

Elementi di Informatica

Tipi ed Espressioni – Logica Booleana

Giordano Da Lozzo e Giuseppe Sansonetti



Algebra di Boole

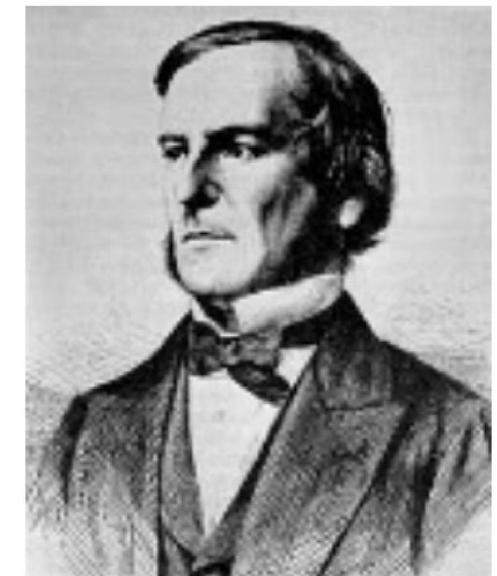
Un tipo astratto, formalizzato da George Boole.

Dominio: {True, False}

3 operatori:

- ! (operatore NOT o di complementazione o di negazione)
- && (operatore AND o di congiunzione)
- || (operatore OR o di disgiunzione)

!, && e || si chiamano operatori logici o booleani



Semantica dell'Operatore NOT

Sia $A \in \{\text{True}, \text{False}\}$

$\neg A$ vale True se A vale False

$\neg A$ vale False se A vale True

Esempi: $\neg \text{True}$ vale False

$\neg(\neg \text{False})$ vale False

A	$\neg A$
True	False
False	True

Tavola di verità

Semantica dell'Operatore AND

Siano $A, B \in \{\text{True}, \text{False}\}$

$A \&& B$ vale True se sia A che B valgono True

$A \&& B$ vale False altrimenti

Esempi: $\text{True} \&& (\text{!False})$ vale True

$(\text{True} \&& \text{False})$ vale False

A	B	$A \&& B$
True	True	True
True	False	False
False	True	False
False	False	False

Tavola di verità

Semantica dell'Operatore OR

Siano $A, B \in \{\text{True}, \text{False}\}$

$A \parallel B$ vale True se almeno uno fra A e B vale True

$A \parallel B$ vale False altrimenti

Esempi: $\text{False} \parallel (\text{!False})$ vale True

$(\text{True} \parallel \text{False})$ vale False

A	B	$A \parallel B$
True	True	True
True	False	True
False	True	True
False	False	False

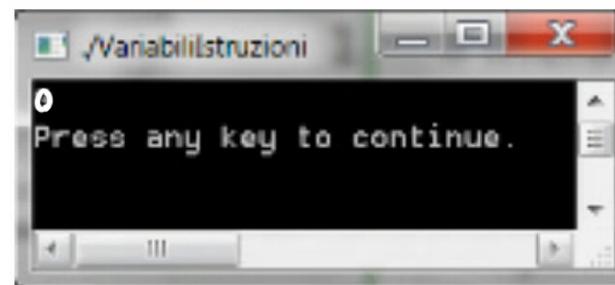
Tavola di verità

Algebra di Boole in C

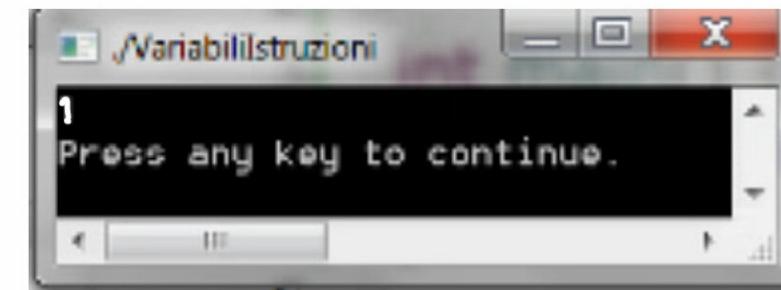
Nel Linguaggio C l'Algebra di Boole viene utilizzata nelle istruzioni di controllo (anche dette istruzioni strutturate) nella seguente maniera:

- Il valore False viene rappresentato da uno 0
- Il valore True viene rappresentato da un 1 (e da qualsiasi numero diverso da 0)
- Gli operatori !, && e || sono definiti nel linguaggio

```
int main() {  
    printf("%d", 1 && 0);  
}
```



```
int main() {  
    printf("%d", (1 && 0) || (1 || 0));  
}
```



Alcune proprietà degli operatori Booleani

Idempotenza di `&&` e `||`

$$A \text{ } \&\& \text{ } A = A$$

$$A \text{ } || \text{ } A = A$$

Commutatività di `&&` e `||`

$$A \text{ } \&\& \text{ } B = B \text{ } \&\& \text{ } A$$

$$A \text{ } || \text{ } B = B \text{ } || \text{ } A$$

Associatività di `&&` e `||`

$$A \text{ } \&\& \text{ } (B \text{ } \&\& \text{ } C) = (A \text{ } \&\& \text{ } B) \text{ } \&\& \text{ } C$$

$$A \text{ } || \text{ } (B \text{ } || \text{ } C) = (A \text{ } || \text{ } B) \text{ } || \text{ } C$$

Distributività

$$A \text{ } \&\& \text{ } (B \text{ } || \text{ } C) = (A \text{ } \&\& \text{ } B) \text{ } || \text{ } (A \text{ } \&\& \text{ } C)$$

$$A \text{ } || \text{ } (B \text{ } \&\& \text{ } C) = (A \text{ } || \text{ } B) \text{ } \&\& \text{ } (A \text{ } || \text{ } C)$$

Alcune proprietà degli operatori Booleani

Elementi neutri di `&&` e `||`

`TRUE && A = A`

`FALSE || A = A`

Elementi assorbenti di `&&` e `||`

`FALSE && A = FALSE`

`TRUE || A = TRUE`

Eliminazione della doppia negazione

`!(!A) = A`

Altre identità notevoli

`A && (!A) = FALSE`

`A || (!A) = TRUE`

Operatori Relazionali

Operatori **binari**, che si applicano ad una **coppia di valori** dello **stesso tipo** (ad esempio due valori di tipo *int*) per verificare l'esistenza di una certa **relazione**, e forniscono il risultato della verifica come valore booleano (quindi **0** o **1**)

$3 == 3$	vale 1	(uguale a)
$4 != 4$	vale 0	(non uguale a)
$5 < (11/2)$	vale 0	(minore di)
$5.5 > 5$	vale 1	(maggiore di)
$-1 <= 0$	vale 1	(minore o uguale a)
$-1/2 >= 0$	vale 1	(maggiore o uguale a)

Espressioni Logiche o Booleane

Anche dette **predicati** o **condizioni**, sono **espressioni** che hanno un valore **Vero** o **Falso** (in C, **1** o **0**) quando vengono valutate.

Sono composte da:

- **1, 0 e variabili booleane** (di tipo *int, short, long, float, ...*)
- **Operatori booleani** (**!**, **&&** e **||**)
- **Funzioni booleane**, ovvero funzioni che restituiscono un valore booleano (ad esempio gli **operatori relazionali**)

Esempio: $((x>10 \parallel (y==5)) \&\& (cond1 \parallel cond2))$

dove **x, y, cond1** e **cond2** sono **variabili** di tipo *int*

Precedenze

In ordine **decrescente** di precedenza

- NOT !
- Relazionali di confronto <, >, <=, >= (**fra loro precedenza a sinistra**)
- Relazionali di uguaglianza ==, != (**fra loro precedenza a sinistra**)
- AND &&
- OR ||

Le **parentesi tonde** possono fugare ogni dubbio.

$x \geq 10 \parallel 1 \And \neg cond1 \parallel x \neq 5$

è equivalente a

$((x \geq 10) \parallel (1 \And (\neg cond1))) \parallel (x \neq 5)$

Esercizi 1

1. Quanto vale l'espressione $1 \parallel 1 \And \And 0$?
2. Siano a e b variabili di tipo *int*. Quanto vale l'espressione $((a \parallel b) \And \And (1 \And \And b)) \parallel b) \And \And 0$?
3. Siano a e b variabili di tipo *int*. Quanto vale l'espressione $(a \And \And b) \parallel a$?
4. Siano a e b variabili di tipo *int*. Quanto vale l'espressione $(!a \And \And b) \parallel a$?

Esercizi 1a

Quanto vale l'espressione $1 \parallel 1 \&\& 0$?

$$1 \parallel 1 \&\& 0$$

=

$$1 \parallel (1 \&\& 0)$$

=

$$1 \parallel 0$$

=

$$1$$

Esercizi 1b

Siano a e b variabili di tipo *int*. Quanto vale l'espressione
 $((a \parallel b) \&\& (1 \&\& b)) \parallel b) \&\& 0$?

$((a \parallel b) \&\& (1 \&\& b)) \parallel b) \&\& 0$

=

X && 0

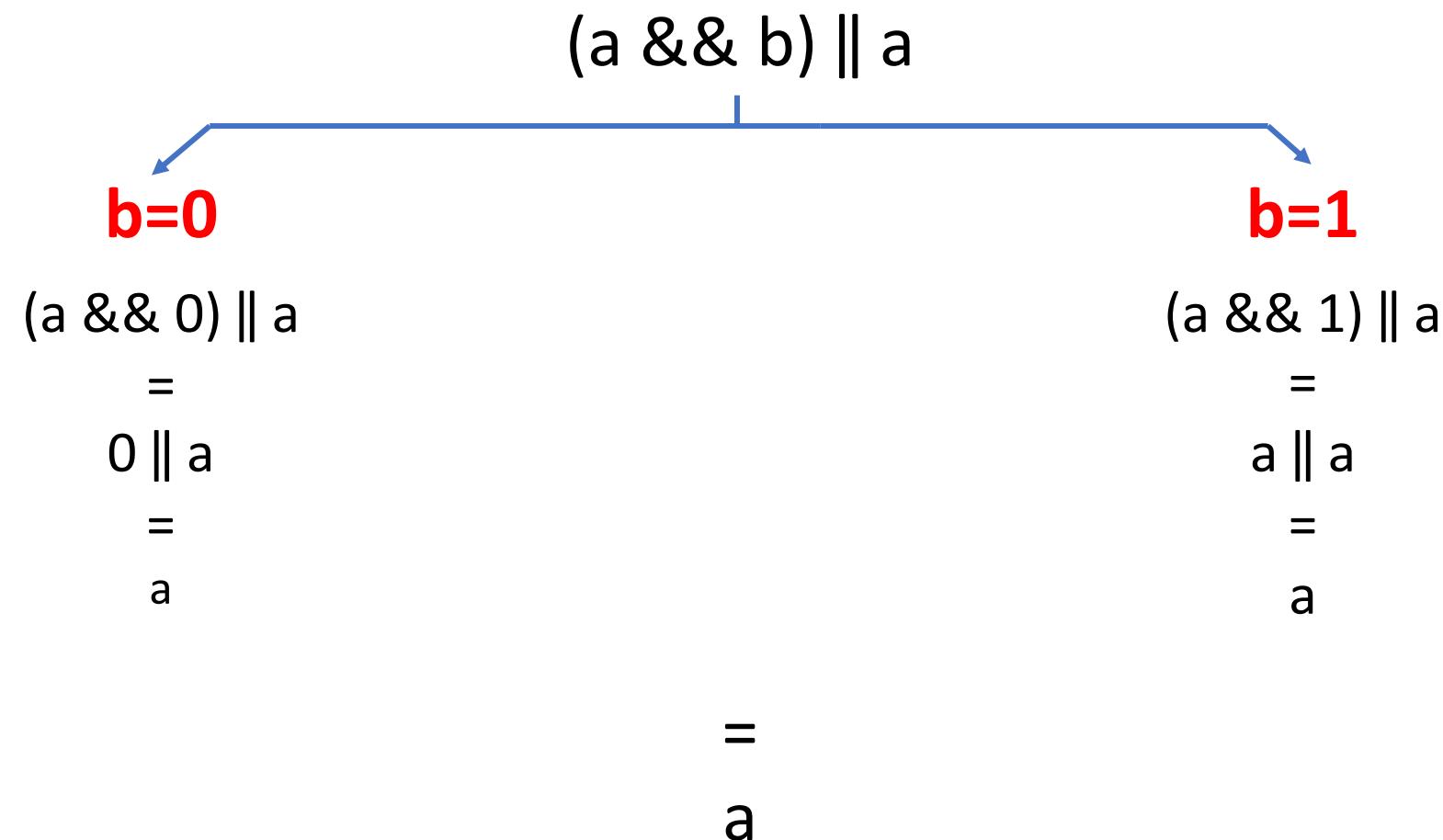
con $X = ((a \parallel b) \&\& (1 \&\& b)) \parallel b)$

=

0 (*indipendentemente dal valore di a e b*)

Esercizi 1c

Siano a e b variabili di tipo *int*. Quanto vale l'espressione $(a \&& b) \parallel a$?



Esercizi 1d

Siano a e b variabili di tipo *int*. Quanto vale l'espressione $(!a \&& b) \parallel a$?

$(!a \&& b) \parallel a$



b=0

$(!a \&& 0) \parallel a$

=

$0 \parallel a$

=

a

b=1

$(!a \&& 1) \parallel a$

=

$!a \parallel a$

=

1

=
 $a \parallel b$

Esercizi 1s

1. Quanto vale l'espressione $1 \parallel 1 \&\& 0$?
⇒ vale 1
2. Siano a e b variabili di tipo *int*. Quanto vale l'espressione $((a \parallel b) \&\& (1 \&\& b)) \parallel b \&\& 0$?
⇒ vale 0, quasiasi sia il valore di a e b
3. Siano a e b variabili di tipo *int*. Quanto vale l'espressione $(a \&\& b) \parallel a$?
⇒ vale 1 se e solo se a vale 1
4. Siano a e b variabili di tipo *int*. Quanto vale l'espressione $(!a \&\& b) \parallel a$?
⇒ vale 1 se e solo se almeno uno fra a e b vale 1

Esercizi 2

5. Sia x una variabile di tipo *int*. Quanto vale l'espressione $(x \leq 10) \parallel (x > 9)$?
6. Siano b e x variabili di tipo *int*.
Scrivere un'istruzione di assegnazione $b = \dots$; tale che b vale 1 se e solo se x è positivo.

Esercizi 2s

5. Sia x una variabile di tipo *int*. Quanto vale l'espressione $(x \leq 10) \parallel (x > 9)$?

⇒ vale 1, qualsiasi sia il valore di x

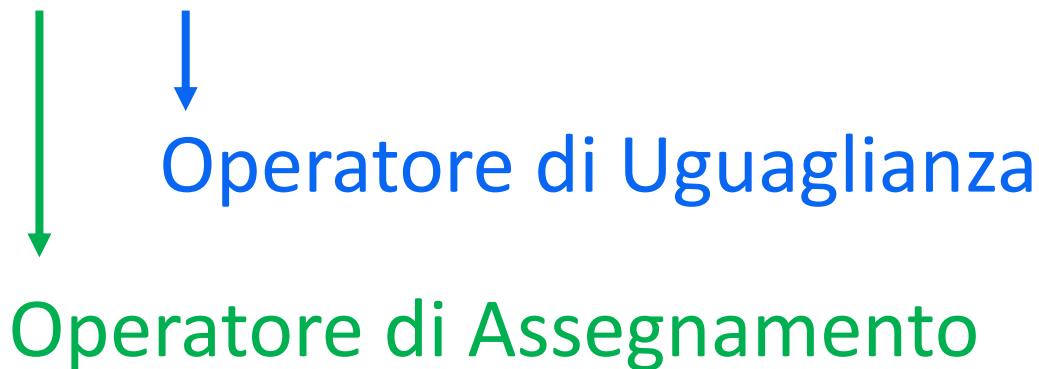
6. Siano b e x variabili di tipo *int*.

Scrivere un'istruzione di assegnazione $b = \dots$; tale che b vale 1 se e solo se x è positivo.

⇒ $b = (x > 0);$

Espressioni Booleane: Ulteriori Considerazioni

- Si può **assegnare** ad una variabile il risultato di una espressione booleana. Es: *int* numero = (x>0) || (!1);
- **Attenzione** alla differenza fra **=** e **==**



Leggi di De Morgan

Siano b_1, b_2, \dots, b_n espressioni di tipo *boolean*.

Allora:

1) $! (b_1 \&\& b_2 \&\& \dots \&\& b_n)$ equivale a $! b_1 \parallel ! b_2 \parallel \dots \parallel ! b_n$

La **negazione di una congiunzione di termini** equivale
alla **disgiunzione della negazione** dei termini

2) $! (b_1 \parallel b_2 \parallel \dots \parallel b_n)$ equivale a $! b_1 \&\& ! b_2 \&\& \dots \&\& ! b_n$

La **negazione di una disgiunzione di termini** equivale
alla **congiunzione della negazione** dei termini



Esempio: $! ((a>=0) \&\& (a<b))$ equivale a $! (a>=0) \parallel ! (a<b)$
che è anche equivalente a $(a<0) \parallel (a>=b)$

Esercizio per casa

Sia **anno** una **variabile intera** che rappresenta un anno. Scrivere una **espressione booleana**, in cui compare la variabile **anno**, che è vera se e solo se **anno** è **bisestile**.

- normalmente, **anno** è bisestile se è divisibile per 4 → cioè la divisione da resto 0
- ma se è anche divisibile per 100 allora non è bisestile → come nel caso dell'anno 1900
- nonostante le condizioni precedenti, se **anno** è divisibile per 400 allora è bisestile → come nel caso dell'anno 2000

Altre Risorse

- Bellini, Guidi: Linguaggio C — Capitoli 6.1 - 6.4, 8.1 - 8.2
Appendice D.5 - D.6, Appendice E