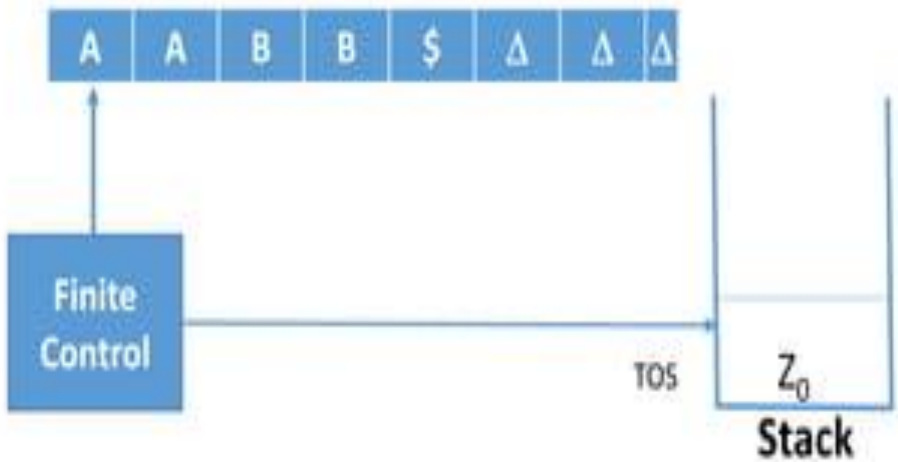
PUSH DOWN AUTOMATA

Prerequisite

- The basic relationship between CFG and R.E is that the CFG can be constructed for every R.E.
- But more than that CFG can also be written for non regular languages like 0^n1^n.
- Thus we can say that regular expressions are the subset of CFG.
- For every R.E we can be drawn F.A., But F.A. is not sufficient to draw the CFG.
- For better understanding of PDA we must know about Stack and its operations.

Push Down Automata

- The PDA will have input tape, finite control and stack.



- The input tape is divided in many cells.
- The finite control has some pointer which points to the current symbol which is to be read.
- At the end of the input $ or Δ blank symbol is placed to identify the end of input.

Definition of a PDA

- A pushdown automaton (PDA) is a seven-tuple:

  $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

  Q A finite set of states

  Σ A finite input alphabet

  Γ A finite stack alphabet

  q_0 The initial/starting state, q_0 is in Q

  z_0 A starting stack symbol, is in Γ // need not always remain at the bottom of stack

  F A set of final/accepting states, which is a subset of Q

  δ A transition function, where

  For DPDA $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow$ finite subsets of $Q \times \Gamma^*$

  For NDPDA $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow$ finite subsets of $2^{Q \times \Gamma^*}$

Push Down Automata

- Any language which is accepted by a F.A. can also accepted by PDA.
- PDA can also accepts the class of languages which are not accepted by F.A., Thus PDA is much more superior to F.A.

Example: Design a PDA for accepting the Language L= $\{a^n b^n | n>=1\}$.

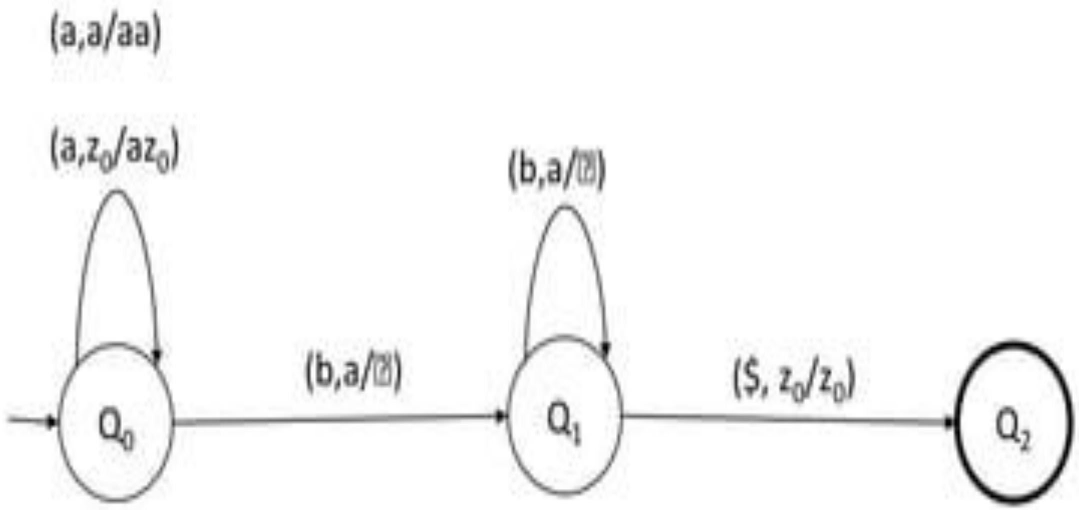
Solution:

The above given language is in which equal number of a's are followed by equal number of b's.

Logic for PDA:

1. Push all a's onto the stack
2. On reading every single b pop one a from the stack.
3. If the input string is reached end and the stack is empty then the string is accepted by the PDA.

PDA



Example(2)

- The Description for the PDA can be given as follows.

$$\delta(q_0,a,z_0)=(q_0,az_0)$$

$$\delta(q_0,a,a)=(q_0,aa)$$

$$\delta(q_0,b,a)=(q_1,\boxed{?})$$

$$\delta(q_1,b,a)=(q_1,\boxed{?})$$

$$\delta(q_1,\$, z_0)=(q_2,\boxed{?})$$

Where q_0 is start state and q_2 is accept state.

- Simulation of PDA for the input string aaabbb as follows

$$\delta(q_0,aaabbb,z_0) \vdash (q_0,aabbb\$,az_0)$$

$$\vdash (q_0,abbb\$,aaz_0)$$

$$\vdash (q_0,bbb\$,aaaz_0)$$

$$\vdash (q_1,bb\$,aaz_0)$$

$$\vdash (q_1,b\$,az_0)$$

$$\vdash (q_1,\$, z_0)$$

$$\vdash (q_2, \boxed{?})$$

Accept State

Hence the string is accepted by the PDA.

Excercise1: Design a PDA that accept a string of well formed parenthesis. Consider the parenthesis is a (,),[,],{,}.

Excercise2: Construct PDA for the language $L=\{a^n b^{2n} | n>=1\}$.

Conversion of CFG to PDA

- For the conversion of CFG to PDA the necessary condition is that the first symbol on R.H.S. production must be a terminal symbol.
  - Rule1: For non terminal symbol, add following rule

    δ(q, ⍰,A)=(q,α)

    Where the production rule is A-> α
  - Rule 2: For each terminal symbol, add the following rule.

    δ(q, a,a)=(q, ⍰) for every terminal symbol.

Example: Construct PDA for the given CFG

S->0BB

B->0S|1S|0 and test whether 010^4 is accepted by this PDA

- Let PDA

A={{q},{0,1},{S,B,0,1}, δ,q,S,F}

The Production rules δ can be written as:

R1: δ(q, ⍰,S) = {(q,0BB)}

R2: δ(q, ⍰,B) = {(q,0S),(q,1S),(q,0)}

R3: δ(q, 0,0) = {(q, ⍰)}

R4: δ(q, 1,1) = {(q, ⍰)}

Example(2):

- Testing 010^4 i.e. 010000 against **PDA**

δ(q,010000,S) ⊢(q,010000,0BB)	Since R1
⊢(q,10000,BB)	Since R3
⊢(q,10000,1SB)	Since R2
⊢(q,0000,SB)	Since R4
⊢(q,0000,0BBB)	Since R1
⊢(q,000,BBB)	Since R3
⊢(q,000,0BB)	Since R2
⊢(q,00,BB)	Since R3
⊢(q,00,0B)	Since R2
⊢(q,0, B)	Since R3
⊢(q,0,0)	Since R2
⊢(q, ⍰)	Since R3

Excercise1: Construct the PDA for the given CFG.

S->0A

A->0AB|1

B->1

Excercise1: Construct the PDA for the given CFG.

S->aSA|bSB|a|b

A->a

B->b

Construction of CFG from PDA

- Algorithm for getting production rules of CFG:

Rule1: The start symbol production can be

$S \text{->} [q_0, Z_0, q]$

Where q indicates the next state an q0 is a start state.

Rule2: If there exists a move of PDA

$\delta(q,a,Z)=\{(q', ⍰)\}$

then the production rule can be written as:

$\delta(q,Z,q')\text{->}a$

Rule3: If there exists a move of PDA as

$\delta(q,a,Z)=\{(q',Z_1,Z_2,.....Z_n)\}$

Then the production rule of CFG can be written as

Example: The PDA is given below $A=(\{q_0, q_1\},\{0,1\},\{S,A\},\delta, q_0, S,\phi)$

Where δ is given as follows:

$\delta(q_0, 1,S)=\{(q_0, AS)\}$

$\delta(q_0, \boxed{?},S)=\{(q_0, \boxed{?})\}$

$\delta(q_0, 1,A)=\{(q_0, AA)\}$

$\delta(q_0, 0,A)=\{(q_1, A)\}$

$\delta(q_1, 1,A)=\{(q_1, \boxed{?})\}$

$\delta(q_1, 0,S)=\{(q_0, S)\}$

Construct the CFG equivalent to this PDA.

Example(2):

- Solution:

Let we will construct a CFG

G=(V,T,P,S)

Here, T={0,1}, V={S ∪ $[q_0,A\,q_0]$,$[q_0,A\,q_1]$,$[q_0,S\,q_0]$,$[q_0,S\,q_1]$, $[q_1,A\,q_1]$, $[q_1,A\,q_0]$, $[q_1,S\,q_1]$, $[q_1,S\,q_0]$}

Now let up build the production rules as:

Using rule1 from the algorithm

P1: S->$[q_0,S\,q_0]$

P2: S->$[q_0,S\,q_1]$

Using rule3 of algorithm for the $\delta(q_0, 1,S)=\{(q_0, AS)\}$ we get,

P3: $[q_0,S\,q_0]$->1$[q_0,A\,q_0]$$[q_0,S\,q_0]$

P4: $[q_0,S\,q_0]$->1$[q_0,A\,q_1]$$[q_1,S\,q_0]$

P5: $[q_0,S\,q_1]$->1$[q_0,A\,q_0]$$[q_0,S\,q_1]$

P6: $[q_0,S\,q_1]$->1$[q_0,A\,q_1]$$[q_1,S\,q_1]$

Example(3):

Now for $\delta(q_0,$ ⍰$,S)=\{(q_0,$ ⍰$)\}$ using rule2 of algorithm we get

P7: $[q_0,S,q_0]\to$ ⍰

Similarly for $\delta(q_0,1,A)=\{(q_0,AA)\}$ Using rule3 we get

P8:$[q_0,A,q_0]\to 1[q_0,A,q_0][q_0,A,q_0]$

P9:$[q_0,A,q_0]\to 1[q_0,A,q_1][q_1,A,q_0]$

P10:$[q_0,A,q_1]\to 1[q_0,A,q_0][q_0,A,q_1]$

P11:$[q_0,A,q_1]\to 1[q_0,A,q_1][q_1,A,q_1]$

Similarly for $\delta(q_0,0,A)=\{(q_1,A)\}$ gives

P12:$[q_0,A,q_0]\to 0[q_1,A,q_0]$

P13:$[q_0,A,q_1]\to 0[q_1,A,q_1]$

$\delta(q_1,1,A)=\{(q_1,$ ⍰$)\}$ Gives

P14:$[q_1,A,q_1]\to 1$

$\delta(q_1,0,S)=\{(q_0,S)\}$

P15: $[q_1,S,q_0]\to 0[q_0,S,q_0]$

Exercise: Find the CFG corresponding to PDA whose transition mapping is as follows:

$\delta(S, a, X) = (s, AX)$

$\delta(S, b, A) = (s, AA)$

$\delta(S, a, A) = (s, ⍰)$

Summary

- From this session we learned what is Push Down Automata.
- How to Construct PDA.
- Conversion of CFG to PDA.
- And Conversion of PDA to CFG.

Next Class

- In the Next Session we will Concentrate on importance of Normal forms .
- Types of Normal Forms.

Given PDA:

δ(S, a , X) = (s, AX)
δ(S, b , A) = (s, AA)
δ(S, a , A) = (s, □)

Solution: Now we will apply conversion algorithm for each transition and obtain the production rules:

For δ(S, a , X) = (s, AX)

P1: (S,X,s)->a(S A S)(S X s) |a(S A s)(s X s)

P2: (S,X,S)->a(S A S)(S X S) | a(S A s)(s X S)

For δ(S, b , A) = (s, AA)

P3: (S, A, s)->b(S A S)(S A s) | b(S A s)(s A s)

P4: (S, A, S)->b(S A S)(S A S) | b(S A s)(s A S)

For δ(S, a , A) = (s, □)

P5: (S, A, s)-> a

Hence , P1,P2,P3,P4 and P5 are productions in CFG.

THE END