

Sistemi - Modulo di Sistemi a Eventi Discreti

Discrete Event and Hybrid Systems

Laurea Magistrale in Ingegneria Informatica per Robotica e Industria Intelligente
Master's degree in Computer Engineering for Robotics and Smart Industry
Tiziano Villa

6 Febbraio 2023

Nome e Cognome:

Matricola:

Posta elettronica:

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

1. Si consideri un impianto G con $E = \{a, b\}$, $E_{uc} = \{b\}$, $L(G) = \overline{a^*ba^*}$ (cioe' il linguaggio ottenuto dai prefissi delle stringhe dell'espressione regolare a^*ba^*), $L_m(G) = a^*ba^*$.

Si supponga che la specifica (il linguaggio generato desiderato) sia $K = \overline{\{a^kba^k, k \geq 0\}} \subseteq L(G)$, cioe' si richiede che l'impianto controllato generi prefissi di stringhe con un numero uguale di a che precedono e seguono un unico b .

- (a) Il linguaggio K e' controllabile ? Si enunci la definizione di controllabilita' di un linguaggio e la si applichi a questo caso.

Traccia della soluzione.

Definizione di controllabilita':

Siano K e $M = \overline{M}$ linguaggi sull'alfabeto di eventi E , con $E_{uc} \subseteq E$. Si dice che K e' controllabile rispetto a M e E_{uc} , se per tutte le stringhe $s \in \overline{K}$ e per tutti gli eventi $\sigma \in E_{uc}$ si ha

$$s\sigma \in M \Rightarrow s\sigma \in \overline{K}.$$

[equivalente a $\overline{K}E_{uc} \cap M \subseteq \overline{K}$]

Si applichi la definizione al nostro esempio dove $M = L(G)$. Si consideri una stringa $s \in \overline{K} = K$,

- se $s = a^*$ e quindi $s\sigma = sb = a^*b \in L(G)$, allora $s\sigma = sb \in \overline{K}$, altrimenti
- se $s \neq a^*$ allora $s\sigma = sb \notin L(G)$ (cioe' se s ha la forma $s = a^kba^l$ ($k \geq l$), allora $sb = a^kba^lb \notin L(G)$, poiche' le parole in $L(G)$ non possono contenere un secondo evento b dopo il primo).

Percio' non esiste una stringa $s \in \overline{K}$ tale che $s\sigma = sb \in L(G) \setminus \overline{K}$, cioe' K e' controllabile.

- (b) Si enunci il teorema di esistenza di un supervisore sotto controllabilita' limitata. In questo esempio, esiste un supervisore S tale che l'impianto controllato generi il linguaggio K ? Se tale supervisore S esiste, lo si descriva enunciando la sua strategia di controllo.

Se tale supervisore S esiste, c'e' una realizzazione di S per mezzo di una automa a stati finiti ? Se tale implementazione esiste, si disegni l'automa a stati finiti corrispondente.

Traccia della soluzione.

Teorema di esistenza di un supervisore:

Siano $G = (X, E, f, \gamma, x_0)$ un impianto, $E_{uc} \subseteq E$ gli eventi incontrollabili, $K \subseteq L(G)$, $K \neq \emptyset$, la specifica. Esiste un supervisore S tale che $L(S/G) = \overline{K}$ se e solo se

$$\overline{K}E_{uc} \cap L(G) \subseteq \overline{K}.$$

Un qualsiasi supervisore S con $L_m(S) = L(S) = \overline{K}$ induce un impianto controllato che genera il linguaggio K , cioè $L(S/G) = \overline{K}$.

In questo caso la strategia del supervisore è semplicemente quella di disabilitare l'evento a dopo ogni stringa del tipo $a^k b a^k$.

Si noti che il linguaggio della specifica non è regolare, quindi non c'è un automa a stati finiti che lo rappresenti. Il fatto che la specifica sia controllabile è indipendente dal fatto che sia realizzabile mediante un automa a stati finiti. In questo caso esiste una strategia di controllo - quella di disabilitare sempre a dopo che l'impianto ha prodotto una parola con un numero di a dopo b pari al numero di a prima di b -, ma non esiste un automa a stati finiti che la realizzi (in questo caso non esiste perché tale automa dovrebbe contare se sono uguali i numeri di eventi a prima e dopo l'evento b senza un limite prestabilito sul numero di eventi a).

English version.

Consider a plant G with $E = \{a, b\}$, $E_{uc} = \{b\}$, $L(G) = \overline{a^*ba^*}$ (i.e., the language given by the prefixes of the strings of the regular expression a^*ba^*), $L_m(G) = a^*ba^*$.

Suppose that the specification (the desired generated language) be $K = \overline{\{a^kba^k, k \geq 0\}} \subseteq L(G)$, i.e., it is required that the controlled plant generates prefixes of strings with an equal number of a s before and after a unique occurrence of b .

- (a) Is the language K controllable ? State precisely the definition of controllable language and apply it to this example.

Hints for the solution.

Definition of controllability:

Let K and $M = \overline{M}$ be languages over the event alphabet E , with $E_{uc} \subseteq E$. K is by definition controllable with respect to M and E_{uc} , if for all the strings $s \in \overline{K}$ and for all the events $\sigma \in E_{uc}$ it holds

$$s\sigma \in M \Rightarrow s\sigma \in \overline{K}.$$

[equivalent to $\overline{K}E_{uc} \cap M \subseteq \overline{K}$]

Let us apply the previous definition to our example where $M = L(G)$. Consider a string $s \in \overline{K} = K$,

- if $s = a^*$ and so $s\sigma = sb = a^*b \in L(G)$, then $s\sigma = sb \in \overline{K}$, otherwise
- if $s \neq a^*$ then $s\sigma = sb \notin L(G)$ (i.e., if s has the form $s = a^kba^l$ ($k \geq l$) then $sb = a^kba^lb \notin L(G)$, since the words in $L(G)$ cannot contain a second event b after the first one).

Therefore there is no string $s \in \overline{K}$ such that $s\sigma = sb \in L(G) \setminus \overline{K}$, i.e., K is controllable.

- (b) State precisely the theorem of existence of a supervisor under limited controllability.

In this example, does a supervisor S exist such that the controlled plant generates the language K ? If such supervisor S exists, describe it by stating its control strategy.

If such supervisor S exists, is there an implementation of S by means of a finite automaton ? If such an implementation exists, draw the corresponding finite automaton.

Hints for the solution.

Theorem on the existence of a supervisor:

Let $G = (X, E, f, \gamma, x_0)$ be a plant, $E_{uc} \subseteq E$ be the uncontrollable events, and $K \subseteq L(G)$, $K \neq \emptyset$, be the specification. There is a supervisor S such that $L(S/G) = \overline{K}$ if and only if

$$\overline{K}E_{uc} \cap L(G) \subseteq \overline{K}.$$

Any supervisor S with $L_m(S) = L(S) = \overline{K}$ induces a controlled plant that generates the language K , i.e., $L(S/G) = \overline{K}$.

In our case the supervisor strategy is simply to disable event a after every string of the form a^kba^k .

Pay attention that the language of the specification is not regular, therefore there is no finite state automaton that realizes it. The fact that the specification is controllable is independent from the fact it is realizable by a finite state automaton. In our example, the control strategy exists - the one that disables always a after that the plant has produced a word with a number of a s after b equal to the number of a s before b -, but there is no finite state automaton that realizes such strategy (such finite state automaton does not exist because it should be able to count to establish if the numbers of events a s before and after the event b are the same, without a predefined bound on the number of events a).

2. Una rete di Petri marcata e' specificata da una quintupla: $\{P, T, A, w, x\}$, dove P sono i posti, T le transizioni, A gli archi, w la funzione di peso sugli archi, e x il vettore di marcamento (numero di gettoni per posto). $I(t_i)$ indica l'insieme dei posti in ingresso alla transizione t_i , $O(t_j)$ indica l'insieme dei posti in uscita dalla transizione t_j .

Per associare un linguaggio a una rete di Petri s'introduce un insieme di eventi E , una funzione che etichetta le transizioni con eventi $l : T \rightarrow E$, e un insieme di stati accettanti (o marcati o finali) $X_m \subseteq N^n$ (n e' il numero di posti).

Si consideri la rete di Petri P_8 definita da:

- $P = \{p_1, p_2\}$
- $T = \{t_1, t_2\}$
- $A = \{(p_1, t_2), (p_2, t_2), (t_1, p_1), (t_2, p_2)\}$
- $w(p_1, t_2) = 1, w(p_2, t_2) = 1, w(t_1, p_1) = 1, w(t_2, p_2) = 1$

- (a) Si disegni il grafo della rete di Petri P_8 con la marcatura iniziale $x_1 = [0, 1]$.
- (b) Si associ a P_8 un linguaggio basato sul seguente alfabeto degli eventi $\{a, d\}$, con $l(t_1) = a$ (cioe', l'evento a e' associato alla transizione t_1) e $l(t_2) = d$.

Si descriva il linguaggio accettato da P_8 .

Si costruisca un automa a stati finiti che riconosca il linguaggio di P_8 , se esiste.

Si costruisca un automa a stati infiniti che riconosca il linguaggio di P_8 , se esiste.

Si commentino i risultati in relazione all'espressivita' delle reti di Petri rispetto agli automi finiti.

Traccia della soluzione.

Il linguaggio accettato da P_8 e' l'insieme di tutte le stringhe in $\{a, d\}^*$ dove ogni prefisso di ogni stringa contiene un numero di eventi d minore o uguale al numero di eventi a .

Non e' un linguaggio regolare, quindi non esiste un automa finito che lo accetta.

L'automata a stati infiniti che lo accetta e':

- $E = \{a, d\}$
- $X = \{0, 1, 2, 3, \dots\}$
- $\Gamma(x) = \{a, d\}$ per $x > 0$, $\Gamma(0) = \{a\}$
- $f(x, a) = x + 1$ per $x \geq 0$
- $f(x, d) = x - 1$ per $x > 0$

La classe dei linguaggi accettati dalle reti di Petri include strettamente quella dei linguaggi accettati da automi a stati finiti (linguaggi regolari).

English version.

A marked Petri net is specified by a quintuple: $\{P, T, A, w, x\}$, where P are the places, T the transitions, A the arcs, w is weight function on the arcs, and x is the marking vector (number of tokens for each place). $I(t_i)$ denotes the set of places entering transition t_i , $O(t_j)$ denotes the set of places exiting transition t_j .

To associate a language to a Petri net, one introduces a set of events E , a function that labels transitions by means of events $l : T \rightarrow E$, and a set of accepting (or marked or final) states $X_m \subseteq N^n$ (n is the number of places).

Consider the Petri net P_8 defined by:

- $P = \{p_1, p_2\}$
- $T = \{t_1, t_2\}$
- $A = \{(p_1, t_2), (p_2, t_2), (t_1, p_1), (t_2, p_2)\}$
- $w(p_1, t_2) = 1, w(p_2, t_2) = 1, w(t_1, p_1) = 1, w(t_2, p_2) = 1$

- (a) Draw the graph of the Petri net P_8 with the initial marking $x_1 = [0, 1]$.
- (b) Associate to P_8 a language over the following event alphabet $\{a, d\}$, with $l(t_1) = a$ (i.e., event a is associated to transition t_1) and $l(t_2) = d$.

Describe the language accepted by P_8 .

Build a finite state automaton which recognizes the language of P_8 , if it exists.

Build an infinite state automaton which recognizes the language of P_8 , if it exists.

Comment the previous answers with respect to the expressivity of Petri nets vs. the expressivity of finite automata.

Hints for the solution.

The language accepted by P_8 is the collection of all strings in $\{a, d\}^*$ where every prefix of every string contains a number of events d less or equal to the number of events a .

It is not a regular language, therefore there is no finite state automaton that accepts it.

The following is an infinite state automaton that accepts it:

- $E = \{a, d\}$
- $X = \{0, 1, 2, 3, \dots\}$

- $\Gamma(x) = \{a, d\}$ per $x > 0$, $\Gamma(0) = \{a\}$
- $f(x, a) = x + 1$ per $x \geq 0$
- $f(x, d) = x - 1$ per $x > 0$

The class of languages accepted by Petri nets is strictly larger than the class of languages accepted by finite state automata (regular languages).