

Architettura degli Elaboratori

Sintesi di reti logiche e minimizzazione



Barbara Masucci

UNIVERSITA' DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INFORMATICA

DIPARTIMENTO DI ECCELLENZA

Punto della situazione

- Stiamo studiando le **reti logiche**
 - Costruite a partire dalle porte logiche **AND, OR, NOT**
 - Ogni rete logica realizza una **espressione booleana**
- Obiettivo di oggi: **sintesi di reti logiche e minimizzazione**
 - Passaggio da una espressione booleana a una rete logica con costo minimale



Operatori logici

- Funzione logica o di commutazione:
 - Una funzione di variabili binarie e a valore binario
- Un risultato importante:
 - Ogni funzione logica può essere espressa in termini soltanto di **AND, OR, NOT**
 - Esistono però anche altri operatori logici importanti: **XOR, NAND, NOR**



XOR o OR esclusivo: somma modulo 2

- $XOR(x,y)$ con x, y variabili che possono assumere valore Vero o Falso dà come risultato
 - Vero se una variabile è posta a Vero, ma non entrambe,
 - Falso, altrimenti
- Interpretando Vero come 1 e Falso come 0
 $XOR(x,y)$ corrisponde a $x \cdot \bar{y} + \bar{x} \cdot y = x \oplus y$

$$XOR(F,F) = F$$

$$XOR(F,V) = V$$

$$XOR(V,F) = V$$

$$XOR(V,V) = F$$

$$0 \oplus 0 = 0$$

$$0 \oplus 1 = 1$$

$$1 \oplus 0 = 1$$

$$1 \oplus 1 = 0$$

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Tavola di verità



NAND: negazione dell'AND

- $NAND(x,y)$, con x, y variabili che possono assumere valore Vero o Falso, dà come risultato
 - Falso se entrambe le variabili sono poste a Vero
 - Vero, altrimenti
- Interpretando Vero come 1 e Falso come 0, $NAND(x,y)$ corrisponde alla negazione di $AND(x,y)$

$$NAND(F,F) = V$$

$$NAND(F,V) = V$$

$$NAND(V,F) = V$$

$$NAND(V,V) = F$$

x	y	$\overline{x \cdot y}$
0	0	1
0	1	1
1	0	1
1	1	0

Tavola di verità



NOR: negazione dell'or

- $NOR(x,y)$, con x, y variabili che possono assumere valore Vero o Falso, dà come risultato
 - Vero se entrambe le variabili sono poste a Falso
 - Falso, altrimenti
- Interpretando Vero come 1 e Falso come 0
 $NOR(x,y)$ corrisponde alla negazione di $OR(x,y)$

$$NOR(F,F) = V$$

$$NOR(F,V) = F$$

$$NOR(V,F) = F$$

$$NOR(V,V) = F$$

x	y	$\overline{x + y}$
0	0	1
0	1	0
1	0	0
1	1	0

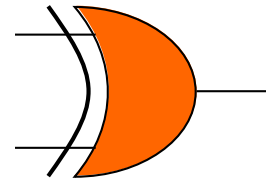
Tavola di verità



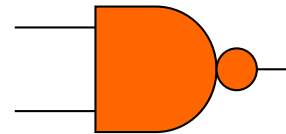
Porte logiche

rappresentazione grafica

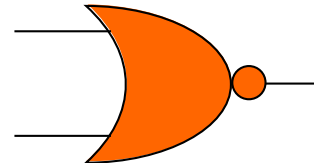
➤ Porta XOR



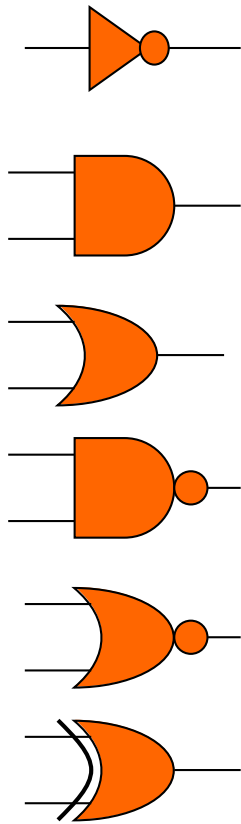
➤ Porta NAND



➤ Porta NOR



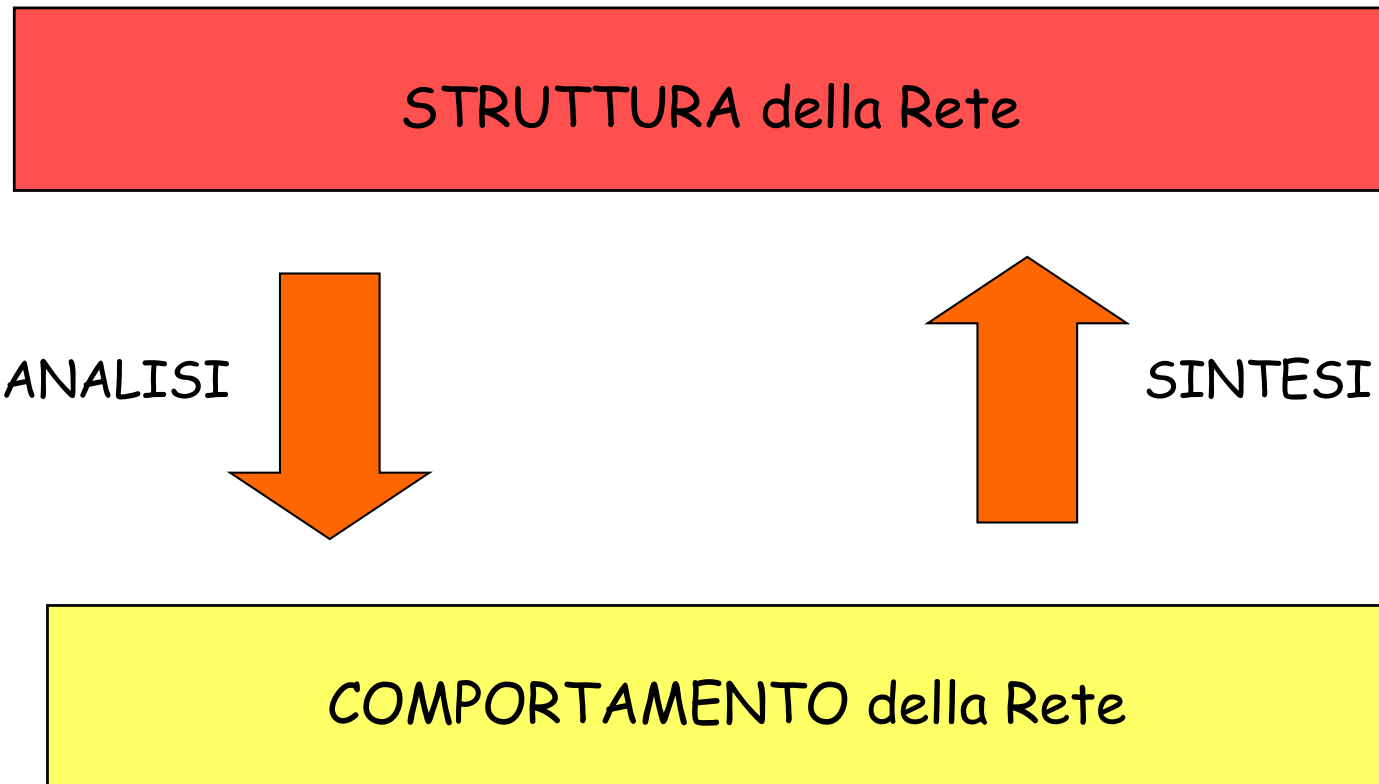
Porte logiche



Operatore	Simbolo	Proprietà
NOT	$y = \bar{x}$	$y=1$ se e solo se $x=0$
AND	$y = x_1x_2$	$y=1$ se e solo se $x_1=x_2=1$
OR	$y = x_1+x_2$	$y=0$ se e solo se $x_1=x_2=0$
NAND	$y = x_1/x_2$	$y=0$ se e solo se $x_1=x_2 = 1$
NOR	$y = x \downarrow x_2$	$y=1$ se e solo se $x_1=x_2 = 0$
XOR	$y = x_1 \oplus x_2$	$y=1$ se e solo se $x_1 \neq x_2$



Analisi e sintesi di reti logiche



Processo di sintesi

- Per ogni funzione logica possiamo costruire una **rete logica che la realizza**
- In realtà, molte reti logiche, non tutte equivalenti in termini di prestazioni
- Il processo di costruzione della rete logica si chiama **sintesi**
- A partire da un tipo particolare di espressione booleana, detta "**forma canonica SOP**", per descrivere la funzione, si ottiene una rete che la realizza
- Il **processo di minimizzazione** consente di ottenere la rete logica con prestazioni migliori



Espressioni booleane

- **Variabile booleana:** una variabile che può assumere solo i valori **1 (Vero)** e **0 (Falso)**
- **Letterale:** una variabile o la sua negazione
- **Operatori booleani:** gli operatori AND, OR, NOT
- **Espressione booleana:** una combinazione di variabili booleane e operatori booleani
 - Un'espressione booleana denota una funzione logica



Espressioni SOP

- Un'espressione booleana è in **forma normale SOP** (Sum Of Products) quando è l'OR di AND di letterali

$$\overline{x_1} \overline{x_2} x_3 + \overline{x_1} x_3 + x_1 x_3$$

- **Mintermine**: prodotto di letterali in cui compare **ogni** variabile o vera o negata
- Una espressione normale SOP è in **forma canonica SOP** se i suoi termini sono tutti mintermini

$$\overline{x_1} \overline{x_2} x_3 + \overline{x_1} x_2 \overline{x_3} + x_1 x_2 x_3$$



Come ottenere una forma normale SOP

- Una espressione booleana può essere ridotta a una **forma normale SOP** mediante le seguenti operazioni:
 - Applicare la proprietà distributiva, sciogliendo tutte le parentesi
 - Eliminare i termini ridondanti, usando le proprietà di idempotenza e complementarietà



Esempio

$$f(x_1, x_2, x_3) = [(x_1x_2 + x_1x_3)(x_1x_2x_3 + \overline{x_2})]x_1 + (x_2x_3 + x_1x_2)\overline{x_3}$$

Non è in forma normale SOP

$$\begin{aligned} f(x_1, x_2, x_3) &= (x_1x_2x_1x_2x_3 + x_1x_2\overline{x_2} + x_1x_3x_1x_2x_3 + x_1\overline{x_2}x_3)x_1 + x_2x_3\overline{x_3} + x_1x_2\overline{x_3} \\ &= (x_1x_2x_3 + x_1x_2x_3 + x_1\overline{x_2}x_3)x_1 + x_1x_2\overline{x_3} \\ &= x_1x_2x_3 + x_1x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2\overline{x_3} \\ &= x_1x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2\overline{x_3} \end{aligned}$$

È in forma normale (anche canonica) SOP!



Da forma normale a forma canonica SOP

- Una espressione booleana in forma normale SOP può essere ridotta a una **forma canonica SOP** mediante le seguenti operazioni:
 - Se un termine prodotto non contiene il letterale x_i , moltiplicare per $(x_i + \bar{x}_i)$
 - Applicare la proprietà distributiva, sciogliendo tutte le parentesi
 - Eliminare i termini ridondanti, usando le proprietà di idempotenza e complementarietà



Esempio

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

Non è in forma canonica SOP

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

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

$$= \overline{x_1} \overline{x_2} x_3 + x_1 \overline{x_2} \overline{x_3} + x_1 \overline{x_2} x_3 + x_1 x_2 \overline{x_3} + x_1 x_2 \overline{x_3} + x_1 x_2 x_3 + x_1 x_2 x_3$$

$$= \overline{x_1} \overline{x_2} x_3 + x_1 \overline{x_2} \overline{x_3} + x_1 \overline{x_2} x_3 + x_1 x_2 x_3$$

È in forma canonica SOP!



Processo di sintesi

- Supponiamo di avere una funzione logica f
- Come risalire ad una rete che la realizza?
- Seguiamo una serie di passi
 - Da f alla tavola di verità
 - Dalla tavola di verità all'espressione in forma canonica SOP
 - Dall'espressione in forma canonica SOP ad una rete a due livelli
 - Il primo di porte AND
 - Il secondo con una sola porta OR



Mintermini

Se la funzione logica è un **mintermine**,
la sua tavola di verità **ha un solo 1**

Esempio:

se $f = \overline{x}_3 x_2 x_1$ la tavola di verità ha un solo
1 in corrispondenza di $x_3=0, x_2=1, x_1=1$

Ricorda:

Il prodotto (AND) è 1 sse ogni fattore è 1

Viceversa, se la tavola di verità
di f **ha un solo 1** necessariamente
 f è un **mintermine**

x_3	x_2	x_1	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

$$f = \overline{x}_3 x_2 x_1$$



Dalla tavola di verità all'espressione SOP

Se invece la tavola di verità ha più
occorrenze di 1, troviamo i mintermini
corrispondenti e li sommiamo

x_3	x_2	x_1	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

$$\bar{x}_3 x_2 x_1$$

$$x_3 \bar{x}_2 x_1$$

f ha valore 1 se:

$$x_3=0, x_2=1, x_1=1$$

oppure

$$x_3=1, x_2=0, x_1=1$$

Espressione canonica SOP

$$f = \bar{x}_3 x_2 x_1 + x_3 \bar{x}_2 x_1$$



Esempio 1

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

$$\bar{x}_1 \bar{x}_2 x_3$$

$$\bar{x}_1 x_2 x_3$$

$$x_1 \bar{x}_2 \bar{x}_3$$

$$x_1 x_2 \bar{x}_3$$

Espressione canonica SOP

$$f(x_1, x_2, x_3) = \bar{x}_1 \bar{x}_2 x_3 + \bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 \bar{x}_3 + x_1 x_2 \bar{x}_3$$



Esempio 2

x_1	x_2	x_3	f_1
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$$\bar{x}_1 \bar{x}_2 \bar{x}_3$$

$$\bar{x}_1 \bar{x}_2 x_3$$

$$\bar{x}_1 x_2 x_3$$

$$x_1 x_2 x_3$$

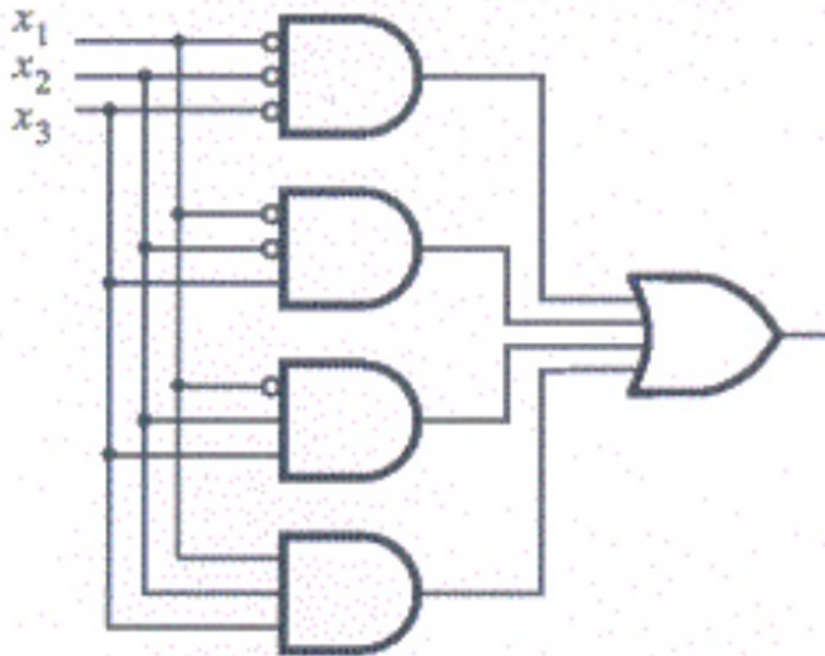
Espressione canonica SOP

$$f(x_1, x_2, x_3) = \bar{x}_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 \bar{x}_2 x_3 + \bar{x}_1 x_2 x_3 + x_1 x_2 x_3$$



Dall'espressione SOP a una rete a due livelli

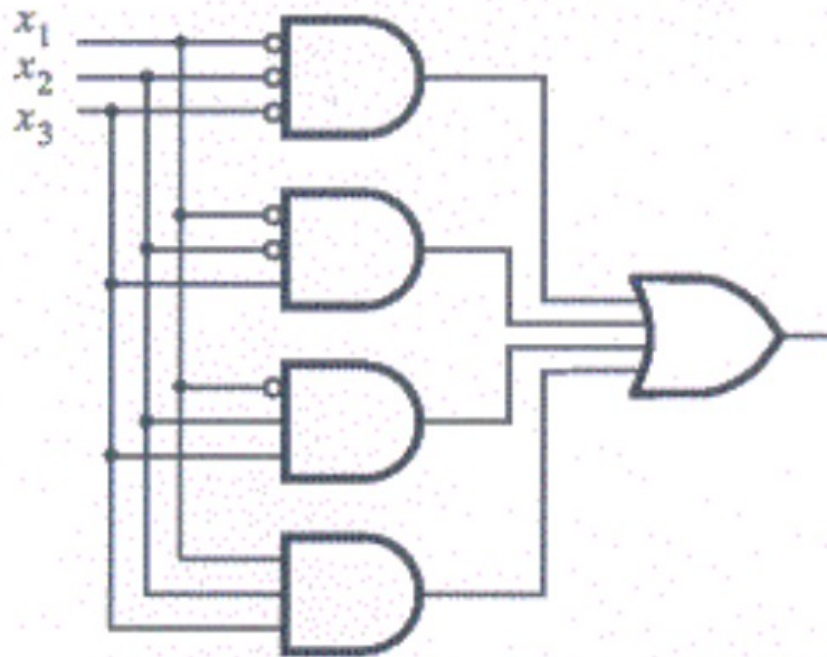
- Nel primo livello **varie porte AND**
 - Tante, quanti sono i mintermini
- Nel secondo livello, **solo una porta OR**



$$f_1 = \bar{x}_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 \bar{x}_2 x_3 + \bar{x}_1 x_2 x_3 + x_1 x_2 x_3$$

Processo di sintesi

Possiamo realizzare la funzione f con un numero minore di porte?



$$f_1 = \bar{x}_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 \bar{x}_2 x_3 + \bar{x}_1 x_2 x_3 + x_1 x_2 x_3$$

Minimizzazione di funzioni

- Data una funzione logica, **esistono molte reti diverse** che la realizzano
 - Non tutte equivalenti in termini di **prestazioni**
- Dalla tavola di verità si ottiene una rete a due livelli
 - Non necessariamente quella di costo minimo
- Il **processo di minimizzazione** consente di ottenere la rete logica con prestazioni migliori



Valutazione delle prestazioni di una rete

➤ Le prestazioni di una rete logica vengono misurate in termini di

➤ Velocità

➤ Costo

Esiste una tecnica che ci permette di sintetizzare la rete in modo da massimizzare le prestazioni?



Velocità di una rete

- Misurata calcolando il **tempo che passa** da quando i segnali sono disponibili sui terminali input a quando un segnale è disponibile sul terminale di output
- Dipende dal **massimo numero di porte** logiche che un segnale deve attraversare da input ad output
 - Quando un segnale attraversa una porta logica subisce un piccolo ritardo
- Faremo riferimento a **reti a due livelli (AND-to-OR)**
 - Sono le più veloci tra quelle che possono realizzare qualsiasi funzione



Costo di una rete

- Il **costo della rete** dipende da
 - Numero di porte logiche
 - Numero di linee di input
- Tra tutte le reti AND-to-OR che realizzano una funzione logica f , una rete è **minimale** se
 - Ha il **minor numero di porte AND**
 - Tra tutte quelle che hanno il minor numero di porte AND ha il **minor numero di linee di input** a quelle porte



Costo di una espressione logica

- Un'espressione in forma normale SOP è **minimale** se
 - Ha il **minor numero di termini prodotto** possibile
 - Tra tutte le espressioni equivalenti che hanno il minor numero di prodotti ha il **minor numero di letterali**



Minimizzazione di espressioni booleane

- La minimizzazione di espressioni booleane non è un processo semplice e diretto
 - Si utilizzano le proprietà dell'algebra booleana
- Se le variabili sono poche, è possibile utilizzare la Mappa di Karnaugh
 - Una rappresentazione particolare delle tavole di verità che consente di semplificare il processo
 - Utile fino a 4 variabili
 - Non la vedremo (argomento presente nel programma del corso corrispondente a 10 CFU)



Minimizzazione di espressioni SOP

- Data una espressione in forma canonica SOP, due **mintermini** sono detti **adiacenti** se differiscono su una sola variabile

$$x_4 \bar{x}_3 \bar{x}_2 \bar{x}_1$$

$$x_4 \bar{x}_3 \bar{x}_2 x_1$$

- Un'espressione che contiene due mintermini adiacenti **può essere minimizzata** nel seguente modo

$$x_4 \bar{x}_3 \bar{x}_2 \bar{x}_1 + x_4 \bar{x}_3 \bar{x}_2 x_1 = x_4 \bar{x}_3 \bar{x}_2 (x_1 + \bar{x}_1) = x_4 \bar{x}_3 \bar{x}_2$$

- Da 2 prodotti e 8 occorrenze di letterali siamo passati a 1 prodotto e 3 letterali



Esempio (maggioranza)

- Sia $f(x_3, x_2, x_1)$ la funzione che vale 1 se (e solo se) la **maggioranza** delle variabili vale 1
- Dalla tavola di verità si ha:

x_3	x_2	x_1	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$f(x_3, x_2, x_1) = \overline{x_3}x_2x_1 + x_3\overline{x_2}x_1 + x_3x_2\overline{x_1} + x_3x_2x_1$$

Da cui si ottiene una rete con
4 porte AND a 3 ingressi ciascuna
e 1 porta OR a 4 ingressi



Esempio (maggioranza)

Applicando la proprietà di idempotenza e del complemento:

$$\begin{aligned} f(x_3, x_2, x_1) &= \overline{x_3}x_2x_1 + x_3\overline{x_2}x_1 + x_3x_2\overline{x_1} + x_3x_2x_1 \\ &= \overline{x_3}x_2x_1 + x_3x_2x_1 + x_3\overline{x_2}x_1 + x_3x_2x_1 + x_3x_2\overline{x_1} + x_3x_2x_1 \\ &= (\overline{x_3} + x_3)x_2x_1 + (\overline{x_2} + x_2)x_3x_1 + x_3x_2(\overline{x_1} + x_1) \\ &= x_2x_1 + x_3x_1 + x_3x_2 \end{aligned}$$

Da cui si ottiene una rete con
3 porte AND a 2 ingressi ciascuna
e 1 porta OR a 3 ingressi!

NOTA: per semplificare abbiamo ridotto due mintermini che differivano solo in 1 variabile



Riepilogo e riferimenti

- Gli operatori NAND, NOR, XOR
 - [P] par. 3.6
- Minimizzazione di reti logiche
 - [P] par. 3.5, 4.1, 4.2, 4.3
- E' possibile consultare in alternativa
 - [PH] Appendice B
(Nella terza edizione, Appendice C)

