CS3300 Quiz 1: Sep 05, 2023. (QP Code: A)

Maximum marks = 30, Time: 45 minutes, Closed Book, Closed Neighbor

Name: _____ Roll: _____

Read the instructions and questions carefully. You can use the given booklet for rough work and stating any reasonable assumptions you make. But write the answers in the QP itself – marks will be given based on the answers in the QP.

- MCQ and True/False questions: Each incorrect answer will lead to a deduction of 0.5 marks.
- MSQ questions:
  - If you choose any wrong option - you will get a 0 for that question.
  - If you choose only a subset of the correct options: you will get proportional marks.

Section 1. Lexical Analysis, 2 marks each

1. Which of the following is/are true about LL(1) grammars?
   (a) Left recursive grammars are not LL(1).
   (b) Some LL(1) grammars may be ambiguous.
   (c) A language that has no LL(1) grammar is ambiguous.
   (d) Left factoring and left recursive removal can be used to convert any grammar to LL(1).
2. The key decision(s) in bottom-up parsing is/are:
   (a) When to reduce?
   (b) What production rules to apply for reduction?
   (c) Which non-terminal to use?
   (d) Which terminals to process?
3. Which of the following is/are NOT a valid token-type(s):
   (a) Type of a variable
   (b) scope of a variable
   (c) operator
   (d) loop
4. Which of the following is/are true with respect to regular expressions:
   (a) ϵ is a regular expression.
   (b) Each regular expression derives unique set of strings.
   (c) Given a finite alphabet L the number of regular expressions over L is finite.
   (d) Every regular expression must derived at least two strings.
5. Which of the following is/are possible attribute-values(s) for lexemes:
   (a) line number
   (b) type of a variable
   (c) operator associativity
   (d) loop
6. Given a production of the form $A \rightarrow \beta$, if β has k symbols (terminals and non-terminals), then how many LR(0) items can the production generate?
   (a) $k-1$
   (b) k
   (c) $k+1$
   (d) Depends on the input

Section 2. Fill in the blank, 3 marks each

1. Consider the grammar and state I_0.

1	S	$\rightarrow$	$E\$$
2	E	$\rightarrow$	$E+T$
3	E	$\rightarrow$	ET
4		$\mid$	T
6	T	$\rightarrow$	Id
7		$\mid$	(E)

I_0 : $S \rightarrow \bullet E\$$
$E \rightarrow \bullet E+T$
$E \rightarrow \bullet ET$
$E \rightarrow \bullet T$
$T \rightarrow \bullet Id$
$T \rightarrow \bullet(E)$

The number of elements in the set returned by $GOTO(I_0, E)$ = _____

2. Consider the table driven parsing algorithm given below. One of the lines is erroneous.

Input: A string w and a parsing table M for a grammar G
Output: If w is in $L(G)$, a leftmost derivation of w; otherwise, indicate an error

```
1  push $ onto the stack; push S onto the stack;
2  a points to the input tape;
3  X = stack.top();
4  while X ≠ $ do
5      if X is a then {stack.pop(); inp++} ;
6      else if X is a terminal then error();
7      else if M[X, a] is an error entry then
           error();
8      else if M[X, a] = X → Y1Y2 ··· Yk then
9          output the production X → Y1Y2 ··· Yk;
10         stack.pop();
11         push Y1, Y2, ··· Yk in that order;
12     X = stack.top();
```

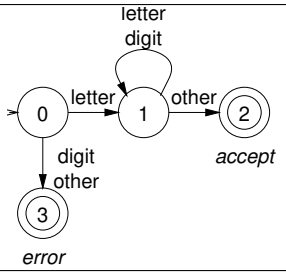
The line number that has the error is _____.

3. Consider the code shown (in the right) for recognizing identifiers.

```
1. state=0; // initial state
2. done=false;
3. tokenVal=""// empty
4. while (not done) {
5.  ch=nextChar();
6.  class=charClass[ch];
7. state=
     nextState[class,state];
8. switch(state) {
9.  case 1:
10.   tokenVal=tokenVal+ch;
11.   break;
12.   case 2: // accept state
13.     tokenType=id;
14.     done = true;
15.     break;
16.   case 3: // error
17.     tokenType=error;
18.     done=true;
19.     break;
20.  } // end switch
21. } // end while
22. return tokenType;
```

The minimum number of lines required to be changed (added/removed) in this code to make it correct are _____. Note: if the code is correct, enter the value 0.

4. Consider the DFA shown to the right.



The number of non-error entries in the `nextState` table are = _____.

Section 3. True or False Answers, 1 mark each

_____ Given an input consisting of m terminals, the LR parsing technique for a grammar with n non-terminals, shifts $m+n$ number of times.

_____ The closure of an item can be a singleton set.

_____ In an LL(1) grammar with no epsilon productions, the FIRST and FOLLOW sets of a non-terminal may have no common elements.

_____ Lexical analysis can be used infer the type of each variable.

_____ A compiler can use error recovery techniques to fix the errors in a program and generate the correct machine-code.

_____ Regular expressions can be used to ensure that all variables are of lower case only.