CS251: Introduction to Language Processing

Mid-semester Examination (2023-24 M Semester)

Max. Points: 100 **Duration:** 1 hour 30 minutes

October 4, 2023

Instructions

- Question paper has 3 pages containing 4 questions.
- All the questions are compulsory.

Question-1 (Lexical Analysis)

Assume that an operating system uses the file names (or directory names), file paths (or directory paths) as described below.

- File name/Directory name can contain only digits or alphabets (upper/lower)
- The length of file/directory name has to be atleast one.
- The file extensions can be either **txt** or **dat**
- The file path should separate the directories with a slash(/). For example, /home/abc/1abc/ is a valid path
- The file path can contain the string .. to express the parent directories. For example, /home/../root/abc.txt is a valid file path

Write a simple lexical analzer that takes any string as an input and prints a token as VALID_PATH if the string is a valid file/directory path. Otherwise, it should print the token as INVALID_PATH.

Note that it is sufficient if you just write down the regular expressions and the corresponding tokens. You need not write the complete lex program.

[**18 Points**]

Question-2 (Context Free Grammar)

Consider the following grammar:

$$\begin{array}{lcl} S & : & E \\ E & : & E - E \\ & | & - - E \\ & | & int \end{array}$$

1. Show that the above grammar is ambiguous by drawing the parse trees corresponding to the string $int - - - int - int$
2. Define your precedence and associative rules that removes the ambiguity.
3. Draw the unique parse trees, by applying left most or right most derivation, for the string $int - - - int - int$ using the above rules.

[10 Points]

Question-3 (Bottom Up Parsing)

Consider the following grammar, with the production numbers as given.

$$\begin{array}{llcl} (1) & S' & : & S \\ (2) & S & : & N * N \\ (3) & S & | & N * P \\ (4) & N & : & -P \\ (5) & P & | & 1P \\ (6) & P & : & 1 \end{array}$$

1. Explain what is the language accepted by the above grammar.
2. Is the string $-11 * -11$ accepted by the grammar? If yes, construct the bottom up parse tree showing the right most derivation applied in the reverse manner.
3. Construct the DFA for LR(0)
4. Construct parser table for LR(0), you are required to use numbering scheme for production rules as given in the question.
5. Is the above grammar LR(0)? Justify your answer.
6. If the grammar is LR(0) show the bottom up parse tree for the string $-11 * -11$ by applying the LR(0) algorithm. If the grammar is not LR(0), explain where the does the algorithm produce a conflict while parsing $-11 * -11$ with the help of bottom up parse tree.

[44 Points]

Question-4 (Top Down Parsing)

Consider the following grammar:

$$
\begin{array}{lcl}
E & : & FH \\
H & : & *E \\
 & | & \epsilon \\
F & : & F! \\
 & | & G \\
G & : & n \\
 & | & (E)
\end{array}
$$

1. Compute the FIRST and FOLLOW sets for all the non terminals
2. Construct the LL(1) parser table
3. Show that the grammar is not LL(1) with a justification.
4. Rewrite the grammar such that grammar is LL(1).

[28 Points]