Algebra Booleana

Alessandro Pellegrini
a.pellegrini@ing.uniroma2.it

Università di Roma
Tor Vergata

Algebra Booleana

- È un tipo di algebra definita dal matematico George Boole (1815-1864)
  - Fu inizialmente proposta nel tentativo di verificare la verità o la falsità di affermazioni in linguaggio naturale, partendo da alcune "verità" di base
- Nel 1936 Claude Shannon propose di utilizzare l'algebra di Boole per studiare e progettare circuiti basati su relé
  - L'algebra booleana si basava sui concetti di vero/falso
  - I relé avevano due stati di funzionamento: aperto/chiuso
- Gli elementi fondamentali dei circuiti elettronici, al giusto livello di astrazione, rispettano ancora le regole dell'algebra booleana

Variabili e funzioni di commutazione

- Una *variabile booleana* (o di commutazione) è una quantità algebrica x definita su un insieme $S = \{0, 1\}$, ossia che può assumere solo due valori

- Una *funzione di commutazione* di una variabile booleana è definita come la proiezione di $\{0,1\}$ su $\{0,1\}$:
  - $f : \{0,1\} \mapsto \{0,1\}$
  - $y = f(x)$
- Una *funzione di commutazione* di n variabili booleane è una funzione il cui dominio è dato da tutte le n-uple $(x_1, x_2, \cdots, x_n)$ in $\{0,1\}^n$ ed il codominio è $\{0,1\}$:
  - $f : \{0,1\}^n \mapsto \{0,1\}$
  - $y = f(x_1, x_2, \cdots, x_n)$

Operatori e assiomi fondamentali dell'algebra booleana

- *Somma logica*: indicata con il segno $+$
- *Prodotto logico*: indicato con il segno $\cdot$
- *Negazione*: dato un valore x, il suo valore negato è $\bar{x}$.

- Essi permettono di definire gli assiomi fondamentali sul dominio S:
  - $\forall a, b \in S;\ a + b \in S;\ a \cdot b \in S$ (chiusura)
  - $\exists 0 \in S | \forall a \in S, a + 0 = a; \exists 1 \in S | \forall a \in S, a \cdot 1 = a$ (elemento identità)
  - $a + b = b + a$ (proprietà commutativa)
  - $(a + b) + c = a + (b + c);\ (a \cdot b) \cdot c = a \cdot (b \cdot c)$ (proprietà associativa)
  - $a \cdot (b + c) = a \cdot b + a \cdot c;\ a + b \cdot c = (a + b) \cdot (a + c)$ (proprietà distributiva)
  - $\forall a \in \mathrm{S}\ \exists \overline{a} \in \mathrm{S} | a + \overline{a} = 1;\ a \cdot \overline{a} = 0$ (elemento inverso)
  - $|S| = 2^n;\ n = 1,2,3, \cdots$ (cardinalità)

Legge di dualità

- Ogni identità booleana rimane invariata scambiando $+$ con $\cdot$ e 0 con 1.

- Questo teorema è vero perché non siamo obbligati ad usare 0 e 1 come simboli per l'algebra booleana binaria.
  - Es: se $1 + 0 = 1$, ponendo $1 \rightarrow \alpha$ e $0 \rightarrow \beta$ otteniamo $\alpha + \beta = \alpha$, che è ancora corretto secondo gli assiomi dell'algebra booleana.
- Se effettuiamo la sostituzione $0 \rightarrow 1$ e $1 \rightarrow 0$, otteniamo:
  - $0 + 1 = 0$, che è ancora una possibile (diversa!) algebra booleana
- A questo punto, se effettuiamo anche la sostituzione $+ \rightarrow \cdot$ e $\cdot \rightarrow +$ otteniamo:
  - $0 \cdot 1 = 0$, che è *la stessa* algebra booleana di partenza

Proprietà di idempotenza

- Nell'algebra booleana binaria vale $a + a = a$ e $a \cdot a = a$

- Infatti:
    - $a = a + 0$
    - $a = a + a \cdot \overline{a}$
    - $a = (a + a) \cdot (a + \overline{a})$
    - $a = a + a$
    - Per la legge di dualità, vale anche: $a = a \cdot a$

Annichilatori funzionali

- Nell'algebra booleana binaria vale $a + 1 = 1$ e $a \cdot 0 = 0$

- Infatti:
  - $a + 1 = a + a + \overline{a}$
  - $a + 1 = a + \overline{a}$
  - $a + 1 = 1$
  - Per la legge di dualità, vale anche: $a \cdot 0 = 0$

Legge dell'assorbimento

- Nell'algebra booleana binaria vale $a + a \cdot b = a$ e $a \cdot (a + b) = a$

- Infatti:
  - $a + a \cdot b = a \cdot 1 + a \cdot b$
  - $a + a \cdot b = a \cdot (1 + b)$
  - $a + a \cdot b = a \cdot 1$
  - $a + a \cdot b = a$
  - Per la legge di dualità, vale anche: $a \cdot (a + b) = a$

Teorema di De Morgan

- Il teorema di De Morgan è un teorema importante perché permette di esprimere gli operatori + e · in funzione degli altri due operatori fondamentali

$$\overline{a+b} = \overline{a} \cdot \overline{b} \qquad \overline{a \cdot b} = \overline{a} + \overline{b}$$

- Questo è vero per un numero qualsiasi di variabili
- Per dimostrare il teorema è sufficiente verificare che $\overline{a} \cdot \overline{b}$ è il complemento di $a + b$:
  - $(a+b) + (\overline{a} \cdot \overline{b}) = (a+b+\overline{a}) \cdot (a+b+\overline{b})$
  - $(a+b) + (\overline{a} \cdot \overline{b}) = (1+b) \cdot (1+a)$
  - $(a+b) + (\overline{a} \cdot \overline{b}) = 1 \cdot 1 = 1$
  - La seconda relazione si ottiene per la legge di dualità

Funzioni di commutazione: rappresentazione

- Gli operatori finora introdotti possono essere utilizzati per definire le *funzioni di commutazione:*
  - Una funzione di commutazione di n variabili $y = f(x_1, x_2, \cdots, x_n)$ è una funzione il cui dominio consiste di tutte e sole le n-uple $(x_1, x_2, \cdots, x_n)$ e il codominio è l'insieme $\{0,1\}$
  - $f: \{0,1\}^n \mapsto \{0,1\}$

- Ci sono vari modi per esprimere una funzione di commutazione:
  - forma tabellare (tabelle di verità)
  - forme canoniche
  - forme decimali

Tabelle di verità

- Una tabella di verità crea una relazione tra le variabili di input ed il valore di output della funzione.
- Nelle tabelle, spesso si utilizza in maniera intercambiabile il *vettore* $\boldsymbol{x} = \langle x_1, x_2, \cdots, x_n \rangle$ e le variabili $x_1, x_2, \cdots, x_n$
- Una tabella di verità di una funzione di n variabili è costituita da 2^n righe

$\boldsymbol{x}$	x_1	x_2	x_3	y
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

Tabelle di verità degli operatori fondamentali

- Anche gli operatori fondamentali possono essere visti come funzioni di commutazione su una (negazione) o due (somma e prodotto) variabili di commutazione
- È quindi possibile esprimerli come tabelle di verità

Negazione

x_1	y
0	1
1	0

Somma logica

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

Prodotto logico

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

Operatori derivati

- Sulla base dei tre operatori fondamentali è possibile definire i seguenti operatori derivati:

- OR esclusivo (XOR), indicato con il simbolo $\oplus$;
- Not AND *(NAND), indicato con il simbolo* $|$;
- NOR, indicato con il simbolo $\downarrow$;
- NOR esclusivo (XNOR), indicato con il simbolo $\odot$.

OR esclusivo (XOR)

- Equivalente alla *somma modulo due*
- Definito come segue:

$$a \oplus b = \overline{a}b + a\overline{b}$$

- Utilizzato per verificare la disuguaglianza tra due variabili
- Proprietà principali:
  - $a \oplus b = b \oplus a$ (proprietà commutativa);
  - $a \oplus (b \oplus c) = (a \oplus b) \oplus c$ (proprietà associativa);
  - $a \oplus a = 0$;
  - $a \oplus \overline{a} = 1$;
  - $a \oplus 1 = \overline{a}$;
  - $\overline{a} \oplus b = a \oplus \overline{b} = \overline{a \oplus b}$

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Not AND (NAND)

- Definito come segue:

$$a|b = \overline{a \cdot b} = \overline{a} + \overline{b}$$

- Proprietà principali:
  - $a|b = b|a$ (proprietà commutativa);
  - $a|1 = \overline{a}$;
  - $a|0 = 1$;
  - $a|\overline{a} = 1$;
  - $a|(b|c) \neq (a|b)|c$ (l'operatore <u>non</u> è associativo)

x_1	x_2	y
0	0	1
0	1	1
1	0	1
1	1	0

NOR

- È il duale dell'operatore NAND
- Definito come segue:

$$a \downarrow b = \overline{a + b} = \overline{a} \cdot \overline{b}$$

- Proprietà principali:
  - $a \downarrow b = b \downarrow a$ (proprietà commutativa);
  - $a \downarrow 1 = 0$;
  - $a \downarrow 0 = \overline{a}$;
  - $a \downarrow \overline{a} = 0$;
  - $a \downarrow (b \downarrow c) \neq (a \downarrow b) \downarrow c$ (l'operatore non è associativo)

x_1	x_2	y
0	0	1
0	1	0
1	0	0
1	1	0

NOR esclusivo (XNOR)

- Definito come segue:

$$a \odot b = (\overline{a} + b) \cdot (a + \overline{b})$$

- Utilizzato per verificare l'uguaglianza tra due variabili
- Proprietà principali:
  - $a \odot b = b \odot a$ (proprietà commutativa);
  - $a \odot (b \odot c) = (a \odot b) \odot c$ (proprietà associativa);
  - $a \odot 1 = a$;
  - $a \odot \overline{a} = 0$;
  - $a \odot 0 = \overline{a}$;

x_1	x_2	y
0	0	1
0	1	0
1	0	0
1	1	1

Compito per casa

- Provate a dimostrare alcune delle proprietà che abbiamo enunciato:
  - $\overline{a} \oplus b = a \oplus \overline{b} = \overline{a \oplus b}$
  - $a|(b|c) \neq (a|b)|c$
  - $a \downarrow (b \downarrow c) \neq (a \downarrow b) \downarrow c$

- Suggerimenti:
  - Utilizzate le definizioni degli operatori
  - Utilizzate il teorema di De Morgan

- Altro suggerimento:
  - Provateci davvero!

Teorema di Shannon (1949)

- Una qualsiasi funzione $y = f(x_1, x_2, \cdots, x_n)$ può essere rappresentata in una delle due seguenti forme duali:

$$f(x_1, x_2, \cdots, x_n) = x_1 \cdot f(1, x_2, \cdots, x_n) + \overline{x_1} \cdot f(0, x_2, \cdots, x_n)$$
$$f(x_1, x_2, \cdots, x_n) = (x_1 + f(0, x_2, \cdots, x_n)) \cdot (\overline{x_1} + f(1, x_2, \cdots, x_n))$$

- I termini che moltiplicano/si sommano a x_1 sono chiamati i *residui* della funzione
- Ciò è vero nel caso di *variabili indipendenti*
- Tale teorema può essere applicato iterativamente a tutte le variabili della funzione

Teorema di Shannon (1949)

- Applicando iterativamente il teorema ai residui si ottiene:

$$f(x_1, \cdots, x_n) = \overline{x_1} \cdot \overline{x_2} \cdot \cdots \cdot \overline{x_n} f(0,0,\cdots,0) + x_1 \cdot \overline{x_2} \cdot \cdots \cdot \overline{x_n} f(1,0,\cdots,0) +$$
$$+ \overline{x_1} \cdot x_2 \cdot \cdots \cdot \overline{x_n} f(0,1,\cdots,0) + \cdots +$$
$$+ x_1 \cdot x_2 \cdot \cdots \cdot \overline{x_n} f(1,1,\cdots,0) + x_1 \cdot x_2 \cdot \cdots \cdot x_n f(1,1,\cdots,1)$$

- Generalizzando, quindi, possiamo esprimere ciascun termine come:

$$x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} f(\alpha_1, \alpha_2, \cdots, \alpha_n)$$

- con $\alpha_i = \{0,1\}$ e $x_i^{\alpha_i} = x_i$ se $\alpha_i =1$, $x_i^{\alpha_i} = \overline{x_i}$ se $\alpha_i = 0$

Forma canonica in somma di prodotti

- Prima forma canonica: *somma di prodotti* (o forma canonica disgiuntiva):

$$f(x_1, x_2, \cdots, x_n) = \sum_{k=0}^{2^n-1} \mathbf{m}_k f(\mathbf{k})$$

  - Il termine $\mathbf{m}_k$ viene chiamato *mintermine* ed è nella forma $x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$

- In ciascun mintermine, le variabili compaiono una ed una sola volta in forma diretta o negata

Forma canonica in prodotti di somme

- Utilizzando il teorema di De Morgan, possiamo trasformare la forma canonica vista precedentemente come segue

$$\overline{f(x_1, x_2, \cdots, x_n)} = \sum_{k=0}^{2^n-1} \mathbf{m}_k \overline{f(\mathbf{k})}$$

$$f(x_1, x_2, \cdots, x_n) = \overline{\sum_{k=0}^{2^n-1} \mathbf{m}_k \overline{f(\mathbf{k})}} = \prod_{k=0}^{2^n-1} \overline{\mathbf{m}_k \overline{f(\mathbf{k})}} = \prod_{k=0}^{2^n-1} (\mathbf{M}_k + f(\mathbf{k}))$$

- Dove Il termine $\mathbf{M}_k$ viene chiamato *maxtermine* ed è nella forma $\mathbf{M}_k = \sum_{i=0}^{n-1} x_i^{\alpha_i}$, con $\alpha_i = \{0,1\}$ e $x_i^{\alpha_i} = x_i$ se $\alpha_i = 0$, $x_i^{\alpha_i} = \overline{x_i}$ se $\alpha_i = 1$

Un esempio: identificazione di mintermini e maxtermini

- Consideriamo la seguente funzione di commutazione definita mediante tabella di verità

$\mathbf{k}$	x_1	x_2	x_3	$f(\mathbf{k})$
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

- I mintermini corrispondono alle configurazioni $\mathbf{k} = 0,4,5,7$
- I maxtermini corrispondono alle configurazioni $\mathbf{k} = 1,2,3,6$

Un esempio: forme canoniche

- Dai mintermini, possiamo definire la seguente rappresentazione in somme di prodotti:

$$f(x_1, x_2, x_3) = \overline{x_1}\,\overline{x_2}\,\overline{x_3} + x_1\,\overline{x_2}\,\overline{x_3} + x_1\overline{x_2}x_3 + x_1x_2x_3$$

k	x_1	x_2	x_3	$f(\mathbf{k})$
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Un esempio: forme canoniche

- Dai maxtermini, possiamo definire la seguente rappresentazione in prodotti di somme:

$$f(x_1, x_2, x_3) = (x_1 + x_2 + \overline{x_3})(x_1 + \overline{x_2} + x_3)(x_1 + \overline{x_2} + \overline{x_3})(\overline{x_1} + \overline{x_2} + x_3)$$

k	x_1	x_2	x_3	$f(\mathbf{k})$
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Forme canoniche

- Le forme canoniche sono rappresentazioni uniformi utilizzate per descrivere una funzione di commutazione.
- Qualsiasi funzione di commutazione può essere trasformata in forma canonica
  - Si può costruire la tabella di verità e utilizzare la tecnica dei mintermini o dei maxtermini
  - Si possono effettuare trasformazioni analitiche: se una variabile x_i non è presente, si può moltiplicare per $(x_i + \overline{x_i})$
- Ad esempio, la funzione $x_1x_3 + \overline{x_1}(x_2 + \overline{x_3})$ può essere trasformata come segue:

$$x_1x_3 + \overline{x_1}(x_2 + \overline{x_3}) =$$
$$= x_1x_3(x_2 + \overline{x_2}) + \overline{x_1}x_2(x_3 + \overline{x_3}) + \overline{x_1}\,\overline{x_3}(x_2 + \overline{x_2}) =$$
$$= x_1x_2x_3 + x_1\overline{x_2}x_3 + \overline{x_1}x_2x_3 + \overline{x_1}x_2\overline{x_3} + \overline{x_1}x_2\overline{x_3} + \overline{x_1x_2x_3}$$

Forma decimale

- La forma tabellare o le forme canoniche possono essere molto lunghe
- Nella *forma decimale*, in cui si indica l'interpretazione decimale delle variabili booleane associate a mintermini/maxtermini.

$$f(\mathbf{k}) = \sum(0,4,5,7) = \prod(1,2,3,6)$$

$\mathbf{k}$	x_1	x_2	x_3	$f(\mathbf{k})$
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Forme semplificate

- Le forme canoniche non sono necessariamente le forme *minime* per rappresentare una funzione booleana
- Identificare una forma minima è importante poiché permetterà di realizzare circuiti più compatti
- Il processo di semplificazione può essere svolto:
  - mediante metodi analitici
  - mediante metodi algoritmici
- I metodi analitici richiedono di applicare le proprietà dell'algebra booleana e i teoremi visti fino ad ora per semplificare l'equazione
- Non c'è una via "certa" da seguire, ci si affida all'intuito e all'esperienza

Semplificazione analitica: un esempio

- Consideriamo la funzione:

$$f(x, y, z, w) = xyzw + xyz\overline{w} + xy\overline{z}\overline{w} + xy\overline{z}w + \overline{x}yzw$$

- Per la proprietà dell'idempotenza, un termine può essere ripetuto più di una volta senza modificare il risultato
- Primo e secondo mintermine: $xyzw + xyz\overline{w} = xyz$
- Primo e quarto mintermine: $xyzw + xy\overline{z}w = xyw$
- Primo e quinto mintermine: $xyzw + \overline{x}yzw = yzw$
- Secondo e terzo mintermine: $xyz\overline{w} + xy\overline{z}\overline{w} = xy\overline{w}$
- Otteniamo:

$$f(x, y, z, w) = xyz + xyw + yzw + xy\overline{w}$$

Semplificazione analitica: un esempio

- Proseguendo:
$$f(x, y, z, w) = xyz + xyw + yzw + xy\overline{w}$$

- Secondo e ultimo termine: $xyw + xy\overline{w} = xy$
$$f(x, y, z, w) = xy + xyz + yzw$$

- Per la legge dell'assorbimento ($a + a \cdot b = a$):
$$f(x, y, z, w) = xy + yzw$$

- In sintesi, abbiamo trovato la seguente uguaglianza:
$f(x, y, z, w) = xyzw + xyz\overline{w} + xy\overline{zw} + xy\overline{z}w + \overline{x}yzw = xy + yzw$

Mappe di Karnaugh

- Le mappe di Karnaugh sono una rappresentazione differente delle tabelle di verità
- Le variabili vengono organizzate in tabelle quadrate o rettangolari, a seconda del loro numero
- I valori che possono assumere le variabili vengono ordinati secondo un Codice di Gray (distanza di Hamming = 1)
- In questo modo, spostandosi da una cella all'altra, si causa il cambiamento del valore *di una sola variabile*
- Poiché $(a + \overline{a}) = 1$, è possibile eliminare (semplificare) una variabile se la funzione assume lo stesso valore in gruppi di celle adiacenti
- È un metodo facile per gli umani (fino a 6 variabili), di difficile realizzazione automatizzata mediante un algoritmo

Mappe di Karnaugh per funzioni di 2, 3, 4 variabili

$x_1 \backslash x_0$	0	1
0	$f(00)$	$f(10)$
1	$f(01)$	$f(10)$

2 variabili

$x_2 \backslash x_0x_1$	00	01	11	10
0	$f(000)$	$f(010)$	$f(110)$	$f(100)$
1	$f(001)$	$f(011)$	$f(111)$	$f(101)$

3 variabili

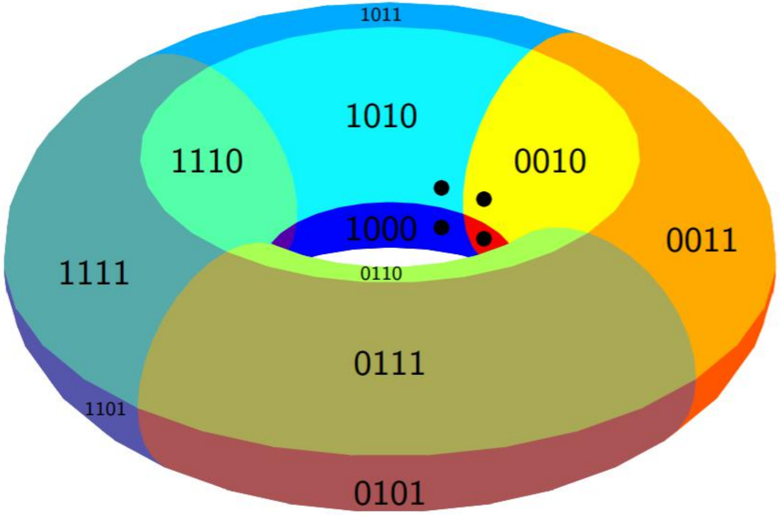
$x_2x_3 \backslash x_0x_1$	00	01	11	10
00	$f(0000)$	$f(0100)$	$f(1100)$	$f(1000)$
01	$f(0001)$	$f(0101)$	$f(1101)$	$f(1001)$
11	$f(0011)$	$f(0111)$	$f(1111)$	$f(1011)$
10	$f(0010)$	$f(0110)$	$f(1110)$	$f(1010)$

4 variabili

- È importante notare che l'uso del codice di Gray garantisce che anche le celle agli estremi siano *adiacenti*

Mappa di Karnaugh per funzioni di 4 variabili

- Le adiacenze ai bordi e agli angoli sono valide poiché la mappa è, in realtà, lo sviluppo della superficie di un solido multidimensionale
- Ad esempio, per 4 variabili, la mappa è lo sviluppo di un *toro*



0000	0100	1100	1000
0001	0101	1101	1001
0011	0111	1111	1011
0010	0110	1110	1010

Mappa di Karnaugh per 5 variabili

$x_2x_3 \backslash x_0x_1$	00	01	11	10
00	$f(00000)$	$f(01000)$	$f(11000)$	$f(10000)$
01	$f(00010)$	$f(01010)$	$f(11010)$	$f(10010)$
11	$f(00110)$	$f(01110)$	$f(11110)$	$f(10110)$
10	$f(00100)$	$f(01100)$	$f(11100)$	$f(10100)$

$x_4 = 0$

	00	01	11	10
00	$f(00001)$	$f(01001)$	$f(11001)$	$f(10001)$
01	$f(00011)$	$f(01011)$	$f(11011)$	$f(10011)$
11	$f(00111)$	$f(01111)$	$f(11111)$	$f(10111)$
10	$f(00101)$	$f(01101)$	$f(11101)$	$f(10101)$

$x_4 = 1$

- L'adiacenza è anche tra le due tabelle, come se queste *fossero sovrapposte*

Mappa di Karnaugh per 6 variabili

- Come nel caso di cinque variabili, anche qui le adiacenze valgono per tabelle sovrapposte

x_2x_3 \ x_0x_1	00	01	11	10
00	$f(000000)$	$f(010000)$	$f(110000)$	$f(100000)$
01	$f(000100)$	$f(010100)$	$f(110100)$	$f(100100)$
11	$f(001100)$	$f(011100)$	$f(111100)$	$f(101100)$
10	$f(001000)$	$f(011000)$	$f(111000)$	$f(101000)$

$x_4x_5= 00$

	00	01	11	10
	$f(000001)$	$f(010001)$	$f(110001)$	$f(100001)$
	$f(000101)$	$f(010101)$	$f(110101)$	$f(100101)$
	$f(001101)$	$f(011101)$	$f(111101)$	$f(101101)$
	$f(001001)$	$f(011001)$	$f(111001)$	$f(101001)$

$x_4x_5= 01$

00	$f(000010)$	$f(010010)$	$f(110010)$	$f(100010)$
01	$f(000110)$	$f(010110)$	$f(110110)$	$f(100110)$
11	$f(001110)$	$f(011110)$	$f(111110)$	$f(101110)$
10	$f(001010)$	$f(011010)$	$f(111010)$	$f(101010)$

$x_4x_5= 10$

$f(000011)$	$f(010011)$	$f(110011)$	$f(100011)$
$f(000111)$	$f(010111)$	$f(110111)$	$f(100111)$
$f(001111)$	$f(011111)$	$f(111111)$	$f(101111)$
$f(001011)$	$f(011011)$	$f(111011)$	$f(101011)$

$x_4x_5= 11$

Mappe di Karnaugh e tabelle di verità

- Per trasformare una tabella di verità in mappa di Karnaugh è sufficiente riempire le celle della mappa in funzione del valore della funzione per le configurazioni delle variabili di ingresso

k	x_0	x_1	x_2	$f(\text{k})$
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

$x_2 \backslash x_0x_1$	00	01	11	10
0	1	0	0	1
1	0	0	1	1

Semplificazioni mediante mappe di Karnaugh

- Per sfruttare le adiacenze, è possibile costruire *insiemi di copertura* di dimensione 1, 2, 4, 8, 16, … celle, raddoppiando via via la dimensione dell’insieme
- Gli insiemi devono coprire *tutti* i termini 1
- In questo modo, si identificano gli *implicanti primi*, ossia gli insiemi di termini che determinano la funzione equivalente minima
- È possibile lavorare anche con i maxtermini, in tal caso si parla di *implicati minimi* e le coperture avvengono sugli 0.
- Non è detto che, data una funzione, esista un solo insieme di implicanti minimi

Esempio di semplificazione

- Consideriamo la funzione $f(x, y, z, w) = \sum(0,1,3,7,15)$
- Rappresentiamo la tabella di verità su una mappa di Karnaugh:

zw \ xy	00	01	11	10
00	1	0	0	0
01	1	0	0	0
11	1	1	1	0
10	0	0	0	0

Esempio di semplificazione

- A questo punto, identifichiamo gli implicanti primi selezionando degli insiemi di dimensione 1, 2, 4, … fino a coprire tutti gli 1 almeno una volta (semplificazione con mintermini)

zw \ xy	00	01	11	10
00	1	0	0	0
01	1	0	0	0
11	1	1	1	0
10	0	0	0	0

zw \ xy	00	01	11	10
00	1	0	0	0
01	1	0	0	0
11	1	1	1	0
10	0	0	0	0

Esempio di semplificazione

- Le variabili che cambiano valore nelle celle adiacenti in ciascun insieme possono essere semplificate:
  - $f(x, y, z, w) = \overline{x}\,\overline{y}\,\overline{z} + \overline{x}zw + yzw$ *(insiemi di sinistra)*
  - $f(x, y, z, w) = \overline{x}\,\overline{y}\,\overline{z} + \overline{x}\,\overline{y}w + yzw$ *(insiemi di destra)*

zw \ xy	00	01	11	10
00	1	0	0	0
01	1	0	0	0
11	1	1	1	0
10	0	0	0	0

zw \ xy	00	01	11	10
00	1	0	0	0
01	1	0	0	0
11	1	1	1	0
10	0	0	0	0

Semplificazione mediante prodotto di somme

- È possibile effettuare la semplificazione creando insiemi che ricoprano gli zeri
  - Le funzioni minime ottenute sono equivalenti
- In questo caso, è necessario esprimere la funzione come prodotto di somme

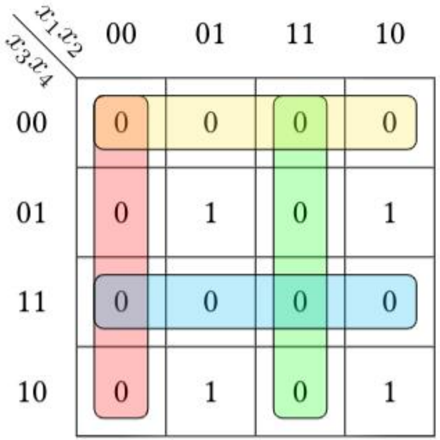
- Quale usare?
  - Regola pratica (ma non generale!): se gli 1 sono meno della metà, si usano i maxtermini. Se gli zeri sono meno della metà, si usano i mintermini
  - In generale, è opportuno identificare la strategia che porta al numero minore di termini o di termini con meno variabili

Semplificazione mediante prodotto di somme: esempio

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	0	0	0	0
01	0	1	0	1
11	0	0	0	0
10	0	1	0	1

- In questo caso la forma minima è data da:
  $f(x_1, x_2, x_3, x_4) = \overline{x_1}x_2\overline{x_3}x_4 + x_1\overline{x_2}\,\overline{x_3}x_4 + \overline{x_1}x_2x_3\overline{x_4} + x_1\overline{x_2}x_3\overline{x_4}$
- È la forma canonica in somma di prodotti: la mappa di Karnaugh non ci permette di semplificare i mintermini

Semplificazione mediante prodotto di somme: esempio



- In questo caso la forma minima è data da:
$$f(x_1, x_2, x_3, x_4) = (x_1 + x_2)(\overline{x_1} + \overline{x_2})(x_3 + x_4)(\overline{x_3} + \overline{x_4})$$
- Abbiamo sempre quattro termini, ma ciascuno di due sole variabili!

Esempi di adiacenze: funzioni di 4 variabili

Si semplificano x_1 e x_3

$x_2x_3 \backslash x_0x_1$	00	01	11	10
00	1	1	1	0
01	1	1	1	0
11	1	0	1	1
10	1	0	1	1

Si semplificano x_0 e x_3

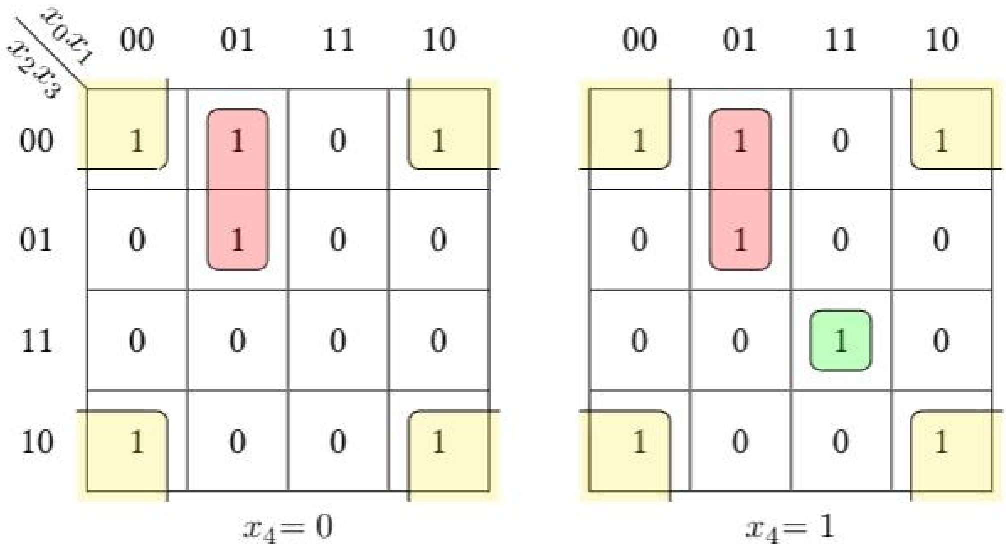
Si semplificano x_2 e x_3

$x_2x_3 \backslash x_0x_1$	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

Si semplificano x_0 e x_2

Esempi di adiacenze: funzioni di 5 variabili

- Gli insiemi rosso e giallo sono due soli insiemi (tabelle sovrapposto)
- L’insieme verde è il caso di un mintermine che è anche implicante primo



Don't Care Conditions

- A volte, una funzione è *parzialmente specificata*
- In questi casi, il valore dell'uscita non è definito per tutte le configurazioni delle variabili di ingresso
  - Variabili dipendenti
  - Configurazioni non di interesse
- In questi casi, i mintermini/maxtermini vengono associati (nella notazione decimale) a un insieme $\sum_{0/1}$ che rappresenta il fatto che non è noto (o di interesse) che il valore della funzione sia 0 o 1
- Nel caso delle mappe di Karnaugh, si indica tale condizione con un trattino (—) e si può far valere la funzione 1 o 0 a seconda di come è più comodo per la minimizzazione

Don’t Care Conditions

x_2x_3 \ x_0x_1	00	01	11	10
00	1	1	1	1
01	-	-	-	-
11	0	0	0	0
10	0	0	0	0

x_2x_3 \ x_0x_1	00	01	11	10
00	1	1	1	1
01	-	-	-	-
11	0	0	0	0
10	0	0	0	0

Esempio con funzione di 6 variabili

- Consideriamo la seguente funzione:

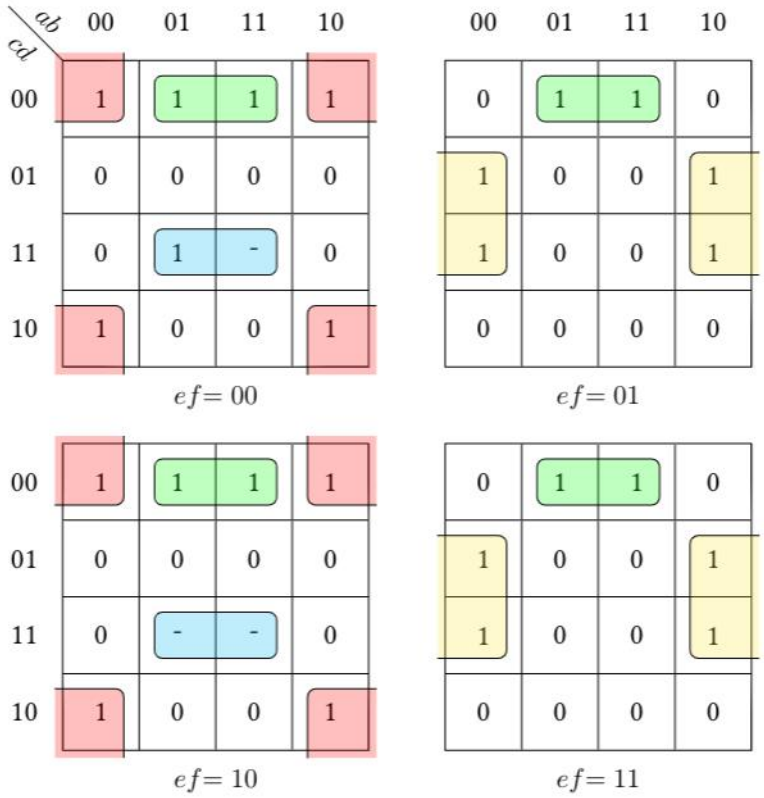
$$f(a,b,c,d,e,f) = \sum(0,1,2,3,4,5,14,16,17,18,19,20,21,29,34,35,40,41,44,50,51,56,57,60,61) + \sum_{0/1}(15,30,31)$$

- Tale funzione ha 3 don't care conditions, che possono essere associate agli 0 o agli 1 come si preferisce

Esempio con funzione di 6 variabili

- L'insieme azzurro costruisce un insieme di copertura che racchiude 4 termini, permettendo una riduzione di due variabili

- Se avessimo considerato il solo mintermine $\overline{a}bcd\overline{e}\overline{f}$ l'espressione sarebbe stata più complessa



Operatori universali

- Sono davvero necessari i tre operatori AND, OR, NOT per definire l'algebra booleana?
  - No: uno dei due può essere eliminato
- In ogni caso, si può effettuare una doppia negazione e sfruttare il teorema di De Morgan
- Esempio: $f(x, y, z) = x + yz + \bar{x}\,\bar{z} = \bar{\bar{f}}(x, y, z) = \overline{\bar{x} \cdot \overline{yz} \cdot \overline{\bar{x}\,\bar{z}}}$
- Applicando nuovamente una doppia negazione e sfruttando il teorema di De Morgan, si sostituisce l'AND con l'OR

- Quindi, è possibile esprimere tutta l'algebra booleana usando solo due operatori.
  - Possiamo ridurre ancora?

Operatori universali

- L’operatore NAND permette, da solo, di esprimere tutta l’algebra booleana (*operatore universale*).
- Infatti:
  - $a|a = \overline{a}$
  - $(a|a)|(b|b) = \overline{\overline{a} \cdot \overline{b}} = a + b$
  - $(a|b)|(a|b) = \overline{\overline{(a \cdot b)} \cdot \overline{(a \cdot b)}} = (a \cdot b) + (a \cdot b) = a \cdot b$
  - $a|(a|a) = \overline{a \cdot \overline{a}} = a + \overline{a} = 1$
  - $(a|(a|a))|(a|(a|a)) = 1|1 = 0$
- È quindi possibile esprimere tutte le costanti e gli operatori fondamentali dell’algebra booleana sfruttando solo ed esclusivamente l’operatore NAND

Operatori universali

- L'operatore NOR è stato definito come duale dell'operatore NAND, quindi anch'esso deve essere universale. Infatti:
  - $a \downarrow a = \overline{a}$
  - $(a \downarrow a) \downarrow (b \downarrow b) = \overline{\overline{a} + \overline{b}} = a \cdot b$
  - $(a \downarrow b) \downarrow (a \downarrow b) = \overline{\overline{(a + b)}} = a + b$
  - $a \downarrow (a \downarrow a) = 0$
  - $(a \downarrow (a \downarrow a)) \downarrow (a \downarrow (a \downarrow a)) = 1$

- Tali operatori universali possono portare benefici nell'implementazione dei circuiti perché, come vedremo, la loro implementazione in hardware può richiedere un numero minore di componenti elettroniche