

# 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

14 Settembre 2023

Nome e Cognome:

Matricola:

Posta elettronica:

problema	punti massimi	i tuoi punti
problema 1	6	
problema 2	8	
problema 3	4	
problema 4	12	
totale	30	

1. Si consideri un processo o impianto  $G$  sull'alfabeto  $E = \{a, b, c\}$  definito come segue:

- stati:  $x_0, x_1, x_2, x_3, x_4$  con  $x_0$  stato iniziale e unico stato accettante;
- transizione da  $x_0$  a  $x_1$ : evento  $a$ ,  
transizione da  $x_1$  a  $x_2$ : evento  $a$ ,  
transizione da  $x_1$  a  $x_3$ : evento  $b$ ,  
transizione da  $x_2$  a  $x_3$ : evento  $a$ ,  
transizione da  $x_2$  a  $x_4$ : evento  $c$ ,  
transizione da  $x_3$  a  $x_0$ : evento  $c$ ,  
transizione da  $x_3$  a  $x_1$ : evento  $a$ ,  
transizione da  $x_4$  a  $x_3$ : evento  $a$ .

- (a) Si disegni il grafo dell'automa  $G$ .
- (b) Si consideri la specifica  $K$  sull'alfabeto  $\{b, c\}$  descritta dall'automa  $H$ .  
La specifica stabilisce che non possono verificarsi due occorrenze dell'evento  $b$  senza che tra esse si verifichi almeno una volta l'evento  $c$ .  
Si disegni l'automa  $H$ .
- (c) Si definisca l'operatore  $\parallel$  di composizione in parallelo di due automi.
- (d) Si costruisca la composizione in parallelo  $G \parallel H$ .

Traccia di soluzione.

Si veda l'allegato.

English version

Consider a manufacturing plant  $G$  on the alphabet  $E = \{a, b, c\}$  defined as follows:

- states:  $x_0, x_1, x_2, x_3, x_4$  with  $x_0$  initial state and unique accepting state;
- transition from  $x_0$  to  $x_1$ : event  $a$ ,  
transition from  $x_1$  to  $x_2$ : event  $a$ ,  
transition from  $x_1$  to  $x_3$ : event  $b$ ,  
transition from  $x_2$  to  $x_3$ : event  $a$ ,  
transition from  $x_2$  to  $x_4$ : event  $c$ ,  
transition from  $x_3$  to  $x_0$ : event  $c$ ,  
transition from  $x_3$  to  $x_1$ : event  $a$ ,  
transition from  $x_4$  to  $x_3$ : event  $a$ .

- (a) Draw the graph of the automaton  $G$ .
- (b) Consider the following specification  $K$  on the alphabet  $\{b, c\}$  described by an automaton  $H$ . The specification requires that two occurrences of the event  $b$  cannot happen if the event  $c$  does not happen at least once between them.  
Draw the automaton  $H$ .
- (c) Define the operator  $||$  of parallel composition of two automata.
- (d) Build the parallel composition  $G||H$ .

Hints for the solution.

See the attachment.

2. Si considerino l'impianto  $G$  e la specifica  $H$  costruiti nel punto precedente. Si assuma che gli eventi incontrollabili siano  $E_{uc} = \{c\} \subseteq E$ .
- (a) Dati i linguaggi  $K$  della specifica e  $M = \overline{M}$  dell'impianto, si scriva la definizione di controllabilit  di  $K$  rispetto a  $M$  e  $E_{uc}$ .
  - (b) Dato che si dispone degli automi a stati finiti  $G$  e  $H$ , si puo determinare se  $K$  e' controllabile applicando l'algoritmo cosiddetto standard, che prevede la costruzione dell'automa  $H||G$  e la sua analisi rispetto alla controllabilit .
    - i. Si descriva con precisione l'algoritmo per determinare la controllabilit  mediante il prodotto  $H||G$ .
    - ii. Si applichi l'algoritmo al nostro caso di studio.  
 $K$  e' controllabile rispetto a  $M$  e  $E_{uc}$  ?
    - iii. Esiste un supervisore  $S$  che realizza il linguaggio  $K$  della specifica ?  
 Se si, si proponga una strategia di controllo.
  - (c) Si definisca il sottolinguaggio controllabile massimo  $K^{\uparrow C}$ .
  - (d) Si calcoli  $K^{\uparrow C}$ , usando ancora l'algoritmo precedente che analizza l'automa prodotto  $H||G$  iterando passi di rimozione e potatura dell'automa prodotto. Si mostrino i passi dell'algoritmo e si disegni l'automa prodotto ad ogni passo.

Traccia di soluzione.

Si veda l'allegato.

$K$  e' controllabile (si puo' vedere applicando l'algoritmo di rimozione o la definizione).

La strategia di controllo e' di disabilitare  $b$  dopo  $ba(aaa)^*$ . L'algoritmo di rimozione si arresta all'inizio senza rimuovere stati dato che si devono disabilitare solo eventi controllabili, per cui  $G||H$  e' il supervisore cercato che realizza il linguaggio della specifica  $K$ .

English version

Consider the plant  $G$  and the specification  $H$  built in the previous question. Suppose that the uncontrollable events are  $E_{uc} = \{c\} \subseteq E$ .

- (a) Given the languages  $K$  of the specification and  $M = \overline{M}$  of the plant, write down the definition of controllability of  $K$  with respect to  $M$  and  $E_{uc}$ .
- (b) Since we have the finite state automata  $G$  and  $H$ , one can determine if  $K$  is controllable by applying the so-called standard algorithm, which requires the construction of the automaton  $H||G$  and its analysis with respect to controllability.
  - i. Describe precisely the algorithm to determine the controllability by analyzing the product  $H||G$ .
  - ii. Apply such algorithm to our example of manufacturing plant.  
Is  $K$  controllable with respect to  $M$  and  $E_{uc}$  ?
  - iii. Does there exist a supervisor  $S$  that realizes the language  $K$  of the specification ?  
If so, describe a control strategy.
- (c) Define the supremal controllable sublanguage  $K^{\uparrow C}$ .
- (d) Compute  $K^{\uparrow C}$ , using again the previous standard algorithm, which builds the product automaton  $H||G$  and iterates steps to remove uncontrollable states and to trim the product automaton.  
Show the steps of the algorithm and draw the product automaton at every step.

Hints for the solution.

See the attachment.

$K$  is controllable (it can be seen by applying the standard removal algorithm or the definition).

The control strategy is to disable  $b$  after  $ba(aaa)^*$ . The standard removal algorithm finishes at the beginning without removing any state, given that one is required to disable only controllable events, so that  $G||H$  is the requested supervisor that realizes the language of the specification  $K$ .

3. Si aggiunga la seguente restrizione al precedente problema di controllo supervisore: il sistema sotto supervisione non deve bloccarsi, cioè deve sempre essere possibile ritornare allo stato  $x_0$  che è il solo stato finale.

Si costruisca il supervisore non-bloccante  $K_{nb}^{\uparrow C}$  (dove  $nb$  indica non bloccante), eliminando gli stati non raggiungibili e non co-raggiungibili.

Traccia di soluzione.

Il supervisore ottenuto precedentemente è già non bloccante (cioè da ogni stato si può raggiungere lo stato iniziale).

English version

Add the following restriction to the previous supervisory control problem: the system under supervision should be non-blocking, i.e., it must always be possible to come back to the state  $x_0$ , which is the only final state.

Build the non-blocking supervisor  $K_{nb}^{\uparrow C}$  (where  $_{nb}$  means non-blocking), removing the states which are not reachable or co-reachable.

Hints for the solution.

The supervisor obtained previously is already non-blocking (i.e. from every state one can reach the initial state).

4. Si aggiungano le seguenti restrizioni al precedente problema di controllo supervisore:

- (a) Il sistema non deve raggiungere lo stato  $x_4$ ;
- (b) Il sistema sotto supervisione non deve bloccarsi, cioè deve sempre essere possibile ritornare allo stato  $x_0$  che è il solo stato finale.

Si costruisca il supervisore richiesto in due fasi:

- (a) A partire da  $K^{\uparrow C}$ , si costruisca prima il supervisore  $K_{!x_4}^{\uparrow C}$  (dove  $!x_4$  indica che non si può raggiungere  $x_4$ ), eliminando  $x_4$  e gli stati divenuti incontrollabili.
- (b) Poi si costruisca il supervisore non-bloccante  $K_{!x_4, nb}^{\uparrow C}$  (dove  $!x_4, nb$  indica che non si può raggiungere  $x_4$  e che il supervisore non è bloccante), eliminando gli stati non raggiungibili e non co-raggiungibili.

Traccia di soluzione.

Nel primo passo si elimina lo stato  $(x_4, y_0)$  e tutti i lati entranti ed uscenti ottenendosi il grafo  $F'$ .

Nel secondo passo si eliminano gli stati  $(x_2, y_0)$  e  $(x_2, y_1)$  in quanto da ognuno di essi esce un lato con l'evento  $c$  incontrollabile, ottenendosi il grafo  $F''$ .

Nel terzo passo si eliminano lo stato irraggiungibile  $(x_3, y_0)$  (non lo si può raggiungere dallo stato iniziale) e lo stato co-irraggiungibile  $(x_1, y_1)$  (da esso non si raggiunge lo stato iniziale), ottenendosi il grafo  $F'''$  che è anche quello finale.

Si vedano i passi grafici nell'allegato.

English version

Add the following restrictions to the previous control supervisor problem:

- (a) The system should not reach the state  $x_4$ ;
- (b) the system under supervision should be non-blocking, i.e., it must always be possible to come back to the state  $x_0$ , which is the only final state.

Build the required supervisor in two phases:

- (a) Starting from  $K^{\uparrow C}$ , build first the supervisor  $K_{!x_4}^{\uparrow C}$  (where  $!x_4$  means that it should not reach  $x_4$ ), removing  $x_4$  and the uncontrollable states.
- (b) Then build the non-blocking supervisor  $K_{!x_4, nb}^{\uparrow C}$  (where  $!x_4, nb$  means that it should not reach  $x_4$  and that the supervisor is non-blocking), removing the states which are not reachable or co-reachable.

Hints for the solution.

In the first step we remove the state  $(x_4, y_0)$  and all entering and exiting transitions, obtaining the graph  $F'$ .

In the second step we remove the states  $(x_2, y_0)$  and  $(x_2, y_1)$  since from each of them there is an exiting transition with the uncontrollable event  $c$ , obtaining the graph  $F''$ .

In the third step we remove the unreachable state  $(x_3, y_0)$  (it cannot be reached from the initial state) and the co-unreachable state  $(x_1, y_1)$  (from which the initial state is unreachable), obtaining the graph  $F'''$ , which is the final one.

See the graphical steps in the attachment.