Algebra di Boole

L'algebra dei calcolatori

L'algebra booleana è un particolare tipo di algebra in cui le variabili e le funzioni possono solo avere valori 0 e 1. Deriva il suo nome dal matematico inglese George Boole che la ideò.

George Boole

Si studia l'algebra booleana poiché le funzioni dell'algebra booleana sono isomorfe ai circuiti digitali. In altre parole, un circuito digitale può essere espresso tramite un'espressione booleana e viceversa.

Una funzione booleana ha una o più variabili in input e fornisce risultati che dipendono solo da queste variabili.

Poiché le variabili possono assumere solo i valori 0 o 1 una funzione booleana con n variabili di input ha solo 2^n combinazioni possibili e può essere descritta dando una tabella, detta **tabella di verità**, con 2^n righe.

$V = f(X, Y, Z)$

Che funzione codifica f?

Input

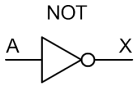
X	Y	Z	V
0	0	0	0
0	0	1	0
0	1	0	0
1	0	0	0
0	1	1	1
1	0	1	1
1	1	0	1
1	1	1	1

Output

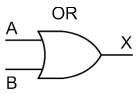
Ogni riga mostra il valore restituito a partire da una particolare configurazione dell'input.

L'algebra di Boole (1)

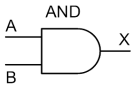
L'algebra di Boole si basa su tre operatori di base: **AND**,**OR**,**NOT**



A	X
0	1
1	0



A	B	X
0	0	0
0	1	1
1	0	1
1	1	1



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Tutte le funzioni booleane possono essere espresse come combinazione di questi tre operatori.

Ad ogni funzione di base corrisponde una porta logica e quindi ogni espressione booleana può essere tradotta in un circuito.

Tramite le proprietà dell'algebra booleana è possibile semplificare espressione booleane complesse

Anche i circuiti corrispondenti saranno più semplici e richiederanno un minor numero di porte logiche

Minori costi di realizzazione dei circuiti
&
Minore occupazione di spazio

Le espressioni booleane vengono utilizzate nei linguaggi di programmazione per la definizione dei criteri decisionali

L'algebra di Boole (2)

Le dimensioni delle tabelle di verità crescono al crescere delle variabili in input, sono quindi necessarie delle rappresentazioni alternative:

- Ordinando le righe dell'input come numeri binari è possibile codificare le tabelle di verità memorizzando solo la colonna dell'output.

X	Y	Z	V
0	1	0	0
1	0	1	1
0	0	0	0
1	0	0	0
0	1	1	1
0	0	1	0
1	1	0	1
1	1	1	1

X	Y	Z	V
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

V
0
0
0
1
0
1
1
1

- Si può specificare l'output di ogni funzione booleana esprimendo, tramite una espressione booleana, quali combinazioni delle variabili di input determinano l'output 1. Per l'esempio precedente: (011-101-110-111). Questa rappresentazione è importante perché può essere direttamente tradotta in un circuito di porte digitali.

Convenzioni notazionali

- Una variabile con valore 1 è indicata dal suo nome
- Una variabile con valore 0 è indicata dal suo nome con sopra una barra
- L'operazione di AND booleano è indicato da un $\cdot$ moltiplicativo oppure viene considerato implicitamente presente
- L'operazione di OR booleano è indicato da un +

$$(011\text{-}101\text{-}110\text{-}111) \Rightarrow \overline{X}YZ + X\overline{Y}Z + XY\overline{Z} + XYZ$$

L'algebra di Boole (3)

Per passare dalla rappresentazione mediante tabella di verità alla notazione tramite espressione booleana è necessario:

1. Identificare tutte le righe della tabella di verità che danno 1 in output;
2. Per ogni riga con un 1 in output scrivere la configurazione delle variabili che la definiscono (tutte le variabili della configurazione saranno in AND tra loro)
3. Collegare tramite OR tutte le configurazioni ottenute.

La rappresentazione così ottenuta è detta in **prima forma canonica**.

Esempio: *Traduzione della funzione rappresentata dalla tabella di verità:*

X	Y	Z	V	
0	0	0	0	
0	0	1	1	$\Leftarrow \overline{X}\overline{Y}Z$
0	1	0	1	$\Leftarrow \overline{X}Y\overline{Z}$
0	1	1	0	
1	0	0	1	$\Leftarrow X\overline{Y}\overline{Z}$
1	0	1	0	
1	1	0	0	
1	1	1	1	$\Leftarrow XYZ$

$$\overline{X}\overline{Y}Z + \overline{X}Y\overline{Z} + X\overline{Y}\overline{Z} + XYZ$$

ATTENZIONE: all'uso della negazione !

$$\overline{AB} \neq \overline{A}\ \overline{B}$$

$\overline{A}\ \overline{B}$	$\overline{B}$	$\overline{A}$	A	B	AB	$\overline{AB}$
1	1	1	0	0	0	1
0	0	1	0	1	0	1
0	1	0	1	0	0	1
0	0	0	1	1	1	0

Semplificazione di funzioni booleane (1)

Le funzioni booleane possono essere descritte da più espressioni equivalenti

$$X\,Y\,\overline{Z} + X\,\overline{Y}\,Z + X\,Y\,Z \equiv X(Y+Z)$$

X	Y	Z	V	Y+Z	X(Y+Z)
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	0	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

Def. Due funzioni booleane si dicono ***equivalenti*** se presentano lo stesso output per qualsiasi configuarazione del'input.

Determinare **la più semplice** funzione booleana equivalente alla funzione data facilita l'interpretazione della funzione stessa e permette di semplificare anche i circuiti logici corrispondenti.

"Cosa si intende per più semplice?"

Una funzione f' è più semplice di una funzione f ($f \equiv f'$) secondo il ***criterio del minor numero di operatori*** se il suo calcolo richiede meno operazioni di AND e OR rispetto al calcolo di f.

N.B. Sebbene questo sia il criterio più utilizzato ne esistono altri

Semplificazione di funzioni booleane (2)

Le funzioni booleane possono essere semplificate applicando ripetutamente le seguenti proprietà:

Name	AND form	OR form
Identity law	1A = A	0 + A = A
Null law	0A = 0	1 + A = 1
Idempotent law	AA = A	A + A = A
Inverse law	$A\bar{A} = 0$	$A + \bar{A} = 1$
Commutative law	AB = BA	A + B = B + A
Associative law	(AB)C = A(BC)	(A + B) + C = A + (B + C)
Distributive law	A + BC = (A + B)(A + C)	A(B + C) = AB + AC
Absorption law	A(A + B) = A	A + AB = A
De Morgan's law	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$

Proprietà di identità:

1	**0**	**A**	**1A**	**0+A**
1	0	0	0	0
1	0	1	1	1

Gli elementi 0 e 1 sono detti **elementi forzanti** rispettivamente per le funzioni OR e AND

Elemento nullo:

1	**0**	**A**	**0A**	**1+A**
1	0	0	0	1
1	0	1	0	1

Proprietà di inverso:

A	$\bar{A}$	$A\bar{A}$	$A+\bar{A}$
0	1	0	1
1	0	0	1

Semplificazione di funzioni booleane (3)

Proprietà associativa:

A	B	C	AB	BC	A(BC)	(AB)C	A+B	B+C	A+(B+C)	(A+B)+C
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	1	1	1	1
0	1	1	0	1	0	0	1	1	1	1
1	0	0	0	0	0	0	1	0	1	1
1	0	1	0	0	0	0	1	1	1	1
1	1	0	1	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

Proprietà distributiva:

A	B	C	BC	A+BC	A+B	A+C	(A+B)(A+C)	B+C	A(B+C)	AB	AC	AB+AC
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	1	0	0	0	0
0	1	0	0	0	1	0	0	1	0	0	0	0
0	1	1	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	0	0	0	0	0
1	0	1	0	1	1	1	1	1	1	0	1	1
1	1	0	0	1	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1

ATTENZIONE: la proprietà distributiva mette in luce una netta differenza rispetto all'aritmetica convenzionale in cui la somma non è distributiva rispetto al prodotto .

Proprietà di assorbimento:

$$A+AB = A\cdot 1+AB = A(1+B) = A\cdot 1 = A$$

Distributiva — **Identità** — **Identità** — **Elemento nullo**

Semplificazione di funzioni booleane (4)

Teorema di DeMorgan:

A	B	$\overline{A}$	$\overline{B}$	A+B	$\overline{A}+\overline{B}$	AB	$\overline{AB}$	$\overline{A}\,\overline{B}$	$\overline{A+B}$
0	0	1	1	0	1	0	1	1	1
0	1	1	0	1	1	0	1	0	0
1	0	0	1	1	1	0	1	0	0
1	1	0	0	1	0	1	0	0	0

Il teorema di DeMorgan si può estendere a più variabili: $\overline{ABC} = \overline{A} + \overline{B} + \overline{C}$

Il teorema di DeMorgan permette di esprimere l'operatore AND/OR in funzione dell'operatore OR/AND.

ATTENZIONE: si ricorda che $\overline{\overline{A}} = A$

La semplificazione delle funzioni booleane può essere ottenuta anche mediante le mappe di Karnaugh o l'algoritmo di Quine/McClusky.

ESERCIZI (1)

1. Si scriva, utilizzando gli operatori booleani AND, OR, NOT, la funzione booleana che ritorna in uscita il valore 1 se sono veri un numero dispari dei tre input

2. Si scriva, utilizzando gli operatori booleani AND, OR, NOT, la funzione booleana che ritorna in uscita il valore 1 se sono veri due dei quattro input

3. Si scriva, utilizzando gli operatori booleani AND, OR, NOT, la funzione booleana che riceve in ingresso un numero binario su 3 bit e ritorna in uscita il valore 1 se e solo se il numero che c'è in ingresso è maggiore o uguale a quattro.

4. Dati gli operatori booleani AND, OR, NOT, scrivere l'espressione di una funzione booleana F avente come ingressi due numeri binari X e Y su 2 bit, che ritorni il valore 1 se X > Y.

ESERCIZI (2)

5. Applicando i teoremi dell'algebra di Boole, verificare la seguente equivalenza tra espressioni

$$\overline{A}\,\overline{B}\,\overline{C} + B\,\overline{C} + A\left(B + \overline{BC}\right) \equiv A + \overline{C}$$

6. Applicando i teoremi dell'algebra di Boole, semplificare le seguenti espressioni e disegnarne la tavola di verità

$$A\,B\,\overline{C} + A\,B + A\,C + C$$

$$\overline{A}\,\overline{B}\,C + A\,\overline{B} + \overline{A}\,\overline{B} + A\,B$$

$$A + A\,B + B + B\,C$$

7. Si valutino le espressioni booleani precedenti assumendo A=1

8. Una cassaforte ha quattro lucchetti, x, y, v, w, che devono essere tutti aperti affinché la cassaforte possa essere aperta. Le chiavi sono distribuite tra 3 persone, A, B, C, come segue: A possiede le chiavi v e y ; B possiede le chiavi v e x ; C possiede le chiavi w e y. Siano le variabili A, B e C uguali a 1 se la persona corrispondente è presente, altrimenti uguali a 0. Costruire la tavola della verità della funzione f(A,B,C) che è uguale ad 1 se e solo se la cassaforte può essere aperta, ed esprimere f in forma algebrica.