

Algebra booleana

Antonella Santone

Anno Accademico 2008/2009

1

Boolean

Il nome *boolean* viene da George Boole, un matematico che inventò delle strutture matematiche, dette appunto algebre di Boole

valori

ha solo due valori: vero (1) e falso (0)

operazioni

operatori logici

not, \neg , $\bar{\quad}$ (unario)
and, \wedge , \bullet (binario)
or, \vee , $+$ (binario)

2

Tavole di verità: not

x	not x
0	1
1	0

3

Esempi di not

x: "piove" **VERO**

not x: "non piove" **FALSO**

Posso scrivere:

not x oppure
 $\neg x$ oppure
 \bar{x}

4

Tavole di verità: and

x	y	x and y
0	0	0
0	1	0
1	0	0
1	1	1

Posso scrivere:

x and y oppure
 $x \wedge y$ oppure
 $x \bullet y$

5

Esempi di and

x: "Benevento è capoluogo di provincia" **VERO**
y: "il Molise è una regione del Nord d'Italia" **FALSO**

x and y:

"Benevento è capoluogo di provincia e il Molise è una regione del Nord d'Italia" **FALSO**

x: "la terra è un pianeta" **VERO**
y: "la luna è un satellite della terra" **VERO**

x and y:

"la terra è un pianeta e la luna è un satellite della terra" **VERO**

6

Tavole di verità: or

x	y	x or y
0	0	0
0	1	1
1	0	1
1	1	1

Posso scrivere:

$x \text{ or } y$ oppure
 $x \vee y$ oppure
 $x + y$

7

Esempi di or

x: "Benevento è capoluogo di provincia" **VERO**
 y: " il Molise è una regione del Nord d'Italia " **FALSO**
 z: "Benevento è capitale d'Italia" **FALSO**

x or y:

"Benevento è capoluogo di provincia oppure il Molise è una regione del Nord d'Italia "

VERO

y or z:

" il Molise è una regione del Nord d'Italia oppure Benevento è capitale d'Italia"

FALSO

8

Precedenza degli operatori logici

Operatori	Associatività
Operatore unario: not	da destra a sinistra
Operatore binario: and	da sinistra a destra
Operatore binario: or	da sinistra a destra

+ alta

+ bassa

9

Esempio

Se a è vero
 b è falso
 c è falso

a or b and c

?

a or (b and c) **vero**

10

Tabella di verità di: a or (b and c)

a	b	c	b and c	a or (b and c)
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

11

Teoremi fondamentali

\bullet = and
 $+$ = or
 $\bar{}$ = not

Identità: $1 \bullet x = x$ $0 + x = x$

Nulla: $0 \bullet x = 0$ $1 + x = 1$

Idempotenza: $x \bullet x = x$ $x + x = x$

Inverso: $\bar{x} \bullet x = 0$ $x + \bar{x} = 1$

Commutativa: $x \bullet y = y \bullet x$ $x + y = y + x$

Associativa: $(x \bullet y) \bullet z = x \bullet (y \bullet z)$ $(x + y) + z = x + (y + z)$

Distributiva: $x \bullet (y + z) = x \bullet y + x \bullet z$ $x + (y \bullet z) = (x + y) \bullet (x + z)$

Principio di dualità:
 derivabili l'una
 dall'altra scambiando:
 $+$ con \bullet
 0 con 1



12

Valgono le proprietà distributive

$$x \text{ and } (y \text{ or } z) = (x \text{ and } y) \text{ or } (x \text{ and } z)$$

"mi piace" e ("ho i soldi" oppure "ho la carta di credito")

=

("mi piace" e "ho i soldi") oppure ("mi piace" e "ho la carta di credito")

13

Regole di De Morgan

$$\overline{(x \bullet y)} = \bar{x} + \bar{y}$$

Esempio:

non (ho i soldi o ho la carta di credito)

=

non ho i soldi e non ho la carta di credito

x	y	$\overline{(x+y)}$	$\bar{x} \bullet \bar{y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

14

Approfondimento

Rappresentazione circuitale delle operazioni logiche

15

Porte logiche

and



or



not

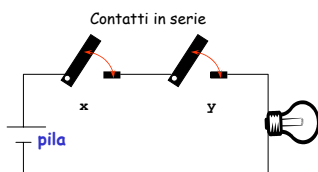


16

and realizzato con interruttori

x	y	x and y
0	0	0
1	0	0
0	1	0
1	1	1

{aperto = 0, chiuso = 1}



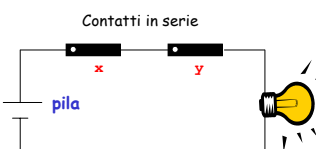
x	y	x and y
aperto	aperto	aperto
chiuso	aperto	aperto
aperto	chiuso	aperto
chiuso	chiuso	chiuso

17

and realizzato con interruttori

x	y	x and y
0	0	0
1	0	0
0	1	0
1	1	1

{aperto = 0, chiuso = 1}



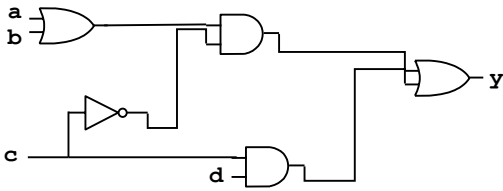
x	y	x and y
aperto	aperto	aperto
chiuso	aperto	aperto
aperto	chiuso	aperto
chiuso	chiuso	chiuso

18

Esempio

$$y = ((a \text{ or } b) \text{ and } (\text{not } c)) \text{ or } (c \text{ and } d)$$

circuito



19

$$y = ((a \text{ or } b) \text{ and } (\text{not } c)) \text{ or } (c \text{ and } d)$$

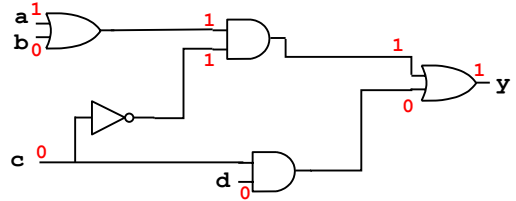
a	b	c	d	a or b	not c	c and d	(a or b) and (not c)	y
0	0	0	0	0	1	0	0	0
0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0
0	0	1	1	0	0	1	0	1
0	1	0	0	1	1	0	1	1
0	1	0	1	1	1	0	1	1
0	1	1	0	1	0	0	0	0
0	1	1	1	1	0	1	0	1

20

$$y = ((a \text{ or } b) \text{ and } (\text{not } c)) \text{ or } (c \text{ and } d)$$

a	b	c	d	a or b	not c	c and d	(a or b) and (not c)	y
1	0	0	0	1	1	0	1	1
1	0	0	1	1	1	0	1	1
1	0	1	0	1	0	0	0	0
1	0	1	1	1	0	1	0	1
1	1	0	0	1	1	0	1	1
1	1	0	1	1	1	0	1	1
1	1	1	0	1	0	0	0	0
1	1	1	1	1	0	1	0	1

21



22

Semplificazione

Il vantaggio dell'algebra di Boole sta nel fatto di permettere la semplificazione dei circuiti

Esempio

$$F = \bar{x} \cdot y \cdot z + \bar{x} \cdot y \cdot \bar{z} + x \cdot z \quad (\text{distributiva})$$

$$= \bar{x} \cdot y \cdot (z + \bar{z}) + x \cdot z \quad (\text{inverso})$$

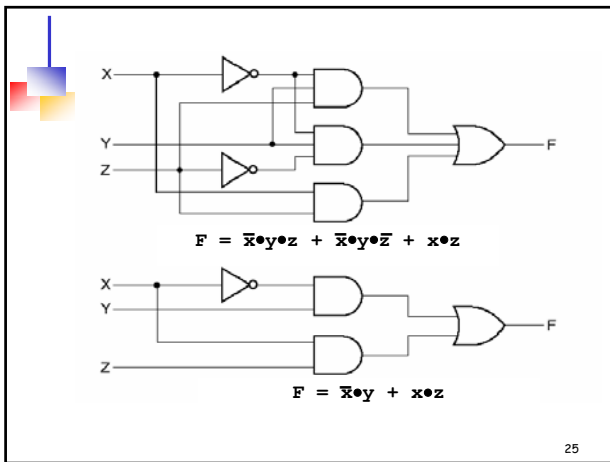
$$= \bar{x} \cdot y \cdot 1 + x \cdot z \quad (\text{identità})$$

$$= \bar{x} \cdot y + x \cdot z$$

23

Le due funzioni sono equivalenti: hanno la stessa tabella di verità ma la seconda funzione è realizzabile con un circuito più semplice

24



Booleani in C?

In C non è presente il tipo booleano; esso viene simulato utilizzando gli interi con la convenzione che:

0 qualunque numero diverso da 0 rappresenta **falso**
 rappresenta **vero**

Le operazioni logiche sono:

not	→	!
and	→	&&
or	→	

Restituiscono 0 quando il risultato è falso,
 1 quando il risultato è vero

Esempi

```
#include<stdio.h>
main()
{
  int i=3, j=2, k=0, ris;
  ris = i && j;           // ris vale 1
  ris = i && k;           // ris vale 0
  ris = i || k;          // ris vale 1
  ris = !(i && k);       // ris vale 1
  ris = !i && k;         // ris vale 0
}
```

Esempi sbagliati

```
a &&           // omissso un operando
a & & b       // spazio non permesso
```

Tipi ordinati

Sono ordinati: interi, reali, caratteri

Su un tipo ordinato sono definiti gli operatori di relazione

Operatori relazionali

== uguale
 != diverso
 < minore
 <= minore o uguale
 > maggiore
 >= maggiore o uguale

Restituiscono 0 quando il risultato è falso,
 1 quando il risultato è vero

Attenzione



31

Esercizi

... sui booleani

32

Esercizio

Scrivere le espressioni booleane che determinano se:

1. il valore di una variabile i è nell'intervallo da 1 a 100, estremi inclusi
2. il valore di una delle due variabili intere j e k è multiplo dell'altro

33

Soluzione

1. $(i \geq 1) \ \&\& \ (i \leq 100)$

con $||$ è sbagliato

34

Soluzione

2. $(j \% k == 0) \ || \ (k \% j == 0)$

Corretto??

Devi imporre con j e k diversi da 0

35

Se scrivo

```
#include<stdio.h>
main()
{int j=9;
  int k=0;
  int ris;
  ris =(j % k == 0) || (k % j == 0);
  printf("%d\n", ris);
}
```

36

succede ...

```
Santone@PC ~  
$ ./a.exe  
Arithmetic exception (core dumped)  
Santone@PC ~  
$ =
```

37

... una soluzione

```
#include<stdio.h>  
main()  
{int j=9;  
int k=0;  
int ris;  
ris = (k!=0) && (j!=0) && ((j % k == 0) || (k % j == 0));  
printf("%d\n", ris);  
}
```

38

Perché funziona?

Il compilatore *C* è sufficientemente furbo da interrompere la valutazione di un'espressione non appena determina la verità o la falsità dell'intera espressione

E1 && E2

Viene valutata **E1**.

Nulla: **0*x=0**

Se **E1** è falsa non viene valutata **E2** perché in ogni caso l'intera espressione sarà falsa.

Se **E1** è vera allora viene valutata **E2**.

39

... lo stesso per ||

E1 || E2

Nulla: **1+x=1**

Viene valutata **E1**.

Se **E1** è vera non viene valutata **E2** perché in ogni caso l'intera espressione sarà vera.

Se **E1** è falsa allora viene valutata **E2**.

Questo modo di valutare le espressioni, velocizza le operazioni, perché non esegue operazioni inutili

40

Esercizio

Se le variabili intere **a**, **b** e **c** hanno rispettivamente valore **3**, **12** e **4**, quale valore viene assegnato alla variabile intera **ris** dalle seguenti espressioni?

```
ris = a+b*c; // 51  
ris = (a>b); // 0  
ris = (a+b) * (a<b); // 15  
ris = (a+b) && (a<b); // 1  
ris = (a+b) || (a>b); // 1  
ris = (a*c-b) || (a>b); // 0  
ris = ((a*c) != b) || (a>b); // 0  
ris = (a>b) || (a<c) || (c==b); // 1
```

41

Esercizio

Supponendo che le variabili intere **x**, **y** abbiano valori **12** e **45**, rispettivamente e che le variabili carattere **a** e **b** abbiano rispettivamente valori **'t'** e **'T'**, cosa stampano le seguenti istruzioni?

```
printf("%d\n", (x>y) || (a!=b)); // 1  
printf("%d\n", (x || a)); // 1  
printf("%d\n", x + (a==b)); // 12  
printf("%d\n", !(x || (x<y)) == (!x && x>=y)); // 1  
printf("%c\n", a*(x||y)); // t
```

42

Esercizio

Si considerino le seguenti 4 variabili:
luiamalei leiamalui luisoldi leisoldi

Scrivere l'istruzione di assegnamento
`sisposano = ...`

affinché due persone si sposino è necessario che entrambe si amino e che almeno una dei 2 abbia i soldi (per il matrimonio)

43

Soluzione

```
sisposano = luiamalei && leiamalui &&  
(luisoldi || leisoldi);
```

Si può verificare facilmente che la variabile SISPOSANO assumerà valore 1 solo nel caso in cui:

LUIAMALEI vale 1
LEIAMALUI vale 1
LUISOLDI vale 1 oppure LEISOLDI vale 1

44

Esercizio

Usando i teoremi fondamentali dell'algebra booleana, si semplifichi la seguente funzione in termini di operatori **and**, **or** e **not**

$$F = ((x+y) \cdot (x+z)) + (x+y) \cdot \bar{z}$$

Scrivere la tavola di verità della funzione ridotta e disegnare il circuito corrispondente

45

Soluzione

Semplificazione:

$$F = ((x+y) \cdot (x+z)) + (x+y) \cdot \bar{z} \quad \textit{distributiva}$$

$$F = (x+y) \cdot (x+z+\bar{z}) \quad \textit{inverso}$$

$$F = (x+y) \cdot (x+1) \quad \textit{nullo}$$

$$F = (x+y) \cdot 1 \quad \textit{identità}$$

$$F = x+y$$

46

Soluzione (cont.)

Circuito: 

Tavola di verità:

x	y	F
0	0	0
0	1	1
1	0	1
1	1	1

47

Esercizio

Valutare e disegnare il circuito

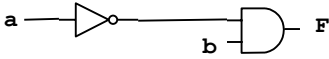
$$F = (\text{not } a) \text{ and } b$$

$$G = (\text{not } a) \text{ or } b$$

48

Soluzione: F

a	b	not a	F = (not a) and b
0	0	1	0
0	1	1	1
1	0	0	0
1	1	0	0



49

Esercizio

Stampare le tabelle booleane relative agli operatori and, or e not

50

Tabella dell'and

```
int x,y;
printf("TABELLA BOOLEANA DELL'AND\n\n");
printf("x \t y \t x and y\n\n");
x = 0;
y = 0;
printf("%d \t %d \t %d\n",x, y, x && y);
x = 0;
y = 1;
printf("%d \t %d \t %d\n",x, y, x && y);
x = 1;
y = 0;
printf("%d \t %d \t %d\n",x, y, x && y);
x = 1;
y = 1;
printf("%d \t %d \t %d\n",x, y, x && y);
```

51

Cygwin

```
Santone@PC ~
$ gcc taband.c
Santone@PC ~
$ ./a.exe
TABELLA BOOLEANA DELL'AND
x      y      x and y
0      0      0
0      1      0
1      0      0
1      1      1
Santone@PC ~
$ =
```

52