Embedded and Cyber-Physical Systems

- Modeling discrete behavior -

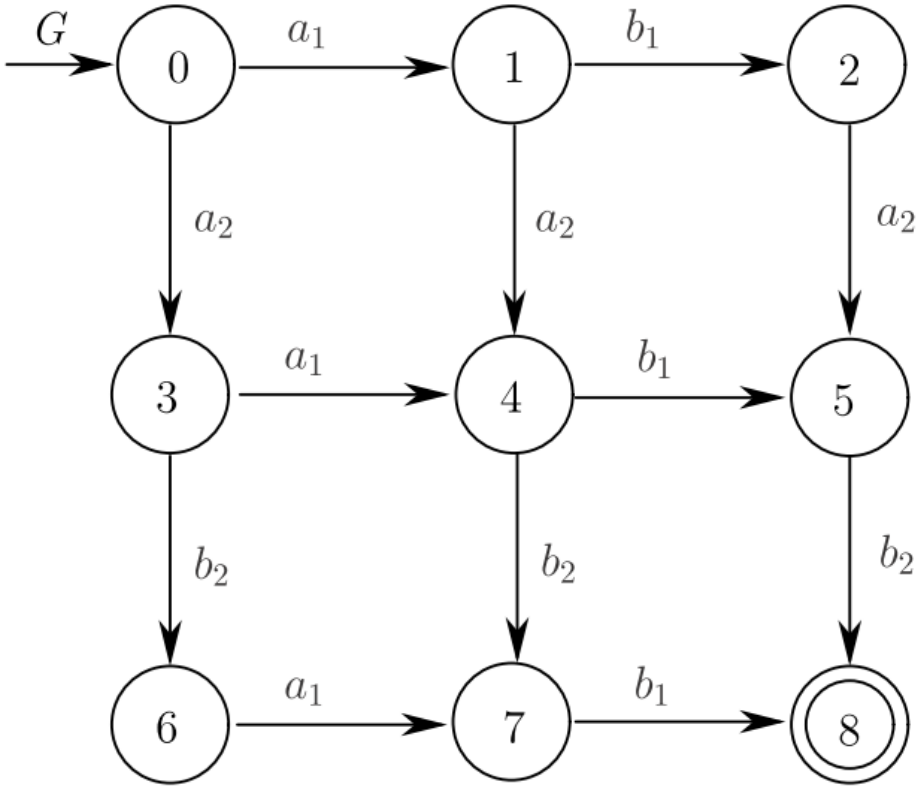
Reference book: Christos G. Cassandras and Stéphane Lafortune, *Introduction to Discrete Event Systems*, Second Edition, Springer, ISBN: 978-0-387-33332-8, 2008.

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

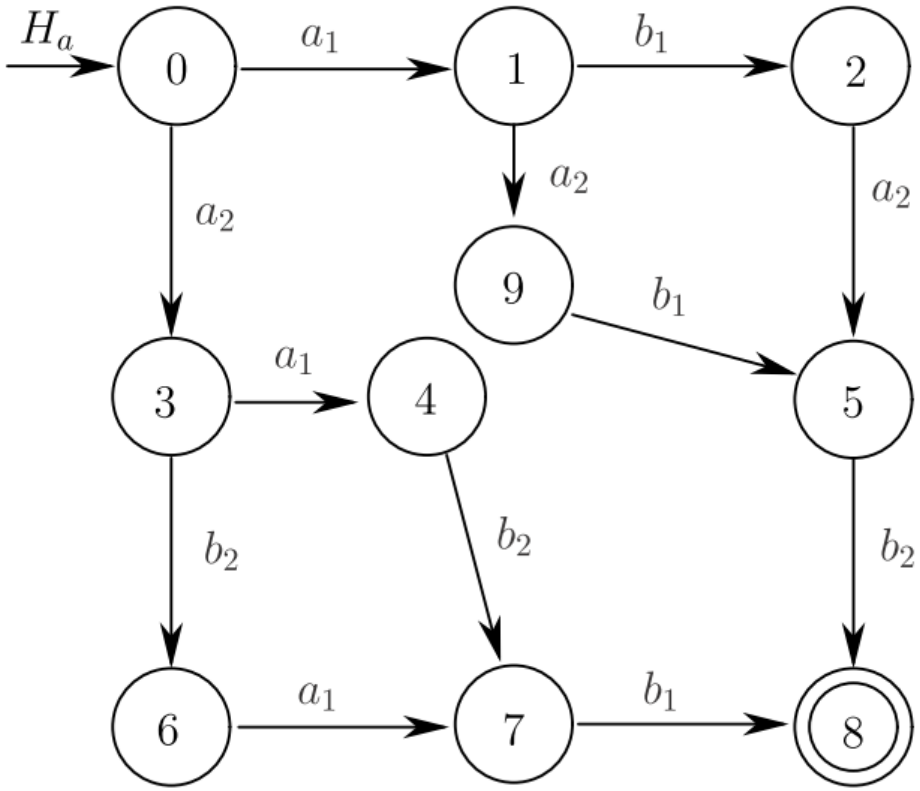
- Uncontrolled behavior for two transactions $T_1 = a_1 b_1$ and $T_2 = a_2 b_2$:



Model of the admissible behavior

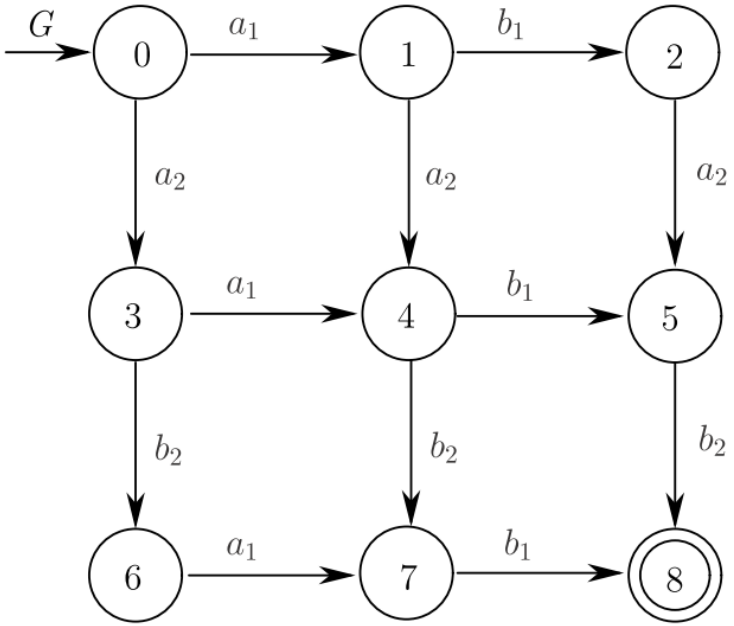
- Ordering constraint

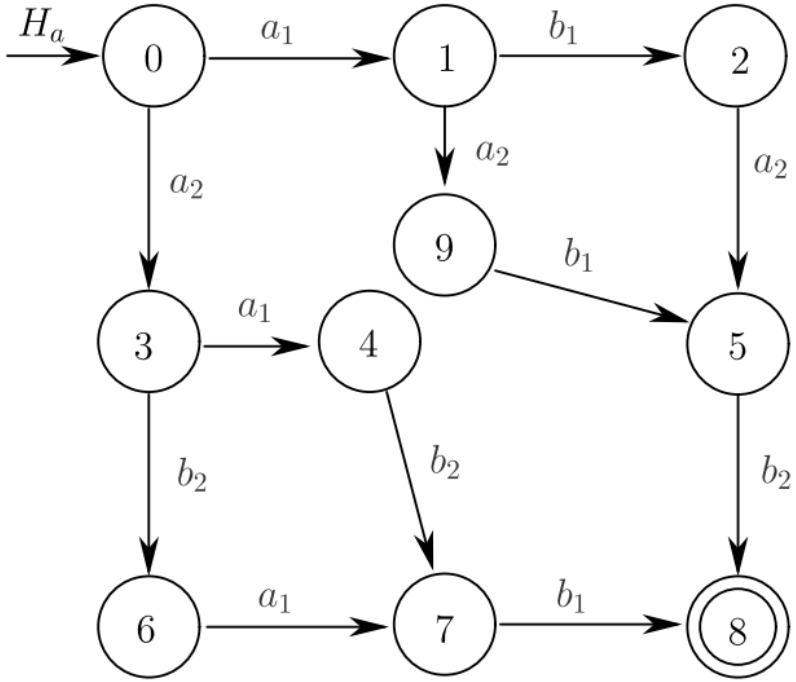
a_1 precedes a_2 if and only if b_1 precedes b_2



All events are controllable and observable

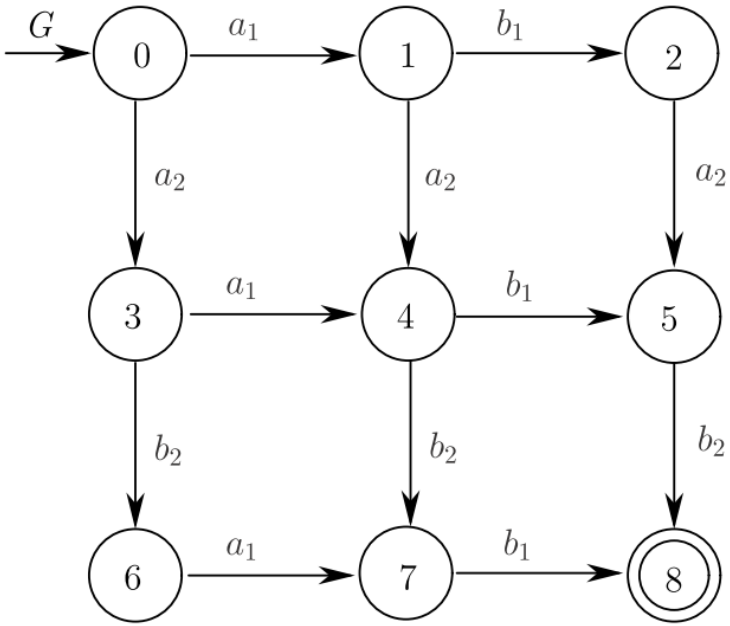
- Supervisor S_1 uses H_a
- Every event executed by G causes the same event in H_a
- When H_a enters state 4, S_1 disables event b_1
- When H_a enters state 9, S_1 disables event b_2

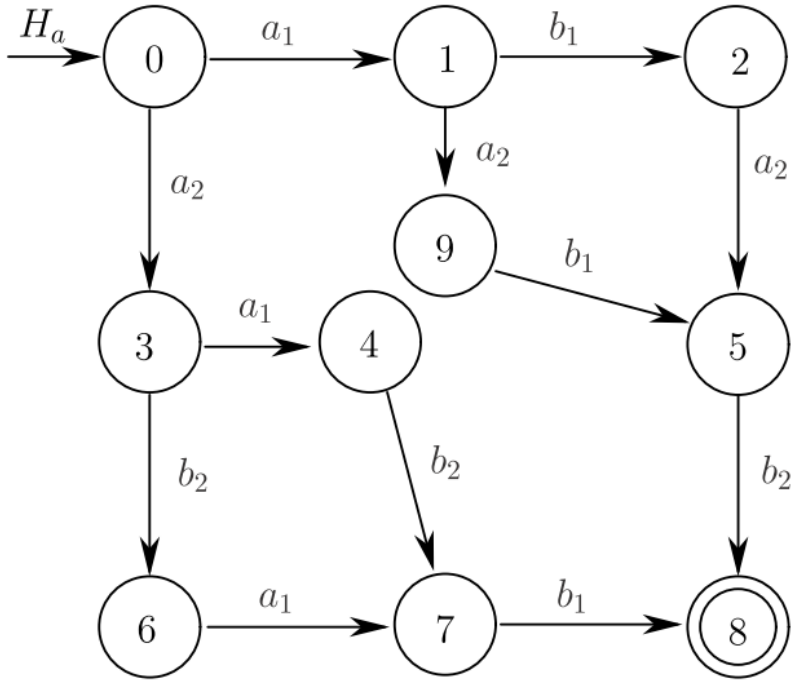




The effect of uncontrollable events

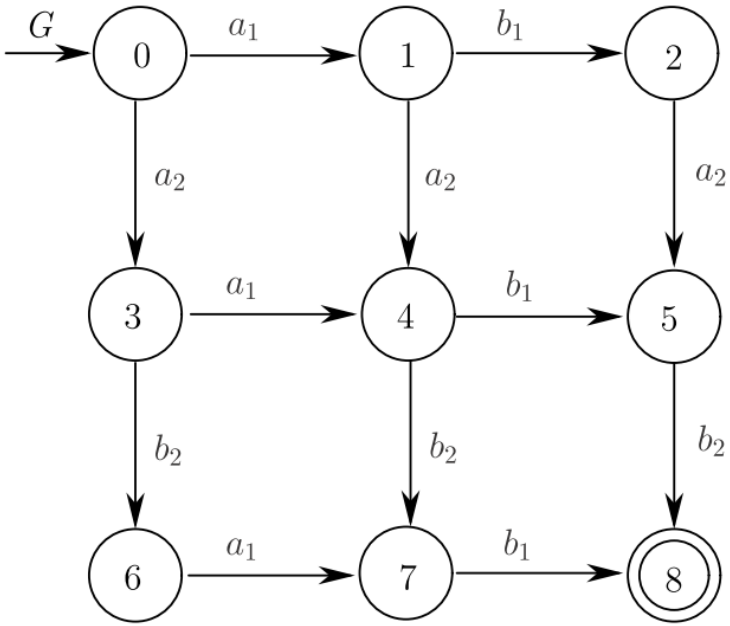
- Let‘s assume a_2 and b_2 are uncontrollable

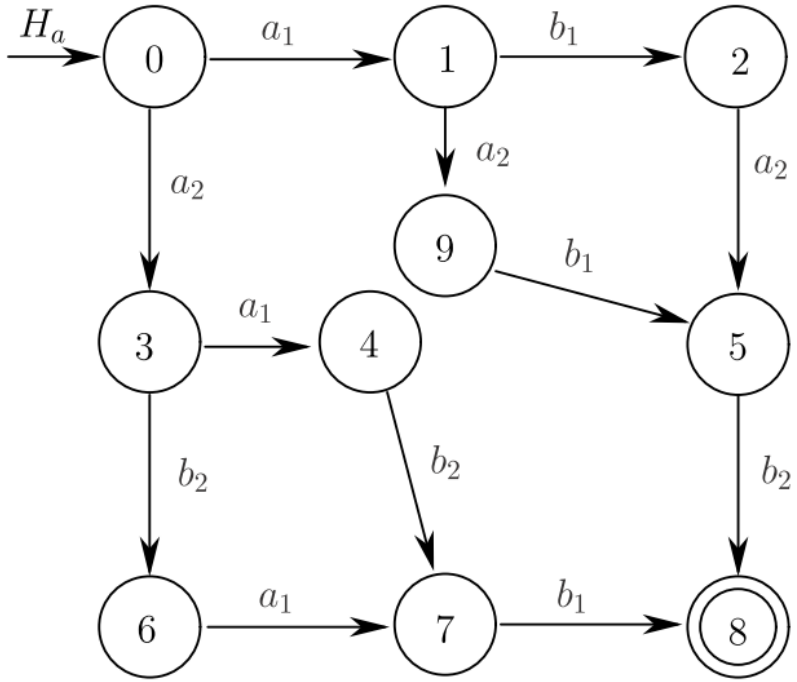




The effect of unobservable events

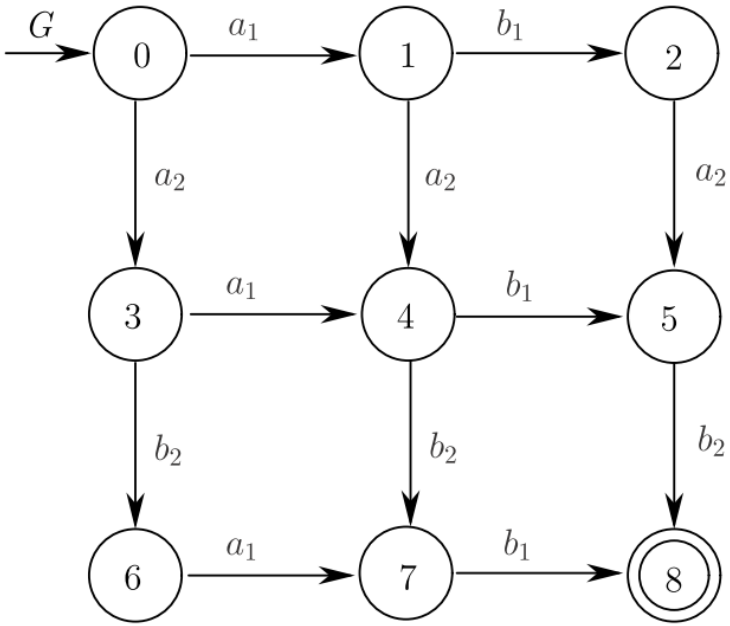
- Let‘s assume a_2 is unobservable

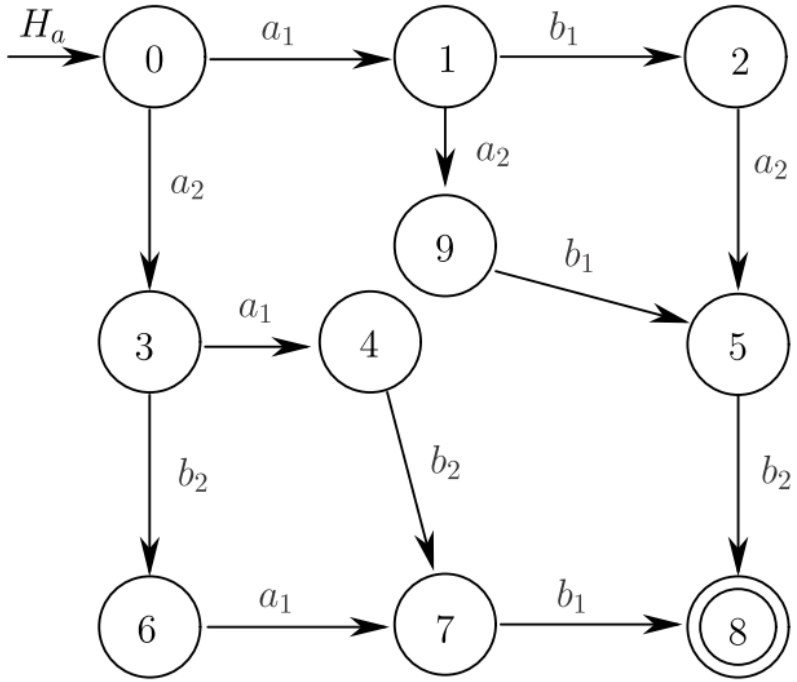




Unobservable and uncontrollable events

- 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

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

Checking Controllability

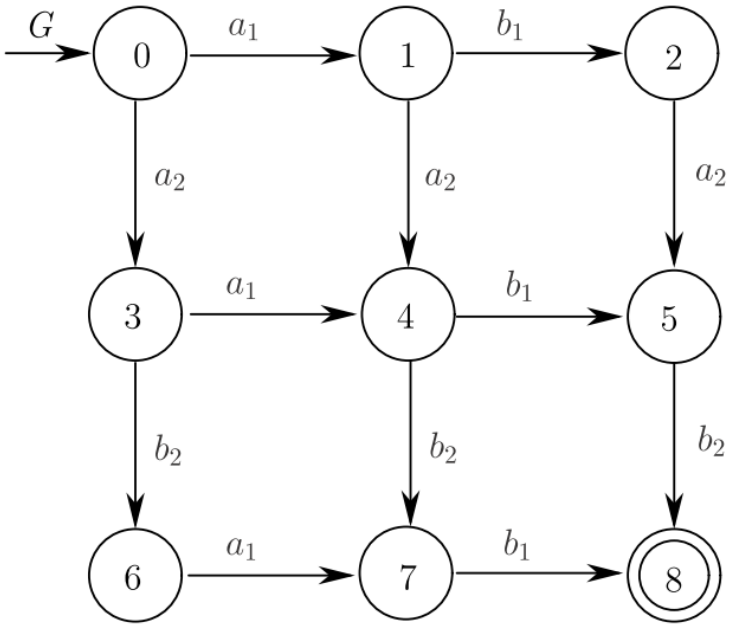
- If H is an automaton that generates $\overline{K}$
- K is controllable with respect to $\mathcal{L}(G)$ and E_{uc} if and only if

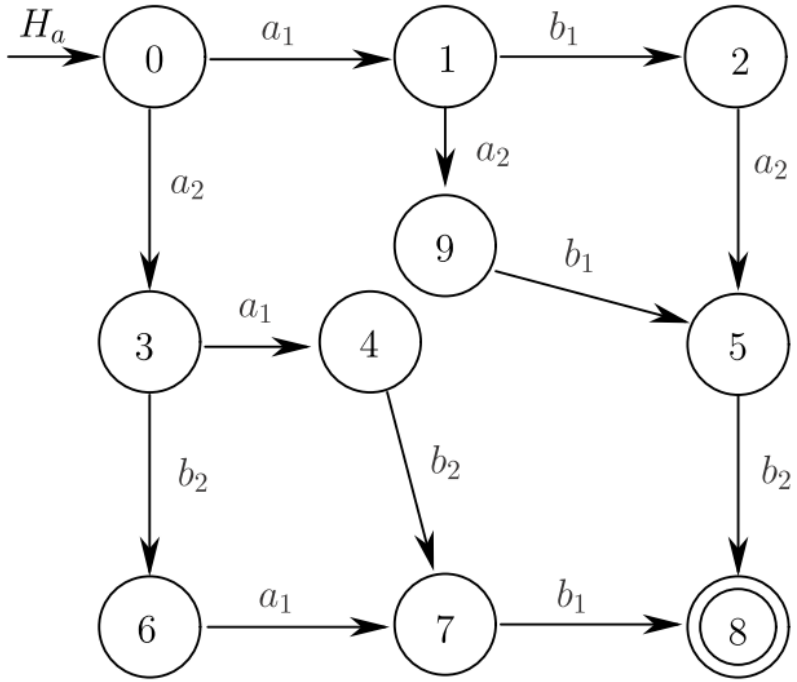
$$\forall x \in X_{H\times G}, x = (x_H, x_G), \forall e \in E_{uc} \bigcap \Gamma(x_G) \Rightarrow e \in \Gamma(x)$$

- What is the worst-case computational complexity of this test?

Example of controllability check

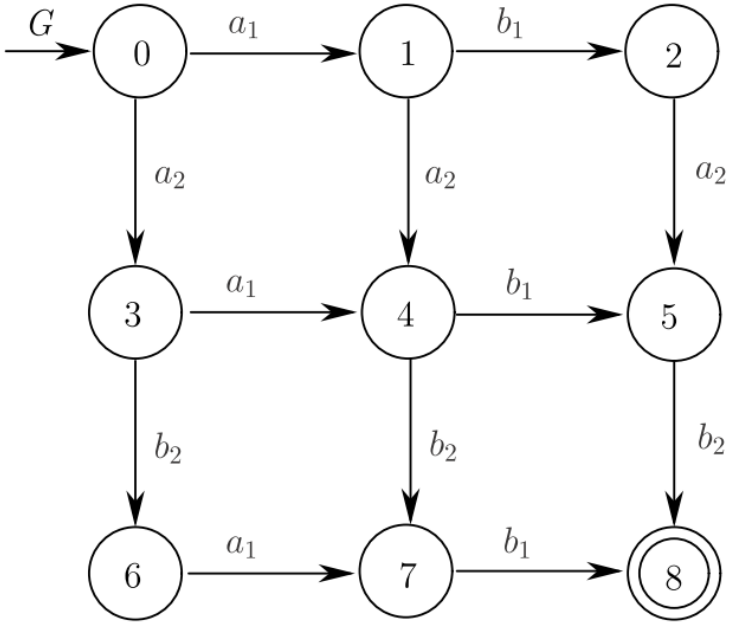
- Is $K = \mathcal{L}_m(H_a)$ controllable?

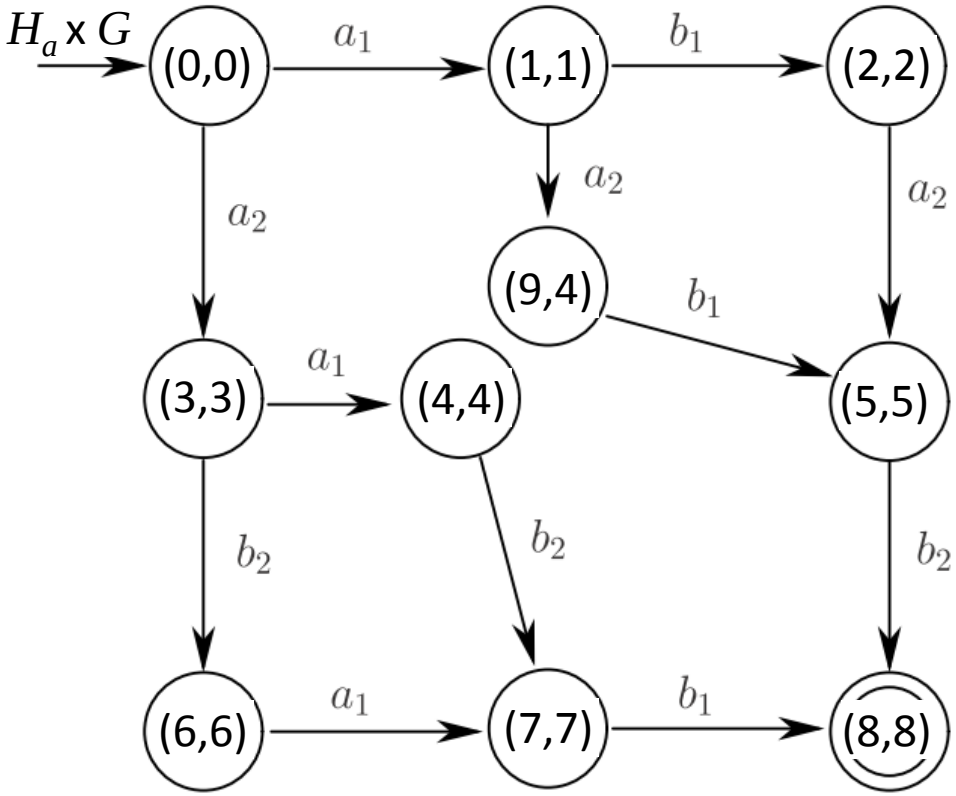




Example of controllability check

- Yes, if b_1 and b_2 are controllable



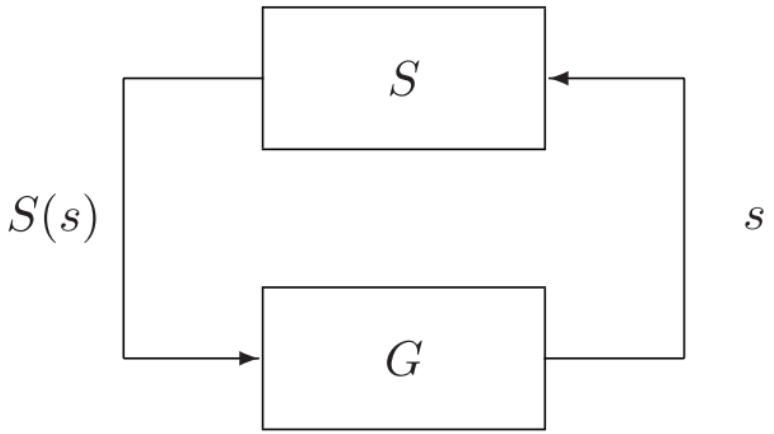


Realization of supervisors

- Let $R := (Y, E, g, \Gamma_R, y_0, Y)$ such that

  $\mathcal{L}_m(R) = \mathcal{L}(R) = \overline{K}$

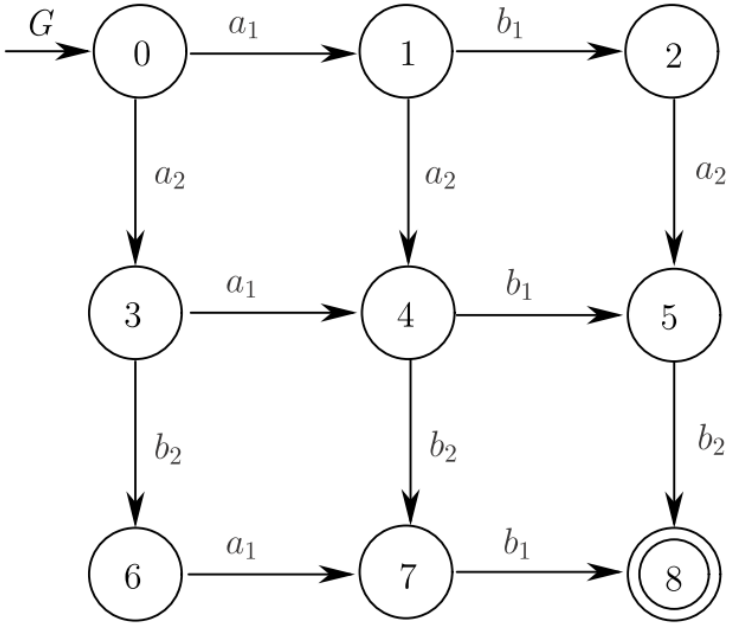
- Then $R||G$ represents the behavior of the closed-loop system *S/G*

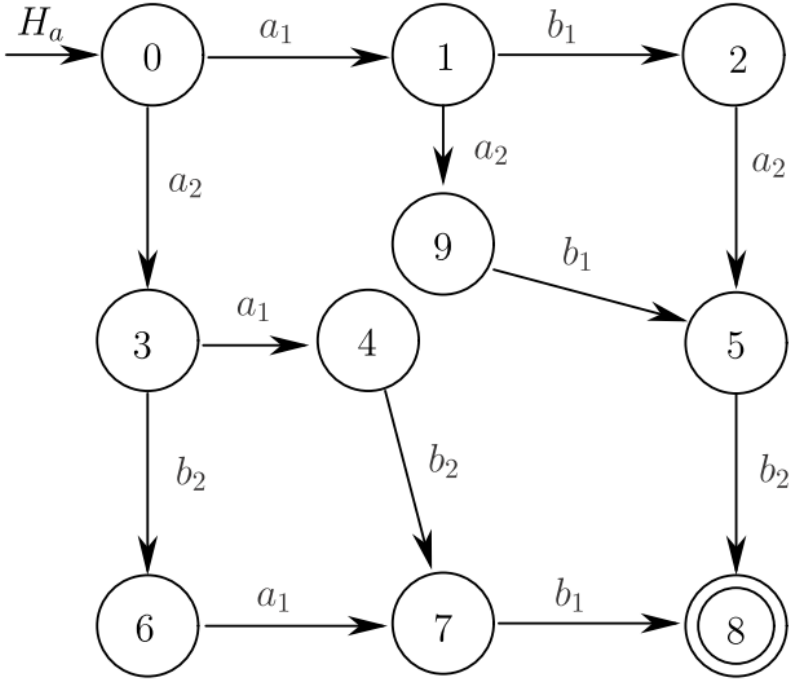


$$
\begin{aligned}
S(s) &= [E_{uc} \cap \Gamma(f(x_0, s))] \cup \{\sigma \in E_c : s\sigma \in \overline{K}\} \\
&= \Gamma_R(g(y_0, s)) \\
&= \Gamma_{R||G}(g||f((y_0, x_0), s))
\end{aligned}
$$

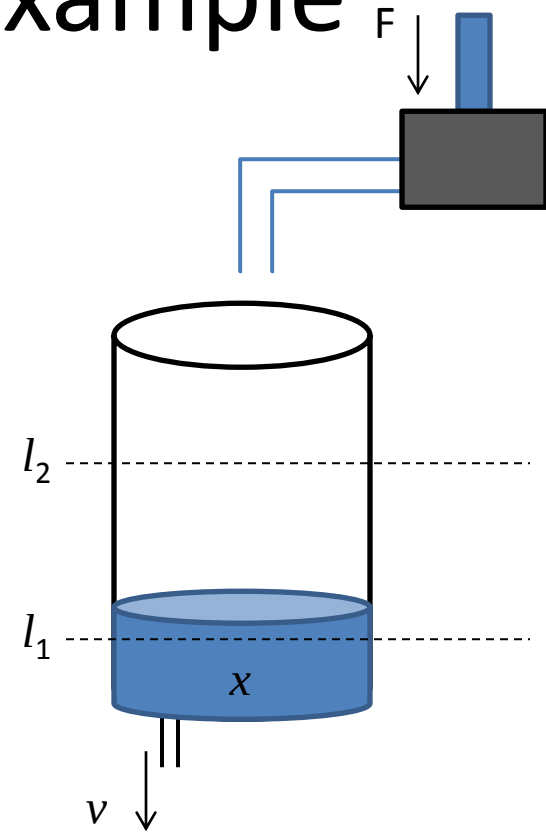
What if K is not controllable?

- 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!

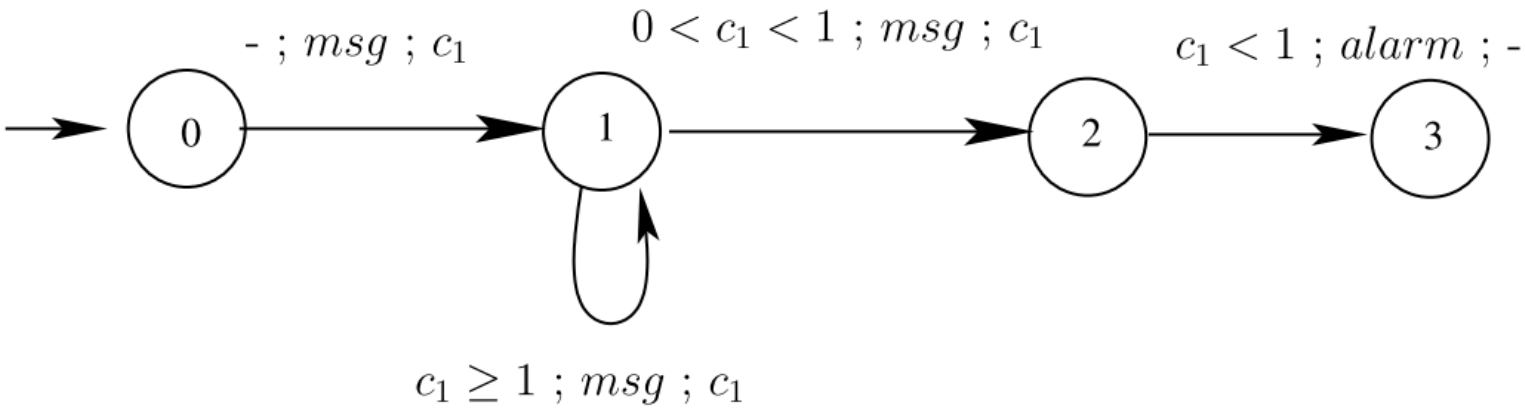
Timed automata with guards

- An automaton with an associated set of *clocks*
- A clock is a continuous variable $c_i(t)$ with

$$\frac{d}{dt}c_i(t) = 1$$

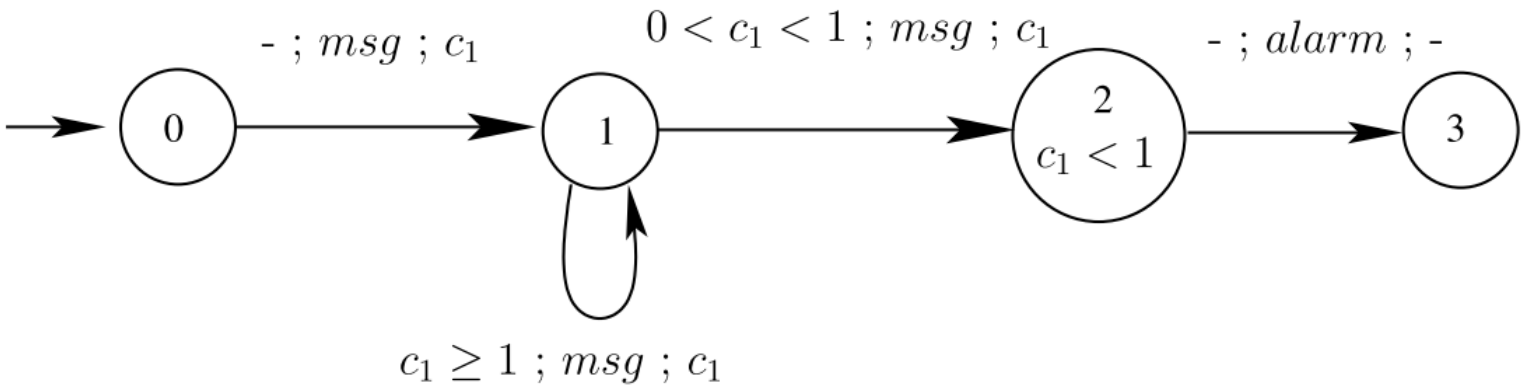
- A transition has the form $(\ guard \ ; \ \ event \ \ ; \ \ reset \ )$
  - *guard* specifies a timing precondition for the transition to occur
  - *event* is generated when the transition occurs
  - *reset* indicates which clock variables are reset to zero upon executing the transition

Example of timed automaton



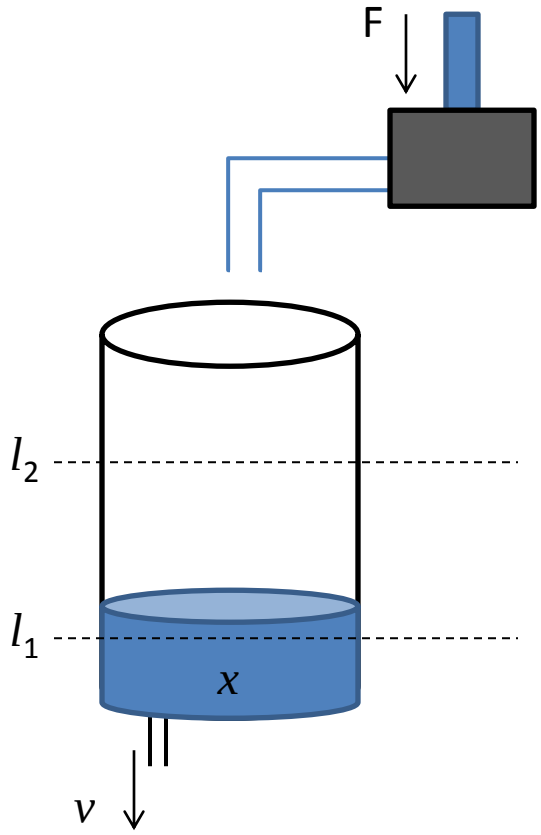
- As long as a *msg* event is generated more than one time unit after the previous *msg* event, then another *msg* event can occur.
- When a *msg* event occurs less than one time unit after the previous *msg* event, then the *alarm* event is generated within the next time unit and the system stops.
- What if *alarm* is not generated „in time“ at state 2?

Timed automata with guards and invariants



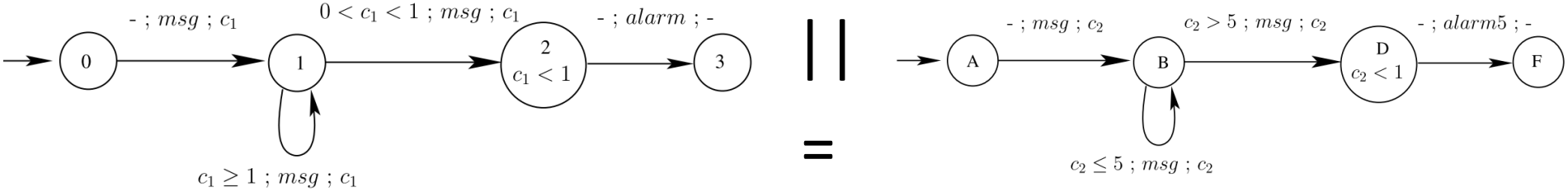
- The system must leave state 2 before the value of $c_1(t)$ reaches 1
- Each state must have at least one outgoing transition whose guard has a non-empty intersection with the state invariant

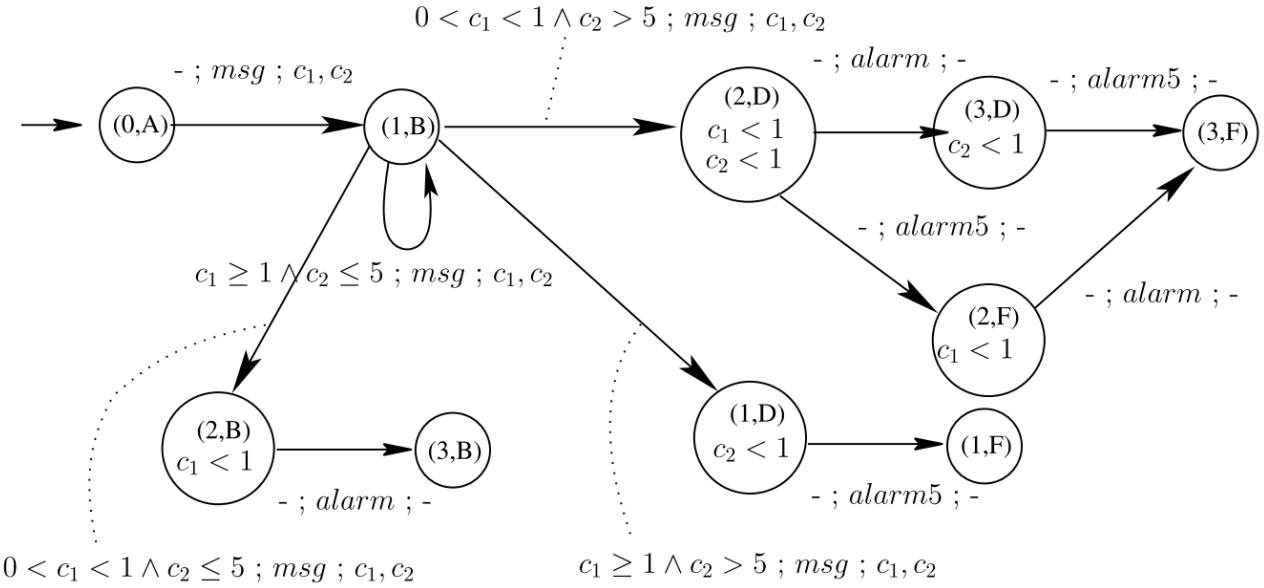
Timed automata model for the tank example



- 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

Parallel composition of timed automata





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