

Sistemi a Tempo Reale

Informatica - Tiziano Villa

26 Gennaio 2007

Nome e Cognome:

Matricola:

Posta elettronica:

problema	punti massimi	i tuoi punti
problema 1	7	
problema 2	7	
problema 3	7	
problema 4	9	
totale	30	

Avvertenza: le tracce di soluzione proposte vogliono chiarire i punti essenziali o sottili, e non costituiscono risposte complete. Si omettono risposte direttamente deducibili dalle dispense del corso.

1. (a) Si definisca la nozione di equivalenza tra macchine a stati finiti deterministiche M_1 e M_2 .

Traccia di risposta.

Per ogni segnale d'ingresso (segnale = successione d'ingressi) M_1 e M_2 producono lo stesso segnale d'uscita.

- (b) Si definisca la nozione di bisimulazione tra macchine a stati finiti deterministiche.

Traccia di risposta.

Esiste una relazione R tale che gli stati iniziali di M_1 e M_2 sono nella relazione R e, per tutti gli stati p di M_1 e q di M_2 nella relazione R , per ogni ingresso, p e q producono la medesima uscita e i loro stati futuri sono anch'essi nella relazione R .

- (c) Si enunci il teorema sulla relazione tra equivalenza e bisimulazione per macchine a stati finiti deterministiche.

Traccia di risposta.

Due macchine a stati finiti deterministiche M_1 e M_2 sono equivalenti se e solo se c'è una bisimulazione tra M_1 e M_2 .

(d) Si considerino le due macchine a stati finiti seguenti:

Macchina M' :

- stati: s'_1, s'_2 con s'_1 stato iniziale;
- transizione da s'_1 a s'_1 : \bullet/\perp ,
transizione da s'_1 a s'_2 : $\bullet/0$,
transizione da s'_2 a s'_2 : $\bullet/\perp, \bullet/0, \bullet/1$.

Macchina M'' :

- stati: s''_1, s''_2 con s''_1 stato iniziale;
- transizione da s''_1 a s''_1 : $\bullet/0, \bullet/\perp$,
transizione da s''_1 a s''_2 : $\bullet/0$,
transizione da s''_2 a s''_2 : $\bullet/\perp, \bullet/0, \bullet/1$.

Si risponda in ordine alle seguenti domande (si indichi sempre il numerale romano in ogni risposta):

- i. Si disegnino i diagrammi di transizione delle due macchine.
- ii. Si classifichino le macchine rispetto al determinismo.

Traccia di risposta.

M' e' pseudo-nondeterministica.

M'' e' nondeterministica, ma non pseudo-nondeterministica.

- iii. Si derivino i comportamenti (successioni d'ingressi/successioni d'uscite) prodotti dalle due macchine e li si confrontino.

Traccia di risposta.

Per descrivere i comportamenti si possono usare le espressioni regolari, es. le uscite di M' sono $\perp^*0\{\perp, 0, 1\}^*$ e le uscite di M'' sono $\{0, \perp\}^*0\{\perp, 0, 1\}^*$. $\text{Comportamenti}(M') = \text{Comportamenti}(M'')$. Per dimostrarlo uno potrebbe riscrivere le espressioni regolari sfruttando identita' valide per esse oppure sfruttare il risultato dei punti successivi (vedi ultima risposta).

- iv. Si trovi una simulazione di M' da parte di M'' , se esiste.

Traccia di risposta.

M'' simula M' come mostrato dalla relazione $R = \{(s'_1, s''_1), (s'_2, s''_2)\}$.

- v. Si trovi una simulazione di M'' da parte di M' , se esiste.

Traccia di risposta.

M' simula M'' come mostrato dalla relazione $R = \{(s''_1, s'_1), (s''_1, s'_2), (s''_2, s'_2)\}$.

vi. Si trovi una bisimulazione tra le due macchine, se esiste.

Traccia di risposta.

Non c'è una bisimulazione perché l'unione delle due precedenti relazioni non è simmetrica, dato che non è presente la coppia (s_2', s_1'') .

vii. Si mostri un esempio di due macchine a stati finiti minimizzate equivalenti, ma non isomorfe.

Traccia di risposta.

Le due macchine M' e M'' sono un tale esempio.

viii. Si commentino i risultati precedenti.

Traccia di risposta.

Le due macchine M' e M'' sono un esempio di macchine a stati finiti nondeterministiche equivalenti e minimizzate, ma non isomorfe; in altri termini, esse mostrano che non esiste un'unica macchina a stati finiti nondeterministica che realizza il sistema originale con il minimo numero di stati.

Si noti che dal fatto che M' simula M'' e M'' simula M' non si può dedurre che c'è una bisimulazione tra M' e M'' (perché non è detto che valga la simmetria, come non vale nell'esempio precedente), mentre ovviamente l'esistenza di una bisimulazione tra M' e M'' implica che M' simula M'' e M'' simula M' .

Dal fatto che M' simula M'' e viceversa consegue che M' è equivalente a M'' (è una condizione sufficiente, ma non necessaria), e questo dimostra l'affermazione di equivalenza al punto iii. che sarebbe stata ben più difficile da dimostrare affrontandola come un problema di equivalenza tra espressioni regolari (trasformandole con delle identità). Trattandosi di una condizione solo sufficiente, dal fatto che non fosse vero che M' simula M'' o M'' simula M' non si potrebbe dedurre che M' non è equivalente a M'' .

.

.

2. Si consideri il seguente automa temporizzato:

- locazioni: l_1, l_2 , dove l_1 e' la locazione iniziale con condizioni iniziali $s(0) := 0, r(0) := 0$;
- dinamica della locazione l_1 : $\dot{s}(t) = 1, \dot{r}(t) = 1$,
dinamica della locazione l_2 : $\dot{s}(t) = 1, \dot{r}(t) = 1$;
- transizione da l_1 a l_2 : $a/s(t), r(t) := 0$,
transizione da l_2 a l_1 : $b/s(t), r(t) := 0$,
dove $a = \{(r(t), s(t)) \mid r(t) = 1\}$,
dove $b = \{(r(t), s(t)) \mid r(t) = 2\}$
(la sintassi delle annotazioni di una transizione e' *guardia/uscita, azione*);
- uscita $y(t) \in Interi \cup \{assente\}$ con $y(t) := s(t)$.

(a) Si disegni il diagramma di transizione degli stati dell'automata, annotando con precisione locazioni e transizioni, e si descriva l'uscita $y(t)$.

Traccia di risposta.

L'automata genera una successione di eventi $\{1, 3, 4, 6, 7, 9, 10, \dots\}$ ai tempi $\{1, 3, 4, 6, 7, 9, 10, \dots\}$. In altri termini, il valore di ogni evento in uscita e' uguale al tempo in cui e' prodotto, e gli intervalli tra eventi si alternano con durate di uno e due secondi; altrimenti l'uscita vale *assente*.

- (b) Si assumo che ci sia un nuovo ingresso $u : Reali \rightarrow Ingresso$ con alfabeto $Ingresso = \{ricomincia, assente\}$ tale che quando il nuovo ingresso abbia valore *ricomincia* il sistema ibrido ricominci da capo ritornando allo stato iniziale.

Si ridesigni il diagramma precedente introducendo questo nuovo segnale e aggiornando le annotazioni.

Traccia di risposta.

- locazioni: l_1, l_2 , dove l_1 e' la locazione iniziale con condizioni iniziali $s(0) := 0, r(0) := 0$;
- dinamica della locazione l_1 : $\dot{s}(t) = 1, \dot{r}(t) = 1$,
dinamica della locazione l_2 : $\dot{s}(t) = 1, \dot{r}(t) = 1$;
- transizione da l_1 a l_2 : $a/s(t), r(t) := 0$,
transizione da l_1 a l_1 : $c/assente, s(t) := 0, r(t) := 0$,
transizione da l_2 a l_1 : $b/s(t), r(t) := 0$,
transizione da l_2 a l_1 : $c/assente, s(t) := 0, r(t) := 0$,
dove $a = \{(r(t), s(t), u(t)) \mid r(t) = 1 \wedge u(t) = assente\}$,
dove $b = \{(r(t), s(t), u(t)) \mid r(t) = 2 \wedge u(t) = assente\}$
dove $c = \{(r(t), s(t), u(t)) \mid u(t) = ricomincia\}$
(la sintassi delle annotazioni di una transizione e' *guardia/uscita, azione*);
- uscita $y(t) \in Interi \cup \{assente\}$ con $y(t) := s(t)$.

3. (a) Si descriva l'algoritmo di assegnamento RMS (Rate Monotonic Scheduling), indicandone le ipotesi di validità e se ne discuta l'ottimalità.

- (b) Dati i seguenti processi periodici che eseguono su una unita' di calcolo singola, si mostri un assegnamento secondo l'algoritmo RMS (Rate Monotonic Scheduling), se esiste. Si supponga che scadenza e periodo coincidano.

processo	durata esecuzione	scadenza
P1	1	4
P2	2	6
P3	3	12

Traccia di risposta.

P1	X	-	-	-		X	-	-	-		X	-	-	-	
P2	-	X	X	-	-	-		X	X	-	-	-	-	-	
P3	-	-	-	X	-	X	-	-	-	X	-	-	-	-	

Si noti che non e' soddisfatta la maggiorazione $U = \sum (c_i/p_i) = 1/4 + 2/6 + 3/12 = 10/12 \sim 0.83 \leq 3 \cdot (2^{1/3} - 1) \sim 0,78$, ma essa e' solo una condizione sufficiente e in questo caso esiste un assegnamento senza che essa sia verificata.

4. (a) Si progetti un semplice processore generico che realizza il seguente insieme di istruzioni:

```
MOV1 Rn, direct
MOV2 direct, Rn
MOV3 @Rn, Rm
MOV4 Rn, #Immed
ADD  Rn, Rm
SUB  Rn, Rm
JZ   Rn, relative
```

- i. Si scriva un simulatore in C del precedente insieme d'istruzioni per definirne la semantica.

ii. Si traduca nel precedente linguaggio macchina l'algoritmo per calcolare l'ennesimo numero di Fibonacci.

Esempio di soluzione (dovuta a Rafael Lopez, Tony Givargis e Randy January):

```
#0  00110000000000001 R0 <-- 1 caso terminale
                                R0 salva Fib(n-1)
#1  00110001000000001 R1 <-- 1 caso terminale
                                R1 salva Fib(n-2)
#2  00110010000000010 R2 <-- 2
                                numero casi terminali
#3  00110011000000000 R3 <-- 0 fisso a zero
#4  00110100000000001 R4 <-- 1
                                passo decremento argomento
#5  0011010100001010 R5 <-- 10 argomento
#6  0101010100100000 R5 <-- R5 - R2
                                decremento argomento
                                per i due casi terminali
#7  0110010100001010 se R5=0 PC <-- PC + 10
                                se finito esci
#8  00010000000000000 M[0] <-- R0
#9  00000110000000000 R6 <-- M[0]
                                [#8 e #9 insieme: R6 <-- R0 ]
                                aggiorna R6 (inutile dopo
                                i casi terminali)
#10 0100011000010000 R6 <-- R6 + R1
                                Fib(n) = Fib(n-1)+Fib(n-2)
#11 00010000000000001 M[1] <-- R0
#12 00000001000000001 R1 <-- M[1]
                                [#11 e #12 insieme: R1 <-- R0 ]
                                aggiorna R1 al nuovo valore di
                                Fib(n-2)
#13 00010110000000000 M[1] <-- R6 [insieme sono
#14 00000000000000000 R0 <-- M[1]   R0 <-- R6 ]
                                [#11 e #12 insieme: R0 <-- R6
                                aggiorna R0 al nuovo valore di
                                Fib(n-1)
#15 0101010101000000 R5 <-- R5 - R4
                                decrementa argomento
#16 0110001111110110 se R3=0 PC <-- PC - 10
                                salto incondizionato
#17 00000000000000000 R0 <-- M[0]
                                mai raggiunta
```

- iii. Si progetti una macchina a stati finiti estesa che modella l'esecuzione dell'insieme d'istruzioni, disegnandone il diagramma delle transizioni annotato con le operazioni sui dati.

iv. Si derivi il controllore dalla precedente macchina a stati finiti estesa.

- v. Si derivi l'unita' esecutiva dalla precedente macchina a stati finiti estesa.