Department of Computer Science
COMP–421 Compiler Design
Practice Exercise Set 1

1 Objectives

- You will gain experience with lexical analysis concepts such as regular expressions and finite automata
- You will familiarize yourself with the various notations that are used to specify regular expressions and languages

2 Exercises

1. ALSU07 Exercise 3.1.1, page 114
2. In a string of length n how many of the following are there?
   (a) prefixes
   (b) suffixes
   (c) substrings
3. Consult the language reference manuals for the C programming language to determine the set of characters that form the input alphabet, the lexical form of the numerical constants and the lexical form of identifiers.
4. Let L be the set of words {A,B,...,A,a,b,...z} and let D be the set of words {1,2,3,4}. Describe the set that is obtained by the following operations on these languages:
   (a) L ∪ D

Prepared by Dr Dionysiou

(b) LD

(c) LL

(d) L(L ∪ D)

5. Describe the languages denoted by the following regular expressions:

   (a) 0(0|1)*0

   (b) ((ϵ|0)1*)*

   (c) 0*A0*A0*A0*

   (d) (a*b*)*

   (e) (1|0)*

6. Write a regular expression (or definition) for each of the following languages. For each case, specify the input alphabet as well.

   (a) Strings over the alphabet $\{a, b, c\}$ where the first a precedes the first b

   (b) Comments consisting of a string surrounded by /* and */

   (c) All strings of 0's and 1's with an even number of 0's , with at least two 0's.

   (d) $\{w \in \{a, b\}^* | w$ starts with a and contains bba as a substring $\}$

   (e) $\{w \in \{0, 1\}^* | w$ contains 111000 as a substring $\}$

   (f) $\{w \in \{0, 1\}^* | w$ consists of alternating 0's and 1's $\}$

7. Explain why there is no regular expression defining strings of $a's$ and $b's$ where there are more $a's$ than $b's$.

8. Construct NFA for the following regular expressions using Thompson's construction algorithm. Show all the intermediate NFA's as well as the transition table for the final NFA. In addition, show the sequence of moves made by each in processing the input string *ababbab*

   (a) $(if|then|else)$

   (b) $(a|b)^*$

   (c) $(a^*|b^*)^*$

(d) $(a|b)^*abc^*$

9. Convert the NFAs in exercise 7 into DFAs using the subset construction algorithm. Show all intermediate steps as well as the transition table for the DFA. In addition, show the sequence of moves made by the DFA in processing the input string *ababbab*