Equivalence of CFG's and PDA's

A_{pq}

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 $A_{pp} \to \varepsilon$

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

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

Proof of Claim 2.30: Induction on the number of steps in the derivation of x.

Base Case: A_{pq} generates x in one step. The only possible one step derivation is $A_{pp} \to \varepsilon$. Clearly ε takes the PDA P from state p to p with empty stack. Base case is true.

Suppose true for $k \geq 1$ steps.

let $A_{pq} \overset{*}{\Rightarrow} x$ with $k+1$ steps. The first step is either $A_{pq} \to a\, A_{rs}\, b$ or $A_{pq} \to A_{pr}\, A_{rq}$

Case 1: $A_{pq} \to a\, A_{rs}\, b \Rightarrow \ldots \Rightarrow \ldots \Rightarrow x$

This means that $x = a\,y\,b$ and $A_{rs} \overset{*}{\Rightarrow} y$ in k steps. This means y takes P from r to s on an empty stack, by induction.

where $y \in \Sigma^*$

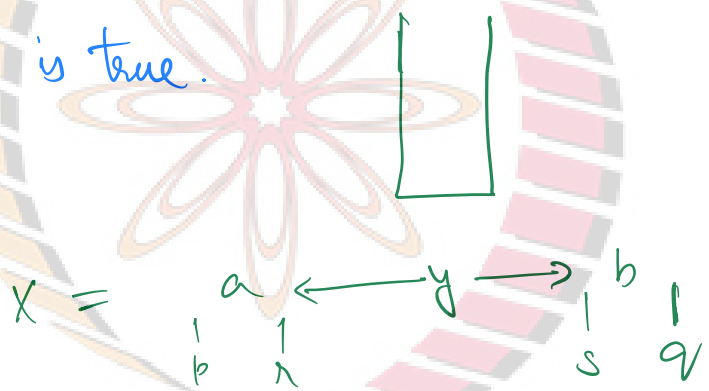
Since $A_{pq} \to a\, A_{rs}\, b$ is a rule, we have $(r,t) \in \delta(p,a,\varepsilon)$ and $(q,\varepsilon) \in \delta(s,b,t)$

⟶ a can take P from p to r and pushes t into the stack.

⟶ y takes P from r to s retaining the stack contents

⟶ b can take P from s to q, popping t from the stack.

So claim is true.



Case 2: $A_{pq} \to A_{pr} A_{rq}$

$\Rightarrow \ldots \Rightarrow yz = x$

This means $x = yz$, where

$$\left.\begin{aligned} A_{pr} &\overset{*}{\Rightarrow} y \\ A_{rq} &\overset{*}{\Rightarrow} z \end{aligned}\right\} \text{ both in } \leq k \text{ steps}$$

By induction, y takes P from p to r on an

empty stack. And z takes P from r to q on an empty stack. So x can take P from p to q on an empty stack.

So claim is true in this case as well.

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

Proof: Induction on the number of steps in the computation of P.

Base Case: The computation has zero steps.

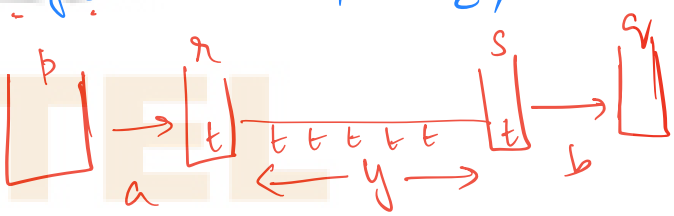
This means P has to start and end at the same state, say p. In zero steps, P can only process the empty string ε, so $x = \varepsilon$. We know that $A_{pp} \to \varepsilon$ is a rule for all states p.

Induction: Assume claim is true when P needs $\leq k$ steps. Let P have a computation where x brings P from p to q with empty stack using $k+1$ steps. We need to show that A_{pq} generates x. There are two cases.

Case 1: The stack never becomes empty. This means that the symbol pushed in step 1 is popped in step $k+1$. Say the symbol is $t. \in \Gamma$ let the first symbol read from the input be a and the last symbol read is $b. \in \Sigma_\varepsilon$

$\in \Sigma_\varepsilon$

This means that $x = a\,y\,b$ where $a, b \in \Sigma_\varepsilon$, and $y \in \Sigma^*$



$\longrightarrow$ a takes P from p to r by pushing t

$\longrightarrow$ b takes P from s to q by popping t.

This implies that y takes P from r to s retaining the stack in $k-1$ steps.

By induction, we have $A_{rs} \overset{*}{\Rightarrow} y$

By assumption, $(r,t) \in \delta(p,a,\varepsilon)$

and, $(q,\varepsilon) \in \delta(s,b,t)$

These two mean that $A_{pq} \to a\, A_{rs}\, b$ is a rule.

$A_{rs} \overset{*}{\Rightarrow} y$, so $a\, A_{rs}\, b \overset{*}{\Rightarrow} a\, y\, b$

Since $x = a\,y\,b$, this means $A_{pq} \overset{*}{\Rightarrow} x$.

Case 2: The stack empties in the middle. Let the stack become empty when P is at the state r.

Let y be the substring that brings P from p to r in $l \leq k$ steps with empty stack.

Let z be the substring that brings P from r to s in $k+1-l \leq k$ steps, again with empty stack.

We have $x = yz$, where by induction

$A_{pr} \overset{*}{\Rightarrow} y$ and $A_{rq} \overset{*}{\Rightarrow} z$

$A_{pq} \rightarrow A_{pr} A_{rq} \overset{*}{\Rightarrow} yz = x$.

Thus $A_{pq} \overset{*}{\Rightarrow} x$.

Theorem: A language is Context-Free if and only if it is recognized by a PDA.