M.E. ELECTRONICS AND TELECOMMUNICATION ENGINEERING FIRST YEAR SECOND SEMESTER EXAMINATION 2017

COMPILER CONSTRUCTION

Time: 3 hours Full Marks : 100

Answer any **five** questions
All questions carry equal marks

1. a) Describe how an NFA may be derived from a given regular expression. Illustrate your answer with $\mathbf{(a|b)^{+}\ abb}$ as an example regular expression.
   b) Derive a DFA for the NFA obtained in part(a). Also minimize the state of the DFA.

10+10

2. Consider the grammar

   S →aAcBe
   A →Ab
   A →b
   B →d

   a) Eliminate immediate left recursion from the grammar
   b) Construct the Predictive parsing table for the grammar obtained in part a
   c) Using you're the table obtained in part(b) parse the string "**abbcde**"

8+8+4

3. a) Define

   i. An Operator grammar
   ii. An operator precedence grammar

   Show that the following grammar is an operator precedence grammar.

   S → cAd
   A → ab
   A → a

   Obtain the precedence relations for the grammar. Using these relations parse the input string **" cabd $"**

   b) Design the necessary procedures for the Recursive – Descent Parsing for the above grammar and parse the input string **"cabd$"**

10+10

4. a) What are the difficulties encountered in top down parsing? What is left factoring? Why this is necessary? Explain your answer by a suitable example in connection with top down parsing
   b) Describe an algorithm that will eliminate left recursion from a given grammar. Use the same to eliminate the left recursion from the following grammar

   S → A a | b
   A → A c | S d | e

10+10

5. Consider the grammar

   S → cAd
   A → ab
   A → a

   a) Apply the necessary left factoring and obtain a predictive parsing table.
   b) Obtain the SLR parsing Table.
   c) Using both the above parser parse the string "**cad**".

8+8+4

[Turn over

6. Consider the following grammar

   S→ AS
   S → b
   A → SA
   A → a

   a) List all the LR(0) items for the above grammar.

   b) Construct an NFA whose states are the LR(0) items from (a). Show that the canonical LR(0) items for the grammar is the same as the states of the equivalent DFA.

10+10

7. Following grammar:

   R → R <> R
   R → R || R
   R → (R)
   R → id

   is used to represent a series parallel connection of a resistive network. **R1<>R2** , for example represents a series connection shown in Fig 1(a) where the string **R1 || R2** represents a parallel connection as shown in Fig 1(b).

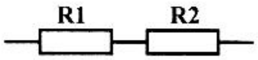

Fig. 1(a)

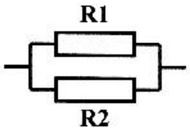

Fig. 1(b)

   a) Write down all the sets of LR(0) items for the above grammar and derive the SLR parsing table for the same.

   b) Show that shift reduce conflict is present in the table. Resolve the conflict by using suitable reasoning. Assume that a series connection has precedence over a parallel connection.

   c) Use the modified table to parse a string **"R1 <> R2 || R3"**.

8+8+4

8. Consider the following grammar

   E → E + T|T
   T → T * F | F
   F → (E) | id

   a) Eliminate immediate left recursion from the above grammar.

   b) Write a mutually recursive procedure for the above grammar to implement a non back tracking recursive-descent parser. Use this parser to parse the input string

   id + id * id $

10+10