Tabelas Verdade

Tabelas verdade podem ser usadas para verificar todos os possíveis valores lógicos de uma fórmula[1] ou de uma expressão booleana.

A construção de tabelas verdade está baseada no princípio da bi-valência. Por esse princípio, sabemos que uma fórmula atômica, representada por uma letra sentencial, é verdadeira ou falsa. Se representarmos falso por 0 e verdadeiro por 1, teremos:

A
0
1

Se uma fórmula é composta por n fórmulas atômicas, teremos 2^n atribuições possíveis.

Assim, a tabela verdade de uma fórmula composta por n fórmulas atômicas terá 2^n linhas.

Com n=2 (A, B), teremos 2^2 = 4 atribuições possíveis

	A	B
1	0	0
2	0	1
3	1	0

[1] Em Lógica, uma fórmula consiste de um conjunto de letras sentenciais – A, B, C etc. – cada uma representando uma proposição, que pode ser verdadeira ou falsa, combinada com conectivos lógicos (aqui estudaremos a negação – o NÂO - a disjunlção – o OU e a conjunção – o E)

4	1	1

Com n=3 (A, B, C), teremos 2^2 = 8 atribuições possíveis

	A	B	C
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

Tabelas Verdade para os Principais Conectivos Lógicos

Negação (~[2] ou ´[3])

A negação de um enunciado A é verdadeira se P for falso e é falsa se P for verdadeiro.

A	A´
0	1
1	0

[2] Representação na Lógica
[3] Representação na Álgebra Booleana

Exemplo:

A = a baleia é um peixe
A é falso e A´ é verdadeiro

A= FHC foi presidente da República Federativa do Brasil
A é verdadeiro e A´ é falso

Disjunção ($\vee$[4] ou +[5]**)**

O conectivo representado acima tem sentido não exclusivo, isto é, uma disjunção é verdadeira se pelo menos um dos disjuntos for verdadeiro.

A disjunção não exclusiva **(**$\vee$**)** só é falsa se os dois disjuntos forem falsos, ao mesmo tempo.

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

[4] Representação na Lógica
[5] Representação na Álgebra Booleana

Exemplo:

A = Lindemberg é professor
B = Lindemberg é funcionário público

A é verdadeiro e B é verdadeiro, logo A + B é verdadeiro

A = Lindemberg é professor
B = Lindemberg nasceu em Ulan Bator, capital da Mongólia

A é verdadeiro e B é falso, logo A + B é verdadeiro.

Obs.: existe um outro conectivo lógico, o “ou exclusivo” ($\oplus$), que estudaremos mais adiante em nosso curso, tal que $A \oplus B$ é falso, se A e B forem, ambos, verdadeiros ou falsos, e verdadeiro, caso contrário (isto é, quando A for verdadeiro e B falso e quando A for verdadeiro e B falso).

Conjunção ($\wedge$[6] ou .[7])

Uma conjunção será verdadeira se ambos os conjuntos forem verdadeiros. Caso contrário, é falsa.

A	B	A . B
0	0	0
0	1	0
1	0	0
1	1	1

[6] Representação na Lógica
[7] Representação na Álgebra Booleana

Exemplo:

A = Belo Horizonte é uma cidade
B = Belo Horizonte é a capital de Minas Gerais

A é verdadeiro e B é verdadeiro, logo P **.** Q é verdadeiro

A = Belo Horizonte é uma cidade
B = Belo Horizonte é a capital do Brasil

A é verdadeiro e B é falso, logo P **.** Q é falso

TABELAS VERDADE PARA FÓRMULAS BEM FORMADAS

Para construir uma tabela-verdade para uma fórmula bem formada wff[8], escrevemos as letras sentenciais à esquerda da tabela e a fórmula à direita da tabela.

Devemos completar com todas as possibilidades de valores verdade para as letras sentenciais.

A seguir, devemos identificar o operador principal[9], pois é ele que determina o valor-verdade para toda a fórmula.

---

[8] Em Lógica, fórmula bem formada (ou wff, proveniente da sigla em inglês para well formed fomula) é uma fórmula que está sintaticamente correta, isto é, que combina conectivos e letras de maneira que faz sentido. As regras para a criação de uma fórmula bem formada – wff fogem ao escopo deste texto.

Por fim, completamos a tabela com os valores-verdade para os operadores, sub-wffs e por fim para a wff (operador principal).

Exemplo:

Tabela verdade para a fórmula A' + B

a) Desenhamos a tabela verdade, com número de linhas igual a 2^n ,onde n é o número de letras sentenciais (fórmulas atômicas) distintas. No exemplo, n = 2 e $2^n = 2^2 = 4$.
b) Preenchemos a coluna da letra sentencial A, fazendo A falso (0) em metade das linhas da tabela verdade, e A verdadeiro (1) na outra metade.
c) Preenchemos a coluna letra sentencial B, fazendo B falso (0) em metade das linhas em que A é falso (0), e B verdadeiro (1) na outra metade. Fazemos, ainda, B falso (0) em metade das linhas em que A é verdadeiro (1), e B verdadeiro (1) na outra metade.
d) preenchemos a coluna da ocorrência de A na fórmula (na wff), repetindo os valores presentes na coluna correspondente a A.

---

[9] Operador principal é aquele que tem como abrangência toda a fórmula. Para identificá-lo, basta construir a fórmula aos poucos, tomando primeiramente os operadores com menor abrangência ou escopo, até chegar à fórmula completa. Deve-se observar a prioridade dos operadores

e) preenchemos a coluna da ocorrência de B na fórmula (na wff) , repetindo os valores presentes na coluna correspondente a B.
f) preenchemos o sinal de negação imediatament à esquerda de A, com o valor verdade correspondente ao conectivo de negação;
g) preenchemos o sinal de disjunção, com o valor verdade correspondente ao conectivo de disjunção, que é o operador principal e, portanto, determina o valor verdade da fórmula; importante observar que está sendo feita a disjunção entre o resultado da negação de A e B.

A	B	A	´	+	B
0	0	0	1	1	0
0	1	0	1	1	1
1	0	1	0	0	0
1	1	1	0	1	1

Observe que essa fórmula às vezes é verdadeira (1^a , 2^a e 4^a linhas) e às vezes é falsa (2^a linha), conforme se pode observar a partir da leitura da coluna do operador principal, que é + . Esse tipo de fórmula é dita **CONTIGENTE**.

Outros exemplos:

A + A´

A	A	+	A	´
0	0	1	0	1
1	1	1	1	0

Essa fórmula é sempre verdadeira (vide coluna do operador principal, +). Esse tipo de fórmula é denominada **TAUTOLOGIA**.

A **.** A’

A	A	.	A	´
0	0	0	0	1
1	1	0	1	0

Essa fórmula é sempre falsa. Vide coluna do operador principal (.). Esse tipo de fórmula é denominada **INCONSISTENTE** ou **CONTRADIÇÃO**.

Exercícios:

Construa as tabelas verdade correspondentes e diga se são tautologias, inconsistentes ou contingentes as seguintes fórmulas:

a) (A + B)’ . (B . A)’
b) (A + (B’ . C)) . (B . (A + C’))
c) ((A . B) + C) + (A’ + (B . C’))