

Aplicações da Tabela Verdade

Eduardo Furlan Miranda

2024-08-01

Baseado em: SCHEFFER, VC; VIEIRA, G; LIMA, TPFS. Lógica Computacional. EDE, 2020. ISBN 978-85-522-1688-9.

- Assim como as fórmulas matemáticas, podemos construir expressões lógicas mais complexas a partir da combinação das proposições, dos conectivos e dos parênteses
- Precedência dos conectivos lógicos
 - 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” e disjunção “OU”)
 - 4. \rightarrow (Implicação)
 - 5. \leftrightarrow (Bicondicional)

- Ao seguir rigorosamente a ordem de precedência dos operadores,
- o uso de parênteses pode ser omitido nos casos adequados
- A fórmula $A \vee (\neg B)$ pode ser escrita como $A \vee \neg B$,
- uma vez que, de acordo com a ordem de precedência, a negação será realizada primeiro

Exemplo 1

Quadro 4.13 | Tabela Verdade das 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 |

- Como o operador de conjunção tem precedência sobre a implicação, chegamos em uma tautologia
- Não ocorreu quando forçamos, por meio de parênteses, a implicação ser efetuada primeiro

Exemplo 1 (continuação)

- Usamos a coluna C3 para efetuar a primeira operação da fórmula $A \wedge B \rightarrow A$, e na coluna C4 usamos o resultado obtido em C3 para fazer a implicação
- O mesmo acontece na coluna C5, em que fazemos a primeira operação da fórmula $A \vee (B \rightarrow A)$, (considerando as regras da ordem de precedência) e depois usamos o resultado de C5 para fazer a conjunção final em C6
- Os resultados intermediários podem ser representados por novas letras, por exemplo: poderíamos chamar o resultado da coluna C3 de P, então na coluna C4 teríamos $P \rightarrow A$

- Da mesma forma, poderíamos chamar o resultado da coluna C5 de R, então na coluna C6 teríamos a fórmula $A \wedge R$
- Dada uma fórmula com várias proposições, conectores e parênteses dentro de parênteses, a resolução deve começar pelos parênteses mais internos.
- Por exemplo, a fórmula $((A \vee B) \rightarrow C) \wedge A$ deve ter a ordem de resolução:
 - 1 - $A \vee B$ (parênteses mais internos)
 - 2 - $((A \vee B) \rightarrow C)$ (parênteses mais externo)
 - 3 - $((A \vee B) \rightarrow C) \wedge A$ (operação fora dos parênteses)

Exemplo 2

Quadro 4.15 | 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 |

Exemplo 2 (continuação)

- Usamos proposições intermediárias para nomear os resultados (P, Q, R)
- $P (A \vee B) = \text{resultado 1}$
- $Q (P \rightarrow C) = \text{resultado 2}$
- $R (Q \wedge A) = \text{resultado final}$

- Todas essas operações e regras no universo da programação vamos utilizar para construir uma sequência de instruções, chamada de algoritmo
- Um algoritmo é uma sequência de passos que soluciona algum problema de diversas áreas do mundo real
- As operações lógicas são usadas em estruturas condicionais (ou estruturas de decisão) e têm o objetivo de realizar testes alterando o fluxo de execução de um programa, de acordo com a resposta obtida

Exemplo 3

- Você acabou de se mudar e está realizando a busca de um apartamento em um site de aluguel de imóveis.
- O site oferece uma interface na qual você clica nas opções que deseja:

| | | | | |
|--------------------------------|---------------------------------------|---|-------------------------------------|----------------------------------|
| Tipo de imóvel | <input type="button" value="Kitnet"/> | <input checked="" type="button" value="Apartamento"/> | <input type="button" value="Casa"/> | |
| Quantidade de dormitórios | <input type="button" value="1"/> | <input type="button" value="2"/> | <input type="button" value="3"/> | <input type="button" value="4"/> |
| Quantidade de banheiros | <input type="button" value="1"/> | <input type="button" value="2"/> | <input type="button" value="3"/> | <input type="button" value="4"/> |
| Quantidade de vagas na garagem | <input type="button" value="0"/> | <input type="button" value="1"/> | <input type="button" value="2"/> | <input type="button" value="3"/> |

- Vamos fazer algumas simulações para avaliarmos a lógica por trás da nossa seleção
- Simulação 1: Selecionar apartamento; 1 dormitório; 1 banheiro; sem vaga de garagem
- A expressão lógica que será construída com base nessa simulação é:
- Apartamento E 1 quarto E 1 banheiro E sem garagem
- Serão exibidos na tela somente os imóveis que satisfazem todas essas alternativas; os demais serão ignorados, pois no algoritmo a instrução é composta pelo conector de conjunção

- Simulação 2: Selecionar Apartamento; 1, 2 dormitórios; 1, 2 banheiros; 0, 1 vaga de garagem
- Nesse caso a expressão lógica será:
- Apartamento E (1 quarto OU 2 quartos) E (1 banheiro OU 2 banheiros)
- E (0 OU 1 vaga de garagem)

- Nesse caso a expressão é mais complexa e envolve a resolução de parênteses internos
- Como resultado da busca apareceriam tanto opções com 1 quanto com 2 quartos e banheiros e também sem garagem ou com 1 vaga
- Todas essas opções devem estar implementadas no algoritmo por meio das estruturas condicionais
- Ao finalizar a compra, o algoritmo faz uma conjunção dos itens que colocamos no carrinho de compras, para calcular o preço final

Exemplo 4

- Vamos utilizar uma base com dados reais e, a partir de uma Tabela Verdade, veremos como classificar os registros
- Trata-se de uma base com os preços de combustíveis em algumas cidades do Brasil no ano de 2019, segregados por estado, cidade, estabelecimento de revenda, produto, data da pesquisa, valor de venda, valor de compra e bandeira do posto

Tabela 4.6 | Amostra dos dados de combustíveis automotivos

| | Estado | Município | Revenda | Produto | Data da Coleta | Valor de Venda | Valor de Compra | Bandeira |
|---|--------|-------------------|---|---------------|----------------|----------------|-----------------|----------|
| 1 | DF | BRASÍLIA | ABRITTA POSTOS DE SERVIÇOS LTDA. | DIESEL S10 | 23/01/2019 | 3,899 | | xx |
| 2 | DF | BRASÍLIA | ABRITTA POSTOS DE SERVIÇOS LTDA. | ETANOL | 16/01/2019 | 3,299 | 2,9999 | xx |
| 3 | RJ | RIO DE JANEIRO | 3POSTO JULIO DE CAS- TILHO LIMITA- DA. | GASOLI- NA | 07/03/2019 | 5,096 | 4,305 | xx |

| | | | | | | | | |
|---|----|-------------------|--|---------------|------------|-------|--------|----|
| 4 | DF | BRASÍLIA | ABRITTA POSTOS DE SERVIÇOS LTDA. | GASOLI- NA | 23/01/2019 | 4,299 | 3,6745 | xx |
| 5 | RJ | RIO DE JANEIRO | ABASTE- CIMEN- TO DE COMBUS- TIVEIS RIO DO A LTDA. | ETANOL | 03/01/2019 | 3,499 | 2,9711 | xx |
| 6 | DF | BRASÍLIA | AM COMER- CIAL DE COMBUS- TIVEIS LTDA. | ETANOL | 23/01/2019 | 3,197 | 2,9416 | xx |

| | | | | | | | | |
|---|----|---------------|---|---------------|------------|-------|-------|----|
| 7 | DF | BRASÍLIA | AM COMER- CIAL DE COMBUS- TIVEIS LTDA. | GASOLI- NA | 08/01/2019 | 4,187 | 3,735 | xx |
| 8 | SC | BLUME- NAU | AUTO POSTO 7 LTDA. | DIESEL S10 | 02/01/2019 | 3,597 | | xx |
| 9 | SC | BLUME- NAU | AUTO POSTO 7 LTDA. | GASOLI- NA | 12/02/2019 | 3,767 | | xx |

- Dadas as seguintes proposições:
 - A: O município é Brasília
 - B: O combustível é gasolina
 - C: A bandeira é Petrobras
- Vamos construir a Tabela Verdade para a regra “Se a cidade não for Brasília e o combustível for gasolina então a bandeira é Petrobras”
- Em seguida, vamos fazer a conexão entre a lógica formal com a lógica computacional

- O primeiro passo é traduzir a regra para uma fórmula lógica, o que nos resulta em $(\neg A \wedge B) \rightarrow C$
- O resultado da Tabela Verdade para a fórmula está no Quadro 4.16 (próximo slide)
- A primeira operação feita foi a negação, que está dentro dos parênteses (seguindo a ordem de precedência dos operadores), depois a conjunção (\wedge), e chamamos o resultado de R
- Usamos esse resultado R para fazer a implicação (\rightarrow)

Quadro 4.16 | Tabela Verdade para a fórmula $(\neg A \wedge B) \rightarrow C$

| | | | | R | |
|---|---|---|----------|-------------------|-------------------|
| A | B | C | $\neg A$ | $\neg A \wedge B$ | $R \rightarrow C$ |
| V | V | V | F | F | V |
| V | V | F | F | F | V |
| V | F | V | F | F | V |
| V | F | F | F | F | V |
| F | V | V | V | V | V |
| F | V | F | V | V | F |
| F | F | V | V | F | V |
| F | F | F | V | F | V |

- a Tabela Verdade nos permite avaliar a veracidade da expressão testando todas as combinações possíveis
- Vamos interpretar alguns resultados do Quadro 4.16
- Na linha 1, temos a seguinte expressão “Se é falso que a cidade não é Brasília e o combustível é gasolina, então a bandeira é Petrobras”
- Tal expressão tem como resultado V, que se deve à falsidade no antecedente, impossibilitando avaliar o resultado que, portanto, é tomado como verdadeiro

- Na linha 2, a expressão é “Se é falso que a cidade não é Brasília e o combustível é gasolina, então é falso que a bandeira é Petrobras”
- Dada a falsidade no antecedente e consequente, a expressão tem como resultado V
- O único caso em que a implicação tem como resultado F, é quando o antecedente é verdadeiro e o consequente é falso

- Os operadores lógicos são usados na lógica de programação para dizer se uma condição é verdadeira ou falsa
- Nesse exemplo, queremos criar uma lógica de programação para avaliar regra “Se a cidade não for Brasília e o combustível for gasolina então a bandeira é Petrobras.”
- Pois bem, o Quadro 4.17 apresenta o resultado da análise das proposições A e B para os registros do Quadro 4.17, bem como da conjunção $\neg A \wedge B$

Quadro 4.17 | Valoração de A e B

| | Município | Produto | | Resultado |
|---|-----------|---------|---------------------|-------------------|
| | A | B | $\leftrightarrow A$ | $\neg A \wedge B$ |
| 1 | V | F | F | F |
| 2 | V | F | F | F |
| 3 | F | V | V | V |
| 4 | V | V | F | F |
| 5 | F | F | V | F |
| 6 | V | F | F | F |
| 7 | V | V | F | F |
| 8 | F | F | V | F |
| 9 | F | V | V | V |

- Dentro de um algoritmo computacional, o resultado da conjunção, dentro de uma estrutura condicional, pode ser usado para classificar a proposição C , pois onde a condição for satisfeita (V), a proposição C também será
- Na programação, o operador de implicação é feito por meio do comando `se... então...`
- No nosso exemplo podemos dizer `se $\neg A \wedge B$ então C` , ou seja, onde $\neg A \wedge B$ for V , C também será
- Para matar um pouco da nossa vontade de começar a programar, veja no Quadro 4.18 como ficaria esse comando dentro de três importantes linguagens de programação

Quadro 4.18 | Estrutura condicional em linguagens de programação

| 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> |

- Como você pode observar no Quadro 4.18, 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
- Além desses conectivos lógicos, também foram usados operadores relacionais

- A sintaxe \neq significa “diferente” e a sintaxe $=$ significa “igual”
- Como você pode ver, a solução geral de um problema está na lógica, que não muda de uma linguagem para outra, o que muda é a sintaxe