INTRODUÇÃO À LÓGICA MATEMÁTICA

Matemática Aplicada a Computação

Professor Rossini A M Bezerra

- Lógica é o estudo dos princípios e métodos usados para distinguir sentenças verdadeiras de falsas.
- **Definição de Proposição**:
  - uma proposição é uma construção que se pode atribuir juízo, ou seja, que pode ser apenas verdadeira ou falsa.
- Exemplos
  - São Paulo é uma cidade grande = Verdade
  - Como vai você =?

- **Conectivos Lógicos**
  - As proposições podem ser simples (atômicas) ou compotas
  - Função:
    - Os conectivos têm a função de combinar sentenças simples para formar sentenças compostas.
- **Proposição Atômica**:
  - são proposições que não podem ser decompostas em proposições mais simples.
- **Proposição Composta**:
  - são proposições mais complexas, compostas por proposições mais simples através dos conectivos lógicos (ou operadores lógicos).
- Exemplos:
  - Animais são peludos **e** aves têm penas.

- Operadores:
  - Negação*:*
    - A negação de uma proposição é construída a partir da introdução da palavra ***não*** ou ***não é o caso que***.
  - Exemplos:
    - Brasil **não** é um país.
  - P denota uma proposição, então sua negação é denotada por:
    - ¬*P* ou ~ *P* (lê-se "*não P*")
    - Interpretamos a negação da seguinte forma: se P é verdadeira, então ¬P é falsa; se P é falsa, então ¬P é verdadeira.

- Para visualizar os valores lógicos de um conectivo utilizamos a tabela-verdade, que descreve as possíveis combinações dos valores lógicos das proposições.

Tabela Verdade Não	
P	~P
V	F
F	V

- *Conjunção*
  - Uma conjunção é verdadeira se ***ambos*** seus conjunctos são verdadeiros. Caso contrário, é falsa.
- É denotada por: $P \wedge Q$

Tabela Verdade de Conjunção		
P	Q	P^Q
V	V	V
V	F	F
F	V	F
F	F	F

- *Disjunção*
  - Uma disjunção é verdadeira se ***pelo menos um*** dos seus disjunctos for verdadeiro. Caso contrário, é falsa.
  - É denotada por: $P \vee Q$

Tabela Verdade de Disjunção		
P	Q	P∨Q
V	V	V
V	F	V
F	V	V
F	F	F

- *Condicional (Implicação)*
  - O condicional é falso se seu antecedente for verdadeiro e seu conseqüente for falso. Caso contrário, ele é verdadeiro.
  - É denotado por: $P \rightarrow Q$ (lê-se "*se P então Q*")
  - Podemos dizer que um enunciado da forma **P → Q** tem o ***mesmo significado*** (semântica) que um enunciado da forma **¬(P ∧ ¬Q)**, ou seja, ambos são verdadeiros sob as mesmas condições.
  - Portanto, podemos obter a tabela-verdade de P → Q construindo a tabela verdade de ¬(P ∧ ¬Q).

Tabela Verdade de Condição		
P	Q	P→Q
V	V	V
V	F	F
F	V	V
F	F	V

- Bicondicional
- Denotado por P↔Q , tem o mesmo significado que (P → Q) ∧ (Q → P).
- Assim, a tabela-verdade de (P ↔ Q) pode ser obtida construindo a tabela-verdade de (P → Q) ∧ (Q → P).

Tabela Verdade de Bicondicional		
P	Q	P→Q
V	V	V
V	F	F
F	V	F
F	F	V

- **Fórmulas Bem-Formadas** (well formed formula - wff)
  - São sentenças lógicas construídas corretamente sobre o alfabeto cujos símbolos são:
    - Conectivos,
    - Parênteses e
    - Letras sentenciais.
  - Exemplos:

$$\neg P,\ P \wedge Q,\ P \vee Q,\ P \rightarrow Q,\ P \leftrightarrow Q$$
$$P \vee \neg Q$$
$$(P \wedge \neg Q) \rightarrow R$$
$$\neg(P \wedge Q) \leftrightarrow (\neg P \vee \neg Q)$$

- **Tabelas-verdade para wffs (Fórmulas Bem-Formadas)**
  - Escrevemos as letras sentenciais à esquerda da tabela
  - Fórmula à direita da tabela.
  - Completaremos com todas as possibilidades de valores verdade para as letras sentenciais.
  - A seguir, devemos identificar o ***operador principal***, pois é ele que determina o valor-verdade para toda a fórmula.
  - Completamos a tabela com os valores-verdade para os operadores
    - sub-wffs e
    - por fim para a wff (operador principal).

- Construa a tabela-verdade para a fórmula ¬¬**P**.
  - Preenchemos a coluna letra sentencial P, completando-se os possíveis valores-verdade que P pode assumir;
  - Preenchemos a coluna da ocorrência de P na fórmula (na wff);
  - Preenchemos o sinal de negação imediatamente à esquerda de P;
  - Preenchemos o segundo sinal de negação, que é o operador principal
  - E, portanto, determina o valor-verdade da fórmula.

Tabela Verdade ¬¬P			
P	¬	¬	P
V	V	F	V
F	F	V	F

- **Equivalência**
- Dizemos que duas fórmulas P e Q são equivalentes se a fórmula $P \leftrightarrow Q$ é uma tautologia.
- Denotamos essa propriedade por $P \Leftrightarrow Q$

A **tautologia** (do grego) é , na retórica, um termo ou texto redundante, que repete a mesma idéia mais de uma vez. Como um vício de linguagem pode ser considerada um sinônimo de pleonasmo ou redundância

- Exemplos de algumas equivalências tautológicas importantes
  - 1 representa uma tautologia e
  - 0 representa uma contradição:

- Comutatividade:	$A \vee B \Leftrightarrow B \vee A$	$A \wedge B \Leftrightarrow B \wedge A$
- Associatividade:	$(A \vee B) \vee C \Leftrightarrow A \vee (B \vee C)$	$(A \wedge B) \wedge C \Leftrightarrow A \wedge (B \wedge C)$
- Distributividade:	$A \vee (B \wedge C) \Leftrightarrow (A \vee B) \wedge (A \vee C)$	$A \wedge (B \vee C) \Leftrightarrow (A \wedge B) \vee (A \wedge C)$
- Elemento Neutro:	$A \vee 0 \Leftrightarrow A$	$A \wedge 1 \Leftrightarrow A$
- Complementares:	$A \vee \neg A \Leftrightarrow 1$	$A \wedge \neg A \Leftrightarrow O$
- DeMorgan:	$\neg(A \vee B) \Leftrightarrow \neg A \wedge \neg B$	$\neg(A \wedge B) \Leftrightarrow \neg A \vee \neg B$

- **Aplicações na Computação**
- Os conectivos lógicos E (AND), OU (OR) e NÃO (NOT), respectivamente $\wedge$, $\vee$ e $\neg$, estão disponíveis em muitas linguagens de programação.
- Eles agem sobre combinações e expressões verdadeiras e falsas para produzir um valor lógico final.
- Tais valores lógicos permitem a decisão do fluxo de controle em programas de computador.
- Assim, em uma ramificação condicional de um programa,
  - se o valor lógico da expressão condicional for verdadeiro,
    - o programa executará um trecho do seu código;
  - se o valor lógico da expressão condicional for falso,
    - execurá outro trecho do seu código.
- Se a expressão condicional for substituída por outra expressão equivalente mais simples,
  - o valor lógico não será afetado,
  - assim como o fluxo de controle do programa,
  - mas o novo código será mais fácil de ser entendido
  - e poderá ser executado mais rapidamente.

- Exemplo:

```
if ((x < y) and not ((x < y) and (z < 1000)))
  do AlgumaCoisa;
else
  do OutraCoisa;
```

- Nesse exemplo, a expressão condicional tem a forma $A \wedge \neg(A \wedge B)$, onde A é "x < y" e B é "z <1000".
- Podemos simplificar essa expressão utilizando as equivalências vistas anteriormente.

$$
\begin{array}{ll}
A \wedge \neg(A \wedge B) \Leftrightarrow & \\
A \wedge (\neg A \vee \neg B) \Leftrightarrow & \text{(DeMorgan)} \\
(A \wedge \neg A) \vee (A \wedge \neg B) \Leftrightarrow & \text{(Distributividade)} \\
0 \vee (A \wedge \neg B) \Leftrightarrow & \text{(Complemetar)} \\
(A \wedge \neg B) \vee 0 \Leftrightarrow & \text{(Comutatividade)} \\
A \wedge \neg B & \text{(Elemento Neutro)}
\end{array}
$$

- Podemos então reecrever a proposição da seguinte forma:

```
if ((x < y) and not (z < 1000))
  do AlgumaCoisa;
else
  do OutraCoisa;
```

- **Quantificadores**
- Wffs formadas apenas pelos cinco operadores lógicos $(\neg, \wedge, \vee, \rightarrow, \leftrightarrow)$ têm possibilidade limitada de expressões.
- Por exemplo, não conseguiríamos simbolizar a sentença "Para todo x, x>0" como sendo uma proposição verdadeira sobre os inteiros positivos.
- Portanto novos conceitos, como o de quantificador, deve ser introduzido.
- Quantificadores são frases do tipo ***para todo***, ***para cada*** ou ***para algum***, isto é, frases que dizem "quantos objetos" apresentam determinada propriedade.

- **Quantificador Universal**:
  - é simbolizado por $\forall$ e lê-se *para todo*, *para qualquer* ou *para cada*.
- Assim, a sentença anterior pode ser simbolizada por: $(\forall x)(x > 0)$
- O valor lógico da expressão $(\forall x)(x>0)$ depende do domínio dos objetos sobre os quais estamos nos referindo, que chamamos de ***conjunto universo***.
- Qual seria o valor lógico da expressão
- $(\forall x)P(x)$ em cada uma das seguintes interpretações?
  - $P(x)$ é a propriedade que x é amarelo e o conjunto universo é o conjunto de todos os botões-de-ouro.
  - $P(x)$ é a propriedade que x é amarelo e o conjunto universo é o conjunto de todas as flores.
  - $P(x)$ é a propriedade que x é positivo ou negativo e o conjunto universo é conjunto de todos os inteiros.

- **Quantificador Existencial**:
  - é simbolizado por $\exists$ e lê-se *existe*, *existe algum*, *para pelo menos um*, *para algum*.
- Assim, a expressão $(\exists x)(x > 0)$
  - É lida como: "existe um x tal que x é maior do que zero".

- A expressão $(\forall x)(\exists y)Q(x, y)$ é lida como:
  - "para todo x existe um y tal que Q(x, y)".
  - Considerando que o conjunto universo é conjuntos dos números inteiros
  - e que Q(x, y) é a propriedade $x < y$,
  - a expressão diz que para todo inteiro x existe um inteiro maior.
- Esta expressão é verdadeira.
- Entretanto, se invertermos a ordem dos quantificadores escrevendo $(\exists y)(\forall x)Q(x, y)$, a mesma interpretação diz que:
  - existe um inteiro y que é maior que qualquer outro inteiro x.
- Neste caso, o valor lógico da expressão é falso.
- Isto ressalta o fato de que a ordem dos quantificadores é importante!