

CS3300

Quiz 1

Dept of CSE, IIT Madras

Total marks = 24

Time = 45 min

03 Sep 2019

Read the instructions and questions carefully. You can make any reasonable 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 lexical tokens:

(i) Division operator ($/$), (ii) Division operator with equality ($/=$), (iii) multi-line comments begin $/*$, (iv) single line comment (starting $//$ and ending with newline).

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(0) and SLR(1) parsing. [1]

Build the LR(0) item sets for the following grammar [3]. Assume that LB, RB and SC are terminal symbols.

B \rightarrow S

S \rightarrow LB S RB

S \rightarrow S S

S \rightarrow SC

Show that the grammar is LR(0) [2], by building the parse table. If there is a conflict, fix the grammar to make it LR(0) [2]. Or else For the input string LB SC SC RB show the trace of parsing by showing the stack, input and action sequences [2]; start from the initial input " \bullet LB SC SC RB" and show the states till you reach either an Accept or Error state.

3. [8] **True/False questions**

(a) Every lexical token can also be specified using context free grammar.

(b) LL parsing not used in practice.

(c) LR parsing stands for "Left-to right parsing Right most derivation".

(d) If a grammar is SLR(1) then LR(1) and SLR(1) tables will have the same number of states.

(e) "Semicolon (;) missing" is an error reported by the lexical analyzer.

(f) "Uninitialized variable" is an error that cannot be reported by the parser.

(g) After left recursion removal and left factoring - a grammar becomes an LL grammar.

(h) If a grammar is context free it is also LR(1).

(i) Every compiler has seven phases.

(j) The number of states in an Characteristic Finite State Machine is dependent on the size of the input string.

(k) The number of entries in the state-stack used by the LR parsing routine is independent of the size of the input string.

(l) SLR(1) has same power as LR(1), but reports fewer errors.

(m) All the compilers of a particular language (say C), for a particular hardware (say x86), use the same intermediate languages.

(n) If there are no errors generated in the front-end of a compiler, the back-end will not throw any further errors.

(o) The complexity of the LR parsing algorithm is less than that of CYK algorithm.

(p) Predictive parsing is less powerful than LL parsing with backtracking.