KØBENHAVNS UNIVERSITET

Diskret Matematik og Formelle Sprog: Problem Set 5

Due: Wednesday March 30 at 09:59 CET.
Submission: Please submit your solutions via *Absalon* as PDF file. State your name and e-mail address close to the top of the first page. Solutions should be written in LaTeX or some other math-aware typesetting system with reasonable margins on all sides (at least 2.5 cm). Please try to be precise and to the point in your solutions and refrain from vague statements. Make sure to explain your reasoning. *Write so that a fellow student of yours can read, understand, and verify your solutions.* In addition to what is stated below, the general rules for problem sets stated on *Absalon* always apply.
Collaboration: Discussions of ideas in groups of two to three people are allowed—and indeed, encouraged—but you should always write up your solutions completely on your own, from start to finish, and you should understand all aspects of them fully. It is not allowed to compose draft solutions together and then continue editing individually, or to share any text, formulas, or pseudocode. Also, no such material may be downloaded from the internet and/or used verbatim. Submitted solutions will be checked for plagiarism.
Grading: A score of 120 points is guaranteed to be enough to pass this problem set.
Questions: Please do not hesitate to ask the instructor or TAs if any problem statement is unclear, but please make sure to send private messages — sometimes specific enough questions could give away the solution to your fellow students, and we want all of you to benefit from working on, and learning from, the problems. Good luck!

1 (60 p) Consider the regular expression

$$a^*(b|c)\big(d^*|ef\big)^*$$

and determine which of the words below belong to the language generated by this regular expression. Motivate your answers briefly but clearly by explaining for each string how it can be generated or by arguing why this is impossible.

1. *defdefdef*
2. *aaaac*
3. *cdef*
4. *aaabcef*
5. *aabddeeff*
6. *aabdddefefd*

2 (90 p) Consider the following context-free grammars, where a, b, c are terminals, B, S are non-terminals, and S is the starting symbol.

Grammar 1:

$$S \to aS \tag{1a}$$
$$S \to B \tag{1b}$$
$$B \to bcB \tag{1c}$$
$$B \to \tag{1d}$$

Grammar 2:

$$S \to aS \tag{2a}$$
$$S \to BS \tag{2b}$$
$$S \to B \tag{2c}$$
$$B \to bcB \tag{2d}$$
$$B \to \tag{2e}$$

Grammar 3:

$$S \to aS \tag{3a}$$
$$S \to BS \tag{3b}$$
$$S \to B \tag{3c}$$
$$B \to bBc \tag{3d}$$
$$B \to \tag{3e}$$

Which of these grammars generate regular languages? For each grammar, write a regular expression that generates the same language (and explain why), or argue why the language generated by the grammar is not regular. In your regular expressions, please use *only* the concatenation, alternative (|) and star (*) operators, and not the syntactic sugar extra operators that we just mentioned briefly in class but never utilized.

3 (80 p) In this problem, we want to write regular expressions and context-free grammars generating specified languages.

3a (40 p) Write a regular expression for the language consisting of all finite (possibly empty) bit strings (i.e., over the alphabet $\{0, 1\}$) that do not contain any consecutive 0s. Examples of such strings are ε (the empty string), 010, 10110111, and 111, whereas 100 and 010010 do not qualify for membership. For partial credit (if your regular expression is wrong or missing), argue why this language is clearly regular.

3b (40 p) Give a context-free grammar for the language $\left\{(a|b)^n(c|d)^n \,\middle|\, n \in \mathbb{N}\right\}$. Examples of strings in this language are ε (the empty string), *aaacdc*, and *abbadddc*, whereas *abc*, *ababcdcdc*, and *abccddba* do not make the cut.

4 (120+ p) In this problem we want to construct a deterministic finite automaton (DFA) that recognizes the language generated by the regular expression $((ab^*)|(aac))^*$.

4a (60 p) Translate this regular expression to a nondeterministic finite automaton (NFA) using the method from Mogensen's notes that we learned in class. Make sure to describe clearly which part of the regular expression corresponds to which part of the NFA, and which states are accepting. Please do not take any shortcuts without explaining what you are doing.

4b (60 p) Translate the nondeterministic finite automaton to a deterministic finite automaton using the method from Mogensen's notes that we learned in class. Make sure to explain in detail how you perform the subset construction, so that it is possible to follow your line of reasoning. Present clearly the resulting DFA, with all states and transitions, and specify which states are accepting.

4c *Bonus problem (worth 40 p extra; might be hard):* How small a DFA can you produce that accepts precisely the language generated by the regular expression above? Can you prove that the size of your DFA is optimal? Note that that you can solve this problem even if you did not solve the other subproblems above.