Languages and Automata

Assignment 3, Tue 18th Feb, 2020

Handing in your answers: There are two options:

1. Brightspace. Before submitting, make sure:
   - the file is a PDF document
   - your name and student number are included in the document (they might be printed).
2. Post box, located in the Mercator building on the ground floor. There will be boxes labelled with *LnA* and the corresponding group teacher's name. Put your work in the post box corresponding to your group. Before putting your solutions in the post box make sure:
   - your name and student number are written clearly on the document.

   There will be 1 box, the *Uitleverbak*, for work that hasn't been picked up at the exercise hours.

Deadline: Fri 28th Feb, 2020, 17:00 (in Nijmegen!). This deadline is strict: submission in brightspace will close at that time.

Goals: After completing these exercises successfully you should be able to construct an NFA from a language description, to construct an NFA-λ from a regular expression, to turn an NFA-λ into an NFA and to determinise an NFA.

There are 4 mandatory exercises, worth **10 points** in total. There are 2 more, extra hard, exercises. Be aware that this exercise is just for fun, you cannot earn any points with it.

1 NFAs and Their Languages

Let $A = \{0, 1, 2\}$ and let L be the set of words in which the last digit occurs at least twice, with no larger digit in between the last two occurences:

$$L = \{w \in A^* \mid \exists x \in A.\ \exists u, v \in A^*.\ w = uxvx,\ \text{and there is no } y \text{ in } v \text{ s.t. } x < y\},$$

where $0 < 1 < 2$. For example, $00, 2111, 2102 \in L$, but $1, 121 \notin L$.

a) Construct an NFA that accepts L. Explain your answer. **(2pt)**

b) Show that, in your automaton from the previous exercise, 2101 is accepted, but 121 not. **(1pt)**

2 NFA-λ

a) Let M be the NFA-λ over the alphabet $A = \{a, b, c\}$ given by the following graph.

$$\begin{tikzcd}
{} \arrow[r] & q_0 \arrow[loop above, "{a,b}"] \arrow[r, "{b,c}"] \arrow[rrr, bend left, "\lambda"] & q_1 \arrow[loop above, "a"] \arrow[r, "\lambda"] & q_2 \arrow[loop above, "{a,b,c}"] \arrow[r, "c"] & q_3
\end{tikzcd}$$

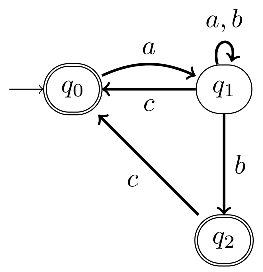
Give an NFA (without λ-transitions) with the same set of states, accepting the same language. **(1pt)**

b) An *NFA-λ with multiple initial states* over an alphabet A is a tuple $\mathcal{A} = (Q, I, \delta, F)$ where $I \subseteq Q$ is a set of initial states; Q, δ and F are the same as with NFA-λ. A word $w \in A^*$ is accepted if there exists a state $q_0 \in I$ such that $\delta^*(q_0, w) \cap F \neq \emptyset$.

Give, for any NFA-λ $\mathcal{A} = (Q, I, \delta, F)$ with multiple initial states, an NFA-λ $\mathcal{A}' = (Q', q_0, \delta', F)$ (with one initial state) such that $\mathcal{A}$ and $\mathcal{A}'$ accept the same words. Explain your answer. **(2pt)**

3 From NFA to DFA

Let M be the NFA over the alphabet $A = \{a, b, c\}$ given by the following graph.



Use the powerset construction from the lecture to turn M into a DFA M' that accepts the same language. Leave out unreachable states, and clearly indicate how a state of M' corresponds to a subset of states of M. **(2pt)**

4 From Regular Expression to NFA-λ

Let e be the regular expression $a(b + a^*)a$.

Use the "toolkit" from the lecture to construct an NFA-λ that accepts $\mathcal{L}(e)$. The (non-trivial) intermediate steps must be given as part of the solution. **(2pt)**

5 Fun Exercises – Properties of Regular Languages

a) Using that regular languages are closed under complement and intersection, show that for regular languages L_1, L_2 their difference $L_1 \setminus L_2$ is regular as well.

b) Give an algorithm that decides whether for a regular expression e its language $\mathcal{L}(e)$ is empty.

c) Give an algorithm that checks for given regular expressions e_1, e_2 whether their languages are equal: $\mathcal{L}(e_1) = \mathcal{L}(e_2)$.

6 Fun Exercise – Constructing an NFA-λ

Give an NFA-λ over $A = \{a\}$ such that it rejects some string and the length of the shortest rejected string is strictly greater than the number of states.