Assignment #1
CS3300

1. [5] **Regular Expressions** Construct regular expression(s) to accept
   (a) (i) ANSI C language comments and (ii) C language Strings.
   (b) All string of a's and b's that contain the substring abb.
   (c) Java numeric constants.
   (d) Pascal keywords: `begin`, `end`, `function`. *Hint: Pascal is case-insensitive.*
   (e) All strings of digits (1, 2, 3) with at most one repeated digit.
2. [10] **Context Free Grammar (1)** We say a grammar to be ϵ-free if it has no non-terminal (except possibly the start symbol) that has ϵ as the RHS in any of its productions. Give a scheme to convert any arbitrary grammar to ϵ-free grammar. Try your scheme on the example $S' \rightarrow S; S \rightarrow aSbS|bSaS|\epsilon$.
3. [5] **Context Free Grammar** Write the CFG for
   (a) the BNF notation for REs.
   (b) the BNF notation for CFGs.
   (c) the set of all strings of 0s, 1s, and 2s that are palindromes.
   (d) accepting statements (including nested statements) that include '`if (expr) Stmt fi`', '`while (expr) do Stmt od`', Assignment statements of the form `x = expr`. Assume that the grammar for the non-terminal `expr` is already given.
4. [10] **Parsing** Give a leftmost derivation, a rightmost derivation, a parse tree for the following grammars and strings. Argue if the grammar is ambiguous or unambiguous, without resorting to building the parse table.
   (a) $S \rightarrow 1\ S\ 0|01$ with the string 110100.
   (b) $S \rightarrow S + S|SS|(S)|S * |a$ with string $(a + a) * a$
   (c) $S \rightarrow S\ \{\ S\ \}\ S|\epsilon$ with the string $\{\{\}\{\}\}$.
5. [2] **SLR Parsing** Construct the SLR sets of items and their GOTO functions for the last two (augmented versions of) grammar of Q4. Indicate any action conflicts in your sets of items and construct the SLR-parsing table, if one exists.
6. [5] **LR(0) parsing** Consider the following grammar:
   `S` $\rightarrow$ `+ S S | * S S | T`
   `T` $\rightarrow$ `id | num`
   (a) Is the grammar LR(0)? Build the LR(0) items and the parsing table to answer this question. (b) Consider the input ** 2 + a + b c* – if you answer the question (a) in affirmative, show the sequence of actions, considering an arbitrary action for each conflict, else give the conflict free sequence of actions and the parse tree
7. [5] Build the LR(1) table for the Crownoise grammar:
   ```
   Crownoise → Kaa Crownoise
             | Kaa
   ```
   Show the skeleton parser's action for parsing "Kaa Kaa Kaa".