CS & IT ENGINEERING

Theory of Computation

Push Down Automata:

PDA – Part 1

Lecture No. 02

By- DEVA Sir

01 Push Down Automata

02

03

04

05

Push Down Automata

Every Reg lang is CFL

But CFL may or may not be Regular

$L = a^*$ over $\Sigma = \{a, b\}$

Regular lang

CFL

Regular Language $\cong$ FA $\equiv$ RG $\equiv$ Reg Exp

CFL $\cong$ PDA $\cong$ CFG

$L = a^n b^n$

↳ Not reg

↳ CFL

Summary

FA < PDA

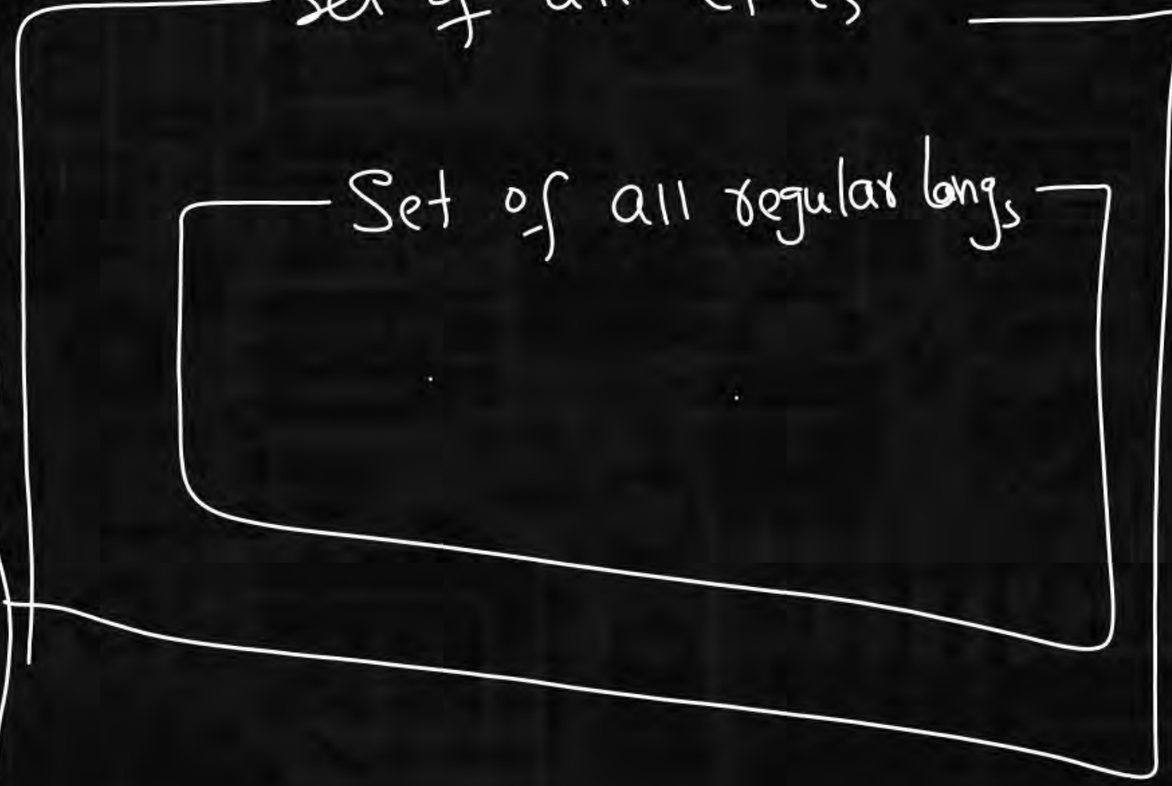
Set of L(FAs) ⊂ Set of L(PDAs)

Set of all regs ⊂ Set of all CFLs

Note:

1) Reg lang need not be subset another reg

2) Reg lang need not be subset of CFL



$(a+b)^* \not\subset a^n b^n$

$a^* \not\subset b^*$

Push Down Automata

PDA ≅ FA + 1 stack

$(Q, \Sigma, \delta, q_0, \underbrace{F}_{\text{optional}}, Z_0, \Gamma)$

- Z_0 → Bottom of Stack (⊥)
- Γ → Stack Alphabet (Set of stack symbols)

I) Uses Final state mechanism to accept string

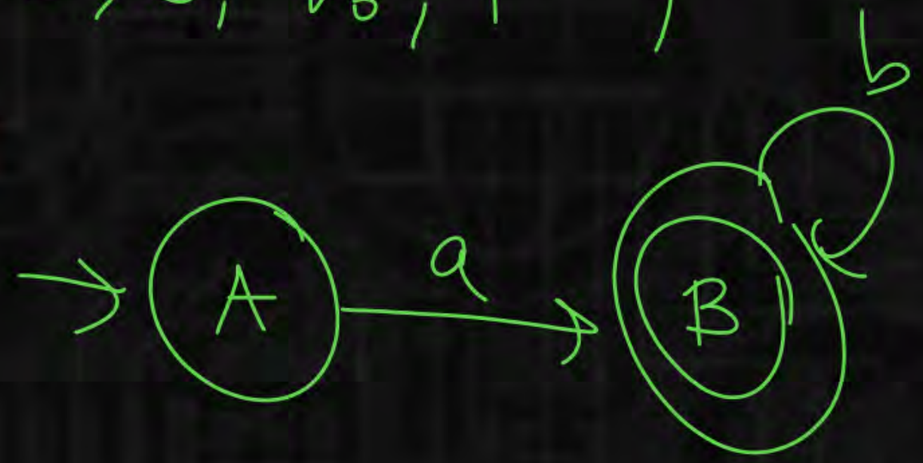
II) Uses Empty stack mechanism to accept string

III) Uses Both Final state & Empty stack

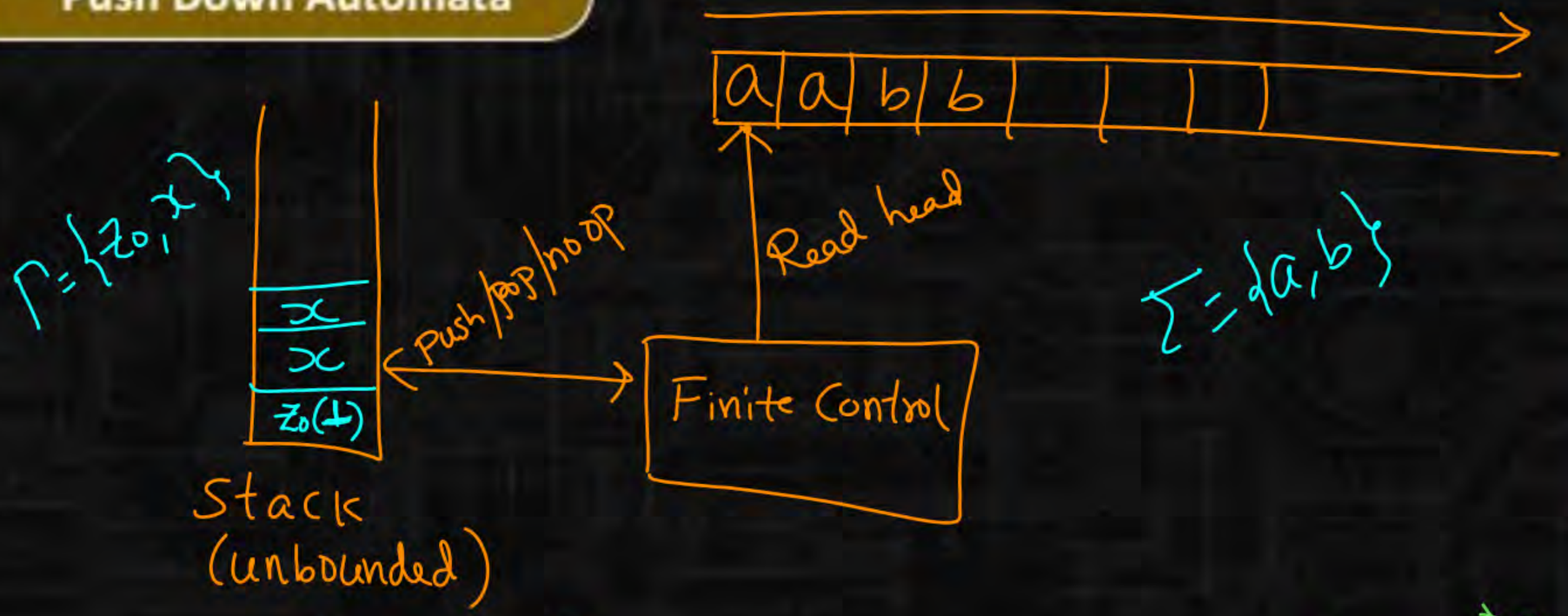
FA

↓ Uses Final state mechanism

$(Q, \Sigma, \delta, q_0, F)$



Push Down Automata



$$\delta_{NPDA} = \delta_{PDA} : Q \times \Sigma_\varepsilon \times \Gamma^* \rightarrow 2^{Q \times \Gamma^*}$$

$$\delta_{DPDA} : Q \times \Sigma \times \Gamma \rightarrow Q \times \Gamma^*$$

Note:

I) Every DPDA is PDA

II) PDA may or may not be DPDA

Push Down Automata

→ By default PDA is non-deterministic m/c

Note:

less powerful

(NFA = DFA) < DPDA < PDA

more powerful

DPDA ⇓ represents DCFLs

PDA ‖ represents CFLs

Note: Every DCFL is CFL

CFL need not be DCFL

PDA stack is restricted to 100 Symbols height

≅

FA

≅

PDA with finite stack

L(NFA) is Regular

L(DFA) is Regular

Every NFA convertible to DFA

Every DFA is NFA

NFA ≅ DFA

Power of NFA & DFA is same

Push Down Automata

DPDA

Uses Final state mechanism to accept Strings

PDA

final state will decide $w \in L$ or $w \notin L$

Stack will decide $w \in L$ or $w \notin L$

Uses Final state $\simeq$ Uses Empty stack

L_1 = Set of all languages accepted by PDA using Final State

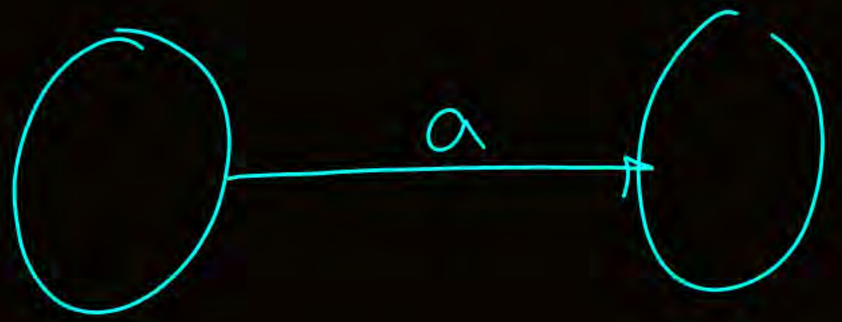
L_2 = Set of all languages accepted by PDA using Empty Stack

$L_1 = L_2$ = Set of all CFLs

State change

Trying to remember a

In FA



I) Change state

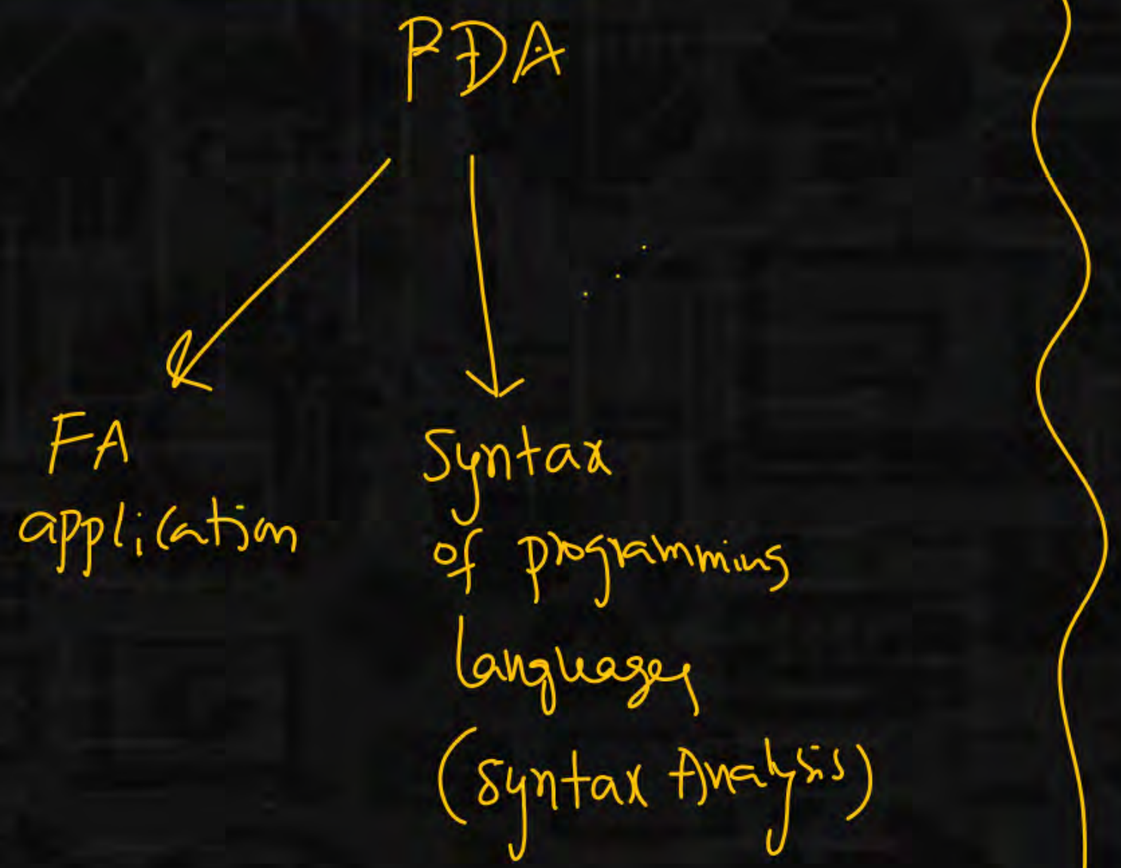
OR

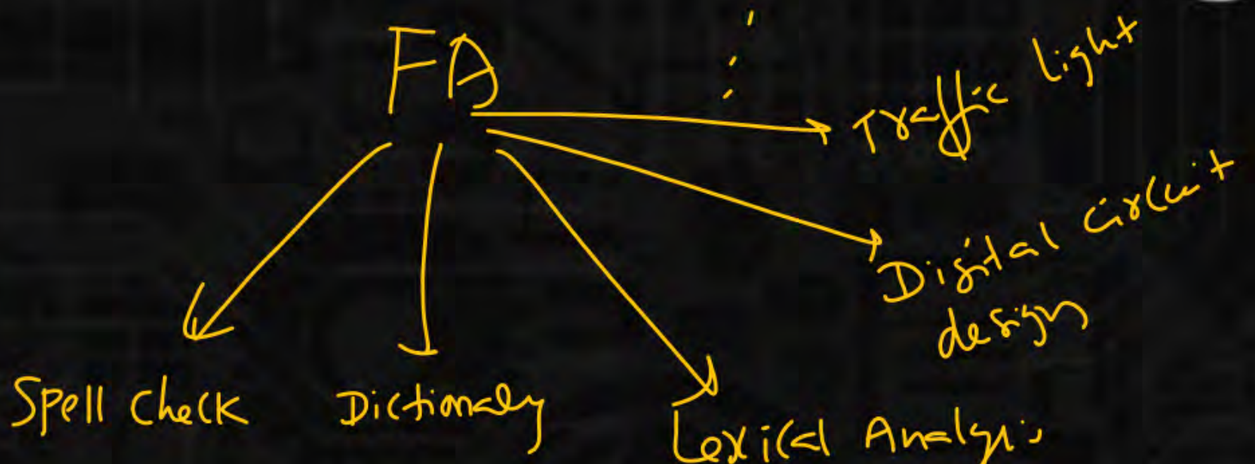
II) push onto stack

OR

III) Both I & II

Push Down Automata





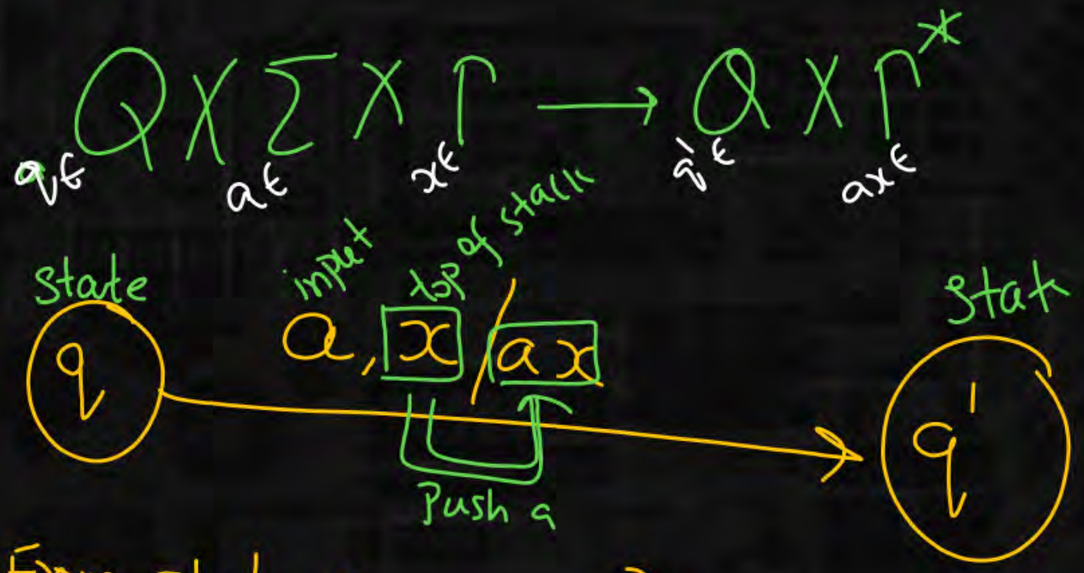
Push Down Automata

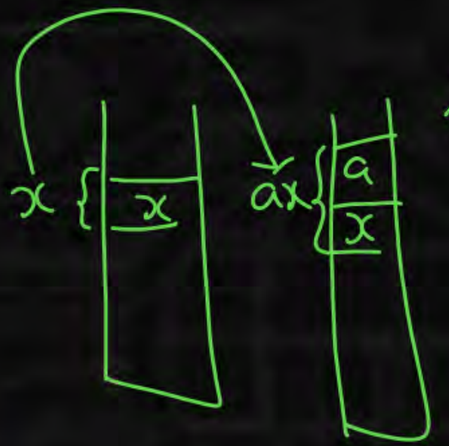
How to understand transition ?

DPDA transition
PDA transition

(1) $\delta(q, a, x) = (q', ax)$

$q \in Q \quad a \in \Sigma \quad x \in \Gamma \longrightarrow q' \in Q \quad ax \in \Gamma^*$



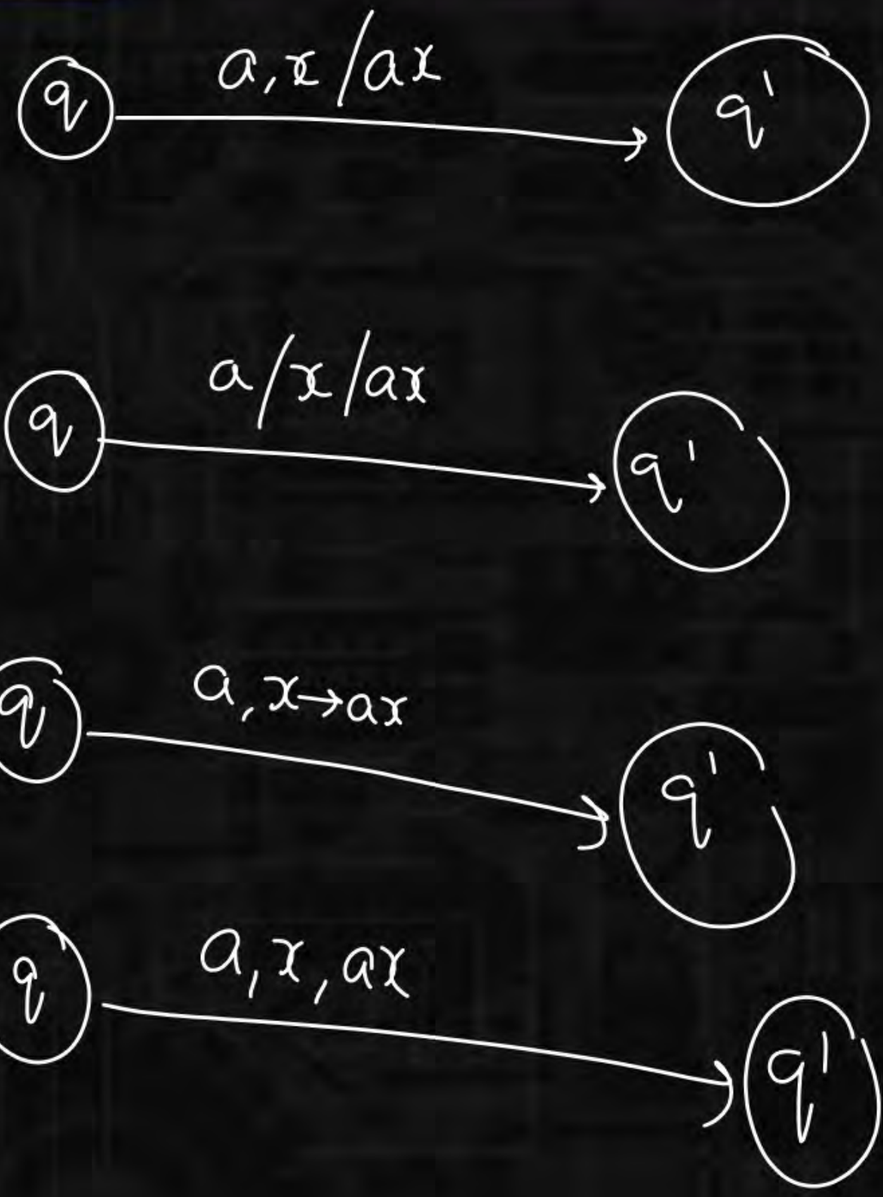


From State q
By reading input a
When top of stack x
⟹ going to q'
Push a onto stack

1st Every Transition ✓
2nd PDA ✓
3rd Strings ε a b aa ✓
4th Language ✓

Push Down Automata

How to understand PDA ?



all notations represent same meaning

$a \in \Sigma$

$x \in \Gamma$

$ax \in \Gamma^*$

Push Down Automata

② $\delta(q, a, a) = (q', a)$

present state, current i/p, top of stack

no change

③ $\delta(q, a, a) = (q', aa)$

i/p ε, ε

push a

i) $q \rightarrow q'$

ii) Input 'a'

iii) tos is 'a'

iv) push a

Push Down Automata

only in PDA

④ $\delta(q, \varepsilon, \boxed{\varepsilon}) = (q, \boxed{\varepsilon})$

no change

i/p tos

By reading no i/p

Without looking at stack

$\Rightarrow$ no operation on stack.

Push Down Automata

		State change	i/p	tos	operation
PDA	⑤ $\delta(q, \varepsilon, \boxed{\varepsilon}) = (q', \boxed{\varepsilon})$	$q \Rightarrow q'$	no i/p	not looking	no operation
PDA	⑥ $\delta(q, \varepsilon, \boxed{\varepsilon}) = (q', \boxed{a})$	$q \Rightarrow q'$	no i/p	not looking	Push 'a'
PDA	⑦ $\delta(q, a, \boxed{\varepsilon}) = (q', \boxed{\varepsilon})$	$q \Rightarrow q'$	read 'a'	not looking	no operation
DPDA & PDA	⑧ $\delta(q, a, \boxed{a}) = (q', \boxed{\varepsilon})$	$q \Rightarrow q'$	read 'a'	tos = a	Pop 'a'
In PDA	⑨ $\delta(q, a, \boxed{aaa}) = (q', \boxed{a})$	$q \Rightarrow q'$	read 'a'	tos seq = aaa	Pop 2 a's

aa's deleted

$Q \times \Sigma_\varepsilon \times \Gamma^* \rightarrow 2^{Q \times \Gamma^*}$

r^* / r^*

① $\varepsilon/\varepsilon \Rightarrow$ no operation

② $\varepsilon/a \Rightarrow$ push a

③ $a/\varepsilon \Rightarrow$ Pop a

④ $aa/a \Rightarrow$ Pop a

⑤ $aaa/aaa \Rightarrow$ no operation

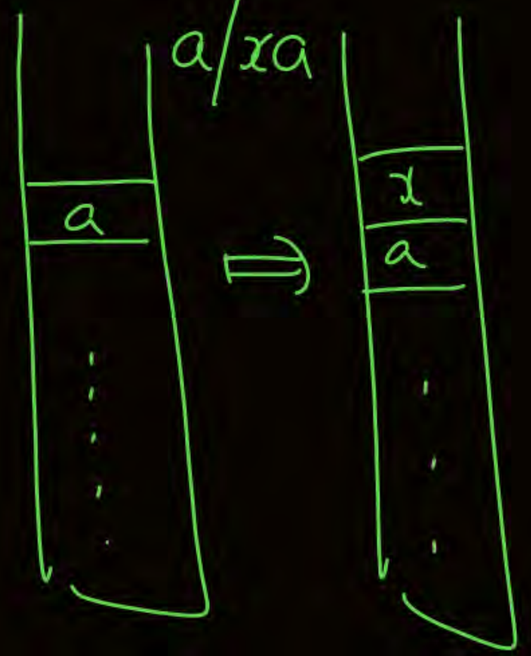
⑥ ~~a/x Invalid~~

⑦ $a/a \Rightarrow$ no operation

⑧ b/a Invalid

Compare r & r^*

operation



Push Down Automata

PDA (10) $\delta(q, a, X) = (q', aX)$: Push 'a'

PDA (11) $\delta(q, a, aX) = (q', X)$: Pop 'a'

PDA (12) $\delta(q, \varepsilon, aX) = (q', aaX)$: push 'a'

PDA & DPDA (13) $\delta(q, a, Z_0) = (q', aZ_0)$: push 'a'

PDA ∉ DPDA (14) $\delta(q, b, a) = (q', \varepsilon)$: Pop 'a'

Push Down Automata

DPDA: (δ) $\quad Q \times \Sigma \times \Gamma \longrightarrow Q \times \Gamma^*$

PDA: (δ) $\quad Q \times \Sigma_\varepsilon \times \Gamma^* \longrightarrow 2^{Q \times \Gamma^*}$

Summary

- PDA & DPDA : δ

Next: What is PDA ?

Construction of PDA & DPDA

Γ = Set of stack Symbols

Γ^* = Set of Strings over Γ


THANK YOU GW
SOLDIERS !