

O documento "codigo" consiste em um **script Python** projetado para ser executado no ambiente Google Colab, com o objetivo de **demonstrar as operações lógicas fundamentais** (Conjunção - AND, Disjunção - OR, e Negação - NOT) através da **geração de tabelas-verdade**.

Este código serve como uma **implementação prática dos conceitos teóricos** amplamente discutidos no "LIVRO LOGICACOMP.pdf", que aborda os fundamentos da lógica computacional. Componentes e Funcionalidades do Código:

### 1. Valores Lógicos Binários:

- O script utiliza `True` (Verdadeiro) e `False` (Falso) para representar os **valores binários 1 e 0**. Essa representação é central na **Álgebra Booleana**, que trata a lógica como um cálculo utilizando esses dois dígitos [62, 63, aula 2], e é a base das proposições na lógica.

### 2. Definição de Funções para Operações Lógicas:

- **Conjunção (AND - E):** A função `operacao_and(a, b)` retorna `True` (Verdadeiro) **somente se ambas as proposições** (`a` e `b`) forem `True`. Caso contrário, retorna `False`. Este comportamento reflete a regra da conjunção, onde "A AND B é Verdadeiro (V) somente se A e B forem Verdadeiros (V)", conforme detalhado na "LIVRO LOGICACOMP.pdf".

- **Disjunção (OR - OU):** A função `operacao_or(a, b)` retorna `False` (Falso) **somente se ambas as proposições** (`a` e `b`) forem `False`. Em qualquer outro caso, retorna `True`. Esta lógica é consistente com a definição de disjunção inclusiva, onde "A OR B é Falso (F) somente se A e B forem Falsos (F)".

- **Negação (NOT - NÃO):** A função `operacao_not(a)` **inverte o valor lógico** da proposição `a`. Se `a` é `True`, retorna `False`; se `False`, retorna `True`. Esta funcionalidade está de acordo com o operador de negação, que "tem a função de inverter, seja uma entrada ou o resultado de uma operação", conforme explicado na "LIVRO LOGICACOMP.pdf".

### 3. Geração de Tabelas-Verdade:

- O script utiliza a biblioteca `prettytable` para exibir as tabelas-verdade de forma organizada.

- Para cada operação lógica (AND, OR, NOT), o código gera uma tabela-verdade que mostra todas as **combinações possíveis de entradas** e seus respectivos resultados.

- Para duas proposições, o código demonstra que há  $2^2 = 4$  combinações possíveis, o que está em linha com a regra geral  $2^n$  para o número de linhas de uma tabela-verdade, onde `n` é o número de proposições.

Relação com os Conceitos do "LIVRO LOGICACOMP.pdf":

O script "codigo" ilustra de maneira prática diversos conceitos fundamentais da lógica computacional apresentados no "LIVRO LOGICACOMP.pdf":

- **Álgebra Booleana:** O uso dos valores `True` e `False` e a implementação dos operadores AND, OR e NOT são a essência da Álgebra Booleana, desenvolvida por George Boole no Período Booleano [62, 63, aula 2].

- **Proposições e Conectivos Lógicos:** O código manipula proposições (representadas pelas variáveis `a` e `b`) e aplica conectivos lógicos (AND, OR, NOT), que são os blocos construtivos da lógica proposicional. Os símbolos utilizados no código ( $\wedge$ ,  $\vee$ ,  $\neg$ ) são os mesmos descritos nas fontes.

- **Tabelas-Verdade:** A principal saída do script são as tabelas-verdade, uma ferramenta fundamental para "analisar TODOS os resultados possíveis" de uma operação lógica. Importância e Aplicação:

A conclusão do próprio script reforça que essas **tabelas-verdade são fundamentais para entender o comportamento das operações lógicas e como elas são aplicadas na construção de algoritmos e sistemas computacionais**. Essa afirmação é um ponto chave no

"LIVRO LOGICACOMP.pdf", que reitera repetidamente que a lógica é a **base para a construção de algoritmos eficazes** e para o desenvolvimento de sistemas digitais e inteligentes