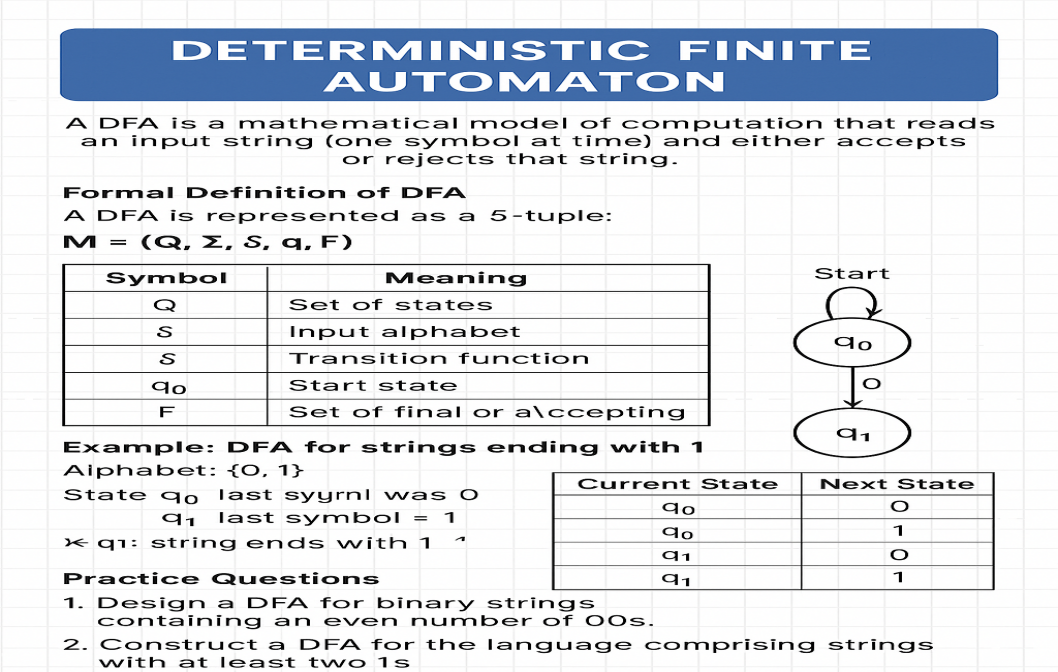
Theory of Computing – DFA, NFA, and NFA to DFA Conversion

1. Deterministic Finite Automaton (DFA)
A Deterministic Finite Automaton (DFA) is a finite state machine that accepts or rejects strings of symbols. It is called deterministic because for each state and input symbol, there is exactly one next state. **Formal Definition:**
A DFA is represented as a 5-tuple: M = (Q, Σ, δ, q■, F), where:
- Q: Finite set of states
- Σ: Input alphabet
- δ: Transition function δ: Q × Σ → Q
- q■: Start state
- F: Set of final (accepting) states

Characteristics:
- Exactly one transition per symbol per state
- No ε (epsilon) transitions
- Used for regular languages

Example:
Design a DFA that accepts binary strings ending with 1.
States: Q = {q■, q■}, Σ = {0, 1}, Start = q■, Final = {q■}
Transition table:
δ(q■, 0) = q■
δ(q■, 1) = q■
δ(q■, 0) = q■
δ(q■, 1) = q■



2. Nondeterministic Finite Automaton (NFA)
An NFA is similar to a DFA but allows multiple transitions for the same symbol or even transitions without consuming any input (ε-transitions). **Formal Definition:**
An NFA is represented as a 5-tuple: M = (Q, Σ, δ, q■, F), where:
- Q: Finite set of states
- Σ: Input alphabet
- δ: Transition function δ: Q × (Σ ∪ {ε}) → 2^Q
- q■: Start state
- F: Set of final (accepting) states

Characteristics:
- Can have multiple or zero transitions for a symbol
- Can have ε (epsilon) transitions
- Easier to design than DFA
- Equivalent in power to DFA (accepts same set of regular languages)

Example:
NFA that accepts strings ending with '1' over {0,1}.
q■ → (1) → q■ (final)
q■ → (0) → q■
q■ → (0,1) → q■

3. NFA to DFA Conversion
The process of converting an NFA into an equivalent DFA is called **subset construction** or **powerset construction**. **Steps:**
1. Start with the ε-closure of the NFA start state — this becomes the DFA start state.
2. For each DFA state and input symbol, find the union of ε-closures of all reachable NFA states.
3. Repeat until no new states are found.
4. Any DFA state containing at least one NFA final state is a DFA final state.

Example:
NFA:
States = {q■, q■}, Alphabet = {0,1}
δ(q■, ε) = {q■}, δ(q■, 0) = {q■}, δ(q■, 1) = {q■}

Equivalent DFA constructed using subsets of NFA states.
Note: Both DFA and NFA recognize exactly the same class of languages — the **regular languages**.

4. Practice Exercises
1. Design a DFA that accepts binary strings with an even number of 0s.
2. Construct an NFA for strings that contain the substring '101'.
3. Convert the above NFA to DFA using the subset construction method.
4. Explain why DFA and NFA are equivalent in expressive power.