

Lógica e Matemática computacional

Unidade 04: Tabela Verdade
Aula 03: Aplicações da Tabela Verdade

Prof. Ms. Romulo de Almeida Neves





Sumário

01

**Precedência dos
operadores lógicos**

Conceitos e Definições

02

**Tabela verdade com
preposições intermediárias**

Conceitos e Exemplos

03

**Estrutura condicional em linguagens de
programação**

Conceitos e Exemplos

.....

01



Precedência dos operadores lógicos

Conceitos e Exemplos

Precedência dos operadores lógicos



- Para resolver uma expressão lógica que combina várias proposições com conectivos lógicos é preciso obedecer a seguinte regra de precedência:
 - 1. Para expressões que tenham parênteses, primeiro efetuam-se as operações lógicas dentro dos parênteses mais internos.
 - 2. \neg (Negação) (maior precedência).
 - 3. $\wedge \vee$, (Conjunção e disjunção).
 - 4. \rightarrow (Implicação).
 - 5. \leftrightarrow (Bicondicional).



Precedência dos operadores lógicos



- Construir uma Tabela Verdade para as fórmulas
 - $A \wedge B \rightarrow A$ e $A \vee (B \rightarrow A)$.

| C1 | C2 | C3 | C4 | C5 | C6 |
|----|----|--------------|----------------------------|-------------------|------------------------------|
| A | B | $A \wedge B$ | $A \wedge B \rightarrow A$ | $B \rightarrow A$ | $A \wedge (B \rightarrow A)$ |
| V | V | V | V | V | V |
| V | F | F | V | V | V |
| F | V | F | V | F | F |
| F | F | F | V | V | F |



02



Tabela verdade com preposições intermediárias

Definição e Exemplos



Tabela verdade com preposições intermediárias

- Dada uma fórmula com várias proposições, conectores e parênteses dentro de parênteses, a resolução deve começar pelos parênteses mais internos.
- A fórmula $((A \vee B) \rightarrow C) \wedge A$ deve ter a seguinte ordem de resolução:
 - 1 - $A \vee B$ (parênteses mais internos)
 - 2 - $((A \vee B \rightarrow C)$ (parênteses mais externos)
 - 3 - $((A \vee B) \rightarrow C) \wedge A$ (operação fora dos parênteses).

Tabela verdade com preposições intermediárias

Tabela Verdade para a fórmula $((A \vee B) \rightarrow C) \wedge A$

| | | | P | Q | R |
|---|---|---|------------|-------------------|--------------|
| A | B | C | $A \vee B$ | $P \rightarrow C$ | $Q \wedge A$ |
| V | V | V | V | V | V |
| V | V | F | V | F | F |
| V | F | V | V | V | V |
| V | F | F | V | F | F |
| F | V | V | V | V | F |
| F | V | F | V | F | F |
| F | F | V | F | V | F |
| F | F | F | F | V | F |

03

Estrutura condicional em linguagens de programação

Conceitos e Exemplos



Estrutura condicional em linguagens de programação

- O comando se é escrito em inglês (if) e o então, nas linguagens C e Java é a chave ({) e em python é o dois pontos (:).
- Outro detalhe é a escrita da conjunção; em C e Java é feito pelo && e em python pelo comando and.

Estrutura condicional em linguagens de programação

- Os conectivos lógicos, também foram usados operadores relacionais. A sintaxe `!=` significa “diferente” e a sintaxe `==` significa “igual”.

| Linguagem | Sintaxe |
|-----------|--|
| C | <pre>if(A != "BRASILIA" && B == "GASOLINA") { printf("Petrobras"); }</pre> |
| Java | <pre>if(A != "BRASILIA" && B == "GASOLINA") { System.out.println("Petrobras"); }</pre> |
| Python | <pre>if A != "BRASILIA" and B == "GASOLINA": print("Petrobras")</pre> |