Tutorial for Cyber-Physical Systems - Discrete Models
Exercise Sheet 8

Exercise 1: NFA → regular expression  2 Points
Consider the following NFA over the alphabet \( \Sigma = \{A, B\} \), which are given in a graphical representation where accepting states have double circles.

\[(a) \quad \begin{array}{c}
q_0 \quad A \\
\downarrow \quad q_1 \quad A \\
\downarrow \quad q_2 \\
A \\
\end{array}
\quad (b) \quad \begin{array}{c}
q_0 \quad A, B \\
\downarrow \quad q_1 \quad A \\
\downarrow \quad q_2 \quad B \\
A \\
\end{array} \]

Write for each NFA a regular expression such that the language described by the regular expression is the language that is accepted by the NFA.

Exercise 2: Regular expression → NFA  2 Points
Consider the following regular expressions over the alphabet \( \Sigma = \{A, B\} \).

\[(a) \quad (AB + B^*)ABA^*A \quad (b) \quad ((ABAB)^* + AB)^*AB(B^* + \emptyset A) \]

Construct for each regular expression an NFA such that the language accepted by the NFA is the language that is described by the regular expression. You may give your NFA as a five tuple or use the graphical representation.

Alternative way to solve this exercise. In our group, we develop the program analysis framework ULTIMATE\(^1\). One component of this framework is an automata library which is available via a web interface\(^2\). Instead of providing your automata on your solution sheet, you may alternatively use the notation of our automata library and send the automata declarations to your tutor via email. Operations of the automata library like, e.g., the acceptance test (Accepts) or the emptiness test (IsEmpty) may help you to find a correct NFA. The sample file `FiniteAutomata.ats`\(^3\) from the web interface shows how to declare automata and apply operations.

---

\(^1\) [http://ultimate.informatik.uni-freiburg.de/](http://ultimate.informatik.uni-freiburg.de/)
\(^2\) [https://ultimate.informatik.uni-freiburg.de/automata_script_interpreter](https://ultimate.informatik.uni-freiburg.de/automata_script_interpreter)
\(^3\) [http://ultimate.informatik.uni-freiburg.de/?ui=int&tool=automata_script_interpreter&task=automataScript&sample=606706886](http://ultimate.informatik.uni-freiburg.de/?ui=int&tool=automata_script_interpreter&task=automataScript&sample=606706886)
Exercise 3: Symbolic automata  
Consider the following symbolic NFA $\mathcal{A}$ over the alphabet $2^{AP}$ with $AP = \{a, b\}$.

Which of the following words is accepted by $\mathcal{A}$? Give a short explanation.

(a) $w_1 = \{a\} \{\} \{a\} \{b\}$
(b) $w_2 = \{a\} \{\} \{a\}$
(c) $w_3 = \{b\} \{\} \{a, b\} \{a\} \{a\}$
(d) $w_4 = \{a\} \{a\} \{a\} \{\} \{a\}$

Hint: You may use the automata library to double check your answer. If you want to use braces in the symbols of your alphabet, you have to put quotation marks around the string that describes the symbol. E.g., $\text{alphabet} = \{"\{\}" "\{a\}" "\{b\}" "\{a, b\}"\}$ could be a useful declaration of your alphabet.

Exercise 4: Regular safety properties  
Let $AP = \{a, b, c\}$. Consider the following regular safety properties:

(a) $P_1$: If $a$ becomes valid, afterward $b$ stays valid ad infinitum or until $c$ holds.

(b) $P_2$: Between two neighboring occurrences of $a$, $b$ always holds.

Construct an NFA $\mathcal{A}_i$ for each property $P_i$ such that $L(\mathcal{A}_i) = \text{BadPref}(P_i)$.

Hint: You may use a symbolic NFA with propositional formulae over the set $AP$ as transition labels.