Pushdown Automata

Reading: Chapter 6
Pushdown Automata (PDA)

• **Informally:**
  – A PDA is an NFA-ε with an **infinite** stack.
  – Transitions are modified to accommodate stack operations.

• **Questions:**
  – What is a stack?
  – How does a stack help?

• A DFA can “remember” only a finite amount of information, whereas a PDA can “remember” an infinite amount of (certain types of) information.
• Example:

\[ \{0^n1^n \mid 0 \leq n \} \]  
Is not regular

\[ \{0^n1^n \mid 0 \leq n \leq k, \text{for some fixed } k \} \]  
Is regular, for any fixed k.

• For k=3:

\[ L = \{\varepsilon, 01, 0011, 000111\} \]
• In a DFA, each state remembers a finite amount of information.

• To get \( \{0^n1^n \mid n \geq 0\} \) with a DFA would require an infinite number of states using the preceding technique.

• An infinite stack solves the problem for \( \{0^n1^n \mid 0 \leq n\} \) as follows:
  – Read all 0’s and place them on a stack
  – Read all 1’s and match with the corresponding 0’s on the stack

• Only need two states to do this in a PDA

• Similarly for \( \{0^n1^m0^{n+m} \mid n,m \geq 0\} \)
Formal Definition of a PDA

- A pushdown automaton (PDA) is a seven-tuple:

\[ M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F) \]

- \( Q \) is a finite set of states
- \( \Sigma \) is a finite input alphabet
- \( \Gamma \) is a finite stack alphabet
- \( q_0 \) is the initial/starting state, in \( Q \)
- \( z_0 \) is a starting stack symbol, in \( \Gamma \)
- \( F \) is a set of final/accepting states, a subset of \( Q \)
- \( \delta \) is a transition function, where

\[ \delta: Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow \text{finite subsets of } Q \times \Gamma^* \]
• Consider the various parts of $\delta$:

$$Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow \text{finite subsets of } Q \times \Gamma^*$$

– $Q$ on the LHS means that at each step in a computation, a PDA must consider its’ current state.
– $\Gamma$ on the LHS means that at each step in a computation, a PDA must consider the symbol on top of its’ stack.
– $\Sigma \cup \{\epsilon\}$ on the LHS means that at each step in a computation, a PDA may or may not consider the current input symbol, i.e., it may have epsilon transitions.

– “Finite subsets” on the RHS means that at each step in a computation, a PDA will have several options.
– $Q$ on the RHS means that each option specifies a new state.
– $\Gamma^*$ on the RHS means that each option specifies zero or more stack symbols that will replace the top stack symbol.
Two types of PDA transitions:

\[ \delta(q, a, z) = \{(p_1, \gamma_1), (p_2, \gamma_2), \ldots, (p_m, \gamma_m)\} \]

- Current state is \( q \)
- Current input symbol is \( a \)
- Symbol currently on top of the stack \( z \)
- Move to state \( p_i \) from \( q \)
- Replace \( z \) with \( \gamma_i \) on the stack (leftmost symbol on top)
- Move the input head to the next input symbol
Two types of PDA transitions:

$$\delta(q, \varepsilon, z) = \{(p_1, \gamma_1), (p_2, \gamma_2), \ldots, (p_m, \gamma_m)\}$$

- Current state is q
- Current input symbol is not considered
- Symbol currently on top of the stack z
- Move to state $p_i$ from q
- Replace z with $\gamma_i$ on the stack (leftmost symbol on top)
- **No input symbol is read**
• **Example PDA #1:** (balanced parentheses)

\[
\text{\texttt{() (()) (()(())) ())))\ v}
\]

**Question:** How could we accept the language with a stack-based Java program?

\[
M = (\{q_1\}, \{\text{"\texttt{(}, \text{\texttt{)}}\}\}, \{L, \#\}, \delta, q_1, \#, \emptyset)
\]

\[
\delta:
\begin{align*}
(1) & \quad \delta(q_1, (, \#) = \{(q_1, L\#)\} \\
(2) & \quad \delta(q_1, ), \#) = \emptyset \\
(3) & \quad \delta(q_1, (, L) = \{(q_1, LL)\} \\
(4) & \quad \delta(q_1, ), L) = \{(q_1, \varepsilon)\} \\
(5) & \quad \delta(q_1, \varepsilon, \#) = \{(q_1, \varepsilon)\} \\
(6) & \quad \delta(q_1, \varepsilon, L) = \emptyset
\end{align*}
\]

• **Goal:** (acceptance)
  – Terminate in a state
  – Read the entire input string
  – Terminate with an empty stack

• Informally, a string is accepted if there exists a computation that uses up all the input and leaves the stack empty.
• Transition Diagram:

![Transition Diagram](image-url)

- Note that the above is not particularly illuminating.

- This is true for just about all PDAs, and consequently we don’t typically draw the transition diagram.

* More generally, states are not particularly important in a PDA.
**Example Computation:**

\[ M = (\{q_1\}, \{\text{"\(\),\ text{"}\)}}\}, \{L, \#\}, \delta, q_1, \#, \emptyset) \]

\[ \delta: \]

(1) \[ \delta(q_1, \text{(#)} = \{(q_1, L\#)\} \]
(2) \[ \delta(q_1, \text{, #}) = \emptyset \]
(3) \[ \delta(q_1, \text{(,), L}) = \{(q_1, LL)\} \]
(4) \[ \delta(q_1, \text{,), L}) = \{(q_1, \varepsilon)\} \]
(5) \[ \delta(q_1, \varepsilon, \#) = \{(q_1, \varepsilon)\} \]
(6) \[ \delta(q_1, \varepsilon, L) = \emptyset \]

<table>
<thead>
<tr>
<th>Current Input</th>
<th>Stack</th>
<th>Rules Applicable</th>
<th>Rule Applied</th>
</tr>
</thead>
<tbody>
<tr>
<td>()</td>
<td>#</td>
<td>(1), (5)</td>
<td>(1) --Why not 5?</td>
</tr>
<tr>
<td>())</td>
<td>L#</td>
<td>(3), (6)</td>
<td>(3)</td>
</tr>
<tr>
<td>))</td>
<td>LL#</td>
<td>(4), (6)</td>
<td>(4)</td>
</tr>
<tr>
<td>)</td>
<td>L#</td>
<td>(4), (6)</td>
<td>(4)</td>
</tr>
<tr>
<td>\varepsilon</td>
<td>#</td>
<td>(5)</td>
<td>(5)</td>
</tr>
<tr>
<td>\varepsilon</td>
<td>\varepsilon</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

- Note that from this point forward, rules such as (2) and (6) will not be listed or referenced in any computations.
• **Example PDA #2:** For the language \( \{ x \mid x = wcw^r \text{ and } w \in \{0,1\}^* \} \)

```
01c10  1101c1011  0010c0100  c
```

Question: How could we accept the language with a stack-based Java program?

\[
M = (\{q_1, q_2\}, \{0, 1, c\}, \{B, G, R\}, \delta, q_1, R, \emptyset)
\]

\[\delta:\]

<table>
<thead>
<tr>
<th>#</th>
<th>Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(\delta(q_1, 0, R) = {(q_1, BR)})</td>
</tr>
<tr>
<td>2</td>
<td>(\delta(q_1, 0, B) = {(q_1, BB)})</td>
</tr>
<tr>
<td>3</td>
<td>(\delta(q_1, 0, G) = {(q_1, BG)})</td>
</tr>
<tr>
<td>4</td>
<td>(\delta(q_1, c, R) = {(q_2, R)})</td>
</tr>
<tr>
<td>5</td>
<td>(\delta(q_1, c, B) = {(q_2, B)})</td>
</tr>
<tr>
<td>6</td>
<td>(\delta(q_1, c, G) = {(q_2, G)})</td>
</tr>
<tr>
<td>7</td>
<td>(\delta(q_2, 0, B) = {(q_2, \varepsilon)})</td>
</tr>
<tr>
<td>8</td>
<td>(\delta(q_2, \varepsilon, R) = {(q_2, \varepsilon)})</td>
</tr>
<tr>
<td>9</td>
<td>(\delta(q_1, 1, R) = {(q_1, GR)})</td>
</tr>
<tr>
<td>10</td>
<td>(\delta(q_1, 1, B) = {(q_1, GB)})</td>
</tr>
<tr>
<td>11</td>
<td>(\delta(q_1, 1, G) = {(q_1, GG)})</td>
</tr>
<tr>
<td>12</td>
<td>(\delta(q_2, 1, G) = {(q_2, \varepsilon)})</td>
</tr>
</tbody>
</table>

*Notes:*

- Rule #8 is used to pop the final stack symbol off at the end of a computation.
• Example Computation:

\[
\begin{align*}
(1) \quad \delta(q_1, 0, R) &= \{(q_1, BR)\} \\
(2) \quad \delta(q_1, 0, B) &= \{(q_1, BB)\} \\
(3) \quad \delta(q_1, 0, G) &= \{(q_1, BG)\} \\
(4) \quad \delta(q_1, c, R) &= \{(q_2, R)\} \\
(5) \quad \delta(q_1, c, B) &= \{(q_2, B)\} \\
(6) \quad \delta(q_1, c, G) &= \{(q_2, G)\} \\
(7) \quad \delta(q_2, 0, B) &= \{(q_2, \varepsilon)\} \\
(8) \quad \delta(q_2, \varepsilon, R) &= \{(q_2, \varepsilon)\} \\
(9) \quad \delta(q_1, 1, R) &= \{(q_1, GR)\} \\
(10) \quad \delta(q_1, 1, B) &= \{(q_1, GB)\} \\
(11) \quad \delta(q_1, 1, G) &= \{(q_1, GG)\} \\
(12) \quad \delta(q_2, 1, G) &= \{(q_2, \varepsilon)\}
\end{align*}
\]

<table>
<thead>
<tr>
<th>State</th>
<th>Input</th>
<th>Stack</th>
<th>Rules Applicable</th>
<th>Rule Applied</th>
</tr>
</thead>
<tbody>
<tr>
<td>q_1</td>
<td>01c10</td>
<td>R</td>
<td>(1)</td>
<td>(1)</td>
</tr>
<tr>
<td>q_1</td>
<td>1c10</td>
<td>BR</td>
<td>(10)</td>
<td>(10)</td>
</tr>
<tr>
<td>q_1</td>
<td>c10</td>
<td>GBR</td>
<td>(6)</td>
<td>(6)</td>
</tr>
<tr>
<td>q_2</td>
<td>10</td>
<td>GBR</td>
<td>(12)</td>
<td>(12)</td>
</tr>
<tr>
<td>q_2</td>
<td>0</td>
<td>BR</td>
<td>(7)</td>
<td>(7)</td>
</tr>
<tr>
<td>q_2</td>
<td>\varepsilon</td>
<td>R</td>
<td>(8)</td>
<td>(8)</td>
</tr>
<tr>
<td>q_2</td>
<td>\varepsilon</td>
<td>\varepsilon</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Example Computation:

1. \(\delta(q_1, 0, R) = \{(q_1, BR)\}\)
2. \(\delta(q_1, 0, B) = \{(q_1, BB)\}\)
3. \(\delta(q_1, 0, G) = \{(q_1, BG)\}\)
4. \(\delta(q_1, c, R) = \{(q_2, R)\}\)
5. \(\delta(q_1, c, B) = \{(q_2, B)\}\)
6. \(\delta(q_1, c, G) = \{(q_2, G)\}\)
7. \(\delta(q_2, 0, B) = \{(q_2, \varepsilon)\}\)
8. \(\delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\}\)
9. \(\delta(q_1, 1, R) = \{(q_1, GR)\}\)
10. \(\delta(q_1, 1, B) = \{(q_1, GB)\}\)
11. \(\delta(q_1, 1, G) = \{(q_1, GG)\}\)
12. \(\delta(q_2, 1, G) = \{(q_2, \varepsilon)\}\)

State | Input | Stack | Rules Applicable | Rule Applied |
--- | --- | --- | --- | --- |
q_1 | 1c1 | R | (9) | (9) |
q_1 | c1 | GR | (6) | (6) |
q_2 | 1 | GR | (12) | (12) |
q_2 | \varepsilon | R | (8) | (8) |
q_2 | \varepsilon | \varepsilon | - | - |

Questions:
- Why isn’t \(\delta(q_2, 0, G)\) defined?
- Why isn’t \(\delta(q_2, 1, B)\) defined?
• **Example PDA #3:** For the language \( \{ x | x = w w^r \text{ and } w \in \{0, 1\}^* \} \)

\[
M = (\{q_1, q_2\}, \{0, 1\}, \{R, B, G\}, \delta, q_1, R, \varnothing)
\]

\[
\delta:
\]

\begin{align*}
(1) & \quad \delta(q_1, 0, R) = \{(q_1, BR)\} & (6) & \quad \delta(q_1, 1, G) = \{(q_1, GG), (q_2, \varepsilon)\} \\
(2) & \quad \delta(q_1, 1, R) = \{(q_1, GR)\} & (7) & \quad \delta(q_2, 0, B) = \{(q_2, \varepsilon)\} \\
(3) & \quad \delta(q_1, 0, B) = \{(q_1, BB), (q_2, \varepsilon)\} & (8) & \quad \delta(q_2, 1, G) = \{(q_2, \varepsilon)\} \\
(4) & \quad \delta(q_1, 0, G) = \{(q_1, BG)\} & (9) & \quad \delta(q_1, \varepsilon, R) = \{(q_2, \varepsilon)\} \\
(5) & \quad \delta(q_1, 1, B) = \{(q_1, GB)\} & (10) & \quad \delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\}
\end{align*}

• **Notes:**
  – Rules #3 and #6 are non-deterministic.
  – Rules #9 and #10 are used to pop the final stack symbol off at the end of a computation.
### Example Computation:

1. \(\delta(q_1, 0, R) = \{ (q_1, BR) \} \)
2. \(\delta(q_1, 1, R) = \{ (q_1, GR) \} \)
3. \(\delta(q_1, 0, B) = \{ (q_1, BB), (q_2, \varepsilon) \} \)
4. \(\delta(q_1, 0, G) = \{ (q_1, BG) \} \)
5. \(\delta(q_1, 1, B) = \{ (q_1, GB) \} \)
6. \(\delta(q_1, 1, G) = \{ (q_1, GG), (q_2, \varepsilon) \} \)
7. \(\delta(q_2, 0, B) = \{ (q_2, \varepsilon) \} \)
8. \(\delta(q_2, 1, G) = \{ (q_2, \varepsilon) \} \)
9. \(\delta(q_1, \varepsilon, R) = \{ (q_2, \varepsilon) \} \)
10. \(\delta(q_2, \varepsilon, R) = \{ (q_2, \varepsilon) \} \)

<table>
<thead>
<tr>
<th>State</th>
<th>Input</th>
<th>Stack</th>
<th>Rules Applicable</th>
<th>Rule Applied</th>
</tr>
</thead>
<tbody>
<tr>
<td>(q_1)</td>
<td>000000</td>
<td>R</td>
<td>(1), (9)</td>
<td>(1)</td>
</tr>
<tr>
<td>(q_1)</td>
<td>00000</td>
<td>BR</td>
<td>(3), both options</td>
<td>(3), option #1</td>
</tr>
<tr>
<td>(q_1)</td>
<td>0000</td>
<td>BBR</td>
<td>(3), both options</td>
<td>(3), option #1</td>
</tr>
<tr>
<td>(q_1)</td>
<td>000</td>
<td>BBBR</td>
<td>(3), both options</td>
<td>(3), option #2</td>
</tr>
<tr>
<td>(q_2)</td>
<td>00</td>
<td>BBBR</td>
<td>(7)</td>
<td>(7)</td>
</tr>
<tr>
<td>(q_2)</td>
<td>0</td>
<td>BR</td>
<td>(7)</td>
<td>(7)</td>
</tr>
<tr>
<td>(q_2)</td>
<td>(\varepsilon)</td>
<td>R</td>
<td>(10)</td>
<td>(10)</td>
</tr>
<tr>
<td>(q_2)</td>
<td>(\varepsilon)</td>
<td>(\varepsilon)</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

### Questions:
- What is rule #10 used for?
- What is rule #9 used for?
- Why do rules #3 and #6 have options?
- Why don’t rules #4 and #5 have similar options?
• Example Computation:

(1) \( \delta(q_1, 0, R) = \{(q_1, BR)\} \)
(2) \( \delta(q_1, 1, R) = \{(q_1, GR)\} \)
(3) \( \delta(q_1, 0, B) = \{(q_1, BB), (q_2, \varepsilon)\} \)
(4) \( \delta(q_1, 0, G) = \{(q_1, BG)\} \)
(5) \( \delta(q_1, 1, B) = \{(q_1, GB)\} \)
(6) \( \delta(q_1, 1, G) = \{(q_1, GG), (q_2, \varepsilon)\} \)
(7) \( \delta(q_2, 0, B) = \{(q_2, \varepsilon)\} \)
(8) \( \delta(q_2, 1, G) = \{(q_2, \varepsilon)\} \)
(9) \( \delta(q_1, \varepsilon, R) = \{(q_2, \varepsilon)\} \)
(10) \( \delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\} \)

<table>
<thead>
<tr>
<th>State</th>
<th>Input</th>
<th>Stack</th>
<th>Rules Applicable</th>
<th>Rule Applied</th>
</tr>
</thead>
<tbody>
<tr>
<td>(q_1)</td>
<td>010010</td>
<td>R</td>
<td>(1), (9)</td>
<td>(1)</td>
</tr>
<tr>
<td>(q_1)</td>
<td>10010</td>
<td>BR</td>
<td>(2)</td>
<td>(5)</td>
</tr>
<tr>
<td>(q_1)</td>
<td>0010</td>
<td>GBR</td>
<td>(3)</td>
<td>(4)</td>
</tr>
<tr>
<td>(q_1)</td>
<td>010</td>
<td>BGBR</td>
<td>(3), both options</td>
<td>(3), option #2</td>
</tr>
<tr>
<td>(q_2)</td>
<td>10</td>
<td>GBR</td>
<td>(4)</td>
<td>(8)</td>
</tr>
<tr>
<td>(q_2)</td>
<td>0</td>
<td>BR</td>
<td>(7)</td>
<td>(7)</td>
</tr>
<tr>
<td>(q_2)</td>
<td>(\varepsilon)</td>
<td>R</td>
<td>(10)</td>
<td>(10)</td>
</tr>
<tr>
<td>(q_2)</td>
<td>(\varepsilon)</td>
<td>(\varepsilon)</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

• Exercises:
  – 0011001100
  – 011110
  – 0111
Exercises:

• Develop PDAs for any of the regular or context-free languages that we have discussed.

• Note that for regular languages an NFA that simply ignores it’s stack will work.

• For languages which are context-free but not regular, first try to envision a Java (or other high-level language) program that uses a stack to accept the language, and then convert it to a PDA.

• For example, for the set of all strings of the form $a^i b^j c^k$, such that either $i \neq j$ or $j \neq k$. Or the set of all strings not of the form $ww$. 
Formal Definitions for PDAs

• Let $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ be a PDA.

• **Definition:** An *instantaneous description* (ID) is a triple $(q, w, \gamma)$, where $q$ is in $Q$, $w$ is in $\Sigma^*$ and $\gamma$ is in $\Gamma^*$.
  - $q$ is the current state
  - $w$ is the unused input
  - $\gamma$ is the current stack contents

• **Example:** (for PDA #3)

  - $(q_1, 111, GBR)$
  - $(q_1, 11, GGBR)$
  - $(q_1, 111, GBR)$
  - $(q_2, 11, BR)$
  - $(q_1, 000, GR)$
  - $(q_2, 00, R)$
• Let $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ be a PDA.

• **Definition:** Let $a$ be in $\Sigma \cup \{\varepsilon\}$, $w$ be in $\Sigma^*$, $z$ be in $\Gamma$, and $\alpha$ and $\beta$ both be in $\Gamma^*$. Then:

\[(q, aw, z\alpha) \vdash_M (p, w, \beta\alpha)\]

if $\delta(q, a, z)$ contains $(p, \beta)$.

• Intuitively, if $I$ and $J$ are instantaneous descriptions, then $I \vdash J$ means that $J$ follows from $I$ by one transition.
• **Examples:** (PDA #3)

(q₁, 111, GBR) ― (q₁, 11, GGBR)  \( (6) \) option #1, with \( a=1 \), \( z=G \), \( \beta=GG \), \( w=11 \), and \( \alpha=BR \)

(q₁, 111, GBR) ― (q₂, 11, BR)  \( (6) \) option #2, with \( a=1 \), \( z=G \), \( \beta=\varepsilon \), \( w=11 \), and \( \alpha=BR \)

(q₁, 000, GR) ― (q₂, 00, R)  Is not true, For any \( a \), \( z \), \( \beta \), \( w \) and \( \alpha \)

• **Examples:** (PDA #1)

(q₁, (())), L#) ― (q₁, ()),LL#)  \( (3) \)

21
A computation by a PDA can be expressed using this notation (PDA #3):

\[(q_1, 010010, R) \mid\rightarrow (q_1, 10010, BR) \quad (1)
\mid\rightarrow (q_1, 0010, GBR) \quad (5)
\mid\rightarrow (q_1, 010, BGBR) \quad (4)
\mid\rightarrow (q_2, 10, GBR) \quad (3), \text{ option } #2
\mid\rightarrow (q_2, 0, BR) \quad (8)
\mid\rightarrow (q_2, \varepsilon, R) \quad (7)
\mid\rightarrow (q_2, \varepsilon, \varepsilon) \quad (10)
\]

\[(q_1, \varepsilon, R) \mid\rightarrow (q_2, \varepsilon, \varepsilon) \quad (9)
\]
• **Definition:** $\rightarrow^*$ is the reflexive and transitive closure of $\rightarrow$.
  
  – $\rightarrow^* I$ for each instantaneous description $I$
  
  – If $I \rightarrow J$ and $J \rightarrow^* K$ then $I \rightarrow^* K$

• Intuitively, if $I$ and $J$ are instantaneous descriptions, then $I \rightarrow^* J$ means that $J$ follows from $I$ by zero or more transitions.

• **Examples:** (PDA #3)

\[
(q_1, 010010, R) \rightarrow^* (q_2, 10, GBR)
\]

\[
(q_1, 010010, R) \rightarrow^* (q_2, \varepsilon, \varepsilon)
\]

\[
(q_1, 111, GBR) \rightarrow^* (q_1, \varepsilon, GGGGBR)
\]

\[
(q_1, 01, GR) \rightarrow^* (q_1, 1, BGR)
\]

\[
(q_1, 101, GBR) \rightarrow^* (q_1, 101, GBR)
\]
• **Definition:** Let $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ be a PDA. The *language accepted by empty stack*, denoted $L_E(M)$, is the set

$$\{w \mid (q_0, w, z_0) \xrightarrow{*} (p, \epsilon, \epsilon) \text{ for some } p \in Q\}$$

• **Definition:** Let $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ be a PDA. The *language accepted by final state*, denoted $L_F(M)$, is the set

$$\{w \mid (q_0, w, z_0) \xrightarrow{*} (p, \epsilon, \gamma) \text{ for some } p \in F \text{ and } \gamma \in \Gamma^*\}$$

• **Definition:** Let $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ be a PDA. The *language accepted by empty stack and final state*, denoted $L(M)$, is the set

$$\{w \mid (q_0, w, z_0) \xrightarrow{*} (p, \epsilon, \epsilon) \text{ for some } p \in F\}$$

• **Questions:**
  – Does the book define string acceptance by empty stack, final state, both, or neither?
  – As an exercise, convert the preceding PDAs to other PDAs with different acceptance criteria.
• **Lemma 1:** Let $L = L_E(M_1)$ for some PDA $M_1$. Then there exists a PDA $M_2$ such that $L = L_F(M_2)$.

• **Lemma 2:** Let $L = L_F(M_1)$ for some PDA $M_1$. Then there exists a PDA $M_2$ such that $L = L_E(M_2)$.

• **Theorem:** Let $L$ be a language. Then there exists a PDA $M_1$ such that $L = L_F(M_1)$ if and only if there exists a PDA $M_2$ such that $L = L_E(M_2)$.

• **Corollary:** The PDAs that accept by empty stack and the PDAs that accept by final state define the same class of languages.

• **Notes:**
  - Similar lemmas and theorems could be stated for PDAs that accept by both final state and empty stack.
  - Part of the lesson here is that one can define “acceptance” in many different ways, e.g., a string is accepted by a DFA if you simply pass through an accepting state, or if you pass through an accepting state exactly twice.
• **Definition:** Let $G = (V, T, P, S)$ be a CFG. If every production in $P$ is of the form

\[ A \rightarrow a\alpha \]

Where $A$ is in $V$, $a$ is in $T$, and $\alpha$ is in $V^*$, then $G$ is said to be in **Greibach Normal Form** (GNF).

• **Example:**

\[
S \rightarrow aAB \mid bB \\
A \rightarrow aA \mid a \\
B \rightarrow bB \mid c
\]

• **Theorem:** Let $L$ be a CFL. Then $L \setminus \{\varepsilon\}$ is a CFL.

• **Theorem:** Let $L$ be a CFL not containing $\{\varepsilon\}$. Then there exists a GNF grammar $G$ such that $L = L(G)$. 
• **Lemma 1:** Let $L$ be a CFL. Then there exists a PDA $M$ such that $L = L_E(M)$.

• **Proof:** Assume without loss of generality that $\varepsilon$ is not in $L$. The construction can be modified to include $\varepsilon$ later.

Let $G = (V, T, P, S)$ be a CFG, where $L = L(G)$, and assume without loss of generality that $G$ is in GNF.

Construct $M = (Q, \Sigma, \Gamma, \delta, q, z, \emptyset)$ where:

- $Q = \{q\}$
- $\Sigma = T$
- $\Gamma = V$
- $z = S$

$\delta$: for all $a$ in $T$, $A$ in $V$ and $\gamma$ in $V^*$, if $A \rightarrow a\gamma$ is in $P$ then $\delta(q, a, A)$ will contain $(q, \gamma)$

Stated another way:

$$\delta(q, a, A) = \{(q, \gamma) \mid A \rightarrow a\gamma \text{ is in } P\}, \text{ for all } a \text{ in } T \text{ and } A \text{ in } V$$

• Huh?

• As we will see, for a given string $x$ in $\Sigma^*$, $M$ will attempt to simulate a leftmost derivation of $x$ with $G$. 
- **Example #1:** Consider the following CFG in GNF.

  $$S \rightarrow aS \quad G \text{ is in GNF}$$

  $$S \rightarrow a \quad L(G) = a^+$$

  Construct M as:

  $$Q = \{q\}$$
  $$\Sigma = T = \{a\}$$
  $$\Gamma = V = \{S\}$$
  $$z = S$$

  $$\delta(q, a, S) = \{(q, S), (q, \varepsilon)\}$$
  $$\delta(q, \varepsilon, S) = \emptyset$$

- **Question:** Is that all? Is $\delta$ complete? Recall that $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow \text{finite subsets of } Q \times \Gamma^*$
• **Example #2:** Consider the following CFG in GNF.

(1) \( S \rightarrow aA \)
(2) \( S \rightarrow aB \)
(3) \( A \rightarrow aA \) \quad \text{G is in GNF}
(4) \( A \rightarrow aB \) \quad L(G) = a^+b^+
(5) \( B \rightarrow bB \)
(6) \( B \rightarrow b \)

Construct M as:

- \( Q = \{q\} \)
- \( \Sigma = T = \{a, b\} \)
- \( \Gamma = V = \{S, A, B\} \)
- \( z = S \)

\[
\begin{align*}
\delta(q, a, S) &= ? \\
\delta(q, a, A) &= ? \\
\delta(q, a, B) &= ? \\
\delta(q, b, S) &= ? \\
\delta(q, b, A) &= ? \\
\delta(q, b, B) &= ? \\
\delta(q, \varepsilon, S) &= ? \\
\delta(q, \varepsilon, A) &= ? \\
\delta(q, \varepsilon, B) &= ?
\end{align*}
\]

Why 9? Recall \( \delta: Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow \text{finite subsets of } Q \times \Gamma^* \)
**Example #2:** Consider the following CFG in GNF.

1. \( S \rightarrow aA \)
2. \( S \rightarrow aB \)
3. \( A \rightarrow aA \) \( \text{G is in GNF} \)
4. \( A \rightarrow aB \) \( L(G) = a^+b^+ \)
5. \( B \rightarrow bB \)
6. \( B \rightarrow b \)

Construct \( M \) as:

- \( Q = \{q\} \)
- \( \Sigma = T = \{a, b\} \)
- \( \Gamma = V = \{S, A, B\} \)
- \( z = S \)

1. \( \delta(q, a, S) = \{(q, A), (q, B)\} \) \( S \rightarrow a\gamma \) \( \text{How many productions are there of this form?} \)
2. \( \delta(q, a, A) = ? \)
3. \( \delta(q, a, B) = ? \)
4. \( \delta(q, b, S) = ? \)
5. \( \delta(q, b, A) = ? \)
6. \( \delta(q, b, B) = ? \)
7. \( \delta(q, \varepsilon, S) = ? \)
8. \( \delta(q, \varepsilon, A) = ? \)
9. \( \delta(q, \varepsilon, B) = ? \) \( \text{Why 9? Recall } \delta: Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow \text{finite subsets of } Q \times \Gamma^* \)
• **Example #2:** Consider the following CFG in GNF.

\[
\begin{align*}
(1) & \quad S \rightarrow aA \\
(2) & \quad S \rightarrow aB \\
(3) & \quad A \rightarrow aA & \text{G is in GNF} \\
(4) & \quad A \rightarrow aB & \text{L(G) = a^*b^+} \\
(5) & \quad B \rightarrow bB \\
(6) & \quad B \rightarrow b
\end{align*}
\]

Construct M as:
\[
\begin{align*}
Q & = \{q\} \\
\Sigma & = \mathcal{T} = \{a, b\} \\
\Gamma & = \mathcal{V} = \{S, A, B\} \\
z & = S
\end{align*}
\]

\[
\begin{align*}
(1) & \quad \delta(q, a, S) = \{(q, A), (q, B)\} \quad \text{From productions #1 and 2, S} \rightarrow \text{aA, S} \rightarrow \text{aB} \\
(2) & \quad \delta(q, a, A) = \{(q, A), (q, B)\} \quad \text{From productions #3 and 4, A} \rightarrow \text{aA, A} \rightarrow \text{aB} \\
(3) & \quad \delta(q, a, B) = \emptyset \\
(4) & \quad \delta(q, b, S) = \emptyset \\
(5) & \quad \delta(q, b, A) = \emptyset \\
(6) & \quad \delta(q, b, B) = \{(q, B), (q, \varepsilon)\} \quad \text{From productions #5 and 6, B} \rightarrow \text{bB, B} \rightarrow \text{b} \\
(7) & \quad \delta(q, \varepsilon, S) = \emptyset \\
(8) & \quad \delta(q, \varepsilon, A) = \emptyset \\
(t9) & \quad \delta(q, \varepsilon, B) = \emptyset \quad \text{Recall } \delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow \text{finite subsets of } Q \times \Gamma^*
\end{align*}
\]
• For a string \( w \) in \( L(G) \) the PDA \( M \) will simulate a leftmost derivation of \( w \).
  
  – If \( w \) is in \( L(G) \) then \((q, w, z_0) \rightarrow^* (q, \varepsilon, \varepsilon)\)
  
  – If \((q, w, z_0) \rightarrow^* (q, \varepsilon, \varepsilon)\) then \( w \) is in \( L(G) \)
  
• Consider generating a string using \( G \). Since \( G \) is in GNF, each sentential form in a *leftmost* derivation has form:

\[
\Rightarrow t_1 t_2 \ldots t_i A_1 A_2 \ldots A_m
\]

terminals non-terminals

• And each step in the derivation (i.e., each application of a production) adds a terminal and some non-terminals.

\[
A_1 \rightarrow t_{i+1} \alpha
\]

\[
\Rightarrow t_1 t_2 \ldots t_i t_{i+1} \alpha A_2 \ldots A_m
\]

• Each transition of the PDA simulates one derivation step. Thus, the \( i^{th} \) step of the PDAs’ computation corresponds to the \( i^{th} \) step in a corresponding leftmost derivation.

• After the \( i^{th} \) step of the computation of the PDA, \( t_1 t_2 \ldots t_i \) are the symbols that have already been read by the PDA and \( A_1 A_2 \ldots A_m \) are the stack contents.
• For each leftmost derivation of a string generated by the grammar, there is an equivalent accepting computation of that string by the PDA.

• Each sentential form in the leftmost derivation corresponds to an instantaneous description in the PDA’s corresponding computation.

• For example, the PDA instantaneous description corresponding to the sentential form:

  \[ \Rightarrow t_1 t_2 \ldots t_i A_1 A_2 \ldots A_m \]

  would be:

  \[ (q, t_{i+1} t_{i+2} \ldots t_n, A_1 A_2 \ldots A_m) \]
• **Example:** Using the grammar from example #2:

\[
\begin{align*}
S & \Rightarrow aA \quad (p1) \\
& \Rightarrow aaA \quad (p3) \\
& \Rightarrow aaaA \quad (p3) \\
& \Rightarrow aaaaB \quad (p4) \\
& \Rightarrow aaaabB \quad (p5) \\
& \Rightarrow aaaaBB \quad (p6)
\end{align*}
\]

• The corresponding computation of the PDA:

\[
\begin{align*}
& \delta(q, a, S) = \{(q, A), (q, B)\} \quad \text{productions p1 and p2} \\
& \delta(q, a, A) = \{(q, A), (q, B)\} \quad \text{productions p3 and p4} \\
& \delta(q, a, B) = \emptyset \\
& \delta(q, b, S) = \emptyset \\
& \delta(q, b, A) = \emptyset \\
& \delta(q, b, B) = \{(q, B), (q, \varepsilon)\} \quad \text{productions p5} \\
& \delta(q, \varepsilon, S) = \emptyset \\
& \delta(q, \varepsilon, A) = \emptyset \\
& \delta(q, \varepsilon, B) = \emptyset
\end{align*}
\]

• (q, aaaabb, S) |— ?
• **Example:** Using the grammar from example #2:

\[
\begin{align*}
S & \Rightarrow aA \quad (p1) \\
& \Rightarrow aaA \quad (p3) \\
& \Rightarrow aaaA \quad (p3) \\
& \Rightarrow aaaB \quad (p4) \\
& \Rightarrow aaaabB \quad (p5) \\
& \Rightarrow aaaaB \quad (p6)
\end{align*}
\]

• The corresponding computation of the PDA:

\[
\begin{align*}
\text{String is read} & \quad \text{Stack is emptied} \\
\text{Therefore the string is accepted by the PDA}
\end{align*}
\]
• **Another Example:** Using the PDA from example #2:

\[
\begin{align*}
(q, aabb, S) & \rightarrow (q, abb, A) \quad (t1)/1 \\
& \rightarrow (q, bb, B) \quad (t2)/2 \\
& \rightarrow (q, b, B) \quad (t6)/1 \\
& \rightarrow (q, \varepsilon, \varepsilon) \quad (t6)/2 \\
\end{align*}
\]

- (p1) \( S \rightarrow aA \)
- (p2) \( S \rightarrow aB \)
- (p3) \( A \rightarrow aA \)
- (p4) \( A \rightarrow aB \)
- (p5) \( B \rightarrow bB \)
- (p6) \( B \rightarrow b \)

- (t1) \( \delta(q, a, S) = \{(q, A), (q, B)\} \) productions p1 and p2
- (t2) \( \delta(q, a, A) = \{(q, A), (q, B)\} \) productions p3 and p4
- (t3) \( \delta(q, a, B) = \emptyset \)
- (t4) \( \delta(q, b, S) = \emptyset \)
- (t5) \( \delta(q, b, A) = \emptyset \)
- (t6) \( \delta(q, b, B) = \{(q, B), (q, \varepsilon)\} \) productions p5
- (t7) \( \delta(q, \varepsilon, S) = \emptyset \)
- (t8) \( \delta(q, \varepsilon, A) = \emptyset \)
- (t9) \( \delta(q, \varepsilon, B) = \emptyset \)

• The corresponding derivation using the grammar:

\[
\begin{align*}
S & \Rightarrow ?
\end{align*}
\]
• **Another Example:** Using the PDA from example #2:

(q, aabb, S) |— (q, abb, A) | (t1)/1

|— (q, bb, B) | (t2)/2

|— (q, b, B) | (t6)/1

|— (q, ε, ε) | (t6)/2

• The corresponding derivation using the grammar:

\[ S \Rightarrow aA \] \hspace{1cm} (p1)

\[ \Rightarrow aaB \] \hspace{1cm} (p4)

\[ \Rightarrow aabB \] \hspace{1cm} (p5)

\[ \Rightarrow aabb \] \hspace{1cm} (p6)
• **Example #3:** Consider the following CFG in GNF.

\[
\begin{align*}
(1) \quad & S \rightarrow aABC \\
(2) \quad & A \rightarrow a \quad \text{G is in GNF} \\
(3) \quad & B \rightarrow b \\
(4) \quad & C \rightarrow cAB \\
(5) \quad & C \rightarrow cC \\

\text{Construct M as:} \\
Q = \{q\} \\
\Sigma = T = \{a, b, c\} \\
\Gamma = V = \{S, A, B, C\} \\
z = S \\
\end{align*}
\]

\[
\begin{align*}
(1) \quad & \delta(q, a, S) = \{(q, ABC)\} \quad S \rightarrow aABC \\
(2) \quad & \delta(q, a, A) = \{(q, \varepsilon)\} \quad A \rightarrow a \\
(3) \quad & \delta(q, a, B) = \emptyset \\
(4) \quad & \delta(q, a, C) = \emptyset \\
(5) \quad & \delta(q, b, S) = \emptyset \\
(6) \quad & \delta(q, b, A) = \emptyset \\
(7) \quad & \delta(q, b, B) = \{(q, \varepsilon)\} \quad B \rightarrow b \\
(8) \quad & \delta(q, b, C) = \emptyset \\
(9) \quad & \delta(q, c, S) = \emptyset \\
(10) \quad & \delta(q, c, A) = \emptyset \\
(11) \quad & \delta(q, c, B) = \emptyset \\
(12) \quad & \delta(q, c, C) = \{(q, AB), (q, C)\} \quad \text{C} \rightarrow \text{cAB}|\text{cC} \\
(13) \quad & \delta(q, \varepsilon, S) = \emptyset \\
(14) \quad & \delta(q, \varepsilon, A) = \emptyset \\
(15) \quad & \delta(q, \varepsilon, B) = \emptyset \\
(16) \quad & \delta(q, \varepsilon, C) = \emptyset \\
\end{align*}
\]
• Notes:
  – Recall that the grammar $G$ was required to be in GNF before the construction could be applied.
  – As a result, it was assumed that $\varepsilon$ was not in the context-free language $L$.
  – What if $\varepsilon$ is in $L$?

• Suppose $\varepsilon$ is in $L$:

1) First, let $L' = L - \{\varepsilon\}$

By an earlier theorem, if $L$ is a CFL, then $L' = L - \{\varepsilon\}$ is a CFL.

By another earlier theorem, there is GNF grammar $G$ such that $L' = L(G)$.

2) Construct a PDA $M$ such that $L' = L_E(M)$

How do we modify $M$ to accept $\varepsilon$?

Add $\delta(q, \varepsilon, S) = \{(q, \varepsilon)\}$? No!
• Counter Example:

Consider \( L = \{\varepsilon, b, ab, aab, aaab, \ldots\} \)  
Then \( L' = \{b, ab, aab, aaab, \ldots\} \)

• The GNF CFG for \( L' \):

(1) \( S \rightarrow aS \)
(2) \( S \rightarrow b \)

• The PDA \( M \) Accepting \( L' \):

\( Q = \{q\} \)
\( \Sigma = T = \{a, b\} \)
\( \Gamma = V = \{S\} \)
\( z = S \)

\( \delta(q, a, S) = \{(q, S)\} \)
\( \delta(q, b, S) = \{(q, \varepsilon)\} \)
\( \delta(q, \varepsilon, S) = \emptyset \)

• If \( \delta(q, \varepsilon, S) = \{(q, \varepsilon)\} \) is added then:

\( L(M) = \{\varepsilon, a, aa, aaa, \ldots, b, ab, aab, aaab, \ldots\} \)
3) Instead, add a new start state \( q' \) with transitions:

\[
\delta(q', \varepsilon, S) = \{(q', \varepsilon), (q, S)\}
\]

where \( q \) is the start state of the machine from the initial construction.

- **Lemma 1**: Let \( L \) be a CFL. Then there exists a PDA \( M \) such that \( L = L_E(M) \).

- **Lemma 2**: Let \( M \) be a PDA. Then there exists a CFG grammar \( G \) such that \( L_E(M) = L(G) \). -- Note that we did not prove this.

- **Theorem**: Let \( L \) be a language. Then there exists a CFG \( G \) such that \( L = L(G) \) iff there exists a PDA \( M \) such that \( L = L_E(M) \).

- **Corollary**: The PDAs define the CFLs.