Formal Definition of Acceptance

PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, F)$ accepts a word $w \in \Sigma^*$
where $w = w_1 w_2 w_3 \ldots w_m$ with $w_i \in \Sigma_{\epsilon}$
if there exists a sequence
\[(q_0, s_0) \rightarrow (q_1, s_1) \rightarrow (q_2, s_2) \rightarrow \ldots \rightarrow (q_m, s_m)\]
where
- $s_i \in \Gamma^*$ (represent the stack), with $s_0 = \epsilon$,
- $q_m \in F$,
- $(q_{i+1}, b) \in \delta(q_i, w_{i+1}, a)$
  where $s_i = at$ and $s_{i+1} = bt$, $a, b \in \Gamma_{\epsilon}$, $t \in \Gamma^*$
Theorem

Suppose L is generated by a CFG $G = (V, \Sigma, R, S)$
Construct $P = (Q, \Sigma, \Gamma, \delta, q, F)$ that recognizes L

A Language is generated by a CFG $\iff$
It is recognized by a PDA
A Language is generated by a CFG

It is recognized by a PDA
Proof Ideas

- $A_{pq} =$ variable generating all $x$ that takes $P$ from $(p, \varepsilon)$ to $(q, \varepsilon)$
- Formal construction:
  - $V = \{ A_{pq} | p, q \in Q \}$
  - $S = A_{q0q}$
  - Defining $R$:
    Intuition: Derivations correspond to computations of $P$;
    There are two cases for a derivation from $A_{pq}$
    1. Stack is empty only at the beginning and end of the derivation from $A_{pq}$
    2. Stack becomes empty somewhere in between
Recap of Proof Ideas

- $A_{pq} =$ variable generating all $x$ that takes $P$ from $(p, \varepsilon)$ to $(q, \varepsilon)$

- Formal construction:
  - $V = \{ A_{pq} | p, q \in Q \}$
  - $S = A_{q0qf}$
  - $R$ defined as follows:
    - $A_{pp} \Rightarrow \varepsilon \forall p \in Q$
    - $A_{pq} \Rightarrow a A_{rs} b \forall p, q, r, s \in Q$
    - s.t. $(r, t) \in \delta(p, a, \varepsilon)$
    - $(q, \varepsilon) \in \delta(s, b, t)$
    - $A_{pq} \Rightarrow A_{pr} A_{rq} \forall p, q, r \in Q$

[Proof sketched on whiteboard, see textbook for details]

---

CFL Pumping Lemma

Let $L$ be a context-free language

Then there exists $p$ such that
for all $w \in L$ and $|w| \geq p$
we can write $w = uvxyz$, where:

1. $uv^ixyz \in L$ for any $i \geq 0$
2. $|vy| > 0$
3. $|vxy| \leq p$
“Surgery” on Parse Trees

Idea: If $w$ is long enough, then any parse tree for $w$ must have a path that contains a variable more than once