Mappe di Karnaugh (K-Map)

▼ Creatore originale: @LucaCaffa

Una mappa di Karnaugh è uno strumento grafico utilizzato in logica booleana per semplificare espressioni logiche o funzioni booleane.

È rappresentata attraverso una tabella che serve per rappresentare i valori di verità di una funzione logica in modo ordinato, così da rendere più facile trovare gruppi di valori uguali (1 o 0) e, quindi, poter semplificare l’espressione logica.

Esempio - K-map di 3 variabili

Prendiamo in esempio la tabella di verità a lato.

Scriviamo una tabella che ha come indice delle colonne le combinazioni degli ingressi xy, mentre come indice delle righe ha le combinazioni di z.

Ogni cella con il valore 1 è descritta da un mintermine.

x	y
0	0
0	0
0	1
0	1
1	0
1	0
1	1
1	1

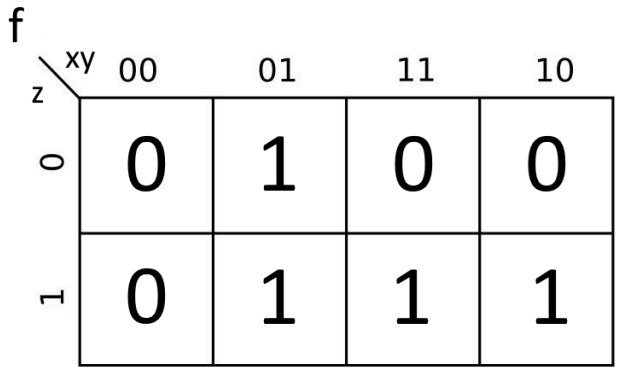


Tabella di Karnaugh associata alla tabella di verità a lato

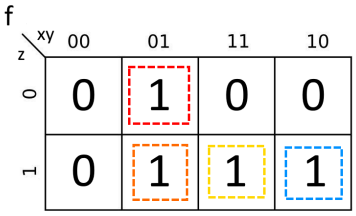
Il nostro obiettivo è semplificare la funzione logica, in modo da avere la sua forma meno costosa possibile dal punto di vista dei mintermini.

Per esempio, se prendessimo tutte le caselle con 1, avremmo 4 mintermini da 3 variabili ciascuno.

Possiamo, invece, prendere gli 1 a rettangoli grandi di dimensione pari a una potenza di due, in modo da semplificare la funzione, ricordando che, rispetto a un determinato rettangolo, si prendono i valori all'interno degli indici che non variano per il rettangolo:

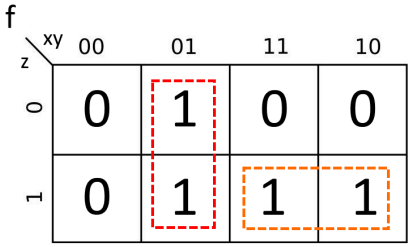
- se prendiamo 4 rettangoli da 1 blocco ognuno avremo:

$$f(x,y,z) = \overline{x}y\overline{z} + \overline{x}yz + xyz + x\overline{y}z$$



- se prendiamo 2 rettangoli da 2 blocchi ognuno avremo:

$$f(x,y,z) = \overline{x}y + xz$$



Si può notare come nel primo tipo di raggruppamento abbiamo 4 mintermini, mentre nel secondo ne abbiamo due, e quindi il secondo possibile raggruppamento risulta meno costoso.

Implicanti

Un implicante è un gruppo di celle adiacenti (dove il valore è 1) in una mappa di Karnaugh che può essere semplificato a un singolo termine booleano.

Caratteristiche

- Ogni implicante rappresenta una combinazione di variabili che è comune a tutte le celle del gruppo;
- Gli implicanti possono essere di diverse dimensioni, da un singolo 1 (minimo implicante) a gruppi più grandi;
- Più grande è l'implicante, più semplice è l'espressione booleana che rappresenta.

Implicanti primi

Un implicante primo è un implicante che non può essere esteso ulteriormente senza includere celle che non contengono il valore 1.

Un implicante è considerato primo se non può essere diviso in gruppi più piccoli che coprono gli stessi 1.

Implicanti essenziali

Un implicante essenziale è un implicante che copre almeno un 1 che non è coperto da alcun altro implicante.

Gli implicanti essenziali sono quelli che non possono essere evitati se si vogliono coprire tutti i valori 1 nella mappa di Karnaugh.

Passi di minimizzazione

In maniera informale, si può individuare una sequenza di passi per trovare la soluzione minima:

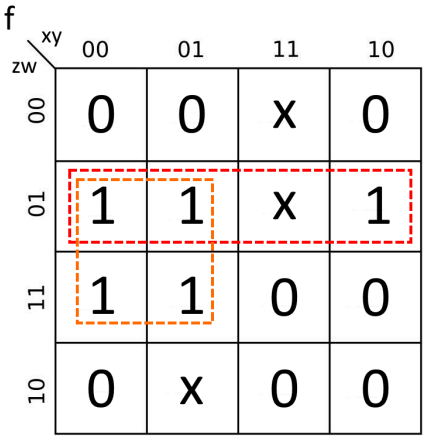
1. si trovano tutti gli implicanti primi;
2. si includono nella soluzione tutti gli implicanti essenziali;
3. si prende un insieme di implicanti primi (non essenziali) che coprono tutti gli 1 non ancora presi;
4. si cerca la minima sovrapposizione tra implicanti primi.

Copertura con valori “don’t care”

Siamo in condizione di “don’t care” quando un determinato valore della funzione non ci interessa. In celle di questo tipo, graficamente si indicano valori di “don’t care” sulla mappa con il simbolo “-” o “x”. Possiamo considerare queste celle come contenenti sia 0 che 1, a seconda dell’utilità nel creare insiemi di implicanti di dimensione massima.

Ad esempio, se abbiamo x=0, y=1, z=1 e w=0 e non ci importa sapere quanto vale la funzione in quella cella, indichiamo il valore non importante con “x”.

Nella K-map di fianco, i simboli “x” sono i valori don’t care. In questo caso, ci conviene prendere il valore “don’t care” in posizione 1101 (considerandolo come 1), ma non ci conviene prendere quello in posizione 1100, né quello in posizione 0110, poiché porterebbero ad una soluzione peggiore.



Esempio di copertura con don’t care

Metodo di Quine-McCluskey (QMC)

Il metodo di Quine-McCluskey è un algoritmo sistematico per la minimizzazione, simile alla mappa di Karnaugh, ma più adatto a funzioni con molte variabili.

Si compone di due fasi che usano delle tabelle per memorizzare i vari risultati intermedi:

- espansione: generazione di tutti gli implicanti primi;
- copertura: scelta del minor numero di implicanti primi.

Spiegazione di ogni passo del metodo QMC

Per spiegare ogni passo, è più comodo utilizzare un esempio. Ipotizziamo di avere la tabella che contiene i valori degli ingressi e della funzione di uscita.

La prima colonna, identificata con il simbolo i, è utilizzata solo per contare e identificare le righe, ma non è un ingresso.

i	x	y
0	0	0
1	0	0
2	0	1
3	0	1
4	1	0
5	1	0
6	1	1

i	x	y
7	1	1

Fase di Copertura

Definiamo i vari passi:

1. guardare la tabella e prendere i mintermini dove la funzione vale 1. In questo esempio, abbiamo i mintermini 0, 2, 3, 4, 6, 7;
2. dobbiamo prendere i mintermini e metterli in dei gruppi, secondo i seguenti ragionamenti:
   a. analizziamo m_0, cercando di capire quanti "1" ha tra i suoi ingressi. Non avendone nessuno, si cercano altri mintermini che non hanno "1" negli ingressi;
   b. dato che non ci sono mintermini che non hanno 1 negli ingressi, m_0 è in un gruppo da solo;
   c. procediamo con il mintermine m_2, cercando di capire quanti "1" ha tra i suoi ingressi. Avendone uno, $(m_2 = 010_2)$, si cercano altri mintermini che contengono un singolo "1" negli ingressi;
   d. si sono altri mintermini che hanno un "1" tra gli ingressi, come $m_4 = 100_2$, e quindi mettiamo m_2 e m_4 nello stesso gruppo;
   e. continuando cosi, si arriva alla situazione descritta nella tabella a lato, dove i gruppi sono descritti dai diversi colori.

i	x
0	0
2	0
4	1
3	0
6	1
7	1

3. una volta costruita la tabella con i gruppi, dobbiamo cercare di ridurla ulteriormente, combinando i gruppi tra loro, secondo i seguenti ragionamenti:
   a. è necessario chiedersi se due mintermini di gruppi diversi possono unirsi, a patto di considerare una variabile in meno, e quindi i due mintermini si possono unire se differiscono solo di un bit;
   b. prendiamo come esempio m_0 e m_2, i bit di ingresso sono, rispettivamente, 000_2 e 010_2. Questi due mintermini differiscono solo per il valore di y, quindi se si scrive $0\text{-}0_2$, si possono unire e indicare entrambi in una singola riga;
   c. ripetere questo ragionamento per tutti i gruppi, ricordando che (0,2) significa che ho unito m_0 e m_2.

i	x	y	z
(0,2)	0	-	0
(0,4)	-	0	0
(2,3)	0	1	-
(2,6)	-	1	0
(4,6)	1	-	0
(3,7)	-	1	1
(6,7)	1	1	-

4. ripartendo dal passo 2, si ripete il ragionamento fatto fino ad ora sui gruppi, cercando di capire se si può semplificare ancora, finché non si arriva ad una situazione di stallo;
5. alla fine delle semplificazioni, si avrà una tabella come quella a lato. La colonna p serve solo per dare un nome alle soluzioni parziali trovate fino ad ora.

i	x	y
(0,2,4,6)	-	-
(2,3,6,7)	-	1

Fase di espansione

Lo scopo di questa fase è scegliere il minor numero di implicanti primi possibili utilizzabili per coprire tutta la funzione. Esistono diversi metodi matematici per risolvere questo problema, ma non sono trattati nel corso, quindi possiamo sceglierli "ad occhio":

1. iniziamo dall'ultima tabella trovata. In questo momento non ci interessa la colonna i, che serviva per distinguere i gruppi, ma dobbiamo ragionare per righe;

i	x	y
(0,2,4,6)	-	-

i	x	y
(2,3,6,7)	-	1

2. costruiamo una tabella di copertura, cioè una tabella che ci dice, per ogni implicante primo, quale mintermine è coperto. Ricordiamo che i mintermini sono tutti i valori nella colonna i, mentre gli implicanti sono le righe della colonna p;

   a. guardiamo l'ultima tabella ottenuta e mettiamo una colonna per ogni mintermine. Quindi, inseriamo i valori $m_0, m_2, m_3, m_4, m_6, m_7$ in ordine sulle colonne;

	m_0	m_2	m_3	m_4	m_6

   b. possiamo inserire, in ragione di uno per riga, gli implicanti primi. Quindi, inseriamo p_0, p_1;

	m_0	m_2	m_3	m_4	m_6	m_7
p_0						
p_1						

   c. segniamo, per ogni riga, e quindi per ogni implicante, quale mintermine copre, mettendo il simbolo ✅, oppure un 1 o una X;

	m_0	m_2	m_3	m_4	m_6	m_7
p_0						
p_1						

   d. per sapere quali mintermini copre un implicante, torno all'ultima tabella della fase di espansione e guardo la corrispondente cella della colonna i.

	m_0	m_2	m_3	m_4	m_6
p_0	✅	✅		✅	✅
p_1		✅	✅		✅

Per p_0 abbiamo (0,2,4,6), e quindi p_0 copre m_0, m_2, m_4, m_6.

Ripeto il procedimento per p_1.

e. abbiamo, quindi, la tabella di copertura completa**.** Dobbiamo prendere degli implicanti (righe) in modo da coprire tutti i mintermini (colonne), cosi che nessun mintermine rimanga tralasciato.

   In questo esempio abbiamo già finito, poiché per coprire tutti i mintermini devo prendere entrambi gli implicanti p_0 e p_1.

Fase di espansione - Ulteriore esempio

Poiché dallo scorso esempio non avevamo bisogno di considerazioni finali, prendo un'altra tabella di copertura già costruita, così da analizzarla di passo in passo.

	m_1	m_2	m_3	m_6	m_9	m_{10}
p_0						
p_1						
p_2	✅		✅		✅	
p_3	✅				✅	
p_4		✅	✅			✅
p_5		✅		✅		✅

In generale, l'unica regola che applichiamo è prendere gli implicanti che coprono più mintermini possibili.

Definiamo i vari passi:

1. controlliamo se esistono degli implicanti primi essenziali, ovvero delle righe che coprono dei mintermini che non copre

	m_1	m_2	m_3	m_6	m_9	m_{10}
p_0						
p_1						
p_2	✅		✅		✅	
p_3	✅				✅	
p_4		✅	✅			✅
p_5		✅		✅		✅

nessun altro.

Nel nostro esempio, p_5 copre m_6, ma m_6 non è coperto da nessun altro implicante primo essenziale. Nella soluzione, p_5 sarà presente.

Segniamo in verde le colonne che abbiamo preso mettendo p_5 nella soluzione;

2. si può notare come m_{12} è ricoperto sia da p_0 che da p_1. In questo caso è indifferente quale dei due viene preso, e noi prendiamo quindi p_0;

	m_1	m_2	m_3	m_6	m_9	m_{10}
p_0						
p_1						
p_2	✅		✅		✅	
p_3	✅				✅	
p_4		✅	✅			✅
p_5		✅		✅		✅

3. I mintermini mancanti sono m_1, m_2, m_9, m_{11}, e sono tutti coperti da p_2, quindi lo prendiamo nella soluzione.

	m_1	m_2	m_3	m_6	m_9	m
p_0						
p_1						
p_2	✅		✅		✅	
p_3	✅				✅	
p_4		✅	✅			✅
p_5		✅		✅		✅

Avremmo potuto prendere anche p_3, p_4 insieme, ma non sarebbe stata la soluzione minima, e quindi neanche la migliore.

Dopo questo procedimento, abbiamo trovato la seguente soluzione.

$$f(x, y, z, w) = p_0 + p_2 + p_5 = xy\overline{z} + \overline{y}w + z\overline{w}$$

Metodo di Espresso

Il metodo Espresso è un algoritmo avanzato per la minimizzazione di funzioni booleane. È più efficiente del metodo di Quine-McCluskey, soprattutto per funzioni complesse con molte variabili.

Il problema del metodo QMC è la crescita esponenziale degli implicanti primi, oltre al fatto che il riconoscimento della copertura minima può rientrare nella classe dei problemi NP-completi.

Il metodo espresso opera in più passaggi:

- espansione (Expand): trasformare ogni termine in un implicante primo più grande possibile;
- copertura ottima (Irredundant Cover): trovare il minor numero di implicanti primi che coprono tutti i mintermini della funzione;
- riduzione (Reduce): ridurre gli implicanti per migliorare la copertura globale;
- iterazione (Repeat Until Convergence): le fasi Expand, Reduce e Irredundant vengono ripetute finché non si raggiunge una soluzione stabile.

▼ Esempio - Minimizzazione con Espresso

Consideriamo la seguente funzione, con valori "don't care" in $\sum_d(0, 3)$.

$$F(A, B, C) = \sum(1, 2, 5, 6, 7)$$

AB/C	00	01
0	X	1
1	1	X

1. Eseguiamo l'espansione.

   I mintermini sono: `001`, `010`, `101`, `110`, `111`.

   Gli implicanti primi trovati potrebbero essere:

   - `0-1` (copre 001 e 101): $\overline{A}C$;
   - `-10` (copre 010 e 110): $B\overline{C}$;
   - `11-` (copre 110 e 111): AB.

2. Eseguiamo la riduzione e la copertura.

   La tabella di copertura mostra che:

   - $B\bar{C}$ copre `010` e `110` ;
   - $\bar{A}C$ copre `001` e `101`;
   - AB copre `110` e `111`.

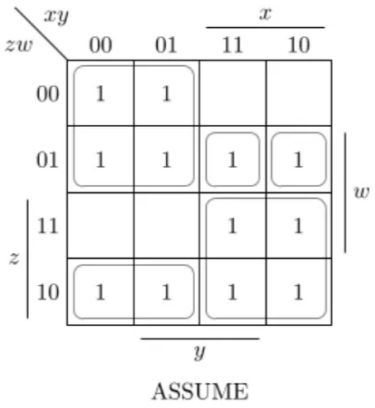
La soluzione minima potrebbe, quindi, essere:

$$F(A, B, C) = B\overline{C} + \overline{A}C + AB$$

▼ Esempio - Minimizzazione con Espresso 2

1. partiamo dalla k-map a lato.

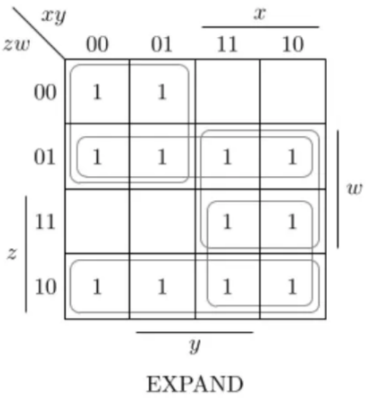
   Il nostro obiettivo è considerare gli implicanti più grandi possibili, per poi andare a ridurli, verificando ogni volta la totale copertura;



Assunzione 1

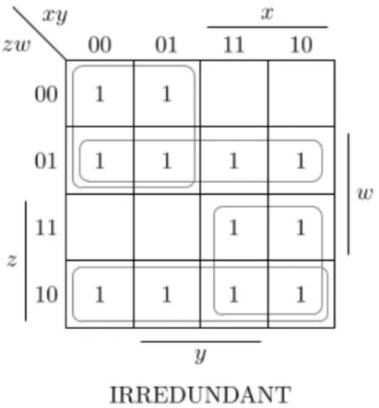
2. fase di espansione, in cui prendiamo gli implicanti in modo da coprire tutta la tabella.

   Stiamo cercando l'insieme con gli implicanti più grandi possibili. La figura a lato mostra come questo sia l'insieme massimale.



Espansione 1

3. fase di copertura ottima, in cui, dall'insieme appena trovato, estraiamo una soluzione non ridondante, in modo che copra tutti i mintermini.



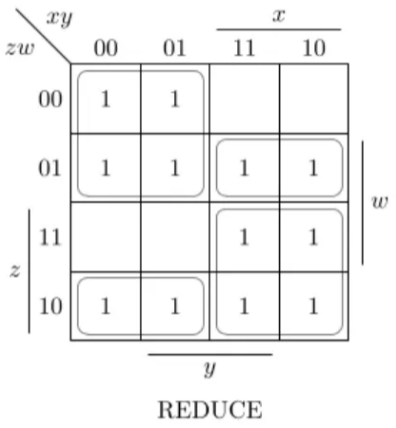
Copertura ottima 1

4. potremmo aver finito, oppure potrebbe esistere una soluzione migliore.

   Quello che si fa è ridurre, ad uno ad uno, la dimensione di ogni implicante, garantendo sempre la copertura totale dei mintermini.

   Se si riesce a fare una modifica, iteriamo i passi in modo continuare la ricerca di una soluzione ottima.

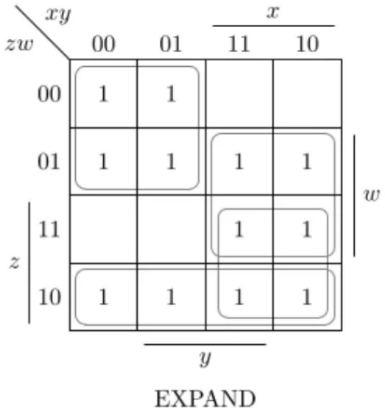
   Qui abbiamo ridotto l'implicante in basso a sinistra, per cui abbiamo capito che il numero di implicanti può essere 4. La soluzione trovata non è ottima, e dobbiamo tornare al punto 2, cercando solo 4 implicanti;



Riduzione 1

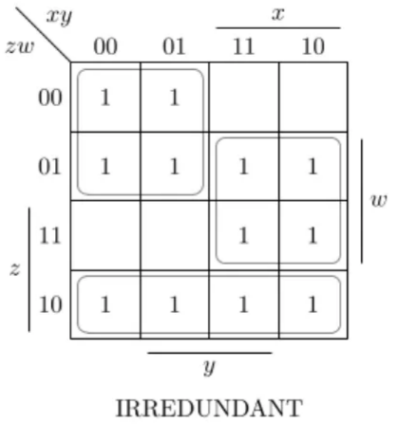
5. si espande nuovamente, ma con 4 implicanti.

   Prendiamo l'insieme più grande di implicanti, cercandone 4;



Espansione 2

6. fase di copertura ottima, in cui prendiamo un sotto insieme dall'insieme trovato al punto 5. Adesso abbiamo trovato una soluzione ottima, e l'algoritmo finisce.



Copertura ottima 2