Sintesi di Reti Combinatorie

Ottimizzazione di Reti Combinatorie a Due Livelli: Metodo di Karnaugh

Introduzione

Metodo di Karnaugh per reti completamente specificate

Le condizioni di indifferenza

Metodo di Karnaugh per reti non completamente specificate

Versione del 22/09/03

Sintesi di reti combinatorie a due livelli: *Introduzione*

- Obiettivo:

Ridurre la complessità di una (o più) funzione(i) booleana(e) espressa(e) in forma di *Prodotto di Somme* o di *Somma di Prodotti* (SOP). Si considerano le forme canoniche come soluzioni iniziali

  - Ci si riferirà alla sola forma *Somma di Prodotti* o *SOP*
    - l'altra ne è la duale ed i principi sono gli stessi.
- Nella sintesi a due livelli gli obiettivi sono due:
  - Riduzione del numero dei termini prodotto (*principale*)
  - Riduzione del numero di letterali (*secondario*)
  - Esempio:
    - `f(a,d,c)=a'b'c'+a'bc'+a'b'c` equivale a `f(a,d,c)=a'b'+a'c'`

- Metodologie di sintesi ottima:
  - Esatte: *Karnaugh* e *Quine - Mc Cluskey;*
  - Euristiche per sintesi a due livelli.

Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Karnaugh
  - Si propone di identificare forme minime a due livelli applicando la regola di riduzione `a Z + a' Z = (a+a') Z = Z` con `Z` termine prodotto di n-1 variabili.
    - Esempio: `abcd' + ab'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`
      - Nota: si osservi che la applicazione della relazione identificata è applicata ad un numero di termini pari a 2^n quindi 2, 4, 8, ...
  - Osservazione: la regola identificata mantiene inalterato il numero dei livelli
    - Cioè, somme di prodotti rimangono tali. Al più, tali espressioni possono banalizzarsi in semplici prodotti o costanti.

Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Karnaugh
  - La formula di riduzione potrebbe essere facilmente applicata direttamente alle espressioni Booleane.

  - Il problema consiste nell'identificare:
    1. sia tutti i termini su cui applicare la riduzione;
       - Non è sempre immediato identificare tutti termini su cui applicare la regola di riduzione identificata.
    2. sia i tutti termini che partecipano a più riduzioni contemporaneamente e replicarli (vedi esempio).
       - Nota: si ricordi che, per le proprietà dell'algebra di Boole, la relazione `x+x=x` può essere applicata anche come `x=x+x`.

- Karnaugh (cont.)
  - Esempio di replicazione dei termini:

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

SOP: $f(a,b) = \underbrace{a'b + ab}_{1}\ldots$ $f(a,b)=a'b+\underbrace{ab}_{1,2}+ab'$

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

1: $=(a'+a)b+ab'=b+ab'$

2: $=a'b+a(b+b')=a'b+a$

  - Nessuna delle due espressioni è ulteriormente riducibile
  - È evidente, comunque, che la soluzione minima sia a+b

- Karnaugh (cont.)
  - Esempio di replicazione dei termini:

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

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

$$= \underbrace{a'b+ab}_{1}+\underbrace{\mathbf{ab}+ab'}_{2}$$

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

Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Karnaugh (cont.)
  - 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.

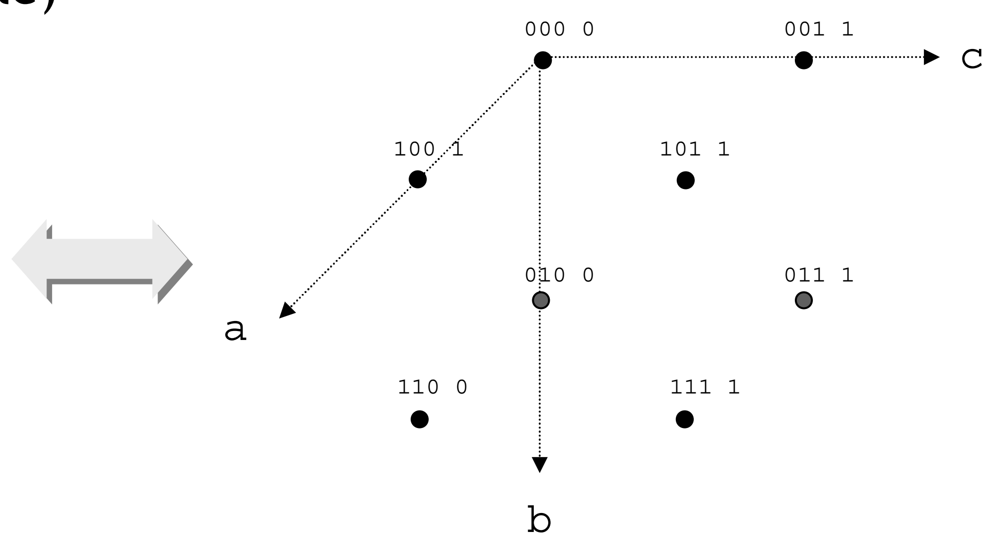
Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Mappe di Karnaugh
  - Una mappa di Karnaugh è uno schema deducibile dalla rappresentazione geometrica delle configurazioni binarie.
  - Definizione utili:
    - *Distanza di Hamming*: numero di bit che cambia nel passare da una configurazione binaria ad un'altra
      - Esempio: la *distanza di Hamming* tra le configurazioni `01001` e `10101` è 3 poiché cambiano 3 bit.
  - L'applicazione della regola di riduzione consiste nell'identificare le configurazioni binarie associate ai termini prodotto che sono a *distanza di Hamming unitaria*.
    - Esempio: i termini prodotto `abcd'` e `ab'cd'` corrispondono a `1110` e `1010` e sono a *distanza di Hamming* pari ad 1.

Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

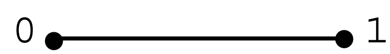
- È noto che una funzione di commutazione a `n` variabili `f: {0,1}`^{n} $\rightarrow$ `{0,1}` può essere rappresenta in modo comodo utilizzando una *tabella della funzione* o tabella della verità.
- In modo assolutamente equivalente una funzione a `n` variabili può essere associata ad una rappresentazione cartesiana in uno *spazio a n dimensioni*.
- Esempio (spazio 3 dimensionale)

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

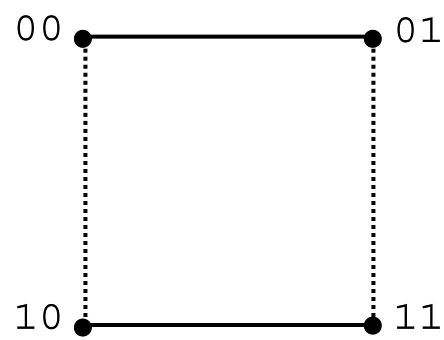


Sintesi di reti combinatorie a due livelli: *Metodi esatti - Karnaugh (cenni)*

- 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*, e l'`1-cubo` è un segmento: i due vertici sono associati alle configurazioni `0` e `1`

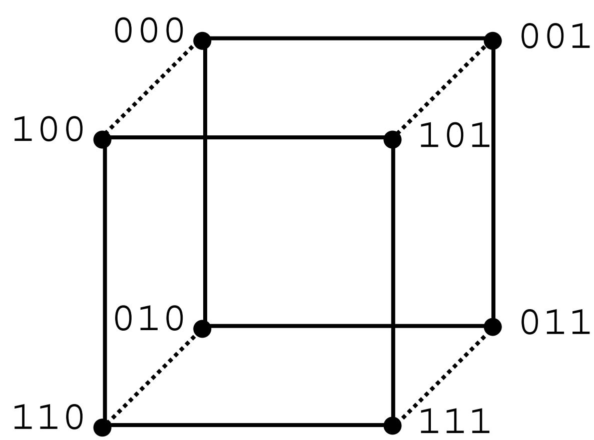


  - Spazio a 2 dimensioni (2 variabili):
    - È il piano, il `2-cubo` è un quadrato che si ottiene dall'`1-cubo` *per proiezione*. Si premette `0` alle configurazioni dei vertici originali, `1` a quelle dei vertici proiettati



Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Spazio a 3 dimensioni (3 variabili)
  - Il `3-cubo` è un solido, che si ottiene dal `2-cubo` *per proiezione*, premettendo `0` alle configurazioni dei vertici originali, `1` a quelle dei vertici proiettati

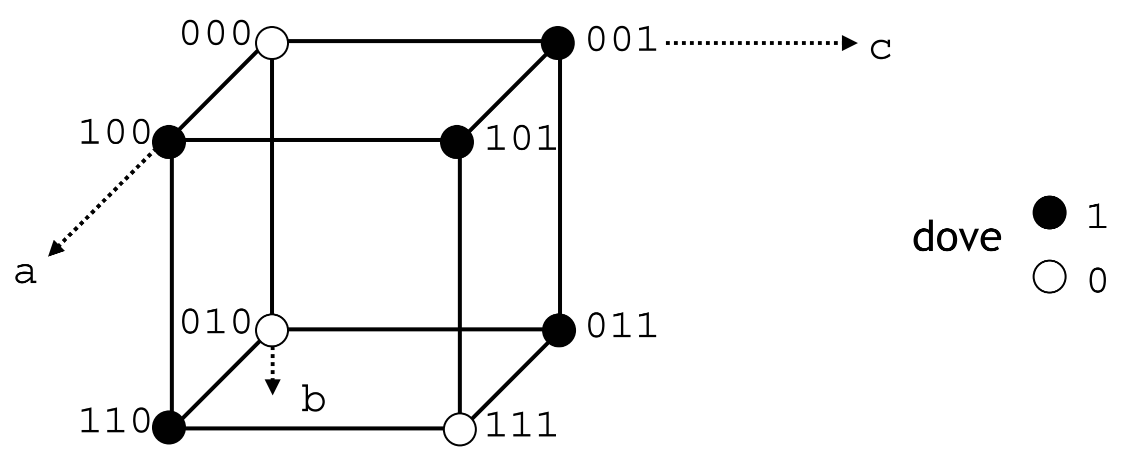


Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Si può pensare di trasportare una tabella delle verità a `n` variabili su un `n-cubo`, marcando opportunamente i nodi associati a `0` e `1`.
  - Si sottolinea nuovamente che due configurazioni sono a distanza unitaria (adiacenti) se e solo se i vertici associati sono collegati da un lato.

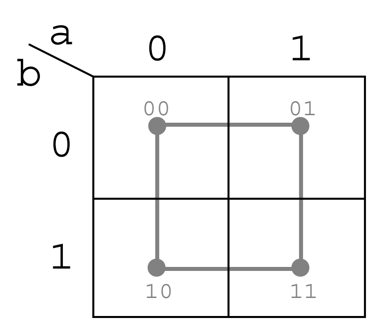
- Esempio: $f(a,b,c)=ON_{set}(1,3,4,5,6)$

ON_{set}=
insieme di 1 della funzione



Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Di fatto, la rappresentazione in uno spazio a n dimensioni non è maneggevole
  - Già per sole tre dimensioni non è di semplice utilizzo.
- Quindi, si passa allo *sviluppo nel piano* dei cubi.
- Al cubo sviluppato nel piano, che ha 2^n vertici, si sovrappone una griglia (*mappa*) con 2^n caselle organizzate secondo righe e colonne
  - Esempio: per il 2-cubo si ha una mappa di 4 caselle su due righe e due colonne, e ad ogni colonna si associa una delle variabili come coordinata



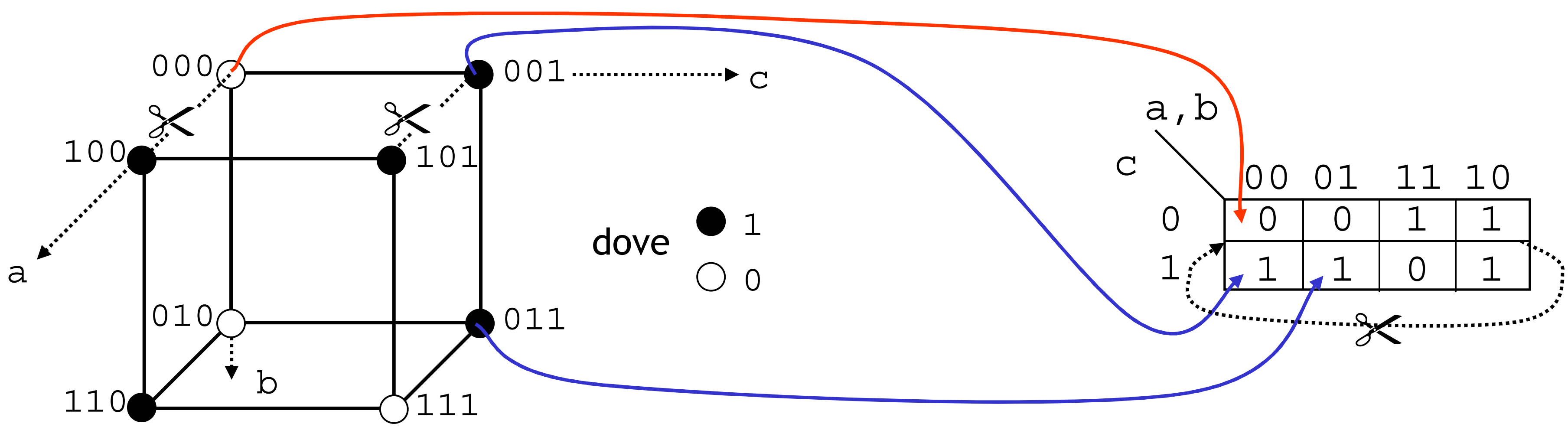
Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Una mappa così realizzata costituisce una *mappa di Karnaugh*:
  - Le configurazioni assunte dalla 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).
  - Esempio: $f(a,b)=ON_{set}(1,2)$

b \ a	0	1
0	0	1
1	1	0

Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

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



Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Caratteristiche delle mappe: riassunto

Indici di riga e colonna: configurazioni adiacenti

Cambia un solo bit nel passaggio da una configurazione ad un'altra

c,d \ a,b	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	0	0	1	1
10	1	0	0	1

Le colonne e righe identificate da `00` e `10` sono adiacenti

Contenuto della matrice: valori della assunti dalla funzione

Esempio: `f(a,b,c,d)` per `a=0` `b=0` `c=0` e `d=0` assume valore `1`

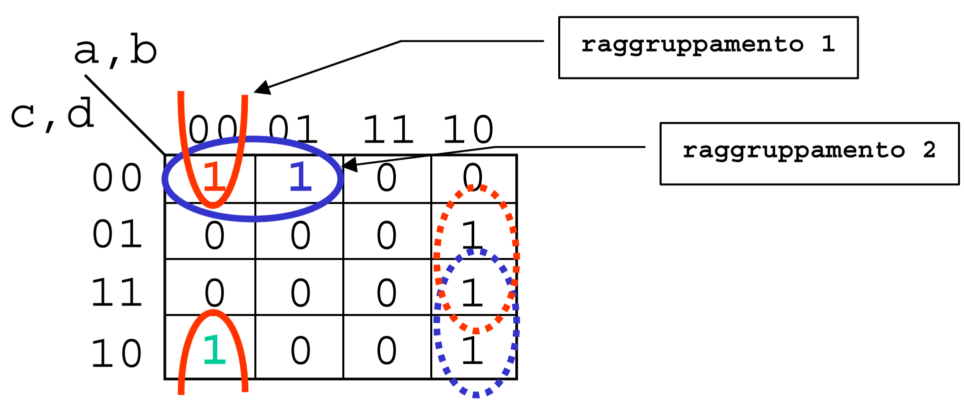
- Caratteristiche delle mappe
  - Si ricorda che: un implicante è un termine prodotto in cui compaiono solo alcuni dei letterali.

F(a,b,c,d)=**a’b’c’d’**+**a’b’cd’**+**a’bc’d’**+ab’c’d+ ab’cd+ab’cd’

raggruppamento 1

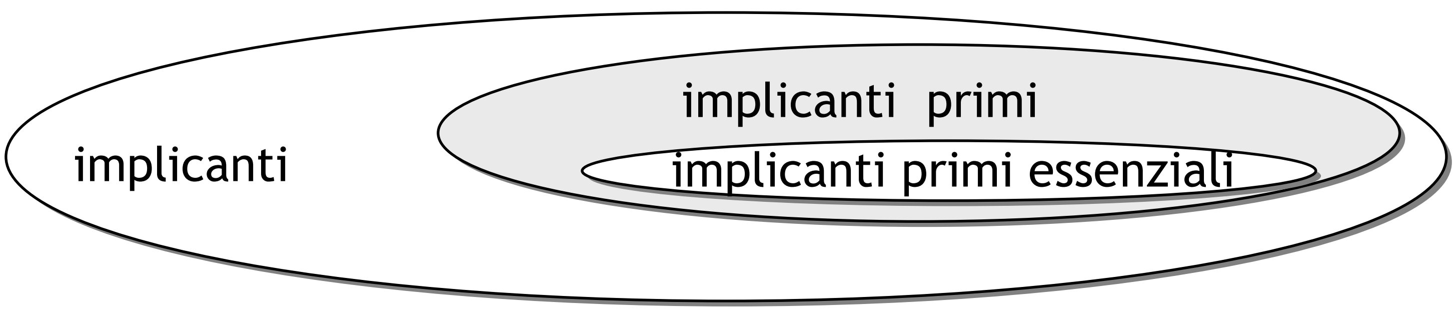
raggruppamento2

$$F(a,b,c,d)=\underbrace{\mathbf{a'b'd'}}_{\textbf{Implicante 1}}+\overbrace{\mathbf{a'c'd'}}^{\textbf{Implicante 2}}+...$$



Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Metodo:
  1. Individuare gli implicanti primi e primi essenziali;
     - Implicante primo
       - Termine prodotto associato ad un *raggruppamento* di dimensione massima.
     - implicante primo essenziale
       - Implicante primo che copre uno o più 1 non coperti da nessun altro implicante primo.
  2. Copertura:
     - Scelta del minor numero di implicanti primi e primi essenziali



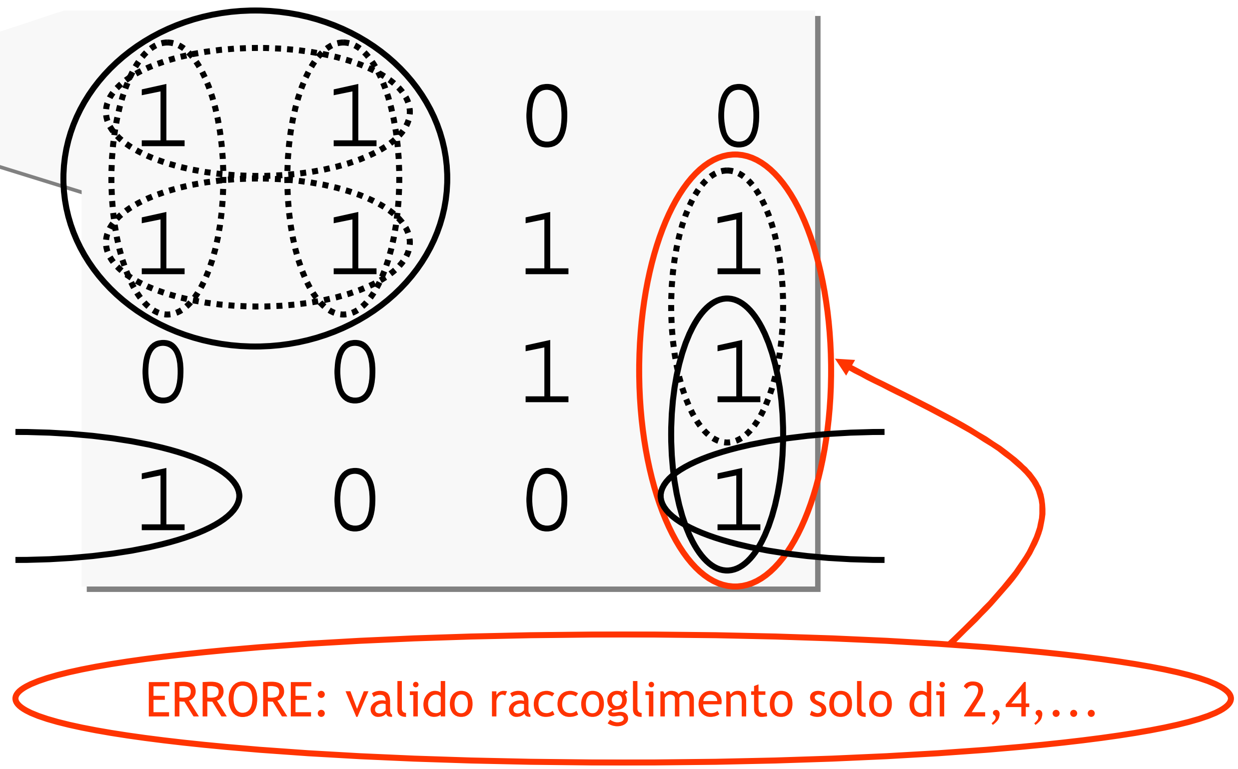
Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Scopo:
  - identificare una forma SoP che includa *il numero minimo di implicanti* e – a parità di numero di prodotti – gli *implicanti* col minimo numero di letterali (definita come *forma minima*) garantendo *la copertura di tutti gli 1 della funzione*.
- Teorema:
  - Esiste sicuramente una forma minima costituita *da soli implicanti primi*
    - sulla mappa di Karnaugh si identificano tutti gli implicanti primi.
      - Nota: la somma di *tutti* gli implicanti primi è spesso *ridondante*.
  - *Implicanti primi essenziali* **devono** essere inclusi nella forma minima.
  - Una forma minima costituita da soli *implicanti primi essenziali* è unica
    - Condizione sufficiente.

Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Esempio: alcuni raccoglimenti

c,d \ a,b	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	0	0	1	1
10	1	0	0	1

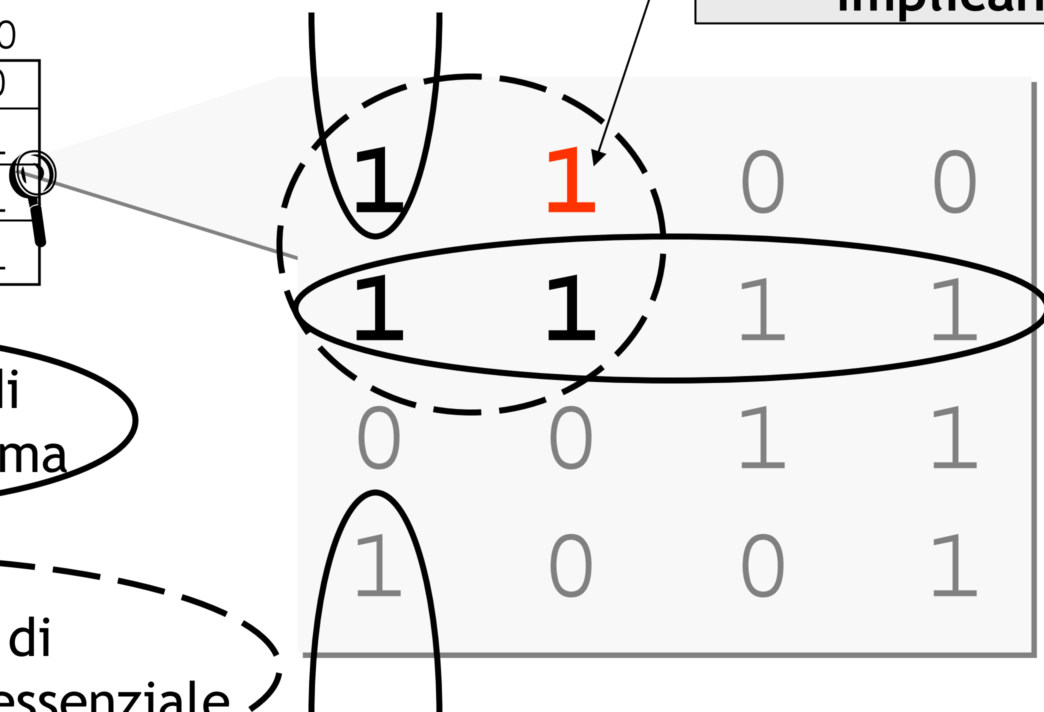


raccoglimento

Raccoglimento di dimensione massima

- Esempio: alcuni raccoglimenti

c,d \ a,b	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	0	0	1	1
10	1	0	0	1



1 appartenente ad un solo implicante primo

Raccoglimento di dimensione massima

Raccoglimento di dimensione massima essenziale

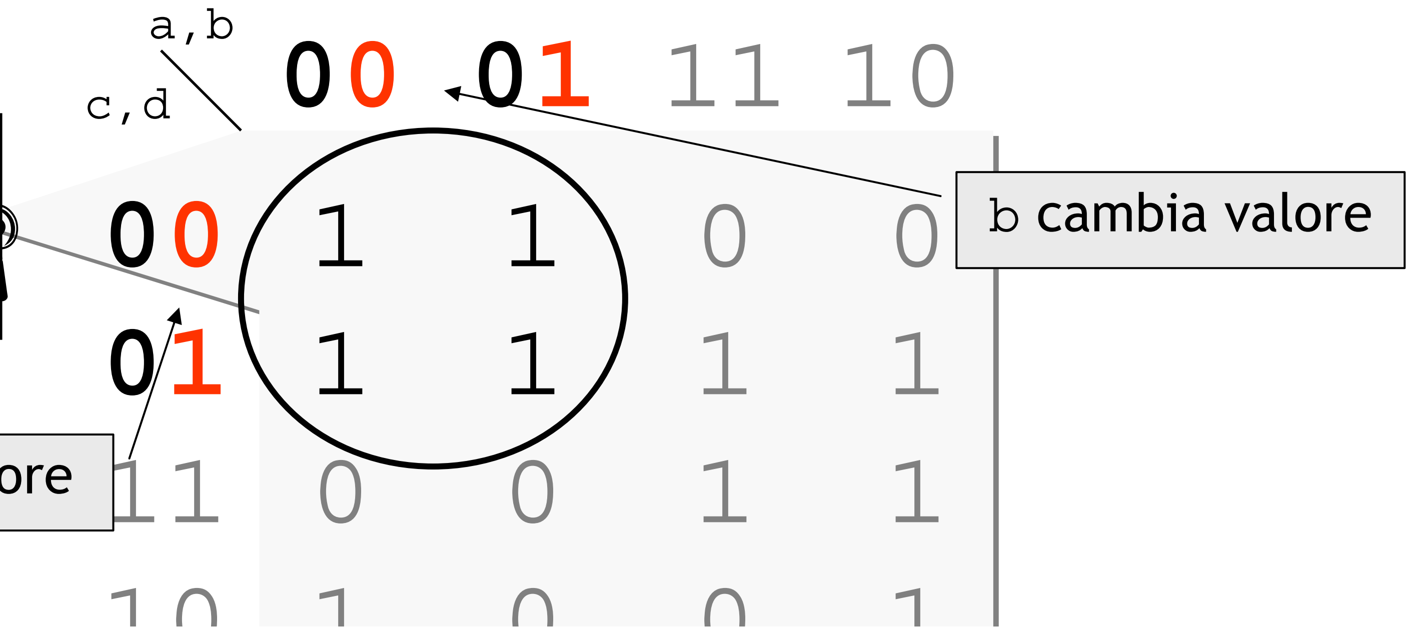
Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Ad ogni raccoglimento è associato un termine prodotto.
- Il termine prodotto (**implicante**) è ottenuto:
  - identificando le variabili che non cambiano mai di valore e riportando ogni variabile in modo *naturale* (esempio: a) se il valore che essa assume è 1 o in modo *complementato* (esempio: a’) se il valore da essa assunto è 0
- Osservazione:
  - un numero di 1 raccolti pari a 2^n produce un implicante di N-n letterali dove N è il numero delle variabili della funzione.
    - Esempio: per una funzione di quattro variabili –es. f(a,b,c,d) - un implicante che raccoglie quattro 1 è associato ad un termine prodotto di 2 variabili (es. a’d)

Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Esempio: identificazione del termine prodotto

c,d \ a,b	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	0	0	1	1
10	1	0	0	1



d cambia valore

b e d cambiano valore: non compaiono nel termine prodotto.
a e c compaiono come 0 quindi a′ e c′ .
Il termine prodotto è a′c′ .

Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Copertura
  - Copertura: sotto insieme 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; questo si traduce nella identificazione della copertura di *minima cardinalità*:
    - sotto insieme degli implicanti primi e primi ed essenziali identificati che realizza una copertura della funzione che è di cardinalità minima.

- Copertura (cont.)
  - Scelta degli implicanti per realizzare la copertura:
    1. Si scelgono tutti gli implicanti primi essenziali.
       - Gli implicanti primi essenziali devono essere parte della copertura poiché “sono essenziali” e, quindi, non è possibile fare a meno di loro.
    2. Si eliminano tutti gli implicanti primi che sono coperti da quelli essenziali (eliminazione implicanti completamente ridondanti)
       - gli implicanti eliminati, detti *completamente ridondanti*, coprono degli 1 che sono già ricoperti da quelli essenziali e, quindi, non servono ed aumentano il costo.
    3. Si seleziona il numero minore degli implicanti primi che sono rimasti.
       - gli implicanti residui sono detti *parzialmente ridondanti*.
  - Osservazione: la scelta viene fatta seguendo un criterio basato sulla pura osservazione della tabella.

Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Esempio:

$$f(a, b, c, d) = \sum (0,1,2,4,5,9,10,11,13,15)$$

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

Implicanti primi essenziali

a'c';   ad

c,d \ a,b	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	0	0	1	1
10	1	0	0	1

Implicanti primi

a'b'd'; b'cd'; ab'c; c'd

Completamente ridondante

- Esempio (cont.):

Tabella ottenuta dopo la selezione degli implicanti primi essenziali

c,d \ a,b	00	01	11	10
00			0	0
01				
11	0	0		
10	**1**	0	0	**1**

1 da coprire

Implicanti primi essenziali

a'c'; ad

Implicanti primi

a'b'd'; **b'cd'**; ab'c; ~~c'd~~

Parzialmente ridondanti

f(a,b,c,d)= a'c'+ad+b'cd'

Forma minima (unica)

Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Esempio:

$$f(a, b, c, d) = \sum (0,2,4,5,10,11,13,15)$$

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

c,d \ a,b	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 essenziali

Nessuno

Implicanti primi

a'c'd'; bc'd; acd; b'cd';
a'b'd'; a'bc'; abd; ab'c

f(a,b,c,d)= a'c'd'+ bc'd+ acd+ b'cd'
f(a,b,c,d)= a'b'd'+ a'bc'+ abd+ ab'c

Due forme minime

Condizioni di Indifferenza (*don’t care*)

- La specifica di un progetto (la descrizione di quello che si vuole progettare) contiene, spesso, delle *condizioni di indifferenza* (denominate anche *don't care* o *DC*).
- 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;

  e/o
  - Le configurazioni di ingresso impediscono all’uscita della rete - in fase di progetto - di essere osservata.

Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Le configurazioni di ingresso per le quali il valore dell'uscita è non specificato vengono definite *condizioni di indifferenza* e costituiscono il $\mathbf{DC}_{\mathbf{set}}$ della funzione stessa.
- Sulla tabella delle verità (o in una mappa di Karnaugh) il valore non specificato della funzione si indica il simbolo "`-`" (o anche "`x`").
- Le condizioni di indifferenza sono gradi di libertà nel processo di sintesi.
  - In fase di sintesi, ai valori non specificati si può assegnare indifferentemente il valore `0` oppure `1` a seconda di quanto conviene per minimizzare la funzione.
  - Una condizione di indifferenza **non** deve necessariamente essere *coperta* da un implicante (forma SoP), ma **può** esserlo se questo conviene cioè se consente di:
    1. o di ridurre il numero degli implicanti;
    2. o di ridurre il numero dei letterali degli implicanti esistenti.

Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Importante:

1. Gli implicanti primi realizzati solamente mediante condizioni di indifferenza non hanno alcuno scopo (non servono).
2. Un implicante primo non diventa essenziale perché è l'unico a coprire una data condizione di indifferenza.

a,b \ c,d

c,d \ a,b	00	01	11	10
00	-	-	0	0
01	-	-	0	0
11	0	0	1	1
10	0	0	0	1

~~a'c'~~

Implicanti primi essenziali

Implicanti primi

NON Implicanti primi

c,d \ a,b	00	01	11	10
00	1	-	0	0
01	-	-	-	-
11	0	0	1	1
10	1	0	0	1

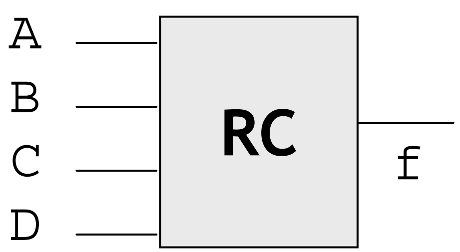
Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh: Esempio 1

- Esempio 1:
  - Si voglia sintetizzare una funzione con quattro ingressi `A,B,C,D` e un'uscita `f`. Gli ingressi rappresentano *cifre decimali codificate in codice BCD*;
  1. l'uscita deve valere `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*.
  - Nota: 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*.

Sintesi di reti combinatorie a due livelli: *Metodi esatti - Karnaugh*: *esempio 1*

- Esempio 1 (cont.)



A	B	C	D	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
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	-

a,b \ c,d	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	-	-	-	-
10	1	1	-	-

- Esempio 1 (cont.)
  - Ignorando la presenza dei gradi di libertà introdotti dalle condizioni di indifferenza, l'utilizzo dei soli `1` porterebbe a identificare due implicanti essenziali.

a,b \ c,d	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'+ c'b'

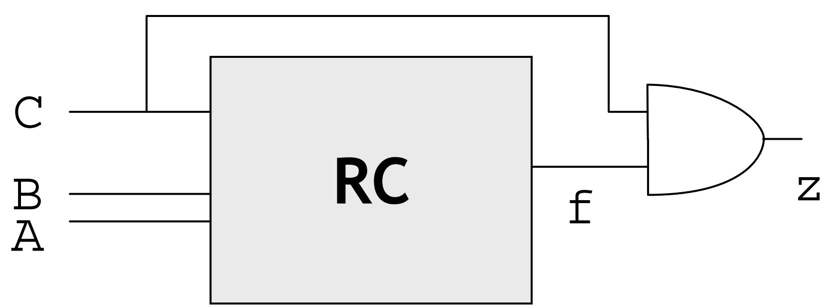
Sintesi di reti combinatorie a due livelli: *Metodi esatti - Karnaugh*: *esempio 1*

- Esempio 1 (cont.)
  - Servendosi delle condizioni di indifferenza si migliora il risulato riducendo il costo della realizzazione.
    - assegnando valore 1 in corrispondenza di 1010 e 1011 e valore 0 in corrispondenza delle altre configurazioni.

a,b \ c,d	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	-	-	-	-
10	1	1	-	-

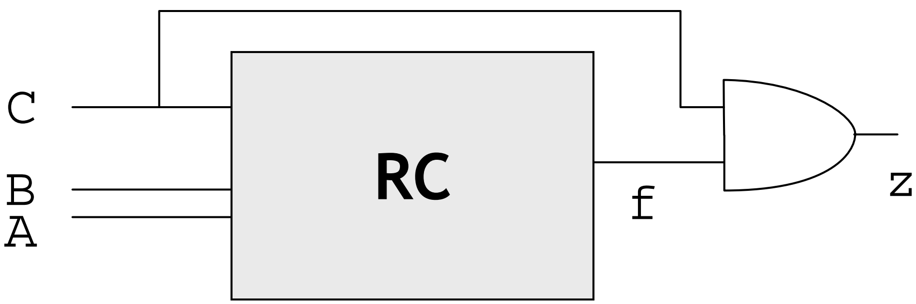
f(a,b,c,d)= b'

- Esempio 2:
  - Si voglia sintetizzare la rete RC di fugura soggetta ai seguenti vincoli di progetto:
    1. il valore assunto da A è sempre uguale a quello di B.
    2. Quando A=0;B=0 e quando A=1;B=1;C=0 il valore di f è 1 mentre, in tutti gli altri casi, f vale 0.
  - Problema: Qual è la funzione associata alla rete combinatoria RC (f=g(a,b,c))?



Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh: esempio 2

- Esempio 2 (cont.):
    - si consideri il seguente esempio dove il valore di A è sempre uguale al valore di B mentre C può assumere qualunque valore. Quando A=B=0 e quando A=B=1 e C=0 il valore di f è 1 mentre, in tutti gli altri casi, f vale 0. Qual è la funzione associata alla rete combinatoria RC (f=g(a,b,c))?
  - Se non facessimo alcuna considerazione né sul fatto che `A` deve essere uguale a `B` né sul contesto in cui è inserito il circuito si avrebbe:



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'

Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh: esempio 2

- Esempio 2 (cont.):
  - si consideri il seguente esempio dove il valore di A è sempre uguale al valore di B mentre C può assumere qualunque valore. Quando A=B=0 e quando A=B=1 e C=0 il valore di f è 1 mentre, in tutti gli altri casi, f vale 0. Qual è la funzione associata alla rete combinatoria RC (f=g(a,b,c))?
  - Considerando il solo vincolo sugli ingressi, espresso da "*A è sempre uguale a B*", si avrebbe:

A	B	C	f
0	0	0	
0	0	1	
0	1	0	A diverso da B
0	1	1	A diverso da B
1	0	0	A diverso da B
1	0	1	A diverso da B
1	1	0	
1	1	1	

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	-

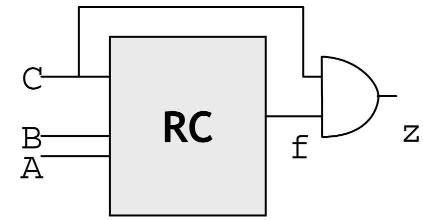
f(a,b,c)= a'+c'

oppure

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

Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh: esempio 2

- Esempio 2 (cont.):
  - Considerando i vincoli imposti sia sugli ingressi sia sulle uscite :



Configurazioni mai prodotte dall'ambiente

A	B	C	f
0	0	0	
0	0	1	
0	1	0	A diverso da B
0	1	1	A diverso da B
1	0	0	A diverso da B
1	0	1	A diverso da B
1	1	0	
1	1	1	

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

Configurazioni mai osservate da Z

A	B	C	f
0	0	0	Z indipendente da f
0	0	1	
0	1	0	Z indipendente da f
0	1	1	
1	0	0	Z indipendente da f
1	0	1	
1	1	0	Z indipendente da f
1	1	1	

Il simbolo "-" indica che il valore assunto dalla uscita non ha alcuna importanza poiché:

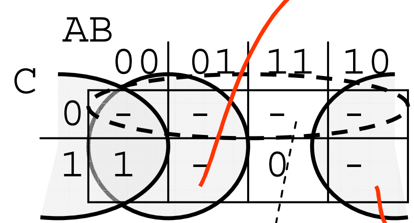
a) La configurazione degli ingressi ad esso relativa non viene mai generata

b) L'uscita corrispondente alla configurazione degli ingressi non viene mai osservata

Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Metodo (riassunto):
  - Rispetto al caso senza condizioni di indifferenza si hanno seguenti variazioni:
  1. Individuare gli implicanti primi e primi essenziali considerando le condizioni di indifferenza come se fossero 1;
     1. Nota: Si ricordi che gli implicanti primi realizzati solamente mediante condizioni di indifferenza non hanno alcun valore.
  2. Coprire solo l'ON_{set} della funzione con gli implicanti identificati.
     - Infatti, i soli termini significativi sono gli 1 della funzione. Questi termini sono gli unici elementi di rilievo (vincoli).
- Esempio:

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



f(a,b,c)= a'

oppure

f(a,b,c)= b'

Trascurato in fase di copertura o
non sviluppato in fase di espansione

Sintesi di reti combinatorie a due livelli:
Metodi esatti - Karnaugh (cenni)

- Esempio

$$f(a,b,c,d) = ON(1,11,12,13,14,15)DC(3,4,5,9)$$

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

Completamente ridondante

c,d \ a,b	00	01	11	10
00	0	-	1	0
01	1	-	1	-
11	-	0	1	1
10	0	0	1	0

Implicanti primi essenziali

ab

Implicanti primi

b'd ; c'd ; ~~bc'~~; ad

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

Forma minima (unica)