Prof. Dr. Andreas Podelski
Dominik Klumpp
Frank Schüssele

Hand in until November 3rd, 2021
23:59 via ILIAS
Discussion: November 8th/9th, 2021

Tutorial for Cyber-Physical Systems - Discrete Models

Exercise Sheet 2

Exercise 1: Implications

3 Points

The goal of this exercise is to familiarize yourself with different mathematical phrases.

A major problem in the development of systems is the correct wording of requirements. Often the requirements come in the form of an implication. Take the requirement *"if the battery is empty then the generator is switched on"*. This can be formulated as the implication $A \rightarrow B$ (A *implies* B) with the assertion A: "the battery is empty" and the assertion B: "the generator is switched on". There are other possible formulations of implications, but often engineers don't get it right. Hence the following exercise.

Group the following statements. Two statements should be in the same group if and only if they are equivalent.

(a) A implies B.

(b) B implies A.

(c) A is a necessary condition for B.

(d) A is a sufficient condition for B.

(e) A if B.

(f) A only if B.

(g) A is stronger than B.

(h) A is weaker than B.

(i) $A \land \neg B$ does not hold.

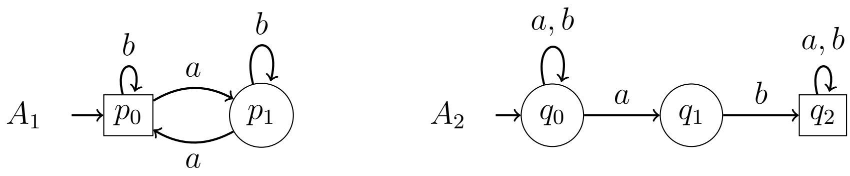
(j) $\neg A \lor B$ holds.

(k) If not B, then not A.

Exercise 2: Intersection of Finite Automata 4 Points

The goal of this exercise is to intersect two finite automata, as we are interested in the intersections in the lecture later on.

Given the two finite automata A_1 and A_2 below, draw a finite automaton A that recognizes the intersection of $L(A_1)$ and $L(A_2)$, i.e. A should accept exactly those words that are accepted by both A_1 and A_2.



Exercise 3: Büchi Automata 9 Points

We practice working with ω-languages and Büchi automata. This serves as preparation for a similar concept in the lecture, so-called transition systems*.*

Consider the following descriptions of two ω-languages over the alphabet $\Sigma = \{a, b\}$.

(L_1) The language of all ω-words that contain only finitely many occurrences of the letter a.

(L_2) The language of all ω-words such that every second letter is an a.

For these two languages, perform the following tasks.

(a) Formally define these languages as sets of words.

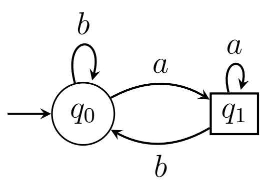
Example: The language L_0 of all words that contain infinitely many occurrences of the letter a can be defined as

$$\begin{aligned} L_0 &= \{\, x_0 x_1 \ldots \mid (\forall i \in \mathbb{N}_0 \,.\, x_i \in \Sigma) \text{ and } \overset{\infty}{\exists} i \in \mathbb{N}_0 \,.\, x_i = a \,\} \\ &= \{\, x_0 x_1 \ldots \mid (\forall i \in \mathbb{N}_0 \,.\, x_i \in \Sigma) \text{ and } \forall j \in \mathbb{N}_0 \,.\, \exists i \in \mathbb{N}_0 \,.\, j < i \text{ and } x_i = a \,\} \end{aligned}$$

(b) For each of these languages, draw a Büchi automaton that recognizes the language. That is, draw a Büchi automaton A_1 that accepts an ω-word w if and only if it contains only finitely many occurrences of the letter a. Similarly, draw an automaton A_2 that accepts an ω-word w if and only if every second letter is an a.

Both automata should have at most 2 states each.

Example: An automaton A_0 that recognizes the language L_0 is shown below:



(c) Describe the automata from exercise *(b)* as a five-tuples.

Example: The automaton A_0 above can be described as $A_0 = (Q_0, \Sigma, \delta_0, Q_0^{\text{init}}, F_0)$ with the set of states $Q_0 = \{q_0, q_1\}$, the set of initial states $Q_0^{\text{init}} = \{q_0\}$, the set of accepting states $F_0 = \{q_1\}$ and the transition relation

$$\delta_0 = \{(q_0, b, q_0), (q_0, a, q_1), (q_1, a, q_1), (q_1, b, q_0)\}$$