3. Deterministic Pushdown Automata

Deterministic Pushdown automata: A deterministic pushdown automata (DPDA) is a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$, where

- Q is a finite of *states*;
- Σ is a finite set of *input alphabet*;
- Γ is a finite set of *stack alphabet*,
- $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\epsilon \to (Q \times \Gamma_\epsilon) \cup \{\varnothing\}$ is the *transition function*;
- $q_0 \in Q$ is the *start state*;
- $F \subseteq Q$ is the set of *acceptance states*.

For every $q \in Q, a \in \Sigma$ and $x \in \Gamma$, *exactly one* of the values

$$\delta(q, a, x), \quad \delta(q, a, \epsilon), \quad \delta(q, \epsilon, x), \quad \delta(q, \epsilon, \epsilon)$$

is not $\varnothing$.

Acceptance of DPDA

- **Accept**: If a DPDA enters an accept state after it has read the last input symbol of an input string, then it accepts that string.
- **Reject**: In all other cases, it reject that string. If could be one of the following cases:
  - The DPDA reads the entire input but does not enter an accept state when it is at the end, or
  - The DPDA fails to read the entire input string,
    - The DPDA tries to pop an empty stack, or
    - The DPDA makes an endless sequence of ϵ - input moves without reading any new inputs.

Deterministic context-free language: The language of a DPDA is a deterministic context-free language (DCFL).

Lemma. Every DPDA has an equivalent DPDA that always reads the entire input string.

Proof. There are two situations, poping an empty stack (hanging) and endless ϵ - input moving (looping).

- ***Hanging***: add a special symbol at the bottom of the stack, and if we pop the empty stack, then go to a special state q_{reject}, and read the rest of the inputs.
- ***Looping***: find all the ϵ - loops and add transitions to q_{reject} of the states in the loops, and read the rest of the inputs.

Theorem. The class of DCFLs is closed under complementation, that is, if A is DCFL, then

$$\Sigma^* - A = \{s \in \Sigma^* \mid s \notin A\}$$

is also a DCFL.

Proof. Add a reading state before any reading transitions, transforming the DPDA to the equivalent form. Then swap all the reading state (from acceptance state to ordinary state, from ordinary state to acceptance state).

Corollary. Any CFL whose complement is not a CFL is not a DCFL.

Definition. (Endmarked languages) For any language A the end marked language $A \dashv$ is defined by

$$\{w \dashv \mid w \in A\}$$

Here $\dashv$ is the special endmarker symbol.

Theorem. A is a DCFL iff $A \dashv$ is a DCFL.

Forced Handles: A handle h of valid string $v = xhy$ is a forced handle if h is the unique handle in every valid string $xh\hat{y}$ where $\hat{y} \in \Sigma^*$.

Deterministic Context-Free Grammar (DCFG): A deterministic context-free grammar is a context-free grammar such that every valid string has a forced handle.

- The definition does not tell us how to check whether a grammar is DCFG.
- We use **DK-test** to check whether a grammar is DCFG.

Theorem. Every DCFG has an equivalent DPDA; every DPDA that recognizes an endmarked language has an equivalent DCFG.