**Pushdown Automata**

NFA with single, last-in-first-out push down stack  
 unlimited memory

\[\begin{array}{c}
\text{NFA} \\
\downarrow \\
x \\
\downarrow \\
\text{from stack alphabet } \Gamma
\end{array}\]

**Stack operations:**
- Read top
- Remove top \((\text{Pop})\)
- Write top \((\text{Push})\)

\[a, x \rightarrow y\] Reading input \(a\) with \(x\) on top, replace \(x\) with \(y\) \((a, x = \varepsilon)\)

\[\text{PDA} \quad \leftarrow \quad \text{CFG}\]

---

**Example PDA**

PDA to recognize \(\{0^n1^n \mid n \geq 0\}\) \((\text{non-regular})\)

\[\begin{array}{c}
\text{NFA} \\
\downarrow \\
\Sigma = \{0,1\} \\
\downarrow \\
\Gamma = \{0, \$\}
\end{array}\]

Push \(\$\) on stack.

Start reading input. As read 0's, push them on stack.
When reach 1, for each 1, pop a 0 off stack. If no 0 to pop, reject.
If reach another 0, after you've read a 1, reject.
Accept if finish input, and stack has \(\$\) on top.
Finish by popping \(\$\) off stack.  \((\text{Deterministic})\)
Formal Definition of PDA

A PDA $M$ is a 6-tuple $(Q, \Sigma, \Gamma, q_0, F)$ where,

1. $Q$ is a finite set of states
2. $\Sigma$ is the input alphabet
3. $\Gamma$ is the stack alphabet
4. $\delta : Q \times \Sigma \times \Gamma \rightarrow \delta' (Q \times \Gamma)$
5. $q_0$ in $Q$, the start state
6. $F \subseteq Q$, set of accept (final) states

Ex. $Q = \{q_1, q_2, q_3, q_4\}$ $\Sigma = \{0, 1\}$ $\Gamma = \{0, 1, \epsilon\}$ $F = \{q_1, q_4\}$

<table>
<thead>
<tr>
<th>input</th>
<th>0</th>
<th>1</th>
<th>$\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>stack</td>
<td>0</td>
<td>$$</td>
<td>$\epsilon$</td>
</tr>
<tr>
<td>$q_1$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q_2$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q_3$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q_4$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

State Diagram of PDA

to recognize $\{0^n1^n | n \geq 0\}$

- $q_1 \xrightarrow{\epsilon, \epsilon, \$} q_2 \xrightarrow{0, \epsilon, 0} \epsilon, 1, 0, \epsilon$
- $q_4 \xrightarrow{\epsilon, \$, \epsilon} q_3 \xrightarrow{1, 0, \epsilon} \epsilon, 1, 0, \epsilon$

a, b, c: input a, top stack b, replace top stack c
- $a = \epsilon$ means don’t read input symbol
- $b = \epsilon$ means don’t read or pop top of stack
- $c = \epsilon$ means don’t write on top
Formal Definition of PDA Acceptance

A PDA $M$ accepts $w = w_1 \ldots w_m$ in $\Sigma^*$ if there is a sequence of states $r_0, r_1, \ldots, r_m$ in $Q$ and strings $s_0, s_1, s_2, \ldots, s_m$ in $\Gamma^*$ with

1. $r_0 = q_0$, $s_0 = \varepsilon$ (starts properly)
2. $(r_{i+1}, b)$ is an element of $\delta(r_i, w_{i+1}, a)$ with $s_i = at$ and $s_{i+1} = bt$, with $a, b$ in $\Gamma$, $t$ in $\Gamma^*$ (moves properly)
3. $r_m$ is in $F$ (ends in final state)

Note: Don’t require stack to be empty to accept.

Example PDA

$L = \{ w \in \{a,b\}^* \}$

$L = \{ w \mid w \text{ has the same number of 0's and 1's} \}$
Example PDA

\[ L = \{ a^i b^j c^k \mid i, j, k \geq 0 \text{ and } i=j \text{ or } i = k \} \]

*Non-deterministically check that #a’s = #b’s OR #a’s = #c’s*