Code : 051616

B.Tech 6th Semester Exam., 2016

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 from the following (any *seven*) : 2×7=14

(a) Right side of three-address code has how many numbers of operator at most?

(i) 1

(ii) 2

(iii) 4

(iv) 3

(b) Token is

(i) a logically cohesive sequence of characters

(ii) always same as lexeme

(iii) non-terminal

(iv) just string

(c) The regular expression for the language of all strings that have zero or more a's followed by zero or more b's is

(i) $(a+b)$

(ii) ab

(iii) $a^* b^*$

(iv) $(ab)^+$

(d) Cross-compiler is a compiler

(i) which is written in a language that is different from the source language

(ii) that generates object code for its host machine

(iii) which is written in a language that is same as the source language

(iv) that runs on the machine but produces object code for another machine

(e) A dangling reference is a

(i) pointer pointing to storage which freed

(ii) pointer pointing to nothing

(iii) pointer pointing to storage which is still in use

(iv) pointer pointing to uninitialized storage

(f) If a grammar is LALR(1), then it is necessarily

(i) SLR(1)

(ii) LR(1)

(iii) LL(1)

(iv) None of the above

(g) After removing left recursion from $A \rightarrow A\alpha / \beta$, the resulting grammar will be

(i) $A \rightarrow \beta A'$

(ii) $A \rightarrow \alpha A'$

(iii) $A \rightarrow \alpha\beta A'$

(iv) $A \rightarrow \beta\alpha A'$

(h) The traversal method translation schema adapted to execute the action is

(i) depth first search

(ii) breath first search

(iii) depth breath first search

(iv) long first search

e above

(i) Consider the following grammar :

$S \rightarrow cAd$

$A \rightarrow ab \mid ac \mid a$

For input string cad, how many times the recursive descent parser will backtrack?

(i) 2

(ii) 3

(iii) 4

(iv) 5

(j) Activation of procedures can be implemented by run-time storage. The part of run-time storage that is responsible for this is

(i) data object

(ii) target code

(iii) stack pointer

(iv) control stack

2. (a) With suitable example, explain the different phases of a compiler.

(b) Define left recursive grammar. Propose an algorithm to remove left recursion and apply that algorithm on the following grammar :

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow id$

(Continued)

(c) Explain the parameter-passing mechanism in the following cases :

(i) Call by value

(ii) Call by reference

(iii) Copy restore

(iv) Call by name 5+5+4

3. A grammar is given below :

$S \rightarrow A$

$A \rightarrow aB \mid Ad$

$B \rightarrow bBC \mid f$

$C \rightarrow g$

(a) Find the FIRST and FOLLOW set.

(b) Construct a predictive parsing table.

(c) Trace whether the string "abbfgg" is accepted or not. 6+6+2

4. Consider the following grammar :

$S \rightarrow 0S0 \mid 1S1 \mid 10$

(a) Find LR(0) collection of items for the above grammar.

(b) Construct SLR parsing table.

(c) Trace whether the string "01100" is accepted or not. 6+6+2

5. Consider the following grammar :

$$S \rightarrow aSbS \mid bSaS \mid \varepsilon$$

(a) Find LR(1) collection of items for the above grammar.

(b) Construct LALR parsing table.

(c) Trace whether the string "01100" is accepted or not. 6+6+2

6. 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

7. (a) Define a basic block.

(b) Write an algorithm to partition, a given sequence of three-address codes into basic blocks.

(c) Apply the above algorithm to find out the basic blocks in the following code fragment :

(i) prod=0

(ii) i=1

(iii) t1=4*i

(iv) t2=a[t1]

(v) t3=4*i

(vi) t4=b[t3]

(vii) t5=t2*t4

(viii) t6=prod+t5

(ix) prod=t6

(x) t7=i+1

(xi) i=t7

(xii) If i<=20 goto (iii) 2+5+7

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

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

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

```
sum=0
for (i=1; i<=20; i++)
sum=sum+a[i]+b[i];
```

2+4+8

9. *(a)* What is activation record? Explain its purpose in compilers.

(b) Write down the algorithm to find the FIRST and FOLLOW set of a grammar. Use that find FIRST and FOLLOW of the following grammar :

$S \rightarrow ACB \mid CbB \mid Ba$

$A \rightarrow da \mid BC$

$B \rightarrow g \mid \varepsilon$

$C \rightarrow h \mid \varepsilon$

(c) What do you mean by left factoring a grammar? Explain with the help of an example.

(d) Explain the difference between syntax-directed definition and translation schemes. 3+6+2+3

★ ★ ★