CS3300 Quiz 1

Dept of CSE, IIT Madras
Total marks = 24
Time = 50 min

06 Sep 2017

Read the instructions and questions carefully. You can make any reasonably assumptions that you think are necessary; but state them clearly. There are total three questions (8 marks each). You will need approximately 15 minutes for answering an 8 marks question (plan your time accordingly). For questions with sub-parts, the division for the sub-parts are given in square brackets.

You will get an answer sheet with 8 pages (if you get a answer booklet with fewer pages then ask for a replacement). Leave the first page empty and start from Page#2. Start each question on a new page. Think about the question before you start writing and write briefly. **For any question, the answer (including the answers for all the sub-parts) should NOT cross more than two pages.** If the answer for any question is spanning more than two pages, we will strictly ignore the spill-over text. If you scratch/cross some part of the answer, you can use space from the next page. You mostly would NOT need any additional sheets.

1. [8] **Lexical Analysis**: Consider the following four operators:
   `&&, &=, &` and `=`.

   Draw a single transition diagram (DFA) for recognizing all these tokens. [2]
   Use the diagram to build the lexical transition table. [2]
   Give a sketch (pseudo code will do) of the lexical recognizer that uses this transition table to recognize a series of these tokens delimited by white space. [4]

2. [8] **Parsing**: Briefly state the differences between LR(1) and LALR(1) parsing. [1]
   Build the LALR(1) item sets for the following grammar [3]. Assume that ID and COMMA are terminal symbols.

   ```
   S := S1 S2
   S1 := ID COMMA S1 | ϵ
   S2 := ID
   ```

   Prove that the grammar is not LALR [2], by building the parse table. Say, we give priority to shift over reduce operation (to resolve conflicts in the LALR(1) table). Give an example valid input string which will not be parsed by the LALR(1) parser [1]. Show the trace of parsing by showing the stack, input and action sequences [1].

3. [8] **Grammar, Precedence, Associativity and Parsing** Write a grammar to parse arithmetic expressions involving identifiers and numbers, over the following operators: +, * and ^. Assume that ^ has highest precedence, followed by * and +. Assume that + and * are left associative and ^ is right associative [5]. Prove that the precedence and associativity are preserved [0.5+0.5].

   Prove that in general, if a grammar is SLR(1), then the LALR(1) and SLR(1) state machines will have the same number of states [1.5]. What is the difference between LR(1) and SLR(1) parsing algorithm [0.5].