$
- $w \vDash_{\mathcal{M}} \varphi \rightarrow \psi$ *sse* $w \nvDash_{\mathcal{M}} \varphi$ *ou* $w \vDash_{\mathcal{M}} \psi$
- $w \vDash_{\mathcal{M}} \Box\varphi$ *sse* $\forall y \in \mathcal{W}$, $(wRy \rightarrow y \vDash_{\mathcal{M}} \varphi)$
- $w \vDash_{\mathcal{M}} \Diamond\varphi$ *sse* $\exists y \in \mathcal{W}$, $(wRy \wedge y \vDash_{\mathcal{M}} \varphi)$ ■

A noção de satisfazibilidade é estendida para modelos, *frames* e para o sistema lógico com um todo conforme as definições a seguir:

Definição 7 (Satisfazibilidade em um modelo)**.** *Uma fórmula φ é satisfazível em um modelo $\mathcal{M} = \langle \mathcal{W}, R, \mathcal{V} \rangle$, denotado por $\mathcal{M} \vDash \varphi$ se $\forall w \in \mathcal{W}$, $w \vDash \varphi$.* ■

No texto que segue, tanto $\mathcal{M} \vDash \varphi$ quanto a notação $\vDash_{\mathcal{M}} \varphi$ podem ser empregadas para indicar que a fórmula φ é satisfeita em $\mathcal{M}$.

Definição 8 (Validade em um *frame*)**.** *Uma fórmula φ é válida em um frame $\mathcal{F} = \langle \mathcal{W}, R \rangle$, denotado por $\mathcal{F} \vDash \varphi$, se $\mathcal{M} \vDash \varphi$ para todo $\mathcal{M} = \langle \mathcal{W}, R, \mathcal{V} \rangle$.* ■

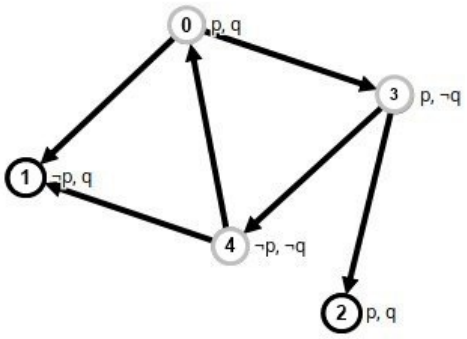
Definição 9 (Fórmula válida)**.** *Uma fórmula φ é válida, denotado por*

$$\vDash \varphi$$

se $\mathcal{F} \vDash \varphi$, para qualquer frame $\mathcal{F} = \langle \mathcal{W}, R \rangle$. ■

Exemplo 1. Seja o modelo $\mathcal{M} = \langle \mathcal{W}, R, \mathcal{V} \rangle$ ilustrado na Figura 2.1, onde os números $0, 1, 2, 3$ e 4 representam respectivamente os mundos $w_0, w_1, w_2, w_3, w_4 \in \mathcal{W}$. A relação de acessibilidade é representada pelas arestas do grafo. Os mundos que possuem um contorno preto também acessam a si próprios. Em cada mundo, usa-se a notação χ para a representação da proposição χ valorada como verdadeira e $\neg\chi$ para indicar sua falsidade.

Figura 2.1: Relação entre mundos



Fonte: Produção do próprio autor.

Sejam as seguintes fórmulas:

1. $\Box p \vee \Box \neg p$
2. $\Box p \vee \neg \Box p$
3. $\Box p$
4. $\Box p \rightarrow \Diamond p$

A primeira fórmula não é satisfeita no mundo w_3, pois analisando a fórmula dada, se tem que $w_2 \vDash p$ e $w_4 \vDash \neg p$, sendo que $w_3 R w_2$ e $w_3 R w_4$. O mesmo se equivale para os mundos w_4 e w_0.

A segunda fórmula é válida para todos os mundos, já que é equivalente à tautologia $\varphi \vee \neg \varphi$, ou seja, qualquer relação de acessibilidade existente entre mundos, há um p ou $\neg p$.

A terceira fórmula é satisfeita apenas no mundo w_2, já que $\forall w' \in \mathcal{W}$, tal que $w_2 R w'$, $w' \vDash p$. Como w_2 se relaciona somente consigo próprio e p é válido neste mundo, então a fórmula é satisfeita.

A quarta fórmula é válida em todos os mundos. A fórmula afirma que, se em um mundo é verdade que necessariamente p, então um mundo acessível por ele também tem que possivelmente p. Pelo observado na 3ª fórmula, todos os mundos exceto w_2 não satisfazem $\Box p$, analisando a tabela verdade da implicação, se o antecedente é falso, então o consequente pode tanto ser verdadeiro ou falso, desta forma, a fórmula torna-se verdadeira. Para o caso de w_2, como $w_2 R w_2$ e $\mathcal{V}(p, w_2) = 1$, então $w_2 \vDash_{\mathcal{M}} \Diamond p$, satisfazendo

a fórmula. Temos, portanto, que $\vDash_{\mathcal{M}} \Box p \rightarrow \Diamond p$. É interessante notar, porém, que a fórmula 4 pode não ser satisfeita em outros modelos.

A partir de um conjunto Γ de fórmulas, define-se a relação de consequência lógica da seguinte maneira:

Definição 10 (Consequência Lógica)**.** *Tem-se que uma fórmula φ é consequência lógica de Γ em um modelo $\mathcal{M}$, denotado por*

$$\Gamma \vDash_{\mathcal{M}} \varphi$$

se e somente se, sempre que todas as fórmulas de Γ forem satisfeitas em $\mathcal{M}$, então φ é satisfeita em $\mathcal{M}$. Ou seja, se para todo $\gamma \in \Gamma$, $\vDash_{\mathcal{M}} \gamma$, então $\vDash_{\mathcal{M}} \varphi$. ∎

2.3 Sistema Dedutivo

Silva, Finger e Melo (2006) expõe que um sistema dedutivo tem o propósito de inferir, derivar ou deduzir novos conjuntos de fórmulas que sejam consequência lógica de um dado conjunto de fórmulas Γ. As regras sintáticas que definem as provas que podem ser obtidas especificam o que chama-se método de prova (ANGELOS, 2016). Há diversos métodos para realizar provas de teoremas, seja por axiomatização, dedução natural ou *tableaux*. O presente trabalho aborda o sistema de Hilbert para a lógica modal na realização de provas apresentadas.

Para a realização de uma prova através do sistema de Hilbert, utiliza-se hipóteses, substituições de axiomas e regras de derivação com o propósito de inferir o objetivo. A substituição é realizada através da troca de uma proposição por uma fórmula: dada uma proposição p em φ, pode-se substituir todas as ocorrências de p por uma fórmula ϱ. Utiliza-se a notação de $\varphi[p := \varrho]$ para indicar tal substituição. Quando uma fórmula ψ é resultante da substituição de um ou mais átomos da fórmula φ, se diz que ψ é uma instância de φ (SILVA; FINGER; MELO, 2006).

Definição 11 (Substituição)**.** *Dadas φ, ψ fórmulas da lógica modal, a substituição de p por ψ, denotada por $\varphi[p := \psi]$ é definida indutivamente por:*

- $p[p := \psi] = \psi$

- $q[p := \psi] = q$ *sse* $q \neq p$
- $(\circ\varphi)[p := \psi] = \circ(\varphi[p := \psi])$, *tal que* $\circ \in \{\neg, \Box, \Diamond\}$.
- $(\varphi_1 \circ \varphi_2)[p := \psi] = (\varphi_1[p := \psi]) \circ (\varphi_2[p := \psi])$, *tal que* $\circ \in \{\wedge, \vee, \rightarrow\}$. ■

Definição 12 (Axiomatização da lógica modal)**.** *A lógica modal possui os seguintes axiomas e conjuntos de regras.*

Ax1 $p \rightarrow (q \rightarrow p)$

Ax2 $(p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$

Ax3 $(\neg q \rightarrow \neg p) \rightarrow (p \rightarrow q)$

Ax4 $p \rightarrow (q \rightarrow (p \wedge q))$

Ax5 $(p \wedge q) \rightarrow p$

Ax6 $(p \wedge q) \rightarrow q$

Ax7 $p \rightarrow (p \vee q)$

Ax8 $q \rightarrow (p \vee q)$

Ax9 $(p \rightarrow r) \rightarrow ((q \rightarrow r) \rightarrow ((p \vee q) \rightarrow r)$

Ax10 $\neg\neg p \rightarrow p$

K $\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$.

Possibilidade $\Diamond(p \vee q) \rightarrow (\Diamond p \vee \Diamond q)$

Modus Ponens *A partir de* φ *e* $\varphi \rightarrow \psi$, *infere-se* ψ

Necessitação *Tendo-se uma prova de* φ, *infere-se* $\Box\varphi$ ■

Os axiomas de 1 a 10 e a regra *Modus Ponens* (MP) são importados da lógica proposicional clássica. Para a distributividade do operador unário $\Box$, se tem o axioma K como especificado, de tal forma que seja φ e $\varphi \rightarrow \psi$ verdadeiros em todos os mundos acessíveis, então haverá ψ verdadeiro também (MALANOVICZ, 2001), já para a distributividade do operador unário $\Diamond$, se tem o axioma da possibilidade, que se refere a existência em pelo menos um mundo acessível das fórmulas φ ou ψ. A regra da necessitação indica

que se uma fórmula é um teorema, então ela deve ser um teorema em todos os mundos possíveis (MALANOVICZ, 2002).

As lógicas mais familiares da família da lógica modal são construídas a partir do sistema mais fraco, denominado como K (GARSON, 2018). Este sistema é considerado o mais fraco da classe de *frames* (BLACKBURN; RIJKE; VENEMA, 2001), no qual não possui restrição a relação de acessibilidade, no caso, qualquer relação satisfaz o sistema.

Exemplo 2. *Seja um sistema da lógica modal que possui o axioma* $\Box p \rightarrow p$*. Deseja-se provar através do método de axiomatização o seguinte sequente.*

$$\Box(p \rightarrow q), \Box(q \rightarrow r) \vdash \Box(p \rightarrow r)$$

1.	$\Box(p \rightarrow q)$	*(premissa)*
2.	$\Box(q \rightarrow r)$	*(premissa)*
3.	$((p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))) \rightarrow ((q \rightarrow r) \rightarrow ((p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))))$	*(Ax1)*
4.	$(p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$	*(Ax2)*
5.	$(q \rightarrow r) \rightarrow ((p \rightarrow (q \rightarrow r)) \rightarrow (p \rightarrow q) \rightarrow (p \rightarrow r))$	*(MP 3, 4)*
6.	$((q \rightarrow r) \rightarrow ((p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r)))) \rightarrow (((q \rightarrow r) \rightarrow (p \rightarrow (q \rightarrow r))) \rightarrow ((q \rightarrow r) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))))$	*(Ax2)*
7.	$((q \rightarrow r) \rightarrow (p \rightarrow (q \rightarrow r))) \rightarrow ((q \rightarrow r) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r)))$	*(MP 5,6)*
8.	$(q \rightarrow r) \rightarrow (p \rightarrow (q \rightarrow r)$	*(Ax1)*
9.	$(q \rightarrow r) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$	*(MP 7,8)*
10.	$\Box(q \rightarrow r) \rightarrow (q \rightarrow r)$	*($\Box p \rightarrow p$)*
11.	$q \rightarrow r$	*(MP 2,10)*
12.	$\Box(p \rightarrow q) \rightarrow (p \rightarrow q)$	*($\Box p \rightarrow p$)*
13.	$(p \rightarrow q)$	*(MP 1,12)*
14.	$(p \rightarrow q) \rightarrow (p \rightarrow r)$	*(MP 11,9)*

15. $p \rightarrow r$ *(MP 13,14)*

16. $\Box(p \rightarrow r)$ *(Necessitação 15)*

2.4 Metapropriedades

Para Angelos (2016), uma prova é um conjunto de regras aplicado a hipóteses com o objetivo de chegar a conclusão de que uma afirmação ou um fato seja verdadeiro. O desenvolvimento da prova de correção e completude possui como objetivo mostrar que se é possível derivar algo, então o fato é verdadeiro e se algo é verdadeiro, então possui uma prova.

Definição 13 (Correção)**.** *Um sistema é dito correto se caso a fórmula* φ *seja derivável em um conjunto de fórmulas* Γ*, então* φ *é consequência lógica de* Γ*, simbolicamente:* $\Gamma \vdash \varphi \Rightarrow \Gamma \vDash \varphi$

■

Teorema 1 (Corretude da lógica modal)**.** *O sistema de Hilbert é correto para a lógica modal.*

Prova: A prova do teorema da corretude possui quatro casos, nos quais consiste em demonstrar que os axiomas, a validação de fórmula em um conjunto de fórmulas, *Modus Ponens* e regra da generalização são satisfeitas no sistema K. O desenvolvimento da prova de correção baseou-se em (COSTA, 1992, p. 58).

Caso 1: α é um axioma da lógica.

Serão provados os dez axiomas de Hilbert e o axioma K, apresentados em 2.3. Adotando φ como Ax1, $\varphi = p \rightarrow (q \rightarrow p)$, será validada a fórmula para todos os mundos que pertencem a qualquer sistema da lógica modal. Esta prova é realizada por contradição, de tal forma que $w \in \mathcal{W}$.

$$w \nvDash p \rightarrow (q \rightarrow p).$$

Não $w \vDash p \rightarrow (q \rightarrow p)$ (Definição do $\nvDash$)

Não ($w \nvDash p$ ou $w \vDash (q \rightarrow p)$) (Semântica da $\rightarrow$)

Não $w \nvDash p$ e Não $w \vDash (q \to p)$	(De Morgan)
Não $w \nvDash p$ e Não ($w \nvDash q$ ou $w \vDash p$)	(Semântica da $\to$)
Não $w \nvDash p$ e Não $w \nVdash q$ e Não $w \vDash p$	(De Morgan)
Não Não $w \vDash p$ e Não Não $w \vDash q$ e Não $w \vDash p$	(Definição do $\nvDash$)
$w \vDash p$ e $w \vDash q$ e Não $w \vDash p$	(Dupla negação)

Como em um mundo $w \in \mathcal{W}$ não pode ter p e $\neg p$, é provado por contradição que $w \vDash p \to (q \to p)$. □

Supondo por contradição que existe um mundo de algum sistema que não satisfaça Ax2, tal que

$$w \nvDash (p \to (q \to r)) \to ((p \to q) \to (p \to r))$$

Não $w \vDash (p \to (q \to r)) \to ((p \to q) \to (p \to r))$	(Definição de $\nvDash$)
Não ($w \nvDash (p \to (q \to r))$ ou $w \vDash ((p \to q) \to (p \to r))$)	(Semântica de $\to$)
Não $w \nvDash p \to (q \to r)^1$ e Não $w \vDash (p \to q) \to (p \to r))^2$	(De Morgan)

Por questão de visualização, a prova será dividida em duas etapas. A primeira etapa é a prova de:

$$\text{Não } w \nvDash p \to (q \to r)$$

Não Não $w \vDash p \to (q \to r)$	(Definição de $\nvDash$)
$w \vDash p \to (q \to r)$	(Dupla negação)
$w \nvDash p$ ou $w \vDash q \to r$	(Semântica de $\to$)
$w \nvDash p$ ou $w \nvDash q$ ou $w \vDash r$	(Semântica de $\to$)

A segunda etapa consiste em:

$$\text{Não } w \vDash (p \to q) \to (p \to r)$$

Não ($w \nvDash p \to q$ ou $w \vDash p \to r$)	(Semântica de $\to$)
Não $w \nvDash p \to q$ e Não $w \vDash p \to r$	(De Morgan)
Não Não $w \vDash p \to q$ e Não $w \vDash p \to r$	(Definição de $\nvDash$)

$w \vDash p \rightarrow q$ e Não $w \vDash p \rightarrow r$	(Dupla Negação)
($w \nvDash p$ ou $w \vDash q$) e Não ($w \nvDash p$ ou $w \vDash r$)	(Semântica de $\rightarrow$)
($w \nvDash p$ ou $w \vDash q$) e Não $w \nvDash p$ e Não $w \vDash r$	(De Morgan)
($w \nvDash p$ ou $w \vDash q$) e Não Não $w \vDash p$ e Não $w \vDash r$	(Definição de $\nvDash$)
($w \nvDash p$ ou $w \vDash q$) e $w \vDash p$ e Não $w \vDash r$	(Dupla Negação)

Unindo a etapa 1 com a etapa 2, se tem que:

$$(w \nvDash p \text{ ou } w \nvDash q \text{ ou } w \vDash r) \text{ e } (w \nvDash p \text{ ou } w \vDash q) \text{ e } w \vDash p \text{ e } w \nvDash r$$

Como não é possível a satisfação das proposições, sem que tenha também sua forma negada, em um mundo qualquer. É provado por contradição que Ax2 é satisfeito em um modelo qualquer. □

Supondo por contradição que qualquer mundo da lógica modal não satisfaça Ax3.

$$w \nvDash (\neg q \rightarrow \neg p) \rightarrow (p \rightarrow q)$$

Não $w \vDash (\neg q \rightarrow \neg p) \rightarrow (p \rightarrow q)$	(Definição de $\nvDash$)
Não ($w \nvDash (\neg q \rightarrow \neg p)$ ou $w \vDash (p \rightarrow q)$)	(Semântica de $\rightarrow$)
Não $w \nvDash (\neg q \rightarrow \neg p)$ e Não $w \vDash (p \rightarrow q)$	(De Morgan)
Não Não $w \vDash (\neg q \rightarrow \neg p)$ e Não $w \vDash (p \rightarrow q)$	(Definição de $\nvDash$)
$w \vDash (\neg q \rightarrow \neg p)$ e Não $w \vDash (p \rightarrow q)$	(Dupla Negação)
($w \nvDash \neg q$ ou $w \vDash \neg p$) e Não ($w \nvDash p$ ou $w \vDash q$)	(Semântica de $\rightarrow$)
($w \nvDash \neg q$ ou $w \vDash \neg p$) e Não $w \nvDash p$ e Não $w \vDash q$	(De Morgan)
(Não $w \vDash \neg q$ ou $w \vDash \neg p$) e Não Não $w \vDash p$ e Não $w \vDash q$	(Definição de $\nvDash$)
(Não $w \nvDash q$ ou $w \nvDash p$) e Não Não $w \vDash p$ e Não $w \vDash q$	(Definição de $\neg$)
(Não $w \nvDash q$ ou $w \nvDash p$) e $w \vDash p$ e Não $w \vDash q$	(Dupla Negação)
(Não Não $w \vDash q$ ou Não $w \vDash p$) e $w \vDash p$ e Não $w \vDash q$	(Definição de $\nvDash$)

($w \vDash q$ ou Não $w \vDash p$) e $w \vDash p$ e Não $w \vDash q$ (Dupla Negação)

Por contradição, é provado que Ax3 é válido em todos os sistemas da lógica modal. □

Seja φ o axioma Ax4 e se deseja provar por contradição que em nenhum mundo da lógica modal é satisfeito.

$$w \nvDash p \rightarrow (q \rightarrow (p \wedge q))$$

Não $w \vDash p \rightarrow (q \rightarrow (p \wedge q)$	(Definição de $\nvDash$)
Não ($w \nvDash p$ ou ($w \vDash q \rightarrow (p \wedge q)$))	(Semântica de $\rightarrow$)
Não ($w \nvDash p$ ou ($w \nvDash q$ ou $w \vDash (p \wedge q)$))	(Semântica de $\rightarrow$)
Não ($w \nvDash p$ ou ($w \nvDash q$ ou ($w \vDash p$ e $w \vDash q$)))	(Semântica da $\wedge$)
Não $w \nvDash p$ e Não $w \nvDash q$ e (Não $w \vDash p$ ou Não $w \vDash q$)	(De Morgan)
Não Não $w \vDash p$ e Não Não $w \vDash q$ e (Não $w \vDash p$ ou Não $w \vDash q$)	(Definição de $\nvDash$)
$w \vDash p$ e $w \vDash q$ e (Não $w \vDash p$ ou Não $w \vDash q$)	(Dupla Negação)

Neste caso temos uma contradição, pois não é permitido uma proposição e sua negação no mesmo mundo.

□

Seja φ o axioma Ax5 e se deseja prová-lo por contradição.

$$w \nvDash (p \wedge q) \rightarrow p$$

Não $w \vDash (p \wedge q) \rightarrow p$	(Definição de $\nvDash$)
Não ($w \vDash \neg(p \wedge q)$ ou $w \vDash p$)	(Semântica da $\rightarrow$)
Não ($w \vDash \neg p \vee \neg q$ ou $w \vDash p$)	(De Morgan)
Não ($w \vDash \neg p$ ou $w \vDash \neg q$ ou $w \vDash p$)	(Definição de $\vee$)
Não ($w \nvDash p$ ou $w \nvDash q$ ou $w \vDash p$)	(Definição de $\nvDash$)
Não (Não $w \vDash p$ ou Não $w \vDash q$ ou $w \vDash p$)	(Definição de $\nvDash$)
Não Não $w \vDash p$ e Não Não $w \vDash q$ e Não $w \vDash p$	(De Morgan)

$w \vDash p$ e $w \vDash q$ e Não $w \vDash p$ (Dupla Negação)

Neste caso, não é possível ter p e $\neg p$ no mesmo mundo, provando-o a contradição.

□

Seja φ o axioma Ax6, a prova segue o mesmo caminho do axioma Ax5. Será provado por contradição.

$$w \nvDash (p \wedge q) \rightarrow q$$

Não $w \vDash (p \wedge q) \rightarrow q$	(Definição de $\nvDash$)
Não ($w \vDash \neg(p \wedge q)$ ou $w \vDash q$)	(Semântica da $\rightarrow$)
Não ($w \vDash \neg p \vee \neg q$ ou $w \vDash q$)	(De Morgan)
Não ($w \vDash \neg p$ ou $w \vDash \neg q$ ou $w \vDash q$)	(Definição de $\vee$)
Não ($w \nvDash p$ ou $w \nvDash q$ ou $w \vDash q$)	(Definição de $\nvDash$)
Não (Não $w \vDash p$ ou Não $w \vDash q$ ou $w \vDash q$)	(Definição de $\nvDash$)
Não Não $w \vDash p$ e Não Não $w \vDash q$ e Não $w \vDash q$	(De Morgan)
$w \vDash p$ e $w \vDash q$ e Não $w \vDash q$	(Dupla Negação)

Neste caso, não é possível ter q e $\neg q$ no mesmo mundo, provando-o a contradição.

□

Seja φ o axioma Ax7. A prova é feita por contradição, onde existe um mundo qualquer na lógica que não satisfaça este axioma.

$$w \nvDash p \rightarrow (p \vee q)$$

Não $w \vDash p \rightarrow (p \vee q)$	(Definição de $\nvDash$)
Não ($w \nvDash p$ ou $w \vDash (p \vee q)$)	(Semântica de $\rightarrow$)
Não ($w \nvDash p$ ou ($w \vDash p$ ou $w \vDash q$))	(Semântica de $\vee$)
Não $w \nvDash p$ e Não $w \vDash p$ e Não $w \vDash q$	(De Morgan)

Como não pode ocorrer uma contradição de proposição em qualquer mundo, se tem a prova que o axioma Ax7 é satisfeito em qualquer mundo.

□

Seja φ o axioma Ax8. A prova é feita por contradição, onde existe um mundo qualquer na lógica que não satisfaça este axioma.

$$w \nvDash q \rightarrow (p \vee q)$$

Não $w \vDash q \rightarrow (p \vee q)$	(Definição de $\nvDash$)
Não ($w \nvDash q$ ou $w \vDash (p \vee q)$)	(Semântica de $\rightarrow$)
Não ($w \nvDash q$ ou ($w \vDash p$ ou $w \vDash q$))	(Semântica de $\vee$)
Não $w \nvDash q$ e Não $w \vDash p$ e Não $w \vDash q$	(De Morgan)

Como não pode ocorrer uma contradição de proposição em qualquer mundo, se tem a prova que o axioma Ax8 é satisfeito em qualquer mundo.

□

Seja φ o axioma Ax9. A prova é realizada por contradição para demonstrar que satisfaz o axioma em qualquer mundo.

$$w \nvDash (p \rightarrow r) \rightarrow ((q \rightarrow r) \rightarrow ((p \vee q) \rightarrow r))$$

Não $w \vDash (p \rightarrow r) \rightarrow ((q \rightarrow r) \rightarrow ((p \vee q) \rightarrow r))$	(Definição de $\nvDash$)
Não ($w \vDash \neg(p \rightarrow r)$ ou $w \vDash ((q \rightarrow r) \rightarrow ((p \vee q) \rightarrow r))$	(Semântica da $\rightarrow$)
Não ($w \vDash \neg(p \rightarrow r)$ ou ($w \vDash \neg(q \rightarrow r)$ ou $w \vDash ((p \vee q) \rightarrow r))$	(Definição de $\nvDash$)
(i) Não $w \vDash \neg(p \rightarrow r)$ e (ii) Não $w \vDash \neg(q \rightarrow r)$ e (iii) Não $w \vDash ((p \vee q) \rightarrow r))$	(De Morgan)

O desenvolvimento da prova sera divido em três subprovas para uma melhor visualização. Por (i) se tem que.

Não $w \vDash \neg(\neg p \vee r)$	(Semântica da $\rightarrow$)
Não $w \vDash \neg\neg p \wedge \neg r$	(De Morgan)
Não ($w \vDash \neg\neg p$ e $w \vDash \neg r$)	(Semântica de $\wedge$)

Não $w \vDash \neg\neg p$ ou Não $w \vDash \neg r$	(De Morgan)
Não Não Não $w \vDash p$ ou Não Não $w \vDash r$	(Definição de $\nvDash$)
Não $w \vDash p$ ou $w \vDash r$	(Dupla Negação)

Por (ii) tem que.

$$\text{Não } w \vDash \neg(q \to r)$$

Não $w \vDash \neg(\neg q \vee r)$	(Semântica de $\to$)
Não $w \vDash \neg\neg q \wedge \neg r)$	(De Morgan)
Não $w \vDash q \wedge \neg r$	(Dupla Negação)
Não ($w \vDash q$ e $w \vDash \neg r$)	(Semântica de $\wedge$)
Não $w \vDash q$ ou Não $w \vDash \neg r$)	(De Morgan)
Não $w \vDash q$ ou Não Não $w \vDash r$)	(Definição de $\nvDash$)
Não $w \vDash q$ ou $w \vDash r$	(Dupla Negação)

Por fim, em (iii) apresenta-se.

$$\text{Não } w \vDash ((p \vee q) \to r))$$

Não $w \vDash \neg(p \vee q) \vee r$	(Semântica da $\to$)
Não $w \vDash (\neg p \wedge \neg q) \vee r$	(De Morgan)
Não ($w \vDash (\neg p \wedge \neg q)$ ou $w \vDash r$)	(Semântica da $\vee$)
Não $w \vDash (\neg p \wedge \neg q)$ e Não $w \vDash r$	(De Morgan)
Não ($w \vDash \neg p$ e $w \vDash \neg q$) e Não $w \vDash r$	(Semântica da $\wedge$)
(Não $w \vDash \neg p$ ou Não $w \vDash \neg q$) e Não $w \vDash r$	(De Morgan)
(Não Não $w \vDash p$ ou Não Não $w \vDash q$) e Não $w \vDash r$	(Definição de $\nvDash$)
($w \vDash p$ ou $w \vDash q$) e Não $w \vDash r$	(Dupla Negação)

Ao unir as subprovas (i), (ii), (iii) obtém-se.

($w \nvDash p$ ou $w \vDash r$) e ($w \nvDash q$ ou $w \vDash r$) e (($w \vDash p$ ou $w \vDash q$) e $w \nvDash r$)

Independente do valor da proposição no mundo w sempre ocorrerá uma contradição, o que prova a validade do axioma em todos os mundos $w \in \mathcal{W}$.

□

Seja φ o axioma Ax10 e se deseja prová-lo com contradição, ou seja, não existe um mundo na lógica que satisfaça tal axioma.

$$w \not\models \neg\neg p \rightarrow p$$

Não $w \vDash \neg\neg p \rightarrow p$	(Definição de $\not\models$)
Não ($w \vDash \neg\neg\neg p$ ou $w \vDash p$)	(Semântica de $\rightarrow$)
Não $w \vDash \neg\neg\neg p$ e Não $w \vDash p$	(De Morgan)
Não Não Não Não $w \vDash p$ e Não $w \vDash p$	(Definição de $\not\models$)
Não Não $w \vDash p$ e Não $w \vDash p$	(Dupla Negação)
$w \vDash p$ e Não $w \vDash p$	(Dupla Negação)

Onde não é possível um mundo satisfazer p e $\neg p$, é provado por contradição que o axioma Ax10 é satisfeito em qualquer mundo $w \in \mathcal{W}$.

□

Seja φ o axioma K, prova-se por contradição que existe um mundo qualquer do sistema K, tal que $w \in \mathcal{W}$, em que:

$$w \not\models \Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$$

Não $w \vDash \Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$	(Definição de $\not\models$)
Não ($w \not\models \Box(p \rightarrow q)$ ou $w \vDash \Box p \rightarrow \Box q$)	(Semântica da $\rightarrow$)
Não $w \not\models \Box(p \rightarrow q)$ e Não $w \vDash \Box p \rightarrow \Box q$)	(De Morgan)
Não Não $w \vDash \Box(p \rightarrow q)$ e Não $w \vDash \Box p \rightarrow \Box q$	(Definição da $\not\models$)
$w \vDash \Box(p \rightarrow q)$ e Não $w \vDash \Box p \rightarrow \Box q$	(Dupla negação)
$w \vDash \Box(p \rightarrow q)$ e Não ($w \not\models \Box p$ ou $\Box q$)	(Semântica da $\rightarrow$)
$w \vDash \Box(p \rightarrow q)$ e Não $w \not\models \Box p$ e Não $w \vDash \Box q$	(De Morgan)
$w \vDash \Box(p \rightarrow q)$ e Não Não $w \vDash \Box p$ e Não $w \vDash \Box q$	(Definição da $\not\models$)
$w \vDash \Box(p \rightarrow q)$ e $w \vDash \Box p$ e Não $w \vDash \Box q$	(Dupla negação)

Este momento da prova será dividido em três componentes.

O primeiro componente é $w \vDash \Box(p \to q)$. Pela definição de $\Box$, se tem:

Para todo $w' \in \mathcal{W}$, tal que wRw', $w' \vDash p \to q$

$w' \nvDash p$ ou $w' \vDash q$ (Semântica de $\to$)

O segundo componente consiste em $w \vDash \Box p$. Pela definição de $\Box$, é demonstrado que:

Para todo $w'' \in \mathcal{W}$, tal que wRw'', $w'' \vDash p$

O terceiro contém Não $w \vDash \Box q$. Pela definição de $\Box$, define-se que:

Para todo $w''' \in \mathcal{W}$, tal que wRw''', $w''' \vDash q$

Conforme apresentado anteriormente, a definição de $\Box$ se diz que para todos os mundos que w se relaciona, tem a proposição verdadeira, logo, a prova do primeiro componente nega o segundo e o terceiro componente, chegando no fim a uma contradição. Desta forma, é provado que o axioma K valida-se em todos os modelos possíveis.

□

Seja φ o axioma da possibilidade e o propósito da prova é por contradição.

$$w \nvDash \Diamond(\varphi \vee \psi) \to (\Diamond\varphi \vee \Diamond\psi)$$

Não $w \vDash \Diamond(\varphi \vee \psi) \to (\Diamond\varphi \vee \Diamond\psi)$ (Definição de $\nvDash$)

Não ($w \nvDash \Diamond(\varphi \vee \psi)$ ou $w \vDash (\Diamond\varphi \vee \Diamond\psi)$) (Semântica de $\to$)

Não $w \nvDash \Diamond(\varphi \vee \psi)$ e Não $w \vDash (\Diamond\varphi \vee \Diamond\psi)$ (De Morgan)

Não Não $w \vDash \Diamond(\varphi \vee \psi)$ e Não $w \vDash (\Diamond\varphi \vee \Diamond\psi)$ (Definição de $\nvDash$)

$w \vDash \Diamond(\varphi \vee \psi)$ e Não $w \vDash (\Diamond\varphi \vee \Diamond\psi)$ (De Morgan)

$w \vDash \Diamond(\varphi \vee \psi)$ e Não ($w \vDash \Diamond\varphi$ ou $w \vDash \Diamond\psi$) (Semântica de $\vee$)

(i) $w \vDash \Diamond(\varphi \vee \psi)$ e (ii) Não $w \vDash \Diamond\varphi$ e
(iii) Não $w \vDash \Diamond\psi$ (De Morgan)

Ao analisar a subprova (ii), obtém-se:

Não $w \vDash \Diamond\varphi$

$w \nvDash \Diamond\varphi$ (Definição de $\nvDash$)

$w \vDash \neg\Diamond\varphi$ (Definição de $\neg$)

$w \vDash \Box\neg\varphi$ (Equivalência de $\Box$)

Pela definição de $\Box$, se tem que $\forall w' \in \mathcal{W},$ tal que $w\mathcal{R}w', w' \vDash \neg\varphi$

A subprova (iii) tem:

$$\text{Não } w \vDash \Diamond\psi$$

$w \nvDash \Diamond\psi$ (Definição de $\nvDash$)

$w \vDash \neg\Diamond\psi$ (Definição de $\neg$)

$w \vDash \Box\neg\psi$ (Equivalência de $\Box$)

Pela definição de $\Box$, se tem que $\forall w' \in \mathcal{W},$ tal que $w\mathcal{R}w', w' \vDash \neg\psi$

A subprova (i) pode ser visto como $\exists w' \in \mathcal{W},$ tal que $w\mathcal{R}w', w' \vDash (\varphi \vee \psi)$. O que contradiz as subprovas (ii) e (iii), pois não é possível um mundo ter uma fórmula e sua negação.

Caso 2: Se uma fórmula φ pertence ao conjunto de fórmulas Γ, então esta fórmula é consequência lógica do conjunto de fórmulas.

Se $\varphi \in \Gamma$, então $\Gamma \vDash \varphi$. □

Caso 3: A prova de que o *Modus Ponens* (MP) é consistente em todos os sistemas normais possíveis é dada na seguinte forma. Como deseja-se provar que β é consequência lógica de α e $\alpha \to \beta$, se tem que $w \vDash \alpha$ e $w \vDash \alpha \to \beta$, então deseja-se provar que $w \vDash \beta$, de tal forma que $w \in \mathcal{W}$.

$$w \vDash (\alpha \to \beta)$$

$w \nvDash \alpha$ ou $w \vDash \beta$ (Semântica de $\to$)

Como na hipótese de indução se tem $w \vDash \alpha$, não é possível ter sua forma negada, por conseguinte, é possível deduzir que $w \vDash \beta$ através de $w \vDash \alpha \to \beta$, como demonstrado acima. □

Caso 4: A prova da regra da generalização se da ao fato de como $\varphi \in \Gamma$ satisfaz todos os mundos de um modelo qualquer $\mathcal{M} = \langle \mathcal{W}, R, \mathcal{V} \rangle$, onde $\mathcal{M} \vdash \varphi$. Seja um mundo $w \in \mathcal{W}, w \vdash \varphi$, pela relação de acessibilidade, se tem que:

$\forall w' \in \mathcal{W}$, tal que wRw', onde $w' \vdash \varphi$.

Utilizando a definição de $\Box$ que é tal que todos os mundos que w se relaciona φ é verdade, então $w \vdash \Box\varphi$. Logo, $\Gamma \vDash \Box\varphi$. $\Box$

Definição 14 (Completude)**.** *Um sistema é dito completo se caso φ é consequência lógica de Γ, então existe φ derivação de φ a partir de Γ.*

$$\Gamma \vDash \varphi \Rightarrow \Gamma \vdash \varphi$$

Teorema 2 (Completude da Lógica Modal)**.** *O sistema de Hilbert para a lógica modal é completo.*

O desenvolvimento da prova da completude é realizado através de modelos canônicos usando o lema de Lindenbaum, a prova é feita para o sistema K, ou seja, o sistema mais genérico da lógica modal e baseou-se em (BLACKBURN; RIJKE; VENEMA, 2001, p. 202).

Definição 15 (Consistência)**.** *Um conjunto de fórmulas Λ é dito consistente se não fere o princípio da não contradição, ou seja, $\Lambda \nvdash \bot$.*

Definição 16 (Maximal Consistente)**.** *O conjunto maximal consistente Λ^+ de um conjunto de fórmulas consistentes Λ é tal que, para qualquer fórmula φ, tem-se $\varphi \in \Lambda^+$ ou $\neg\varphi \in \Lambda^+$ e Λ^+ é consistente.*

Lema 1. *Se Δ é consistente e $\Gamma \subseteq \Delta$, então Γ é consistente.*

Prova: Seja Δ consistente e $\Gamma \subseteq \Delta$. Supomos que Γ é inconsistente, ou seja, existem $\varphi_1, \ldots, \varphi_n$ tais que $\{\varphi_1, \ldots, \varphi_n\} \subseteq \Gamma$ e $\vdash_\Gamma \neg(\varphi_1 \wedge \ldots \wedge \varphi_n)$. Como $\Gamma \subseteq \Delta$, temos que existem $\varphi_1, \ldots, \varphi_n \in \Delta$, tal que $\vdash_\Delta \neg(\varphi_1 \wedge \ldots \wedge \varphi_n)$, tornando-o inconsistente, contradizendo a hipótese inicial.

$\Box$

Lema 2 (Lema de Lindenbaum)**.** *Dado um conjunto de fórmulas Γ consistente, existe um Δ maximal consistente tal que $\Gamma \subseteq \Delta$.*

Prova: A partir de Γ, mostra-se a existência de um Δ maximal consistente em que $\Gamma \subseteq \Delta$.

Para o desenvolvimento da prova, assume-se um conjunto fixo e enumerável de todas as fórmulas, $\{\sigma_1, \sigma_2, \ldots\}$ que define uma sequência de conjuntos de fórmulas $\Delta_0, \Delta_1, \ldots$ a partir de Γ.

Definição 17. *Dada enumeração de fórmulas* $\{\sigma_1, \sigma_2, \ldots\}$, Δ *é um conjunto de fórmulas definido de forma que:*

1. $\Delta_0 = \Gamma$
2. $\Delta_n = \begin{cases} \Delta_{n-1} \cup \{\sigma_n\}, \textit{ se } \Delta_{n-1} \cup \{\sigma_n\} \textit{ for consistente} \\ \Delta_{n-1}, \textit{ caso contrario} \end{cases}$
3. $\Delta = \bigcup_{n=0}^{\infty} \Delta_n$

Lema 3. Δ_n *é consistente, sendo* $0 \leq n$.

Prova:

Caso n = 0, temos que Γ é consistente.

Caso n > 0, com base na Definição 17.2, Δ_n é construído a partir de um conjunto consistente Δ_{n-1}.

Lema 4. $\Delta_n \subseteq \Delta$, *para* $0 \leq n$.

Prova: Como apresentado pela Definição 17.3, Δ se apresenta como a união de todos os conjuntos Δ_n, onde $n \in \mathbb{N}$. Logo, seja qualquer valor de n, $\Delta_n \subseteq \Delta$.

Lema 5. $\Gamma \subseteq \Delta$

Prova: Como em 17.1 se tem $\Gamma \subseteq \Delta_0$, e pelo Lema 4 tem-se $\Delta_0 \subseteq \Delta$, então $\Gamma \subseteq \Delta$.

Assim, prova-se que Δ é uma extensão de Γ.

A prova de que Γ é maximal necessita dos seguintes lemas.

Lema 6. $\Delta_k \subseteq \Delta_n$, *onde* $0 \leq k \leq n$.

Prova: Seja um valor qualquer de k, onde $0 \leq k \leq n$, pela Definição 17.2, um conjunto Δ_n é construído pelo conjunto de seus predecessores. Logo, para qualquer valor $k \leq n$, $\Delta_k \subseteq \Delta_n$.

Lema 7. *$\sigma_k \in \Delta_k$, sempre que $\sigma_k \in \Delta$ para algum $0 < k$.*

Prova: Supondo $\sigma_k \in \Delta$ e $\sigma_k \notin \Delta_k$. Pelo Lema 3, temos que Δ_{k-1} é consistente. Como $\sigma_k \notin \Delta_k$, pela definição 17.2 temos que $\Delta_{k-1} \bigcup \{\sigma_k\}$ não é consistente e $\Delta_k = \Delta_{k-1}$. Para qualquer $k < n$ temos, pelo Lema 3 que Δ_n é consistente, logo $\sigma_k \notin \Delta_n$. Portanto $\sigma_k \notin \Delta$, o que contradiz a hipótese inicial. Logo, $\sigma_k \in \Delta$.

Lema 8. *Qualquer subconjunto finito Δ' de Δ, se tem que $\Delta' \subseteq \Delta_n$, para algum $0 \leq n$.*

Prova: Como Δ' é finito, a partir da enumeração $\sigma_1, \sigma_2, \dots$ de fórmulas utilizada na construção de Δ, há uma fórmula $\sigma_n \in \Delta'$ tal que $k \leq n$ para qualquer $\sigma_k \in \Delta'$, como $\Delta' \subseteq \Delta$, temos que $\sigma_n \in \Delta$ e, portanto, pelo Lema 7, $\sigma_n \in \Delta_n$. Qualquer outro elemento $\sigma_k \in \Delta'$ se tem $k \leq n$, pelo Lema 7 $\sigma_k \in \Delta_k$ e Lema 6, $\Delta_k \subseteq \Delta_n$, logo todos os elementos de Δ' pertencem a Δ_n.

Lema 9. *Δ é maximal consistente, isto é, (i) Δ é consistente e (ii) para cada σ se $\Delta \cup \{\sigma\}$ é consistente, então $\sigma \in \Delta$.*

Prova (i): Supondo que Δ tem um subconjunto inconsistente denominado Δ'. Seja um $\sigma_n \in \Delta'$, onde n é o maior termo do conjunto, pelo Lema 8, temos que $\Delta' \subseteq \Delta_n$, logo Δ_n é inconsistente, o que contradiz o Lema 1.

Prova (ii): Supondo $\Delta \cup \{\sigma\}$ consistente. Existe um $n \in \mathbb{N}$, no conjunto $\{\sigma_1, \sigma_2, \dots\}$ usado para a construção de Δ, $\sigma = \sigma_n$. Pelo Lema 1, qualquer subconjunto de Δ consistente, também é consistente. Pela Definição 17.2, se tem $\Delta_n = \Delta_{n-1} \cup \{\sigma_n\}$. Então $\sigma_n \in \Delta_n$, como $\Delta_n \subseteq \Delta$, logo, $\sigma \in \Delta$.

□

Definição 18. *O modelo canônico $\mathcal{M}^\Lambda$ para uma lógica modal Λ é dado como uma tripla $(\mathcal{W}^\Lambda, \mathcal{R}^\Lambda, \mathcal{V}^\Lambda)$.*

1. $\mathcal{W}^\Lambda$ é um conjunto de todos os maximais consistentes de Λ.
2. $\mathcal{R}^\Lambda$ é uma relação binária sobre $\mathcal{W}^\Lambda$ denominada de relação canônica, onde $w\mathcal{R}^\Lambda w'$ se para todas as fórmulas φ, temos que $\varphi \in w' \implies \Diamond\varphi \in w$.
3. $\mathcal{V}^\Lambda$, chamada valoração canônica, é definida por $\mathcal{V}^\Lambda(p, w) = \begin{cases} 1, \text{ se } p \in w \\ 0, \text{ caso contrário} \end{cases}$.

O Frame canônico para Λ é dado por $\mathcal{F}^{\Lambda} = (\mathcal{W}^{\Lambda}, \mathcal{R}^{\Lambda})$.

Lema 10. *Para qualquer lógica normal* Λ, $w\mathcal{R}^{\Lambda}w' \Longleftrightarrow (\forall\varphi, \Box\varphi \in w \rightarrow \varphi \in w')$.

Prova:

($\Rightarrow$) Supondo $w\mathcal{R}^{\Lambda}w'$ e $\varphi \notin w'$. Como w' é um conjunto maximal consistente, temos que $\neg\varphi \in w'$. Como $w\mathcal{R}w'$, pela Definição 18.2, temos $\Diamond\neg\varphi \in w$.Como w é maximal consistente temos $\neg\Diamond\neg\varphi \notin w$. Pela dualidade dos operadores modais demonstramos que $\Box\varphi \notin w$. Neste caso está provado por contraposição.

($\Leftarrow$) Supondo ($\Box\varphi \in w \rightarrow \varphi \in w'$) e $\Diamond\varphi \notin w$.Como w é maximal consistente, temos $\neg\Diamond\varphi \in w$. Pela dualidade dos operadores modais, $\Box\neg\varphi \in w$, aplicando a hipótese temos $\neg\varphi \in w'$, ou seja, $\varphi \notin w'$, uma vez que w' é maximal consistente. Logo, $\Diamond\varphi \notin w \rightarrow \varphi \notin w'$, por contraposição, esta é a definição 18.2, portanto $w\mathcal{R}^{\Lambda}w'$.

□

Lema 11 (Lema Existencial)**.** *Dada* Λ *lógica modal normal e* $w \in \mathcal{W}^{\Lambda}$, *se* $\Diamond\varphi \in w$ *então existe* $w \in \mathcal{W}^{\Lambda}$ *tal que* $w\mathcal{R}^{\Lambda}w'$ *e* $\varphi \in w'$.

Prova: Suponhamos $\Diamond\varphi \in w$. Então o conjunto $v = \{\varphi\} \cup \{\psi | \Box\psi \in w\}$ é consistente. Para demonstrar tal fato, vamos supor que v' é inconsistente, então há $\psi_1, \ldots, \psi_n$ tais que

$$\vdash_{\Lambda} (\psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_n) \rightarrow \neg\varphi \qquad \text{(Hipótese)}$$

$$\vdash_{\Lambda} \Box((\psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_n) \rightarrow \neg\varphi) \qquad \text{(Necessitação)}$$

$$\vdash_{\Lambda} \Box(\psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_n) \rightarrow \Box\neg\varphi \qquad \text{(Axioma K + MP)}$$

$$\vdash_{\Lambda} (\Box\psi_1 \wedge \Box\psi_2 \wedge \ldots \wedge \Box\psi_n) \rightarrow \Box\neg\varphi \qquad \text{(Distributiva do } \Box \text{ para } \wedge)$$

Como $\Box\psi_i \in w$ para $1 \leq i \leq n$, temos que $\Box\psi_1 \wedge \cdots \wedge_n \in w$, portanto $\Box\neg\varphi \in w$. Pela dualidade dos operadores modais, $\neg\Diamond\varphi \in w$ o que não é possível pois w é maximal consistente e $\Diamond\varphi \in w$. Logo v' é consistente.

Seja v uma extensão maximal consistente de v', garantida pelo Lema 2. Então temos que $\varphi \in v$. Além disso, para qualquer fórmula ψ, se $\Box\psi \in w$ então $\psi \in v$. pelo Lema 10, tem-se então $w\mathcal{R}^{\Lambda}v$.

□

Lema 12 (Lema da Verdade)**.** *Para qualquer lógica modal normal* Λ *e qualquer fórmula* φ, *tem-se,* $\mathcal{M}^{\Lambda}, w \vDash \varphi \Longleftrightarrow \varphi \in w$.

Prova: A prova é realizada por indução de tamanho φ.

Base: $\varphi = p$ onde p é uma proposição atômica. Pela definição de $\mathcal{M}^\Lambda$ temos que $\mathcal{V}^\Lambda(p, w) = 1 \Longleftrightarrow p \in w$.

Hipótese de Indução: Seja φ com tamanho $1 \leq n$ então $\mathcal{M}^\Lambda, w \vDash \varphi \Longleftrightarrow \varphi \in w$.

Passo: Temos 6 casos a analisar.

1. **A fórmula φ é do tipo $\neg\psi$:**

   ($\Rightarrow$) Supondo $\mathcal{M}^\Lambda, w \vDash \neg\psi$ então $\mathcal{M}^\Lambda, w \nvDash \psi$ pela hipótese de indução $\psi \notin w$, como w é maximal, $\neg\psi \in w$.

   ($\Leftarrow$) Supondo $\neg\psi \in w$, como w é maximal consistente temos $\psi \notin w$. Pela hipótese de indução $\mathcal{M}^\Lambda, w \nvDash \psi$, logo $\mathcal{M}^\Lambda, w \vDash \neg\psi$.

2. **A fórmula φ é do tipo $\psi_1 \to \psi_2$:**

   ($\Rightarrow$) Supondo $\mathcal{M}^\Lambda, w \vDash \psi_1 \to \psi_2$ então $\mathcal{M}^\Lambda, w \nvDash \psi_1$ ou $\mathcal{M}^\Lambda, w \nvDash \psi_2$. Caso $\mathcal{M}^\Lambda, w \nvDash \psi_1$ pela hipótese de indução $\psi_1 \notin w$, logo $\neg\psi_1 \in w$ e como w é maximal consistente, $(\psi_1 \to \psi_2) \in w$. Caso $\mathcal{M}^\Lambda, w \vDash \psi_2$ pela hipótese de indução $\psi_2 \in w$, portanto $(\psi_1 \to \psi_2) \in w$, pois w é maximal consistente.

   ($\Leftarrow$) Supondo $(\psi_1 \to \psi_2) \in w$ então como w é maximal consistente temos $\psi_1 \notin w$ ou $\psi_2 \in w$. Caso $\psi_1 \notin w$, pela hipótese de indução temos $\mathcal{M}^\Lambda, w \nvDash \psi_1$, logo $\mathcal{M}^\Lambda, w \vDash (\psi_1 \to \psi_2)$. Caso $\psi_2 \in w$, pela hipótese de indução tem-se $\mathcal{M}^\Lambda, w \vDash \psi_2$, logo $\mathcal{M}^\Lambda, w \vDash \psi_1 \to \psi_2$.

3. **A fórmula φ é do tipo $\psi_1 \wedge \psi_2$:**

   ($\Rightarrow$) Supondo $\mathcal{M}^\Lambda, w \vDash \psi_1 \wedge \psi_2$, portanto $\mathcal{M}^\Lambda, w \vDash \psi_1$ e $\mathcal{M}^\Lambda, w \vDash \psi_2$. Pela hipótese de indução, $\psi_1 \in w$ e $\psi_2 \in w$, como w é maximal consistente $(\psi_1 \wedge \psi_2) \in w$.

   ($\Leftarrow$) Supondo $(\psi_1 \wedge \psi_2) \in w$, como w é maximal consistente temos $\psi_1 \in w$ e $\psi_2 \in w$. Pela hipótese de indução $\mathcal{M}^\Lambda, w \vDash \psi_1$ e $\mathcal{M}^\Lambda, w \vDash \psi_2$ logo $\mathcal{M}^\Lambda, w \vDash \psi_1 \wedge \psi_2$.

4. **A fórmula φ é do tipo $\psi_1 \vee \psi_2$:**

   ($\Rightarrow$) Supondo $\mathcal{M}^\Lambda, w \vDash \psi_1 \vee \psi_2$, portanto $\mathcal{M}^\Lambda, w \vDash \psi_1$ ou $\mathcal{M}^\Lambda, w \vDash \psi_2$, vamos analisar ambos os casos.

   (a) $\mathcal{M}^\Lambda, w \vDash \psi_1$. Neste caso, pela hipótese de indução, $\psi_1 \in w$. Como w é maximal consistente temos $(\psi_1 \vee \psi_2) \in w$

(b) $\mathcal{M}^\Lambda, w \vDash \psi_2$ é análogo a (a), o que leva a $(\psi_1 \vee \psi_2) \in w$.

($\Leftarrow$) Supondo que $(\psi_1 \vee \psi_2) \in w$, como w é maximal consistente temos $\psi_1 \in w$ ou $\psi_2 \in w$.

(a) Caso $\psi_1 \in w$ então pela hipótese de indução tem-se que $\mathcal{M}^\Lambda, w \vDash \psi_1$, logo $\mathcal{M}^\Lambda, w \vDash \psi_1 \vee \psi_2$.

(b) Caso $\psi_2 \in w$ então pela hipótese de indução tem-se que $\mathcal{M}^\Lambda, w \vDash \psi_2$, logo $\mathcal{M}^\Lambda, w \vDash \psi_1 \vee \psi_2$.

Portanto $\mathcal{M}^\Lambda, w \vDash \psi_1 \vee \psi_2$.

5. **A fórmula φ é do tipo $\Diamond\psi$:**

($\Rightarrow$) Seja $\mathcal{M}^\Lambda, w \vDash \Diamond\psi$.

$\exists w'(w\mathcal{R}^\Lambda w' \wedge \mathcal{M}^\Lambda, w' \vDash \psi)$ (Definição de Possibilidade)

$\exists w'(w\mathcal{R}^\Lambda w' \wedge \psi \in w')$ (Hipótese de Indução)

$\Diamond\psi \in w$ (Definição de $\mathcal{R}^\Lambda$)

($\Leftarrow$) Seja $\Diamond\psi \in w$, pelas equivalências acima basta encontrar um conjunto maximal consistente, tal que $w\mathcal{R}^\Lambda w'$ e $\psi \in w'$. Pelo Lema 11 é provado.

6. **A fórmula φ é do tipo $\Box\psi$:**

($\Rightarrow$) Supondo $\mathcal{M}^\Lambda, w \vDash \Box\psi$ e $\Box\psi \notin w$. Então como w é maximal consistente, $\neg\Box\psi \in w$. Pela dualidade dos operadores modais $\Diamond\neg\psi \in w$. Pelo Lema 11 existe $w' \in \mathcal{W}^\Lambda$ em que $w\mathcal{R}^\Lambda w'$ e $\neg\psi \in w'$. Porém como $\mathcal{M}^\Lambda, w \vDash \Box\psi$ temos que para qualquer $w' \in \mathcal{W}^\Lambda$ tal que $w\mathcal{R}^\Lambda w'$, $\mathcal{M}^\Lambda, w' \vDash \psi$. Pela hipótese de indução $\psi \in w'$ para todo w' em que $w\mathcal{R}^\Lambda w'$. Temos que $\psi \in w'$ e $\neg\psi \in w'$, o que é absurdo, já que w' é consistente. Portanto $\Box\psi \in w$.

($\Leftarrow$) Supondo $\Box\psi \in w$, pelo Lema 10, para todo $w' \in \mathcal{W}^\Lambda$ tal que $w\mathcal{R}^\Lambda w'$ tem-se $\psi \in w'$. Pela hipótese de indução, $\mathcal{M}^\Lambda, w \vDash \psi$ para todo w' em que $w\mathcal{R}^\Lambda w'$, logo $\mathcal{M}^\Lambda, w \vDash \Box\psi$.

□

Teorema 3 (Teorema dos Modelos Canônicos)**.** *Qualquer lógica modal normal é fortemente completa em relação aos modelos canônicos.*

Prova: Supondo Δ um conjunto consistente de uma lógica modal Λ. Pelo Lema 2 tem-se $\Delta \subseteq \Delta^+$. Pelo Lema 12 prova-se que $\mathcal{M}^\Lambda, \Delta^+ \vDash \Delta$.

□

2.5 Sistemas da Lógica Modal

A relação de acessibilidade possui fundamental importância na semântica da lógica modal. Para verificar a validade de uma fórmula eventualmente se observa a relação de acessibilidade daquele sistema. A partir das restrições impostas sobre a relação de acessibilidade, obtém-se diferentes sistemas modais (MALANOVICZ, 2001). Cada sistema da lógica modal possui uma classe de *frames* diferente, onde estes *frames* são construídos através da relação de acessibilidade e suportam os axiomas e regras do sistema K. Os axiomas, quando válidos em uma estrutura de Kripke, implicam em determinadas propriedades das suas relações de acessibilidade (SILVA, 2013), estas nas quais podem ser reflexivas, transitivas, euclidianas, funcionais, entre outras. Diferentes propriedades acerca da relação de acessibilidade definem diferentes sistemas de lógica modal. Deste modo, tem-se a definição de sistemas como $D, T, S4$, etc.

Um axioma que satisfaz um determinado sistema não necessariamente deverá satisfazer um outro sistema, porém existem sistemas que estendem outros, por conseguinte, podem utilizar dois ou mais axiomas que caracterizam a relação de acessibilidade.

Diferentes *frames* possuem axiomas que são satisfeitos em $\mathcal{W}$, estes axiomas são definidos através da relação de acessibilidade, logo, uma relação específica possui um axioma específico. A Tabela 2.2 apresenta algumas relações presentes na lógica modal.

Seja o sistema modal D, constituído pelos axiomas K e $D = \Box p \rightarrow \Diamond p$. Os *frames* para tal sistema são aqueles que respeitam a propriedade de serialidade, ou seja para cada mundo existe um próximo mundo acessível. Voltando ao Exemplo 1, o axioma do sistema D é válido em todos os mundos, visto que a relação de acessibilidade determina a satisfação.

O sistema constituído pelos *frames* que possuem a propriedade da reflexividade é chamado de sistema T. Tal sistema é composto pelos axiomas K e $T = \Box p \rightarrow p$, ou seja, cada mundo está relacionado consigo mesmo. Desta forma, a importância de KT como sistema básico faz com que ele seja usualmente referido apenas como T (SILVA,

Tabela 2.2: Diferentes relações para a construção de sistemas modais.

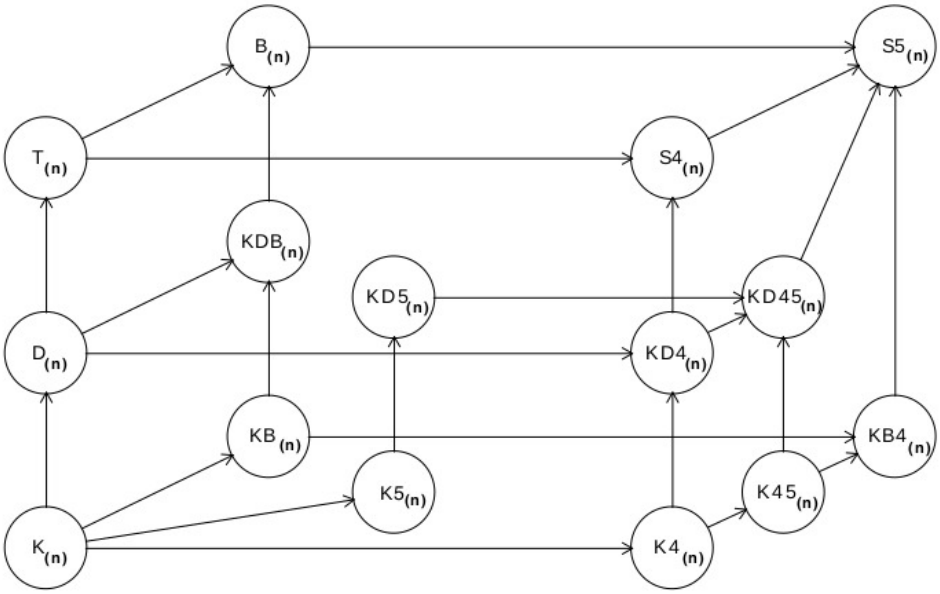
Relação de acessibilidade	Condição no *Frame*	Axioma
Reflexiva	$\forall w \in \mathcal{W}$, tal que wRw	$\Box p \rightarrow p$
Transitiva	$\forall w, x, y \in \mathcal{W}$, tal que $(wRx \wedge xRy) \Rightarrow wRy$	$\Box p \rightarrow \Box\Box p$
Simétrica	$\forall w, y \in \mathcal{W}$, tal que $wRy \Rightarrow yRw$	$p \rightarrow \Box\Diamond p$
Euclidiana	$\forall w, x, y \in \mathcal{W}$, tal que $(wRx \wedge wRy) \Rightarrow xRy$	$\Diamond p \rightarrow \Box\Diamond p$
Serial	$\forall w \exists x \in \mathcal{W}$, tal que wRx	$\Box p \rightarrow \Diamond p$
Funcional	$\forall w, x, y \in \mathcal{W}$, tal que $(wRx \wedge wRy) \Rightarrow x = y$	$\Diamond p \rightarrow \Box p$
Densa	$\forall w, x, \exists y \in \mathcal{W}$, tal que $wRx \Rightarrow (wRy \wedge yRx)$	$\Box\Box p \rightarrow \Box p$
Convergente	$\forall w, x, y, \exists z \in \mathcal{W}$, tal que $(wRx \wedge wRy) \Rightarrow (xRz \wedge yRz)$	$\Diamond\Box p \rightarrow \Box\Diamond p$

Fonte: Produção do próprio autor.

2013). O sistema B, que respeita as propriedades da reflexividade e simetria na classe de *frames*, é tal que os axiomas K, T e $B = p \rightarrow \Box\Diamond p$ estão presentes neste sistema.

Lewis apresentou dois novos sistemas, $S4$ e $S5$ (BLACKBURN; RIJKE; VENEMA, 2001). O sistema $S4$, possui os axiomas K, T e $4 = \Box p \rightarrow \Box\Box p$ e como propriedades do *frame* a reflexividade e a transitividade. Por outro lado, o sistema $S5$ pode ser construído de duas formas. A primeira adicionando o axioma B no sistema $S4$. A segunda possui os axiomas K, T, $5 = \Diamond p \rightarrow \Box\Diamond p$. Desta forma, um *frame* possui uma relação de acessibilidade reflexiva, transitiva e euclidiana. A Figura 2.2 representa a hierarquia dos principais sistemas modais presentes na literatura.

Figura 2.2: Hierarquia entre os sistemas modais normais



Fonte: (SILVA, 2013, p. 19)

2.6 Extensões da Lógica Modal

Para Gorsky (2008) com o acréscimo de novos operadores, a lógica em sua forma alética comporta-se sobre a veracidade de fórmulas vistas como necessárias ou possíveis, porém, ao tratar sobre o aspecto não alético, aborda novos conceitos em que surgem novas lógicas, tais quais temporal, epistêmica, deôntica e doxástica.

A lógica temporal aplica-se na verificação de ocorrência de eventos em um programa, ou seja, utilizada para raciocinar sobre sequência de estados induzidos por tais programas (MALANOVICZ, 2001). Para Mafra (2019), uma fórmula é dita como válida se para cada execução realizada é possível chegar em um estado desejável, desta forma, pode-se verificar a existência de *deadlock* em um programa dentre outras propriedades.

O estudo da epistemologia juntamente com a lógica chama-se lógica epistêmica (GARSON, 2018). A lógica epistêmica trata sobre a crença e o conhecimento. Através do sistema B da lógica modal, sua aplicação pode ser voltada para teoria dos jogos, inteligência artificial e análise de sistemas multiagentes (GABBAY et al., 2003).

A lógica deôntica trata sobre a permissão e obrigatoriedade. Através do operador O, equivalente ao de necessidade da lógica modal, define-se a semântica de permissão e proibido (BLACKBURN; RIJKE; VENEMA, 2001). Na área da ciência da computa-

ção a lógica pode ser aplicada em modelagem de sistemas concorrentes e programação paralela, entretanto no ramo do direito, é visto como a compreensão de normas e leis (MALANOVICZ, 2002).

3 Assistentes de Provas

Assistentes de provas, ou provadores semi-automáticos, são sistemas computacionais que permitem a formalização de teorias matemáticas com o objetivo de provar teoremas, de tal forma que verifica correção de programas (GEUVERS, 2009). A descrição para verificar a validade dos sistemas de *hardware* e *software* requer uma formalização matemática (MOURA; KONG; AVIGAD, 2019), nos quais constrói-se a partir de axiomas, hipóteses, parâmetros, conjuntos, entre outros (TEAM, 2019).

Para Yang e Deng (2019), a realização de uma prova compreende no desenvolvimento de conceitos e domínios de técnicas no estabelecimento de uma hipótese até o propósito de conclusão. Uma prova formal consiste no envolvimento lógico e métodos computacionais para designar afirmações em termos matemáticos precisos (MOURA; KONG; AVIGAD, 2019). Segundo (SILVA, 2019), o desenvolvimento de uma prova formal através de um provador semi-automático consiste no auxílio do desenvolvedor para guiar a prova, diferentemente de provadores automáticos em que o próprio *software* desenvolve a prova a partir de uma proposição dada de entrada.

Uma prova possui uma diferença considerável nos detalhes quando realizada manualmente e quando produzida em um computador (GEUVERS, 2009). Conforme apresentado por Moura, Kong e Avigad (2019), no caso dos *softwares* de assistentes de provas, sua funcionalidade é auxiliar, através de regras e axiomas, o desenvolvimento de forma prática e conclusiva do objetivo e que o mesmo está correto, no qual os passos lógicos para esta conclusão podem variar de quem estiver desenvolvendo.

Neste Capítulo apresentam-se conceitos e diferentes ferramentas de assistentes de provas, de forma que estes *softwares* auxiliam na verificação de validade de teoremas. A Sessão 3.1 destaca alguns diferentes assistentes de provas. A Sessão 3.2 expõe uma explicação a fundo do assistente de provas Coq.

3.1 Provadores semi-automáticos

Segundo Geuvers (2009), diversas provas complexas da matemática já foram formalizadas e desenvolvidas em diferentes assistentes de provas. Os mais distintos provadores semi-automáticos possuem propriedades e se adéquam ao desenvolvimento de tarefas de forma mais prática do que outros (SILVA, 2019), como é o caso do HOL-Light (HARRISON, 2017), Mizar (MUZALEWSKI, 1993), Isabelle (NIPKOW; PAULSON; WENZEL, 2019), Lean (MOURA; KONG; AVIGAD, 2019), entre outros.

Desenvolvido no ano de 2013, o assistente de provas Lean foi implementado pela equipe da *Microsoft Research* (MOURA; KONG; AVIGAD, 2019). O desenvolvimento de provas é utilizado a partir de comandos imperativos denominados como táticas (BENTZEN, 2019), de forma que aplica-se regras para concluir o objetivo fornecido. Baseado no *Calculus of Constructions* (CoC), o Lean conta com uma hierarquia contável de universos não cumulativos e tipos indutivos (MOURA; KONG; AVIGAD, 2019). A vantagem do assistente Lean é com base no fornecimento *online* para a prova de teoremas. O Lean pode lidar com classes de classificação superior, reduções de definição, coerções, sobrecargas e tipos, de maneira integrada (MOURA et al., 2015).

Implementado em ML, Isabelle é um sistema genérico para implementação de formalismos lógicos (NIPKOW; PAULSON; WENZEL, 2019), este assistente de provas possui extensões como Isabelle/HOL, voltado para lógica de ordem superior (AVIGAD et al., 2007) e Isabelle/Isar, no qual adaptou-se à teoria e desenvolvimento de provas (WENZEL, 2019). Segundo Avigad et al. (2007), o suporte oferecido é de forma automatizada, no qual inclui simplificador de termos, raciocínio automatizado e procedimento de decisão para aritmética linear e aritmética de Presburger.

O Mizar é baseado na teoria de conjuntos axiomática de Tarski-Grothendieck (MUZALEWSKI, 1993). O uso de símbolos que diferem de outros assistentes de provas faz com que a prova desenvolvida se torne mais próxima de textos matemáticos (BANCEREK et al., 2017). Segundo Naumowicz, lowicz e Grabowski (2016) o sistema utiliza lógica de primeira-ordem e suporta instruções com variáveis de segunda ordem, como por exemplo esquema de indução, e seus tipos são voltados para quantificação e qualificação para inferência de propriedades de objetos.

3.2 Coq

O assistente de provas Coq foi desenvolvido no final da década de 80 por pesquisadores do Instituto Nacional de Pesquisa em Informação e Automática (INRIA) (BöHNE; KREITZ, 2018). Segundo Paulin-Mohring (2012), o Coq possui a capacidade para o desenvolvimento matemáticos, tais como definições de objetos, declarações de predicados, conectivos lógicos e provas de escrita; já na computação, é voltado para especificação formal de programas e verificação de correção destes programas. Seu desenvolvimento é baseado na linguagem OCaml, a qual é uma linguagem funcional fortemente tipada e com tipagem estática, em que é uma extensão da linguagem Caml com o acréscimo de suporte a orientação a objeto (LEROY et al., 2019).

A linguagem lógica utilizada no Coq é uma variedade de teoria de tipos, chamada de CIC (TEAM, 2019). O CIC é uma linguagem representativa de programas funcionais, tais quais linguagem ML e provas de lógicas de ordem superior (PAULIN-MOHRING, 2015), no qual se estende do CoC para o suporte de tipos indutivos. O CoC é classificado como um Cálculo Lambda polimórfico de ordem superior e com tipos dependentes (SILVA, 2019). Através da utilização do isomorfismo de Curry-Howard as proposições possuem tipos e desta forma pode-se dizer que no Coq todo programa é uma prova e toda prova é um programa ou pode ser extraída como um programa.

O Coq faz uso de duas linguagens, mas com propósito diferente: Gallina e Vernacular (ALVES, 2018). A linguagem especificada para o desenvolvimento de provas é denominada Gallina, se caracterizando por ser uma linguagem funcional e uma linguagem de provas (SILVA, 2019). Vernacular é a linguagem de comandos onde o usuário interage com o *software*. O compilador Coq verifica automaticamente a exatidão das definições e de provas (PAULIN-MOHRING, 2012), desta forma, pode-se extrair o código gerado para as linguagens OCaml e Haskell.

Segundo Silva (2019), o Coq é de uso frequente por alunos e pesquisadores para a formalização de programas e prova de propriedades. Diferentes propriedades já foram provadas em Coq, tal qual o teorema das quatro cores (GONTHIER, 2008), provas que a construção de uma nova linguagem para *blockchain* é válida (OCONNOR, 2017) e a conjetura de Kepler (HALES et al., 2010). A demonstração do teorema das quatro cores é um caso de complexidade elevada, pois existem muitos casos de comparação para se analisar (SILVA, 2019). Tal teorema consiste em um grafo planar em que cada vértice

possui uma cor diferente dos seus vizinhos e o número total de cores nunca será maior do que quatro.

Os termos presentes no Coq possuem tipos, ao combinar diferentes termos com certos conectivos, criam-se novos termos nos quais também possuem um tipo (BARROS, 2010). Os tipos que o Coq suporta, além dos já conhecidos, como *nat* e *bool*, são *Prop*, *Set* e *SProp*, estes tipos são conhecidos como *sorts*, existe o tipo de *sort* universal, conhecido como *Type* (TEAM, 2019).

Pela necessidade de se quantificar sobre tipos, tipos também precisam possuir tipos dentro do sistema. Com a existência do paradoxo de Girard, não se pode definir *Type* com o tipo *Type*, pois gera inconsistência, para isso o Coq faz uma hierarquia de tipos, ou seja, $\forall i \in \mathbb{N}, Type_i : Type_{i+1}$. O algoritmo do Coq permite a transparência por parte do usuário para o índice do tipo, porém internamente se infere o tipo do *Type*. O *Type* mais baixo da hierarquia possui os *sorts Prop*, *Set* e *SProp* e os universos no Coq são cumulativos, ou seja, $Type_i \subseteq Type_{i+1}$.

Por questão da extração de código que acontece no Coq, é definido que *Set* e *Type* são computacionalmente relevantes, ou seja, são extraídos para código, já *Prop* e *SProp* não e são apagados no momento da extração. A questão do universo *Prop* ser irrelevante computacionalmente pode ser dada pela permissão de se utilizar o axioma do terceiro excluído: quando utilizado este axioma, Prop implica em *proof irrelevance*. O argumento de Prop ser impredicativo se dá pela construção de um objeto do grupo quantificado por todos os objetos desse grupo. Por questão do cálculo de construções dar a base da teoria do Coq, só tem *Prop* e $Type_0$, onde *Prop* é impredicativo e $Type_0$ como predicativo, dessa forma o Coq estende $Type_0$ para *Set* e uma hierarquia de *Types*. *Set*, ao contrário de *Prop*, é relevante e predicativo. Por outro lado, *SProp* já foi construído como *irrelevante*, sem a necessidade de utilizar o axioma do terceiro excluído, como em *Prop*.

Tabela 3.1: Tabela comparativa entre os *Type* e *sort*

Tipo	Relevância computacional	Predicativo	Relevância da prova
Type	Sim	Sim	Sim
Set	Sim	Sim	Sim
Prop	Não	Não	Sim[1]
SProp	Não	Não	Não

Fonte: Produção do próprio autor.

Uma vez que o Coq utiliza uma linguagem funcional com notação prefixa, a conversão de notação pode ser dada pelo comando *Notation*. A realização de uma transformação de uma função matemática, representada de forma infixa, para uma representação mais simples tem o propósito de facilitar a escrita e interpretação de definição das provas. A ordem de precedência da utilização destas notações é dada pelo intervalo de 0 a 200, onde o menor valor representa uma precedência maior e o maior valor retrata uma baixa precedência.

Uma definição indutiva é especificada através de nomes e tipos de construtores, esta declaração indutiva se enquadra em três contextos, *Inductive*, *Definition* e *Fixpoint*.

- *Inductive*: Cria novos tipos e atribui nomes, desta forma introduz tipos indutivos.
- *Definition*: Atribui um nome a um termo, desta forma, realiza *patter matching* para definir o valor de retorno. Definição não recursiva de um tipo já declarado.
- *Fixpoint*: Semelhante ao *Definition*, porém, torna possível a programação funcional recursiva de ordem superior (BARROS, 2010).

A representação de *Notation*, *Inductive*, *Definition* e *Fixpoint* em Coq é exemplificada abaixo.

```
1 Notation "x + y" := (plus x y)
2                     (at level 50, left associativity)
3                     : nat_scope.
4
5 Inductive bool : Type :=
6   | true
```

[1]Exceto ao assumir o lema do terceiro excluído

```
7     | false.
8
9   Definition negb (b:bool) : bool :=
10    match b with
11      | true ⇒ false
12      | false ⇒ true
13    end.
14
15  Fixpoint plus (n : nat) (m : nat) : nat :=
16    match n with
17      | O ⇒ m
18      | S n' ⇒ S (plus n' m)
19    end.
```

Para o desenvolvimento de uma prova no Coq utiliza-se regras de dedução, estas regras são táticas aplicadas em premissas e hipóteses para a conclusão de um objetivo. Segundo (DELAHAYE, 2000) a linguagem tática, conhecida por $\mathcal{L}_{Tac}$, é a linguagem implementada do provador, por ser Turing-completa, não é imposto limitações para sua utilização durante a prova. A Tabela 3.2 apresenta algumas táticas aplicáveis para a realização de uma prova.

Tabela 3.2: Algumas táticas existentes em Coq

apply	assert	auto	compare
compute	cycle	destruct	decompose
do	easy	exfalso	exists
first	fold	generalize	intros
let	match goal	now	omega
only	pattern	reflexivity	rewrite
set	simpl	solve	subst
tauto	transitivity	unfold	wlog

Fonte: Adaptado de (TEAM, 2019, p.618-621)

O desenvolvimento da modelagem de sistemas através do provador semi-automático Coq possui uma restrição quando abordado em lógicas de primeira ordem e intuicionista. O auxílio que conduzirá a construção da biblioteca de lógica modal pro-

porciona um aumento na construção de modelos, nos quais argumentos persistentes como necessários e possíveis podem ser adquiridos de forma mais clara e objetiva.

4 Trabalhos Relacionados

Neste capítulo serão relatados os trabalhos que apresentam uma relação com a proposta deste trabalho. O desenvolvimento da lógica modal em Coq tem diferentes tratamentos, como é explicado nas Sessões 4.1, 4.2 e 4.3. Além do que será apresentado, a literatura apresenta implementações em outros assistentes de provas, tais quais Isabelle (BENZMüLLER; CLAUS; SULTANA, 2015), HOL-Light (HARRISON, 2017) e Lean (BENTZEN, 2019).

4.1 Wind (2001)

A autora efetuou uma implementação de lógica modal com base no sistema S5 no assistente de provas Coq, tal sistema consiste nos axiomas K, T, 4 e 5. O objetivo para o desenvolvimento deste sistema é voltado para a lógica epistêmica, no qual realizou-se a modelagem de dois *puzzles*. O primeiro é o quebra-cabeça dos homens sábios, que consiste no objetivo de cada sábio descobrir de forma independente a cor do próprio chapéu, sem que possa visualizá-lo. O segundo é o quebra-cabeça das crianças enlameadas, onde existe uma quantidade de crianças com a testa suja e as mesmas precisam descobrir se estão com a testa suja.

Wind (2001) desenvolveu a prova em Coq para a aplicação dos *puzzles* através do método de prova de dedução natural, onde a demonstração das regras foram apresentadas para a lógica modal e a lógica do conhecimento. O desenvolvimento do código pode ser visto no próprio trabalho a partir do apêndice.

O desenvolvimento da biblioteca foi separado em nove componentes, de forma que o primeiro apresenta a estrutura de Kripke no Coq, definição de proposições, mundos, *frames* e modelos. O segundo componente demonstra exemplos da estrutura de Kripke, com a construção de mundos, as proposições em que cada mundo possui e como se comunicam, com e sem a utilização dos operadores modais. Em seguida implementou-se a teoria da correspondência, que provou-se, através de lemas, os sistemas que compõem o sistema S5. O desenvolvimento de regras de introdução e eliminação, em dedução natural,

e regras de derivação correspondeu ao componente quatro. Os dois seguintes componentes ficam responsáveis pela construção do sistema S5 e $S5^n$, respectivamente, com a restrição de regras impostas pela relação de acessibilidade do respectivo sistema. Os dois últimos componentes correspondem ao desenvolvimento da prova dos *puzzles* propostos no texto.

4.2 Doczkal e Smolka (2011)

Os autores Doczkal e Smolka (2011) obtiveram como foco uma formalização, em Coq, de lógica decidível para teoria de tipo construtivo. Os sistemas modais implementados no trabalho foram o K, T e $K4$, com o foco para a lógica dinâmica e a temporal. Os capítulos apresentam a lógica proposicional clássica e a lógica modal, tal qual a sua linguagem, semântica e sua formalização em *Coq* . A conclusão obtida foi uma representação fiel da lógica modal para a teoria de tipos construtivos e demonstrou-se a prova de um teorema de modelo pequeno e a decidibilidade computacional.

A implementação que foi proposta pelos autores pode ser encontrada na página da Universidade de Saarland[2]. O desenvolvimento do código estabelece a construção e definições de táticas, através de lemas foram construídos operadores proposicionais e propriedades de relação. Cada modelo foi definido através de estruturas, no qual consistem mundos, relações e proposições.

4.3 Benzmüller e Paleo (2015)

A proposta utilizada pelos autores é compreender se provar teoremas quando utilizado o assistente de provas Coq se adequa no raciocínio interativo. Desta forma, utilizou-se a implementação da lógica modal, sistemas K e $S5$, para pessoas com o conhecimento básico da lógica e de táticas do Coq com o objetivo de resolver problemas em diversas áreas aplicáveis. Além dos operadores modais utilizados, implementou-se os quantificadores para a lógica modal de primeira ordem. A construção de novas táticas viabilizou a aplicabilidade da dedução natural para seu método de prova, o desenvolvimento pode ser visto no próprio artigo. A conclusão dos autores se tornou positiva com o desenvolvimento fornecido para o usuário, de forma que o desenvolvimento implícito dos mundos possíveis

[2]https://www.ps.uni-saarland.de/ doczkal/coq-3/

e a relação de acessibilidade não permaneceu a cargo do usuário.

A implementação da lógica modal em Coq, constitui em construção de mundos, proposições, relações de acessibilidade e a definição dos operadores de necessidade e possibilidade. Definiu-se lemas e axiomas, como K, B, T e 4, no qual foram realizadas suas provas a partir de linguagem tática. A construção de mundos, proposições e relação de acessibilidade, foram definidas como *Parameter*, que se diz por ser um parâmetro compartilhado e utiliza-se para definir uma variável como implícita.

4.4 Considerações sobre os trabalhos relacionados

Conforme apresentado nas sessões anteriores, diferentes autores dispuseram o desenvolvimento de alguns sistemas modais. O desenvolvimento deste trabalho consiste na implementação dos sistemas encontrados na literatura, utilizando o método axiomático, e que seja escalável à introdução de novos axiomas e sistemas para o auxílio na prova. A Tabela 4.1 demonstra o objetivo dos autores citados e deste trabalho.

Tabela 4.1: Comparação entre os trabalhos relacionados

Autores	Sistemas							
	K	D	B	T	K4	K5	S4	S5
Wind (2001)	X			X	X	X	X	X
Doczkal e Smolka (2011)	X			X	X			
Benzmüller e Paleo (2015)	X		X	X	X			X
Desenvolvimento do Autor	X	X	X	X	X	X	X	X

Fonte: Produção do próprio autor.

5 Desenvolvimento da Biblioteca Modal

Neste capítulo será apresentado o desenvolvimento realizado na biblioteca modal, juntamente com as escolhas de implementação, estrutura do projeto e definições de mundos possíveis, conceito de *Frame* e Modelo, as propriedades e metapropriedades que consistem na lógica e os diferentes sistemas propostos. Por fim, serão relatadas as dificuldades que estiveram presentes durante a implementação da biblioteca.

5.1 Desenvolvimento e estrutura da implementação

O desenvolvimento do código consistiu na preservação da estrutura lógica apresentada na Sessão 2. A implementação da lógica foi realizada com *deep embedding*. Um desenvolvimento de *deep embedding* é a representação da lógica em um objeto, interpretação deste objeto como um tipo e prová-lo. Essa escolha se dá pela compreensão da lógica de como ela é, de modo que ao analisar o trabalho escrito e a biblioteca implementada, possa identificar de forma evidente a explanação teórica.

Diferentes módulos da biblioteca padrão no Coq foram utilizados para o desenvolvimento da biblioteca modal proposta pelo autor. O uso destas bibliotecas serviram para diferentes contextos, partindo da nomenclatura de operadores até o auxílio no desenvolvimento de provas.

As definições *Frame* e Modelo foram implementadas a partir de *records*; a formalização se assemelha à Definição 4 e à Definição 5, respectivamente, como demonstrados abaixo.

```
1 Record Frame : Type :={
2     W : Set;
3     R : W → W → Prop;
4 }.
5
6 Record Model : Type := {
7     F : Frame;
8     v : nat → (W F) → Prop;
```

```
9 }.
```

A estrutura do *record* faz com que os tipos sejam dependentes das instâncias dos tipos anteriores, ou seja, no *Frame*, a lista de relações entre mundos depende do conjunto de mundos daquele mesmo *Frame*. O *Model* possui um *Frame*, e o mesmo é carregado para a valoração de proposições nos respectivos mundos. Ambos os códigos podem ser reimplementados como tipo indutivo.

```
1 Inductive Frame : Type :=
2     Build_Frame : forall W : Set, (W → W → Prop) → Frame.
3
4 Inductive Model : Type :=
5     Build_Model : forall F : Frame, (nat → W F → Prop) → Model.
```

A construção de uma fórmula modal é realizada a partir de um tipo indutivo representando os operadores modais. Uma proposição atômica é caracterizada por $\#n$, onde $n \in \mathbb{N}$, tornando esta representação equivalente a p_n. A verificação de uma proposição válida em um mundo é formalizado pela relação dada `v M`, onde representa a função de valoração de um modelo.

A função recursiva `fun_validation` é a principal responsável pelo *deep embedding*, onde é feita uma interpretação da fórmula simbólica para a lógica do Coq. Esta função faz a quebra da fórmula para análise de validade em um determinado mundo, onde recebe por parâmetro o Modelo, mundo e a fórmula para análise. Quando realizado uma indução na fórmula, os casos criados são os operadores lógicos e o literal. Os casos `Box` e `Dia` determinam a construção dos operadores necessário e possível, respectivamente. Para o operador $\Box$ é analisada a validade da fórmula para todos os mundos que w possui uma relação, já para o operador $\Diamond$, deve existir pelo menos um mundo em que a fórmula é válida.

A validação da fórmula em um mundo é dada pela desconstrução indutiva e se assemelha à Definição 6.

```
1 Inductive modalFormula : Set :=
2     | Lit       : nat → modalFormula
3     | Neg       : modalFormula → modalFormula
4     | Box       : modalFormula → modalFormula
5     | Dia       : modalFormula → modalFormula
```

```
6     | And       : modalFormula → modalFormula → modalFormula
7     | Or        : modalFormula → modalFormula → modalFormula
8     | Implies   : modalFormula → modalFormula → modalFormula.
9
10 Fixpoint fun_validation (M: Model) (w: W (F M)) (φ: modalFormula): Prop :=
11   match φ with
12   | Lit    p     ⇒ v M w p
13   | Box    ψ     ⇒ forall w': W (F M), R (F M) w w' → fun_validation M w' ψ
14   | Dia    ψ     ⇒ exists w': W (F M), R (F M) w w' ∧ fun_validation M w' ψ
15   | Neg    ψ     ⇒ ¬fun_validation M w ψ
16   | And    ψ γ   ⇒ fun_validation M w ψ ∧ fun_validation M w γ
17   | Or     ψ γ   ⇒ fun_validation M w ψ ∨ fun_validation M w γ
18   | Implies ψ γ ⇒ fun_validation M w ψ → fun_validation M w γ
19   end.
```

A equivalência lógica foi implementada tanto para os operadores da lógica clássica, quanto para os operadores da lógica modal. Visto que classificamente a paridade do $\Diamond\varphi$ é a mesma da $\neg\Box\neg\varphi$. A estrutura dos operadores que tratam fórmulas modais se antecedem por um ponto, esta característica sintática se dá para uma diferenciação dos conectivos lógicos já instanciados pelo Coq.

Foram iniciadas as diferentes relações de equivalência entre sistemas da lógica modal e seus respectivos axiomas. Quando imposta uma restrição na relação de acessibilidade, se constrói um determinado sistema, simultaneamente um axioma representativo desta restrição passa a existir. A seguinte implementação demonstra uma representação da Tabela 2.2.

```
1  Theorem validation_reflexivity_frame:
2     forall M φ,
3     reflexivity_frame (F M) ↔
4     (M |= .□ φ .→ φ).
5  Admitted.
6
7  Theorem validation_simmetry_frame:
8    forall M φ,
9    simmetry_frame (F M) ↔
10   (M |= φ .→ .□ .◇ φ).
```

```
11 Admitted.
```

O sistema dedutivo foi formalizado dado um tipo indutivo para a dedução da lógica modal. A dedução é formada na lógica clássica a partir de um conjunto de premissas, axiomas e *Modus Ponens*, porém, para a lógica modal há inclusão de axiomas e da regra da Necessitação.

A dedução implementada recebe como argumento um sistema da lógica modal, uma teoria contida neste sistema e a conclusão em que se deseja obter. A teoria é composta por conjunto de fórmulas bem formadas que auxiliam no desenvolvimento da prova. O primeiro caso da dedução avalia se a premissa está presente na teoria, retornando `Some elemento`, se o elemento existir, ou `Nothing` caso contrário.

O segundo caso é a construção dos axiomas presentes durante a prova. Dado um sistema que se deseja provar uma propriedade, é verificada a disponibilidade destes axiomas. O sistema mais genérico, determinado como sistema K, contém os dez axiomas de Hilbert, o axioma da distributividade e o da possibilidade, e qualquer construção de outro sistema a partir do K, será carregando estes axiomas.

O terceiro e quarto caso é a aplicação das regras de *Modus Ponens* e generalização, respectivamente. O sistema dedutivo da biblioteca de lógica modal é representado a seguir.

```
1  Inductive deduction (A: axiom → Prop): theory → modalFormula → Prop :=
2    (* Premise. *)
3    | Prem: forall (t: theory)
4                   (f: modalFormula)
5                   (i: nat),
6          (nth_error t i = Some f) → deduction A t f
7    (* Axiom. *)
8    | Ax: forall (t: theory)
9                 (a: axiom)
10                (f: modalFormula),
11         A a → instantiate a = f → deduction A t f
12   (* Modus Ponens. *)
13   | Mp: forall (t: theory)
14                (f g: modalFormula)
```

```
15              (d1: deduction A t (f .→ g))
16              (d2: deduction A t f),
17        deduction A t g
18    (* Generalization. *)
19    | Nec: forall (t: theory)
20                  (f: modalFormula)
21                  (d1: deduction A t f),
22          deduction A t (□ f).
```

O sistema passado como argumento para a dedução expressa o conjunto de axiomas disponíveis para o desenvolvimento da prova. O sistema K representa o sistema mais genérico e quando instanciado para outro sistema, seus axiomas são carregados. Cada sistema diferente consiste em uma instanciação de outro sistema e o acréscimo de um ou mais axiomas, e semanticamente isso se determina por conta de alguma restrição realizada na relação de acessibilidade. O retorno de erro é dado quando utilizado um axioma que não pertence ao sistema.

O exemplo a seguir demonstra como se instancia um sistema dentro de outro. Para o sistema T, a construção base é feita a partir do sistema K e o acréscimo do axioma da reflexividade. Ao construir o sistema B, ou seja, acrescentando o axioma da simetria junto com o sistema T, por transitividade fica disponível também os axiomas do sistema K.

```
1   Inductive K: axiom → Prop :=
2     | K_ax1: forall φ ψ, K (ax1 φ ψ)
3     | K_ax2: forall φ ψ γ, K (ax2 φ ψ γ)
4     | K_ax3: forall φ ψ, K (ax3 φ ψ)
5     | K_ax4: forall φ ψ, K (ax4 φ ψ)
6     | K_ax5: forall φ ψ, K (ax5 φ ψ)
7     | K_ax6: forall φ ψ, K (ax6 φ ψ)
8     | K_ax7: forall φ ψ, K (ax7 φ ψ)
9     | K_ax8: forall φ ψ, K (ax8 φ ψ)
10    | K_ax9: forall φ ψ γ, K (ax9 φ ψ γ)
11    | K_ax10: forall φ ψ, K (ax10 φ ψ)
12    | K_axK: forall φ ψ, K (axK φ ψ)
13    | K_axPos: forall φ ψ, K (axPos φ ψ).
```

```
14
15 (* Reflexive *)
16 Inductive T: axiom → Prop :=
17   | T_K: forall φ, K φ → T φ
18   | T_axT: forall φ , T (axT φ).
19
20 (* Reflexive and Symmetry *)
21 Inductive B: axiom → Prop :=
22   | B_T: forall φ, T φ → B φ
23   | B_axB: forall φ , B (axB φ).
```

O desenvolvimento da prova da correção foi realizado através da indução do sistema K. Cada um dos casos da prova se refere aos respectivos axiomas e regras de derivação do sistema. Foram implementados, separadamente, as provas dos doze axiomas e as duas regras para um melhor auxílio na prova da correção e se assemelhar à demonstração apresentada à Definição 13.

```
1 Theorem soundness:
2   forall (Γ: theory) (φ: modalFormula),
3   (K; Γ |-- φ) →
4   (Γ ||= φ).
5 Admitted.
```

Iniciou-se o desenvolvimento da completude através da construção do conjunto de Lindenbaum. Esta construção indutiva parte do princípio na adição de fórmulas em Δ, enquanto houver consistência. A função `Consistency` verifica se não é possível derivar uma contradição em qualquer sistema para o conjunto de fórmulas `(P n :: Δ)`, ambos passados como parâmetros. A representação a seguir se remete à Definição 17.

```
1 Variable P: nat → modalFormula.
2 Variable Γ: theory.
3 Variable A: axiom → Prop.
4
5 Inductive Lindenbaum_set : nat → theory → Prop :=
6   | Lindenbaum_zero:
7     Lindenbaum_set 0 Γ
8   | Lindenbaum_succ1:
```

```
9     forall n Δ,
10    Lindenbaum_set n Δ →
11    Consistency A (P n :: Δ) →
12    Lindenbaum_set (S n) (P n :: Δ)
13  | Lindenbaum_succ2:
14    forall n Δ,
15    Lindenbaum_set n Δ →
16    ¬Consistency A (P n :: Δ) →
17    Lindenbaum_set (S n) Δ.
```

5.2 Dificuldades na implementação

Uma das principais dificuldades encontradas para a implementação da biblioteca consiste na transcrição da lógica modal no formato *deep embedding*. Embora a compreensão permaneça plausível, o aumento da complexidade para o desenvolvimento de provas no sistema é visível. Algumas abordagens utilizadas inicialmente tiveram que ser readaptadas para uma melhor modelagem da biblioteca.

O problema na modelagem do sistema dedutivo era carregar os subsistemas em um sistema maior, onde os axiomas precisavam ser mantidos e deixá-lo mais genérico para uma nova construção de sistema. Inicialmente foram implementados diferentes sistemas dedutivos, onde cada construção de sistema da lógica precisava instanciar as premissas, axiomas e regras de derivação. Logo houve uma necessidade de um desenvolvimento mais dinâmico, parametrizando o conjunto de axiomas.

Outra dificuldade encontrada foi a adaptação do conjunto de Lindenbaum, utilizado na teoria de conjuntos, para a lógica do Coq. Definiu-se similarmente através do tipo indutivo a construção original, ou seja, adicionando fórmulas que não gerem inconsistência ao conjunto ao longo do processo. O uso de uma representação do conjunto infinito dificultaria a identificação do conjunto maximal consistente, por conta disso, se determinou por meio da construção do conjunto finito.

6 Considerações Finais

Existe uma dificuldade vigente para a modelagem de certos tipos de sistemas computacionais com lógica clássica. As propriedades que consistem na lógica podem contribuir para os ramos da computação e facilitar no auxílio de modelagens, de forma que em diferentes contextos a representação matemática seja coerente com a prova realizada em provadores semi-automáticos. Uma prova realizada em *software* contribui para as propriedades representativas em que o programa em desenvolvimento consiste. Em vista disso, a construção da biblioteca de lógica modal torna as provas mais compreensíveis para o usuário que está programando.

A apresentação formal da lógica modal, no Capítulo 2, auxilia na compreensão do leitor a entender de forma aprofundada conceitos técnicos e, em seguida, modelar sistemas na biblioteca desenvolvida. Diferentes sistemas modais foram desenvolvidos na biblioteca modal para abranger diversas áreas de estudo, com o propósito de auxiliar e detectar erros em desenvolvimento de prova axiomática de forma escrita. O Exemplo 2 foi desenvolvido e pode ser encontrado em `exemplos.v` como entendimento de resolução da prova na biblioteca.

O diferencial do trabalho apresentado em relação aos trabalhos relacionados foi a dinamicidade no tratamento de diferentes sistemas da lógica modal. Conforme apresentado no Capítulo 4, os diferentes autores apresentam soluções para sistemas específicos, ou seja, a implementação do sistema dedutivo se restringiu à construção para o tratamento de um problema. O desenvolvimento retratado no trabalho trouxe de forma dinâmica a elaboração de um sistema desejado pelo usuário. Como apresentado no Capítulo 5, independente do sistema no qual se deseja trabalhar, é possível construir um novo sistema somente através de instanciações. Desta forma é possível modelar diversos problemas, seja com um sistema já apresentado na biblioteca ou construído pelo próprio usuário.

A biblioteca modal apresentada se demonstrou robusta para as propostas oferecidas. Foram desenvolvidas as propriedades da lógica modal, como as validações de fórmulas em mundos, modelos e sistemas, a equivalência lógica entre os conectivos lógicos, tanto para as lógica clássica, como para a lógica modal, a validação dos axiomas

para os respectivos *Frames* e o sistema dedutivo, onde consiste a estrutura dos diferentes sistemas, como propriedades para o sistema K. O desenvolvimento da correção da lógica modal foi apresentado para o sistema K, como mostrado durante o trabalho, já para a completude iniciou-se a construção da prova através do conjunto de Lindenbaum, deixando aberto o desenvolvimento para possíveis trabalhos futuros.

6.1 Trabalhos Futuros

Como proposta de trabalhos futuros para o presente trabalho, pode ser dividido em duas categorias, sendo uma continuação do desenvolvimento que ficara em aberto e uma melhoria para a biblioteca. As possíveis continuações no desenvolvimento podem ser dadas pela implementação da prova da correção para os diferentes sistemas da lógica modal. Sugere-se também a prova da completude da lógica modal utilizando o conjunto maximal consistente, através do lema de Lindenbaum. Adicionalmente, é recomendado construir uma biblioteca de táticas no Coq para a automatizar e simplificar provas no *deep embedding*.

Para uma melhoria e reestruturação de alguns conceitos implementados na biblioteca sugere-se a remodelagem da lógica modal no Coq sem a utilização do método *deep embedding*, por fins de diferenciação da estrutura e desenvolvimento de provas. Por fim, analisar como definir o conjunto maximal consistente de Lindenbaum no Coq.

Referências

AGUDELO, J. C. A. *Computação Paraconsistente: Uma Abordagem Lógica à Computação Quântica*. Tese (Doutorado) — Universidade Estadual de Campinas, São Paulo, 11 2009.

ALLWEIN, G.; HARRISON, W. L. Distributed modal logic. In: *Outstanding Contributions to Logic*. Springer International Publishing, 2016. p. 331–362. Disponível em: <https://doi.org/10.1007/978-3-319-29300-4_16>.

ALVES, T. de P. *Portando teorias entre assistentes de prova*: Um estudo de caso. (Projeto de Diplomação) — Centro de Informática, Universidade Federal de Pernambuco, Pernambuco, 2018.

ANGELOS, D. A. d. *Tableaux clausal para lógica modal*. 49 p. (Projeto de Diplomação) — Universidade de Brasília – Instituto de Ciências Exatas, Brasília, 2016.

AVIGAD, J. et al. A formally verified proof of the prime number theorem. *ACM Transactions on Computational Logic*, Association for Computing Machinery (ACM), v. 9, n. 1, p. 2–es, dec 2007. Disponível em: <https://doi.org/10.1145/1297658.1297660>.

BANCEREK, G. et al. The role of the mizar mathematical library for interactive proof development in mizar. *Journal of Automated Reasoning*, Springer Science and Business Media LLC, v. 61, n. 1-4, p. 9–32, nov 2017. Disponível em: <https://doi.org/10.1007-/s10817-017-9440-6>.

BARROS, F. J. F. *Uma Formalização da Composicionalidade do Cálculo λ em Coq*. 69 p. Dissertação (Mestrado) — Universidade de Brasília, Brasília, 2010.

BENTZEN, B. A henkin-style completeness proof for the modal logic s5. *arXiv preprint arXiv:1910.01697*, Carnegie Mellon University, p. 12, out 2019. Disponível em: <https://arxiv.org/pdf/1910.01697v1.pdf>.

BENZMüLLER, C.; CLAUS, M.; SULTANA, N. Systematic verification of the modal logic cube in isabelle/HOL. *Electronic Proceedings in Theoretical Computer Science*, Open Publishing Association, v. 186, p. 27–41, jul 2015. Disponível em: <https://doi.org/10.4204/eptcs.186.5>.

BENZMÜLLER, C.; PALEO, B. W. Interacting with modal logics in the coq proof assistant. In: BEKLEMISHEV, L. D.; MUSATOV, D. V. (Ed.). *Computer Science – Theory and Applications*. Cham: Springer International Publishing, 2015. p. 398–411. ISBN 978-3-319-20297-6.

BERTOLINI, C.; CUNHA, G. B. da; FORTES, P. R. *Lógica Matemática*. 1. ed. Santa Maria, Rio Grande do Sul: UFSM-NTE, 2017. ISBN 978 85 8341 184 0. Disponível em: <https://nte.ufsm.br/images/identidade_visual/LogicaMatematica.pdf>.

BLACKBURN, P.; RIJKE, M. d.; VENEMA, Y. *Modal Logic*. Cambridge, Inglaterra: Cambridge University Press, 2001. (Cambridge Tracts in Theoretical Computer Science).

BöHNE, S.; KREITZ, C. Learning how to prove: From the coq proof assistant to textbook style. *Electronic Proceedings in Theoretical Computer Science*, Open Publishing Association, v. 267, p. 1–18, mar 2018. Disponível em: <https://doi.org/10.4204/eptcs-.267.1>.

COCCHIARELLA, N. B.; FREUND, M. A. *Modal Logic*: An introduction to its syntax and semantics. Oxford University Press, 2008. Disponível em: <https://doi.org/10.1093-/acprof:oso/9780195366587.001.0001>.

COSTA, M. d. C. *Introdução à lógica modal aplicada à Computação*. Porto Alegre: UFRGS, 1992.

DELAHAYE, D. A tactic language for the system coq. In: *Logic for Programming and Automated Reasoning*. Springer Berlin Heidelberg, 2000. p. 85–95. Disponível em: <https://doi.org/10.1007/3-540-44404-1_7>.

DOCZKAL, C.; SMOLKA, G. *Constructive Formalization of Classical Modal Logic*. Universidade de Saarland, Alemanha, 2011.

GABBAY, D. et al. *Many-Dimensional Modal Logics - Theory and Applications*. 1. ed. Amsterdam, Netherlands: Elsevier, 2003. ISBN 0 444 50826 0. Disponível em: <https://doi.org/10.1016/s0049-237x(03)x8001-5>.

GARSON, J. Modal logic. In: ZALTA, E. N. (Ed.). *The Stanford Encyclopedia of Philosophy*. Fall 2018. [S.l.]: Metaphysics Research Lab, Stanford University, 2018.

GEUVERS, H. Proof assistants: History, ideas and future. *Sadhana*, Springer Science and Business Media LLC, v. 34, n. 1, p. 3–25, feb 2009. Disponível em: <https://doi.org/10.1007/s12046-009-0001-5>.

GOLINSKA-PILAREK, J.; MUNOZ-VELASCO, E.; MORA, A. A new deduction system for deciding validity in modal logic k. *Logic Journal of IGPL*, Oxford University Press (OUP), v. 19, n. 2, p. 425–434, jul 2010. Disponível em: <https://doi.org/10.1093-/jigpal/jzq033>.

GONTHIER, G. Formal proof – the four-color theorem. *Notices of the American Mathematical Society*, AMS, v. 55, n. 11, p. 1382–1393, Dec 2008. Disponível em: <https://www.ams.org/notices/200811/tx081101382p.pdf>.

GORANKO, V. Springer Nature, 1999. 255–258 p. Disponível em: <https://doi.org/10-.1023/a:1008282618104>.

GORSKY, S. B. *Semântica algébrica para as lógicas modais e seu interesse filosófico*. Dissertação (Mestrado) — Universidade Campinas, 2008.

GRUPO DE LÓGICA E FUNDAMENTOS DA FÍSICA. *Lógica*: Aspectos da lógica atual. Santa Catarina, Brasil: Universidade Federal de Santa Catarina, UFSC, 2008.

HAACK, S. *Philosophy of Logics*. Cambridge, England: Cambridge University Press, 1978. ISBN 0-521-29329-4.

HALES, T. C. et al. A revision of the proof of the kepler conjecture. *Discrete & Computational Geometry*, v. 44, n. 1, p. 1–34, Jul 2010. ISSN 1432-0444. Disponível em: <https://doi.org/10.1007/s00454-009-9148-4>.

HARRISON, J. *HOL Light Tutorial*. Hillsboro, Oregon, USA: Intel - Ronler Acres Campus, 2017. 230 p. Disponível em: <https://www.cl.cam.ac.uk/~jrh13/hol-light-/tutorial.pdf>.

HILPINEN, R. Deontic, epistemic, and temporal modal logics. In: *A Companion to Philosophical Logic*. Blackwell Publishing Ltd, 2006. p. 491–509. Disponível em: <https://doi.org/10.1002\%2F9780470996751.ch32>.

HUTH, M.; RYAN, M. *Lógica em Ciência da Computação*: modelagem e argumentação sobre sistemas. 2. ed. Rio de Janeiro: LTC, 2008.

KRIPKE, S. A. Semantical analysis of modal logic i normal modal propositional calculi. *Mathematical Logic Quarterly*, Wiley Online Library, v. 9, n. 5-6, p. 67–96, 1963.

LEROY, X. et al. *The OCaml system release 4.09*. Rocquencourt, França, 2019. Version 4.09. Disponível em: <https://ocaml.org/docs/>.

MAFRA, G. M. *Tradução automática de especificação formal modelada em TLA + para linguagem de programação*. (Projeto de Diplomação) — Bacharelado em Ciência da Computação–Centro de Ciências Tecnológicas, UDESC, Joinville, 2019.

MALANOVICZ, A. V. *Lógicas modais: fundamentos e aplicações*. 56 p. (Projeto de Diplomação) — Bacharelado em Ciência da Computação–Instituto de Informática, UFRGS, Porto Alegre, 2001.

MALANOVICZ, A. V. *Sistemas de dedução para lógicas modais proposicionais*. Rio Grande do Sul, 2002.

MASTOP, R. *Modal Logic for Artificial Intelligence*. Países Baixos, 2012.

MELO, D. H. F. de. Pressuposições metafísicas em semântica modal. *Kínesis - Revista de Estudos dos Pós-Graduandos em Filosofia*, Faculdade de Filosofia e Ciências, v. 9, n. 20, mar. 2018. Disponível em: <https://doi.org/10.36311/1984-8900.2017.v9n20.07.p87>.

MOURA, L. de; KONG, S.; AVIGAD, J. *Theorem Prooving in Lean*. [S.l.], 2019. Version 3.4.0. Disponível em: <leanprover.github.io>.

MOURA, L. de et al. The lean theorem prover (system description). In: *Automated Deduction - CADE-25*. Springer International Publishing, 2015. p. 378–388. Disponível em: <https://doi.org/10.1007/978-3-319-21401-6_26>.

MUZALEWSKI, M. *An Outline of PC Mizar*. Poland, 1993. Disponível em: <http://www.cs.ru.nl/~freek/mizar/>.

NAUMOWICZ, A.; LOWICZ, A. K.; GRABOWSKI, A. *Mizar Hands-on Tutorial*. Białystok, Polônia, 2016.

NIPKOW, T.; PAULSON, L. C.; WENZEL, M. *A Proof Assistant for Higher-Order Logic*. [S.l.], 2019. Isabelle2019. Disponível em: <https://isabelle.in.tum.de/documentation-.html>.

OCONNOR, R. Simplicity. In: *Proceedings of the 2017 Workshop on Programming Languages and Analysis for Security - PLAS 17*. ACM Press, 2017. Disponível em: <https://doi.org/10.1145/3139337.3139340>.

PAULIN-MOHRING, C. Introduction to the coq proof-assistant for practical software verification. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012. p. 45–95. Disponível em: <https://doi.org/10.1007/978-3-642-35746-6_3>.

PAULIN-MOHRING, C. Introduction to the calculus of inductive constructions. In: PALEO, B. W.; DELAHAYE, D. (Ed.). *All about Proofs, Proofs for All*. College Publications, 2015, (Studies in Logic (Mathematical logic and foundations), v. 55). Disponível em: <https://hal.inria.fr/hal-01094195>.

PRIMO, G. A. L. A linguagem dos mundos possíveis. 4^o *Encontro de Pesquisa na Graduação em Filosofia da UNESP*, v. 2, n. 2, p. 9, 2009.

ROCHA, R. M. *O Realismo modal de David Lewis: uma opção pragmática*. 117 p. Dissertação (Mestrado) — Universidade Federal de Goiás, Goiás, 2010.

SILVA, F. S. C. da; FINGER, M.; MELO, A. C. V. de. *Lógica para computação*. [S.l.]: Thomson Learning, 2006.

SILVA, G. B. *Implementação de um Provador de Teoremas para Lógica Modais Normais*. 113 p. (Projeto de Diplomação) — Universidade de Brasília – Instituto de Ciências Exatas, Brasília, 2013.

SILVA, R. C. G. *Um algoritmo verificado para inferência de tipos na presença de recursão polimórfica*. 54 p. Dissertação (Mestrado) — Universidade do Estado de Santa Catarina, Joinville, Santa Catarina, 2019.

TEAM, T. C. development. *The Coq proof assistant reference manual*. [S.l.], 2019. Version 8.9.0. Disponível em: <http://coq.inria.fr>.

VARDI, M. Y. *Why is modal logic so robustly decidable?* Texas, Estados Unidos, 1997.

WENZEL, M. *The Isabelle/Isar Reference Manual*. [S.l.], 2019. 354 p. Isabelle2019. Disponível em: <https://isabelle.in.tum.de/doc/isar-ref.pdf>.

WIND, P. de. *Modal logic in coq*. Dissertação (Mestrado) — Vrije Universiteit, 2001.

YANG, K.; DENG, J. Learning to prove theorems via interacting with proof assistants. In: CHAUDHURI, K.; SALAKHUTDINOV, R. (Ed.). *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. PMLR, 2019. (Proceedings of Machine Learning Research, v. 97), p. 6984–6994. Disponível em: <http://proceedings.mlr.press/v97/>.

ZALTA, E. N. *Basic Concepts In Modal Logic*. Center for the Study of Language and Information Stanford University, 1995. Disponível em: <http://mally.stanford.edu-/notes.pdf>.