Capitolo 7

Equivalenza e minimizzazione

Abbiamo più volte ripetuto che ogni stato interno di una rete sequenziale è indicativo di una particolare informazione sulle sequenze di ingresso. Non essendovi però un metodo generale per individuare e codificare tali informazioni, è possibile che nella sintesi di una rete si introducano stati distinti per memorizzare informazioni sostanzialmente uguali. La rete così costruita può cioè avere stati ridondanti, e potrebbe essere semplificata raggruppando tali stati in uno solo.

Abbiamo già incontrato questa situazione nella rete per addizionare tre numeri (§ 5.3, diagramma di fig. 5.6), ove lo stato A "iniziale" e lo stato B "di riporto 0", introdotti per rappresentare due condizioni apparentemente diverse, svolgono in pratica la stessa funzione. Studiamo ora il problema in modo sistematico, utilizzando il modello generale di automa (§ 5.1). Una volta noto, il procedimento di riduzione degli stati costituirà un passo standard nella sintesi delle reti sequenziali: costruita infatti la tabella di flusso senza preoccuparsi che contenga stati ridondanti, essa sarà ridotta prima di essere trasformata in tabella delle transizioni.

Descriviamo il diagramma degli stati dell'automa attraverso una *funzione di prossimo stato* Σ, e una *funzione di uscita* Ω, definite sull'insieme $S \times X$ delle coppie (stato interno, stato di ingresso). Cioè, se l'automa si trova nello stato interno S_i e riceve lo stato di ingresso X_j:

$\Sigma(S_i, X_j)$ indica il prossimo stato interno;

$\Omega(S_i, X_j)$ indica lo stato di uscita.

Lo studio, come vedremo, differisce sostanzialmente se le funzioni Σ e Ω sono definite su tutto il dominio $S \times X$ o su parte di esso.

7.1 Funzionamento completamente specificato

Ammettiamo che il funzionamento dell'automa sia *completamente specificato*. Ciò significa che le funzioni Σ e Ω sono definite su tutti i punti del dominio $S \times X$, ovvero che per ogni coppia (stato interno, stato di ingresso) sono specificati il prossimo stato interno e lo stato di uscita. Tutti gli sviluppi contenuti nel presente paragrafo sono validi nell'ipotesi di completa specificazione; i casi in cui quest'ipotesi cada (già accennati nell'esercizio 5.2) saranno approfonditi nel prossimo paragrafo.

Due stati svolgono identiche funzioni se non possono essere distinti mediante osservazioni esterne sul funzionamento dell'automa. Poniamo perciò la seguente definizione:

Definizione 1 Lo stato S_1 dell'automa A_1 e lo stato S_2 dell'automa A_2 sono *equivalenti* se e solo se, per qualsiasi sequenza di ingresso applicata ad A_1 nello stato iniziale S_1, e ad A_2 nello stato iniziale S_2, si ottiene la stessa sequenza di uscita. In particolare A_1 può coincidere con A_2 (equivalenza tra stati dello stesso automa).

L'equivalenza tra stati si indica con il simbolo: $S_1 \sim S_2$, e gode ovviamente delle proprietà riflessiva, simmetrica e transitiva. Essa è cioè una relazione di equivalenza in senso algebrico, e ripartisce l'insieme S degli stati di un automa in *classi di equivalenza* disgiunte: ogni classe di equivalenza contiene tutti stati mutuamente equivalenti, né alcuno stato equivalente a questi è contenuto in una diversa classe.

Dall'equivalenza tra stati si passa a definire l'equivalenza tra automi:

Definizione 2 Due automi A_1 e A_2 sono *equivalenti* se per ogni stato S_i di A_1 esiste uno stato S_j di A_2 con $S_i \sim S_j$, e viceversa.

Anche la relazione di equivalenza tra automi è una relazione di equivalenza in senso algebrico, e si indica anch'essa con: $A_1 \sim A_2$.

E' ora chiaro che, dato un automa A con s classi di equivalenza, tutti gli stati di ogni classe possono essere sostituiti con uno solo di essi, dando luogo a un nuovo automa A_m equivalente ad A. A_m ha s stati, e possiamo immediatamente constatare che nessun automa $A' \sim A$ può avere un numero di stati $s' < s$. Infatti ogni stato di A' deve avere uno stato equivalente in A (definizione 2); dunque, se fosse $s' < s$, dovrebbe esistere almeno uno stato S_i di A' equivalente a due stati S_j, S_h di A non appartenenti alla stessa classe di equivalenza; cioè $S_j \not\sim S_h$, contro l'ipotesi $S_j \sim S_i$, $S_h \sim S_i$ e la transitività della relazione. A_m è quindi un *automa minimo* equivalente ad A, ed è anche l'unico automa minimo poiché unica è la partizione in classi di equivalenza per A.

Se il precedente ragionamento garantisce l'esistenza e l'unicità dell'automa minimo A_m equivalente a un automa dato A, resta da definire un procedimento di costruzione di A_m. Tale procedimento consisterà nella determinazione delle classi di equivalenza di A, e nell'associazione degli stati del nuovo automa a queste classi, secondo quanto ora indicheremo.

Notiamo anzitutto che la definizione 1 non è direttamente utilizzabile per stabilire se due stati sono equivalenti, poiché implicherebbe l'esame di tutte le sequenze di ingresso che non sono in numero finito. Si ricorre pertanto alla seguente proposizione, che è immediata conseguenza della definizione 1.

Proposizione 1 $S_i \sim S_j$ se e solo se, $\forall X_h \in X$:

$$\Omega(S_i, X_h) = \Omega(S_j, X_h),$$
$$\Sigma(S_i, X_h) \sim \Sigma(S_j, X_h).$$

L'equivalenza tra S_i e S_j può così essere affermata esaminando i singoli stati (non le sequenze) di ingresso X_h, constatando che le uscite $\Omega(S_i, X_h)$ e $\Omega(S_j, X_h)$ prodotte sono uguali, e ripetendo ricorsivamente il controllo sulle coppie dei prossimi stati $\Sigma(S_i, X_h)$ e $\Sigma(S_j, X_h)$ raggiunti. L'equivalenza di due stati richiede cioè che siano equivalenti altre coppie, secondo un meccanismo che ora facilmente trasformeremo in procedura di calcolo. Vedremo che le catene di richieste hanno sempre lunghezza finita e quindi il procedimento converge sempre.

Consideriamo l'automa descritto dalla tabella di flusso della figura 7.1*a*, e vediamo su tale esempio come si costruiscono le classi di equivalenza degli stati. Tutte le coppie di stati vengono esaminate alla luce della proposizione 1, e i risultati sono riportati nella *tabella a scala* indicata nella figura 7.1*b* (che ha una casella per ogni coppia S_i, S_j) secondo le regole seguenti.

1) Se esiste uno stato di ingresso X_h (cioè una colonna della tabella di flusso) per cui $\Omega(S_i, X_h) \neq \Omega(S_j, X_h)$, gli stati S_i e S_j sono immediatamente riconosciuti non equivalenti, e la corrispondente casella della tabella a scala è sbarrata con una croce: per esempio gli stati A e

C della figura 7.1*a* hanno uscite differenti nella colonna 0, e quindi si sbarra la casella A-C della tabella a scala.

2) Se, per ogni ingresso X_h, si ha $\Omega(S_i, X_h)=\Omega(S_j, X_h)$, gli stati S_i e S_j sono equivalenti, a condizione che lo siano, colonna per colonna, i prossimi stati. Nella casella S_i-S_j della tabella a scala si riportano, per ogni X_h, le coppie $\Sigma(S_i, X_h)$, $\Sigma(S_j, X_h)$ di stati di cui si richiede l'equivalenza, tralasciando però di registrare le coppie per cui

$$\Sigma(S_i, X_h)=\Sigma(S_j, X_h), \qquad [7.1]$$

oppure

$$\Sigma(S_i, X_h)=S_i, \quad \Sigma(S_j, X_h)=S_j, \qquad [7.2]$$

oppure

$$\Sigma(S_i, X_h)=S_j, \quad \Sigma(S_j, X_h)=S_i, \qquad [7.3]$$

che non pongono significative condizioni all'equivalenza di S_i, S_j.

Stato interno \ Ingresso	0	1
A	G, 0	E, 1
B	C, 0	A, 1
C	E, 1	G, 0
D	B, 0	E, 1
E	G, 0	A, 1
F	D, 1	F, 0
G	A, 1	G, 0

(*a*)

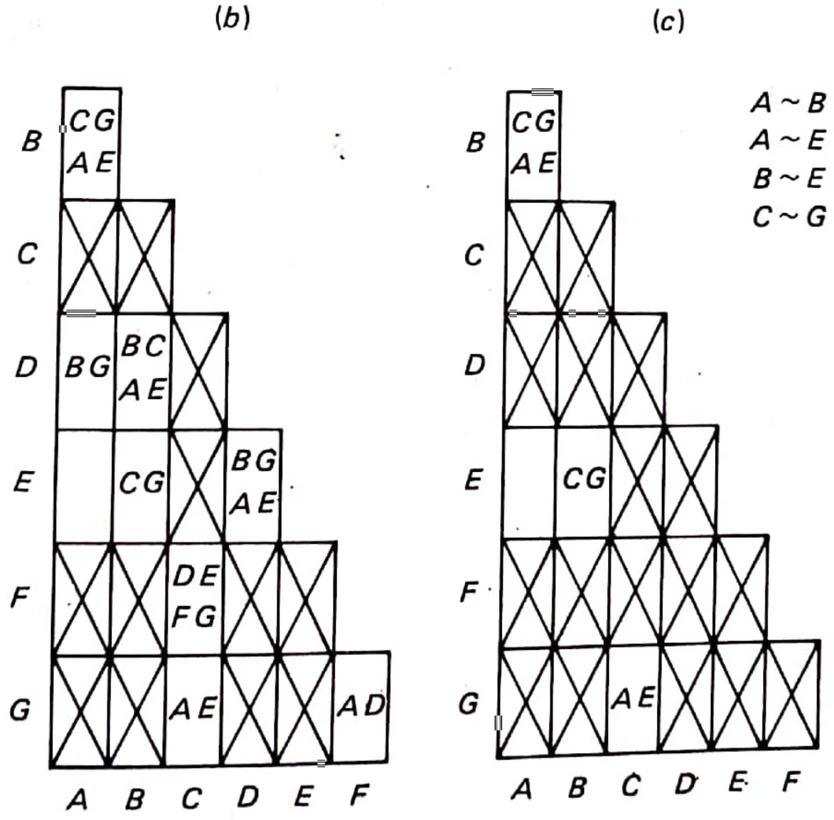

Figura 7.1
Determinazione delle coppie di stati equivalenti.

Per esempio gli stati A e B della figura 7.1*a* richiedono l'equivalenza di C e G, e di A ed E, e tali coppie sono riportate nella casella A-B della tabella a scala; gli stati A e D richiedono l'equivalenza di B e G per ingresso 0, mentre nessuna condizione è imposta per ingresso 1 poiché i prossimi stati coincidono (situazione [7.1]); gli stati A ed E non richiedono alcuna condizione, poiché per ingressi 0 e 1 si hanno rispettivamente le situazioni [7.1] e [7.3]: nessuna condizione è riportata nella casella A-E della tabella a scala, a indicare che A ed E sono direttamente riconosciuti equivalenti.

Riempita la tabella a scala (fig. 7.1*b*), le catene di richieste devono essere propagate al suo interno. Si esaminano le caselle dall'alto verso il basso e da sinistra a destra; se la casella S_i-S_j contiene la condizione $S_r\ S_t$, e la casella S_r-S_t contiene una croce, tale croce si propaga nella casella S_i-S_j, a indicare che anche questi stati non sono equivalenti (fig. 7.1*c*). Per esempio la casella A-D contiene la condizione $B\ G$, la casella B-G contiene una croce, dunque anche la casella A-D viene sbarrata. Esaminate una volta tutte le caselle si ha un primo aggiornamento della tabella a scala (fig. 7.1*c*, ove la casella C-F non è ancora sbarrata).

Si itera il procedimento propagando ulteriormente le croci: la casella C-F contiene le condizioni $D\ E$ ed $F\ G$, e viene sbarrata nel corso della seconda iterazione (croce tratteggiata) quando si incontra la croce nella casella D-E. Si procede così per iterazioni successive, finché un'intera iterazione non dà luogo ad alcuna variazione della tabella e il procedimento di aggiornamento è concluso (fig. 7.1*c* con tutte le croci ivi segnate).

Le coppie di stati equivalenti *corrispondono alle caselle della tabella finale che non contengono croci*: infatti le richieste ivi specificate sono tutte compatibili con tali equivalenze. Nel nostro esempio si hanno le seguenti equivalenze (caselle non sbarrate) e richieste:

$$A \sim B \Rightarrow C \sim G, \qquad A \sim E,$$
$$A \sim E \Rightarrow \text{nulla},$$
$$B \sim E \Rightarrow C \sim G,$$
$$C \sim G \Rightarrow A \sim E.$$

Il procedimento considerato ha complessità polinomiale nelle dimensioni dell'automa. Infatti, indicati con k e p il numero di stati interni e di stati di ingresso, la costruzione iniziale della tabella a scala richiede il confronto delle $(k-1)k/2$ coppie di stati interni, ciascuna esaminata sulle p colonne della tabella di flusso: tale costruzione richiede quindi tempo di $O(k^2 p)$. Ogni iterazione per l'aggiornamento della tabella a scala richiede, per ognuna delle sue $(k-1)k/2$ caselle, l'esame delle condizioni (contenuti di altre caselle) ivi specificate, che sono al più p: dunque anche ogni iterazione richiede tempo di $O(k^2 p)$. Poiché il procedimento si arresta quando in un'iterazione non si propaga alcuna croce, esso può al più richiedere $(k-1)k/2$ iterazioni, se la (unica) croce contenuta nella configurazione iniziale si propaga, un passo per iterazione, in tutte le altre caselle. Il procedimento converge dunque in $O(k^2)$ iterazioni, e la sua complessità in tempo è nel caso pessimo di $O(k^4 p)$. Nei casi usuali, ovviamente, il tempo richiesto è drasticamente inferiore.

Trovate le c coppie di stati equivalenti, si possono costruire le classi di equivalenza con un procedimento di complessità lineare in c, che indichiamo brevemente per l'esempio della figura 7.1. Si parte dalla lista ordinata L di coppie:

$$L = A \sim B, A \sim E, B \sim E, C \sim G.$$

Tutte le coppie di L che contengono il primo stato (A) formano la classe di equivalenza di tale stato: (A, B, E). Le coppie contenute in questa classe ($A \sim B, A \sim E, B \sim E$) si tolgono da L, e si ripete il procedimento sulle coppie rimaste in lista fino a che questa si vuota. Nel nostro esempio la seconda e ultima classe individuata è (C, G). Tutti gli stati non contenuti nelle coppie di L formano classi composte di singoli stati: (D), (F). Concludendo, l'automa della figura 7.1a ha le classi di equivalenza

$$(A, B, E), (C, G), (D), (F).$$

A questo punto è immediata la costruzione dell'automa minimo A_m equivalente a un automa dato A. A_m ha uno stato, diciamo α, per ogni classe di equivalenza C di A; la tabella di flusso di A_m si costruisce, per ogni coppia (stato interno α, stato di ingresso X_h), prendendo a caso uno stato $S_i \in C$ e assegnando come uscita di A_m l'uscita $\Omega(S_i, X_h)$ di A, e come prossimo stato di A_m lo stato β associato alla classe di equivalenza cui appartiene il prossimo stato $\Sigma(S_i, X_h)$ di A.

Nel nostro esempio costruiamo l'automa minimo a quattro stati dalla figura 7.2, equivalente all'automa della figura 7.1a.

	Stato interno \ Ingresso	0	1
α: (A, B, E)	α	$\beta, 0$	$\alpha, 1$
β: (C, G)	β	$\alpha, 1$	$\beta, 0$
γ: (D)	γ	$\alpha, 0$	$\alpha, 1$
δ: (F)	δ	$\gamma, 1$	$\delta, 0$

Figura 7.2
Automa minimo equivalente all'automa della figura 7.1*a*.

7.2 Funzionamento incompletamente specificato

Come già accennato, non sempre le funzioni Σ e Ω sono definite su tutti i punti del dominio $S \times X$. Se ciò non avviene, cioè se esistono coppie (stato interno, stato di ingresso) per cui non è specificato il prossimo stato interno, o lo stato di uscita, o entrambi, l'automa ha funzionamento *incompletamente specificato*, e tutta la teoria dell'equivalenza e minimizzazione deve essere riformulata.

Osserviamo anzitutto che la non specificazione dell'automa in un punto S_i, X_h è decretata durante il procedimento di sintesi di una rete sequenziale: se è impossibile, o non significativo, che tale punto sia raggiunto durante il funzionamento della rete, sono non specificati il prossimo stato e l'uscita; se il punto può essere raggiunto solo come ultimo di una sequenza significativa, è non specificato il prossimo stato; se, infine, il punto appartiene a una sequenza per cui non è interessante l'uscita (per esempio una sequenza che ha il solo scopo di condurre a un determinato stato interno), tale uscita non è specificata.

Per il funzionamento incompletamente specificato, il concetto di equivalenza tra stati viene esteso come qui sotto indicato, notando che, quando un automa raggiunge uno stato non specificato, qualsiasi sequenza di uscita sarà da tal punto completamente non specificata.

Definizione 3 Lo stato S_1 dell'automa A_1 e lo stato S_2 dell'automa A_2 sono *compatibili* se e solo se, per qualsiasi sequenza di ingresso applicata ad A_1 nello stato S_1 e ad A_2 nello stato S_2, si ottengono due sequenze di uscita uguali ovunque entrambe sono specificate. In