Embedded and Cyber-Physical Systems

- Modeling discrete behavior -

Classic DES control example: Database concurrency

- Read/write events
- Transaction: a sequence of events from the same user
- Schedule: interleaving of events from different (concurrent) transactions
- Control problem:
  - Allow only admissible schedules
  - Allow as many admissible schedules as possible
Database concurrency example

- Uncontrolled behavior for two transactions $T_1 = a_1 b_1$ and $T_2 = a_2 b_2$:
Model of the admissible behavior

• Ordering constraint

\[ a_1 \text{ precedes } a_2 \text{ if and only if } b_1 \text{ precedes } b_2 \]
All events are controllable and observable

- Supervisor $S_1$ uses $H_a$
- Every event executed by $G$ causes the same event in $H_a$
- When $H_a$ enters state 4, $S_1$ disables event $b_1$
- When $H_a$ enters state 9, $S_1$ disables event $b_2$
The effect of uncontrollable events

• Let’s assume $a_2$ and $b_2$ are uncontrollable
The effect of unobservable events

• Let’s assume $a_2$ is unobservable
Unobservable and uncontrollable events

- Event $a_2$ is unobservable
The Controllability Theorem

- \( G = (X, E, f, \Gamma, x_0) \) with \( E_{uc} \subseteq E \)
- \( K \subseteq \mathcal{L}(G) \), where \( K \neq \emptyset \).

- There exists supervisor \( S \) such that \( \mathcal{L}(S/G) = \overline{K} \)
  if and only if

\[
\overline{K}E_{uc} \cap \mathcal{L}(G) \subseteq \overline{K}
\]
Checking Controllability

• If $H$ is an automaton that generates $\overline{K}$
• $K$ is controllable with respect to $\mathcal{L}(G)$ and $E_{uc}$ if and only if
  \[ \forall x \in X_{H \times G}, x = (x_H, x_G), \forall e \in E_{uc} \cap \Gamma(x_G) \implies e \in \Gamma(x) \]
• What is the worst-case computational complexity of this test?
Example of controllability check

- Is $K = \mathcal{L}_m(H_a)$ controllable?
Example of controllability check

- Yes, if $b_1$ and $b_2$ are controllable
Realization of supervisors

- Let $R := (Y, E, g, \Gamma_R, y_0, Y)$ such that
  \[ \mathcal{L}_m(R) = \mathcal{L}(R) = \overline{K} \]
- Then $R||G$ represents the behavior of the closed-loop system $S/G$

\[
S(s) = [E_{uc} \cap \Gamma(f(x_0, s))] \cup \{\sigma \in E_c : s\sigma \in \overline{K}\} \\
= \Gamma_R(g(y_0, s)) \\
= \Gamma_R||G(g||f((y_0, x_0), s))
\]
What if $K$ is not controllable?

- Supremal controllable sublanguage
- Example: $K = \mathcal{L}_m(H_a)$ and $E_{uc} = \{a_2, b_2\}$
DES model for the tank example

• Variable outflow
• A 2-speed pump
• Level between l1 and l2
• Minimize load on external supply
• Maximal input flow greater than maximal output flow
• Safety specification: tank should not be empty
• Design a supervisor for this system!
Timed automata with guards

• An automaton with an associated set of clocks
• A clock is a continuous variable \( c_i(t) \) with
  \[
  \frac{d}{dt} c_i(t) = 1
  \]
• A transition has the form \(( \text{guard} ; \text{event} ; \text{reset} )\)
  – \text{guard} specifies a timing precondition for the transition to occur
  – \text{event} is generated when the transition occurs
  – \text{reset} indicates which clock variables are reset to zero upon executing the transition
Example of timed automaton

- ; \textit{msg} ; c_1

0 < c_1 < 1 ; \textit{msg} ; c_1

c_1 < 1 ; \textit{alarm} ; -

c_1 \geq 1 ; \textit{msg} ; c_1

• As long as a \textit{msg} event is generated more than one time unit after the previous \textit{msg} event, then another \textit{msg} event can occur.

• When a \textit{msg} event occurs less than one time unit after the previous \textit{msg} event, then the \textit{alarm} event is generated within the next time unit and the system stops.

• What if \textit{alarm} is not generated „in time“ at state 2?
Timed automata with guards and invariants

- The system must leave state 2 before the value of $c_1(t)$ reaches 1
- Each state must have at least one outgoing transition whose guard has a non-empty intersection with the state invariant

\[ \begin{align*}
0 & \rightarrow 1 & 0 < c_1 < 1 \; ; \; \text{msg} \; ; \; c_1 \\
1 & \rightarrow 2 & \; ; \; \text{msg} \; ; \; c_1 \; \Rightarrow \; c_1 < 1 \\
2 & \rightarrow 3 & - \; ; \; \text{alarm} \\
1 & \rightarrow 0 & c_1 \geq 1 \; ; \; \text{msg} \; ; \; c_1
\end{align*} \]
Timed automata model for the tank example

- Variable outflow
- A 2-speed pump
- Level between $l_1$ and $l_2$
- Minimize load on external supply
- Maximal input flow greater than maximal output flow
- Safety specification: tank should not be empty
Parallel composition of timed automata

\[ 0 < c_1 < 1; \text{msg} ; c_1 \quad \text{||} \quad 0 < c_1 < 1 \land c_2 > 5; \text{msg} ; c_1, c_2 \]

\[ c_1 < 1 \quad \text{||} \quad c_1 < 1 \land c_2 < 1 \]

\[ c_1 \geq 1; \text{msg} ; c_1 \quad \text{||} \quad c_1 \geq 1 \land c_2 \leq 5; \text{msg} ; c_1, c_2 \]

\[ c_1 < 1 \quad \text{||} \quad c_1 < 1 \land c_2 \leq 5; \text{msg} ; c_1, c_2 \]

\[ c_2 < 1 \quad \text{||} \quad c_2 < 1 \]

\[ c_2 \geq 1 \land c_2 > 5; \text{msg} ; c_1, c_2 \]

\[ c_2 \leq 5; \text{msg} ; c_2 \]

\[ c_2 > 5; \text{msg} ; c_2 \]

\[ -; \text{alarm} ; - \]

\[ -; \text{alarm5} ; - \]
Hybrid automata

• Extensions of timed automata where the simple clock dynamics are replaced by arbitrary time-driven dynamics

• Example: the two-tank system from the introduction lecture

\[
\begin{align*}
\dot{x}_1 &= k_{11} \\
\dot{x}_2 &= k_{12}
\end{align*}
\]

\[
\begin{align*}
\dot{x}_1 &= k_{21} \\
\dot{x}_2 &= k_{22}
\end{align*}
\]