Equivalence of CFG's and PDA's

In Lecture 22, we saw the following.

Theorem 2.20: A language is context-free if and only if some PDA recognizes it.

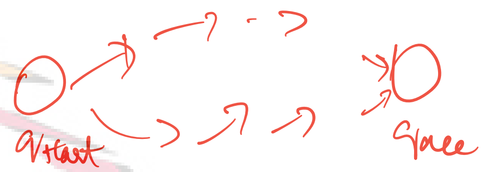
Lemma 2.21: If a language is context-free, then some PDA recognizes it.

We proved Lemma 2.21 in Lecture 22. To complete the proof of Theorem 2.20, we need to show the other direction as well.

Lemma 2.27: If a pushdown automaton recognizes a language, then it is context-free.

Assume that there is a PDA P.
Because of the normalizations discussed in Lecture 21, we may assume WLOG the following:

1. P has a single accepting state
2. P empties stack before accepting.
3. Each transition either pushes or pops, but not both.



GOAL : To obtain a CFG G that generates all the strings that can take P from q_{start} to q_{accept}.

We set variable A_{pq} to generate all strings that can take the PDA P from state p to state q, with an empty stack.

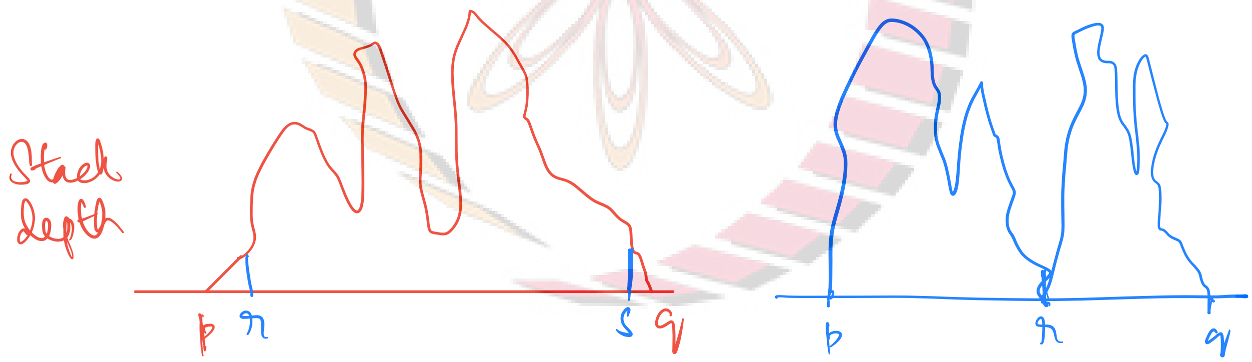
$$A_{p,q} = \{ \text{all strings that move P from } (p, \text{empty stack}) \rightarrow (q, \text{empty stack}) \}$$

<u>Note</u> : These strings also allow the stack to be retained.

Then $A_{q_{start}, q_{accept}}$ generates $L(P)$.

While processing any string, P's first move must involve a push into the stack. The last move (while accepting a string) must be a pop. There are two possibilities.

→ last move pops the same symbol that was pushed in the first move. (That is, stack never gets emptied till the end)



Now consider a string in A_{pq}. We add the following rules.

where $a, b \in \Sigma_\varepsilon$

$$A_{pq} \rightarrow \boxed{a\ A_{rs}\ b}$$

where a is input read in first move, and b is input read in last move. and state r follows p and state q follows s.

→ State becomes empty in between, at state r.

$$A_{pq} \to A_{pr} A_{rq}$$

Proof: Let $P = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{accept}\})$ and we will construct G.

G has variables $\{A_{pq} \mid p, q \in Q\}$.

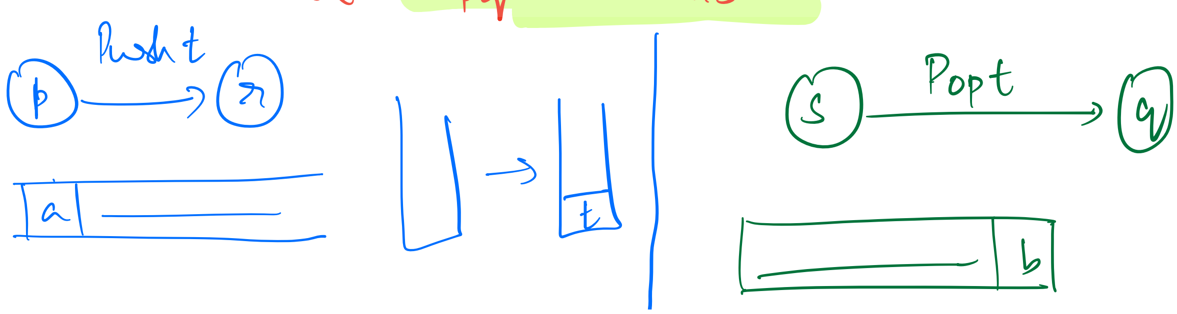
The start variable is $A_{q_0, q_{accept}}$

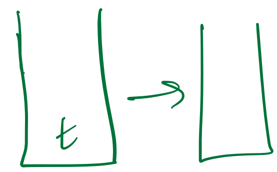
The rules of G are

→ For each $p, q, r, s \in Q$, $t \in \Gamma$, $a, b \in \Sigma_\varepsilon$,

If $(r, t) \in \delta(p, a, \varepsilon)$ and $(q, \varepsilon) \in \delta(s, b, t)$

add $A_{pq} \to a A_{rs} b$





→ For each $p, q, r \in Q$

add $A_{pq} \to A_{pr} A_{rq}$

→ For all $p \in Q$, add $A_{pp} \to \varepsilon$

This completes the construction. Now we have to show that $A_{p,q}$ indeed generates string x if and only if x can take P from (p, empty stack) to (q, empty stack).

The two directions of this proof use induction. This is proved in Claim 2.30 and Claim 2.31

$\Longrightarrow$

Claim 2.30: If $A_{p,q}$ generates x, then x can take P from p with empty stack to q with empty stack.

$\Longleftarrow$

Claim 2.31: If x can bring P from p with empty stack to q with empty stack, then A_{pq} generates x.