

## Assignment-IV

Dharmajay Porwal  
EN19CS301110

1 Define Pushdown Automata. What are the different types of language accepted by a PDA (to define them)  
And Pushdown Automata is a finite Automata with extra memory called stack which helps pushdown Automata to recognize context free language.

PDA can be defined as -

- a)  $Q$  is the set of states.
- b)  $\Sigma$  is the set of input symbol.
- c)  $\Gamma$  is the set of pushdown symbols.
- d)  $q_0$  is the initial state.
- e)  $Z$  is the initial pushdown system.
- f)  $f$  is the set of final states.
- g)  $\delta$  is the transition function.

Pushdown Automata accepts context free languages  
A they can be accepted by following two approaches:

## i) Acceptance by final state

Let  $P = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$  be a PDA.

The language accepted by the final state can be defined as

$$L(PDA) = \{w / (q_0, w, Z) \vdash^* (P, \epsilon, \epsilon), q \in F\}$$

## ii) Acceptance by empty stack

Let  $P = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$  be a PDA.

The languages accepted by empty stack can be defined as

$$N(PDA) = \{w / (q_0, w, Z) \vdash^* (P, \epsilon, \epsilon), q_0 \in Q\}$$

## 2 what is CFG?

Ans CFG stands for Context free Grammar which is classified as Type-2 Grammar in Chomsky classification of grammar.

CFG can be defined by four tuples as -

$$G = (V, T, P, S)$$



where ,

$C_1 = C_{\text{irammese}}$

$T =$  A finite set of terminal symbols

$V =$  A finite set of non-terminal symbols

$P =$  Set of production rules.

$S =$  Starting symbol.

for ea -

$$L = \{w c w^R \mid w \in (a, b)^*\}$$

Production rules  $\rightarrow$

$$S \rightarrow a S a$$

$$S \rightarrow b S b$$

$$S \rightarrow c$$

Now check abbcbbba string can be derived from the given CFG.

$$S \Rightarrow a S a$$

$$S \Rightarrow a b S b a \quad (:\because S \rightarrow b S b)$$

$$S \Rightarrow a b b S b b a \quad (:\because S \rightarrow b S b)$$

$$S \Rightarrow a b b c b b a \quad (:\because S \rightarrow c)$$

String abcbba is derived.

3 Compare NFA & PDA. Does a pushdown automata have memory? Justify.

Ans

PDA

NFA

a) for Type-2 Grammar we can design PDA

for Type-3 Grammar we use NFA.

b) NPDA is more powerful than DPDA

NFA has same powers as in DFA.

c) Not every NPDA is transformed into equivalent DPDA.

Every NFA is transformed into equivalent DFA.

d) Context free languages can be recognized by PDA

Regular languages can be recognized by NFA.

It can be access a limited amount of information on the stack. A PDA can push an element onto the top of stack & pop off an element from the top of stack. To read an element into the stack, the top elements are popped off & are lost.

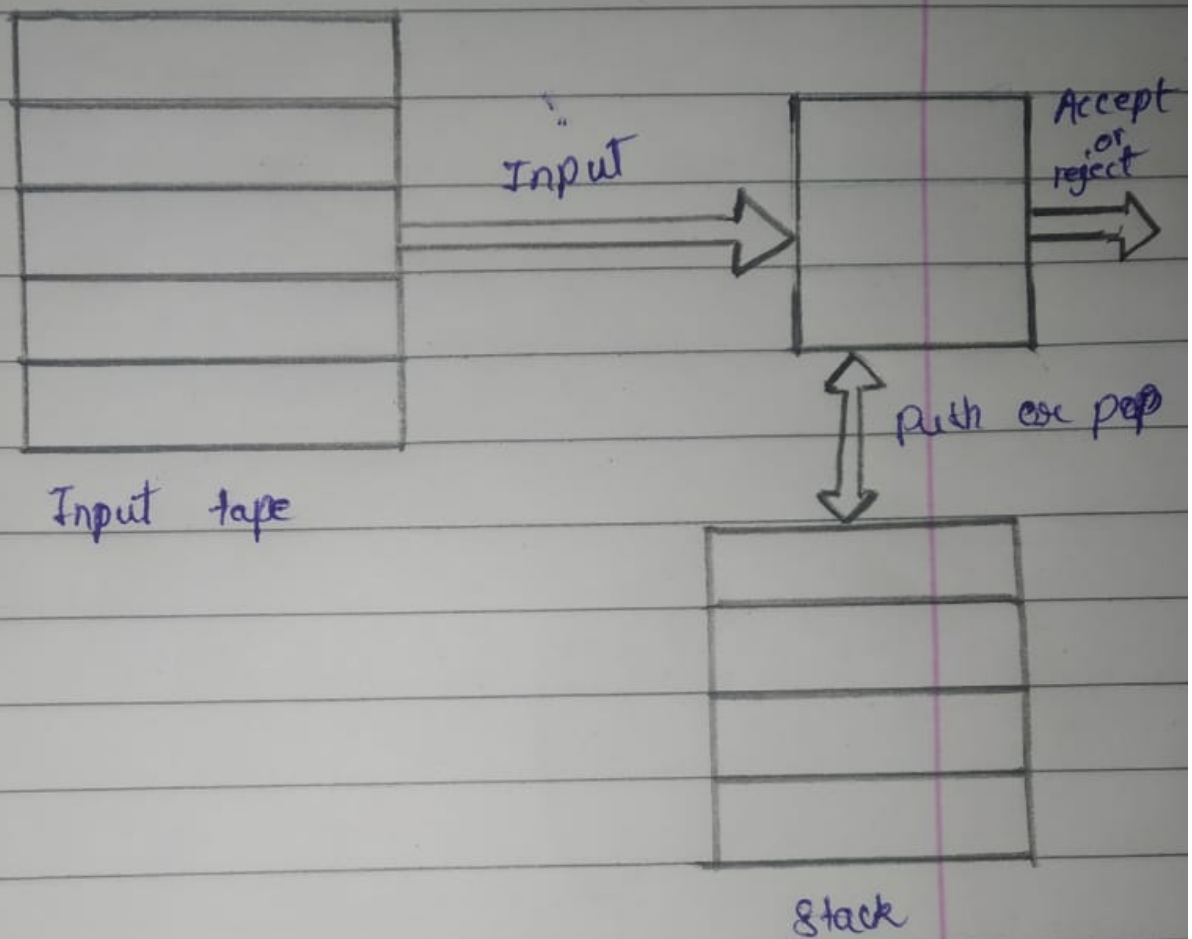


Fig: PDA