Talen en Automaten

Assignment 2, Tue 21st Nov, 2017

Exercise teachers. The student groups are supervised by the following teachers:

Teacher	E-Mail	Room	Time
Michiel de Bondt	`M.deBondt@math.ru.nl`		8:45 – 10:30
Demian Janssen	`wd.janssen@student.ru.nl`		8:45 – 10:30
Leon Gondelman	`lgondelmann@gmail.com`		8:45 – 10:30
Tom van Bussel	`tom.van.bussel@student.ru.nl`		8:45 – 10:30
David Venhoek	`david@venhoek.nl`		8:45 – 10:30
Alexis Linard	`A.linard@cs.ru.nl`		8:45 – 10:30
Bas Steeg	`bas.steeg@student.ru.nl`		10:45 – 12:30
Ties Robroek	`ties.robroek@student.ru.nl`		10:45 – 12:30
Jan Martens	`j.martens@student.ru.nl`		15:45 – 17:30

Postboxes are located in the Mercator building on the ground floor There will be boxes labelled with *Talen en Automaten* and the corresponding group teacher's name. There will be 1 box, the *Uitleverbak*, for work that hasn't been picked up at the exercise hours.

Handing in your answers: There are two options:

1. E-mail: Send your solutions by e-mail to your exercise class teacher (see above) with subject "**T&A: assignment 2**". This e-mail should only contain a single PDF document as attachment (unless explicitly stated otherwise). Before sending an e-mail make sure:
   - the file is a PDF document
   - your name is part of the filename (for example MyName_assignment-2.pdf)
   - your name and student number are included in the document (they will be printed).
2. Post box: Put your solutions in the appropriate post box (see above). Before putting your solutions in the post box make sure:
   - your name, student number, and IC, KI or Wiskunde are written clearly on the document.

Deadline: Tue 28th Nov, 2017, 8:45 (in Nijmegen!)

Goals: After completing these exercises successfully you should be able to construct an automaton from a description of a language, describe the language of a basic automaton, perform the complement and product constructions, and compute a regular expression from an automaton.

There are 3 mandatory exercises, worth **10 points** in total. There is 1 more, extra hard, exercise. Be aware that this exercise is just for fun, you cannot earn any points with it.

1 DFAs and Their Languages

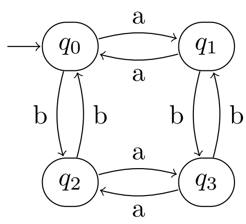
a) Let L be the following language over $A = \{a, b\}$. **(3pt)**

$$L = \{w \in A^* \mid |w|_a \text{ is } \textit{not} \text{ divisible by 3 and } w \text{ ends with } a\}$$

Use the constructions for product and complement automata, given in the lecture, to construct an automaton M with $\mathcal{L}(M) = L$.

b) Which of the words $abaa$, ba are accepted by your automaton from the previous answer? Justify your answer with accepting or rejecting computations. **(1pt)**

c) We define a family of DFA over the alphabet $A = \{a, b\}$ by letting the states Q, the transition map δ and the initial state q_0 be as in the following graph.



Describe explicitly the languages L_1, L_2, L_3 accepted by the automata $M_i = (Q, q_0, F_i, \delta)$ with accepting states $F_1 = \emptyset$, $F_2 = \{q_0\}$ and $F_3 = \{q_1, q_2\}$, respectively. **(3pt)**

2 DFAs and Regular Expressions

Let M be the DFA given by:

- set of states $Q = \{q_0, q_1, q_2\}$
- initial state q_0
- set of final states $F = \{q_0, q_2\}$
- transition function δ given by

δ	a	b
q_0	q_1	q_0
q_1	q_0	q_2
q_2	q_0	q_1

a) Draw a state/transition diagram for the automaton M. **(1pt)**

b) Construct a regular expression e such that $\mathcal{L}(M) = \mathcal{L}(e)$. **(2pt)**

3 Fun Exercises – Regular Languages and DFAs

In this exercise, you will show that regular languages can be characterised in even another way, and you will show some properties of this characterisation. On the one hand should this exercise especially appeal to people with some background in algebra, but it can be solved without any deep knowledge there. On the other hand, the Myhill-Nerode is also important for automata minimisation, giving us a way to optimise recognisers for regular languages.

Let A be a finite alphabet. We define for each $a \in A$ a map $(-)_a : \mathcal{P}(A^*) \to \mathcal{P}(A^*)$ on the languages over A by

$$L_a = \{w \in A^* \mid aw \in L\},$$

called the *language derivative* (with respect to a) of L. This map can be extended inductively to words by

$$L_\lambda = L \qquad L_{aw} = (L_a)_w.$$

Finally, we denote by $\ker L$ the *kernel* of a language, which is given by

$$\ker L = \{L_w \mid w \in A^*\}.$$

a) Show that a language L over A is regular if and only if $\ker L$ is finite. [Hint: Use that regular languages are exactly those that are accepted by finite DFAs, even though this has not been fully proved in the lecture, yet.]

Next, we define for $L \subseteq A^*$ an equivalence relation $\sim_L$ on words by

$$w \sim_L v \iff (\forall u \in A^*.\ wu \in L \Leftrightarrow vu \in L).$$

You can just assume that this is an equivalence relation, so we can form equivalence classes of words

$$[w]_L = \{v \in A^* \mid w \sim_L v\}$$

and thus the quotient

$$A^*/{\sim_L} = \{[w]_L \mid w \in A^*\}.$$

b) Show, using the previous exercise, that L is regular if and only if $A^*/{\sim_L}$ is finite. This is known as the Myhill-Nerode theorem.

c) Show that $\ker L$ is a monoid for any language, that is, there is an associative, binary operation with a unit on $\ker L$.

d) Let L be given by

$$L = \{w \in \{a, b\}^* \mid |w| \text{ is even}\}.$$

Show that $\ker L$ is actually a group, that is, every element has an inverse with respect to the operation defined in the last exercise.