Theory of Computation Slides
based on Michael Sipser's Textbook

Moses A. Boudourides[1]

Visiting Associate Professor of Computer Science
Haverford College

[1] Moses.Boudourides@cs.haverford.edu

Section 2.2

Pushdown Automata

March 1, 2022

Definition of Pushdown Automata (PDA)

Definition: **A (Nondeterministic) Pushdown Automaton**

A (nondeterministic) **pushdown automaton** (**PDA**) is a 7–tuple $(Q, \Sigma, \Gamma, q_0, Z_0, F, \delta)$, where

- Q is the (finite) set of **states**,
- Σ is the (finite) **input alphabet**,
- Γ is the (finite) **stack alphabet**,
- $q_0 \in Q$ is the (input) **start state**,
- $Z_0 \in \Gamma$ is the **stack start symbol**,
- $F \subseteq Q$ is the **set of accepting** or (**final**) **states**, and
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \to \mathcal{P}(Q \times \Gamma^*)$ is the **transition function** (or better said **multi–function** or **relation**) (where $\mathcal{P}(X)$ denotes the **power set** of set of X).

Transitions of a PDA, I

How a PDA computes

Let us denote $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ and $\Gamma_\varepsilon = \Gamma \cup \{\varepsilon\}$. Writing the transition function of a PDA M as:

$$\delta(p, \sigma, Z) = \{(q_1, \gamma_1), (q_2, \gamma_2), \ldots, (q_m, \gamma_m)\},$$

where $p, q_1, q_2, \ldots q_m \in Q$ are states, $\sigma \in \Sigma_\varepsilon$ is an input symbol, $Z \in \Gamma_\varepsilon$ is a stack symbol, and $\gamma_1, \gamma_2, \ldots, \gamma_m \in \Gamma^*$ are stack strings, means that:

- when M is on state p and it is reading the **leftmost** input symbol σ (notice that, since $\sigma \in \Sigma_\varepsilon$, it is possible that $\sigma = \varepsilon$), while the **top of the stack** is occupied by stack symbol Z (notice that, since $Z \in \Sigma_\varepsilon$, one may consider that $Z = \varepsilon$),
- then M transitions to states $q_1, q_2, \ldots q_m$ by removing σ from the input string and replacing Z with γ_i, for each $i = 1, 2, \ldots, m$ (respectively).

Transitions of a PDA, II

How a PDA computes (cont.)

For $p, q \in Q, \sigma \in \Sigma_\varepsilon, Z \in \Gamma_\varepsilon, \gamma \in \Gamma^*$,

$$p \xrightarrow{\sigma \mid Z,\gamma} q \text{ means } \delta(p, \sigma, Z) \ni (p, \gamma),$$

i.e., when a PDA M is on state p, reads the rightmost input symbol σ, while symbol Z is on top of the stack, then M transitions to state q, removes σ from the input and replaces Z with γ on the top of the stack. Particular cases:

- $\sigma \mid \varepsilon, \gamma$ means reading input or stack symbol σ, adding string γ on the top of the stack.
- $\sigma \mid Z, \varepsilon$ means reading input symbol σ, removing symbol Z from the top of the stack.
- $\sigma \mid \varepsilon, \varepsilon$ means reading input symbol σ, without making any change on the stack.
- $\varepsilon \mid Z, \gamma$ means without reading any input symbol, replacing Z with γ on the top of the stack.
- $\varepsilon \mid \varepsilon, \gamma$ means without reading any input or stack symbol, adding string γ on the top of the stack.
- $\varepsilon \mid Z, \varepsilon$ means without reading any input symbol, removing symbol Z from the top of the stack.

Configurations and Moves

Definition: Configurations and Moves

Let $M = (Q, \Sigma, \Gamma, q_0, Z_0, F, \delta)$ be a PDA.

- A **configuration** (or **instantaneous discription**) of M is a triplet (p, w, γ) such that, when M is on state $p \in Q$, the part of the input string that is about to be read is string $w \in \Sigma^*$ and, at the same instance, the contents of the (whole) stack are given by string $\gamma \in \Gamma^*$.
- Two configurations $(p, \sigma w, Z\alpha)$ and $(q, w, \gamma\alpha)$ (for $p, q \in Q, \sigma \in \Sigma_\varepsilon, w \in \Sigma^*, Z \in \Gamma_\varepsilon, \alpha, \gamma \in \Gamma^*$) are said to form a **move in one step**, written as

$$(p, \sigma w, Z\alpha) \vdash (q, w, \gamma\alpha),$$

whenever $\delta(p, \sigma, Z) \ni (q, \gamma)$, i.e., whenever M is on state p, reads the rightmost input symbol σ and at the stack's top is symbol Z, then M transitions to state q replacing Z with (string) γ on the stack.

Chains of Moves

Definition: Chains of Moves

Let $C_0, C_1, \ldots, C_n$ be a sequence of configurations such that every successive two configurations form a move in one step. Then these configurations are said to form (a **chain of**) **moves in** n **steps**, $C_0 \vdash C_1 \vdash \cdots \vdash C_n$, which is symbolically written as

$$C_0 \vdash^n C_n.$$

As previously, the notation

$$C_0 \vdash^* C_n.$$

will refer to a move of 1 or more steps.

Languages Accepted by PDAs

Definition: Languages accepted by PDAs

Let $M = (Q, \Sigma, \Gamma, q_0, Z_0, F, \delta)$ be a PDA.

- The **language accepted by empty stack** by M is the set

  $$L(M) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash^* (q_f, \varepsilon, \varepsilon),\ \text{for } q_f \in F\}.$$

- The **language accepted by final state** by M is the set

  $$L_{FS}(M) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash^* (q_f, \varepsilon, \gamma), \\ \text{for } q_f \in F \text{ and } \gamma \in \Gamma^*\}.$$

Theorem 1

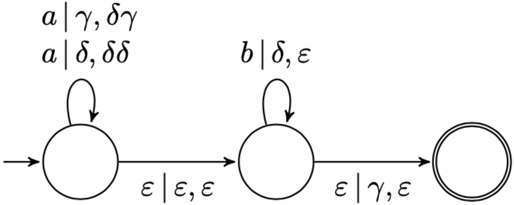
$L = L_{FS}(M_1)$, for some PDA M_1, if and only if there exists PDA M such that $L = L(M)$.

Theorem 1

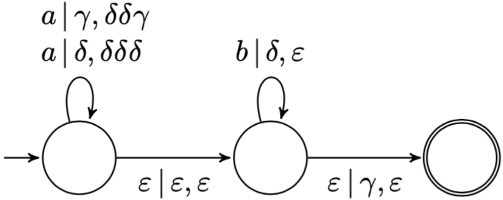
L is a context–free language (CFL) if and only if $L = L(M)$, for some PDA M.

Examples of Pushdown Automata, I

Example 1: $L = \{a^i b^i \mid i \geq 0\}$

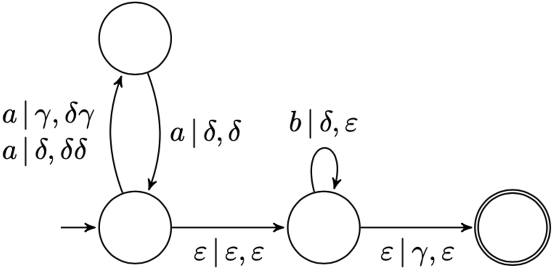


Example 2: $L = \{a^i b^{2i} \mid i \geq 0\}$

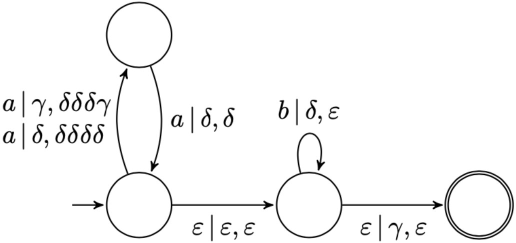


Examples of Pushdown Automata, II

Example 3: $L = \{a^{2i}b^i \mid i \geq 0\}$

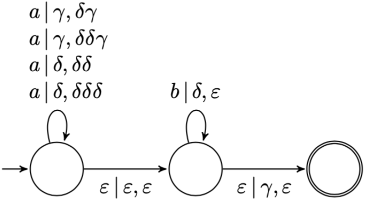


Example 4: $L = \{a^{2i}b^{3i} \mid i \geq 1\}$



Examples of Pushdown Automata, III

Example 5: $L = \{a^i b^j \mid 0 \leq i \leq j \leq 2i\}$



Example 6: $L = \{w \in (a+b)^\star \mid n_a(w) = n_b(w) \geq 0\}$