Disciplina 1030760 - Lógica para Computação

Profa. Me Monica Pereira

Ementa da disciplina

A lógica matemática é amplamente utilizada em soluções algorítmicas e baseia todos os dispositivos digitais. Assim, os estudantes precisam compreender e aplicar conceitos de álgebra das proposições e álgebra booleana e seus teoremas, na resolução de problemas e implementação de programas.

Lógica

A lógica preocupa-se com a forma como uma proposição ou declaração é realizada, de acordo com:

a) Estudo do raciocínio;

b) Estudo do pensamento correto e verdadeiro;

c) Regras para demonstração científica verdadeira;

d) Regras para verificação de verdade ou falsidade de um pensamento.

Lógica

A lógica busca verificar a validade de um raciocínio para um determinado contexto ou realidade.

Premissa maior:

- Todo homem com mais de 100 quilos é pesado.

- Premissa menor:
  - João tem 120 quilos.
- Conclusão:
  - Portanto, João é pesado.

Lógica

A lógica busca verificar a validade de um raciocínio.
A lógica matemática é livre de contexto.

Premissa maior:

- Todo homem com mais de 100 quilos é pesado.

- Se João tem 120 quilos, então João é pesado.

Premissa menor

Conclusão

Lógica

p : João tem 120 quilos.
q : João está na Terra.
r : Então João é pesado.

$p \wedge q \to r$

Lógica

p : João tem 120 quilos.

q : João está na Lua.

r : Então João é leve.

p ∧ q -> r

Lógica

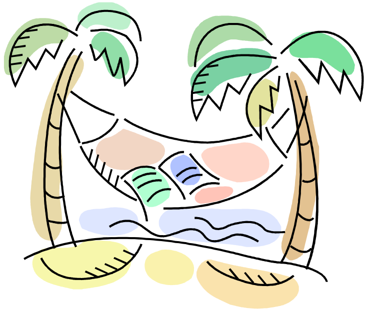
- Sintaxe:

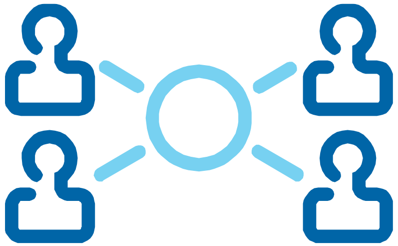
A sintaxe é relativa a representação e organização dos símbolos em uma linguagem.

A sintaxe é baseada em convenções.

- Semântica:

A semântica é formada pelos conceitos ligados a um contexto. Considere como exemplo a palavra REDE.

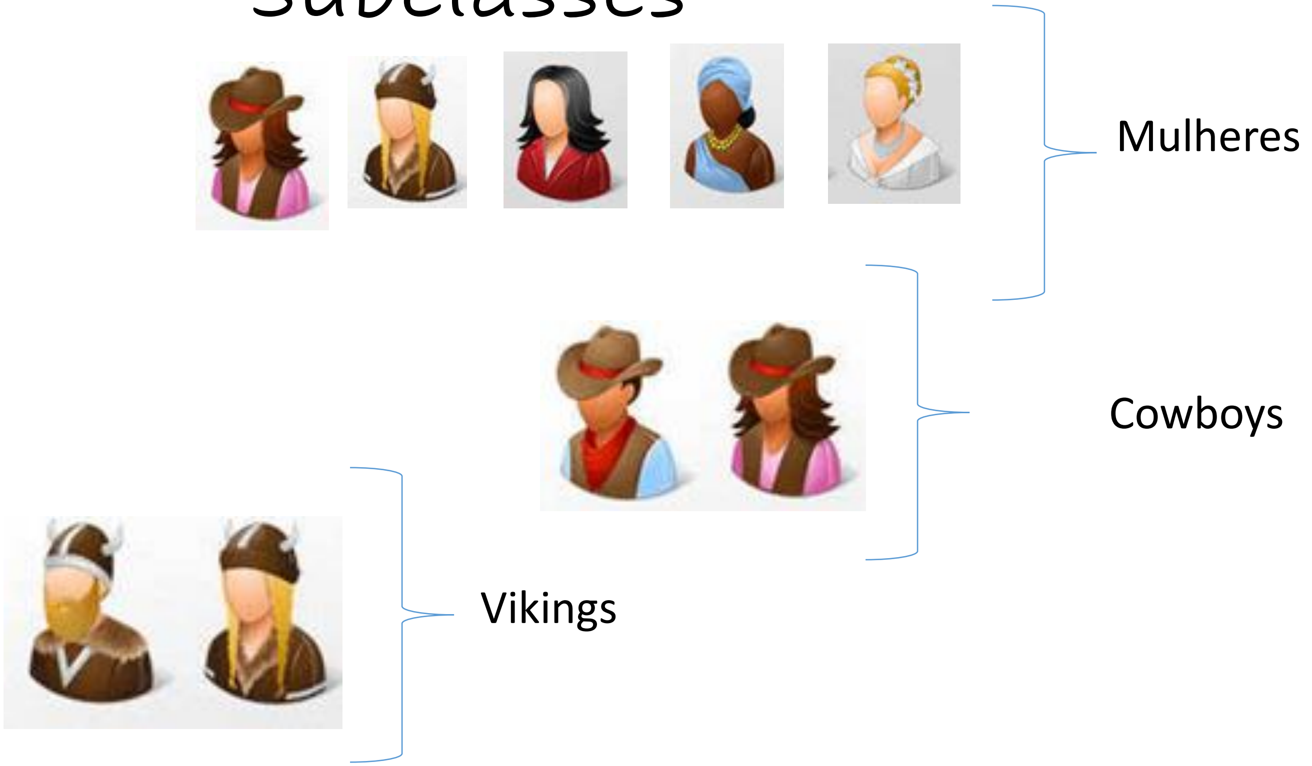




A lógica proposicional refere-se a classes ou conjuntos de coisas ou entes.

Classe: Pessoas

- Subclasses



A lógica proposicional refere-se a classes ou conjuntos de coisas ou entes.

Classe: Animais

Subclasses

Mamíferos

Aves

A lógica proposicional refere-se a classes ou conjuntos de coisas ou entes.

Conjunto: Números inteiros

$\{ \ldots -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5 \ldots \}$

Subconjunto: Números negativos ou não positivos

$\{ \ldots -5, -4, -3, -2, -1 \}$

Lógica proposicional

Proposição: é todo conjunto de palavras ou símbolos que exprimem um pensamento de sentido completo.

Proposição ⇔ Declaração ⇔ Sentença

Sujeito + Predicado

1) Está fazendo sol.
2) Está chovendo.
3) O quadrado tem três lados.
4) A lua não tem gravidade.
5) Jacarés tem dentes.
6) A luz está acesa.

Lógica proposicional

Proposições simples:	Proposições compostas:
Está chovendo.	A luz está acesa E não enxergo.
O céu está nublado.	É noite E não enxergo.
A luz está acesa.	Se está chovendo ENTÃO o céu está nublado.
Não enxergo.	Se faz sol OU a luz está acesa então Maria lê.
É noite.	
Faz sol.	
Maria lê.	

Lógica proposicional

Proposições simples:	Proposições compostas:
a = 2	a é positivo e c = 0
b < 5	a = 2 e c = 0
b = -3	d = -1 ou b < 5
c = 0	Se b < 5 e b = -3 então b é negativo
d = -1	
a é positivo	
c é positivo	
b é negativo	
d é negativo	

Lógica proposicional

A lógica matemática adota como regras fundamentais do pensamento dois princípios:

a) Princípio da não contradição: uma proposição não pode ser verdadeira ou falsa ao mesmo tempo.

b) Princípio do terceiro excluído: não existe um outro caso ou estado além de verdadeiro ou falso para uma proposição. Verifica-se sempre um destes casos e nunca um terceiro.

Algumas Aplicações da Lógica na Computação

a) Construção de linguagens formais
b) Compiladores
c) Construção de linguagens de programação (a partir de linguagens formais)
d) Algoritmos e programação
e) Banco de Dados (Linguagens formais como Álgebra Relacional e Cálculo Relacional – que dão origem ao SQL)
f) Programação em Lógica
g) Inteligência Artificial
h) Circuitos lógicos
i) Entre outros ...

A linguagem da Lógica Proposicional:

Alfabeto

a) A lógica é uma linguagem formal composta de:

Gramática

b) A lógica proposicional ocupa-se da forma como os argumentos são organizados, a fim de validar ou refutar uma fórmula.

A linguagem da Lógica Proposicional:

Alfabeto da lógica proposicional:

- Símbolos de pontuação: (,)
- Símbolos de verdade: V ou F ou *true* e *false*
- Proposições compostas: P, Q, R, S ...;
- Proposições simples: p, q, r, s ...;
- Conectivos proposicionais:
  - $\vee$ = ou;
  - $\wedge$ = e;
  - ~ = não;
  - -> = se ... então;
  - <-> = se e somente se

A linguagem da Lógica Proposicional:

SUJEITO

José é eleitor.

PREDICADO

Ana tem 10 anos.

Classe Pessoa

Maria é motorista.

Maria é idosa.

A linguagem da Lógica Proposicional:

Classe Z

$a > 5$

$a < 10$

$b = 30$

$c = -9$

$d < -5$

$d > -20$

A linguagem da Lógica Proposicional:

Fórmulas consistem em:

a) Um símbolo de verdade (true ou false);
b) Um símbolo proposicional ou uma proposição (P, Q);
c) Se as proposições P e Q são fórmulas então $P \wedge Q$ é uma fórmula;
d) Se as proposições P e Q são fórmulas então $P \vee Q$ é uma fórmula;
e) Se as proposições P e Q são fórmulas então $P \rightarrow Q$ é uma fórmula;
f) Se as proposições P e Q são fórmulas então P <-> Q é uma fórmula.

A linguagem da Lógica Proposicional:

Fórmulas

a) Conjunção: $p \wedge q$

b) Disjunção: $p \vee q$

c) Implicação: $p \rightarrow q$

d) Bi-implicação: p <-> q

A linguagem da Lógica Proposicional:

Considere as proposições p, q, r, s. Adota-se V para *true* e F para *false*.

a) (p ∨ q) → V

b) (q ^ p) → F

c) (p <-> q) <-> V

d) (~s ∨ ~r) → p

e) (~~s ∧ p)

f) r

g) F

A linguagem da Lógica Proposicional:

Precedência de conectivos:

- Maior precedência: ~
- Precedência intermediária: ^ e v
- Menor precedência: → e <->

Aplicações:

SELECT NOME, DATANASC FROM FUNCIONARIO WHERE

DATANASC > ‘01/01/1950’ AND DATANASC < ‘31/12/1980’;

SELECT NOME, ENDERECO, CIDADE, UF FROM FUNCIONARIO WHERE

CIDADE = ‘CURITIBA’ OR CIDADE = ‘PORTO ALEGRE’;

```
#include <stdio.h>
main ()
{
int idade;
char titulo;
char residente;
printf("digite a idade da pessoa:\n");
scanf("%i",&idade);
printf ("digite n ou s para informar se a pessoa é residente ou não:\n");
scanf("%s",&residente);
printf ("digite n ou s para informar se a pessoa tem título ou não:\n");
scanf("%s",&titulo);
          if (idade >= 16 && residente == 's' && titulo == 's')
                    printf("é eleitor\n");
                    else
                              if (idade < 16)
                              printf("é menor de idade\n");
                              else
                                        if ( residente == 'n');
                                        printf("não é residente\n");
                                        else
                                                  printf("não tem título\n");}
```

Entre as sentenças abaixo, indique quais não são fórmulas da lógica proposicional.

a) $(p \vee q \rightarrow u)$

b) $((p \vee q) \rightarrow \ u) \rightarrow \ F$

c) $\sim x \rightarrow u$

d) $((p \wedge q) \rightarrow V) \rightarrow F$

e) $t \wedge u$

f) $p \wedge q) \rightarrow V \rightarrow F$

g) $\rightarrow u \rightarrow \sim x$

h) $\sim p \wedge \sim q$

i) $\vee q$

j) $(v \ , \ q)$

k) $\sim p \wedge \sim q <-> x$

l) $<-> x$

m) $\sim q <->$

n) $x <-> \neg p$

A linguagem da Lógica Proposicional:

Crie fórmulas considerando as seguintes proposições:

p : imc < 18,5
q : imc >= 18,5
r : imc <= 25
s : imc >= 25
t : imc <= 30
u : imc > 30
v : Abaixo do peso
x : Peso normal
y : Acima do peso
z : Obeso

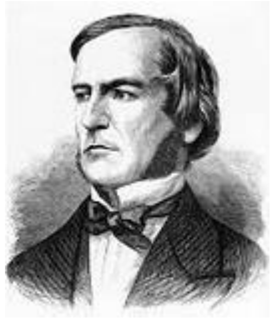
A linguagem da Lógica Proposicional:

Validação de pensamentos:

$(p <-> q) \rightarrow (q <-> p)$

$\sim(p \rightarrow \sim p)$

$((1 - q) \rightarrow p) \rightarrow (p + q) = 1$



Para George Boole o valor (1) representa a totalidade de uma classe ou conjunto, como por exemplo homens ou humanos. As variáveis p ou q representam subconjuntos da classe.

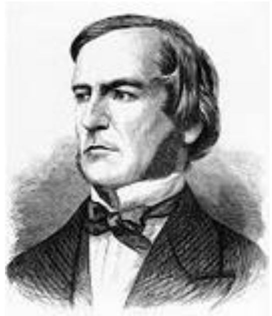
A linguagem da Lógica Proposicional:

Validação de pensamentos:

p = brasileiros nascidos em Chapecó

q = ~(brasileiros nascidos em Chapecó)

$$((1 - q) \rightarrow p) \rightarrow (p + q) = 1$$



Para George Boole o valor (1) representa a totalidade de uma classe ou conjunto, como por exemplo homens ou humanos. As variáveis p ou q representam subconjuntos da classe.

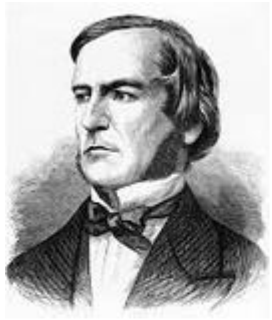
A linguagem da Lógica Proposicional:

Validação de pensamentos:

p = números inteiros positivos

q = ~(números inteiros positivos)

$$((1 - q) \rightarrow p) \rightarrow (p + q) = 1$$



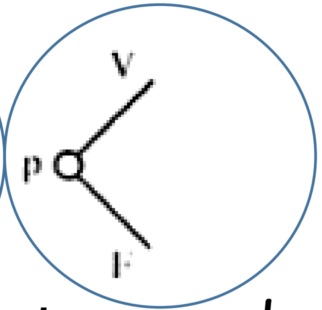
Para George Boole o valor (1) representa a totalidade de uma classe ou conjunto, como por exemplo homens ou humanos. As variáveis p ou q representam subconjuntos da classe.

Métodos de representação semântica:

Para uma proposição simples, somente dois valores são possíveis.

Tabela-verdade

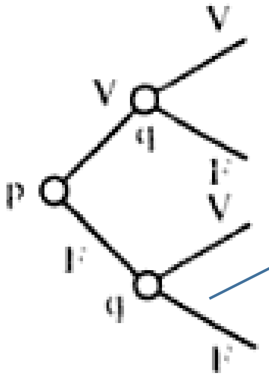
p
V
F



Árvore semântica

As proposições compostas dependem dos valores das proposições simples, de acordo com as combinações possíveis, para verdade e falsidade.

	p	q
1	V	V
2	V	F
3	F	V
4	F	F

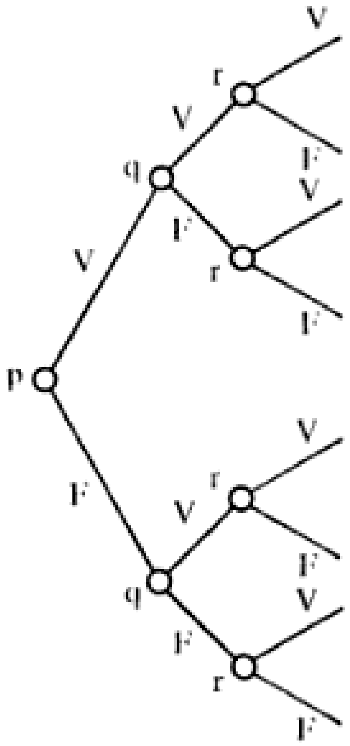


As arestas dos grafos representam o valor de V ou F de cada uma das proposições.

Tabela-Verdade

As proposições compostas dependem dos valores das proposições simples, de acordo com as combinações possíveis, para verdade e falsidade.

	p	q	r
1	V	V	V
2	V	V	F
3	V	F	V
4	V	F	F
5	F	V	V
6	F	V	F
7	F	F	V
8	F	F	F



As arestas dos grafos representam o valor de V ou F de cada uma das proposições.

Valor lógico de proposições

O valor lógico de uma proposição é dado por:
V(p), que pode assumir V ou F.

Se p for verdadeira então escreve-se V(p) = V.

Se p for falsa então escreve-se V(p) = F.

Valor lógico de proposições

Defina os valores lógicos das seguintes proposições:

p : Tartarugas tem três patas.
q : Árvores tem folhas.
r : A capital do Brasil é Buenos Aires.
s : O triângulo equilátero tem todos os lados de igual comprimento.
u : O triângulo equilátero tem todos lados e ângulos de mesma medida.
v : Todas as janelas tem quatro lados.
x : A água quente é formada por dois átomos de hidrogênio e um átomo de oxigênio.
y : Um losango tem seis lados.

Operações lógicas sobre proposições

Negação:

Dada uma proposição p, a sua negação é denotada como:

~p (não p).

Se V(p) = V

então

V(~p) = F

p	~p
V	F
F	V

Operações lógicas sobre proposições

Negação:

$p : 5 + 3 = 8$

Definindo os valores lógicos para p e ¬p.

$V(p) = V$

$V(\sim p\ ) = F$

Definindo os valores lógicos para p e ¬p.

$p : 5 + 3 \neq 8$

$V(p) = F$

$V(\sim p) = V$

p	~p
V	F
F	V

Operações lógicas sobre proposições

Negação:

p : O gato não tem quatro patas.

~p : É negado que o gato não tem quatro patas. (Não é verdade que o gato não tem quatro patas.)

Ou

p : O gato tem quatro patas.

~p : O gato não tem quatro patas.

Operações lógicas sobre proposições

Conjunção: é a sentença formada por duas proposições simples e pelo conectivo "e". Considere duas proposições p e q.

$$(V(p \wedge q) = V) <-> ((V(p) = V) \wedge (V(q) = V))$$

Isto é, o valor lógico de uma conjunção é V se, e somente se, ambas as proposições tiverem valores lógicos iguais a V(erdade).

p	q	p ^ q
V	V	V
V	F	F
F	V	F
F	F	F

Operações lógicas sobre proposições

Dado o conjunto $C = \{ y \mid y \in Z \wedge 2 > y < 5 \}$ e as proposições a seguir, indique o valor lógico das conjunções.

p : Está chovendo.

q : A rua está molhada.

$r : x = y + 9$

$s : x >= 9$

a) $p \wedge q$

b) $r \wedge s$

c) $p \wedge \sim q$

d) $r \wedge \sim s$

Operações lógicas sobre proposições

Disjunção: é a sentença formada por duas proposições simples e pelo conectivo “ou”. Considere duas proposições p e q.

Para que a disjunção seja verdadeira é preciso que uma das proposições simples tenha como valor lógico V(erdade).

Se $(V(p)=V) \vee (V(q)=V) \rightarrow (p \vee q) = V$

p	q	p ˅ q
V	V	V
V	F	V
F	V	V
F	F	F

Operações lógicas sobre proposições

Disjunção exclusiva: é a sentença formada por duas proposições simples e pelo conectivo "ou", no entanto quando uma das proposições for verdadeira a outra não pode ser. Considere assim, duas proposições p e q.

Para que a disjunção exclusiva seja verdadeira é preciso que somente uma das proposições simples tenha como valor lógico V(erdade). Observe os exemplos a seguir.

p : Vou ao cinema.

q : Vou ao teatro.

p	q	$p \underline{\vee} q$
V	V	F
V	F	V
F	V	V
F	F	F

Exercícios
Traduza as sentenças da lógica proposicional para a linguagem natural.

p : Juca tem um carro.
q : Paulo é pobre.
r : Juca não dirige.
s : Paulo é motorista.
u : Paulo tem muito dinheiro.
x : Maria vai a escola.
y : Maria vai a igreja.
z : Maria tem uma moto.

a) $p \wedge \sim r$
b) $s \wedge r$
c) $s \wedge \sim r$
d) $\sim s$
e) $\sim x \wedge y$
f) $\sim q \vee p$
g) $x \underline{\vee} y$
h) $z \wedge p \wedge s$
i) $\sim p \wedge r$
j) $\sim u \wedge q$
k) $u \vee q$
l) $z \wedge y \underline{\vee} z$

Exercícios
Traduza as sentenças da lógica proposicional para a linguagem natural e determine o seu valor lógico para o conjunto Z.

$p : x = 4$

$q : x = 5$

$r : y = -9$

$s : y < 0$

$u : z >= 0$

$m : z < 0$

a) $p \wedge \sim r$

b) $s \vee r$

c) $p \wedge q$

d) $r \wedge s$

e) $u \vee m$

f) $u \vee m$

g) $u \wedge m$

Operações lógicas sobre proposições

- **Condicional:** implicação. Sentença formada por uma proposição (p) chamada de antecedente e outra (q) chamada de consequente.
- $p \rightarrow q$ (se p então q)
- No entanto, p não determina o
- valor de q.
- p : Pedro tem 80 anos.
- q : Pedro é jovem.
- $p \rightarrow q$

p	q	$p \rightarrow q$
V	V	V
F	V	V
V	F	F
F	F	V

Operações lógicas sobre proposições

- **Bicondicional:** sentença formada por proposições p e q, de forma que p implica em q e q implica em p. Isto é, p
- p <-> q (p, se e somente se q)
- p é condição necessária para q e
- q é condição necessária para p
- p : Pedro tem 60 anos ou mais.
- q : Pedro é idoso.
- q <-> p

p	q	$p \leftrightarrow q$
V	V	V
F	V	F
V	F	F
F	F	V

Operações lógicas sobre proposições

- Bicondicional:
- p : Pedro tem 80 anos.
- q : Pedro é velho.

  p <-> q
- r : 5 > 3
- s : 3 < 5

  r <-> s

p	q	p ↔ q
V	V	V
F	V	F
V	F	F
F	F	V

Exercícios:

1. Sejam as proposições p : Está frio e q : Está chovendo. Traduzir para a **linguagem corrente** as seguintes proposições:

(a) $\sim p$	(b) $p \wedge q$	(c) $p \vee q$
(d) $q \leftrightarrow p$	(e) $p \rightarrow \sim q$	(f) $p \vee \sim q$
(g) $\sim p \wedge \sim q$	(h) $p \leftrightarrow \sim q$	(i) $p \wedge \sim q \rightarrow p$

5. Sejam as proposições p : Marcos é alto e q : Marcos é elegante. Traduzir para a **linguagem simbólica** as seguintes proposições:

(a) Marcos é alto e elegante
(b) Marcos é alto, mas não é elegante
(c) Não é verdade que Marcos é baixo ou elegante
(d) Marcos não é nem alto e nem elegante
(e) Marcos é alto ou é baixo e elegante
(f) É falso que Marcos é baixo ou que não é elegante

Fonte: Iniciação a lógica matemática de Edgard de Alencar Filho – página 25

Exercícios:

9. Traduzir para a **linguagem simbólica** as seguintes proposições matemáticas:

(a) $(x + y = 0$ e $z > 0)$ ou $z = 0$
(b) $x = 0$ e $(y + z > x$ ou $z = 0)$
(c) $x \neq 0$ ou $(x = 0$ e $y < 0)$
(d) $(x = y$ e $z = t)$ ou $(x < y$ e $z = 0)$

10. Traduzir para a **linguagem simbólica** as seguintes proposições matemáticas:

(a) Se $x > 0$ então $y = 2$
(b) Se $x + y = 2$ então $z > 0$
(c) Se $x = 1$ ou $z = 2$ então $y > 1$
(d) Se $z > 5$ então $x \neq 1$ e $x \neq 2$
(e) Se $x \neq y$ então $x + z > 5$ e $y + z < 5$
(f) Se $x + y > z$ e $z = 1$ então $x + y > 1$
(g) Se $x < 2$ então $x = 1$ ou $x = 0$
(h) $y = 4$ e se $x < y$ então $x < 5$

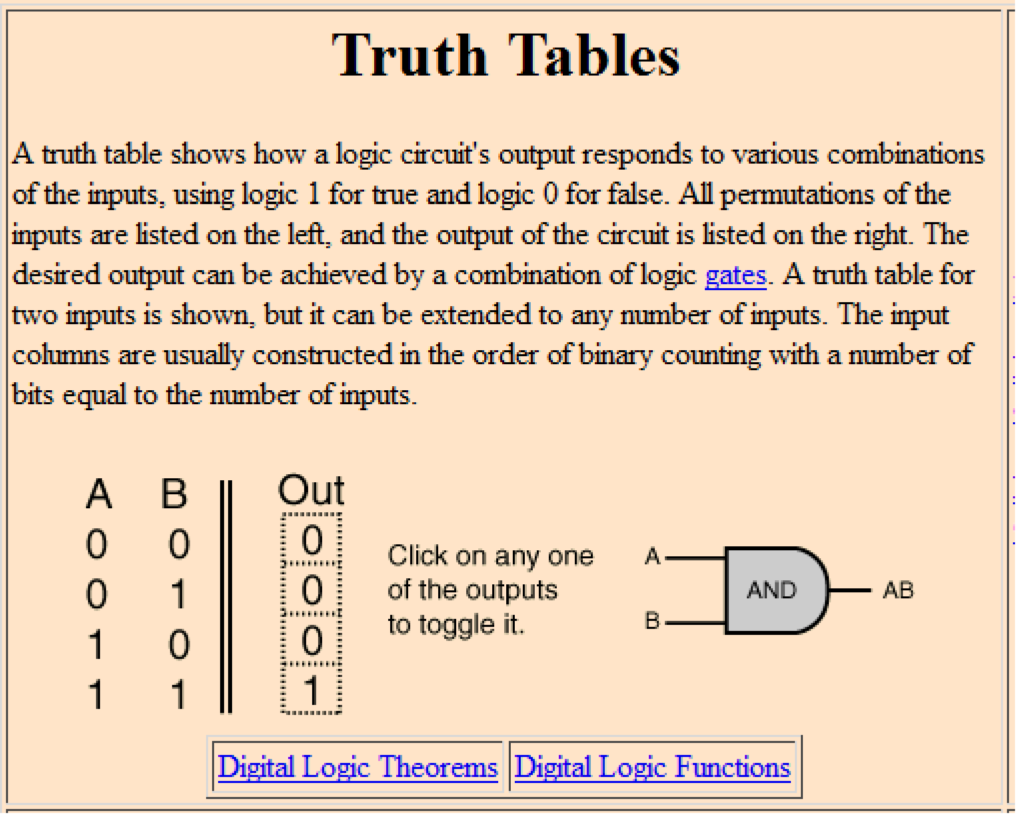
Fonte: Iniciação a lógica matemática de Edgard de Alencar Filho – página 26

Construção de tabelas-verdade

- Aplicação prática das tabelas-verdade

As tabelas-verdade são utilizadas na construção de circuitos lógicos. Neste caso, as entradas são representadas pelos valores lógicos 0 (zero para ausência de corrente) e 1 (um para presença de corrente).
As entradas são sempre definidas a esquerda da porta lógica e as saídas a direita da porta lógica.

http://hyperphysics.phy-astr.gsu.edu



Truth Tables

A truth table shows how a logic circuit's output responds to various combinations of the inputs, using logic 1 for true and logic 0 for false. All permutations of the inputs are listed on the left, and the output of the circuit is listed on the right. The desired output can be achieved by a combination of logic gates. A truth table for two inputs is shown, but it can be extended to any number of inputs. The input columns are usually constructed in the order of binary counting with a number of bits equal to the number of inputs.

A	B	Out
0	0	0
0	1	0
1	0	0
1	1	1

Click on any one of the outputs to toggle it.

Digital Logic Theorems | Digital Logic Functions

Construção de tabelas-verdade

- Aplicação prática das tabelas-verdade:
  - Determinar a equivalência de fórmulas diferentes para resolução de um mesmo problema.
  - Testar a validade da estrutura lógica de argumentos.

Construção de tabelas-verdade

- Dadas várias proposições simples (p, q, r ...) e o conjunto de conectivos da lógica matemática (~ , →, <->, v, ^), podemos construir proposições compostas e verificar o seu valor lógico através do método da tabela-verdade.
- Proposições compostas:
  - $P(p, q) = \sim p \vee (p \rightarrow q)$
  - $Q(p, q) = (p \leftrightarrow \sim q) \wedge q$
  - $R(p, q, r) = (p \wedge q) \rightarrow r$

Construção de tabelas-verdade

- A partir das tabelas-vedade das operações lógicas fundamentais:

$\sim p$, $p \rightarrow q$, $p <-> q$, $p \vee q$, $p \wedge q$

podemos obter os valores lógicos das proposições compostas.

  - $P(p, q) = \sim p \vee (p \rightarrow q)$
  - $Q(p, q) = (p <-> \sim q) \wedge q$
  - $R(p, q, r) = (p \wedge q) \rightarrow r$

A tabela-verdade de uma proposição composta com n proposições simples contém 2^n linhas.

UNOCHAPECÓ

Construção de tabelas-verdade

- Passos:

1) Conta-se o número de proposições simples e formam-se as colunas para cada proposição;
2) Para a primeira proposição simples atribui-se 2^{n-1} valores V(verdade) e 2^{n-1} valores F(falsidade);
3) Para a segunda proposição simples atribui-se 2^{n-2} valores V(verdade) e 2^{n-2} valores F(falsidade) e assim sucessivamente, de acordo com o número de proposições.

Construção de tabelas-verdade

- Montando a tabela verdade das proposições simples (p, q, r):

Para n = 3 , número de linhas da tabela-verdade = 8.

Atribuindo os valores lógicos:

a) Para p, 2^{3-1} valores V e 2^{3-1} valores F;

b) Para q, 2^{3-2} valores V e 2^{3-2} valores F;

c) Para r, 2^{3-3} valores V e 2^{3-3} valores F;

d) Da quarta coluna em diante, inserem-se as fórmulas, conforme o método adotado.

p	q	r	...
V	V	V	
V	V	F	
V	F	V	
V	F	F	
F	V	V	
F	V	F	
F	F	V	
F	F	F	

Construção de tabelas-verdade Método I

Tabela verdade para a proposição composta: $(\neg p \wedge r) <-> (q \vee \neg r)$

p	q	r	~p	~p ^ r	~r	q ~r	(~p ^ r) ↔ (q ˅ ~r)
V	V	V	F	F	F	V	F
V	V	F	F	F	V	V	F
V	F	V	F	F	F	F	V
V	F	F	F	F	V	V	F
F	V	V	V	V	F	V	V
F	V	F	V	F	V	V	F
F	F	V	V	V	F	F	F
F	F	F	V	F	V	V	F

Construção de tabelas-verdade Método II

Tabela verdade para a proposição composta: $(\sim p \wedge r) \leftrightarrow (q \vee \sim r)$

p	q	r	(~p	^	r)	↔	(q	v	~r)
V	V	V	F	F	V	F	V	V	F
V	V	F	F	F	F	F	V	V	V
V	F	V	F	F	V	V	F	F	F
V	F	F	F	F	F	F	F	V	V
F	V	V	V	V	V	V	V	V	F
F	V	F	V	F	F	F	V	V	V
F	F	V	V	V	V	F	F	F	F
F	F	F	V	F	F	F	F	V	V

Construção de tabelas-verdade
Método III

Tabela verdade para a proposição composta: (~p ∧ r) <-> (q ∨ ~r)

(~	p	^	r)	↔	(q	˅	~	r)
F	V	F	V	F	V	V	F	V
F	V	F	F	F	V	V	V	F
F	V	F	V	V	F	F	F	V
F	V	F	F	F	F	V	V	F
V	F	V	V	V	V	V	F	V
V	F	F	F	F	V	V	V	F
V	F	V	V	F	F	F	F	V
V	F	F	F	F	F	V	V	F

Construção de tabelas-verdade

Exercícios

Faça as tabelas-verdade das proposições a seguir:

a) $\sim(p \vee \sim q)$

b) $\sim(p \rightarrow \sim q)$

c) $p \wedge q \rightarrow p \vee q$

d) $\sim p \rightarrow (q \rightarrow p)$

e) $(p \rightarrow q) \rightarrow p \wedge q$

f) $q <-> \sim q \wedge p$

g) $\sim r \rightarrow p \wedge q$

h) $\sim((r \rightarrow p) \vee (s \rightarrow q))$

j) $(s <-> r) <-> (p <-> q)$

k) $(p \wedge q) <-> (r \wedge \sim s)$

Uso de parênteses nas proposições compostas

Usam-se parênteses para eliminar ambiguidades.

Exemplos:

- $p \wedge q \vee r$ - pode ser lida como $(p \wedge q) \vee r$ ou $p \wedge (q \vee r)$
- $p \wedge q \rightarrow r \vee s$ - pode ser lida como $(p \wedge q) \rightarrow (r \vee s)$ ou $((p \wedge q) \rightarrow r) \vee s$, e assim por diante.

Tautologias, Contradições e Contingências

Tautologia ou proposição logicamente verdadeira

É toda proposição composta que encerra em sua última coluna, somente valores V(erdade).

Se P(p, q, r ...) é tautologia em V(P)=V.

Princípio de Identidade

Todo objeto é igual a si mesmo.

p	p	$p \rightarrow p$	$p \leftrightarrow p$
V	V	V	V
F	F	V	V

Tautologias, Contradições e Contingências

Tautologia ou proposição logicamente verdadeira

É toda proposição composta que encerra em sua última coluna, somente valores V(erdade).

Se P(p, q, r ...) é tautologia em V(P)=V.

Princípio da Não contradição

p	~p	p ^ ~ p	~(p ^ ~ p)
V	F	F	V
F	V	F	V

Uma proposição não pode assumir os valores V(erdade) ou F(alsidade) ao mesmo tempo.

Tautologias, Contradições e Contingências

Tautologia ou proposição logicamente verdadeira

É toda proposição composta que encerra em sua última coluna, somente valores V(erdade).

Se P(p, q, r ...) é tautologia em V(P)=V.

Princípio do Terceiro Excluído

p	~p	p ᵛ ~ p
V	F	V
F	V	V

Uma proposição pode ter valor lógico V(erdade) ou F(alsidade).Qualquer outro valor é desconsiderado.

Tautologias, Contradições e Contingências

Tautologia ou proposição logicamente verdadeira

Princípio da Substituição:

- Dadas as proposições

$P(p, q, r \ldots)$, $P_0(p, q, r \ldots)$, $Q_0(p, q, r \ldots)$, $R_0(p, q, r \ldots)$

- Se $P(p, q, r \ldots)$ é uma tautologia então todas as proposições P_0, Q_0 e R_0 formada pelas proposições simples p, q e r também são tautologias.

Tautologias, Contradições e Contingências

Tautologia

Faça as tabelas-verdade das proposições:

$p \vee \sim (p \wedge q)$

$(p \wedge q) \rightarrow (p \leftrightarrow q)$

Tautologias, Contradições e Contingências

Contradição

Uma contradição é uma proposição composta que tem na sua tabela verdade, em sua última coluna somente os valores lógicos F(alsidade).

p ^ ~ p

p <-> ~ p

(p ^ q) ^ ~ (p v q)

- Dadas as proposições

  P(p, q, r ...), P_0(p, q, r ...), Q_0(p, q, r ...), R_0(p, q, r ...)

- Se P(p, q, r ...) é uma contradição então todas as proposições P_0, Q_0 e R_0 formada pelas proposições simples p, q e r também são contradições.

Tautologias, Contradições e Contingências

Contingência ou Satisfabilidade

É toda proposição composta que encerra em sua última coluna valores V(erdade) e F(alsidade), de forma que cada um deve aparecer pelo menos uma vez.

Assim, toda proposição que não é tautologia e também não é contradição, é uma proposição contingente ou proposição indeterminada.

Tautologias, Contradições e Contingências

Satisfabilidade

"Um conjunto de fórmulas {H_1, H_2, H_3 ...} é satisfatível quando existe pelo menos uma interpretação I, que interpreta as fórmulas H_1, H_2, H_3 como sendo igual a T(rue) ... Uma propriedade desejável dos programas lógicos, por exemplo, é que eles sejam conjuntos de fórmulas satisfatíveis. Isso significa que deve haver pelo menos uma interpretação que satisfaça o programa."

Pesquisar: complexidade computacional ou complexidade e algoritmos.

Fonte: SOUZA, João Nunes de. **Lógica para ciência da computação:** uma introdução concisa. 2. ed. rev. e atual. Rio de Janeiro: Elsevier, 2008. 220 p.

Tautologias, Contradições e Contingências

Exercícios:

Demonstrar que as seguintes proposições são tautológicas:

a) $(p \rightarrow p) \vee (p \rightarrow \sim p)$

b) $p \leftrightarrow p \wedge (p \vee q)$

c) $\sim (p \vee \sim p) \vee (q \vee \sim q)$

Tautologias, Contradições e Contingências

Exercícios:

Demonstrar que as seguintes proposições são contingentes:

a) $p \vee q \rightarrow p \wedge q$

b) $p \rightarrow (p \rightarrow q \wedge \sim q)$

Demonstrar que as seguintes proposições são tautológicas, contraválidas ou contingentes:

a) $\sim p \vee q \rightarrow (p \rightarrow q)$

b) $p \wedge q \rightarrow (p \leftrightarrow q \vee r)$

Implicação lógica (=>)

P => Q

Diz-se que uma proposição P(p, q, r ...) implica uma proposição Q(p, q, r ...) se todas as vezes que P for verdadeira Q também é verdadeira.

Demonstrar que q => p ^ q <-> p

Implicação lógica

Propriedades da implicação lógica:

Reflexividade:

P(p, q, r ...) => P(p, q, r ...)

Transitividade:

Se P(p, q, r ...) => Q(p, q, r ...) e

Q(p, q, r ...) => R(p, q, r ...) então

P(p, q, r ...) => R(p, q, r ...)

Implicação lógica

Demonstração da implicação das proposições:

$(p \wedge q)$, $(p \vee q)$ e $(p \leftrightarrow q)$

Isto é:

$(p \wedge q) \Rightarrow (p \vee q) \Rightarrow (p \leftrightarrow q)$

$(((p \wedge q) \rightarrow (p \vee q)) \wedge ((p \vee q) \rightarrow (p \leftrightarrow q))) \rightarrow ((p \wedge q) \rightarrow (p \leftrightarrow q))$

Implicação lógica

Regras de inferência:

Sempre que p tiver valor lógico igual a V(erdade), para qualquer proposição que seja acrescentada (adicionada) a operação “ou” (v) com p, a sentença
(p v q) será sempre verdadeira.

Adição

p	q	p v q	p → p v q
V	V	V	V
V	F	V	V
F	V	V	V
F	F	F	V

p => p v q

Implicação lógica

Regras de inferência: Simplificação

p	q	p ^ q	p ^ q → p
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	V

$p \wedge q \Rightarrow p$

$p \wedge q \Rightarrow q$

Na conjunção toda vez que p acontecer, q também deve acontecer, assim a simplificação é possível, pois a existência de um implica na existência do outro, para que a sentença seja verdadeira.

Implicação lógica

Verifique a implicação lógica das proposições:

a) $p <-> q$, $p \rightarrow q$, $q \rightarrow p$

b) $((x \neq 0 \rightarrow x = 0) \wedge x \neq y) \quad => \quad (x = 0)$

Obs.: para a solução das letra “b”, considere as proposições e sua negação.

c) $q => p \wedge q \rightarrow p$

Implicação lógica

Silogismo disjuntivo – argumento formado por três partes.

(modus tollendo ponens) Antagonismo.

Ou é dia ou é noite. (Se existe o dia, então existe a noite!!!!!)

Não é dia.

Portanto, é noite.

$(p \vee q) \wedge \sim p \Rightarrow q$

Se $((p \vee q) \wedge \sim p)$ então q.

Faça a tabela verdade de $((p \vee q) \wedge \sim p) \rightarrow q$

Implicação lógica

Silogismo disjuntivo – Regras:

Modus tollens (Afirmar negando)

$(p \to q) \wedge \sim q \Rightarrow p$

Se é dia então é noite.

Não é noite.

Portanto, é dia.

Se o antecedente (p) implica (é condição necessária para) no consequente (q) e o consequente é negado, então resta o antecedente.

Implicação lógica

Silogismo disjuntivo – Regras:

Modus ponens (Afirmar afirmando)

$(p \rightarrow q) \wedge p \Rightarrow q$

Se é dia então é noite.

Se é dia.

Portanto, é noite.

Se o antecedente (p) implica no consequente (q) e o antecedente existe, então o consequente também existe.

Implicação lógica

Silogismo hipotético

$$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$$

Propriedade de Transitividade da implicação lógica.