

PROGRAMMAZIONE INFORMATICA

3. ALGEBRA BOOLEANA

RICCARDO ZAMOLO
rzamolo@units.it

UNIVERSITÀ DEGLI STUDI TRIESTE
INGEGNERIA CIVILE E AMBIENTALE



A.A. 2023-24

- Utilizzata per trattare i segnali binari ($\mathcal{B} = \{0,1\}$), cioè quelli che vengono utilizzati materialmente nel calcolatore.
- Princìpi generali pubblicati nel 1854 da George Boole (1815-1864).
- Postulati di Huntington (1874-1952): dato un insieme S ed una relazione di uguaglianza che gode delle proprietà riflessiva ($x = x$), simmetrica ($x = y \Leftrightarrow y = x$) e transitiva ($x = y, y = z \Rightarrow x = z$), si suppone S sia dotato delle seguenti proprietà:

A_1, A_2 . Sono definite due leggi di composizione:

- somma logica “ $+$ ” $+: S \times S \rightarrow S, (x, y) \mapsto x + y$

- prodotto logico “ $\cdot$ ” $\cdot: S \times S \rightarrow S, (x, y) \mapsto x \cdot y$

B_1, B_2 . Esistono due elementi di S , denominati *zero* (0) e *unità* (1), tali che:

$$x + 0 = x \quad \text{e} \quad x \cdot 1 = x$$

C_1, C_2 . Proprietà commutativa di “ $+$ ” e “ $\cdot$ ”:

$$x + y = y + x \quad \text{e} \quad x \cdot y = y \cdot x$$

D_1, D_2 . Proprietà distributiva di “ $+$ ” rispetto a “ $\cdot$ ” e viceversa:

$$x + (y \cdot z) = (x + y) \cdot (x + z)$$

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$$

E . Per ogni $x \in S$ esiste un $\bar{x} \in S$, denominato *complemento* di x , tale che:

$$x + \bar{x} = 1 \quad \text{e} \quad x \cdot \bar{x} = 0$$

F . Esiste almeno una coppia di elementi x e y di S tali che $x \neq y$.

- I postulati di Huntington definiscono sull'insieme S una struttura algebrica denominata *Algebra Booleana*.
- La compatibilità dei postulati di Huntington può essere verificata con l'esistenza di modelli per i quali i postulati sono validi e non si contraddicono.
- Modello binario (*algebra binaria*) dove $S = \mathcal{B} = \{0, 1\}$ e dove le due operazioni logiche fondamentali “+” e “ $\cdot$ ” sono definite da:

OR			AND			$\Rightarrow$	NOT		
+	0	1	$\cdot$	0	1		x	0	1
0	0	1	0	0	0		$\bar{x}$	1	0
1	1	1	1	0	1				

- il complemento (NOT) può essere costruito dalla definizione delle due operazioni logiche “+” e “ $\cdot$ ” soddisfacendo il postulato E;
- il postulato F vale per la definizione di $S = \mathcal{B} = \{0, 1\}$;
- le operazioni logiche “+” e “ $\cdot$ ” così definite rispettano i postulati A_1 , A_2 , B_1 , B_2 , C_1 , C_2 ;
- la proprietà distributiva (postulati D_1 e D_2) va verificata.

- Verifica della proprietà distributiva D_1 e D_2 . Va effettuata per tutte le $2^n = 8$ combinazioni dei possibili valori delle $n = 3$ variabili x, y e z :

$$u = x + (y \cdot z)$$

$$v = (x + y) \cdot (x + z)$$

x	y	z	u	v
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

$$u = x \cdot (y + z)$$

$$v = (x \cdot y) + (x \cdot z)$$

x	y	z	u	v
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

- I postulati di Huntington applicati al modello binario costituiscono un sistema compatibile.
- Dai postulati è possibile dedurre infiniti *teoremi*, cioè delle identità tra due termini Booleani basati sugli operatori somma, prodotto e complemento logici applicati alle variabili ed alle costanti fondamentali (zero e unità).

- Teoremi fondamentali dell'Algebra Booleana:

Teorema	Base (1)	Duale (2)
T1	$x + x = x$	$x \cdot x = x$
T2	$x + 1 = 1$	$x \cdot 0 = 0$
T3	$x + (x \cdot y) = x$	$x \cdot (x + y) = x$
T4	$x + (\bar{x} \cdot y) = x + y$	$x \cdot (\bar{x} + y) = x \cdot y$
T5	$\overline{(\bar{x})} = x$	
T6	$\overline{(x + y)} = \bar{x} \cdot \bar{y}$	$\overline{(x \cdot y)} = \bar{x} + \bar{y}$

T1, T2 e T5 sono verificabili in maniera diretta;

$$\text{T3}_1: x + (x \cdot y) \stackrel{\text{B}_2}{=} (x \cdot 1) + (x \cdot y) \stackrel{\text{D}_2}{=} x \cdot (1 + y) \stackrel{\text{T}_{21}}{=} x$$

$$\text{T3}_2: x \cdot (x + y) \stackrel{\text{B}_1}{=} (x + 0) \cdot (x + y) \stackrel{\text{D}_1}{=} x + (0 \cdot y) \stackrel{\text{T}_{22}}{=} x$$

$$\text{T4}_1: x + (\bar{x} \cdot y) \stackrel{\text{D}_1}{=} (x + \bar{x}) \cdot (x + y) \stackrel{\text{E}}{=} x + y$$

$$\text{T4}_2: x \cdot (\bar{x} + y) \stackrel{\text{D}_2}{=} (x \cdot \bar{x}) + (x \cdot y) \stackrel{\text{E}}{=} x \cdot y$$

T6 (De Morgan) con verifica diretta:

x	y	$\overline{x+y}$	$\bar{x} \cdot \bar{y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

x	y	$\overline{x \cdot y}$	$\bar{x} + \bar{y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

- *Variabile Booleana*: $x \in \mathcal{B} = \{0, 1\}$;
- *Funzione Booleana* f di n variabili Booleane $x_1, \dots, x_n$:

$$f : \mathcal{B}^n \rightarrow \mathcal{B}$$

$$(x_1, \dots, x_n) \mapsto f(x_1, \dots, x_n)$$

- Le funzioni Booleane si possono esprimere algebricamente tramite operatori logici (somma, prodotto, complemento) applicati alle variabili Booleane ed alle costanti 0 e 1. Ad esempio $f(x, y, z) = \bar{x} \cdot y + z$.
- Essendo $\mathcal{B}$ un insieme finito, è possibile definire una funzione booleana anche per mezzo di una *tabella di verità* che riporta il valore di f per ogni combinazione dei valori binari delle variabili $x_1, \dots, x_n$:

x_1	$\dots$	x_n	$f(x_1, \dots, x_n)$
0	0	0	$f(0, \dots, 0)$
0	0	1	$f(0, \dots, 1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	1	1	$f(1, \dots, 1)$

- Le righe della precedente tabella sono $m = 2^n$, ossia il numero delle n -uple Booleane in $\mathcal{B}^n$.
- Il numero delle possibili funzioni Booleane di n variabili Booleane è quindi $2^m = 2^{2^n}$.

- Funzioni di una variabile ($n = 1$, $2^{2^1} = 4$ possibili funzioni):

	f_0	f_1	f_2	f_3
x	0	x	$\bar{x}$	1
0	0	0	1	1
1	0	1	0	1

- Funzioni di due variabili ($n = 2$, $2^{2^2} = 16$ possibili funzioni):

	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
	AND				XOR				OR	NOR	XNOR	NAND				
x	y	0	$\cdot$	x	y	$\oplus$	$+$	$\downarrow$	$\odot$	$\bar{y}$	$\bar{x}$					1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	1	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1
1	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	1
1	1	0	1	0	1	0	0	1	1	0	0	1	0	1	0	1

- Oltre alle funzioni OR e AND già viste, assumono particolare interesse pratico le seguenti funzioni, simmetriche rispetto ai due ingressi x e y :
 - XOR (OR esclusivo): $x \oplus y$
 - NOR (OR negato): $x \downarrow y = \overline{x + y} = \bar{x} \cdot \bar{y}$
 - XNOR (XOR negato): $x \odot y = \overline{x \oplus y}$
 - NAND (AND negato): $x | y = \overline{x \cdot y} = \bar{x} + \bar{y}$

- Addizione di due cifre binarie x e y con riporto, tabella di verità:

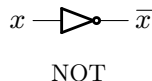
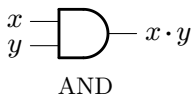
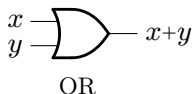
		Somma	Riporto
x	y	s	r
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- s ed r : funzioni Booleane da determinare algebricamente.
- La somma s è equivalente alla funzione XOR:

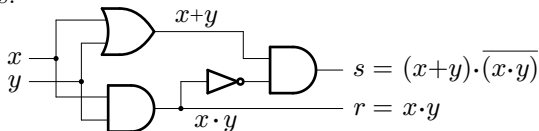
$$\begin{aligned}
 s = x \oplus y &= \overbrace{(\bar{x} \cdot y) + (x \cdot \bar{y})}^{s_1} = \\
 &= \underbrace{(x + y)}_{s_2} \cdot \underbrace{(\overline{x \cdot y})}_{s_3} = \underbrace{(x + y)}_{s_2} \cdot \underbrace{(\bar{x} + \bar{y})}_{s_3}
 \end{aligned}$$

- Il riporto r è equivalente alla funzione AND: $r = x \cdot y$.
- Numero di blocchi logici necessari per l'addizione:
 - s_1 : 6 blocchi (2 AND, 1 OR e 2 complementi per s_1 ; 1 AND per r);
 - s_2 : 4 blocchi (2 AND [$x \cdot y$ contato una volta sola per s_2 e r], 1 OR e 1 complemento per s_2);
 - s_3 : 6 blocchi (1 AND, 2 OR e 2 complementi per s_3 ; 1 AND per r);
- La forma s_2 per la funzione Booleana somma consente l'uso del minor numero di blocchi logici per la realizzazione dell'addizione di due cifre binarie con riporto.

- Alcuni simboli circuitali (*porte logiche*):



- Le porte logiche vengono realizzate materialmente impiegando dispositivi elettronici come diodi, transistor, resistenze e C-MOS.
- Funzioni Booleane più articolate si ottengono combinando gli ingressi e le uscite delle porte logiche fondamentali (OR, AND, NOT, NAND, NOR), dando origine alle *reti logiche*.
- Esempio di rete logica che realizza l'addizione di due cifre binarie x e y con riporto (*Half-adder*), utilizzando la forma $s_2 = (x + y) \cdot \overline{(x \cdot y)}$ per la somma s :



- Tali reti logiche vengono impiegate nei calcolatori elettronici per realizzare determinate funzioni Booleane che permettono di operare sui segnali binari.