

CS3300 Quiz 1

Dept of CSE, IIT Madras

Total marks = 24+2

07 Sep 2016

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) + one bonus question (2 marks). 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. Attempt the bonus question (marks disproportional to the time needed) at the end.

You will get an answer sheet with 8 pages (if you get a answer sheet with fewer pages then ask for a replacement sheet). 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:** State briefly the difference between LR(0) and SLR(1) parsing. [2]
Build the LR(0) item sets for the following grammar [3].

$S := S S \mid (S) \mid id$

Q: Build parsing table to justify if the grammar LR(0)? SLR(1)? [3]

3. [8] **Type checking** Our goal is to do type checking for simple C code. Describe the actions you will perform when you encounter the following statements (to infer types and perform type checking). [2+2+2+1+1]

1. Scalar (non structs/unions/enums) variable declarations. 2. Assignment statements of the form $a=b + c$ or $a = b \mid \mid c$ 3. Function declarations of the form $t1 \text{ foo } (t2 \ x) \{ S \}$ 4. Function calls of the form $a=\text{foo}(b)$ 5. Return statement of the form **return a**

4. [2] **Bonus: Remove conflicts:** The following grammar parses the list of parameters in a function definition:

```
params  : param
         | params "," param
param   : type_name ids
type_name: id
ids    : id | ids "," id
```

This grammar leads to a shift-reduce conflict on ",", during LALR(1) parsing. Rewrite the grammar (must accept the same language) to eliminate the conflict. [1] Show that the rewritten grammar is LALR(1). [1]