

COMS W4115

Programming Languages and Translators

Homework Assignment #1

Submit solutions electronically on

Courseworks/COMSW4115/Assignments

by 2:40pm, February 13, 2013

Instructions

- You may discuss the questions with others but your answers must be your own words and your own code.
- Problems 1-4 are each worth 25 points.
- Solutions to these problems will be posted on Courseworks on February 20, 2013.
- This assignment may be submitted electronically on Courseworks by 2:40pm, February 20, 2013 for 50% credit.

Problems

1. Construct Lex-style regular expressions for the following patterns. For each pattern, create a Lex program to find the first longest lowercase word in the dictionary satisfying that pattern. As part of your answer list your program and the longest word found. State what dictionary you used (e.g., on Linux systems `/usr/dict/words`). You can use any variant of Lex such as Flex, JLex, Ocamllex, PLY, etc.
 - a. All lowercase English words that can be made up using only the letters in your first name. Each letter in your name can be used zero or more times and in any order in a word.
 - b. All lowercase English words that can be made up using exactly the letters in your first name. Each letter in your name must be used exactly the number of times it appears in your name but the letters can appear in any order in a word. If your name has more than five letters, use the prefix of length five.
2. Let L be the language consisting of all strings of a 's, b 's, and c 's such that each string is of the form $abxba$ where x does not contain ba as a substring. These strings model comments in the programming language C.
 - a. Write down a deterministic finite automaton (DFA) for L .
 - b. Minimize the number of states in your DFA. Briefly explain how you did the minimization.
 - c. Explain what property of the scanned input each state of your minimized DFA recognizes.
 - d. Write down a regular expression for L .
3. Let L be the language consisting of just balanced parentheses, i.e., $\{\epsilon, (), (()), () () , ((())) , (() ()) , \dots\}$
 - a. Write down a recursive definition for L .
 - b. Use the pumping lemma for regular languages to show that L cannot be specified by a regular expression.
4. Let R be a regular expression of length m and let w be an input string of length n . Briefly discuss in terms of m and n the time-space complexity of the McNaughton-Yamada-Thompson algorithm to determine whether w is in $L(R)$.