VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
VELLORE ▪ CHENNAI
www.vit.ac.in

School of Computer Science and Engineering
G1 +TG1-Slot CAT-II (Oct-2017)
B.Tech (CSE, BCI, BCB)
Subject: TOC and Compiler Design – CSE2002

Time: 1 Hr 30 Mins Max.Marks:50

Answer ALL questions
(5 X 10 = 50 marks)

1. (a) An *LL(0)* parser is a similar to an *LL(1)* parser, except that there are zero tokens of look ahead to determine which production to use. Describe the set of grammars that can be parsed with an *LL(0)* parser. [5]

(b) Is it possible to find, for every context-free language, a grammar such that all its productions are either of the form $A \rightarrow BCD$ (the body of the production consisting of three variables) or $A \rightarrow a$ (the body of the production consisting of a single terminal)? Give either a proof or a counter example? [5]

2. (a) Consider the following (already augmented) grammars for the language a^* [5]

$$S \rightarrow A$$
$$A \rightarrow Aa \mid \varepsilon$$

and

$$S \rightarrow A$$
$$A \rightarrow aA \mid \varepsilon$$

Both of these grammars are *SLR(1)*. However, the *SLR(1)* parser for one of these grammars will use *O(n)* space in its parsing stack when run on the string an, while the other parser will only use *O(1)* stack space.
Identify which grammar's parser uses *O(n)* stack space and which grammar's parser uses *O(1)* stack space. Justify your answer by making specific references to how an *SLR(1)* parser for each grammar parses strings of the form a^n. In other words, even if you can figure out why one parser uses *O(n)* stack space, you should still explain why the other parser uses only *O(1)* stack space.

(b) Consider the following expression grammar [5]

*exprs := exprs + expr | exprs * expr | expr*
expr := x

(a) Is the grammar ambiguous or unambiguous? If ambiguous show two different parse trees for the same string. If it is unambiguous give an informal argument?
(b) Does this grammar properly capture the normal precedence of arithmetic operators * and +. If so, give a brief argument as to why it does, if not, give an example that shows where it fails to capture the correct precedence relationship.

3. Let *G* be an LL(1) grammar with Σ a set of terminal symbols and Π as the set of non-terminal symbols (Note: For each of the following the answer should be given in big-O

notation. For example $O(x)$, if the quantity is linear in x)

(a) What is the maximum size of LL(1) parsing table for G? Why?
(b) What is the size of the largest parse tree produced the grammar in CNF for a string of length n? Why? (Assume that the grammar G is not having ϵ-productions)
(c) How long does it take to parse a string of length n with respect to G using *LL(1)* parsing algorithm? Why?
(d) What is the maximum size of SLR(1) parsing table for G? Why?

4. (a) Suppose we want to add the following conditional statement to MiniJava: [5]

```
ifequal (exp1, exp2)
        statement1
smaller
        statement2
larger
        statement3
```

The meaning of this is that *statement1* is executed if the integer expressions *exp1 and exp2 are equal*; *statement2* is executed *if exp1 < exp2*, and *statement3* is executed *if exp1 > exp2.*
Give context-free grammar production(s) for the ***ifequal*** statement that allows either or both of the ***smaller*** and ***larger*** parts of the statement to be omitted. If both the *smaller* and ***larger*** parts of the statement appear, they should appear in that order.

(b) Consider the following grammar for a simplified Clike function prototype. [5]
The terminals are {***T_Ident***, ***T_Double***, ***T_Char***, (,), ; ,}. The tokens will be recognized by the scanner and passed to the parser.

Proto → ***Type T_Ident (ParamList) ;***
Type → ***T_Int*** | ***T_Double*** | ***T_Char***
ParamList → ***ParamList , Param*** | ***Param***
Param → ***Type T_Ident***

00101

Why does the LR(0) parser not attempt a reduction to Param after pushing the first sequence of type and identifier onto the parse stack? Doesn't the sequence on top of the stack match the right side. What other requirements must be met before a reduction is performed?

5. (a) Give a formal description and the corresponding state diagram of a PDA that recognizes the language $L = \{w \mid 2\#a(w) \neq 3\#b(w),\ w \in \{a, b\}^*\}$, where $\#a(w)$ and $\#b(w)$ denotes the number of a's and b's occurring in the string w. [5]
(b) Let $L_1 = \{0^n1^m \mid 0 < n \leq m < 2n\}$. Let G_1 be the grammar with starting symbol S and the following rules: [5]

Rule 1: $S \rightarrow 0S11$
Rule 2: $S \rightarrow T$
Rule 3: $T \rightarrow 0T1$
Rule 4: $T \rightarrow 01$

For each n and m satisfying $0 < n \leq m < 2n$, describe a leftmost derivation of 0^n1^m using the grammar G_1. (That is, say how many times to apply each rule and in what order).