Università degli studi di Parma
Dipartimento di Ingegneria dell'Informazione

Algebra di Boole

Cenni

(versione del 1 ottobre 2002)

Introduzione

Rappresentazione di una funzione combinatoria

Proprietà dell'algebra di commutazione

Forme canoniche

Teorema di espansione di Shannon

Introduzione

- Operazione
  - Una operazione α sull'insieme $S=\{s_1,s_2,...\}$ è una funzione che da S x S porta in S, cioè:

$$\alpha: S \times S \rightarrow S$$

  - Esempio:
    - 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]
- Sistema algebrico:
  - Combinazione di un insieme e di una o più operazioni.
  - Esempio:
    - ([0,1], *) è un sistema algebrico

Introduzione

- Algebra Booleana B:
  - È un sistema algebrico identificato dalla quintupla:

    $(B, \oplus, \odot, O, I)$
  - In cui:
    - B è l'insieme di specificazione (carrier)
    - $\oplus$, $\odot$ sono operatori binari (operazioni a due elementi)
      - Associano a coppie di elementi in B un elemento dello stesso insieme
      - $\oplus$ è il simbolo associato all'operazione *somma*
      - $\odot$ è il simbolo associato all'operazione *prodotto*
    - O, I sono elementi speciali di B

Introduzione

- Le proprietà dei due operatori dedotte da assiomi
  - $\oplus$ (simbolo *somma*) e $\odot$ (simbolo *prodotto*) sono commutative per ogni x, y $\in$ B
  - La somma è distributiva rispetto al prodotto ed il prodotto è distributivo rispetto alla somma per ogni x, y, z $\in$ B
  - O è l'elemento neutro rispetto a $\oplus$ ed I è l'elemento neutro rispetto a $\odot$
  - Ogni elemento x $\in$ B ammette un elemento x' complemento di x tale che (x $\oplus$ x') = I e (x $\odot$ x') = O
    - L'elemento x' complemento di x è unico.
    - Questa proprietà autorizza l'inserimento del nuovo operatore *complemento* (~) intendendo con ~x l'elemento x' complemento di x
      - ~ è il simbolo associato all'operazione *complemento*

Algebra di Commutazione

- Algebra Booleana a due valori (o di *commutazione*)
  - È definita da

    ({0, 1}, $\oplus$, $\odot$, ~, 0, 1)
  - In cui
    - L'insieme di specificazione B = {0,1}
    - le operazioni $\oplus$, $\odot$ e ~ sono definite come:

$\oplus$ (a \ b)	0	1
0	0	1
1	1	1

$\odot$ (a \ b)	0	1
0	0	0
1	0	1

~ (a)	
0	1
1	0

- Nota: D'ora in poi si farà riferimento all'algebra di commutazione.

Algebra di Commutazione

"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
Prentice-Hall, 1976

Algebra di Commutazione

- Definizioni:
  - Un letterale è una coppia (variabile,valore)
    - (x,1) È indicato come x mentre (x,0) È indicato come ~x
  - Un termine prodotto è il prodotto logico o disgiunzione (AND) di più letterali
    - Ad esempio, una funzione nelle variabili a,b,c e d potrebbe avere una espressione che contiene i termini prodotto $(\sim a \odot b \odot \sim c)$ e $(a \odot \sim b \odot c)$.
  - Un termine somma è la somma logica o congiunzione (OR) di più letterali
    - Ad esempio, una funzione nelle variabili a,b,c e d potrebbe avere una espressione che contiene i termini somma $(\sim a \oplus b \oplus \sim c)$ e $(a \oplus \sim b \oplus c)$.

Algebra di Commutazione

- Espressione Boolana
  - Una espressione Booleana E è definita in modo induttivo come parola composta da operatori booleani, parentesi, costanti e letterali nel modo seguente:
    - Sia gli elementi di B, chiamati costanti, che i letterali x,y,z,… sono espressioni booleane.
    - Se E1 e E2 sono espressioni booleane anche (E1 ⊕ E2) (E1 ⊙ E2) e (~E1) lo sono.
    - Non esistono altre espressioni booleane oltre a quelle che possono essere generate da un numero finito di applicazioni delle due regole precedenti.
- Nota: D'ora in avanti, i simboli degli operatori "⊕", "⊙" e "~" saranno sostituiti con "+", "*" e "!" (o "'") oppure da OR, AND e NOT.

Proprietà

- Associativa
  - Somma: **a+(b+c)=(a+b)+c**
  - Prodotto: **a*(b*c)=(a*b)*c**
- Distributiva
  - Somma: **a*(b+c)=a*b+b*c**
  - Prodotto: **a+(b*c)=(a+b)*(a+c)**
- Idempotenza
  - Somma: **a+a=a**
  - Prodotto: **a*a=a**
- Elemento neutro
  - Somma: **a+1=1** **a+0=a**
  - Prodotto: **a*0=0** **a*1=a**

Proprietà

- Assorbimento
  - Somma: `a+(a*b)=a`
  - Prodotto: `a*(a+b)=a`
- Involuzione
  - Negazione: `(a')'=a`
- Leggi di De Morgan
  - Somma: `(a+b)'=a'*b'`
  - Prodotto: `(a*b)'=a'+b'`
- Consenso
  - Somma: `a*b+a'*c+b*c = a*b + a'*c`
  - Prodotto: `(a+b)*(a'+c)*(b+c)=(a+b)*(a'+c)`

Principio di dualità

- Ogni identità deducibile dai postulati dell'algebra di Boole è trasformata in un'altra identità se:
  - Ogni somma è sostituita da un prodotto, e vice versa
  - Ogni elemento identità 0 è sostituito da un elemento identità 1, e vice versa
- Esempi:

Legge di assorbimento	Elemento neutro	Proprietà distributiva
`a + (a * b) = a`	`a + 1 = 1`	`a * (b + c) = (a*b) + (a*c)`
`a * (a + b) = a`	`a * 0 = 0`	`a + (b * c) = (a+b) * (a+c)`

Dimostrazioni

- Dimostrazioni delle proprietà
  - Metodo esaustivo
  - Metodo algebrico
- Esempio: **a + a'b = a + b**

Metodo algebrico

Operazione	Proprietà
`a*1 + a'b`	Elemento neutro
`a(b+b') + a'b`	Negazione: `x+x'=1`
`ab + ab' + a'b`	Proprietà distributiva
`ab + ab + ab' + a'b`	Idempotenza
`a(b+b') + b(a+a')`	Distributiva
`a*1 + b*1`	Negazione: `x+x'=1`
`a + b`	Elemento neutro

Metodo esaustivo

a	b	a+a'*b	a+b
0	0	0+1*0=0	0+0=0
0	1	0+1*1=1	0+1=1
1	0	1+0*0=1	1+0=1
1	1	1+0*1=1	1+1=1

Rappresentazione di una funzione

- Funzione di commutazione
  - Una funzione di commutazione a n variabili $f(x_0, ...x_{n-1})$ è una funzione che ha dominio in $\{0,1\}^n$ e codominio in $\{0,1\}$: $f: \{0,1\}^n \rightarrow \{0,1\}$
- Una funzione di commutazione $f:\{0, 1\}^n \rightarrow \{0, 1\}$ può essere rappresenta in modo comodo utilizzando una tabella della funzione o tabella della verità
  - Una tabella della verità specifica la relazione tra ogni elemento del dominio $\{0, 1\}^n$ di f e l'immagine nel codominio.

Rappresentazione di una funzione

- 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

Espressioni e funzioni

- Una funzione booleana di *n* variabili può essere espressa
  - Da una espressione booleana di *n* variabili E
- Le proprietà dell'algebra di commutazione
  - Possono essere utilizzate per manipolare una espressione booleana ed ottenerne una equivalente
  - Due espressioni booleane E1 e E2 sono equivalenti se e solo se sono riconducibili alla stessa funzione booleana

Esempio:

f(a,b,c) = (a'b')'a = (a+b)*a = aa + ba = a + ba = a

E1 = (a'b')'a; **E2 = (a+b)*a;** **E3 = aa + ba**
E4 = a + ba; **E5 = a**

E1, E2, E3, E4 ed **E5** sono espressioni booleane equivalenti

Espressioni e funzioni

- Relazione tra funzioni ed espressioni:
  - Ad una espressione E di *n* variabili corrisponde un'unica funzione f di n variabili
  - Viceversa, ad una funzione f di n variabili corrispondono infinite espressioni di n variabili.

Funzione booleana

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

Espressioni booleane

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

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

f(a,b,c)= a

Espressioni e funzioni

- Data una funzione booleana, 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
  - È necessario definire una metrica che indichi la qualità di una soluzione rispetto ad uno o più aspetti (es. Costo, prestazioni, energia…)
    - Esempio: se l'obiettivo è minimizzare il costo del circuito corrispondente a un'espressione, una possibile metrica è il numero di letterali presenti nell'espressione.
      - Si osservi che la metrica è caratterizzata da un valore atteso e da una varianza; quindi, due soluzioni di costo simile (es. $\pm$ 20%) sono da considerarsi equivalenti.

Espressioni e funzioni

- Data una funzione booleana
  - La soluzione iniziale al problema di determinare una sua espressione consiste nel ricorso alle forme canoniche
  - La soluzione iniziale può essere manipolata per raggiungere l'obiettivo desiderato
    - Ad esempio, riduzione del costo
- Le forme canoniche più comuni sono:
  - La forma somma di prodotti, o SoP
  - La forma prodotto di somme, o PoS
- Data una funzione boolena è rappresentata da
  - Una ed una sola forma canonica SoP (disgiuntiva)
  - Una ed una sola forma canonica PoS (congiuntiva)

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 dalla somma (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

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

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

- Infatti
  - Quando **a=0** e **b=1** il prodotto **a'b** assume valore 1 mentre vale 0 in tutti gli altri casi (f_1)
  - Quando **a=1** e **b=1** il prodotto **ab** assume valore 1 mentre vale 0 in tutti gli altri casi (f_2)
- Osservazione: **"a=0** e **b=1"** significa anche letterale (a,0) in AND con la coppia (variabile, valore) (b,1) cioè, termine prodotto **a'b**

Forme canoniche: Definizioni

- Mintermine
  - È un termine prodotto in cui compaiono letterali corrispondenti a tutte le variabili della funzione
    - Ad esempio, una funzione nelle variabili **a,b,c** e **d** potrebbe avere una espressione che contiene i mintermini **a'bc'd** e **ab'cd'.**

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

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

- In generale
  - Mettendo in OR i mintermini della funzione si ottiene una espressione della funzione: Prima Forma Canonica
  - Nel mintermine una variabile compare:
    - Nella forma **x** se nella configurazione di ingresso ha valore 1
    - Nella forma **x'** se nella configurazione di ingresso ha valore 0

Forme canoniche

- Esempio
  - Tabella della verità:

a	b	c	f(a,b,c)	Mintermini
0	0	0	0	
0	0	1	1	a'b'c
0	1	0	1	a'bc'
0	1	1	1	a'bc
1	0	0	1	ab'c'
1	0	1	0	
1	1	0	0	
1	1	1	1	abc

  - Prima forma canonica:

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

Forme canoniche

- Si consideri il nuovamente l'esempio iniziale

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 prodotto (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

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
  - Quando **a=0** e **b=0** il termine **(a'b')'** assume valore 0 mentre vale 1 in tutti gli altri casi (f_1)
  - Quando **a=1** e **b=0** il termine **(ab')'** assume valore 0 mentre vale 1 in tutti gli altri casi (f_2)

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	1
1	1	1

*

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

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

- Applicando le leggi di De Morgan:

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

Forme canoniche

- In generale
  - Mettendo in AND i maxtermini della funzione si ottiene una espressione della funzione: Seconda Forma Canonica
  - Nel maxtermine una variabile compare:
    - Nella forma **x** se nella configurazione di ingresso ha valore 0
    - Nella forma **x'** se nella configurazione di ingresso ha valore 1
- Dove, Maxtermine
  - È un termine somma in cui compaiono letterali corrispondenti a tutte le variabili della funzione
    - Ad esempio, una funzione nelle variabili **a,b,c** e **d** potrebbe avere una espressione che contiene i maxtermini **a'+b+c'+d** e **a+b'+c+d'.**

Forme canoniche

- Esempio
  - Tabella della verità:

a b c	f(a,b,c)	Maxtermini
0 0 0	0	a + b + c
0 0 1	1	
0 1 0	1	
0 1 1	1	
1 0 0	1	
1 0 1	0	a' + b + c'
1 1 0	0	a' + b' + c
1 1 1	1	

  - Seconda forma canonica:

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

Prima forma canonica

- La prima forma canonica ha come espressione generale

  $$f = (x_1' \ldots x_n') * f(0,\ldots,0) +$$
  $$+ (x_1' \ldots x_n) * f(0,\ldots,1) +$$
  $$+ \ldots +$$
  $$+ (x_1 \ldots x_n) * f(1,\ldots,1)$$

- Mintermini della funzione **f**

  $$(x_1' \ldots x_n'),\ (x_1' \ldots x_n),\ \ldots,\ (x_1 \ldots x_n)$$

- Valori che la funzione assume quando la configurazione delle variabili in ingresso è (0,…,0), …, (1,….,1)

  $$f(0,\ldots,0),\ f(0,\ldots,1),\ \ldots.,\ f\ (1,\ldots,1)$$

Prima forma canonica

- La seconda forma canonica ha come espressione generale

$$
\begin{aligned}
f = & ((x_1'+...+x_n') + f(1,...,1)) * \\
& * ((x_1'+...+x_n) + f(1,..,0)) * \\
& * \ ... \ * \\
& * ((x_1+...+x_n) + f(0,...,0))
\end{aligned}
$$

- Maxtermini della funzione **f**

$$(x_1'+...+x_n'),\ (x_1'+...+x_n),\ ...,\ (x_1+...+x_n)$$

- Valori che la funzione assume quando la configurazione delle variabili in ingresso è (0,…,0), …, (1,….,1)

$$f(0,...,0),\ f(0,...,1),\ ....,\ f\ (1,...,1)$$

Teorema di espansione di Shannon

- La descrizione formale introdotta in precedenza deriva direttamente dall'applicazione iterativa del Teorema di espansione di Shannon
- Sia **f**: $B^n \rightarrow B$ è una funzione booleana
  - Per ogni **(x1, x2, ..., xn)** in B^n, si ha:

$$\begin{aligned} f(x_1, x_2, ..., x_n) &= x_1' * f_{x1'} + x_1 * f_{x1} = \\ &= x_2' * f_{x2'} + x_2 * f_{x2} = \\ &= ... \end{aligned}$$

  - E, dualmente:

$$\begin{aligned} f(x_1, x_2, ..., x_n) &= (x_1' + f_{x1}) * (x_1 + f_{x1'}) = \\ &= (x_2' + f_{x2}) * (x_2 + f_{x2'}) = \\ &= ... \end{aligned}$$

Teorema di espansione di Shannon

- Esempio, SoP:

```
f(a, b, c) = a’ * f(0,b,c) + a * f(1,b,c) =
           = b’ * f(a,0,c) + b * f(a,1,c) =
           = c’ * f(a,b,0) + c * f(a,b,1)
```

- Esempio, PoS:

```
f(a, b, c) = (a’ + f(1,b,c)) * (a + f(0,b,c)) =
           = (b’ + f(a,1,c)) * (b + f(a,0,c)) =
           = (c’ + f(a,b,1)) * (c + f(a,b,0))
```

Teorema di espansione di Shannon

- 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

Teorema di espansione di Shannon

- Esempio di espansione completa

```
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	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Prima forma canonica

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

Seconda forma canonica

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

Teorema di espansione di Shannon

- Può essere utilizzato anche su espressioni Booleane
  - Espandendo rispetto ad **a** l'espressione booleana:

    f(a,b,c) = ab + b′ + a′bc′

  - Si ottiene la forma equivalente:

    f = ab + (ab′+a′b′) + a′bc′

    f = a′(b′+bc′) + a(b+b′) = a′b′ + a′bc′ + a

  - Espandendo rispetto ad **a**, **b** e **c** l'espressione booleana:

    f(a,b,c) = ab + b′ + a′bc′

  - Si ottiene la forma equivalente:

    f = a′(b′+bc′) + a(b+b′) =

    = a′(b′1+bc′) + a(b′1+b1) =

    = 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

Manipolazione delle espressioni

- Data un'espressione di una funzione booleana
  - L'algebra di commutazione permette di manipolarla per 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
    - In questo caso vale 9
  - Obiettivo: ridurre il costo

Manipolazione delle espressioni

- Dall'espressione di partenza:

  $f(x,y,z) = x'yz' + xyz' + xyz$

- Applicando la proprietà distributiva e quella della complementazione si ottiene:

  $f(x,y,z) = (x'+x)yz' + xyz =$

  $= 1*yz' + xyz = yz' + xyz$

- Applicando di nuovo la proprietà distributiva si ottiene:

  $f = y(z' + xz)$

- E ricordando che $a + a'b = a + b$, si ottiene infine:

  $f = y(z' + x) = yz'+xy$

- La nuova espressione ha un costo di 4 letterali

Manipolazione delle espressioni

- Allo stesso risultato si sarebbe giunti anche:

  `f(x,y,z) = x’yz’ + xyz’ + xyz`

- Applicando la proprietà dell’idempotenza si ottiene:

  `f(x,y,z) = x’yz’ + xyz’ + xyz’ + xyz`

- Applicando la proprietà distributiva

  `f = yz’(x’+x) + xy(z’+z)`

- Da cui infine

  `f = yz’1 + xy1 = yz’ + xy`

Manipolazione delle espressioni

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