CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY

Seventh Semester of B. Tech. Examination (C.E.)
Nov 2013
CE401 Compiler Construction (CC)

Date: 14.11.2013, Thursday **Time: 10:00 a.m. To 01:00 p.m.** **Maximum Marks: 70**

Instructions:

1. The question paper comprises of two sections.
2. Section I and II must be attempted in separate answer sheets.
3. Make suitable assumptions and draw neat figures wherever required.
4. Rough work is to be done in the last page of main supplementary, please don't write anything on the question paper.
5. Indicate clearly, the option(s) you attempt along with its respective question no.
6. Figures to the right indicate full marks.

SECTION-I

Q-1 **Answer the following questions.**

1. Why analysis phase is divided into lexical analysis and parsing? Why input buffer in Lexical Analyzer is divided into two halves? 3
2. Differentiate following terms. 4
   a) Parse Tree and Annotated Parse Tree
   b) Parse Tree and Syntax Tree
3. Construct DFA and minimize it 4
   (0*1*)* 0

Q-2

[A] Consider the grammar: S→SS+| SS*|a 3

(1) Show that the string aa+a* can be generated by the grammar.
(2) Construct the parse tree for the grammar.
(3) Is the grammar ambiguous?

[B] Write an algorithm for scanner, which can identify following 'C' language tokens. 5

(i) Keywords (if, else, while)
(ii) String constants e.g. "this is constant"
(iii) single line comments-/* this is comment*/
(iv) punctuation symbols – ",", ',', %,;,{,}

OR

[B] Give the translation of three address code for the following stmt: 5

```
while a < b do
 if c < d then
   x:= y + z
 else
   x:= y-z.
```

[C] What are the design considerations for constructing a compiler? Which are the major issues in designing/constructing the compiler? 4

Q-3

[A] Identify the phase of a compiler detecting following types of errors: 4

a. Logical errors
b. Code is unreachable
c. Function prototype not specified
d. Type Mismatch

[B] What are the roles of Lexical Analyzer? Give the need for Lexical Analysis in compilation process. Also explain the roles performed by Syntax Analyzer phase. Give the need for Syntax Analysis in compilation process. 4

OR

[B] Write **syntax directed definition** for translating following grammar to *postfix* notation. Also draw the **annotated parse tree** for input 9-5+2. 4

expr -> expr + term
expr -> expr – term
term -> 0 | 1 | .....| 9

[C] Construct the **Non-Recursive Predictive Parsing table** for the following grammar. 4

S -> aBDh
B -> cC
C -> bC | null
D -> EF
E -> g | null
F -> f | null

OR

[C] Discuss on **the factors** affecting **Target Code Generation** (phase) of a compiler. 4

SECTION-II

Q-4

1. What is *left factoring* problems? Explain in detail. Also explain how to solve these problems. **Left factor** the following grammar: 4
   S→ iEtS | iEtSeS | a
   E→ b
2. Write a grammar to generate the following language 4
   L= { SREV(S)S| S∈{a,b}*}
3. What are the goals of *error handler*? Explain: **Panic Mode** Error Recovery Strategy. 3

Q-5

[A] Name at least six items of run-time information stored in an *activation record/frame*. For each item, identify whether its value is set before the procedure is called, during procedure execution or right before procedure return. 4

[B] What is the need for *symbol table*? How does the *symbol table* represented in memory? Explain *Hash symbol* table. 4

OR

[B] Eliminate the ***left recursion*** from the following grammar: 4
S→a | ^ | (T)
T→ T, S | S
Whether the transformed grammar is LL (1)?

[C] Check whether the following grammar is **LL (1) or not?** 4
S→ aABC
A→ a | bbD
B→ a | ε
C→ b | ε
D→ c | ε

OR

[C] What is *code optimization*? What are the sources of *optimization*? List the criteria for selecting *code optimization* technique. 4

Q-6

[A] State the following Grammar is **LR (1)** but **not LALR (1).** 4
S→ Aa | bAc | Bc | bBa
A→ d
B→ d

[B] Compute the **FIRST** and **FOLLOW** sets for the following grammar: 4

S→ aAbc | BCf
B→ Cd | ε
A→ C | ε
C→df | ε

OR

[B] Construct **canonical LR (0)** parsing states for following grammar: 4

S→ L=R
S→R
L→*R
L→id
R→L

[C] Consider the following **augmented grammar**: 4

S'→ S
S →CC
C →cC|d
Prepare the ***goto graph*** for it.

OR

[C] Construct SLR parsing table and find set of **LR (1)** items. 4

S→ aABe
A→Abc
A→b
B→d