COSC 4785
Compiler Construction I, Exam 1
September 22, 2022
Due: Beginning of class Tuesday (0800), September 27, 2022
SUBMIT ON WYOCOURSES

Questions marked with ‡ MAY have scanned images included with their answer.

This is a take home examination. You are expected to do your own work and not discuss this examination with others. You may use the course text and the texts that I have provided on the course site. Even though I cannot stop you from using the World Wide Web, I prefer that you NOT use it. I expect that you will provide YOUR OWN ANSWERS to all the questions and not verbatim or poorly paraphrased copies of something you found.

Read each question carefully and completely before answering. If you have any questions concerning the exam, ask the instructor. Your grade will be based on the number of questions answered correctly and partial credit may be awarded. There are 10 questions worth a total of 100 points on this exam.

****

Question 1: [10 points]

Consider the language over the alphabet $\Sigma = \{0, 1\}$ consisting of all strings which begin with "11" and end with "000".

(a) [4 points] Write a regular expression for this language (see Section 3.3.3). NOTE: Character classes are NOT part of regular expressions.

(b) [6 points] Write an **unambiguous** grammar for the same language.

Question 2: [6 points]

Why is it important to understand the distinction between l-values and r-values when creating an intermediate code representation?

Question 3: [4 points]

What information is normally stored in a symbol table? I do not want to know HOW it is stored but you cannot just wave your hands with this.

Question 4: [16 points]

You are given the following grammar:

$$\begin{array}{lll} \mathbf{M} & \rightarrow & \mathbf{MM} \\ & | & \mathbf{ab} \\ & | & \mathbf{a} \\ & | & \mathbf{b} \end{array}$$

(a) [8 points] ‡ Construct a parse tree for the string **aaba**.

(b) [8 points] ‡ Is this grammar ambiguous? **Prove** your answer.

Question 5: [10 points]

Discuss how, for a block-structured language, a symbol table lookup for an identifier would be done. I do not want any code or pseudo-code. Just explain it. Of course that will mean explaining your basic symbol table implementation. This requires that you think not just try to find the answer.

Question 6: [10 points]

‡ Draw a transition diagram for the the keywords { **dog**, **doggerel**, **ogre**, **gerund**, **underdog**} and compute the *failure* function. The failure function should be in the form of a table. A VERY READABLE typewritten table.

Question 7: [18 points]

Discuss the following. Tell what part each plays in the compilation process and include approximately where in the compilation process each would (or could or should) be done.

(a) [6 points] Semantic analysis

(b) [6 points] Syntactic analysis

(c) [6 points] Lexical analysis

Question 8: [6 points]

One of the elements most used in a Flex program is a *character class*. What is a *character class* and how is it used? I **do not** want any code examples. Just give an organized discussion.

Question 9: [10 points]

‡ Create an NFA for the regular expression $a^*(ab)^+b(ab)^*$. Follow the algorithm given for this and do NOT try to create a DFA.

Question 10: [10 points]

We have briefly discussed using 3-address code as the intermediate code representation in a compiler.

(a) [5 points] Why do we bother with intermediate code representations.

(b) [5 points] What makes 3-address code very reasonable to use for this representation?