Discrete Event Systems

Exercise Session 1

Roland Schmid
nsg.ee.ethz.ch

ETH Zürich (D-ITET)
23 September 2021
Formal Definition of a Finite Automaton

A finite automaton (FA) is a 5-tuple \((Q, \Sigma, \delta, q_0, F)\), where

- \(Q\) is a finite set called the states
- \(\Sigma\) is a finite set called the alphabet
- \(\delta: Q \times \Sigma \rightarrow Q\) is the transformation function
- \(q_0 \in Q\) is the start state
- \(F \subseteq Q\) is the set of accept states (a.k.a. final states).
Vending Machine Java Code

Soda vend(){
    int total = 0, coin;
    while (total != 45){
        receive(coin);
        if ((coin==10 && total==40)
            ||(coin==25 && total>=25))
            reject(coin);
        else
            total += coin;
    }
    return new Soda();
}
Vending Machine “Logics”
Cartesian Product Construction

• We want to construct a finite automaton $M$ that recognizes any strings belonging to $L_1$ or $L_2$.

• Idea: Build $M$ such that it simulates both $M_1$ and $M_2$ simultaneously and accept if either of the automatons accepts.
Formal Definition

- Given two automata
  \( M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1) \) and \( M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2) \)

- Define the unioner of \( M_1 \) and \( M_2 \) by:
  \( M_\cup = (Q_1 \times Q_2, \Sigma, \delta_1 \times \delta_2, (q_1, q_2), F_\cup) \)

  - where the accept state \((q_1, q_2)\) is the combined start state of both automata

  - where \( F_\cup \) is the set of ordered pairs in \( Q_1 \times Q_2 \) with at least one state an accept state. That is: \( F_\cup = Q_1 \times F_2 \cup F_1 \times Q_2 \)

  - where the transition function \( \delta \) is defined as
    \( \delta((q_1, q_2), j) = (\delta_1(q_1, j), \delta_2(q_2, j)) = \delta_1 \times \delta_2 \)
Other constructions: Intersector

- Other constructions are possible, for example an intersector:

- Accept only when both ending states are accept states. So the only difference is in the set of accept states. Formally the intersector of $M_1$ and $M_2$ is given by $M_\cap = (Q_1 \times Q_2, \Sigma, \delta_1 \times \delta_2, (q_{0,1}, q_{0,2}), F_\cap)$, where $F_\cap = F_1 \times F_2$. 
Complement

• How about the complement? The complement is only defined with respect to some universe.

• Given the alphabet $\Sigma$, the *default universe* is just the set of all possible strings $\Sigma^*$. Therefore, given a language $L$ over $\Sigma$, i.e. $L \subseteq \Sigma^*$ the complement of $L$ is $\Sigma^* - L$

• Note: Since we know how to compute set difference, and we know how to construct the automaton for $\Sigma^*$ we can construct the automaton for $\overline{L}$.

• Question: Is there a simpler construction for $\overline{L}$?

• Answer: Just switch accept-states with non-accept states.