Prof. Dr. Andreas Podelski
Dr. Matthias Heizmann
Christian Schilling

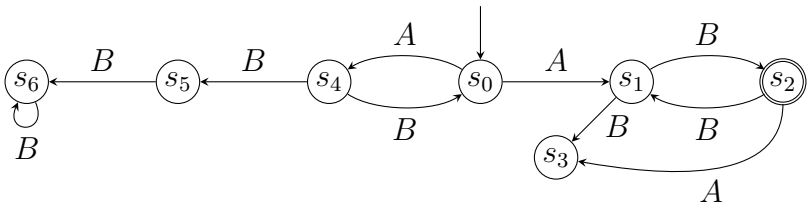
Delivery: January 16th, 2017
16:15 via the post boxes
Discussion: January 18th, 2017

Tutorial for Cyber-Physical Systems - Discrete Models
Exercise Sheet 11

Exercise 1: SCCs and NBA emptiness 2 Points

Given a directed graph, a *connected component* is a set of nodes X such that for each two nodes $x_1, x_2 \in X$ there is a path from node x_1 to node x_2 and vice versa there is a path from node x_2 to node x_1.
A *strongly connected component (SCC)* is a connected component that is *maximal*, i.e., a connected component X such that each strict superset $Y \supsetneq X$ is not a connected component.
We call a strongly connected component *nontrivial* if it contains at least one edge (i.e., if there is one node that has a self loop or if there are at least two nodes).

Consider the following graph representation of an NBA $\mathcal{A}$.



(a) Determine all strongly connected components (SCCs) of this graph.

(b) Does $\mathcal{A}$ accept the word AB^ω?

(c) In the lecture we learned a naive algorithm for checking language emptiness of NBA.

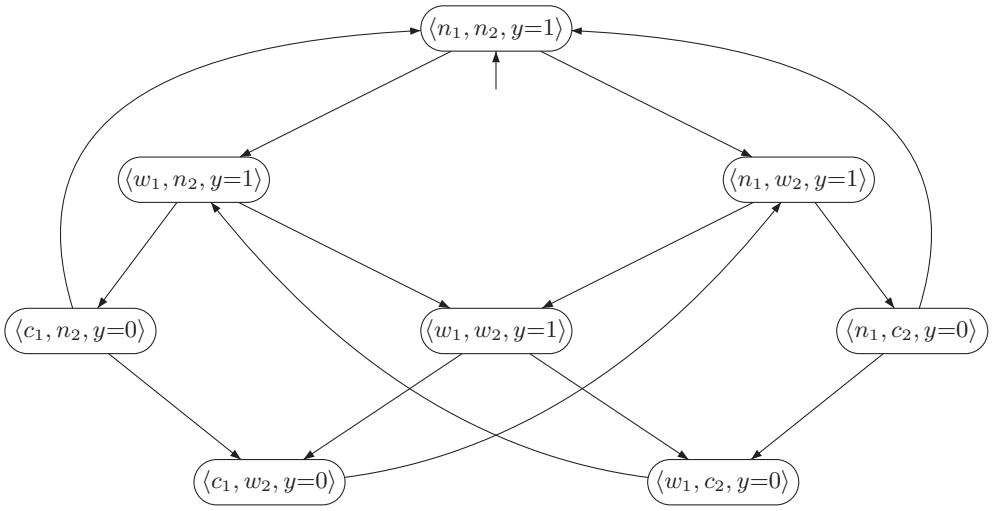
The idea of the algorithm was: Check for each accepting state (1) if the state is reachable from some initial state and (2) if the state is reachable from itself.

Assume that you have given not only the NBA but also all its SCCs. Describe an algorithm for checking language emptiness that is more efficient that the naive algorithm. It is sufficient if your algorithm returns "YES" if the automaton does not accept any word, and "NO" otherwise, the algorithm does not have to provide a counterexample in case the answer is "No".

You do not need to argue why your algorithm is correct.

Exercise 2: Checking ω-regular properties 4 Points

Consider the transition system TS_{Sem} for mutual exclusion with a semaphore below.



Let P_{live} be the following ω-regular property over $AP = \{w_1, c_1\}$:

> "Whenever process 1 is in its waiting location (w_1), it will eventually enter its critical section (c_1)."

Note that the labeling function is given implicitly by the state names, e.g., w_1 holds in all states whose name contains this string.

(a) Depict an NBA $\mathcal{A}$ for P_{live} and an NBA $\overline{\mathcal{A}}$ for the complement property $\overline{P}_{live} = \left(2^{AP}\right)^{\omega} \setminus P_{live}$.

(b) Check if $TS_{Sem} \not\models P_{live}$ holds by performing the following steps that were presented in the lecture.

(i) Depict the reachable fragment of the product $TS_{Sem} \otimes \overline{\mathcal{A}}$.

(ii) Check if this product transition system satisfies the persistence property "eventually forever $\neg F$". In case the product satisfies the property argue why this is the case. Otherwise, give a path in the product that shows the violation of the persistence property and give the corresponding path in TS_{Sem} that shows the violation of P_{live}.

Exercise 3*: Checking ω-regular properties with automata 1 Point

Consider again the transition system TS_{Sem} and the property P_{live} from Exercise 2. Use the web interface of the ULTIMATE Automata Library[1] to check (again) if the transition system TS_{Sem} satisfies the property P_{live}.

Follow the approach outlined on slides 82 f. of lecture 21.

- Construct an NBA $\mathcal{A}_{TS_{Sem}}$ that accepts the traces of TS_{Sem}.
- Construct the NBA $\overline{\mathcal{A}}$.
- Construct an NBA $\mathcal{A}_{\mathsf{inter}}$ that accepts the language $\mathcal{L}(\mathcal{A}_{TS_{Sem}}) \cap \mathcal{L}(\overline{\mathcal{A}})$ (operation `BuchiIntersect`).
- Check if this language is the empty set (operation `BuchiIsEmpty`).

You should submit a file containing the automata declarations and the required operations using the notation from the web interface to your tutor via email.

Hint: the following code can be used to obtain some ω-word that is accepted by the automaton `A_inter`.

```
LassoWord omegaWord = getAcceptedLassoWord(A_inter);
print(omegaWord);
```

[1] `https://ultimate.informatik.uni-freiburg.de/automata_script_interpreter`