Mathematical Methods for Computer Science I

Fall 2016

Series 8 – Hand in before Monday, 21.11.2016 - 12.00

1. Construct an NFA accepting the following languages defined over $\Sigma = \{a, b, c\}$.
   a) Words whose last letter is the same as the first.
   b) Words whose last letter is different from the first.
   c) Words whose third last character is the character b.

2. Consider the following transition table of an NFA, where the start state is marked with an arrow and the accepting state is marked with a star:

δ	0	1
$\rightarrow p$	$\{p, q\}$	$\{p\}$
q	$\{r\}$	$\{r\}$
r	$\{s\}$	$\emptyset$
$* s$	$\{s\}$	$\{s\}$

   a) Describe the language accepted by this NFA.
   b) Convert this NFA into a DFA using the subset construction.

3. * By construction, NFAs have the power to be in several states at once. This makes them typically more succinct and easier to design than DFAs. We have seen that NFAs accept exactly the regular languages which DFAs do. While we can always convert an NFA into a DFA, the latter may have exponentially more states than the NFA, since $|2^Q| = 2^{|Q|}$, which is the maximum number of states of the DFA obtained from transforming an NFA.
   a) Find a non-trivial example of a language whose NFA and DFA, obtained by transforming the NFA, have the same number of states.
   b) Find an example of a language L whose DFA, obtained by transforming the NFA accepting L, has $|2^Q|$ states, where Q is the set of states of the NFA.

4. The aim of this exercise is to directly construct a DFA $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$ from an ε-NFA $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$ without passing via an NFA.
   Using the notions of *subset construction* and the *closure*,
   a) construct the set of states Q_D of D;
   b) define the starting state q_D of D;
   c) define the set of accepting states F_D of D;
   d) define the transition function δ_D of D.
   e) Apply this procedure to the ε-NFA seen in class, which accepts finite decimal numbers (in order to avoid making the DFA excessively huge, you can ignore the empty state which the automaton would reach while reading non-acceptable input):

---

* Exercises with a $\star$ are intended for Discrete Mathematics I students only. However, MMI I students can gain additional bonus points by attempting them.