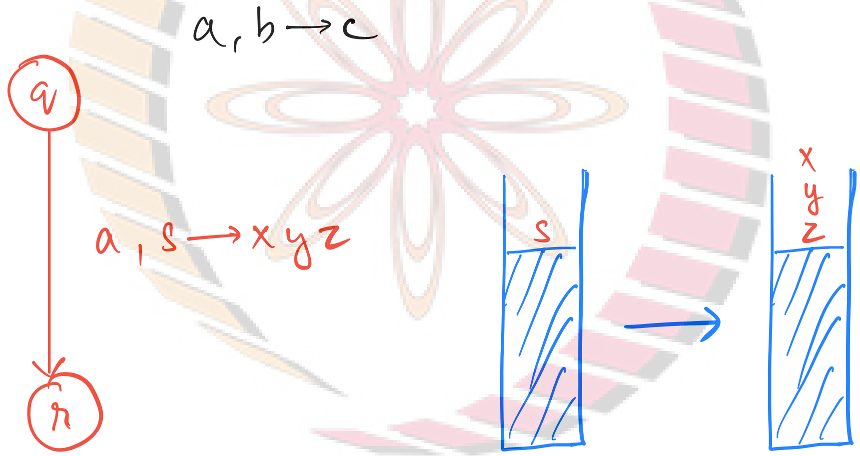
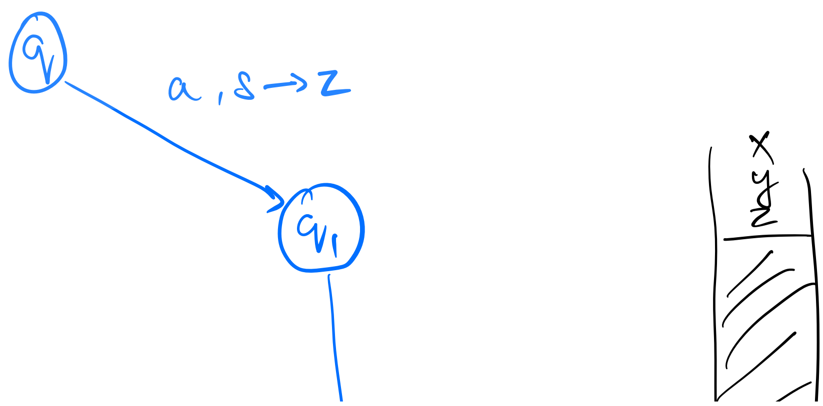
Equivalence of CFG and PDA's

We have seen two models - CFG's and PDA's.
We will now see that they have the same computational power.

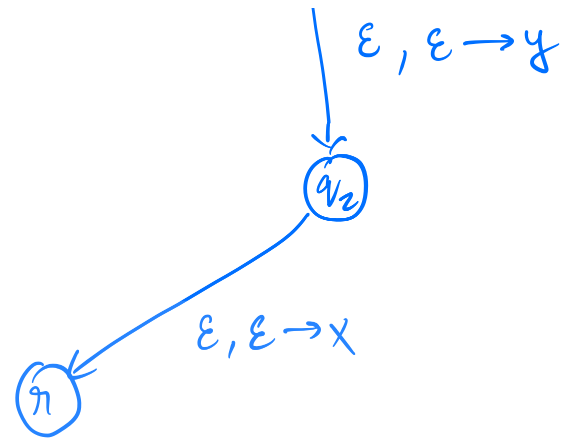
First, we define a shorthand notation.



We accomplish this in the following way.



$a, S \to xyz$



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

CF $\Longrightarrow$ PDA

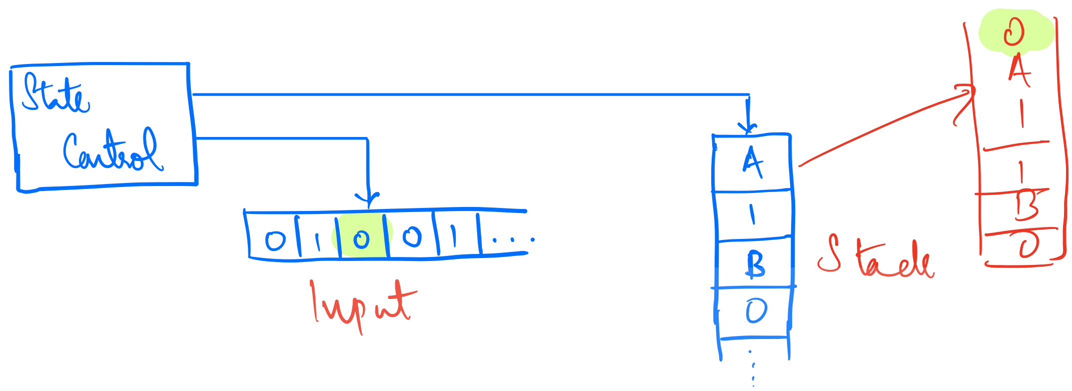
PDA $\Longrightarrow$ CF

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

Proof: Suppose there is a language A which is generated by a CFG G. We will construct a PDA P for A.

P accepts w $\Longleftrightarrow$ w is generated by G.

$A \to 0A1 \to 00B1 \to 00A11$



The PDA can access only the top of the stack. So it cannot apply production rules to the intermediate symbols.

1. Keep $ in the stack at the beginning, followed by the start variable.

2. Repeat (a) If top = variable, choose a substitution rule non deterministically and replace

   (b) If top = terminal, pop off and verify that the next symbol of the input is the same. If yes, advance.

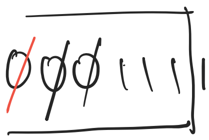
   (c) If top = $, move to accept state.

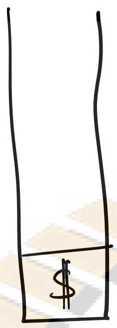
$\{0^n 1^n \mid n \geq 0\}$

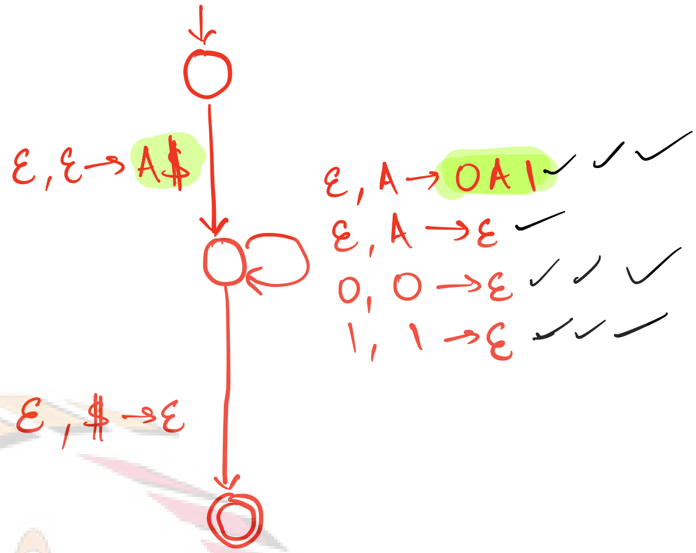
Grammar:

$A \to 0A1$
$A \to \varepsilon$

0101







More formally, $P = (Q, \Sigma, \Gamma, \delta, q_{start}, F)$

$Q = \{q_{start}, q_{loop}, q_{accept}\} \cup E$

additional states required to implement the shorthand notation.

$\delta(q_{start}, \varepsilon, \varepsilon) = \{(q_{loop}, S\$)\}$

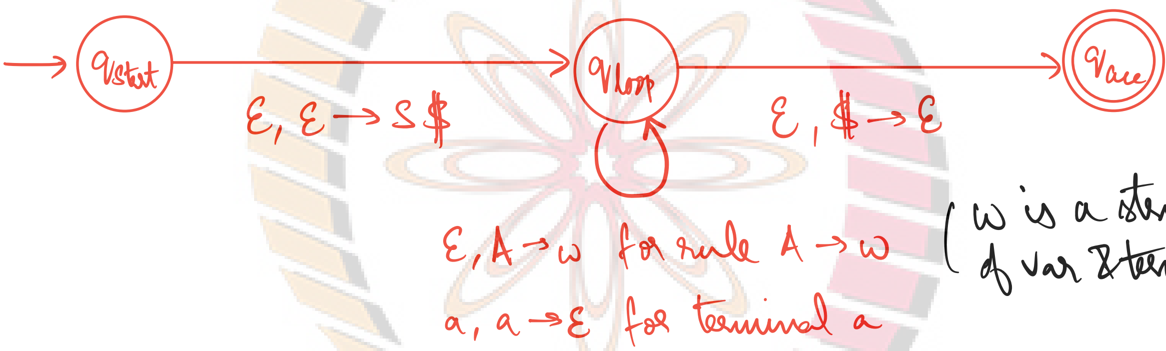
$\delta(q_{loop}, \varepsilon, A) = \{(q_{loop}, w) \mid \text{where } A \to w \text{ is a rule of } G\}$

For all variables A in G

$\delta(q_{loop}, a, a) = \{(q_{loop}, \varepsilon)\}$

For all terminals
a in G

$\delta(q_{loop}, \varepsilon, \$) = \{(q_{accept}, \varepsilon)\}$



(w is a string of var & terminals)

Exercise : Read Example 2.25