M–11112

Register No. | | | | | | | | |

BE Degree Examination April 2018

Sixth Semester

Computer Science and Engineering

14CST61 – COMPILER DESIGN

(Regulations 2014)

Common to BTech Information Technology

Time: Three hours | Maximum: 100 marks

Answer all Questions

Part – A (10 × 2 = 20 marks)

1. Given $\Sigma = \{a, b\}$ write a regular expression to define the language $L = \{w / w$ begins with 'a' and ends with 'b'$\}$. [CO1,K3]
2. Define Deterministic Finite Automata. [CO1,K2]
3. State the differences between top down and bottom up parsing techniques. [CO2,K4]
4. Write the language define by the Grammar $S \rightarrow aAB; A \rightarrow bBb; B \rightarrow A / \varepsilon$. [CO3,K3]
5. Why do you need intermediate code generation in compiler? [CO4,K2]
6. Give the grammar $E \rightarrow E + T / T$, $T \rightarrow T * F / F$, $F \rightarrow id$, draw the syntax tree for the sentence 'id+id*id'. [CO3,K3]
7. What is meant by the term 'Basic block'? [CO4,K1]
8. Define Peephole optimization. [CO5,K1]
9. What are the items that appear in the activation record? [CO5,K1]
10. List the types of storage allocation techniques and their use. [CO5,K2]

Part – B (5 × 13 = 65 marks)

11. a. i) Specify an input buffering technique for lexical analysis. Discuss about its construction an working principle. (7) [CO1,K1]

    ii) State a simple approach for the design of a lexical analyzer. (6) [CO1,K1]

    (OR)

    b. Given $\Sigma = \{a, b\}$ and the regular expression (a/b)*abb, construct an equivalent $\in$-NFA and convert it into a Deterministic Finite Automate using subset construction method. (13) [CO2,K3]

12. a. i) Given $\Sigma = \{0,1,2\}$, write a grammar to generate the set $\{w2w^R / w$ is in $(0/1)^*$ and w^R is the reverse of $w\}$. (3) [CO2,K3]

    ii) Define LR parser. Mention the different type of LR parsers and discuss in detail. (10) [CO1,K2]

    (OR)

    b. Construct the predictive parsing table for the grammar $E \rightarrow TE'; E' \rightarrow TE' / \varepsilon; T \rightarrow (E) / id$ and parse the string (id+id). (13) [CO2,K3]

13. a. Explain how three address codes are more suitable for the generation of object code. Discuss the different ways of implementing three address codes. (13) [CO3,K2]

(OR)

b. Given the grammar $A \rightarrow id := E; E \rightarrow E + E / E * E / - E / (E) / id$, write the abstract and concrete translation schemes for generation of three address codes. (13) [CO3,K2]

14. a. Consider the following piece of code. Write the equivalent three address codes for a machine with 4 bytes/word, draw the flow graph and do the necessary optimizations. (13) [CO4,K3]

```
Begin
P := 0;
For I = 1 to 20 do
P := P + A[I] * B[I];
end
```

(OR)

b. Describe the significance of loop optimization and state the different optimizations that can be made with respect to loops in the program. (13) [CO4,K1]

15. a. Why are heap management and garbage collection used in compiler? Explain in detail. (13) [CO5,K2]

(OR)

b. Explain in detail how runtime storage in managed for languages with stack allocation of storage. (13) [CO5,K2]

Part – C ($1 \times 15 = 15$ marks)

16. a. Mention the different phases of a compiler. With suitable program codes as example, illustrate the translation of code after each phase. (15) [CO1,K1]

(OR)

b. Construct the SLR parser for the grammar (15) [CO2,K3]

$E \rightarrow E + T / T$
$T \rightarrow T * F / F$.
$F \rightarrow (E) / id$

Show the parsing action for the sentence '$(id + id)$'.

Bloom's Taxonomy Level	Remembering (K1)	Understanding (K2)	Applying (K3)	Analysing (K4)	Evaluating (K5)	Creating (K6)
Percentage	26	37.8	35	2	-	-