UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

INSTITUTO DE INFORMÁTICA

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LÓGICAS MODAIS:

FUNDAMENTOS E APLICAÇÕES

ALINE VIEIRA MALANOVICZ

Projeto de Diplomação

Prof. Dr. Tiarajú Asmuz Diverio

Orientador

Porto Alegre, maio de 2002.

Sumário

Lista de Figuras

Resumo

Este trabalho apresenta os fundamentos e algumas aplicações de lógica modal, especialmente na formalização de raciocínio sobre conhecimento e de sistemas concorrentes. É apresentada uma visão geral de lógica, incluindo a lógica clássica proposicional e a de primeira ordem, com destaque para suas propriedades formais. Sobre as lógicas modais proposicionais, são apresentados seus fundamentos, com destaque para o aspecto semântico, e seus sistemas de prova tais como dedução natural, *tableau* e resolução. É também apresentada brevemente uma visão geral sobre aplicações de lógica modal a algumas das diversas áreas da Ciência da Computação.

Abstract

This work presents the basics and applications of modal logics, specially in knowledge representation and in concurrent systems. We begin with an overview of logics, including classical propositional and first-order logic, with an emphasis on its formal properties. Next, we describe the foundations of modal logics, with an emphasis on Kripke semantics and corresponding proof systems such as natural deduction, resolution and tableux for propositional modal logics. Finally, a survey of applications of modal logics in Computing Science is presented.

1 Introdução

1.1 Motivação

Os aspectos teóricos da computação, incluindo as lógicas, têm fundamental importância para a Ciência da Computação. Proporcionam não só um adequado embasamento teórico necessário para um correto e amplo entendimento da ciência envolvida na computação, como também mais um estágio na formação do desenvolvimento de um raciocínio lógico (cada vez mais necessário em todas as subáreas da computação), no desenvolvimento de demonstrações e de suas técnicas, tanto formal, como informalmente. Assim, um raciocínio seguindo algumas regras simples e universalmente aplicáveis pode constituir uma prova precisa, mas que pode ser adequadamente seguida em uma argumentação informal. Assim, propriedades abstratas podem ser especificadas e estudadas independentemente de estruturas e de implementações.

Deve-se notar que o alto nível abstrato em que esses estudos são feitos proporciona conclusões que transcendem a evolução tecnológica pela qual estão passando os computadores modernos. [DIV2000] As tecnologias podem mudar, porém os fundamentos permanecem os mesmos, e a lógica matemática é uma ferramenta fundamental na definição de conceitos computacionais, pois o seu estudo permite a formação de um embasamento teórico expressivo para formalizar o funcionamento de sistemas. Além disso, uma motivação adicional – e um tanto mais voltada ao aspecto prático – ao estudo de lógicas modais e sua aplicação à Ciência da Computação é que elas podem ser tratadas como uma sugestão de meios ou formas para os projetistas de sistemas desenvolverem produtos melhores, mais elegantes e com descrições formais mais simples. [SOU99]

1.2 Objetivos

Este trabalho tem como objetivo apresentar uma introdução às lógicas modais de forma que possam ser usadas dentro da Ciência da Computação. Essas lógicas são utilizadas para especificar e raciocinar sobre propriedades comportamentais de modelos e aplicações em Ciência da Computação. [SOU99] Como essa lógica não é um tópico abordado no curso de graduação, este trabalho pretende ser uma referência introdutória aos fundamentos e às possibilidades de aplicação das lógicas modais. Para essa introdução, serão apresentados os conceitos básicos de Lógica Modal e um estudo de suas aplicações, especialmente em programação, na especificação de sistemas concorrentes e na formalização do raciocínio sobre conhecimento.

Tal objetivo impõe que sejam primeiramente definidos os símbolos utilizados, discutidos os conceitos de necessidade e possibilidade e então apresentados o sistema semântico e os sistemas de dedução. Serão também estudadas as propriedades fundamentais de sistemas lógicos como satisfação, relação de conseqüência, completude e corretude, e os sistemas de prova para lógicas modais (incluindo *tableau* e dedução natural), pois são fundamentais no estudo da formalização de sistemas concorrentes e no estudo da automação do raciocínio sobre conhecimento.

1.3 Metodologia

Foi realizada extensa pesquisa bibliográfica, a qual produziu a presente compilação de conceitos, provas, exemplos e aplicações. Foram pesquisados os conceitos básicos da Lógica Modal Proposicional, as propriedades fundamentais de sistemas de prova de Lógica Modal Proposicional, e suas aplicações à Ciência da Computação, especialmente na análise de concorrência e na representação do raciocínio sobre conhecimento. Foram produzidos resumos de cada obra pesquisada, de modo a permitir a compilação dos mesmos para a composição de grandes porções dos capítulos deste trabalho. Ainda, foram coletados e descritos exemplos de aplicações pesquisadas. O material escrito resultante de todo esse processo metódico foi reunido e organizado neste volume.

1.4 Organização do texto

O *Capítulo 2 – Visão Geral de Lógica* apresenta uma descrição da Lógica Matemática, apresentando a Lógica e suas propriedades de uma maneira geral. São ressaltadas as suas principais características e é definida uma estrutura conceitual que servirá como base para os capítulos seguintes. A Lógica Clássica (Proposicional e de Primeira Ordem) também é apresentada sucintamente, como uma ilustração dos conceitos básicos apresentados.

O *Capítulo 3 – Fundamentos de Lógica Modal* apresenta os aspectos gerais dessa lógica, com o seu conceito de *modalidades da verdade* de uma proposição. É introduzida também a Semântica dos Mundos Possíveis, as propriedades das lógicas modais e alguns dos diferentes sistemas de Lógica Modal (K, T, D, K4, S4, KB, B, B4, S5). Os sistemas de prova para Lógica Modal Proposicional são descritos, porém um estudo mais detalhado dos mesmos é reservado para os trabalhos futuros. A Lógica Modal de Primeira Ordem não é tratada neste trabalho.

O *Capítulo 4 – Aplicações de Lógica Modal* reúne parte do material pesquisado a respeito das possíveis formas de utilização das lógicas modais nas diversas áreas da Ciência da Computação, com ênfase em aspectos de programação, na formalização de raciocínio sobre conhecimento e na formalização de sistemas concorrentes.

A parte final do trabalho, intitulada *Conclusões*, faz uma revisão e uma avaliação do mesmo, descrevendo as contribuições trazidas por ele e apontando perspectivas de trabalhos futuros.

Um ponto importante a ressaltar é que, neste trabalho, são descritos conceitos básicos referentes aos principais tópicos apresentados, e não é feito qualquer aprofundamento em detalhes formais dos mesmos. Tal detalhamento pode ser buscado pelo leitor nas referências às fontes pesquisadas, as quais são comentadas na seção apropriada. Além disso, algumas notações tomadas de diferentes fontes foram unificadas, com a finalidade de manter a uniformidade do texto e a facilidade de leitura do mesmo.

2 Visão Geral de Lógica

2.1 Introdução

Uma lógica é um sistema formal, um sistema de manipulação de símbolos contendo uma linguagem formalmente descrita e um conjunto de regras para manipulá-los previamente estabelecidos. [SOU99] Uma lógica simbólica é um conjunto de estudos tendentes a expressar em linguagem matemática as estruturas e operações do pensamento, deduzindo-as a partir de um número reduzido de axiomas, com a intenção de criar uma linguagem rigorosa, adequada ao pensamento científico. [GIR95]

Na lógica, a linguagem formal que se manipula é formada de entidades básicas chamadas sentenças, as quais são usualmente dadas por definições indutivas e constituem uma linguagem formal. As regras da lógica devem permitir estabelecer relacionamentos, chamados de inferências, entre as sentenças. Inferência é um tipo de argumento que tem premissas e conclusão. As regras da lógica definem quais são as inferências válidas, ou seja, aquelas que, se aceitas as premissas, obrigam a que se aceitem as conclusões. [SOU99]

Considera-se aqui a validade formal, ou seja, um argumento (inferência) será formalmente válido se for válido em função da forma (estrutura sintática) das premissas e da conclusão. As regras de manipulação (no caso acima, o estabelecimento de relações entre premissas e conclusão) das sentenças definem o que se chama de relação de conseqüência de uma lógica.

Para tanto, há a *abordagem baseada em satisfação*, que focaliza a manipulação de sistemas de avaliação – por exemplo, o raciocínio sobre tabelas-verdade –, interessando-se em definir qual o valor-verdade que uma *interpretação* decide para as sentenças da linguagem da lógica; e a *abordagem baseada em provas*, que focaliza a definição de meios para a manipulação de expressões sintáticas com o objetivo de formular e construir provas de argumentos, segundo a *teoria da prova*, pois as premissas e a conclusão são ligadas por uma *prova*, construída com base na forma sintática das premissas e da conclusão. [GIR95]

2.1.1 Satisfação

A definição de uma linguagem exige a especificação de uma *sintaxe* e de uma *interpretação* que seja capaz de avaliar *todas* as sentenças da linguagem com um dos valores-verdade pré-definidos na interpretação dada. Essa interpretação (dos operadores da linguagem) também pode ser equivalentemente apresentada na forma de tabelas-verdade.

A notação de símbolos A, B, etc. para sentenças e Γ, Δ, etc. para conjuntos de sentenças é muito usada. Se A é verdadeiro na interpretação *m*, escreve-se *m* $\Vdash$ A, e diz-se que *m satisfaz A.* Também se diz que *A vale em m*, e que *m é um modelo para A.* Chama-se $\Vdash$ uma *relação de satisfação.* Dadas uma coleção de sentenças e uma coleção de interpretações, a abordagem baseada em satisfação da lógica preocupa-se em identificar essa relação.

2.1.2 Sistemas de Avaliação

Na definição das linguagens (gramáticas), podem existir categorias gramaticais maiores (fórmulas, por exemplo) definidas em termos das mais simples (fórmulas atômicas e operadores, por exemplo), por motivos que serão esclarecidos nesta seção.

Uma *linguagem proposicional* L é composta de um conjunto contável de sentenças atômicas P e um conjunto de operadores (ou conectivos) O. Os operadores possuem uma aridade, que é o número de sentenças que eles tomam para formar outra sentença. O conjunto de sentenças de uma linguagem proposicional é o menor conjunto que contém o conjunto contável de sentenças atômicas e que é fechado sob os operadores do conjunto de operadores. Na seção seguinte, a qual trata da Lógica Clássica Proposicional, ela será apresentada segundo esta definição.

Um *sistema de avaliação* para uma linguagem proposicional *m* é composto por um conjunto de valores-verdade (com pelo menos dois elementos) M, um subconjunto próprio não-vazio dos valores-verdade (chamado conjunto dos valores-verdade designados) D, e um conjunto de funções F, cada uma correspondendo a um dos operadores do conjunto de operadores, tal que f_o: $M^{n0} \rightarrow M$ (onde n_0 é a aridade do operador o). Diz-se que f_o interpreta o.

Um sistema de avaliação fornece uma descrição composicional dos operadores da linguagem da lógica considerada, mas não diz como avaliar sentenças atômicas. Para que seja possível saber o valor-verdade de uma sentença, é necessário que se atribuam valores-verdade às sentenças atômicas que a constituem.

Uma *atribuição* ou *valoração* (a) relativa a um sistema de avaliação para uma linguagem é uma função que parte do conjunto contável de sentenças atômicas da linguagem (P) e chega ao conjunto de valores-verdade (M). Ou seja, a : P $\rightarrow$ M. Dada uma atribuição, é possível calcular o valor-verdade de qualquer sentença da linguagem.

Cada atribuição (a) relativa a um sistema de avaliação (M) induz uma *interpretação* ou *avaliação* v_a dada por:

- $v_a(p) = a(p)$, para $p \in P$ (conjunto contável das de sentenças atômicas da linguagem)
- $v_a(o(A_1,...,A_n)) = f_o(v_a(A_1),...,v_a(A_n))$, onde *n* é a aridade do operador o, f_o interpreta o em M, e $A_1,...,A_n$ são sentenças da linguagem.

Uma *interpretação* é um sistema de avaliação mais uma atribuição. A parte do sistema de avaliação garante a avaliação dos operadores, enquanto a parte da atribuição especifica os valores-verdade das sentenças atômicas. A divisão da interpretação em dois componentes permite que os valores-verdade das sentenças atômicas (atribuição) variem, enquanto o tratamento dos operadores permanece fixo. Dessa forma, o sistema de avaliação *m* é uma coleção de interpretações, cada uma das quais tratando os operadores da linguagem de uma mesma maneira.

Uma sentença A é uma *tautologia* em um sistema de avaliação *m* se, para toda atribuição (a) relacionada a *m*, $v_a(A) \in D$.

2.1.3 Teoria da Prova

Uma descrição de uma lógica através da abordagem baseada em prova é chamada de *apresentação*. Uma apresentação especifica uma relação de conseqüência $\vdash$ como segue: $\Gamma \vdash A$ (lê-se *"A é derivável, ou dedutível, a partir de* Γ") se existe uma *prova* de A a partir de Γ, ou seja, uma prova com *conclusão* A e *premissas* Γ. Existem na literatura três estilos de apresentações de uso bastante difundido [RYA92]: axiomáticas; em dedução natural; e em seqüentes (a expressão $\Gamma \vdash A$ é um seqüente, representando o argumento com *premissas* Γ e *conclusão* A). Na seção seguinte, a qual trata da Lógica Clássica Proposicional, elas será apresentada utilizando-se o estilo axiomático.

No estilo axiomático, a relação de conseqüência é caracterizada por um conjunto dado de axiomas e um conjunto finito de regras de inferência. Os axiomas fazem o papel de "verdades básicas", e a idéia é gerar verdades adicionais pela aplicação das regras de inferência. Uma prova de A, a partir de premissas Γ, é uma seqüência finita de fórmulas, terminando com A, tal que cada fórmula da seqüência atende a pelo menos uma das seguintes condições: é um dos axiomas; ou é um membro de Γ; ou é derivável das fórmulas anteriores na seqüência por meio de uma regra de inferência. Se existe uma prova de A a partir de premissas Γ, escreve-se $\Gamma \vdash A$. Uma sentença A é um *teorema* da apresentação se existe uma prova de A a partir de um conjunto vazio de premissas (ou seja, $\emptyset \vdash A$).

2.1.4 Relações de Conseqüência

Uma relação de conseqüência sobre um conjunto de sentenças (de uma linguagem L) é a relação, escrita $\vdash$, entre *P*(L) (o conjunto potência de L) e L, a qual satisfaz pelo menos as seguintes propriedades tidas como essenciais em um sistema lógico (há outras propriedades, como compacidade e propriedade da dedução, por exemplo). Na notação, as vírgulas significam união de conjuntos; daí Δ,Γ significa $\Delta\cup\Gamma$ e Δ,C significa $\Delta\cup\{C\}$ [SER98]:

- *inclusão*: **se** $A \in \Gamma$ **então** $\Gamma \vdash A$
  - se A está explicitamente entre as premissas de um argumento, então A é uma conseqüência válida;
- *monotonicidade*: **se** $\Gamma \vdash A$, **então** $\Gamma, \Delta \vdash A$
  - (regra de *enfraquecimento*): se a sentença A segue de Γ, então ela segue de qualquer conjunto em que Γ esteja incluído (ou seja, o conjunto de conseqüências de Γ não decresce quando Γ cresce);
- *corte*: **se** $\Gamma \vdash C, \Delta$ **e** $C \vdash A$, **então** $\Delta, \Gamma \vdash A$
  - se é possível derivar A de algumas premissas Δ e de uma sentença C, então A pode ser derivada daquelas premissas (Δ) e de mais algumas outras premissas a partir das quais se tem C (essa propriedade é chamada de *corte*, pois podemos tirar/cortar a fórmula C e ir diretamente para a fórmula A).

2.1.5 Relações de Acarretamento (*entailment relations*)

Relações de acarretamento são um caso especial de relações de conseqüência e são definidas em termos de sistemas de avaliação. São escritas $\models$, possivelmente subscritas pelo sistema de avaliação em questão.

Define-se que um conjunto de sentenças Γ de uma linguagem *acarreta* uma sentença A dessa linguagem em um sistema de avaliação *m* para essa linguagem (com seus valores-verdades em M e seus valores designados em D), e escreve-se $\Gamma \models_m A$, se, para toda atribuição **a** tal que $v_a(B) \in D$ para cada $B \in \Gamma$, então $v_a(A) \in D$ também. Ou seja, se as premissas são todas designadas (isto é, avaliadas para valores-verdade designados), o mesmo acontece com a conclusão. $\Gamma \models_m A$ significa que, de acordo com o sistema de avaliação *m*, o argumento com premissas Γ e conclusão A é válido. A prova de que a relação de acarretamento é uma relação de conseqüência é encontrada em [RYA92].

2.1.6 Propriedades Semânticas das Apresentações

Duas propriedades são de extrema importância quando se comparam $\vdash$ (relação de conseqüência definida por teoria da prova – uma apresentação) e $\models$ (relação de acarretamento): corretude (*soundness*) e completude (*completeness*).

Corretude (*soundness*): $\vdash$ é correta com relação a $\models$ se

$$\Gamma \vdash A \text{ implica } \Gamma \models A.$$

Provar que *G |¾ A implica G |= A* mostra que o sistema de prova é *correto* (*sound*) com relação ao sistema de avaliação no sentido de que ele *não é capaz de provar algo que não seja válido*. Para provar a *corretude*, é suficiente que se mostre que as regras de construção de provas preservam *acarretamento*.

Completude (*completeness*): $\vdash$ é completa com relação a $\models$ se

$$\Gamma \models A \text{ implica } \Gamma \vdash A.$$

Provar que $\Gamma \models A$ implica $\Gamma \vdash A$ mostra que o sistema de prova é suficientemente forte para provar tudo o que é válido (com relação ao sistema de avaliação). A prova da completude de um sistema de prova é mais difícil que a prova da corretude do mesmo. Uma maneira de proceder é mostrar que o sistema de prova pode ser estendido pela adição de qualquer sentença para a qual nem ela própria nem a sua negação seja teorema como um axioma, de tal forma que exista uma atribuição para o sistema de avaliação a qual faça todos os sistemas verdadeiros [SOU99].

Há também a questão da *decidibilidade* (a grosso modo: mesmo sabendo que o sistema dedutivo é capaz de provar todas as fórmulas válidas, existe um procedimento que garanta que se pode sempre obter tal prova?; ou, no caso de uma fórmula não válida, mostrar que tal prova não existe?). Essa questão não será descrita formalmente aqui.

2.2 Lógica Clássica Proposicional

Nesta seção, a Lógica Clássica Proposicional é apresentada sucintamente.

2.2.1 A linguagem da lógica clássica proposicional

Os símbolos da Lógica Clássica Proposicional são

- um conjunto enumerável φ de símbolos proposicionais;
- pontuação: “(“ e “)”;
- *conectivos* ou *operadores*: “$\neg$”, “$\vee$”, “$\wedge$” e “$\rightarrow$”

As *fórmulas* ou *sentenças* da Lógica Clássica Proposicional são:

- todo símbolo proposicional é uma fórmula chamada *fórmula atômica*;
- se A é uma fórmula, então $(\neg A)$ também é uma fórmula;
- se A e B são fórmulas, então $(A \vee B)$, $(A \wedge B)$ e $(A \rightarrow B)$ também são fórmulas;
- nada é fórmula, a menos que seja forçada a ser por um dos itens acima.

2.2.2 Apresentação Axiomática

O *esquema de axiomas* da Lógica Clássica Proposicional é definido considerando que, se A, B e C são fórmulas (da Lógica Clássica Proposicional), então as seguintes fórmulas são axiomas [RYA92]:

AXM1 $A \rightarrow (B \rightarrow A)$

AXM2 $(C \rightarrow (A \rightarrow B)) \rightarrow ((C \rightarrow A) \rightarrow (C \rightarrow B))$

AXM3 $(\neg A \rightarrow \neg B) \rightarrow ((\neg A \rightarrow B) \rightarrow A)$

Utiliza-se a *regra de inferência* chamada *modus ponens* para a formação do sistema dedutivo da Lógica Clássica Proposicional. Ela é definida (sendo A e B fórmulas e Γ um conjunto de fórmulas) como:

$$\textbf{se } \Gamma \vdash A \textbf{ e } \Gamma \vdash (A \rightarrow B) \textbf{ então } \Gamma \vdash B$$

É importante ressaltar que o conjunto de axiomas acima foi apresentado sob a forma de *esquema de axiomas*, ou seja, os esquemas apresentados não são axiomas – nem ao menos são fórmulas – são fôrmas em que cada letra (A, B e C) é receptáculo para as fórmulas que as substituem (instanciam), obtendo-se assim os axiomas. Na definição da regra de inferência, a existência de Γ nos dois "lados" da regra mostra que a conclusão depende das mesmas hipóteses de que dependem as premissas. [RYA92]

2.2.3 Sistema de Avaliação

O *sistema* clássico *de avaliação* para a Lógica Clássica Proposicional é definido sobre os seguintes conjuntos: conjunto O dos operadores da linguagem = $\{\vee, \wedge, \rightarrow, \neg\}$; conjunto M dos valores-verdade = {V, F}; conjunto D dos valores-verdade designados = {V}; e conjunto F de funções sobre o conjunto o dos operadores dado pelas seguintes tabelas:

fÚ	**V**	**F**
V	V	V
F	V	F

fÙ	**V**	**F**
V	V	F
F	F	F

f$_{\rightarrow}$	**V**	**F**
V	V	F
F	V	V

fØ	
V	F
F	V

2.2.4 Semântica

Uma semântica para a Lógica Clássica Proposicional tem aqui sua definição apresentada pela composição de diversos conceitos relacionados.

Uma função de valoração V tem a seguinte assinatura: $V : \varphi \rightarrow \{V, F\}$. Uma extensão da função de valoração V, chamada de V^o, tem a assinatura $V^o : F \rightarrow \{V, F\}$ (onde *F* é o conjunto de todas as fórmulas, *p* é um símbolo proposicional, V é uma função de valoração, e A e B $\in F$) e é definida da seguinte forma:

(S p) $V^o(p) = V(p)$;

(S ¬) $V^o(\neg A) = V$, se $V^o(A) = F$
$V^o(\neg A) = F$, caso contrário;

(S ∧) $V^o(A \wedge B) = V$, se $V^o(A) = V$ e $V^o(B) = V$;
$V^o(A \wedge B) = F$, caso contrário;

(S ∨) $V^o(A \vee B) = V$, se $V^o(A) = V$ ou $V^o(B) = V$;

$V^o(A \vee B) = F$, caso contrário;
($S \rightarrow$) $V^o(A \rightarrow B) = V$, se $V^o(A) = F$ ou $V^o(B) = V$;
$V^o(A \rightarrow B) = F$, caso contrário;

Uma fórmula A é *satisfatível* se existir uma função de valoração $V^o(A) = V$. Nesse caso, diz-se que a função de valoração V satisfaz A. Uma fórmula A é *válida* se ela for satisfeita por toda função de valoração. Uma fórmula A é uma *contradição* se ela não é satisfeita por nenhuma função de valoração. Uma fórmula A é uma *conseqüência tautológica* de um conjunto de fórmulas Γ – escreve-se $\Gamma \models A$ – se toda função de valoração que satisfizer todo membro de Γ também satisfaz A.

A expressão $\emptyset \models A$ pode ser interpretada da seguinte forma: como não se pode apresentar nenhuma função de valoração que não satisfaça algum membro de $\emptyset$, pois esse não possui membro algum, podemos dizer, por *vacuidade*, que qualquer função de valoração satisfaz todos os membros de $\emptyset$. Assim, se $\emptyset \models A$, então A é satisfeita por todas as funções de valoração, ou seja, é válida. Nesse caso, escreve-se $\models A$ e diz-se que A é uma *tautologia*.

2.2.5 Corretude e Completude

Como visto em uma seção anterior, o problema de provar a corretude e a completude de uma apresentação refere-se a mostrar, respectivamente, que:

$$\Gamma \vdash A \text{ implica } \Gamma \models A$$

e que

$$\Gamma \models A \text{ implica } \Gamma \vdash A$$

para A sendo uma fórmula e Γ, um conjunto de fórmulas.

Essas demonstrações são parte das demonstrações correspondentes para a Lógica Modal (capítulo 3), portanto não serão provadas neste ponto, mas no capítulo 3. As provas de tais propriedades e de outros resultados para a lógica clássica podem ser encontrados na obra de Nolt [NOL91]. Sendo assim, apenas se afirma que, para a semântica e para a apresentação dadas neste capítulo para a Lógica Clássica Proposicional, temos que o conjunto de fórmulas válidas (*tautologias*) é igual ao conjunto dos teoremas, ou seja:

$$\Gamma \vdash A \text{ se e somente se } \Gamma \models A.$$

2.3 Lógica Clássica de Primeira Ordem

Esta seção descreverá brevemente a lógica clássica de primeira ordem. Alguns detalhes de formalização não serão apresentados; apenas as idéias mais gerais. Uma referência em português é a obra de Nolt e Rohatyn [NOL91], pois aborda tanto os conceitos como os formalismos com uma linguagem de fácil entendimento, apresentando inúmeros demonstrações e exemplos e ainda propondo exercícios com respostas ao final da obra.

Uma lógica de primeira ordem (ou cálculo de predicados) é um sistema formal aplicado à definição de teorias do universo de discurso da matemática. Uma teoria é um conjunto de assertivas tradicionalmente chamadas de proposições, lemas, teoremas, etc., acerca de um universo de discurso. [GIR95] Com o uso de Lógica Clássica de Primeira Ordem, é possível representar afirmações sobre os indivíduos e suas propriedades ou relações.

A Lógica Clássica de Primeira Ordem utiliza, além dos conectivos lógicos da Lógica Clássica Proposicional, os símbolos lógicos: $\forall$ (para todo) quantificador universal; $\exists$ (existe) quantificador existencial; variáveis (para denotar indivíduos arbitrários do universo do discurso) e, opcionalmente, o símbolo de igualdade. Também são utilizados os símbolos não-lógicos constantes e símbolos funcionais (para denotar elementos específicos do universo de discurso) e símbolos predicativos para denotar relacionamentos entre indivíduos e que recebem os valores-verdade falso ou verdadeiro. [CAS87]

A sintaxe da lógica de primeira ordem baseia-se em um conjunto de variáveis individuais x_0, ..., x_n, ... junto com um conjunto R_n (para cada n>0) de símbolos de relações de *n*-lugares e um conjunto f_n de símbolos de funções *n*-lugares (para cada n=0). Mais tarde usaremos *Var* para o conjunto de variáveis e *Fun* para o conjunto de símbolos de funções. Adicionalmente, os símbolos básicos da linguagem *L* incluem os conetivos lógicos $\wedge$, $\vee$, $\neg$, $\rightarrow$, os quantificadores $\forall$ e $\exists$, e a igualdade. [TUR84]

Fórmulas bem formadas (wff) são construídas a partir de certas wff atômicas através dos conetivos e quantificadores lógicos. As wff atômicas são da forma $C(t_0, \ldots, t_{n-1})$ onde *C* é um símbolo de relação de *n*-lugares (a igualdade (=) é de dois-lugares) e cada t_i é um termo onde *t* é um termo se é uma variável individual ou se é da forma $f(t'_0, \ldots, t'_{m-1})$ onde os t'_i são termos e *f* é um símbolo de função *m*-lugares (em particular, os símbolos de função de zero-lugares são termos). Wff mais complexas são formadas por conjunção ($\wedge$), disjunção ($\vee$), negação ($\neg$) e quantificação da seguinte forma: se *A* e *B* já são wff, então *AÙB*, *AÚB*, *ØA*, *A*$\rightarrow$*B*, *"xA*, *$xA*. Onde não há perigo de confusão, não usaremos os subscritos e os superescritos nas constantes e nas variáveis.

Lógica clássica de primeira ordem é definida, especificando-se um conjunto finito de esquemas de axiomas e regras de inferência. A maneira mais comum de apresentá-lo é a seguinte [TUR84]:

2.3.1 Axiomas

Para quaisquer wff *A*, *B*, *C* de *L*

1. *A* $\rightarrow$ *(B* $\rightarrow$ *A)*
2. *(A* $\rightarrow$ *(B* $\rightarrow$ *C))* $\rightarrow$ *((A* $\rightarrow$ *B)* $\rightarrow$ *(A* $\rightarrow$ *C))*
3. *(ØB* $\rightarrow$ *ØA)* $\rightarrow$ *((ØB* $\rightarrow$ *A)* $\rightarrow$ *B)*
4. *"xA(x)* $\rightarrow$ *A(t)* onde *t* é um termo livre de *x* em *A(x)*
5. *("x)(A*$\rightarrow$*B)* $\rightarrow$ *(A* $\rightarrow$ *"xB)* onde *A* não contém ocorrências livres de *x*.

2.3.2 Regras de Inferência

MP A, A$\rightarrow$B + B

GEN A + $\forall$xA

Uma *prova* é qualquer seqüência da forma A_1, ..., A_n onde cada A_i é ou uma instância de um esquema de axiomas como segue de membros anteriores da seqüência por uma aplicação de MP ou GEN. Um *teorema* é qualquer wff que resulta de uma prova, isto é, o último membro de uma seqüência como essa. A semântica para o cálculo de predicados é fornecida pela seguinte noção.

Um *frame M de primeira ordem* consiste de um domínio *D* não-vazio, junto com uma função *F* que associa a cada símbolo de função *n*-lugares *f* uma função *f'* de $D^n \rightarrow D$ e a cada constante de relação de *n*-lugares, *C*, um elemento *C'* de 2^{Dn}.

Para fornecer a semântica para *L*, com respeito a tal frame, empregaremos uma função de associação g que associa a cada variável individual um elemento de *D*. Empregaremos a notação

M |=$_g$ A

para indicar que a função de associação *g satisfaz* a wff *A* no frame M.

1. *M* |=$_g$ *C*(t_0, ..., t_{n-1}) se e somente se *(Val* (t_0, *g*), …, *Val*(t_{n-1}, *g*)) Î *C'*.

   onde *Val(t,g)=g(t)* se *t* é uma variável individual e *f'(Val*(t'_0,*g*), ..., *Val*(t'_{m-1},*g*)) se *t* é da forma *f*(t'_0, ..., t'_{m-1}).

2. *M* |=$_g$ Ø*A* se e somente se *M* |=/=$_g$ *A*
3. *M* |=$_g$ *A*Ù*B* se e somente se *M*|=$_g$ *A* e *M* |=$_g$ *B*
4. *M* |=$_g$ "*xA* se e somente se *M*|=$_g$ (*d*|*x*)*A* para cada *d* em *D*

onde g(d|x) *é aquela função de associação idêntica a* g *exceto na variável* x; *aqui ela associa o valor* d.

As condições-verdade para os outros conetivos podem ser deduzidas a partir das equivalências

$A \vee B$? $\neg(\neg A \wedge \neg B)$

$A \rightarrow B$? $\neg A \vee B$

A? B ? $(A \rightarrow B) \wedge (B \rightarrow A)$

$\exists x A$? $\neg \forall x \neg A$

Uma wff *A* é *universalmente válida* se e somente se, para cada frame *M* e cada função de associação *g*, *M*|=$_g$ *A*.

Essas duas noções (isto é, sendo um teorema e sendo universalmente válida) são conectadas com o teorema da completude.

Teorema da Completude: Uma wff do cálculo de predicados é um teorema se e somente se ela é universalmente válida. (não será demonstrado).

Como referência para um aprofundamento maior sobre o assunto Lógica Clássica de Primeira Ordem pode-se indicar, como já observado, a obra de Nolt e Rohatyn [NOL91].

3 Fundamentos de Lógica Modal

3.1 Introdução

Uma Lógica é um sistema formal, ou seja, um sistema de manipulação de símbolos contendo uma linguagem formalmente descrita e um conjunto de regras previamente estabelecido para manipulá-los. Essas regras permitem estabelecer relações (inferências, ou seja, argumentos que têm premissas e conclusão) entre as sentenças e definem quais são as inferências válidas (ou seja, aquelas nas quais a verdade das premissas "garante" a verdade da conclusão). [SOU99]

Com a Lógica Clássica Proposicional, capturamos parte do conteúdo lógico de um discurso. Com a Lógica Clássica de Primeira Ordem, temos a possibilidade de nos referirmos a indivíduos específicos dentro do universo de discurso. Uma limitação da lógica clássica é que os valores-verdade considerados nela são somente dois, e dois extremos: verdadeiro ou falso, sem "matizes", nem valores intermediários. Ou seja, não se admitem "modalidades" dessa verdade ou dessa falsidade. A lógica que trata desses "matizes" é a lógica modal. [DEA78] Com a Lógica Modal Proposicional, aumentamos significativamente nossa capacidade de comunicação, podendo expressar as noções de necessidade, impossibilidade, contingência e possibilidade. [GIR95]

Pode-se entender a lógica modal em dois sentidos: restrito e amplo. O sentido restrito da lógica modal é o sentido clássico. Na lógica modal assim entendida, só se estudariam as chamadas "modalidades *aléticas*" ou modalidades da verdade. Modalidades de verdade seriam "necessário", "possível", "impossível" e "contingente". A lógica modal alética – organizada, como qualquer outra lógica formal, em um cálculo de enunciados e um cálculo de predicados de ordens distintas – estudaria as relações de inferência entre enunciados afetados por alguns dos operadores modais. [DEA78] Estudaremos a lógica no sentido clássico.

A Lógica Modal trata de argumentos que envolvem os conceitos de necessidade, possibilidade, impossibilidade e contingência da verdade (ou da falsidade) de uma sentença. Assim, uma sentença pode ser "(necessariamente) verdadeira", "(necessariamente) falsa", "contingente" ou "possível", conforme a necessidade da verdade da proposição em relação ao mundo a que ela se refere. [COS92] A Lógica Modal aceita todos os seus teoremas da Lógica Clássica e suplementa-os, estendendo a linguagem (apresentando novos operadores) e criando novos teoremas. [GIR95]. Dessa forma, existe um aumento acentuado da capacidade de expressar-se estruturas de espaço, estado ou tempo, comparativamente à capacidade apresentada pela lógica clássica. [DIM88]

Ela pode ser entendida como um sistema lógico específico no qual se estudam as relações de inferência entre proposições "afetadas" por operadores modais, mas também pode ser entendida como o estudo da noção de *necessidade lógica*, [DEA78] ou seja, como a lógica da necessidade e da possibilidade, do que "deve ser" e do que "pode ser". [Hughes e Cresswell apud Costa] Outra 'versão' diz que Lógica Modal pode ser descrita resumidamente como a lógica da necessidade e da possibilidade, do que "deve ser" e do que "pode ser" [HUG96].

Uma proposição que é verdadeira em todas as situações, ou seja, em todos os mundos ou estados possíveis (nos quais seja possível verificar a sua validade) é considerada *necessariamente verdadeira* (*necessária*). Analogamente, uma proposição que é falsa em todas as situações, ou seja, em todos os mundos ou estados possíveis (nos quais seja possível verificar a sua validade) é considerada *necessariamente falsa* (*impossível*). Se ela não se encaixa em nenhum dos casos acima, ela é considerada uma *contingência* (ou seja, pode ser verdadeira no estado atual, mas não em todos os estados); e se ela simplesmente não é *impossível* (*necessariamente falsa*), ela é considerada *possível*. [COS92]

Uma assertiva que é forçada a ser verdadeira em todas as situações em que pudermos verificar a sua condição de verdade é dita *necessariamente verdadeira*, ou simplesmente *necessária*. Assim, uma verdade necessária é aquela que não poderia ser falsa em nenhuma circunstância.

Se pudermos verificar que, para todas as situações, uma assertiva é falsa, dizemos que ela é *necessariamente falsa*, ou simplesmente *impossível*. Se uma assertiva não é nem impossível, nem necessária, dizemos que ela é *contingente*. Assim, uma verdade contingente é aquela que poderia ser falsa em circunstâncias específicas. Admitindo a noção de mundos possíveis onde a lógica é aplicada, podemos afirmar que uma verdade necessária é verdade em todos os mundos possíveis, e que uma verdade contingente é verdade no mundo atual, mas não se mantém verdade em todos os mundos possíveis.

Se uma assertiva não é impossível, dizemos que ela é *possível*. Essas quatro noções (necessidade, impossibilidade, contingência e possibilidade) definem os *modos* básicos nos quais uma assertiva pode ser falsa ou verdadeira e são os principais objetos de estudo da lógica modal.

A lógica modal é enriquecida pela adição de dois operadores novos: ? (é necessário que) para representar necessidade e $\Diamond$ (é possível que) para representar possibilidade, os quais, combinados com a negação, expressam as modalidades básicas de impossibilidade e contingência. [GIR95] Sob esse novo regime, a sentença A $\rightarrow$? A é tomada como axiomática. A adição de tais axiomas e de regras apropriadas de inferência envolvendo esses operadores facilita a derivação de teoremas que nem mesmo são exprimíveis na linguagem do cálculo de predicados. [COS92]

3.1.1 A noção de universo, mundos e a relação de acesso

Esta seção é baseada na descrição de lógica modal encontrada na obra de Dimuro [DIM88] e na de Giraffa [GIR95].

A semântica da Lógica Clássica Proposicional considera o significado de uma fórmula baseado no resultado de uma função de avaliação (valoração) e de sua extensão, induzida por uma estrutura (interpretação). No entanto, os operadores modais implicam afirmações a respeito de outras possíveis situações ou mundos. [GIR95]

Logo necessitamos de uma estrutura capaz de conter essas noções de mundos possíveis (acessíveis a partir do mundo atual) e de avaliação em mundos possíveis.

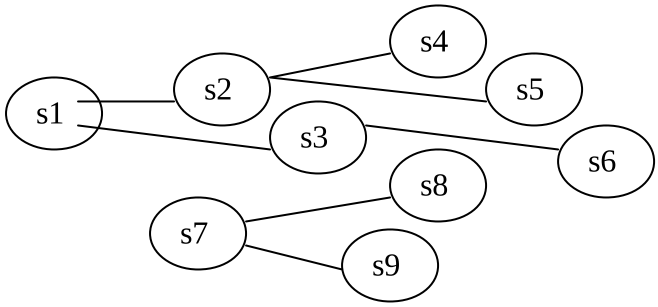

Figura 3.1: Existência ou não de uma relação de acesso para diversos mundos

De acordo com a Figura 1, podemos afirmar, por exemplo, que

s1 R s2, s1 R s3, s2 R s4, s2 R s5,

s3 R s6, s7 R s8 e s7 R s9, mas $\neg$(s1 R s7), $\neg$(s2 R s8), etc.

Uma estrutura como essa foi proposta por Kripke [KRI59], a qual é denotada por <W, R, v> para o alfabeto de uma linguagem de uma Lógica Modal Proposicional e é definida como:

a) W é um conjunto não-vazio de mundos possíveis (o "universo");
b) R é uma relação binária de acessibilidade em W (entre esses mundos), isto é, dado w $\in$ W, outros mundos w' podem ser imaginados possíveis se (w, w') pertence à relação R (fato representado por wRw'). [GIR95] Essa relação especifica a possibilidade de acessar um mundo w' a partir de um mundo w [DIM88];
c) v é uma função binária, chamada de função de avaliação, definida pela assinatura:

    v: W $\times$ p $\rightarrow$ {V,.F}, onde p é o conjunto de todos os símbolos proposicionais.

3.2 A linguagem da lógica modal Proposicional

Lógica Modal Proposicional é uma extensão da Lógica Clássica Proposicional, obtida pela adição dos operadores $\Box$ e $\Diamond$, ou seja, os símbolos utilizados para expressar sentenças em lógica modal são os mesmos utilizados na lógica clássica (com a mesmo significado definido nessa lógica) mais o operador necessidade ($\Box$ – L para alguns autores) e o operador possibilidade ($\Diamond$ – M para alguns autores). Em outras palavras [COS92]:

- um conjunto enumerável de símbolos representando proposições (diferentes dos símbolos lógicos);
- pontuação: "(" e ")";
- conectivos: "$\neg$", "$\wedge$", "$\vee$" e "$\rightarrow$"
- modais: "$\Box$" e "$\Diamond$".

3.2.1 Operadores

Denotando por [w]s o valor verdade de uma fórmula w em um mundo s, então os operadores especiais que descrevem propriedades dos mundos que são acessíveis a partir de um mundo de referência em um universo são [DIM88]:

- operador de necessidade: $\Box$. $[\Box w]s \equiv (\forall s')(R(s, s') \rightarrow [w]s')$

Assim, $\Box$w é verdadeiro no mundo s se a fórmula w é Verdadeira em todos os mundos R-acessíveis a partir de s, ou: w é necessariamente V em s se, para todo mundo s'tal que s R s', w é V em s'.

- operador de possibilidade: $\Diamond$: $[\Diamond w]s \equiv (\exists s')(R(s,s') \wedge [w]s')$

Logo, $\Diamond$w é Verdadeira em um mundo s se w é Verdadeira em pelo menos um mundo R-acessível a partir de s, ou: w é possivelmente V em s se existe pelo menos um mundo sé tal que s R s' e w é V em s'.

3.2.2 Fórmulas Modais

Uma fórmula modal é uma fórmula constituída de símbolos proposicionais, símbolos de predicados (incluindo a igualdade), símbolos funcionais, constantes individuais, variáveis individuais, operadores clássicos e operadores modais.

Uma fórmula estática é toda fórmula que não apresenta nenhum operador modal.

Uma fórmula modal plena (dinâmica) é toda fórmula constituída de subfórmulas estáticas às quais operadores clássicos e modais são aplicados [DIM88].

O conjunto de *fórmulas* válidas da Lógica Modal Proposicional é o menor conjunto de "strings" do alfabeto modal proposicional obtidas recursivamente satisfazendo as seguintes condições [COS92] e [GIR95]:

- Todo símbolo proposicional é uma fórmula, chamada *fórmula atômica*.
- Se α é uma fórmula, então $(\neg\alpha)$ também é uma fórmula.
- Se α e β são fórmulas, então $(\alpha\vee\beta)$, $(\alpha\wedge\beta)$ e $(\alpha\rightarrow\beta)$ e $(\alpha\Leftrightarrow\beta)$ também são fórmulas.
- Se α é uma fórmula, então $\Box\alpha$ e $\Diamond\alpha$ também o são.
- Nada é uma fórmula, a menos que seja forçada a ser por um dos itens acima.

Quando não existir nenhuma ambigüidade, podemos omitir alguns parênteses na escrita das fórmulas.

Como foi definido, o acréscimo à notação da lógica clássica resume-se a: se P é uma fórmula bem formada da lógica clássica ou da lógica modal, então $\Box$P e $\Diamond$P também o são na lógica modal. Simplificando a linguagem, $\Box$P será verdadeira se P for verdadeira em todos os mundos possíveis, e $\Diamond$P será verdadeira se P for verdadeira em algum dos mundos possíveis. Utilizando os símbolos apresentados, podemos dizer, por exemplo:

é necessário que P (não é possível que não-P)	$\Box P$
é possível que P (não é necessário que não-P)	$\Diamond P$
não é necessário que não-P (é possível que P)	$\neg\Box\neg P$
é impossível que P (é necessário que não-P)	$\Box\neg P$
é impossível que P (não é possível que P)	$\neg\Diamond P$
não é possível que não-P (é necessário que P)	$\neg\Diamond\neg P$

Com a ajuda da negação, todas as notações modais podem se reduzir a uma só das noções, que pode ser tanto a necessidade como a possibilidade [DEA78]. O símbolo "$\Box$" ("*é necessário que...*") poderia ser considerado o símbolo primitivo, enquanto o símbolo "$\Diamond$" ("*é possível que...*") poderia ser definido como "$\neg\Box\neg$" ("*não é necessário que não-...*"). Ou o símbolo "$\Diamond$" ("*é possível que...*") poderia ser considerado o símbolo primitivo, enquanto o símbolo "$\Box$" ("*é necessário que...*") poderia ser definido como "$\neg\Diamond\neg$" ("*não é possível que não-...*").

Conforme [DIM88], o valor verdade de uma fórmula modal em algum mundo de um dado universo é determinado pelo emprego repetido das regras para os operadores modais, e pela avaliação das subfórmulas estáticas no próprio mundo.

Assumiremos que cada mundo contém uma interpretação total de todos os símbolos clássicos na fórmula, logo o valor verdade de qualquer fórmula estática é totalmente determinado.

Extraindo um exemplo de [MANN], consideremos a existência de um universo constituído de mundos que são dias, e para cada dia, o predicado chove(1), o qual, conhecida a data, depende somente da localização l. Se o mundo de referência fixo possui significado hoje, então:

$\Box$chove(l) $\rightarrow$ $\neg$chove(l)

é interpretada como: para um dado dia e uma dada localização l, se chove naquele dia em l, então há um outro dia no futuro (depende do significado de R) no qual não irá chover em l; assim, toda chuva irá eventualmente cessar.

Similarmente, a fórmula:

chove(l) $\rightarrow$ $\Box$chove(l)

afirma que, se chove naquele dia, então irá chover para sempre (também depende de R).

Seja a fórmula geral:

$\Box$ ($\neg$w) $\Leftrightarrow$ $\neg$($\Diamond$w)

Ela afirma que todos os mundos R-acessíveis satisfazem $\neg$w se e somente se não existe um mundo R-acessível satisfazendo w. Essa fórmula é Verdadeira em qualquer mundo para qualquer universo com uma relação arbitrária R.

Algumas leis da lógica modal proposicional verificáveis intuitivamente são, por exemplo:

1. $P \rightarrow \Diamond P$ — *"ab opportere ad esse valet consequentia"*
2. $\Box P \rightarrow P$ — *"ab esse ad posee valet consequentia"*
3. $\Box (P \rightarrow Q) \rightarrow (\Box P \rightarrow \Box Q)$
4. $\neg \Diamond (P \vee Q) \Leftrightarrow (\neg \Diamond P \wedge \neg \Diamond Q)$
5. $\Diamond (P \vee Q) \Leftrightarrow (\Diamond P \vee \Diamond Q)$
6. $(\Box P \vee \Box Q) \rightarrow \Box (P \vee Q)$
7. $\Diamond (P \wedge Q) \rightarrow (\Diamond P \wedge \Diamond Q)$

3.2.3 Universo de uma Fórmula Modal

Um universo U de uma fórmula modal w consiste de um conjunto de estados (ou mundos) S, uma relação binária R em S, denominada de relação de acesso, e um domínio D.

Cada mundo s provê uma interpretação de primeira ordem sobre o domínio para todos os símbolos proposicionais, símbolos predicativos, símbolos funcionais, constantes individuais e variáveis individuais (livres) em w. [DIM88]

Um modelo (U, s0) é um universo U com um dos estados de U, so $\in$ S, designado como o mundo inicial ou de referência (interpretação).

Em resumo,

- Universo de w:
  - conjunto de mundos S
  - relação de acesso entre mundos R
  - domínio D

3.3 Semântica dos Mundos Possíveis de Kripke

Na estrutura M que determina a semântica da lógica modal, R é uma relação binária de acessibilidade sobre o conjunto W' de mundos possíveis. As propriedades formais de R determinam os axiomas e as regras de inferência da lógica modal que são considerados válidos.

Todo o esquema axiomático (tautologias e regras) da Lógica Proposicional é válido na Lógica Modal Proposicional.

Em quaisquer tipos de estruturas, são válidos os axiomas: $\neg$

$\Box P \rightarrow \neg\Diamond\neg P$

$\Box (P\rightarrow Q)\rightarrow(\Box P\rightarrow\Box Q)$

3.3.1 Valor-verdade de uma Fórmula Modal

Definimos o valor verdade de uma fórmula modal w em um mundo s (denotado por [w]s) em um universo conhecido U indutivamente:

1. Se w é estático, então o valor verdade de [w]s é calculado, interpretando w em s;
2. $[\Box w]s$ é $(\forall s')(R(s,s') \rightarrow [w]s')$;
3. $[\Diamond w]s$ é $(\exists s')(R(s, s') \wedge [w]s')$;
4. $[w1 \vee w2]$ é verdadeiro se e somente se ou [w1]s ou [w2]s é verdadeiro;
5. $[\neg w]s$ é verdadeiro se e somente se [w]s é falso;
6. $[\exists xw]s$ é verdadeiro se e somente se existe um universo U' diferente de U no máximo pela interpretação dada a x em cada mundo de U, tal que [w]s é verdadeiro em U'.

Vejamos o exemplo extraído de [MAN92]. Consideremos a interpretação da fórmula w

$$\forall x\ \Box\exists y,\ p(x,y)$$

para um universo U com um conjunto de mundos S, relação de acesso R e domínio D. O significado de [w]s, para $s \in S$, é

$$(\forall x \in D)(\forall s' \in S)(R(s, s') \rightarrow (\exists y \in D)\ [p(x, y)]s')$$

Então essa fórmula será verdadeira em um mundo s de um universo U para todas as atribuições de valores a x, e para cada mundo s' acessível a partir de s, existe uma atribuição de valores a y, possivelmente dependente da atribuição de x e da escolha de s', que tornará p(x,y) verdadeira em s'.

3.3.2 Modelo Satisfatório para uma Fórmula

Se uma fórmula w é verdadeira em algum mundo s0 em um universo U, dizemos que (U, s0) é um modelo satisfatório para aquela fórmula, ou que a fórmula é satisfeita em (U, s0).

3.3.3 Fórmula Válida

Uma fórmula w que é verdadeira em todos os mundos de todos os universos é denominada de válida, isto é, para todo universo U de w e para cada mundo s em U, [w]s é verdadeira. Por exemplo, a fórmula:

$$\Box\neg w \equiv \neg\Diamond w$$

é uma fórmula válida.

Essa fórmula estabelece a conexão entre necessidade e possibilidade. Outra fórmula válida é:

$$\Box(w1 \rightarrow w2) \rightarrow (\Box w1 \rightarrow \Box w2) \qquad \text{Normalidade}$$

Isto é, se, em todos os mundos acessíveis, é válido w1 → w2 e, se w1 é verdadeiro em todos os mundos acessíveis, então w2 deve ser também verdadeiro em todos esses mundos.

As fórmulas acima são válidas para qualquer relação de acesso. Se adicionarmos restrições à relação R, obteremos fórmulas adicionais que são verdadeiras para qualquer modelo com uma relação satisfazendo essas restrições.

De acordo com as restrições que podemos impor sobre R, obtemos diferentes sistemas modais. Neste estudo, R será estabelecida sempre reflexiva e transitiva, isto é, consideramos que uma fórmula é válida se ela é verdadeira em todos os mundos de cada universo com uma relação de acesso que é reflexiva e transitiva.

Por exemplo, a fórmula:

$$\Box w \rightarrow w$$

é válida se é verdadeira para qualquer modelo reflexivo. Ela é afirmada para um mundo s tal que, se todos os mundos acessíveis a partir de s satisfazem w, então w é satisfeita pelo próprio s, já que, pela reflexividade, s é acessível a partir de si mesmo.

A fórmula

$$\Diamond\Diamond w \rightarrow \Diamond w$$

é válida, pois é verdadeira para todos os modelos transitivos. Podemos escrever:

$$(\Diamond(\Diamond w)) \rightarrow (\Diamond w)$$

então:

$$(\exists s1)(R(s0,s1) \wedge (\Diamond w)) \rightarrow \Diamond w$$

ou

$$(\exists s1)(R(s0,s1) \wedge (\exists s2)(R(S1,s2) \wedge [w]s2)) \rightarrow (\exists s3)(R(s0,s3) \wedge [w]s3)$$

Afirma-se que, para um mundo s0, se existe um s2 acessível a partir de s1, que é acessível a partir de s0 tal que s2 satisfaz w, então existe um s3, acessível a partir de s0, que satisfaz w.

Isso sempre acontece em um modelo transitivo, pois, pela transitividade, s2 também é acessível a partir de s0 e podemos tomar s3 = s2.

A função de avaliação v pode ser estendida para permitir a avaliação de fórmulas (P) em geral (e não apenas de proposições) através da noção de satisfação (|=) em uma estrutura (M), onde a notação M $|=_w$ P indica que a estrutura M satisfaz a fórmula P no mundo w; e está definida recursivamente como [GIR95]:

a) (SA) M $|=_w$ A se e somente se v(w, A) = V;

b) (S$\neg$) M $|=_w$ ($\neg$P) se e somente se M $|\neq_w$ P;

c) (S$\wedge$) M $|=_w$ (P$\wedge$Q) se e somente se M $|=_w$ P e M $|=_w$ Q;

d) (S$\vee$) M $|=_w$ (P$\vee$Q) se e somente se M $|=_w$ P ou M $|=_w$ Q;

e) (S$\rightarrow$) M $|=_w$ (P$\rightarrow$Q) se e somente se M $|\neq_w$ P ou M $|\neq_w$ Q;

f) (S$\Leftrightarrow$) M $|=_w$ (P$\Leftrightarrow$Q) se e somente se (M $|=_w$ P e M $|=_w$ Q) ou (M $|\neq_w$ P e M $|\neq_w$ Q);

g) (S$\Diamond$) M $|=_w$ $\Diamond$P se e somente se existir w' $\in$ M tal que wRw'e M $|=_w$ P;

h) (S$\Box$) M $|=_w$ $\Box$P se e somente se para todo w'$\in$ M tal que wRw'e M $|=_w$ P;

onde A é um símbolo proposicional, P e Q são fórmulas modais proposicionais e $|\neq$ é a negação de |=.

- Dizemos que a *estrutura* M *satisfaz a fórmula* P se existe algum mundo w $\in$ W tal que M $|=_w$ P; isto é, P é verdadeira para o mundo w, na estrutura M;
- Dizemos que a *fórmula* P *é satisfatível* se existe alguma estrutura que a satisfaz; caso contrário, dizemos que a fórmula P é *insatisfatível*;
- Uma *fórmula* P *é válida em uma estrutura* M se, para todo mundo w$\in$ W, M satisfaz P, ou seja, M $|=_w$ P; dizemos que M *é um modelo para* P.

Se tomarmos uma classe de estruturas nas quais a relação de acessibilidade R entre os mundos possíveis seja somente *reflexiva* (isto é: para todo w, temos wRw, obteremos a lógica modal conhecida como *sistema T.* Se, porém, em adição, considerarmos que a relação seja também *transitiva* (isto é, para todo w, w', w" $\in$ W, temos que, se wRw' e w'Rw" então wRw"), obteremos a lógica modal conhecida como *sistema S4*. Já se R for *reflexiva*, *transitiva* e *simétrica* (isto é, para todo w, w' $\in$ W, se wRw' então w'Rw), obteremos a lógica modal conhecida como *sistema S5*.

3.3.4 Sistemas

Sistema K

Os axiomas $\Box$P$\rightarrow\neg\Diamond\neg$P e $\Box$(P$\rightarrow$Q)$\rightarrow$($\Box$P$\rightarrow\Box$Q) são válidos em qualquer classe de estruturas (sistemas), independente das características da relação de acessibilidade R.

Demonstração

a) $\Box$P$\rightarrow\neg\Diamond\neg$P

Suponhamos, por contradição, que existe uma estrutura M = <W, R, v>, com w$\in$W tal que:

$$M \;|\neq_w (\Box P \rightarrow \neg\Diamond\neg P).$$

Logo, pela definição de $|\neq$, temos:

$\neg(M \models_w (\Box P \rightarrow \neg\Diamond\neg P))$

se e somente se $\neg(M \not\models_w \Box P$ ou $M \models_w \neg\Diamond\neg P)$ (S$\rightarrow$)

se e somente se $\neg(M \not\models_w \Box P$ ou $M \not\models_w \Diamond\neg P)$ (S$\neg$)

se e somente se $\neg M \not\models_w \Box P$ e $\neg M \not\models_w \Diamond\neg P$ (De Morgan)

se e somente se $\neg\neg M \models_w \Box P$ e $\neg\neg M \models_w \Diamond\neg P$ (Definição de $|\neq$)

se e somente se $M \models_w \Box P$ e $M \models_w \Diamond\neg P$

se e somente se, para todo $w' \in W$, tal que wRw', $M \models_{w'} P$ e (S$\Box$)

existir $w'' \in W$ tal que wRw'', $M \models_{w''} \neg P$ (S$\Diamond$)

o que é uma contradição.

Portanto, $M \models_w (\Box P \rightarrow \neg\Box\neg P)$.

b) $\Box(P \rightarrow Q) \rightarrow (\Box P \rightarrow \Box Q)$

Suponhamos, por contradição, que existe uma estrutura M = <W, R, v> com $w \in W$ tal que:

$$M \not\models_w (\Box(P \rightarrow Q) \rightarrow (\Box P \rightarrow \Box Q))$$

Logo, pela definição de $|\neq$, temos:

$\neg M \not\models_w (\Box(P \rightarrow Q) \rightarrow (\Box P \rightarrow \Box Q))$

se e somente se $M \not\models_w$? $(P \rightarrow Q)$ ou $M \models_w$ (? $P \rightarrow$? Q) (S$\rightarrow$)

se e somente se $\neg M \not\models_w$? $(P \rightarrow Q)$ e $\neg M \models_w$ (? $P \rightarrow$? Q) (De Morgan)

se e somente se $\neg\neg M \not\models_w$? $(P \rightarrow Q)$ e $\neg(M \not\models_w$? P ou $M \models_w$? $Q)$ (Definição de $|\neq$ e S$\rightarrow$)

se e somente se $M \models_w$? $(P \rightarrow Q)$ e $\neg M \not\models_w$? P e $\neg M \models_w$? Q (De Morgan)

se e somente se $M \models_w$? $(P \rightarrow Q)$ e $\neg\neg M \models_w$? P e $\neg M \models_w$? Q (Definição de $|\neq$)

se e somente se $M \models_w$? $(P \rightarrow Q)$ e $M \models_w$? P e $\neg M \models_w$? Q

Mas $M \models_w$? $(P \rightarrow Q)$

se e somente se, para todo $w' \in W$, tal que wRw', $M \models_{w'} (P \rightarrow Q)$ (S?)

se e somente se, para todo $w' \in W$, tal que wRw', $M \not\models_{w'} P$ ou $M \models_{w'} Q$ (S$\rightarrow$)

se e somente se, para todo $w' \in W$, tal que wRw', $\neg M \models_{w'} P$ ou $M \models_{w'} Q$ (Definição de $|\neq$)**(1)**

Por outro lado, $M \models_w$? P e $\neg M \models_w$? Q

se e somente se, para todo $w' \in W$ tal que wRw', $M \models_{w'} P$

e $\neg$(para todo $w'' \in W$ tal que wRw'', $M \models_{w''} Q$) (S?)

se e somente se, para todo $w' \in W$ tal que wRw', $M \models_{w'} P$ e existir $w'' \in W$,

tal que wRw'', $M \models_{w''} \neg Q$ $(\neg\forall xP \Leftrightarrow \exists x\neg P)$

se e somente se, para todo, $w' \in W$, tal que wRw', $M \models w'P$,

e existir $w'' \in W$, tal que wRw'', $M \not\models w'' Q$ (Definição de $|\neq$)

que contradiz **(1)**. Portanto, não existe uma estrutura M que não satisfaça a fórmula ? (P→Q)→(? P→? Q). Logo todas as estruturas satisfazem-na.

A fórmula acima é conhecida em lógica modal com *fórmula K* e é válida em todas as estruturas, sem restrições. Denomina-se *sistema K* o sistema que tenha como exigência mínima a validação da fórmula K, e, como K mantém a regra *modus ponens* através das modalidades, esses sistemas são chamados *sistemas normais*.

Embora não apresentemos demonstração da menutenção da regra *modus ponens* através de modalidades, é de se esperar que ela seja válida para os operadores ? e ◇, devido à manutenção do esquema axiomático da Lógica Proposicional.

EXEMPLO:

Suponha que a fórmula (? P∧? (P→Q))→? Q é válida em uma estrutura M e que ? P∧? (P→Q) seja satisfeita em um mundo possível w ∈ W. Então, por S→, a fórmula ? Q também deve ser satisfeita em w, assim como ? P e ? (P→Q), por S∧.

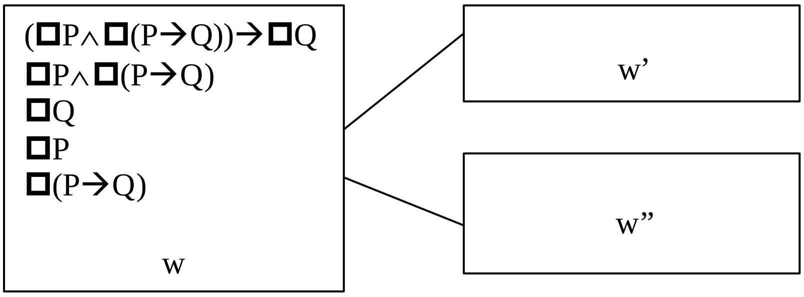

Figura 3.2: Fórmulas satisfeitas em w ∈ W.

Mas, pelo significado do operador ? , é preciso que P, P→Q e Q sejam satisfeitas em todos os mundos w' ∈ W, tal que wRw'.

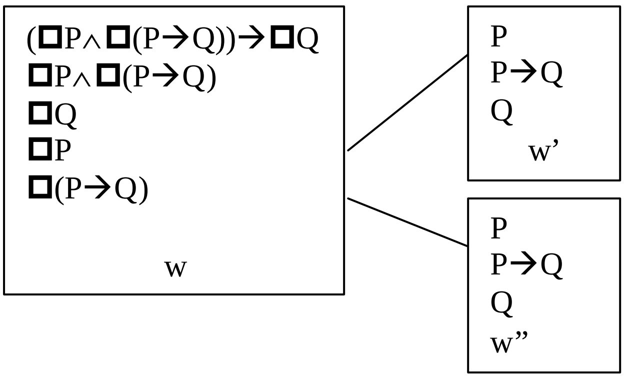

Figura 3.3: Fórmulas satisfeitas nos mundos acessíveis a partir de w.

Sistema T

A relação R é reflexiva. Provaremos que, com essa condição, a fórmula ? P→P é um axioma.

Demonstração

Suponhamos, por contradição, que exista uma estrutura M = <W, R, v> com $w \in W$, que não satisfaça ? P$\rightarrow$P. Então temos:

M $|\neq_w$ (? P$\rightarrow$P)

se e somente se $\neg$M $|=_w$ (? P$\rightarrow$p) (Definição de $|\neq$)

se e somente se $\neg$(M $|\neq_w$? P ou M $|=_w$ P) (S$\rightarrow$)

se e somente se $\neg$($\neg$M $|=_w$? P ou M $|=_w$ P) (Definição de $|\neq$)

se e somente se $\neg\neg$M $|=_w$? P e $\neg$M $|=_w$ P (De Morgan)

se e somente se M $|=_w$? P e $\neg$M $|=_w$ P

o que é uma contradição, por S? , sendo R reflexiva.

? P$\rightarrow$P é denominada *fórmula característica* do sistema T e nos dá exatamente a noção de reflexibilidade, ou seja, um mundo ser acessível por ele mesmo.

De ? P, podemos concluir P, o que nos leva à intuição de algo necessariamente verdadeiro como sendo algo que deve ser aceito em todas as situações (mundos) que consigamos imaginar (ou tivermos acesso) especialmente a atual (mundo atual).

Sistema S4

Nos sistemas de lógica modal que possuem acessibilidade reflexiva e transitiva, verifica-se que ? P$\rightarrow$? ? P (1)

Consideremos uma estrutura M = <W, R, v> com w_1, w_2, w_3, w_4 em W, e a relação de acessibilidade R transitiva tal que w_1Rw_2, w_1Rw_3, w_3Rw_1, w_2Rw_4, w_4Rw_3 e, pela transitividade, w_1Rw_1, w_1Rw_4, como pode ser observado nas figuras a seguir.

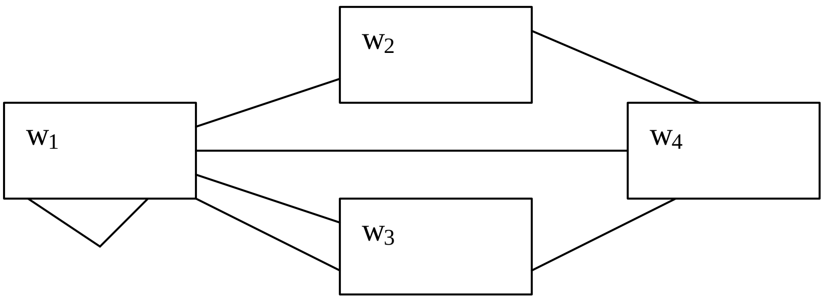

Figura 3.4: Estrutura com relação de acessibilidade transitiva.

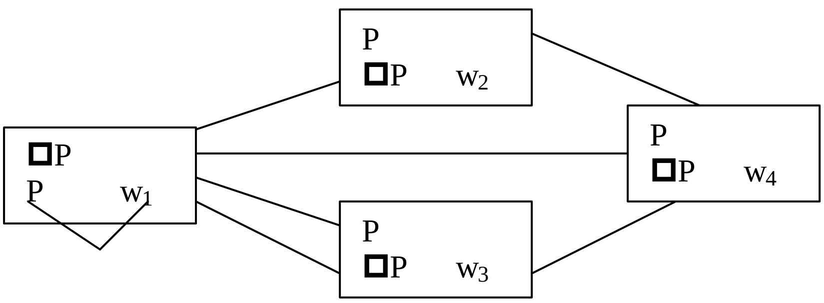

Figura 3.5: Aplicação da relação de acessibilidade transitiva (1).

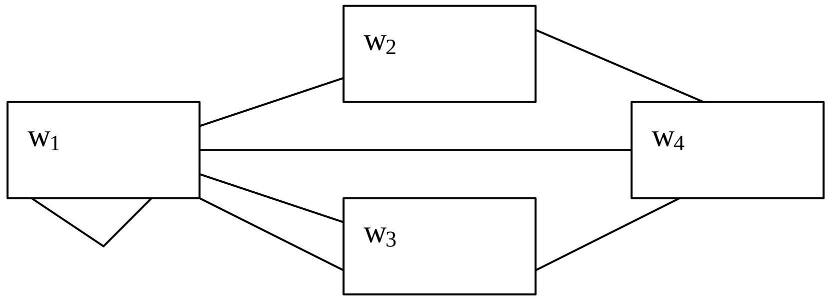

Figura 3.6: Aplicação da relação de acessibilidade transitiva (2).

Admitindo-se que o antecedente de (1) ? P é válido, então:

M |=$_{w1}$? P

Mas

M |=$_{w1}$ P, M |=$_{w2}$ P, M |=$_{w3}$ P e M |=$_{w4}$ P, em todos os mundos acessíveis.

Valendo P, em todos os mundos acessíveis, também vale ? P. (S?)

Então, temos:

M |=$_{w1}$? P, M |=$_{w2}$? P, M |=$_{w3}$? P e M |=$_{w4}$? P. (S?)

como mostra a Figura 5

Como ? P é satisfatível em todos os mundos acessíveis a partir de w1, podemos concluir por (S?) que:

M |=$_{w1}$? ? P, e (S?)

(M |=$_{w1}$? P) → (M |=$_{w1}$? ? P), (S→)

como mostra a Figura 6.

Sistema S5

Nos sistemas modais que admitem relações simétricas de acessibilidade, se existe um mundo w $\in$ W que M |=$_w$ P, então, se wRw'e wRw", por simetria temos w'Rw, w"Rw, wRw, então M |=$_w$ ◇P, M |=$_{w'}$ ◇P e M |=$_{w''}$ ◇P por S◇.

Ora, como M satisfaz a ◇P em todos os mundos possíveis, então, por S? , temos que M |=$_w$? ◇P. Pela definição de S→, podemos concluir que M |=$_w$ (◇P→? ◇P).

Logo, ◇P→? ◇P.

A tabela a seguir apresenta fórmulas características para os sistemas K, T, S4 e S5 (já demonstradas) e para os outros sistemas de Kripke para a Lógica Modal Proposicional (sem demonstração).

Sistema	Acessibilidade	Fórmulas Características
K	Nenhuma restrição	? (P→Q)→(? P→? Q)
T	Reflexiva	? P→P
D	Serial: para todo w$\in$W, há um w'$\in$W tal que wRw'	? P→◇P
K4	Transitiva	? P→? ? P
S4	Reflexiva e Transitiva	Axiomas de T e K4

KB	Simétrica	$P \rightarrow ?\ \Diamond P$
B	Reflexiva e Simétrica	Axiomas de T e KB
B4	Transitiva e Simétrica	Axiomas de K4 e KB
S5	Reflexiva, Simétrica e Transitiva	$\Diamond P \rightarrow ?\ \Diamond P$

3.4 Sistemas de Prova para Lógica Modal Proposicional

3.4.1 Axiomático

A apresentação deste assunto neste trabalho está baseada no resumo de [GIR95].

Serão considerados apenas os símbolos $\rightarrow$, $\neg$ e ? como primitivos, pois os símbolos $\wedge$, $\vee$, $\Leftrightarrow$ e $\Diamond$ podem ser deduzidos por definição como segue:

$(P \wedge Q)\ \underline{\underline{def}}\ \neg(P \rightarrow \neg Q)$

$(P \vee Q)\ \underline{\underline{def}}\ \neg P \rightarrow Q$

$(P \Leftrightarrow Q)\ \underline{\underline{def}}\ (P \rightarrow Q) \wedge (Q \rightarrow P)$

$\Diamond P\ \underline{\underline{def}}\ \neg ?\ \neg P$

Sistema K

O conjunto de axiomas e regras de inferência é o mesmo da representação axiomática da Lógica Proposicional, acrescido da fórmula característica do sistema K e da regra de inferência de *necessitação*.

a) Axiomas:

tautologias (da Lógica Proposicional) (tautologias)

$?\ (P \rightarrow Q) \rightarrow (?\ P \rightarrow ?\ Q)$ (K)

b) Regras de inferência

se $\Gamma \vdash P, \Gamma \vdash (P \rightarrow Q)$ **então** $\Gamma \vdash Q$ (Modus Ponens)

se $\vdash P$ **então** $\vdash ?\ P$ (Necessitação)

onde P e Q são fórmulas modais proposicionais, e Γ é um conjunto de fórmulas.

Modus Ponens diz que, se a partir de um conjunto de fórmulas, deduz-se P e $P \rightarrow Q$, então, a partir desse mesmo conjunto de fórmulas, deduz-se Q.

Necessitação diz que, se uma fórmula é um teorema, então ela deve ser um teorema em todos os mundos possíveis. Queremos que os teoremas da Lógica Proposicional sejam não apenas aceitos na Lógica Modal Proposicional, mas também aceitos necessariamente em todos os mundos possíveis.

A tabela a seguir apresenta o sistema axiomático para as lógicas modais de outros sistemas de Kripke.

Sistema	Sistema Axiomático	
T	Axiomas e regras de inferência do sistema K e ? P→P	
D	Axiomas e regras de inferência do sistema K e ? P→◇P	(D)
K4	Axiomas e regras de inferência do sistema K e ? P→? ? P	(K4)
S4	Axiomas e regras de inferência do sistema T e o Axioma K4	
KB	Axiomas e regras de inferência do sistema K e P→? ◇P	(B)
B	Axiomas e regras de inferência do sistema T e o Axioma B	
B4	Axiomas e regras de inferência do sistema K4 e o Axioma B	
S5	Axiomas e regras de inferência do sistema S4 e ◇P→? ◇P	

Principais propriedades

O sistema dedutivo determina um conjunto gerado a partir do conjunto de axiomas, por aplicações das regras de inferência, a saber, o conjunto dos teoremas. Da mesma forma, o sistema semântico determina um conjunto das fórmulas válidas. Poderíamos perguntar, então, se o conjunto dos teoremas é um subconjunto do conjunto das fórmulas válidas. Essa questão é endereçada pelo *teorema da corretude*, que estabelece que, se uma fórmula é provada pelo sistema dedutivo, então ela é válida. Assim, o teorema da corretude vem mostrar que o sistema dedutivo está de acordo com o significado dado às fórmulas, ou, em outras palavras, que o sistema dedutivo tem “um respaldo nas observações sobre as coisas do ‘mundo real’ sendo tratado”. Entre as várias conclusões importantes que podem ser extraídas desse teorema, destaca-se a seguinte: como no mundo real não pode acontecer que algo *seja* e *não seja* ao mesmo tempo, tem-se, pela corretude, que não pode acontecer: $\vdash$ P e $\vdash \neg$P.

Da mesma forma, seria possível perguntar se o conjunto das fórmulas válidas é um subconjunto do conjunto dos teoremas. Essa questão é endereçada pelo *teorema da completude*, que vem estabelecer que o sistema dedutivo é suficiente para gerar todas as fórmulas válidas.

Corretude

Para toda linguagem modal proposicional LMP, para todo conjunto de fórmulas Γ de LMP, para toda fórmula P de LMP, se $\Gamma \vdash$ P então $\Gamma \models$ (Teorema da Corretude). Ou seja, se P é deduzida a partir de Γ, então P é satisfeita (válida) em todas as estruturas de M nas quais os membros de Γ são válidos. [GIR95]

Completude

Para toda linguagem modal proposicional LMP, para todo conjunto de fórmulas Γ de LMP, para toda fórmula P de LMP, se $\Gamma \models$ P então $\Gamma \vdash$ (Teorema da Completude). Ou seja, se P é válida em todas as estruturas de M nas quais os membros de Γ são válidos, então P pode ser deduzida logicamente a partir de Γ. [GIR95]

Demonstrações dos teoremas da corretude e da completude do sistema axiomático da lógica modal proposicional D encontram-se na obra de Costa [COS92] e não serão apresentadas neste trabalho.

3.4.2 Dedução Natural

A prova de uma fórmula da lógica modal no sistema de dedução natural pode implicar a criação de subprovas, de acordo com as modalidades envolvidas. Assim, se estamos interessados em provar $\Diamond P$, poderíamos criar uma subprova para P. Essas subprovas correspondem a uma mudança de mundo e a verificar a validade da fórmula nesse mundo.

Existem duas formas de considerarmos essas criações de subprovas. Em uma, a criação de uma subprova pode ser interpretada como sendo a mudança para um mundo específico, como se estivéssemos procurando um contra-exemplo. Assim, se em uma prova encontrarmos $\Diamond P$, podemos criar uma subprova com P e se chegarmos a uma contradição, ou seja, se escrevermos duas fórmulas do tipo P e $\neg P$, então podemos retornar, escrevendo essa contradição. O mesmo acontece com ¬? P, que podemos criar uma nova subprova com $\neg P$ devido à interdefinibilidade dos operadores modais. Para simplificar a escrita, o símbolo $\perp$ será usado para representar a contradição.

Nessa situação, se encontrarmos no desenvolvimento da prova ? P – como P deve ser válida em todo mundo acessível a partir do atual – poderemos escrever P na subprova. O mesmo raciocínio se aplica ao encontrarmos ¬? P, que nos permitirá escrever $\neg P$ na subprova. Agora, como a relação de acessibilidade varia de acordo com o sistema que se está utilizando, deve-se ter uma forma de aplicar essa regra para cada sistema da lógica modal. Por exemplo, se a relação de acessibilidade é reflexiva, então podemos escrever P na prova atual também.

Na abordagem chamada por Fitch de *I-estilo*, "as mudanças de mundo" ocorrem de uma forma muito semelhante às que ocorrem no sistema de *tableau* que será visto adiante.

A outra abordagem para criação de subprovas pode ser interpretada como a mudança para um mundo genérico qualquer. Assim, se for encontrado ? P, pode-se abrir uma subprova com P. Se nessa subprova for possível concluir Q, e como essa subprova pode ser interpretada como a verificação da fórmula em um mundo qualquer, ou seja, ela se aplica a todos os mundos acessíveis a partir do atual (mesmo que não haja nenhum), pode-se retornar e concluir ? Q.

Note que já surgiram situações de criação de subprovas na lógica clássica proposicional. Ao ser feita uma suposição, é como se se estivesse criando uma subprova – não envolvendo mudança de mundo, é claro. Como, nesse caso, não há mudança de mundo, qualquer fórmula escrita acima poderia ser escrita dentro da suposição, desde que respeitada a visibilidade do aninhamento. Essas suposições serão representadas através de caixas para facilitar a visualização. No caso de subprovas da lógica modal, para diferenciar e ainda para facilitar a visualização, a criação de uma subprova será representada por uma "caixa estrita", na qual as linhas superiores e inferiores são duplas.

Será utilizada a seguinte notação:

- as fórmulas dos tipos ? P e $\neg\Diamond Q$ serão referenciadas por "ν" e os componentes P e $\neg Q$, por "ν_0";

- as fórmulas dos tipos $\Diamond P$ e $\neg$? Q serão referenciadas por "π_0", e os componentes P e $\neg$Q, por "π_0";

O entendimento das regras de dedução natural para os sistemas de lógica modal depende fortemente do conceito de relação de acessibilidade, que é uma noção semântica. Portanto, na maioria das vezes, as regras serão explicadas através do sistema semântico.

I-estilo

As regras da abordagem *I-estilo* de dedução natural para a lógica modal proposicional serão apresentadas, iniciando pelo sistema K. Com o sistema K está contido em todos os outros sistemas que estão sendo considerados (embora haja outros sistemas – não-normais – que não contêm K [KRI59]), essas regras devem também ser aceitas nos outros sistemas, com os devidos ajustes e acréscimos necessários. Os outros sistemas serão considerados nas seções seguintes.

Sistema K

O conjunto de regras da abordagem *I-estilo* de dedução natural para o sistema K da lógica modal proposicional é formado pelas seguintes regras:

REGRAS DE INTRODUÇÃO	REGRAS DE ELIMINAÇÃO
$\wedge$I se P e Q então $P \wedge Q$	$\wedge$E se $P \wedge Q$ então P se $P \wedge Q$ então Q
$\vee$I se P então $P \vee Q$ se Q então $P \vee Q$	$\vee$E se $P \vee Q$ e [supondo P] conclui-se R e [suponto Q] conclui-se R então R
$\rightarrow$I se [supondo P] conclui-se Q então $P \rightarrow Q$	$\rightarrow$E se P e $P \rightarrow Q$ então Q
$\neg$I se[supondo P] conclui-se *contradição* então $\neg$P	$\neg$E se *contradição* então P se [supondo $\neg$P] conclui-se *contradição* então P
νI se ν_0 então ν	νE se ν então [pode-se usar ν_0 em uma subprova]
πI	πE se π então [pode-se usar π_0 em uma subprova]

REGRA DE RETORNO

RET se em uma subprova conclui-se *contradição* então *contradição*

As restrições impostas pelas regras acima são as seguintes:

- A ocorrência da fórmula ν_0 a ser usada pela regra νI não pode estar dentro do escopo de nenhuma suposição não descarregada – obviamente que se ela estiver dentro de uma suposição descarregada, pela visibilidade do aninhamento, ela também não poderá ser usada. Assim, ela não pode depender de nenhuma hipótese.
- A subprova da regra νE deve ser uma já anteriormente criada (por aplicação da regra πE).

São apresentadas algumas explicações sobre as regras acima:

- As regras $\neg$I e $\neg$E tiveram uma reformulação (não obrigatória), representando a contradição por "$\perp$".
- É desejado, certamente, que os teoremas da lógica clássica proposicional sejam também teoremas da lógica modal proposicional, uma vez que estamos usando os mesmos conectivos lógicos com os mesmos significados. Portanto, as regras da lógica clássica proposicional são também regras da lógica modal proposicional.
- É desejado não apenas que os teoremas da lógica proposicional sejam aceitos na lógica modal proposicional, mas que sejam necessariamente, ou seja, que sejam aceitos em todos os mundos possíveis. Esta é a explicação para a regra νI que é chamada *necessitação*.
- Ao aparecer uma fórmula do tipo π, pode-se afirmar que existe um mundo acessível a partir do atual no qual π_0 ocorre. Então, com a regra πE, cria-se uma subprova – vai-se para esse mundo – para verificar essa situação.
- Se em uma subprova, pode-se deduzir uma contradição ($\perp$), então as afirmações de que se dispõe sobre esse "mundo possível" são também contraditórias. Então pode-se voltar e afirmar a contradição "$\perp$". Essa é a justificativa para a regra RET.
- O componente ν_0 de uma fórmula ν deve ser aceito em todos os mundos acessíveis a partir do atual. Portanto, como uma subprova pode ser interpretada como um mundo específico, ao se obter uma fórmula ν, pode-se também escrever ν_0 em uma subprova existente, através da regra νE.
- A regra νE é, em geral, apresentada com uma regra à parte denominada *iteração*. Optou-se por essa forma de apresentação para manter a padronização da apresentação do sistema de dedução natural.

Neste trabalho, não serão apresentadas provas ou demonstrações do sistema de dedução natural; apenas serão descritas as regras de cada sistema da lógica modal proposicional. Esse tópico será tema de trabalhos futuros.

Sistemas T, D, K4, S4, KB, B e S5

Serão apresentadas a seguir as regras da abordagem I-estilo de dedução natural para os sistemas *T, D, K4, S4, KB, B e S5* da lógica modal proposicional.

Regras do sistema T: o conjunto das regras do sistema I-estilo de dedução natural para a lógica modal T é formado pelas regras do sistema K mais a regra de eliminação:

νE se ν então ν_0

Essa regra se justifica pelo fato de que, no sistema T, a relação de acessibilidade é reflexiva. Dessa forma, se uma fórmula deve ser aceita necessariamente – ou seja, em todos os mundos acessíveis – então ela deve ser aceita também no mundo atual, que é um mundo acessível, pela reflexividade de acessibilidade.

Regras do sistema D: As regras do sistema K, eliminando-se a restrição imposta à regra νE, que limita a aplicação da regra somente a subprovas já existentes. Isso se justifica pois a relação de acessibilidade, sendo serial, implica que sempre existe um próximo mundo possível. Então, ao encontrarmos uma fórmula do tipo ν, podemos criar uma subprova para ν_0, uma vez que sabemos que existe um possível mundo acessível a partir do atual.

Regras do sistema K4: As regras do sistema K mais a seguinte regra de iteração:

νIT se ν então [podemos usar ν em uma subprova]

Restrição: a subprova deve ser uma já existente.

Explicação: como a relação de acessibilidade é transitiva, se uma fórmula P é aceita necessariamente, P deve ser aceita em todos os mundos acessíveis a partir do atual. Pela transtividade, todos os mundos acessíveis a partir dos mundos acessíveis do atual são acessíveis a partir do atual. Logo P deve ser aceita também nesses mundos. E portanto P deve ser aceita necessariamente em todos os mundos acessíveis a partir do mundo atual. Dessa forma, se for encontrada uma fórmula do tipo ν em uma prova, será possível escrever ν em qualquer subprova existente.

Regras do sistema S4: As regras dos sistemas T e K4.

Regras do sistema KB: As regras do sistema K mais a regra de introdução:

πI se π_0 então pode-se utilizar π em uma subprova

Restrição: a subprova deve ser uma já existente

Explicação: supondo que uma fórmula do tipo π_0 seja escrita em uma prova. Em termos do sistema semântico, π_0 é satisfeita em um mundo possível w. Como a relação de acessibilidade é simétrica, todos os mundos acessíveis a partir de w acessam w. Então, em cada um desses mundos, a fórmula π é satisfeita. Dessa forma, se a fórmula π_0 for encontrada em uma prova, será possível escreve-la em qualquer subprova existente.

Regras do sistema B: As regras dos sistemas T e KB.

Regras do sistema S5: As regras dos sistemas B e K4.

πIT se π então [pode-se utilizar π em uma subprova]

Restrição: a subprova deve ser uma já existente

Explicação: considerando uma estrutura <W, R, v> cuja relação de acessibilidade R é simétrica e transitiva, supõe-se que existam mundos possíveis w, w' $\in$ W tal que wRw' e uma fórmula do tipo π, por exemplo, $\Diamond$P, seja satisfeita em w, ou seja w $\vdash$ $\Diamond$P. Então existe um mundo w" $\in$ W tal que wRw" e w" $\vdash$ P. Pela simetria, w'Rw e pela transitividade w'Rw". Portanto, w' $\vdash$ $\Diamond$P. Então, se a relação é transitiva e simétrica e for encontrada uma fórmula do tipo π em uma prova, pode-se escrever π em uma subprova já existente.

Observação: o conjunto de regras de dedução natural formado pelas regras dos sistemas B e K4 é suficiente para o sistema S5. A adição dessa regra ao conjunto vem facilitar sobremaneira a prova de várias fórmulas.

A-estilo

Como para a abordagem I-estilo, será feita a apresentação das regras de dedução natural *A-estilo* para a lógica modal proposicional, começando pelo sistema K, com explicações um pouco mais detalhadas e, então, serão apresentadas as regras para os outros sistemas considerados.

Sistema K

O conjunto de regras da abordagem *A-estilo* de dedução natural para o sistema K da lógica modal proposicional é formado pelas regras de dedução natural para a lógica clássica proposicional mais as seguintes regras:

REGRAS DE ELIMINAÇÃO

REGRAS DE INTRODUÇÃO	REGRAS DE ELIMINAÇÃO
νI se ν_0 foi encontrado em uma subprova então ν	νE se ν então pode-se usar ν_0 em uma subprova

Restrição: na aplicação da regra νI, ocorrência da fórmula ν_0 na subprova não pode depender de nenhuma hipótese (da qual ν não dependa).

Explicações: ao ser encontrada uma fórmula do tipo ν em uma prova, em termos de semântica isso reflete que ν é satisfeita em um mundo possível w e que ν_0 é satisfeita em todos os mundos acessíveis a partir de w. Então se pode criar uma subprova com a aplicação da regra νE e escrever a fórmula ν_0. A grande diferença para a abordagem I-estilo é que, nesta abordagem, uma subprova corresponde a um mundo acessível (a partir do atual) específico, e a subprova que se está criando agora, na abordagem A-estilo, representa um mundo acessível (a partir do atual) genérico qualquer. Assim, se for encontrada uma fórmula do tipo ν_0 em uma subprova, pode-se retornar e escrever ν por aplicação da regra νI.

Sistemas T, D, K4, S4, KB, B e S5

Serão apresentadas a seguir as regras da abordagem A-estilo de dedução natural para os sistemas *T, D, K4, S4, KB, B e S5* da lógica modal proposicional.

Regras do sistema D: As regras do sistema K mais a seguinte regra:

◇I   se ? P então ◇P

Explicação: isso se justifica pois a relação de acessibilidade sendo serial implica que sempre existe um próximo mundo possível. Então, ao ser encontrada uma fórmula do tipo ? P, como P deve ser aceita em todos os mundos possíveis a partir do atual, e uma vez que se sabe que existe um mundo acessível a partir do atual, então se pode afirmar que P é aceita nesse mundo. Portanto, P é possível, ou seja, ◇P.

Regras dos sistemas T, K4, S4, KB, B e S5: as regras específicas da abordagem A-estilo de dedução natural para os sistemas *T, K4, S4, KB, B e S5* são as mesmas de cada um dos sistemas na abordagem I-estilo.

As provas das propriedades de corretude e completude do sistema de dedução natural para lógica modal proposicional não são apresentadas neste trabalho, porém serão tema de trabalhos futuros e podem ser conferidas na obra de Costa [COS92] e em suas referências. A seguir, é apresentada uma recente "tendência" na área de dedução natural que tem o mérito de facilitar o processo de prova em lógica modal e outras lógicas utilizando-se esse sistema de provas.

Dedução Natural rotulada

O artigo "Labelled natural Deduction for Substructural Logics", de Broda, Finger e Russo [BRO98], propõe uma metodologia uniforme para desenvolver a Dedução Natural sobre a família de lógicas lineares, de relevância e intuicionista. A metodologia segue a disciplina de Sistemas Dedutivos Rotulados (LDS – Labelled Deductive Systems), na qual o processo dedutivo manipula unidades declarativas – fórmulas rotuladas de acordo com uma álgebra de rotulação. Nos sistemas descritos nesse artigo, os rótulos são tanto termos básicos como variáveis de uma dada linguagem de rotulação, e as regras de inferência manipulam fórmulas e rótulos simultaneamente, gerando (sempre que necessário) restrições sobre os rótulos utilizados nas regras.

É fornecido um conjunto de estilos de regras de inferência de Dedução Natural, e a noção de derivação é definida, associando uma "derivação estrutural" de estilo dedução natural rotulada com um conjunto de restrições geradas. Procedimentos algorítmicos, baseados em uma técnica chamada abdução de recursos, são definidos para resolver as restrições geradas com uma derivação, e suas condições de término são discutidas no artigo.

Uma derivação por dedução natural está correta em relação a uma dada lógica subestrutural se, sob as condições em que os procedimentos algorítmicos terminam, o conjunto de restrições associado é satisfeito em relação à álgebra de rotulação que o suporta. Isso é demonstrado, provando que o sistema de dedução natural é completo e correto em relação ao sistema de *tableau*.

3.4.3 Resolução

Mints, em seu artigo "*Resolution Calculi for Modal Logics*" [MIN89] propõe uma modificação no método de resolução clássico para prova de teoremas de lógica clássica e de teorias axiomáticas baseadas nela proposto por Robinson em "*A machine-ordered logic based on the resolution principle*", de 1965 (Mints já havia proposto modificações do método original, completas e corretas para lógica intuicionista e para o cálculo modal proposicional S5). A abordagem deste artigo ([MIN89]) permite a obtenção de sistemas completos e corretos para as lógicas modais mais "populares", em particular para S4, K, K4, T, Br e G. Na realidade, o que é descrito é um esquema geral que permite estabelecer a completude do cálculo construído dessa forma para sistemas modais que tenham formulações do tipo de Gentzen (as quais não são tratadas neste trabalho) com a propriedade de subfórmula. Os sistemas G e Br (que também não são tratados) são usados para ilustrar como manipular tênues violações da propriedade de subfórmulas.

3.4.4 Tableau

Esta apresentação do sistema de tableau é baseada na descrição de [COS92].

Uma refutação baseada em tableau para lógica modal requer o uso de diversos sub-tableaux. Os sub-tableaux são criados como que simulando as mudanças de mundo. Assim, ao encontrar-se, por exemplo, uma fórmula do tipo $\Diamond P$ em um ramo, como se sabe que existe um mundo possível acessível a partir do atual, onde P é aceita, pode-se criar um sub-tableau com P para verificarmos se as informações que temos a respeito desse mundo possível são contraditórias ou não. Se essas informações são contraditórias, nesse "mundo possível", então, obviamente, as informações que se tem no mundo atual – sobre esse mundo possível – também são contraditórias. Da mesma forma, se esse sub-tableau for fechado, então o ramo que lhe deu origem também é fechado. É claro após a criação de um sub-tableau, a partir do ramo X, se tivermos em X uma fórmula do tipo ? P, como P deve ser aceita em todos os mundos possíveis, pode-se escrever P nesse sub-tableau, para completar as informações que se tem sobre esse mundo possível.

Se se estiver considerando o sistema K, então ao se encontrar ? P não se pode garantir, somente com essa fórmula, que existe um próximo mundo possível e, logo, só se pode escrever P em um sub-tableau já existente, ao passo que, se o sistema da lógica modal em consideração for o D, então, pela serialidade da relação de acessibilidade, pode-se garantir que sempre haverá um próximo mundo e pode-se criar um sub-tableau novo a partir de ? P. Dessa forma, o sistema que se está considerando irá influenciar na forma de criação de sub-tableaux e na forma de levar informação para esses sub-tableaux, obtendo-se assim um conjunto de regras de tableau para cada sistema da lógica modal, como será visto a seguir.

Esse mecanismo não exige a análise de todas as fórmulas através das regras de tableau para a lógica clássica (regras A e B) antes de serem aplicadas as regras para tratar as modalidades, facilitando a criação de provadores automáticos de teoremas através de estratégia linear. Portanto, adotaremos como básico o sistema de tableau no qual os sub-tableau podem ser referenciados e para tanto, faz-se necessário definir um aparato para identificação dos tableaux e suas fórmulas.

Um *prefixo* é qualquer expressão utilizada para dar nome a um tableau que pode aparecer na refutação por tableau de uma dada fórmula.

A idéia é ter um nome diferente para cada tableau de uma refutação. Dessa forma, uma fórmula P em uma refutação por tableau é unicamente identificada pelo par (γ, P), onde γ é o prefixo do tableau em que P ocorre.

Para administrar a criação de novos tableaux e adição de fórmulas a tableaux existentes, é criado o operador ρ que, aplicado a um prefixo γ e uma fórmula P produz um dos dois resultados abaixo:

- Cria um novo tableau, cujo nome é γ, começando com P, se γ não é um nome de algum tableau existente subordinado ao ramo que contém P, ou
- Adiciona P ao tableau designado pelo prefixo γ, caso contrário.

Regras do sistema de tableau para a Lógica Clássica Proposicional

Regras tipo A:	Regras tipo B:
Se $P \wedge Q$ então P e Q	Se $P \vee Q$ então P ou Q
Se $\neg(P \vee Q)$ então $\neg P$ e $\neg Q$	Se $P \rightarrow Q$ então $\neg P$ ou Q
Se $\neg(P \rightarrow Q)$ então P e $\neg Q$	Se $\neg(P \wedge Q)$ então $\neg P$ ou $\neg Q$
Se $\neg\neg P$ então P	

Sistema K: Formado pelas regras da lógica clássica mais as seguintes regras;

Regras Tipo E: Se ? P então $\rho(\gamma', P)$ Se $\neg\Diamond$P então $\rho(\gamma', \neg P)$ onde γ' é um tableau previamente gerado por aplicação da regra F a uma fórmula do ramo	Regras tipo F: se $\Diamond$P então $\rho(\gamma', P)$ se $\neg$? P então $\rho(\gamma', \neg P)$ onde γ' é um novo tableau.

Sistema T: Formado pelas regras da lógica clássica mais as seguintes regras:

Regras tipo E: Se ? P então $\rho(\gamma', P)$ Se $\neg\Diamond$P então $\rho(\gamma', \neg P)$ onde γ' é um tableau previamente gerado por aplicação da regra F a uma fórmula do ramo.	Regras tipo F: Se $\Diamond$P então $\rho(\gamma', P)$ Se $\neg$? P então $\rho(\gamma', \neg P)$ onde γ' é um novo tableau.	Regras tipo E-R (R de reflexiva): Se ? P então P Se $\neg\Diamond$P então $\neg$P

Sistema D: formado pelas regras da lógica clássica mais as seguintes regras:

Regras tipo E: Se ? P então $\rho(\gamma', P)$ Se $\neg\Diamond$P então $\rho(\gamma', \neg P)$ onde γ' é um novo tableau ou um tableau previamente gerado por aplicação da regra E ou F a uma fórmula do ramo.	Regras tipo F: Se $\Diamond$P então $\rho(\gamma', P)$ Se $\neg$? P então $\rho(\gamma', \neg P)$ onde γ' é um novo tableau.

Sistema K4 formado pelas regras da lógica clássica mais as seguintes regras:

Regras tipo E: Se ? P então $\rho(\gamma', P)$ Se $\neg\Diamond$P então $\rho(\gamma', \neg P)$ onde γ' é um tableau previamente gerado por aplicação da regra F a uma fórmula do ramo.	Regras tipo E-T (T de transitiva): Se ? P então $\rho(\gamma'$, ? P) Se $\neg\Diamond$P então $\rho(\gamma', \neg\Diamond P)$ onde γ' é um tableau previamente gerado por aplicação da regra F a uma fórmula do ramo.	Regras tipo F: Se $\Diamond$P então $\rho(\gamma', P)$ Se $\neg$? P então $\rho(\gamma', \neg P)$ onde γ' é um novo tableau.

Sistema S4: formado pelas regras dos sistemas T e K4.

Sistemas KB, B, B4 e S5

Esses sistemas foram agrupados pela dificuldade de obtenção de sistema de tableau que possa dar origem a um procedimento de decisão para essas lógicas. De fato, dentre essas lógicas com relação de acessibilidade simétrica, somente para S5 foi apresentado um sistema decidível. [COS92] Devido a tais dificuldades, os sistemas de tableau para eles não são desenvolvidos neste trabalho, podendo ser encontrados em [COS92] e em suas referências.

Completude e corretude

Costa [COS92] considera o método dos tableaux para o sistema D da lógica modal proposicional e prova a sua *corretude*, isto é, mostra que nenhuma fórmula e sua negação são ambas provadas por esse sistema. Também prova a sua *completude*, mostrando que toda fórmula válida pode ser provada por esse sistema. Assim, fica estabelecida a equivalência entre o sistema de tableau e a semântica de mundos possíveis.

As provas da completude e da corretude do sistema de tableau, assim como do sistema de dedução natural, serão tema de trabalhos futuros, não sendo, por isso, apresentadas neste trabalho. Essas provas podem ser encontradas tanto na obra de Costa [COS92] comentada acima como nas referências a outras fontes que essa obra apresenta.

4 Aplicações de Lógica Modal

4.1 Introdução

Dois aspectos podem ser distinguidos na conexão entre Lógica e Ciência da Computação [RYA92]. Um é o aspecto "fundacional", no qual a lógica é usada para dar um modelo de fenômeno de computação. Por exemplo: o isomorfismo Curry-Howard (explicado com algum detalhe em [MEN2001]) é uma descrição do paralelo entre computação e transformação de provas na lógica intuicionista. Outro aspecto é o de "sistema de raciocínio", no qual a lógica é usada para a descrição e para a implementação de sistemas que raciocinam sobre um domínio particular. Por exemplo: a lógica temporal é usada para raciocinar sobre domínios em que o tempo desempenha um papel chave; a lógica deôntica, para domínios que envolvem o comportamento normativo, etc. [SOU99]

Esta seção apresenta uma visão geral das aplicações da lógica modal nas diversas áreas da Ciência da Computação. A exposição aqui descrita é resultado da coleta de descrições feitas por diversos autores, muitas vezes encontradas na literatura na forma de comentário sobre tais aplicações idealizadas ou implementadas por outros. Os registros coletados são aqui expostos em ordem cronológica para facilitar a percepção da evolução dessas aplicações das lógicas às diversas áreas da Ciência da Computação. As seções seguintes abordam com um pouco mais de profundidade algumas das principais aplicações de lógica modal, quais sejam programação e sistemas concorrentes. A seção seguinte a essas apresenta também uma visão geral das aplicações de lógica modal na representação do raciocínio sobre o conhecimento.

Lógica modal foi empregada na forma de lógica dinâmica [HAR79] para facilitar a declaração e a prova de propriedades de programas. Programas são vistos como relações entre estados e, conseqüentemente, cada programa implicitamente induz um operador modal. Isso possibilita alguém a expressar, de uma forma mais natural, propriedades de programas tais como correção parcial.

Lógica temporal tem encontrado aplicação à especificação e à verificação de programas concorrentes [MAN79]. Eles apresentam uma forma de lógica temporal como um meio de raciocinar sobre seqüências de estados induzidos por tais programas. Halpern, Manna e Moszwkowski empregaram certas variações de lógica temporal para especificar circuitos de hardware [HAL83]. A grande força de lógicas modais e temporais refere-se ao seu poder de expressão. Nesses sistemas, é possível expressar propriedades de programas de uma maneira elegante e natural. Isso é, em grande parte, devido ao vocabulário enriquecido de tais lógicas sobre o do cálculo de predicados. A introdução de Lógica Temporal na IA tem como principal área de aplicação a formalização de eventos, ações e planos. Dois artigos importantes de menção especial aqui são os de McDermott e Allen. [TUR84]

Lógicas Modais, na forma de lógicas de conhecimento, de crenças e de ações, foram introduzidas na IA por Moore [MOO84] e [KON82]. Moore apresentou uma lógica de conhecimento, equivalente em poder à Lógica Modal S4, e desenvolveu uma forma de Lógica Modal para modelar agentes capazes de desempenhar tarefas cooperativas que envolvem a interação entre conhecimento, ação e planejamento.

Entre outras lógicas não-clássicas, as lógicas de necessidade, de possibilidade, de tempo, de conhecimento e de crença têm uma teoria semântica bem definida e limpa cuja força reside em sua base matemática. Tais lógicas podem proporcionar a construção de ferramentas de grande precisão e elegância para aplicação à formalização de aspectos do raciocínio e à representação do conhecimento. [TUR84]

Várias formas de programação em lógica foram desenvolvidas, principalmente com o objetivo de superar as perdas expressivas da capacidade de estruturas de espaço, estado ou tempo, que ocorrem na programação em lógica que se baseia somente na lógica de predicados de primeira ordem, cujo domínio é constituído por somente um mundo, e no qual o valor verdade de uma expressão nunca se modifica e não depende do lugar nem do tempo. Diversos aperfeiçoamentos foram estudados em [DIM88], cujo relatório descreve e analisa diversas propostas desenvolvidas, as quais são extensões da programação em lógica considerando a lógica modal.

Há também a lógica modal de ação, que foi desenvolvida para especificação de requisitos de sistemas de tempo real. [COS92] enfatiza uma ferramenta formal para animação de sistemas obtida através de modificações simples em um método de dedução sistemático.

Segundo Giraffa [GIR95], Lógica Modal foi empregada por Harel na forma de lógica dinâmica para facilitar o estabelecimento e a prova de propriedades na área de especificação e verificação de programas. Também foi utilizada por Moore e Konolige [COS92] na forma de lógica de conhecimento, crença e ação, em Inteligência Artificial, para o desenvolvimento de programas com facilidades para inferências com base no conhecimento de agentes e para modelar agentes capazes de executar tarefas cooperativas que envolvem a interação de conhecimento, ação e planejamento.

Mais recentemente, a modelagem de sistemas multiagentes tem sido tema de intensa investigação [HAL95]. Uma das principais áreas de aplicação das lógicas modais é a Inteligência Artificial. A formalização de propriedades relacionadas a tais sistemas tem sido efetuada através de diversas lógicas, entre as quais lógicas epistêmicas (ou “do conhecimento”), condicionais, fuzzy e temporais, entre outras. Muitas dessas lógicas são fundamentadas sobre lógica modal. É nesse contexto que se insere este trabalho, visto que o conhecimento de lógicas não-clássicas (lógicas modais) é fundamental na formalização de sistemas multiagentes e outros sistemas de computação.

4.2 Programação

Dimuro [DIM88] apresenta um estudo de extensões à programação em lógica baseadas na Lógica Modal. Expõe que, na época da publicação do trabalho, várias formas de programação em lógica eram apresentadas e desenvolvidas, assim como era ampliado o estudo de lógicas não-clássicas (de acordo com [TUR84], utiliza-se a denominação de lógica não-clássica para referir qualquer lógica que não seja a do cálculo proposicional ou de predicados, puramente.) A programação em lógica vinha sendo baseada na Lógica de Predicados de Primeira Ordem e utilizava-se somente desta como uma linguagem de programação.

Entretanto, a Lógica de Predicados de Primeira Ordem tem seu domínio em somente um mundo, no qual o valor verdade de uma expressão nunca se modifica e não depende do local e do tempo. Desta forma, existe uma perda expressiva de capacidade de estrutura de espaço, estado ou tempo. Devido a essa restrição, começou-se a estudar a semântica da Lógica Modal, que é uma lógica não-clássica, possuindo semânticas de mundos possíveis, baseada nos modelos de mundos possíveis desenvolvido inicialmente por Kripke e Hintikka.

Baseada na Lógica Modal, foi proposta por Sakakibara [SAK76] uma extensão da programação em Lógica, chamada de Programação em Lógica Modal, onde foi fornecida uma interpretação procedimental para a Lógica Modal. Neste estudo, foi empregada a semântica de mundos possíveis, com a utilização de multimundos como módulos ou como procedimentos na programação em lógica.

Uma outra abordagem foi proposta por Farinas Del Cerro [FAR86], que é uma tentativa de "casar" a Lógica Modal com PROLOG, recebendo a denominação de MOLOG. MOLOG pode ser considerada como uma linguagem do tipo PROLOG, na qual literais e cláusulas de Horn podem ser qualificados por expressões modais. A abordagem de MOLOG utiliza regras de resolução como uma regra de inferência. Algumas tarefas realizadas pelas regras do método de dedução podem ser feitas durante a análise sintática das cláusulas. Assim, um novo método de dedução, denominado Compilação, também foi introduzido por [FAR86].

A Programação em Lógica Modal é uma extensão de PROLOG baseada em lógica modal. Na verdade, é uma extensão simples e natural da programação em lógica, possuindo ótimas vantagens na análise de estruturas ou modularidade de programas e na prática de depuração de programas, baseada na semântica da lógica. [DIM88]

Existem duas formas de programação em lógica modal. Uma delas é dirigida a observar um programa escrito como axiomas e derivar teoremas como respostas desses axiomas, como em PROLOG. Por outro lado, é possível descrever um modelo de mundo possível como um programa e utilizar expressões modais interpretadas nesse modelo. A Programação em Lógica Modal desenvolvida em [SAK76] baseia-se nesse segundo tipo de abordagem.

Na época em que o artigo de Dimuro [DIM88] foi escrito, várias formas de programação em lógica vinham sendo desenvolvidas, principalmente com o objetivo de superar as perdas expressivas da capacidade de estrutura de espaço, estado ou tempo que ocorrem na programação em lógica que se baseia somente na lógica proposicional ou de predicados de primeira ordem, cujo domínio é constituído por somente um mundo, e no qual o valor verdade de uma expressão jamais se modifica e não depende do lugar e do tempo.

Diversos aperfeiçoamentos foram estudados, porém esta seção concentra-se nas propostas desenvolvidas por [FAR86] e [SAK76], que são extensões da programação em lógica considerando a lógica modal. No segundo, encontramos uma interpretação procedimental para a lógica modal, utilizando a semântica do mundo possível e, conseqüentemente, obtendo multimundos como módulos ou como procedimentos na programação em lógica.

Comparando com outros estudos de extensões, Dimuro [DIM88] obteve o argumento lógico exato da programação em lógica modal e nunca foi introduzida qualquer primitiva impura e não se constitui na forma integrada dos dois diferentes paradigmas de programação, como a programação em lógica e a programação orientada a objetos. Assim, essa programação em Lógica Modal apresentada por [SAK76] pode ser vista como uma extensão simples e natural da programação em lógica.

Dimuro [DIM88] também defende suas vantagens na análise de estruturas ou de modularidade de programas e na prática de depuração baseada na semântica da lógica. Em resumo, um programa em lógica modal de [SAK76] expressa um modelo de mundo possível, e expressões modais, nesse mundo, são interpretadas como procedimentos no modelo de mundo possível.

O procedimento de prova pode corresponder ao método *tableau* semântico. Vários relacionamentos entre mundos podem ser definidos pelas definições de relações. Além disso, a programação em Lógica Modal de [SAK76] pode manter sua consistência lógica. Essa é uma das vantagens em comparação com outras formas de introdução de primitivas impuras como os módulos na programação em lógica. Isso é de muita utilidade na depuração de programas.

Do ponto de vista de linguagem de programação, considera mundos como módulos ou procedimentos e, assim, um mundo limita o escopo de validade de cada nome de predicado, e uma definição de relação limita o escopo de avaliação do programa em cada mundo, isto é, define a relação de acesso de cada programa. Este fato é que introduz estruturas de programas e também a hierarquia na programação em lógica. Então podemos utilizar o mecanismo de multimundos na programação em Lógica Modal de [SAK76]. Isso também torna possível a polissemia de nomes de predicados.

Observando a proposta de [FAR86], podemos concluir que o MOLOG é uma extensão do PROLOG com operadores modais. No estudo de Dimuro, dois métodos foram definidos: O primeiro método utiliza resolução modal restrita a cláusulas de Horn modais, e o segundo método, baseado no primeiro, é denominado compilação e possibilita a obtenção de cláusulas de Horn modais similares a cláusulas de Horn clássicas. Para um adequado esclarecimento sobre cláusulas de Horn, uma boa referência é a própria obra de Farinas [FAR86].

A metodologia da abordagem de [FAR86] foi definida, considerando modalidades como termos que modificam ou qualificam fatos ou regras. Uma semântica em termos de mundos possíveis foi associada a cada operador modal, e foi apresentado um conjunto completo de regras de resolução e transformação de compilação, permitindo a dedução no conjunto de operadores modais. [DIM88]

Um resultado interessante desse estudo é o não fornecimento das regras de resolução ou transformações de compilação como dados para MOLOG, mas a obtenção automática do conjunto de regras de resolução e transformações de compilação para os operadores modais definidos da sua axiomatização. [DIM88]

MOLOG é uma extensão do PROLOG muito similar à Programação em Lógica Modal de [SAK76]. Entretanto, o sistema MOLOG interpreta axiomaticamente os operadores modais e difere da Programação em Lógica Modal de [SAK76], que se baseia na semântica dos mundos possíveis.

Ainda podem ser citadas também extensões do PROLOG como PROLOG/KR [NAK82], e METAPROLOG [BOW85], que estão muito relacionado entre si e com a Programação em Lógica Modal de Farinas Del Cerro. PROLOG/KR possui o conceito de "mundos" e é também baseado na semântica da lógica modal. O METAPROLOG é também uma extensão para considerar metateoria e possui "teorias", como os "mundos" aqui referidos. [DIM88]

4.3 Sistemas concorrentes

Nesta seção, é apresentada uma visão geral de como a lógica temporal, uma lógica modal específica, pode ser aplicada à formalização de sistemas concorrentes. Não são apresentados detalhes de formalismos, nem da linguagem da lógica temporal; apenas exemplos ilustrativos. Uma excelente fonte de referência para um aprofundamento em detalhes sobre o tema é a obra de Goldblatt [GOL87], em cuja bibliografia encontram-se também outras excelentes referências.

O estudo de sistemas concorrentes e distribuídos é uma área de pesquisa importante na Ciência da Computação. Um sistema concorrente consiste de um certo número de componentes autônomos que interagem entre si a fim de realizarem uma tarefa conjunta. [SOU99]

Sistemas concorrentes reais são usualmente compostos de vários processadores independentes, cada um deles executando um programa próprio. Em tais sistemas, a execução dos comandos nos diferentes processadores comumente se sobrepõe em vez de se intercalar. Informalmente, pode-se dizer que *dois eventos são "concorrentes" se eles ocorrem sem uma ordem prévia estabelecida para suas ocorrências*. [SOU99] Isso contrasta com os sistemas seqüenciais, em que quaisquer dois eventos que ocorrem em uma computação estão obrigatoriamente ordenados.

A teoria dos sistemas concorrentes consiste da formulação de modelos matemáticos abstratos e do estudo das propriedades desses modelos. Uma das principais características dos sistemas concorrentes é a concorrência, ou seja, o fato de que há vários processos envolvidos cujos eventos podem ocorrer simultaneamente. [SOU99] Além da concorrência, dois outros aspectos de interesse da teoria dos sistemas distribuídos são a causalidade e a escolha:

- a *causalidade* refere-se ao fato de que certos eventos em um sistema distribuído podem somente ocorrer em uma ordem fixa. Por exemplo, uma mensagem somente pode ser recebida após ter sido enviada. O recebimento da mensagem é dito ser *causalmente* dependente do envio da mesma.

- a *escolha* captura o fato de que sistemas podem comportar-se de uma maneira indeterminada. Ou seja, em certos pontos da computação, o sistema pode escolher entre eventos alternativos, levando a diferentes comportamentos.

Considerando-se uma *linguagem de especificação* como sendo um formalismo no qual se especificam comportamentos dos sistemas sob estudo, evitando-se referências ao método ou detalhes de implementação, [SOU99] temos que uma linguagem de especificação para sistemas distribuídos é aquela em que podem ser descritas propriedades comportamentais dos sistemas distribuídos. É necessário que a linguagem de especificação permita que se combinem especificações simples a fim de construir especificações mais complexas – o que reflete a intuição sobre sistemas grandes poderem ser partidos em subsistemas menores e mais bem gerenciáveis. [SOU99]

Além disso, tratando-se de sistemas distribuídos, há de se esperar que a linguagem de especificação possa descrever propriedades como causalidade, escolha e concorrência. É necessário, pois, que se possam especificar os relacionamentos válidos entre os estados do sistema enquanto a computação procede. Vendo-se um programa como um gerador de um conjunto de computações, espera-se que a especificação de um programa forneça uma caracterização alternativa, preferencialmente mais descritiva e menos operacional, do conjunto de comutações geradas pelo programa. Esses requisitos sugerem o uso da lógica formal com conectivos booleanos e modalidades temporais como a linguagem de especificação desejada. [SOU99]

Sistemas de transições são meios adequados para a definição dos significados de programas. Entretanto, para muitos sistemas computacionais, especialmente os que envolvem concorrência, seu comportamento em andamento como seqüências de ações ou mudanças de estados é mais importante. Lógicas Temporais são projetadas para raciocinar sobre tal comportamento. [SOU99]

A Lógica Temporal é uma lógica modal que consiste em uma linguagem formal de especificação para a descrição e análise de aspectos comportamentais e dependentes do tempo de sistemas concorrentes. Ela fornece uma maneira simples e natural, porém precisa, de falar sobre o ordenamento da ocorrência das interações, sem a adaptação de medidas absolutas de tempo [MAN92].

Como linguagem de especificação, a Lógica Temporal é apropriada e conveniente para especificar o comportamento dinâmico de programas concorrentes e descrever suas propriedades. A principal vantagem da Lógica Temporal é que ela fornece, através do uso de um conjunto especial de operadores, expressão sucinta e natural de propriedades de programas que ocorrem freqüentemente. Pode-se dizer que o objetivo da Lógica Temporal é tornar claros os raciocínios que envolvam declarações com algum aspecto temporal. [SOU99]

Há diferentes pontos de vista quanto ao desenvolvimento da Lógica Temporal e quanto a sua relação com a Lógica Modal, quais sejam [MAN92]:

- um dos pontos de vista considera a Lógica Temporal como uma evolução (um desenvolvimento) da Lógica Modal através da interpretação dos operadores modais em um contexto dependente de tempo, ou, equivalentemente, pela especialização dessa Lógica para modalidades de tempo.
- um outro ponto de vista (e uma outra motivação) para o estudo da Lógica Temporal surge a partir da análise lógica de linguagens naturais. Essa abordagem vê o desenvolvimento da Lógica Temporal como a formalização de convenções lingüísticas que se referem a tempos verbais num cálculo formal.

Apresenta-se, nesta seção, uma breve exploração do primeiro ponto de vista. O primeiro passo é considerar as Lógicas Temporais como Lógicas Modais sobre Sistemas de Transições Não-Rotuladas cujas relações de transição são ordenamentos. Considerando esses ordenamentos com temporais, e o conjunto de estados consistindo de tempos, temos a seguinte interpretação para os operadores modais: ? expressa *invariavelmente*, e $\Diamond$ expressa *eventualmente.*

Assim, um exemplo adaptado de Souza (Manna e Pnueli *apud* [SOU99]) ilustra a comparação que pode ser feita entre a expressão de uma situação utilizando-se apenas Lógica Clássica e a expressão da mesma situação com Lógica Temporal. Dado o argumento:

PREMISSAS

1. Uma requisição (*r*) do recurso *disco* não pode ser satisfeita (*s*) a não ser que *disco* tenha sido previamente liberado (*l*)
2. Toda requisição (*r*) do recurso *disco* acabará por ser satisfeita (*s*)
3. *Disco* ainda não foi liberado (*l*)
4. Porém *disco* encontra-se requisitado (*r*)

CONCLUSÃO

5. *Disco* será liberado (*l*)

Com Lógica Clássica apenas

l(t): *disco* foi liberado no instante *t*

r(t): *disco* foi requisitado no instante *t*

s(t): a requisição de *disco* foi satisfeita no instante *t*

n: constante que denota o tempo corrente

PREMISSAS

1. $\forall t\ (s(t)\ \exists t'\ (\ (t' < t) \wedge l(t')\ )\ )$
2. $\forall t\ (r(t) \rightarrow \exists t'(\ (t < t') \wedge s(t')\ )\ )$
3. $\forall t\ (\ (t < n) \rightarrow \neg l(t)\ )$
4. $\exists t\ (\ (t < n) \wedge r(t)\ )$

CONCLUSÃO

5. $\exists t\ (\ (n < t) \wedge l(t)\ )$

Informação sobre o domínio (tempo), para o argumento ser válido:

$\forall t\ (\ \neg(t < t)\ )$

$\forall t, t'(\ (t < t') \vee (t = t') \vee (t > t')\ )$

$\forall t, t', t''\ (\ (t < t') \wedge (t' < t'') \rightarrow (t < t'')\ )$

Com Lógica Temporal

? sempre

$\Diamond$ alguma vez

$?_{+}$ sempre no futuro

$\Diamond_{+}$alguma vez no futuro

$?_{-}$ sempre no passado

$\Diamond_{-}$ alguma vez no passado

PREMISSAS

1. $?\ (s \rightarrow \Diamond_{-}\ l)$
2. $?\ (r \rightarrow \Diamond_{+}\ s)$
3. $?_{-}\ \neg l$
4. $\Diamond_{-}\ r$

CONCLUSÃO

5. $\Diamond_{+}\ l$

Outro exemplo, agora adaptado de Goldblatt [GOL87], considera a seguinte descrição de programa concorrente: Há *n* processos diferentes agindo em paralelo e usando um ambiente de memória compartilhada, de forma que cada um pode alterar os valores das variáveis dos outros. Para fins ilustrativos, os processos podem ser tomados como fluxogramas disjuntos, com nodos rotulados. Um programa concorrente consiste de um número de processos maior ou igual a 1 que estão rodando em paralelo. Um estado do programa pode ser definido como um vetor de processos e variáveis. Especifica-se um rótulo para cada processo (denotando o ponto em que o processo está atualmente), e o valor atual de cada variável. Cada estado sucessivo é obtido a partir do estado predecessor pela escolha de exatamente um processador para agir a cada passo. Portanto, a partir de um estado inicial (vetor de processos e variáveis), várias seqüências diferentes de execução podem ser geradas, dependendo de qual processador for escolhido para agir a cada passo.

A sintaxe da lógica temporal (Lógica Temporal Proposicional Sobre Seqüências de Estados – LTPSSE) tem como símbolos os mesmos da lógica modal mais os seguintes operadores temporais: O??υ. As fórmulas ou sentenças válidas são da mesma forma que as da lógica modal, e incluem ainda OA e A υ B, se A e B são fórmulas. A semântica dos novos operadores (e a nova semântica dos operadores modais "Box" - ? - e "Diamond" - $\Diamond$) é a seguinte:

- ? ?significa *daqui para a frente, de agora em diante*, incluindo o momento (estado) presente;

- O? significa *próximo*, ou seja, o próximo estado;
- υ? significa *até que*;
- ◇ é equivalente a ¬? ¬, como usual, e significa *em algum ponto no futuro ou no presente*.

Um modelo sobre uma seqüência de estados é definido da maneira usual, e a relação de satisfação pode ser informalmente descrita como a mesma da lógica modal, acrescida do seguinte:

- *m* ||— ? A : "*daqui para a frente* A" é verdadeira no *j*-ésimo estado da seqüência do modelo *m* se e somente se, para todo k ≥ j, A for verdadeira no *k*-ésimo estado da seqüência do modelo;
- *m* ||— ◇A : "*em algum ponto no futuro ou no presente* A" é verdadeira no *j*-ésimo estado da seqüência do modelo *m* se e somente se, para algum k ≥ j, A for verdadeira no *k*-ésimo estado da seqüência do modelo *m*;
- *m* ||— OA : "*próximo* A" é verdadeira no *j*-ésimo estado da seqüência do modelo *m* se e somente se A for verdadeira no *j+1*-ésimo estado da seqüência do modelo *m*;
- *m* ||— AυB : "A *até que* B" é verdadeira no *j*-ésimo estado da seqüência do modelo *m* se e somente se, para algum k ≥ j, B for verdadeira no k-ésimo estado da seqüência do modelo *m* e, para todo *i* tal que k > i ≥ j, A for verdadeira no *i*-ésimo estado da seqüência do modelo *m*;

Intuitivamente, a semântica apresentada leva a interpretar ? através da relação ≤ (menor ou igual), e O pela relação *R*, em que *j R k* se e somente se *k = j+1*. *R* é funcional, e a conexão entre as duas relações é que ? é o fechamento transitivo e reflexivo de *R*.

Entre as propriedades temporais de programas, a exclusão mútua tem grande importância para a corretude dos mesmos. Para compreender sua modelagem utilizando lógica temporal, considerem-se dois processos que são executados em paralelo, como P1 e P2 no exemplo a seguir, adaptado de [SOU99]. Assuma-se que cada processo contém uma seção que inclui algumas das tarefas críticas para a cooperação dos dois processos (grifada no exemplo). A propriedade que declara que os processos nunca executarão simultaneamente suas respectivas seções críticas é chamada de *exclusão mútua* com respeito a esse par de seções críticas.

Sejam C1 e C2, respectivamente, as regiões críticas de P1 e P2, ou seja, as instruções r_3, r_4, r_5 grifadas em P1 e as instruções m_2, m_3, m_4 e m_5 grifadas em P2. Elas são seções obviamente críticas, pois fazem acessos à variável compartilhada *buffer*. A fim de obter o resultado correto, deve ser assegurado que nenhum outro acesso (ou modificação) a *buffer* seja feito durante a computação que envolve *buffer*. A propriedade de exclusão mútua para as duas seções críticas de P1 e P2 pode ser descrita por

$$? \neg(exec\mathrm{C}_1 \wedge exec\mathrm{C}_2),$$

onde a condição inicial associada à computação admissível é

$$exec\mathrm{r}_0 \wedge exec\mathrm{m}_0 \wedge buffer = \emptyset \wedge \text{semáforo} = 1 \wedge \text{contaElementos} = 0 \wedge \text{contaEspaços} = \text{N}$$

Programa Produtor-Consumidor

buffer = ∅; semáforo = 1;

contaElementos = 0;

contaEspaços = N (*número máximo de elementos no buffer*)

P1 : Produtor	**P2 : Consumidor**
r_0 : computa y_1	m_0 : requisita (contaElementos)
r_1 : requisita (contaEspaços)	m_1 : requisita (semáforo)
r_2 : requisita (semáforo)	**m_2 : y_2 := cabeça (buffer)**
r_3 : t_1 := buffer || y_1	**m_3 : t_2 := cauda (buffer)**
r_4 : buffer := t_1	**m_4 : b := t_2**
r_5 : libera (semáforo)	**m_5 : libera (semáforo)**
r_6 : libera (contaEspaços)	m_6 : libera (contaEspaços)
r_7 : vai para r_0	m_7 : computa usando y_2
	m_8 : vai para m_0

Isso declara que nunca é o caso de o programa Produtor-Consumidor, ou seja, a execução conjunta dos processos P1 e P2, alcançar as seções críticas C_1 e C_2 simultaneamente. Dessa forma, a exclusão mútua é implicada [SOU99].

Outra propriedade temporal importante de programas é a acessibilidade.

$$? \, (exec_i\,(m) \rightarrow exec_j \Diamond (m'))$$

expressa que, se o programa alcançar *m*, então eventualmente ele alcançará *m'*.

Outras propriedades que podem ser especificadas de maneira semelhante são, por exemplo: ausência de deadlock, corretude parcial, corretude total, terminação, responsividade e vivacidade. Souza [SOU99] apresenta referências para obras que tecem explicações bastante detalhadas sobre como a Lógica Temporal é utilizada para especificação e verificação de programas concorrentes. Também descreve estruturas de eventos, definindo a sua lógica associada, cuja semântica captura as noções de conflito, concorrência e conflito mínimo, entre outras.

4.4 Representação do Conhecimento

Nesta seção, são apresentados os resultados da pesquisa das aplicações de lógicas modais no campo de representação do conhecimento e de raciocínio sobre o conhecimento, especialmente sobre a obra de Joseph Y. Halpern (e seus colegas) e Melvin Fitting. As exposições são feitas através de breves resumos elaborados sobre os artigos e obras pesquisados. É interessante observar o interrelacionamento dos artigos, pois, afinal, tratam todos de uma área comum, embora "multidisciplinar" da Ciência da Computação. Os artigos foram dispostos na ordem cronológica de suas publicações para facilitar a leitura e a busca por referências entre eles próprios.

Uma importante ressalva em relação a esta seção específica é que algumas das obras pesquisadas abordam temas mais formalmente aprofundados dos sistemas de lógicas modais, alguns dos quais não são tratados por este trabalho, por isso não são apresentados os aspectos formais de modelagem das aplicações apresentadas. Esses aspectos são tema para trabalhos futuros.

O artigo "*Modelling Knowledge and Action in Distributed Systems*", de Halpern e Fagin [HAL89] apresenta um modelo formal que captura a sutil interação entre conhecimento e ação em sistemas distribuídos. Um sistema distribuído é visto como um conjunto de execuções, no qual uma execução é uma função que parte do tempo e chega a estados globais, e um estado global é uma tupla que consiste de um estado de meio ambiente e um estado local para cada processo no sistema. Esse modelo é uma generalização daquele usado em muitos artigos anteriores. Ações, nesse modelo, são associadas com funções que partem de estados globais e chegam a estados globais. Um protocolo é uma função que parte de estados locais e chega a ações.

Os autores estendem a noção padrão de protocolo definindo "protocolos baseados em conhecimento", aqueles nos quais ações de processos podem depender explicitamente de seus conhecimentos. Os protocolos baseados em conhecimento fornecem um meio natural de descrever como ações deveriam acontecer em um sistema distribuído. Finalmente, eles mostram como a noção de um protocolo que "implementa" outro pode ser capturada no modelo proposto.

O artigo "*Knowledge and Common Knowledge in a Distributed Environment*", de Halpern e Moses [HAL90] trata de raciocínio sobre o conhecimento, que parece exercer um papel fundamental nos sistemas distribuídos. Devido a isso, esse raciocínio é uma parte central dos argumentos intuitivos informais usados no projeto de protocolos distribuídos. A comunicação em um sistema distribuído pode ser vista como a ação de transformação do estado do sistema de conhecimento. O artigo apresenta um *framework* geral para a formalização e o raciocínio sobre conhecimento em sistemas distribuídos.

É defendido também que os estados de conhecimento de grupos de processadores são conceitos úteis para o projeto e a análise de protocolos distribuídos. Em particular, o "conhecimento distribuído" corresponde ao conhecimento que é "distribuído" entre os membros do grupo, enquanto o "conhecimento comum" corresponde a um fato que é "publicamente conhecido". A relação entre conhecimento comum e uma variedade de ações desejáveis em um sistema distribuído é ilustrada. Além disso, é demonstrado que, formalmente, em sistemas na prática, o conhecimento comum não é atingível. Também é apresentado e investigado um grande número de variantes mais fracas de conhecimento comum que são atingíveis em diversos casos de interesse.

Devido ao fato de a abordagem deste artigo ser bastante polêmica, ele representa uma das mais importantes referências para artigos posteriores que tratam de temas correlatos, tanto defendendo o ponto de vista apresentado neste como contestando, ou ainda, trazendo contribuições ou extensões ao problema apresentado.

O artigo "*A Theory of Knowledge and Ignorance for Many Agents*", de Halpern [HAL93], estende a noção de "conhecimento único" apresentada por Halpern e Moses em [HAL90] para muitos agentes e para um certo número de lógicas modais. Na abordagem deste artigo, "tudo o que um agente conhece é *alfa*" é verdadeiro em uma estrutura M se, em M, o agente sabe *alfa* e tem um conjunto máximo de "possibilidades". Para estender essa abordagem, é preciso definir o que conta como uma "possibilidade". No caso de um agente único, uma possibilidade pode ser identificada com uma designação de verdade. No caso de multiagentes, as coisas são mais complicadas. São, então, consideradas três noções de possibilidade (todas relacionadas).

Os autores defendem que a primeira é mais apropriada para lógicas não-introspectivas, tais como K_n, T_n, e $S4_n$, a segunda é mais apropriada para $K45_n$ e $KD45_n$, e a última é mais apropriada para $S5_n$ (onde n subscrito representa a quantidade de agentes do sistema multiagentes). Com a noção apropriada de possibilidade, demonstra-se que elas são extensões razoáveis em todos os casos. Os resultados apresentados no artigo remetem a conceitos de lógicas modais que não foram sequer mencionados neste trabalho e, portanto.

O artigo "*Common Knowledge Revisited*", de Fagin, Halpern, Moses e Vardi [FAG95] considera o paradoxo do conhecimento comum levantado por [HAL90]: pode-se demonstrar que o conhecimento comum é um pré-requisito para as atividades de coordenação e concordância do dia-a-dia, mas também se pode demonstrar que o conhecimento comum é inatingível na prática, no mundo real, por causa da imprecisão temporal. A solução desse paradoxo leva a um entendimento mais profundo da natureza do conhecimento comum e da simultaneidade, e mostra, mais uma vez, a importância do processo de modelagem.

O artigo discute duas soluções para o paradoxo: a modelagem do mundo com uma granularidade mais "grossa" e o relaxamento das exigências (requisitos) para a coordenação. Em particular, ele traz à tona a importância da granularidade na qual o tempo é modelado, e estressa a necessidade de considerar as aplicações para as quais essas noções estão sendo usadas. Além disso, utilizando-se a noção de conjunto de eventos, é possível esclarecer a tênue relação entre o conhecimento comum e a coordenação.

O artigo "*Reasoning About Knowledge: A Survey*", de Halpern [HAL95], tenta identificar e descrever algumas das tarefas comuns em que são utilizadas lógicas modais para trabalhar com raciocínio sobre conhecimento, em campos tão diferentes quanto Filosofia, Economia, Lingüística, Inteligência Artificial e Teoria da Computação, com ênfase particular nos trabalhos dos últimos cinco anos, e mais particularmente na Ciência da Computação.

Nesse apanhado, é apresentado o modelo "clássico", discorre-se sobre o Conhecimento Comum, é feita uma interpretação concreta de sistemas multi-agentes, tratando temas como tempo, conhecimento em sistemas distribuídos, conhecimento "externo", discute-se o que é conhecimento (o seu uso comum), por que o "ataque coordenado" não é possível, discutem-se soluções para isso, discute-se a incerteza (com o uso de intervalos) e a profundidade do conhecimento exigido.

Também é discutido o problema da onisciência lógica, utilizando para isso fórmulas logicamente equivalentes distinguidas por rótulos. Também são comentadas as comunicações e ações do conhecimento e a utilização de probabilidades para o tratamento do conhecimento "incerto". Seus trabalhos futuros apontam para lógicas epistêmicas (lógicas do conhecimento) que serão úteis para entender melhor aspectos da não-monotonicidade e de provas que partem de zero conhecimento.

Este artigo é, como o próprio subtítulo expressa, uma pesquisa sobre as principais idéias apresentadas na obra de mesmo nome publicada como livro (com mais de 400 páginas) pelos mesmos autores. A obra completa dedica um capítulo inteiro a cada um dos aspectos que aparecem comentados neste artigo, fornecendo uma visão, ao mesmo tempo, ampla e aprofundada das principais aplicações de lógicas modais na formalização do raciocínio sobre conhecimento.

O artigo "*Multi-Agent Only Knowing*", de Halpern e Lakemeyer [HAL96], comenta a proposta de Levesque [LEV90], que apresentou a noção de "conhecimento único", co ma meta de capturar certos tipos de raciocínio não-monotônico. A lógica de Levesque lida apenas com o caso de um único agente. Na época da publicação do artigo, tanto Halpern como Lakemeyer estavam tentando independentemente estender a lógica de Levesque para o caso de multiagentes. Embora houvesse uma grande quantidade de similaridades em suas abordagens, havia algumas diferenças significativas. Neste artigo, eles reexaminam a noção de conhecimento único, voltando as "primeiros" princípios.

Nesse processo, simplificam as provas de completude de Levesque, e apontam alguns problemas com as definições anteriores. Isso os leva a reconsiderar quais deveriam ser as propriedades de conhecimento único. Apresentam também um sistema de axiomas que capturam "desiderata" e mostram que isso tem uma semântica que lhe corresponde. O sistema de axiomas tem uma característica de interesse adicionada: ele inclui um operador modal para a satisfatibilidade, e então fornece uma axiomatização completa para a satisfatibilidade na lógica K45.

O artigo "*Modality and Databases*", de Melvin Fitting [FIT2000], apresenta uma lógica modal na qual alguém pode quantificar tanto objetos como conceitos, e oferece uma semântica e um sistema de *tableau*. é uma lógica modal natural, que estende as versões padrão e é capaz de tratar diversas dificuldades bem conhecidas com sucesso. O artigo também apresenta o uso dessa lógica modal para apresentar uma maneira diferente (melhor) de tratar bancos de dados relacionais. A idéia central é tratar registros como mundos possíveis, entradas de registros como objetos, e atributos como conceitos, no sentido modal. Uma aplicação interessante dessa idéia central é que tornaria possível uma teoria de bancos de dados relacionais satisfatória intuitivamente, e também poderia ser estendido pela introdução de tipos de mais alto nível, para lidar com atributos multivalorados e estruturas mais complexas.

Conclusões

Este trabalho apresentou um estudo introdutório sobre os fundamentos de lógica modal e sobre suas aplicações à Ciência da Computação. Esse estudo foi desenvolvido iniciando pela revisão de alguns aspectos genéricos de sistemas formais e da Lógica Clássica Proposicional, e a parte principal do trabalho apresentou os conceitos básicos das Lógicas Modais, tais como notações da linguagem, propriedades, apresentações e sistemas de prova. Buscou-se uma abordagem concisa dessas lógicas, devido aos objetivos e à abordagem introdutória deste trabalho.

A seguir, foram apresentados brevemente alguns exemplos de aplicações que já foram propostas para a utilização de lógicas modais em áreas específicas da Ciência da Computação, tais como formalização de sistemas concorrentes, programação e representação de conhecimento. Tais exemplos fizeram uso dos conceitos desenvolvidos anteriormente e puderam oferecer uma noção da relevância do tema para o desenvolvimento da Ciência da Computação.

Algumas das principais fontes de pesquisa para este trabalho são [COS92], [DEA75], [GOL87], [TUR84]. A leitura de tais obras é sugerida para um aprofundamento (embora ainda introdutório) sobre as bases teóricas do tema tratado, pelo fato de apresentarem linguagem bastante acessível. Além dessas fontes, foram consultados diversos artigos recentemente publicados, os quais abordam aspectos mais específicos de aplicações das lógicas modais.

Devido ao seu escopo limitado, o presente trabalho não apresenta aprofundamento na formalização das lógicas nem nas suas aplicações. Esse estudo de aspectos mais detalhados representa uma interessante possibilidade de trabalhos futuros sobre o tema e suas expansões.

Bibliografia

[BLO89] BLOK, W.J. Algebraizable logics. In: *Memoirs of the American Mathematical Society*. New York. Providence, R.I. vol. 77, n. 396 (jan. 1989), p. 1-78 v. 95-99

[BOW85] BOWEN, K. H. *Meta-level programming and knowledge representation*. New Generation Computing, 3 (1985), p. 359-383.

[BOY79] BOYER, R. S.; MOORE, J. S. *A computational logic*. New York: Academic Press, 1979.

[BOY88] BOYER, Robert S. *A computational logic handbook*. Boston: Academic Press, 1988. 408 p.

[BRA2001] BRASIL. MINISTÉRIO DA EDUCAÇÃO E CULTURA. COMISSÃO DE ESPECIALISTAS DE ENSINO DE COMPUTAÇÃO E INFORMÁTICA. *Diretrizes curriculares de cursos de graduação na área de computação e informática*. Disponível em: <*http://www.mec.gov.br/sesu/ftp/curdiretriz/computacao/co_diretriz.rtf*> Acessado em 12 dez.2001.

[BRO98] BRODA, Krysia; FINGER, Marcelo; RUSSO, Alessandra. Labelled Natural Deduction for Substructural Logics. In: *Logic Journal of the IGPL*. 1998.

[CAN55] CANNABRAVA, Euryalo. Lógica modal e dedução. In: *Revista Brasileira de Filosofia*. Sao Paulo vol.5, n.1 (jan./mar. 1955), p. 60-68

[CAR46] CARNAP, R. *Modalities and quantification*. J.S.L., vol. 1, n. 12, 1946.

[CAR56] CARNAP, Rudolf. *Meaning and necessity*: a study in semantics and modal logic. Chicago: The University of Chicago Press, 1956. 258p.

[CAS87] CASANOVA. M. A.; GIORNO, F. A. C.; FURTADO, A. L. *Programação em lógica e a linguagem Prolog*. Rio de Janeiro: E. Blücher, 1987.

[CHE80] CHELLAS, B. *Modal logic*: an introduction. Cambridge University Press, 1980.

[CLA82] CLARK, K. L.; TÄRNLUND, S. A. (eds.) *Logic programming*. London: Academic Press, 1982. 366P (A.P.I.C. Studies in Data Processing, n. 16).

[COS92] COSTA, Marcos Mota do Carmo. *Introdução à lógica modal aplicada à computação*. Porto Alegre: Instituto de Informática da UFRGS, 1992. 210 p. (Trabalho apresentado na Escola de Computação (8. : 1992 agosto 3-12 : Gramado, RS) ; editado por Taisy Silva Weber)

[DAL94] DALEN, D. van. *Logic and structure*. Springer Verlag, 1994. 3ª edição.

[DEA78] DEAÑO, Alfredo. *Introducción a la lógica formal*. Madri: Alianza Universidad Textos, 1978. 424p.

[DIM88] DIMURO, Graçaliz Pereira. *Um estudo de extensões à programação em lógica baseadas na lógica modal*. Porto Alegre: CPGCC da UFRGS, 1988. 48 p.

[DIV2000] DIVERIO, T. A.; MENEZES, P. F. B. *Teoria da computação*: máquinas universais e computabilidade. 2.ed. Porto Alegre: Instituto de Informática da UFRGS: Sagra Luzzatto, 2000.

[FAG95] FAGIN, Ronald; HALPERN, Joseph Y.; MOSES, Yoram; MOSHE, Y. Vardi. *Common knowledge revisited* 1995. Disponível em: <*http://www.cs.cornell.edu/home/halpern/abstract.html#journal55*>

[FAG98] FAGIN, R.; HALPERN, J.; MOSES, Y. & VARDI, M. *Reasoning about knowledge*. MIT Press, 1998.

[FAR86] FARINAS DEL CERRO, L. *Molog*: a system that extendes PROLOG with modal logic. New generation computing, 4 (1986).

[FIT2000] FITTING, Melvin. *Modality and Databases*. 2000. Disponível em: http://comet.lehman.cuny.edu/fitting. Acesso realizado em: 23.abr.2002.

[FIT83] FITTING. *Proof methods for modal and intuitionistic*. 1983.

[FIT99a] FITTING, Melvin. & MENDELSON, R. *First order modal logic*. Kluwer, 1999.

[FIT99b] FITTING, Melvin. *On quantified modal logic* /Fundamenta Informaticae/, 39:1-5-121,1999. Acesso realizado em: 13.abr.2002. Disponível em: <*http://comet.lehman.cuny.edu/fitting/bookspapers/pdf/papers/quantmodal.pdf*>

[GAB92] GABBAY, D.; HOGGER, C. & ROBINSON, J. A. *Handbook of logic in artificial intelligence and logic programming*. Oxford University Press, 1992-98. 6 v.

[GAB94a] GABBAY, D.; HODKINSON, I.; REYNOLDS, M. *Temporal logic*: mathematical foundations and computational aspects. v.1. Oxford: Oxford University Press, 1994.

[GAB94b] GABBAY, D. GUENTNER, F. *Handbook of philosophical logic*. Dordrecht: Kluwer Academic, 1994. 4v.

[GIR95] GIRAFFA, Lucia Maria Martins, LADEIRA, Marcelo, SILVEIRA, Ricardo Azambuja. *Lógica modal*. Porto Alegre: CPGCC da UFRGS, 1995. 34 p. : il. (Trabalho orientado pelo professor José Mauro Volkmer de Castilho)

[GOL87] GOLDBLATT, R. *Logics of time and computation*. CLSI Lecture Notes, 1987.

[HAA84] HAACK, S. Deviant logic. Cambridge: Cambridge University Press, 1974. In: TURNER, R. *Logics for artificial intelligence.* Great Britain: Ellis Horwood, 1984.

[HAL89] HALPERN, Joseph Y.; FAGIN, Ronald. *Modelling knowledge and action in distributed systems.* In: Distributed Computing. Springer-Verlag, 1989 (3) p. 159-177. Disponível em <*http://www.cs.cornell.edu/home/halpern/*>. Acesso em: 20.abr.2002.

[HAL90] HALPERN, Joseph Y.; MOSES, Yoram. *Knowledge and common knowledge in a distributed environment*. Journal of the ACM, 37(3): 1990, pp. 549-587. Disponível em *http://www.cs.cornell.edu/home/halpern/abstract.html#journal23*

[HAL93] HALPERN, Joseph Y. *A theory of knowledge and ignorance for many agents.* In: Proceedings of the Twelfth National Conference on Artificial Intelligence, 1993. Disponível em: <*http://www.cs.cornell.edu/home/halpern/abstract.html#journal55*> Acesso em: 20.abr.2002.

[HAL95] HALPERN, J. *Reasoning about knowledge*: a survey. In: GABBAY, D.; HOGGER, C. J.; ROBINSON, A. J. (eds.) Handbook of logic in artificial intelligence and logic programming. v.4 Oxford: Oxford University Press, 1995.

[HAL96] HALPERN, Joseph Y.; LAKEMEYER, Gerhard. *Multi-agent only knowing* (1996) Disponível em: <*http://www.cs.cornell.edu/home/halpern/abstract.html #journal55*> Acesso em: 22.abr.2002.

[HAR79] HAREL, D. *First order dynamic logic*. Berlin: Springer-Verlag, 1979. In: TURNER, R. Logics for artificial intelligence. Great Britain: Ellis Horwood, 1984.

[HEG73] HEGENBERG, L. *Lógica*: cálculo sentencial. São Paulo: EDUSP, 1973.

[HOA84] HOARE, C. A. R. *Communicating sequential process*. Prentice-Hall, 1984.

[HUG96] HUGHES, G. & CRESSWELL, M. *A new introduction to modal logic*. Routledge, 1996.

[HUT2000] HUTH, Michael. *Logic in computer science*: modelling and reasoning about systems. Cambridge: Cambridge University, c2000. 387 p.

[KOW84] KOWALSKI, R. *Predicate logic as programming language*. Proc. of IFIP, pp. 569-574, 1984.

[KRI59] KRIPKE, S. *A completeness theorem in modal logic*. Journal of Simbolic Logic 24 (1), março 1959.

[LEV90] LEVESQUE, H. J. “All I know: a study in autoepistemic logic.” *Artificial Intelligence.* 1990. n. 42 v. 3. p. 263-309.

[MAN92] MANNA, Z.; PNUELI, A. *The temporal logic of reactive and concurrent systems.* vol. 1. Specification. Springer Verlag, 1992.

[MAR96] MARX, Maarten. *Multi-dimensional modal logic*. Buenos Aires: Departamento de Computacion/Universidad de Buenos Aires, 1996. 1 v.

[MEN2001] MENEZES, P. F. B; HAEUSLER, E. H. *Teoria das Categorias para Ciência da Computação*. Porto Alegre: Instituto de Informática da UFRGS: Sagra Luzzatto, 2001. (Série Livros Didáticos, n. 12). 324p.

[MIL89] MILNER, R. *Communication and concurrency*. Prentice-Hall, 1989.

[MIN89] MINTS, G.. *Resolution calculi for modal logics*. In: American Mathematical Society Translations. Series 2. New York. Providence, R.I. vol. 143, (1989), p. 1-14 v. 149-151

[MOO84] MOORE, R. C. A. *Formal theory of knowledge and action*. In: TURNER, R. Logics for artificial intelligence. Great Britain: Ellis Horwood, 1984.

[MUK92] MUKUND, M.; THIAGARAJAN, P. S. *A logical characterization of well branching structures*. Theoretical computer science, 96, 1992.

[NAK82] NAKASHIMA, H. *PROLOG/KR* - language features. Proc. of the 1st Intern. Logic Programming Conference, Marseille, 1982-65-70.

[NOL91] NOLT, John; ROHATYN, Dennis. *Lógica*. São Paulo: Makron Books, 1991 (Coleção Schaum).

[OLI91] OLIVEIRA, A. J. F. de. *Lógica e aritmética* - uma introdução informal aos métodos formais. Gradiva, Lisboa, 1991.

[OLI94] OLIVEIRA, F. M. *Critérios de equilibração para sistemas tutores inteligentes*. Porto Alegre: CPGCC da UFRGS, 1994.

[PNU81] PNUELI, A. *The temporal semantics of concurrent programs*. Theoretical Computer Science, 13, p. 45-60.

[PRI2001] PRIEST, G. *An introduction to non-classical logic*. Cambridge University Press, 2001.

[RYA92] RYAN, M.; SADLER, M.. *Valuation systems and consequence relations*. In: Handbook of logic in computer science. Eds.: S. Abramski, D. M. Gabbay, and T. S. E. Maibaum. vol. 1, pp. 1-78, Oxford.

[SAK76] SAKAKIBARA, Yasubumi. *Programming in modal logic*: an extension of PROLOG based on modal logic. LNCCC 264, Logic Programming 76.

[SER98] SÉRATES, Jonofon. *Raciocínio lógico*: lógico matemático, lógico quantitativo, lógico numérico, lógico analítico, lógico crítico. 7. ed. Brasilia: Jonofon, 1998. 2v.

[SHA81] SHAPIRO, E. Y. *Inductive inference of theories from facts*. Research report 192, Depto of Computer Science, Yale University, 1981.

[SOU99] SOUZA, Sandro Silva de. *Um estudo introdutório de lógicas modais e temporais na formalização de sistemas concorrentes*. Porto Alegre: Instituto de Informática da UFRGS, 1999. (Projeto de Diplomação)

[STU83] STUHLMANN-LAEISZ, Rainer. *Das sein-sollen-problem*: eine modallogische studie. Stuttgart: Frrommann-Holzboog, 1983. 200 p.

[TUR84] TURNER, Raymond W. *Logics for artificial intelligence*. Chichester: Ellis Horwook, 1984. 121p.