



Insegnamento di Elementi di informatica (6 CFU)

Algebra Booleana

ing. Nadia Ranaldo
RCOST - Dipartimento di Ingegneria
Università degli Studi del Sannio

Algebra di Boole



• Il nome *booleano* (o *boolean* in inglese) viene da George Boole, un matematico inglese del XIX secolo che inventò una aritmetica particolare, detta appunto **Algebra di Boole**

valori

Una **variabile booleana** ha solo due valori: **vero** (1) e **falso** (0)

operazioni

Gli operatori che operano su variabili booleane si chiamano **operatori logici**. Gli operatori logici elementari sono:

not, \neg , (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 NA gioca in serie A"

FALSO

x and y:

"Benevento è capoluogo di provincia e il NA gioca in serie A"

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

Esercizio



x: "Sette è un numero pari"

y: "Sei è un numero divisibile per tre"

Quanto vale x and y?

x: "L'estate è una stagione"

y: "Gennaio è il primo mese dell'anno"

Quanto vale x and y?

7

Tavole di verità: or



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

Posso scrivere:

x or y oppure

$x \vee y$ oppure

$x + y$

8

Esempi di or



x: "Benevento è capoluogo di provincia"
 y: "il NA gioca in serie A"
 z: "Benevento è capitale d'Italia"

VERO
 FALSO
 FALSO

x or y:

"Benevento è capoluogo di provincia oppure il NA gioca in serie A"

VERO

y or z:

"Il NA gioca in serie A oppure Benevento è capitale d'Italia"

FALSO

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

Esempio



Se a è vero
 b è falso
 c è falso

a or b and c ?

a or (b and c) vero

Esercizio



Se x è falso
 y è falso
 z è vero

x and z or not y ?



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

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

Le operazioni logiche sono:

not	→	!
and	→	&&
or	→	

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



```
#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
}
```



```
#include<stdio.h>
main()
{
    int a=0, b=1, k=2, ris;
    ris = k || !a;         // ris?
    ris = !a && b;         // ris?
    ris = a || b;         // ris?
    ris = !a && (b || k);  // ris?
    ris = !a && k||b;      // ris?
}
```



a && // omesso un operando

a & & b // spazio non permesso

Tipi ordinati



- Sono ordinati i tipi intero, reale e carattere
- Su un tipo ordinato sono definiti gli **operatori di relazione**

17

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

•Essi hanno tutti lo stesso livello di priorità, (tranne l'operatore di uguaglianza che ha precedenza più bassa) e associano da sinistra a destra

18

Esercizi

... sui booleani



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

20

Soluzione



```
(i>=1) && (i<=100)
```

con || è sbagliato

21

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);
}
```

22

succede ...



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

23

... 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);
}
```

24



- 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 $E1$.
- Se $E1$ è falsa non viene valutata $E2$ perché in ogni caso l'intera espressione sarà falsa.
- Se $E1$ è vera allora viene valutata $E2$.

25



$E1 \ || \ E2$

- 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

26



Se le variabili intere a , b e c hanno rispettivamente valore 5, 35 e 7, quale valore viene assegnato alla variabile intera ris dalle seguenti espressioni?

```
ris = a+b*c           // 250
ris = (a>b)           // 0
ris = (a+b) * (a<b)  // 40
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
```

27



- 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)*

28



```
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
```



Stampare la tabella booleana relativa all'operatore 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);
```



```
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 ~  
$ -
```

Esercizio



Stampare la tabella booleana relativa all' operatore `or` e `not`