

Sistemi - Modulo di Sistemi a Eventi Discreti

Laurea Magistrale in Ingegneria e Scienze Informatiche
Tiziano Villa

24 Febbraio 2014

Nome e Cognome:

Matricola:

Posta elettronica:

problema	punti massimi	i tuoi punti
problema 1	21	
problema 2	9	
totale	30	

1. (a) Si consideri la seguente macchina a stati finiti:

Macchina M_1 :

- Ingresso: $U = \{0, 1\}$;
- Uscita: $V = \{0, 1\}$;
- Stati: $S = \{s_{1A}, s_{2B}, s_{3B}, s_d\}$ con s_{1A} stato iniziale;
- Transizioni (etichetta U/V):
 - transizione da s_{1A} a s_{1A} : $1/-$,
 - transizione da s_{1A} a s_{2B} : $1/0$,
 - transizione da s_{1A} a s_{3B} : $1/1$,
 - transizione da s_{1A} a s_d : $0/-$,
 - transizione da s_{2B} a s_{1A} : $0/0$,
 - transizione da s_{2B} a s_d : $0/1$,
 - transizione da s_{2B} a s_d : $1/0$,
 - transizione da s_{3B} a s_{1A} : $0/1$,
 - transizione da s_{3B} a s_d : $1/-$,
 - transizione da s_d a s_d : $-/-$.

- (b) Si disegni il diagramma di transizione della macchina M_1 . Si classifichi M_1 rispetto al determinismo.

Traccia di soluzione.

M_1 e' nondeterministica, ma non pseudo-nondeterministica.

Si noti che una macchina e' non-deterministica se esiste almeno una sequenza d'ingresso cui risponde con almeno due sequenze d'uscita diverse.

- (c) Si descriva l'algoritmo per determinizzare una macchina a stati finiti non-deterministica.

Traccia di soluzione.

Si rimanda alle dispense sulle macchine a stati finiti (Lez. 16, TAH).

Si sottolinea un errore comune: nel costruire gli stati della macchina determinizzata, si devono considerare le coppie di ingressi/uscite $(u, v) \in U \times V$ e non solo gl'ingressi $u \in U$ (come si farebbe per determinizzare un automa non deterministico).

- (d) Si determinizzi M_1 ottenendo la macchina $Det(M_1)$. Si mostri il procedimento e la macchina risultante $Det(M_1)$. Si classifichi $Det(M_1)$ rispetto al determinismo.

Traccia di soluzione.

Si mostra la tavola delle transizioni della macchina determinizzata $Det(M_1)$.

	<i>stati presenti</i>						
	(s_{1A})	(s_d)	(s_{1A}, s_{2B})	(s_{1A}, s_{3B})	(s_{1A}, s_d)	(s_{1A}, s_{2B}, s_d)	(s_{1A}, s_{3B}, s_d)
<i>U/V</i>	<i>stati futuri</i>						
0/0	(s_d)	(s_d)	(s_{1A}, s_d)	(s_d)	(s_d)	(s_{1A}, s_d)	(s_d)
0/1	(s_d)	(s_d)	(s_d)	(s_{1A}, s_d)	(s_d)	(s_d)	(s_{1A}, s_d)
1/0	(s_{1A}, s_{2B})	(s_d)	(s_{1A}, s_{2B}, s_d)	(s_{1A}, s_{2B}, s_d)	(s_{1A}, s_{2B}, s_d)	(s_{1A}, s_{2B}, s_d)	(s_{1A}, s_{2B}, s_d)
1/1	(s_{1A}, s_{3B})	(s_d)	(s_{1A}, s_{3B})	(s_{1A}, s_{3B}, s_d)	(s_{1A}, s_{3B}, s_d)	(s_{1A}, s_{3B}, s_d)	(s_{1A}, s_{3B}, s_d)

Si noti che $Det(M_1)$ e' ancora una macchina non-deterministica. E come potrebbe essere altrimenti ? Se la macchina originale puo' produrre piu' sequenze di uscita diverse in risposta a una sequenza d'ingresso e la procedura di determinizzazione ne preserva il comportamento, anche $Det(M_1)$ potra' produrre piu' sequenze di uscita diverse in risposta a una sequenza d'ingresso (che e' la definizione di macchina non-deterministica). Ma $Det(M_1)$ e' una macchina pseudo-nondeterministica (aliter deterministica rispetto all'uscita), poiche' dato un ingresso, un'uscita e uno stato presente c'e' un unico stato futuro. In altre parole, l'automa sottostante (ottenuto fondendo le etichette d'ingresso e uscita in un'unica etichetta d'ingresso) e' deterministico.

- (e) Si descriva l'algoritmo per la minimizzazione degli stati di una macchina a stati finiti non-deterministica che produce una macchina equivalente con un numero minimo di stati tra quelle bisimili.

Traccia di soluzione.

Si rimanda alle dispense sulle macchine a stati finiti (Lez. 16, TAH).

Si sottolinea un errore comune: dato che $Det(M_1)$ e' una macchina pseudo-nondeterministica, cioe' non-deterministica, si deve usare l'algoritmo per la minimizzazione non-deterministica, non quello per la minimizzazione deterministica. In particolare la versione non-deterministica richiede di applicare ripetutamente un operatore di separazione che si riferisce alle coppie di ingressi/uscite $(u, v) \in U \times V$ e non solo gl'ingressi $u \in U$ (come si farebbe per minimizzare una macchina deterministica).

- (f) Si minimizzi il numero degli stati di $Det(M_1)$ ottenendo la macchina $Min(Det(M_1))$.

Si mostri il procedimento e la macchina risultante $Min(Det(M_1))$. Si classifichi $Min(Det(M_1))$ rispetto al determinismo.

Traccia della soluzione.

Per semplificare la derivazione, si e' applicato l'algoritmo tabulare alla tavola delle transizioni.

Si parte dalla tavola

	<i>stati presenti</i>						
	(s_{1A})	(s_d)	(s_{1A}, s_{2B})	(s_{1A}, s_{3B})	(s_{1A}, s_d)	(s_{1A}, s_{2B}, s_d)	(s_{1A}, s_{3B}, s_d)
<i>U/V</i>	<i>stati futuri</i>						
0/0	(s_d)	(s_d)	(s_{1A}, s_d)	(s_d)	(s_d)	(s_{1A}, s_d)	(s_d)
0/1	(s_d)	(s_d)	(s_d)	(s_{1A}, s_d)	(s_d)	(s_d)	(s_{1A}, s_d)
1/0	(s_{1A}, s_{2B})	(s_d)	(s_{1A}, s_{2B}, s_d)	(s_{1A}, s_{2B}, s_d)	(s_{1A}, s_{2B}, s_d)	(s_{1A}, s_{2B}, s_d)	(s_{1A}, s_{2B}, s_d)
1/1	(s_{1A}, s_{3B})	(s_d)	(s_{1A}, s_{3B})	(s_{1A}, s_{3B}, s_d)	(s_{1A}, s_{3B}, s_d)	(s_{1A}, s_{3B}, s_d)	(s_{1A}, s_{3B}, s_d)

Tutti gli stati di $Det(M_1)$ sono equivalenti (da ciascuno di essi sotto qualsiasi ingresso si puo' produrre qualsiasi uscita), percio' ci si riduce alla macchina con un solo stato, diciamo (s_d) . Da cui otteniamo la macchina $Min(Det(M_1))$.

	<i>stati presenti</i>
	(s_d)
<i>U/V</i>	<i>stati futuri</i>
0/0	(s_d)
0/1	(s_d)
1/0	(s_d)
1/1	(s_d)

$Min(Det(M_1))$ e' pseudo-nondeterministica.

2. Si consideri il seguente grafo delle transizioni dei segnali (STG, Signal Transition Graph, nella letteratura in inglese):

- $V = \{v_1(a+), v_2(b+), v_3(a-), v_4(b-)\}$,
insieme dei vertici, dove accanto a ogni vertice e' indicata tra parentesi la transizione di segnale corrispondente;
- $E = \{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_1)\}$,
insieme degli archi;
- l'arco (v_4, v_1) ha un gettone.

(a) Si spieghi che cos'e' un grafo delle transizioni dei segnali e la sua interpretazione come rete di Petri.

Traccia di soluzione.

Un GTS - grafo di transizione dei segnali (STG - signal transition graph, in inglese) e' un modello formale per diagrammi temporali (che riportano le relazioni di causalita' tra i cambiamenti di livello di forme d'onda). Piu' formalmente, e' una rete di Petri interpretata dove le transizioni sono rappresentate direttamente dalle loro etichette (transizioni di segnali), e si omettono i posti (tutti i posti hanno una sola transizione in ingresso e una sola transizione in uscita - Grafo Marcato, MG - Marked Graph). I gettoni indicano lo stato iniziale del sistema.

(b) Si disegni il grafo delle transizioni dei segnali e la rete di Petri corrispondente.

Traccia di soluzione.

Si puo' rappresentare un grafo di transizione dei segnali (disegnando le transizioni come sbarrette e i posti come cerchietti), o nella sua forma semplificata (le transizioni sono rappresentate direttamente dalle etichette e si omettono i posti dato che hanno un'unica transizione in ingresso e in uscita). Qui si chiede di disegnare entrambe le rappresentazioni.

Vedi foglio allegato.

(c) Si disegni il grafo di raggiungibilit  di tale rete di Petri.

Traccia di soluzione.

Il grafo di raggiungibilit  GR (RG - reachability graph) di una rete di Petri   un sistema di transizione ST (TS - transition system) i cui stati sono gli stati raggiungibili della rete e gli eventi sono le transizioni della rete.

Vedi foglio allegato.

- (d) Codificando il grafo di raggiungibilit  si ottenga il grafo degli stati, e lo si disegni.

Traccia di soluzione.

Si puo' dire che il grafo degli stati GS (SG - state graph) e' l'interpretazione binaria del sistema di transizione che rappresenta il grafo di raggiungibilit . Ogni nodo e' etichettato con un vettore binario che riporta il valore binario di ogni segnale, e gli eventi rappresentano transizioni di segnali.

Vedi foglio allegato.

- (e) Si enuncino le propriet  di codifica unica e di codifica completa. Si verifichi se il grafo ottenuto le soddisfa.

Traccia di soluzione.

Codifica Consistente: per ogni transizione tra due stati del grafo degli stati, i due vettori binari degli stati differiscono solo per il cambio di valore del segnale che etichetta la transizione.

Codifica Unica: a ogni stato del grafo degli stati e' assegnato un codice binario unico.

E' sufficiente per derivare senza ambiguit  le funzioni di stato futuro. Non e' una condizione necessaria perche' alcuni stati nel grafo degli stati possono essere equivalenti, oppure l'ambiguit  puo' riguardare i segnali d'ingresso (che non devono essere sintetizzati, ma e' compito dell'ambiente produrre correttamente).

Codifica Completa: se due stati del grafo degli stati hanno lo stesso codice binario, essi hanno gli stessi segnali d'uscita abilitati.

Intuitivamente, la codifica completa indica che il sistema ha abbastanza memoria per ricordarsi in che stato e' (e quindi che le funzioni di stato futuro sono deterministiche e percio' realizzabili). In particolare, se la codifica e' unica, essa e' completa.

Le propriet  precedenti servono per garantire la realizzabilit  di un circuito (assumendo ovviamente anche la propriet  della limitatezza).

Limitatezza: il grafo degli stati ha un numero finito di stati, cioe' la rete di Petri originale ha un insieme di raggiungibilit  finito.

Il grafo ottenuto soddisfa la propriet  di Codifica Unica (e quindi di Codifica Completa).

- (f) Si sintetizzi il segnale di uscita b in funzione del segnale d'ingresso a , minimizzandone la funzione logica. Se ne mostri la realizzazione con porte logiche.

Si commenti il circuito ottenuto.

Traccia di soluzione.

In generale, le regioni di eccitazione $ER(x+)$ e $ER(x-)$ sono gl'insiemi di stati in cui sono abilitati rispettivamente $x+$ e $x-$; le regioni di quiescenza $QR(x+)$ e $QR(x-)$ sono gl'insiemi di stati in cui x ha il medesimo valore, 1 o 0, ed e' stabile:

$$\begin{aligned} ER(x+) &= \{s \in S \mid s \xrightarrow{x+}\}, \\ ER(x-) &= \{s \in S \mid s \xrightarrow{x-}\}, \\ QR(x+) &= \{s \in S \mid s(x) = 1 \wedge s \notin ER(x-)\}, \\ QR(x-) &= \{s \in S \mid s(x) = 0 \wedge s \notin ER(x+)\}. \end{aligned}$$

Calcolando le regioni di eccitazione $ER(b+)$, $ER(b-)$ (b cambia di livello) e di quiescenza $QR(b+)$, $QR(b-)$ (b non cambia di livello) del segnale b , si determina la funzione di stato futuro e la si minimizza (ad esempio con una mappa di Karnaugh).

$$ER(b+) = \{10\}, ER(b-) = \{01\}, QR(b+) = \{11\}, QR(b-) = \{00\}.$$

In questo caso si ottiene la realizzazione banale $b = a$, cioe' il circuito identita'.

Vedi foglio allegato.

Nota finale.

Una realizzazione corretta delle uscite richiede che si abbiano transizioni sulle uscite se e solo se l'ambiente ne e' in attesa; transizioni non attese, o il non generare transizioni attese possono produrre malfunzionamenti.

Per garantire che il circuito sia indipendente dalla velocita' (speed independent), cioe' che la sua correttezza non dipenda dai ritardi effettivi delle componenti, s'impone la condizione di persistenza delle uscite: se c'e' una coppia di segnali di cui l'uno disabilita l'altro, entrambi devono essere segnali d'ingresso.

Vale il teorema: *Un grafo di transizione dei segnali (GTS, STG) e' realizzabile come un circuito indipendente dalle velocita' se e solo se il grafo degli stati (GS, SG) e' finito, consistente, persistente rispetto alle uscite e soddisfa la proprieta' di codifica completa.*

Inoltre per garantire che il circuito sintetizzato sia indipendente dalla velocità l'ipotesi di porte complesse atomiche richiede che tutti i ritardi interni ad ogni porta siano trascurabili e non producano alcun comportamento spurio osservabile esternamente. Le porte complesse sono astrazioni di realizzazioni fisiche e dipendono dalle tecnologie disponibili. A volte si allude a questo fatto disegnando le porte logiche come tutte "appiccicate" senza fili interni, per significare che si vorrebbe una realizzazione a porte complesse senza ritardi interni. Tale problematica esula dal nostro corso.