Università degli studi di Parma
Dipartimento di Ingegneria dell'Informazione
Reti Logiche A – a.a. 2007/08

Sintesi di Reti Combinatorie
Mappe di Karnaugh

Introduzione
Funzioni completamente specificate
Le condizioni di indifferenza
Funzioni non completamente specificate

stefano.caselli@unipr.it
http://www.ce.unipr.it/people/caselli
http://corsi.unipr.it

Sintesi di reti combinatorie a due livelli

- Obiettivo:
  - Ridurre la complessità di una (o più) funzioni booleane espresse in forma di Prodotto di Somme (PS) o di Somma di Prodotti (SP)
  - Ci si riferirà principalmente alla forma SP; la metodologia, con ovvie differenze, si applica anche alla forma PS
- Nella sintesi a due livelli gli obiettivi sono due:
  - Riduzione del numero dei termini prodotto (principale)
  - Riduzione del numero di letterali (secondario)
- Metodologie di sintesi ottima:
  - Metodo delle mappe di Karnaugh
  - Metodo di Quine - Mc Cluskey
  - Euristiche per sintesi a due livelli

Semplificazione di espressioni SP

- Identificare forme minime a due livelli applicando la regola di riduzione:

  aZ + a'Z = (a + a')Z = Z
- In cui **Z** è un termine prodotto di *n-1* variabili
  - Esempio

    a**b**cd' + a**b'**cd' = **acd'**
- La riduzione può essere applicata iterativamente
  - Esempio

    abc'd' + abc'd + abcd' + abcd =

    abc'(d'+d) + abc(d'+d) =

    abc' + abc =

    ab(c'+c) = ab

Semplificazione di espressioni SP

- Il metodo appena visto
  - È applicato ad un numero di termini pari a 2^n
  - Mantiene inalterato il numero dei livelli
  - Somme di prodotti rimangono tali
  - Al più, tali espressioni possono banalizzarsi
    - Divengono semplici prodotti
    - Divengono somme di letterali
    - Divengono costanti

Semplificazione di espressioni SP

- La relazione vista può essere applicata direttamente alle espressioni algebriche che definiscono una rete
- Il problema consiste nell'identificare:
  - Tutti i termini su cui applicare la riduzione
    - Non è sempre immediato identificare tutti termini su cui applicare la regola di riduzione identificata
  - Tutti i termini che partecipano a più riduzioni contemporaneamente e replicarli
    - Si ricordi che, per le proprietà dell'algebra di Boole, la relazione

      x + x = x
    - può essere applicata anche come

      x = x + x

Semplificazione di espressioni SP

- Esempio di replicazione dei termini:

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

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

1 2

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

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

Semplificazione di espressioni SP

- Esempio di replicazione dei termini:

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

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

$$\begin{aligned} f(a,b) &= a'b + ab + ab + ab' = \\ &= (a'+a)b + a(b+b') = \\ &= a + b \end{aligned}$$

Le mappe di Karnaugh

- Il metodo delle mappe di Karnaugh consente di risolvere direttamente i problemi identificati
  - Replicazione dei termini
  - Identificazione dei termini da raggruppare
- Il metodo delle mappe di Karnaugh è *grafico*
  - La sua applicazione è semplice per funzioni di un numero di variabili fino a 4
  - Risulta più complesso per un numero di variabili da 5 a 6
  - È praticamente inattuabile per un numero di variabili superiore a 6

Le mappe di Karnaugh

- Una mappa di Karnaugh
  - È uno schema deducibile dalla rappresentazione geometrica delle configurazioni binarie
- Definizione di *distanza di Hamming*
  - Numero di bit che cambia nel passare da una configurazione binaria ad un'altra
  - Esempio
    - Distanza di Hamming tre le configurazioni 01001 e 10101

      `01001`

      `10101`
    - La distanza è pari a 3 poiché cambiano 3 bit

Le mappe di Karnaugh

- La regola di riduzione
  - Consiste nell'identificare le configurazioni binarie associate ai termini prodotto che sono a distanza di Hamming unitaria
  - A tali configurazioni corrispondono coppie di mintermini in cui una sola variabile è in forma vera in un mintermine e complementata nell'altro
  - Esempio:
    - **abcd' + ab'cd'**

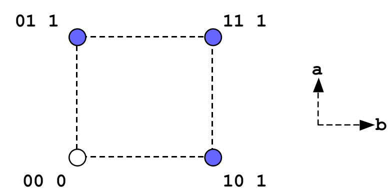
      abcd' = 1110

      ab'cd' = 1010
    - I mintermini 1110 e 1010 sono ad una distanza di Hamming pari ad 1

Le mappe di Karnaugh

- Funzione binaria di *n* variabili
  $f:\ \{0,1\}^n \rightarrow \{0,1\}$
- Può essere rappresenta
  - Mediante tabella della verità
  - Mediante rappresentazione geometrica cartesiana in uno spazio a *n* dimensioni in cui gli assi sono le variabili della funzione
- Esempio a 2 variabili

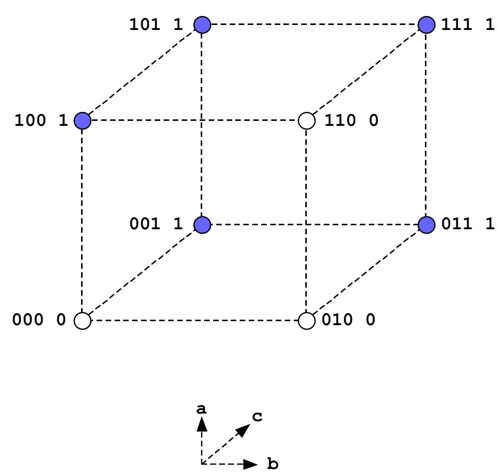
a b	f(a,b)
0 0	0
0 1	1
1 0	1
1 1	1



Le mappe di Karnaugh

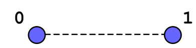
- Esempio a 3 variabili

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



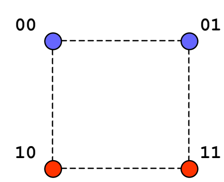
N-Cubi

- Nella rappresentazione cartesiana di una funzione in uno spazio a n dimensioni, collegando i vertici le cui configurazioni sono a distanza di Hamming unitaria si ottiene un *n-cubo*
- Spazio a 1 dimensione (1 variabile)
  - È una linea
  - L’1-cubo è un segmento
    - I due vertici sono associati alle configurazioni 0 e 1



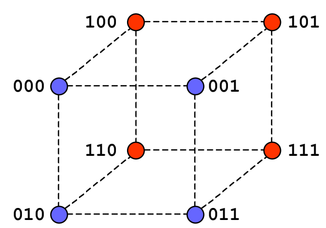
N-Cubi

- Spazio a 2 dimensioni (2 variabili)
  - È il piano
  - Il 2-cubo è un quadrato
    - Si ottiene dall’1-cubo per proiezione
    - Si premette 0 alle configurazioni dei vertici originali, 1 a quelle dei vertici proiettati



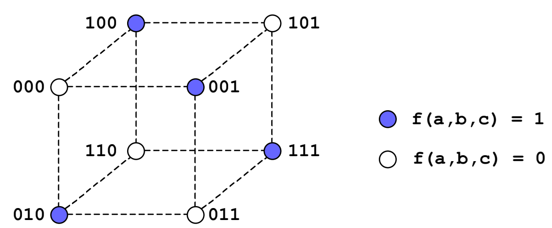
N-Cubi

- Spazio a 3 dimensioni (3 variabili)
  - È lo spazio tridimensionale
  - Il 3-cubo è un solido
    - Si ottiene dal 2-cubo per proiezione
    - Premettendo 0 alle configurazioni dei vertici originali, 1 a quelle dei vertici proiettati



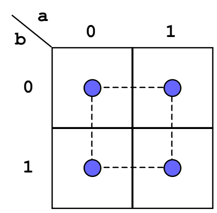
N-Cubi e Tabelle della verità

- Si può trasportare
  - Una tabella delle verità a n variabili su un n-cubo
  - Marcando opportunamente i nodi associati a 0 e 1
- Si ricorda che
  - Due configurazioni sono a distanza unitaria (adiacenti) se e solo se i vertici associati sono collegati da un lato



N-Cubi e Mappe

- La rappresentazione in uno spazio a n dimensioni non è maneggevole
  - Già per sole tre dimensioni non è di semplice utilizzo
  - Si passa allo sviluppo nel piano dei cubi
- Al cubo sviluppato nel piano
  - Che ha 2^n vertici
- Si sovrappone una mappa
  - Con 2^n caselle organizzate secondo righe e colonne



N-Cubi e Mappe

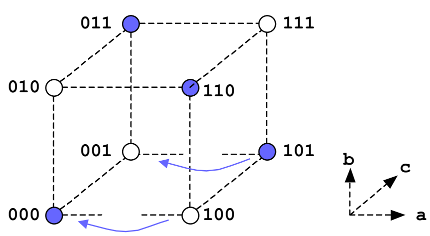
- Una mappa così realizzata è una *mappa di Karnaugh*:
  - Le configurazioni assunte dalle variabili di ingresso danno origine gli indici di riga e colonna della mappa
  - In ogni casella si trascrive il valore assunto dalla funzione quando la configurazione delle variabili corrisponde a quella delle coordinate che contrassegnano le caselle
  - In una mappa di Karnaugh, due caselle che condividono un lato di un n-cubo corrispondono a due configurazioni di variabili adiacenti (distanza di Hamming pari ad 1)

f(a,b) = ONset(1,2)

b \ a	0	1
0	0	1
1	1	0

N-Cubi e Mappe

- Lo sviluppo di un 3-cubo implica il taglio del cubo
- Il taglio deve mantenere intatta la adiacenza fra vertici
  - Si presti molta attenzione all'ordinamento delle coordinate
  - L'ordinamento delle coordinate mantiene le distanze di Hamming e non coincide con la numerazione consecutiva



adiacenti

c \ ab	00	01	11	10
0	1	0	1	1
1	0	1	0	1

adiacenti

Raggruppamenti

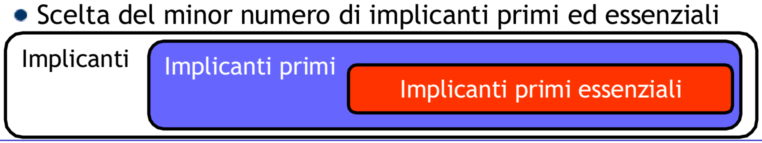
- Caratteristiche delle mappe
  - Un implicante è una funzione p associata ad un termine prodotto di m letterali con $1 \leq m \leq n$ tale per cui $f \geq p$
    - Cioè p *implica* f.
      - Per ogni 1 in p corrisponde un 1 in f.
    - Un mintermine è un implicante in cui m=n.

```
F(a,b,c,d) = a'b'c'd' +
             a'bc'd' +
             a'b'cd' +
             ab'c'd +
             ab'cd +
             ab'cd'
```

cd \ ab	00	01	11	10
00	1	1	0	0
01	0	0	0	1
11	0	0	0	1
10	1	0	0	1

Metodo

- Individuare gli implicanti primi e primi essenziali:
  - Implicante primo
    - Funzione p associata ad un termine prodotto a cui corrisponde un raggruppamento di dimensione massima
      - Cioè, l'eliminazione di un qualsiasi letterale dal prodotto genera un prodotto tale che la nuova funzione q *non implica* f
  - Implicante primo essenziale
    - Implicante primo che copre uno o più 1 non coperti da nessun altro implicante primo.
- Copertura:
    - Scelta del minor numero di implicanti primi ed essenziali



Metodo

- Identificare una forma SP che
  - Includa il numero minimo di implicanti
    - A parità di numero di prodotti, l'implicante associato al prodotto col minimo numero di letterali (definita come forma minima)
  - Garantisca la copertura di tutti gli 1 della funzione
- Teorema:
  - Esiste una forma minima costituita da soli implicanti primi
    - Gli implicanti primi essenziali devono essere inclusi nella forma minima
    - Una forma minima costituita da soli implicanti primi essenziali è unica
      - La condizione è solo sufficiente

Metodo

- Esempio 1

ab \ cd	00	01	11	10
00	1	1	0	0
01	1	1	0	1
11	0	0	0	1
10	1	0	0	1

Raggruppamento rettangolare di dimensione massima

Raggruppamento

Raggruppamento scorretto:
Sono possibili solo 2, 4, 8, 2^k caselle

Metodo

- Esempio 2

ab \ cd	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	0	0	1	1
10	0	0	0	1

Raggruppamento essenziale di dimensione massima

Raggruppamento di dimensione massima

1 Mintermine appartenente ad un solo implicante primo

Metodo

- Ad ogni raggruppamento è associato un termine prodotto
- Il termine prodotto associato ad un implicante è ottenuto:
  - Identificando le variabili che non cambiano mai di valore
  - Riportando ogni variabile in modo diretto
    - Se il valore che essa assume è 1
  - In modo complementato
    - Se il valore da essa assunto è 0
- Osservazione:
  - Un numero di 2^m "1" raccolti costituisce un raggruppamento rettangolare di ordine *m*
  - Esso produce un termine prodotto di *n-m* letterali dove *n* è il numero di variabili della funzione
    - Esempio: per una funzione di 4 variabili un implicante che raccoglie 4 "1" è associato ad un prodotto di 2 variabili

Metodo

- Esempio 3

ab \ cd	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	0	0	1	1
10	0	0	0	1

Raggruppamento

0/1 Cambiamento di valore all'interno dell'n-cubo

La variabile **a** non cambia valore: **a** = 0, **a** compare negata nel prodotto
La variabile **b** cambia valore: **b** non compare nel termine prodotto
La variabile **c** non cambia valore: **c** = 0, **c** compare negata nel prodotto
La variabile **d** cambia valore: **d** non compare nel termine prodotto

Il termine prodotto corrispondente è **a'c'**

Copertura

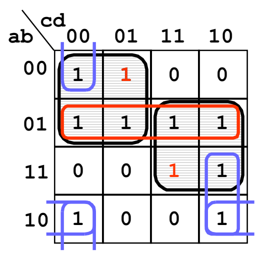
- Una copertura è
  - Un sottoinsieme degli implicanti identificati tale per cui nessun 1 della funzione rimane scoperto
  - Poiché ogni implicante scelto aumenta il costo della realizzazione della funzione, il numero di implicanti da scegliere deve essere il minore possibile.
- L'obiettivo è la riduzione del costo
  - Identificazione della copertura di minima cardinalità:
    - Sottoinsieme degli implicanti primi e primi ed essenziali identificati che realizza una copertura della funzione che è di cardinalità minima

Copertura

- Scelta degli implicanti per realizzare la copertura:
  - Si scelgono tutti gli implicanti primi essenziali
    - Sono parte della copertura poiché "sono essenziali" e, quindi, non è possibile fare a meno di loro
  - Si eliminano gli implicanti primi coperti da quelli essenziali
    - Gli implicanti eliminati, detti completamente ridondanti, coprono degli 1 che sono già ricoperti da quelli essenziali
  - Si seleziona il numero minore di implicanti primi rimasti
    - Gli implicanti residui sono detti parzialmente ridondanti
  - Osservazione:
    - La scelta viene fatta seguendo un criterio basato sulla sola osservazione della mappa

Copertura

- Esempio 1:
  - Selezione degli implicanti primi essenziali



Implicanti primi essenziali

a'c' bc

Implicanti primi completamente ridondanti

a'b

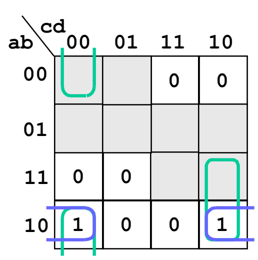
Implicanti primi

b'c'd' acd' ab'd'

F(a,b,c,d) = a'c' + bc + ...

Copertura

- Esempio 1:
  - Copertura dei rimanenti termini
  - Forma minima



Implicanti primi

b'c'd' acd' ab'd'

Implicanti primi parzialmente ridondanti

b'c'd' acd'

F(a,b,c,d) = a'c' + bc + ab'd'

Copertura

- Esempio 2:
  - Forme equivalenti

ab \ cd	00	01	11	10
00	1	1	0	0
01	0	1	1	0
11	0	0	1	1
10	1	0	0	1

Implicanti primi

a'b'c'   a'bd   abc   ab'd'

b'c'd'   a'c'd   bcd   acd'

F(a,b,c,d) = a'b'c' + a'bd + abc + ab'd'

F(a,b,c,d) = b'c'd' + a'c'd + bcd + acd'

Condizioni di indifferenza

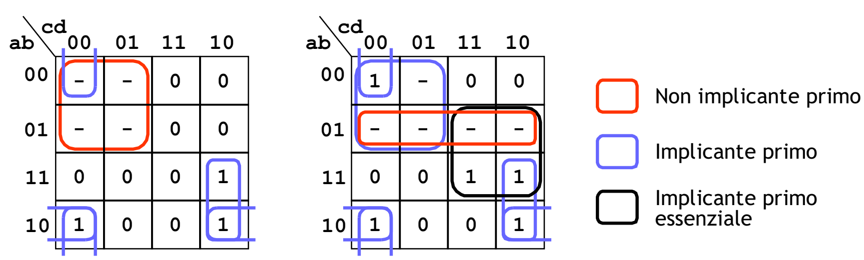
- *Condizioni di indifferenza* o *don't care*
  - La specifica di un progetto (la descrizione di quello che si vuole progettare) contiene, spesso, delle condizioni di indifferenza
  - Le condizioni di indifferenza corrispondono a configurazioni di ingresso per le quali il valore dell'uscita non è noto e non è neppure di interesse sapere quanto può valere
  - Questo accade quando:
    - Le configurazioni di ingresso non si presentano mai
    - Le configurazioni di ingresso consentono di trascurare l'uscita della rete (ad esempio perchè la risposta complessiva del circuito è determinata da un'altra uscita).

Condizioni di indifferenza

- Le configurazioni di ingresso per le quali il valore dell'uscita è non specificato costituiscono il *DCset* della funzione
- Sulla tabella della verità (o in una mappa di Karnaugh) il valore non specificato della funzione si indica con i simboli "**-**" o "**x**"
- Le condizioni di indifferenza sono *gradi di libertà* nel processo di sintesi
  - Ai DC si può assegnare il valore 0 o 1 a seconda di quanto conviene per minimizzare la funzione
  - Una condizione di indifferenza non deve necessariamente essere coperta da un implicante

Condizioni di indifferenza

- Importante:
  - Gli implicanti realizzati solamente mediante condizioni di indifferenza non hanno alcuna utilità e non sono primi
  - Un implicante primo non diventa essenziale quando è l'unico a coprire una data condizione di indifferenza



Esempio 1

- Sintetizzare una funzione di 4 ingressi a, b, c, d
  - Gli ingressi codificano cifre decimali in codice BCD
  - L'uscita vale 1 se e solo se la cifra in ingresso è minore o uguale a 3 oppure maggiore o uguale a 8
- Dalla specifica risulta che
  - Delle 16 possibili configurazioni degli ingressi solo 10 potranno effettivamente presentarsi (Codifica BCD)
  - In corrispondenza delle configurazioni di valori impossibili, non interessa il valore che la funzione può assumere
  - In questi casi, il valore dell'uscita è non specificato

Esempio 1

Tabella della verità

BCD	a	b	c	d	f
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	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	-

Mappa di Karnaugh

ab \ cd	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	-	-	-	-
10	1	1	-	-

Esempio 1

- Ignorando la presenza dei gradi di libertà introdotti dalle condizioni di indifferenza
  - L'utilizzo dei soli 1 porterebbe a identificare due implicanti primi essenziali

ab \ cd	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	-	-	-	-
10	1	1	-	-

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

Esempio 1

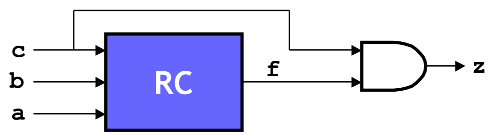
- Servendosi delle condizioni di indifferenza si migliora il risultato riducendo il costo della realizzazione
  - Assegnando valore 1 in corrispondenza di 1010 e 1011 e valore 0 in corrispondenza delle altre configurazioni

ab \ cd	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	-	-	-	-
10	1	1	-	-

$$f(a,b,c,d) = b'$$

Esempio 2

- Si voglia sintetizzare la rete RC di figura soggetta ai seguenti vincoli di progetto:
  - Il valore assunto da **a** è sempre uguale a quello di **b**
  - Il valore di **f** è **1**
    - Quando **a=0**, **b=0**
    - Quando **a=1**, **b=1**, **c=0**
  - Il valore di **f** è **0**
    - In tutti gli altri casi



Esempio 2

- Non facendo alcuna considerazione
  - Sul contesto in cui è inserito il circuito
  - Sul fatto che **a** deve essere uguale a **b**

a	b	c	f
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

c \ ab	00	01	11	10
0	1	0	1	0
1	1	0	0	0

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

Esempio 2

- Considerando il solo vincolo sugli ingressi
  - **a** è sempre uguale a **b**

a	b	c	f
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

→

a	b	c	f
0	0	0	1
0	0	1	1
0	1	0	-
0	1	1	-
1	0	0	-
1	0	1	-
1	1	0	1
1	1	1	0

c \ ab	00	01	11	10
0	1	-	1	-
1	1	-	0	-

f(a,b,c) = a′ + c′
f(a,b,c) = b′ + c′

Esempio 2

- Considerando i vincoli imposti sulle uscite
  - Dato che **z=cf**, quando **c=0** allora **z** non dipende da **f**

a	b	c	f	z	
0	0	0	1	0	**z** indipendente da **f**
0	0	1	1	1	
0	1	0	-	-	**z** indipendente da **f**
0	1	1	-	-	
1	0	0	-	-	**z** indipendente da **f**
1	0	1	-	-	
1	1	0	1	0	**z** indipendente da **f**
1	1	1	0	0	

a	b	c	f
0	0	0	-
0	0	1	1
0	1	0	-
0	1	1	-
1	0	0	-
1	0	1	-
1	1	0	-
1	1	1	0

c \ ab	00	01	11	10
0	-	-	-	-
1	1	-	0	-

f(a,b,c) = a′
f(a,b,c) = b′

Sintesi su mappe di funzioni non completamente specificate

- Rispetto al caso senza condizioni di indifferenza si hanno le seguenti variazioni nel metodo di sintesi:
  - Individuare gli implicanti primi e primi essenziali considerando le condizioni di indifferenza come se fossero tutte 1
    - Gli implicanti primi realizzati solamente mediante condizioni di indifferenza non hanno alcun valore
  - Coprire solo l'ONset della funzione con gli implicanti
    - Infatti, i soli termini significativi sono gli 1 della funzione
    - Questi termini sono gli unici elementi di rilievo (vincoli)

Mappe di Karnaugh per espressioni PS

- E' indispensabile sapere effettuare la sintesi anche per il caso PS (anche ai fini dell'esame)
- In questo caso si raggruppano assieme gli "0" e le eventuali condizioni di indifferenza utili ad allargare i raggruppamenti rettangolari
- Si utilizza una corrispondenza negata:
  - Un numero di 2^m "0" o "—" raccolti costituisce un raggruppamento rettangolare di ordine *m*
  - Esso produce un termine somma di *n*-*m* letterali dove *n* è il numero di variabili della funzione
- Il termine somma associato ad un implicato è ottenuto:
  - Identificando le variabili che non cambiano mai di valore
  - Riportando ogni variabile in forma diretta se essa assume il valore 0, in forma complementata se assume il valore 1