Prof. Dr. Andreas Podelski
Tanja Schindler

Hand in until January 19, 2018
11:59 via the post boxes
Discussion: January 22, 2018

Tutorial for Cyber-Physical Systems - Discrete Models

Exercise Sheet 11

The goal of this sheet, similar to the last one, is to get a deeper understanding of automata that can be used to verify certain linear-time properties.

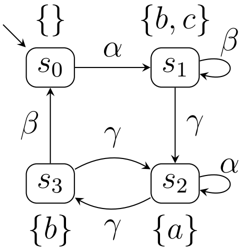
NFAs can be used to verify regular safety properties. We have learned how to read and construct NFAs in the last sheet. In the first part of this sheet, we will see how to apply them to verify regular safety properties.

However, we are also interested in verifying liveness properties. NFAs only accept finite words. This is fine for safety properties as it is sufficient to look at bad prefixes. In contrast, one needs to look at infinite words in order to verify ω-regular liveness properties (and more general ω-regular properties).

Therefore, the goal of the second part of this sheet is to understand Büchi automata. Later on, we will see how to apply these Büchi automata to verify ω-regular properties.

Exercise 1: Checking regular safety properties

Consider the following transition system TS over the atomic propositions $AP = \{a, b, c\}$.



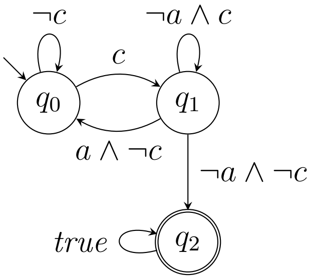
The book describes an algorithm for checking regular safety properties. The safety property E is given as an NFA $\mathcal{A}$ that accepts the bad prefixes of E.

The algorithm first computes the product $TS \otimes \mathcal{A}$ and then checks whether the invariant $\neg F$ holds, where F is the set of final states of $\mathcal{A}$.

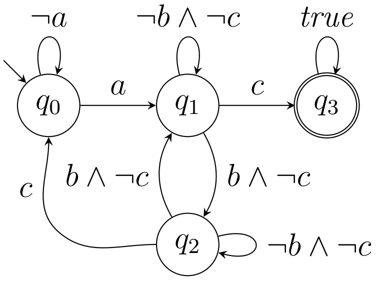
If the invariant holds for $TS \otimes \mathcal{A}$, then the property E holds for TS. Otherwise, the property E does not hold and the algorithm returns a sequence of states of TS as an error indication.

Apply the algorithm to the properties that are given by the following NFA.

(a) $\mathcal{A}_1$:



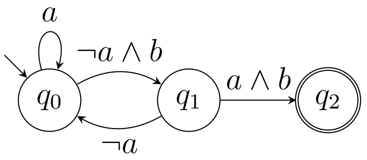
(b) $\mathcal{A}_2$:



Motivation: We have seen in the previous sheet how to construct an NFA for the set of bad prefixes of a regular safety property P. The goal of this exercise is to learn how to construct the product of the transition system TS and the NFA. Then we can apply the invariant checking algorithm that we have seen before (e.g. in its recursive version on Sheet 7).

Exercise 2: Nonblocking symbolic NFA

Consider the following DFA (i.e., deterministic NFA) $\mathcal{A}$ over the alphabet $\Sigma = 2^{AP}$, where $AP = \{a, b, c\}$.



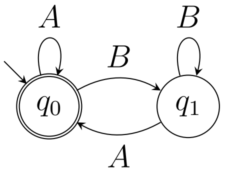
Give a nonblocking DFA $\mathcal{A}'$ such that both automata accept the same language (i.e., $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$).

Motivation: The goal of this exercise is to understand the concept of "non-blocking" and to see that there is a general way to construct a non-blocking DFA from a given DFA, and this in the context of symbolic automata (i.e., with symbolic transitions).

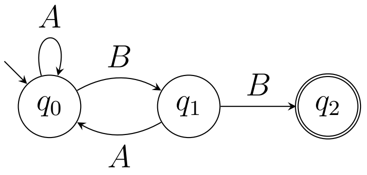
Exercise 3: Büchi automata I

Describe the ω-languages of the following Büchi automata over the alphabet $\Sigma = \{A, B\}$. You may use ω-regular expressions or natural language.

(a)



(b)



Motivation: This exercise is a warm-up for applying Büchi automata to verify ω-regular liveness properties. It aims at learning how to "analyze" a given Büchi automaton, i.e., determine the ω-regular language recognized by a given Büchi automaton.

Exercise 4: Büchi automata II

Construct a Büchi automaton over the alphabet $\Sigma = \{A, B\}$ whose language consists of all ω-words that contain only finitely many A.

Motivation: This exercise complements Exercise 3. It aims at learning how to "synthesize" a Büchi automaton, i.e., construct a Büchi automaton that accepts a given ω-regular language.

Exercise 5: Minimal bad prefixes
Provide an example for a regular safety property P_{safe} over some set of atomic propositions AP and an NFA $\mathcal{A}$ for its *minimal* bad prefixes such that

$$L_\omega(\mathcal{A}) \neq \left(2^{AP}\right)^\omega \setminus P_{\text{safe}}$$

when $\mathcal{A}$ is viewed as a Büchi automaton.

Motivation: The same automaton can be read as an automaton over finite words and as an automaton over infinite words. The goal is to understand that going from one to the other can be more subtle than one might expect. The goal is also to be able to manipulate the notion of a safety property in the context of the two kinds of relevant automata (over finite words and, respectively, over infinite words). If this is not sufficient for motivation: it is rather likely that this exercise will appear in the exam.

Exercise 6: Inclusion
In the algorithm for checking regular safety properties we exploited the following equivalence for languages $L_1, L_2 \subseteq \Sigma^*$ for some alphabet Σ.

$$L_1 \subseteq L_2 \text{ iff } L_1 \cap \overline{L_2} = \emptyset$$

Here we use $\overline{L_2}$ to denote the complement $\Sigma^* \setminus L_2$.

Show that this equivalence holds.

Motivation: The goal of the exercise is to go back to the basics and spell out the proof of something that seems obvious (if it's easy, all the better).