

Algebra di Boole

Franco Sartore
marzo 2003

Algebra di Boole

- **George Boole**: matematico inglese (1815 - 1864)
- La sua Applicazione alla **Logica** del formalismo dell'**Algebra** è nota come **Algebra di Boole** e si è dimostrata di fondamentale importanza per l'informatica

Algebra di Boole

- Due **costanti**:
 - il valore **VERO** comunemente rappresentato col bit **1**
 - il valore **FALSO** comunemente rappresentato col bit **0**
- Le **variabili** (**variabili logiche**, **variabili booleane**) possono assumere solo i valori **VERO, FALSO** (**1, 0**)

Algebra di Boole a due valori

- Le **operazioni fondamentali** sono
 - Il **Prodotto logico** (operatore **AND**) così chiamato per la sua analogia con il tradizionale prodotto dell'Algebra dei numeri
 - La **Somma logica** (operatore **OR**) così chiamata per la sua (parziale) analogia con la tradizionale somma dell'Algebra dei numeri
 - La **Negazione** (operatore **NOT**)

Algebra di Boole a due valori

- **AND, OR, NOT** sono chiamati **Operatori logici** o **Operatori booleani** e sono definiti assiomaticamente tramite le rispettive **Tavole di Verità**
- Una Tavola di Verità ha tante righe quante sono le combinazioni dei valori che gli **operandi** (valori d'ingresso, **input**) possono assumere

Prodotto logico **AND**

- **AND** è un **operatore binario** (si applica a **due operandi**)
- Dà come risultato **VERO** (**1**) se e solo se gli operandi sono **entrambi VERI**

x	y	x AND y
0	0	0
0	1	0
1	0	0
1	1	1

Somma logica **OR**

- **OR** è un **operatore binario** (si applica a **due operandi**)
- Dà come risultato **VERO (1)** se **almeno uno** degli operandi è **VERO** (quindi anche se **entrambi** gli operandi sono **VERI**)

x	y	x OR y
0	0	0
0	1	1
1	0	1
1	1	1

Negazione **NOT**

- **NOT** è un **operatore unario** (si applica ad **un solo operando**)
- Dà come risultato **VERO (1)** se l'operando è **FALSO (0)** e viceversa

x	NOT x
0	1
1	0

Funzioni logiche

- Tramite gli operatori logici (booleani) si possono costruire **Funzioni logiche (booleane)** di Variabili e costanti logiche, che possono assumere solo i valori **VERO**, **FALSO (1, 0)**
- Le Funzioni logiche sono definite per mezzo delle loro Tavole di Verità

Funzioni logiche

- Tra gli operatori logici è fissata la seguente **gerarchia di esecuzione (precedenze)**:
 - 1 - NOT
 - 2 - AND
 - 3 - OR
- che può essere modificata con l'uso di **parentesi**. Le parentesi possono essere utilizzate anche per aumentare la leggibilità

Funzioni logiche

x	y	z	x OR y	NOT z	u
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	1	0	0
1	0	0	1	1	1
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	0	0

Funzioni logiche

x	y	z	NOT z	y AND NOT z	u
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	1	1	1
1	1	1	0	0	1

Funzioni logiche

- Negli esempi **x, y, z** sono **variabili d'ingresso (input)**, **u** è la **variabile d'uscita (output)**
- Le variabili d'ingresso sono dette **indipendenti**: possono assumere qualunque valore (logico, booleano)
- La variabile d'uscita è detta **dipendente**: il suo valore è determinato dai valori assunti dalle variabili d'ingresso

Reti logiche - Porte logiche

- In un computer l'insieme dei circuiti che elaborano l'informazione può essere visto come una **rete logica**
- Una rete logica è un insieme di dispositivi opportunamente connessi chiamati **porte logiche**
- Le porte logiche sono dispositivi in grado di eseguire operazioni logiche su **input** di tipo **binario**



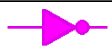
Reti logiche - Porte logiche

- Gli input binari sono di fatto **2 livelli di tensione: alto, basso**
- Si può realizzare con semplici circuiti elettronici una porta logica in corrispondenza di ogni operatore logico (booleano).
- Nella realtà essa avrà come input valori alti o bassi di potenziale che, per comodità rappresenteremo come **1, 0 (VERO, FALSO)**

Reti logiche - Porte logiche





- L'algebra di Boole è lo strumento d'elezione per la rappresentazione e la progettazione delle reti logiche e quindi dei circuiti elettronici che elaborano i dati nei computer

Porte logiche e Operatori booleani Rappresentazione circuitale

X y		u	$x \text{ AND } y = u$
X y		u	$x \text{ OR } y = u$
X		u	$\text{NOT } x = u$

Porte logiche e Operatori booleani Rappresentazione circuitale

- Esistono altri Operatori e Porte logiche che si possono ottenere dalla composizione delle tre fondamentali

NAND		<table border="1"> <thead> <tr><th>A</th><th>B</th><th>$\overline{A \cdot B}$</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	$\overline{A \cdot B}$	0	0	1	0	1	1	1	0	1	1	1	0
A	B	$\overline{A \cdot B}$															
0	0	1															
0	1	1															
1	0	1															
1	1	0															
NOR		<table border="1"> <thead> <tr><th>A</th><th>B</th><th>$\overline{A + B}$</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	$\overline{A + B}$	0	0	1	0	1	0	1	0	0	1	1	0
A	B	$\overline{A + B}$															
0	0	1															
0	1	0															
1	0	0															
1	1	0															
XOR		<table border="1"> <thead> <tr><th>A</th><th>B</th><th>$A \oplus B$</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	$A \oplus B$	0	0	0	0	1	1	1	0	1	1	1	0
A	B	$A \oplus B$															
0	0	0															
0	1	1															
1	0	1															
1	1	0															
XNOR		<table border="1"> <thead> <tr><th>A</th><th>B</th><th>$\overline{A \oplus B}$</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	$\overline{A \oplus B}$	0	0	1	0	1	0	1	0	0	1	1	1
A	B	$\overline{A \oplus B}$															
0	0	1															
0	1	0															
1	0	0															
1	1	1															

Operatori logici e linguaggi di programmazione

- Gli operatori booleani vengono utilizzati anche a livello più alto
- Ogni **linguaggio di programmazione** li mette a disposizione, spesso con simboli più sintetici
- Nel linguaggio **C**:
 - **AND** → `&`
 - **OR** → `|`
 - **NOT** → `!`