Name:- Prof. Anand Gharu
class:- TE COMPUTER
Page No.
Subject:- TOC
MB NO:- 8087777708

5. PUSHDOWN AUTOMATA:

PROF. ANAND GHARU
ASSISTANT PROFESSOR
PVGCOE NASHIK.
anandgharu.wordpress.com

* Introduction to Pushdown Automata (PDA)

→ Pushdown Automata (PDA) can be viewed as a finite automata with stack.

An added stack provides memory and increase capability g machine.

A pushdown Automata can do the following.

1. Read i/p symbol
2. Perform stack operation
   - push oper^n
   - POP oper^n
   - check empty condition g a stack thro. initial stack symbol
   - Read top symbol g stack without a pop.
3. make state change.

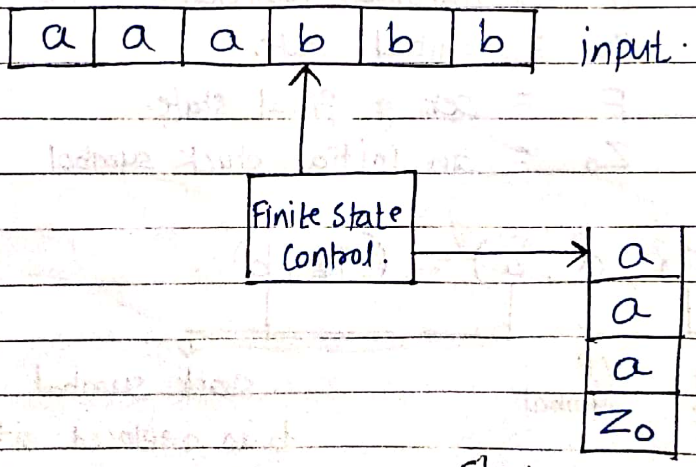
- PDA is more powerful than FA.
- A context free lang (CFL) can be recognized by PDA
- A Context free lang is a PDA lang.

Example :-

A string g the form $a^n b^n$ can't be handled by FA. But same can be handled by PDA.

$n = 3 \qquad a^3 b^3$

a a a b b b



Stack After Reading the 1st half g $a^3 b^3$

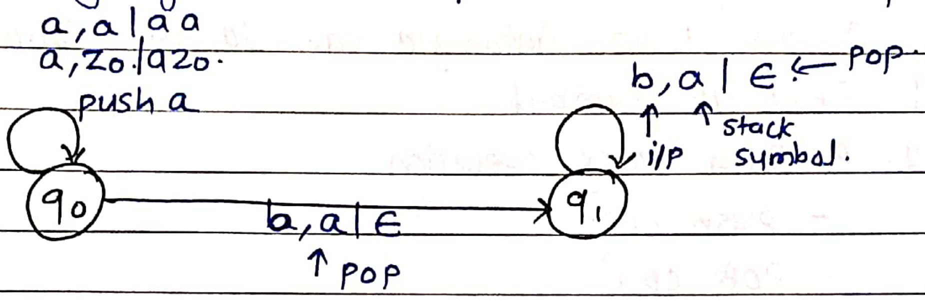
A PDA uses three stack operation :-

1) POP operation, - it removes the top symbol of stack.

2) PUSH operation - it insert symbol onto the top of stack.

3) NOP operation - it does nothing to stack.

The Language $\{a^n b^n \mid n \geq 1\}$ can be accepted by PDA.



PDA for $a^n b^n$.

* Definition of PDA.

→ A PDA can be defined as 7 tuple.

$$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

where,

Q = set of state

Σ = i/p alphabets

T = stack symbol

δ = transition function

q_0 = Initial state.

F = set of final state.

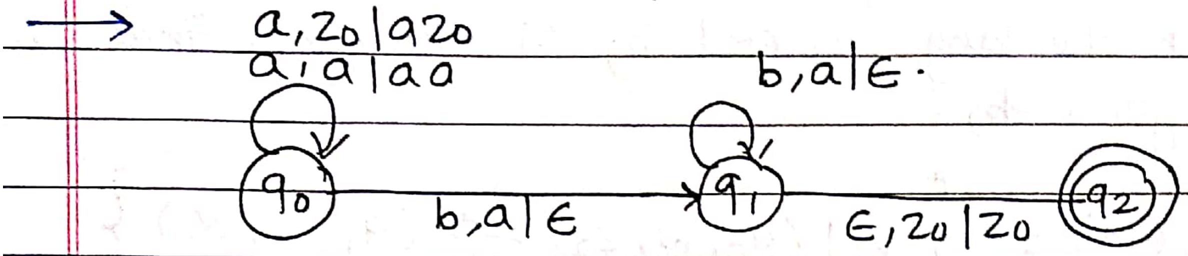
Z_0 = an initial stack symbol.

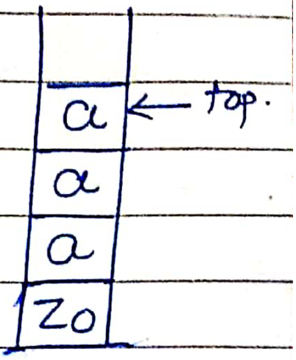
1) $\delta(\underbrace{q_1}_{\text{current state}}, \underbrace{a}_{\text{i/p symbol}}, b) = (q_2, b)$

stack symbol
b is replaced with b.
i.e. no stack oper.

* Example :-

$a^n b^n \mid n \geq 1$ aaabbb.





Transition Diagram.

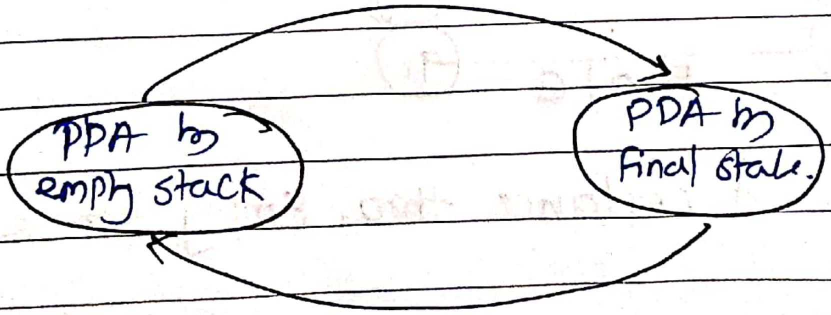
Using set of eqⁿ :- (transition Rules)

1) $\delta(q_0, a, Z_0) = (q_0, aZ_0)$
2) $\delta(q_0, a, a) = (q_0, aa)$
3) $\delta(q_0, b, a) = (q_1, \epsilon)$
4) $\delta(q_1, b, a) = (q_1, \epsilon)$
5) $\delta(q_1, \epsilon, Z_0) = (q_2, Z_0)$

* The Language of PDA :-

→ A Lang, L can be accepted by a PDA in two ways
1) Through final state
2) Through Empty stack.

It is possible to convert bet' two classes
1) From final state to empty stack.
2) From empty stack to final state.

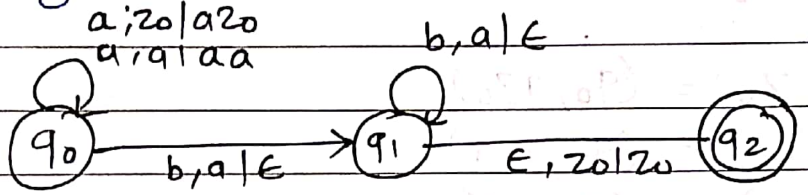


1) Acceptance by final state :-

Let the PDA, $M = \{Q, \Sigma, \Gamma, \delta, q_0, z_0, F\}$.
then the lang accepted by M. through a final state is given by.

$$L(M) = \left\{ w \mid (q_0, w, z_0) \overset{*}{\underset{M}{\Longrightarrow}} (q_1, \epsilon, \alpha) \right\}$$

where the state $q_1 \in F$.. α, the final contents of the stack are irrelevant. as a string accepted through a final state.



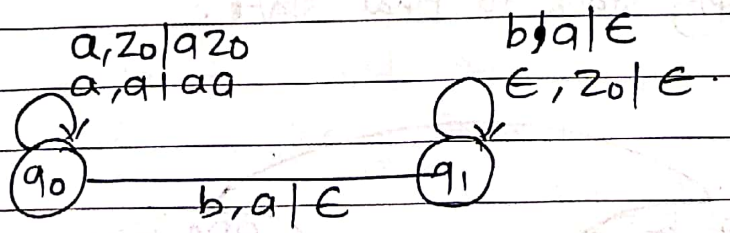
Acceptance thro. final state.

2) Acceptance by Empty stack :-

Let the PDA, $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, \phi\}$.
then the lang accepted through an empty stack is given by :

$$L(M) = \left\{ w \mid (q_0, w, z_0) \overset{*}{\underset{M}{\Longrightarrow}} (q_1, \epsilon, \epsilon) \right\}$$

Where, q_1 is any state belonging to Q. and the stack becomes empty on application of i/p string w.



Acceptance thro. Empty stack.

Note :-

1) Final state having more state — 1) Empty stack having less state.

2) ~~less~~ more time required — 2) less time required.

* Non-deterministic PDA :-

→ There are 2 types g PDA
1) Deterministic PDA (DPDA)
2) Non-deterministic PDA. (NPDA)

- In a DPDA, there is only one moves in every situati whereas ain NPDA, there are multiple moves under situatio
- DPDA is less powerful than NPDA
- Every context free lang. cann't be recognized by a DPDA but It can be recognised by NPDA.
- The class g lang. a DPDA can accept lies in between a regular lang. and CFL.
- A palindrom can be accepted by NPDA but. it cann't be accepted by a DPDA.

* Application g PDA :-

→ 1) PDA is machine for CFL
2) A string belonging to CFL can be recognized by PDA.
3) PDA is extensively used for parsing.
4) PDA is an abstract machine, it can also be used for giving proofs g lemma on CFL.
5) PDA can write symbol on stack.
6) PDA can Push or POP symbol on stack