

LÓGICA MATEMÁTICA

AUTORES

Cristiano Bertolini

Guilherme Bernardino da Cunha

Patricia Rodrigues Fortes



LICENCIATURA EM COMPUTAÇÃO

LÓGICA MATEMÁTICA

AUTORES

Cristiano Bertolini

Guilherme Bernardino da Cunha

Patricia Rodrigues Fortes

1ª Edição

UAB/NTE/UFSM

UNIVERSIDADE FEDERAL DE SANTA MARIA

Santa Maria | RS

2017

©Núcleo de Tecnologia Educacional – NTE.
Este caderno foi elaborado pelo Núcleo de Tecnologia Educacional da
Universidade Federal de Santa Maria para os cursos da UAB.

PRESIDENTE DA REPÚBLICA FEDERATIVA DO BRASIL

Michel Temer

MINISTRO DA EDUCAÇÃO

Mendonça Filho

PRESIDENTE DA CAPES

Abilio A. Baeta Neves

UNIVERSIDADE FEDERAL DE SANTA MARIA

REITOR

Paulo Afonso Burmann

VICE-REITOR

Paulo Bayard Dias Gonçalves

PRÓ-REITOR DE PLANEJAMENTO

Frank Leonardo Casado

PRÓ-REITOR DE GRADUAÇÃO

Martha Bohrer Adaime

COORDENADOR DE PLANEJAMENTO ACADÊMICO E DE EDUCAÇÃO A DISTÂNCIA

Jerônimo Siqueira Tybusch

COORDENADOR DO CURSO DE LICENCIATURA EM COMPUTAÇÃO

Sidnei Renato Silveira

NÚCLEO DE TECNOLOGIA EDUCACIONAL

DIRETOR DO NTE

Paulo Roberto Colusso

COORDENADOR UAB

Reisoli Bender Filho

COORDENADOR ADJUNTO UAB

Paulo Roberto Colusso

NÚCLEO DE TECNOLOGIA EDUCACIONAL

DIRETOR DO NTE

Paulo Roberto Colusso

ELABORAÇÃO DO CONTEÚDO

Cristiano Bertolini, Guilherme Bernardino da Cunha
e Patrícia Rodrigues Fortes

REVISÃO LINGUÍSTICA

Camila Marchesan Cargnelutti

APOIO PEDAGÓGICO

Magda Schmidt
Siméia Tussi Jacques

EQUIPE DE DESIGN

Carlo Pozzobon de Moraes
Mariana Panta Millani
Matheus Tanuri Pascotini

PROJETO GRÁFICO

Ana Letícia Oliveira do Amaral



B546l Bertolini, Cristiano

Lógica matemática [recurso eletrônico] / Cristiano Bertolini,
Guilherme Bernardino da Cunha, Patrícia Rodrigues Fortes. – Santa
Maria, RS : UFSM, NTE, 2017.

1 e-book

ISBN: 978-85-8341-184-0

Este caderno foi elaborado pelo Núcleo de Tecnologia Educacional
da Universidade Federal de Santa Maria para os cursos da UAB
Acima do título: Licenciatura em computação

1. Matemática 2. Lógica matemática I. Universidade Aberta do
Brasil II. Título.

CDU 510.6

Ficha catalográfica elaborada por Alenir Goularte - CRB-10/990
Biblioteca Central da UFSM



Ministério da
Educação



PROGRAD



APRESENTAÇÃO

A disciplina de *Lógica Matemática* é a base do pensamento computacional, ou seja, a forma como os computadores realizam tarefas. Apesar de ter origem na filosofia há muitos anos, com o desenvolvimento da computação, a Lógica tem evoluído muito. Nesta disciplina, abordaremos a lógica proposicional e de predicados que são fundamentais para o entendimento de diversas outras matérias na área de Ciência da Computação. Também versaremos sobre lógicas não clássicas, que serão vistas novamente em disciplinas mais avançadas do curso, como *Inteligência Artificial*.

O professor Cristiano Bertolini possui graduação em Ciência da Computação pela Universidade de Passo Fundo (2001), mestrado em Ciência da Computação pela Pontifícia Universidade Católica do Rio Grande do Sul (2005) e doutorado em Ciências da Computação pela Universidade Federal de Pernambuco (2010). Tem experiência na área de Engenharia de Software, com ênfase em Teste de Software, Métodos Formais e Engenharia de Software Experimental. Atualmente é professor adjunto da UFSM no campus de Frederico Westphalen.

O professor Guilherme Bernardino da Cunha possui graduação em Ciência da Computação, mestrado em Ciências (2003) com Ênfase em Inteligência Artificial e Processamento Digital de Imagens e doutorado em Ciências, com ênfase em Engenharia Biomédica, pela Universidade Federal de Uberlândia. Atualmente é professor Adjunto da Universidade Federal de Santa Maria – campus Frederico Westphalen. Tem experiência na área de Ciência da Computação, atuando principalmente nos seguintes temas: Engenharia de Software, Inteligência Artificial, Análise de Séries Temporais, Epidemiologia, Banco de Dados, Redes Neurais Artificiais, Algoritmos Genéticos e Bioinformática.

Patricia Rodrigues Fortes é licenciada em Matemática pela Universidade Federal de Santa Maria (1997), possui mestrado em Matemática Aplicada (1998) e doutorado em Engenharia Mecânica pela Universidade Federal do Rio Grande do Sul (2003). Trabalhou por 12 anos na URI – Campus de Frederico Westphalen, tendo sido coordenadora do Curso de Licenciatura em Matemática e coordenadora do Comitê de Pesquisa. Em março de 2011, tornou-se professora da Universidade Federal de Santa Maria – UFSM/campus de Frederico Westphalen, onde trabalha atualmente com disciplinas da área de Matemática nos cursos de graduação em Engenharia Ambiental e Sanitária, Engenharia Florestal, Agronomia e Sistemas de Informação. Suas pesquisas relacionam-se à Matemática Aplicada (Fenômenos de Transporte de Partículas) e ao Ensino de Matemática. Trabalha com extensão universitária, associando conhecimentos matemáticos às ações de projetos da área ambiental, executados na região de abrangência da UFSM/Campus de Frederico Westphalen. Por dez anos trabalhou como Professora Orientadora do Programa de Iniciação Científica da OBMEP (Olimpíada Brasileira de Matemática das Escolas Públicas). Desde 2007, tem sido membro do Banco de Avaliadores INEP/MEC (Instituto Nacional de Estudos e Pesquisas Educacionais / Ministério da Educação). É membro

da SBMAC (Sociedade Brasileira de Matemática Aplicada e Computacional) desde 2000, foi vice-coordenadora da Região 13 – Rio Grande do Sul em 2012, e é a atual Secretária Geral da Diretoria Gestão 2016-2017.

ENTENDA OS ÍCONES



ATENÇÃO: faz uma chamada ao leitor sobre um assunto, abordado no texto, que merece destaque pela relevância.



INTERATIVIDADE: aponta recursos disponíveis na internet (sites, vídeos, jogos, artigos, objetos de aprendizagem) que auxiliam na compreensão do conteúdo da disciplina.



SAIBA MAIS: traz sugestões de conhecimentos relacionados ao tema abordado, facilitando a aprendizagem do aluno.



TERMO DO GLOSSÁRIO: indica definição mais detalhada de um termo, palavra ou expressão utilizada no texto.

SUMÁRIO

▷ APRESENTAÇÃO ·5

▷ UNIDADE 1 – LÓGICA CLÁSSICA E PROPOSICIONAL ·10

Introdução ·12

1.1 Proposições ·13

1.2 Operações lógicas sobre proposições ·15

1.2.1 Negação ·15

1.2.2 Conjunção ·16

1.2.3 Disjunção ·17

1.2.4 Disjunção Exclusiva ·17

1.2.5 Condicional ·18

1.2.6 Bicondicional ·19

Atividades ·21

▷ UNIDADE 2 – TABELA VERDADE ·22

Introdução ·24

2.1 Conceitos básicos ·25

2.2 Implicações e equivalências ·28

2.3 Tautologias ·33

Atividades ·34

▷ UNIDADE 3 – MÉTODO DEDUTIVO ·36

Introdução ·38

3.1 Definições ·39

3.1.1 Exemplificando ·39

3.1.2 Forma Normal Conjuntiva ·44

3.1.3 Forma Normal Disjuntiva ·46

Atividades ·48

▷ UNIDADE 4 – LÓGICA DE PREDICADOS ·49

Introdução ·51

4.1 Lógica proposicional versus Lógica de primeira ordem ·52

4.2 Sintaxe ·53

4.2.1 Definições ·54

4.2 Semântica ·60

Atividades ·66

▷ **UNIDADE 5 – LÓGICAS NÃO CLÁSSICAS ·67**

Introdução ·69

5.1 Lógica Fuzzy ·70

5.2 Lógica Modal ·72

5.3 Lógica Paraconsistente ·73

Atividades ·75

▷ **UNIDADE 6 – LÓGICA TEMPORAL ·76**

Introdução ·78

6.1 Elementos básicos ·79

6.2 Semântica de Kripke ·80

6.3 Conceitos básicos da lógica temporal ·81

6.4 Aplicações em lógica temporal ·82

Atividades ·83

▷ **CONSIDERAÇÕES FINAIS ·84**

▷ **REFERÊNCIAS ·85**

1

LÓGICA CLÁSICA
E PROPOSICIONAL

INTRODUÇÃO

A lógica matemática é de fundamental importância para as linguagens de programação necessárias para a construção de programas de computador (*softwares*). É com base na lógica matemática que as linguagens de computador são descritas. Em lógica, uma linguagem de computador é dita como linguagem formal, pois o formalismo é dado pela representação matemática.

A linguagem natural é um meio de comunicação utilizado no cotidiano das pessoas, por exemplo, Português, Inglês, Espanhol. Uma das características dessas linguagens é a ambiguidade, ou seja, uma sentença pode ser interpretada de diferentes formas. Em um sistema computacional não podemos ter ambiguidades; portanto, precisamos de mecanismos que permitam expressar os sistemas computacionais de forma não ambígua. A lógica é o fundamento mais básico desses sistemas e tem sido amplamente estudada.

Tanto as linguagens naturais quanto as formais possuem sintaxe (como se escreve) e semântica (significado). No entanto, apenas linguagens formais são livres de ambiguidade. Para que isso seja possível, estudaremos, neste capítulo, os fundamentos da lógica matemática.

Os fundamentos que veremos a seguir serão utilizados nas demais disciplinas no curso, principalmente naquelas que abordam linguagens de programação. Dessa forma, é de extrema importância o estudo da lógica clássica e, principalmente, da lógica proposicional. A lógica proposicional deverá ser compreendida e posteriormente será constantemente revista nas outras disciplinas. Entender a lógica proposicional capacitará o aluno a resolver problemas computacionais.

1.1

PROPOSIÇÕES

No estudo da lógica matemática, o conceito mais básico é o de Proposição, que vem do verbo propor, ou seja, significa submeter à apreciação. Uma proposição também é vista como uma sentença e pode ser descrita em uma linguagem (formal ou não).

Uma proposição é uma declaração afirmativa à qual se pode associar um valor verdadeiro ou falso, mas não ambos. Por exemplo, “O Brasil fica na América” é uma proposição verdadeira (V), enquanto “A lua é de queijo” é uma proposição falsa (F). A proposição é o elemento básico a partir do qual os argumentos são construídos, sendo também o principal objeto de estudo na lógica proposicional.

Sentenças não declarativas não podem ser consideradas proposições pois não possuem apenas um valor associado (verdadeiro ou falso). Por exemplo:

» Sentenças Interrogativas: Onde você mora? Qual é o carro preto?

» Sentenças Imperativas: Maria, compareça à recepção, por favor. Faça todos os exercícios solicitados agora.

» Sentenças Exclamativas: Todo mundo está sujeito a cometer erros! Nossa, que foto bonita!

Cabe ressaltar que algumas sentenças declarativas também não podem ser analisadas, principalmente quando são descritas por linguagens não formais ou quando apresentam alguma ambiguidade e não é possível atribuir um valor lógico (verdadeiro ou falso). Por exemplo: Eu encontrei Maria com uma camisa amarela. Nesse caso, a sentença é ambígua, pois pode significar que “eu estava com uma camisa amarela” ou que “Maria estava com uma camisa amarela”.

Desta forma, temos que proposições são sentenças onde é possível atribuir apenas um valor lógico: verdadeiro ou falso. Usualmente as proposições são representadas por letras minúsculas (por exemplo: p, q, r, s, t), e no decorrer deste livro usaremos letras para representarmos as proposições. Alguns exemplos de proposições válidas:

p: Carlos é professor

q: $1 > 5$

Se afirmarmos que a proposição p é verdadeira, ou seja, Carlos é professor, então podemos dizer que o valor lógico da proposição p é verdadeiro – ou pela equação:

$$VL(p)=V$$

No caso da proposição q, é falsa pois 1 não é maior que 5, então temos que:

$$VL(q)=F$$

Podemos observar que as proposições p e q assumem sempre um valor lógico e isso é válido para qualquer proposição lógica. Não existe a possibilidade de uma

proposição assumir mais de um valor lógico. Isso se aplica porque as proposições seguem algumas propriedades que devem ser sempre obedecidas:

» **Princípio da identidade:** tudo é idêntico a si mesmo. Por exemplo, a proposição p é igual à p ($p = p$), mesmo se existir $p = q$.

» **Princípio da não-contradição:** uma proposição não pode ser verdadeira e falsa ao mesmo tempo. Por exemplo, dada uma proposição p ela é ou verdadeira ou falsa e nunca assume os dois estados ao mesmo tempo.

» **Princípio do terceiro excluído:** toda proposição ou é verdadeira ou é falsa, isto é, verifica-se sempre um destes casos e nunca um terceiro. Ou seja, neste sistema de raciocínio tem-se estabelecido somente dois “estados de verdade”, isto é, a “verdade” e a “não verdade” (“falsidade”).

Desta forma, a Lógica Matemática é um sistema bivalente ou dicotômico, no qual os dois estados de verdade servem para caracterizar todas as situações possíveis, sendo mutuamente excludentes, isto é, a ocorrência de um estado exclui a existência do outro (ou é verdade ou é não verdade).

Veremos a seguir como analisamos proposições para determinar se são verdadeiras ou falsas. Nesse contexto, podemos ter proposições bem complexas e usaremos o conceito de Tabela verdade, que consiste em uma tabela matemática utilizada para determinar o estado (verdadeiro ou falso) das proposições.

1.2

OPERAÇÕES LÓGICAS SOBRE PROPOSIÇÕES

Proposições podem ser consideradas simples ou compostas. Serão proposições simples aquelas que vêm sozinhas, desacompanhadas de outras proposições. Por exemplo, duas proposições simples:

p: O carro é azul
q: João é motorista

Caso combinarmos as duas proposições, temos uma proposição composta, ou seja, duas (ou mais) proposições vêm conectadas entre si, formando uma só sentença. Por exemplo, podemos conectar as proposições p e q com conectores lógicos:

r: O carro é azul e João é motorista

Desta forma, quando pensamos, muitas vezes efetuamos cálculos (ou operações) sobre proposições. Esse cálculo é denominado cálculo proposicional e se assemelha à aritmética dos números. Essas operações também resultam em um valor verdadeiro ou negativo. A seguir são apresentadas as operações sobre proposições:

1.2.1 Negação

Definição: Dada uma proposição p, denomina-se a negação de p a proposição representada por “não p”, no qual o valor lógico é verdade quando p é falso e falso quando o valor de p é verdadeiro.

Desta forma, “não p” tem o valor lógico oposto daquele de p. Simbolicamente, podemos expressar a negação de um valor p por $\sim p$, que se lê “não p”. O valor lógico da negação de uma proposição é, portanto, definido pela Tabela 1, denominada tabela verdade.

p	$\sim p$
V	F
F	V

Assim, podemos concluir que:

$\sim V = F$
 $\sim F = V$

Ou seja, não verdade é igual a falso e não falso é igual a verdade. Alguns exemplos do operador de negação:

Se $p = 1+1 = 2$ então p é verdade, ou seja, $p=V$
Se $\sim p = 1+1=2$ então p é falso, ou seja, $p=F$
 $p = \text{Carlos é casado}$
 $\sim p = \text{Carlos não é casado}$

1.2.2 Conjunção

Definição: Dada duas proposições p e q , define-se por conjunção o operador “e” (simbolicamente representado por \wedge), onde “ $p \wedge q$ ” possui o valor lógico Verdade se ambas as proposições (p e q) são verdade. Da mesma forma, pode possuir o valor lógico Falso se ambas as proposições (p e q) são falsas.

Simbolicamente, a representação da conjunção entre duas proposições é representada por “ $p \wedge q$ ”, onde se lê “ p e q ”. O valor lógico da conjunção de duas proposições é, portanto, definido pela tabela verdade a seguir:

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

Assim, podemos concluir que:

$V \wedge V = V$
 $F \wedge F = F$
 $V \wedge F = F$
 $F \wedge V = F$

Em termos gerais temos ainda que:

$$V(p \wedge q) = V(p) \wedge V(q)$$

Alguns exemplos do operador de conjunção:

$p = \text{O céu é azul}$
 $q = 5+2=7$
 $p \wedge q = V$
 $p = \text{Rio de Janeiro é a capital do Brasil}$
 $q = 1+1 = 2$
 $p \wedge q = F$

1.2.3 Disjunção

Definição: Dada duas proposições p e q , define-se por disjunção o operador “ou” (simbolicamente representado por \vee), onde “ $p \vee q$ ” possui o valor lógico Verdade se pelo menos uma das proposições ($p \vee q$) forem verdade. Da mesma forma, pode possuir o valor lógico Falso se, e apenas se, ambas as proposições ($p \vee q$) forem falsas.

Simbolicamente, a representação da disjunção entre duas proposições é representada por “ $p \vee q$ ” onde se lê “ p ou q ”. O valor lógico da disjunção de duas proposições é, portanto, definido pela tabela verdade a seguir:

p	q	$p \vee q$
V	V	V
V	F	V
F	V	V
F	F	F

Assim, podemos concluir que:

$$V \vee V = V$$

$$F \vee F = F$$

$$V \vee F = V$$

$$F \vee V = V$$

Em termos gerais temos ainda que:

$$V(p \vee q) = V(p) \vee V(q)$$

Alguns exemplos do operador de disjunção:

$$p = \text{A bola é quadrada}$$

$$q = 1+1 = 3$$

$$p \vee q = F$$

$$p = \text{Rio de Janeiro é a capital do Brasil}$$

$$q = 1+1 = 2$$

$$p \vee q = V$$

1.2.4 Disjunção Exclusiva

A disjunção exclusiva é um caso específico da disjunção. Suponhamos as duas proposições abaixo:

$$p = \text{Carlos é casado ou gaúcho}$$

$$q = \text{Maria é carioca ou gaúcha}$$

Considerando a proposição p , podemos ter que “Carlos é casado” é verdade e “Carlos é gaúcho” também é verdade. Ou seja, ambas podem ser verdades. No entanto, na

proposição q, se maria for carioca ela não poderá ser também gaúcha, ou seja, ou Maria é Carioca ou é gaúcha. Então temos, na proposição q, uma disjunção exclusiva pois o ou é exclusivo, enquanto que na proposição p o ou é inclusivo.

Definição: Dada duas proposições p e q, define-se por disjunção exclusiva o operador “ou exclusivo” (simbolicamente representado por $\underline{\vee}$), onde “ $p \underline{\vee} q$ ” possui o valor lógico Verdade se, e apenas se, uma das proposições ($p \underline{\vee} q$) for verdade. Da mesma forma, pode possuir o valor lógico Falso se, e apenas se, as duas proposições ($p \underline{\vee} q$) forem verdadeiras ou as duas proposições forem falsas.

Simbolicamente, a representação da disjunção exclusiva entre duas proposições é representada por “ $p \underline{\vee} q$ ” onde se lê “p ou exclusivo q”. O valor lógico da disjunção de duas proposições é, portanto, definido pela tabela verdade a seguir:

p	q	$p \underline{\vee} q$
V	V	F
V	F	V
F	V	V
F	F	F

Assim, podemos concluir que:

$$V \underline{\vee} V = F$$

$$F \underline{\vee} F = V$$

$$V \underline{\vee} F = V$$

$$F \underline{\vee} V = F$$

Em termos gerais temos ainda que:

$$V(p \underline{\vee} q) = (V(p) \underline{\vee} F(q)) \vee (F(p) \underline{\vee} V(q))$$

Alguns exemplos do operador de disjunção exclusiva:

$$p = 2+2 = 4$$

$$q = 1+1 = 3$$

$$p \underline{\vee} q = V$$

$$p = \text{Brasília é a capital do Brasil}$$

$$q = 1+1 = 2$$

$$p \underline{\vee} q = F$$

1.2.5 Condicional

Definição: Dada duas proposições p e q, define-se por condicional o operador “se” (simbolicamente representado por $-->$), onde “ $p --> q$ ” possui o valor lógico falso se p for verdade e q for falso. Em todos os outros casos o valor lógico sempre será verdadeiro.

Simbolicamente, a representação do condicional entre duas proposições é representada por “ $p \rightarrow q$ ” onde se lê “se p então q ”. O valor lógico do condicional de duas proposições é, portanto, definido pela tabela verdade a seguir:

p	q	$p \rightarrow q$
V	V	V
V	F	F
F	V	V
F	F	V

Assim, podemos concluir que:

$$V \rightarrow V = V$$

$$F \rightarrow F = V$$

$$V \rightarrow F = F$$

$$F \rightarrow V = V$$

Em termos gerais temos ainda que:

$$V(p \rightarrow q) = V(p) \rightarrow V(q)$$

Alguns exemplos do operador de condicional:

$$p = 2+2 = 4$$

$$q = 1+1 = 3$$

$$p \rightarrow q = V$$

$$p = \text{Brasília é a capital do Brasil}$$

$$q = 1+1 = 2$$

$$p \rightarrow q = V$$

1.2.6 Bicondicional

Definição: Dada duas proposições p e q , define-se por bicondicional o operador “se e somente se” (simbolicamente representado por \leftrightarrow), onde “ $p \leftrightarrow q$ ” possui o valor lógico Verdade se ambas as proposições forem verdadeiras ou falsas. Em todos os outros casos o valor lógico sempre será Falso.

Simbolicamente, a representação do bicondicional entre duas proposições é representada por “ $p \leftrightarrow q$ ”, onde se lê “ p se e somente se q ”. O valor lógico do bicondicional de duas proposições é, portanto, definido pela tabela verdade a seguir:

p	q	$p \leftrightarrow q$
V	V	V
V	F	F
F	V	F
F	F	V

Assim, podemos concluir que:

$$V \leftrightarrow V = V$$

$$F \leftrightarrow F = V$$

$$V \leftrightarrow F = F$$

$$F \leftrightarrow V = F$$

Em termos gerais temos ainda que:

$$V(p \leftrightarrow q) = V(p) \leftrightarrow V(q)$$

Alguns exemplos do operador de bicondicional:

$$p = 2+2 = 4$$

$$q = 1+1 = 3$$

$$p \leftrightarrow q = F$$

$$p = \text{Brasília é a capital do Brasil}$$

$$q = 1+1 = 2$$

$$p \leftrightarrow q = V$$

ATIVIDADES – UNIDADE 1

1. Dada as seguintes proposições:

p : está quente

q : está chovendo

Traduzir para a linguagem natural as seguintes proposições:

a) $\sim p$

b) $p \wedge q$

c) $p \vee q$

d) $q \leftrightarrow p$

e) $P \rightarrow \sim q$

f) $p \vee \sim q$

g) $\sim p \wedge \sim q$

h) $p \leftrightarrow \sim q$

i) $p \wedge \sim q \rightarrow p$

2. Dada as seguintes proposições:

p : Maria é bonita

q : Carlos é feliz

Traduzir para a linguagem natural as seguintes proposições:

a) $Q \rightarrow p$

b) $\sim \sim p$

c) $\sim(\sim p \wedge \sim q)$

2

TABELA VERDADE

INTRODUÇÃO

As tabelas verdade mostram como determinamos o valor de verdade de sentenças formadas com os operadores da lógica vistos anteriormente. As tabelas de verdade exprimem o significado dos operadores relevantes para a lógica proposicional clássica. São representadas por tabelas, como o próprio nome já diz, onde, no lado esquerdo da tabela de verdade, temos as sentenças a partir das quais a sentença composta foi formada. O valor da sentença composta é dado pela coluna da direita.

Na lógica proposicional, as tabelas verdade podem ser utilizadas para resolver diversos problemas; no entanto, também possuem limitações. Como veremos, resolver uma tabela verdade com poucos elementos é trivial; porém, uma proposição composta com diversos elementos distintos pode ser bem trabalhosa de resolver. Mesmo assim, a forma de resolução segue a mesma sempre.

Neste capítulo, estudaremos as tabelas verdade e como podem ser utilizadas para determinar se uma proposição é verdadeira ou não. No final do capítulo, são apresentados diversos exercícios extremamente importantes para a compreensão das tabelas verdade.

2.1

CONCEITOS BÁSICOS

A construção de uma tabela verdade envolve os operadores apresentados anteriormente e a combinação de várias proposições simples, que, por sua vez, tornam-se uma única proposição composta. Desta forma, é possível construir a tabela verdade para qualquer proposição composta usando os operadores lógicos. O resultado de uma proposição composta será verdadeiro (V) ou falso (F).

O número de linhas da tabela verdade de uma proposição composta depende do número de proposições simples, e pode ser definida pelo seguinte teorema:

Teorema: a tabela verdade de uma proposição composta representada por n proposições simples possui 2^n linhas.

Demonstração: o teorema nada mais é do que uma análise combinatória onde tem-se: para cada proposição simples um único valor lógico é assumido (V ou F). Assim, uma proposição composta $R(p_1, p_2, \dots, p_n)$, onde cada uma das proposições p pode assumir um dos valores lógicos e a combinação das n proposições acontece com a combinação de todas as possibilidades (n a n), considerando os valores lógicos V ou F. Por exemplo, para $R(p_1, p_2, p_3, p_4, p_5)$ tem-se que o número de linhas será $2^5 = 32$ linhas.

A construção de uma tabela verdade pode acontecer por diferentes formas, pois depende da decomposição de uma proposição composta, que pode acontecer de diferentes maneiras. Por exemplo, a proposição composta:

$$\sim(p \wedge \sim q)$$

Pode ser reescrita como:

$$\sim p \wedge q$$

Observa-se que essa decomposição é semelhante à decomposição realizada em operações com equações matemáticas, onde os parênteses têm uma função importante na forma de resolução.

Vamos considerar o seguinte exemplo:

$$R(p, q) = \sim(p \wedge \sim q)$$

Temos que a proposição composta R possui duas proposições simples. Assim, a tabela verdade deverá ter 4 linhas ($2^2 = 4$).

p	q	$\sim q$	$p \wedge \sim q$	$\sim(p \wedge \sim q)$
V	V	F	F	V
V	F	V	V	F
F	V	F	F	V
F	F	V	F	V

O primeiro passo é colocar os valores das proposições simples, no caso p e q. Também podemos decompor o $\sim q$ que representa a negação de q. Após, podemos resolver o operador de conjunção e, por fim, aplicamos o operador de negação e temos o resultado da proposição composta.

Vejamos agora um exemplo mais complexo. Dada a seguinte proposição:

$$R(p,q,t) = p \vee \sim t \rightarrow q \wedge \sim t$$

Temos que a quantidade de linhas da tabela verdade será de 8 linhas ($2^3 = 8$):

p	q	t	$\sim t$	$p \vee \sim t$	$q \wedge \sim t$	$p \vee \sim t \rightarrow q \wedge \sim t$
V	V	V	F	V	F	F
V	V	F	V	V	V	V
V	F	V	F	V	F	F
V	F	F	V	V	F	F
F	V	V	F	F	F	V
F	V	F	V	V	V	V
F	F	V	F	F	F	V
F	F	F	V	V	F	F

Observamos que novamente atribuímos valores lógicos para as 3 proposições simples, no caso p, q e t. Essa atribuição geralmente se dá da seguinte forma: a primeira proposição (p) assume metade dos valores verdade e metade falso; a segunda proposição, metade da proposição anterior; e a terceira, metade da anterior. Desta forma, garantimos todas as combinações de Vs e Fs possíveis. Após a definição dos valores lógicos de cada proposição simples, começamos a decompor as operações, sendo a primeira a negação do valor t, ou seja, $\sim t$. Depois resolvemos a disjunção e a conjunção. Por fim, resolvemos o condicional e temos como resultado os valores lógicos apresentados na última coluna.

Agora, resolveremos uma tabela verdade um pouco mais complexa, com 5 proposições simples:

$$R(p,q,r,t,s) = (p \vee q) \wedge (r \vee t) \rightarrow s$$

Desta forma temos que a quantidade de linhas da tabela verdade será de 8 linhas ($2^5 = 32$):

p	q	r	t	s	pvt	rvt	$(pvq) \wedge (rvt)$	$(pvq) \wedge (rvt) \rightarrow s$
V	V	V	V	V	V	V	V	V
V	V	V	V	F	V	V	V	F
V	V	V	F	V	V	V	V	V
V	V	V	F	F	V	V	V	F
V	V	F	V	V	V	V	V	V
V	V	F	V	F	V	V	V	F
V	V	F	F	V	V	F	F	V
V	V	F	F	F	V	F	F	V
V	F	V	V	V	V	V	V	V
V	F	V	V	F	V	V	V	F
V	F	V	F	V	V	V	V	V
V	F	V	F	F	V	V	V	F
V	F	F	V	V	V	V	V	V
V	F	F	V	F	V	V	V	F
V	F	F	F	V	V	F	F	V
V	F	F	F	F	V	F	F	F
F	V	V	V	V	V	V	V	V
F	V	V	V	F	V	V	V	F
F	V	V	F	V	V	V	V	V
F	V	V	F	F	V	V	V	F
F	V	F	V	V	V	V	V	V
F	V	F	V	F	V	V	V	F
F	V	F	F	V	V	F	F	V
F	V	F	F	F	V	F	F	F
F	F	V	V	V	F	V	F	V
F	F	V	V	F	F	V	F	V
F	F	V	F	V	F	V	F	V
F	F	V	F	F	F	V	F	V
F	F	F	V	V	F	V	F	V
F	F	F	V	F	F	V	F	V
F	F	F	F	V	F	F	F	V
F	F	F	F	F	F	F	F	V

Observamos que podemos resolver qualquer proposição composta apenas decompondo as proposições simples e suas operações. No entanto, quanto mais proposições simples maior será a tabela verdade.

2.2

IMPLICAÇÕES E EQUIVALÊNCIAS

Existe uma relação entre proposições que pode ser de implicação ou equivalência. É de fundamental importância para o estudo da lógica matemática entender essas relações. As proposições podem ser também independentes, ou seja, as proposições possuem os 4 valores lógicos, por exemplo:

p	q
V	V
V	F
F	V
F	F

Quando as proposições são dependentes, as suas tabelas verdade não possuem alguma possível combinação de Vs e Fs., por exemplo:

p	q	$p \rightarrow q$
V	V	F
V	F	V
F	V	F
F	F	F

Observa-se que ocorre as alternativas VF, VV e FF, mas não ocorre a alternativa FV. Assim, podemos dizer que existe uma relação simples entre as proposições p e $q \rightarrow p$.

A relação de implicação entre duas proposições, representada simbolicamente por \Rightarrow , é diferente da operação condicional representada por \rightarrow , ou seja, enquanto uma operação lógica dá origem a uma nova proposição, quando definimos que $p \Rightarrow q$ apenas indicamos uma relação entre as proposições.

A relação de implicação entre duas proposições ocorre apenas quando em suas tabelas verdade não ocorre a alternativa FV. Por exemplo, podemos verificar se $p \Rightarrow q \rightarrow p$, ou seja, se p implica em se q então p:

p	q	$p \rightarrow q$
V	V	F
V	F	V
F	V	F
F	F	F

Portanto, verifica-se que a alternativa FV não ocorre, sendo assim: $p \Rightarrow q \rightarrow p$.

As proposições também podem ser equivalentes. Simbolicamente temos: $p \iff q$ onde significa que p é equivalente a q . Por exemplo, podemos verificar se:

$$p \wedge q \iff \sim(\sim p \vee \sim q)$$

onde temos a seguinte tabela verdade:

p	q	$p \wedge q$	$\sim p$	$\sim q$	$\sim p \vee \sim q$	$\sim(\sim p \vee \sim q)$
V	V	V	F	F	F	V
V	F	F	F	V	V	F
F	V	F	V	F	V	F
F	F	F	V	V	V	F

Podemos concluir que existe equivalência entre as proposições, ou seja, $p \wedge q \iff \sim(\sim p \vee \sim q)$. Observamos que não existe a ocorrência de VF e FV na mesma linha. De forma bem simples, podemos concluir que quando as tabelas verdade são iguais temos uma relação de equivalência.

Algumas equivalências são bem simples, por exemplo, a dupla negação. Considerando a seguinte proposição:

$$R(p) = \sim(\sim p) \iff p$$

onde a dupla negação em p é exatamente a própria proposição p . Podemos observar a equivalência também pela tabela verdade:

p	$\sim p$	$\sim(\sim p)$
V	F	V
F	V	F

Para a conjunção e disjunção também seguimos as mesmas regras, também conhecidas por:

Leis Idempotentes:

$$R(p) = p \wedge p \iff p$$

$$R(p) = p \vee p \iff p$$

Ou seja, p conjunção ou disjunção com ele mesmo sempre resulta em p , como podemos observar na tabela verdade abaixo:

p	$p \wedge p$	$p \vee p$
V	V	V
F	F	F

Quando consideramos proposições diferentes temos as seguintes regras, que são determinadas por leis.

Leis cumulativas:

$$R(p,q) = p \wedge q \iff q \wedge p$$

$$R(p,q) = p \vee q \iff q \vee p$$

As tabelas verdade correspondente são:

p	q	$p \wedge q$	$q \wedge p$
V	V	V	V
V	F	F	F
F	V	F	F
F	F	F	F

p	q	$p \vee q$	$q \vee p$
V	V	V	V
V	F	V	V
F	V	V	V
F	F	F	F

Leis Associativas:

$$R(p,q,r) = p \wedge (q \wedge r) \iff (p \wedge q) \wedge r$$

$$R(p,q,r) = p \vee (q \vee r) \iff (p \vee q) \vee r$$

Observamos que as leis associativas demonstram que o uso dos parentes, dada a mesma operação, gera uma equivalência. As tabelas verdade correspondente são:

p	q	r	$q \wedge r$	$p \wedge (q \wedge r)$	$p \wedge q$	$(p \wedge q) \wedge r$
V	V	V	V	V	V	V
V	V	F	F	F	V	F
V	F	V	F	F	F	F
V	F	F	F	F	F	F
F	V	V	V	F	F	F
F	V	F	F	F	F	F
F	F	V	F	F	F	F
F	F	F	F	F	F	F

Da mesma forma, para a disjunção temos:

p	q	r	$q \vee r$	$p \vee (q \vee r)$	$p \vee q$	$(p \vee q) \vee r$
V	V	V	V	V	V	V
V	V	F	V	V	V	V
V	F	V	V	V	V	V
V	F	F	F	V	V	V
F	V	V	V	V	V	V
F	V	F	V	V	V	V
F	F	V	V	V	F	V
F	F	F	F	F	F	F

Leis de Morgan:

$$R(p,q) = \sim(p \wedge q) \iff \sim p \vee \sim q$$

$$R(p,q) = \sim(p \vee q) \iff \sim p \wedge \sim q$$

As respectivas tabelas verdade são:

p	q	$\sim p$	$\sim q$	$p \wedge q$	$\sim(p \wedge q)$	$\sim p \vee \sim q$
V	V	F	F	V	F	F
V	F	F	V	F	V	V
F	V	V	F	F	V	V
F	F	V	V	F	V	V

p	q	$\sim p$	$\sim q$	$p \vee q$	$\sim(p \vee q)$	$\sim p \wedge \sim q$
V	V	F	F	V	F	F
V	F	F	V	V	F	F
F	V	V	F	V	F	F
F	F	V	V	F	V	V

Leis Distributivas:

$$R(p,q,r) = p \wedge (q \vee r) \iff (p \wedge q) \vee (p \wedge r)$$

$$R(p,q,r) = p \vee (q \wedge r) \iff (p \vee q) \wedge (p \vee r)$$

As respectivas tabelas verdade são:

p	q	r	$q \vee r$	$p \wedge (q \vee r)$	$p \wedge q$	$p \wedge r$	$(p \wedge q) \vee (p \wedge r)$
V	V	V	V	V	V	V	V
V	V	F	V	V	V	F	V
V	F	V	V	V	F	V	V
V	F	F	F	F	F	F	F
F	V	V	V	F	F	F	F
F	V	F	V	F	F	F	F
F	F	V	V	F	F	F	F
F	F	F	F	F	F	F	F

Bicondicional:

$$R(p,q) = p \leftrightarrow q \iff (p \rightarrow q) \wedge (q \rightarrow p)$$

A tabela verdade correspondente é:

p	q	$p \leftrightarrow q$	$p \rightarrow q$	$q \rightarrow p$	$(p \rightarrow q) \wedge (q \rightarrow p)$
V	V	V	V	V	V
V	F	F	F	V	F
F	V	F	F	F	F
F	F	V	V	V	V

Condicionais:

$$R(p,q) = p \rightarrow q$$

$$R(p,q) = \sim p \rightarrow \sim q \text{ (contrapositivo)}$$

$$R(p,q) = q \rightarrow p \text{ (recíproca do condicional)}$$

$$R(p,q) = \sim q \rightarrow \sim p \text{ (recíproca do contrapositivo)}$$

p	q	$\sim p$	$\sim q$	$p \rightarrow q$	$\sim p \rightarrow \sim q$	$q \rightarrow p$	$\sim q \rightarrow \sim p$
V	V	F	F	V	V	V	V
V	F	F	V	F	V	V	F
F	V	V	F	V	F	F	V
F	F	V	V	V	V	V	V

Desta forma, podemos concluir as seguintes equivalências:

$$p \rightarrow q \iff \sim q \rightarrow \sim p$$

$$q \rightarrow p \iff \sim p \rightarrow \sim q$$

2.3

TAUTOLOGIAS

Definição: é toda proposição composta na qual a sua última coluna da tabela verdade é sempre V.

Ou seja, tautologia é toda proposição composta $R(p,q,r\dots)$ cujo valor lógico é sempre verdade, independente de quaisquer que sejam os valores lógicos das proposições simples componentes $(p,q,r\dots)$. Por exemplo:

$R(p) = p \vee \sim p$ possui a seguinte tabela verdade:

p	$\sim p$	$p \vee \sim p$
V	F	V
F	V	V

Assim, concluímos que a proposição $p \vee \sim p$ é uma tautologia e podemos demonstrar duas importantes propriedades:

a) A condição necessária e suficiente para que $p \implies q$ é que o condicional $p \implies q$ seja uma tautologia.

Demonstração:

A condição é necessária: $(p \implies q) \implies (p \implies q)$. Se $p \implies q$, não ocorre VF, então o condicional $p \implies q$ é uma tautologia.

A condição é suficiente: $(p \implies q) \implies (p \implies q)$. Se $(p \implies q)$, não ocorre em sua tabela verdade a alternativa VF, então $p \implies q$.

b) A condição necessária e suficiente para que $p \iff q$ é que o condicional $p \iff q$ seja uma tautologia.

Demonstração:

A condição é necessária: $(p \iff q) \iff (p \iff q)$. Se $p \iff q$, não ocorre VF, então o bicondicional $p \iff q$ é uma tautologia.

A condição é suficiente: $(p \iff q) \iff (p \iff q)$. Se $(p \iff q)$, não ocorre em sua tabela verdade a alternativa VF, então $p \iff q$.

ATIVIDADES – UNIDADE 2

1. Sabendo-se que $V(p) = V(q) = T$ (true) e $V(r) = V(s) = F$ (false), determine os valores lógicos das seguintes proposições:

- $(p \wedge (q \vee r)) \rightarrow (p \rightarrow (r \vee q))$
- $(q \rightarrow r) \iff (\sim q \vee r)$
- $(\sim p \vee \sim(r \wedge s))$
- $\sim(q \iff (\sim p \wedge s))$
- $(p \iff q) \vee (q \rightarrow \sim p)$
- $\sim(\sim q \wedge (p \wedge \sim s))$
- $\sim q \wedge ((\sim r \vee s) \iff (p \rightarrow \sim q))$
- $\sim(\sim p \vee (q \wedge s)) \rightarrow (r \rightarrow \sim s)$
- $\sim(p \rightarrow (q \rightarrow r)) \rightarrow s$

2. Determinar $P(VV, VE, FV, FF)$ em cada um dos seguintes casos:

- $P(p, q) = \sim(\sim p \iff q)$
- $P(p, q) = \sim p \vee q \rightarrow p$
- $P(p, q) = (p \vee q) \wedge \sim(p \wedge q)$
- $P(p, q) = (p \vee \sim q) \vee (\sim p \vee q)$
- $P(p, q) = \sim((p \vee q) \wedge (\sim p \vee \sim q))$

3. Determinar $P(VVV, VVE, VFV, VFE, FVV, FVE, FFV, FFF)$ em cada um dos seguintes casos:

- $P(p, q, r) = p \vee (q \wedge r)$
- $P(p, q, r) = (p \wedge \sim q) \vee r$
- $P(p, q, r) = \sim p \vee (q \wedge \sim r)$
- $P(p, q, r) = (p \vee q) \wedge (p \vee r)$
- $P(p, q, r) = (p \vee \sim r) \wedge (q \vee \sim r)$

4. Determinar $P(VFV)$ em cada um dos seguintes casos:

- $P(p, q, r) = p \wedge \sim r \rightarrow \sim q$
- $P(p, q, r) = \sim p \wedge (q \vee \sim r)$
- $P(p, q, r) = \sim(p \wedge q) \iff \sim(p \vee \sim r)$
- $P(p, q, r) = (r \wedge (p \vee \sim q)) \wedge \sim(\sim r \vee (p \wedge q))$
- $P(p, q, r) = (p \wedge q \rightarrow r) \rightarrow q \vee \sim r$

5. Construir as tabelas verdade para as proposições abaixo:

- $P(p, q) = (\sim p \vee \sim q)$
- $P(p, q, r) = p \vee \sim r \rightarrow q \wedge \sim r$
- $P(p, q) = \sim(p \wedge q) \vee \sim(q \iff p)$
- $P(p, q, r) = (p \wedge q \rightarrow r) \vee (\sim p \iff q \vee \sim r)$
- $P(p, q) = \sim(p \vee \sim q)$
- $P(p, q) = \sim(p \rightarrow \sim q)$
- $P(p, q) = p \wedge q \rightarrow p \vee q$

- h) $P(p,q) = \sim p \rightarrow (q \rightarrow p)$
- i) $P(p,q) = (p \rightarrow q) \rightarrow p \wedge q$
- j) $P(p,q,r) = \sim p \wedge r \rightarrow q \vee \sim r$
- k) $P(p,q,r) = p \rightarrow r \iff q \vee \sim r$
- l) $P(p,q,r) = p \rightarrow (p \rightarrow \sim r) \iff q \vee r$
- m) $P(p,q,r) = (p \wedge q \rightarrow r) \vee (\sim p \iff q \vee \sim r)$

3

MÉTODO DEDUTIVO

INTRODUÇÃO

Método dedutivo é um processo lógico no qual uma conclusão é baseada na concordância de múltiplas premissas que são, em geral, consideradas verdade. Em contraste com o método dedutivo, temos o método indutivo, o qual não será abordado nesse material didático. O método dedutivo também é considerado como um método *top-down* (lógica de cima para baixo), ou seja, dada uma sentença geral, a mesma é deduzida em uma conclusão específica.

Um exemplo clássico foi introduzido pelo filósofo Aristóteles, considerado o pai do método dedutivo:

- » Todos os homens são mortais
- » Sócrates é um homem
- » Logo, Sócrates é mortal

No exemplo acima, que estabelece que todos os homens são mortais e Sócrates é um homem, observa-se que a evidência é verdadeira, pois a premissa estabelece que Sócrates é um indivíduo do grupo onde os membros são mortais e, desta forma, Sócrates é mortal.

Neste capítulo, estudaremos o método dedutivo e como podemos estabelecer verdades ou falsidades dada uma premissa. Esse estudo é extremamente importante para a lógica proposicional, pois permite a verificação de conclusões lógicas.

3.1

DEFINIÇÕES

Como vimos até agora, todas as implicações e equivalências foram apresentadas e demonstradas usando tabelas verdade. Uma forma mais eficiente de demonstrar implicações e equivalências é o chamado método dedutivo.

Definição: é o processo para estabelecer de forma rigorosa a validade dos argumentos, derivando a conclusão do argumento a partir das proposições e usando um sistema de regras.

Ou seja, se há certa afirmação (ou negação) que aceitamos como verdadeira, argumentamos para justificar porque é que todos devem aceitá-la como verdadeira. Em outras palavras, acreditamos em alguma coisa e queremos provar que aquilo em que acreditamos é verdadeiro.

Será utilizada álgebra das proposições, na qual os exemplos são baseados na nomenclatura abaixo, também utilizada por inúmeros outros autores:

Proposições Simples:

p,q,r,t (Verdadeira)
c (Falsa)

Proposições Compostas:

P,Q,R,T (Verdadeira)
C (Contradição)

3.1.1 Exemplificando

Vamos exemplificar como utilizar o método dedutivo com base em vários exemplos:

1. Demonstrar as implicações:

- a) $c \implies p$
- b) $p \implies t$

onde p é uma proposição qualquer e c e t são proposições cujos valores lógicos são respectivamente F (falso) e V (verdade). Então temos que:

- a) $c \implies p \iff \sim c \vee p$
 $\iff t \vee p$
 $\iff t$
- b) $p \implies t \iff \sim p \vee t$
 $\iff t$

Podemos observar claramente as duas demonstrações com o auxílio da tabela verdade:

p	c	t	$c \rightarrow t$	$p \rightarrow t$
V	F	V	V	V
F	F	V	V	V

A seguir, iremos demonstrar várias implicações usando o método dedutivo, considerando diferentes regras:

2. Considerando a seguinte implicação:

$p \wedge q \Rightarrow p$ (**Simplificação**)

temos, sucessivamente, que:

$$\begin{aligned}
 p \wedge q \rightarrow p &\iff \sim(p \wedge q) \vee p \\
 &\iff (\sim p \vee \sim q) \vee p \\
 &\iff (\sim p \vee p) \vee \sim q \\
 &\iff T \vee \sim q \\
 &\iff T
 \end{aligned}$$

3. Considerando a seguinte implicação:

$p \Rightarrow p \vee q$ (**Adição**)

temos, sucessivamente, que:

$$\begin{aligned}
 p \rightarrow p \vee q &\iff \sim p \vee (p \vee q) \\
 &\iff (\sim p \vee p) \vee q \\
 &\iff T \vee q \\
 &\iff T
 \end{aligned}$$

4. Considerando a seguinte implicação:

$(p \rightarrow q) \wedge p \Rightarrow q$ (**Modus ponens**)

temos, sucessivamente, que:

$$\begin{aligned}
 (p \rightarrow q) \wedge p &\iff p \wedge (\sim p \vee q) \\
 &\iff (\sim p \wedge p) \vee (p \wedge q) \\
 &\iff C \vee (p \wedge q) \\
 &\iff p \wedge q \Rightarrow q
 \end{aligned}$$

5. Considerando a seguinte implicação:

$(p \rightarrow q) \wedge \sim q \Rightarrow \sim p$ (**Modus tollens**)

temos, sucessivamente, que:

$$\begin{aligned}
 (p \rightarrow q) \wedge \sim q &\iff (\sim p \vee q) \wedge \sim q \\
 &\iff (\sim p \vee \sim q) \vee (q \vee \sim q) \\
 &\iff (\sim p \wedge \sim q) \vee C \\
 &\iff \sim p \wedge \sim q \Rightarrow \sim p
 \end{aligned}$$

6. Considerando a seguinte implicação:
 $(p \vee q) \wedge \sim p \Rightarrow q$ (**Silogismo disjuntivo**)

temos, sucessivamente, que:

$$\begin{aligned} (p \vee q) \wedge \sim p &\iff (p \wedge \sim p) \vee (q \wedge \sim p) \\ &\iff C \vee (q \wedge \sim p) \\ &\iff q \wedge \sim p \Rightarrow q \end{aligned}$$

7. Considerando a seguinte implicação:

$$p \wedge q \Rightarrow p \vee q$$

temos, sucessivamente, que:

$$\begin{aligned} p \wedge q \dashv\vdash p \vee q &\iff \sim(p \wedge q) \vee (p \vee q) \\ &\iff (\sim p \vee \sim q) \vee (p \vee q) \\ &\iff (\sim p \vee p) \vee (\sim q \vee q) \\ &\iff T \vee T \Rightarrow T \end{aligned}$$

8. Considerando a seguinte implicação:

$$p \Rightarrow q \dashv\vdash p$$

temos, sucessivamente, que:

$$\begin{aligned} p \dashv\vdash (q \dashv\vdash p) &\iff \sim p \vee (q \dashv\vdash p) \\ &\iff \sim p \vee (\sim q \vee p) \iff (\sim p \vee p) \vee \sim q \\ &\iff T \vee \sim q \\ &\iff T \end{aligned}$$

9. Considerando a seguinte implicação:

$$p \Rightarrow \sim p \dashv\vdash q$$

temos, sucessivamente, que:

$$\begin{aligned} p \dashv\vdash (\sim p \dashv\vdash q) &\iff \sim p \vee (\sim p \dashv\vdash q) \\ &\iff \sim p \vee (\sim \sim p \vee q) \\ &\iff \sim p \vee (p \vee q) \\ &\iff (\sim p \vee p) \vee q \\ &\iff T \vee q \\ &\iff T \end{aligned}$$

10. Considerando a seguinte implicação:

$$p \dashv\vdash q \Rightarrow p \wedge r \dashv\vdash q$$

temos, sucessivamente, que:

$$\begin{aligned} p \dashv\vdash q \dashv\vdash p \wedge r \dashv\vdash q &\iff \sim(p \dashv\vdash q) \vee (p \wedge r \dashv\vdash q) \\ &\iff \sim(\sim p \vee q) \vee (\sim(p \wedge r) \vee q) \\ &\iff (\sim \sim p \wedge \sim q) \vee ((\sim p \vee \sim r) \vee q) \\ &\iff (p \wedge \sim q) \vee ((\sim p \vee q) \vee r) \\ &\iff (p \wedge \sim q) \vee \sim(p \wedge \sim q) \vee \sim r \end{aligned}$$

$$\iff T \vee \sim r$$

$$\iff T$$

11. Considerando a seguinte equivalência:

$$p \rightarrow q \iff p \wedge \sim q \rightarrow c \text{ (Redução a absurdo)}$$

temos, sucessivamente, que:

$$p \wedge \sim q \rightarrow c \iff \sim(p \wedge \sim q) \vee c$$

$$\iff \sim(p \wedge \sim q)$$

$$\iff \sim p \vee \sim \sim q$$

$$\iff \sim p \vee q$$

$$\iff p \rightarrow q$$

12. Considerando a seguinte equivalência:

$$p \rightarrow q \iff p \vee q \rightarrow q$$

temos, sucessivamente, que:

$$p \vee q \rightarrow q \iff \sim(p \vee q) \vee q$$

$$\iff (\sim p \wedge \sim q) \vee q$$

$$\iff (\sim p \vee q) \wedge (\sim q \vee q)$$

$$\iff (\sim p \vee q) \wedge T$$

$$\iff \sim p \vee q$$

$$\iff p \rightarrow q$$

13. Considerando a seguinte equivalência:

$$(p \rightarrow q) \wedge (p \rightarrow \sim q) \iff \sim p$$

temos, sucessivamente, que:

$$(p \rightarrow q) \wedge (p \rightarrow \sim q) \iff (\sim p \vee q) \wedge (\sim p \vee \sim q)$$

$$\iff \sim p \wedge (q \wedge \sim q)$$

$$\iff \sim p \vee C$$

$$\iff \sim p$$

14. Considerando a seguinte equivalência:

$$p \wedge q \rightarrow r \iff p \rightarrow (q \rightarrow r) \text{ (Exportação-Importação)}$$

temos, sucessivamente, que:

$$p \rightarrow (q \rightarrow r) \iff \sim p \vee (q \rightarrow r)$$

$$\iff \sim p \vee (\sim q \vee r)$$

$$\iff (\sim p \vee \sim q) \vee r$$

$$\iff \sim(p \wedge q) \vee r$$

$$\iff p \wedge q \rightarrow r$$

15. Considerando a seguinte equivalência:

$$(p \rightarrow r) \wedge (q \rightarrow r) \iff p \vee q \rightarrow r$$

temos, sucessivamente, que:

$$\begin{aligned} (p \rightarrow r) \wedge (q \rightarrow r) &\iff (\sim p \vee r) \wedge (\sim q \vee r) \\ &\iff (\sim p \vee \sim q) \wedge r \\ &\iff \sim(p \vee q) \vee r \\ &\iff p \vee q \rightarrow r \end{aligned}$$

16. Considerando a seguinte equivalência:

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

temos, sucessivamente, que:

$$\begin{aligned} (p \rightarrow q) \vee (p \rightarrow r) &\iff (\sim p \vee q) \vee (\sim p \vee r) \\ &\iff (\sim p \vee \sim p) \vee (q \vee r) \\ &\iff \sim p \vee (q \vee r) \\ &\iff p \rightarrow q \vee r \end{aligned}$$

17. Considerando a seguinte equivalência:

$$(p \rightarrow r) \vee (q \rightarrow s) \iff p \wedge q \rightarrow r \vee s$$

temos, sucessivamente, que:

$$\begin{aligned} (p \rightarrow r) \vee (q \rightarrow s) &\iff (\sim p \vee r) \vee (\sim q \vee s) \\ &\iff (\sim p \vee \sim q) \vee (r \vee s) \\ &\iff \sim(p \wedge q) \vee (r \vee s) \\ &\iff p \wedge q \rightarrow r \vee s \end{aligned}$$

Como observamos nessas demonstrações, uma das formas para provar a implicação ou equivalência é na diminuição do número de operadores. Para isso, temos o seguinte teorema:

Teorema: entre os cinco operadores fundamentais (\sim , \vee , \wedge , \rightarrow , \leftrightarrow), três exprimem-se em termos de apenas dois dos seguintes passos:

- (i) \sim e \vee
- (ii) \sim e \wedge
- (iii) \sim e \rightarrow

Por exemplo:

(i) \wedge , \rightarrow e \leftrightarrow exprimem-se em termos de \sim e \vee

$$\begin{aligned} p \wedge q &\iff \sim \sim p \wedge \sim \sim q \\ &\iff \sim(\sim p \vee \sim q) \\ p \rightarrow q &\iff \sim p \vee q \\ p \leftrightarrow q &\iff (p \rightarrow q) \wedge (q \rightarrow p) \\ &\iff \sim(\sim(\sim p \vee q) \vee \sim(\sim q \vee p)) \end{aligned}$$

(ii) \vee , \rightarrow e \leftrightarrow exprimem-se em termos de \sim e \wedge

$$\begin{aligned} p \vee q &\iff \sim \sim p \vee \sim \sim q \\ &\iff \sim(\sim p \wedge \sim q) \\ p \rightarrow q &\iff \sim p \vee q \end{aligned}$$

$$\begin{aligned} & \iff \sim(p \wedge \sim q) \\ p \leftrightarrow q & \iff (p \rightarrow q) \wedge (q \rightarrow p) \\ & \iff \sim(p \wedge \sim q) \wedge \sim(\sim p \wedge q) \end{aligned}$$

(iii) $\vee, \wedge, \leftrightarrow$ exprimem-se em termos de \sim e \rightarrow

$$\begin{aligned} p \wedge q & \iff \sim(\sim p \vee \sim q) \\ & \iff \sim(p \rightarrow \sim q) \\ p \vee q & \iff \sim\sim p \vee q \\ & \iff \sim p \rightarrow q \\ p \leftrightarrow q & \iff (p \rightarrow q) \wedge (q \rightarrow p) \\ & \iff \sim((p \rightarrow q) \rightarrow \sim(q \rightarrow p)) \end{aligned}$$

Para os operadores mais comuns (negação, junção e disjunção) define-se:

Definição: uma proposição está em sua Forma Normal (FN) se e somente se contém no máximo os operadores \sim, \vee e \wedge .

Por exemplo, as seguintes proposições estão em sua forma normal:

- » $p \vee q$
- » $p \wedge (p \vee q)$
- » $p \rightarrow q$
- » $p \vee (p \rightarrow (p \wedge q))$

É importante observar que qualquer proposição pode ser levada a sua FN equivalente pela eliminação dos operadores \rightarrow e \leftrightarrow , observando as seguintes regras de substituição:

- (i) $p \rightarrow q$ substituindo o operador temos $\sim p \vee q$
- (ii) $p \leftrightarrow q$ substituindo o operador temos $(\sim p \vee q) \wedge (p \vee \sim q)$

A Forma Normal das proposições pode ser classificada em Forma Normal Conjuntiva (FNC) e Forma Normal Disjuntiva (FND), como veremos a seguir.

3.1.2 Forma Normal Conjuntiva

Definição: uma proposição encontra-se na Forma Normal Conjuntiva (FNC) se e somente se ela satisfaz as seguintes condições:

- a) Contém no máximo os seguintes operadores: \sim, \wedge, \vee
- b) \sim não aparece repetido (por exemplo $\sim\sim p$) e não tem alcance sobre \wedge e \vee , ou seja, só incide sobre proposições simples (letras proposicionais)
- c) \vee não tem alcance sobre \wedge (por exemplo, não há proposições compostas do tipo $p \vee (p \wedge q)$)

Por exemplo, as seguintes proposições se enquadram em uma FNC:

$$\begin{aligned} & \sim p \vee \sim q \\ & (p \wedge q) \wedge \sim q \end{aligned}$$

Para transformar uma proposição qualquer em uma FNC equivalente, aplicam-se as seguintes regras:

(1) Eliminar os operadores \rightarrow e \leftrightarrow substituindo:

$$\begin{aligned} p \rightarrow q & \text{ por } \sim p \vee q \text{ e} \\ p \leftrightarrow q & \text{ por } (\sim p \vee q) \wedge (p \vee \sim q) \end{aligned}$$

(2) Eliminar negações repetidas e parênteses precedidos por \sim pelas regras de dupla negação e Morgan.

(3) Substituir:

$$\begin{aligned} p \vee (q \wedge r) & \text{ por } (p \vee q) \wedge (p \vee r) \\ (p \wedge q) \vee r & \text{ por } (p \vee r) \wedge (q \vee r) \end{aligned}$$

Por exemplo, vamos determinar a FNC das seguintes proposições:

1. Dada a seguinte proposição: $\sim(((p \vee q) \wedge \sim q) \vee (q \wedge r))$

Demonstração:

$$\begin{aligned} \sim(((p \vee q) \wedge \sim q) \vee (q \wedge r)) & \iff \sim((p \vee q) \vee \sim\sim q) \wedge (\sim q \vee \sim r) \\ & \iff ((\sim p \wedge \sim q) \vee q) \wedge (\sim q \vee \sim r) \\ & \iff (\sim p \vee q) \wedge (\sim q \vee q) \wedge (\sim q \vee \sim r) \end{aligned}$$

Observe que a proposição resultante ainda poderia ser simplificada resultando em:

$$(\sim p \vee q) \wedge (\sim q \vee \sim r)$$

Desta forma, observamos que podemos ter mais de um resultado (FNC equivalente) para uma dada proposição.

2. Dada a seguinte proposição: $p \leftrightarrow q \vee \sim r$

Demonstração:

$$\begin{aligned} p \leftrightarrow q \vee \sim r & \iff (p \rightarrow (q \vee \sim r)) \wedge ((q \vee \sim r) \rightarrow p) \\ & \iff (\sim p \vee q \vee \sim r) \wedge (\sim(q \vee \sim r) \vee p) \\ & \iff (\sim p \vee q \vee \sim r) \wedge ((\sim q \wedge r) \vee p) \\ & \iff (\sim p \vee q \vee \sim r) \wedge (p \vee \sim q) \wedge (p \vee r) \end{aligned}$$

3. Dada a seguinte proposição: $(p \rightarrow q) \leftrightarrow (\sim q \rightarrow \sim p)$

Demonstração:

$$\begin{aligned} (p \rightarrow q) \leftrightarrow (\sim q \rightarrow \sim p) & \iff (\sim p \vee q) \leftrightarrow (\sim\sim q \vee \sim p) \\ & \iff (\sim p \vee q) \leftrightarrow (q \vee \sim p) \\ & \iff (\sim(\sim p \vee q) \vee (q \vee p)) \wedge ((\sim p \vee q) \vee \sim(q \vee \sim p)) \\ & \iff ((\sim\sim p \wedge \sim q) \vee (q \vee p)) \wedge ((\sim p \vee q) \vee (\sim q \vee \sim\sim p)) \\ & \iff ((p \wedge \sim q) \vee (q \vee p)) \wedge ((\sim p \vee q) \vee (\sim q \vee p)) \\ & \iff (p \vee q \vee \sim p) \wedge (\sim q \vee q \vee \sim p) \wedge (\sim p \vee q \vee \sim q) \wedge (\sim p \vee q \vee p) \end{aligned}$$

3.1.3 Forma Normal Disjuntiva

Definição: uma proposição encontra-se na Forma Normal Disjuntiva (FND) se e somente se ela satisfaz as seguintes condições:

- a) Contém no máximo os seguintes operadores: \sim , \wedge , \vee ;
- b) \sim não aparece repetido (por exemplo $\sim\sim p$) e não tem alcance sobre \wedge e \vee , ou seja, só incide sobre proposições simples (letras proposicionais)
- c) \wedge não tem alcance sobre \vee (por exemplo, não há proposições compostas do tipo $p \wedge (p \vee q)$)

Observa-se que a diferença entre uma FNC e uma FND é apenas na condição que tem como base o \vee e \wedge .

Por exemplo, as seguintes proposições se enquadram em uma FNC:

- $p \vee \sim q$
- $(p \wedge q) \vee \sim q$

Para transformar uma proposição qualquer em uma FND equivalente, aplicam-se as seguintes regras:

- (1) Eliminar os operadores \rightarrow e \leftrightarrow substituindo:

$$p \rightarrow q \text{ por } \sim p \vee q \text{ e}$$
$$p \leftrightarrow q \text{ por } (\sim p \vee q) \wedge (p \vee \sim q)$$

- (2) Eliminar negações repetidas e parênteses precedidos por \sim pelas regras de dupla negação e Morgan.

- (3) Substituir:

$$p \wedge (q \vee r) \text{ por } (p \wedge q) \vee (p \wedge r)$$
$$(p \vee q) \wedge r \text{ por } (p \wedge r) \vee (q \wedge r)$$

Por exemplo, vamos determinar a FND das seguintes proposições:

- 1. Dada a seguinte proposição: $(p \rightarrow q) \wedge (p \rightarrow \sim p)$

Demonstração:

$$(p \rightarrow q) \wedge (p \rightarrow \sim p) \iff ((\sim p \vee q) \wedge \sim q) \vee ((\sim p \vee q) \wedge p)$$
$$\iff (\sim p \vee \sim q) \vee (q \wedge \sim q) \vee (\sim p \wedge p) \vee (p \wedge q)$$

Da mesma forma que as FNC, aqui também podemos ter mais de uma FND equivalente. Por exemplo:

$$(\sim p \wedge \sim q) \vee (p \wedge q)$$

Ou seja, para uma dada proposição poderá haver mais de uma FND equivalente.

2. Dada a seguinte proposição: $\sim((p \vee q) \wedge \sim q) \vee (q \wedge r)$

Demonstração:

$$\begin{aligned}\sim((p \vee q) \wedge \sim q) \vee (q \wedge r) &\iff \sim((p \vee q) \wedge \sim q) \wedge \sim(q \wedge r) \\ &\iff (\sim(p \vee q) \vee \sim\sim q) \wedge (\sim q \vee \sim r) \\ &\iff ((\sim p \wedge \sim q) \vee q) \wedge (\sim q \vee \sim r) \\ &\iff (((\sim p \wedge \sim q) \vee q) \wedge \sim q) \vee (((\sim p \wedge \sim q) \vee q) \wedge \sim r) \\ &\iff (\sim p \wedge \sim q \wedge \sim q) \vee (q \wedge \sim q) \vee (\sim p \wedge \sim q \wedge \sim r) \vee (q \wedge \sim r) \\ &\iff (\sim p \wedge \sim q) \vee (\sim p \wedge \sim q \wedge \sim r) \vee (q \wedge \sim r)\end{aligned}$$

ATIVIDADES – UNIDADE 3

1. Mostrar que as seguintes proposições são tautológicas:

- a) $(p \rightarrow q) \vee (p \rightarrow \sim p)$
- b) $(p \rightarrow q) \wedge (p \rightarrow q)$
- c) $(p \rightarrow q) \wedge \sim q \rightarrow \sim p$
- d) $p \iff q \wedge (p \vee q)$
- e) $\sim(p \wedge \sim p) \vee (q \rightarrow \sim q)$
- f) $\sim(p \vee q) \rightarrow (p \iff q)$
- g) $(p \iff p \wedge \sim p) \iff \sim p$
- h) $p \vee (q \vee \sim p)$
- i) $(p \vee q) \wedge \sim p \rightarrow q$
- j) $\sim(p \vee \sim p) \vee (q \vee \sim q)$
- k) $p \vee (p \wedge q) \iff p$
- l) $(p \iff q) \wedge p \rightarrow q$

2. Demonstre, utilizando tabelas verdade, as seguintes relações de equivalência:

- a) $p \wedge (p \vee q) \iff p$
- b) $p \vee (p \wedge q) \iff p$
- c) $p \vee q \iff (p \vee q) \wedge \sim(p \wedge q)$

3. Simplifique as proposições abaixo utilizando as leis de equivalência:

- a) $p \vee (p \rightarrow q) \vee (p \rightarrow \sim q)$
- b) $(p \rightarrow q) \vee (\sim p \rightarrow q)$
- c) $(p \rightarrow q) \rightarrow r$
- d) $(p \vee q) \rightarrow (\sim r \rightarrow \sim q)$
- e) $p \rightarrow (p \wedge q)$
- f) $p \iff q$

4. Demonstre as relações abaixo utilizando as equivalências:

- a) $p \rightarrow q \wedge r \iff (p \rightarrow q) \vee (p \rightarrow r)$
- b) $p \rightarrow q \vee r \iff (p \rightarrow q) \wedge (p \rightarrow r)$
- c) $p \wedge (r \vee s \vee t) \iff (p \vee r) \vee (p \wedge s) \vee (p \wedge t)$
- d) $p \wedge q \rightarrow r \iff p \rightarrow (q \rightarrow r)$
- e) $\sim(\sim p \rightarrow \sim q) \iff \sim p \wedge q$

4

LÓGICA DE PREDICADOS

INTRODUÇÃO

A linguagem da lógica proposicional não é adequada para representar relações entre objetos. Por exemplo, se fôssemos usar uma linguagem proposicional para representar "João é pai de Maria e José é pai de João", usaríamos duas letras sentenciais diferentes para expressar ideias semelhantes (por exemplo, P para simbolizar "João é pai de Maria" e Q para simbolizar "José é pai de João") e não estaríamos captando com esta representação o fato de que as duas frases falam sobre a mesma relação de parentesco entre João e Maria e entre José e João.

Outro exemplo do limite do poder de expressão da linguagem proposicional é sua incapacidade de representar instâncias de uma propriedade geral. Por exemplo, se quiséssemos representar em linguagem proposicional "Qualquer objeto é igual a si mesmo" e "3 é igual a 3", usaríamos letras sentenciais distintas para representar cada uma das frases, sem captar que a segunda frase é uma instância particular da primeira.

Da mesma forma, se por algum processo de dedução chegássemos à conclusão que um indivíduo arbitrário de um universo tem uma certa propriedade, seria razoável querermos concluir que esta propriedade vale para qualquer indivíduo do universo. Porém, usando uma linguagem proposicional para expressar "um indivíduo arbitrário de um universo tem uma certa propriedade" e "esta propriedade vale para qualquer indivíduo do universo", usaríamos dois símbolos proposicionais distintos e não teríamos como concluir o segundo do primeiro.

Neste capítulo, veremos que nem sempre a lógica proposicional é suficiente e precisamos de algo mais poderoso para a representação de sentenças. Observa-se que a lógica de predicados é bem mais complexa e exige, do aluno um domínio básico de matemática discreta.

4.1

LÓGICA PROPOSICIONAL *VERSUS* LÓGICA DE PRIMEIRA ORDEM

O que não é possível expressar em Lógica Proposicional?

- » Todo tricolor é um campeão. Roberto é tricolor. Logo, Roberto é um campeão.
- » A adição de dois números ímpares quaisquer é um número par.
- » Acesso a esse recinto é permitido somente para as pessoas autorizadas ou conhecidas de pessoas autorizadas.
- » Quantificadores: todo, qualquer, existe, alguns, nenhum.
- » Objetos: indivíduos do universo de discurso, sobre o qual quantificadores podem ser aplicados.

4.2

SINTAXE

A linguagem de primeira ordem captura relações entre indivíduos de um mesmo universo de discurso e a lógica de primeira ordem vai permitir concluir particularizações de uma propriedade geral dos indivíduos de um universo de discurso. Para ter tal poder de expressão, a linguagem de primeira ordem vai usar vários símbolos mais sofisticado do que o da linguagem proposicional vista no capítulo anterior.

Considere a sentença:

"Todo objeto é igual a si mesmo"

Esta sentença fala de uma propriedade (a de ser igual a si mesmo) que vale para todos os indivíduos de um universo de discurso, sem identificar os objetos deste universo.

Considere agora a sentença:

"Existem números naturais que são pares"

Esta sentença fala de uma propriedade (a de ser par) que vale para alguns indivíduos do universo dos números naturais, sem, no entanto, falar no número "2" ou "4" ou "8" e assim por diante, em particular. Para expressar propriedades gerais (que valem para todos os indivíduos) ou existenciais (que valem para alguns indivíduos) de um universo, são utilizados os **quantificadores**:

- » \forall (universal)
- » \exists (existencial)

Estes quantificadores virão sempre seguidos de um símbolo de variável, captando, desta forma, a ideia de estarem simbolizando as palavras "para qualquer" e "para algum".

Considere as sentenças:

- a) "Sócrates é homem"
- b) "Todo aluno do curso de Licenciatura em Ciência da Computação estuda lógica"

A primeira frase fala de uma propriedade (ser homem) de um indivíduo distinguido ("Sócrates") de um domínio de discurso. A segunda frase fala sobre objetos distinguidos "Licenciatura em Ciência da Computação" e "lógica". Tais objetos poderão ser representados usando os símbolos, soc para "Sócrates", lcc para "Licenciatura em Ciência da Computação", lg para "lógica". Tais símbolos são chamados de símbolos de **constantes**.

As propriedades "ser aluno de" e "estuda" relacionam objetos do universo de discurso considerado, isto é, "ser aluno de" relaciona os indivíduos de uma universidade com os seus cursos, "estuda" relaciona os indivíduos de uma universidade com as disciplinas.

Para representar tais relações são usados símbolos de *predicados* (ou *relações*). Nos exemplos citados, podemos usar Estuda e Aluno, que são símbolos de relação binária. As relações unárias expressam propriedades dos indivíduos do universo (por exemplo "ser par", "ser homem"). A relação "ser igual a" é tratada de forma especial, sendo representada pelo símbolo de igualdade.

Desta forma podemos simbolizar as sentenças consideradas nos exemplos da seguinte forma:

- » "Todo mundo é igual a si mesmo" por $\forall x x \approx x$;
- » "Existem números naturais que são pares" por $\exists x \text{Par}(x)$;
- » "Sócrates é homem" por $\text{Homem}(\text{soc})$;
- » "Todo aluno de Licenciatura em Ciência da Computação estuda lógica" por $\forall x(\text{Aluno}(x, \text{lcc}) \rightarrow \text{Estuda}(x, \text{lg}))$.

Observamos que podemos representar objetos do domínio por meio de constantes. No entanto, uma outra maneira de representá-los é por meio do uso de símbolos de função.

Por exemplo, podemos representar os números naturais "1", "2", "3", e assim por diante, por meio do uso de símbolo de função, digamos, *suc*, que vai gerar nomes para os números naturais "1", "2", "3", etc. a partir da constante 0, por exemplo, "1" vai ser denotado por $\text{suc}(0)$, "3" vai ser denotado por $\text{suc}(\text{suc}(\text{suc}(0)))$. Sequências de símbolos tais como $\text{suc}(0)$ e $\text{suc}(\text{suc}(\text{suc}(0)))$ são chamadas *termos*.

Desta forma, considerando a sentença:

"Todo número natural diferente de zero é sucessor de um número natural"

podemos simbolizá-la por:

$$\forall x(\neg x \approx 0 \rightarrow \exists y \text{suc}(y) \approx x)$$

4.2.1 Definições

Os símbolos de uma função e de um predicado tem *aridades* que dependem de quantos são seus argumentos. Uma maneira simples de indicar a aridade de um símbolo de função *f* ou de um símbolo de predicado *P* é usar um símbolo de tipo *s* e indicar que *f* é *n*-ária por:

$$f: s^n \rightarrow s$$

$$P \text{ é } n\text{-ário por } P: s^n$$

Observa-se que um elemento distinguido *u* de um universo pode ser considerado como uma função de aridade zero; assim, os símbolos de constante podem ser declarados, usando a notação acima, como:

$$c: s^0 \rightarrow s$$

Desta forma, temos:

$$\text{suc}: s^1 \rightarrow s \text{ ou mais simplesmente, } \text{suc}: s \rightarrow s, \text{ Par}: s, \text{ Aluno}: s^2 \text{ e } 0: s^0 \rightarrow s.$$

Os símbolos de constantes, funções e predicados são chamados de **símbolos não lógicos** de uma linguagem de primeira ordem. Os símbolos não lógicos de uma linguagem de primeira ordem, junto com suas respectivas aridades são chamados de **assinatura**.

A seguir, veremos as definições dos conceitos discutidos acima.

Definição Sintática 1:

Uma **assinatura** de uma linguagem de primeira ordem consiste de um conjunto de sequências cujos elementos são da forma:

f: $s^n \rightarrow s$, com $n > 0$ e/ou da forma P: s^n , com $n > 0$

Dada uma assinatura ρ a linguagem de primeira ordem com assinatura ρ , $L(\rho)$, tem como alfabeto:

a) Conectivos proposicionais: \neg (negação), \wedge (&) conjunção, \vee disjunção, \leftrightarrow dupla implicação e \rightarrow (\supset) implicação.

b) Quantificadores: \forall quantificador universal e \exists quantificador existencial

c) Símbolo de igualdade (opcional): \approx

d) Símbolos de variáveis: VAR

e) Símbolos de predicados de ρ : o conjunto consistindo do primeiro elemento de cada uma das sequências da forma P: s^n que ocorrem em ρ . Para cada n , se P: s^n ocorre em ρ , P é chamado de símbolo de predicado n -ário.

f) Símbolos de função de ρ : o conjunto consistindo do primeiro elemento de cada uma das sequências da forma f: $s^n \rightarrow s$ que ocorrem em ρ . Para cada n , se f: $s^n \rightarrow s$ ocorre em ρ , f é chamado de símbolo de predicado n -ário. Os símbolos de função 0-ária são chamados de símbolos de constantes.

g) Símbolos de pontuação: "(", ")", "[", "]", "{", "}".

É importante observar que, por convenção, usaremos letras maiúsculas ou palavras da língua portuguesa inicializadas com letras maiúsculas para representar símbolos de predicados; e usaremos letras minúsculas ou palavras da língua portuguesa inicializadas com letras minúsculas para representar símbolos de função e de variáveis.

Como foi mencionado anteriormente, os símbolos de predicados pretendem representar relações entre objetos de um mesmo universo de discurso. A definição de conjunto de fórmulas bem formadas de uma linguagem de primeira ordem $L(\rho)$ faz uso do conceito de termos. Os termos pretendem nomear os objetos do universo de um discurso.

Definição Sintática 2:

Dada uma assinatura ρ , consideremos o conjunto U das sequências finitas formadas pelos símbolos de VAR, símbolos de pontuação e os símbolos de função de ρ . Para cada símbolo de função n -ário ($n > 0$) f , então:

$F_f : U^n \rightarrow U$ por $F_f(e_1, \dots, e_n) = f(e_1, \dots, e_n)$.

Definição Sintática 3 (*termos de uma linguagem de primeira ordem*)

Dada uma assinatura ρ , o conjunto de termos de $L(\rho)$, $TERM(\rho)$, é o conjunto gerado a partir de VAR pelas funções geradoras $F = \{F_f / f \text{ símbolo de função } n\text{-ária de } \rho, n > 0\}$, i.e. o conjunto $TERM(\rho)$ de termos é o menor conjunto de U tal que:

- » Todo símbolo de variável pertence a $TERM(\rho)$
- » Todo símbolo de constante pertence a $TERM(\rho)$
- » Se t_1, t_2, \dots, t_n estão em $TERM(\rho)$ e f é um símbolo de função n -ário de ρ , então $f(t_1, t_2, \dots, t_n)$ está em $TERM(\rho)$.

Como $TERM(\rho)$ é um conjunto indutivo, vale o princípio de indução em $TERM(\rho)$, sendo:

Princípio de indução para termos: seja A uma propriedade que diz respeito a termos. Se cada símbolo de variável e cada símbolo de constante tem a propriedade A e se t_1, t_2, \dots, t_n tem a propriedade A implica que $f(t_1, t_2, \dots, t_n)$ tem a propriedade A, para $t_1, t_2, \dots, t_n \in TERM(\rho)$ e f símbolo de função n -ária, então todos os termos em $TERM(\rho)$ têm a propriedade A.

Dadas as definições até agora, podemos considerar o seguinte exemplo:

Vamos mostrar que todo termo tem o mesmo número de abre parênteses "(" e de fecha parênteses ")" usando o princípio da indução em $TERM$.

Considerando a notação $na(\alpha)$ e $nf(\alpha)$ para o número de abre e fecha parênteses de uma expressão α ,² respectivamente.

Seja $S = \{\alpha \in TERM / na(\alpha) = nf(\alpha)\}$

Vamos mostrar que S é um conjunto indutivo, i.e.:

- a) $VAR \subseteq S$, sendo $na(x) = 0 = nf(x)$, para $x \in VAR$.
- b) Se c é símbolo de constante $na(c) = 0 = nf(c)$. Logo $c \in S$
- c) Se $t_1, t_2, \dots, t_n \in S$ e f é símbolo de função n -ário, então $na(f(t_1, t_2, \dots, t_n)) = na(t_1) + \dots + na(t_n)$

Por hipótese de indução, temos que: $na(t_1) = nf(t_1) \dots na(t_n) = nf(t_n)$

Logo: $na(f(t_1, t_2, \dots, t_n)) = na(t_1) + \dots + na(t_n) = nf(t_1) + \dots + nf(t_n) = nf(f(t_1, t_2, \dots, t_n))$.

Portanto: $f(t_1, t_2, \dots, t_n) \in S$.

Definição Sintática 4: (*fórmula atômica*)

Dada uma assinatura ρ , o conjunto das fórmulas atômicas de $L(\rho)$, $FORMAT(\rho)$, é o conjunto das expressões da forma:

- » $t_1 \approx t_2$, para t_1 e $t_2 \in TERM(\rho)$ (se $L(\rho)^2$ tem \approx);
- » $P(t_1, t_2, \dots, t_n)$, onde P é símbolo de predicado n -ário de ρ e $t_1, t_2, \dots, t_n \in TERM(\rho)$.

As fórmulas atômicas expressam as mais simples relações sobre os objetos do universo de discurso.

Definição Sintática 5:

Dada uma assinatura ρ , seja W o conjunto das sequências finitas de símbolos do alfabeto de $L(\rho)$ e de $TERM(\rho)$. Para $x \in VAR$, defina $f_{\forall x}: W \rightarrow W$ por $f_{\forall x}(\varphi) = (\forall x \varphi)$ e $f_{\exists x}: W \rightarrow W$ por $f_{\exists x}(\varphi) = (\exists x \varphi)$

Definição Sintática 6 (fórmulas de uma linguagem de primeira ordem)

Dada um assinatura ρ , o conjunto das fórmulas de $L(\rho)$, $FORM(\rho)$, é o conjunto gerado a partir de $FORMAT(\rho)$ pela aplicação dos geradores:

$f_{\vee}, f_{\wedge}, f_{\rightarrow}, f_{\leftrightarrow}, f_{\neg},$ e $f_{\forall x}, f_{\exists x}$, para cada $x \in VAR$

Ou seja, $FORM(\rho)$ é o menor conjunto de expressões de W tal que:

- Se t_1 e $t_2 \in TERM(\rho)$, então $t_1 \approx t_2 \in FORM(\rho)$;
- $P(t_1, t_2, \dots, t_n) \in FORM(\rho)$, onde P é símbolo de predicado n -ário P e $t_1, t_2, \dots, t_n \in TERM(\rho)$;
- Se $\alpha \in FORM(\rho)$, então $(\neg \alpha) \in FORM(\rho)$;
- Se α e $\beta \in FORM(\rho)$, então $(\alpha \vee \beta), (\alpha \wedge \beta), (\alpha \rightarrow \beta)$ e $(\alpha \leftrightarrow \beta) \in FORM(\rho)$;
- Se $\alpha \in FORM(\rho)$ e $x \in VAR$, então $(\forall x \alpha)$ e $(\exists x \alpha) \in FORM(\rho)$.

Dada uma assinatura ρ , a linguagem de primeira ordem com assinatura $\rho, L(\rho)$, é o conjunto das fórmulas bem formadas, $FORM(\rho)$. Em geral uma linguagem de primeira ordem $L(\rho)$ é apresentada por sua assinatura e pela indicação se nela ocorre ou não o símbolo de igualdade.

Para cada fórmula existe uma árvore cuja raiz é a fórmula e as folhas são as fórmulas atômicas que ocorrem na fórmula. Em cada nível, os nós da árvore desse nível são as subfórmulas que formam a fórmula do nível anterior pela aplicação de uma das regras (1-5) da definição de FORM.

Como FORM é um conjunto indutivo, vale o princípio da indução em FORM, a saber:

Princípio de indução para fórmulas: seja A uma propriedade que diz respeito a fórmulas, sendo:

- » se cada fórmula atômica tem a propriedade A e
- » se α_1 e α_2 têm a propriedade A , então $(\alpha_1 * \alpha_2)$ tem a propriedade A , para $*$ $\in \{\vee, \wedge, \leftrightarrow, \rightarrow\}$ e
- » se α tem a propriedade A , então $(\neg \alpha)$ tem a propriedade A e
- » se α tem a propriedade A , então $(\forall x \alpha)$ e $(\exists x \alpha)$ têm a propriedade A .

Desta forma, podemos concluir que todas as fórmulas em FORM têm a propriedade A .

Usamos a seguinte convenção para uso dos parênteses:

- » as mesmas convenções usadas para os conectivos sentenciais;
- » os quantificadores $\forall x$ e $\exists x$ ligam mais que os outros conectivos;
- » omitir os parênteses mais externos das fórmulas.

As definições a seguir apresentam alguns conceitos de manipulação sintática de termos e de fórmulas.

Considere os termos $f(x)$, $f(c)$ e $f(f(x))$. Note que $f(c)$ pode ser visto como o termo $f(x)$ quando escrevemos c no lugar de x , assim como $f(f(x))$ pode ser considerado como uma reescrita do termo $f(x)$ quando escrevemos $f(x)$ no lugar de x . Em termos intuitivos, $f(x)$ denota um objeto do universo de discurso que depende do objeto denotado por x . Esta mesma relação de dependência vai ser observada entre o objeto denotado por $f(c)$ e o denotado por c , assim como entre os objetos denotados por $f(f(x))$ e por $f(x)$.

Considere agora os termos $f(x,y)$, $g(x)$ e $f(x,g(x))$. Note que $f(x,g(x))$ pode ser visto como o termo $f(x,y)$ quando escrevemos $g(x)$ no lugar de y . Assim temos o conceito de aplicar substituição de variáveis por termos em termos.

Definição sintática 7 (*substituição simples*)

Para uma variável $x \in \text{VAR}$ e um termo $t \in \text{TEM}(\rho)$, a expressão x/t é chamada de *substituição simples* (de x por t).

Agora considere os termos $f(x,y)$, $g(x)$ e $f(g(x),x)$. Note que $f(g(x),x)$ pode ser visto como o termo $f(x,y)$ quando substituimos simultaneamente $g(x)$ por x e x por y . Este é um exemplo de aplicação simultânea de duas aplicações simples, sendo, $x:=g(x)$ e $y:=x$.

Definição Sintática 8 (*substituição*)

Uma substituição é um conjunto finito de substituições simples:

$\theta = \{x_1 := t_1, x_2 := t_2, \dots, x_n := t_n\}$, tal que $x_i \neq x_j$ para $i \neq j$

Uma substituição $\theta = \{x_1 := y_1, x_2 := y_2, \dots, x_n := y_n\}$ é chamada de **renomeação de variáveis**.

A aplicação da substituição $\theta = \{x_1 := t_1, \dots, x_n := t_n\}$ a um termo τ , denotada por $\tau\theta$, é o termo resultante da aplicação simultânea das substituições $x_i := t_i$ de θ em τ .

Mais formalmente, temos uma definição recursiva da aplicação da substituição de variáveis em termos, i.e. vamos definir $\tau\theta$ recursivamente.

Definição Sintática 9 (*substituição de variáveis em termos*)

Dada uma substituição θ , vamos definir o termo $\tau\theta$ para cada termo τ

1. para $\tau = x$

$\tau\theta = t$, se $x := t \in \theta = x$, caso contrário

2. para $\tau = c$

$\tau\theta = c$

3. para $\tau = f(t_1, \dots, x_n := t_n)$

$\tau\theta = f(t_1\theta, \dots, t_n\theta)$.

Agora considere os termos $f(x,y)$, $g(x)$ e $f(g(y),y)$. Note que $f(g(y),y)$ pode ser visto como o termo $f(x,y)$ quando substituimos x por $g(y)$, que, por sua vez, é o resultado da substituição de x por y em $g(x)$. Este é um exemplo de aplicação de duas substituições, uma após a outra (substituição de x por y em $g(x)$ e de x pelo termo resultante em $f(x,y)$).

Definição Sintática 10 (*composição de substituições*)

Sejam $\theta = \{x_1 := t_1, x_2 := t_2, \dots, x_n := t_n\}$ e $\tau = \{v_1 := \tau_1, v_2 := \tau_2, \dots, v_m := \tau_m\}$ substituições. A *composição de substituições* θ e τ é a substituição $\theta \circ \tau = \{x_1 := t_1\tau, x_2 := t_2\tau, \dots, x_n := t_n\tau\} \cup \{v_i := \tau_i \in \tau\}$

v_i não ocorre em nenhuma substituição de θ . Ou seja, $\theta^\circ \tau$ é a substituição obtida aplicando-se τ aos segundos termos das substituições simples de θ e acrescentando-se o conjunto de substituições simples de τ cujos primeiros elementos não são primeiros elementos de θ .

Definição Sintática 11 (*ocorrência de variáveis livres e ligadas*)

Uma ocorrência da variável v na fórmula F é *ligada* quando está dentro do escopo de algum quantificador:

$$Q v (\forall v \text{ ou } \exists v)$$

Observa-se que uma ocorrência da variável v combinada a um quantificador (da forma $Q v$) costuma ser chamada de ligada ou então recebe um nome especial, como ligante.

As ocorrências *livres* da variável v na fórmula F são as que não são ligadas nem da forma:

$$Q v (\forall v \text{ ou } \exists v)$$

Por exemplo, considere a fórmula $P(v) \wedge \forall v [R(w,v) \rightarrow S(v,w)]$. Observamos que as duas ocorrências da variável w (em $R(w,v)$ e em $S(v,w)$) são livres; e a variável u têm ocorrências livres (em $P(v)$) e ligadas (em $R(u,v)$ e em $S(v,u)$).

Definição Sintática 12 (*variáveis livres e ligadas*)

As variáveis livres na fórmula F são as que têm ocorrências livres em F . O conjunto de tais variáveis será denotado por $VL[F]$.

As *variáveis livres em um conjunto* Γ de fórmulas são as livres em alguma fórmula F de Γ , formando o conjunto $VL[\Gamma] = \cup_{F \in \Gamma} VL[F]$.

Uma *sentença* é uma fórmula F sem ocorrências livres de variáveis, ou seja, $VL[F] = \emptyset$.

4.3

SEMÂNTICA

A semântica fornece significado para as fórmulas de uma linguagem de primeira ordem. Desta forma, dada uma linguagem de primeira ordem podemos traduzir os objetos formais em relações entre indivíduos de um universo de discurso e em descrições de propriedades deste universo. É preciso notar que nem sempre a tradução de uma fórmula vai descrever propriedades que são verdadeiras no universo de discurso escolhido.

Por exemplo, considere a fórmula:

$$\exists x(\neg x \approx c)$$

onde o significado intuitivo é: existe um elemento diferente do objeto distinguido nomeado por c . Observa-se que esta assertiva é verdadeira num universo com pelo menos dois elementos, mas não é verdade num universo cujo único elemento é o indivíduo denotado por c .

Assim, para se dar um significado para uma linguagem de primeira ordem, temos de fornecer o universo de discurso, os indivíduos distinguidos deste universo que são denotados pelas constantes da linguagem, as funções e as relações definidas neste universo de discurso que interpretam os símbolos de funções e predicados, respectivamente, da linguagem.

Definição Semântica 1

Dada um assinatura ρ , uma estrutura A para uma linguagem de primeira ordem $L(\rho)$ consiste de:

- a) um conjunto não vazio, A , chamado de universo de A ;
- b) uma função n -ária, $f^A: A^n \rightarrow A$, para cada símbolo de função n -ária f em $L(\rho)$;
- c) uma relação n -ária, $P^A \subseteq A^n$, para cada símbolo de predicado n -ário P em $L(\rho)$.

Assim, símbolos de relações serão interpretados como relações no universo da estrutura e os símbolos de função serão interpretados como funções no universo da estrutura, respeitando-se a aridade dos símbolos de predicados e de funções. Observa-se que a interpretação de um símbolo de constante c é um elemento distinguido de A , c^A .

Uma estrutura A para uma linguagem de primeira ordem $L(\rho)$ pode ser apresentada como uma n -upla:

$$A = \langle A, F, P \rangle,$$

onde F é o conjunto das funções n -árias, $f^A: A^n \rightarrow A$, onde f é símbolo de função n -ária em $L(\rho)$ e P é o conjunto das relações n -ária $P^A \subseteq A^n$, onde P é símbolo de predicado n -ário em $L(\rho)$. No caso de F ou P serem conjuntos unitários, não será usada a notação de conjuntos, o conjunto unitário com seu único elemento.

Considere a estrutura $A = \langle \mathbb{N}, \{c^A, d^A, f^A\}, P^A \rangle$, com $c^A = 0$, $d^A = 2$, P^A como a relação $<$ (menor que) entre números naturais e $f^A(n) = 2n$. É razoável interpretarmos as fórmulas:

- » $P(c, d)$ como $0 < 2$ (que é verdade),
- » a fórmula $P(d, c)$ como $2 < 0$ (que é falso) e
- » $f(c) \approx d$ como $0 = 2$ (que é falso).

Mas como seria interpretada a fórmula $P(x, d)$ nesta estrutura?

À primeira vista, tentaríamos como resposta $x > 2$. Mas isto não nos dá uma assertiva verdadeira nem falsa. Parece razoável atribuir-se um valor a x e aí verificarmos se com esta atribuição a x , a tradução da fórmula $P(x, d)$ é uma assertiva verdadeira ou falsa sobre os números naturais. O mesmo tipo de questão aparece ao considerarmos a fórmula $P(f(x), d)$.

Assim, se associarmos o valor 2 a x , a fórmula $P(x, d)$ é interpretada como $2 > 2$ (que é falso), se dermos o valor 3 a x , a fórmula $P(x, d)$ é interpretada como $3 > 2$ (que é verdadeiro). No caso da fórmula $P(f(x), d)$, ao associarmos o valor 0 a x , parece claro que sua interpretação na estrutura dada é $0 < 2$, já que $2 * 0 = 0$. Como vemos o significado de uma fórmula (verdade ou falso) numa estrutura depende do valor que seus termos denotam dentro do domínio da estrutura.

Definição Semântica 2 – Atribuição de valores a termos

Seja A uma estrutura com domínio A para uma linguagem de primeira ordem $L(\rho)$.

Seja $v : \text{VAR} \rightarrow A$, uma atribuição de valores às variáveis em A .

A extensão $\upsilon : \text{TERM}(\rho) \rightarrow A$ de v é uma atribuição de valores aos termos em A , definida recursivamente da seguinte forma:

- $\upsilon(x) = v(x)$, para $x \in \text{VAR}$;
- $\upsilon(c) = c^A$ para cada símbolo de constante c de $L(\rho)$;
- $(f(t_1, t_2, \dots, t_n)) = f^A(\upsilon(t_1), \upsilon(t_2), \dots, \upsilon(t_n))$, para cada símbolo de função n -ário f de $L(\rho)$ e $t_1, t_2, \dots, t_n \in \text{TERM}(\rho)$.

Note que pelo fato de $\text{TERM}(\rho)$ ser livremente gerado a partir de VAR , o teorema da recursão (não visto nesse livro) garante que υ existe e é a única extensão de v com as propriedades descritas acima. Por simplicidade, podemos usar v em substituição da extensão υ de v .

Exemplificando:

Para ilustrar o conceito de estrutura e atribuição de valores a termos, considere uma linguagem de primeira ordem $L(\rho)$ com símbolos $f : s \rightarrow s$ de função unária e $c : s^0 \rightarrow s$, de constante.

Seja A a estrutura para esta linguagem tendo como universo \mathbb{N} , o conjunto dos naturais, $f^A : \mathbb{N} \rightarrow \mathbb{N}$ definida por $f^A(n) = 2n$ e $c^A = 3$. Tome $v : \text{VAR} \rightarrow \mathbb{N}$ tal que $v(x) = 2$. Assim, temos que:

- » o termo $f(x)$ é interpretado nesta estrutura com a atribuição v como o número 4;
- » o termo $f(f(x))$ é interpretado nesta estrutura com a atribuição v como o número 8;
- » o termo $f(c)$ é interpretado nesta estrutura com a atribuição v como o número 6.

Observe que os termos $f(x)$ e $f(f(x))$ podem ter outras interpretações na mesma estrutura, dependendo da atribuição de valores v , enquanto que $f(c)$ sempre aponta para o número 6, qualquer que seja a atribuição de valores v .

As fórmulas pretendem expressar relações entre elementos do universo ou propriedades do universo. A ideia intuitiva é que, por exemplo, a fórmula $P(x,c)$, numa estrutura com predicado binário P^A e constante c^A , vai ter um significado se a x estiver associado um valor no universo e, neste caso, seu significado pode ou não corresponder à realidade. Por exemplo, considere a estrutura A tendo como universo N , o conjunto dos naturais:

$$P^A = \{(n,m) / n > m\} \subseteq N^2 \text{ e } c^A = 0.$$

Nesta estrutura, com a atribuição v , tal que $v(x)=0$, $P(x,c)$ significa $0 > 0$, que é falso; com a atribuição v tal que $v(x)=3$, $P(x,c)$ significa $3 > 0$, que é verdade. Por outro lado, a interpretação de $(\exists x P(x,c))$ expressa que existe um número natural menor que 0, que é falso, qualquer que seja a atribuição dada a x . Esta diferença se dá pelo fato de na fórmula $P(x,c)$, o valor que se dá a x é relevante para o significado da fórmula, enquanto que em $(\exists x P(x,c))$ a variável x faz o papel de uma variável “dummy”, isto é, x aparece na fórmula só para constar.

Dada uma estrutura A com domínio A para uma linguagem de primeira ordem $L(\rho)$ e uma fórmula α , seja v uma atribuição de valores a VAR em A . Vamos definir a noção A satisfaz α com a atribuição v , que é uma relação entre A , v e fórmulas, notada por $A \models \alpha[v]$. A definição é por recursão, assim daremos a definição de forma direta para as fórmulas atômicas e para as demais será dada recursivamente.

Definição Semântica 3 – Fórmula logicamente válida $\models \alpha$

Uma fórmula α é logicamente válida, denotada por $\models \alpha$, se e somente se para cada estrutura A para a linguagem de α , A satisfaz α , ou $\sigma \models \alpha, A \models \alpha$, para toda estrutura A .

Exemplo 1:

$$P(c) \rightarrow \exists x P(x) \text{ é logicamente válida.}$$

Exemplo 2:

$$\forall x P(x) \rightarrow P(y) \text{ é logicamente válida.}$$

Toda fórmula que é obtida pela substituição sistemática das letras sentenciais de uma tautologia por fórmulas de lógica de primeira ordem é uma fórmula logicamente válida, por exemplo $P(x) \rightarrow P(x)$.

Definição Semântica 4 – Equivalência lógica: $\alpha \text{ eq } \beta$

Duas fórmulas α e β são logicamente equivalentes se e só se a fórmula $\alpha \leftrightarrow \beta$ é logicamente válida.

Exemplos:

1. $\forall x \alpha$ e $\forall x \forall y \alpha$ são logicamente equivalentes.
2. $\neg \forall x \gamma$ e $\exists x \neg \gamma$ são logicamente equivalentes.
3. $\neg \exists x \gamma$ e $\forall x \neg \gamma$ são logicamente equivalentes.
4. $\exists x(\gamma \vee \beta)$ e $\exists x \gamma \vee \beta$ são logicamente equivalentes se x não ocorre livre em β .
5. $\exists x(\beta \vee \gamma) \leftrightarrow \beta \vee \exists x \gamma$ são logicamente equivalentes se x não ocorre livre em β .
6. $\forall x(\gamma \wedge \beta)$ e $\forall x \gamma \wedge \beta$ são logicamente equivalentes se x não ocorre livre em β .
7. $\forall x(\beta \wedge \gamma)$ e $\beta \wedge \forall x \gamma$ são logicamente equivalentes se x não ocorre livre em β .
8. $\exists x(\gamma \wedge \beta)$ e $\exists x \gamma \wedge \beta$ são logicamente equivalentes se x não ocorre livre em β .
9. $\exists x(\beta \wedge \gamma)$ e $\beta \wedge \exists x \gamma$ são logicamente equivalentes se x não ocorre livre em β .

Definição Semântica 5 – Consequência lógica $\Gamma \models \alpha$

Dado um conjunto de fórmulas Γ e uma fórmula α , dizemos que α é consequência lógica de Γ ($\Gamma \models \alpha$) se e só se para toda estrutura A e toda atribuição v , se $A \models \Gamma[v]$, então $A \models \alpha[v]$.

Exemplos

$$\{\forall y \forall x P(x,y)\} \models \forall x \forall y P(x,y)$$
$$\{\forall x P(x)\} \models P(f(x))$$

Se Γ é o conjunto vazio, usamos a notação $\models \alpha$. Note que neste caso a definição coincide com a de α ser logicamente válida. Como no caso proposicional, usaremos a notação $\Gamma \not\models \alpha$ para indicar que α não é uma consequência de Γ .

Definição Semântica 6 – $\Gamma: A \models \Gamma$

Dada uma estrutura A e um conjunto de fórmulas Γ dizemos que A satisfaz Γ (A é um modelo de Γ), $A \models \Gamma$ se e só se $A \models \Gamma[v]$, para toda atribuição de valores v .

Observa-se que se Γ for um conjunto de sentenças, $A \models \Gamma$ e e só se A satisfaz α para toda sentença $\alpha \in \Gamma$.

Por exemplo, considere o conjunto de sentenças:

$$\Gamma = \{\forall x \forall y (f(x)=f(y) \rightarrow x=y), \forall x Q(f(x),x)\}$$
 e a estrutura $A = \langle \mathbb{N}, f^A, Q^A \rangle$

onde \mathbb{N} é o conjunto dos naturais, $f^A(n) = n+1$ e $Q^A = \{(n, m) \in \mathbb{N}^2 / n > m\}$. Temos $A \models \Gamma$.

A definição da semântica para uma linguagem de primeira ordem (satisfação de fórmulas por estruturas) é bastante técnica, pois dá o significado de um linguagem formal de maneira precisa. A seguir, uma aplicação menos formal da semântica, que é o uso da linguagem de primeira ordem para representação de conhecimento.

O que se pretende aqui é, dada uma(s) sentença(s) em português (ou qualquer outra linguagem natural) obter uma fórmula que a(s) represente(m) simbolicamente, no sentido de que ao interpretarmos os símbolos de volta pelas palavras que eles simbolizam tenhamos uma interpretação que satisfaça a fórmula. Não há uma única forma de representação.

Deve-se identificar uma assinatura, isto é, quais os símbolos de constantes, funções e predicados que serão usados, de forma que:

- » nomes próprios sejam simbolizados por símbolos de constantes;
- » relações e propriedades sejam simbolizados por símbolos de predicados;
- » “algum”, “alguém” e seus sinônimos são simbolizados por $\exists x$;
- » “nenhum”, “ninguém” (e seus sinônimos) são simbolizados por $\neg\exists x$;
- » “para todo”, “cada” e seus sinônimos são simbolizados por $\forall x$.

Exemplificando:

Para simbolizar "Maria vai à escola" temos a sentença Escola (Maria), escolhendo Maria como símbolo de constante e Escola como símbolo de predicado, com significado pretendido, x vai à escola para Escola (x).

Para simbolizar "Lógica é difícil" temos a sentença Difícil(lógica), escolhendo lógica como símbolo de constante e Difícil como símbolo de predicado, com significado pretendido x é difícil para Difícil (x).

Para simbolizar "A menor bola da loja de João é vermelha" temos: $\forall x(\text{Loja}(\text{joão},x) \wedge \text{Bola}(x) \wedge \forall y(\text{Loja}(\text{joão},y) \wedge \text{Bola}(y) \wedge x \neq y \rightarrow \text{Menor}(x,y)) \rightarrow \text{Vermelha}(x))$, escolhendo Loja e Menor como símbolos de predicados binários, Bola e Vermelha como símbolos de predicado unários e joão como símbolo de constante, com significados pretendidos:

- » Loja(u,v) - v é vendido na loja de u
- » Menor (u,v) – u é menor que v
- » Bola(v) – v é bola
- » Vermelha(v) – v é vermelha

Para simbolizar "o avô de uma pessoa é o pai da mãe ou do pai desta pessoa" temos $\forall x \forall y (\text{Avô}(x,y) \leftrightarrow x = \text{pai}(y) \vee x = \text{mãe}(y))$, escolhendo Avô como símbolo de predicado binário, pai e mãe como símbolos de função unários, com significados pretendidos:

- » Avô(v,u) – v é avô de u
- » pai(u) – o pai de u
- » mãe(u) – a mãe de u.

Se escolhêssemos Mãe e Pai como predicados binários com significados pretendidos Mãe(u,v) – u é mãe de v e Pai(u,v) – u é pai de v, teríamos: $\forall x \forall y \forall z (\text{Avô}(x,y) \leftrightarrow \text{Pai}(x,z) \wedge (\text{Pai}(z,y) \vee \text{Mãe}(z,y))$ simbolizando a mesma frase acima.

Para simbolizar "A derivada de f é definida" temos a fórmula atômica Definida(derivada(f)), escolhendo Definida como símbolo de predicado unário, derivada como símbolo de função unária, e f como símbolo de constante com os significados pretendidos óbvios. Alternativamente, podemos ter $\forall y (y = \text{derivada}(f) \rightarrow \text{Definida}(y))$.

Para simbolizar "Alguém mora na casa velha" temos $\exists x (\text{Mora}(x, \text{velha}))$, escolhendo Mora como símbolo de predicado binário e velha como símbolo de constante, com significados pretendidos:

- » Mora(u,v) – u mora em v
- » velha – a casa velha

Todo fantasma da região mora na casa velha pode ser simbolizado por: $\forall x(\text{Fantasma}(x) \rightarrow \text{Mora}(x, \text{velha}))$.

Existe um estudante que é amigo de Jorge mas não é amigo de Oscar pode ser simbolizado por:

$$\exists x(\text{Estudante}(x) \wedge \text{Amigo}(x, \text{jorge}) \wedge \neg \text{Amigo}(x, \text{oscar}))$$

Todo estudante que é amigo de Jorge é amigo de Oscar pode ser simbolizado por:

$$\forall x((\text{Estudante}(x) \wedge \text{Amigo}(x, \text{jorge})) \rightarrow \text{Amigo}(x, \text{oscar}))$$

ATIVIDADES – UNIDADE 4

1. Dados símbolos de predicados P e Q (unários), R e S (binários) e T (ternário), símbolos de operações f (unário), g (binário) e h (ternário); símbolos de constantes c e d, considere as fórmulas abaixo e responda:

- Identifique as ocorrências ligadas e livres de variáveis em cada fórmula.
- Quais fórmulas têm ocorrências ligadas e livres de uma mesma variável?

$$F1: \forall u [(\exists w R(u,g(v,w))) \rightarrow S(w,h(u,v,w))];$$

$$F2: \forall v [R(u,v) \rightarrow \forall u S(u,v)];$$

$$F3: \forall v [\exists u R(u,f(v)) \rightarrow S(w,g(u,c))];$$

$$F4: P(u) \rightarrow \forall v [S(u,v) \rightarrow \neg S(v,v)];$$

$$F5: P(u) \rightarrow \forall v [S(u,v) \rightarrow \neg S(v,v)];$$

$$F6: \forall v [(\exists u T(u,v,w) \rightarrow (S(u,w) \wedge Q(v)))];$$

$$F7: \exists u [P(u) \rightarrow \forall w (T(u,v,w) \rightarrow T(v,u,v))];$$

$$F8: \forall u T(g(v,u),v,h(f(u),c,w));$$

$$F9: \forall u [P(u) \rightarrow P(f(u))];$$

$$F10: P(c) \wedge \forall u [P(u) \rightarrow P(h(f(v),g(v,c),d))];$$

$$F11: P(c) \wedge \forall v [P(v) \rightarrow P(f(v))] \rightarrow P(u);$$

$$F12: \neg P(u) \rightarrow [\exists v R(u,v) \vee \exists w S(u,w)];$$

$$F13: \forall v \exists u R(h(u,c,d),f(v));$$

$$F14: \forall u \forall v [\exists w T(u,v,w) \times (R(u,v) \vee \neg S(v,u))];$$

$$F15: \forall v S(w,g(u,c));$$

$$F16: \forall u \forall v [\exists w T(u,v,w) \times (R(u,v) \vee \neg S(v,u))];$$

$$F17: \forall u \forall v [\exists w (P(f(u)) \wedge T(u,h(c,v,w),w) \wedge Q(g(d,w))) \rightarrow (R(f(u),v) \vee \neg S(f(v),u))].$$

2. Considere as fórmulas do exercício acima e responda quais dessas fórmulas são sentenças.

3. Simbolize as sentenças abaixo por sentenças de lógica de primeira ordem.

- Só os bravos sabem perdoar.
- Nenhum homem é uma ilha.
- Todo país tem o governo que merece.
- Não há nenhuma certeza, exceto a Lógica.
- Miséria gosta de companhia.
- Nem tudo que reluz é ouro.
- Se você agrada a todo mundo, você não agrada a ninguém.
- Existe algo de podre no reino da Dinamarca.
- Todo mundo ama alguém.
- Alguém ama todo mundo.
- Todo projeto em andamento tem um gerente.
- José é o gerente de todos os projetos.
- Existe um gerente de todos os projetos.

5

LÓGICAS NÃO CLÁSSICAS

INTRODUÇÃO

Como vimos anteriormente, a lógica clássica possui alguns paradigmas. Ela tem em comum os princípios de bivalência ou terceiro excluído, não-contradição e identidade. O princípio da bivalência ou terceiro excluído afirma que uma proposição tem apenas dois valores; possível e impossível, certa e errada, verdadeira e falsa. O princípio da não-contradição afirma que uma proposição não pode assumir dois valores distintos; ela é possível ou impossível, certa ou errada, verdadeira ou falsa. O princípio da identidade afirma que toda proposição é idêntica a si mesma.

Desta forma, podemos concluir que toda lógica que quebra um desses paradigmas é considerada uma lógica não clássica. Entretanto, se estudarmos mais a fundo, veremos que não é toda lógica não clássica que quebra esses paradigmas. Muitas vezes, a lógica não clássica também estende a clássica. A lógica clássica trabalha apenas com afirmações, como: “Todos homens são mortais”, “Sócrates é homem”. Para essas proposições, é possível dar um valor verdade. Observamos que a lógica clássica é incapaz de representar diferentes argumentos informais existentes, por exemplo “Eventualmente chove na cidade”. Considerando que a lógica é a ciência do pensamento, e tem como papel fundamental o entendimento de diferentes argumentos, seria necessário que ela conseguisse formalizar qualquer tipo de argumento.

Veremos nesse capítulo como funcionam diferentes lógicas não clássicas. Não iremos abordar cada uma em detalhes, pois elas compreendem uma fundamentação teórica complexa e extensa. A ideia desse capítulo é apresentar diferentes lógicas não clássicas e a importância de cada uma delas.

5.1

LÓGICA FUZZY

A lógica Fuzzy, também conhecida em português como lógica difusa ou lógica multivalorada, tem por objetivo modelar raciocínios aproximados e não precisos. Por exemplo:

O recipiente possui um pouco de água e por um tempo a água permanecerá no recipiente.

Nessa frase, existem dois termos (um pouco e um tempo) bastante subjetivos e difíceis de representar. Para um especialista que esteja conversando com outro especialista, o entendimento seria normal; no entanto, durante o processo de aquisição, fica bastante complicado para entender e representar esse tipo de conhecimento. Dessa forma, uma maneira de tentar solucionar o processo de representação de conhecimento impreciso é através da Lógica Fuzzy. Observa-se que a Lógica Fuzzy objetiva a modelagem computacional do raciocínio humano, impreciso, ambíguo e vago.

A teoria clássica de conjuntos permite o tratamento de classes de objetos e suas inter-relações em um universo definido. Nessa teoria, a pertinência de um dado elemento com relação a um conjunto refere-se ao fato de tal elemento pertencer ou não a esse conjunto. Diferente da Lógica Booleana, que admite apenas valores booleanos, ou seja, verdadeiro ou falso, a lógica Fuzzy, trata de valores que variam entre 0 e 1. Assim, um valor de 0.5 pode representar meia verdade; logo, 0.9 e 0.1 representam quase verdade e quase falso, respectivamente, ou seja, podemos modelar situações imprecisas. A lógica Fuzzy é capaz de representar informações vagas, em geral, descritas em linguagem natural e convertê-las para um formato numérico, de fácil manipulação.

Como discutido anteriormente, na Lógica clássica os conjuntos são bem definidos, de modo que um elemento pertence ou não a um conjunto; se pertencer, pertence somente a um. Isso evita que ambiguidades apareçam e tornam a lógica mais simples. Ainda considerando o exemplo da utilização de conjuntos para separar pessoas pela altura, uma pessoa com 1,69m seria considerada uma pessoa de altura mediana, se assim fosse definido, estando apenas nesse conjunto e em nenhum outro; já uma pessoa com 1,71m faria parte do conjunto das pessoas altas, e somente deste. Todavia, na realidade, fica bem difícil ver que pessoas com uma diferença de altura tão mínima pertencem a conjuntos diferentes. Por outro lado, pela ótica da Lógica Fuzzy, ter-se-ia as duas pessoas com certo grau de pertinência aos dois conjuntos, variando entre 0 e 1, ou seja, teríamos a tomada de decisão baseada em fatores mais humanos, mais maleáveis. Assim, pode-se concluir que os conjuntos Fuzzy que classificam os elementos de um dado universo são menos rígidos do que aqueles utilizados na teoria clássica, visto que eles admitem graus parciais de pertinência.

O primeiro passo na representação de conjuntos Fuzzy é a escolha da função de pertinência. A escolha dessa função depende do problema a ser modelado e também

da capacidade computacional disponível para processar o que se deseja. Funções não-lineares podem ser mais eficientes para problemas mais complicados; porém, elas demandam um poder computacional muito maior do que as funções lineares. Se o universo a ser trabalhado for curto, ou contínuo, torna-se bem mais simples a aplicação de uma função para separar adequadamente os elementos em conjuntos.

A lógica Fuzzy é muito utilizada na área de Inteligência Artificial, visto que ela é o principal formalismo de representação do conhecimento e, portanto, é muito útil no desenvolvimento de sistemas inteligentes, em especial os especialistas e os multiagentes. Segue abaixo uma lista não-exaustiva de domínios de aplicação da Lógica Fuzzy, no contexto da Inteligência Artificial:

- » Sistemas especialistas;
- » Sistemas multiagentes;
- » Reconhecimento de padrões;
- » Robótica;
- » Sistemas de controle inteligentes;
- » Sistemas de apoio à tomada de decisão;
- » Algoritmos genéticos;
- » Data mining.

5.2

LÓGICA MODAL

Lógica modal é o estudo do comportamento dedutivo de expressões que tratam de modos quanto ao tempo, possibilidade, probabilidade, entre outras. As mais usadas são a de possibilidade e necessidade, como por exemplo, “necessariamente” e “possivelmente”.

O fundador da lógica formal moderna, Gottlob Frege, duvidava que a lógica modal fosse viável, então, em 1933, Rudolf Carnap e Kurt Gödel começaram a busca por uma estrutura matemática de uma lógica que lidasse com as três modalidades clássicas (possibilidade, necessidade e probabilidade). Em 1937, Robert Feys, seguidor de Gödel, propôs o sistema T de lógica modal. Em 1951, Georg Henrik Von Wright propôs o sistema M, que é elaborado sobre o sistema T. Também nos anos 1950, C. I. Lewis construiu, sobre o sistema M, seus conhecidos sistemas modais S1, S2, S3, S4 e S5. Em 1965, Saul Kripke estabeleceu o sistema modal normal mínimo K.

A lógica modal define o termo "alética", que deriva da palavra grega "aleteia", significando verdade. Assim, falar de modalidades aléticas é falar dos modos como uma frase pode ser verdadeira ou falsa. Temos como exemplo de modalidades aléticas a necessidade e possibilidade.

As proposições na lógica modal podem ser classificadas como:

» Necessárias: proposições que necessariamente são verdadeiras ou falsas, ou seja, sua negação é impossível. Por exemplo: $1+1 = 2$

» Possíveis: proposições que podem levar a uma ocorrência, ou seja, ela não é necessariamente falsa. Por exemplo: Pode estar chovendo em Santa Maria agora.

» Contingentes: proposições que podem ser ou não verdades. Por exemplo: João era um filósofo.

» Impossíveis: proposições que marcam a impossibilidade de um acontecimento. Por exemplo: Uma pedra tem emoções.

» No próximo capítulo, iremos ver um pouco mais sobre lógica temporal, que teve como base a lógica modal.

5.3

LÓGICA PARACONSISTENTE

O aparecimento da lógica paraconsistente somente ocorreu em 1963, com um trabalho do lógico brasileiro Newton Carneiro Affonso da Costa. Da Costa já havia exposto suas ideias sobre o conceito da contradição anteriormente, mas foi só em 1963 que ele formulou, não um sistema, mas uma hierarquia enumerável de lógicas paraconsistentes de primeira ordem, dos respectivos cálculos de descrições e um esboço de teorias paraconsistentes de conjuntos construídos sobre sua lógica. Até 1976, o termo utilizado era lógica para sistemas formais inconsistentes. A partir dessa data, em um congresso, foi instituído o termo lógica paraconsistente. O lógico brasileiro Newton iniciou estudos no sentido de desenvolver sistemas lógicos que pudessem envolver contradições, motivado por questões de natureza tanto filosófica quanto matemáticas. Ele é conhecido internacionalmente como o criador das lógicas paraconsistentes.

As pesquisas em lógica paraconsistentes desenvolveram-se muito rapidamente, em parte como consequência dos trabalhos do brasileiro Da Costa e sua escola e, em parte, independentemente. Hoje, a lógica paraconsistente é um ramo bastante estudado no Brasil, na Austrália, na Polônia e nos Estados Unidos.

A lógica paraconsistente diverge da lógica clássica no sentido de que possam alicerçar sistemas teóricos que admitam contradições, expressões do tipo "A e não A" sem que se tornem triviais, ou seja, sem que todas as expressões bem formadas de sua linguagem possam ser provadas como teoremas do sistema. As lógicas paraconsistentes permitem que se possa aceitar a existência de teorias inconsistentes e a coexistência de sistemas lógicos incompatíveis entre si. Há duas classificações para esse tipo de lógica:

» Fraca, é quando pode servir de base tanto para teorias paraconsistentes quanto para teorias consistentes.

» Forte, geralmente, já existe uma fórmula tal que ela e sua negação são teoremas nessa lógica.

Uma teoria dedutiva T , cuja linguagem contenha um símbolo de negação (\sim), é dita inconsistente se o conjunto de seus teoremas contém ao menos dois deles, um dos quais é a negação do outro. Neste caso, sendo A e $\sim A$ tais teoremas, normalmente deriva-se em T uma contradição, isto é, uma expressão da forma $A \sim A$; caso isso aconteça, T é consistente.

A teoria T é trivial se o conjunto de suas fórmulas coincide com o de seus teoremas, ou seja, dito informalmente, se todos os enunciados sintaticamente corretos do ponto de vista da linguagem de T puderem ser provados em T . Uma lógica paraconsistente só pode ser utilizada como lógica subjacente a teorias inconsistentes, mas não triviais. Isso implica que o princípio da não contradição deve ser de alguma forma restringido, a fim de que possam parecer contradições, mas deve-se evitar que de duas premissas contraditórias se possa deduzir uma fórmula qualquer.

A classe das proposições é decomposta em proposições de dois tipos: na classe das bem comportadas (obediência à lógica clássica, para as lógicas bem comportadas A , tem-se que $\neg(A \rightarrow A)$ é verdadeiro), toda fórmula válida do cálculo clássico também o será nos cálculos de Sistemas Formais Inconsistentes, com exceção de um deles; se A for mal comportada, pode-se ter $A \rightarrow \neg A$. As lógicas paraconsistentes de certa forma estendem-se à lógica tradicional, permitindo certas investigações que não seriam possíveis à luz da lógica clássica. Elas não visam eliminar a lógica tradicional, que permanece válida em seu particular domínio de aplicabilidade.

ATIVIDADES – UNIDADE 5

1. Para cada uma das lógicas apresentadas nesse capítulo, responda:

- a) Porque ela é considerada não clássica?
- b) Quais são as suas principais características?

2. Pesquise exemplos de onde essas lógicas podem ser aplicadas na área de Ciência da Computação.

6

LÓGICA TEMPORAL

INTRODUÇÃO

O termo Lógica Temporal é usado para descrever um sistema de regras e simbolismos de representação do raciocínio, tendo a presença do tempo como elemento de destaque. O conceito de Lógica Temporal foi introduzido por volta dos anos de 1960 por Arthur Prior, com a designação de Lógica Tensa e consequentemente usada por lógicos e cientistas da computação. Atualmente, na área de Ciência da Computação, o estudo de verificação de sistemas é muito abordado. Tal estudo serve para demonstrar se um sistema computacional é livre de falhas ou não. A lógica temporal tem sido base para o estudo de verificação de sistemas.

Alguns sistemas lógicos são baseados na Lógica Temporal, como exemplo: Lógica computacional em árvore (CTL), Lógica para revestir tempo (LTL), Lógica Temporal de intervalos (ITL) e Lógica de tempo de ações (TLA). As mesmas incluem seu uso para formalizar o entendimento em assuntos filosóficos sobre tempo, definir as semânticas de expressões temporais em linguagem natural, definir uma linguagem para codificar conhecimento temporal em inteligência artificial e atuar como uma ferramenta no controle de aspectos temporais da execução de programas.

Neste capítulo, veremos a semântica de Kripke e uma descrição da lógica temporal e seus principais operadores.

6.1

ELEMENTOS BÁSICOS

Como visto nos capítulos anteriores, a lógica de primeira ordem tem limitação no sentido de não diferenciar entre o conceito de Possível e Necessário, pois de acordo com o formalismo clássico, fórmulas são apenas verdadeiras ou falsas, sem nenhum tipo de qualificação. A lógica modal permite representar o conceito de necessidade e possibilidade, ou seja, representa o estudo do comportamento das expressões de necessidade (é necessário que) e possibilidade (é possível que).

Estes conceitos são formalizados a partir da noção de Mundo Possível, cuja interpretação pode ser descrita como uma alternativa imaginável ao mundo real. A lógica modal surge dentro do contexto da Lógica Temporal, pois a linguagem desta contém, além dos operadores verdade funcional, os operadores da lógica modal, que são:

- » Necessidade: representada pelo símbolo de \Box
- » Possibilidade: representada pelo símbolo de \Diamond .

A sintaxe e a semântica da lógica modal é derivada da lógica de primeira ordem. A Lógica Temporal especifica suas funções a partir da semântica da modal agregando o fator tempo. Exemplificando os quantificadores da lógica modal – Necessidade \Box e Possibilidade \Diamond :

- » $\Diamond\varphi$: é possível que φ seja verdade;
- » $\Box\varphi$: é necessário que φ seja verdade;
- » $\Box\varphi \rightarrow \Diamond\varphi$: tudo que é necessário é possível;
- » $\varphi \rightarrow \Diamond\varphi$: se algo é verdadeiro, então é possível;
- » $\varphi \rightarrow \Box\Diamond\varphi$: algo que é verdadeiro é necessariamente possível;
- » $\Diamond\varphi \rightarrow \Box\Diamond\varphi$: tudo que é possível é necessariamente possível.

6.2

SEMÂNTICA DE KRIPKE

A semântica de Kripke é uma classe Kr de modelos de Kripke, onde o sistema K é o menor dos sistemas modais normais. Ou seja, ela é a interseção de todos os sistemas modais normais, justificado pelos seguintes princípios: trata-se de um sistema de lógica modal, com um conjunto de axiomas e regras de inferência que representam formalmente o raciocínio.

Seja P um conjunto de proposições atômicas, logo uma estrutura de Kripke é uma tupla (S, i, R, L) , onde:

S é um conjunto finito de estados;

» $i \in S$ é o estado inicial;

» $R \subseteq S \times S$ é uma relação de transição total: $\forall s \in S . \exists s' \in S \cdot (s, s') \in R$

» $L : S \rightarrow \mathcal{P}(P)$ é uma função que marca cada estado com o conjunto de fórmulas atômicas válidas nesse estado.

A semântica da Lógica Temporal pode ser definida sobre estruturas de Kripke. Desta forma, as técnicas de especificação e verificação de propriedades podem ser apresentadas independentemente do formalismo do modelo a ser adotado.

6.3

CONCEITOS BÁSICOS DA LÓGICA TEMPORAL

A lógica clássica trata de uma espécie de “verdade verdadeira”, no sentido em que a mesma não dispõe de nenhum mecanismo que represente o tempo e suas consequências sobre os valores verdade nas fórmulas lógicas.

A Lógica Temporal possui como base a lógica modal e permite representar a variação dos estados do mundo em diferentes períodos de tempo, ou seja, permite verificar a veracidade das sentenças ao longo do tempo, já que uma sentença pode ser verdadeira em determinado instante de tempo e falsa em um outro instante.

Na abordagem da Lógica Temporal, as propriedades a serem verificadas são dadas por meio de fórmulas de Kripke (abordagem dos mundos possíveis) que representam um conjunto de estados, de transições entre estados e funções que rotulam cada estado com o conjunto de propriedades verdadeiras dentro dele.

A principal característica da Lógica Temporal é o fato de uma determinada fórmula lógica poder representar valores verdade diferentes em instantes distintos de tempo. Essa característica é formalizada através da introdução na sintaxe da linguagem lógica de primeira ordem de diversos operadores temporais. Neste caso, tem-se o Método de Argumentos Temporais, onde:

- » A é uma fórmula qualquer,
- » P é um instante no passado,
- » F é um instante no futuro,
- » G todos os instantes no futuro e
- » H todos os instantes do passado.

Com base nesses operadores podemos definir métodos de argumentos temporais para a Fórmula A:

- » PA – $t < \text{now} \ \& \ A(t)$ – A foi verdade em algum instante do passado
- » FA – $\text{now} < t \ \& \ A(t)$ – A será verdade em algum instante do futuro
- » GA – $\forall t (A \text{ do } t < \text{now} \rightarrow (t))$ – A será verdade em todos os instantes do futuro
- » HA – $\forall t (A \text{ do } \text{now} < t \rightarrow (t))$ – A foi verdade em todos os instantes do passado

6.4

APLICAÇÕES EM LÓGICA TEMPORAL

As aplicações da Lógica Temporal em ciência da computação são apresentadas nos trabalhos de renomados pesquisadores com C. Caleiro, Michel Fisher, Chiara Ghidini e outros. No final da década de 1970, o estilo modal da Lógica Temporal encontrou a aplicação extensiva na área de informática. Sendo aplicada e relacionada à especificação e verificação dos programas, em especial naqueles com execução simultânea, isto é, o processamento é executado por dois ou mais processadores, que trabalham em paralelo, com o objetivo de assegurar o comportamento correto dos programas, sendo necessário especificar a maneira em que as ações dos vários processadores são relacionadas.

A Lógica Temporal pode ser também encontrada em aplicações de inteligência artificial, como a construção de agentes, sendo um desses tipos os agentes BDI (Belief, Desire and Intention), que possuem uma arquitetura de crenças, desejos e intenções que necessitam formalização para que tenham real utilidade. A formalização de tais termos pode ser realizada através de sua representação lógica. Geralmente, utiliza-se lógica modal para essas situações, mas nos anos 1990, o formalismo lógico baseado em Lógica Temporal passou a ser utilizado, já que permite a análise de como a crença do agente futuro pode afetar os desejos e intenções do presente.

Outra aplicação da Lógica Temporal é na engenharia de software, através do uso de técnicas de inteligência artificial, pois esta última utiliza Lógica Temporal na melhoria dos ambientes de construção e qualidade de software. A Lógica Temporal é utilizada nos componentes do sistema relacionados principalmente na construção dos programas de forma geral, no armazenamento e consulta aos dados. No nível mais próximo ao usuário, há a interface homem-máquina; no nível intermediário, o sistema de informação deve ter mecanismos de processamento de dados para a entrada, edição, análise, visualização e saída dos dados; e no nível mais interno do sistema, deve ter um sistema de gerência de bancos de dados.

Atualmente, as aplicações estão utilizando Lógica Temporal para resolver muitos problemas da engenharia, da medicina, de grandes bases de dados como as de sequência genética, entre outros. A Lógica Temporal apresenta-se como uma alternativa para o tratamento de dados e recuperação de informações históricas. Isto se deve à sua principal característica, ou seja, uma determinada fórmula lógica poder representar valores verdade diferentes em instantes distintos de tempo. A utilização da Lógica Temporal em sistemas de informações é de essencial importância, principalmente naqueles em que a evolução dos dados no decorrer do tempo é fator de confiabilidade e integridade.

ATIVIDADES – UNIDADE 6

1. Quais são os principais problemas computacionais que podem ser resolvidos utilizando a lógica temporal?

CONSIDERAÇÕES FINAIS

A Lógica Matemática é fundamental para o estudo da Ciência da Computação. Foi a lógica formal, como vimos nesse livro, que possibilitou e ainda possibilita o avanço da computação. No decorrer do curso de Licenciatura em Ciência da Computação, os conceitos vistos aqui serão aplicados e estendidos. Por exemplo, a programação de computadores é realizada pela definição de sequência de comandos lógicos. Esses comandos definem passo a passo o que um programa de computador pode fazer e o que ele não deve fazer. Os conceitos de lógica também são fundamentais para o estudo da Inteligência Artificial. Quando alguém fala em programas que apreendem, nada mais são do que conjuntos de comandos lógicos, onde as possíveis decisões são definidas pelos programadores com base em uma lógica matemática.

Os fundamentos mais básicos da lógica têm como base a lógica proposicional. Vimos que as proposições são importantes e podemos defini-las de forma matemática (usando símbolos e operações). Sendo assim, podemos expressar uma série de sentenças de forma não ambígua, possibilitando o processamento delas por um computador. A ambiguidade, por sua vez, é um conceito importante na lógica, pois o que buscamos é uma linguagem sem ambiguidades, ou seja, sem significados diferentes. E foi o entendimento da lógica proposicional que possibilitou o desenvolvimento dos primeiros computadores, os quais executavam apenas cálculos e comandos simples.

Os primeiros três capítulos do livro são considerados como a base da lógica proposicional. No entanto, não é possível descrever tudo com a lógica proposicional – precisamos de algo mais abrangente, como vimos no capítulo sobre lógica de predicados. Os dois últimos capítulos, sobre lógicas não clássicas, são ilustrados de forma resumida, pois demandariam, no mínimo, uma disciplina inteira para estudá-las de forma mais ampla. Lógicas não clássicas ainda são um importante tópico de pesquisa em Ciência da Computação.

A compreensão da lógica matemática ocorre por meio do entendimento dos conceitos básicos e da resolução de exercícios. Este livro demonstra todos os conceitos básicos importantes que veremos na disciplina. No entanto, a resolução dos exercícios será de fundamental importância para o aprendizado da lógica matemática.

REFERÊNCIAS

ALENCAR FILHO, E. **Iniciação à lógica matemática**. São Paulo: Nobel, 2002.

GERSTING, J. L. **Fundamentos Matemáticos para a Ciência da Computação**. 5. ed. Rio de Janeiro: LTC, 2004.

SOUZA, J. N. **Lógica para Ciência da Computação**. Rio de Janeiro: Campus, 2002.