Algebra di Boole
ed
Elementi di Logica

Cenni all'algebra di Boole

- L'*algebra di Boole* (inventata da G. Boole, britannico, seconda metà '800), o *algebra della logica*, si basa su *operazioni logiche*
- Le operazioni logiche sono applicabili a *operandi logici*, cioè a operandi in grado di assumere solo i valori ***vero*** *e* ***falso***
- Si può rappresentare *vero* con il bit **1** e *falso* con il bit **0** (convenzione di *logica positiva*)

Operazioni logiche fondamentali

- Operatori logici binari (con 2 operandi logici)
  - Operatore **OR**, o *somma logica*
  - Operatore **AND**, o *prodotto logico*
- Operatore logico unario (con 1 operando)
  - Operatore **NOT**, o *negazione*, o *inversione*

- Poiché gli operandi logici ammettono due soli valori, si può definire compiutamente ogni operatore logico tramite una ***tabella*** di associazione operandi-risultato

Operatori logici di base e loro tabelle di verità

A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

(somma logica)

A	B	A and B
0	0	0
0	1	0
1	0	0
1	1	1

(prodotto logico)

A	not A
0	1
1	0

(negazione)

Le tabelle elencano tutte le possibili combinazioni in ingresso e il risultato associato a ciascuna combinazione

Espressioni logiche (o Booleane)

- Come le espressioni algebriche, costruite con:
  - Variabili logiche (letterali): p. es. A, B, C = 0 oppure 1
  - Operatori logici: and, or, not
- Esempi:

  A or (B and C)

  (A and (not B)) or (B and C)
- **Precedenza**: l'operatore "not" precede l'operatore "and", che a sua volta precede l'operatore "or"

  A and not B or B and C = (A and (not B)) or (B and C)
- Per ricordarlo, si pensi OR come "+" (più), AND come "×" (per) e NOT come "–" (cambia segno)

Tabelle di verità delle **espressioni logiche**

A	B	NOT ((A OR B) AND (NOT A))
0	0	1
0	1	0
1	0	1
1	1	1

Specificano i valori di verità
per tutti i possibili valori
delle variabili

Tabella di verità
di un'espressione logica

A and B or not C

A B C	**X** = A and B	**Y** = not C	**X** or **Y**			
0 0 0	0 and 0 = 0	not 0 = 1	0	or	1	**= 1**
0 0 1	0 and 0 = 0	not 1 = 0	0	or	0	**= 0**
0 1 0	0 and 1 = 0	not 0 = 1	0	or	1	**= 1**
0 1 1	0 and 1 = 0	not 1 = 0	0	or	0	**= 0**
1 0 0	1 and 0 = 0	not 0 = 1	0	or	1	**= 1**
1 0 1	1 and 0 = 0	not 1 = 0	0	or	0	**= 0**
1 1 0	1 and 1 = 1	not 0 = 1	1	or	1	**= 1**
1 1 1	1 and 1 = 1	not 1 = 0	1	or	0	**= 1**

Due esercizi

A	B	NOT	((A	OR	B)	AND	(NOT	A))
0	0	**1**	0	0	0	0	1	0
0	1	**0**	0	1	1	1	1	0
1	0	**1**	1	1	0	0	0	1
		1	1	1	1	0	0	1

A	B	C	(B	OR	NOT	C)	AND	(A	OR	NOT	C)
0	0	0	0	1	1	0	**1**	0	1	1	0
0	0	1	0	0	0	1	**0**	0	0	0	1
0	1	0	1	1	1	0	**1**	0	1	1	0
0	1	1	1	1	0	1	**0**	0	0	0	1
1	0	0	0	1	1	0	**1**	1	1	1	0
1	0	1	0	0	0	1	**0**	1	1	0	1
1	1	0	1	1	1	0	**1**	1	1	1	0
1	1	1	1	1	0	1	**1**	1	1	0	1

A che cosa servono le espressioni logiche?

- A *modellare* alcune (non tutte) forme di *ragionamento*
  - A = è vero che 1 è maggiore di 2 ? (sì o no, qui è no) = 0
  - B = è vero che 2 più 2 fa 4 ? (sì o no, qui è sì) = 1
  - A and B = è vero che 1 sia maggiore di 2 e che 2 più 2 faccia 4 ?
    Si ha che A and B = 0 and 1 = 0, dunque no
  - A or B = è vero che 1 sia maggiore di 2 o che 2 più 2 faccia 4 ?
    Si ha che A or B = 0 and 1 = 1, dunque sì
- OR, AND e NOT vengono anche chiamati *connettivi logici*, perché funzionano come le congiunzioni coordinanti "o" ed "e", e come la negazione "non", del linguaggio naturale
- Si modellano ragionamenti (o *deduzioni*) basati solo sull'uso di "o", "e" e "non" (non è molto, ma è utile)

Che cosa **non** si può modellare tramite espressioni logiche?

- Le espressioni logiche (booleane) *non modellano*:
  - Domande *esistenziali*: "***c'è almeno*** un numero reale x tale che il suo quadrato valga −1 ?" (si sa bene che *non c'è*)
    $\exists x \mid x^2 = -1$ è falso
  - Domande *universali*: "***ogni*** numero naturale è la somma di quattro quadrati di numeri naturali ?" (si è dimostrato *di sì*)
    $\forall x \mid x = a^2+b^2+c^2+d^2$ è vero ("teorema dei 4 quadrati")
    Più esattamente andrebbe scritto: $\forall x\ \exists a,b,c,d \mid x = a^2+b^2+c^2+d^2$
- $\exists$ e $\forall$ sono chiamati "operatori di quantificazione", e sono ben diversi da or, and e not
- La parte della logica che tratta solo degli operatori or, and e not si chiama ***calcolo proposizionale***
- Aggiungendo gli operatori di quantificazione, si ha il ***calcolo dei predicati*** (che è molto più complesso)

Tautologie e Contraddizioni

- *Tautologia*
  - Una espressione logica che è sempre **vera**, per qualunque combinazione di valori delle variabili
    - Esempio: principio del “terzo escluso”: **A or not A** (*tertium non datur*, non si dà un terzo caso tra l’evento A e la sua negazione)
- *Contraddizione*
  - Una espressione logica che è sempre **falsa**, per qualunque combinazione di valori delle variabili
    - Esempio: principio di “non contraddizione”: **A and not A** (l’evento A e la sua negazione non possono essere entrambi veri)

Equivalenza tra espressioni

- Due espressioni logiche si dicono ***equivalenti*** (e si indica con ⇔) **se hanno la medesima tabella di verità**. La verifica è *algoritmica*. Per esempio:

A B	not A and not B	⇔	not (A or B)
0 0	1 and 1 = 1		not 0 = 1
0 1	1 and 0 = 0		not 1 = 0
1 0	0 and 1 = 0		not 1 = 0
1 1	0 and 0 = 0		not 1 = 0

- Espressioni logiche equivalenti modellano gli stessi *stati di verità* a fronte delle medesime variabili

Proprietà dell'algebra di Boole

- L'algebra di Boole gode di svariate *proprietà*, formulabili sotto specie di *identità*
  - (cioè formulabili come equivalenze tra espressioni logiche, valide per qualunque combinazione di valori delle variabili)
- Esempio celebre: le "Leggi di De Morgan"

not (A **and** B) = not A **or** not B (1ª legge)

not (A **or** B) = not A **and** not B (2ª legge)

Ancora sulle proprietà

- Alcune proprietà somigliano a quelle dell'algebra numerica tradizionale:
  - Proprietà *associativa*: A or (B or C) = (A or B) or C (idem per AND)
  - Proprietà *commutativa*: A or B = B or A (idem per AND)
  - Proprietà *distributiva* di AND rispetto a OR:
    A and (B or C) = A and B or A and C
  - … e altre ancora
- Ma parecchie altre sono alquanto insolite…
  - Proprietà *distributiva* di OR rispetto a AND:
    A or B and C = (A or B) and (A or C)
  - Proprietà di *assorbimento* (A assorbe B):
    A or A and B = A
  - *Legge dell'elemento* 1: not A or A = 1
  - … e altre ancora

Uso delle proprietà

- *Trasformare* un'espressione logica in un'altra, differente per aspetto ma equivalente:

not A and B or A =	(assorbimento)
= not A and B or (A or A and B) =	(togli le parentesi)
= not A and B or A or A and B =	(commutativa)
= not A and B or A and B or A =	(distributiva)
= (not A or A) and B or A =	(legge dell'elemento 1)
= ***true*** and B or A =	(vero and B = B)
= B or A è più semplice dell'espressione originale !	

- Si *verifichi* l'equivalenza con le tabelle di verità!
- Occorre conoscere un'ampia lista di proprietà e si deve riuscire a "vederle" nell'espressione (qui è il difficile)