ECE 641 Advanced Topics in Supervisory Control for Discrete Event Systems

Lecture 7

Associate Prof. Dr. Klaus Schmidt

Department of Mechatronics Engineering – Çankaya University

PhD Course in Electronic and Communication Engineering
Credits (3/0/3)
Course webpage: http://ece641.cankaya.edu.tr/

Reminder: Observability

Definition (Observability)

K is denoted as *observable* for G, Σ_c and Σ_o if for all $s, s' \in \overline{K}$ with $p(s) = p(s')$ and for all $\sigma \in \Sigma_\mathrm{c}$

$$s\sigma \in L(G) \text{ and } s'\sigma \in \overline{K} \Rightarrow s\sigma \in \overline{K}.$$

Theorem (Observability Theorem)

There exists a nonblocking supervisor under partial observation S such that $L_\mathrm{m}(G||S) = K$ and $L(G||S) = \overline{K}$ if and only if

- *K is controllable for G and Σ_u*
- *K is observable for G, Σ_c and Σ_o*
- *K is $L_\mathrm{m}(G)$-closed*

Theorem (Observability Verification)

K is observable for G, Σ_c and Σ_o if and only if $L_\mathrm{m}(V) = \emptyset$.

Properties: Union of Observable Languges

Example

Gap 1

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Properties: Results

Lemma

Let G be a plant, Σ_{c} a set of controllable events, Σ_{o} a set of observable events. Assume that $K_1, K_2 \subseteq L_{\mathrm{m}}(G)$ are observable for G, Σ_{c}, Σ_{o}. Then, $K_1 \cup K_2$ is not necessarily observable for G, Σ_{c}, Σ_{o}.

Remarks

- The lemma implies that there is no supremal observable sublanguage!

Lemma

Let G be a plant, Σ_{c} a set of controllable events, Σ_{o} a set of observable events. Assume that $K_1, K_2 \subseteq L_{\mathrm{m}}(G)$ are observable for G, Σ_{c}, Σ_{o}. Then, $K_1 \cap K_2$ is observable for G, Σ_{c}, Σ_{o}.

Remarks

- There exists an "infimal observable superlanguage"

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Properties: Results

Proof

Gap 2

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Supervisory Control: Basic Problem

Problem (Observability Theorem)

Let G be a plant automaton, $K \subseteq L_{\mathrm{m}}(G)$ a specification language, Σ_c a controllable event set, Σ_{o} an observable event set and $p : \Sigma^\star \to \Sigma_{\mathrm{o}}^\star$ the associated natural projection. Assume that K is $L_{\mathrm{m}}(G)$-closed. Find a nonblocking supervisor under partial observation S such that

- $L_{\mathrm{m}}(S||G) \subseteq K$
- $L_{\mathrm{m}}(S||G)$ *is as large as possible*

Remarks

- We already know that there might not be a supremal solution
  $\Rightarrow$ We can look for maximal solutions such that for all other supervisors S': $L_{\mathrm{m}}(S'||G) \not\supset L_{\mathrm{m}}(S||G)$
  $\Rightarrow$ We can look for conditions such that a supremal solution exists

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Supervisory Control: Example

Database Concurrency Control

Gap 3

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Supervisory Control: Example

Database Concurrency Control

Gap 4

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Supervisory Control: Example

Database Concurrency Control

Gap 5

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Normality: Definition

Definition

Let G be an automaton with the alphabet Σ, $K \subseteq L_{\mathrm{m}}(G)$ be a specification, Σ_{o} a set of observable events and $p : \Sigma^\star \to \Sigma_{\mathrm{o}}^\star$ be a natural projection. K is denoted as normal for G and p if

$$\overline{K} = p^{-1}p(\overline{K}) \cap L(G)$$

Remarks

- $p(\overline{K})$ is the "observation" of $\overline{K}$
- Normality means that $\overline{K}$ can be recovered from its observation and G
- Normality is a condition for closed languages

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Normality: Example

Database Concurrency Control

Gap 6

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Normality: Properties

Lemma

Assume that $K \subseteq L_{\mathrm{m}}(G)$ is normal for G and p. Then, K is observable for G, any $\Sigma_{\mathrm{c}} \subseteq \Sigma$ and Σ_{o}

Proof

Gap 7

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Normality: Properties

Lemma

Normality is stronger than observability.

Proof

Gap 8

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Normality: Properties

Lemma

Assume that $K_1, K_2 \subseteq L_{\mathrm{m}}(G)$ are normal for G and p. Then, $K_1 \cup K_2$ is normal for G and p.

Proof

Gap 9

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Normality: Properties

Theorem

Assume that $\Sigma_c \subseteq \Sigma_o$. Then, observability and normality are equivalent.

Proof

Gap 10

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Supremal Normal Sublanguage: Discussion

Observation

- There exists a supremal normal sublanguage $SupN(K, G, p)$
- There exists a supremal controllable sublanguage
- If $\Sigma_c \subseteq \Sigma_o$, there is a supremal observable sublanguage

$\Rightarrow$ Compute supremal controllable and normal sublanguage in order to solve the supervisory control problem under partial observation

Computation of $SupN(K, G, p)$

- Lin-Brandt formula for prefix-closed languages:

$$SupN(K, G, p) = K \setminus (p^{-1}p(L(G) \setminus K)\Sigma^\star)$$

- $p^{-1}p(L(G) \setminus K)$: strings with same observation as forbidden strings

$\Rightarrow$ Remove all potentially forbidden strings and their successors from K

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Supremal Normal Sublanguage: Solution

Computation of $SupN(K, G, p)$ (General Case)

- $K_1 := K$
- Iteration: $K_{i+1} = SupN(\overline{K}_i, G, p) \cap K$ until $K_{i+1} = K_i$
  $\Rightarrow$ Result is $SupN(K, G, p)$

Computation of $SupCN(K, G, p, \Sigma_u)$

- Iterate until $K_{i+1} = K_i$ starting from $K_1 = K$
  - Compute $K_{i+1} = SupCon(K_i, G, \Sigma_u)$, $i = i + 1$
  - Compute $K_{i+1} = SupN(K_i, G, p)$, $i = i + 1$
- Complexity is exponential in the worst-case because of projection!

Example

Gap 11

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Supremal Normal Sublanguage: Example

Database Concurrency Control

Gap 12

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University