A pushdown automaton (PDA) is essentially an $\epsilon$-NFA with a stack.

On a transition the PDA:

1. Consumes an input symbol. or $\epsilon$
2. Goes to a new state (or stays in the old).
3. Replaces the top of the stack by any string (does nothing, pops the stack, or pushes a string onto the stack)
Example: Let’s consider

\[ L_{wwr} = \{ww^R : w \in \{0, 1\}^*\}, \]

with “grammar” \( P \rightarrow 0P0, \ P \rightarrow 1P1, \ P \rightarrow \epsilon. \)

A PDA for \( L_{wwr} \) has three states, and operates as follows:

1. Guess that you are reading \( w \). Stay in state 0, and push the input symbol onto the stack.

2. Guess that you’re in the middle of \( ww^R \). Go spontaneously to state 1.

3. You’re now reading the head of \( w^R \). Compare it to the top of the stack. If they match, pop the stack, and remain in state 1. If they don’t match, go to sleep.

4. If the stack is empty, go to state 2 and accept.
The PDA for $L_{wwr}$ as a transition diagram:

- $0, Z_0 / 0 Z_0$
- $1, Z_0 / 1 Z_0$
- $0, 0 / 0 0$
- $0, 1 / 0 1$
- $1, 0 / 1 0$
- $1, 1 / 1 1$

- $0, 0 / ε$
- $1, 1 / ε$

Start

$q_0$

$ε, Z_0 / Z_0$
$ε, 0 / 0$
$ε, 1 / 1$

$q_1$

$ε, Z_0 / Z_0$

$q_2$
Actions of the Example PDA

q_0

0 0 1 1 0 0

Z_0
Actions of the Example PDA
Actions of the **Example** PDA
Actions of the Example PDA
Actions of the Example PDA
Actions of the Example PDA
Actions of the Example PDA
Actions of the Example PDA
Actions of the Example PDA
A PDA is a seven-tuple:

\[ P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F), \]

where

- \( Q \) is a finite set of states,
- \( \Sigma \) is a finite input alphabet,
- \( \Gamma \) is a finite stack alphabet,
- \( \delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow 2^{Q \times \Gamma^*} \) is the transition function,
- \( q_0 \) is the start state,
- \( Z_0 \in \Gamma \) is the start symbol for the stack, and
- \( F \subseteq Q \) is the set of accepting states.
Example: The PDA

is actually the seven-tuple

\[ P = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, Z_0\}, \delta, q_0, Z_0, \{q_2\}) \]

where \( \delta \) is given by the following table (set brackets missing):

<table>
<thead>
<tr>
<th></th>
<th>0, Z_0</th>
<th>1, Z_0</th>
<th>0,0</th>
<th>0,1</th>
<th>1,0</th>
<th>1,1</th>
<th>( \epsilon, Z_0 )</th>
<th>( \epsilon, 0 )</th>
<th>( \epsilon, 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_0 )</td>
<td>q_0, 0Z_0</td>
<td>q_0, 1Z_0</td>
<td>q_0, 00</td>
<td>q_0, 01</td>
<td>q_0, 10</td>
<td>q_0, 11</td>
<td>q_1, Z_0</td>
<td>q_1, 0</td>
<td>q_1, 1</td>
</tr>
<tr>
<td>( q_1 )</td>
<td>( q_1, \epsilon )</td>
<td>( q_1, \epsilon )</td>
<td>( q_1, \epsilon )</td>
<td>( q_2, Z_0 )</td>
<td>( q_1, 0 )</td>
<td>( q_1, 1 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( * q_2 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

185
A PDA goes from configuration to configuration when consuming input.

To reason about PDA computation, we use *instantaneous descriptions* of the PDA. An ID is a triple

\[(q, w, \gamma)\]

where \(q\) is the state, \(w\) the remaining input, and \(\gamma\) the stack contents.

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

\[
\forall w \in \Sigma^*, \beta \in \Gamma^* : \\
(p, \alpha) \in \delta(q, a, X) \Rightarrow (q, aw, X\beta) \vdash (p, w, \alpha\beta). \\
\text{yield}
\]

We define \(\vdash^*\) to be the reflexive-transitive closure of \(\vdash\).
Example: On input 1111 the PDA

0, Z₀ / 0 Z₀
1, Z₀ / 1 Z₀
0, 0 / 0 0
0, 1 / 0 1
1, 0 / 1 0
1, 1 / 1 1
0, 0 / ε
1, 1 / ε

has the following computation sequences:
The following properties hold:

1. If an ID sequence is a legal computation for a PDA, then so is the sequence obtained by adding an additional string at the end of component number two.

2. If an ID sequence is a legal computation for a PDA, then so is the sequence obtained by adding an additional string at the bottom of component number three.

3. If an ID sequence is a legal computation for a PDA, and some tail of the input is not consumed, then removing this tail from all ID’s result in a legal computation sequence.
**Theorem 6.5:** \( \forall w \in \Sigma^*, \gamma \in \Gamma^* : \)

\[(q, x, \alpha) \vdash^* (p, y, \beta) \Rightarrow (q, xw, \alpha\gamma) \vdash^* (p, yw, \beta\gamma).\]

**Proof:** Induction on the length of the sequence to the left.

Note: If \( \gamma = \epsilon \) we have property 1, and if \( w = \epsilon \) we have property 2.

Note2: The reverse of the theorem is false.

For property 3 we have

**Theorem 6.6:**

\[(q, xw, \alpha) \vdash^* (p, yw, \beta) \Rightarrow (q, x, \alpha) \vdash^* (p, y, \beta).\]
Acceptance by final state

Let \( P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F) \) be a PDA. The language accepted by \( P \) by final state is

\[
L(P) = \{ w : (q_0, w, Z_0) \vdash^* (q, \epsilon, \alpha), q \in F \}.
\]

Example: The PDA on slide 183 accepts exactly \( L_{wwr} \).

Let \( P \) be the machine. We prove that \( L(P) = L_{wwr} \).

(\( \supseteq \)-direction.) Let \( x \in L_{wwr} \). Then \( x = ww^R \), and the following is a legal computation sequence

\[
(q_0, ww^R, Z_0) \vdash^* (q_0, w^R, w^R Z_0) \vdash (q_1, w^R, w^R Z_0) \vdash^* (q_1, \epsilon, Z_0) \vdash (q_2, \epsilon, Z_0).
\]
Observe that the only way the PDA can enter $q_2$ is if it is in state $q_1$ with top stack symbol $= z_0$

Thus it is sufficient to show that if $(q_0, x, Z_0) \vdash^* (q_1, \epsilon, Z_0)$ then $x = w w^R$, for some word $w$.

We’ll show by induction on $|x|$ that

$$(q_0, x, \alpha) \vdash^* (q_1, \epsilon, \alpha) \Rightarrow x = w w^R.$$ 

**Basis:** If $x = \epsilon$ then $x$ is a palindrome.

**Induction:** Suppose $x = a_1 a_2 \cdots a_n$, where $n > 0$, and the IH holds for shorter strings.

There are two moves for the PDA from ID $(q_0, x, \alpha)$:
Move 1: The spontaneous \((q_0, x, \alpha) \vdash (q_1, x, \alpha)\). Now \((q_1, x, \alpha) \vdash^* (q_1, \epsilon, \beta)\) implies that \(|\beta| < |\alpha|\), which implies \(\beta \neq \alpha\).

Move 2: Loop and push \((q_0, a_1a_2 \cdots a_n, \alpha) \vdash (q_0, a_2 \cdots a_n, a_1\alpha)\).

In this case there is a sequence

\[(q_0, a_1a_2 \cdots a_n, \alpha) \vdash (q_0, a_2 \cdots a_n, a_1\alpha) \vdash \cdots \vdash (q_1, a_n, a_1\alpha) \vdash (q_1, \epsilon, \alpha)\]

Thus \(a_1 = a_n\) and

\[(q_0, a_2 \cdots a_n, a_1\alpha) \vdash^* (q_1, a_n, a_1\alpha)\].

By Theorem 6.6 we can remove \(a_n\). Therefore

\[(q_0, a_2 \cdots a_{n-1}, a_1\alpha) \vdash^* (q_1, \epsilon, a_1\alpha)\].

Then, by the IH \(a_2 \cdots a_{n-1} = yy^R\). Then \(x = a_1yy^R a_n\) is a palindrome.
Let $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F')$ be a PDA. The language accepted by $P$ by empty stack is

$$N(P) = \{w : (q_0, w, Z_0) \vdash^* (q, \epsilon, \epsilon)\}.$$ 

Note: $q$ can be any state.

Question: How to modify the palindrome-PDA to accept by empty stack? two ways to do it!

Give a final-state PDA for balanced brackets (or Dyck language): $B \rightarrow BB \mid (B) \mid \epsilon$

$L_2 = \{0^m 1^n 2^p \mid m,n,p \geq 0, m+n = p\}$

Acceptance by Empty Stack
Theorem 6.9: If $L = N(P_N)$ for some PDA $P_N = (Q, \Sigma, \Gamma, \delta_N, q_0, Z_0)$, then $\exists$ PDA $P_F$, such that $L = L(P_F)$.

Proof: Let

$P_F = (Q \cup \{p_0, p_f\}, \Sigma, \Gamma \cup \{X_0\}, \delta_F, p_0, X_0, \{p_f\})$

where $\delta_F(p_0, \epsilon, X_0) = \{(q_0, Z_0X_0)\}$, and for all $q \in Q, a \in \Sigma \cup \{\epsilon\}, Y \in \Gamma : \delta_F(q, a, Y) = \delta_N(q, a, Y)$, and in addition $(p_f, \epsilon) \in \delta_F(q, \epsilon, X_0)$.

![Diagram of PDA transition system](image.png)
We have to show that $L(P_F) = N(P_N)$.

($\supseteq$ direction.) Let $w \in N(P_N)$. Then 

$$(q_0, w, Z_0) \vdash^*_N (q, \epsilon, \epsilon),$$

for some $q$. From Theorem 6.5 we get 

$$(q_0, w, Z_0X_0) \vdash^*_N (q, \epsilon, X_0).$$

Since $\delta_N \subset \delta_F$ we have 

$$(q_0, w, Z_0X_0) \vdash^*_F (q, \epsilon, X_0).$$

We conclude that 

$$(p_0, w, X_0) \vdash_F (q_0, w, Z_0X_0) \vdash^*_F (q, \epsilon, X_0) \vdash_F (p_f, \epsilon, \epsilon).$$

($\subseteq$ direction.) By inspecting the diagram.
Let’s design $P_N$ for for catching errors in strings meant to be in the \textit{if-else-grammar} $G$

\[ S \rightarrow \epsilon | SS | iS | iSe. \]

Here e.g. $\{ieie, iie, ie\} \subseteq L(G)$ and e.g. $\{ei, ieei\} \cap L(G) = \emptyset$.

The diagram for $P_N$ is

Formally,

\[ P_N = (\{q\}, \{i, e\}, \{Z\}, \delta_N, q, Z), \]

where $\delta_N(q, i, Z) = \{(q, ZZ)\}$,

and $\delta_N(q, e, Z) = \{(q, \epsilon)\}$.

Question: Does one state suffice for empty-stack PDAs?
From $P_N$ we can construct

$$P_F = (\{p, q, r\}, \{i, e\}, \{Z, X_0\}, \delta_F, p, X_0, \{r\})$$

where

$$\delta_F(p, \epsilon, X_0) = \{(q, Z X_0)\},$$
$$\delta_F(q, i, Z) = \delta_N(q, i, Z) = \{(q, ZZ)\},$$
$$\delta_F(q, e, Z) = \delta_N(q, e, Z) = \{(q, \epsilon)\},$$

and

$$\delta_F(q, \epsilon, X_0) = \{(r, \epsilon)\}$$

The diagram for $P_F$ is
**Theorem 6.11:** Let \( L = L(P_F) \), for some PDA \( P_F = (Q, \Sigma, \Gamma, \delta_F, q_0, Z_0, F) \). Then \( \exists \) PDA \( P_n \), such that \( L = N(P_N) \).

**Proof:** Let

\[
P_N = (Q \cup \{p_0, p\}, \Sigma, \Gamma \cup \{X_0\}, \delta_N, p_0, X_0)
\]

where \( \delta_N(p_0, \epsilon, X_0) = \{(q_0, Z_0X_0)\} \), \( \delta_N(p, \epsilon, Y) = \{(p, \epsilon)\} \), for \( Y \in \Gamma \cup \{X_0\} \), and for all \( q \in Q, a \in \Sigma \cup \{\epsilon\}, Y \in \Gamma : \delta_N(q, a, Y) = \delta_F(q, a, Y) \), and in addition \( \forall q \in F, \text{ and } Y \in \Gamma \cup \{X_0\} : (p, \epsilon) \in \delta_N(q, \epsilon, Y) \).
We have to show that $N(P_N) = L(P_F)$.

($\subseteq$-direction.) By inspecting the diagram.

($\supseteq$-direction.) Let $w \in L(P_F)$. Then

$$(q_0, w, Z_0) \vdash_F^* (q, \epsilon, \alpha),$$

for some $q \in F, \alpha \in \Gamma^*$. Since $\delta_F \subseteq \delta_N$, and Theorem 6.5 says that $X_0$ can be slid under the stack, we get

$$(q_0, w, Z_0X_0) \vdash_N^* (q, \epsilon, \alpha X_0).$$

The $P_N$ can compute:

$$(p_0, w, X_0) \vdash_N (q_0, w, Z_0X_0) \vdash_N^* (q, \epsilon, \alpha X_0) \vdash_N^* (p, \epsilon, \epsilon).$$
A language is 

*generated by a CFG*

if and only if it is 

*accepted by a PDA by empty stack*

if and only if it is 

*accepted by a PDA by final state*

We already know how to go between null stack and final state.
From CFG's to PDA's

Given $G$, we construct a PDA that simulates $\Rightarrow_{lm}^*$. We write left-sentential forms as $xA\alpha$ where $A$ is the leftmost variable in the form. For instance,

\[
\begin{array}{c}
(a + E) \\
x \\
A \\
\alpha
\end{array}
\]

Let $xA\alpha \Rightarrow x\beta\alpha$. This corresponds to the PDA first having consumed $x$ and having $A\alpha$ on the stack, and then on $\epsilon$ it pops $A$ and pushes $\beta$.

More formally, let $y$, s.t. $w = xy$. Then the PDA goes non-deterministically from configuration $(q, y, A\alpha)$ to configuration $(q, y, \beta\alpha)$. 

202
At \((q, y, \beta \alpha)\) the PDA behaves as before, unless there are terminals in the prefix of \(\beta\). In that case, the PDA pops them, provided it can consume matching input.

If all guesses are right, the PDA ends up with empty stack and input.

Formally, let \(G = (V, T, Q, S)\) be a CFG. Define \(P_G\) as

\[
(\{q\}, T, V \cup T, \delta, q, S),
\]

where

\[
\delta(q, \epsilon, A) = \{(q, \beta) : A \to \beta \in Q\},
\]

for \(A \in V\), and

\[
\delta(q, a, a) = \{(q, \epsilon)\},
\]

for \(a \in T\).

Example: On blackboard in class.

\[
S \to \epsilon, S/0S0 | 1S1 | SS | \epsilon
\]
Theorem 6.13: \( N(P_G) = L(G) \).

Proof:

\( \supseteq \)-direction.) Let \( w \in L(G) \). Then

\[ S = \gamma_1 \Rightarrow_{l_m} \gamma_2 \Rightarrow_{l_m} \cdots \Rightarrow_{l_m} \gamma_n = w \]

Let \( \gamma_i = x_i\alpha_i \). We show by induction on \( i \) that

where \( x_i \) is a string of terminals and \( \alpha_i \) begins with a variable

\[
(q, w, S) \vdash^* (q, y_i, \alpha_i),
\]

where \( w = x_i y_i \).
**Basis:** For $i = 1$, $\gamma_1 = S$. Thus $x_1 = \epsilon$, and $y_1 = w$. Clearly $(q, w, S) \vdash^* (q, w, S)$.

**Induction:** IH is $(q, w, S) \vdash^* (q, y_i, \alpha_i)$. We have to show that

$$(q, y_i, \alpha_i) \vdash^* (q, y_{i+1}, \alpha_{i+1})$$

Now $\alpha_i$ begins with a variable $A$, and we have the form

$$\frac{x_i A \chi \Rightarrow x_i \beta \chi}{\gamma_i} \left\downarrow \right\downarrow \frac{\gamma_{i+1}}{\gamma_{i+1}}$$

By IH $A \chi$ is on the stack, and $y_i$ is unconsumed. From the construction of $P_G$ it follows that we can make the move

$$(q, y_i, A \chi) \vdash (q, y_i, \beta \chi).$$

If $\beta$ has a prefix of terminals, we can pop them with matching terminals in a prefix of $y_i$, ending up in configuration $(q, y_{i+1}, \alpha_{i+1})$, where $\alpha_{i+1}$ is the tail of the sentential form $x_{i+1} \alpha_{i+1} = \gamma_{i+1}$.

Finally, since $\gamma_n = w$, we have $\alpha_n = \epsilon$, and $y_n = \epsilon$, and thus $(q, w, S) \vdash^* (q, \epsilon, \epsilon)$, i.e. $w \in N(P_G)$.
(⊆-direction.) We shall show by an induction on the length of $\vdash^*$, that

(♣) If $(q, x, A) \vdash^* (q, \epsilon, \epsilon)$, then $A \Rightarrow x$.

**Basis:** Length 1. Then it must be that $A \rightarrow \epsilon$ is in $G$, and we have $(q, \epsilon) \in \delta(q, \epsilon, A)$. Thus $A \Rightarrow \epsilon$.

**Induction:** Length is $n > 1$, and the IH holds for lengths $< n$.

Since $A$ is a variable, we must have

$$(q, x, A) \vdash (q, x, Y_1Y_2 \cdots Y_k) \vdash \cdots \vdash (q, \epsilon, \epsilon)$$

where $A \rightarrow Y_1Y_2 \cdots Y_k$ is in $G$. 

206
We can now write $x$ as $x_1 x_2 \cdots x_k$, according to the figure below, where $Y_1 = B$, $Y_2 = a$, and $Y_3 = C$. 

![Diagram]

$= a$
Now we can conclude that

\[(q, x_i x_{i+1} \cdots x_k, Y_i) \vdash^* (q, x_{i+1} \cdots x_k, \epsilon)\]

is less than \(n\) steps, for all \(i \in \{1, \ldots, k\}\). If \(Y_i\) is a variable we have by the IH and Theorem 6.6 that

\[Y_i \Rightarrow^* x_i\]

If \(Y_i\) is a terminal, we have \(|x_i| = 1\), and \(Y_i = x_i\). Thus \(Y_i \Rightarrow^* x_i\) by the reflexivity of \(\Rightarrow^*\).

Hence, \(A \leadsto Y_1 Y_2 \cdots Y_k \Rightarrow^* x_1 x_2 \cdots x_k = x\)

The claim of the theorem now follows by choosing \(A = S\), and \(x = w\). Suppose \(w \in N(P)\). Then \((q, w, S) \vdash^* (q, \epsilon, \epsilon)\), and by (\(\spadesuit\)), we have \(S \Rightarrow^* w\), meaning \(w \in L(G)\).
Let’s look at how a PDA can consume \( x = x_1 x_2 \cdots x_k \) and empty the stack.

We shall define a grammar with variables of the form \([p_{i-1} Y_i p_i]\) representing going from \( p_{i-1} \) to \( p_i \) with net effect of popping \( Y_i \).
Formally, let \( P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0) \) be a PDA. Define \( G = (V, \Sigma, R, S) \), where

\[
V = \{ [pXq] : \{p, q\} \subseteq Q, X \in \Gamma \} \cup \{S\}
\]

\[
R = \{ S \to [q_0Z_0p] : p \in Q \} \cup
\]

\[
\{ [qXr_k] \to a[rY_1r_1] \cdots [r_{k-1}Y_kr_k] : a \in \Sigma \cup \{\epsilon\}, \{r_1, \ldots, r_k\} \subseteq Q, (r, Y_1Y_2\cdots Y_k) \in \delta(q, a, X) \}\]

If \( k = 0 \), i.e. \( Y_1Y_2\cdots Y_k = \epsilon \), then \([qXr] \rightarrow a\)
Example: Let’s convert

\[ P_N = (\{q\}, \{i, e\}, \{Z\}, \delta_N, q, Z), \]

where \( \delta_N(q, i, Z) = \{(q, ZZ)\} \),
and \( \delta_N(q, e, Z) = \{(q, \epsilon)\} \) to a grammar

\[ G = (V, \{i, e\}, R, S), \]

where \( V = \{[qZq], S\} \), and
\[ R = \{[qZq] \to i[qZq][qZq], [qZq] \to e, S \to [qZq] \} \]

If we replace \([qZq]\) by \(A\) we get the productions
\( S \to A \) and \( A \to iAA|e. \)
Example: Let $P = (\{p, q\}, \{0, 1\}, \{X, Z_0\}, \delta, q, Z_0)$, where $\delta$ is given by

1. $\delta(q, 1, Z_0) = \{(q, XZ_0)\}$

2. $\delta(q, 1, X) = \{(q, XX)\}$

3. $\delta(q, 0, X) = \{(p, X)\}$

4. $\delta(q, \epsilon, X) = \{(q, \epsilon)\}$

5. $\delta(p, 1, X) = \{(p, \epsilon)\}$

6. $\delta(p, 0, Z_0) = \{(q, Z_0)\}$

What language does this PDA accept?

to a CFG.
We get $G = (V, \{0, 1\}, R, S)$, where

$$V = \{[pXp], [pXq], [pZ_0p], [pZ_0q], [qXq], [pXq], [qZ_0p], [qZ_0q]\}$$

and the productions in $R$ are

$$S \rightarrow [qZ_0q][qZ_0p]$$

From rule (1):

$$[qZ_0q] \rightarrow 1[qXq][qZ_0q]$$
$$[qZ_0q] \rightarrow 1[qXp][pZ_0q]$$
$$[qZ_0p] \rightarrow 1[qXq][qZ_0p]$$
$$[qZ_0p] \rightarrow 1[qXp][pZ_0p]$$

From rule (2):

$$[qXq] \rightarrow 1[qXq][qXq]$$
$$[qXq] \rightarrow 1[qXp][pXq]$$
$$[qXp] \rightarrow 1[qXq][qXp]$$
$$[qXp] \rightarrow 1[qXp][pXp]$$
From rule (3):

\[ q X q \rightarrow 0[pXq] \]
\[ q X p \rightarrow 0[pXp] \]

From rule (4):

\[ q X q \rightarrow \epsilon \]

From rule (5):

\[ p X p \rightarrow 1 \]

From rule (6):

\[ p Z_0 q \rightarrow 0[q Z_0 q] \]
\[ p Z_0 p \rightarrow 0[q Z_0 p] \]
**Theorem 6.14:** Let $G$ be constructed from a PDA $P$ as above. Then $L(G) = N(P)$

**Proof:**

($\supseteq$-direction.) We shall show by an induction on the length of the sequence $\vdash^*$ that

(♠) If $(q, w, X) \vdash^* (p, \epsilon, \epsilon)$ then $[qXp] \Rightarrow^* w$.

**Basis:** Length 1. Then $w$ is an $a$ or $\epsilon$, and $(p, \epsilon) \in \delta(q, w, X)$. By the construction of $G$ we have $[qXp] \rightarrow w$ and thus $[qXp] \Rightarrow^* w$. 
**Induction:** Length is $n > 1$, and $\blacklozenge$ holds for lengths $< n$. We must have

$$(q, w, X) \vdash (r_0, x, Y_1 Y_2 \cdots Y_k) \vdash \cdots \vdash (p, \epsilon, \epsilon),$$

where $w = ax$ or $w = \epsilon x$. It follows that $(r_0, Y_1 Y_2 \cdots Y_k) \in \delta(q, a, X)$. Then we have a production

$$[q X r_k] \rightarrow a[r_0 Y_1 r_1] \cdots [r_{k-1} Y_k r_k],$$

for all $\{r_1, \ldots, r_k\} \subset Q$.

We may now choose $r_i$ to be the state in the sequence $\vdash^*$ when $Y_i$ is popped. Let $x = w_1 w_2 \cdots w_k$, where $w_i$ is consumed while $Y_i$ is popped. Then

$$(r_{i-1}, w_i, Y_i) \vdash^* (r_i, \epsilon, \epsilon).$$

**Note that** $r_k = p$

By the IH we get

$$[r_{i-1}, Y, r_i] \Rightarrow w_i$$
We then get the following derivation sequence:

\[
[qXr_k] \Rightarrow a[r_0Y_1r_1] \cdots [r_{k-1}Y_kr_k] \Rightarrow^* \\
aw_1[r_1Y_2r_2][r_2Y_3r_3] \cdots [r_{k-1}Y_kr_k] \Rightarrow^* \\
aw_1w_2[r_2Y_3r_3] \cdots [r_{k-1}Y_kr_k] \Rightarrow^* \\
\cdots \\
aw_1w_2 \cdots w_k = w = ax
\]
We shall show by an induction on the length of the derivation $\Rightarrow^*$ that

($\heartsuit$) If $[qXp] \Rightarrow^* w$ then $(q, w, X) \vdash^* (p, \epsilon, \epsilon)$

**Basis:** One step. Then we have a production $[qXp] \rightarrow w$. From the construction of $G$ it follows that $(p, \epsilon) \in \delta(q, a, X)$, where $w = a$. But then $(q, w, X) \vdash^* (p, \epsilon, \epsilon)$.

**Induction:** Length of $\Rightarrow$ is $n > 1$, and $\heartsuit$ holds for lengths $< n$. Then we must have

$$[qXr_k] \Rightarrow a[r_0Y_1r_1][r_1Y_2r_2] \cdots [r_{k-1}Y_kr_k] \Rightarrow^* w$$

We can break $w$ into $aw_1 \cdots w_k$ such that $[r_{i-1}Y_ir_i] \Rightarrow^* w_i$. From the IH we get

$$(r_{i-1}, w_i, Y_i) \vdash^* (r_i, \epsilon, \epsilon)$$
From Theorem 6.5 we get
\[(r_{i-1}, w_i w_{i+1} \cdots w_k, Y_i Y_{i+1} \cdots Y_k) \vdash^* \]
\[(r_i, w_{i+1} \cdots w_k, Y_{i+1} \cdots Y_k) \]

Since this holds for all \( i \in \{1, \ldots, k\} \), we get
\[(q, aw_1 w_2 \cdots w_k, X) \vdash \]
\[(r_0, w_1 w_2 \cdots w_k, Y_1 Y_2 \cdots Y_k) \vdash^* \]
\[(r_1, w_2 \cdots w_k, Y_2 \cdots Y_k) \vdash^* \]
\[(r_2, w_3 \cdots w_k, Y_3 \cdots Y_k) \vdash^* \]
\[(p, \epsilon, \epsilon). \]

\[p = r_k\]

Q1. Can you give a 1-state empty stack PDA for \( L_1 = \{0^n 1^n \mid n \geq 0\} \)?

Q2: How to decide if a PDA \( M \) accepts a string \( w \)?
A PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ is deterministic iff

1. $\delta(q, a, X)$ is always empty or a singleton.
2. If $\delta(q, a, X)$ is nonempty, then $\delta(q, \epsilon, X)$ must be empty.

Example: Let us define

$$L_{wcwr} = \{wcw^R : w \in \{0, 1\}^*\}$$

Then $L_{wcwr}$ is recognized by the following DPDA
We’ll show that Regular \( \subseteq L(DPDA) \subseteq CFL \)

**Theorem 6.17:** If \( L \) is regular, then \( L = L(P) \) for some DPDA \( P \).

**Proof:** Since \( L \) is regular there is a DFA \( A \) s.t. \( L = L(A) \). Let

\[
A = (Q, \Sigma, \delta_A, q_0, F)
\]

We define the DPDA

\[
P = (Q, \Sigma, \{Z_0\}, \delta_P, q_0, Z_0, F),
\]

where

\[
\delta_P(q, a, Z_0) = \{(\delta_A(q, a), Z_0)\},
\]

for all \( p, q \in Q \), and \( a \in \Sigma \).

An easy induction (do it!) on \(|w|\) gives

\[
(q_0, w, Z_0) \xrightarrow{*} (p, \epsilon, Z_0) \iff \delta_A(q_0, w) = p
\]

The theorem then follows (why?)
What about DPDA’s that accept by null stack?

They can recognize only CFL’s with the prefix property.

A language $L$ has the prefix property if there are no two distinct strings in $L$, such that one is a prefix of the other.

Example: $L_{wcwr}$ has the prefix property.

Example: $\{0\}^*$ does not have the prefix property.

**Theorem 6.19:** $L$ is $N(P)$ for some DPDA $P$ if and only if $L$ has the prefix property and $L$ is $L(P')$ for some DPDA $P'$.

**Proof:** Homework
- We have seen that $\text{Regular} \subseteq L(\text{DPDA})$.

- $L_{\text{wcwr}} \in L(\text{DPDA}) \setminus \text{Regular}$

- Are there languages in $\text{CFL} \setminus L(\text{DPDA})$.

  Yes, for example $L_{\text{wwr}}$.

- What about DPDA’s and Ambiguous Grammars?

  $L_{\text{wwr}}$ has unamb. grammar $S \rightarrow 0S0 | 1S1 | \epsilon$

  but is not $L(\text{DPDA})$.

For the converse we have

**Theorem 6.20:** If $L = N(P)$ for some DPDA $P$, then $L$ has an unambiguous CFG.

**Proof:** By inspecting the proof of Theorem 6.14 we see that if the construction is applied to a DPDA the result is a CFG with unique leftmost derivations.
Theorem 6.20 can actually be strengthened as follows

**Theorem 6.21:** If $L = L(P)$ for some DPDA $P$, then $L$ has an unambiguous CFG.

**Proof:** Let $\$ be a symbol outside the alphabet of $L$, and let $L' = L\$.

It is easy to see that $L'$ has the prefix property.

By Theorem 6.20 we have $L' = N(P')$ for some DPDA $P'$.

By Theorem 6.20 $N(P')$ can be generated by an unambiguous CFG $G'$.

Modify $G'$ into $G$, s.t. $L(G') = L$, by adding the production

$$\$ \rightarrow \epsilon$$

Since $G'$ has unique leftmost derivations, $G$ also has unique lm’s, since the only new thing we’re doing is adding derivations

$$w\$ \Rightarrow^{lm} w$$

to the end.