

Algebra di Boole

Contenuti rieditati delle slide della
Prof. L. Caponetti

Algebra di Boole

- L'algebra di Boole descrive variabili ed operatori detti booleani
- Una variabile si dice booleana se può assumere solo 2 valori
- I valori costanti possono essere denotati con **vero/falso, V/F, 1/0, true/false, T/F**

Operatori dell'algebra di Boole

- Gli operatori possono essere denotati con:

not, \neg

and, \wedge

or, \vee

Tavola di verità

A	B	A and B	A or B
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	F

Contraddizione

- Una contraddizione è una espressione sempre falsa

$$A \text{ and } (\text{not } A) = F$$

A	not A	A and (not A)
V	F	F
F	V	F

Tautologia

- Una tautologia è una espressione sempre vera
 $A \text{ or } (\text{not } A) = V$

A	not A	A or (not A)
V	F	V
F	V	V

Involuzione

$$\text{not (not } A) = A$$

A	not A	not (not A)
V	F	V
F	V	F

Teoremi di De Morgan

- $\text{not } (A \text{ or } B) = (\text{not } A) \text{ and } (\text{not } B)$
- $\text{not } (A \text{ and } B) = (\text{not } A) \text{ or } (\text{not } B)$

Espressioni booleane semplici

- Con le variabili, costanti ed operatori booleani si possono costruire delle espressioni il cui valore è ancora booleano – **espressioni booleane semplici**
- Esempi: not B
not (A or B)

Espressioni relazionali

- Operatori relazionali:

= <> < <= > >=

- In una espressione relazionale gli operatori sono applicati ad espressioni semplici dello stesso tipo, ad esempio espressioni aritmetiche
- Il risultato è **vero** oppure **falso**

Espressioni booleane

- In generale una espressione booleana o proposizione è costituita da 2 termini separati da un operatore booleano
- Ad esempio **($x > 0$) and ($x < 100$)**
- **Ogni termine ha un valore di verità ottenuto mediante una espressione relazionale**

Esempi

- $\text{not } (a > b) = a \leq b$
- $(a > 5) \text{ and } (a \leq 10)$
- $\text{not } ((a > 5) \text{ and } (a \leq 10)) =$
 $\text{not } (a > 5) \text{ or } \text{not } (a \leq 10) =$
 $(a \leq 5) \text{ or } (a > 10)$

Variabili booleane

- I linguaggi di programmazione consentono di elaborare i valori di verità, vero e falso
- Il linguaggio Pascal fornisce il tipo predefinito Boolean per trattare con variabili ed operatori booleani
- Il linguaggio C non fornisce un tipo predefinito, considera:
 - Falso il **valore 0**
 - Vero, **un qualsiasi valore diverso da 0**

Operatori relazionali e logici nel linguaggio C

- Operatori relazionali

==	Uguale
!=	Diverso
<	Minore
<=	Minore uguale
>	Maggiore
>=	Maggiore uguale

- Operatori logici

&&	And
 	Or
!	Not

Asserzioni

- Le **asserzioni** in un linguaggio di programmazione sono delle **espressioni booleane**, che possono essere utilizzate per:
 - Documentare un programma, descrivendo la semantica del programma
 - Verificare un programma, provandone la correttezza formale

Asserzioni

- Le **asserzioni** possono essere precondizioni e postcondizioni
 - **Precondizione** o **premessa**, è una espressione booleana che deve essere verificata prima di una istruzione o blocco di istruzioni
 - **Postcondizione** o **conseguenza**, è una espressione booleana che deve essere verificata dopo una istruzione o blocco di istruzioni

Esempio - fattoriale

- Consideriamo l'algoritmo del fattoriale descritto mediante pseudocodifica ed introduciamo:
 - **Una preconditione**, espressione da verificare sui dati di input, prima della esecuzione dell'algoritmo
 - **Una postcondizione**, espressione da verificare dopo l'esecuzione dell'algoritmo
 - **Una asserzione invariante all'interno del ciclo**, espressione che deve sempre essere verificata nel ciclo

Esempio - fattoriale

```
/* Precondizione dell'algoritmo */  
/* PREC:  $n \geq 0$  */  
/* Postcondizione dell'algoritmo */  
/* POST:  $(fatt \geq 1)$  and  $(fatt = n!)$  */  
i=0  
fatt=1  
while (i<n)  
{  
    /* Asserzione invariante */  
    /* INV:  $(fatt = i!)$  and  $(i < n)$  */  
    i=i+1  
    fatt=fatt*i  
}
```

Quesiti

- **Le seguenti affermazioni sono corrette?**
 1. Le espressioni relazionali e logiche in C hanno un valore di tipo int che può essere 0 o 1
 2. L'operatore ! di negazione è unario
 3. L'espressione !a ha un valore di tipo int che può essere 0 oppure 1

Esercizio

- **Scrivere le espressioni equivalenti alle seguenti senza utilizzare la negazione**

1. $(a > b) =$

2. $(a \leq b \ \&\& \ c \leq d) =$

3. $(a+1 == b+1) =$

4. $(a < 1 \ || \ v < 2 \ \&\& \ c < 3)$