B5.2-R4: AUTOMATA THEORY AND COMPILER DESIGN

NOTE:

1. Answer question 1 and any FOUR from questions 2 to 7.
2. Parts of the same question should be answered together and in the same sequence.

Time: 3 Hours
Total Marks: 100

1. 
   a) What is Syntax directed translation? In this context define synthesized attributes with example.
   b) Explain different error recovery strategies used in syntax analysis.
   c) Consider the following grammar
      \[ \text{stmt} \rightarrow \text{if expr then stmt} \]
      \[ \quad | \text{if expr then stmt else stmt} \]
      \[ \quad | \text{other} \]
      Write down the language generated by the above grammar. Convert the above grammar into an unambiguous form.
   d) Operator precedence parsing is suitable for operator grammar - Justify.
   e) What is an activation record used in runtime storage management? What does a typical activation record consist of? What is the role of processor register used in this management?
   f) What do you mean by l-values and r-values of an identifier? How are they used in the translation of expressions?
   g) Compare and contrast triples with indirect triples?

(7x4)

2. 
   a) Prove the following theorem by induction: \(1+2+3+\ldots+n = \frac{n(n+1)}{2}\).
   b) Define Kleene closure. Explain with an example.
   c) Consider the following grammar \(G=(T=\{m,n,q,\$\}, NT=\{S, S0, A, B, C\}, S0, P)\).
      \[ P: \]
      \[ (1) \quad S' \rightarrow S\$ \]
      \[ (2) \quad S \rightarrow AB \]
      \[ (3) \quad A \rightarrow mA \]
      \[ (4) \quad A \rightarrow nA \]
      \[ (5) \quad A \rightarrow \epsilon \]
      \[ (6) \quad B \rightarrow qCn \]
      \[ (7) \quad C \rightarrow Cm \]
      \[ (8) \quad C \rightarrow m \]
      Compute the FIRST and FOLLOW sets for the non-terminals in \(G\).

(5+6+7)

3. 
   a) Find a grammar generating \(L = \{a^n b^n c^i | n \geq 1, i \geq 0\}\). Prove that it does so.
   b) Design a Turing machine \(M\) to recognize the language \(\{1^n 2^n 3^n | n \geq 1\}\)
4.
   a) Consider the following grammar S→AB|CA, B→AB|BC, A→ a, C→ a B |c. Convert it into reduced form.
   b) Construct a PDA accepting the set of all strings over {0,1} with equal no. of 0’s & 1’s.
   c) Define Type 2 and Type 3 grammar. Find the highest type number which can be applied for the following grammars.
      i. S→Aa, A→c|Ba B→abc
      ii. S→ASB|d A→aA

5.
   a) Remove the left recursion from the following grammar.

\[
\begin{align*}
E &\rightarrow E+T \mid T \\
T &\rightarrow T*F \mid F \\
F &\rightarrow (E) \mid id
\end{align*}
\]

   b) Describe the use of stack & heap in runtime allocation.
   c) To improve the target code we generally use copy propagation, code motion and reduction in strength. Explain and give example in each case.

6.
   a) Draw the steps of parsing id+ id2*id 3 with the help of shift reduces parser. Define viable prefix in this parser.
   b) How can you define a Direct Acyclic Graph (DAG). Write down its applications.
   c) Draw the DAG for the following code:

\[
\begin{align*}
1. &\quad t_1 := 4*i \\
2. &\quad t_2 := a[t_1] \\
3. &\quad t_3 := 4*i \\
4. &\quad t_4 := b[t_3] \\
5. &\quad t_5 := t_2*t_4 \\
6. &\quad t_6 := prod+t_5 \\
7. &\quad prod := t_6 \\
8. &\quad t_7 := i+1 \\
9. &\quad i := t_7 \\
10. &\quad \text{if } i <= 20 \text{ goto(1)}
\end{align*}
\]

7.
   a) Define LR parser. What are its merits? Also point out the drawbacks of LR parsing method.
   b) Suppose a DAG is given as input. How can you get an optimal code for the DAG? Write a algorithm for getting a heuristic ordering of DAG.
   c) How can you speed up the lexical analyzer using input buffering?