

Sistemi - Modulo di Sistemi a Eventi Discreti
Discrete Event and Hybrid Systems

Laurea Magistrale in Ingegneria e Scienze Informatiche
Tiziano Villa

21 Febbraio 2022

Nome e Cognome:

Matricola:

Posta elettronica:

problema	punti massimi	i tuoi punti
problema 1	18	
problema 2	12	
totale	30	

1. (a) Si definisca la nozione di macchina a stati finiti nondeterministica progressiva.

Define a non-deterministic completely-specified finite-state machine.

Traccia di soluzione.

Si vedano le dispense per la definizione formale. Si noti che progressiva significa che l'evoluzione e' definita per ogni ingresso, cioe' la funzione di transizione e' definita come

$$Stati \times Ingressi \longrightarrow P(Stati \times Uscite) \setminus \emptyset$$

dove P rappresenta l'insieme potenza. La clausola " $\setminus \emptyset$ " impone che sia progressiva.

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

Define simulation between two non-deterministic finite-state machines.

Traccia di soluzione.

Si vedano le dispense per la definizione formale. Intuitivamente: ogni stato iniziale di M_1 e' simulato da uno stato iniziale di M_2 , e se uno stato p di M_1 e' simulato da uno stato q di M_2 , per ogni ingresso e per ogni stato futuro p' di p c'e' uno stato futuro q' di q tali che p' e q' producono la medesima uscita e p' e' simulato da q' .

- (c) Si enunci il teorema sulla relazione tra raffinamento e simulazione di una macchina a stati finiti nondeterministica da parte di una macchina a stati finiti deterministica.

State the theorem on the relation between refinement and simulation of a non-deterministic finite-state machine by a deterministic finite-state machine.

Traccia di soluzione.

Se M_2 e' deterministica, M_1 e' simulata da M_2 se e solo se M_1 e' equivalente a M_2 (ed e' equivalente se e solo se M_1 raffina M_2 e M_2 raffina M_1).

If M_2 is deterministic, M_1 is simulated by M_2 if and only if M_1 is equivalent to M_2 (and it is equivalent if and only if M_1 refines M_2 and M_2 refines M_1).

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

Consider the two following finite-state machines:

Macchina (Machine) M' :

- stati (states): s'_1, s'_2, s'_3 con s'_1 stato iniziale (initial state);
- transizione da (edge from) s'_1 a (to) s'_3 : $\bullet/1$,
transizione da (edge from) s'_3 a (to) s'_3 : $\bullet/\perp, \bullet/0, \bullet/1$,
transizione da (edge from) s'_1 a (to) s'_1 : \bullet/\perp ,
transizione da (edge from) s'_1 a (to) s'_2 : $\bullet/0$,
transizione da (edge from) s'_2 a (to) s'_2 : $\bullet/\perp, \bullet/0, \bullet/1$.

Macchina (Machine) M'' :

- stati (states): s''_1, s''_2, s''_3 con s''_1 stato iniziale (initial state);
- transizione da (edge from) s''_1 a (to) s''_3 : $\bullet/1$,
transizione da (edge from) s''_3 a (to) s''_3 : $\bullet/\perp, \bullet/0, \bullet/1$,
transizione da (edge from) s''_1 a (to) s''_1 : $\bullet/0, \bullet/\perp$,
transizione da (edge from) s''_1 a (to) s''_2 : $\bullet/0$,
transizione da (edge from) s''_2 a (to) s''_2 : $\bullet/\perp, \bullet/0, \bullet/1$.

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

Answer in order the following questions (specify the Roman numeral in every answer)

i. Si disegnino i diagrammi di transizione delle due macchine.

Draw the graphs of the two finite-state machines.

ii. Si classifichino le macchine rispetto al determinismo.

Classify the two finite-state machines with respect to determinism.

Traccia di soluzione.

M' e' pseudo-nondeterministica.

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

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

Find a simulation of M' by M'' , if it exists.

Traccia di soluzione.

M'' simula M' come mostrato dalla relazione

$$R_{M'-M''} = \{(s'_1, s''_1), (s'_2, s''_2), (s'_2, s''_1), (s'_3, s''_3), (s'_2, s''_3)\}.$$

Si noti che per simulare la transizione di M' da s'_1 a s'_2 sotto $\bullet/0$ la macchina M'' puo' scegliere la transizione da s''_1 a s''_2 oppure quella da s''_1 a s''_1 ; nel primo caso s'introduce nella relazione la coppia (s'_2, s''_2) ,

nel secondo caso s'introduce la coppia (s'_2, s''_1) e poi per conseguenza la coppia (s'_2, s''_3) . Si potrebbe scegliere una simulazione minima mettendoci nel primo o nel secondo caso, ma decidiamo d'inserire tutte e tre tali coppie per costruire una simulazione piu' grande.

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

Find a simulation of M'' by M' , if it exists.

Traccia di soluzione.

M' simula M'' come mostrato dalla relazione

$$R_{M''-M'} = \{(s''_1, s'_1), (s''_1, s'_2), (s''_2, s'_2), (s''_3, s'_3), (s''_3, s'_2)\}.$$

Si noti che per simulare la transizione di M'' da s''_1 a s''_1 sotto $\bullet/0$ la macchina M' deve usare la transizione da s'_1 a s'_2 che introduce la coppia (s''_1, s'_2) e per conseguenza la coppia (s''_3, s'_2) .

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

Find a bisimulation between the two finite-state machines, if it exists.

Traccia di soluzione.

Una bisimulazione e' data dall'unione delle due precedenti relazioni, che e' simmetrica: $R_{M'-M''} \cup R_{M''-M'} = \{(s'_1, s''_1), (s'_2, s''_2), (s'_2, s''_1), (s'_3, s''_3), (s'_2, s''_3), (s'_1, s''_1), (s'_1, s''_2), (s'_2, s''_2), (s'_3, s''_3), (s'_3, s''_2)\}.$

- vi. Si determinizzi la macchina M'' e si mostri il diagramma di transizione della macchina deterministica cosi' trovata $det(M'')$.

Determinize the finite-state machine M'' and show the graph of the finite-state machine $det(M'')$ obtained by determinization.

Traccia di soluzione.

Macchina $det(M'')$:

- stati: $\{s''_1\}, \{s''_1, s''_2\}, \{s''_2, s''_3\}, \{s''_3\}$ con $\{s''_1\}$ stato iniziale;
- transizione da $\{s''_1\}$ a $\{s''_1\}$: \bullet/\perp ,
transizione da $\{s''_1\}$ a $\{s''_1, s''_2\}$: $\bullet/0$,
transizione da $\{s''_1, s''_2\}$ a $\{s''_1, s''_2\}$: $\bullet/0, \bullet/\perp$,
transizione da $\{s''_1\}$ a $\{s''_3\}$: $\bullet/1$,
transizione da $\{s''_3\}$ a $\{s''_3\}$: $\bullet/\perp, \bullet/0, \bullet/1$,
transizione da $\{s''_1, s''_2\}$ a $\{s''_2, s''_3\}$: $\bullet/1$,
transizione da $\{s''_2, s''_3\}$ a $\{s''_2, s''_3\}$: $\bullet/\perp, \bullet/0, \bullet/1$.

- vii. Si trovi una simulazione di M' da parte di $det(M'')$, se esiste.

Find a simulation of M' by $det(M'')$, if it exists.

Traccia di soluzione.

$det(M'')$ simula M' come mostrato dalla relazione

$$R_{M' - \det(M'')} = \{(s'_1, \{s''_1\}), (s'_2, \{s''_1, s''_2\}), (s'_2, \{s''_2, s''_3\}), (s'_3, \{s''_3\})\}.$$

viii. Si trovi una simulazione di $\det(M'')$ da parte di M' , se esiste.

Find a simulation of $\det(M'')$ by M' , if it exists.

Traccia di soluzione.

M' simula $\det(M'')$ come mostrato dalla relazione

$$R_{\det(M'') - M'} = \{(\{s''_1\}, s'_1), (\{s''_1, s''_2\}, s'_2), (\{s''_2, s''_3\}, s'_2), (\{s''_3\}, s'_3)\}.$$

ix. Si trovi una bisimulazione tra le due macchine M' e $\det(M'')$, se esiste.

Find a bisimulation between the two finite-state machines M' and $\det(M'')$, if it exists.

Traccia di soluzione.

L'unione delle precedenti relazioni $R_{M' - \det(M'')} \cup R_{\det(M'') - M'} =$

$\{(s'_1, \{s''_1\}), (s'_2, \{s''_1, s''_2\}), (s'_2, \{s''_2, s''_3\}), (s'_3, \{s''_3\}), (\{s''_1\}, s'_1), (\{s''_1, s''_2\}, s'_2), (\{s''_2, s''_3\}, s'_2), (\{s''_3\}, s'_3)\}$ e' simmetrica, quindi costituisce una bisimulazione tra M' e $\det(M'')$.

x. Si commentino i risultati precedenti.

Write down any appropriate comment on the previous results.

Traccia di soluzione.

M' e M'' sono esempi di macchine a stati finiti minimizzate equivalenti (M' raffina M'' e M'' raffina M'), e pure bisimili (date le ipotesi non era scontato che fossero anche bisimili, ma in questo esempio si da' il caso che lo siano). Si noti che minimizzando gli stati di M' e M'' si ottiene in entrambi i casi una medesima macchina costituita da un solo stato con i tre auto-anelli per ogni coppia d'ingressi/uscite.

Determinizzando gli stati di M'' si ottiene una macchina $\det(M'')$ pseudo-nondeterministica equivalente a M'' . Ne consegue che anche M' e $\det(M'')$ sono bisimili, poiche' macchine pseudo-nondeterministiche sono equivalenti se e solo se sono bisimili.

Si noti che minimizzando gli stati di $\det(M'')$ si ottiene una macchina a stati finiti isomorfa a M' (in $\det(M'')$ gli stati $\{s''_1, s''_2\}$ e $\{s''_2, s''_3\}$ sono equivalenti).

2. Si consideri il sistema a eventi discreti G dove $\mathcal{L}(G) = \overline{a^*ba^*}$, $\mathcal{L}_m(G) = a^*ba^*$. Sia $E_{uc} = \{b\}$.

Sia dato il linguaggio ammissibile $L_{am} = \{a^m ba^n : m \geq n \geq 0\}$.

Consider the discrete event system G where $\mathcal{L}(G) = \overline{a^*ba^*}$, $\mathcal{L}_m(G) = a^*ba^*$. Let $E_{uc} = \{b\}$.

Take the admissible language to be $L_{am} = \{a^m ba^n : m \geq n \geq 0\}$.

- (a) Il linguaggio $\overline{L_{am}}$ e' regolare ?

Is the language $\overline{L_{am}}$ regular ?

Traccia di soluzione.

No. Bisogna contare le "a".

- (b) Dato un impianto G e un supervisore S si scriva la definizione di supervisore S non-bloccante per G .

Given a plant G and a supervisor S write the definition of non-blocking supervisor S for G .

Traccia di soluzione.

$$\overline{\mathcal{L}_m(S/G)} = \mathcal{L}(S/G).$$

- (c) Si enunci il teorema di esistenza di un supervisore non-bloccante S tale che $\mathcal{L}_m(S/G) = L_{am}$.

State the theorem for the existence of a non-blocking supervisor S such that $\mathcal{L}_m(S/G) = L_{am}$.

Traccia di soluzione.

Sia dato il sistema a eventi discreti $G = (X, E, f, \Gamma, x_o, X_m)$, dove $E_{uc} \subseteq E$ sono gli eventi incontrollabili (per cui $E_c = E \setminus E_{uc}$). Si consideri il linguaggio $L_{am} \subseteq \mathcal{L}_m(G)$, dove $L_{am} \neq \emptyset$. Esiste un supervisore non-bloccante S per G tale che

$$\mathcal{L}_m(S/G) = L_{am}, \quad \mathcal{L}(S/G) = \overline{L_{am}}$$

se e solo se le due condizioni seguenti valgono:

- i. L_{am} e' controllabile rispetto a $\mathcal{L}(G)$ e E_{uc} , cioe' $\overline{K}E_{uc} \cap \mathcal{L}(G) \subseteq \overline{K}$.
- ii. L_{am} e' $\mathcal{L}_m(G)$ -chiuso, cioe' $L_{am} = \overline{L_{am}} \cap \mathcal{L}_m(G)$.

- (d) Applicando il precedente teorema di esistenza, si verifichi se esiste un supervisore non-bloccante.

Se esiste, si verifichi che non e' bloccante applicando la definizione di supervisore non-bloccante.

Applying the previous theorem for the existence of a non-blocking supervisor. verify whether a non-blocking supervisor exists.

If it exists, verify that it is non-blocking using the definition of non-blocking supervisor.

Traccia di soluzione.

La controllabilita' di L_{am} e' verificata: si disabilita "a" (evento controllabile) dopo che il numero di "a" che seguono "b" e' pari a quello di "a" che precedono "b".

Quindi per il teorema di controllabilita' esiste un supervisore S tale che $\mathcal{L}(S/G) = \overline{L_{am}}$.

Inoltre L_{am} soddisfa la $\mathcal{L}_m(G)$ -chiusura: $L_{am} = \overline{L_{am}} \cap \mathcal{L}_m(G)$, che e' vera per costruzione poiche'

$$\overline{L_{am}} \equiv \{a^m b a^n : m \geq n \geq 0\} = \overline{\{a^m b a^n : m \geq n \geq 0\}} \cap a^* b a^* \equiv \overline{L_{am}} \cap \mathcal{L}_m(G).$$

Da cui si ottiene: $\mathcal{L}_m(S/G) = \mathcal{L}(S/G) \cap \mathcal{L}_m(G) = \overline{L_{am}} \cap \mathcal{L}_m(G) = L_{am}$, quindi esiste un supervisore S tale che $\mathcal{L}_m(S/G) = L_{am}$.

Tale supervisore e' non bloccante in quanto soddisfa la definizione $\overline{\mathcal{L}_m(S/G)} = \mathcal{L}(S/G)$, come si vede da $\overline{\mathcal{L}_m(S/G)} = \overline{L_{am}} = \mathcal{L}(S/G)$.

Mettendo tutto insieme: esiste un supervisore S non bloccante tale che $\mathcal{L}(S/G) = \overline{L_{am}}$ e $\mathcal{L}_m(S/G) = L_{am}$.

- (e) Esiste una realizzazione a stati finiti del supervisore S , cioe' un automa a stati finiti R che marca $\overline{L_{am}}$ e quindi tale che $\mathcal{L}_m(R) = \mathcal{L}(R) = \overline{L_{am}}$?

Se no, si motivi la risposta. Se si, si mostri tale realizzazione.

Is there a finite-state realization of the supervisor S , i.e., a finite-state automaton R which marks $\overline{L_{am}}$ and so it is such that $\mathcal{L}_m(R) = \mathcal{L}(R) = \overline{L_{am}}$?

If not, explain the negative answer. If yes, show such realization.

Traccia di soluzione.

No. Il supervisore deve generare e marcare il linguaggio $\overline{L_{am}}$, per cui deve avere un numero infinito di stati.

(f) Esiste una realizzazione con una rete di Petri del supervisore S ?

Se no, si motivi la risposta. Se si, si mostri tale realizzazione.

Does there exist a realization of the supervisor S by a Petri net ?

If not, explain the negative answer. If yes, show such realization.

Traccia di soluzione.

Si.

La seguente rete di Petri P_{sup} realizza il supervisore S .

Si consideri la rete di Petri P_{sup} definita da:

- $P = \{p_1, p_2, p_3\}$
- $T = \{t_1, t_2, t_3\}$
- $A = \{(p_1, t_1), (p_1, t_2), (p_2, t_3), (p_3, t_2), (p_3, t_3), (t_1, p_1), (t_1, p_3), (t_2, p_2), (t_3, p_2)\}$
- $\forall i, j \ w(p_i, t_j) = 1$
- $\forall i, j \ w(t_i, p_j) = 1$, tranne che $w(t_1, p_2) = 2$

Sia $x_0 = [1, 0, 1]$ la marcatura iniziale.

Per associare un linguaggio a una rete di Petri s'introduce un insieme di $l : T \rightarrow E$, e un insieme di stati che accettano $X_m \subseteq N^n$ (n e' il numero di posti) per il linguaggio marcato.

Si assuma che alle transizioni t_1 e t_3 sia associato l'evento a , e che a t_2 sia associato l'evento b .

Una variante piu' semplice e' la seguente:

- $P = \{p_1, p_2, p_3\}$
- $T = \{t_1, t_2, t_3\}$
- $A = \{(p_1, t_1), (p_1, t_2), (p_2, t_3), (p_3, t_3), (t_1, p_1), (t_1, p_3), (t_2, p_2), (t_3, p_2)\}$
- $\forall i, j \ w(p_i, t_j) = 1$
- $\forall i, j \ w(t_i, p_j) = 1$

Sia $x_0 = [1, 0, 0]$ la marcatura iniziale.

Ed altre varianti ancora ...