CSc 30400 Introduction to Theory of Computer Science

2nd Homework Set

Due Date: 3/4

Instructions:

- This homework set covers whole Chapter 1.
- Submit your solutions by email or hand them in class **before the beginning of the class!**
- Prepare your solution either in electronic format or on paper. Using JFLAP earns a 5% credit, thus you are very encouraged to use it. You should create jflap files for all the finite automata and the regular expressions that you are asked to create. You should prepare your solutions on your own, **do not use JFLAP's automatic feature!**
- If your electronic solutions consist of more than one files, compress all your files into one file, name it using your first and last name and the homework set number and send the zipped file in an attachment. Try to give indicative names to your files (for example easy_ex1_1.jff). You should also include a word or pdf report. This file should contain solutions for all the questions that require writing a proof (or text in general) and any comments you want to make regarding the answers in the jflap files (example Easy questions, Question 1: see files easy_ex1_1.jff to easy_ex1_10.jff).
- Exercises are divided into three sections: Practice, Easy and Hard. Practice questions just help you understand the material. "Easy questions" should be relatively easy (though not all easy questions have the same difficulty)! Hard questions are a bit tougher. All the questions earn the same amount of points.

1 Practice questions

P 1. Give the language on the alphabet $\Sigma = \{0, 1\}$ that each of the following regular expressions represent

$r_1 = 0^+1$

$r_2 = 11 + 01$

$r_3 = 1(\varepsilon + (0 + 1)1)$

$r_4 = 1^*01$

P 2. Give a regular expression that represents the following language on $\Sigma = \{0, 1\}$.

$L_1 = \{10, 110, 111\}$

$L_2 = \{w | w \text{ starts with } 110\}$

$L_3 = \{w | w \text{ contains an even number of 1s}\}$

$L_4 = \{w | w \text{ ends with } 01\}$

2 Easy questions

E 1. For each of the first 5 DFAs of exercise E 1 of homework set 1 give equivalent regular expressions. In some cases it might help you if you used the conversion algorithm we discussed in class but you don't have to include the intermediate steps in your solution, just the regular expressions.

E 2. Convert the regular expressions of exercise P 1 of this homework set into equivalent NFA_ε (you don't have to include all the intermediate steps, just the final answer).

E 3. Prove or disprove: the following regular expressions represent the same language on the alphabet $\Sigma = \{0, 1\}$.

(a) 0^*1^* and $(01)^*$

(b) $(0 + 1)^*$ and $(01)^*$

(c) $(0^*1)^*$ and $(0^*1^*)^*$

(d) $(0^*1^*)^*$ and $(0 + 1)^*$

(e) $(0 + 1)^*$ and $(0 + 1)^*(00((10)^*1)^+010)^*$

E 4. Lexical analysis is the very first step in compilation. During the lexical analysis we can identify "tokens" (small groups of symbols) that make up a program.

Consider (for simplicity) the following set of symbols:

- Alphanumeric: $a - e$, $0 - 3$,
- The blank space: $\_$
- The characters: / and #,

and the following examples of tokens over those symbols:

- **Integers:** An integer is any sequence of digits $0 - 3$ that either does not start with zero, or, if it starts with zero, it contains only one digit (the zero itself). For instance, 123, 0 are valid integers but 012 is not.
- **Identifiers:** An identifier starts with any letter $a - e$. After that it can contain any alpha-numeric symbol ($a - e$ and $0 - 3$). For instance abe, $bc1$, $dae32b$ are all valid identifiers but 31, a_3, $3b$, $/ad$ are not.
- **Whitespaces:** Whitespaces include any number of blank spaces.
- **Comments:** A comment starts with /# and ends with #/ with anything in between except a comment ending sequence (#/). For instance
  - $/\#abcde\#/$,
  - $/\#\#\#\#\_ab\_ae\#\#\#/$,
  - $/\#ace\_\#\#\_/\_\#\#\_aaa\#/$

  are all valid comments. But
  - $\#/acbd\#$,
  - $/\#bbb$,
  - $/\#cde\_\#/aba\_\#/$

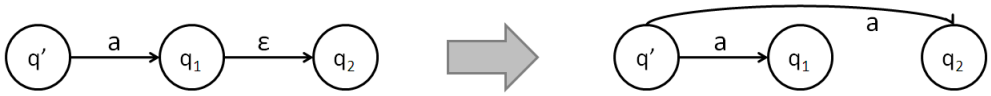
  are not.

Write a regular expression that represents each class. For the forth class you might find it helpful first to design a finite automaton and then to convert this to an equivalent regular expression.

3 Hard questions

H 1. In order to prove that NFA_ε are equivalent to NFA we should eliminate the ε-moves. Your book presents one way to do this.

Consider the following alternative way: For every ε-move $\delta(q_1, \varepsilon) = q_2$, like the one shown in the figure, eliminate the ε-move and for every transition $\delta(q', a) = q_1$ add also the transition $\delta(q', a) = q_2$.



Is this transformation completely correct? What is missing?

H 2. Decide (giving a proof) whether the following languages on the alphabet $\Sigma = \{0, 1\}$ are regular or not.

$L_1 =$ $\{w | w$ contains an equal number of 0s and 1s$\}$.
Example: 011001 belongs in the language but 001 doesn't.

$L_2 =$ $\{w | w$ contains an equal number of occurrences of the substrings 01 and 10$\}$.
Example: 010 belongs in the language but 0101 doesn't.