Fundamentos

Lógica Proposicional Booleana

Prof. Edson Alves
Faculdade UnB Gama

Termos primitivos e axiomas

A Lógica Proposicional Booleana é construída a partir de dois axiomas fundamentais, que relaciona os termos primitivos **proposição**, **verdadeiro** e **falso**:

1. **Princípio do Terceiro Excluído:** toda proposição ou é verdadeira ou é falsa;
2. **Princípio da Não Contradição:** uma proposição não poder ser, simultaneamente, verdadeira e falsa.

Proposições simples e compostas

- As proposições são representadas por letras minúsculas (por exemplo, $p, q, r, \ldots$)
- Duas ou mais proposições simples podem ser combinadas por meio de conectivos lógicos para formar proposições compostas
- Proposições compostas são representadas por letras maiúsculas (por exemplo, $P, Q, R, \ldots$)
- Proposições compostas também podem ser combinadas por meio de conectivos lógicos

Conectivos lógicos

Conectivo	Notação	Valor Lógico
Conjunção (e)	$p \wedge q$	verdadeira apenas quando ambas p e q são verdadeiras
Disjunção (ou)	$p \vee q$	falsa apenas quando ambas p e q são falsas
Disjunção exclusiva (*xor*)	$p \veebar q$	verdadeira apenas quando p e q tem valores lógicos opostos

Conectivos lógicos

Conectivo	Notação	Valor Lógico
Condicional (se, então)	$p \rightarrow q$	falsa apenas quando p é verdadeira e q é falsa
Bicondicional (se, e somente se)	$p \leftrightarrow q$	verdadeira apenas quando p e q tem mesmo valor lógico
Negação (não)	$\neg p$	inverte o valor lógico de p

Leis de Morgan

Sejam p e q duas proposições. Vale que

$$\neg(p \wedge q) = \neg p \ \vee \neg q$$

e que

$$\neg(p \vee q) = \neg p \ \wedge \neg q$$

C e C++

- As linguagens C e C++ tem, em sua sintaxe, operadores lógicos relacionais, que representam os conectivos lógicos
- Também há suporte para operadores lógicos *bit* a *bit*, que aplicam cada operação lógica aos pares de *bits* correspondentes
- Importante notar que, embora a linguagem C++ tenha um tipo de dado booleano, ambas linguagens interpretam como verdadeiro qualquer valor inteiro diferente de zero, e o zero como falso

Operadores lógicos em C/C++

Operador	Aridade	Símbolo em C	Palavra reservada em C++
e	binário	`&&`	`and`
ou	binário	`||`	`or`
não	unário	`!`	`not`

Operadores lógicos em C/C++

- Observe que apenas 3 dos conectivos lógicos tem símbolos ou palavras reservadas equivalentes em C e C++
- Isto se dá porque é possível, a partir destes três, definir quaisquer um dos demais conectivos
- Por exemplo, $p \rightarrow q$ é logicamente equivalente a $\neg p \vee q$
- Estes operadores são curto-circuito: se o valor lógico da primeira proposição é suficiente para definir o valor da expressão, a segunda proposição não é avaliada

Operadores *bit* a *bit* em C/C++

Operador	Aridade	Símbolo
e	binário	`&`
ou	binário	`|`
ou exclusivo	binário	`^`
não	unário	`~`

Exemplo de operadores *bit* a *bit* em C/C++

```c
int main()
{
    unsigned char a = 46;          //  46 = 00101110
    unsigned char b = 151;         // 150 = 10010111

    unsigned char c = a & b;       //   6 = 00000110
    unsigned char d = a | b;       // 191 = 10111111
    unsigned char e = a ^ b;       // 185 = 10111001
    unsigned char f = ~a;          // 209 = 11010001
    unsigned char g = ~b;          // 104 = 01101000

    return 0;
}
```

Problemas

- AtCoder
  1. ABC 097A - Colorful Transceivers
  2. ABC 117D - XXOR
  3. ABC 148A - Round One
- Codeforces
  1. 276D - Little Girl and Maximum XOR
- OJ
  1. 10718 - Bit Mask

Referências

1. **ALENCAR FILHO**, Edgard de. *Iniciação à Lógica Matemática*. São Paulo, Nobel, 2002.
2. **HALE**, Margie. *Essentials of Mathematics: Introduction to Theory, Proof, and the Professional Culture*. Mathematical Association of America, 2003.
3. **MORTARI**, Cezar A. *Introdução à Lógica*. Editora Unesp, 2ª edição, 2017.