- **OBJECTIVE :** To learn about push down automata

UNIT III PUSH DOWN AUTOMATA 9

Definition - Moves - Instantaneous descriptions -- Equivalence of Pushdown automata and CFG - Deterministic pushdown automata - Pumping lemma for Regular Languages and CFL - Application of Pumping Lemma

PUSH DOWN AUTOMATA BASIC DEFINITIONS

Push Down Automata :

- ✶ The PDA will have an input tape a finite control and a stack.
- ✶ The stack holds a string of symbols from some alphabet
- ✶ The device will be non-deterministic having some finite number of choices of moves in each situations.

Formal Definition :

- ✶ A push Down Automata M is a system

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

Where

$Q \rightarrow$ is a finite set of states

$\Sigma \rightarrow$ is an alphabet called the input alphabet

$\Gamma \rightarrow$ is an alphabet called the stack alphabet

$q_0 \rightarrow$ in Q is the initial state

$Z_0 \rightarrow$ is Γ is a particular stack symbol called the start symbol

$F \rightarrow F \subseteq Q$, is the set of final states

$\delta \rightarrow$ is a transition function mapping from Q x $(\Sigma \cup \{\in\})$ x $\Gamma \rightarrow$ finite subset of $Q \times \Gamma^*$

Moves of the Push Down automata

- The moves will be of two types
- In the first type of move,
  - An input symbol is used. Depending on the input symbol, the top symbol on the stack and the state of a finite control, a number of choice are possible.
  - Each choice consists of next state for the finite control and a string of symbols to replace the top tack symbol.
  - After selecting a choice, the input head is advanced one symbol.

$$\delta(q, a, z) = \{(p_1, \upsilon,), (p_2, \upsilon_2), \ldots (p_m, \upsilon_m)\}$$

where

q and p_1 $1 \le i \le m$ are states

a is in Σ

$Z \rightarrow$ is a stack symbol

$\upsilon_1 \rightarrow \text{is in}\, \Gamma^*$

i.e) The PDA in state q with the input symbol a and z top symbol on the stack can for any i enter state p_1, replace symbol z by string υ_i and advance the input head one symbol.

- The second type of move called an $\in$ is similar to the first, except that the input symbol is not used and the input head is not advanced after the move

$$\delta(q, \in, z) = \{(p_1, \upsilon,), (p_2, \upsilon_2), \ldots (p_m, \upsilon_i)\}$$

ie) The PDA in state q, independent of the input symbol being scanned and with z, the top symbol on stack, can enter state p_i, for any I and replace z by υ_i. The input head is not advanced.

Instantaneous descriptions : (ID)

- ✶ The ID records the state, input and stack contents
- ✶ ID is defined as a triple

  $(q,w,\ \upsilon_i)$

  Where $q \rightarrow$ is a state in the finite control

  $w \rightarrow$ is a string of input

  $\upsilon \rightarrow$ is a string of stack symbols.

- ✶ If $M=(Q,\Sigma,\Gamma,\delta,q_0,z_0,f)$ is a PDA we say

  $(q,aw,z\delta)\ \perp(p,w,\beta,\alpha)$

  if $\delta(q,a,z)=(p,\beta)$

Language accepted by PDA

1. **Language accepted by empty stack**
   - ✶ To define the language accepted to be the set of all inputs for which some sequence of moves causes the PDA to empty its stack.
   - ✶ For PDA $M=(Q,\Sigma,\Gamma,\delta,q_0,z_0,f)$, we define N(M), the language accepted by empty stack to be

     $N(M)=\{W1(q_0,w,z_0)\perp,(P,\varepsilon,\in),\text{for some in Q}\}$
2. **Language accepted by final state:**
   - ✶ To define the language accepted to be the set of all inputs for which some choices of moves causes the PDA to enter a final state.
   - ✶ We define L(M), the language accepted by PDA M by final state to be

     $L(M)=\{W1(q_0,w,z_0)\overset{*}{\perp},(P,\varepsilon,\upsilon),\text{for some in P in f and } \upsilon \text{ in } \Gamma^*\}$

- If a set of can be accepted by empty stack by some PDA, it can be accepted by final state by some other PDA and vice versa.

Deterministic push down Automata :

- The PDA is deterministic, if atmost one move is possible from any ID.
- The PDA M = $(Q,\Sigma,\Gamma,\delta,q_0,z_0,F)$ is deterministic if
  1) For each q in Q and z in Γ, whenever $\delta(q,\in,z)$ is non empty then $\delta(q,a,z)$ is empty for all a in Σ.
  2) For no q in Q,Z in Γ and a in $\Sigma U \{\in\}$ does $\delta(q,a,z)$ contain more than one element.
- The deterministic and non- deterministic models of PDA, are not equivalent with respect to the language accepted.
- ww^R is accepted by Non-deterministic PDA but not by any Deterministic PDA

EQUIVALENCE OF PUSHDOWN AUTOMATA AND CFL

Equivalence of acceptance by final state and empty stack

- If a Language is accepted by empty stack, by some PDA, it can be accepted by final stat by some other PDA and Vice versa

Theorem 1 :

- If L is L(M_2) for some PDA M_2, then L is N(M_1) for some PDA M_1.

Proof :

- We prove M_1 to simulate M_2 with the option for
  - M_1 should erase its stack whenever M_2 enters a final state.
  - We use state q_e of M_1 to erase the stack
  - We use a bottom of stack marker X_0 for M_1 does not accidently accept if M_2 empties its stack without entering a final state
- Let M_2 = $(Q,\Sigma,\Gamma,\delta,q_0,z_0,F)$ be a PDA such that L = L (M_2)
- Let M_1 = (QU $\{q_e,q_0,'\}$, $\Sigma,\Gamma U\{x_0\},\delta^1,q_0{}^1,x_0,\phi)$

Where δ^1 is defined as follows

1) $\delta^1(q_0{}^1,\in,x_0)=\{(q_0,z_0,x_0)\}$

2) $\delta^1(q,a,z)=\delta(q,a,z\ )$

for all q in Q, a in $\in \text{or} \{\in\} \text{and} z \text{in} \Gamma$

3) For all q in F and Z in $\Gamma U\{x_0\}$

$\delta^1(q,\in,z_0)=(q_e,\in)$

4) For all z in $\Gamma U\{x_0\}$

$\delta^1(q_e,\in,z)=(q_e,\in)$

- ✶ Rule (1) causes M_1 to enter the initial ID of M_2
- ✶ Rule (2) allows M_1 to simulate M_2
- ✶ Rule (3) and (4) allow M_1 to the choice of entering state q_e and erasing its stack thereby accepting the input or continuing to simulate M_2
- ✶ Let x be in $L(M_2)$, then
  $(q_0, x,z_0) \perp (q,\in,\upsilon)$ for some q in F
- ✶ Now consider M_1 with input x By rule (1)
  $(q_0^1, x,x_0) \perp (q_0,x,z_0,x_0)$

By rule (2) every move of M_2 is a legal move for M_1 thus.

$(q_0,x,z_0) \perp (q,\in,\upsilon)$

By rule (3) and (4)

$(q,\in,\upsilon,x_0) \perp (q_e,\in,\in)$

That is

$(q_0^1, x,x_0) \perp (q_0,x,z_0,x_0) \perp (q,\in,\upsilon,x_0) \perp (q_e,\in,\in)$

M_1 accepts x by empty stack

$L(M_2) = N(M_1)$

Theorem 2:

If L is N(M) for some PDA M_1, then L is $L(M_1)$ for some PDA M_2

Proof :

- ✶ We prove M_2 to simulate M_1
- ✶ M_2 enters a final state when and only when M_1 empties its stack
- ✶ Let $M_1 = (Q,\Sigma,\Gamma,\delta,q_0,z_0,\phi)$ be a PDA such that $L = N(M_1)$
- ✶ Let $M_2 = (QU\{q_0^1, q_f\}, \Sigma, \Gamma U \{x_0\}, \delta^1, q_0^1, x_0, \{q_f\})$

Where δ' is defined as follows

1) $\delta^1(q_0^1,\in,x_0)=\{(q_0,z_0,x_0)\})$

2) For all q in Q, a in $\Sigma U\{\in\}$, and Z in Γ

$\delta^1(q,a,z)=\delta(q,a,z)$

3) For all q in Q

$\delta^1(q,\in,x_0)=(q_f,\in)$

- Rule(1) causes M_2 to enter the initial ID of M_1
- Rule(2) allows M_2 to simulate M_1
- Rule(3) causes M_2, when X_0 appears to enter a final state, thereby accepting the input x
- Let x be in $N(M_1)$

$$(q_0,x,z_0)\perp(q,\in,\in)$$

- Now, consider M_2 with input x

$$(q_0{}^1,x,x_0)\perp(q_0,x,z_0)$$

$$(q_0,x,x_0)\perp(q,\in,x_0)$$

$$(q,\in,x_0)\perp(q_f,\in,\in)$$

- Thus

$$L(M_2)=N(M_1)$$

EQUIVALENCE OF CFL AND PDA

Theorem 3:

If L is a context free Language, then there exists a PDA M, such that L = N(M)

Proof:

Let G = (V,T,P,S) be a CFG in Greibach Norm from generating L

- Let M is defined as

$$M=(\{q\},T,V,\delta,q,s,\phi)$$

Where

$$\delta(q,a,A)=(q,\upsilon)$$

Whenever A $\rightarrow a\upsilon$ is in p

- The PDA M simulates left most derivations of G is in GNF each sentential form in a left most derivation consist of a string of terminals x followed by a string of variables α
- M stores the suffix α on the left sentential form on its stack after processing the prefix x.

Theorem 4:

If L is n(m) for some PDA M, then L is a context free language.

Proof :

Let M = ($Q,\Sigma,\Gamma,\delta,q_0,z_0,\phi$) be the PDA such that L = N(M)

Let G = (V,T,P,S) be a CFG

Where

V $\rightarrow$ is the set of objects of the form [q,A,p], for each q and p in q and A in Γ plus the new symbol S.

T = Σ

P $\rightarrow$ is the set of productions.

1) S $\rightarrow$ $[q_0,z_0,q]$ for each q in Q

2) $[q,A,q_{m+1}]$ $\rightarrow a[q_1,B_1,q_2][q_2,B_2,q_3],...[q_m,B_m,q_{m+1}]$

for each $q,q_1,q_2,\ldots q_{m+1}$ in Q each a in $\Sigma U\{\in\}$ and $A,B_1,B_2,\ldots,B_m$ in Γ such that $\delta(q,a,A)=(q_1,B_1,B_2...B_m)$

(If m = 0, then the productin is $[q,A,q_1]$ $\rightarrow$ a

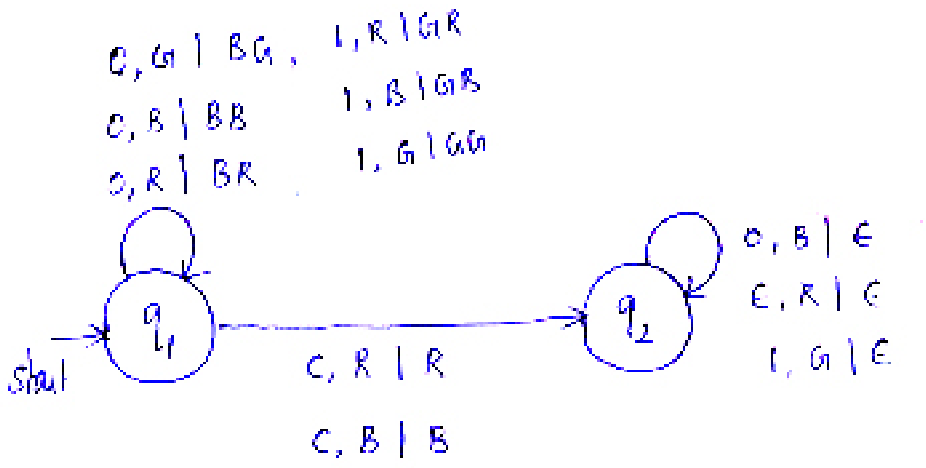
CONVERSION OF CFL TO PDA

1.Design a PDA accepting L = {wcw^R | win (0+1)*} by empty stack.

Solution

$M=(\{q_1,q_2\},\{0,1,c\},\{R,B,G\},\delta,q_1,R,\theta)$

Transition Diagram:



where δ is defined as

1) $\delta(q_1, O, R) = \{(q_1, BR)\}$

2) $\delta(q_1, O, B) = \{(q_1, BB)\}$

3) $\delta(q_1, O, G) = \{(q_1, BG)\}$

4) $\delta(q_1, C, R) = \{(q_2, R)\}$

5) $\delta(q_1, C, B) = \{(q_2, B)\}$

6) $\delta(q_1, C, G) = \{(q_2, G)\}$

7) $\delta(q_1, 1, R) = \{(q_1, GR)\}$

8) $\delta(q_1, 1, B) = \{(q_1, GB)\}$

9) $\delta(q_1, 1, G) = \{(q_1, GG)\}$

10) $\delta(q_2, O, B) = \{(q_2, \in)\}$

11) $\delta(q_2, \in, R) = \{(q_2, \in)\}$

12) $\delta(q_2, 1, G) = \{(q_2, \in)\}$

Let x = O1C1O

$(q_1, O1C1O, R) \perp (q_1, 1C1O, BR) \perp (q_1, clo, GBR)$

$\perp (q_2, 1O, GBR) \perp (q_2, O, BR) \perp (q_2, \in, R)$

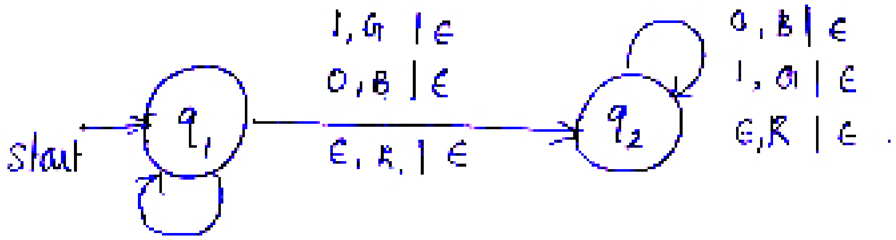
$\vdash (q_2, \in, \in)$

x = O1C1O is accepted by PDA

2) Design a PDA accepting L = {WW[R]|W in (0+1)[R]} by empty stack

Solution

Transition Diagram



PDA M is defined as,

1) $\delta(q_1, O, R) = \{(q_1, BR)\}$
2) $\delta(q_1, 1, R) = \{(q_1, GR)\}$
3) $\delta(q_1, O, B) = \{(q_1, BB)\}, (q_2, \in)\}$
4) $\delta(q_1, O, G) = \{(q_1, BG)\}$
5) $\delta(q_1, 1, B) = \{(q_1, GB)\}$
6) $\delta(q_1, 1, G) = \{(q_1, GG)\}(q_2, \in)\}$
7) $\delta(q_2, O, B) = \{(q_2, \in)\}$
8) $\delta(q_2, 1, G) = \{(q_2, \in)\}$
9) $\delta(q_1, \in, R) = \{(q_2, \in)\}$
10) $\delta(q_2, \in, R) = \{(q_2, \in)\}$

$M = (\{q_1, q_2,\}, \{0,1\}, \{R, B, G\}, \delta, 9_1, R, \phi)$

Let x=0110

$(q_1, 0110, R) \vdash (q_1, 110, BR)$

$\vdash (q_1, 10, GBR) \vdash (q_1, O, BR)$

$\vdash (q_1, O, GGBR) \vdash (q_2, \in, R)$

$\vdash (q_1, O, BGGBR) \vdash (q_2, \in, \in)$

$\vdash (q_1, \in, BBGGBR)$

x = 0110 is accepted by PDA

4) Convert the Grammar

$S \rightarrow OSI | A$

$A \rightarrow IAO | S | E$

Design a PDA that accepts the same language by empty stack.

Solution

The Given grammar

G=(V,T,P,S)

Where V = {S,A}

T = {0,1}

To construct a PDA M = $Q, \Sigma, \Gamma, \delta, q_0, z_0, \phi)$

Where

Q = $\{P_1, q_2\}$

$\Sigma = T = \{0,1\}$

$\Gamma = \{v\} U \{z_0\}$

= $\{S,A,O,1,Z_0\}$

where δ is defined as,

1) $\delta(q_1, \in, z_0) = q_2, S)$

For each production

2) $\delta(q_2, \in, s) = (q_2, OSI)$
3) $\delta(q_2, \in, s) = (q_2, A)$
4) $\delta(q_2, \in, A) = (q_2, S)$
5) $\delta(q_2, \in, A) = (q_2, IAO)$
6) $\delta(q_2, \in, A) = (q_2, \in)$

For each terminal

7) $\delta(q_2, o, O) = (q_2, \in)$

8) $\delta(q_2,1,1)=(q_2,\in)$

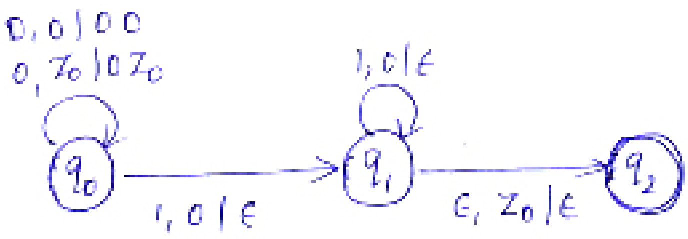
9) $\delta(q_2,o,1)=(q_2,\in)$

10) $\delta(q_2,1,0)=(q_2,\in)$

4) Design a PDA accepting L = {$a^n b^n$ | n $\geq$1} by final state

Solution:

Transition Diagram:



Transition function δ

1) $\delta(q_0,o,z_0)=(q_0,Oz_0)$

2) $\delta(q_0,o,o)=(q_0,oo)$

3) $\delta(q_0,1,o)=(q_1,\in)$

4) $\delta(q_1,1,o)=(q_1,\in)$

5) $\delta(q_1,\in,z_0)=(q_2,\in)$

The PDA is defined as

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

$Q = \{q_0,q_1,q_2\}$

$\Sigma=\{0,1)$

$\Gamma=\{0;1,z_0\}$

$F = \{q_2\}$

Let us consider the string w = 0011

$(q_0,0011;z_0)\perp(q_0,011;oz_0)\perp(q_1,1;0z_0)\perp(q_1,\in;z_0)\perp(q_2,\in)$

Final state is reached so the string w = 0011 is accepted.

CONVERSION OF PDA TO CFL

Given PDA

$M = (\{q_0,q_1\},\{0,1\},\{x,z_0\},\delta,q_0,q_0,\phi)$

where δ is given by

$\delta(q_0,o,z_0)=(q_0,xz_0)$

$\delta(q_0,o,x)=(q_0,xx)$

$\delta(q_0,1,x)=(q_1,\in)$

$\delta(q_1,1,x)=(q_1,\in)$

$\delta(q_1,\in,x)=(q_1,\in)$

$\delta(q_1,\in,z_0)=(q_1,\in)$

Construct a CFG G, generating N(M)

Solution :

$V=\{s,[q_0,x,q_0],[q_0,x,q_1],[q_1,x,q_0],[q_1,x,q_1][q_0,z_0,q_0][q_0,z_0,q_1][q_1,z_0,q_0][q_1,z_0,q_1]\}$

$T = \{0,1\}$

The productions for s are

$S\rightarrow[q_0,z_0,q_0]$ (from rule 1)

$S\rightarrow[q_0,z_0,q_1]$

The Productions for each transition functions of M.

1) $\delta(q_0,o,z_0)=(q_0,x,z_0)$

The Productions are

$[q_0,z_0,q_0]\rightarrow o\,[q_0,x,q_0][q_0,z_0,q_0]$

$[q_0,z_0,q_0]\rightarrow o\,[q_0,x,q_1][q_1,z_0,q_0]$

$[q_0,z_0,q_1]\rightarrow o\,[q_0,x,q_0][q_0,z_0,q_1]$

$[q_0, z_0, q_1] \rightarrow o\,[q_0, x, q_1][q_1, z_0, q_1]$

2) There is a move $\delta\ (q_0, o, x) = \{(q_0, xx)\}$

The productions are

$[q_0, x, q_0] \rightarrow o\,[q_0, x, q_0][q_0, z_0, q_0]$

$[q_0, x, q_0] \rightarrow o\,[q_0, x, q_1][q_1, x, q_0]$

$[q_0, x, q_1] \rightarrow o\,[q_0, x, q_0][q_0, x, q_1]$

$[q_0, x, q_1] \rightarrow o\,[q_0, x, q_1][q_1, x, q_1]$

3) $\delta\ (q_0, 1, x) = \{(q_1, \in)\}$

The production is

$[q_0, x, q_1] \rightarrow 1$

4) $\delta\ (q_1, 1, x) = \{(q_1, \in)\}$

The production is

$[q_1, x, q_1] \rightarrow 1$

5) $\delta\ (q_1, \in, x) = \{(q_1, \in)\}$

The production is

$[q_1, x, q_1] \rightarrow \in$

6) $\delta\ (q_0, 1, x) = \{(q_1, \in)\}$

The production is

$[q_0, x, q_1] \rightarrow 1$

- ✶ There are no production for the variable $[q_1, x, q_0]$ and $[q_1, z_0, q_0]$
- ✶ As all the productions for $[q_0, x, q_0]$ and $[q_0, z_0, q_0]$ have $[q_1, x, q_0]$ or $[q_1, z_0, q_0]$ on the right, no string of terminals can be derived from $[q_0, x, q_0]$ or $[q_0, z_0, q_0]$
- ✶ Deleting all production involving one of these variables on either the right or left, we end up with the following productions.

$S \rightarrow [q_0, z_0, q_1]$

$[q_0,z_0,q_1]\rightarrow o\,[q_0,x,q_1][q_1,z_0,q_1]$

$[q_0,x,q_1]\rightarrow o\,[q_0,x,q_1][q_1,x,q_1]$

$[q_0,x,q_1]\rightarrow 1$

$[q_1,z_0,q_1]\rightarrow \in$

$[q_1,x,q_1]\rightarrow \in$

$[q_1,x,q_1]\rightarrow 1$

1) Give a grammer for the language N(M) where
$M=(\{q_0,q_1\},\{0,1\},\{z_0,x\},\delta,q_0,z_0,\phi)$ and δ is given by

$\delta(q_0,1,z_0)=\{(q_0,xz_0)\}$

$\delta(q_0,1,x)=\{(q_0,xx)\}$

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

$\delta(q_0,\in,z_0)=\{(q_0,\in)\}$

$\delta(q_1,1,x)=\{(q_1,\in)\}$

$\delta(q_1,0,z_0)=\{(q_0,z_0)\}$

$V=\{S_1,[q_0,z_0,q_0],[q_0,z_0,q_1],[q_1,z_0,q_0],[q_1,z_0,q_1][q_0,x,q_0][q_0,x,q_1][q_1,x,q_0][q_1,x,q_1]\}$

T = {0,1}

P = set of productions

$S\rightarrow[q_0,z_0,q_0]$

$S\rightarrow[q_0,z_0,q_1]$

1) $\delta(q_0,1,z_0)=\{(q_0,xz_0)\}$

$[q_0,z_0,q_0]\rightarrow 1\,[q_0,x,q_0][q_0,z_0,q_0]$

$[q_0,z_0,q_0]\rightarrow 1\,[q_0,x,q_1][q_1,z_0,q_0]$

$[q_0,z_0,q_1]\rightarrow 1\,[q_0,x,q_0][q_0,z_0,q_1]$

$[q_0,z_0,q_1]\rightarrow 1\,[q_0,x,q_1][q_1,z_0,q_1]$

2) $\delta\ (q_0,1,x) = \{(q_0,xx)\}$

$[q_0,x,q_0]\rightarrow 1\,[q_0,x,q_0][q_0,x,q_0]$

$[q_0,x,q_0]\rightarrow 1\,[q_0,x,q_1][q_1,x,q_0]$

$[q_0,x,q_1]\rightarrow 1\,[q_0,x,q_0][q_0,x,q_1]$

$[q_0,x,q_1]\rightarrow 1\,[q_0,x,q_1][q_1,x,q_1]$

3) $\delta\ (q_0,0,x) = (q_1,x)$

$[q_0,x,q_0]\rightarrow 0\,[q_1,x,q_0]$

$[q_0,x,q_1]\rightarrow 0\,[q_1,x,q_1]$

4) $\delta\ (q_0,\in,z_0) = \{(q_0,\in)\}$

$[q_0,z_0,q_0]\rightarrow\in$

5) $\delta\ (q_1,1,x) = \{(q_1,\in)\}$

$[q_1,x,q_1]\rightarrow 1$

6) $\delta\ (q_1,0,z_0) = \{(q_0,z_0)\}$

$[q_1,z_0,q_0]\rightarrow 0\,[q_0,z_0,q_0]$

$[q_1,z_0,q_1]\rightarrow 0\,[q_0,z_0,q_1]$

- ✶ There are no production for the variable $[q_1,x,q_0]$
- ✶ As all the productions for $[q_0,x,q_0]$ have $[q_1,x,q_0]$ on the right, no string of terminals can be derived from $[q_0,x,q_0]$
- ✶ Deleting all production involving one of these variables on either the right or left, we end up with the following productions.

$[q_0,z_0,q_1]\rightarrow 1\,[q_0,x,q_1][q_1,z_0,q_1]$

$[q_0,z_o,q_0]\rightarrow 1\,[q_0,x,q_1][q_1,z_o,q_0]$

$[q_0,x,q_1]\rightarrow 1\,[q_0,x,q_1][q_1,x,q_1]$

$[q_0,x,q_1]\rightarrow 0\,[q_1,x,q_1]$

$[q_1,z_0,q_0]\rightarrow\in$

$[q_1, x, q_1] \rightarrow 1$

$[q_1, z_o, q_0] \rightarrow 0\, [q_0, z_o, q_0]$

$[q_1, z_o, q_1] \rightarrow 0\, [q_0, z_o, q_1]$

PUMPING LEMMA FOR CFL

Pumping Lemma for CFL:

- The pumping lemma for CFL's states that there are two short substrings close together that can be repeated both the same number of times.

Lemma

Let L be any CFL. Then there is a constant n, depending only on >, such that if z is in L and

$| z | \geq$ n then we may write

z = uvwxy such that

1) $| Vx | \geq 1$
2) $| Vwx | \leq n$
3) for all I $\geq$ $uv^i sx^i y$ is in L

Proof :

- Let G be a CFG in CNF generating L $\{\in\}$
- If Z is in L(G) and Z is long, then any parse tree for z must contain a long path
- We prove this by mathematical induction on i, that path of length for z.
- If the word generated by a CNF grammar has no path of length greater than I, then the word length is no greater than 2^{i-1}

Basis :

i = l is trivial

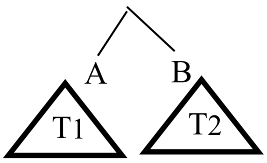
Since the tree must be of the form

Induction step:

Let I > 1

- If there are no paths of length greater than i - 1 in trees T_1 and T_2, then the trees generate words of 2^{i-1} of fewer symbols
- Thus the entire tree generates a word no longer than 2^{i-1}

S



- Let G have k variables and let $n = 2^k$
- If z is in L(G) and $| z | \geq n$, then since $| z | >^{k-1}$, any parse tree for z must have a path of length at least k +1.
- But such a path has k + 2 vertices, and all except last vertex are labeled by variables.
- Thus there must be some variables that appears twice on the path.
- Some variables must appear twice near the bottom of the path.
- Then there must be two variables V_1 and V_2 on the path satisfying the following conditions.
  1. The vertices V_1 and V_2both have the same label say A
  2. Vertex V_1 is closer to the root than vertex V_2
  3. The portion of the path from V_1 to the leaf is of length atmost k + 1
- Now the subtree T_1 and T_2 is the subtree generated by vertex V_2 and Let Z_2 is the yield of subtree T_2

  Then we can write

  $$Z_1 = Z_3 Z_2 Z_4$$

- Z_3 and Z_4 cannot both be $\in$. Since the production used in the derivation of z_1 must be of the form $A \rightarrow BC$, for some variable B and C
- The Subtree T_2 must be completely within either the subtree generated by B or the subtree generated by C.
- Example :

  G = ({A,B,C}, {a,b}, P,A)

  P : $\{A \rightarrow BC, B \rightarrow BA, A \rightarrow a, B \rightarrow b, \{C \rightarrow BA\}$

Applications of pumping lemma:

- The pumping lemma can be used to prove a variety of languages not to be context free.
- The pumping lemma can also be used to show that certain languages similar to L_1 are not context free.

Example :

show that $L = \{a^i b^i c^i \mid 1 \geq 1\}$ is not context free. Language.

Solution :

Let L = $\{a^i b^i c^i \mid i \geq 1\}$ is a CFL

Let n $\rightarrow$ be the constant of Lemma 1.

consider Z = $a^n b^n c^n$

write Z = uvwxy so as to satisfy the conditions of pumping lemma

$u = a^m$

$vwx = a^{n-m}\, b^m$, $w = a^j\, a^j$

$y = b^{n-m}\, c^n$

Verify

$uvwxy = a^m\, a^{n-m}\, b^m\, b^{n-m}\, c^n$

$= a^n\, b^n\, c^n$

$uv^iwx^iy = a^m\, (a^{n-m-j})^{-i}\, a^j\, b^j\, (b^{m-j})^I\, b^{n-m}\, c^n$

For I = o

$uv^iwx^iy = a^m\, a^j\, b^j\, b^{n-m}\, c^n$

$= a^{m+j}\, b^{n-m\,j} c^n$ # L

For i = 2

$uv^iwx^iy = a^m\, a^{2n-2m-2j}\, a^j\, b^j\, b^{n-m}\, c^n$

$= a^{2n-m-j}, b^{m+n-jj} c^n$ # L

For i = 0 and i = 2 uv^iwx^iy is not in L so given L = $\{a^i b^i c^i \mid i \geq 1\}$ is a CFL