Theory of Computation – PYQ and Suggestions

1. What is a finite automaton? Explain the difference between DFA and NFA.

Answer:

- A **finite automaton (FA)** is a mathematical model of computation used to recognize **regular languages**.
- **DFA (Deterministic FA):** Each state has exactly one transition for each symbol.
- **NFA (Nondeterministic FA):** A state can have zero, one, or more transitions for a symbol; can include ε-transitions.

---

2. Explain the concept of ε-closure in NFA.

Answer:

- The **ε-closure** of a state is the set of states reachable from it by **ε-transitions only**.
- Used in converting **NFA to DFA**.

---

3. Define Pumping Lemma for Regular Languages. Give its significance.

Answer:

- **Pumping Lemma:** If L is regular, there exists an integer p such that any string $s \in L$ with $|s| \geq p$ can be divided as $s = xyz$, satisfying:
  1. $|y| > 0$
  2. $|xy| \leq p$
  3. For all $i \geq 0$, $xy^iz \in L$
- **Significance:** Used to **prove that a language is not regular**.

---

4. What is a context-free grammar (CFG)? Write its components.

Answer:

- A CFG is a 4-tuple **G = (V, Σ, R, S)**:
  1. **V** = set of variables (non-terminals)
  2. **Σ** = set of terminals
  3. **R** = set of production rules ($A \rightarrow \alpha$, $A \in V$, $\alpha \in (V \cup \Sigma)^*$)
  4. **S** = start symbol
- CFGs generate **context-free languages (CFLs)**.

---

5. Explain ambiguity in CFG. Give an example.

Answer:

- A CFG is **ambiguous** if there exists a string with **more than one leftmost or rightmost derivation**.
- **Example:**
  Grammar: E → E+E | E*E* | *id*
  *String: id+id*id has **two different parse trees**.

---

6. Define derivation tree (parse tree).

Answer:

- A **derivation tree** is a tree that represents the application of CFG rules to generate a string.
- Root = Start symbol, leaves = terminals of the string.

---

7. Explain Chomsky hierarchy of languages.

Answer:

Type	Grammar	Language	Machine
0	Unrestricted	Recursively enumerable	Turing machine
1	Context-sensitive	CSL	Linear bounded automaton (LBA)
2	Context-free	CFL	Pushdown automaton (PDA)
3	Regular	RL	Finite automaton (DFA/NFA)

---

8. What are closure properties of Regular Languages?

Answer:

- **Regular languages are closed under:**

1. Union: $L_1 \cup L_2$
2. Intersection: $L_1 \cap L_2$
3. Complement: L_1'
4. Concatenation: L_1L_2
5. Kleene Star: L_1^*
6. Reversal

---

9. Define linear bounded automaton (LBA).

Answer:

- An LBA is a **Turing machine with restricted tape**.
- Tape size is bounded by the **input length**.
- Recognizes **context-sensitive languages (CSL)**.

---

10. Why is Type-1 grammar called context-sensitive?

Answer:

- In Type-1 grammar, productions are of the form **$\alpha A \beta \rightarrow \alpha \gamma \beta$**, where $\gamma \neq \varepsilon$.
- A non-terminal **A can be replaced depending on the surrounding context (α, β)**, hence **context-sensitive**.

---

11. Define Turing machine and its components.

Answer:

- A Turing machine (TM) is a **mathematical model of computation** capable of simulating any algorithm.
- **Components:** (Q, Σ, Γ, δ, q_o, q_accept, q_reject)
  1. **Q** = set of states
  2. **Σ** = input alphabet
  3. **Γ** = tape alphabet ($\Sigma \subseteq \Gamma$)
  4. **δ** = transition function
  5. **q_o** = start state
  6. **q_accept, q_reject** = accept/reject states

---

12. Define recursively enumerable (RE) and recursive languages.

Answer:

- **Recursively enumerable (RE):** TM **may halt** on strings in L, may not halt otherwise.

- **Recursive:** TM **halts for every input**, accepts strings in L, rejects strings not in L.

---

13. Define left recursion in CFG and how it is removed.
Answer:

- **Left recursion:** A → Aα | β, leads to **infinite recursion in top-down parsing**.
- **Removed by:**
  1. Introduce new non-terminal A'.
  2. Rewrite:
     A → βA'
     A' → αA' | ε

---

14. Explain right-linear and left-linear grammars.
Answer:

- **Right-linear grammar:** A → aB or A → a
- **Left-linear grammar:** A → Ba or A → a
- Both generate **regular languages**, equivalent to **finite automata**.

---

15. Define language. How many languages are possible for an alphabet Σ = {a, b}?
Answer:

- **Language:** Any subset of **Σ*** (set of all strings over Σ).
- Σ = {a,b}, strings = {ε, a, b, aa, ab, ba, bb, …}
- Number of languages = **2^(number of strings in Σ*)**, which is **infinite**.

---

16. Prove that regular languages are closed under intersection.
Answer (Idea):

- Let L_1 and L_2 be regular, with DFA $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$, $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$
- Construct DFA $M = (Q_1 \times Q_2, \Sigma, \delta, (q_1, q_2), F_1 \times F_2)$
- $\delta((p, q), a) = (\delta_1(p, a), \delta_2(q, a))$
- M accepts $L_1 \cap L_2$, so regular languages are closed under intersection.

---

17. Define Myhill-Nerode theorem (short).
Answer:

- It provides a **necessary and sufficient condition for regularity**:

- A language L is regular ⇔ it has **finite equivalence classes** under relation: x ~ y iff ∀z, xz ∈ L ⇔ yz ∈ L.

---

18. Define pushdown automaton (PDA) and its relation to CFG.
Answer:

- PDA is a finite automaton with a **stack**, used to recognize **context-free languages (CFL)**.
- **Relation:** For every CFG, there exists an equivalent PDA, and vice versa.

---

19. Explain non-determinism in Turing machines.
Answer:

- **Non-deterministic TM (NTM):** Can move to **multiple states for the same input and tape symbol**.
- Accepts a string if **any computation path leads to accept**.

---

20. What is a regular expression? How is it related to FA?
Answer:

- **Regular expression (RE):** Symbolic representation of **regular languages** using operations: union, concatenation, Kleene star.
- **Equivalence:** Every RE ↔ Finite Automaton.