Theory of Computer Science

G. Röger
S. Eriksson
Spring Term 2023

University of Basel
Computer Science

Exercise Sheet 2

Due: Wednesday, March 15, 2023

Exercise 2.1 (JFLAP; 1 point)

This exercise is about getting familiar with the program JFLAP. We will use it on this sheet and in the future for several exercises as it covers a wide range of topics in the area of theoretical computer science.
In JFLAP you can define and simulate formal languages and different automata that we will cover in the lecture. The program also contains several algorithms from the lecture that you can use to convert between different models of computation.
Unfortunately, there are also some differences to the definitions from the lecture. For example, by default JFLAP uses λ instead of ε for the empty word and for NFA transitions that do not consume a symbol. This can be changed in the menu Preferences. For DFAs JFLAP assumes that all non-defined transitions go to a "trap state". According to our definition, this is not possible because for every combination of a state and a symbol there has to be a defined successor state. However, it is possible to create the trap state explicitly in JFLAP (Convert → Add Trap State to DFA). In general, the definitions from the lecture have preference in all cases with differences.
Furthermore, JFLAP offers a lot of support with executing algorithms. Depending on the algorithm there are functions to execute a single or all steps of the algorithm automatically. The practice effect is obviously small when using this. We thus recommend to use these automated steps only to check your solution or when you are stuck. You should be able to solve all exercises on paper without this support. Remember that this kind of automated support will not be available in the exam.
For submissions of exercises with JFLAP, please create an easy-to-read layout (avoid overlapping edges and make sure edge labels are readable). Then create an image (File → Save Image As → PNG) and include it in your submission. Please also submit your JFLAP file. A clearly defined filename is important so we can automatically analyze these files. Please use exactly this format: `ex1a.jff`, `ex2a.jff`, `ex2b.jff`, ... Please do not create subdirectories in your submission.

(a) Download JFLAP from `https://www.jflap.org/` and start it.

(b) Work through the tutorial on finite automata under `https://www.jflap.org/tutorial/` up to the section "Convert to DFA".

For this exercise, no answer is necessary. Just confirm that you completed the tutorial.

Exercise 2.2 (DFA; 1+1+2 points)

Consider the DFA $M = \langle \{q_0, q_1, q_2, q_3\}, \{\mathtt{a}, \mathtt{b}\}, \delta, q_0, \{q_2\}\rangle$ with transition function

$$\begin{aligned}\delta = \{&\langle q_0, \mathtt{a}\rangle \mapsto q_1, \langle q_0, \mathtt{b}\rangle \mapsto q_2, \langle q_1, \mathtt{a}\rangle \mapsto q_0, \langle q_1, \mathtt{b}\rangle \mapsto q_3,\\ &\langle q_2, \mathtt{a}\rangle \mapsto q_3, \langle q_2, \mathtt{b}\rangle \mapsto q_0, \langle q_3, \mathtt{a}\rangle \mapsto q_2, \langle q_3, \mathtt{b}\rangle \mapsto q_1\}.\end{aligned}$$

(a) Create a graphical representation of M in JFLAP.

(b) Specify the sequence of states visited with input `abbab`. Is the word accepted?

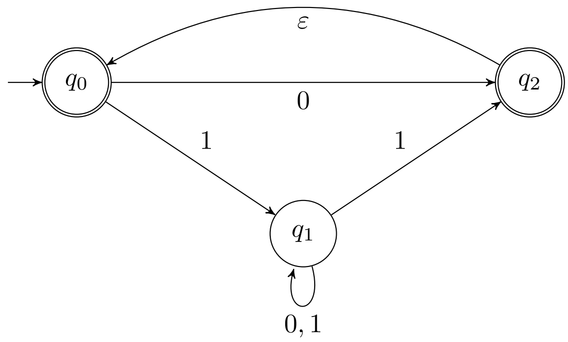
(c) Describe in natural language which language M recognizes.

Exercise 2.3 (DFAs; 2 points)

Specify a DFA in JFLAP that recognizes the language $L = \{a^{n_1} b^{n_2} c^{n_3} \mid n_1 \geq 1, n_2 \geq 0, n_3 \geq 2\}$.

Exercise 2.4 (DFA and NFA; 1+2 points)

Consider the following NFA:



(a) Does the NFA accept the word 0101010? Justify your answer.

(b) Specify a DFA in JFLAP that is equivalent to the NFA.

Hint: The proof to the theorem by Rabin and Scott shows one possible way to construct a DFA from an NFA, but you can omit states if they cannot be reached from the initial state.