Corso di Calcolatori Elettronici I

Macchine sequenziali: minimizzazione degli stati

Prof. Roberto Canonico

Università degli Studi di Napoli Federico II
Dipartimento di Ingegneria Elettrica
e delle Tecnologie dell'Informazione
Corso di Laurea in Ingegneria Informatica
Corso di Laurea in Ingegneria dell'Automazione

Funzioni uscita e stato prossimo

- L'uscita e lo stato prossimo sono funzioni della sequenza di ingressi applicata a partire da uno "stato iniziale":

  $u_{\mathrm{k}} = \lambda(q_0, J_k)$

  $q_{k+1} = \delta(q_0, J_k)$
- *con* $J_k = i_0, i_1, i_2, \ldots, i_k$

Macchine complete e incomplete

Applicabilità di una sequenza

- Una sequenza di ingressi J è *applicabile* a M in q - *si dice a M(q)* - se è definita *u=λ(q, J), la funzione che fornisce l'uscita u che si ottiene applicando alla macchina la sequenza di ingressi J a partire da uno "stato iniziale" q.*
- Se la funzione di uscita *λ* è definita ovunque, la macchina si dice **completa**, **incompleta** altrimenti
- **Per le macchine incomplete esistono sequenze non applicabili**
  - Sequenza J_j *non applicabile* in q_0: *λ* non definita
    - Potrebbe essere applicabile una sequenza più lunga

Equivalenza

- Occorre formalizzare il fatto che due macchine possano avere lo stesso funzionamento

  → reagire nello stesso modo (con le stesse uscite) alle stesse sequenze di ingressi
- Definizione di ***stati equivalenti*** in macchine complete*:*
  - **Producono la stessa sequenza di uscite per qualsiasi sequenza di ingressi**

Equivalenza

- Un modo per riconoscere stati equivalenti (fondamentale negli algoritmi che vedremo) è usare la proprietà ricorsiva degli stati equivalenti:

  Due stati sono equivalenti se lo sono tutte le possibili coppie di stati successivi, e sono uguali tutte le possibili uscite successive.

Equivalenza: riassumendo

- Concetto di equivalenza: avere lo stesso funzionamento "esterno"
  - Reagire nello stesso modo (con le stesse uscite) alle stesse sequenze di ingressi
  - Due stati sono equivalenti se, per ciascun ingresso:
    - sono eguali le uscite
    - sono equivalenti gli stati successivi
  - Definizione di *stati equivalenti* in macchine **complete***:*
    - Producono la stessa sequenza di uscite per qualsiasi sequenza di ingressi

Equivalenza

- I due stati possono appartenere anche alla stessa macchina
- Due macchine *complete* M e M' sono equivalenti se per ciascuno stato q di M esiste almeno uno stato q' di M' ad esso equivalente e, viceversa

Equivalenza e macchine incompete

- La definizione precedente non può essere applicata così com'è
  → non tutte le possibili sequenze sono applicabili a tutti gli stati
- Si introducono i concetti di:
  - ***Compatibilità*** tra stati
  - ***Inclusione*** tra macchine

Equivalenza: ricapitolando

Stati equivalenti (macchine complete)

- Due stati (della stessa macchina o di macchine diverse) sono equivalenti se producono la stessa sequenza di uscite per qualsiasi sequenza di ingressi
- Definizione ricorsiva
  - Due stati sono equivalenti se, per ciascun ingresso sono eguali le uscite e sono equivalenti gli stati successivi

Macchine equivalenti (complete)

- Due macchine complete M e M' sono equivalenti se per ciascuno stato q di M esiste almeno uno stato q' di M' ad esso equivalente e viceversa

Macchine equivalenti (incomplete): non tutte le sequenze di ingresso sono applicabili

- il concetto è sostituito da quelli di **Compatibilità** tra stati ed **Inclusione** tra macchine

Stati compatibili

- Due stati sono **compatibili** se *per ogni* sequenza di ingressi applicabili ad *entrambi,* le uscite prodotte sono identiche
- A differenza della relazione vista prima per macchine complete, la compatibilità **NON** è una relazione di equivalenza:

  Gode delle proprietà
  - Riflessiva
  - Simmetrica
  - Ma NON di quella transitiva
    - p.e.: q1~q2, q2~q3, λ(q1,J)=a, , λ(q2,J)=-, λ(q3,J)=b
- Ciò complica la ricerca di macchine equivalenti minime

Compatibilità

- Così come l'equivalenza, anche la compatibilità può essere definita ricorsivamente
- **Due stati sono compatibili se tutti i possibili stati prossimi sono compatibili, e tutte le possibili uscite prossime sono uguali**
- Non essendo una relazione di equivalenza, non è possibile utilizzare le proprietà delle classi di equivalenza.
- Si generalizza con il concetto di *famiglia di insiemi di stati compatibili massimi*

Compatibilità

- Per le macchine incomplete, non si parla quindi di equivalenza, ma di inclusione:
  - Una macchina M' ne include una M, *in una coppia di stati q* e *q' ,* se tutte le sequenze di ingressi applicabili ad M a partire da q lo sono anche per M' a partire da q' producendo la stessa uscita
  - Se è possibile trovare per ciascuno stato di M uno q che soddisfa la precedente definizione, allora M' include M
    → è possibile usare M' in luogo della M
  - M ed M' possono includersi l'un l'altra
    → diremo in questo caso che le due macchine sono equivalenti

Compatibilità: formalmente

Inclusione fra macchine

- **Concettualmente:** una macchina include un'altra se "fa qualcosa in più"
- **Formalmente:**
  - M'(q') include M(q):
    $$M'(q') \supseteq M(q) \Leftrightarrow \forall (J \text{ applicabile a } M(q)) : \lambda(q', J) = \lambda(q, J)$$
    - M' include M
    $$M' \supseteq M \Leftrightarrow \forall (q \in M) \exists (q' \in M') : M'(q') \supseteq M(q)$$

Compatibilità ed equivalenza

- Nel caso di macchine complete le due definizioni coincidono.
- Tra due macchine equivalenti, conviene scegliere quella con meno stati
- → problema di minimizzazione

  individuare la macchina con il minor numero di stati tra tutte le possibili macchine equivalenti

Minimizzazione degli stati

Minimizzazione e classi di equivalenza

- Data una macchina M se ne vuole trovare una equivalente, M', con un numero minimo di stati.
- **La famiglia delle classi di equivalenza degli stati di M è la soluzione**: M' ha uno stato per ogni elemento C della famiglia con:
  - **uscita:** pari alla uscita degli elementi di C, tutti eguali per essere questi equivalenti fra loro,
  - **stato seguente:** quello determinato da C stesso; essendo infatti gli stati di C_i equivalenti fra loro, devono avere tutti per seguenti stati equivalenti e quindi appartenere ad un unico elemento C' della famiglia delle classi di equivalenza.
  - **M' ha un numero di stati minimo, per le proprietà delle classi di equivalenza**.

Minimizzazione degli stati

Minimizzazione e classi di compatibilità massime

- Per le macchine incomplete, proprietà analoghe a quelle delle classi di equivalenza, ma più complesse, hanno le classi di compatibilità massime
- Si procede analogamente, con le seguenti peculiarità:
  - M' include M (non è equivalente)
  - La soluzione potrebbe non essere minima, ma in generale presenta una macchina a "stati ridotti"
  - Si possono avere più soluzioni, potendo uno "stato seguente" essere incluso in più elementi distinti della famiglia di compatibilità massima

Problema della Minimizzazione

- Partendo da una macchina *M(Q,I,U,τ,ω)* , ne vogliamo trovare a macchina *M'(Q',I,U,τ',ω')* equivalente ad M e con il minor numero di stati
- Partiamo dalla famiglia di insiemi di stati compatibili massimi $F=(S_1,S_2,...,S_n)$

Problema della Minimizzazione

- La F gode delle seguenti proprietà, essenziali nei metodi di minimizzazione:
  - Gli elementi S di F sono disgiunti
  - Gli elementi S di F coprono l'insieme degli stati Q
  - Tutti gli stati di un elemento S di F portano alla stessa uscita (eventualmente non definita)
  - F è chiusa: da due stati di uno stesso elemento S di F si arriva a due stati che appartengono ad una stessa S'
- Ricerchiamo la *M'(F,I,U,τ',ω')*
  → M' ha un numero di stati non superiore a M

Ricerca della famiglia F

- Algoritmo del partizionamento
- Metodo tabellare di Paull-Unger

- Procedono per “eliminazione”

  Partono da una presunta F (inizialmente coincidente con Q) e cercano di individuare incompatibilità fin quando è possibile

Algoritmo del partizionamento

- Si individuano gli stati incompatibili rispetto alle uscite per ciascun ingresso
- Le partizioni individuate si esaminano rispetto allo stato prossimo
- Si itera fintantoché tutte le partizioni non verificano la definizione di compatibilità

Algoritmo del partizionamento

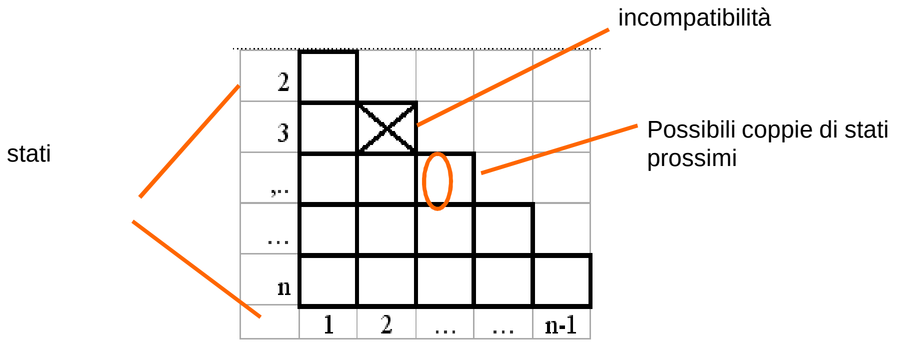
statii	i_1	i_2
q_1	q_2/u_1	q_7/u_2
q_2	q_4/u_2	q_7/u_1
q_3	q_1/u_1	q_5/u_2
q_4	q_2/u_2	q_3/u_1
q_5	q_4/u_1	q_3/u_2
q_6	q_1/u_2	q_2/u_2
q_7	q_5/u_1	q_5/u_2

i	Elemento in esame	Analisi uscite	Partizione elementi	Famiglia
				(1,2,3,4,5,6,7)
1	(1,2,3,4,5,6,7)	u_1: (1,3,5,7); u_2:(2,4,6)	(1,3,5,7) (2,4,6)	(1,3,5,7) (2,4,6)
2	(1,3,5,7)	u_2: (1,3,5,7)		
2	(2,4,6)	u_1:(2,4); u_2: (6)	(2,4) (6)	(1,3,5,7) (2,4,)(6)

i	Stati seguenti	Analisi stati seguenti	Partizione elemento	Famiglia
Passo 3				(1,3,5,7) (2,4) (6)
1	(1,3,5,7)→(2,1,4,5)	(2,4) (1,5)→(1,5) (3,7)	(1,5) (3,7)	(1,5) (3,7) (2,4) (6)
1	(2,4)→(4,2)			
2	(2,4)→(7,3)			

Metodo di Paull-Unger

- Riorganizza il procedimento visto prima in forma di matrice diagonale



Metodo di Paull-Unger

- Si marcano come incompatibili le coppie di stati che portano ad uscite differenti per almeno un ingresso
- Si indicano le coppie di possibili stati prossimi in ogni casella
- Si continua iterativamente il procedimento partizionando rispetto agli stati

Metodo di Paull-Unger

- Ogni elemento della Tabella delle Implicazioni contiene:
  - Il simbolo di non equivalenza (X)
  - Il simbolo di equivalenza (~)
  - La coppia di stati *condizionanti* se non è possible stabilire immediatamente l'equivalenza (o non equivalenza)

S1	X		
S2	X	~	
S3	S1,S2	X	X
	S0	S1	S2

Esempio – Paull-Unger

	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

b	cg ae					
c	x	x				
d	bg	bc ae	x			
e		cg	x	bg ae		
f	x	x	de fg	x	x	
g	x		ae	x	x	
	a	b	c	d	e	f

"e" ed "a" hanno le stesse uscite per ogni ingresso

"g" ed "f" sono equivalenti se lo sono "a" e "d"

"g" e "b" hanno uscite differenti (con rif ad uno stesso ingresso)

Esempio - cont

- Procedendo iterativamente si giunge a determinare le classi di equivalenza

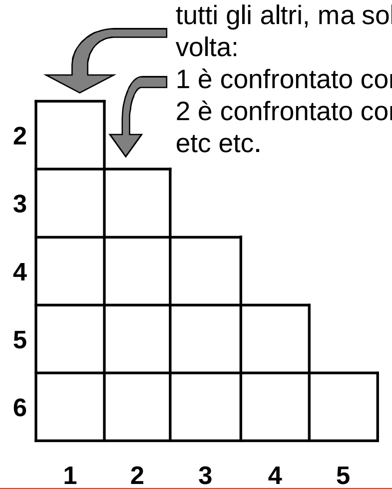
b	~					
c	x	x				
d	x	x	x			
e	~	~	x	x		
f	x	x	x	x	x	
g	x	x	~	x	x	x
	a	b	c	d	e	f

$\alpha = \{a,b,e\}$
$\beta = \{c,g\}$
$\gamma = \{d\}$
$\delta = \{f\}$

Metodo di Paull-Unger

	i_1	i_2
1	2/A	4/B
2	3/B	5/A
3	5/A	4/B
4	2/B	2/B
5	2/B	5/A
6	3/A	5/B

Ogni stato è confrontato con tutti gli altri, ma solo una volta:
1 è confrontato con 2…6
2 è confrontato con 3…6
etc etc.



Esempio: Riconoscitore di codice 8-4-2-1

- Costruire una rete nella quale entrano serialmente i bit di un codice decimale 8-4-2-1 a partire dal bit meno significativo e dalla quale esce un segnale impulsivo che individua se i quattro bit costituiscono o meno una delle 10 parole-codice previste

Riferimento: “Reti logiche-Complementi ed Esercizi” CAP 5, es n.6

Esempio: Riconoscitore di codice 8-4-2-1

- Procediamo per elencazione di tutti le possibili sequenze
  - Individuiamo tutti i possibili stati
  - Partizioniamo rispetto alle uscite

cifra	8-4-2-1
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

	1	0
0	1/0	2/0
1	3/0	4/0
2	5/0	6/0
3	7/0	8/0
4	9/0	10/0
5	11/0	12/0
6	13/0	14/0
7	0/0	0/1
8	0/0	0/1
9	0/0	0/1
10	0/1	0/1
11	0/0	0/1
12	0/0	0/1
13	0/0	0/1
14	0/1	0/1

	1	0
0	1/0	2/0
1	3/0	4/0
2	5/0	6/0
3	7/0	8/0
4	9/0	10/0
5	11/0	12/0
6	13/0	14/0
7	0/0	0/1
8	0/0	0/1
9	0/0	0/1
11	0/0	0/1
12	0/0	0/1
13	0/0	0/1
10	0/1	0/1
14	0/1	0/1

Esempio: Riconoscitore di codice 8-4-2-1

- Eliminiamo le righe uguali
  - ne resta soltanto una e lo stato non eliminato (ad es. 7) viene sostituito a tutti quelli eliminati nella colonna degli stati futuri

	1	0
0	1/0	2/0
1	3/0	4/0
2	5/0	6/0
3	7/0	8/0
4	9/0	10/0
5	11/0	12/0
6	13/0	14/0
7	0/0	0/1
~~8~~	~~0/0~~	~~0/1~~
~~9~~	~~0/0~~	~~0/1~~
~~11~~	~~0/0~~	~~0/1~~
~~12~~	~~0/0~~	~~0/1~~
~~13~~	~~0/0~~	~~0/1~~
10	0/1	0/1
~~14~~	~~0/1~~	~~0/1~~

	1	0	
0	1/0	2/0	
1	3/0	4/0	
2	5/0	6/0	
3	7/0	7/0	Non 8*
4	9/0	10/0	
5	11/0	7/0	
6	13/0	10/0	Non 14
7	0/0	0/1	
10	0/1	0/1	

* Lo stesso vale per 9, 11, 12 e 13

Esempio: Riconoscitore di codice 8-4-2-1

- Tracciamo la tabella delle implicazioni

1	~~1-3;2-4~~							
2	~~1-5;2-6~~	3-5;4-6						
3	~~1-7;2-7~~	~~3-7;4-7~~	~~5-7;6-7~~					
4	~~1-7;2-10~~	~~3-7;4-10~~	~~5-7;6-10~~	~~7-10~~				
5	~~1-7;2-7~~	~~3-7;4-7~~	~~5-7;6-7~~	~	~~7-10~~			
6	~~1-7;2-10~~	~~3-7;4-10~~	~~5-7;6-10~~	~~7-10~~	~	~~7-10~~		
7	X	X	X	X	X	X	X	
10	X	X	X	X	X	X	X	X
	0	1	2	3	4	5	6	7

Esempio: Riconoscitore di codice 8-4-2-1

- Classi di equivalenza
  - Ricordando anche gli stati “fusi” in precedenza

S0=(0),
S1=(1,2),
S2=(3,5),
S3=(4,6),
S4= **(7,8,9,11,12,13),**
S5= (10,14).

	1	0
S0	S1/0	S1/0
S1	S2/0	S3/0
S2	S4/0	S4/0
S3	S4/0	S5/0
S4	S0/0	S0/1
S5	S0/1	S0/1

Tabella degli stati ridotti