

Chap. 6 Pushdown Automata

6.1 Definition of Pushdown Automata

Example 6.2 $L_{ww^R} = \{ww^R \mid w \in (0+1)^*\}$ **Palindromes over $\{0, 1\}$.**

A cfg $P \rightarrow \varepsilon \mid 0 \mid 1 \mid 0P0 \mid 1P1$.

Consider a FA with a stack(= a Pushdown automaton; PDA).

q_0 : Push input symbol onto stack, and stay in q_0 (**gathering mode**) or

Go to state q_1 (**matching mode**) **nondeterministically**(ε -moves)

Guess that it is the **center** of the palrindrome, now.

q_1 : **If** the input symbol is **same** as the top of the stack,

then pop the top of the stack and stay in q_1 .(**matching mode**)

If the input symbol is **not** same as the top of the stack,

then reject.

If **no more input symbol** and **empty stack** in state q_1 ,

then go to the final state q_2 and **accept**, **else** reject.

6.1.2 The Formal Definition of Pushdown Automata

A pushdown automaton (PDA) $P = (Q, T, \Gamma, \delta, q_0, Z_0, F)$ is

1. Q is a **finite** set of state alphabet,
2. T is a **finite** set of input alphabet,
3. Γ is a **finite** stack alphabet,
4. δ is a **transition function**. $\delta: Q \times (T \cup \{\varepsilon\}) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$.

If $(p, \gamma) \in \delta(q, a, X)$ for $q, p \in Q, a \in T \cup \{\varepsilon\}, \gamma \in \Gamma^*$.

pop stack top X and **push** stack string $\gamma \in \Gamma^*$ onto stack.

- i) $(p, \varepsilon) \in \delta(q, a, X)$ **pop** X
- ii) $(p, X) \in \delta(q, a, X)$ no stack change
- iii) $(p, YX) \in \delta(q, a, X)$ **push** Y

5. $q_0 \in Q$ is an **initial state**,
6. $Z_0 \in \Gamma^*$ is an **initial stack content**,
7. $F \subseteq Q$ is a set of **final states**.

Example 6.2 $P_{\text{wwwR}} = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, Z_0\}, \delta, q_0, Z_0, \{q_2\})$

1. $\forall X \in \Gamma, \forall a \in T: \delta(q_0, a, X) = \{(q_0, aX)\}$. *gathering mode(push)*

State q_0 : If see 0, then push 0 and if see 1, then push 1; stay in q_0 .

2. $\delta(q_0, \varepsilon, X) = \{(q_1, X)\}$. *goto matching mode*

Go to the state q_1 on ε input(*nondeterministic*): guess center of Pal.

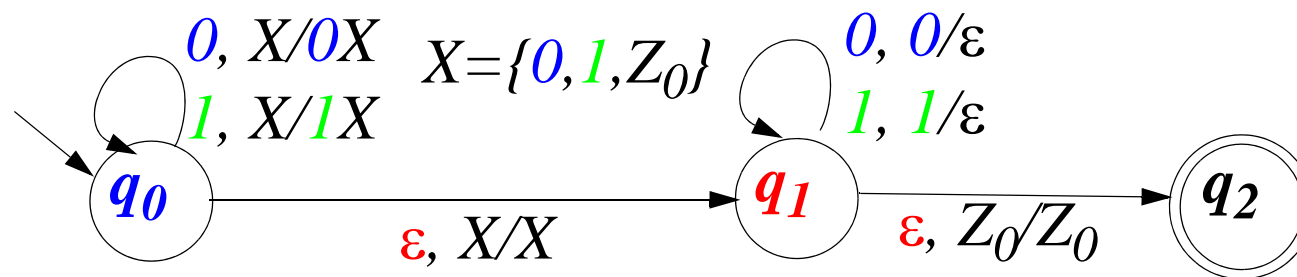
3. $\forall a \in T: \delta(q_1, a, a) = \{(q_1, \varepsilon)\}$. *matching mode(pop)*

State q_1 : *match* input symbol against stack top symbol and *pop* it.

else *error!* $\delta(q_1, 0, 1) = \delta(q_1, 1, 0) = \delta(q_1, 0, Z_0) = \delta(q_1, 1, Z_0) = \emptyset$.

4. $\delta(q_1, \varepsilon, Z_0) = \{(q_2, Z_0)\}$. *end of matching! Accept!*

$L(P_{\text{wwwR}}) = L_{\text{wwwR}}$ but P_{wwwR} is *not deterministic!* Ex. 6.3 Fig. 6.2 in p 230.



Configuration (Instantaneous description) of PDA

(current state, remained input string, current stack contents)

$$(q, x, \gamma) \in Q \times T^* \times \Gamma^*.$$

$$\vdash_P \subseteq (Q \times T^* \times \Gamma^*) \times (Q \times T^* \times \Gamma^*)$$

$$(q, \mathbf{ax} X\beta) \vdash_P (p, \mathbf{x}, \gamma\beta), \text{ if } (p, \gamma) \in \delta(q, \mathbf{a}, X)$$

$$(q, \mathbf{x}, X\beta) \vdash_P (p, \mathbf{x}, \gamma\beta), \text{ if } (p, \gamma) \in \delta(q, \varepsilon, X)$$

We may use \vdash instead of \vdash_P if P is understood.

\vdash is a binary relation on $(Q \times T^* \times \Gamma^*)$.

\vdash^* is a **reflexive and transitive closure** of \vdash .

Recursive definition of \vdash^* .

$$\forall I, J, K \in Q \times T^* \times \Gamma^*, I \vdash_B^* I.$$

$$\text{If } I \vdash J \text{ and } J \vdash^* K, I \vdash_R^* K.$$

Let $P = (Q, T, \Gamma, \delta, q_0, Z_0, F)$ is a PDA. Then

Thm. 6.5 If $(q, x, \alpha) \vdash^* (p, y, \beta)$ for $q, p \in Q$, $x, y \in T^*$ and $\alpha, \beta \in \Gamma^*$.

Then $(q, xw, \alpha\gamma) \vdash^* (p, yw, \beta\gamma)$ for any $w \in T^*$ and $\gamma \in \Gamma^*$.

Adding lookahead input strings and lookback stack strings.

Thm. 6.6 If $(q, xw, \alpha) \vdash^* (p, yw, \beta)$ for $q, p \in Q$, $x, y, w \in T^*$ and $\alpha, \beta \in \Gamma^*$.

Then $(q, x, \alpha) \vdash^* (p, y, \beta)$.

Remooving lookahead input strings.

6.2 The language of a PDA

6.2.1 Acceptance by Final State

$$L(P) = \{w \in T^* \mid (q_0, w, Z_0) \vdash^* (f, \varepsilon, \alpha), f \in F\}$$

6.2.2 Acceptance by Null Stack

$$N(P) = \{w \in T^* \mid (q_0, w, Z_0) \vdash^* (f, \varepsilon, \varepsilon)\}$$

6.2.3 From Empty Stack to Final State

Thm. 6.9 If $L = N(P_N)$ for some PDA $P_N = (Q_N, T, \Gamma_N, \delta_N, q_0^N, Z_0^N, \emptyset)$.

Then there is a PDA P_F such that $L = L(P_F)$.

$$P_F = (Q_N \cup \{q_0^F, q_F\}, T, \Gamma_N \cup \{Z_0^F\}, \delta_F, q_0^F, Z_0^F, \{q_F\})$$

where $q_0^F, q_F \notin Q_N, Z_0^F \notin \Gamma_N$.

- δ_F :
1. $\delta_F(q_0^F, \varepsilon, Z_0^F) = \{(q_0^N, Z_0^N Z_0^F)\}$. push old stack **bottom** Z_0^N .
 2. $\delta_F \supseteq \delta_N$, simulate P_N with δ_N .
 3. $\forall q \in Q_N, \delta_F(q, \varepsilon, Z_0^F) = \{(q_F, \varepsilon \text{ (or } Z_0^F \text{ or any } \alpha \in \Gamma_N^*))\}$.

If stack is **empty** (Z_0^F : stack top), go to the final **final** state q_F

See **Fig. 6.4** in p. 237.

$$p_0 = q_0^F, X_0 = Z_0^F, Z_0 = Z_0^N, q_0/\varepsilon = q_0^N/\text{any}.$$

6.2.4 From Final State to Empty Stack

Thm. 6.11 If $L = L(P_F)$ for some PDA $P_F = (Q_F, T, \Gamma_F, \delta_F, q_0^F, Z_0^F, F)$.

Then there is a PDA P_N such that $L = N(P_N)$.

$$P_N = (Q_F \cup \{q_0^N, q_E\}, T, \Gamma_N, \delta_N, q_0^E, Z_0^N, \emptyset)$$

where $q_0^E, q_E \notin Q_F, Z_0^N \notin \Gamma_F$ and $\Gamma_N = \Gamma_F \cup \{Z_0^N\}$.

- δ_N :
1. $\delta_N(q_0^E, \varepsilon, Z_0^N) = \{(q_0, Z_0^F Z_0^N)\}$ push old stack **bottom** Z_0^F .
 2. $\delta_N \supseteq \delta_F$ simulate P_F with δ_F
 3. $\forall f \in F, \forall Z \in \Gamma_N, \delta_N(f, \varepsilon, Z) \supseteq \{(q_E, \varepsilon)\}$.

If **final** state, **pop** a stack symbol and go to the **empty** state q_E .

4. $\forall Z \in \Gamma_N, \delta_N(q_E, \varepsilon, Z) = \{(q_E, \varepsilon)\}$

Pop all of the stack symbols in the **empty** state q_E .

See **Fig. 6.7** in p. 240 $p_0 = q_0^E, X_0 = Z_0^N, Z_0 = Z_0^F, q_0 = q_0^F$, and
 ... any/ $\varepsilon = Z/\varepsilon$ in the state $p = q_E$.

6.3 Equivalence of PDA's and CFG's

6.3.1 From Grammars to Pushdown Automata

Theorem 6.13 If $G = (N, T, P, S)$ is a cfg. Then \exists PDA P . \exists . $L(G) = N(P)$.

Construct $P = (\{q\}, T, N \cup T, \delta, q, S, \emptyset)$ *guess and verify parser*

$\forall A \in N, \delta(q, \varepsilon, A) = \{(q, \alpha) \mid A \rightarrow \alpha \in P\}$ *guess A as α ($A \rightarrow \alpha \in P$).*

$\forall a \in \Sigma, \delta(q, a, a) = \{(q, a) \mid a \in T\}$ *verify $a \in \Sigma$.*

Proof $A \Rightarrow_{lm}^* x\alpha$ if and only if $(q, x, A) \vdash^* (q, \varepsilon, \alpha), x \in \Sigma^*, \alpha \in (N \cup \Sigma)^*$.

(If) If $(q, x, A) \vdash^i (q, \varepsilon, \alpha)$, then $A \Rightarrow_{lm}^* x\alpha$ for $i \geq 0$.

basis $i = 0$: $x = \varepsilon$, and $A = \alpha$. $\therefore (q, \varepsilon, A) \vdash^0 (q, \varepsilon, A)$. $\therefore A \Rightarrow_{lm}^* A$.

induction Let $i \geq 1$, and consider the next-to-last step.

i) $(q, x, A) \vdash^{i-1} (q, \varepsilon, B\gamma) \vdash_{\text{guess}}^{B \rightarrow \beta} (q, \varepsilon, \beta\gamma) = (q, \varepsilon, \alpha)$

$\therefore A \Rightarrow_{lm}^* xB\gamma$ by IH and $B \rightarrow \beta \in P$ by construction of δ_{guess} .

$\therefore A \Rightarrow_{lm}^* xB\gamma \Rightarrow_{lm} x\beta\gamma = x\alpha$.

$$\begin{aligned}
ii) (q, x, A) = (q, ya, A) &\vdash^{i-1} (q, a, a\alpha) \vdash_{\text{verify}} (q, \varepsilon, \alpha) \\
(q, y, A) &\vdash^{i-1} (q, \varepsilon, a\alpha) \quad (\text{Thm 6.5; } (q, \underline{\varepsilon}, a\alpha)) \\
\therefore A &\Rightarrow_{lm}^* ya\alpha = x\alpha \text{ by IH}
\end{aligned}$$

(Only if) If $A \Rightarrow_{lm}^i x\alpha$, then $(q, x, A) \vdash^* (q, \varepsilon, \alpha)$ for $i \geq 0$.

basis $i = 0$, $A \Rightarrow_{lm}^0 A$. $x = \varepsilon$ and $A = \alpha$. $(q, \varepsilon, A) \vdash^0 (q, \varepsilon, A)$.

induction Let $i \geq 1$, and consider the next-to-last step.

$$A \Rightarrow_{lm}^{i-1} yB\gamma \Rightarrow_{lm} y\beta\gamma = yy'\gamma'\gamma = xA \text{ where } \beta = y'\gamma', y' \in \Sigma^*, \gamma' \in (N \cup \Sigma)^*.$$

$$\begin{aligned}
(q, y, A) &\vdash^* (q, \varepsilon, B\gamma) \text{ by IH. } \therefore (q, yy', A) \vdash^* (q, y', B\gamma) \text{ (by T.6.5)} \\
&\vdash_G^{B \rightarrow \beta} (q, y', \beta\gamma) = (q, y', y'\gamma'\gamma) \vdash_{\sqrt{|y'|}} (q, \varepsilon, \gamma'\gamma) = (q, \varepsilon, \alpha)
\end{aligned}$$

$$\therefore A \Rightarrow_{lm}^* x\alpha \text{ if and only if } (q, x, A) \vdash^* (q, \varepsilon, \alpha).$$

If $A = S$, $\alpha = \varepsilon$, then $S \Rightarrow_{lm}^* x$ if and only if $(q, x, S) \vdash^* (q, \varepsilon, \varepsilon)$

$$\therefore L(G) = N(P).$$

6.3.2 From PDA's to Grammars

Theorem 6.14 If a PDA $P = (Q, T, \Gamma, \delta, q_0, Z_0, \emptyset)$.

Then there is a CFG G such that $L(G) = N(P)$. **null stack**

proof $G = (Q \times \Gamma \times Q \cup \{S\}, T, P, S)$

$P = \{S \rightarrow [q_0 Z_0 q] \mid \forall q \in Q\}$ use $[q_0 Z_0 q]$ instead of (q_0, Z_0, q)

$\cup \{[q A p_m] \rightarrow a [p Y_1 p_1][p_1 Y_2 p_2] \dots [p_{m-1} Y_m p_m] \mid$

$(p, Y_1 Y_2 \dots Y_m) \in \delta(q, a, A), a \in T \cup \{\varepsilon\}, \forall q \in Q, 1 \leq \forall i \leq m: p_i \in Q, Y_i \in \Gamma\}$

$(\text{if } m = 0, [q A p] \rightarrow a \in P, a \in T \cup \{\varepsilon\})$

To prove that $[q A p] \Rightarrow_{lm}^* x \in T^*$, if and only if

PDA Configurations (ID's) $(q, x, A) \vdash^* (p, \varepsilon, \varepsilon) \subseteq Q \times T^* \times \Gamma^*$.

Nonterminal $[q A p]$ derives terminal string x if and only if

x causes PDA P to pop A from stack

starting in the state q and ending in the state p .

1) If $(q, x, A) \vdash^i (p, \varepsilon, \varepsilon)$, then $[qAp] \Rightarrow_{lm}^* x$ for $i \geq 1$.

basis $i = 1$, $(q, x, A) \vdash (p, \varepsilon, \varepsilon)$. $\therefore (p, \varepsilon) \in \delta(q, x, A)$, $x \in \Sigma \cup \{\varepsilon\}$.

$\therefore [qAp] \rightarrow x \in P$ where $x \in T \cup \{\varepsilon\}$. $\therefore [qAp] \Rightarrow_{lm}^* x$.

induction $(q, x, A) = (q, ay, A) \vdash (p_1, y, Y_1 \dots Y_m) \vdash^{i-1} (p, \varepsilon, \varepsilon)$.

$\therefore \exists p_2, \dots, p_m$, $p \in q$ and assume $y = y_1 \dots y_m \in T^*$.

$(p_1, y_1 \dots y_m, Y_1 \dots Y_m) \vdash^* (p_2, y_2 \dots y_m, Y_2 \dots Y_m) \vdash^* \dots (p_m, y_m, Y_m) \vdash (p, \varepsilon, \varepsilon)$.

$1 \leq \forall i \leq m$, $(p_i, y_i, Y_i) \vdash^* (p_{i+1}, \varepsilon, \varepsilon)$. (Thm 6.5 and y_i depends on Y_i only)

$\therefore [p_i Y_i p_{i+1}] \Rightarrow_{lm}^* y_i$ by IH.

$\therefore [p_1 Y_1 p_2][p_2 Y_2 p_3] \dots [p_m Y_m p] \Rightarrow_{lm}^* y_1 y_2 \dots y_m = y$

Since $(p_1, Y_1 \dots Y_m) \in \delta(q, a, A)$, $(q, ay, A) \vdash (p_1, y, Y_1 \dots Y_m)$.

$\therefore \exists [qAp] \rightarrow a [p_1 Y_1 p_2][p_2 Y_2 p_3] \dots [p_m Y_m p] \in P$. ($\forall p_i \in Q$)

$\therefore [qAp] \Rightarrow_{lm} a [p_1 Y_1 p_2][p_2 Y_2 p_3] \dots [p_m Y_m p] \Rightarrow_{lm}^* ay = x$.

2) If $[qAp] \Rightarrow_{lm}^i x$, then $(q, x, A) \vdash^* (p, \varepsilon, \varepsilon)$ for $i \geq 1$.

basis $i = 1$, $[qAp] \rightarrow x \in P$, $(p, \varepsilon) \in \delta(q, x, A)$ where $x \in T \cup \{\varepsilon\}$.

induction $[qAp] \Rightarrow_{lm} a [p_1Y_1p_2][p_2Y_2p_3] \dots [p_mY_mp] \Rightarrow_{lm}^{i-1} x \in \Sigma^*$.

$x = ay_1 \dots y_m$ where $1 \leq \forall i \leq m$, $[p_iY_ip_{i+1}] \Rightarrow_{lm}^i y_i$ where $p_{m+1} = p$.

$\therefore (p_i, y_i, Y_i) \vdash^* (p_{i+1}, \varepsilon, \varepsilon)$ by IH.

Since $[qAp] \rightarrow a [p_1Y_1p_2][p_2Y_2p_3] \dots [p_mY_mp] \in P$,

$a [p_1Y_1p_2][p_2Y_2p_3] \dots [p_mY_mp] \in \delta(q, a, A)$ where $a \in T \cup \{\varepsilon\}$.

$\therefore (q, ay_1 \dots y_m, A) \vdash (p_1, y_1 \dots y_m, Y_1 \dots Y_m) \vdash^* \dots \vdash^* (p, \varepsilon, \varepsilon)$.

$[qAp] \Rightarrow_{lm}^* x$, $\forall q, p \in q$ if and only if $(q, x, A) \vdash^* (p, \varepsilon, \varepsilon)$.

$\therefore [q_0Z_0p] \Rightarrow_{lm}^* x$, $\forall q, p \in q$ if and only if $(q_0, x, Z_0) \vdash^* (p, \varepsilon, \varepsilon)$.

$S \Rightarrow_{lm} [q_0Z_0p] \Rightarrow_{lm}^* x$, $\forall q, p \in q$ if and only if $(q_0, x, Z_0) \vdash^* (p, \varepsilon, \varepsilon)$.

6.4 Deterministic Pushdown Automata

6.4.1 Definition of a Deterministic PDA

A PDA $P = (Q, T, \Gamma, \delta, q_0, Z_0, F)$ with

$\delta: Q \times (T \cup \{\varepsilon\}) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$ is **deterministic**, if

1. $\forall q \in Q, \forall a \in T \cup \{\varepsilon\}: |\delta(q, a, X)| \leq 1$ or

$\delta: q \times (T \cup \{\varepsilon\}) \times \Gamma \rightarrow Q \times \Gamma^*$.

2. If $\exists \delta(q, a, X)$, then $\delta(q, \varepsilon, X) = \emptyset$ or

If $\exists (q, \varepsilon, X)$, then $\forall a \in \Sigma: \delta(q, a, X) = \emptyset$.

Example 6.16 Palindromes over $\{0, 1\}$ with center marker c .

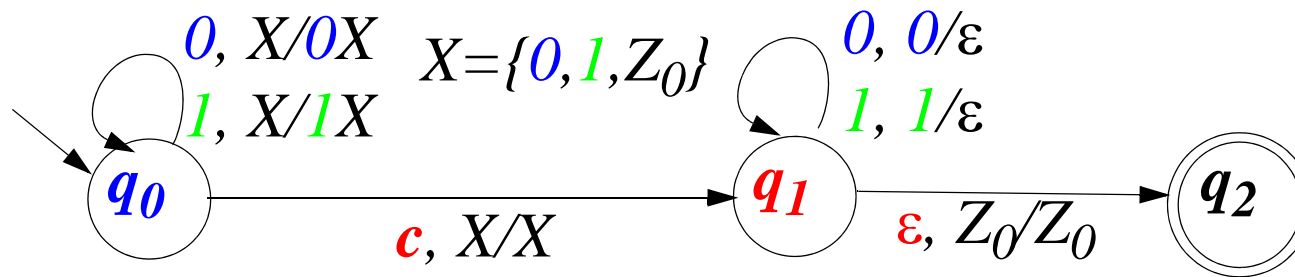
$$L_{wcwR} = \{w\mathbf{c}w^R \mid w \in (0+1)^*\}$$

$$P_{wcwR} = (\{\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2\}, \{0, 1, c\}, \{0, 1, Z_0\}, \delta, \mathbf{q}_0, Z_0, \{\mathbf{q}_2\})$$

$$\cdot \exists. L_{wcwR} = L(P_{wcwR}).$$

$P_{WCWR} = (Q, T, \Gamma, \delta, q_0, Z_0, F)$ is a **deterministic PDA!**

1. $\forall X \in \Gamma, \forall a \in \{0, 1\}: \delta(q_0, a, X) = \{(q_0, aX)\}$. *gathering mode(push)*
2. $\delta(q_0, c, X) = \{(q_1, X)\}$. *goto matching mode*
3. $\forall a \in \{0, 1\}: \delta(q_1, a, a) = \{(q_1, \varepsilon)\}$. *matching mode(pop)*
4. $\delta(q_1, \varepsilon, Z_0) = \{(q_2, \varepsilon)\}$. *end of matching!*



Stronger version of deterministic PDA

$$\delta: Q \times T \times \Gamma \rightarrow Q \times \Gamma^*.$$

6.4.2 Regular Languages and Deterministic PDA'

Theorem 6.17 *If L is regular, then $L = L(P)$ for some PDA.*

proof Let $A = (Q, T, \delta_A, q_0, F)$ is a DFA $\exists. L = L(A)$. Then

$P = (Q, T, \{Z_0\}, \delta_P, q_0, Z_0, F)$ with

$\delta_P(q, a, Z_0) = \{(p, Z_0) \mid \delta_A(q, a) = p\}$ is a **deterministic PDA**.

DFA is a DPDA and FA is a PDA!

6.4.3 DPDA's and Context-free Languages

P_{wCwR} is a DPDA. But L_{wCwR} is not **regular** but **context-free**.

$$L_{wCwR} = L(P_{wCwR}).$$

L_{wwR} is not **regular** but **context-free**. But there is no DPDA P

$$\exists. L_{wwR} = L(P)$$

Regular Languages $\subset L(\text{DPDA}) \subset$ **Context-free Languages**

6.4.4 DPDA's and Ambiguous Grammars

Theorem 6.20 *If $L = L(P)$ for some DPDA A , Then L has an **unambiguous** context-free grammar.*

proof *If a PDA P is DPDA, then the CFG in Ex. 6.16 is unambiguous.*

Theorem 6.21 *If $L = L(P)$ for some DPDA A , Then L has an ambiguous grammar.*

proof *end marker($\$$) vs ϵ . **But not important!** ($\Sigma \cup \{\$$ } vs $\Sigma \cup \{\epsilon\}$)*

Context-free Languages = Context-free grammars = Pushdown Automata
 \subset **Regular Languages = Regular Grammars = Finite Automata**
= Regular Expressions

