**COMP603: MIDTERM I**

NAME: _____

Complete within 120 minutes. Read each question carefully. Write legibly and check your work. No calculators, phones, or laptops are allowed. Good luck!

## 1. Short Definitions

Correctly define 8 of the following terms for full credit. Correctly define all for extra credit.

(1) String: a sequence of character

(2) Language: set of strings

(3) Compiler: transform source language into a target language

(4) Interpreter: accepts code and runtime data and runs with it. We're not generating code.

(5) Bootstrapping: a compiler written in the language it compiles

making a compiler "self hosting"

(6) Visitor: visit (traverse) nodes in a tree to do some computation, without mixing computation into the nodes themselves

(7) Nondeterminism: having more than one option about which static to

(8) Ambiguity: _____

(9) First set: the set of terminals (excluding empty string) that can appear first in any derivation of a nonterminal.

(10) Follow set the set of terminals (excluding empty string) that can appear first after derivation of a nonterminal.

---

## 2. Lists

Complete 3 of the following lists for full credit. Complete all for extra credit.

(1) Compiler phases, in order. Briefly describe what each phase does.

(a) Scan / Tokenize / Lexical analysis: Split source code into small chunks (tokens) such as identifiers, reserved words, literals, operators, etc.

(b) Parse:Check the syntax of the source code

(c) Translate: Translate low level syntax into high-level abstract syntax tree

(d) Optimize:Improve performance or structure

(e) Generate code: Traverse the AST to generate code.

(2) Primitive regular expressions. Briefly describe what each regular expression matches.

(a) Empty Set: Reject everything.

(b) Empty String:Match the empty string.

(c) Symbol: Match a single character.

(d) Sequence: Match regex a followed by regex b.

(e) Alternation: Match regex a or match regex b, but not both.

(f) Kleene Star:Match regex a zero or more times {"",a,aa,aaa,...}

(3) Finite automaton elements. Describe each.

(a) _____

(b) _____

(c) _____

(d) _____

(e) _____

(4) For a grammar to be LL(1),[1] it must be:

(a) _____

(b) _____

(c) _____

(d) _____

## 3. Fill in the blank

Complete the following statements for full credit.

(1) A pushdown automaton is a finite automaton with _____

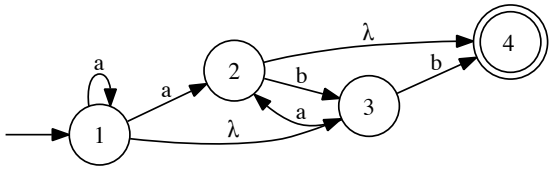
(2) A Turing machine is a finite automation with _____

(3) It is _____ possible to define an NFA which cannot be converted into a DFA.

[1] Left-right, Leftmost derivation, 1 token lookahead

## 4. Regular languages

Refer to the Figure below. Answer 3 of the following questions. Answer all for extra credit.




(1) What is the initial state of the DFA using subset construction?

(2) Draw the equivalent DFA using subset construction.

(3) Write the equivalent regular expression.

(4) IPv4 addresses are written as four integers, separated by dots (e.g., `173.203.204.223`). Each integer ranges from 0 to 255. Write a regular expression to match precisely these addresses.

## 5. Context-free languages

Refer to the context-free grammar below. $S$ is the start symbol. Answer 4 of the following questions. Answer all for extra credit.

$$
\begin{array}{ll}
S \to T & T \to \mathbf{x} \\
S \to S + T & T \to \mathbf{y} \\
S \to S - T & T \to \mathbf{z} \\
S \to S * T & T \to (S) \\
S \to S/T &
\end{array}
$$

(1) Is the grammar above ambiguous? Why or why not?

(2) Explain why the grammar above is not LL(1).

(3) What is $First(T)$?

(4) What is $Follow(S)$?

(5) Perform a leftmost derivation of the following string: $\mathbf{x} * (\mathbf{y} + \mathbf{z})$

## 6. Extra credit

Complete any of the following for extra credit.

(1) List all possible *Sentences* that can be matched by the grammar below.

$$Sentence \rightarrow NounPhrase\ VerbPhrase$$
$$NounPhrase \rightarrow Article\ Noun$$
$$VerbPhrase \rightarrow Verb\ NounPhrase$$
$$Article \rightarrow \textbf{the}$$
$$Noun \rightarrow \textbf{boy}$$
$$Noun \rightarrow \textbf{ball}$$
$$Verb \rightarrow \textbf{kicked}$$

(2) Rewrite the grammar on the previous page to be LL(1).