Lógicas Modais

Prof. Mário Benevides

mario@cos.ufrj.br
16 de Outubro de 2012
UFRJ

Sumário

Capítulo 1

Introdução

Este material está sendo construido durante o curso. Faltam várias figuras, provas, exemplos e explicações. Este material deve ser usado como material suplementar.

No capítulo 2 apresentamos uma revisão sobre Lógica Clássica Proposicional e Lógica Clássica de Primeira Ordem. A intensão deste capítulo é prover material sobre estes assuntos para ajudar no entendimento do restante do curso.

Capítulo 2

Lógica Clássica

2.1 Lógica Clássica Proposicional

Neste capítulo nós apresentaremos a Lógica Clássica Proposicional. Na seção 2.1.1 nós definimos a linguagem. Na seção 2.1.2 nós apresentamos a semântica e definimos a importante noção de conseguência lógica. Na seção 2.1.3 apresentamos algoritmos para verificar conseguência lógica e satisfabilidade e discutimos a complexidades destes problenas. Na seção 2.1.4 são apresentados alguns sistemas dedutivos. Finalmente, na seção **??**, enunciamos e provamos os teoremas de Correção e Completude da Lógica Clássica Proposicional.

2.1.1 Linguagem da Lógica Clássica Proposicional

Alfabeto

Dado um conjunto Φ de símbolos proposicionais, $\Phi = \{p, q, ...\}$, o *alfabeto* sobre Φ é constituído por: cada um dos elementos de Φ; o símbolo $\perp$ (absurdo); os conectivos lógicos $\neg$ (negação), $\rightarrow$ (implicação), $\wedge$ (conjunção) e $\vee$ (disjunção) e os parênteses, como símbolos auxiliares.

Linguagem proposicional induzida pelo alfabeto sobre Φ

A *linguagem proposicional induzida pelo alfabeto sobre* Φ é definida indutivamente da seguinte forma:

$$\varphi ::= p \mid \perp \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \rightarrow \varphi_2 \mid \neg\varphi$$

Exemplo

Sócrates é um homem.
Se Sócrates é um homem então Sócrates é mortal.

$A-$Sócrates é um homem.
$B-$Sócrates é mortal.

A
$A \rightarrow B$

Algumas vezes utilizamos o conectivo *se e somente se* $\leftrightarrow$ que é definido como:

$$(\alpha \leftrightarrow \beta) \equiv ((\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha))$$

Exemplos de fórmulas bem formadas:

- $A \to B$ (não é fórmula)
- $(A) \to \neg(B)$ (não é fórmula)
- $(\neg A \vee B) \wedge (B \vee C) \to D$ (não é fórmula)
- $((A \to (B \to \neg A)) \to (A \vee B))$ (é fórmula)
- $(A \to (B \wedge C))$ (é fórmula)

Observação:

- Convenções sobre omissão de parênteses:
  $\neg > \wedge > \vee > \to$

- Parênteses mais externos podem ser omitidos:
  $A \to (B \to C) \equiv (A \to (B \to C))$

2.1.2 Semântica da Lógica Clássica Proposicional

- A semântica da lógica clássica proposicional consiste na atribuição de significado às fórmulas da linguagem.
- Isto é feito através da atribuição de valor verdade.
- Para cada fórmula é atribuído um valor verdadeiro ou falso.
  valores-verdade:
  V - verdadeiro
  F - falso
- O valor verdade de uma fórmula depende unicamente dos valores verdade atribuídos aos seus símbolos proposicionais.

Tabela Verdade

Conjunção:

A	B	$A \wedge B$
V	V	V
V	F	F
F	V	F
F	F	F

Hoje tem aula **e** hoje é quinta-feira.

Disjunção (não-exclusiva):

A	B	$A \vee B$
V	V	V
V	F	V
F	V	V
F	F	F

Hoje tem aula **ou** hoje é quinta-feira.

Negação:

A	$\neg A$
V	F
F	V

Hoje **não** tem aula.

Implicação:

A	B	$A \to B$
V	V	V
V	F	F
F	V	V
F	F	V

Existem lógicas que discordam da linha 4 Ex: 3-valores, intuicionista, relevante...

Exercício:

Construa a tabela verdade de: $(\neg A \vee B) \to C$

A	B	C	$\neg A$	$(\neg A \vee B)$	$(\neg A \vee B) \to C$
V	V	V			
V	V	F			
V	F	V			
V	F	F			
F	V	V			
F	V	F			
F	F	V			
F	F	F			

Função de Atribuição de Valor Verdade

A cada símbolo proposicional nós queremos atribuir um valor verdadeiro ou falso. Isto é feito através de uma função **v** de atribuição de valor verdade. $\mathbf{v}$:$\mathcal{P} \mapsto \{V, F\}$, onde $\mathcal{P}$ é conjunto dos símbolos proposicionais

Exemplos:$\mathbf{v}(A) = F$, $\mathbf{v}(B) = V$, $\mathbf{v}(C) = V$

Uma vez atribuído valor verdade a cada símbolo proposicional em $\mathcal{P}$, queremos estender esta atribuição para o conjunto de todas as fórmulas da linguagem proposicional, que denotaremos por W. Na definição a seguir α e β denotam fórmulas e A denota um símbolo proposicional, isto é, $\alpha,\ \beta \in W$ e $A \in \mathcal{P}$.

Definimos uma função v de atribuição de valor verdade a fórmulas da linguagem como uma extensão da função **v** tal que:

$v : W \mapsto \{V, F\}$, onde v deve satisfazer as seguintes condições:

1. $v(A) = \mathbf{v}(A), \text{se} A \in \mathcal{P}$

2. $v(\neg\alpha) = \begin{cases} V & \text{se} \quad v(\alpha) = F \\ F & \text{se} \quad v(\alpha) = V \end{cases}$

3. $v(\alpha \wedge \beta) = \begin{cases} V & \text{se} \quad v(\alpha) = v(\beta) = V \\ F & \quad \text{caso contrário} \end{cases}$

4. $v(\alpha \vee \beta) = \begin{cases} F & \text{se} \quad v(\alpha) = v(\beta) = F \\ V & \quad \text{caso contrário} \end{cases}$

5. $v(\alpha \to \beta) = \begin{cases} F & \text{se} \quad v(\alpha) = V \text{ e } v(\beta) = F \\ V & \quad \text{se} \quad \text{caso contrário} \end{cases}$

Exemplo: Ache o valor verdade da seguinte fórmula para a valoração $\mathbf{v}(A) = V,\ \mathbf{v}(B) = F,\ \mathbf{v}(C) = F$:

$$\begin{tikzcd}
 & v(A \to (B \vee \neg C)) = V \arrow[dl] \arrow[dr] & & \\
v(A) = V \arrow[d] & & v((B \vee \neg C)) = V \arrow[dl] \arrow[dr] & \\
\mathbf{v}(A) = V & v(B) = F \arrow[d] & & v(\neg C) = V \arrow[d] \\
 & \mathbf{v}(B) = F & & v(C) = F \arrow[d] \\
 & & & \mathbf{v}(C) = F
\end{tikzcd}$$

Exemplo: Ache o valor verdade da seguinte fórmula para a valoração $\mathbf{v}(A) = F,\ \mathbf{v}(B) = F, \mathbf{v}(C) = V,\ \mathbf{v}(D) = V$:

$$\begin{tikzcd}
 & & v((A \wedge \neg D) \to (\neg C \vee B)) = V \arrow[dll] \arrow[dr] & \\
v((A \wedge \neg D)) = F \arrow[d] \arrow[dr] & & & v((\neg C \vee B)) = F \arrow[dl] \arrow[d] \\
v(A) = F \arrow[d] & v(\neg D) = F \arrow[d] & v(\neg C) = F \arrow[d] & v(B) = F \arrow[d] \\
\mathbf{v}(A) = F & v(D) = V \arrow[d] & v(C) = V \arrow[d] & \mathbf{v}(B) = F \\
 & \mathbf{v}(D) = V & \mathbf{v}(C) = V &
\end{tikzcd}$$

Algoritmo para Construir Tabela Verdade

Quantas linhas possui uma tabela verdade para $(A \wedge \neg D) \rightarrow (\neg C \vee B)$?

Cada linha corresponde a uma possível atribuição de valores verdade aos símbolos proposicionais que compõe a fórmula. Como esta fórmula possui 4 símbolos proposicionais (A,B,C e D), sua tabela verdade deve ter $2^4 = 16$ linhas.

Tabela Verdade computa o valor verdade de uma fórmula para todas as possíveis atribuições **v** a seus símbolos proposicionais.

Logo, o problema de se saber todos os valores verdades de uma fórmula na lógica clássica proposicional, para todas as atribuições **v** a seus símbolos proposicionais, é decidível; o algoritmo é o seguinte:

passo 1: conte o número de símbolos proposicionais;

passo2: monte uma tabela com 2^n linhas e com quantas colunas for o número de subfórmulas da fómula;

passo 3: preencha as colunas dos símbolos proposicionais com V ou F alternando de cima para baixo VFVF para a 1a coluna, VVFF... para a 2a, VVVVFFFF para a 3a e assim por diante, nas potências de 2.

passo 4: compute o valor verdade das outras colunas usando as tabelas básicas fornecidas.

Exemplo: $(\neg A \rightarrow B) \vee C$

$2^3 = 8$

A	B	C	$\neg A$	$(\neg A \rightarrow B)$	$(\neg A \rightarrow B) \vee C$
V	V	V	F	V	V
V	V	F	F	V	V
V	F	V	F	V	V
V	F	F	F	V	V
F	V	V	V	V	V
F	V	F	V	V	V
F	F	V	V	F	V
F	F	F	V	F	F

Tautologias, Contradições, Fórmula Equivalentes

Existem fórmulas onde todas as linhas da Tabela Verdade dão verdade. Elas são verdadeiras não importando os valores verdade que atribuímos aos seus símbolos proposicionais. Estas fórmulas são chamadas **tautologias**. Da mesma forma, existem fórmulas que são sempre falsas, independente dos valores verdade atribuídos aos seus símbolos proposicionais. Estas são chamadas **contradições**. Além disso, existem fórmulas que, embora diferentes, têm tabelas verdade que coincidem linha a linha. Tais fórmulas são ditas **equivalentes**.

Exemplos:

A	$A \to A$
V	V
F	V

$A \to A$ é uma tautologia.

A	B	$B \to A$	$A \to (B \to A)$
V	V	V	V
V	F	V	V
F	V	F	V
F	F	V	V

A	B	$(B \vee A)$	$\neg(A \vee B)$	$A \wedge \neg(A \vee B)$
V	V	V	F	F
V	F	V	F	F
F	V	V	F	F
F	F	F	V	F

$A \wedge \neg(A \vee B)$ é uma contradição.

A	B	$B \wedge A$	$\neg(A \wedge B)$
V	V	V	F
V	F	F	V
F	V	F	V
F	F	F	V

A	B	$\neg A$	$\neg B$	$\neg A \vee \neg B$
V	V	F	F	F
V	F	F	V	V
F	V	V	F	V
F	F	V	V	V

$\neg(A \wedge B)$ é equivalente a $\neg A \vee \neg B$.

Definição 1. *Tautologia e Contradição:*

- *Uma fórmula α é uma tautologia se e somente se, para toda atribuição* $\boldsymbol{v}$, $v(\alpha) = V$.

- *Uma fórmula α é uma contradição se e somente se, para toda atribuição* $\boldsymbol{v}$, $v(\alpha) = F$.

Exemplos de tautologias "famosas":

- $A \vee \neg A$
- $A \rightarrow A$
- $(A \rightarrow ((A \rightarrow B) \rightarrow B)$
- $A \wedge B \rightarrow A$
- $A \wedge B \rightarrow B$
- $\neg\neg A \rightarrow A$
- $A \rightarrow A \vee B$
- $B \rightarrow A \vee B$
- $((A \rightarrow B) \wedge \neg B) \rightarrow \neg A$

Exemplos de contradições:

- $A \wedge \neg A$
- $\neg(A \rightarrow A)$
- $A \wedge (A \rightarrow B) \wedge \neg B$

Exercício
Verificar se estas fórmulas são realmente tautologias e contradições.

Definição 1. *Equivalência entre Fórmulas:*
Duas fórmulas α e β são ditas equivalentes, $\alpha \equiv \beta$, se e somente se, para toda atribuição $\boldsymbol{v}$, $v(\alpha) = v(\beta)$.

Intuitivamente, duas fórmulas são equivalentes se, linha a linha, elas tem a mesma tabela verdade.

Exemplos de equivalências:

$\neg\neg A \equiv A$

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

Exercício: Verificar se as seguintes fórmulas são equivalentes:

1. $\neg(A \wedge B) \equiv \neg A \vee \neg B$
2. $\neg(P \to Q) \equiv (P \wedge \neg Q)$
3. $P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$
4. $P \to Q \equiv \neg Q \to \neg P$
5. $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$

Observação:

Utilizando a noção de equivalência, é possível definir alguns dos conectivos a partir de outros. Por exemplo, utilizando a negação ($\neg$) e mais um conectivo qualquer ($\wedge$, $\vee$ ou $\to$) podemos definir todos os outros. Assim:

Definimos $\to$ e $\wedge$ usando $\neg$ e $\vee$

$P \to Q \equiv \neg P \vee Q$

$P \wedge Q \equiv \neg(\neg P \vee \neg Q)$

Definimos $\to$ e $\vee$ usando $\neg$ e $\wedge$

$P \to Q \equiv \neg(P \wedge \neg Q)$

$P \vee Q \equiv \neg(\neg P \wedge \neg Q)$

Definimos $\wedge$ e $\vee$ usando $\neg$ e $\to$

$P \wedge Q \equiv \neg(P \to \neg Q)$

$P \vee Q \equiv \neg P \to Q$

Exercício: Verificar as equivalências acima.

Na verdade todos os conectivos podem ser definido a partir de um único novo conectivo chamado. Isto é o que vamos ver no exercício seguinte.

Definição:

Seja α uma fórmula e Γ um conjunto de fórmulas:

1. Uma atribuição de valor verdade $\mathbf{v}$:$\mathcal{P} \mapsto \{V, F\}$ satisfaz α se e somente se $v(\alpha) = V$. E **v** <u>satisfaz</u> Γ se e somente se v satisfaz cada membro de Γ.

2. Γ é <u>satisfatível</u> se e somente se existe uma atribuição **v** que satisfaz Γ. Caso contrário, Γ é <u>insatisfatível</u>.

Definição:

Um conjunto de fórmulas $\alpha_1, \alpha_2, ..., \alpha_n$ ***implica logicamente*** numa fórmula β, $\alpha_1, \alpha_2, ..., \alpha_n \models \beta$, se somente se para toda valoração **v** se $v(\alpha_1 \wedge \alpha_2 \wedge ... \wedge \alpha_n) = V$, então $v(\beta) = V$.

Teorema:

Um conjunto de fórmulas $\alpha_1, \alpha_2, ..., \alpha_n$, implica logicamente em β, ou seja, $\alpha_1, \alpha_2, ..., \alpha_n \models \beta$ se somente se $(\alpha_1 \wedge \alpha_2 \wedge ... \wedge \alpha_n) \rightarrow \beta$ é uma tautologia.

Exemplo:

$(C \vee T) \wedge \neg T \rightarrow C$ é tautologia $\Rightarrow (C \vee T) \wedge \neg T \models C$

2.1.3 Complexidade

Nesta seção gostariamos de investigar dois problemas distintos:

Problema 1: Dada uma fórmula φ com comprimento n e uma valoração v para os símbolos proposicionais. Qual a complexidade de se calcular o valor de $v(\varphi)$ para a atribuição **v**? Calcular $\nu(\varphi, \mathbf{v})$.

Onde o comprimento de uma fórmula é o número de símbolos da fórmula, i.e., núumero de símbolos proposicionais + número de conectivos lógicos.

A seguir especificamos a função $\nu(\varphi,\mathbf{v})$ que implementa $v(\varphi)$ para a atribuição **v**.

Função $\nu(\varphi,\mathbf{v})$: **bool**

caso φ
$= P$ onde P é um símbolo proposicional, **retorna** $\mathbf{v}(P)$;
$= \neg\varphi_1$, **retornar NOT** $\nu(\varphi_1,\mathbf{v})$;
$= \varphi_1 \wedge \varphi_2$, **retornar** $\nu(\varphi_1,\mathbf{v})$ **AND** $\nu(\varphi_2,\mathbf{v})$;
$= \varphi_1 \vee \varphi_2$, **retornar** $\nu(\varphi_1,\mathbf{v})$ **OR** $\nu(\varphi_2,\mathbf{v})$;
$= \varphi_1 \rightarrow \varphi_2$, **retornar NOT** $\nu(\varphi_1,\mathbf{v})$ **OR** $\nu(\varphi_2,\mathbf{v})$;

Complexidade da função $\nu(\varphi,\mathbf{v})$ é O(n), pois a função é chamada uma vez para cada símbolo proposicional e uma vez para cada conectivo lógico.

Problema 2: Dada uma fórmula φ com comprimento n e m símbolos proposicionais. Verificar se existe alguma valoração que satisfaz φ.

Função SAT(φ): **bool**

para cada valoração **v faça**
  se $\nu(\varphi,\mathbf{v})$ **então retorna verdadeiro**
retorna falso

Complexidade da função SAT(φ)

Complexidade da função SAT $\approx$ número de valorações diferentes $\times$ complexidade de $\nu(\varphi,\mathbf{v})$
Complexidade da função SAT $\approx O(2^m) \times O(n) \approx O(2^m.n)$

Obs.:
1) problema 1 é polinomial (linear) no comprimento da fórmula;

2) problema 2 é NP completo.

2.1.4 Sistemas Dedutivos

Nas seções anteriores apresentamos a linguagem e a semântica da Lógica Clássica Proposicional. Voltaremos agora a problema central deste curso. Dado Um banco de dados (conjunto de fórmulas), BD = $\{\alpha_1, \cdots, \alpha_n\}$, e uma pergunta (fórmula), α com saber se o banco de dados implica logicamente na pergunta.

Nós já temos um algoritmo para responder BD $\models \alpha$ montando a tabela verdade para $\alpha_1 \wedge \alpha_2 \wedge ... \wedge \alpha_n \rightarrow \alpha$. Se for uma tautologia responde SIM, senão responde NÃO.

Existem várias outras formas de se responder a pergunta acima de uma forma mais sintática, as dentre estas podemos destacar:

- Dedução Natural
- Tableaux
- Resolução
- Provado do Dov Gabbay
- Axiomático

A seguir revisaremos os métodos de Tableaux e Axiomáticos que serão importantes no decorrer do nosso curso.

2.1.5 Método de Tableaux

As deduções são feitas por refutação, i.e., se queremos deduzir α a partir de um banco de fórmulas BD, BD $\vdash \alpha$, partimos da negação de α e tentamos chegar no absurdo. As deduções têm forma de árvore.

A seguir apresentamos todas as regras de de Tableaux.

Tableaux para a Lógica Proposicional Clássica

R_1

$$\frac{\alpha \wedge \beta}{\begin{array}{c}\alpha \\ \beta\end{array}}$$

R_2

$$\frac{\alpha \vee \beta}{\alpha \quad \beta}$$

R_3

$$\frac{\alpha \rightarrow \beta}{\neg\alpha \quad \beta}$$

R_4

$$\frac{\neg\neg\alpha}{\alpha}$$

R_5

$$\frac{\neg(\alpha \wedge \beta)}{\neg\alpha \quad \neg\beta}$$

R_6

$$\frac{\neg(\alpha \vee \beta)}{\begin{array}{c}\neg\alpha \\ \neg\beta\end{array}}$$

R_7

$$\frac{\neg(\alpha \rightarrow \beta)}{\begin{array}{c}\alpha \\ \neg\beta\end{array}}$$

Motivação

Se aplicarmos as regras a uma fórmula, vamos gerar uma árvore, onde cada ramo corresponde a uma ou mais valorações que satisfazem a fórmula, por isso é chamado Tableau Semântico.

Lembrando do nosso método semântico para verificar consequência lógica, i.e., dado um $BD = \{\phi_1, \cdots, \phi_n\}$ e uma pergunta φ, temos que $BD \models \varphi$ se e somente se $(\phi_1 \wedge \cdots \wedge \phi_n) \rightarrow \varphi$ é uma tautologia. Mas verificar se esta fórmula é uma tautologia é equivalente a verificar se sua negação é uma contradição. A intuição do método de Tableaux é aplicar as regras para mostrar que $\neg((\phi_1 \wedge \cdots \wedge \phi_n) \rightarrow \varphi)$ não possui nenhuma valoração que a faça verdadeira, i.e., ela é uma contradição. E portanto, $(\phi_1 \wedge \cdots \wedge \phi_n) \rightarrow \varphi$ é uma tautologia.

Definição: Um **ramo** θ de um tableaux τ é dito **fechado** se ele contiver α e $\neg\alpha$ para qualquer fórmula α.
Definição: Um **tableaux** τ é dito **fechado** se cada um dos seus ramos for fechado. E aberto caso contrário.

Método

1. O ramo inicial deve conter todas as fórmulas do BD seguidas da negação da pergunta;
2. aplique as regras as fórmulas no mesmo ramo no máximo uma vez;
3. se o tableaux fechar responda SIM;
4. se , em todos os ramos, todas as fórmulas já foram usadas uma vez e mesmo assim o tableaux não fechou responda NÃO.

Exemplo 1: $\{A \to B,\ B \to C\} \vdash A \to C$

$$\begin{tikzcd}
BD & A \to B \\
BD & B \to C \\
Neg.\ perg. & \neg(A \to C) \arrow[d, "R_7"] \\
 & A
\end{tikzcd}$$

$$\begin{tikzcd}
 & & \neg C \arrow[dl, "R_3"'] \arrow[dr, "R_3"] & \\
 & \neg B \arrow[dl, "R_3"'] \arrow[dr, "R_3"] & & \underline{C} \\
\underline{\neg A} & & \underline{B} &
\end{tikzcd}$$

Exemplo 2: A $\vee B \vdash \neg(\neg A \wedge \neg B)$

$$
\begin{tikzcd}
BD & & A \lor B & \\
Neg.\ perg. & & \neg\neg(\neg A \land \neg B) \arrow[d, "R_4"] & \\
& & (\neg A \land \neg B) \arrow[d, "R_1"] & \\
& & \neg A & \\
& & \neg B \arrow[dll, "R_2"'] \arrow[dr, "R_2"] & \\
\underline{A} & & & \underline{B}
\end{tikzcd}
$$

Este tableaux é fechado, pois todas as valorações são contraditórias, logo $A \lor B$ e $\neg(\neg A \land \neg B)$ não é satisfatível.

Teorema (Correção): se existe um tableaux fechado para BD, $\neg\alpha$., então BD $\models \alpha$.

Teorema (Completude): se BD $\models \alpha$ então existe tableaux fechado para BD, $\neg\alpha$.

O método de Tableaux é refutacionalmente completo.
Exercícios:

1. $A \to B, \neg(A \lor B) \vdash \neg(C \to A)$
2. $(P \to Q), \neg(P \leftrightarrow Q) \vdash \neg P$
3. Guga é inteligente. Guga é determinado. Se Guga é determinado e atleta então ele não é um perdedor. Guga é atleta se ele é amante do tênis. Guga é amante do tênis se é inteligente. Guga não é um perdedor?
4. $(P \to (R \to Q)), (P \to R) \vdash (P \to Q)$
5. $\neg A \lor B, \neg(B \lor \neg C), C \to D \vdash \neg A \lor D$

2.1.6 Sistema Axiomático

- Outro sistema dedutivo.
- Mais antigo e mais utilizado para fins teóricos.
- Vários axiomas e uma única regra de inferência.

Sejam α, β e γ fórmulas quaisquer da linguagem proposicional.

Axiomas Lógicos:

Implicação:
(1) $\alpha \to (\beta \to \alpha)$
(2)$(\alpha \to (\beta \to \gamma)) \to ((\alpha \to \beta) \to \alpha \to \gamma))$

Conjunção:
(3) $(\alpha \wedge \beta) \to \alpha$
(4) $(\alpha \wedge \beta) \to \beta$
(5) $\alpha \to (\beta \to (\alpha \wedge \beta))$

Disjunção:
(6) $\alpha \to \alpha \vee \beta$
(7) $\beta \to \alpha \vee \beta$
(8) $((\alpha \to \gamma) \wedge \beta \to \gamma)) \to ((\alpha \vee \beta) \to \gamma)$

Negação:
(9) $\alpha \to \neg\neg\alpha$
(10) $\neg\neg\alpha \to \alpha$
(11) $(\alpha \to \beta) \to ((\alpha \to \neg\beta) \to \neg\alpha)$

Regra de Inferência:
Modus Pones (M.P.)

$$\frac{\alpha \quad \alpha \to \beta}{\beta}$$

- Nosso cálculo dedutivo possui um conjunto infinito de axiomas lógicos. Para cada fórmula α, β e γ, nós temos axiomas diferentes.

- (1),...,(11) são chamadas de axiomas esquema.
- A única regra é a Modus Pones (M.P.).

Definição:

Uma fórmula α é dita um ***teorema*** de um conjunto de fórmulas $\Gamma(\Gamma \vdash \alpha)$ se e somente se existe uma seqüência de fórmulas $\alpha_1, ..., \alpha_n$ tal que $\alpha_n = \alpha$ e cada α_i é:

(i) uma instância de um axioma esquema;
(ii) ou for obtida por M.P. aplicada a α_l e α_k e $l, k < i$.
(iii) ou um membro de Γ.

A seqüência de fórmulas $\alpha_1, ..., \alpha_n$ é chamada de uma prova de α a partir de Γ.

Exemplos:

(1) $\Gamma = \{A \wedge B, A \to C\} \vdash C \vee D$?

1. $A \wedge B \to A$ axioma 3
2. $A \wedge B$ Γ
3. A M.P.(1,2)
4. $A \to C$ Γ
5. C M.P.(3,4)
6. $C \to (C \vee D)$ axioma 6
7. $C \vee D$ M.P.(5,6)

(2) $\vdash A \to A$

1. $A \to ((A \to A) \to A)$ axioma 1
2. $(A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))$ axioma 2
3. $(A \to (A \to A)) \to (A \to A)$ M.P.(1,2)
4. $A \to (A \to A)$ axioma 1

5. $A \to A$ M.P.(4,3)

Exercícios:

Provar usando o Método Axiomático:

1) $A \to B, B \to C \vdash A \to C$
2) $(A \vee B) \to C \vdash A \to (B \vee C)$
3) $A \to (B \vee C) \vdash (A \to B) \vee (A \to C)$

Observação: É importante notar (e possível provar) que os todos métodos dedutivos estudados para a lógica clássica proposicional são equivalentes, ou seja, uma fórmula que pode ser provada utilizando um deles, sempre poderá ser provada utilizando qualquer dos outros. Isso é importante, na medida em que nos permite provar uma determinada propriedade dos sistemas dedutivos em geral, provando-a apenas para o método axiomático, que embora difícil de ser usado na prática para provar um teorema, é bastante simples no que diz respeito à sua construção, o que facilita a demonstração de propriedades teóricas, como a completude e a corretude.

2.1.7 Relações entre Sintaxe e Semântica

Uma das aspectos mais importantes da lógica proposicional é a maneira como a sintaxe se relaciona com a semântica.

Nós queremos relacionar o fato de uma fórmula α ser um teorema de um conjunto de fórmulas Γ ($\Gamma\tilde{A}\ \alpha$) com a propriedade de α ser uma consequência lógica de Γ ($\Gamma \models \alpha$).

Teorema da Corretude

"Tudo que o cálculo dedutivo prova é semanticamente válido."
Se $\Gamma \vdash \alpha$ **então** $\Gamma \models \alpha$

Se uma fórmula é provada a partir de um conjunto de fórmulas então ela é consequência lógica deste conjunto de fórmulas.

Este teorema nos assegura que tudo que provamos no sistema dedutivo é ***correto*** em relação à semântica. Isto é, nosso sistema dedutivo só prova teoremas que semanticamente estão ***corretos***.

A prova é feita por indução no comprimento das provas. Como se prova:

1) Prova-se que os axiomas do cálculo dedutivo são semânticamente válidos, isto é, são tautologias;

2) Prova-se que as regras de inferência sempre derivam conclusões verdadeiras a partir de premissas verdadeiras.

Teorema da Completude

"Tudo que é semânticamente válido é provado pelo cálculo dedutivo."

Se $\Gamma \models \alpha$ **então** $\Gamma \vdash \alpha$

Se Γ implica logicamente em α então existe uma prova de α a partir de Γ no sistema dedutivo.

O sistema dedutivo é ***completo*** em relação à semântica pois para toda fórmula α que é consequência lógica de Γ existe uma prova α a partir de Γ no sistema dedutivo.

Tudo que é semanticamente obtido pode ser também obtido no sistema dedutivo.

Prova-se utilizaqndo-se a técnica do modelo canônico.

2.2 Lógica Clássica de Primeira Ordem

2.2.1 Linguagem

Linguagem da Lógica Clássica Proposicional
+
Variáveis
+
Cosntantes
+
Funções
+
Tabelas (Predicados)

Linguagem: alfabeto + regras gramaticais

Definição 1. *Um alfabeto de 1a ordem consiste dos seguintes conjuntos de símbolos:*

Símbolos Lógicos:

1. ***Conectivos lógicos:*** $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\neg$, $\forall$, $\exists$.
2. ***Símbolos auxiliares:*** *(e).*
3. ***Conjunto enumerável de variáveis:*** $V = \{v1, v2, ...\}$

   Símbolos não Lógicos:

4. ***Conjunto enumerável de constantes:*** $C = \{c1, c2...\}$
5. ***Conjunto enumerável de símbolos de função:*** $F = \{f1, f2, ...\}$ *A cada símbolo funcional está associado um número inteiro* $n > 0$*, chamado de aridade.*
6. ***Conjunto enumerável de símbolos predicativos (Predicados):*** $P = \{P1, P2, ...\}$. *A cada símbolo predicativo está associado um número inteiro* $n > 0$*, chamado aridade.*

Exemplo: $\forall x(\exists y ANCESTRAL(y, x) \land ANCESTRAL(Joao, Jose))$

Definição 1. *Os* ***termos*** *da linguagem de 1a ordem são definidos recursivamente como:*

(i) *toda variável e constante é um termo;*

(ii) *se* $t1, t2, ..., tn$ *são termos e* f *um símbolo funcional de aridade* n*,* $f(t1, t2, ..., tn)$ *é um termo;*

(iii) *nada mais é termo.*

Definição 1. *As* ***fórmulas*** *da lógica de 1a ordem são definidas recursivamente como:*

(i) *Se* P *é um predicado de aridade* n *e* $t1, t2, ..., tn$ *são termos, então* $P(t1, t2, ..., tn)$ *é uma fórmula chamada* ***fórmula atômica****;*

(ii) *Se* α *e* β *são fórmulas, então* $(\neg\alpha), (\alpha \land \beta), (\alpha \lor \beta), (\alpha \to \beta)$ *também são fórmulas;*

(iii) *Se alpha é uma fórmula e* x *uma variável, então* $\forall x\ \alpha$ *e* $\exists x\ \alpha$ *também são fórmulas;*

(iv) *Nada mais é fórmula*

De uma forma alternativa podemos definir a linguagem de primeira ordem por meio de uma notação BNF.

Termos:

$t ::= x \mid c \mid f(t_1, ..., t_n)$

Fórmulas:

$\alpha ::= P(t_1, ..., t_n) \mid (\alpha_1 \land \alpha_2) \mid (\alpha_1 \lor \alpha_2) \mid (\alpha_1 \to \alpha_2) \mid \neg\alpha \mid \forall x \alpha(x) \mid \exists x \alpha(x)$ onde P é um símbolo predicativo n-ário e $t_1, ..., t_n$ são termos.

Observações:

1. $\alpha \leftrightarrow \beta \equiv (\alpha \to \beta) \land (\beta \to \alpha)$

2. Convenções:

   (i) x, y, z, ... Variáveis;
   (ii) a, b, c, ... Constantes;
   (iii) f, g, h, ... Funções;
   (iv) A, B, C, P, U, ... Predicados;

Definição 1. *Dizemos que uma variável x ocorre livre em uma fórmula α se somente se:*

(i) *α é uma fórmula atômica e x ocorre em α;*

(ii) *α é uma fórmula da forma $\beta \wedge \gamma$, $\beta \vee \gamma$, $\beta \rightarrow \gamma$ e x ocorre livre em β ou γ;*

(iii) *α é uma fórmula da forma $\neg\beta$ e x ocorre livre em β;*

(iv) *α é uma fórmula da forma $\forall y\beta$ ou $\exists y\beta$ e x ocorre livre em β e $x \neq y$.*

Exemplos: x ocorre livre?

1. P(x,y) SIM
2. ∀y(P(x,y) ∧ Q(y,x) → R(y)) SIM
3. ∀y(∀x(P(x) → Q(y)) → R(x)) SIM
4. ∀y ∀z ((∀xP(x,y) → Q(z)) ∧ (Q(x) → R(x,y))) SIM
5. P(z,y) NÃO
6. ∀y ∃x (P(x,y) → Q(y)) NÃO

Definição 1. *Uma fórmula α é uma sentença (ou uma fórmula fechada) se somente se α não tem nenhuma variável ocorrendo livre.*

Definição 1. *Seja α uma fórmula, x uma variável e t um termo. Pela substituição de x por t em $\alpha(\alpha(x/t))$ entendemos a expressão resultante da troca de todas as ocorrências livres de x por t.*

Exemplos:

1. $\forall$ y(P(x, y,f(x, y))) $\rightarrow$ Q(g(x), h(g(x)) x/h(a)
   $\forall$ y(P(h(a), y, f(h(a), y)) $\rightarrow$ Q(g(h(a), h(g(h(a))))

2. $\forall$ y($\forall$ x(Q(x, y, g(z)) $\rightarrow$ P(f(x), y))) $\rightarrow$ R(y(g(x))) x/f(z)
   $\forall$ y($\forall$ x(Q(x, y, g(z)) $\rightarrow$ P(f(x), y))) $\rightarrow$ R(y(g(f(z))))

3. [$\forall$y(P(x, y, f(x,y))) $\rightarrow$ Q(y,z) x/g(z)] z/a
   $\forall$y (P(g(z), y, f(g(z), y)) $\rightarrow$ Q(y, z) z/a
   $\forall$y (P(g(a), y, f(g(a), y)) $\rightarrow$ Q(y, a)

Definição:
Uma variável x é substituível em uma fórmula α por um termo t se, para cada variável y ocorrendo em t, não existe nenhuma subfórmula de α da forma $\forall$ yβ ou $\exists$ yβ onde x ocorre livre em β.

O que queremos evitar com esta condição é que o quantificador $\forall$ y ou $\exists$ y capture alguma variável de t.

Exemplo:
($\forall$ y CHEFE(x,y) $\rightarrow$ GERENTE(x)) x/y
($\forall$ y CHEFE(y,y) $\rightarrow$ GERENTE(y))

2.2.2 Semântica da LCPO

Nesta seção apresentaremos a semântica da Lógica Clássica de Primeira Ordem somente para sentenças, isto é, fórmulas sem ocorrência de varáveis livres.

A semântica da lógica de primeira ordem tem como objetivo atribuir significados às fórmulas da linguagem.

- Uma fórmula só tem significado quando uma interpretação é dada a seus símbolos não lógicos.

- $\forall x(Q(x) \rightarrow P(x))$ é verdadeira ou falsa?

Nós só podemos dizer se esta fórmula é V ou F se interpretarmos seus símbolos não-lógicos.

Primeiro, precisamos saber qual o universo em que as variáveis estão quantificando. Por exemplo: números inteiros, números reais, pessoas...

Depois, precisamos interpretar os predicados, funções e constantes.

Exemplo: $\forall x(Q(x) \rightarrow P(x))$

Interpretação:
•**universo:**pessoas
•**predicados:** Q: é funcionário da UFRJ. P: é funcionário público.
$\forall x(Q(x) \rightarrow P(x))$ é verdadeira na interpretação acima.
Exemplo 2:
U={João, José, Pedro}
$Q^I = \{< Joao >, < Jose >\}$
$P^I = \{< Jose >, < Pedro >\}$

$\forall x(Q(x) \rightarrow P(x))$ é falsa nesta interpretação.

Exemplo 3: $\forall x(Q(x) \rightarrow P(x))$
$U = Z$ (inteiros)
$Q^I = \{< 0 >, < 1 >, ...\}$(naturais)
$P^I = \{... < -2 >, < -1 >, < 0 >, < 1 >, < 2 >, ...\}$ (inteiros)

$\forall x(Q(x) \rightarrow P(x))$ é verdadeira nesta interpretação.

Exemplo 4: $\exists x(P(x) \wedge Q(x, c))$
$U = R$ (reais)
$Q^I = x > c$
$P^I = x$ é racional
$c^I = 0$
"Existe algum nÚmero real que também é racional e maior do que zero."
Exemplo 5: $\forall x(P(x) \wedge Q(x) \rightarrow R(x, f(c)))$
$U = Z$ (inteiros)
$c^I = 0$
$f^I = x + 1$
$Q^I = \{< 2 >, < 4 >, < 6 >, ...\}$
$P^I = \{< 1 >, < 2 >, < 3 >, ...\}$
$R^I = x > y$

“Todo número inteiro positivo e par é maior do que 1.” (verdadeiro)

Exemplo 6: $\forall x(P(x) \land Q(x) \rightarrow R(x, f(c)))$
$U = Z$ (inteiros)
$c^I = 4$
$f^I = x + 1$
$Q^I = \{< 2 >, < 4 >, < 6 >, ...\}$
$P^I = \{< 1 >, < 2 >, < 3 >, ...\}$
$R^I = x > y$

“Todo número inteiro positivo e par é maior do que 4.” (falso)

Exemplo 7: $(\forall y C(x, y)) \rightarrow G(x)$
U = {José,João,Pedro,Paulo}
C : x é chefe de y
G : x é gerente

C^I	
João	José
João	Paulo
João	Pedro
João	João
Paulo	João
Paulo	Paulo
Paulo	Pedro
Paulo	José
Paulo	José

G^I
João
Pedro

x = João V
x = José V
x = Pedro V
x = Paulo F

Porém, pela nossa definição da linguagem de LPO, podemos ter variáveis livres ocorrendo nas fórmulas, por exemplo

$$\forall x C(x, y)$$

A variável y ocorre livre nesta fórmula.

Em geral, para sabermos se uma fórmula é verdadeira ou falsa, nós precisamos saber o universo e interpretar cada símbolo não-lógico neste universo e dar valor as variáveis livres.

(1) Interpretar variáveis livres e constantes em elementos do domínio.
(2) Interpretar predicados em relações entre elementos do domínio.
(3) Interpretar funções em funções sobre o domínio.

Definição: Definimos uma **interpretação** como sendo um par ordenado $< D, I >$ onde D é um conjunto não-vazio de indivíduos chamado **domínio**. E V é uma função chamada de **função de interpretação**, definida como:

1. I associa a cada variável livre x um elemento do domínio $d^I \in D$.

   $I(x) = d^I$

2. I associa a cada constante c, um elemento do domínio $c^I \in D$.

   $I(c) = c^I$

3. I associa a cada símbolo funcional n-ário f uma função n-ária $f^I : D^n \to D$ tal que $I(f(t_1,...,t_n)) = f^I(I(t_1),...,I(t_n))$, onde $t_1,...,t_n$ são termos.

4. I associa a cada símbolo predicativo n-ário P uma relação n-ária sobre D.
   $I(P) = P^I$, $P^I \subseteq D^n$, ie, $P^I \subseteq D \times D \times, , , \times D$, n vezes.

Definição:

Seja L uma linguagem de primeira ordem e α e β, fórmulas de L, $t_1, ..., t^n$ termos, P um símbolo predicativo n-ário e $< D, I >$ uma interpretação. Definimos a função de avaliação de fórmulas de L como:

$V_I : W \rightarrow \{V, F\}$, onde W é o conjunto de fórmulas, tal que:

(1) $V_I(P(t_1, ..., t_n)) = V$ se somente se $< I(t_1), ..., I(t_n) > \in P^I$. F caso contrário.

(2) $V_I(\neg\alpha) = V$ se $V_I(\alpha) = F$. F caso contrário.

(3) $V_I(\alpha \wedge \beta) = V$ se $V_I(\alpha) = V$ e $V_I(\beta) = V$. F caso contrário.

(4) $V_I(\alpha \vee \beta) = F$ se $V_I(\alpha) = F$ e $V_I(\beta) = F$. V caso contrário.

(5) $V_I(\alpha \rightarrow \beta) = F$ se $V_I(\alpha) = V$ e $V_I(\beta) = F$. V caso contrário.

(6) $V_I(\forall x\alpha) = V$ se somente se para todo $d \in D$, se $I(x) = d$ então $V_I(\alpha) = V$. F caso contrário.

(7) $V_I(\exists x\alpha) = V$ se para algum $d \in D$, $I(x) = d$ e $V_I(\alpha) = V$. F caso contrário.

Definição:

Seja L uma linguagem de 1a ordem. I uma interpretação para L, Γ um conjunto de fórmulas de L e α uma fórmula.

1. I **satisfaz** $\alpha(\models_I \alpha)$ se somente se $V_I(\alpha) = V$;
2. I **satisfaz** Γ se somente se satisfaz cada membro de Γ ;
3. Γ é **satisfatível** se somente se existe uma interpretação I que satisfaça Γ;
4. α é **válida** ($\models \alpha$) se somente se para toda interpretação I, $\models_I \alpha$, i.e., $V_I(\alpha) = V$ para todo I; (*válida é equivalente a tautologia*)
5. Γ **implica logicamente** em $\alpha(\Gamma \models \alpha$) se somente se para toda interpretação I, se I satisfaz Γ, então I satisfaz α;
6. Γ é **insatisfatível** se somente se Γ não é satisfatível, i.e., não existe uma interpretação I que satisfaz Γ;

7. Uma interpretação I que satisfaz Γ é dita **modelo** para Γ.

   Exercício:

   Dada a seguinte estrutura:

   $D = \{joao, jose, ana, maria\}$

Filhiacao	
jose	joao
maria	jose
joao	ana

Homem
jose
jose

Mulher
ana
maria

Pai	
joao	jose
jose	maria

Interprete a fórmula $\forall x \forall y(F(y,x) \land H(x) \rightarrow P(x,y))$ e verifique formalmente se ela é verdadeira ou falsa.

2.2.3 Axiomatização da LCPO

Axiomas Lógicos:

Implicação:
(1) $\alpha \rightarrow (\beta \rightarrow \alpha)$
(2)$(\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \gamma))$

Conjunção:
(3) $(\alpha \land \beta) \rightarrow \alpha$
(4) $(\alpha \land \beta) \rightarrow \beta$
(5) $\alpha \rightarrow (\beta \rightarrow (\alpha \land \beta))$

Disjunção:
(6) $\alpha \rightarrow \alpha \lor \beta$
(7) $\beta \rightarrow \alpha \lor \beta$
(8) $((\alpha \rightarrow \gamma) \land \beta \rightarrow \gamma)) \rightarrow ((\alpha \lor \beta) \rightarrow \gamma)$

Negação:
(9) $\alpha \rightarrow \neg\neg\alpha$
(10) $\neg\neg\alpha \rightarrow \alpha$
(11) $(\alpha \rightarrow \beta) \rightarrow ((\alpha \rightarrow \neg\beta) \rightarrow \neg\alpha)$

Quantificador

(12) $\forall x\alpha(x) \rightarrow \alpha(x/t)$
(13) $\forall(\alpha \rightarrow \beta) \rightarrow (\forall x\alpha \rightarrow \forall x\beta)$
(14) $\alpha \rightarrow \forall x\alpha$, onde x não ocorre livre em α.

Igualdade

(15) $x \ = \ x$
(16) $x \ = \ y \rightarrow (\alpha \rightarrow \alpha')$, onde α é uma fórmula atômica e α' é obtida de α substituindo-se zero ou mais ocorrências de x (mas não necessariamente todos) por y.

Regra de Inferência:

Modus Pones (M.P.)

$$\frac{\alpha \quad \alpha \rightarrow \beta}{\beta}$$

Abreviatura: $\exists x\alpha \equiv \neg\forall x\neg\alpha$

Definição:
Uma fórmula α é dita um ***teorema*** de um conjunto de fórmulas $\Gamma(\Gamma \vdash \alpha)$ se e somente se existe uma seqüência de fórmulas $\alpha_1, ..., \alpha_n$ tal que $\alpha_n = \alpha$ e cada α_i é:
(i) uma instância de um axioma esquema;
(ii) ou for obtida por M.P. aplicada a α_l e α_k e $l, k < i$.
(iii) ou um membro de Γ.

A seqüência de fórmulas $\alpha_1, ..., \alpha_n$ é chamada de uma prova de α a partir de Γ.

Relação entre Sintaxe e Semântica

TEOREMA DA CORRETUDE:
Se $\Gamma \vdash \alpha$ então $\Gamma \models \alpha$..
TEOREMA DA COMPLETUDE:
Se $\Gamma \models \alpha$ então $\Gamma \vdash \alpha$.

2.2.4 Estruturas e Teorias

Nesta seção gostariamos de apresentar alguns exemplos de estruras relacionais conhecidas e como certas formulas podem ser interpretadas nestas. Achar um conjunto de fórmulas que são verdadeiras exatamente em uma certa clásse de estruturas. Estudaremos os números naturais, grafos, ordens e árvores.

Quando juntamos um conjunto de fórmulas não lógicas a axiomatização da Lógica de Primeira ordem obtemos uma **Teoria**. A partir da teoria podemos deduzir propriedades (teoremas) sobre a estrutura sendo representada pela teoria.

Grafos, Ordens e Árvores

Garfos

Um grafo $G = (V, A)$ é uma par onde V é um conjunto não vazio de vértices e A é uma relação binária sobre V, $A \subseteq V \times V$.

Um linguagem, básica, de primeira ordem para representar grafos deverá ter um símbolo predicativo 2-ário para ser interpretado como A. E o domínio da interpretação deve ser o conjunto de vétices V.

Linguagem: predicado 2-ário R.

Interpretação:

- $D = V$
- $I(R) = A$

Podemos escrever fórmulas que impõem condições sobre o tipo de grafo. Por exemplo, a fómula

$$\forall x R(x, x)$$

é verdadeira, na interpretação a cima se e somente se a relação A for **refleiva**.

Outros exemplos de condições são:

	Condição	Fórmula
Rx.	Reflexividade	$\forall x R(x,x)$
IRx.	Ireflexividade	$\forall x \neg R(x,x)$
Sm.	Simétria	$\forall x \forall y R(x,y) \rightarrow R(y,x)$
Tr.	Transitividade	$\forall x \forall y (\exists z (R(x,z) \wedge R(z,y))) \rightarrow R(x,y)$
Sl.	Serial (Total)	$\forall x \exists y R(x,y)$
Eu.	Euclidiana	$\forall x \forall y \forall z (R(x,z) \wedge R(x,y)) \rightarrow R(z,y)$
ASm.	Anti-Simétrica	$\forall x \forall y R(x,y) \wedge R(y,x) \rightarrow x = y$
Tc.	Tricotomia	$\forall x \forall y (R(x,y) \vee x = y \vee R(y,x)$

Outra clásse de grafos muito usada em computação é clásse dos grafos k-coloríveis. Estes são os grafos que podem ser coloríveis com k cores respeitando as seguintes condições:

1. todo vértice é atribuida uma única cor;
2. vértices vizinhos tem cores distintas.

Estes grafos formam uma estrutura com mais k relações unárias para representar as cores, $G = (V, A, Cor_1, \cdots, Cor_k)$. Para expressar estes grafos precisamos estender nossa linguagem comok símbolos de predicados $C_1, \cdots C_k$ e interpretá-los como

- $I(C_i) = Cor_i$, para todo $1 \leq i \leq k$

exercício: Escreva as fórmulas para expressar as condições 1 e 2 para um grafo ser 3-colorível.

Se juntar algumas destas fórmulas aos axiomas da Lógica de Primeira Ordem obteremos uma teoria dos grafos, por exemplo podemos ter a teoria dos grafos reflexivos e simétricos e etc.

Ordens

Um relação de ordem pode ser vista como um grafo onde o conjunto de aresta A é a prórpria relação de ordem $\leq$ ou $<$ dependendo se a ordem é estrita ou não. Para ter uma ordem algumas condições devem ser impostas:

	Ordem	Fórmulas
Pré	Pré-Ordem	Rx + Tr
Par.	Ordem Parcial	Rx + Tr + ASm
Tot	Odem Total(linear)	Rx + Tr + ASm + Tc
Est.	Estrita	Subst. Rx por IRx em Pré, Par, Tot

Se juntar algumas destas fórmulas aos axiomas da Lógica de Primeira Ordem obteremos uma teoria das ordens, por exemplo podemos ter a teoria dos grafos parciais e etc.

Árvores

Uma árvore é um grafo conexo com um vértice especial chamado raiz tal que deste vértice só existe um único caminho para qualquer outro vértice. Uma árvore pode ser vista como um grafo $G = (V, A, raiz)$. Nós vamos estender a linguagem dos grafos com uma constante r para denotar a $raiz$,

- $I(r) = raiz$

exercício: Escreva as fórmulas para expressar que um grafo é uma árvore. Dica: defina um novo símbolo de predicado, na linguagem, para expressar caminho entre dois vértices, $C(x, y)$ se exsite um caminho de x para y e/ou use a relação de $=$.

Se juntar estas fórmulas aos axiomas da Lógica de Primeira Ordem obteremos uma teoria das árvores.

Teoria dos Números

Outro exemplo de estrutura são os números Naturais e as operações básicas de aritimética. Dada a seguinte estrutura $A_E = \langle \mathbb{N}, 0, \mathrm{S}, <, +, ., \mathrm{E} \rangle$ sobre os Naturais nós podemos escrever as seguintes fórmulas (axiomas) e interpretá-los nesta estrutura.

Axiomas de A_E

S1.	$\forall x S(x) \neq 0$
S2.	$\forall x \forall y (S(x) = S(y) \rightarrow x = y)$
L1.	$\forall x \forall y (x < S(y) \leftrightarrow x \leq y)$
L2.	$\forall x \not< 0$
L3.	$\forall x \forall y (x < y \vee x = y \vee y < x)$
A1.	$\forall x (x + 0) = x$
A2.	$\forall x \forall y (x + S(y)) = S(x + y)$
M1.	$\forall x (x.0) = 0$
M2.	$\forall x \forall y (x.S(y)) = (x.y) + x$
E1.	$\forall x (xE0) = S(0)$
E2.	$\forall x \forall y (xES(y)) = (xEy).x$

Um leitor mais familiarizado notará que os seguintes axiomas foram retirados de A_E:

S3.	$\forall y (y \neq 0 \rightarrow \exists x \; y = S(x))$
Indução.	$(\varphi(0) \wedge \forall x (\varphi(x) \rightarrow \varphi(S(x)))) \rightarrow \forall x \varphi(x)$

Se juntarmos estes axiomas com a nossa axiomatixação da Lógica de Primeira Ordem teremos uma axiomatização para a aritimética dos números naturais, i.e, uma teoria dos números Naturais.

De fato, mesmo sem estes axiomas, nós podemos provar um teorema muito iteressante:

Teorema 2.1. *Uma relação R é recursiva sse R é representável em $Cn(A_E)$.*

Capítulo 3

Lógicas Modais

3.1 Linguagem

3.1.1 Alfabeto modal sobre Φ

Dado um conjunto Φ de símbolos proposicionais, $\Phi = \{p, q, ...\}$, o *alfabeto modal* sobre Φ é constituído por: cada um dos elementos de Φ; o símbolo $\perp$ (absurdo); os conectivos lógicos $\neg$ (negação), $\rightarrow$ (implicação), $\wedge$ (conjunção) e $\vee$ (disjunção); os operadores modais $\Box$ (necessidade) e $\Diamond$ (possibilidade); e os parênteses, como símbolos auxiliares.

3.1.2 Linguagem modal induzida pelo alfabeto modal sobre Φ

A *linguagem modal induzida pelo alfabeto modal sobre* Φ é definida indutivamente da seguinte forma:

$$\varphi ::= p \mid \bot \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \rightarrow \varphi_2 \mid \neg\varphi \mid \Box\varphi \mid \Diamond\varphi$$

3.2 Semântica

3.2.1 *Frames*

Um *frame* é um par $F = (W, R)$ onde W é um conjunto não-vazio de *estados* e R é uma relação binária em W dita *relação de acessibilidade*. Diz-se que $s_2 \in W$ é *acessível* a partir de $s_1 \in W$ se, e somente se, $(s_1, s_2) \in R$.

Figura 3.1: Exemplo de um Frame.

No exemplo da figura 3.1 o conjundo de estados é $W = \{s_1, s_2, s_3, s_4, s_5\}$ e a relação de acessibilidade é $R = \{(s_1, s_2), (s_1, s_3), (s_3, s_3), (s_3, s_4)$ $(s_4, s_1), (s_4, s_5), (s_5, s_5)\}$. O frame é $F = (W, R)$.

3.2.2 Modelos

Um *modelo* sobre o conjunto Φ é um par $M = (F, V)$ onde $F = (W, R)$ é um *frame* e V é uma função de Φ no conjunto das partes de W, que faz corresponder a todo símbolo proposicional $p \in \Phi$ o conjunto de estados nos quais p é satisfeito, i.e., $V : \Phi \longmapsto Pow(W)$.

Figura 3.2: Exemplo de um Modelo.

No exemplo da figura 3.2 o frame é o mesmo da figura 3.1 e a função V é:

- $V(p) = \{s_3, s_4, s_5\}$
- $V(q) = \{s_1, s_5\}$
- $V(r) = \{s_1\}$

3.2.3 Satisfação

Seja $M = (F, V)$ um modelo e $w \in W$ um estado. A notação $M, w \Vdash \varphi$ indica que a fórmula φ é *satisfeita* pelo modelo M no estado w, o que é definido indutivamente como:

- $M, w \Vdash p$ sse $w \in V(p)(\forall p \in \Phi)$
- $M, w \nVdash \bot$
- $M, w \Vdash \neg\varphi$ iff $M, w \nVdash \varphi$,
- $M, w \Vdash \varphi \to \varphi'$ sse $M, w \nvDash \varphi$ **ou** $M, w \Vdash \varphi'$
- $M, w \Vdash \varphi \wedge \varphi'$ sse $M, w \Vdash \varphi$ **e** $M, w \Vdash \varphi'$
- $M, w \Vdash \varphi \vee \varphi'$ sse $M, w \Vdash \varphi$ **ou** $M, w \Vdash \varphi'$
- $M, w \Vdash \Box\varphi$ sse para todo $w' \in W$ **se** wRw' **implica** $M, w' \Vdash \varphi$
- $M, w \Vdash \Diamond\varphi$ sse existe $w' \in W$, wRw' **e** $M, w' \Vdash \varphi$

Nós podemos generalizar a noção de satisfação para conjuntos de fórmulas. Se $\Gamma = \{\phi_1, \cdots, \phi_n\}$ então $M, w \Vdash \Gamma$ sse $M, w \Vdash \phi_i$, para todo $1 \leq i \leq n$.

Exemplo: Seja M o modelo da figura 3.2. Queremos verificar se $M, s_2 \Vdash \Box p$.

$M, s_2 \Vdash \Box p$ sse para todo $w' \in W$ **se** s_2Rw' **implica** $M, w' \Vdash p$, nós precisamos verificar para $w' \in \{s_1, s_2, s_3, s_4, s_5\}$. Como temos uma implicação, para os não vizinhos de s_2 a implicação é vacoamente verdadeira. Então precisamos verificar somente para

- $w' = s_4$, $M, s_4 \Vdash p$ sse $s_4 \in V(p)$ o que é verdade;
- $w' = s_5$, $M, s_5 \Vdash p$ sse $s_5 \in V(p)$ o que é verdade.

A seguir apresentamos um algoritmo para verificar se uma fórmula modal φ é satisfeita num modelo $M = (W, R, V)$[1] num estado w.

função Satisfaz(φ,M, w): **booleano**

caso φ:

p:	**se** $w \in V(p)$ **então retorna verdadeiro**
	senão retorna falso
$\bot$:	**retorna falso**
$\neg\varphi_1$:	**retorna not** Satisfaz(φ_1,M,w)
$\varphi_1 \wedge \varphi_2$:	**retorna** Satisfaz(φ_1,M,w) **and** Satisfaz(φ_2,M,w)
$\varphi_1 \vee \varphi_2$:	**retorna** Satisfaz(φ_1,M,w) **or** Satisfaz(φ_2,M,w)
$\varphi_1 \rightarrow \varphi_2$:	**retorna not** Satisfaz(φ_1,M,w) **or** Satisfaz(φ_2,M,w)
$\Diamond\varphi_1$:	**para todo** w' t. q. wRw' **faça**
	se Satisfaz(φ_1,M,w')**então retorna verdadeiro**
	retorna falso
$\Box\varphi_1$:	**para todo** w' t. q. wRw' **faça**
	se not Satisfaz(φ_1,M,w')**então retorna falso**
	retorna verdadeiro

Complexidade: para cada conectivo booleano são feitas, no pior caso, duas chamadas e para cada ocorrência de símbolo proposicional temos uma chamada. Para os conectivos modais temos que percorrer a lista de adjacências, no pior caso, para todos os estados de W. Logo a complexidade é $O(|\varphi| \times (|W| + |R|))$, isto é, linear no tamanho da fórmula e no tamanho do modelo.

[1] Usaremos no texto $M = (W, R, V)$ quando na verdade deveriamos usar $M = (F, V)$ e $F = (W, R)$.

3.2.4 Tradução Padrão

$\mathcal{LM}$: Linguagem modal com conjunto Φ de símbolos proposicionais, $\Phi = \{p_1, p_2, p_3, ...\}$
$\mathcal{LPO}$: Linguagem de primeira ordem com um predicado binário R e um conjunto de predicados unários $\{P_1, P_2, P_3, ...\}$

Definição 1. Tradução Padrão: *Seja x uma variável de primeira ordem. A tradução padrão $\mathcal{T}$ é uma função que mapeia fórmulas da linguagem modal $\mathcal{LM}$ para linguagem de primeira ordem $\mathcal{LPO}$: $\mathcal{LM} \mapsto^{\mathcal{T}} \mathcal{LPO}$, definida a seguir:*

$$\begin{array}{lcl}
\mathcal{T}_x(\bot) & = & \bot \\
\mathcal{T}_x(p_i) & = & P_i(x), \textit{ para todo } i \in \mathbb{N}^+ \\
\mathcal{T}_x(\neg\varphi) & = & \neg\mathcal{T}_x(\varphi) \\
\mathcal{T}_x(\varphi_1 \wedge \varphi_2) & = & \mathcal{T}_x(\varphi_1) \wedge \mathcal{T}_x(\varphi_2) \\
\mathcal{T}_x(\varphi_1 \vee \varphi_2) & = & \mathcal{T}_x(\varphi_1) \vee \mathcal{T}_x(\varphi_2) \\
\mathcal{T}_x(\varphi_1 \to \varphi_2) & = & \mathcal{T}_x(\varphi_1) \to \mathcal{T}_x(\varphi_2) \\
\mathcal{T}_x(\Diamond\varphi) & = & \exists y(xRy \wedge \mathcal{T}_y(\varphi)) \\
\mathcal{T}_x(\Box\varphi) & = & \forall y(xRy \to \mathcal{T}_y(\varphi))
\end{array}$$

Dado um modelo modal $M = (W, R, V)$ nós podemos ver este modelo modal como um modelo para a liguagem de primeira ordem $\mathcal{LPO}$ interpretando W como o domínio, o predicado R como a relação binária R e cada predicado unário P_i como o conjunto $V(p_i)$.

Teorema 3.1. *Seja φ uma fórmula modal na linguagem $\mathcal{LM}$. Para todo modelo modal M e todo estado w temos*[2]*,*

$$M, w \Vdash \varphi \quad sse \quad M \models \mathcal{T}_x(\varphi)[x/w]$$

[2] Onde $\models$ é a relação de satisfação da Lógica de Primeira Ordem

Prova: *Por indução no comprimento da fórmula φ. A prova pode ser encontrada no apendice A na seção A.1.*

$\triangle$

Exemplo 1. *Obter a tradução padrão de $\Box(p \to \Diamond q)$*

$$\begin{aligned}
\mathcal{T}_x(\Box(p \to \Diamond q)) &= \forall y(xRy \to \mathcal{T}_y((p \to \Diamond q))) \\
&= \forall y(xRy \to (\mathcal{T}_y(p) \to \mathcal{T}_y(\Diamond q))) \\
&= \forall y(xRy \to (\mathcal{T}_y(p) \to \exists z(yRz \wedge \mathcal{T}_z(q)))) \\
&= \forall y(xRy \to (P(y) \to \exists z(yRz \wedge Q(z))))
\end{aligned}$$

Desafio1: No exemplo anterior usamos três variáveis quando na verdade só precisavamos usar duas. Na realidade, qualquer fórmula modal pode ser traduzida para uma de primeira ordem usando-se somente duas variáveis. Por que? Escreva a fórmula do exemplo anterior somente com duas variáveis. Explique como conseguiu.

Desafio 2: Dadas:

$\mathcal{LM}$: Linguagem modal com conjunto Φ de símbolos proposicionais, $\Phi = \{p_1, p_2, p_3, ...\}$

$\mathcal{LPO}$: Linguagem de primeira ordem com um predicado binário R e um conjunto de predicados unários $\{P_1, P_2, P_3, ...\}$ e duas variáveis.

É verdade que toda fórmula em $\mathcal{LPO}$, φ, com uma variável livre x, pode ser traduzida (de volta), por uma função de tradução $\mathcal{T}^- : \mathcal{LPO} \mapsto \mathcal{LM}$, numa fórmula modal em $\mathcal{LM}$ equivalente tal que

$$M, w \Vdash \mathcal{T}^{-}(\varphi) \;\; sse \;\; M \models \varphi(x)[x/w]$$

Exercício 1. *Obtenha a tradução padrão para as seguintes fórmulas modais:*

1. $\Box p \to p$
2. $p \to \Diamond p$, *qual a relação com a fórmula anterior?*
3. $\Box p \to \Box\Box p$
4. $\Diamond\Diamond p \to \Diamond p$, *qual a relação com a fórmula anterior?*
5. $p \to \Box\Diamond p$
6. $\Diamond p \to \Box\Diamond p$
7. $\Box(p \wedge \Diamond(q \to \Diamond\neg p))$

Um resposta parcial para o desafio 2 pode ser dada pela noção de bissimulação.

3.2.5 Bissimulação

Bissimulação é uma ferramenta poderosa para comparar modelos. A intuição deste conceito vem de álgebras de processos [1], onde se deseja estabelecer um relação de equivalência entre processos. Lá dos processos são bissimilares se eles são observacionalmente equivalentes, i.e., um observador externo não consegue distinguir um do outro somente observando o comportamento (ações de comunicação) deles. Esta noçaõ é muito útil, por exemplo, dado uma especificação de um processo $\mathcal{P}_{esp}$ e sua implementação $\mathcal{P}_{impl}$, nós gostariamos que sob o ponto de vista de um usuário externo elas fossem observacionalmente equivalentes, i.e., $\mathcal{P}_{esp} \cong^{eq}_{obs} \mathcal{P}_{impl}$.

Um modelo **enraizado** $M_w = \langle W, R, V \rangle$ com raiz w é um modelo com um elemento distinguido $w \in W$.

Definição 2. *Sejam* $M_w = \langle W_1, R_1, V_1 \rangle$ *e* $N_v = \langle W_2, R_2, V_2 \rangle$ *dois modelos enraizados. Nós dizemos que* M_w *e* N_v *são* **bissimilares**, *notação* $M_w \approx N_v$, *sse*

1. $w \in V_1(p)$ *sse* $v \in V_2(p)$, *para todo símbolo proposicional* $p \in \Phi$;
2. *se* wR_1w', *então existe um* v' *tal que* vR_2v' *e* $M_{w'} \approx N_{v'}$;
3. *se* vR_2v', *então existe um* w' *tal que* wR_1w' *e* $M_{w'} \approx N_{v'}$.

Ilustrando a definição 2.

$$\begin{array}{ccc} w & \approx & v \\ \downarrow{\scriptstyle R_1} & & \downarrow{\scriptstyle R_2} \\ w' & \approx & v' \end{array}$$

Exercício 2. *Mostre que as seguintes modelos enraizados são ou não bissimilares;*

1. $M_w = \langle \{w\}, wR_1w, V_1(p) = \{w\} \rangle$ *e* $N_v = \langle \{v, v'\}, vR_2v', v'R_2v, V_2(p) = \{v, v'\} \rangle$
2. $M_w = \langle \{w\}, wR_1w, V_1(p) = \{w\} \rangle$ *e* $N_v = \langle \{v, v'\}, vR_2v', V_2(p) = \{v, v'\} \rangle$

3. $M_w = \langle \{w, w'\}, wR_1w', V_1(p) = \{w, w'\}\rangle$ *e* $N_v = \langle \{v, v', v''\}, vR_2v', vR_2v'', V_2(p$
$\{v, v', v''\}\rangle$

A seguir vamos enunciar e provar um importante teorema.

Teorema 3.2. *Sejam* $M_w = \langle W_1, R_1, V_1\rangle$ *e* $N_v = \langle W_2, R_2, V_2\rangle$ *dois modelos enraizados tal que* $M_w \approx N_v$*. Então,*

$$M, w \Vdash \varphi \ sse \ N, v \Vdash \varphi$$

,

Para um fórmula modal qualquer φ*.*

> **Prova:** *Por indução no comprimento da fórmula* φ*. A prova pode ser encontrada no apendice A na seção A.2.*
>
> $\triangle$

Este teorema 3.2 junto com o teorema 3.1 nos dão uma resposta parcial ao desafio 2: toda fórmula no fragmento da $\mathcal{LPO}$ usado para a tradução padrão corresponde a uma tradução de uma fórmula modal equivalente? Estes dois teoremas juntos nos axiliarão na resposta desta pergunta. Vejamos o exemplo a seguir.

Exemplo 2. *Dada a fórmula em* $\mathcal{LPO}$ *que expressa a propriedade de reflexividade,* $\forall x R(x, x)$*, ela corresponde a uma fórmula modal* φ_{lm} *equivalente? Isto é, existe uma fórmula modal* φ_{lm} *tal que*

$$M, w \Vdash \varphi_{lm} \quad sse \quad M \models \forall x R(x, x) \ ?$$

No exercício 2 item 1 nós mostramos que os modelos enraizados M_w e N_v são bissimilares e portante pelo teorema 3.2 eles satisfazem as mesma fórmulas modais

$M, w \Vdash \varphi_{lm}$ *sse* $N, v \Vdash \varphi_{lm}$

Porém, nós sabemos que

$M \models \forall x R(x, x)$ *e* $N \not\models \forall x R(x, x)$?

Mas aplicando o teorema da tradução 3.1 padrão aos dois lados do **e** *temos*

$M, w \Vdash \varphi_{lm}$ *e* $N, v \nVdash \varphi_{lm}$

O que é uma contradição, e portanto a fórmula $\forall x R(x, x)$ não corresponde a tradução padrão de nenhuma fórmula modal.

Exercício 3. *Mostre que as seguintes condições em $\mathcal{LPO}$ correspondem ou não a uma fórmula modal usando os teoremas 3.2 e 3.1:*

1. *Simetria:* $\forall x \forall y (R(x, y) \to R(y, x))$
2. *Transitividade:* $\forall x, y, z \in W \ (xRy \land yRz \to xRz)$
3. *Euclidiano:* $\forall x, y, z \in W \ (xRy \land xRz \to yRz)$

Dica: Ache dois modelos bissimilares tal que um satisfaça a condição e outro não, e então aplique o raciocínio do exemplo anterior.

Desafio 3: Se uma certa fórmula em $\mathcal{LPO}$ é sempre verdadeira em modelos bissimilares (invariante por bissimulação) ela corresponde a tradução padrão de uma fórmula modal?

3.2.6 Clásses de Frames

Nesta seção apresentamos algumas clásses de frames que são mais usuais.

Seja um frame $F = (W, R)$ e $\mathcal{F}$ a clásse de todos os frames.

Clásse dos Frames Reflexivos $\mathcal{F}_r$

Composta pelos frames cuja a relação de acessibilidade seja reflexiva.

$$\forall x \in W \ (xRx)$$

$\Longrightarrow$: *Inserir exemplo de frame reflexivo*

Clásse dos Frames Simétricos $\mathcal{F}_s$

Composta pelos frames cuja a relação de acessibilidade seja simétrica.

$$\forall x, y \in W \ (xRy \to yRx)$$

$\Longrightarrow$: *Inserir exemplo de frame simétrico*

Clásse dos Frames Transistivos $\mathcal{F}_t$

Composta pelos frames cuja a relação de acessibilidade seja transitiva.

$$\forall x, y, z \in W \ (xRy \land yRz \to xRz)$$

$\Longrightarrow$: *Inserir exemplo de frame transitivo*

Clásse dos Frames Seriais $\mathcal{F}_{serial}$

Composta pelos frames cuja a relação de acessibilidade seja serial.

$$\forall x \exists y \ (xRy)$$

$\Longrightarrow$: *Inserir exemplo de frame serial*

Clásse dos Frames Euclideanos $\mathcal{F}_{eucl}$

Composta pelos frames cuja a relação de acessibilidade seja Euclideana.

$$\forall x, y, z \in W \ (xRy \land xRz \to yRz)$$

$\Longrightarrow$: *Inserir exemplo de frame euclideano*

3.2.7 Validade

1. φ é **verdadeira em um modelo** M, $M \Vdash \varphi$, sse φ é verdadeira em todos os estados de M;

2. φ é **válida em um frame** F, $F \Vdash \varphi$, sse φ é verdadeira em todos os modelos M baseados em F;

3. φ é **válida numa clásse de frames** $\mathcal{F}$, $\mathcal{F} \Vdash \varphi$, sse φ válida em todos os frames $F \in \mathcal{F}$.

Lema 1. *: $\mathcal{F} \Vdash \Box(\varphi \to \psi) \to \Box\varphi \to \Box\psi$, onde $\mathcal{F}$ é a classe de todos os frames.*

Prova: *Suponha, por contradição, que existe um modelo $\mathcal{M}= (\mathcal{F}, V)$ com um mundo possível $w \in W$ tal que*
$(\mathcal{M}, w) \nVdash \Box(\varphi \to \psi) \to \Box\varphi \to \Box\psi$
Então,
(1) $\mathcal{M}, w \Vdash \Box(\varphi \to \psi)$ *e*
(2) $\mathcal{M}, w \nVdash \Box\varphi \to \Box\psi$
(1) *se e somente se* $\forall w^{'} \in W$*, se* $wR_\alpha w^{'}$ *então* (3) $\mathcal{M}, w^{'} \Vdash (\varphi \to \psi)$.
(2) *se e somente se* (4) $\mathcal{M}, w \Vdash \Box\varphi$ *e* (5) $\mathcal{M}, w \nVdash \Box\psi$.
(4) *se e somente se* $\forall w^{'} \in W$*, se* $wR_\alpha w^{'}$ *então* (6) $\mathcal{M}, w^{'} \Vdash \varphi$.
De (3) *e* (6) *e pela definição de satisfação,* $\forall w^{'} \in W$*, se* $wR_\alpha w^{'}$ *então* $\mathcal{M}, w^{'} \Vdash \psi$*, mas isto é se e somente se* $\mathcal{M}, w \Vdash \Box\psi$*. O que contraria* (5).

$\triangle$

Exercício 4. *Mostre que as seguintes fórmulas são* **válidas** *ou* **não** *na clásse $\mathcal{F}$ de todos os frames.*

1. $\mathcal{F} \Vdash \Box(\phi \wedge \psi) \rightarrow (\Box\phi \wedge \Box\psi)$
2. $\mathcal{F} \Vdash (\Box\phi \wedge \Box\psi) \rightarrow \Box(\phi \wedge \psi)$
3. $\mathcal{F} \Vdash \Diamond(\phi \wedge \psi) \rightarrow (\Diamond\phi \wedge \Diamond\psi)$
4. $\mathcal{F} \Vdash (\Diamond\phi \wedge \Diamond\psi) \rightarrow \Diamond(\phi \wedge \psi)$
5. $\mathcal{F} \Vdash \Box(\phi \vee \psi) \rightarrow (\Box\phi \vee \Box\psi)$
6. $\mathcal{F} \Vdash (\Box\phi \vee \Box\psi) \rightarrow \Box(\phi \vee \psi)$
7. $\mathcal{F} \Vdash \Diamond(\phi \vee \psi) \rightarrow (\Diamond\phi \vee \Diamond\psi)$
8. $\mathcal{F} \Vdash (\Diamond\phi \vee \Diamond\psi) \rightarrow \Diamond(\phi \vee \psi)$
9. $\mathcal{F} \Vdash \Box(\phi \rightarrow \psi) \rightarrow (\Box\phi \rightarrow \Box\psi)$ *Lema 1*
10. $\mathcal{F} \Vdash (\Box\phi \rightarrow \Box\psi) \rightarrow \Box(\phi \rightarrow \psi)$
11. $\mathcal{F} \Vdash \Diamond(\phi \rightarrow \psi) \rightarrow (\Diamond\phi \rightarrow \Diamond\psi)$
12. $\mathcal{F} \Vdash (\Diamond\phi \rightarrow \Diamond\psi) \rightarrow \Diamond(\phi \rightarrow \psi)$
13. $\mathcal{F} \Vdash \Diamond\phi \rightarrow \neg\Box\neg\phi$
14. $\mathcal{F} \Vdash \Box\phi \rightarrow \neg\Diamond\neg\phi$

Validade na Clásse dos Frames Reflexivos $\mathcal{F}_r$

Composta pelos frames cuja a relação de acessibilidade seja reflexiva.

$$\forall x \in W \ (xRx)$$

Lema 2. *:* $\mathcal{F}_r \Vdash \Box\varphi \to \varphi$*, onde* $\mathcal{F}_r$ *é a classe dos os frames reflexivos.*

> **Prova:** *Suponha, por contradição, que existe um modelo* $M = (F, V)$ *com um estado* $w \in W$ *tal que*
> $M, w \nVdash \Box p \to p \Leftrightarrow$
> (1) $M, w \Vdash \Box p$ *e*
> (2) $M, w \nVdash p$
> (1) $\Leftrightarrow$ *para todo* w'*, se* wRw' *então* $M, w' \Vdash p$*. Mas como o frame* F *é reflexivo,* wRw *e protanto* $M, w \Vdash p$*, o que contraria* (2)*.*
>
> $\triangle$

Lema 3. *: Se* $F \Vdash \Box p \to p$ *então* F *é reflexivo.*

> **Prova:** *Vamos provar a contra-positiva. Suponha que* F *não é reflexivo. Precisamos mostrar que* $F \nVdash \Box p \to p$*. Para tanto, vamos construir um modelo* $W = (W, R, V)$ *baseado em* F *tal que* $M, w \nVdash \Box p \to p$*, onde* w *é um estado de* F *que não é reflexivo, i. e.,* $(w, w) \notin R$*. Seja* $V(p) = \{v \in W \mid v \neq w\}$*.*
> $M, w \Vdash \Box p \to p \Leftrightarrow M, w \nVdash \Box p$ (1) *ou* $M, w \Vdash p$ (2)
> (1) $\Leftrightarrow$ *existe* w'*,* wRw' *e* $M, w' \nVdash p \Leftrightarrow w' \notin V(p)$*. Mas como* w *não é reflexivo,* $w' \neq w$ *e pela definição de* $V(p)$*,* $w' \in V(p)$ *o que é uma contradição.*
> (2) $\Leftrightarrow w \in V(p)$*, o que é uma contradiçao com a definição de* $V(p)$*.*
>
> $\triangle$

Teorema 3.3. $F \Vdash \Box p \to p$ *se e somente se* F *é reflexivo.*

Prova: *Direta do lema 2 e lema 3.*

$\triangle$

Corolário 1. *: $\mathcal{F}_r \Vdash \varphi \to \Diamond\varphi$, onde $\mathcal{F}_r$ é a classe dos os frames reflexivos.*

Prova: *Direta do lema 2 e do item 14 do exrecício 4.*

$\triangle$

Validade na Clásse dos Frames Transitivos $\mathcal{F}_t$

Composta pelos frames cuja a relação de acessibilidade seja transitiva.

$$\forall x, y, z \in W\ (xRy \wedge yRz \to xRz)$$

Lema 4. *: $\mathcal{F}_t \Vdash \Box\varphi \to \Box\Box\varphi$, onde $\mathcal{F}_t$ é a classe dos frames transitivos.*

Prova: *COLOCAR A PROVA!!!*

$\triangle$

Corolário 2. *: $\mathcal{F}_t \Vdash \Diamond\Diamond\varphi \to \Diamond\varphi$, onde $\mathcal{F}_t$ é a classe dos frames transitivos.*

Prova: *Direta do lema 4 e do item 14 do exercício 4.*

$\triangle$

Validade na Clásse dos Frames Simétricos $\mathcal{F}_s$

Composta pelos frames cuja a relação de acessibilidade seja simétrica.

$$\forall x, y \in W\ (xRy \to yRx)$$

Lema 5. *: $\mathcal{F}_s \Vdash \varphi \to \Box\Diamond\varphi$, onde $\mathcal{F}_s$ é a classe dos frames simétricos.*

> **Prova:** *COLOCAR A PROVA!!!*
>
> $\triangle$

Corolário 3. *: $\mathcal{F}_s \Vdash \Diamond\Box\varphi \to \varphi$, onde $\mathcal{F}_s$ é a classe dos frames simétricos.*

> **Prova:** *Direta do lema 5 e dos itens 13 e 14 do exercício 4.*
>
> $\triangle$

$\Longrightarrow$: *Inserir outras classes*

Exercício 5. *Prove:*

1. *Prove Corolário 1 sem usar lema 2;*
2. *Prove Corolário 3 sem usar lema 5;*
3. *Prove Corolário 2 sem usar lema 4;*
4. *Euclidiano: $\forall x, y, z \in W\ (xRy \wedge xRz \to yRz)$; E. $\Diamond p \to \Box\Diamond p$*
5. *combinações*
6. *Serial*

3.2.8 Conseguência Lógica

Uma fórmula φ é **conseguência lógica** de um conjunto de fórmulas Γ, $\Gamma \Vdash_{\mathcal{F}} \varphi$, com respeito a uma clásse de frames $\mathcal{F}$, se e somente se para todo modelo $M = (W, R, V)$, baseado em frames em $\mathcal{F}$, e para todo $w \in W$ se $M, w \Vdash \Gamma$ então $M, w \Vdash \varphi$.

Quando a clásse de frames estiver subentendida nós usaremos somente $\Gamma \Vdash \varphi$.

3.3 Sistema Modais Normais

3.3.1 Sistema K

O sistema modal K é o menor sistema modal normal contendo os seguintes axiomas e regras de inferência:

Axiomas

ax.1 todas as tautologias proposicionais

ax.2 $\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$ axioma K

ax.3 $\Diamond\varphi \leftrightarrow \neg\Box\neg\varphi$ axioma *Dual*

Regras de Inferência

Substituição Uniforme:

$$\frac{\vdash \varphi}{\vdash \varphi(p_1/\phi_1, \cdots, p_n/\phi_n)}$$

Onde $p_1, \cdots, p_n$ são todos os símbolos proposicionais ocorrendo em φ

Modus Ponens:

$$\frac{\varphi, \varphi \to \psi}{\psi}$$

Generalização:

$$\frac{\vdash \varphi}{\vdash \Box\varphi}$$

Uma fórmula φ é dita ser um **teorema** de um conjunto de fórmulas Γ, $\Gamma \vdash \varphi$, se e somente se existe uma seqüência $\varphi_0, \varphi_1, ..., \varphi_n$ de fórmulas tal que φ_i é um axioma ou foi obtido aplicando uma regra de inferência para fórmulas de $\{\varphi_0, \varphi_1, ..., \varphi_{i-1}\}$ e φ é último elemento φ_n. Dizemos que um conjunto de fórmulas é **inconsistente** se e somente se $\Gamma \vdash \bot$ caso contrário, Γ é dito ser **consistente**. Uma fórmula φ é consistente se e somente se $\{\varphi\}$ é consistente.

Exemplo: $\Box(p \wedge q), \Box(p \to r) \vdash \Box r$

1. $\Box(p \wedge q)$
2. $\Box(p \to r)$
3. $(p \wedge q) \to p$ ax.1 tautologia
4. $\Box((p \wedge q) \to p)$ Generalização
5. $\Box((p \wedge q) \to p) \to (\Box(p \wedge q) \to \Box p)$ ax.2
6. $(\Box(p \wedge q) \to \Box p)$ MP(4,5)
7. $\Box p$ MP(1,6)
8. $\Box(p \to r) \to (\Box p \to \Box r)$ ax.2
9. $(\Box p \to \Box r)$ MP(2,8)

10. $\Box r$ MP(7,9)

O sistema modal K é correto e completo em relação a classe de todos os frames.

Teorema 3.4 (Correção). *Se* $\Gamma \vdash_K \varphi$ *then* $\Gamma \Vdash \varphi$.

> **Prova:**
>
> $\triangle$

Teorema 3.5 (Completude). *Se* $\Gamma \Vdash \varphi$ *then* $\Gamma \vdash_K \varphi$.

> **Prova:** *A prova deste teorema usa um técnica chamada Modelo Canônico e se encontra no Apendice A.3.*
>
> $\triangle$

3.3.2 Sistema T

O sistema modal T é obtido acrescentando o axioma T ao sistema modal K. Todos os axiomas e regras de inferência de K também pertencem a T.

Axioma T

T. $\Box\varphi \to \varphi$

O sistema modal T é correto e completo em relação a classe dos frames reflexivos $\mathcal{F}_r$.

Teorema 3.6 (Correção). *Se* $\Gamma \vdash_T \varphi$ *then* $\Gamma \Vdash_{\mathcal{F}_r} \varphi$.

> **Prova:** *Adicionar comentários em relação a prova para K.*

$\triangle$

Teorema 3.7 (Completude). *Se* $\Gamma \Vdash_{\mathcal{F}_r} \varphi$ *then* $\Gamma \vdash_T \varphi$.

> **Prova:** *Adicionar comentários em relação a prova para* K.

$\triangle$

3.3.3 Sistema KD

O sistema modal KD é obtido acrescentando o axioma D ao sistema modal K. Todos os axiomas e regras de inferência de K também pertencem a KD.

O sistema KD vem originalmente de Lógica Deôntica onde se estuda conceitos de obrigações e permissões.

Axioma D

$$D.\ \Box\varphi \rightarrow \Diamond\varphi$$

O sistema modal KD é correto e completo em relação a classe dos frames seriais $\mathcal{F}_{serial}$.

Teorema 3.8 (Correção). *Se* $\Gamma \vdash_{KD} \varphi$ *then* $\Gamma \Vdash_{\mathcal{F}_{serial}} \varphi$.

> **Prova:** *Adicionar comentários em relação a prova para* K.

$\triangle$

Teorema 3.9 (Completude). *Se* $\Gamma \Vdash_{\mathcal{F}_{serial}} \varphi$ *then* $\Gamma \vdash_{KD} \varphi$.

> **Prova:** *Adicionar comentários em relação a prova para* K.

$\triangle$

3.3.4 Sistema $S4$

O sistema modal $S4$ é obtido acrescentando o axioma 4 ao sistema modal T. Todos os axiomas e regras de inferência de K e T também pertencem a $S4$.

Axioma 4

4. $\Box\varphi \rightarrow \Box\Box\varphi$

O sistema modal $S4$ é correto e completo em relação a classe dos frames reflexivos e transitivos $\mathcal{F}_{rt}$.

Teorema 3.10 (Correção)**.** *Se* $\Gamma \vdash_{S4} \varphi$ *then* $\Gamma \Vdash_{\mathcal{F}_{rt}} \varphi$.

> **Prova:** *Adicionar comentários em relação a prova para* K*.*
>
> $\triangle$

Teorema 3.11 (Completude)**.** *Se* $\Gamma \Vdash_{\mathcal{F}_{rt}} \varphi$ *then* $\Gamma \vdash_{S4} \varphi$.

> **Prova:** *Adicionar comentários em relação a prova para* K*.*
>
> $\triangle$

3.3.5 Sistema $S5$

O sistema modal $S5$ é obtido acrescentando o axioma 5 ao sistema modal $S4$. Todos os axiomas e regras de inferência de K, T e $S4$ também pertencem a $S5$.

O nome E deste axioma vem de Eulidiano.

Axioma 5

$$E.\ \Diamond\varphi \to \Box\Diamond\varphi$$

O sistema modal $S5$ é correto e completo em relação a classe dos frames reflexivos, transitivos e euclidianos $\mathcal{F}_{rts}$. Na verdade a clásse dos frames reflexivos, transitivos e euclidianos coincide com a clásse dos frames reflexivos, transitivos e simétricos.

Teorema 3.12 (Correção)**.** *Se* $\Gamma \vdash_{S5} \varphi$ *then* $\Gamma \Vdash_{\mathcal{F}_{rts}} \varphi$*.*

> **Prova:** *Adicionar comentários em relação a prova para* K*.*
>
> $\triangle$

Teorema 3.13 (Completude)**.** *Se* $\Gamma \Vdash_{\mathcal{F}_{rts}} \varphi$ *then* $\Gamma \vdash_{S5} \varphi$*.*

> **Prova:** *Adicionar comentários em relação a prova para* K*.*
>
> $\triangle$

3.3.6 Outros Sistemas Modais

$\Longrightarrow$: Fazer tabela com sistemas modais e seus axiomas e condições nos framnes.

$\Longrightarrow$: falar de outros sistemas não normais

Tabela 3.1: Fórmulas Válidas em Clásses de Frames

Nome	Fórmula	Dual	Clásse de Frames
K	$\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$	—	nenhuma
T	$\Box\varphi \to \varphi$	$\varphi \to \Diamond\varphi$	Reflexivos
D	$\Box\varphi \to \Diamond\varphi$	—	Seriais
4	$\Box\varphi \to \Box\Box\varphi$	$\Diamond\Diamond\varphi \to \Diamond\varphi$	Transitivos
B	$\varphi \to \Box\Diamond\varphi$	$\Diamond\Box\varphi \to \varphi$	Simétrico
E	$\Diamond\varphi \to \Box\Diamond\varphi$	$\Diamond\Box\varphi \to \Box\varphi$	Euclidiano

3.3.7 Tableaux para Sistemas Modais

O método de Tableaux para os sistemas modais é uma outra forma de se estabelecer consequência lógica, i.e., $BD \models \varphi$, de uma forma "sintática". O método é idêntico ao da Lógica Proposicional, somente acrescentando regras para tratar dos operadore smodais: $\Box$ e $\Diamond$. O sistema apresentado a seguir é baseado no apresentado no livro [3].

Definição: Um **ramo** θ de um tableaux τ é dito **fechado** se ele contiver α e $\neg\alpha$ para qualquer fórmula α.
Definição: Um **tableaux** τ é dito **fechado** se cada um dos seus ramos for fechado. E aberto caso contrário.

Método

1. O ramo inicial deve conter todas as fórmulas do BD seguidas da negação da pergunta;

2. aplique as regras as fórmulas no mesmo ramo no máximo uma vez;

3. se o tableaux fechar responda SIM;

4. se , em todos os ramos, todas as fórmulas já foram usadas uma vez e mesmo assim o tableaux não fechou responda NÃO.

Tableaux para o Sistema K

Regras para os Operadores da Lógica Proposicional

R$_1$

$$\frac{\alpha \wedge \beta}{\begin{array}{c}\alpha \\ \beta\end{array}}$$

R$_2$

$$\frac{\alpha \vee \beta}{\alpha \quad \beta}$$

R$_3$

$$\frac{\alpha \to \beta}{\neg\alpha \quad \beta}$$

R$_4$

$$\frac{\neg\neg\alpha}{\alpha}$$

R$_5$

$$\frac{\neg(\alpha \wedge \beta)}{\neg\alpha \quad \neg\beta}$$

R$_6$

$$\frac{\neg(\alpha \vee \beta)}{\begin{array}{c}\neg\alpha \\ \neg\beta\end{array}}$$

R$_7$

$$\frac{\neg(\alpha \to \beta)}{\begin{array}{c}\alpha \\ \neg\beta\end{array}}$$

Regras para os Operadores Modais

Regras do Tipo E

R$_\Box$

$$\frac{\Box\alpha}{\rho(T', \alpha)}$$

R$_{\neg\Diamond}$

$$\frac{\neg\Diamond\alpha}{\rho(T', \neg\alpha)}$$

Regras do Tipo F

$R_\Diamond$

$$\frac{\Diamond\alpha}{\rho(T', \alpha)}$$

$R_{\neg\Box}$

$$\frac{\neg\Box\alpha}{\rho(T', \neg\alpha)}$$

A função ρ é definida da seguinte forma:

- **se** regra do tipo E, **então** adicione α a um tableaux existente T' no mesmo ramo;

- **se** regra do tipo F, **então** crie um tablaux novo T' e coloque α como a primeira fórmula;

Teorema (Correção): se existe um tableaux fechado para BD, $\neg\alpha$., então BD $\models \alpha$.

Teorema (Completude): se BD $\models \alpha$ então existe tableaux fechado para BD, $\neg\alpha$.

O método de Tableaux é refutacionalmente completo.

Exemplo: $BD = \{\Box(p \to q),\ \Diamond p\} \vdash \Diamond q$

1.	$\Box(p \to q)$		BD
2.	$\Diamond p$		BD
3.	$\neg\Diamond q$		Neg. Perg.
2.1	p		$F(2)$
2.2	$\neg q$		$E(3)$
2.3	$p \to q$		$E(1)$
2.4	$\underline{\neg p}$	$\underline{q}$	$\mathrm{R}_3(2.3)$

Exercícios:

1. $BD = \{\Box(p \to \Diamond q),\ \Diamond p\} \vdash \Diamond\Diamond q$

2. Faça todos os exercícios da seção 3.2.7.

Desafio: Como você modificaria o Tableaux de **K** para **D** (Serial)?

3.4 Lógicas Multi-Modais

Uma lógica multi-modal é uma generalização da lógica modal estudada nas seções anteriores, na verdade esta é uma lógica mono-modal. Um lógica multi-modal é uma lógica modal com mais de um operador modal e seu respectivo dual. O vocabulário é o da linguagem modal básica estendido com um conjunto, possivelmente infinito, de operadores modais. A linguagem pode ser definida induticamente a partir de um conjunto Φ de símbolos proposicionais, como a seguir:

$$\varphi ::= p \mid \perp \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \rightarrow \varphi_2 \mid \neg\varphi \mid \Box_1\varphi \mid \Box_2\varphi \mid \cdots \mid \Diamond_1\varphi \mid \Diamond_2\varphi \mid \cdots$$

Um *frame* multi-modal é uma túpla $F = (W, R_1, R_2, \cdots)$ onde W é um conjunto não-vazio de *estados* e R_i, para $1 \leq i$, é uma relação binária em W. Diz-se que $s_2 \in W$ é $i-$*acessível* a partir de $s_1 \in W$ se, e somente se, $(s_1, s_2) \in R_i$. A noção de *modelo multi-modal* é analoga, só que baseada em frames multi-modais. A noção de *satisfação* é também analoga, aprentaremos a seguir somente para cada modalidade $\Box_i$ e $\Diamond_i$, para $1 \leq i$,

- $M, w \Vdash \Box_i\varphi$ sse para todo $w' \in W$ **se** wR_iw' **implica** $M, w' \Vdash \varphi$
- $M, w \Vdash \Diamond_i\varphi$ sse existe $w' \in W$, wR_iw' **e** $M, w' \Vdash \varphi$

As noções de *validade* e *conseguência lógica* permanencem as mesmas.

3.4.1 Sistema Multi-Modal K_i

O sistema multi-modal K_i é o menor sistema multi-modal normal contendo os seguintes axiomas e regras de inferência para cada par de operadores multi-modais $\Box_i$ e $\Diamond_i$, para $1 \leq i$:

Axiomas

ax.1 todas as tautologias proposicionais

ax.2 $\Box_i(\varphi \to \psi) \to (\Box_i\varphi \to \Box_i\psi)$ axioma K_i

ax.3 $\Diamond_i\varphi \leftrightarrow \neg\Box_i\neg\varphi$ axioma $Dual_i$

Regras de Inferência

Substituição Uniforme:

$$\frac{\vdash \varphi}{\vdash \varphi(p_1/\phi_1, \cdots, p_n/\phi_n)}$$

Onde $p_1, \cdots, p_n$ são todos os símbolos proposicionais ocorrendo em φ

Modus Ponens:

$$\frac{\varphi, \varphi \to \psi}{\psi}$$

Generalização:

$$\frac{\vdash \varphi}{\vdash \Box_i\varphi}$$

O sistema modal K é correto e completo em relação a classe de todos os frames.

Teorema 3.14 (Correção)**.** *Se* $\Gamma \vdash_{K_i} \varphi$ *then* $\Gamma \Vdash \varphi$*.*

Prova: *Adicionar comentários em relação a prova para* K

$\triangle$

Teorema 3.15 (Completude). *Se* $\Gamma \Vdash \varphi$ *then* $\Gamma \vdash_{K_i} \varphi$.

> **Prova:** *Adicionar comentários em relação a prova para K*
>
> $\triangle$

3.4.2 O sistema KV_{ab}

Esta seção tem como objetivo ilustrar um sistema multi-modal com duas modalidades e seus respectivos duais. Este é um sistema K_2 que chamaremos de KV_{ab} devido as modalidades $[a]$ ($\langle a \rangle$) e $[b]$ ($\langle b \rangle$). A linguage deste sistema pode ser definida indutivamente como

$$\varphi ::= p \mid \bot \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \rightarrow \varphi_2 \mid \neg\varphi \mid [a]\varphi \mid [b]\varphi \mid \langle a \rangle\varphi \mid \langle b \rangle\varphi$$

Um *frame* multi-modal de KV_{ab} é uma tripla $F = (W, R_a, R_b)$ onde W é um conjunto não-vazio de *estados* e $R_a, R_b \subseteq W \times W$ A noção de *modelo multi-modal* é analoga, só que baseada em frames multi-modais. A noção de *satisfação* é também analoga, aprentaremos a seguir somente para cada modalidade,

- $M, w \Vdash [a]\varphi$ sse para todo $w' \in W$ **se** wR_aw' **implica** $M, w' \Vdash \varphi$
- $M, w \Vdash [b]\varphi$ sse para todo $w' \in W$ **se** wR_bw' **implica** $M, w' \Vdash \varphi$
- $M, w \Vdash \langle a \rangle\varphi$ sse existe $w' \in W$, wR_aw' **e** $M, w' \Vdash \varphi$
- $M, w \Vdash \langle b \rangle\varphi$ sse existe $w' \in W$, wR_bw' **e** $M, w' \Vdash \varphi$

As noções de *validade* e *conseguência lógica* permanencem as mesmas.

O sistema multi-modal $K_a b$ é o menor sistema multi-modal normal contendo os seguintes axiomas e regras de inferência para cada par de operadores multi-modais:

Axiomas

ax.1 todas as tautologias proposicionais

ax.2a $[a](\varphi \to \psi) \to ([a]\varphi \to [a]\psi)$ axioma K_a

ax.2b $[b](\varphi \to \psi) \to ([b]\varphi \to [b]\psi)$ axioma K_b

ax.3a $\langle a \rangle \varphi \leftrightarrow \neg[a]\neg\varphi$ axioma $Dual_a$

ax.3b $\langle b \rangle \varphi \leftrightarrow \neg[b]\neg\varphi$ axioma $Dual_b$

Regras de Inferência

Substituição Uniforme:

$$\frac{\vdash \varphi}{\vdash \varphi(p_1/\phi_1, \cdots, p_n/\phi_n)}$$

Onde $p_1, \cdots, p_n$ são todos os símbolos proposicionais ocorrendo em φ

Modus Ponens:

$$\frac{\varphi, \varphi \to \psi}{\psi}$$

Generalização:

$$\frac{\vdash \varphi}{\vdash [a]\varphi} \qquad \frac{\vdash \varphi}{\vdash [b]\varphi}$$

O sistema modal Kab é correto e completo em relação a classe de todos os frames multi-modais com duas relações binárias.

Teorema 3.16 (Correção)**.** *Se* $\Gamma \vdash_{K_{ab}} \varphi$ *then* $\Gamma \Vdash \varphi$*.*

Prova: *Adicionar comentários em relação a prova para K*

$\triangle$

Teorema 3.17 (Completude)**.** *Se* $\Gamma \Vdash \varphi$ *then* $\Gamma \vdash_{K_{ab}} \varphi$*.*

Prova: *Adicionar comentários em relação a prova para K*

$\triangle$

Se acrescentarmos ao siatema K_{ab} o seguinte axioma obtemos um novo sistema onde a relação $R_a \subseteq R_b$.

ax.4 $\langle a \rangle \varphi \to \langle b \rangle \varphi$

Lema 6. *: Se* $F \Vdash \langle a \rangle \varphi \to \langle b \rangle \varphi$ *se e somente se* F *tem a propriedade* $R_a \subseteq R_b$*.*

Prova: *Exercício para casa.*

$\triangle$

Se acrescentarmos ao siatema K_{ab} os seguintes axiomas obtemos um novo sistema onde a relação $R_a = R_b^{-1}$.

ax.5 $\varphi \to [a]\langle b \rangle \varphi$

ax.6 $\varphi \to [b]\langle a \rangle \varphi$

Lema 7. *: Se* $F \Vdash \varphi \to [a]\langle b \rangle \varphi$ *e* $\varphi \to [b]\langle a \rangle \varphi$ *se e somente se* F *tem a propriedade* $R_a = R_b^{-1}$*.*

Prova: *Exercício para casa.*

$\triangle$

3.4.3 Complexidade

Dada uma fórmula modal φ, com comprimento $\mid \varphi \mid$, nós vimos na seção 3.2.3 que a complexidade de se verificar se φ é satisfeita em um modelo $M = (W, R, V)$ é $O(|\varphi| \times (|W| + |R|))$, isto é, linear no tamanho da fórmula e no tamanho do modelo. Esta complexidade não se altera se o frame for reflexivo, simétrico e/ou transitivo. Esta talvez seja uma das razões do sucesso das lógicas modais.

Outro problema bem mais difícil é o de validade. Dado um sistema modal normal decidir se um fórmula φ é válida na clásse de frames correspondente.

Teorema 3.18. *O problema de validade para* **K**, **T** *e* **S4** *é* **PSPACE**-*Completo.*

Teorema 3.19. *O problema de validade para* **S5** *é* **NP**-*Completo*[3].

Em Complexidade Computacional definimos várias clásses de complexidade. Esta são baseadas na quantidade de recuro computacional que necessitam ser consumidos para se resolver o problema. Normalmente, os recurso são medidos pelo tempo e/ou espaço que precisamos para resolver o problema numa Máquin de Turing. As clásses mais conhecidas são:

P : Esta é a clásse dos problemas que podem ser resolvidos em tempo polinomial por uma Máquina de Turing Determinística;

NP : Estes são os problemas que podem ser resolvidos em tempo polinomial por uma máquina de Turing Não-determinística;

[3] O problema de validade para a Lógica Clássica Proposicional é **NP**-Completo

PSPACE : Esta é a clásse dos problemas que podem ser resolvidos usando-se espaço polinomial por uma Máquina de Turing Determinística.

EXPTIME : Esta é a clásse dos problemas que podem ser resolvidos em tempo exponecial por uma Máquina de Turing Determinística.

As comparaçôes entre estas clásses estão entre os grandes problemas em aberto em complexidade computacional. Por exemplo, não se sabe se $\mathbf{P} \neq \mathbf{NP}$. Sabe-se algumas destas relações, por exemplo:

$\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PSPACE} \subseteq \mathbf{EXPTIME}$

Nós dizemos que um dado problema é **NP**-Completo, se todo problema na mesmas clásse pode ser reduzido a ele. Isto quer dizer que consiguir uma "boa" solução para ele é a mesma coisa que conseguir uma "boa" solução para todos na clásse, i.e., ele é tão difícil quanto qualquer

Existem algumas lógicas modais conhecidas cujo problema de validade é **EXPTIME**-Completo, Lógica Dinâmica Proposicional PDL é a mais famosas delas.

Referências Bibliográficas

[1] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[2] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Theoretical Tracts in Computer Science. Cambridge University Press, 2001.

[3] M. M. C. Costa. *Introdução à Lógica Modal Aplicada à Computação.* VIII Escola de Computação. Gramado, 1992.

[4] R. Goldblatt. Logics of Time and Computation. *CSLI* Lecture Notes 7,1992.

[5] D. Harel, D. Kozen D. and Tiuryn. Dynamic Logics. MIT Press, 2000.

[6] Chellas, B. (1980). *Modal Logic, An Introduction.* Cambridge UP, Cambridge, U.K.

[7] Halpern, J. Y., R. Fagin, Y. Moses and M. Y. Vardi (1995). *Reasoning about knowledge.* MIT Press, Massachusets, U.S.A.

[8] Hughes, G. E, Cresswell, M.J. (1996). *A New Introduction to Modal Logic.* Routledge, London and New York.

Apêndice A

Provas

A.1 Prova do Teorema 3.1 Tradução Padrão

Prova: *A prova é por indução no tamanho da fórmula* φ

Base: Seja $\varphi = p$ *um símbolo proposicional, então temos* $M, w \Vdash p$ *sse e somente se* $w \in V(p)$ *e, pela tradução padrão, sabemos que* $\mathcal{T}_x(p) = P(x)$, *e*

$M \models P[x/w]$ *sse* $w \in I(P)$ *sse* $w \in V(p)$

Logo, $M, w \Vdash p$ *sse* $M \models P[x/w]$

Hipótese de Indução (HI) : Suponha que

$$M, w \Vdash \varphi \quad sse \quad M \models \mathcal{T}_x(\varphi)[x/w]$$

valha para toda fórmula φ *de tamanho k.*

Passo indutivo: Agora temos que mostrar que vale se φ *tiver tamanho k+1.*

Vamos mostrar que vale para cada possível forma que φ pode assumir, isto é, se φ tem como conectivo principal: $\neg, \rightarrow, \vee, \wedge, \diamond$ e $\Box$.

Vamos começar com os operadores modais.

Seja ψ uma fórmula de tamanho k.

- $\varphi = \Box\psi$

$M \models \mathcal{T}_x(\Box\psi)[x/w]$ *sse*

$M \models \forall y(R(x,y) \rightarrow \mathcal{T}_y(\psi))[x/w]$ *sse*

para todo $w' \in M$, $R(w,w')$ *implica* $M \models \mathcal{T}_y(\psi)[y/w']$ *sse*

para todo $w' \in M$, $R(w,w')$ *implica* $M, w' \Vdash \psi$, *(H.I.) sse*

$M, w \Vdash \Box\psi$

- $\varphi = \Diamond\psi$

Este caso é análogo ao caso anterior.

- $\varphi = \neg\psi$

$M, w \Vdash \neg\psi$ *sse* **NÂO** $M, w \Vdash \psi$ *sse* *bf NÃO* $M \models \mathcal{T}_x(\psi)[x/w]$, *(H.I.) sse*

$M \models \neg\mathcal{T}_x(\psi)[x/w]$ *sse*

$M \models \mathcal{T}_x(\neg\psi)[x/w]$ *sse*

- $\varphi = \psi \wedge \phi$

  $M, w \Vdash \psi \wedge \phi$ *sse* $M, w \Vdash \psi$ **E** $M, w \Vdash \phi$ *sse*

  $M \models \mathcal{T}_x(\psi)[x/w]$ **E** $M \models \mathcal{T}_x(\phi)[x/w]$, *(H.I.) sse*

  $M \models \mathcal{T}_x(\psi)[x/w] \wedge \mathcal{T}_x(\phi)[x/w]$ *sse*

  $M \models \mathcal{T}_x(\psi \wedge \phi)[x/w]$

- $\varphi = \psi \vee \phi$ *e* $\varphi = \psi \rightarrow \phi$

  Estes casos são análogos ao caso anterior.

$\triangle$

A.2 Prova do Teorema 3.2 Bissimulação

Prova: *Tem uma prova bem simples no livro do Hans na página 25. Só precisa adaptar para o nosso caso.*

A prova é por indução no tamanho da fórmula φ

Base: Seja $\varphi = p$ um símbolo proposicional, então temos $M, w \Vdash p$ sse e somente se $w \in V_1(p)$, mas como $M_w \approx N_v$ pela condição 1. $v \in V_2(p)$ que pela definição de satisfação sse $N, v \Vdash p$.

Hipótese de Indução (HI) : Suponha que

$$M, w \Vdash \varphi \;\; sse \; N, v \Vdash \varphi$$

valha para toda fórmula φ de tamanho k.

Passo indutivo: Agora temos que mostrar que vale se φ tiver tamanho k+1.

Vamos mostrar que vale para cada possível forma que φ pode assumir, isto é, se φ tem como conectivo principal: $\neg, \rightarrow, \vee, \wedge, \diamond$ e $\Box$.

- $\varphi = \neg\psi$

  $M, w \Vdash \neg\psi$ sse $M, w \nVdash \psi$ sse

  Pela H. I. sse $N, v \nVdash \psi$, sse $N, v \Vdash \neg\psi$

- $\varphi = \psi \vee \phi$, $\varphi = \psi \wedge \phi$ e $\varphi = \psi \rightarrow \phi$

  Estes casos são análogos ao caso anterior.

- $\varphi = \Box\psi$

$M, w \Vdash \Box\psi$. Seja v' um mundo arbitrário tal que vR_2v'.

Pela condição 3 da definição de bissimulação, existe um w' tal que wR_1w' e $M_{w'} \approx N_{v'}$

Mas pela definição de satisfação nós temos que $M, w' \Vdash \psi$ e pela H. I. $N, v' \Vdash \psi$, mas como isto vale para todo v' tal que vR_2v', então $N, v \Vdash \Box\psi$.

- $\varphi = \Diamond\psi$

Este caso é análogo ao caso anterior.

$\triangle$

A.3 Prova do Teorema 3.5 Completude para K

Uma Lógica é fortemente completa se, dado um conjunto de fórmulas $\Gamma \cup \varphi$, temos que se φ é consequencia lógica de Γ, então pode-se apresentar uma dedução formal de φ a partir de Γ.

$$\Gamma \models \varphi \Rightarrow \Gamma \vdash \varphi$$

Dada uma lógica modal Λ, provaremos que é fortemente completa com respeito a alguma classe de estruturas mostrando que todo conjunto de fórmulas Λ-consistente pode ser satisfatível em algum modelo. Para isso, vamos construir um modelo no a fórmula é satisfatível - **Modelo Canônico.**

Definição 3. *Um conjunto de fórmulas* Γ *é maximal* Λ*-consistente se é* Λ*-consistente e algum conjunto de fórmulas que contenha* Γ *propriamente é* Λ*-inconsistente. Se* Γ *é um conjunto de fórmulas maximal consistente, então dizemos que ele é* Λ**-CMC**.

Usamos **CMCs** na prova da completude por dois motivos:

1. Note que todo ponto w em todo modelo M para uma lógica Λ é associado a um conjunto de fórmulas: $\{\phi | M, w \Vdash \phi\}$ que é um Λ**-CMC**.

    Prova: *Queremos mostrar que* $\lambda = \{\phi | M, w \Vdash \phi\}$ *é* Λ**-CMC**.

    Suponha que ϕ *é verdade em algum modelo M para a lógica* Λ*. E suponha, por contradição, que* λ *é* Λ*-inconsistente, ou seja,* $\lambda \vdash \phi$ *e* $\lambda \vdash \neg\phi$*.*

    Se ϕ *é verdade em algum modelo M, então temos que* $M, w \Vdash \phi$*. Logo,* $\phi \in \lambda$*. Mas, por hipótese, sabemos que* $\lambda \vdash \neg\phi$*, e pelo teorema de Correção 3.4 temos que* $\lambda \Vdash \neg\phi$*, como por definição todas as fórmulas de* λ *são satisfeitas no estado w do modelo* M*, o que implica que* $M, w \Vdash \neg\phi$*, isto é* $M, w \nVdash \phi$*. Logo,* $\phi \notin \lambda$*. Contradição! Portanto,* λ *é* Λ*-consistente.*

    Para completar a prova de que $\lambda = \{\phi | M, w \Vdash \phi\}$ *é* Λ**-CMC** *precisamos mostrar que algum conjunto que contenha* λ *propriamente é* Λ*-inconsistente.*

Seja K um conjunto consistente tal que $\lambda \subset K$, então existe um $\psi \in K$ e $\psi \notin \lambda$. Pela definição de λ, $M, w \nVdash \psi$, e potanto $M, w \Vdash \neg\psi$ o mque implica que $\neg\psi \in \lambda$ e como $\lambda \subset K$ então $\neg\psi \in K$. E, portanto, $(K \vdash \psi) \wedge (K \vdash \neg\psi)$. Logo, K é Λ-inconsistente.

$\triangle$

2. Se w está relacionado com w' em algum modelo M, então é claro que a informação envolvida no **CMC** associado com w e w' é *c*oerentemente relacionado. Isto é, Modelos dão origem a coleções de **CMC coerentemente relacionados**.

Sendo assim, a ideia da construção do Modelo Canônico é trabalhar atrás de coleções de CMC coerentemente relacionados para os modelos desejados. Construímos, então, um modelo canônico cujos pontos são todos os CMC da lógica do nosso interesse.

Proposição 1. - Propriedades dos CMCs

Se Λ é uma lógica e Γ é um Λ-CMC. Então:

1. *Γ é fechado em relação a Modus Ponens: se ϕ, $\phi \rightarrow \psi$, então $\psi \in \Gamma$;*
2. *$\Lambda \subseteq \Gamma$;*
3. *$\forall\phi, \phi \in \Gamma$ ou $\neg\phi \in \Gamma$;*
4. *$\forall\phi, \psi$, $\phi \vee \psi \in \Gamma$ se e, somente se, $\phi \in \Gamma$ ou $\psi \in \Gamma$.*

No lema a seguir, vamos provar que algum conjunto consistente de fórmulas pode ser extendido até um CMC.

Lema 8. - Lindenbaum's Lemma

Se Σ é um conjunto de fórmulas Λ-consistente, então existe um Λ-CMC Σ^+ tal que $\Sigma \subseteq \Sigma^+$.

Prova: *Seja $\phi_0, \phi_1, \phi_2, \ldots$ uma enumeração de fórmulas da nossa linguagem. Definimos o conjunto Σ^+ como a união de cadeias de conjuntos Λ-consistente, como a seguir:*

$$\Sigma_0 = \Sigma$$

$$\Sigma_{n+1} = \begin{cases} \Sigma_n \cup \{\phi_n\}, & se\ for\ \Lambda - consistente \\ \Sigma_n \cup \{\neg\phi_n\}, & caso\ contrario \end{cases}$$

$$\Sigma^+ = \cup_{n \geq 0} \Sigma_n$$

Propriedades de Σ^+ :

1. *Σ_k é Λ-consistente $\forall k$.*

Prova: Vamos provar por indução em k.

Base: $\Sigma_0 = \Sigma$ que é Λ-consistente por hipótese.

Hipótese de Indução: Suponha que Σ_k seja Λ-consistente.

Agora queremos mostrar que Σ_{k+1} também é Λ-consistente.

Por construção, temos $\Sigma_{k+1} = \begin{cases} \Sigma_k \cup \{\phi_k\}, & se\ for\ \Lambda - consiste \\ \Sigma_k \cup \{\neg\phi_k\}, & caso\ contrario \end{cases}$

Usando a sguinte tautologia temos:

$$\Sigma_k \Leftrightarrow (\Sigma_k \wedge \phi_k) \vee (\Sigma_k \wedge \neg\phi_k)$$

que, $\Sigma_k \Leftrightarrow \Sigma_{k+1}$. E como, por hipótese, Σ_k é Λ-consistente, podemos concluir que Σ_n é Λ-consistente, $\forall n$.

$\triangle$

$\Longrightarrow$: PAREI AQUI em 10/10/11!!!

2. *Exatamente um de ϕ e $\neg\phi$ está em Σ_{k+1}, pra toda fórmula ϕ;*
3. *Se $\Sigma_{k+1} \vdash_\Lambda \phi$, então $\phi \in \Sigma_{k+1}$;*
4. *Σ_{k+1} é um Λ-CMC.*

$\triangle$

Definição 4. *O modelo canônico $\mathfrak{M}^\Lambda$ para uma lógica modal Λ é a tripla $(W^\Lambda, R^\Lambda, V^\Lambda)$, onde:*

1. *W^Λ é o conjunto de todos os Λ-CMC;*
2. *R^Λ é a relação binária em W^Λ definida por $wR^\Lambda u$ se, para toda fórmula $\psi, \psi \in u \Rightarrow \Diamond\psi \in w$. R^Λ é chamado de relação canônica.*

3. V^Λ *é a valoração definida por* $V^\Lambda(p) = \{w \in w^\Lambda | p \in w\}$. V^Λ *é chamado de valoração canônica.*

$\mathfrak{F} = (W^\Lambda, R^\Lambda)$ *é chamado frame canônimo.*

Lema 9. *Para alguma fórmula normal* Λ, $wR^\Lambda u$ *se e, somente se,* $\forall\psi, \Box\psi \in w \Rightarrow \psi \in u$.

Prova: *Suponha* $wR^\Lambda u$ *e* $\psi \notin u$. *Como* u *é um CMC, (pela* **proposição 1** *)* $\neg\psi \in u$. *Como* $wR^\Lambda u$*,(pela definição anterior) temos* $\Diamond\neg\psi \in w$. *Como* w *é consistente,*$\neg\Diamond\neg\psi \notin w$. *Então,* $\Box\psi \notin w$. *Provamos, então, por contrapositiva.*

$$\neg(\psi \in u) \to \neg(\Box\psi \in w)$$

Reciprocamente, $\forall\psi$, $\Box\psi \in w \Rightarrow \psi \in u$. *Queremos mostrar que* $wR^\Lambda u$.

$\Box\psi \in w \equiv \forall x \in w$, *se* $wR^\Lambda x$ *então* $M, x \models \psi$. *Ou seja, para todo vértice x adjacente ao w, tem-se* $\psi \in x$. *Como* $\psi \in u$, *concluímos que u é uma vértice adjacente de w. E, portanto,* $wR^\Lambda u$.

$\triangle$

Lema 10. - Lema da Existência *Para alguma lógica modal normal* Λ *e algum estado* $w \in W^\Lambda$, *se* $\Diamond\phi \in w$, *então existe um estado* $v \in W^\Lambda$ *tal que* $wR^\Lambda v$ *e* $\phi \notin v$.

Prova: *Suponha que* $\Diamond\phi \in w$. *Então para toda fórmula* $\Box\varphi_1, \Box\varphi_2, ...\Box\varphi_n \in w$ *queremos mostrar que* $\varphi_1 \wedge \varphi_2 \wedge ... \wedge \varphi_n \wedge \phi$ *é consistente.*

Vamos supor que $\varphi_1 \wedge \varphi_2 \wedge ... \wedge \varphi_n \wedge \phi$ *é inconsistente,* *então* $\neg(\varphi_1 \wedge \varphi_2 \wedge ... \wedge \varphi_n \wedge \phi)$ *é teorema.*

$\vdash (\varphi_1 \wedge \varphi_2 \wedge ... \wedge \varphi_n \wedge) \Rightarrow \neg\phi$
$\vdash \Box((\varphi_1 \wedge \varphi_2 \wedge ... \wedge \varphi_n \wedge) \Rightarrow \neg\phi)$
$\vdash \Box(\varphi_1 \wedge \varphi_2 \wedge ... \wedge \varphi_n \wedge) \Rightarrow \Box\neg\phi$
$\vdash \Box\varphi_1 \wedge \Box\varphi_2 \wedge ... \wedge \Box\varphi_n \wedge \Rightarrow \Box\neg\phi$

Pela hipótese de que toda fórmula do tipo $\Box\varphi_i \in w$ *temos então que* $\Box\neg\phi \in w$

$\Box\neg\phi \in w \equiv \neg\Diamond\phi \in w$ *Contradição com a hipótese inicial.*

Logo, $\phi \in w$.

$\triangle$

Lema 11. - THUTH LEMMA *Para alguma lógica modal normal* Λ *e alguma fórmula* ϕ, $M^{\Lambda}, w \models \phi \Leftrightarrow \phi \in w$.

Prova: *Indução no grau de* ϕ. *O caso base segue da definição de* V^{Λ}. *Casos booleanos seguem da proposição 1. Então faremos para os operadores modais.*

Hipótese de Indução: *Se* ϕ *tem tamanho n, então* $\mathfrak{M}^{\Lambda}, w \models \phi \Leftrightarrow \phi \in w$.

$$\begin{aligned} \mathfrak{M}, w \models \Diamond\phi &\Leftrightarrow \exists v(wR^{\Lambda}v \wedge \mathfrak{M}, v \models \phi) \\ &\Leftrightarrow_{pelaHI} \exists v(wR^{\Lambda}v \wedge \phi \in v) \Rightarrow \\ &\Rightarrow_{R^{\Lambda}} \Diamond\phi \in w \end{aligned}$$

Reciprocamente, suponha $\Diamond\phi \in w$, *pelo Lema da Existência, existe um estado v em* W^{Λ} *tal que* $(wR^{\Lambda}v \wedge \phi \in v) \Leftrightarrow \exists v(wR^{\Lambda}v \wedge \mathfrak{M}, v \models \phi) \Leftrightarrow \mathfrak{M}, w \models \Diamond\phi$

O resultado é análogo para o outro operador modal. Logo, o lema está provado.

$\triangle$

Teorema A.1. - Teorema do Modelo Canônico *Alguma lógica modal é fortemente completa com respeito a um modelo canônico.*

Prova: *Suponha que* Σ *é um conjunto consistente de uma lógica modeal normal. Pelo* **Lindebaum's Lemma**, *existe um conjunto maximal consistente (*Λ*-CMC)* Σ^+ *, extensão de* Σ*. Então* $\Sigma \in \Sigma^+$ *e, pelo* **Truth Lemma**, $\mathfrak{M}, \Sigma^+ \models \Sigma$.

$\triangle$