CS 4410

Automata, Computability, and Formal Language

Dr. Xuejun Liang

Spring 2019

Chapter 7

Pushdown Automata

1. Nondeterministic Pushdown Automata
   - Definition of a Pushdown Automata
   - The Language Accepted by a Pushdown Automaton
2. Pushdown Automata and Context-Free Languages
   - Pushdown Automata for Context-Free Languages
   - Context-Free Grammar for Pushdown Automata
3. Deterministic Pushdown Automata and Deterministic Context-Free Languages
4. Grammars for Deterministic Context-Free Languages*

Learning Objectives

At the conclusion of the chapter, the student will be able to:

- Describe the components of a nondeterministic pushdown automaton
- State whether an input string is accepted by a nondeterministic pushdown automaton
- Construct a pushdown automaton to accept a specific language
- Given a context-free grammar in Greibach normal form, construct the corresponding pushdown automaton
- Describe the differences between deterministic and nondeterministic pushdown automata
- Describe the differences between deterministic and general context-free languages

Nondeterministic Pushdown Automata

Definition 7.1: A **nondeterministic pushdown accepter** (npda) is defined by the sep-tuple $M=(Q, \Sigma, \Gamma, \delta, q_0, z, F)$
where

Q is a finite set of internal states of the control unit,
Σ is the input alphabet,
Γ is a finite set of symbols called the **stack alphabet**,
$\delta: Q\times(\Sigma\cup\{\lambda\})\times\Gamma \rightarrow$ finite subsets of $Q\times\Gamma^*$ is the transition function,
$q_0 \in Q$ is the initial state of the control unit
$z \in \Gamma$ is the **stack start symbol**
$F \subseteq Q$ is the set of final states

Example 7.1: $\delta(q_1, a, b) =\{(q_2, cd), (q_3, \lambda)\}$ means …

Example 7.2: $Q=\{q_0, q_1, q_2, q_3\}$, $\Sigma=\{a, b\}$, $\Gamma=\{0, 1\}$, $z=0$, $F=\{q_3\}$
$\delta(q_0, a, 0)=\{(q_1, 10), (q_3, \lambda)\}$, $\delta(q_0, \lambda, 0)=\{(q_3, \lambda)\}$, $\delta(q_1, a, 1)=\{(q_1,11)\}$
$\delta(q_1, b, 1)=\{(q_2, \lambda)\}$, $\delta(q_2, b, 1)=\{(q_2, \lambda)\}$, $\delta(q_2, \lambda, 0)=\{(q_3, \lambda)\}$

Language accepted by a Pushdown Automata

In the triplet (q, w, u)

q: current state, w: unread part of input string, and u: stack content

A move from (q_1, aw, bx) to (q_2, w, yx) denoted by $(q_1, aw, bx) \vdash (q_2, w, yx)$ is possible, if and only if $(q_2, y) \in \delta(q_1, a, b)$.

Definition 7.2: Let $M=(Q, \Sigma, \Gamma, \delta, q_0, z, F)$ be a nondeterministic pushdown automaton. The language accepted by M is the set

$$L(M) = \{w \in \Sigma^* : (q_0, w, z) \vdash_M^* (p, \lambda, u), p \in F, u \in \Gamma^*\}$$

In words, the language accepted by M is the set of all strings that can put M into a final state at the end of the string. The final stack content u is irrelevant to this definition of acceptance.

Example 7.4: Construct an npda for the language

$$L=\{ w \in \{a, b\}^* : n_a(w) = n_b(w)\}$$

Example 7.5: Construct an npda for the language

$$L=\{ ww^R : w \in \{a, b\}^+ \}$$

Pushdown Automata for Context-Free Languages

Example 7.6: Construct an npda that accepts the language generated by grammar with productions

$S \rightarrow aSbb|a$

$(q_0, \lambda, z) \vdash (q_1, \lambda, Sz)$	
$(q_1, a, S) \vdash (q_1, \lambda, SA)$	$S \rightarrow aSA$
$(q_1, a, S) \vdash (q_1, \lambda, \lambda)$	$S \rightarrow a$
$(q_1, b, A) \vdash (q_1, \lambda, B)$	$A \rightarrow bB$
$(q_1, b, A) \vdash (q_1, \lambda, \lambda)$	$B \rightarrow b$
$(q_1, \lambda, z) \vdash (q_f, \lambda, z)$	

Theorem 7.1: For every context-free language L, there exists an npda M such that L=L(M).

Example 7.7: Construct an npda that accepts the language generated by grammar with productions

$S \rightarrow aA,$
$A \rightarrow aABC|bB|a,$
$B \rightarrow b,$
$C \rightarrow c.$

Context-Free Grammars for Pushdown Automata

Two properties of an npda

1. Single final state q_f, which is entered if and only if the stack is empty
2. All transitions must have the form $\delta(q_i, a, A) = \{c_1, c_2, .., c_n\}$, where

   $c_j = (q_j, \lambda)$ $((q_i, a, A) \vdash (q_j, \lambda, \lambda))$ or

   $c_j = (q_j, BC)$ $((q_i, a, A) \vdash (q_j, \lambda, BC))$

Build the grammar from an npda with the two properties

1. Variable: (q_iAq_j) and Staring variable: (q_0zq_f)
2. Production: $(q_iAq_j) \rightarrow a$ if $(q_i, a, A) \vdash (q_j, \lambda, \lambda)$

   $\forall q_l, q_k \in Q$, $(q_iAq_k) \rightarrow a\,(q_jBq_l)\,(q_lCq_k)$ if $(q_i, a, A) \vdash (q_j, \lambda, BC)$

Example 7.8: Consider the npda with transitions

$\delta(q_0, a, z) = \{(q_0, Az)\}$, $\delta(q_0, a, A) = \{(q_0, A)\}$,

$\delta(q_0, b, A) = \{(q_1, \lambda)\}$, $\delta(q_1, \lambda, z) = \{(q_2, \lambda)\}$. q_2 is the final state.

Theorem 7.2: If L=L(M) for some npda M, then L is a context-free language.

Deterministic Pushdown Automata and Deterministic Context-Free Languages

Definition 7.3: A **deterministic pushdown accepter** (dpda) is a pushdown automata as defined in Definition 7.1, subject to the restrictions that, for every $q \in Q$, $a \in \Sigma \cup \{\lambda\}$ and $b \in \Gamma$,

1. $\delta(q, a, b)$ contains at most one element,
2. If $\delta(q, \lambda, b)$ is not empty, then $\delta(q, c, b)$ must be empty for every $c \in \Sigma$.

Definition 7.4: A language L is said to be a deterministic context-free language if and only if there exists a dpda such that L = L(M).

In contrast to finite automata, deterministic and non-deterministic pushdown automata are not equivalent.

Examples 7.10: $L=\{a^n b^n : n \geq 0\}$ is deterministic context-free language.
$L=\{ww^R : w \in \{a, b\}^+\}$ is not deterministic.