
Mappe di Karnaugh

Maurizio Palesi

Obiettivi

- Trovare una espressione in forma SP o PS minima rispetto a certi criteri di costo
- Nella ottimizzazione delle espressioni SP (PS) a due livelli l'obiettivo è
 - Ridurre il numero di mintermini (maxtermini)
 - Ridurre il numero di letterali
 - $f(a,b,c)=\underline{abc}+\underline{abc}+\underline{abc}$ equivale a $f(a,b,c)=\underline{ab}+\underline{ac}$
- Metodologie di minimizzazione
 - Karnaugh
 - Quine-Mc Cluskey

Generalità

- Si propone di identificare forme minime a due livelli applicando
 - Per *SP* la riduzione $aZ + \underline{a}Z = (a + \underline{a})Z = Z$ con Z termine prodotto (implicante) di $n-1$ variabili
 - Per *PS* la riduzione $(\underline{a} + Z)(a + Z) = Z$ con Z termine somma (implicato) di $n-1$ variabili
 - Esempio: $ab\underline{c} + abc = ab$

Esempio di Riduzione

- La riduzione può essere applicata iterativamente
$$\begin{aligned} \underline{a}bcd + a\underline{b}cd + abc\underline{d} + abcd &= \\ &= abc(\underline{d}+d) + abc(\underline{d}+d) = \\ &= abc\underline{c} + abc = ab(\underline{c}+c) = ab \end{aligned}$$
- La formula di riduzione potrebbe essere facilmente applicata direttamente alle espressioni Booleane
- Il problema consiste nell'identificare
 - Sia tutti i termini su cui applicare la riduzione
 - Sia i tutti termini che partecipano a più riduzioni contemporaneamente e replicarli

Problemi

■ $f(a,b) = \underline{a}b + ab + a\underline{b}$

→ $(\underline{a}+a)b + a\underline{b} = b + a\underline{b}$

→ $\underline{a}b + a(b+\underline{b}) = \underline{a}b + a$



Nessuna delle due espressioni è minima!

■ L'espressione minima è $a+b$ ottenuta come

$$\underline{a}b + ab + a\underline{b} =$$

$$= \underline{a}b + ab + \underline{a}b + a\underline{b} =$$

$$= (a+\underline{a})b + a(b+\underline{b}) = b+a$$

Mappe di Karnaugh

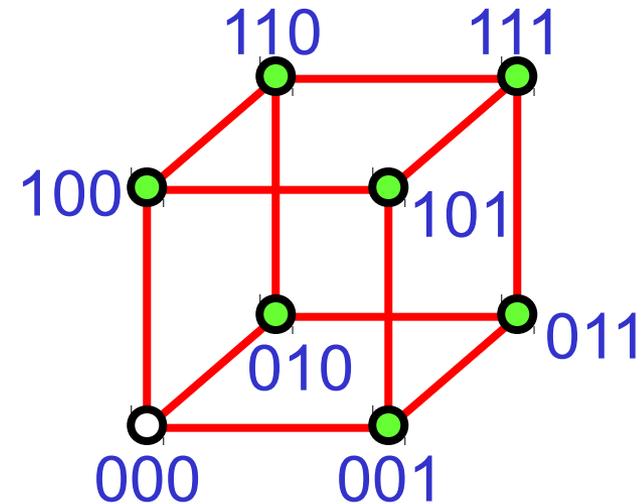
- Il metodo delle **mappe di Karnaugh** consente di risolvere direttamente i problemi identificati
 - Sia dovuti alla **replicazione dei termini**
 - Sia legati alla **identificazione dei termini da raggruppare**
- Il metodo delle mappe di Karnaugh è **grafico**
- La sua applicazione è semplice per un numero di variabili **fino a 4**
 - Risulta **complesso** per un numero di variabili da **5 a 6**
 - É **praticamente inattuabile** per un numero di variabili **superiori a 6**

Esempio

- Vogliamo trovare una copertura minima della funzione $OR(a,b,c)$
- Occorre
 - Identificare tutti gli implicanti primi essenziali
 - Un insieme minimo di implicanti che coprano i mintermini non coperti dagli implicanti primi essenziali

Rappresentazione nD

OR(a,b,c)			
a	b	c	o
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



- Si vede subito che gli implicant primari essenziali sono a , b e c che coprono interamente la funzione
- Ma questa rappresentazione è scomoda al crescere del numero di variabili!

Rappresentazione 2D

- Rappresentiamo lo spazio Booleano n -dimensionale sullo spazio a due dimensioni

		a	
		0	1
b	0	00	10
	1	01	11

2 variabili

		ab			
		00	01	11	10
c	0	000	010	110	100
	1	001	011	111	101

3 variabili

→ Gli indici delle colonne e delle righe in posizione adiacente differiscono solo di un bit

→ La prima e l'ultima colonna (riga) devono essere considerate adiacenti

→ Per le forme SP ogni casella in cui è presente un 1 corrisponde ad un mintermine

4 variabili

		ab			
		00	01	11	10
cd	00	0000	0100	1100	1000
	01	0001	0101	1101	1001
	11	0011	0111	1111	1011
	10	0010	0110	1110	1010

Mappe di Karnaugh

- Ogni casella della mappa corrisponde ad un punto dello spazio Booleano
- In ogni casella può essere messo il valore della funzione in quel punto

		<i>ab</i>			
		00	01	11	10
<i>c</i>	0	0	1	1	1
	1	1	1	1	1

Mappa di Karnaugh della funzione $OR(a,b,c)$

- Su questa mappa si possono identificare facilmente i sottocubi di dimensione massima (**implicanti primi**)

Mappe di Karnaugh

		<i>ab</i>			
		00	01	11	10
<i>c</i>	0	0	1	1	1
	1	1	1	1	1

Diagram illustrating a Karnaugh map for the function $OR(a,b,c)$. The map is a 2x4 grid with columns labeled *ab* (00, 01, 11, 10) and rows labeled *c* (0, 1). The cells containing 1s are circled in red, indicating the prime implicants: a (circles around the 1s in the 01 and 11 columns), b (circles around the 1s in the 01 and 11 columns), and c (circles around the 1s in the 01 and 11 columns). The labels *a*, *b*, and *c* are placed below the map to indicate the variables covered by the prime implicants.

$$OR(a,b,c) = a + b + c$$

Esempio

$$\begin{aligned} & \underline{a}bcd + a\underline{b}cd + ab\underline{c}d + abc\underline{d} = \\ & = \underline{a}c(\underline{b}d + \underline{b}d + \underline{b}d + \underline{b}d) = \\ & = \underline{a}c(\underline{b}(d + \underline{d}) + b(d + \underline{d})) = \\ & = \underline{a}c(\underline{b} + b) = \underline{a}c \end{aligned}$$

		<i>ab</i>			
		00	01	11	10
<i>cd</i>	00	1	0	1	1
	01	0	0	1	0
	11	1	1	0	0
	10	1	1	0	0

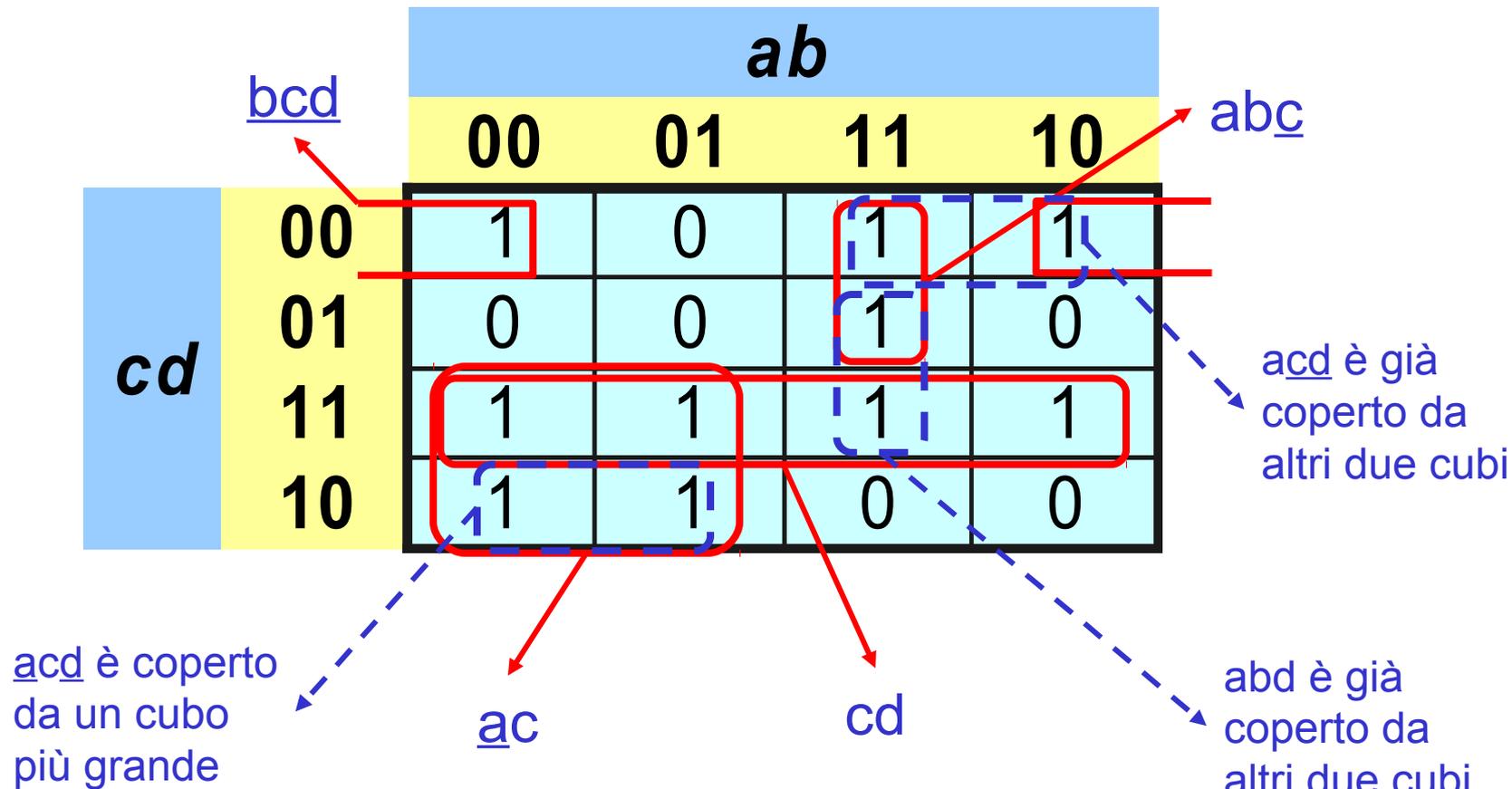
$$\begin{aligned} & \underline{a}bcd + a\underline{b}cd = \\ & = \underline{a}bc(d + \underline{d}) = \underline{a}bc \end{aligned}$$

- In una mappa a n variabili ad un cubo di 2^m caselle adiacenti corrisponde un termine prodotto di $n-m$ variabili
- Le $n-m$ variabili che restano sono quelle che nel cubo hanno lo stesso valore in tutte le caselle
- Una funzione f può essere rappresentata da una espressione SP nella quale i prodotti corrispondono ai cubi necessari per coprire tutte le caselle in cui è presente il valore 1

Regole

- La minimizzazione è ottenuta individuando il minimo numero di cubi e, a parità di numero, quelli col la **massima dimensione** garantendo la **copertura di tutti gli 1**
- Per ottenere un'espressione minima
 - **Non** si deve scegliere un cubo le cui caselle sono coperte da un cubo di dimensione maggiore
 - Se esistono più modi di coprire gli 1, bisogna scegliere la copertura con i cubi di **massima dimensione**
 - **Non** si devono scegliere cubi che coprono solo 1 di ***f*** già coperti da un insieme di altri cubi già scelti

Esempio



Esempio

		<i>ab</i>			
		00	01	11	10
<i>cd</i>	00	1	0	1	1
	01	0	0	1	0
	11	1	1	1	1
	10	1	1	0	0

$$f(a,b,c,d) = abc + bcd + ac + cd$$

Esempio

		<i>ab</i>			
		00	01	11	10
<i>cd</i>	00	1	0	1	1
	01	0	0	1	0
	11	1	1	1	1
	10	1	1	0	0

$$f(a,b,c,d) = abc + bcd + ac + acd$$

Esempio

		<i>ab</i>			
		00	01	11	10
<i>cd</i>	00	1	0	1	1
	01	0	0	1	0
	11	1	1	1	1
	10	1	1	0	0

$$f(a,b,c,d) = abc + bcd + acd + cd$$

Esempio

■ Le tre funzioni precedenti

$$\rightarrow f(a,b,c,d) = abc\underline{c} + \underline{bcd} + \underline{ac} + cd$$

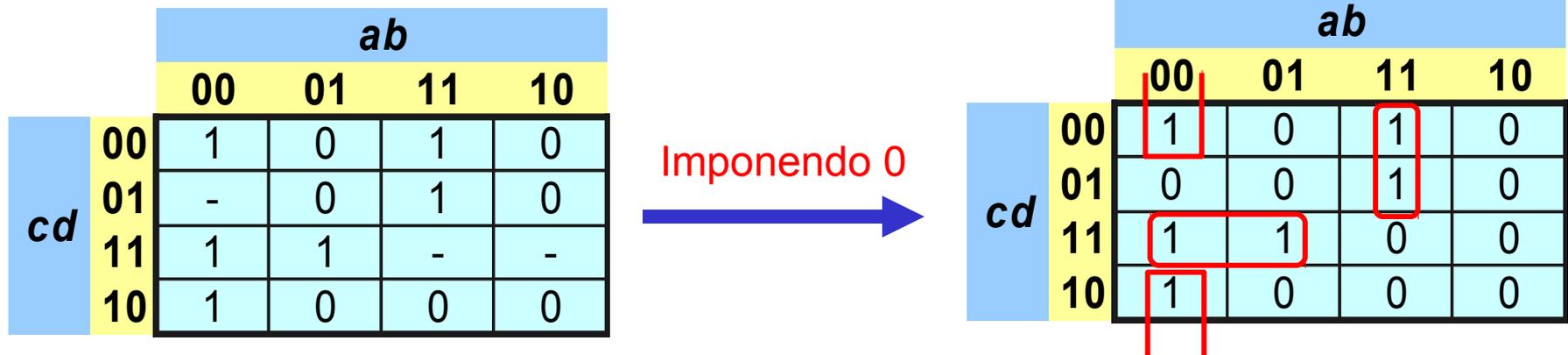
$$\rightarrow f(a,b,c,d) = abc\underline{c} + \underline{bcd} + \underline{ac} + acd$$

$$\rightarrow f(a,b,c,d) = abc\underline{c} + \underline{bcd} + \underline{acd} + cd$$

Sono equivalenti da un punto di vista funzionale ma la prima è quella minima

Funzioni non Completamente Specificate

- Le condizioni di indifferenza possono essere sfruttate per incrementare la dimensione di cubi



$$f(a,b,c,d) = \underline{a}b\underline{d} + a\underline{b}\underline{c} + \underline{a}c\underline{d}$$

Funzioni non Completamente Specificate

- Le condizioni di indifferenza possono essere sfruttate per incrementare la dimensione di cubi

		<i>ab</i>			
		00	01	11	10
<i>cd</i>	00	1	0	1	0
	01	-	0	1	0
	11	1	1	-	-
	10	1	0	0	0

Imponendo 1

		<i>ab</i>			
		00	01	11	10
<i>cd</i>	00	1	0	1	0
	01	1	0	1	0
	11	1	1	1	1
	10	1	0	0	0

$$f(a,b,c,d) = \underline{a}b + ab\underline{c} + cd$$