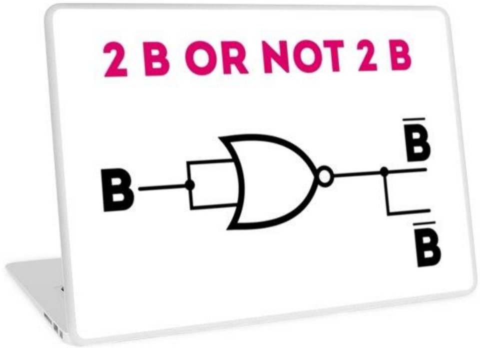
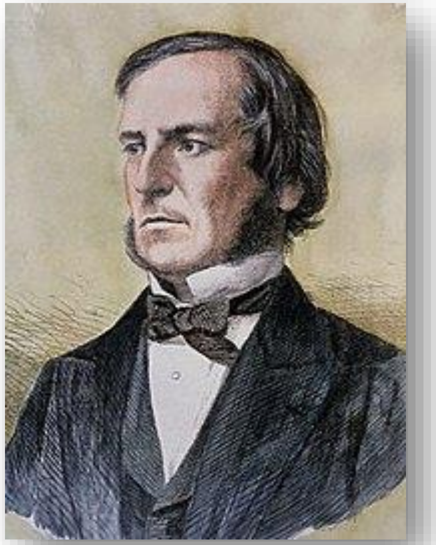
UNIVERSITÀ
DI PARMA

Informatica e Laboratorio di Programmazione

algebra di Boole

Alberto Ferrari





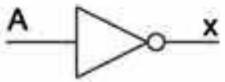
algebra di Boole

- l'algebra di Boole è un ***formalismo*** che opera su variabili (dette variabili booleane)
- le ***variabili booleane*** possono assumere due soli valori: ***vero***, ***falso***
- sulle variabili booleane è possibile definire delle funzioni (dette funzioni booleane)
- le ***funzioni booleane*** possono assumere solo i due valori ***vero, falso***

- una ***tabella di verità*** ***definisce*** una funzione booleana
  - valore risultante per ciascuna ***combinazione*** dei valori in ingresso
- a volte, *specifica incompleta*
  (certe combinazioni di ingressi non possono verificarsi) → non è specificato alcun valore

#	w	x	y	z	f
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

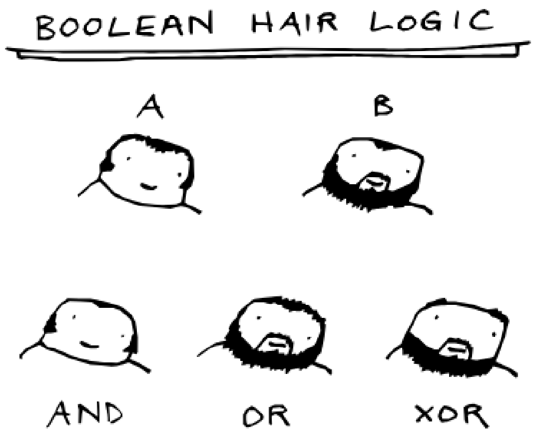
operatori di base

Name	NOT	AND	NAND	OR	NOR	XOR	XNOR
Alg. Expr.	$\overline{A}$	AB	$\overline{AB}$	$A+B$	$\overline{A+B}$	$A\oplus B$	$\overline{A\oplus B}$
Symbol							
Truth Table	A X 0 1 1 0	B A X 0 0 0 0 1 0 1 0 0 1 1 1	B A X 0 0 1 0 1 1 1 0 1 1 1 0	B A X 0 0 0 0 1 1 1 0 1 1 1 1	B A X 0 0 1 0 1 0 1 0 0 1 1 0	B A X 0 0 0 0 1 1 1 0 1 1 1 0	B A X 0 0 1 0 1 0 1 0 0 1 1 1

espressione booleana

- operatori possono essere ***combinati*** in espressioni
  - altra forma di definizione di funzioni booleane
  - es. $F_2(A, B, C) = A \cdot B + C$

Operatore	Simbolo
And	· (∧)
Or	+ (∨)
Not	¬
Xor	⊕
Nand	↑
Nor	↓



proprietà degli operatori

Proprietà	Not
Complemento	¬¬A = A

Proprietà	And	Or
Commutativa	A · B = B · A	A + B = B + A
Associativa	(A·B) · C = A · (B·C)	(A+B) + C = A + (B+C)
Distributiva	A + (B·C) = (A+B) · (A+C)	A · (B+C) = (A·B) + (A·C)
Idempotenza	A · A = A	A + A = A
Identità	A · 1 = A	A + 0 = A
Del limite	A · 0 = 0	A + 1 = 1
Assorbimento	A · (A + B) = A	A + (A · B) = A
Inverso	A · ¬ A = 0	A + ¬A = 1
De Morgan	¬(A·B·C...) = ¬A + ¬B + ¬C...	¬(A+B+C...) = ¬A · ¬B · ¬C...

Attenzione a De Morgan: errore comune!

forme canoniche

- ***somma di prodotti (SP):*** si considerano le righe a *1*
  - $F1(A, B, C) = (\neg A \cdot \neg B \cdot \neg C) + (\neg A \cdot B \cdot C) + (A \cdot \neg B \cdot C) + (A \cdot B \cdot \neg C) + (A \cdot B \cdot C)$
- ***prodotto di somme (PS):*** si considerano le righe a *0*
  - $F1(A, B, C) = (A + B + \neg C) \cdot (A + \neg B + C) \cdot (\neg A + B + C)$

A	B	C	F_1	→ Forma canonica...
0	0	0	1	→ SP
0	0	1	0	→ PS
0	1	0	0	→ PS
0	1	1	1	→ SP
1	0	0	0	→ PS
1	0	1	1	→ SP
1	1	0	1	→ SP
1	1	1	1	→ SP

python – operatori binari

- x << shift # x = x * (2^shift)
- x >> shift # x = x / (2^shift), con segno
- x & y # AND applicato bit a bit
- x | y # OR applicato bit a bit
- x ^ y # XOR bit a bit
- ~x # complemento di ogni bit

Da non confondere con operatori logici (and, or, not)