Logica e Reti Logiche
(Episodio 12: Semplificazioni di formule in forma normale)

Francesco Pasquale

24 novembre 2025

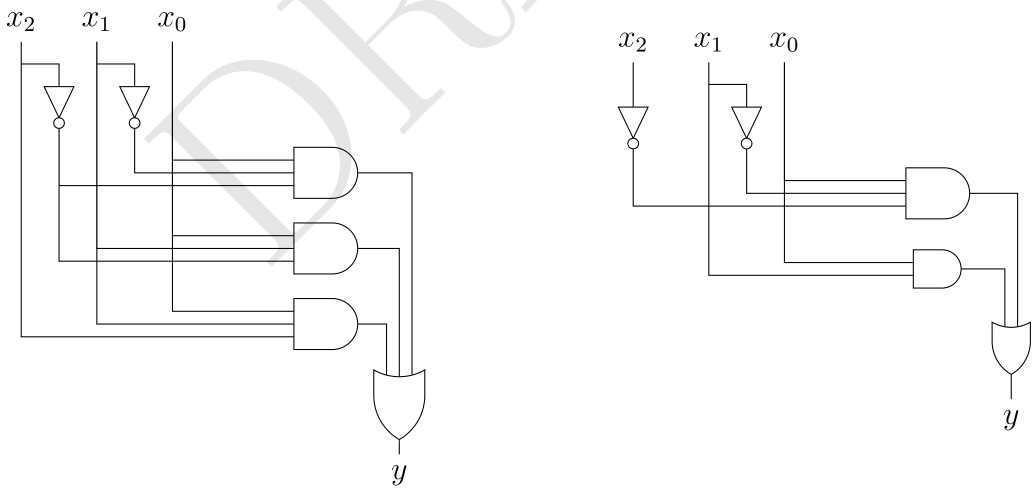
Nell'episodio precedente abbiamo visto che, per una data formula in forma normale, per esempio

$$y = x_0\bar{x}_1\bar{x}_2 + x_0x_1\bar{x}_2 + x_0x_1x_2 \tag{1}$$

è facile costruire immediatamente un circuito che la implementi (Fig. 1a). Tuttavia, se riusciamo a semplificare la formula preservandone la forma normale

$$\begin{aligned} y &= x_0\bar{x}_1\bar{x}_2 + x_0x_1\bar{x}_2 + x_0x_1x_2 \\ &= x_0\bar{x}_1\bar{x}_2 + x_0x_1(\bar{x}_2 + x_2) \\ &= x_0\bar{x}_1\bar{x}_2 + x_0x_1 \end{aligned} \tag{2}$$

possiamo costruire un circuito equivalente "risparmiando" qualche porta (Fig. 1b).



(a) Circuito originale

(b) Circuito semplificato.

Figura 1: Circuiti equivalenti che implementano la Formula 1

Esercizio 1. È possibile semplificare ulteriormente la formula in (2) in modo da avere una formula in forma normale che dia luogo a un circuito con un numero di porte logiche inferiore?

In generale, trovare "il più piccolo circuito" che implementi una data formula è un problema *computazionalmente difficile*[1]. Tuttavia, qualche euristica si può applicare, in alcuni casi particolari.

1 Algebra Booleana

Prima di tutto ricordiamoci qualche semplice proprietà dell'algebra *Booleana*, ossia quella in cui i valori delle variabili sono soltanto 0 (`False`) e 1 (`True`), con i simboli "$\cdot$" e "$+$" indichiamo rispettivamente le operazioni di AND e OR, e con una barretta $\bar{x}$ sopra una variabile x ne indichiamo la negazione.

Osservate che per l'algebra Booleana vale il seguente principio di *dualità*: se un'uguaglianza è vera, allora è vera anche l'uguaglianza che si ottiene scambiando gli zeri con gli uni e gli AND con gli OR. Per esempio, se faccio l'AND di una variabile x con 1 ottengo la variabile stessa $x \cdot 1 = x$. L'uguaglianza duale di questa è quella che si ottiene scambiando l'AND con l'OR e 1 con 0, ossia $x + 0 = x$. Anche questa è vera perché l'OR di una variabile Booleana con 0 è uguale alla variabile stessa. La tabella seguente contiene le principali proprietà dell'AND e dell'OR: le uguaglianze nella colonna di destra sono le duali di quelle nella colonna di sinistra.

	AND	OR
Elemento neutro	$x \cdot 1 = x$	$x + 0 = x$
Annullamento	$x \cdot 0 = 0$	$x + 1 = 1$
Idempotenza	$x \cdot x = x$	$x + x = x$
Complementarità	$x \cdot \bar{x} = 0$	$x + \bar{x} = 1$

Spesso ometteremo del tutto il simbolo "$\cdot$": per esempio, scriveremo direttamente xy per indicare l'AND di x e y.

Nell'algebra Booleana le operazioni di AND e OR sono *commutative*, *associative* e *distributive*:

	AND	OR
Commutatività	$xy = yx$	$x + y = y + x$
Associatività	$x(yz) = (xy)z$	$x + (y + z) = (x + y) + z$
Distributività	$x(y + z) = xy + xz$	$x + yz = (x + y)(x + z)$

Dalle proprietà precedenti possiamo ricavare delle uguaglianze molto utili per semplificare le formule

Assorbimento	$x(x + y) = x$	$x + xy = x$
Combinazione	$xy + x\bar{y} = x$	$(x + y)(x + \bar{y}) = x$
Consenso	$xy + \bar{x}z + yz = xy + \bar{x}z$	$(x + y)(\bar{x} + z)(y + z) = (x + y)(\bar{x} + z)$

Osservate che, nella semplificazione della formula che abbiamo fatto all'inizio di questo episodio, per ottenere la (2) dalla (1) abbiamo usato prima la *distributività* dell'AND rispetto all'OR [$x_0x_1\bar{x}_2 + x_0x_1x_2 = x_0x_1(\bar{x}_2 + x_2)$], poi la *complementarità* dell'OR [$\bar{x}_2 + x_2 = 1$], infine il fatto che 1 è *elemento neutro* per l'AND [$x_0x_1 \cdot 1 = x_0x_1$]. Questi tre passaggi ci danno la semplificazione che nella tabella precedente abbiamo indicato col nome di "combinazione".

[1] Vedrete come si può rendere rigoroso il significato di "computazionalmente difficile" nei corsi di informatica teorica del secondo anno. Nel caso del problema in questione, la "difficoltà" è stata dimostrata qui [Umans, 2001]

Esercizio 2. Ricavare le semplificazioni "assorbimento" e "consenso" dalle proprietà precedenti.

Infine, fra le semplificazioni utili vale senz'altro la pena di ricordare le formule di De Morgan [$\overline{xy} = \bar{x} + \bar{y}$] e [$\overline{x + y} = \bar{x}\bar{y}$], che si possono generalizzare a un numero arbitrario di variabili

$$\overline{(x_1 x_2 \cdots x_n)} = \overline{x1} + \overline{x_2} + \cdots + \overline{x_n} \qquad \text{e} \qquad \overline{(x_1 + x_2 + \cdots + x_n)} = \overline{x1} \cdot \overline{x_2} \cdots \overline{x_n}$$

2 Codici Gray e Mappe di Karnaugh

In una formula Booleana con n variabili, chiamiamo *letterale* una variabile o una variabile negata, chiamiamo *implicante* l'AND di uno o più letterali, chiamiamo *mintermine* un implicante che contiene tutte le n variabili nella formula. Per esempio, nella formula (2) [$x_0\bar{x}_1\bar{x}_2 + x_0x_1$] ci sono i letterali $x_0, x_1, \bar{x}_1$ e $\bar{x}_2$; $x_0\bar{x}_1\bar{x}_2$ è un mintermine; x_0x_1 è un implicante ma non è un mintermine.

Se confrontiamo la semplificazione che abbiamo fatto per passare dalla (1) alla (2)

$$x_0x_1\bar{x}_2 + x_0x_1x_2 = x_0x_1(\bar{x}_2 + x_2) = x_0x_1$$

con la tabella di verità della formula

x_0	x_1	x_2	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

vediamo che abbiamo "raccolto" i due mintermini relativi agli 1 cerchiati in blu in un unico implicante, x_0x_1 dove la variabile x_2 è stata eliminata perché assume valori di verità diversi nelle due righe cerchiate in verde della tabella di verità, mentre x_0 e x_1 compaiono asseriti, perché il loro valore di verità è 1 in entrambe le righe. Si noti che allo stesso modo avremmo potuto semplificare

$$x_0\bar{x}_1\bar{x}_2 + x_0x_1\bar{x}_2 = x_0\bar{x}_2$$

dove a essere eliminata è la variabile x_1, mentre x_0 compare asserita e x_2 compare negata.

x_0	x_1	x_2	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

(3)

Per evidenziare direttamente dalla tabella quali semplificazioni si possono fare, sarebbe utile scrivere le righe con un ordinamento in cui ogni riga differisca dalla precedente e dalla successiva per un unico bit. È possibile farlo? Vediamo.

L'ordinamento usuale con cui scriviamo le 2^n righe di una tabella con n variabili si può generare ricorsivamente in questo modo: per corstruire le righe della tabella con $n+1$ variabili, partiamo dalla righe della tabella con n variabili, la "duplichiamo" e aggiungiamo 0 davanti a tutte le righe dell'"originale" e 1 davanti a tutte le righe della "copia"

$$\begin{array}{c} 0\\ 1 \end{array} \to \begin{array}{c} 0\\ 1\\ \hline 0\\ 1 \end{array} \longrightarrow \begin{array}{cc} 0 & 0\\ 0 & 1\\ 1 & 0\\ 1 & 1 \end{array} \to \begin{array}{cc} 0 & 0\\ 0 & 1\\ 1 & 0\\ 1 & 1\\ \hline 0 & 0\\ 0 & 1\\ 1 & 0\\ 1 & 1 \end{array} \longrightarrow \begin{array}{ccc} 0 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 0\\ 0 & 1 & 1\\ 1 & 0 & 0\\ 1 & 0 & 1\\ 1 & 1 & 0\\ 1 & 1 & 1 \end{array}$$

Se nella costruzione precedente, quando "duplichiamo" la sequenza, la "ribaltiamo" anche, otteniamo un ordinamento chiamato *codice Gray*

$$\begin{array}{c} 0\\ 1 \end{array} \to \begin{array}{c} 0\\ 1\\ \hline 1\\ 0 \end{array} \longrightarrow \begin{array}{cc} 0 & 0\\ 0 & 1\\ 1 & 1\\ 1 & 0 \end{array} \to \begin{array}{cc} 0 & 0\\ 0 & 1\\ 1 & 1\\ 1 & 0\\ \hline 1 & 0\\ 1 & 1\\ 0 & 1\\ 0 & 0 \end{array} \longrightarrow \begin{array}{ccc} 0 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 1\\ 0 & 1 & 0\\ 1 & 1 & 0\\ 1 & 1 & 1\\ 1 & 0 & 1\\ 1 & 0 & 0 \end{array}$$

Esercizio 3. Osservare che, per costruzione, in un codice Gray a n bit

- Ogni sequenza differisce dalla successiva per un unico bit;
- L'ultima sequenza differisce dalla prima per un unico bit.

Quindi se scriviamo una tabella di verità con le righe ordinate secondo il codice Gray, alcune semplificazioni appariranno immediatamente evidenti. Per esempio, la tabella nella (3) verrebbe scritta così

x_0	x_1	x_2	y
0	0	0	0
0	0	1	0
0	1	1	0
0	1	0	0
1	1	0	1
1	1	1	1
1	0	1	0
1	0	0	1

(4)

dove i due 1 adiacenti cerchiati in blu evidenziano la semplificazione ($x_0x_1\bar{x}_2 + x_0x_1x_2 = x_0x_1$). Osservate però che mentre *alcune* semplificazioni si evidenziano, *altre* rimangono "nascoste", come per esempio la semplificazione che si può ottenere dalla quinta e ultima riga della tabella precedente ($x_0x_1\bar{x}_2 + x_0\bar{x}_1\bar{x}_2 = x_0\bar{x}_2$).

Per ottenere il massimo possibile da questa schematizzazione non è sufficiente ordinare le righe di una tabella secondo il codice Gray, ma dobbiamo scrivere le tabelle di verità in modo *bidimensionale*.

Mappe di Karnaugh. Possiamo scrivere la tabella di verità in (4) anche in questo modo

$x_0 \backslash^{x_1}{}_{x_2}$	00	01	11	10
0	0	0	0	0
1	1	0	1	1

dove, per esempio, il numero cerchiato in verde rappresenta il valore che assume l'output y quando la terna in input è $(x_0, x_1, x_2) = (0, 0, 1)$. Una tabella scritta in questo modo si chiama *Mappa di Karnaugh*. Si noti che in una mappa di Karnaugh l'ordine con cui sono indicizzate righe e colonne è quello dato dal codice Gray.

Esercizio 4. Scrivere la mappa di Karnaugh per la tabella di verità

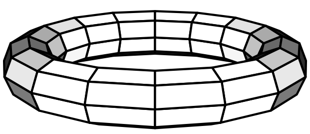
x_0	x_1	x_2	y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

In una mappa di Karnaugh tutti i mintermini che si possono semplificare sono adiacenti

$x_0 \backslash^{x_1}{}_{x_2}$	00	01	11	10
0	0	0	0	0
1	1	0	1	1

(5)

Si noti inoltre che, siccome le colonne della mappa di Karnaugh sono indicizzate con il codice Gray, l'ultima colonna è adiacente alla prima (la semplificazione segnata in rosso qui sopra). *[Regola 1: La mappa di Karnaugh si deve immaginare "arrotolata" su se stessa (l'ultima colonna è adiacente alla prima colonna e l'ultima riga è adiacente alla prima riga)].*



Si noti anche che l'1 in posizione $(x_0, x_1, x_2) = (1, 1, 0)$ nella mappa in (5) è stato considerato due volte. Complessivamente quindi la formula semplificata che otteniamo dalla (5) è

$$y = x_0 x_1 + x_0 \bar{x}_2 \tag{6}$$

dove l'implicante $x_0 x_1$ corrisponde al rettangolo blu in (5) mentre l'implicante $x_0 \bar{x}_2$ corrisponde a quello rosso. *[Regola 2: Gli 1 possono essere inseriti in più di un rettangolo].*

Esercizio 5. Verificare che la formula (6) è equivalente alla (1) e osservare che il circuito corrispondente alla (6) ha un NOT in meno e un AND a due ingressi invece che uno a tre, rispetto al circui in Fig.1b.

Osservate che, nonostante gli 1 siano tutti adiacenti nella (5) non possiamo raggruparli tutti e tre insieme in un unico implicante, mentre avremmo potuto farlo se fossero stati "quattro" adiacenti. *[Regola 3: I lati dei rettangoli in cui raccogliamo gli 1 devono essere potenze di 2].* Per esempio, nella mappa qui sotto possiamo raccogliere gli 1 in questo modo

$x_0 \backslash {}^{x_1}{}_{x_2}$	00	01	11	10
0	0	1	0	0
1	1	1	1	1

(7)

Ottenendo la formula $x_0 + \bar{x}_1 x_2$, dove l'implicante x_0 viene dal rettangolo blu (x_0 vale 1 in tutte le celle del rettangolo blu, mentre sia x_1 che x_2 valgono 0 in alcune celle e 1 in altre) e l'implicante $\bar{x}_1 x_2$ viene dal rettangolo rosso (x_1 vale 0 in entrambe le celle del rettangolo e x_2 vale 1 in entrambe le celle, mentre x_0 vale 1 in una cella e 0 nell'altra).

Esercizio 6. Scrivere la tabella di verità nella forma standard della mappa di Karnaugh in (7) e verificare che corrisponde alla formula $x_0 + \bar{x}_1 x_2$.

Se ci limitiamo a formule con al più quattro variabili, le mappe di Karnaugh ci consentono di semplificare una formula in forma normale disgiuntiva fino a ottenere il minor numero possibile di implicanti e con il minor numero di variabili per implicante. Per esempio, dalla seguente mappa di Karnaugh

${}^{x_0}{}_{x_1} \backslash {}^{x_2}{}_{x_3}$	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	0	0	0	0
10	0	1	1	0

(8)

corrisponde la formula

$$\bar{x}_0 \bar{x}_3 + \bar{x}_0 x_1 + x_0 \bar{x}_1 x_3$$

dove il primo implicante, $\bar{x}_0 \bar{x}_3$, viene dal quadrato blu (x_0 e x_3 valgono 0 in tutte le celle del quadrato, mentre x_1 e x_2 sono 0 in alcune celle e 1 in altre), il secondo implicante $\bar{x}_0 x_1$ viene dal rettangolo verde e il terzo implicante, $x_0 \bar{x}_1 x_3$, viene dal rettangolo rosso.

Osservate che se nella mappa (8) avessimo considerato dei rettangoli diversi (per esempio, se avessimo cerchiato in verde soltanto i due 1 centrali invece che includere anche i due 1 già considerati nel rettangolo blu) avremmo ottenuto una formula equivalente ma non minimizzata. *[Regola 4: Ogni rettangolo deve essere il più grande possibile].*

Esercizio 7. Scrivere la mappa di Karnaugh della seguente tabella di verità e disegnare il circuito corrispondente

x_0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x_1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
y	0	0	0	1	1	0	1	1	0	0	0	0	1	0	1	0

Esercizio 8. Abbiamo imparato come costruire una formula in forma normale disgiuntiva (somma di prodotti) a partire da una mappa di Karnaugh. Come possiamo costruire una formula in forma normale congiuntiva (prodotto di somme) a partire dalla stessa mappa (senza costruire prima la disgiuntiva)?

Esercizio 9. Usando solo porte AND, OR e NOT, progettare un circuito che implementi la seguente funzione booleana $f : \{0,1\}^4 \to \{0,1\}$

$$f(x_1, x_2, x_3, x_4) = \begin{cases} x_3 x_4 & \text{se } x_1 \neq x_2 \\ x_3 + x_4 & \text{altrimenti} \end{cases}$$