Algebra di Boole

Cenni all'Algebra di Boole

Introduzione
Rappresentazione di una funzione combinatoria
Proprietà dell'algebra di commutazione
Forme canoniche
Teorema di espansione di Shannon

Versione del 19/09/03

Algebra Booleana: *sistema algebrico*

- Operazione:
  - una operazione α sull'insieme `S={s1,s2,...}` è una funzione che da `SxS` (`S` cartesiano `S`) porta in `S`.
  - Quindi, per cui ad ogni coppia ordinata appartenente ad `SxS` corrisponde un elemento di `S`, cioè

$$\alpha: SxS \rightarrow S.$$

  - Alcune considerazioni:
    - L'operazione `*` (di moltiplicazione) sull'intervallo `[0,1]` consente di ottenere un valore incluso in `[0,1]` a partire da elementi inclusi in `[0,1]`
    - La sottrazione sull'insieme dei naturali non è una operazione.
      - Es: `5-10` non appartiene ai naturali.
- Sistema Algebrico:
  - Combinazione di un insieme e di una o più operazioni.
    - esempio: `([0,1], *)` è un sistema algebrico.

Algebra Booleana: *introduzione (1)*

- Algebra Booleana B:
  è un sistema algebrico identificato dalla quintupla `(B,+,*,0,1)` dove:
  - `B` è l'insieme su cui vengono definite le operazioni
  - `+,*` sono le operazioni OR e AND (operazioni a due elementi)
  - `0,1` sono elementi speciali di B.
    - `0` è l'elemento neutro rispetto a +
    - `1` è l'elemento neutro rispetto a *
- Esempio:
  - Algebra Booleana a due valori: `({0,1},+,*,0,1)` dove + e * sono definiti come

+	0	1
0	0	1
1	1	1

*	0	1
0	0	0
1	0	1

Algebra Booleana: *introduzione (2)*

- Le *variabili* dell'algebra booleana a due valori possono assumere solo i due valori 0 e 1
  - precisamente, se `x` indica una variabile, è
    - $x = 0$ se e solo se $x \neq 1$
    - $x = 1$ se e solo se $x \neq 0$
- Si considera inoltre una *operazione a un solo elemento* (unary operation) detta *complementazione* o *negazione* (`NOT`), definita come

!	
0	1
1	0

  - Nota: il simbolo associato al `NOT` è spesso indicato come `'` (esempio `x'`), `!`(esempio `!x`) o sopra segnando la variabile.

Algebra di Commutazione

- L'*Algebra di Commutazione* è un'algebra booleana a due elementi.
- Una funzione di commutazione a n variabili è una funzione del tipo:

$$f: \{0,1\}^n \rightarrow \{0,1\}$$

- *"Tra tutte le algebre booleane, l'algebra booleana a due valori........è la più utile. Essa è la base matematica della analisi e progetto di circuiti di commutazione che realizzano i sistemi digitali."*
  - [Lee, S.C., *Digital Circuit And Logic Design*. Englewood Cliffs, NJ: Prentice-Hall, 1976]

Algebra di Commutazione: *rappresentazione di una funzione*

- Una funzione di commutazione a n variabili $f: \{0,1\}^n \rightarrow \{0,1\}$ può essere rappresenta in modo comodo utilizzando una *tabella della funzione* o tabella della verità
  - Una tabella della verità specifica la relazione che esiste tra ogni elemento del dominio di f ($\{0,1\}^n$) e la corrispondente immagine (elemento del codominio)
- Esempio:

a b c	f(a,b,c)
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	1
1 0 0	1
1 0 1	1
1 1 0	0
1 1 1	1

Algebra di Commutazione: *definizioni (1)*

- Letterale:
  - un letterale è una coppia (Variabile,Valore)
  - (x,1) è indicato come x (variabile in forma naturale);
  - (x,0) rappresenta la *variabile x* in forma *negata* (complementata) ed è indicato come x′ (oppure !x).
  - In modo equivalente, dato $a \in \{0,1\}$ un letterale è espresso come x^a dove, per $a=1$ $x^a = x$ e per $a=0$ $x^a = x'$.
    - Ad esempio, il letterale z vale 1 ogni qual volta che la variabile z vale 1, mentre il letterale z′ vale 1 ogni qual volta che la variabile z vale 0.

Algebra di Commutazione: *definizioni (2)*

- Termine prodotto:
  - Un termine prodotto è il *prodotto logico o disgiunzione* (AND) di più letterali.
  - Un termine prodotto in cui compaiono letterali corrispondenti a tutte le variabili della funzione e tale per cui la configurazione di valori delle variabili definite dai letterali genera un valore 1 della funzione stessa nella tabella delle verità, costituisce un **mintermine** della funzione (spesso si sottintende il segno *)
    - Ad esempio, a′b′c e ab′c rappresentano due mintermini della funzione di cui si è prima data la tabella delle verità
  - Un termine prodotto in cui compaiono solo alcuni dei letterali e che corrisponda a un *insieme* di 1 della funzione è denominato **implicante**.
    - Ad esempio, a′c rappresenta un implicante della funzione data

Algebra di Commutazione: *Definizioni (3)*

- Termine somma (duale):
  - Un termine somma è la *somma logica o congiunzione* (`OR`) di più letterali.
  - Un termine somma in cui compaiono letterali corrispondenti a tutte le variabili della funzione e tale per cui la configurazione di valori delle variabili definite dai letterali genera un valore 0 della funzione stessa nella tabella delle verità, costituisce un ***maxtermine*** della funzione
    - Ad esempio, `a+b+c` e `a+b'+c` rappresentano due *maxtermini* della funzione data
  - Un termine somma in cui compaiono solo alcuni dei letterali e che corrisponda a un *insieme* di 0 della funzione è denominato **implicato**.
    - Ad esempio, `a+c` rappresenta un implicato della funzione data

Algebra di Commutazione: *proprietà (1)*

- 1: elemento neutro
  - `a+1=1` `a*0=0` (elemento nullo)
  - `a+0=a` `a*1=a` (identità)
- 2: idempotenza
  - `a+a=a` `a*a=a`
- 3: inverso
  - `a+a'=1` `a*a'=0`
- 4: commutativa
  - `a+b=b+a` `a*b=b*a`
- 5: associativa
  - `a+(b+c)=(a+b)+c` `a*(b*c)=(a*b)*c`

Algebra di Commutazione: *proprietà (2)*

- 6: distributiva
  - a*(b+c) = a*b + b*c   a+(b*c)=(a+b)*(a+c)
    - Nota: vale per la somma rispetto al prodotto come per il prodotto rispetto alla somma - non esiste precedenza fra le due operazioni, occorre sempre immaginare le parentesi "sottintese" intorno a ogni applicazione di un'operazione.
- 7: assorbimento
  - a+(a*b)=a   a*(a+b)=a
- 8: Leggi di De Morgan
  - (a+b)' = a'*b'   (a*b)' = a'+b'
- 9:
  - a+a'b = a+b   a*(a'+b) = a*b
- 10: consenso
  - a*b+a'*c+b*c = a*b + a'*c
  - (a+b)*(a'+c)*(b+c)=(a+b)*(a'+c)

Algebra di Commutazione: *proprietà (3)*

- 11: principio di dualità
  - Ogni identità deducibile dai postulati dell'algebra di Boole è trasformata in un'altra identità se:
  1. Ogni operazione + viene sostituita da una operazione * e vice versa.
  2. Ogni elemento identità 0 viene sostituito da un elemento identità 1 e vice versa.
  - Esempio: (assorbimento)
    - a+(a*b)=a
    - a*(a+b)=a

Algebra di Commutazione: *proprietà (4)*

- Il modo più semplice per dimostrare le proprietà è quello esaustivo (si dimostra per tutti i possibili valori di tutte le variabili).

- Sono possibili altri tipi di dimostrazione:
  - Ad esempio, si voglia dimostrare `a+a'b = a+b`
    - Si sostituisce `a` con `a*1`
    - Dalla proprietà della negazione (`b+b'=1`) applicata *da destra verso sinistra* si sostituisce `a1 + a'b` con `a(b+b')+a'b`
    - Applicando la proprietà distributiva si ottiene `ab+ab'+a'b`
    - Applicando la proprietà di idempotenza *da destra verso sinistra* al termine `ab` si ottiene `ab + ab + ab' + a'b`
    - Applicando la proprietà distributiva *da destra verso sinistra* si ottiene `a(b+b') + b(a + a')`
    - Applicando le proprietà delle negazione si ha infine `a*1 + b*1= a + b`

Algebra di Commutazione: *funzioni*

- Una *funzione booleana* di `n` variabili può essere espressa attraverso una *espressione booleana* di `n` variabili costituita da letterali, costanti, operatori `AND`, `OR` e `NOT`.
  - Esempio di espressione booleana: `f(a,b,c)=ab+a'c'`
- Le proprietà dell'algebra di commutazione possono essere utilizzate per manipolare una espressione booleana ed ottenerne una equivalente.
  - Due *espressioni booleane* sono *equivalenti* se e solo se sono riconducibili alla stessa *funzione booleana*.
- Esempio:
  - `f(a,b,c)= (a'*b')'*a = (a+b)*a = a`

De Morgan

Assorbimento

Algebra di commutazione: *espressioni e funzioni (1)*

- Il numero di *espressioni booleane* di n variabili definite su una *algebra booleana* B è **infinito**.
  - La relazione tra *espressioni booleane* e *funzioni booleane* **non è 1 a 1**.
- Esempio:

a b c	f(a,b,c)
0 0 0	0
0 0 1	0
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	1

f(a,b,c)= (a'*b')'*a

f(a,b,c)= ...

f(a,b,c)= a

Algebra di commutazione: *espressioni e funzioni (2)*

- Data una *funzione booleana* – ad esempio, mediante la tabella delle verità - il problema è identificare almeno una *espressione booleana* ad essa corrispondente
  - In molte applicazioni dell'*algebra booleana* uno scopo fondamentale è determinare una *buona* rappresentazione della *funzione booleana*, avendo preventivamente definito il concetto di *buono ed un modo per valutarlo* : *obiettivo e cifra di merito*
    - Ad esempio: ***l'obiettivo è minimizzare il costo del circuito** corrispondente a un'espressione*, la cifra di merito usata è il *numero di letterali* presenti nell'espressione.
  - Solitamente la *buona* rappresentazione algebrica viene ricavata manipolando una soluzione iniziale.

Algebra di commutazione: *espressioni e funzioni (3)*

- Data una *funzione booleana*, la soluzione iniziale al problema di determinare una sua espressione consiste nel ricorso alle *forme canoniche*.
- Le forme canoniche sono, rispettivamente, la forma *somma di prodotti* (SoP) e quella *prodotto di somme* (PoS).
- Data una funzione boolena esistono una ed una sola forma canonica SoP ed una e una sola forma PoS che la rappresenta.

Algebra Booleana: *Forme canoniche*

- Si consideri il seguente esempio:

a b	f(a,b)
0 0	0
0 1	1
1 0	0
1 1	1

- È intuitivo osservare che la funzione possa essere ottenuta dal `OR` delle seguenti funzioni:

a b	f(a,b)
0 0	0
0 1	1
1 0	0
1 1	1

=

a b	f_1(a,b)
0 0	0
0 1	1
1 0	0
1 1	0

+

a b	f_2(a,b)
0 0	0
0 1	0
1 0	0
1 1	1

Algebra Booleana: *Forme canoniche*

- Per cui, intuitivamente, si ottiene:

a b	f(a,b)
0 0	0
0 1	1
1 0	0
1 1	1

=

a b	f_1(a,b)
0 0	0
0 1	1
1 0	0
1 1	0

+

a b	f_2(a,b)
0 0	0
0 1	0
1 0	0
1 1	1

$f_1(a,b)=a'b$ $\quad$ $f_2(a,b)=ab$

  - Poiché, ad esempio, quando `a=0` e `b=1` il prodotto `a'b` assume valore 1 mentre vale 0 in tutti gli altri casi.

Algebra Booleana: *Forme canoniche*

- Ne consegue:

a b	f(a,b)
0 0	0
0 1	1
1 0	0
1 1	1

=

a b	f_1(a,b)
0 0	0
0 1	1
1 0	0
1 1	0

+

a b	f_2(a,b)
0 0	0
0 1	0
1 0	0
1 1	1

$$f(a,b) = a'b + ab$$

- Mettendo in `OR` i ***mintermini*** della funzione si ottiene l'*espressione booleana* della funzione stessa espressa come somma di prodotti. Questa *espressione booleana* è denominata ***prima forma canonica***.
  - Si ricorda che nel *mintermine* una variabile compare nella forma naturale `x` se nella corrispondente configurazione di ingresso ha valore `1`, nella forma complementata `x'` se ha valore `0`

Algebra Booleana: *Forme canoniche*

- Esempio:

a b c	f(a,b,c)
0 0 0	0
0 0 1	1
0 1 0	1
0 1 1	1
1 0 0	1
1 0 1	0
1 1 0	0
1 1 1	1

f(a,b,c)=a'b'c+a'bc'+a'bc+ab'c'+abc

Prima Forma Canonica

Algebra Booleana: *Forme canoniche*

- Si consideri nuovamente lo stesso esempio:

a b	f(a,b)
0 0	0
0 1	1
1 0	0
1 1	1

- È intuitivo osservare che la funzione possa essere ottenuta dall'AND delle seguenti funzioni:

a b	f(a,b)
0 0	0
0 1	1
1 0	0
1 1	1

=

a b	f_1(a,b)
0 0	0
0 1	1
1 0	1
1 1	1

*

a b	f_2(a,b)
0 0	1
0 1	1
1 0	0
1 1	1

Algebra Booleana: *Forme canoniche*

- Per cui, intuitivamente, si ottiene:

a b	f(a,b)
0 0	0
0 1	1
1 0	0
1 1	1

=

a b	$f_1(a,b)$
0 0	0
0 1	1
1 0	1
1 1	1

*

a b	$f_2(a,b)$
0 0	1
0 1	1
1 0	0
1 1	1

$f_1(a,b)=(a'b')'$ $f_2(a,b)=(ab')'$

- Infatti, ad esempio, quando a=0 e b=0 il termine (a'b')' assume valore 0 mentre vale 1 in tutti gli altri casi.
- o anche

$f_1(a,b)= a+b$ $f_2(a,b)= a'+b$

Algebra Booleana: *Forme canoniche*

- Applicando le leggi di De Morgan, si ottiene le seguente trasformazione:

$f_1(a,b)=(a'b')'$ $f_2(a,b)=(ab')'$

$f_1(a,b)=(a+b)$ $f_2(a,b)=(a'+b)$

$f(a,b) = (a+b)*(a'+b)$

- Mettendo in AND i ***maxtermini*** della funzione si ottiene l'*espressione booleana* della funzione stessa espressa come prodotto di somme. Questa *espressione booleana* è denominata ***seconda forma canonica***.
  - Si ricorda che nel *maxtermine* una variabile compare nella forma naturale x se nella corrispondente configurazione di ingresso ha valore 0, nella forma complementata x' se ha valore 1

Algebra Booleana: *Forme canoniche*

- Esempio:

a	b	c	f(a,b,c)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$f(a,b,c)=(a+b+c)*(a'+b+c')*(a'+b'+c)$

Seconda Forma Canonica

Algebra Booleana: *Espansione di Shannon*

- Formalmente, quanto esposto dal punto di vista intuitivo produce le forme canoniche come segue:
  - **prima forma canonica**:
    - $f = (x_1'\ldots x_n')*f(0,\ldots,0)+(x_1'\ldots x_n)*f(0,\ldots,1)+\ldots+(x_1\ldots x_n)*f(1,\ldots,1)$ dove
      - $(x_1'\ldots x_n')$, $(x_1'\ldots x_n)$, …, $(x_1\ldots x_n)$ sono i *mintermini* della funzione f,
      - $f(0,\ldots,0)$, …., $f(1,\ldots,1)$ sono i *valori* che la funzione assume quando la configurazione delle variabili sia, rispettivamente, $(0,\ldots,0),\ldots,(1,\ldots,1)$
  - **seconda forma canonica**:
    - $f=((x_1'+\ldots+x_n')+f(1,\ldots,1))*((x_1'+\ldots+x_n)+f(1,\ldots,0))*\ldots*((x_1+\ldots+x_n)+f(0,\ldots,0))$ dove
      - $(x_1'+\ldots+x_n')$, $(x_1'+\ldots+x_n)$, …, $(x_1+\ldots+x_n)$ sono i *maxtermini* di f.
  - Nota: *f(0,0,…,0), f(0,0,…,1) . . . f(1,1,…,1)* sono noti con il nome di *discriminante della funzione f e* il loro valore appartiene a B.

Algebra Booleana: *Espansione di Shannon*

- La descrizione formale introdotta in precedenza deriva direttamente dall'applicazione iterativa del *Teorema di espansione di Shannon*
  - se f: $B^n \rightarrow B$ è una *funzione booleana* si ha

    $$f(x_1,x_2,...,x_n) = x_1'*f_{x1'} + x_1*f_{x1}$$

    per ogni $(x_1,x_2,...,x_n)$ in B^n.

    •Ad esempio, `f(a,b,c)=a'*f(0,b,c)+a*f(1,b,c)`
  - Dualmente, se f: $B^n \rightarrow B$ è una *funzione booleana* si ha

    $$f(x_1,x_2,...,x_n) = (x_1'+f_{x1})*(x_1+f_{x1'})$$

    per ogni $(x_1,x_2,...,x_n)$ in B^n.

    •Ad esempio, `f(a,b,c)=(a'+f(1,b,c))*(a+f(0,b,c))`

Algebra Booleana: *Espansione di Shannon e Forme canoniche*

- Ad esempio:

```
f(a,b,c)=a'b'c'*f(0,0,0)+a'b'c*f(0,0,1)+a'bc'*f(0,1,0)
         +a'bc*f(0,1,1)+ab'c'*f(1,0,0)+ab'c*f(1,0,1)+
         +abc'*f(1,1,0)+abc*f(1,1,1)
```

a b c	f(a,b,c)
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	1
1 0 0	1
1 0 1	1
1 1 0	0
1 1 1	1

Prima Forma Canonica

`a'b'c+a'bc+ab'c'+ab'c+abc`

`(a+b+c)*(a+b'+c)*(a'+b'+c)`

Seconda forma Canonica

Algebra Booleana: *Espansione di Shannon*

- Osservazione: il teorema di espansione di Shannon può essere utilizzato anche su espressioni Booleane. Esempio:
  - Espandendo rispetto ad `a` l'espressione booleana `f(a,b,c)=ab+b'+a'bc'`, si ha la forma equivalente `f(a,b,c)=a'*(b'+bc')+ a*(b+b')=a'b'+a'bc'+ab + ab'` `= a'b'+a'bc'+a`
  - Espandendo rispetto ad `a` `b` e `c` la espressione booleana `f(a,b,c)=ab+b'+a'bc'`, si ha la forma equivalente
  - `f(a,b,c)=a'*(b'+bc')+ a*(b+b')`
    `=a'*(b'*(1)+b*(c'))+ a*(b'*(1)+b*(1))`
    `=a'*(b'*(c'+c)+b*(c'))+a*(b'(c'+c)+b*(c'+c))`

    `=a'b'c'+a'b'c+a'bc'+ab'c'+ab'c+abc'+abc`
    - » è la prima forma canonica della funzione associata alla espressione booleana di partenza.

Algebra Booleana: *Manipolazione delle espressioni (1)*

- Data un'espressione di una funzione booleana, le proprietà dell'algebra di commutazione permettono di manipolarla in modo da ottenere un'espressione equivalente, ma di forma diversa
  - eventualmente con caratteristiche meglio rispondenti a particolari requisiti.
- Esempio:
  - sia data la forma canonica
    - `f(x,y,z) = x'yz' + xyz' + xyz`
  - e sia data la funzione di costo costituita dal *numero di letterali presenti* – che in questo caso vale 9.
  - Obiettivo: ridurre il *costo*.

Algebra Booleana: *Manipolazione delle espressioni (2)*

- Una prima manipolazione mediante le regole dell'algebra dà:
  - `f(x,y,z) = x'yz' + xyz' + xyz`

1. applicando la proprietà distributiva e quella della complementazione:
   - `f(x,y,z)= (x'+ x)yz'+xyz= 1yz'+xyz= yz'+xyz.`
2. poi, applicando di nuovo la proprietà distributiva
   - `f = y(z'+xz)`
3. E ricordando che `a + a'b = a + b,` si ottiene infine
   - `f= y(z'+x)= yz'+xy`

Algebra Booleana: *Manipolazione delle espressioni (3)*

- Allo stesso risultato si sarebbe giunti anche:
  - `f(x,y,z) = x'yz' + xyz' + xyz`

1. applicando dapprima la proprietà dell'idempotenza:
   - `f(x,y,z) = x'yz' + xyz' + xyz' + xyz`
2. poi applicando la proprietà distributiva
   - `f = yz'(x'+x) + xy (z' + z)`
3. Da cui infine
   - `f = yz'1 +xy1 = yz' + xy`

- Si osservi che, rispetto alla forma canonica di partenza, l'espressione logica ottenuta è di costo inferiore (*4 letterali*).

Algebra Booleana: *Manipolazione delle espressioni (5)*

- Si osservi che l'applicazione delle trasformazioni algebriche non permette di identificare una procedura sistematica
- Come conseguenza:
  - Non è possibie identificare un algoritmo
    - non possono essere realizzati strumenti CAD che consentano di produrre una soluzione ottima a due livelli utilizzando le proprietà dell'algebra
  - Non è possibile sapere se l'espressione ottenuta è quella minima
    - L'immediatezza della bontà del risultato dipende molto dalla scelta delle proprietà da applicare e dall'ordine in cui sono applicate.
- In pratica, non è questa la via che si sceglie!