3.13 Pushdown Automata

We have seen that the regular languages are exactly the languages accepted by DFA's or NFA's. The context-free languages are exactly the languages accepted by pushdown automata, for short, PDA's. However, although there are two versions of PDA's, deterministic and nondeterministic, contrary to the fact that every NFA can be converted to a DFA, nondeterministic PDA's are strictly more poweful than deterministic PDA's (DPDA's). Indeed, there are context-free languages that cannot be accepted by DPDA's.

Thus, the natural machine model for the context-free languages is nondeterministic, and for this reason, we just use the abbreviation PDA, as opposed to NPDA.

We adopt a definition of a PDA in which the pushdown store, or stack, must not be empty for a move to take place. Other authors allow PDA's to make move when the stack is empty. Novices seem to be confused by such moves, and this is why we do not allow moves with an empty stack.

Intuitively, a PDA consists of an input tape, a nondeterministic finite-state control, and a stack.

Given any set X possibly infinite, let $\mathcal{P}_{fin}(X)$ be the set of all finite subsets of X.

Definition 3.13.1 A *pushdown automaton* is a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, where

- Q is a finite set of *states*;
- Σ is a finite *input alphabet*;
- Γ is a finite *pushdown store (or stack) alphabet*;
- $q_0 \in Q$ is the *start state* (or *initial state*);
- $Z_0 \in \Gamma$ is the *initial stack symbol* (or *bottom marker*);
- $F \subseteq Q$ is the set of *final (or accepting) states*;
- $\delta: Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to \mathcal{P}_{fin}(Q \times \Gamma^*)$ is the *transition function*.

A transition is of the form $(q, \gamma) \in \delta(p, a, Z)$, where $p, q \in Q$, $Z \in \Gamma$, and $a \in \Sigma \cup \{\epsilon\}$. A transition of the form $(q, \gamma) \in \delta(p, \epsilon, Z)$ is called an *ϵ-transition (or ϵ-move)*.

The way a PDA operates is explained in terms of *Instantaneous Descriptions*, for short ID's. Intuitively, an Instantaneous Description is a snapshot of the PDA. An ID is a triple of the form

$$(p, u, \alpha) \in Q \times \Sigma^* \times \Gamma^*.$$

The idea is that p is the current state, u is the remaining input, and α represents the stack.

It is important to note that we use the convention that the **leftmost** symbol in α represents the topmost stack symbol.

Given a PDA M, we define a relation $\vdash_M$ between pairs of ID's. This is very similar to the derivation relation $\Longrightarrow_G$ associated with a context-free grammar.

Intuitively, a PDA scans the input tape symbol by symbol from left to right, making moves that cause a change of state, an update to the stack (but only at the top), and either advancing the reading head to the next symbol, or not moving the reading head during an ϵ-move.

Definition 3.13.2 Given a PDA

$$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F),$$

the relation $\vdash_M$ is defined as follows:

(1) For any move $(q, \gamma) \in \delta(p, a, Z)$, where $p, q \in Q$, $Z \in \Gamma$, $a \in \Sigma$, for every ID of the form $(p, av, Z\alpha)$, we have

$$(p, av, Z\alpha) \vdash_M (q, v, \gamma\alpha).$$

(2) For any move $(q, \gamma) \in \delta(p, \epsilon, Z)$, where $p, q \in Q$, $Z \in \Gamma$, for every ID of the form $(p, u, Z\alpha)$, we have

$$(p, u, Z\alpha) \vdash_M (q, u, \gamma\alpha).$$

As usual, $\vdash_M^+$ is the transitive closure of $\vdash_M$, and $\vdash_M^*$ is the reflexive and transitive closure of $\vdash_M$.

A move of the form

$$(p, au, Z\alpha) \vdash_M (q, u, \alpha)$$

where $a \in \Sigma \cup \{\epsilon\}$, is called a *pop move*.

A move on a real input symbol $a \in \Sigma$ causes this input symbol to be consumed, and the reading head advances to the next input symbol. On the other hand, during an ϵ-move, the reading head stays put.

When

$$(p, u, \alpha) \vdash_M^* (q, v, \beta)$$

we say that we have a *computation*.

There are several equivalent ways of defining acceptance by a PDA.

Definition 3.13.3 Given a PDA

$$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F),$$

the following languages are defined:

(1) $T(M) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash_M^* (f, \epsilon, \alpha)$, where $f \in F$, and $\alpha \in \Gamma^*\}$.

We say that $T(M)$ is the *language accepted by M by final state*.

(2) $N(M) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash_M^* (q, \epsilon, \epsilon)$, where $q \in Q\}$.

We say that $N(M)$ is the *language accepted by M by empty stack*.

(3) $L(M) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash_M^* (f, \epsilon, \epsilon)$, where $f \in F\}$.

We say that $L(M)$ is the *language accepted by M by final state and empty stack*.

In all cases, note that the input w must be consumed entirely.

The following lemma shows that the acceptance mode does not matter for PDA's. As we will see shortly, it does matter for DPDAs.

Lemma 3.13.4 *For any language L, the following facts hold.*

(1) If $L = T(M)$ for some PDA M, then $L = L(M')$ for some PDA M'.

(2) If $L = N(M)$ for some PDA M, then $L = L(M')$ for some PDA M'.

(3) If $L = L(M)$ for some PDA M, then $L = T(M')$ for some PDA M'.

(4) If $L = L(M)$ for some PDA M, then $L = N(M')$ for some PDA M'.

In view of lemma 3.13.4, the three acceptance modes T, N, L are equivalent.

The following PDA accepts the language

$$L = \{a^n b^n \mid n \geq 1\}$$

by empty stack.

$Q = \{1, 2\}$, $\Gamma = \{Z_0, a\}$;
$(1, a) \in \delta(1, a, Z_0)$,
$(1, aa) \in \delta(1, a, a)$,
$(2, \epsilon) \in \delta(1, b, a)$,
$(2, \epsilon) \in \delta(2, b, a)$.

The following PDA accepts the language

$$L = \{a^n b^n \mid n \geq 1\}$$

by final state (and also by empty stack).

$Q = \{1, 2, 3\}$, $\Gamma = \{Z_0, A, a\}$, $F = \{3\}$;
$(1, A) \in \delta(1, a, Z_0)$,
$(1, aA) \in \delta(1, a, A)$,
$(1, aa) \in \delta(1, a, a)$,
$(2, \epsilon) \in \delta(1, b, a)$,
$(2, \epsilon) \in \delta(2, b, a)$,
$(3, \epsilon) \in \delta(1, b, A)$,
$(3, \epsilon) \in \delta(2, b, A)$.

DPDA's are defined as follows.

Definition 3.13.5 A PDA

$$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

is a *deterministic PDA* (for short, DPDA), iff the following conditions hold for all $(p, Z) \in Q \times \Gamma$: either

(1) $|\delta(p, a, Z)| = 1$ for all $a \in \Sigma$, and $\delta(p, \epsilon, Z) = \emptyset$, or

(2) $\delta(p, a, Z) = \emptyset$ for all $a \in \Sigma$, and $|\delta(p, \epsilon, Z)| = 1$.

A DPDA *operates in realtime* iff it has no ϵ-transitions.

It turns out that for DPDA's the most general acceptance mode is by final state. Indeed, there are language that can only be accepted deterministically as $T(M)$. The language

$$L = \{a^m b^n \mid m \geq n \geq 1\}$$

is such an example. The problem is that $a^m b$ is a prefix of all strings $a^m b^n$, with $m \geq n \geq 2$.

A language L is a *deterministic context-free language* iff $L = T(M)$ for some DPDA M.

It is easily shown that if $L = N(M)$ (or $L = L(M)$) for some DPDA M, then $L = T(M')$ for some DPDA M' easily constructed from M.

A PDA is *unambiguous* iff for every $w \in \Sigma^*$, there is at most one computation

$$(q_0, w, Z_0) \vdash^* ID_n,$$

where ID_n is an accepting ID.

There are context-free languages that are not accepted by any DPDA. For example, it can be shown that the languages

$$L_1 = \{a^n b^n \mid n \geq 1\} \cup \{a^n b^{2n} \mid n \geq 1\},$$

and

$$L_2 = \{ww^R \mid w \in \{a, b\}^*\},$$

are not accepted by any DPDA.

Also note that unambiguous grammars for these languages can be easily given.

We now show that every context-free language is accepted by a PDA.

3.14 From Context-Free Grammars To PDA's

We show how a PDA can be easily constructed from a context-free grammar. Although simple, the construction is not practical for parsing purposes, since the resulting PDA is horribly nondeterministic.

Given a context-free grammar $G = (V, \Sigma, P, S)$, we define a one-state PDA M as follows:

$Q = \{q_0\}$; $\Gamma = V$; $q_0 = S$; $Z_0 = S$; $F = \emptyset$;

For every rule $(A \to \alpha) \in P$, there is a transition

$$(q_0, \alpha) \in \delta(q_0, \epsilon, A).$$

For every $a \in \Sigma$, there is a transition

$$(q_0, \epsilon) \in \delta(q_0, a, a).$$

The intuition is that a computation of M mimics a leftmost derivation in G. One might say that we have a "pop/expand" PDA.

Lemma 3.14.1 *Given any context-free grammar $G = (V, \Sigma, P, S)$, the PDA M just described accepts $L(G)$ by empty stack, i.e., $L(G) = N(M)$.*

Proof. The following two claims are proved by induction.

Claim 1:

For all $u, v \in \Sigma^*$ and all $\alpha \in NV^* \cup \{\epsilon\}$, if $S \underset{lm}{\overset{*}{\Longrightarrow}} u\alpha$, then

$$(q_0, uv, S) \vdash^* (q_0, v, \alpha).$$

Claim 2:

For all $u, v \in \Sigma^*$ and all $\alpha \in V^*$, if

$$(q_0, uv, S) \vdash^* (q_0, v, \alpha)$$

then $S \underset{lm}{\overset{*}{\Longrightarrow}} u\alpha$. $\square$

We now show how a PDA can be converted to a context-free grammar

3.15 From PDA's To Context-Free Grammars

The construction of a context-free grammar from a PDA is not really difficult, but it is quite messy. The construction is simplified if we first convert a PDA to an equivalent PDA such that for every move $(q, \gamma) \in \delta(p, a, Z)$ (where $a \in \Sigma \cup \{\epsilon\}$), we have $|\gamma| \leq 2$. In some sense, we form a kind of PDA in Chomsky Normal Form.

Lemma 3.15.1 *Given any PDA*

$$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F),$$

another PDA

$$M' = (Q', \Sigma, \Gamma', \delta', q_0', Z_0', F')$$

can be constructed, such that $L(M) = L(M')$ *and the following conditions hold:*

(1) There is a one-to-one correspondence between accepting computations of M *and* M'*;*

(2) If M *has no* ϵ*-moves, then* M' *has no* ϵ*-moves; If* M *is unambiguous, then* M' *is unambiguous;*

(3) For all $p \in Q'$*, all* $a \in \Sigma \cup \{\epsilon\}$*, and all* $Z \in \Gamma'$*, if* $(q, \gamma) \in \delta'(p, a, Z)$*, then* $q \neq q_0'$ *and* $|\gamma| \leq 2$*.*

The crucial point of the construction is that accepting computations of a PDA accepting by empty stack and final state can be decomposed into subcomputations of the form

$$(p, uv, Z\alpha) \vdash^* (q, v, \alpha),$$

where for every intermediate ID (s, w, β), we have $\beta = \gamma\alpha$ for some $\gamma \neq \epsilon$.

The nonterminals of the grammar constructed from the PDA M are triples of the form $[p, Z, q]$ such that

$$(p, u, Z) \vdash^+ (q, \epsilon, \epsilon)$$

for some $u \in \Sigma^*$.

Given a PDA

$$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

satisfying the conditions of lemma 3.15.1, we construct a context-free grammar $G = (V, \Sigma, P, S)$ as follows:

$$V = \{[p, Z, q] \mid p, q \in Q, Z \in \Gamma\} \cup \Sigma \cup \{S\},$$

where S is a new symbol, and the productions are defined as follows: for all $p, q \in Q$, all $a \in \Sigma \cup \{\epsilon\}$, all $X, Y, Z \in \Gamma$, we have:

(1) $S \to \epsilon \in P$, if $q_0 \in F$;

(2) $S \to a \in P$, if $(f, \epsilon) \in \delta(q_0, a, Z_0)$, and $f \in F$;

(3) $S \to a[p, X, f] \in P$, for every $f \in F$, if $(p, X) \in \delta(q_0, a, Z_0)$;

(4) $S \to a[p, X, s][s, Y, f] \in P$, for every $f \in F$, for every $s \in Q$, if $(p, XY) \in \delta(q_0, a, Z_0)$;

(5) $[p, Z, q] \to a \in P$, if $(q, \epsilon) \in \delta(p, a, Z)$ and $p \neq q_0$;

(6) $[p, Z, s] \to a[q, X, s] \in P$, for every $s \in Q$, if $(q, X) \in \delta(p, a, Z)$ and $p \neq q_0$;

(7) $[p, Z, t] \to a[q, X, s][s, Y, t] \in P$, for every $s, t \in Q$, if $(q, XY) \in \delta(p, a, Z)$ and $p \neq q_0$.

Lemma 3.15.2 *Given any PDA*

$$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

satisfying the conditions of lemma 3.15.1, the context-free grammar $G = (V, \Sigma, P, S)$ *constructed as above generates* $L(M)$*, i.e.,* $L(G) = L(M)$*. Furthermore,* G *is unambiguous iff* M *is unambiguous.*

Proof. We have to prove that

$$L(G) = \{w \in \Sigma^+ \mid (q_0, w, Z_0) \vdash^+ (f, \epsilon, \epsilon),\ f \in F\} \cup \{\epsilon \mid q_0 \in F\}.$$

For this, the following claim is proved by induction.

Claim:

For all $p, q \in Q$, all $Z \in \Gamma$, all $k \geq 1$, and all $w \in \Sigma^*$,

$$[p, Z, q] \underset{lm}{\overset{k}{\Longrightarrow}} w \quad \text{iff} \quad (p, w, Z) \vdash^+ (q, \epsilon, \epsilon).$$

Using the claim, it is possible to prove that $L(G) = L(M)$. $\square$

In view of lemmas 3.15.1 and 3.15.2, the family of context-free languages is exactly the family of languages accepted by PDA's. It is harder to give a grammatical characterization of the deterministic context-free languages. One method is to use Knuth $LR(k)$-grammars.

Another characterization can be given in terms of *strict deterministic grammars* due to Harrison and Havel.