classmate
Date 07/11/23
Page

Assignment NO. 4

Q.1) Explain Instantaneous Description (ID) of PDA.

current ~~se~~ status of push down automata at particulare part of time is known as instantaneous description (ID) of PDA.

ID informs about current situation of push down automata.

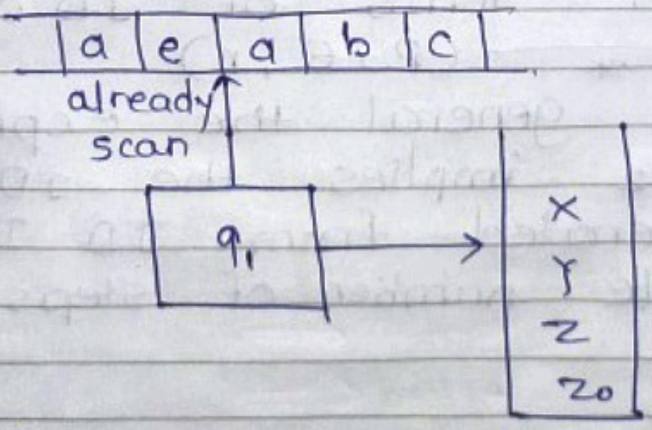
The term ID is basically a tuple consisting of 3 parts.

① current state OR present state of machine.

② The I/P symbol which is being processed currently it is also known as lookahead.

③ current contents of the stack.

consider the following sinario



By observing above diagram we came to know that machine is currently in a state q_1.

The string ae is already scan the remaining subsequent string abc is to be scanned yet.

current the stack is containing the string xyz based on this description, we can easily formulate the ID as $(q_1, "abc", "xyz")$

So, the ID is set of parameters required to execute the transitn fun.

If we consider the transitn function as $\delta(q, \alpha, Z_0) = (P, r)$

It means that when current charactor α (lookahead) is processed from state q when Z_0 is an the top of stack.

Machine changes in responce, machine changes q to P. machine will load another string r on another string.

This execution of transitn fun visualized in terms of ID as

$(q, \alpha, Z_0) \vdash^* (P, \epsilon, r)$

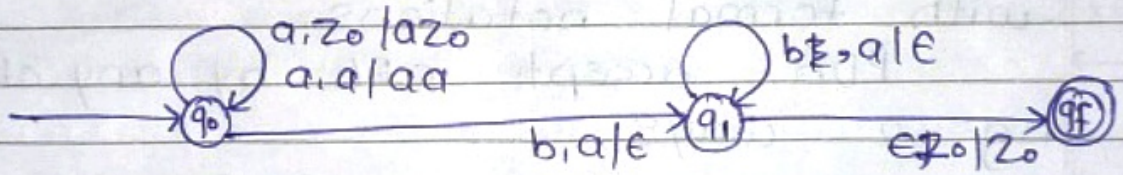
so, in general the expression $I_1 \vdash^* I_2$ implies the ID I_2 has been generated from ID I_1 in some finite number of steps.

Q.2) Discuss union of two PDAs with examples?

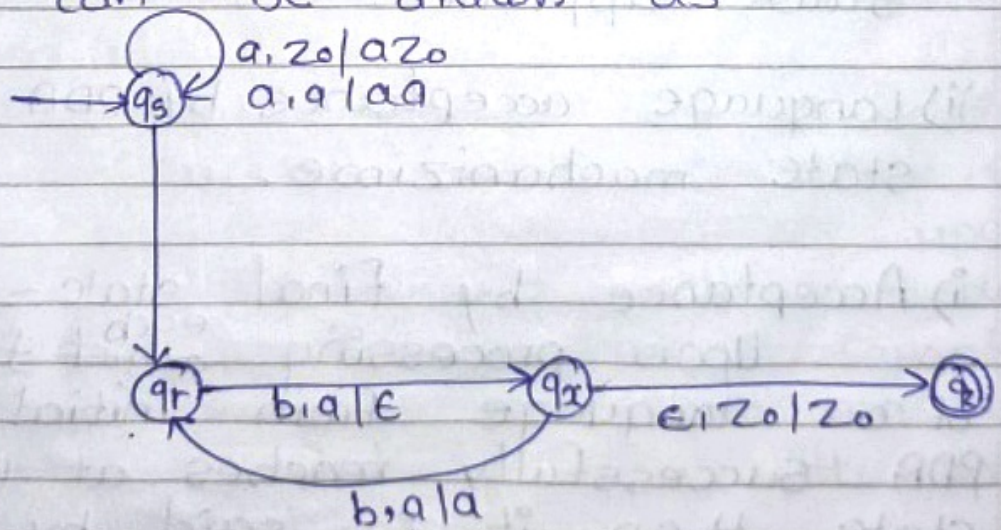
→ Union of two PDAs

Just like we can perform union operation in CFG. It is quite possible to perform union operat^n of PDA. consider, the language $L_1 = \{a^n b^n \mid n \geq 1\}$. consider another language $L_2 = \{a^n b^{2n} \mid n \geq 1\}$

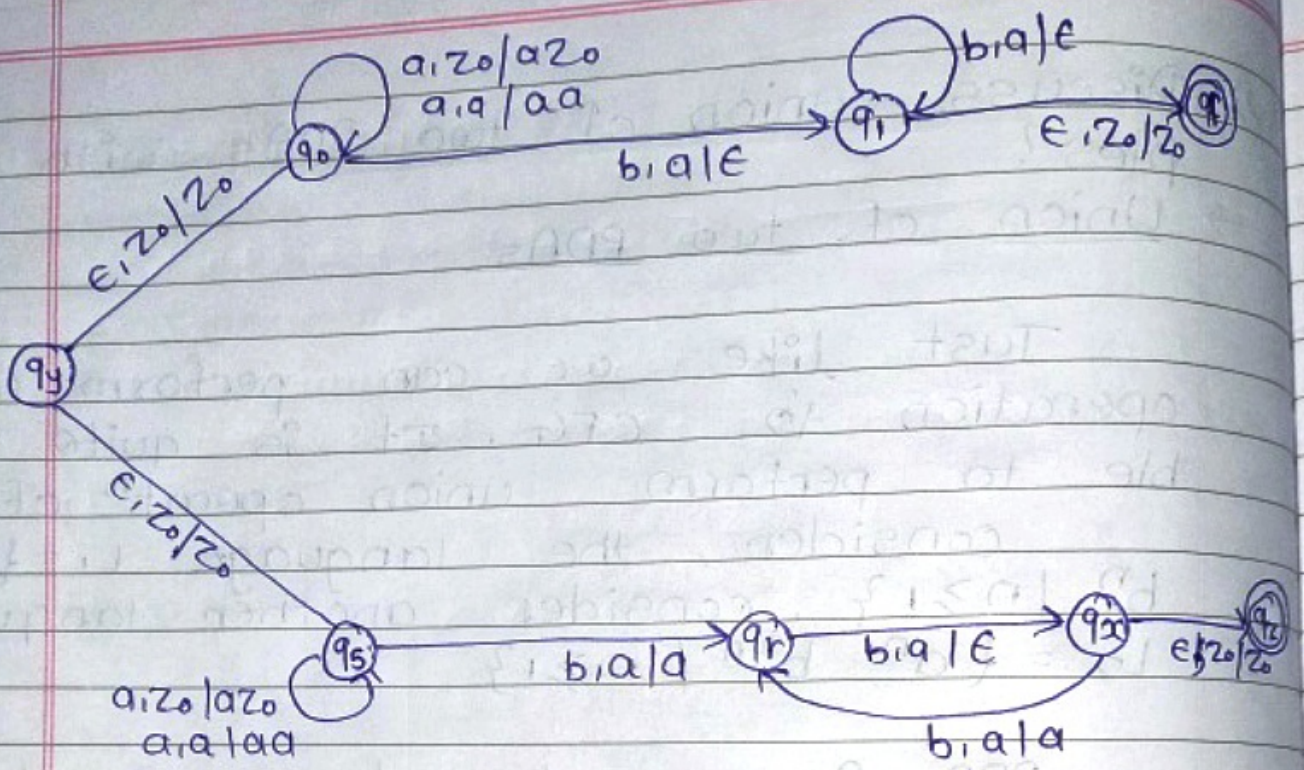
PDA P_1 can be drawn for L_1 as follows



Another PDA P_2 to accept CFL L_2 can be drawn as



Now, this two machine can be put under the union as shown below.



Q.3) Explain the language acceptance by PDA with formal notations.

→ PDA accept CFL by any of following 2 ways.

i) Language acceptance by PDA by ε-stack approach.

ii) Language acceptance by PDA by final state machanizime.

i) Acceptance by final state -

Upon processing ~~which~~ each & every string of a language from initial state, if PDA successfully reaches at the final state then it is said by concern language has been accepted by PDA by final state machanism.

Let, P is the PDA then $L_f(P)$ is the language accepted by PDA with final state approach.

Formally, we can define $L_f(P)$ as follows

$$L_f(P) = \{ w / (q_0, w, z_0) \vdash^* (q_f, \epsilon, z_0) \text{ where } q_f \in F \}$$

That means, for initial Id (q_0, w, z_0) PDA should obtain final Id (q_f, ϵ, z_0) consisting any of the final state q_f such as (q_f, ϵ, z_0) in same finite number of steps.

In final state approach the contains of stack are irrelevant. PDA successfully halt some final state.

ii) Acceptance by Empty stack -

It is another way to accept the CFL. In this approach final state don't exist.

That means $F = \phi$.

In this approach PDA mainly imphsises an making stack completely empty.

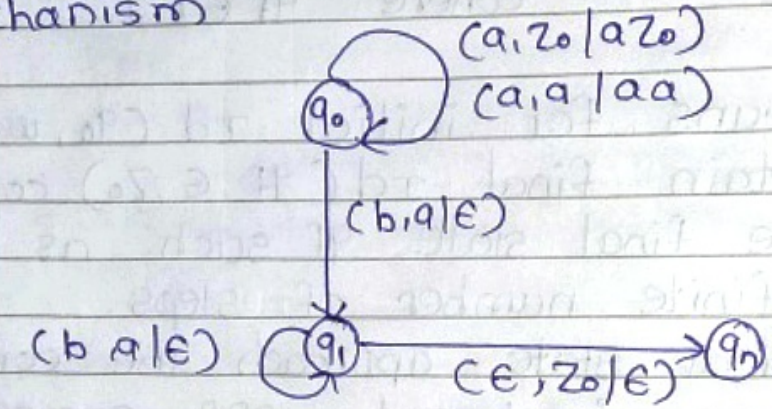
That means more prisisly upon processing. upon processing each an every string of the language if PDA is completely empty then, it is said that conserned language accepted by PDA with empty stack approach.

Let P is the PDA then $L_N(P)$ is language accepted by push-down automata. by empty stack machanism.

Formally $L_N(P)$ can be defined as,

$$L_N(P) = \{w \mid (q_0, w, Z_0) \vdash^{*} (q_n, \epsilon, \epsilon)\}$$

For example, consider the language $L = \{a^m b^m \mid m \geq 1\}$ by empty stack mechanism



Observing above transition diagram upon processing enter I/P string from inital state q_0 macheine is reaching to the non-final state q_n by making stack completly empty.

∴ It can be concluted that above PDA accepts CFL - $L = \{a^m b^m \mid m \geq 1\}$ by empty stack machanism.

Q.4) What is NPDA? Explain the non deterministic behavior of machine with example.

→ - It is another form of PDA.
- NPDA is an extension of ε-NFA.
- If we include 1 stack along with ε-NFA resulting structure is called as a NPDA.
- NPDA shows non-deterministic behaviour

- Every DPDA is NPDA. However, every NPDA may or may not be DPDA.
- by default PDA is NPDA.
- NPDA are used to accept NCFL or simply CFL.

consider the $L = \{ w w^R \mid w \in \{a,b\}^* \}$

Above language is consisting all even length palindrome.

Let's just identify some of the string in this languages

$\{ \epsilon, aa, bb, abba \ldots \}$.

In constract to add length palindrome use easily predict the centre in even length polendrom.

we connot locate the centre.

since, non of the fixed character resides at the centre.

so, in this case machine consider empty charactoe ϵ to predict the centre.

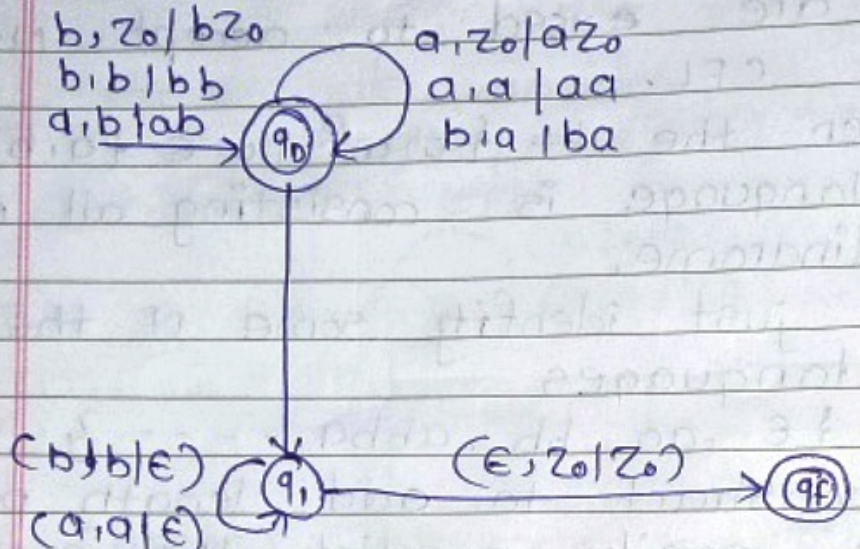
However ϵ can appeare anywhere in the I/p String.

∴ machine cannot able to findout the exact instance of ϵ which will appeare at the centre.

Due to which in each of the move machine goes on processing empty character ϵ, as well as actual charactoe from the I/p string.

Due to which behaviour of the string due to non-deterministic.

The perticulate non-deterministic machine NPDA as drawn can be sketched as,



Now, we shall demonstrate the process of I/p string abba with respect to abba.

Machine can visualize as ϵ.a.ϵ.b. ϵ.b.ϵ.a.ϵ.

Machine cann't locate the exact instance of ϵ to divide I/p in two half.

∴ Machine goes on processing ϵ character 'ϵ' blindly along with actual character in the I/p & this is main clause of the non-deterministic behaviour.

For initial ID machine goes on processing empty character of e along with actual character of I/p string.

~~The~~

So, actual part of acceptance for the I/p string abba w.r. to above machi

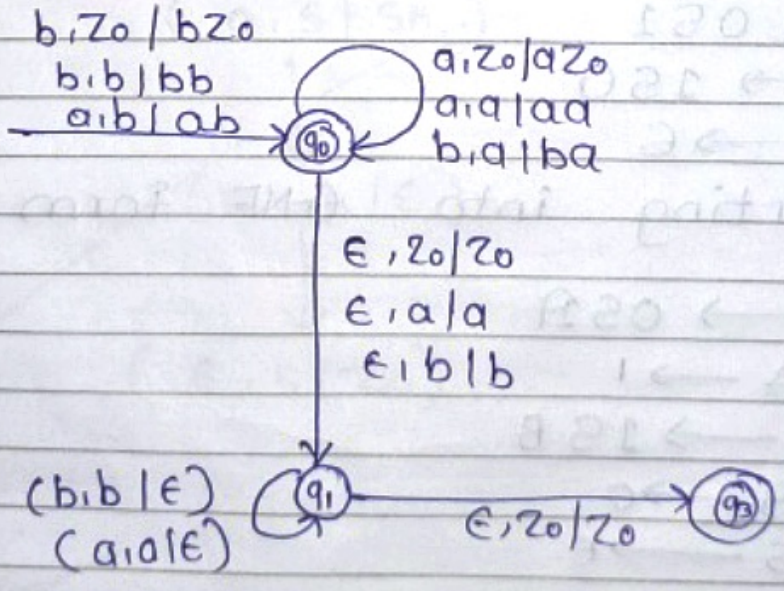
ne can be interpreted in the form of sequence of ID.

$(q_0, abba, Z_0) \vdash (q_0, bba, aZ_0) \vdash (q_0, ba, baZ_0)$
$\vdash (q_1, ba, baZ_0) \vdash (q_1, a, aZ_0) \vdash (q_1, \epsilon, Z_0)$
$\vdash (q_f, \epsilon, Z_0)$

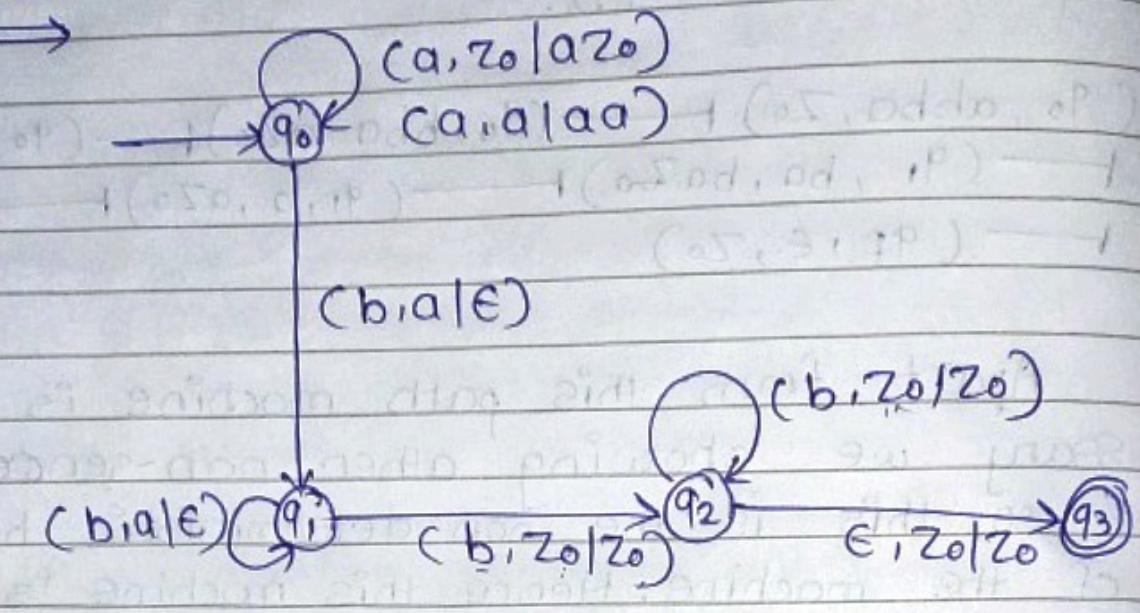
Apart from this path machine is unnecessary we showing other non-sence moves. so, this is the non-deterministic behaviour of the machine. Hence this machine is known as NPDA.

Q.5) Construct the PDA for:

i) $L = \{ww^R / w \in \{a,b\}^*\}$



ii) $L = \{a^m b^n / m < n\}$



Q.6) Convert following grammar in PDA

$S \longrightarrow 0S1 \mid 1S0 \mid \epsilon$

And simulate the derivation of I/P string 0101 with above PDA.

→ $S \longrightarrow 0S1$

$S \longrightarrow 1S0$

$S \longrightarrow \epsilon$

Converting into GNF form

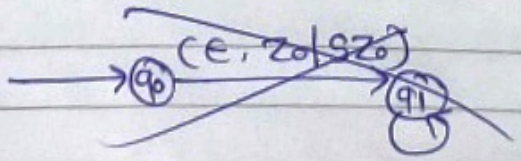
$S \longrightarrow 0SA$

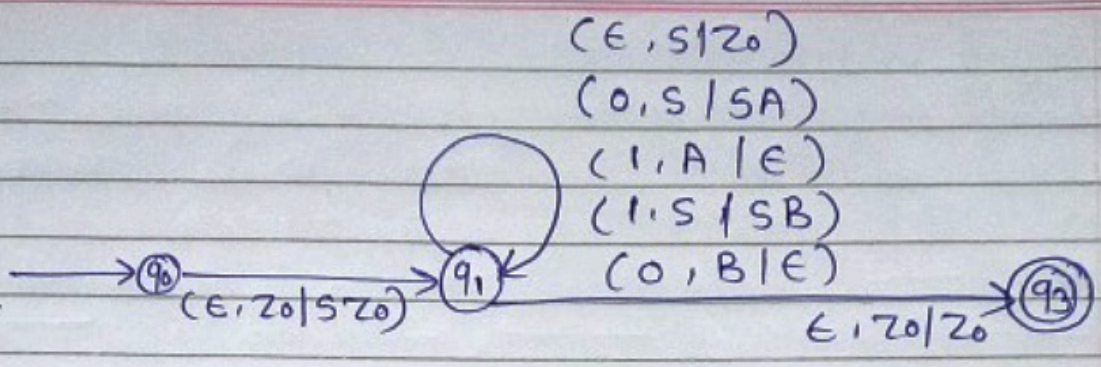
$A \longrightarrow 1$

$S \longrightarrow 1SB$

$B \longrightarrow 0$

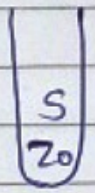
$S \longrightarrow \epsilon$



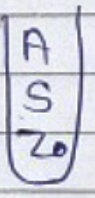


Input string 0101 -

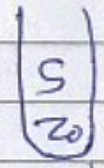
(ϵ, Z0|SZ0)



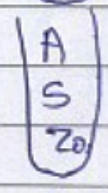
(0, S|SA)



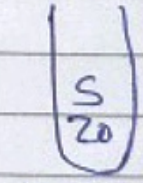
(1, A|ϵ)



(0, S|SA)



(1, A|ϵ)



(ϵ, S|Z0)