

B.Tech 6th Semester Exam., 2018

COMPILER DESIGN

Time : 3 hours

Full Marks : 70

Instructions :

- (i) The marks are indicated in the right-hand margin.
- (ii) There are **NINE** questions in this paper.
- (iii) Attempt **FIVE** questions in all.
- (iv) Question No. 1 is compulsory.

1. Choose the correct answer for any seven of the following : 2×7=14

(a) What is the number of tokens in the following C statement?

```
printf("I=%d, and I=%x", I, I) and I;
```

- (i) 3
- (ii) 26
- (iii) 10
- (iv) 21

(b) Consider the following two statements :
 P : Every regular grammar is LL(1).
 Q : Every regular set has LR(1) grammar.
 Which of the following is true?

- (i) Both P and Q are true
- (ii) P is true and Q is false
- (iii) P is false and Q is true
- (iv) Both P and Q are false

(c) What is the similarity between LR, LALR and SLR?

- (i) Use of same algorithm, but different parsing table
- (ii) Same parsing table, but different algorithm
- (iii) Their parsing tables and algorithms are similar but use top-down approach
- (iv) Both parsing tables and algorithms are different

(d) Consider the following grammar :

$$S \rightarrow aSbS \mid bSaS \mid \epsilon$$

How many different parse trees are possible for string ababab?

- (i) 2
- (ii) 3
- (iii) 4
- (iv) 5

(e) A shift reduce parser carries out the actions specified within braces immediately after reducing with the corresponding rule of grammar :

$S \rightarrow xxW$ (PRINT "1")

$S \rightarrow y$ {print "2"}

$S \rightarrow Sz$ {print "3"}

What is the translation of xxxxyz using the syntax directed translation scheme described by the above rules?

(i) 23131

(ii) 11233

(iii) 11231

(iv) 33211

(f) The most powerful parser is

(i) SLR

(ii) LALR

(iii) Canonical LR

(iv) Operator-precedence

(g) In some programming languages, an identifier is permitted to be a letter followed by any number of letters or digits. If L and D denote the sets of letters and digits respectively, which of the following expressions defines an identifier?

(i) $(L \cup D)^*$

(ii) $L(L \cup D)^*$

(iii) $(L.D)^*$

(iv) $L.(L.D)^*$

(h) In operator precedence parsing, precedence relations are defined

(i) for all pairs of non-terminals

(ii) for all pairs of terminals

(iii) to delimit the handle

(iv) None of the above

(i) A bottom-up parser generates

(i) right-most derivation

(ii) right-most derivation in reverse

(iii) left-most derivation

(iv) left-most derivation in reverse

(i) A grammar that produces more than one parse tree for some sentence is called

(i) ambiguous

(ii) unambiguous

(iii) regular

(iv) None of the above

2. (a) What do you understand by left recursion and left factoring? How are these eliminated? Explain with the help of suitable example. 10

(b) What do you understand by bootstrapping? Explain with the help of suitable example. 4

3. (a) How can addressing modes be used for reducing the memory access time? 4

(b) What are activation records? Explain its purpose in compilers. 4

(c) What are the optimization techniques applied on procedure calls? Explain with an example. 6

4. (a) Given the Syntax-Directed Definition below with the synthesized attribute val. Draw the annotated parse tree for the expression $(3 + 4) * (5 + 6)$: 7

$L \rightarrow E$	$L.val = E.val$
$E \rightarrow T$	$E.val = T.val$
$E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
$T \rightarrow F$	$T.val = F.val$
$T \rightarrow T_1 * F$	$T.val = T_1.val * F.val$
$F \rightarrow (E)$	$F.val = E.val$
$F \rightarrow \text{digit}$	$F.val = \text{digit.lexval}$

(b) Construct a Syntax-Directed Translation scheme that translates arithmetic expressions from infix into postfix notation. Your solution should include the context-free grammar, the semantic attributes for each of the grammar symbols, and semantic rules. Show the application of your scheme to the input and "3*4+5*2". 7

5. (a) What is the advantage of using machine independent intermediate representation during translation process? 2

(b) Explain, with example, what three address codes are. Give some common three address statements. 4

- (c) Find the three address codes of the following program. There are four bytes per word : 8

```
Sum = 0
For (i=1; i<=20; i++)
Sum = Sum+a[i]+b[i];
```

6. (a) Find the CNF (Chomsky Normal Form) for the following CFG : 5

```
S → ABA
A → aA | ε
B → bB | ε
```

- (b) Eliminate null productions from the following CFG : 4

```
S → Xa
X → aX | bX | ε
```

- (c) Convert the following grammar to GNF : 5

```
A1 → A2A3
A2 → A3A1 | b
A3 → A1A2 | a
```

7. (a) What are the advantages of LALR parsing over SLR and CLR methods? 5

- (b) Write the algorithm to compute FIRST and FOLLOW for a given grammar.. 5

- (c) What is shift-reduce conflict? 4

8. What are intermediate codes in compilers? Why is it needed in compiler design? Discuss different types of intermediate codes generated by intermediate code generation phase. 14

9. (a) What is parsing? What is the difference between Top-down parsing, Bottom-up parsing and Predictive parsing? 7

- (b) Explain the working of each phase of compiler in detail with an example. 7
