UNIT 1

1. Finite Automaton

- **Definition**: A mathematical model with a finite number of states used to recognize patterns in strings.
- **Components**:
  - **States (Q)**: Set of all possible states.
  - **Alphabet (Σ)**: Set of input symbols.
  - **Transition Function (δ)**: Defines state changes based on input.
  - **Start State (q_0)**: Initial state.
  - **Accept States (F)**: States where the string is accepted.

2. Deterministic and Non-Deterministic Finite Automata

- **DFA (Deterministic Finite Automaton)**:
  - For every state and input, there is **exactly one transition**.
  - Easier to implement.
  - Example: Recognizing binary strings divisible by 3.
- **NDFA (Non-Deterministic Finite Automaton)**:
  - Allows **multiple transitions** for a single input or state.
  - Uses ε (epsilon) transitions (moves without input).
  - More abstract and flexible.

3. Transition Systems and Properties

- **Transition Function**: δ(q,a)→q′δ(q, a) \to q'δ(q,a)→q′, maps state qqq and input aaa to a new state q′q'q′.
- **Properties**:
  - Completeness: Defined for all states and inputs.
  - Closure: DFA and NDFA transitions are closed under certain operations like union, concatenation, etc.

4. Acceptability of a String

- A string is **accepted** if, starting from the initial state, it ends in an accept state after processing all input symbols.

5. Equivalence of DFA and NDFA

- **Theorem**: Every NDFA can be converted to an equivalent DFA.
- **Process**:
    - Construct subsets of NDFA states (subset construction).
    - Each subset corresponds to a DFA state.

6. Mealy and Moore Machines

- **Mealy Machine**:
    - Outputs depend on both the state and input.
    - Output: λ(q,a)→outputλ(q, a) \to outputλ(q,a)→output.
- **Moore Machine**:
    - Outputs depend only on the state.
    - Output: λ(q)→outputλ(q) \to outputλ(q)→output.

7. Minimization of Finite Automata

- **Goal**: Reduce the number of states in DFA while preserving functionality.
- **Steps**:
    1. Remove unreachable states.
    2. Combine equivalent states.

8. Basics of Strings and Alphabets

- **Alphabet (Σ)**: Set of symbols (e.g., {0, 1} for binary).
- **String**: Sequence of symbols from the alphabet.
- **Language**: Set of strings over an alphabet.

9. Transition Graph and Properties

- **Transition Graph**: Visual representation of states and transitions.
- **Properties**:
    - Can represent DFA, NDFA, or ε-NDFA.
    - Helps in visualizing language recognition.

10. Regular Languages

- Defined by regular expressions and recognized by finite automata.
- **Operations**:
  - Union (A ∪ B).
  - Concatenation (A · B).
  - Kleene Star (A*).

11. Equivalence of Deterministic and Non-Deterministic Automata

- Proven by subset construction method.
- Both recognize the same class of languages: **regular languages**.

Quick Example

- Alphabet: {a, b}
- DFA Example:
  - States: {q_0, q_1}, Start: q_0, Accept: q_1.
  - Transition: δ(q0,a)→q1,δ(q1,b)→q0δ(q_0, a) \to q_1, δ(q_1, b) \to q_0δ(q0,a)→q1,δ(q1,b)→q0.
  - Accepts strings with alternating "a" and "b".

Important MCQ

1. Basic Definitions

1. What is a Finite Automaton?
   a) A device with finite memory
   b) A mathematical model with finite states
   c) A machine with infinite memory
   d) None of the above
   Answer: b

2. The components of a finite automaton are:
   a) States, Alphabet, Transition Function, Start State, Accept States
   b) Input, Output, Memory
   c) Transition Table and Stack
   d) None of the above
   Answer: a

3. What does DFA stand for?
   a) Deterministic Functional Automaton
   b) Deterministic Finite Automaton
   c) Deterministic Finite Algorithm
   d) Deterministic Formal Algorithm
   Answer: b

2. Deterministic and Non-Deterministic Automata

4. In a DFA, for every state and input symbol, there is:
   a) Exactly one transition
   b) At most one transition
   c) Zero transitions
   d) Multiple transitions
   Answer: a

5. NDFA allows:
   a) Only one transition per input
   b) Multiple transitions and ε-transitions
   c) No transitions for some inputs
   d) None of the above
   Answer: b

6. Which is true about DFA and NDFA?
   a) NDFA is more powerful than DFA
   b) DFA is faster than NDFA
   c) DFA and NDFA recognize the same languages
   d) None of the above
   Answer: c

3. Transition Systems

7. What is the purpose of the transition function in a finite automaton?
   a) Determines state changes based on input
   b) Converts NDFA to DFA
   c) Accepts or rejects strings
   d) None of the above
   Answer: a

8. Transition functions in DFA are:
   a) Total functions
   b) Partial functions
   c) Recursive functions
   d) None of the above
   Answer: a

9. What is the output of the transition function $\delta(q0,a)\delta(q_0, a)\delta(q0,a)$ if $q0q_0q0$ is the start state?
   a) Input symbol
   b) Next state
   c) Current state
   d) None of the above
   Answer: b

4. Acceptability of Strings

10. A string is accepted by a finite automaton if:
    a) It starts with an accept state
    b) It ends in an accept state
    c) It is processed by all states
    d) None of the above
    Answer: b

11. What is the language of a finite automaton?
    a) Set of states
    b) Set of all strings it accepts
    c) Set of transitions
    d) None of the above
    Answer: b

5. Equivalence of DFA and NDFA

12. Every NDFA can be converted to:
    a) DFA
    b) Moore Machine
    c) Mealy Machine
    d) None of the above
    Answer: a

13. Which method is used to convert NDFA to DFA?
    a) Transition reduction
    b) Subset construction
    c) State minimization
    d) State partitioning
    Answer: b

6. Mealy and Moore Machines

14. In a Mealy Machine, the output depends on:
    a) State only
    b) Input and State
    c) Input only
    d) None of the above
    Answer: b

15. In a Moore Machine, the output depends on:
    a) State only
    b) Input and State
    c) Input only
    d) None of the above
    Answer: a

16. Which is true for Mealy and Moore machines?
    a) Moore machines are faster than Mealy machines
    b) Mealy machines produce output earlier
    c) Mealy and Moore machines recognize the same languages
    d) None of the above
    Answer: c

7. Minimization of Finite Automata

17. Why is DFA minimization performed?
    a) To reduce memory usage
    b) To improve efficiency
    c) To reduce states while preserving behavior
    d) All of the above
    Answer: d

18. Which step is part of DFA minimization?
    a) Removing unreachable states
    b) Adding new states
    c) Splitting transitions
    d) None of the above
    Answer: a

8. Strings and Alphabets

19. What is an alphabet in the context of finite automata?
    a) Set of strings
    b) Set of symbols
    c) Transition table
    d) None of the above
    Answer: b

20. Which of the following represents a string?
    a) Sequence of states
    b) Sequence of symbols from an alphabet
    c) Sequence of transitions
    d) None of the above
    Answer: b

9. Transition Graphs

21. A transition graph is used to:
    a) Represent states and transitions visually
    b) Minimize DFA
    c) Convert regular expressions
    d) None of the above
    Answer: a

22. Transition graphs are equivalent to:
    a) Transition tables
    b) Regular expressions
    c) Both a and b
    d) None of the above
    Answer: c

10. Regular Languages

23. Regular languages can be recognized by:
    a) Finite automata
    b) Turing machines
    c) Context-free grammars
    d) None of the above
    Answer: a

24. Regular languages are closed under:
    a) Union
    b) Concatenation
    c) Kleene Star
    d) All of the above
    Answer: d

25. Which is true about regular languages?
    a) They are recognized by all machines
    b) They can be expressed with finite automata
    c) They require infinite memory to recognize
    d) None of the above
    Answer: b

11. Equivalence of DFA and NDFA

26. Which languages do DFA and NDFA both recognize?
    a) Regular languages
    b) Context-free languages
    c) Recursive languages
    d) None of the above
    Answer: a

27. Which of the following is NOT a property of DFA?
    a) Deterministic transitions
    b) Unique next state for each input
    c) ε-transitions allowed
    d) Finite number of states
    Answer: c

28. Which type of automaton uses ε-transitions?
    a) DFA
    b) NDFA
    c) Turing Machine
    d) None of the above
    Answer: b

29. Which finite automaton is easier to implement in hardware?
    a) DFA
    b) NDFA
    c) Moore Machine
    d) Mealy Machine
    Answer: a

30. What is a distinguishing feature of regular languages?
    a) Infinite memory
    b) Can be recognized by finite automata
    c) Require a stack for recognition
    d) None of the above
    Answer: b