National Technical University of Athens

School of Electrical and Computer Engineering

Foundations of Computer Science 2024-25

1st set of exercises

(automata – formal languages – grammars)

Exercise 1.

Construct a DFA, a regular expression and a regular grammar for each of the following languages:

(a) Set of strings of $\Sigma_1 = \{0, 1\}$ which are binary representations of integer powers of 8.

(b) Set of strings of $\Sigma_2 = \{a, b\}$ that do not contain the string $'abba'$.

(c) Set of strings of $\Sigma_2 = \{a, b\}$ that have an odd number of $'a'$, each followed by at least two $'b'$.

Exercise 2.

(a) Design an NFA$_\varepsilon$ that accepts strings of $\Sigma_1 = \{0, 1\}$ that are of the form $((00)^*(11) + 01)^*$.

(b) Convert the NFA$_\varepsilon$ of (a) to DFA.

(c) Minimize the DFA of (b) or prove that it is minimal.

(d) Give a regular grammar for this language.

Exercise 3.

(a) Design an efficient algorithm that takes as input a DNA sequence S (i.e. a string from the alphabet $\Sigma = \{A, C, G, T\}$) and a string $s \in \Sigma^*$ and outputs whether s is contained in S.

Your algorithm should generate a suitable DFA in a manner similar to the one shown on slide 20. Describe the steps of the algorithm in detail, using pseudocode.

Note: Your algorithm should construct the DFA "directly", without using the method on slide 36 (i.e. not by constructing the NFA first).

(b) [Optional - bonus] Implement the algorithm in a programming language of your choice and test its operation with appropriate inputs.

Note: Consider that S may contain up to 10^6 symbols, while s may contain up to 100 symbols.

Exercise 4. Write a regular expression for the following language. Justify the correctness of your expression.

The set of strings from the alphabet $\Sigma = \{\ [\ ,\ ]\ \}$ with an equal number of occurrences of «[» and «]», in which there is no prefix containing more «]» than «[». Also, no prefix cannot contain three more «[» than «]». What does this language describe?

Hint: if it is more convenient, you may design an automaton first and find the regular expression corresponding to it.

Exercise 5. Prove that the language defined as in the previous exercise, but without the second constraint (i.e. there can be any more «[» than «]» in the prefix) is not regular. Describe an appropriate automaton and grammar for this language. What does this language describe?

Exercise 6.

Are the following languages regular? If a language is not regular, prove it using either the Pumping Lemma, a closure property, or a combination of both. If a language is regular, write a regular grammar or design the NFA/DFA.

(a) Language $L_1 = \{(a^n b^3)^2 : n \in N^*\}$

(b) $L_2 = \{a^n b^m c^k : n, m, k \in N^*$ such that $n \neq m$ and $n \neq k\}$

(c) $L_3 = \{a^n \mid n$ is a perfect square$\}$

(d) $L_4 = \{0^n \mid n$ is a prime number$\}$

(e) $L_5 = \{a^i b^j \mid \text{gcd}(i, j) = 1\}$

Exercise 7.

Show that the following languages are regular if $L \subseteq \Sigma^*$ is regular:

(a) $\text{Pre}(L) = \{w \in \Sigma^* \mid \exists x \in \Sigma^* : wx \in L\}$ (i.e., $\text{Pre}(L)$ is the language of the prefixes of stings of L).

(b) $\text{Suf}(L) = \{w \in \Sigma^* \mid \exists x \in \Sigma^* : xw \in L\}$ (i.e., $\text{Suf}(L)$ is the language of the suffixes of strings of L).

Exercise 8.

(a) Describe in words the language that is generated by each of the following grammars:

$G_1 : S \to 0S0 \mid A, A \to 1A0 \mid 10.$

$G_2 : S \to aaSbbb \mid aaacbb.$

(b) Develop concise grammars for the following context-free languages:

$L_1 = \{0^i 2^j x \mid x \in \{1, 2\}^*, j \geq 0, i > 3j + 2|x|\}.$

$L_2 = \{a^{2i} b^{i+2} c^{3j+1} d^j \mid i > 0, j \geq 0\}$

Deadline for submission and instructions. Your deadline for submission is 10/11/2024, exclusively through Helios (make sure the final file is <5MB in total).

It is strongly recommended that you dedicate enough time to solving the exercises on your own before seeking assistance from external sources (such as the internet, literature, or discussions with peers). In any case, the solutions should be strictly *individual*.

To be graded, you will need to briefly present your solutions on a date and time that will be announced later.

For questions/clarifications: send an email to `focs@corelab.ntua.gr`.