Theory of Computation

Introduced By:

Prof. Safia Abbas

Introduced By: Prof. Safia Abbas

Text Books:

"***An Introduction to the theory of computation***" by Michael Sipser, 2nd Edition, PWS Publishing Company, Inc. 2006; ISBN: 0-534-95097-3

"***An Introduction to Formal Languages & Automata***" by Peter Linz, 5th Edition, Jones & Bartlett Publishers, Inc. January 2012; ISBN:978-1-4496-1552-9 .

Agenda

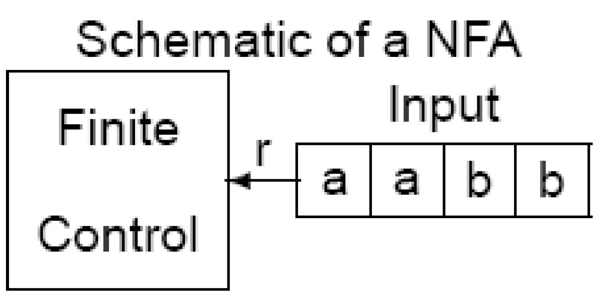
- How to identify that the language is CFL?
- Push Down Automaton?
- Greibach Normal Form
- Relation between CFG and PDA

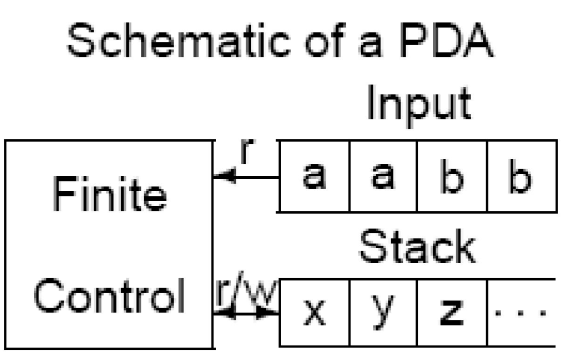
Pushdown Automata

- A pushdown automata recognizes the context free language.
- the CFG like the regular expression and the PDA like the FA.
  - PDAs are like FAs but have an extra component called a *stack* which provides additional memory.
- The stack allows PDA to recognize some non-regular languages.

For proving that a language is context-free:

1. construct a CFG that generates the language or
2. construct a PDA that recognizes the language



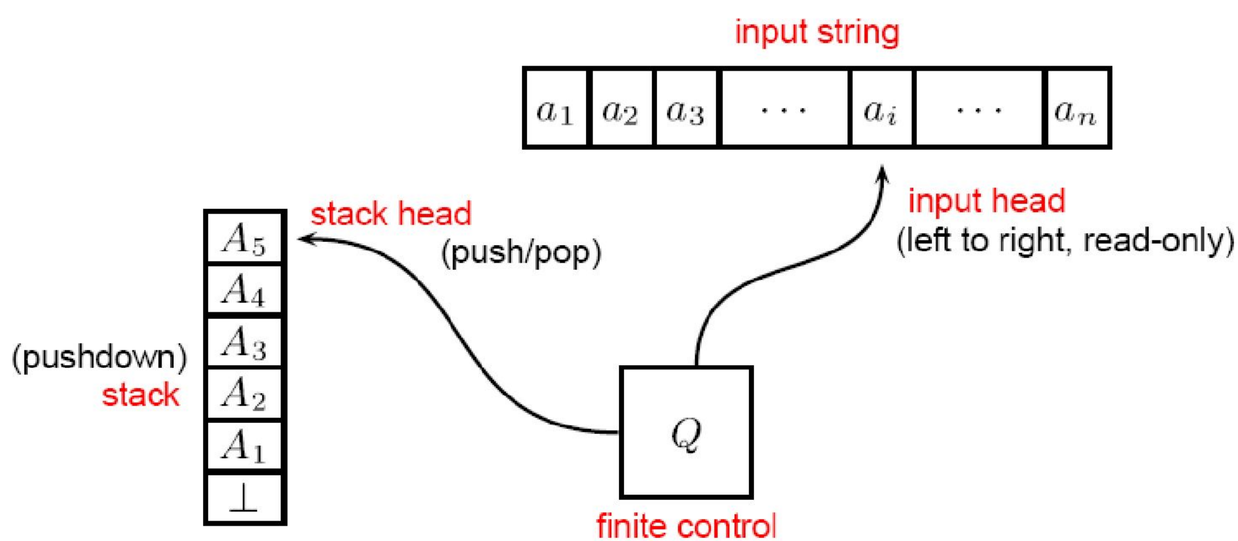


Pushdown Automata

- As with finite automata, there are deterministic and nondeterministic PDA.
- However, deterministic and nondeterministic PDA are not equivalent.
- Some CFLs are only recognizable in the presence of nondeterminism.
- We will consider nondeterministic PDA since they are equivalent to CFGs.
- Deterministic PDA are weaker.

Pushdown Automata

- A (non-deterministic) pushdown automaton is like an NFA, except it has a *stack* (pushdown store) for recording a potentially unbounded amount of information, in a last-in-first-out (LIFO) fashion



PDA Transitions

At any point, the PDA is

1. in a certain state,
2. scanning a certain symbol on the input tape, and
3. with a certain top-of-the-stack symbol.

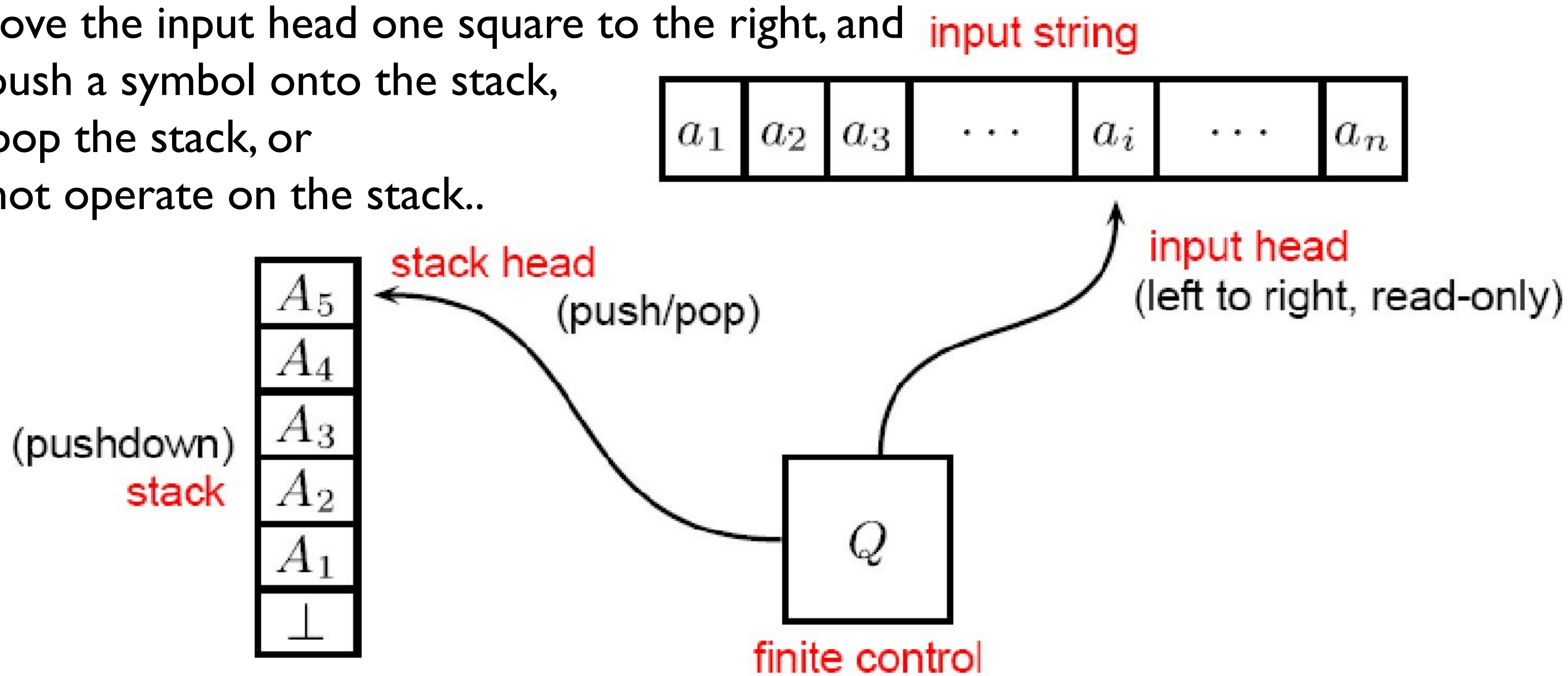
Depending on that the PDA would

1. enter a new state,
2. move the input head one square to the right, and

(a) push a symbol onto the stack,

(b) pop the stack, or

(c) not operate on the stack..

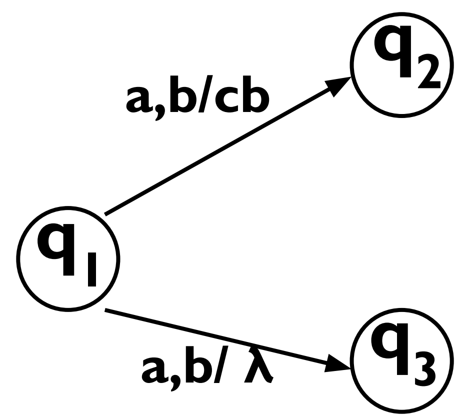


Formal Definition of PDA

A PDA is a 7-tuple $(Q, \Sigma, \Gamma, \delta, q_0, z, F)$, where

- Q is a finite set of states,
- Σ is the input alphabet,
- Γ is the stack alphabet,
- $\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow$ finite set of $Q \times \Gamma^*$,
- $q0 \in Q$ is the start state, and
- $F \subseteq Q$ is the set of accept states.
- $z \in \Gamma$ is the stack start symbol

Example : $\delta(q1, a, b) = \{(q2, cb), (q3, \lambda)\}$



Examples

$Q=\{q_0, q_1, q_2, q_3\}$, $\Sigma=\{a, b\}$, $\Gamma=\{0, 1\}$, $q0$, $z=0$, $F=\{q_3\}$

$\delta(q_0, a, 0) = \{(q_1, 10), (q_3, \lambda)\}$,

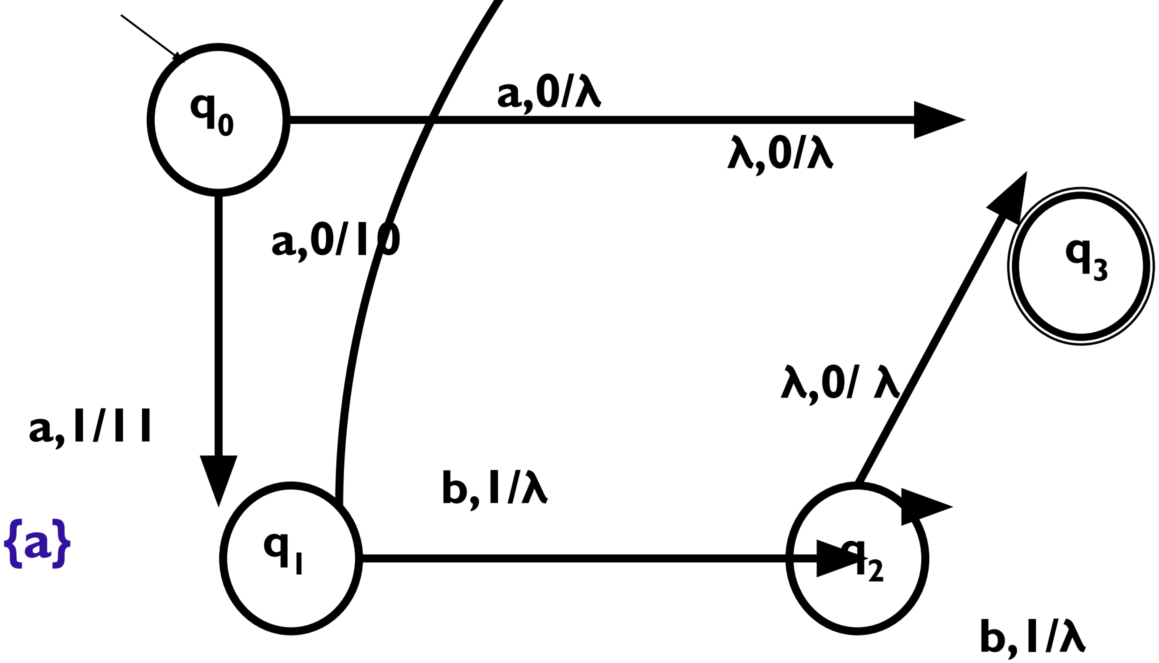
$\delta(q_0, \lambda, 0) = \{(q_3, \lambda)\}$

$\delta(q_1, a, 1) = \{(q_1, 11)\}$

$\delta(q_1, b, 1) = \{(q_2, \lambda)\}$

$\delta(q_2, b, 1) = \{(q_2, \lambda)\}$

$\delta(q_2, \lambda, 0) = \{(q_3, \lambda)\}$



$L = \{a^n b^n : n >= 0\} \cup \{a\}$

let us trace the string aaabbb and aaabb

The Language of a PDA

Let M be a nondeterministic pushdown automata,

A PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$, accepts a string w if

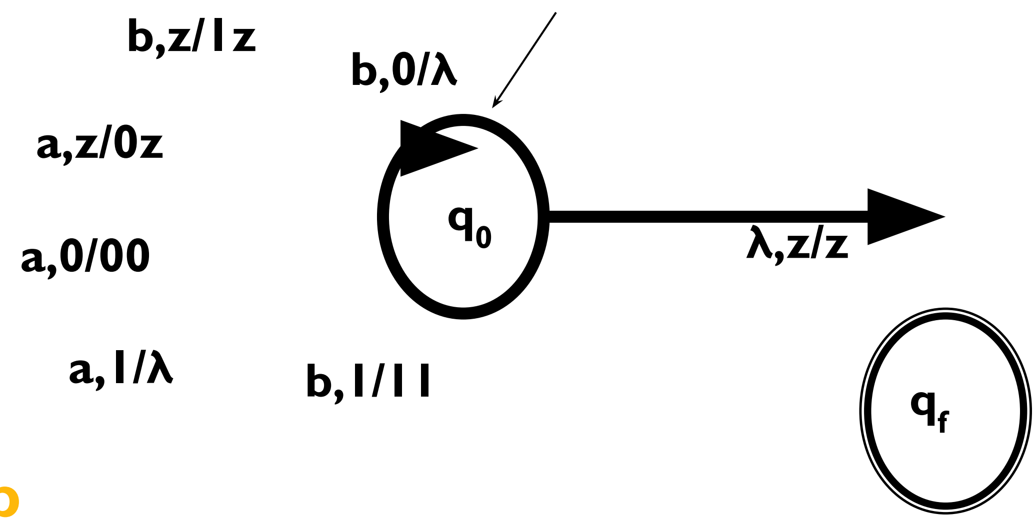
$$L(M) = \{w \in \Sigma^* : \delta^*(q_0, w, z) \vdash^*_M (q_f, \lambda, u),$$
$$q_f \in F, u \in \Gamma^*\}$$

Examples

Construct an NPDA for $L=\{w \in \{a, b\}^*: n_a(w)=n_b(w)\}$

$M = (\{q_0, q_f\}, \{a, b\}, \{0, 1, z\}, \delta, q_0, z, \{q_f\})$

$\delta(q_0, \lambda, z) = \{(q_f, z)\}$
$\delta(q_0, a, z) = \{(q_0, 0z)\}$
$\delta(q_0, b, z) = \{(q_0, 1z)\}$
$\delta(q_0, a, 0) = \{(q_0, 00)\}$
$\delta(q_0, b, 0) = \{(q_0, \lambda)\}$
$\delta(q_0, a, 1) = \{(q_0, \lambda)\}$
$\delta(q_0, b, 1) = \{(q_0, 11)\}$



Example: let w=baab

$(q0, baab, z) \vdash (q0, aab, 1z) \vdash (q0, ab, z) \vdash (q0, b, 0z) \vdash (q0, \lambda, z)$

Examples

$L=\{ww^R: w \in \{a, b\}^*\}$ $M = (\{q_0, q_1, q_2\}, \{a, b\}, \{a, b, z\}, \delta, q_0, \{q_2\})$

push in the stack

$\delta(q_0, a, a) = \{(q_0, aa)\}$

$\delta(q_0, b, a) = \{(q_0, ba)\}$

$\delta(q_0, a, b) = \{(q_0, ab)\}$

$\delta(q_0, b, b) = \{(q_0, bb)\}$

$\delta(q_0, a, z) = \{(q_0, az)\}$

$\delta(q_0, b, z) = \{(q_0, bz)\}$

guess middle

$\delta(q_0, \lambda, a) = \{(q_1, a)\}$

$\delta(q_0, \lambda, b) = \{(q_1, b)\}$

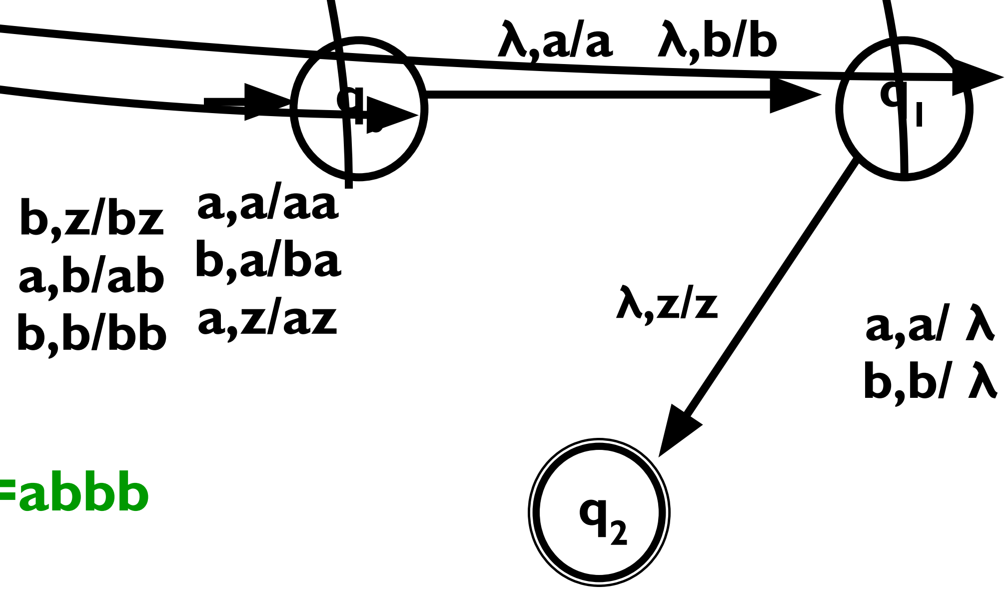
match w^R

$\delta(q_1, a, a) = \{(q_1, \lambda)\}$

$\delta(q_1, b, b) = \{(q_1, \lambda)\}$

successful match

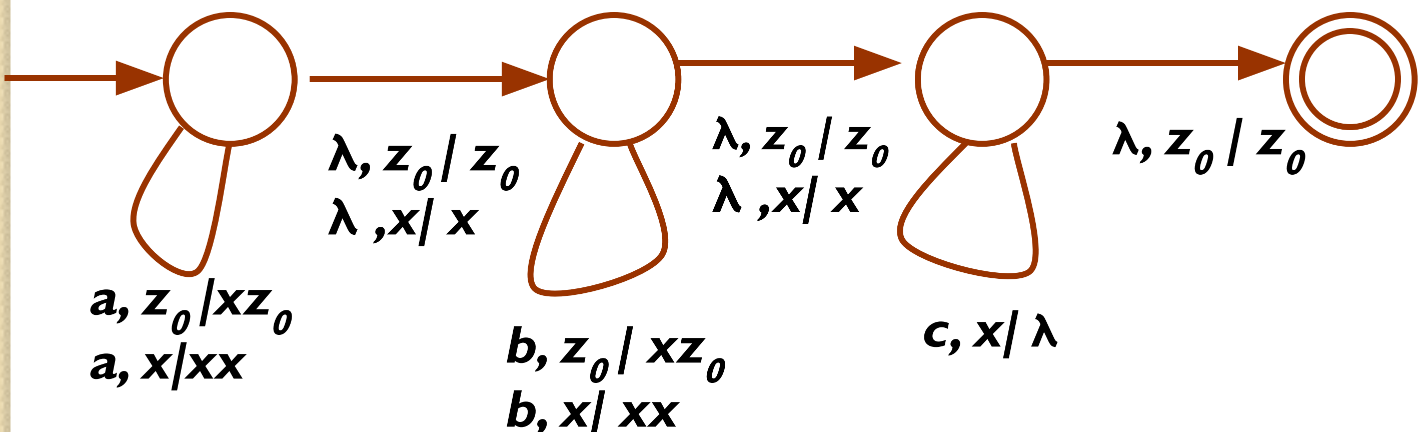
$\delta(q_1, \lambda, z) = \{(q_2, z)\}$



Let us trace the w=abba and w=abbb

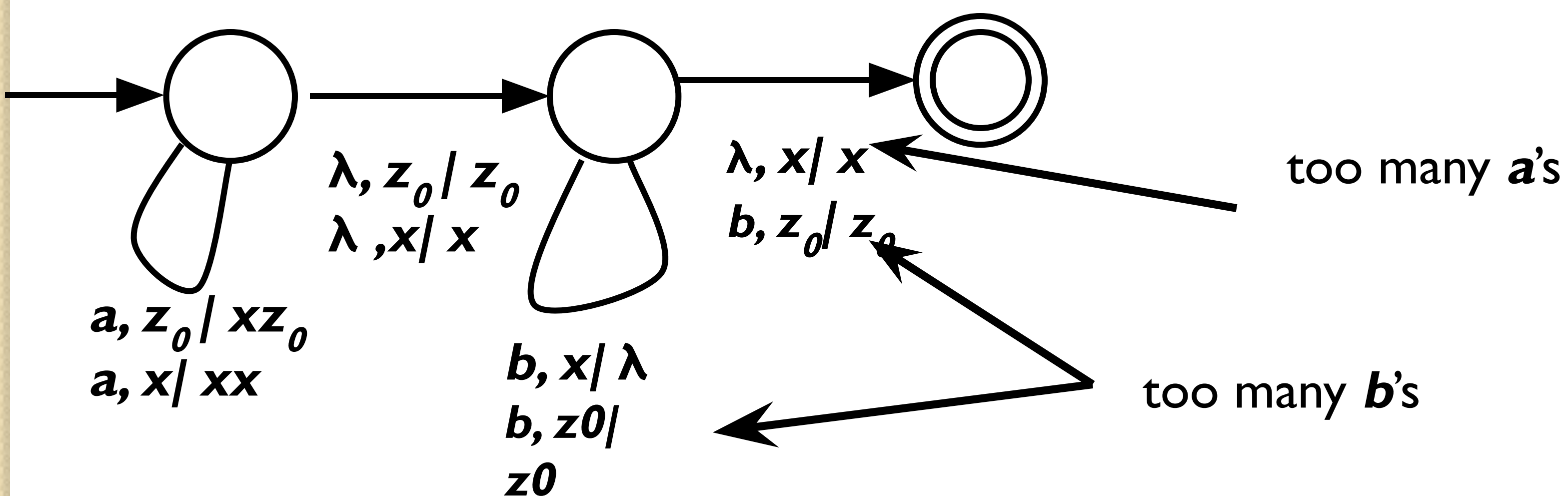
Context-free Languages

- ***Review***
  - Draw the Pushdown automata for
    - The language $\{a^n b^m c^{n+m}\}$



Context-free Languages

- **Review**
  - Pushdown automata
    - The language $\{a^n b^m \mid n \neq m\}$



Greibach Normal Form

Definition: A context-free grammar is in Greibach normal form if all productions are of the form $A \rightarrow ax$, where $a \in T$ and $x \in N^*$.

Example : $S \rightarrow AB, A \rightarrow aA|bB|b, B \rightarrow b$ is not GNF,

$S \rightarrow aAB|\ bBB|\ bB, A \rightarrow aA|bB|b, B \rightarrow b$ is GNF

Example: Convert $S \rightarrow abSb|aa$ into GNF.

$S \rightarrow aBSB|\ aA, A \rightarrow a, B \rightarrow b$

Theorem: Any context-free grammar with no λ in $L(G)$ has an equivalent grammar G` in GN form.

Question

- Prove or disprove the regularity of the language
  - $L=\{a^{n+1}b^{2n}:n>=0\}$
- Find the context free grammar that describes L.
- Find the NPDA that recognizes L.

- Since we can get the CFG it is easier to draw the PDA based on this grammar.

Theorem

- A language is context free if and only if some pushdown automaton recognizes it.
- We will focus on converting from CFG to PDA

- How can we draw NPDA from CFG???