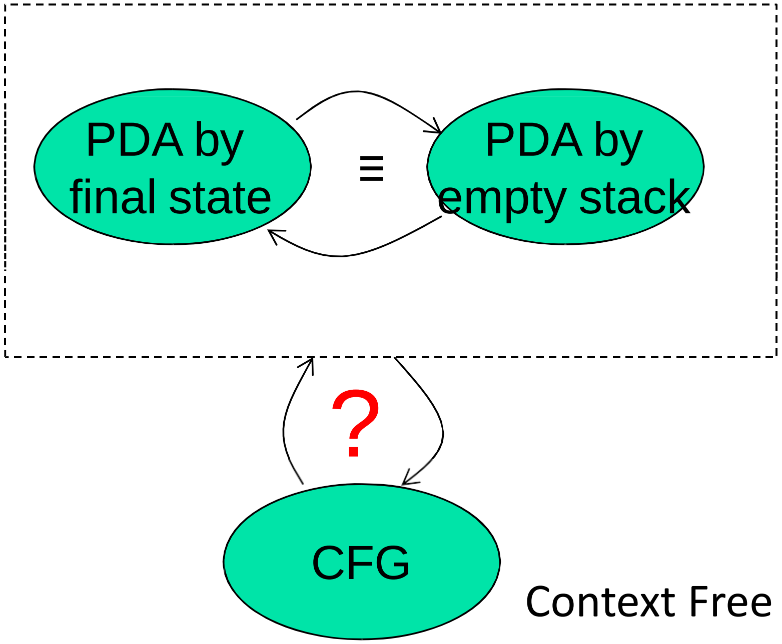
Designing PDA for given CFG

CFG ←→ PDA ←→ CFL



Context Free Grammar

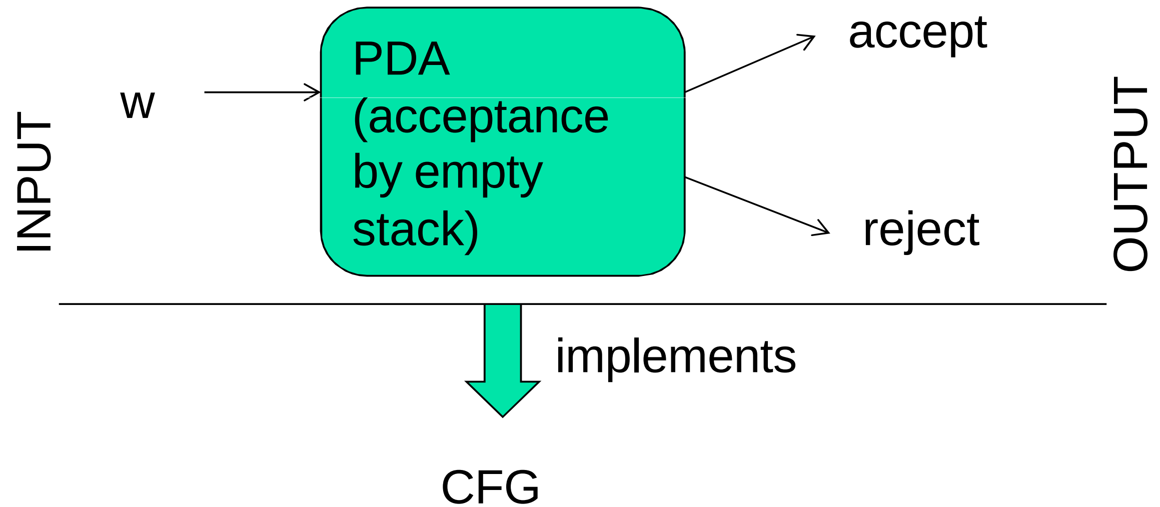
Q) Does a PDA that accepts by <u>empty stack</u> need any final state specified in the design?

Ans: **No**, there is no need of final state, though the string is accepted in such cases

i.e. A= Φ

Converting CFG to PDA

Main idea: The PDA simulates the ***leftmost derivation*** *of grammar tree* on a given w, and upon consuming w fully it either arrives at acceptance (by empty stack) or non-acceptance.



Converting a CFG into a PDA

Main idea: The PDA simulates the leftmost derivation on a given w, and upon consuming it fully it either arrives at acceptance (by empty stack) or non-acceptance.

Steps:

1. Push the right hand side of the production onto the stack, with leftmost symbol at the stack top
2. If stack top is the leftmost variable, then replace it by all its productions (each possible substitution will represent a *distinct* path taken by the non-deterministic PDA)
3. If stack top has a terminal symbol, and if it matches with the next symbol in the input string, then pop it

State is inconsequential (only one state is needed)

Design of Transition function δ:

Rule1:

$\delta(q, \wedge, A) = \{(q, \alpha) \mid A \rightarrow \alpha \text{ is in } P\}$

for every $A \in V_n$

Rule2:

$\delta(q, a, a) = \{(q, \wedge)\}$ for every $a \in \Sigma$

PDA Design

Input: CFG $(Vn, \sum, P, S)$

Process: Expanding CFG productions for stack operations

Output: PDA accepting CFL by empty stack approach

$T_{empty\_stack} = (\{q\}, \sum, q, \delta, \emptyset, (Vn \cup \sum), S)$

Example: CFG to PDA

- G = ({S,A}, {0,1}, P, S)
- P:
  - S → AS | ε
  - A →0A1 | A1 | 01
- PDA = ({q}, {0,1}, {0,1,A,S}, δ, q, S, Φ)
- δ:
  - δ(q, ε , S) = {(q, AS), (q, ε)}
  - δ(q, ε , A) = { (q,0A1), (q,A1), (q,01) }
  - δ(q, 0, 0) = { (q, ε) }
  - δ(q, 1, 1) = { (q, ε) }

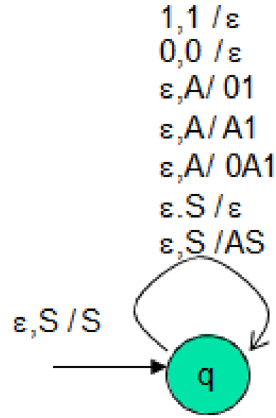
How will this new PDA work?

Lets simulate string <u>0011</u>

Simulating string 0011

PDA (δ):

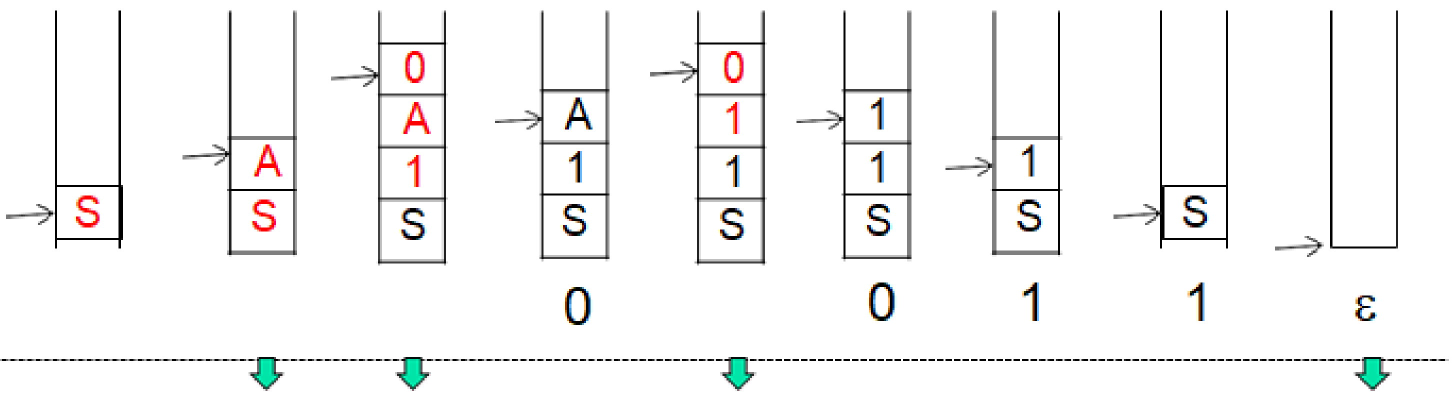
$\delta(q, \varepsilon, S) = \{ (q, AS), (q, \varepsilon) \}$
$\delta(q, \varepsilon, A) = \{ (q,0A1), (q,A1), (q,01) \}$
$\delta(q, 0, 0) = \{ (q, \varepsilon) \}$
$\delta(q, 1, 1) = \{ (q, \varepsilon) \}$



Leftmost deriv.:

$S \rightarrow AS$
$\rightarrow 0A1S$
$\rightarrow 0011S$
$\rightarrow 0011$

Stack moves (shows only the successful path):



Accept by empty stack

$S \rightarrow AS \rightarrow 0A1S \rightarrow 0011S \rightarrow 0011$

Ex1.

Q) Construct a PDA equivalent to following CFG:

$S \rightarrow 0BB$, $B \rightarrow 0S|1S|0$

Test if string $w=010^4$ is recognized by the PDA ?

$G \equiv \{S \rightarrow 0BB,\ B \rightarrow 0S \mid 1S \mid 0\}$

Let $T = (\{q\}, \{0,1\}, q, \{S, B, 0, 1\}, \delta, S, \emptyset)$

$\delta(q, \wedge, S) = \{(q, 0BB)\}$

$\delta(q, \wedge, B) = \{\ (q, 0S), (q, 1S), (q, 0)\ \}$

$\delta(q, 0, 0) = \{(q, \wedge)\}$

$\delta(q, 1, 1) = \{(q, \wedge)\}$

$(q, 010^4, S) \vdash (q, 010^4, 0BB) \vdash (q, 10^4, BB)$

$\vdash (q, 10^4, 1SB) \vdash (q, 0^4, SB) \vdash (q, 0^4, 0BBB)$

$\vdash (q, 0^3, BBB) \vdash (q, 0^3, 0BB)$ .... $\vdash^* (q, \wedge, \wedge)$

➔ Accepted

Ex2.

Q-i) Generate a grammar over $\sum = \{a, b\}$ to accept all words containing equal number of a’s and b’s

Q-ii)Construct a PDA equivalent to the designed CFG

iii) IF designed PDA is Deterministic or Non-deterministic ?

iv) Test if string w= abbaab is recognized by the PDA ?

Ex2.

$G \equiv S \rightarrow aSbS \mid bSaS \mid \wedge$

Ex2.

$G \equiv S \rightarrow aSbS \mid bSaS \mid \wedge$

$\delta(q, \wedge, S) = \{(q, aSbS), (q, bSaS), (q, \wedge)\}$

$\delta(q, a, a) = \{(q, \wedge)\}$

$\delta(q, b, b) = \{(q, \wedge)\}$

$T = (\{q\}, \{a,b\}, q, \{S, a, b\}, \delta, S, \emptyset)$

$\rightarrow$ Nondeterministic

$(q, abbaab, S) \vdash (q, abbaab, aSbS) \vdash$

$(q, bbaab, SbS) \vdash \ldots\ldots$