# RACIOCÍNIO LÓGICO

AULA 2

Prof. André Roberto Guerra

## CONVERSA INICIAL

O roteiro proposto contempla, nesta aula, as operações lógicas sobre as proposições e a construção das tabelas-verdade. Serão apresentadas as regras específicas das operações e sua aplicação – o cálculo proposicional.

Definidos os termos e símbolos que compõem a lógica proposicional, inicia-se o cálculo das sentenças/proposições, permitindo a validação da resposta encontrada (Quadro 1).

Quadro 1 – Objetivos da lógica proposicional

| | |
|---|---|
| **Objetivos de acordo com a taxonomia de Bloom revisada** | Operações lógicas sobre proposições<br>Construção de tabelas-verdade |
| | Objetivos específicos |

Fonte: Elaborado pelo autor.

## TEMA 1 – OPERAÇÕES LÓGICAS

Os **princípios da lógica** (identidade, contradição e terceiro excluído) e o **alfabeto** (símbolos da linguagem, variáveis proposicionais, conectivos lógicos e símbolos especiais), apresentados anteriormente, são a base do cálculo das proposições.

Proposição é um conjunto de símbolos (alfabeto, combinação de palavras, frase) que deve atender a duas condições básicas:

- deve ter sentido completo;
- ser apenas verdadeira (V) ou falsa (F).

São também classificadas como:

- simples;
- ou compostas (caso em que possuem os operadores lógicos).

Proposições **simples** são representadas por símbolos: as letras minúsculas $\boldsymbol{p, q, r, s}$ etc. São, portanto, denominadas *variáveis proposicionais*, as quais podem assumir os valores lógicos de verdadeiras ou falsas (**V** ou **F**).

Exemplos de variáveis proposicionais:

- $\boldsymbol{p}$: “A lua é quadrada”;

- $\boldsymbol{q}$: “A neve é branca”.

Proposições **compostas**, ou **fórmulas proposicionais**, são representadas por letras maiúsculas ($\boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{R}, \boldsymbol{S}$ etc.), indicando as variáveis proposicionais que as compõem ($\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{r}, \boldsymbol{s}$ etc.).

Exemplo de fórmula proposicional (proposição composta):

- $\boldsymbol{p}$: “A lua é quadrada”;
- $\boldsymbol{q}$: “A neve é branca”;
- $\boldsymbol{P}(\boldsymbol{p}, \boldsymbol{q})$: “**Se** a lua é quadrada, **então** a neve é branca”. “Se $\boldsymbol{p}$, então $\boldsymbol{q}$”.

## 1.1 Operações e conectivos

Em lógica simbólica, a ação de combinar proposições é chamada de *operação* e os conectivos são chamados de *operadores* e são representados por símbolos específicos, conforme Quadro 2.

Quadro 2 – As operações lógicas e seus respectivos conectivos e símbolos

| **Operação** | **Conectivo** | **Símbolo** |
|---|---|---|
| Negação | Não | $\sim$ |
| Conjunção | E | $\wedge$ |
| Disjunção | Ou | $\vee$ |
| Implicação ou condicional | se... então | $\rightarrow$ |
| Bi-implicação ou bicondicional | se e somente se | $\leftrightarrow$ |

Fonte: Elaborado pelo autor.

Para determinar o valor lógico de uma proposição composta, em função dos valores lógicos das proposições que a compõem, deve-se definir as operações, isto é, dar o resultado da operação para cada possível conjunto de valores dos operandos.

Operação é definida como **a ação de combinar proposições** e os conectivos (representados por símbolos) são os operadores.

Os conectivos lógicos são usados para construir proposições compostas (fórmulas atômicas), formando argumentos mais complexos. São os conectivos lógicos que permitirão construir proposições compostas para formar argumentos mais complexos.

Usando conectivos, é possível construir proposições como nos exemplos:

- “Pedro é magro **e** Mário é alto”;
- “José foi ao cinema, Roberto foi ao parque **e** Marcelo ficou em casa”;
- “Roberta foi ao mercado **ou** à padaria”;
- “João saiu **e** Marcos ficou em casa”;
- “A Lua **não** é satélite da Terra”;
- “**Se** a chuva continuar a cair, **então** o rio vai transbordar”;
- “**Se** Pedro estudar, **então** será aprovado”;
- “Pedro será aprovado **se e somente se** estudar”.

## 1.2 Fórmulas proposicionais

Fórmulas proposicionais (proposições compostas) são representadas por letras maiúsculas ($\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{R}$ etc.) indicando as variáveis que as compõem ($\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{r}, \boldsymbol{s}$ etc.). Como em:

$\boldsymbol{p}$: **“**A lua é quadrada”;

$\boldsymbol{q}$: “A neve é branca”; então

$\boldsymbol{A}(\boldsymbol{p}, \boldsymbol{q})$: “**Se** a lua é quadrada, **então** a neve é branca” = “$\boldsymbol{Se\ p, ent}$ã$\boldsymbol{o\ q}$” $\boldsymbol{p} \rightarrow \boldsymbol{q}$.

Toda fórmula atômica (proposição composta) é uma fórmula. Se $\boldsymbol{A}$ e $\boldsymbol{B}$ são fórmulas, então: $(\sim\boldsymbol{A}), (\boldsymbol{A} \wedge \boldsymbol{B}), (\boldsymbol{A} \vee \boldsymbol{B}), (\boldsymbol{A} \rightarrow \boldsymbol{B}), (\boldsymbol{A} \leftrightarrow \boldsymbol{B})$ também são fórmulas.

Todos os símbolos proposicionais e de verdade são fórmulas. Se $\boldsymbol{A}$ é uma **fórmula**, então $\sim\boldsymbol{A}$ também é uma fórmula.

Constituem operações sobre as fórmulas proposicionais:

- Negação: $\sim\boldsymbol{A}, \sim\boldsymbol{B}$;
- Conjunção: $\boldsymbol{A} \wedge \boldsymbol{B}$;
- Disjunção: $\boldsymbol{A} \vee \boldsymbol{B}$;
- Implicação: $\boldsymbol{A} \rightarrow \boldsymbol{B}$;
- Bi-implicação: $\boldsymbol{A} \leftrightarrow \boldsymbol{B}$, sendo: $\boldsymbol{A}$: antecedente; $\boldsymbol{B}$: consequente.

Exemplo de fórmula proposicional (proposição composta):

$\boldsymbol{p}$: “Eu estudo”;

$\boldsymbol{q}$: “Sou aprovado”;

$P(p)$: **"Não** estudo" = $\sim p$;

$P(p, q)$: "Estudo **e** sou aprovado" = $p \wedge q$;

$P(p, q)$: "Estudo **ou** sou aprovado" = $p \vee q$;

$P(p, q)$: **"Se** estudo **então** sou aprovado" = $p \rightarrow q$;

$P(p, q)$: "Sou aprovado **se e somente se** estudo" = $q \leftrightarrow p$.

### 1.3 Ordem de precedência dos conectivos/operações

Em proposições mais longas, o uso de muitos parênteses para definir a precedência das operações pode tornar sua análise mais complexa. Para resolver isso, é comum se estabelecer uma ordem de precedência dos conectivos lógicos que torna desnecessária a colocação de parênteses:

| | | | | | |
|---|---|---|---|---|---|
| **Precedência:** | 1ª | 2ª | 3ª | 4ª | 5ª |
| **Operador:** | $\sim$ | $\wedge$ | $\vee$ | $\rightarrow$ | $\leftrightarrow$ |

Os parênteses são utilizados apenas para alterar a ordem de precedência de avaliação dos operadores. Assim, na expressão $p \wedge (q \vee r)$, o *ou* é avaliado antes do *e*.

Para os operadores binários ($\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$), quando uma expressão contém vários operadores iguais, avalia-se **da esquerda para a direita**.

Para o operador unário $\sim$, quando houver vários consecutivos, resolve-se **da direita para a esquerda** ("de dentro para fora").

Em relação ao algoritmo ordem de precedência, há quatro passos:

1. Percorra a expressão **da esquerda para a direita**, executando as operações de **negação**, na ordem em que aparecerem.

2. Percorra novamente a expressão, **da esquerda para a direita**, executando as operações de **conjunção** e **disjunção**, na ordem em que aparecerem.

3. Percorra outra vez a expressão, **da esquerda para a direita**, executando dessa vez as operações de **implicação** (**condicionais**), na ordem em que aparecerem.

4. Percorra uma última vez a expressão, **da esquerda para a direita**, executando as operações de **bi-implicação** (**bicondicionais**), na ordem em que aparecerem.

Dessa forma, as operações da expressão $\boldsymbol{p} \wedge \sim \boldsymbol{q} \rightarrow \boldsymbol{r} \vee \boldsymbol{s}$ serão executadas na seguinte ordem:

$$\boldsymbol{p} \wedge \sim \boldsymbol{q} \rightarrow \boldsymbol{r} \vee \boldsymbol{s}$$

2 1 4 3.

Um caso especial é a utilização de negações consecutivas; por exemplo, a proposição "é falso que eu não tenha saído" pode ser simbolizada por $\sim\sim \boldsymbol{p}$ (em que $\boldsymbol{p}$ representa "eu tenha saído"); nesse caso, a segunda negação deve ser executada antes.

Quando necessário modificar a ordem de precedência, utilizam-se parênteses. Assim, no exemplo dado, a expressão $\boldsymbol{p} \wedge \boldsymbol{q} \rightarrow \boldsymbol{r}$ significa:

"**Se** Mário foi ao cinema **e** João foi ao teatro, **então** Marcelo ficou em casa";

e a expressão $\boldsymbol{p} \wedge (\boldsymbol{q} \rightarrow \boldsymbol{r})$ significa:

"Mário foi ao cinema **e**, **se** João foi ao teatro, **então** Marcelo ficou em casa".

A utilização dos conectivos $\wedge$ e $\vee$ pode causar ambiguidade até mesmo em linguagem natural, como, por exemplo, na expressão:

"Mário foi ao cinema **e** Marcelo ficou em casa **ou** Maria foi à praia",

representada por $\boldsymbol{p} \wedge \boldsymbol{q} \vee \boldsymbol{s}$, que não deixa claro seu significado, que tanto pode significar:

"Mário foi ao cinema **e** Marcelo ficou em casa **ou** então Maria foi à praia", representada por $(\boldsymbol{p} \wedge \boldsymbol{q}) \vee \boldsymbol{s}$; como pode significar:

"Mário foi ao cinema **e** Marcelo ficou em casa **ou** Maria foi à praia", representada por $\boldsymbol{p} \wedge (\boldsymbol{q} \vee \boldsymbol{s})$, que são claramente afirmações distintas.

Segundo a ordem de precedência da lógica, a expressão dada corresponde à primeira forma apresentada; contudo, para evitar incertezas, utilizam-se parênteses.

Assim, as expressões simbólicas podem assumir aspectos ainda mais complexos, como, por exemplo:

$$(\boldsymbol{p} \leftrightarrow \boldsymbol{q} \vee (\sim \boldsymbol{r} \rightarrow \boldsymbol{s})) \wedge \sim \boldsymbol{t}.$$

Para determinar a ordem de execução das operações que possuem parênteses, utiliza-se o **algoritmo ordem de precedência com parênteses**, em cinco passos, a saber:

1. Percorra a expressão até encontrar o primeiro ")".

2. Volte até encontrar o "(" correspondente, delimitando assim um trecho da expressão sem parênteses.
3. Execute o **algoritmo ordem de precedência** sobre a expressão delimitada.
4. Elimine o par de parênteses encontrado.
5. Volte ao passo 1.

De acordo com esse algoritmo, as operações da expressão anterior seriam executadas nesta ordem:

$$(\boldsymbol{p} \leftrightarrow \boldsymbol{q} \vee (\sim \boldsymbol{r} \rightarrow \boldsymbol{s})) \wedge \sim \boldsymbol{t}$$

4 3 1 2 6 5.

Uma proposição composta é, portanto, formada por conexões de proposições simples. Ou seja, uma proposição composta é uma cadeia constituída pelos símbolos $\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{r}$ etc., representando proposições simples, símbolos de conectivos e parênteses.

No entanto, nem toda cadeia desses símbolos representa uma proposição composta; por exemplo, a cadeia $\boldsymbol{AB}$ ↔ ) ∧∧∨ ( $\boldsymbol{C}$ → não tem nenhum significado em lógica. Temos então o problema de reconhecer quando uma cadeia desses símbolos representa realmente uma proposição composta.

As proposições são também conhecidas por *fórmulas bem formadas* (ou, simplesmente, *fórmulas*) e possuem uma lei de formação.

**Nota**

Estabelecer precedência entre operadores é uma questão basicamente de convenção. Por exemplo, alguns autores estabelecem a precedência dos operadores de implicação e bi-implicação (→ e ↔, respectivamente) antes dos de conjunção e disjunção (∧ e ∨, respectivamente). Também se convencionou que, quando os operadores são repetidos, resolve-se a expressão da esquerda para a direita (exceto para a negação). Todavia, em outras referências, convenciona-se que, quando há repetição do operador de implicação (→) ou bi-implicação (↔), resolve-se a expressão da direita para a esquerda. Assim, para essa outra convenção, a expressão que aqui seria escrita como $\boldsymbol{p} \rightarrow \boldsymbol{q} \rightarrow \boldsymbol{r}$, lá deve ser escrita como $(\boldsymbol{p} \rightarrow \boldsymbol{q}) \rightarrow \boldsymbol{r}$. Dessa forma, uma expressão como essa será avaliada da mesma forma, nas duas convenções, por meio do uso de parênteses para ajustar as precedências. Portanto, quando for consultado um livro sobre lógica ou utilizada uma calculadora proposicional (existe disponível na *web*), deve-se verificar qual a precedência utilizada para poder compatibilizar as proposições.

## 1.4 Tradução da linguagem natural para a linguagem dos símbolos

Trata-se de procedimento inicial para o cálculo proposicional a tradução da linguagem natural para a linguagem dos símbolos (proposições lógicas) e vice-versa.

Segue exemplo de tradução de linguagem natural para simbólica com *mas* e *nem* e uso de parênteses:

$\boldsymbol{p}$: “Está quente”;

$\boldsymbol{q}$: “Tem sol”;

“**Não** está quente **mas** tem sol” equivale a “**Não** está quente **e** tem sol”; $(\sim\boldsymbol{p}) \wedge \boldsymbol{q}$ (*mas* equivale a ∧).

Já “**Não** está quente **nem** ensolarado” equivale a “**Não** está quente e **não** está ensolarado”; $(\sim\boldsymbol{p}) \wedge (\sim\boldsymbol{q})$.

Parênteses (símbolos auxiliares) são utilizados para estabelecer o alcance ou a precedência na avaliação de expressões mais longas, com mais de um operador. E enquanto não for definida uma precedência para os diversos operadores ($\sim, \wedge, \vee, \rightarrow, \leftrightarrow$). Um exemplo de operações lógicas simbólicas com parênteses:

$\boldsymbol{p}$: “A lua é quadrada”;

$\boldsymbol{q}$: “A neve é branca”;

“**Se** a lua é quadrada **e** a neve é branca, **então** a lua **não** é quadrada”: $(\boldsymbol{p} \wedge \boldsymbol{q}) \rightarrow (\sim\boldsymbol{p})$.

“A lua **não** é quadrada **se e somente se** a neve é branca”: $(\sim\boldsymbol{p}) \leftrightarrow \boldsymbol{q}$.

<u>Outro </u>exemplo de tradução de linguagem natural para simbólica encontra-se com base na seguinte proposição composta:

“Meu peso aumenta **se e somente se não** faço nem dieta nem exercícios, **então** vou para o trabalho a pé **ou** de bicicleta”.

Determinam-se as proposições simples e, com elas, monta-se a proposição simbólica correspondente a essa proposição composta.

$\boldsymbol{p}$: “Meu peso aumenta”;

$\boldsymbol{q}$: “Eu faço dieta”;

$\boldsymbol{r}$: “Eu faço exercícios”;

$\boldsymbol{s}$: “Eu vou para o trabalho a pé”;

$\boldsymbol{t}$: “Eu vou para o trabalho de bicicleta”, logo:

$$(\boldsymbol{p} \leftrightarrow (\sim \boldsymbol{q} \wedge \sim \boldsymbol{r})) \rightarrow (\boldsymbol{s} \vee \boldsymbol{t}).$$

## TEMA 2 – TABELAS-VERDADE

Numa proposição composta, as diversas proposições simples podem assumir os valores lógicos de verdadeiro ou falso ($\boldsymbol{V}$ ou $\boldsymbol{F}$). Uma vez estabelecida a proposição composta, é necessário poder se calcular o seu valor final em função das diversas combinações possíveis para os valores das proposições simples ($\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{r}$ etc.), o que também resultará em um valor lógico ($\boldsymbol{V}$ ou $\boldsymbol{F}$).

Portanto, é preciso conhecer o resultado das operações resultantes da utilização dos diversos conectivos lógicos representados por $\sim, \wedge, \vee, \rightarrow$ e $\leftrightarrow$.

Dados uma expressão proposicional e os valores lógicos das proposições simples que a compõem, utilizando-se a ordem de precedência é possível calcular o valor lógico da expressão.

A construção de tabela-verdade é a maneira de confirmar os valores que são apresentados em cada proposição.

Para se determinar o valor (**verdadeiro** ou **falso**) das proposições compostas (moleculares), após conhecidos os valores das proposições simples (atômicas) que as compõem, são utilizadas as tabelas-verdade.

A tabela-verdade de uma proposição composta é uma tabela na qual são apresentados todos os valores-verdade possíveis de uma proposição composta, para cada combinação dos valores-verdade de suas proposições componentes.

Cada linha da tabela corresponde a uma possível combinação dos valores lógicos das proposições componentes; como são dois os valores lógicos, existem, para $\boldsymbol{n}$ componentes, $\mathbf{2^n}$ combinações possíveis.

Ao aplicar um operador lógico qualquer entre dois operandos lógicos $\boldsymbol{p}$ e $\boldsymbol{q}$, existirão quatro possibilidades de combinação entre $\boldsymbol{p}$ e $\boldsymbol{q}$; e, assim, serão quatro os resultados possíveis para a operação, cada um sendo $\boldsymbol{V}$ ou $\boldsymbol{F}$.

A Tabela 1 ilustra essa situação.

Tabela 1 – Exemplo de tabela-verdade

| p | q | p, q |
|---|---|---|
| V | V | V ou F |
| V | F | V ou F |
| F | V | V ou F |
| F | F | V ou F |

Fonte: Elaborada pelo autor.

A tabela-verdade possui dois tipos de colunas:

- **Colunas** para as **proposições componentes** (variáveis), onde são distribuídos os valores $\boldsymbol{V}$ e $\boldsymbol{F}$ de forma a incluir cada possível combinação.
- **Colunas** para as **operações**, onde os valores $\boldsymbol{V}$ e $\boldsymbol{F}$ são obtidos pelas operações.

Assim, se a expressão possui $\boldsymbol{n}$ componentes e $\boldsymbol{m}$ operações, a tabela terá $\boldsymbol{m} + \boldsymbol{n}$ colunas.

## 2.1 Valor lógico proposicional

Para recuperar o valor lógico de uma proposição $\boldsymbol{p}$, será utilizada a seguinte notação: $\boldsymbol{VL}(\boldsymbol{p})$. Assim, se $\boldsymbol{p}$ for verdadeira, então $\boldsymbol{VL}(\boldsymbol{p}) = \boldsymbol{V}$; se for falsa, $\boldsymbol{VL}(\boldsymbol{p}) = \boldsymbol{F}$.

# TEMA 3 – CONSTRUÇÃO DE TABELAS-VERDADE

Dada uma expressão proposicional e dados os valores lógicos das proposições simples que a compõem, é possível, com a ordem de precedência, calcular o valor lógico da expressão dada. Contudo, o objetivo, muitas vezes, é se determinar o conjunto de valores lógicos que a expressão pode assumir, para quaisquer valores lógicos das proposições componentes.

Por exemplo, considere a expressão proposicional: $\boldsymbol{p} \vee \boldsymbol{q} \rightarrow \boldsymbol{p} \wedge \boldsymbol{q}$. Na expressão, existem apenas duas variáveis, $\boldsymbol{p}$ e $\boldsymbol{q}$. Cada uma pode ser verdadeira ou falsa, então temos quatro possibilidades: $\boldsymbol{p}$ e $\boldsymbol{q}$, ambas, verdadeiras; $\boldsymbol{p}$ verdadeira e $\boldsymbol{q}$ falsa; $\boldsymbol{p}$ falsa e $\boldsymbol{q}$ verdadeira; ou, finalmente, $\boldsymbol{p}$ e $\boldsymbol{q}$, ambas, falsas.

Cada linha da tabela-verdade corresponde a uma possível combinação dos valores lógicos das proposições componentes; como são dois os valores lógicos, existem, para $\boldsymbol{n}$ componentes, $\boldsymbol{2^n}$ combinações possíveis. Portanto, a tabela-verdade tem $\boldsymbol{2^n}$ linhas, além do cabeçalho.

Para determinar a tabela-verdade, são estabelecidas certas convenções para sua construção:

- Para as colunas:

1. dispor as proposições componentes em ordem alfabética;
2. dispor as operações na ordem de precedência determinada pelo algoritmo ordem de precedência (com parênteses, se for o caso).

- Para as linhas:

1. alternar $\boldsymbol{V}$ e $\boldsymbol{F}$ na coluna do último componente;
2. alternar $\boldsymbol{V}, \boldsymbol{V}$ e $\boldsymbol{F}, \boldsymbol{F}$ na coluna do penúltimo componente;
3. alternar $\boldsymbol{V}, \boldsymbol{V}, \boldsymbol{V}, \boldsymbol{V}$ e $\boldsymbol{F}, \boldsymbol{F}, \boldsymbol{F}, \boldsymbol{F}$ na coluna do antepenúltimo componente;
4. prosseguir dessa forma se houver mais componentes, sempre dobrando o número de $\boldsymbol{V}$ e $\boldsymbol{F}$ a cada coluna à esquerda.

Utilizando-se as convenções estabelecidas para construção das colunas, calculado o número de linhas e obedecendo-se à ordem de precedência das operações, a tabela assume a forma qual a Tabela 2.

Tabela 2 – Tabela-verdade em construção

| $\boldsymbol{p}$ | $\boldsymbol{q}$ | $\boldsymbol{p} \vee \boldsymbol{q}$ | $\boldsymbol{p} \wedge \boldsymbol{q}$ | $\boldsymbol{p} \vee \boldsymbol{q} \rightarrow \boldsymbol{p} \wedge \boldsymbol{q}$ |
|---|---|---|---|---|
| V | V | V | V | V |
| V | F | V | F | F |
| F | V | V | F | F |
| F | F | F | F | V |

Fonte: Elaborada pelo autor.

Uma tabela-verdade como a da Tabela 2, na qual são apresentados todos os valores-verdade possíveis de uma proposição composta, para cada combinação dos valores-verdade das proposições componentes, é chamada de *tabela-verdade da proposição composta*.

Ex.: considere a expressão proposicional

$(\boldsymbol{p} \rightarrow \boldsymbol{q}) \vee \sim ((\boldsymbol{p} \leftrightarrow \boldsymbol{r}) \rightarrow \sim \boldsymbol{r})$.

A precedência das operações é dada por:

$(\boldsymbol{p} \rightarrow \boldsymbol{q}) \vee \sim ((\boldsymbol{p} \leftrightarrow \boldsymbol{r}) \rightarrow \sim \boldsymbol{r})$

1 6 5 2 4 3.

A tabela-verdade correspondente assume o aspecto da Tabela 3.

Tabela 3 – Exemplo de aplicação de tabela-verdade

| | $p$ | $q$ | $r$ | $p \to q$ (1) | $p \leftrightarrow r$ (2) | $\sim r$ (3) | $(2) \to (3)$ (4) | $\sim (4)$ (5) | $(p \to q) \vee \sim ((p \leftrightarrow r) \to \sim r)$ (6) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | V | V | V | V | V | F | F | V | V |
| 2 | V | V | F | V | F | V | V | F | V |
| 3 | V | F | V | F | V | F | F | V | V |
| 4 | V | F | F | F | F | V | V | F | F |
| 5 | F | V | V | V | F | F | V | F | V |
| 6 | F | V | F | V | V | V | V | F | V |
| 7 | F | F | V | V | F | F | V | F | V |
| 8 | F | F | F | V | V | V | V | F | V |

Fonte: Elaborada pelo autor.

A atribuição de valores lógicos aos componentes simples de uma proposição composta é chamada de *uma interpretação* dessa proposição. Assim, uma proposição com $\boldsymbol{n}$ componentes, $\mathbf{2}^{\boldsymbol{n}}$ combinações possíveis, $\boldsymbol{n}$ componentes simples distintos, admitirá $\mathbf{2}^{\boldsymbol{n}}$ interpretações.

## TEMA 4 – REGRAS ESPECÍFICAS DAS OPERAÇÕES

Para a construção das tabelas-verdade, algumas **regras**/**normas** específicas para cada **operação lógica** individualmente aqui apresentadas devem ser seguidas/observadas.

### 4.1 Negação

Chama-se *negação de uma proposição* $\boldsymbol{p}$ a proposição representada por $\sim\boldsymbol{p}$ (não $\boldsymbol{p}$), cujo valor lógico é $\boldsymbol{F}$ (falso) quando $\boldsymbol{p}$ é $\boldsymbol{V}$ (verdadeiro) e $\boldsymbol{V}$ (verdadeiro) quando $\boldsymbol{p}$ é $\boldsymbol{F}$ (falso).

Seja a proposição $\boldsymbol{p}$: “Paulo vai a festa”. Se $\boldsymbol{VL}(\boldsymbol{p}) = \boldsymbol{V}$, então $\boldsymbol{VL}(\sim\boldsymbol{p}) = \boldsymbol{F}$, ou seja, $\sim \boldsymbol{p}$: “Paulo **não** vai à festa”.

A negação inverte o valor-verdade de uma expressão. Se $\boldsymbol{p}$ é $\boldsymbol{V}$ (verdadeira), a negação de $\boldsymbol{p}$ é $\boldsymbol{F}$ (falsa), ao passo que se $\boldsymbol{p}$ é $\boldsymbol{F}$ (falsa), $\sim \boldsymbol{p}$ é $\boldsymbol{V}$ (verdadeira).

A tabela-verdade da operação de negação segue as normas e regras definidas, em especial por essa ser a única operação unária (uma função com somente uma variável). Portanto, com $(\mathbf{2}^{\mathbf{1}})$, 2 linhas (Tabela 4).

Tabela 4 – Tabela-verdade de uma operação de negação

| $p$ | $\sim p$ |
|---|---|
| V | F |
| F | V |

Fonte: Elaborada pelo autor.

## 4.2 Conjunção

Chama-se *conjunção* de duas proposições $p$ e $q$ a proposição representada por $p \wedge q$ (p **e** q), cujo valor lógico é $V$ (verdadeiro) quando ambas as proposições ($p$, $q$) são $V$ (verdadeiras) e $F$ (falso), nos demais casos.

Seja a proposição: “Paulo e José irão à festa” ($p$: “Paulo vai à festa”; $q$: “José vai à festa”). Essa proposição será $V$ (verdadeira) apenas quando Paulo for **e** José for, sendo $F$ (falsa) em qualquer outra situação (quando Paulo for **e** José **não** for ou Paulo **não** for **e** José for ou quando ambos **não** forem).

Se $p$ e $q$ são proposições, a expressão $p \wedge q$ é chamada de *conjunção* de $p$ e $q$ e as proposições $p$ e $q$ são chamadas de *fatores da expressão*.

A tabela-verdade da operação de **conjunção** segue normas e regras definidas e, como nas outras operações, essa é uma função com pelo menos duas variáveis. Portanto, com ($\mathbf{2^2}$), 4 linhas (Tabela 5).

Tabela 5 – Tabela-verdade de uma operação de conjunção

| $p$ | $q$ | $p \wedge q$ |
|---|---|---|
| V | V | V |
| V | F | F |
| F | V | F |
| F | F | F |

Fonte: Elaborada pelo autor.

## 4.3 Disjunção

Chama-se *disjunção* (*inclusiva*) de duas proposições $p$ e $q$ a proposição representada por $p \vee q$ (p **ou** q), cujo valor lógico é $V$ (verdadeiro) quando pelo menos uma das proposições ($p$, $q$) é $V$ (verdadeira), sendo $F$ (falso) apenas quando ambos os valores lógicos das proposições forem $F$ (falsos), o que pode ser representado por:

$(VL(p) = F, VL(q) = F)$.

Seja a proposição: “Paulo ou José irão à festa” ($p$: “Paulo vai à festa”; $q$: “José vai à festa”). Essa proposição será $V$ (verdadeira) em qualquer situação em que Paulo for **ou** José for (quando Paulo for **ou** José **não** for; ou quando Paulo **não** for **e** José for; ou quando Paulo for e José for), sendo $F$ (falsa) apenas quando ambos **não** forem.

Se $p$ e $q$ são proposições, a expressão $p \vee q$ é chamada de *disjunção* de $p$ e $q$ e as proposições $p$ e $q$ são chamadas de *parcelas da expressão*.

A tabela-verdade da operação de *disjunção* segue normas e regras definidas, como nas outras operações, também com $(2^2)$, 4 linhas (Tabela 6).

Tabela 6 – Tabela-verdade de uma operação de disjunção

| $p$ | $q$ | $p \vee q$ |
|---|---|---|
| V | V | V |
| V | F | V |
| F | V | V |
| F | F | F |

Fonte: Elaborada pelo autor.

## 4.4 Implicação (condicional)

Chama-se de *implicação* (ou *condicional*), de duas proposições $p$ e $q$, a proposição representada por $p \rightarrow q$ (**se** p, **então** q), cujo valor lógico é $F$ (falso) apenas quando o valor lógico da proposição **antecedente** é $V$ (verdadeiro) e o **consequente** é $F$ (falso), sendo $V$ (verdadeiro) em todos os demais casos.

Seja a proposição: “**Se** Paulo for à festa, **então** José vai à festa” ($p$: “Paulo vai à festa”; $q$: “José vai à festa”). Essa proposição será $F$ (falsa) apenas quando Paulo (**antecedente**) for e José (**consequente**) **não** for. Será $V$ (verdadeira) em qualquer outra situação (quando Paulo for **e** José for, quando José for **ou** Paulo **não** for ou quando Paulo **não** for **e** José **não** for).

A tabela-verdade da operação de **implicação** segue normas e regras definidas, como nas outras operações, também com $(2^2)$, 4 linhas (Tabela 7).

Tabela 7 – Tabela-verdade de uma operação de implicação (condicional)

| $p$ | $q$ | $p \rightarrow q$ |
|---|---|---|
| **V** | **V** | **V** |
| **V** | **F** | **F** |
| **F** | **V** | **V** |
| **F** | **F** | **V** |

Fonte: Elaborada pelo autor.

## 4.5 Bi-implicação (bicondicional)

Chama-se de *bi-implicação* (ou *bicondicional*), de duas proposições $\boldsymbol{p}$ e $\boldsymbol{q}$, a proposição representada por $\boldsymbol{p} \leftrightarrow \boldsymbol{q}$ (p **se e somente se** q), cujo valor lógico é $\boldsymbol{V}$ (verdadeiro) quando os valores lógicos das proposições são **iguais** e $\boldsymbol{F}$ (falso) quando os valores lógicos das proposições são **diferentes**.

Seja a proposição: “Paulo vai à festa **se e somente se** José vai à festa” ($\boldsymbol{p}$: “Paulo vai à festa”; $\boldsymbol{q}$: “José vai à festa”). Essa proposição será $\boldsymbol{V}$ (verdadeira) sempre que ambos forem ou não. Será $\boldsymbol{F}$ (falsa) sempre quando um for e o outro **não** for.

A tabela-verdade da operação de **bi-implicação** segue normas e regras definidas, como nas outras operações, também com $(\mathbf{2^2})$, 4 linhas (Tabela 8).

Tabela 8 – Tabela-verdade de uma operação de bi-implicação (bicondicional)

| $p$ | $q$ | $p \leftrightarrow q$ |
|---|---|---|
| **V** | **V** | **V** |
| **V** | **F** | **F** |
| **F** | **V** | **F** |
| **F** | **F** | **V** |

Fonte: Elaborada pelo autor.

## 4.6 Ou exclusivo

Importante ressaltar que a operação chamada de *ou* pode ter dois sentidos na linguagem habitual: inclusivo (**disjunção** “∨”) e exclusivo (“$\underline{\vee}$”).

É representada por $\boldsymbol{p} \ \underline{\vee} \ \boldsymbol{q}$ (p **ou exclusivo** q), que significa $((\boldsymbol{p} \vee \boldsymbol{q}) \wedge \sim(\boldsymbol{p} \wedge \boldsymbol{q}))$, cujo valor lógico é $\boldsymbol{V}$ (verdadeiro) quando os valores lógicos das proposições são **diferentes** e $\boldsymbol{F}$ (falso) quando os valores lógicos das proposições são **iguais**.

A tabela-verdade da operação **ou exclusivo** segue normas e regras definidas, como nas outras operações, também com ($\mathbf{2^2}$), 4 linhas (Tabela 9).

Tabela 9 – Tabela-verdade de uma operação de *ou exclusivo*

| $\boldsymbol{p}$ | $\boldsymbol{q}$ | $((\boldsymbol{p} \vee \boldsymbol{q}) \wedge \sim(\boldsymbol{p} \wedge \boldsymbol{q}))$ |
|---|---|---|
| **V** | **V** | **V F F V** |
| **V** | **F** | **V V V F** |
| **F** | **V** | **V V V F** |
| **F** | **F** | **F F V F** |

Fonte: Elaborada pelo autor.

## TEMA 5 – APLICAÇÃO: CÁLCULO PROPOSICIONAL

Construir a tabela-verdade da seguinte proposição:

$\boldsymbol{p} \rightarrow (\boldsymbol{q} \wedge \sim \boldsymbol{p}) \leftrightarrow \sim (\boldsymbol{p} \vee \boldsymbol{q}) \wedge (\sim \boldsymbol{p} \veebar \boldsymbol{q})$.

Seguir as convenções estabelecidas para construção da tabela-verdade, que tem $\mathbf{2^n}$ linhas, além do cabeçalho; logo, ($\mathbf{2^2}$), 4 linhas.

**Para as colunas**:

1. Dispor as proposições componentes em ordem alfabética.

2. Dispor as operações na ordem de precedência determinada pelo algoritmo ordem de precedência (com parênteses, se for o caso).

Tabela 10 – Colunas da tabela-verdade proposta

| | $\boldsymbol{p}$ | $\boldsymbol{q}$ | $\boldsymbol{p} \rightarrow (\boldsymbol{q} \wedge \sim \boldsymbol{p}) \leftrightarrow \sim (\boldsymbol{p} \vee \boldsymbol{q}) \wedge (\sim \boldsymbol{p} \veebar \boldsymbol{q})$<br>8 2 1 9 6 3 7 4 5 |
|---|---|---|---|
| 1 | | | |
| 2 | | | |
| 3 | | | |
| 4 | | | |

Fonte: Elaborada pelo autor.

**Para as linhas**:

3. Alternar $\boldsymbol{V}$ e $\boldsymbol{F}$ na coluna do último componente.

4. Alternar $\boldsymbol{V}, \boldsymbol{V}$ e $\boldsymbol{F}, \boldsymbol{F}$ na coluna do penúltimo componente.

5. Alternar $\boldsymbol{V}, \boldsymbol{V}, \boldsymbol{V}, \boldsymbol{V}$ e $\boldsymbol{F}, \boldsymbol{F}, \boldsymbol{F}, \boldsymbol{F}$ na coluna do antepenúltimo componente.

6. Prosseguir dessa forma se houver mais componentes, sempre dobrando o número de $\boldsymbol{V}$ e $\boldsymbol{F}$ a cada coluna à esquerda.

Tabela 11 – Linhas da tabela-verdade proposta

| | $p$ | $q$ | $p$ | $\rightarrow$ | $(q$ | $\wedge$ | $\sim$ | $p)$ | $\leftrightarrow$ | $\sim$ | $(p$ | $\vee$ | $q)$ | $\wedge$ | $(\sim$ | $p$ | $\veebar$ | $q)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 8 | | 2 | 1 | | 9 | 6 | | 3 | | 7 | 4 | | 5 | |
| 1 | **V** | **V** | | | | | | | | | | | | | | | | |
| 2 | **V** | **F** | | | | | | | | | | | | | | | | |
| 3 | **F** | **V** | | | | | | | | | | | | | | | | |
| 4 | **F** | **F** | | | | | | | | | | | | | | | | |

Fonte: Elaborada pelo autor.

Aplicar as regras específicas das operações.

Tabela 12 – Resolução da tabela-verdade proposta

| | $p$ | $q$ | $p$ | $\rightarrow$ | $(q$ | $\wedge$ | $\sim$ | $p)$ | $\leftrightarrow$ | $\sim$ | $(p$ | $\vee$ | $q)$ | $\wedge$ | $(\sim$ | $p$ | $\veebar$ | $q)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 8 | | 2 | 1 | | 9 | 6 | | 3 | | 7 | 4 | | 5 | |
| 1 | **V** | **V** | | | | | **F** | | | | | | | | | | | |
| 2 | **V** | **F** | | | | | **F** | | | | | | | | | | | |
| 3 | **F** | **V** | | | | | **V** | | | | | | | | | | | |
| 4 | **F** | **F** | | | | | **V** | | | | | | | | | | | |

| | $p$ | $q$ | $p$ | $\rightarrow$ | $(q$ | $\wedge$ | $\sim$ | $p)$ | $\leftrightarrow$ | $\sim$ | $(p$ | $\vee$ | $q)$ | $\wedge$ | $(\sim$ | $p$ | $\veebar$ | $q)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 8 | | 2 | 1 | | 9 | 6 | | 3 | | 7 | 4 | | 5 | |
| 1 | **V** | **V** | | | | **F** | **F** | | | | | | | | | | | |
| 2 | **V** | **F** | | | | **F** | **F** | | | | | | | | | | | |
| 3 | **F** | **V** | | | | **V** | **V** | | | | | | | | | | | |
| 4 | **F** | **F** | | | | **F** | **V** | | | | | | | | | | | |

| | $p$ | $q$ | $p$ | $\rightarrow$ | $(q$ | $\wedge$ | $\sim$ | $p)$ | $\leftrightarrow$ | $\sim$ | $(p$ | $\vee$ | $q)$ | $\wedge$ | $(\sim$ | $p$ | $\veebar$ | $q)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 8 | | 2 | 1 | | 9 | 6 | | 3 | | 7 | 4 | | 5 | |
| 1 | **V** | **V** | | | | **F** | **F** | | | | | **V** | | | | | | |
| 2 | **V** | **F** | | | | **F** | **F** | | | | | **V** | | | | | | |
| 3 | **F** | **V** | | | | **V** | **V** | | | | | **V** | | | | | | |
| 4 | **F** | **F** | | | | **F** | **V** | | | | | **F** | | | | | | |

| | $p$ | $q$ | $p$ | $\rightarrow$ | $(q$ | $\wedge$ | $\sim$ | $p)$ | $\leftrightarrow$ | $\sim$ | $(p$ | $\vee$ | $q)$ | $\wedge$ | $(\sim$ | $p$ | $\veebar$ | $q)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 8 | | 2 | 1 | | 9 | 6 | | 3 | | 7 | 4 | | 5 | |
| 1 | **V** | **V** | | | | **F** | **F** | | | | | **V** | | | **F** | | | |
| 2 | **V** | **F** | | | | **F** | **F** | | | | | **V** | | | **F** | | | |
| 3 | **F** | **V** | | | | **V** | **V** | | | | | **V** | | | **V** | | | |
| 4 | **F** | **F** | | | | **F** | **V** | | | | | **F** | | | **V** | | | |

| | $p$ | $q$ | $p \rightarrow$ | $(q \wedge$ | $\sim p)$ | $\leftrightarrow$ | $\sim$ | $(p \vee q)$ | $\wedge$ | $(\sim p$ | $\veebar q)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 8 | 2 | 1 | 9 | 6 | 3 | 7 | 4 | 5 |
| 1 | **V** | **V** | | **F** | **F** | | | **V** | | **F** | **V** |
| 2 | **V** | **F** | | **F** | **F** | | | **V** | | **F** | **F** |
| 3 | **F** | **V** | | **V** | **V** | | | **V** | | **V** | **F** |
| 4 | **F** | **F** | | **F** | **V** | | | **F** | | **V** | **V** |

| | $p$ | $q$ | $p \rightarrow$ | $(q \wedge$ | $\sim p)$ | $\leftrightarrow$ | $\sim$ | $(p \vee q)$ | $\wedge$ | $(\sim p$ | $\veebar q)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 8 | 2 | 1 | 9 | 6 | 3 | 7 | 4 | 5 |
| 1 | **V** | **V** | | **F** | **F** | | **F** | **V** | | **F** | **V** |
| 2 | **V** | **F** | | **F** | **F** | | **F** | **V** | | **F** | **F** |
| 3 | **F** | **V** | | **V** | **V** | | **F** | **V** | | **V** | **F** |
| 4 | **F** | **F** | | **F** | **V** | | **V** | **F** | | **V** | **V** |

| | $p$ | $q$ | $p \rightarrow$ | $(q \wedge$ | $\sim p)$ | $\leftrightarrow$ | $\sim$ | $(p \vee q)$ | $\wedge$ | $(\sim p$ | $\veebar q)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 8 | 2 | 1 | 9 | 6 | 3 | 7 | 4 | 5 |
| 1 | **V** | **V** | | **F** | **F** | | **F** | **V** | **F** | **F** | **V** |
| 2 | **V** | **F** | | **F** | **F** | | **F** | **V** | **F** | **F** | **F** |
| 3 | **F** | **V** | | **V** | **V** | | **F** | **V** | **F** | **V** | **F** |
| 4 | **F** | **F** | | **F** | **V** | | **V** | **F** | **V** | **V** | **V** |

| | $p$ | $q$ | $p \rightarrow$ | $(q \wedge$ | $\sim p)$ | $\leftrightarrow$ | $\sim$ | $(p \vee q)$ | $\wedge$ | $(\sim p$ | $\veebar q)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 8 | 2 | 1 | 9 | 6 | 3 | 7 | 4 | 5 |
| 1 | **V** | **V** | **F** | **F** | **F** | | **F** | **V** | **F** | **F** | **V** |
| 2 | **V** | **F** | **F** | **F** | **F** | | **F** | **V** | **F** | **F** | **F** |
| 3 | **F** | **V** | **V** | **V** | **V** | | **F** | **V** | **F** | **V** | **F** |
| 4 | **F** | **F** | **V** | **F** | **V** | | **V** | **F** | **V** | **V** | **V** |

| | $p$ | $q$ | $p \rightarrow$ | $(q \wedge$ | $\sim p)$ | $\leftrightarrow$ | $\sim$ | $(p \vee q)$ | $\wedge$ | $(\sim p$ | $\veebar q)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 8 | 2 | 1 | 9 | 6 | 3 | 7 | 4 | 5 |
| 1 | **V** | **V** | **F** | **F** | **F** | **V** | **F** | **V** | **F** | **F** | **V** |
| 2 | **V** | **F** | **F** | **F** | **F** | **V** | **F** | **V** | **F** | **F** | **F** |
| 3 | **F** | **V** | **V** | **V** | **V** | **F** | **F** | **V** | **F** | **V** | **F** |
| 4 | **F** | **F** | **V** | **F** | **V** | **V** | **V** | **F** | **V** | **V** | **V** |

Fonte: Elaborada pelo autor.

## FINALIZANDO

Nesta aula foram apresentados os conteúdos que descrevem as operações lógicas sobre as proposições. Foram também apresentadas as regras para a construção das tabelas-verdade.

## REFERÊNCIAS


ABAR, C. A. A. P. **Noções de lógica matemática**. São Paulo: Ed. PUC-SP, 2011.

BOROWSKI, B. S. **Truth table constructor**. *Software*. Versão 3.0. [S.l.], jul. 2013. Disponível em: <http://www.brian-borowski.com/software/truth/TruthTableConstructor.jar>. Acesso em: 14 nov. 2018.

COPI, I. M. **Introdução à lógica**. São Paulo: Mestre Jou, 1968.

COPPIN, B. **Inteligência artificial**. Rio de Janeiro: LTC, 2017.

DOPP, J. **Noções de lógica formal**. São Paulo: Herder, 1970.

LUGER, G. F. **Inteligência artificial**. 6. ed. São Paulo: Pearson, 2013.

NOLT, J.; ROHATYN, D. **Lógica**. São Paulo: McGraw-Hill, 1991.

TANENBAUM, A. S. **Organização estruturada de computadores**. 6. ed. São Paulo: Pearson, 2013.