Definitions

📘 Automata Theory Definitions

🔹 Deterministic Finite Automaton (DFA)

A DFA is a 5-tuple:

```
M = (Q, Σ, δ, q₀, F)
```

Where:

- `Q`: finite set of states
- `Σ`: finite input alphabet
- `δ: Q × Σ → Q`: transition function
- `q₀ ∈ Q`: start state
- `F ⊆ Q`: set of accepting (final) states

✅ For every state and symbol, **exactly one transition** exists.

🔹 Nondeterministic Finite Automaton (NFA)

An NFA is also a 5-tuple:

```
M = (Q, Σ, δ, q₀, F)
```

Where:

- `δ: Q × Σ → 2^Q`

✅ Allows **multiple transitions** for the same symbol and state.

🔹 Epsilon-NFA (ε-NFA)

Similar to NFA but includes ε-transitions (moves without consuming input):

```
δ: Q × (Σ ∪ {ε}) → 2^Q
```

✅ Can change state **without reading input**.

🔹 Mealy Machine

A Mealy machine is a 6-tuple:

```
M = (Q, Σ, Γ, δ, λ, q₀)
```

Where:

- Γ: output alphabet
- $\delta: Q \times \Sigma \to Q$: transition function
- $\lambda: Q \times \Sigma \to \Gamma$: output function

✅ **Output depends on current state and input.**

🔹 Moore Machine

A Moore machine is a 6-tuple:

```
M = (Q, Σ, Γ, δ, λ, q₀)
```

Where:

- $\lambda: Q \to \Gamma$: output depends only on current state

✅ **Output is associated with states**, not transitions.

🔹 Context-Free Grammar (CFG)

A CFG is a 4-tuple:

```
G = (V, Σ, R, S)
```

Where:

- V: variables (non-terminals)
- Σ: terminals
- R: rules $A \to \alpha$, where $\alpha \in (V \cup \Sigma)^*$
- $S \in V$: start symbol

✅ Generates **Context-Free Languages (CFLs)**.

🔹 Chomsky Normal Form (CNF)

A CFG is in CNF if every production is of the form:

- `A → BC` (two non-terminals)
- `A → a` (a terminal)
- `S → ε` (optional, if empty string is in language)

✅ Useful for parsing algorithms like **CYK**.

🔹 Greibach Normal Form (GNF)

A CFG is in GNF if every production is of the form:

```
A → a α
```

Where:

- `a ∈ Σ` (a terminal)
- `α ∈ V*` (zero or more non-terminals)

✅ Starts with a terminal.

🔹 Backus-Naur Form (BNF)

A **notation** used to write CFGs:

Example:

```
<expr> ::= <term> | <expr> + <term>
```

Where:

- `::=` means "is defined as"
- `<...>` denotes non-terminals

✅ Common in language syntax specs.

🔹 Pushdown Automaton (PDA)

A PDA is a 7-tuple:

```
M = (Q, Σ, Γ, δ, q₀, Z₀, F)
```

Where:

- `Γ`: stack alphabet
- `δ: Q × (Σ ∪ {ε}) × Γ → 2^(Q × Γ*)`: transition function
- `Z₀`: initial stack symbol

✅ Recognizes **context-free languages** using a stack.

🔹 Nondeterministic PDA (N-PDA)

Same as PDA, but allows multiple choices:

- All PDAs are **nondeterministic** unless specified otherwise.
- A **DPDA** has at most one possible move per step.

🔹 Turing Machine (TM)

A TM is a 7-tuple:

```
M = (Q, Σ, Γ, δ, q₀, q_accept, q_reject)
```

Where:

- `Γ`: tape alphabet (includes blank symbol)
- `δ: Q × Γ → Q × Γ × {L, R}`: transition function
- `q₀`: start state
- `q_accept`, `q_reject`: halting states

✅ Recognizes **recursively enumerable languages**
✅ Has **infinite tape** and serves as a general model of computation.
