6. PUSH DOWN AUTOMATA (PDA)

CHAPTER - 6
PUSH DOWN AUTOMATA (PDA)

6.1 INTRODUCTION

(i) The regular languages have an equivalent automaton - *the finite automaton.*

(ii) An automaton equivalent to context free language is *push down automata*.

(iii) Finite automata cannot recognise all context free languages. Because some context free languages are not regular.

(iv) Finite automata have strictly finite memories, whereas recognition of context-free language may require storing an unbounded amount of information.

(v) Push down automata is a machine similar to finite automata that will accept context free languages, except more powerful.

Examples 6.1

(i) $L=\{a^n b^n : n \geq 0\}$, to handle this language, we must not only check that all *a*'s procede the first b, we must also count the number of *a*'s. Since *n* is unbounded, this counting cannot be done with a finite memory. So we require PDA (Push Down Automata), a machine that can count without limit.

(ii) $L = \{ww^R : w \in \{a,b\}^*\}$, to handle this language, we need more than unlimited counting ability, we need to store and match a sequence of symbols in reverse order.

6.2 PUSH DOWN AUTOMATA (PDA)

The push down automata is essentially a finite automaton with control of both an *input tape* and a *stack* (or) *Last in-first out (Lifo) list*. The symbols can be entered (or) removed only on top of the storage (stack). When a symbol is added on top, the symbol previously on top becomes the second and so on. Similarly, when a symbol is removed from the top of stack, the symbol previously second from top becomes top symbol and so on.

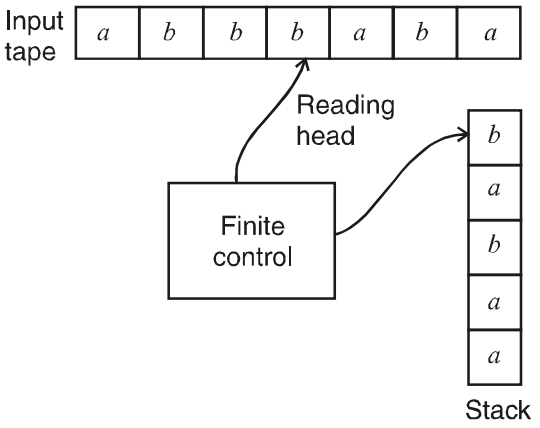


Fig. 6.1 Push Down Automata

6.2.1 Comparison of NFA and PDA

NFA

(i) $(p, a, q) \in \Delta$ means if machine M is in state p, then on reading 'a' from input tape go to state q.

(ii) $(p, \varepsilon, q) \in \Delta$ means if machine M is in state p, goes to state q, without consuming input.

PDA

(i) $((p, a, \beta), (q,y)) \rightarrow$ if machine M is in state p, the symbol read from input tape is 'a', and β is on top of stack, goes to state q, and replace β by y on top of stack.

(ii) $((s, a, e), (s, a)) \rightarrow$ if machine M is in state s, reads 'a', remains in state s and push a onto stack (e-empty stack).

(iii) $((s, c, e), (f, e)) \rightarrow$ if read 'c' in state s and stack is empty, goes to final state f and nothing to push onto stack.

(iv) $((s, e, e), (f, e)) \rightarrow$ if in state s, go to state f.

(v) $((f, q, a), (f, \phi)) \rightarrow$ if read a in state f, remain in state f and pop a from stack.

(vi) PDA's are non-deterministic.

6.2.2 Definition of the Push Down Automata

A push down automaton M is defined by $(Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$, where

- ⋆ Q is a *finite set of states*.
- ⋆ Σ is an alphabet called the *input alphabet* - is a finite alphabet of tape symbols.
- ⋆ Γ is an alphabet called *stack alphabet* - is a finite alphabet of stack symbols.

- ★ $q_0 \in Q$ is the *start state* (or) *initial state*
- ★ z_0 in Γ is a particular stack symbol called *start symbol*.
- ★ $F \subseteq Q$ is the set of *final* (or) *favorable states*.
- ★ δ is the *transition relation*.

  i.e. δ is a subset of $(Q \times (\Sigma \cup \{e\}) \times \Gamma^*) \rightarrow (2^{Q \times \Gamma^*})$

6.2.3 Transitions and Graphical Notation

Let $((p, a, \beta), (q, \gamma)) \in \delta$. It means that

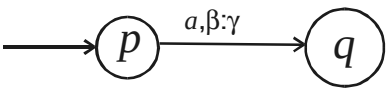
(a) read a from the tape

(b) pop the string β from the stack

(c) move from state p to state q

(d) push string γ on to stack.

The equivalent transition diagram is:



Sometimes, a diagram, generalizing the transition diagram of a finite automaton, will make aspects of the behavior of a given PDA clearer.

(a) The nodes corresponds to the states of the PDA.

(b) An arrow labeled start indicates the start state, and doubly circled states are accepting.

(c) The arc corresponds to the transition of the PDA.

for ex : $\delta(q, a, X) = \{(p, \alpha)\}$

is denoted by an arc labeled $a, X/d$ from state q to state p.

Example 6.2

Consider the Language $L_{ww^r} = \{ww^R \mid w \text{ is in } (0+1)^*\}$

Solution :

The design of pushdown automaton to accept the language is as follows.

Language is palindrome over alphabet {0, 1}

(i) Start in a state q_0, read symbols till the end of the string w and store them on the stack, by pushing a copy of each input symbol onto stack.

(ii) We guess we have seen the middle, so go to state q_1.

(iii) Once in q_1, we compare input symbols with the symbol at the top of the stack. If they match, we consume the input symbol, pop the stack and proceed. If they do not match, then input string is not palindrome.

(iv) If we empty the stack, we have seen input w followed by w^R. The string is accepted.

PDA P(Q, Σ, Γ, δ, q_0, z_0, ϕ)

Q = $\{q_0, q_1, q_2\}$ = No. of states

Σ = {0, 1} = Input symbol alphabet

Γ = $\{0, 1, z_0\}$

Start state = q_0

Start stack symbol = z_0

Final state = $\{q_2\}$

δ is :

a) Set to push w on to stack

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

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

$\delta(q_0, 0, 0) = \{(q_0, 00)\}$

$\delta(q_0, 0, 1) = \{(q_0, 01)\}$

$\delta(q_0, 1, 0) = \{(q_0, 10)\}$

$\delta(q_0, 1, 1) = \{(q_0, 11)\}$

b) To find middle of the string, *npda* switches state from q_0 to q_1

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

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

$\delta(q_0, \varepsilon, 1) = \{(q_1, 1)\}$

c) Match w^R against contents of stack

$\delta(q_1, 0, 0) = \{(q_1, \varepsilon)\}$

$\delta(q_1, 1, 1) = \{(q_1, \varepsilon)\}$

d) To reach final state

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

the corresponding transition diagram is:

6.3 INSTANTANEOUS DESCRIPTIONS (ID)

A PDA goes from configuration to configuration when consuming input. To reason about PDA computation, we use *Instantaneous Descriptions (ID)* of the PDA. An ID is a triple (q, w, γ) where

- ⋆ q is the *current state*
- ⋆ w is the *remaining input*
- ⋆ γ is the *stack contents*.

Definition

Let P be a PDA. P = (Q, Σ, Γ, δ, q_0, z_0 F)

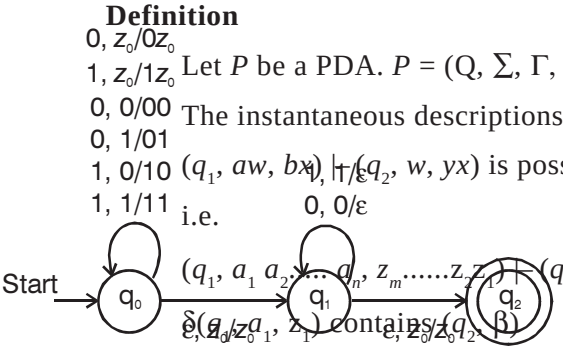
The instantaneous descriptions of pushdown automata is such that

$(q_1, aw, bx) \vdash (q_2, w, yx)$ is possible if and only if $(q_2, y) \in \delta\ (q, a, b)$

i.e.

$(q_1, a_1\ a_2, \ldots\ a_n, z_m \ldots\ldots z_2 z_1) \vdash (q_2, a_2, \ldots\ldots\ldots\ a_n, \beta z_m \ldots\ldots\ z_2)$ is possible if and only if

$\delta(q_1, a_1, z_1)$ contains (q_2, β)



Note :

As $\vdash$ defines a relation in the set of all IDs of a PDA, we can define the reflexive-transitive closure $\vdash^*$ which represents a definite sequence of n moves, where n is any non-negative integer.

If $(q, x, \alpha) \vdash^* (q^1, y, \beta)$ represents n moves, we write

$(q, x, \alpha) \vdash^n (q^1, y, \beta)$

In particular $(q, x, \alpha) \vdash^0 (q, x, \alpha)$.

Also $(q, x, \alpha) \vdash^* (q, y\ \beta)$ can be split as

$(q, x, \alpha) \vdash (q_1, x_1, \alpha_1) \vdash (q_2, x_2, \alpha_2) \vdash \ldots\ldots \vdash (q^1, y, \beta)$ for some $x_1, x_2 \ldots\ldots \in \Sigma^*$ and $\alpha_1, \alpha_2 \ldots\ldots \in \Gamma^*$

Definition of Move interpretations

(i) The meaning of $\delta(q, a, z) = \{(p_1, \nu_2), \ldots\ldots (p_m, \nu_m)\}$ where q and p_i, $1 \le i \le m$ are states, $a \in \Sigma$, $z \in \Gamma$, $\nu_i \in \Gamma^*$, $1 \le i \le m$, is that the PDA in state q_i with input symbol a and z the top symbol on the stack. For any i, enter state p_i, replace symbol z by string ν_i and advance the input head one symbol. The convention for stack is left most symbol of ν_i will be placed highest on the stack and the right most symbol lowest on the stack.

(ii) The meaning of $\delta(q, \varepsilon, z) = \{(p_1, \nu_1) (p_2, \nu_2), \ldots\ldots (p_m, \nu_m)\}$ is that the PDA in state q, independent of the input symbol being scanned and with z the top symbol on the stack, can enter state p_i and replace z by ν_i for any i, $1 \le i \le m$. In this case, the input head is not advanced.

Properties of $\vdash^*$

Property 1

If $(q_1, x, \alpha) \vdash^* (q_2, \lambda, \beta)$..... (1)

then for every $y \in \Sigma^*$,

$(q_1, xy, \alpha) \vdash^* (q_2, y, \beta)$..... (2)

conversely, if $(q_1, xy, \alpha) \vdash^* (q_2, y, \beta)$ for some $y \in \Sigma^*$

then $(q_1, x, \alpha) \vdash^* (q_2, \lambda, \beta)$

Proof

If the PDA is in state q_1 with α in pushdown store (PDS (or) stack), and the moves given by (1) is effected by processing the string x, the PDA moves to state q_2 with β in PDS.

The same transition is effected by starting with the input symbol xy and processing only x. In this case, y remains to be processed and hence we get (2).

Property 2

If $(q, x, \alpha) \vdash^* (q^1, \lambda, \gamma)$..... (3)

then for every $\beta \in \Gamma^*$

$(q, x, \alpha\beta) \vdash^* (q^1, \lambda, \gamma\beta)$..... (4)

Proof

The sequence of moves given by (3) can be split as

$(q, x, \alpha) \vdash (q_1, x_1, \alpha_1) \vdash^* (q_2, x_2, \alpha_2) \vdash \ldots\ldots\ldots \vdash (q^1, \lambda, \gamma)$

Consider $(q_i, x_i, \alpha_i) \vdash (q_{i+1}, x_{i+1}, \alpha_{i+1})$.

Let $\alpha_i = z_1 z_2 \ldots\ldots z_m$. As a result of the move, z_1 is erased and some string is placed above $z_2 \ldots\ldots z_m$. So $z_2 \ldots\ldots z_m$ is not affected.

If we have β below z_2.... z_m, then also z_2..... z_mβ is not affected. So we obtain

$(q, x, \alpha\beta) \vdash (q_1, x_1, \alpha_1\beta) \vdash$ .....$\vdash (q^1, \lambda, \gamma\beta)$

i.e., $(q_1, x, \alpha\beta) \vdash (q_1, \lambda, \gamma\beta)$

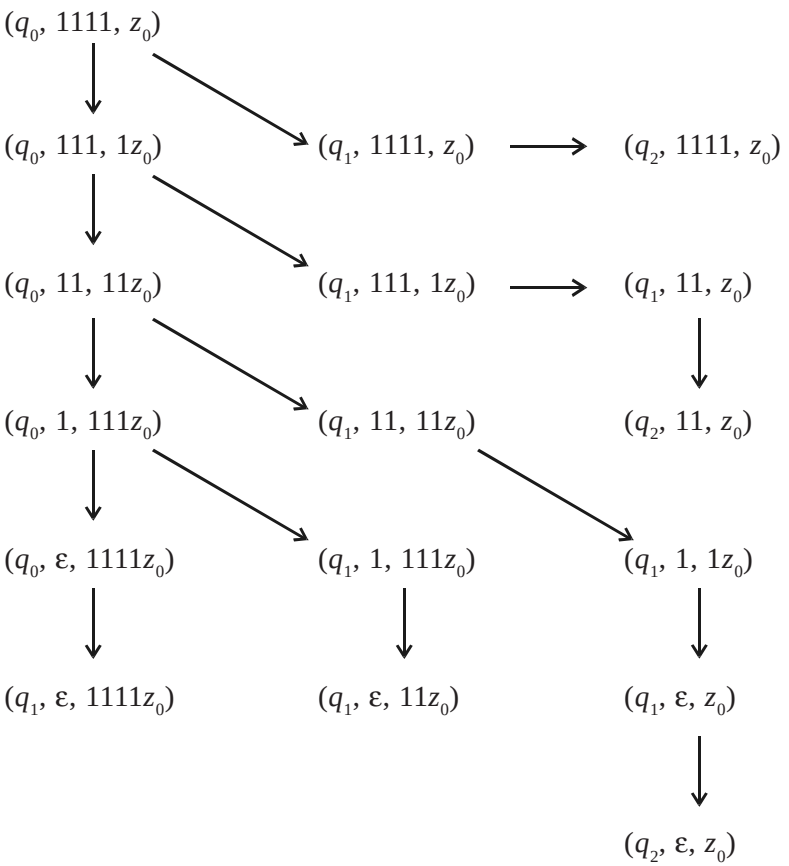
Note :

(i) The equation (3) implies (4)

(ii) In general equation (4) need not imply (3)

Example 6.3

Consider the Example 6.2. For the PDA, process the input 1111. Find the instantanceous descriptions.



Example 6.4

Suppose the PDA P = $(\{q, p\}), \{0, 1\}, \{z_0, X\}, \delta, q, z_0, \{p\})$ has the following transition function :

(i) $\delta(q, 0, z_0) = \{(q, Xz_0)\}$

(ii) $\delta(q, 0, X) = \{(q, XX)\}$

(iii) $\delta(q, 1, X) = \{(q, X)\}$

(iv) $\delta(q, \varepsilon, X) = \{(p, \varepsilon)\}$

(v) $\delta(p, \varepsilon, X) = \{(p, \varepsilon)\}$

(vi) $\delta(p, 1, X) = \{(p, XX)\}$

(vii) $\delta(p, 1, z_0) = \{(p, \varepsilon)\}$

Starting from initial ID (q, w, z_0), show all the reachable ID's when the input w is 01.

Solution :

$(q, 01, z_0) \vdash (q, 1, Xz_0) \vdash (q, \varepsilon, Xz_0) \vdash (p, \varepsilon, z_0)$

$(p, 1, z_0) \vdash (p, \varepsilon, \varepsilon)$

6.4 THE LANGUAGES OF A PDA

1. Acceptance by final state

Let M = (Q, $\sum$, Γ, δ,. q_0, z_0, F) be a PDA. Then, the language accepted by M is the set of strings such that

$L(M) = (w \mid (q_0, w, z_0) \vdash^* (P, \varepsilon, \gamma), P \in F, \gamma \in \Gamma^*)$. i.e., PDA M consumes w from the input and enters an accepting state.

2. Acceptance by empty stack

Let N(M) be the language accepted by M is given by

$N(M) = \{w \mid (q_0, w, z_0) \vdash^* (P, \varepsilon, \varepsilon)$ for some $P \in Q\}$.

Example 6.5

Consider a npda with

$Q = \{q_0, q_1, q_2, q_3\}$ = a finite set of states

$\sum = \{a, b\}$ = input alphabet

$\Gamma = \{0, 1\}$ = stack alphabet

$z = 0$ = stack start symbol

$F = \{q_3\}$ = set of final states

$\delta(q_0, a, 0) = \{(q_1, 10)\}$

$\delta(q_0, \lambda, 0) = \{(q_3, \lambda)\}$

$\delta(q_1, a, 1) = \{(q_1, 11)\}$

$\delta(q_1,b,1) = \{(q_2, \lambda)\}$

$\delta(q_2,b,1) = \{(q_2, \lambda)\}$

$\delta(q_2, \lambda, 0) = \{(q_3, \lambda)\}$

What can we say about the transition?

Solution :

(i) Transitions are not specified for all possible combinations of input and stack symbols.

Example : $\delta(q_0, b, 0)$ is not specified. Unspecified transition is null state.

(ii) The crucial transitions are

(a) $\delta(q_1, a, 1) = \{(q_1, 11)\}$

which adds a '1' to the stack when an '*a*' is read, and

(b) $\delta(q_2, b, 1) = \{(q_2,\lambda)\}$

which removes a 1 when a *b* is encountered.

(iii) The above steps count the number of *a*'s and match that count against the number of *b*'s. The control unit is in state q_1 until the first *b* is encountered, and it goes to state q_2.

(iv) After analysing the remaining transitions, *npda* will end in final state q_3 if and only if the input string is in the Language $L = \{a^n b^n : n \geq 0\}$

Example 6.6

Consider a npda with

$Q = \{q_0, q_1, q_2, q_3\}$ = set of states

$\sum = \{a, b\}$ = input alphabet

$\Gamma = \{0,1\}$ = stack alphabet

$z = 0$ = stack start symbol

$F = \{q_3\}$ = set of final states

The transitions are:

$\delta(q_0, a, 0) = \{(q_1,10), (q_3,\lambda)\}$

$\delta(q_0,\lambda,0) = \{(q_3,\lambda)\}$

$\delta(q_1,a,1) = \{(q_1,11)\}$

$\delta(q_1,b,1) = \{(q_2,\lambda)\}$

$\delta(q_2,b,1) = \{(q_2,\lambda)\}$

$\delta(q_2,\lambda,0) = \{(q_3,\lambda)\}$

What is the language accepted by this *npda*?

Solution :

This *npda* is similar to the one given in Example 6.5 except that

$\delta(q_0, a, 0) = \{(q_1, 10), (q_3, \lambda)\}$

Here $\delta(q_0, a, 0)$ can also go to (q_3, λ) which is an accepting state. So it can accept string *a*.

So the language accepted is $L = \{a^n b^n : n \geq 0\} \cup \{a\}$

Example 6.7

Construct a *npda* for the language

$L = \{w \in \{a,b\}^* : n_a(w) = n_b(w)\}$

Solution :

(i) It involves counting the number of *a*'s and *b*'s, which can be easily done with a stack.

(ii) The solution is

$M = (\{q_0, q_f\}, \{a, b\}, \{0,1,z\}, \delta, q_0, z, \{q_f\})$

with δ given by

$\delta(q_0, \lambda, z) = \{(q_f, z)\}$ ·····> (1)

$\delta(q_0, a, z) = \{(q_0, 0z)\}$ ·····> (2)

$\delta(q_0, b, z) = \{(q_0, 1z)\}$ ·····> (3)

$\delta(q_0, a, 0) = \{(q_0, 00)\}$ ·····> (4)

$\delta(q_0, b, 0) = \{(q_0, \lambda)\}$ ·····> (5)

$\delta(q_0, a, 1) = \{(q_0, \lambda)\}$ ·····> (6)

$\delta(q_0, b, 1) = \{(q_0, 11)\}$ ·····> (7)

We use symbol '0' to count number of *a*'s to match with *b*.

Similary we use symbol '1' to count number of *b*'s. to match with *a*.

For example, in processing string *baab*, the *npda* makes the moves

$(q_0, baab, z) \vdash (q_0, aab, 1z)$ ·····> by rule 3

$\vdash (q_0, ab, z)$ ·····> by rule 6

$\vdash (q_0, b, 0z)$ ·····> by rule 2

$\vdash (q_0, \lambda, z)$ ·····> by rule 5

$\vdash (q_f, z)$ ·····> by rule 1

and hence string is accepted.

Example 6.8

Construct a *npda* for accepting the language

$L = \{ww^R : w \in \{a, b\}^+\}$

Solution :

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

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

$\Sigma = \{a, b\}$

$\Gamma = \{z, a, b\}$

$F = \{q_2\}$

The transition function has several parts :

(i) set to push w on to stack.

$\delta(q_0, a, a) = \{(q_0, aa)\}$ ·····> (1)

$\delta(q_0, b, a) = \{(q_0, ba)\}$ ·····> (2)

$\delta(q_0, a, b) = \{(q_0, ab)\}$ ·····> (3)

$\delta(q_0, b, b) = \{(q_0, bb)\}$ ·····> (4)

$\delta(q_0, a, z) = \{(q_0, az)\}$ ·····> (5)

$\delta(q_0, b, z) = \{(q_0, bz)\}$ ·····> (6)

(ii) To find middle of the string, where *npda* switches from q_0 to state q_1

$\delta(q_0, \lambda, a) = \{(q_1, a)\}$ ·····> (7)

$\delta(q_0, \lambda, b) = \{(q_1, b)\}$ ·····> (8)

(iii) Set to match w^R against contents of the stack.

$\delta(q_1, a, a) = \{(q_1, \lambda)\}$ ·····> (9)

$\delta(q_1, b, b) = \{(q_1, \lambda)$ ·····> (10)

and finally

$\delta(q_1, \lambda, z) = \{(q_2, z)\}$ ·····> (11)

to recognise successful match.

The processing of string $w = abba$

$(q_0, abba, z) \vdash (q_0, bba, az)$ ·····> by rule 5

$\vdash$ (q_0, ba, baz) ·····> by rule 2

$\vdash$ (q_1, ba, baz) ·····> by rule 8

$\vdash$ (q_1, a, az) ·····> by rule 10

$\vdash$ (q_1, a, az) ·····> by rule 9

$\vdash$ (q_1, λ, z) ·····> by rule 9

$\vdash$ (q_2, λ) ·····> by rule 11

Hence string is accepted.

Note :

At 3rd move, to locate middle of the string, (q_0, ba, baz) we have two choices for next move

(i) $\delta(q_0, b, b) = \{(q_0, bb)\}$ (or)

(ii) $\delta(q_0, \lambda, b) = \{(q_1, b)\}$

Example 6.9

Consider a npda with

$A = (\{s_0, p, q\}, \{a,b\}, \{a,b,e\}, \delta, s_0, e\ \{q\})$

where δ is

$\delta(s_0, a, e) = \{(p,a)\}$ (1)

$\delta(p,a,a) = \{(p, aa)\}$ (2)

$\delta(p,a,e) = \{(p,a)\}$ (3)

$\delta(p,b,a) = \{(p,e), (q,e)\}$ (4)

process string *aaabbb*

Solution :

$(s_0, aaabbb, e) \Rightarrow (p, aabbb, a)$ by rule 1

$\vdash$ $(p, abbb, aa)$ by rule 2

$\vdash$ (p, bbb, aaa) by rule 2

$\vdash$ (p, bb, aa) by rule 4

$\vdash$ (p, b, a) by rule 5

$\vdash$ (q, e, e) by rule 5

Hence string is accepted

Example 6.10

Draw the presentation of PDA for the language L = $\{0^n1^n : n \geq 1\}$

Solution :

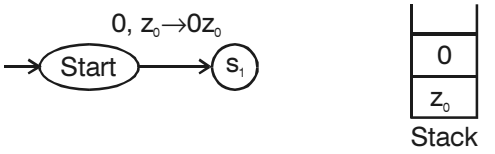
Step 1 :

1. Stack is empty
2. Finite control state is in start state

Step 2 :

When the input comes as '0', the state transition is from start state to s_1 and the stack symbol is '0'

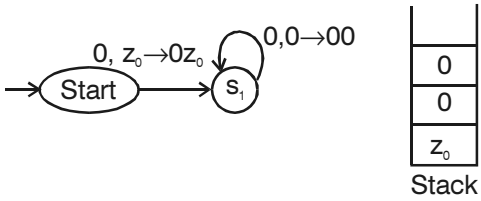
i.e. $\delta(\text{start}, 0, z_0) = \{(s_1, 0z_0)\}$



Step 3 :

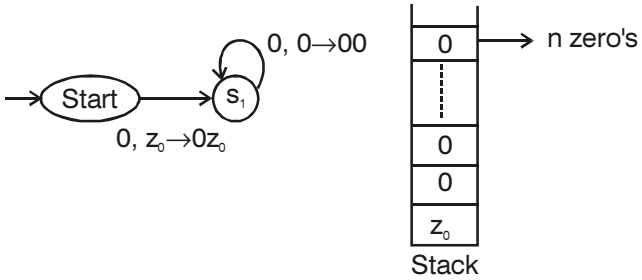
When more than one input comes as '0', the state transition is from the state s_1 to itself. Stack symbol is '0'.

i.e. $\delta(s_1,0,0) = \{s_1,00)\}$



Step 4:

When number of incoming zero's increases, stack size grows:



Step 5 :

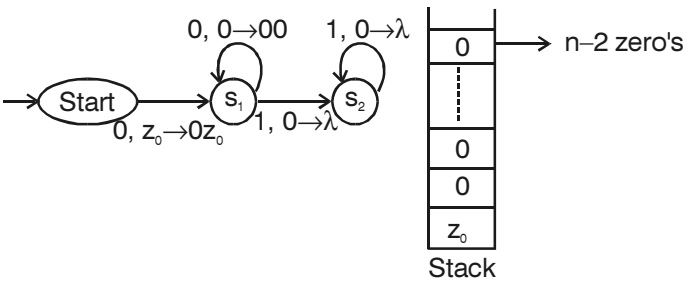
When the input comes as '1', the state transition is from state s_1 to state s_2 and empties (or) pops the top most symbol of the stack.

$\delta(s_1, 1, 0) = \{(s_2, \lambda)\}$

Step 6:

When more than one input comes as '1's, the state transition remains in s_2 itself and empties top most symbol of the stack.

$\delta(s_2, 1, 0) = \{(s_2, \lambda)\}$



The same operation is performed till all 1's are exhausted.

Step 7 :

When the input is empty (ε)(or)λ and the stack content is z_0, then the state transition is from state s_2 to state s_3

$\delta(s_2, \lambda, z_0) = \{(s_3, \lambda)\}$

Example 6.11

Construct a *npda* accepting L = {wcw^R : $w \in \{0,1\}^*$} by reaching final state.

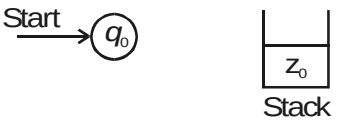
Solution :

w^R is reverse order of group of symbols. (For example w=01 w^R=10)

Step 1 :

Let the stack content be z_0

The control unit is in state q_0



Step 2:

The finite automata will be in state q_0 until it recognises a separator symbol 'c'.

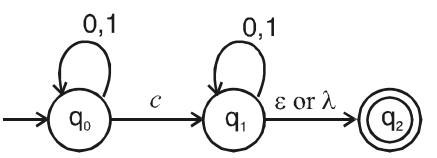
It checks whether the string $w \in \{0, 1\}$ and then pushes on to stack.

Step 3

Whenever the npda recognises a separator symbol 'c' the state changes from q_0 to state q_1 and no more symbols are pushed on to the stack.

Step 4 :

The symbols are read after the separator are compared with the elements in the stack and if they match then they are popped out from the stack.



Therefore the machine is as follows:

M = (Q, Σ, Γ, δ, q_0, z_0, F) where

Q = {q_0, q_1, q_2}

Σ = {0, 1}

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

$F = \{q_2\}$

where δ (or) rules are

(i) *Set of rules to push w on to stack is*

$R_1 : \delta(q_0, 0, z_0) = \{(q_0, 0z_0)\}$

$R_2 : \delta(q_0, 1, z_0) = \{(q_0, 1z_0)\}$

$R_3 : \delta(q_0, 0, 0) = \{(q_0, 00)\}$

$R_4 : \delta(q_0, 1, 1) = \{(q_0, 11)\}$

$R_5 : \delta(q_0, 0, 1) = \{(q_0, 01)\}$

$R_6 : \delta(q_0, 1, 0) = \{(q_0, 10)\}$

The above 6 rules are used to push all symbols {0,1} on to stack.

(ii) *To accept the separator symbol*

$R_7 : \delta(q_0, c, 0) = \{(q_1, 0)\}$

$R_8 : \delta(q_0, c, 1) = \{(q_1, 1)\}$

(iii) *Rules to match* w^R

$R_9 : \delta(q_1, 0, 0) = \{(q_1, \varepsilon)\}$

$R_{10} : \delta(q_1, 1, 1) = \{(q_1, \varepsilon)\}$

(iv) *To accept string*

$R_{11} : \delta(q_1, \lambda, z_0) = \{(q_2, z_0)\}$

Example 6.12

Consider a npda

$A = (\{q_0, q_1, q_f\}, \{a, b, c\}, \{a, b, z_0\}, \delta, q_0, z_0, \{q_f\})$

where δ is defined as

$R_1 : \delta(q_0, a, z_0) = \{(q_0, az_0)\}$

$R_2 : \delta(q_0, b, z_0) = \{(q_0, bz_0)\}$

$R_3 : \delta(q_0, a, a) = \{(q_0, aa)\}$

$R_4 : \delta(q_0, b, a) = \{(q_0, ba)\}$

$R_5 : \delta(q_0, a, b) = \{(q_0, ab)\}$

$R_6 : \delta(q_0, b, b) = \{(q_0, bb)\}$

$R_7 : \delta(q_0, c, a) = \{(q_1, a)\}$

R_8 : $\delta(q_0, c, b) = \{(q_1, b)\}$

R_9 : $\delta(q_0, c, z_0) = \{(q_1, z_0)\}$

R_{10} : $\delta(q_1, a, a) = \{(q_1, \lambda)\}$

R_{11} : $\delta(q_1, b, b) = \{(q_1, \lambda)\}$

R_{12} : $\delta(q_1, \lambda, z_0) = \{(q_f, z_0)\}$

(a) Explain δ

(b) Process input string *bacab*.

(c) Process input string *abcbb*

Solution :

(a) The explanation of δ (or) rule is as follows:

(i) By rule R_1 and R_2, PDA pushes the first symbol of the input string on to stack if it is *a* or *b*.

(ii) By rule R_3–R_6, the symbols of input string is pushed on to stack until it sees center marker '*c*'.

(iii) By rule R_7–R_9, on seeing '*c*' (center marker), PDA moves to state q_1 without making any change in stack.

(iv) By rule R_{10}, R_{11}, PDA erases top most symbol if it coincides with the current input symbol.

(v) By rule R_{12}, PDA reaches final state q_f only when all input symbols are exhausted and stack has only z_0.

(b) Processing *bacab*.

(i) Initially stack has z_0

(ii) Control unit is in state q_0

$\therefore (q_0, bacab, z_0) \vdash (q_0, acab, bz_0)$ ·····> by rule R_2

$\vdash (q_0, cab, abz_0)$ ·····> by rule R_5

$\vdash (q_1, ab, abz_0)$ ·····> by rule R_7

$\vdash (q_1, b, bz_0)$ ·····> by rule R_{10}

$\vdash (q_1, \lambda, z_0)$ ·····> by rule R_{11}

$\vdash (q_f, \lambda, z_0)$ ·····> by rule R_{12}

$\vdash (q_f, z_0)$ ·····> by rule R_{12}

Hence input string is accepted.

(c) Processing *abcbb*

(i) Initially stack has z_0

(ii) Control unit is in state q_0.

$\therefore (q_0, abcbb, z_0) \vdash (q_0, bcbb, az_0)$ ·····> by rule R_1

$\vdash (q_0, cbb, baz_0)$ ·····> by rule R_4

$\vdash (q_1, bb, baz_0)$ ·····> by rule R_8

$\vdash (q_1, b, az_0)$ ·····> by rule R_{11}

$\vdash$ It has to halt

because $\delta(q_1, b, a) = \phi$. Hence string is rejected.

Example 6.13

Construct a *pda* A accepting L = $\{wcw^T \mid w \in (a,b)^*\}$ by final state

Solution :

Let A = $(Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ be a *pda*. The set accepted by *pda* by final state is defined by

$T(A) = \{w \in \Sigma^* \mid (q_0, w, z_0) \vdash^* (q_f, \lambda, \alpha)$ for some $q_f \in F$ and $\alpha \in \Gamma^*\}$

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

$Q = \{q_0, q_1, q_f\}$ – set of states

$\Sigma = \{a, b, c\}$ – input alphabet

$\Gamma = \{a, b, z_0\}$ – stack alphabet

$F = \{q_f\}$ – final state

where δ (or) rule is defined as

R_1 : $\delta(q_0, a, z_0) = \{(q_0, az_0)\}$

R_2 : $\delta(q_0, b, z_0) = \{(q_0, bz_0)\}$

R_3 : $\delta(q_0, a, a) = \{(q_0, aa)\}$

R_4 : $\delta(q_0, b, a) = \{(q_0, ba)\}$

R_5 : $\delta(q_0, a, b) = \{(q_0, ab)\}$

R_6 : $\delta(q_0, b, b) = \{(q_0, bb)\}$

R_7 : $\delta(q_0, c, a) = \{(q_1, a)\}$

R_8 : $\delta(q_0, c, b) = \{(q_1, b)\}$

R_9 : $\delta(q_0, c, z_0) = \{(q_1, z_0)\}$

R_{10} : $\delta(q_1, a, a) = \{(q_1, \lambda)\}$

R_{11} : $\delta(q_1, b, b) = \{(q_1, \lambda)\}$

R_{12} : $\delta(q_1, \lambda, z_0) = \{(q_f, z_0)\}$

Let $w = a_1a_2 \ldots\ldots a_n$ where each a_i is either a or b.

Then we write

$(q_0, a_1a_2 \ldots\ldots a_n c w^T, z_0) \vdash (q_0, a_2 \ldots\ldots a_n c w^T, a_1 z_0)$ ·····> by rule R_1, R_2

$\vdash^* (q_0, c w^T, a_n a_{n-1} \ldots\ldots a_2 a_1 z_0)$ ·····> by rule R_1–R_6.

$\vdash (q_1, a_n a_{n-1} \ldots a_2\ a_1, a_n a_{n-1} \ldots a_1 z_0)$ ·····> by rule R_7, R_8

$\vdash^* (q_1, \lambda, z_0)$ ·····> by rule R_{10}, R_{11}

$\vdash (q_f, \lambda, z_0)$ ·····> by rule R_{12}

Therefore $wcw^T \in T(A)$

Example 6.14

Construct a *pda* A to accept the Language $L = \{wcw^T \mid w \text{ in } (a,b)^*\}$ by empty store.

Solution :

Let us recall acceptance by empty store.

Let $A = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ be a *pda*. The set N(A) accepted by null store is defined by $N(A) = \{w \in \Sigma^* \mid (q_0, w, z_0) \vdash^* (q, \lambda, \lambda)$ for some $q \in Q\}$

$A = (\{q_0, q_1, q_f\}, \{a, b, c\}, \{a, b, z_0\}, \delta, q_0, z_0, \{q_f\})$ is a *pda*, where δ is defined as

$R_1 : \delta(q_0, a, z_0) = \{(q_0, az_0)\}$

$R_2 : \delta(q_0, b, z_0) = \{(q_0, bz_0)\}$

$R_3 : \delta(q_0, a, a) = \{(q_0, aa)\}$

$R_4 : \delta(q_0, b, a) = \{(q_0, ba)\}$

$R_5 : \delta(q_0, a, b) = \{(q_0, ab)\}$

$R_6 : \delta(q_0, b, b) = \{(q_0, bb)\}$

$R_7 : \delta(q_0, c, a) = \{(q_1, a)\}$

$R_8 : \delta(q_0, c, b) = \{(q_1, b)\}$

$R_9 : \delta(q_0, c, z_0) = \{(q_1, z_0)\}$

$R_{10} : \delta(q_1, a, a) = \{(q_1, \lambda)\}$

$R_{11} : \delta(q_1, b, b) = \{(q_1, \lambda)\}$

$R_{12} : \delta(q_1, \lambda, z_0) = \{(q_f, z_0)\}$

To accept by empty store, we should make provision to erase z_0.

$\therefore R_{13} : \delta(q_f, \lambda, z_0) = \{(q_f, \lambda)\}$

Let $w = a_1 a_2 \ldots\ldots a_n$.

$(q_0, a_1 a_2 \ldots\ldots a_n c w^T, z_0) \vdash^* (q_0, c w^T, a_n a_{n-1} \ldots\ldots a_2 a_1 z_0)$ by rule R_1–R_6

$\vdash (q_1, a_n a_{n-1} \ldots\ldots a_2 a_1, a_n a_{n-1} \ldots\ldots a_2 a_1 z_0)$ by rule R_7, R_8

$\vdash^* (q, \lambda, z_0)$ by rule R_{10}, R_{11}

$\vdash (q_f, \lambda, z_0)$ by rule R_{12}

$\vdash (q_f, \lambda, \lambda)$ by rule R_{13}.

Hence $N(A) = \{wcw^T \mid w \in (a, b)^*\}$

6.4.1 Equivalence of Acceptance of PDA from Empty Stack to Final State

Theorem

If $A = (Q, \Sigma, \Gamma, \delta, q_0, z_0, \phi)$ is a *pda* accepting L by empty store, we can find a *pda* $B=(Q^1, \Sigma, \Gamma^1, \delta_B, q_0^1, z_0^1, F^1)$ which accepts L by final state

i.e. $L = N(A) = T(B)$

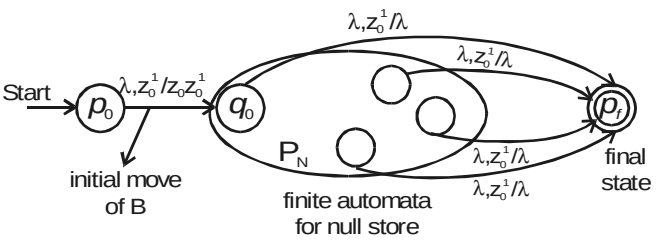
Proof

B is constructed in such a way that

(a) by the initial moves of B, it reaches an initial ID of A.

(b) by the final move of B, it reaches its final state, and

(c) all intermediate moves of B are in A.



Let us define B as follows:

$B = (Q^1, \Sigma, \Gamma^1, \delta_B, q_0^1, z_0^1, F^1)$

where

$Q^1 = Q \cup \{p_0, p_f\}$

$\Gamma^1 = \Gamma \cup \{z_0^1\}$

$F^1 = \{p_f\}$ = new final state (not in Q)

$q_0^1 = p_0$ = new start state.

z_0^1 = new start symbol for stack.

δ_B is given by rules.

$R_1 : \delta_B(p_0, \lambda, z_0^1) = \{(q_0, z_0 z_0^1)\}$

$R_2 : \delta_B(q, a, z) = \delta(q, a, z)$ for all q in Q, a in $\sum$ or λ and z in Γ

$R_3 : \delta_B(q, \lambda, z_0^1) = \{(p_f, \lambda)\}$

By rule R_1, the *pda* B moves from initial ID of B to an initial ID of A. R_1 gives a λ move. As a result of R_1, B moves to the initial state of A with the start symbol z_0 on top of the stack.

Rule R_2 is used to simulate A. Once B reaches an initial ID of A, R_2 is used to simulate moves of A. We can repeatedly apply R_2 until z_0^1 is pushed to the top of stack.

Rule R_3 is also an λ move. Using R_3, B moves to new final state p_f by erasing z_0^1 in stack.

We have to show N(A) = T(B)

Let $w \in$ N(A). Then by definition of N(A),

$(q_0, w, z_0) \overset{*}{\underset{A}{\vdash}} (q, \lambda, \lambda)$ for some $q \in Q$.

By theorem

$(q, x, \alpha) \overset{*}{\vdash} (p, y, \beta) \Rightarrow (q, xw, \alpha\gamma) \overset{*}{\vdash} (p, yw, \beta\gamma)$

we get

$(q_0, w, z_0 z_0^1) \overset{*}{\underset{A}{\vdash}} (q, \lambda, z_0^1)$

Since null store (or) empty store (δ) is a subset of δ_B

i.e.. $\delta \subset \delta_B$ we have

$(q_0, w, z_0 z_0^1) \overset{*}{\underset{B}{\vdash}} (q, \lambda, z_0^1)$

$\therefore$ we conclude that

$$(p_0, w, z_0^1) \underset{B}{\vdash} (q_0, w, z_0 z_0^1)$$
$$\overset{*}{\underset{B}{\vdash}} (q, \lambda, z_0^1)$$
$$\underset{B}{\vdash} (p_f, \lambda, \lambda)$$

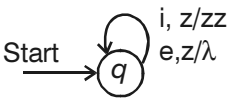
Example 6.15

Consider a pda A = ({q}, {i,e}, {z}, δ, q, z)

where δ is

$\delta(q, i, z) = \{(q, zz)\}$

$\delta(q, e, z) = \{(q, \lambda)\}$

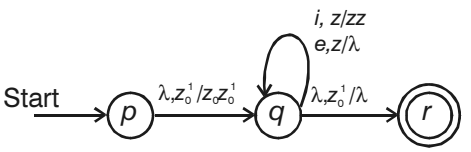


Find the equivalent *pda* (B) accepting by final state.

Solution :

Let B be *pda* accepting by final state.

Include new start state and final state.



$B = \{Q^1, \sum, \Gamma^1, \delta_B, p, z_0^1, F^1\}$

where

$Q^1 = \{p, q, r\}$

$\sum = \{i, e\}$ – input alphabet

$\Gamma^1 = \{z, z_0^1\}$ – stack alphabet

p = start state

$F^1 = \{r\}$

and δ_B is

$\delta_B(p, \lambda, z_0^1) = \{(q, z_0 z_0^1)\} \rightarrow$ by Rule 1 of theorem

$\delta_B(q, i, z) = \delta(q, i, z) = \{(q, zz)\}$

$\delta_B(q, e, z) = \delta(q, e, z) = \{(q, \lambda)\}$

Rule of empty stack acceptance.

$\delta_B(q, \lambda, z_0^1) = \{(r, \lambda\}) \rightarrow$ by Rule 3 of theorem.

6.4.2 Equivalance of Acceptance of PDA from Final State to Empty Stack

Theorem

If A = (Q, Σ, Γ, δ, q_0, z_0, F) accepts L by final state, we can find a *pda* B, accepting L by empty store.

i.e., L = T(A) = N(B)

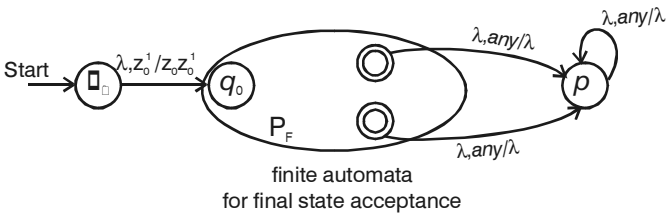
Proof

B is constructed from A in such a way that

(a) by the initial move of B an initial ID of A is reached.

(b) once B reaches an initial ID of A, it behaves like A until a final state of A is reached.

(c) when B reaches final state of A, it guesses whether the input string is exhausted. Then B simulates A or it erases all the symbols in stack.



B is as follows:

B = ($Q \cup \{p_0, p\}$, Σ, $\Gamma \cup \{z_0^1\}$, δ_B, p_0, z_0^1, ϕ) where

p_0 – new state not in Q.

δ_B is defined by R_1, R_2, R_3 and R_4 as:

$R_1 : \delta_B (p_0, \lambda, z_0^1) = \{(q_0, z_0 z_0^1)\}$

$R_2 : \delta_B(p, \lambda, z) = \{(p,\lambda)\}$ for all $z \in \Gamma \cup \{z_0^1\}$

$R_3 : \delta_B (q, a, z) = \delta(q, a, z)$ for all $a \in z$, $q \in Q$, $z \in \Gamma$.

$R_4 : \delta_B (q, \lambda, z) = \delta(q, \lambda, z) \cup \{(p, \lambda)\}$ for all $z \in \Gamma \cup \{z_0^1\}$ and $q \in F$.

Using R_1, B enters an initial ID of A and the start symbol z_0 is placed on top of stack.

R_2 is a λ-move, using this B erases all the symbols on stack.

R_3 is used to make B simulate A until it reaches the final state of A.

We have to show that T(A) = N(B)

Let $w \in$ T(A). Then

$(q_0, w, z_0) \overset{*}{\underset{A}{\vdash}} (q, \lambda, \alpha)$..... (1)

for some $q \in F$, $\alpha \in \Gamma^*$.

Since $\delta_B \subseteq \delta$ and by theorem

$(q, x, \alpha) \overset{*}{\vdash} (p, y, \beta) \Rightarrow (q, xw, \alpha\gamma) \overset{*}{\vdash} (p, yw, \beta\gamma)$ we can write (1) has

$(q_0, w, z_0 z_0^1) \overset{*}{\underset{B}{\vdash}} (q, \lambda, \alpha z_0^1)$

Then B can be computed has

$(p_0, w, z_0^1) \underset{B}{\vdash} (q_0, w, z_0 z_0^1) \overset{*}{\underset{B}{\vdash}} (q, \lambda, \alpha z_0^1) \overset{*}{\underset{B}{\vdash}} (p, \lambda, \lambda)$

Thus L = N(B) = T(A)

6.5 EQUIVALENCE OF PDA AND CFG

A language is generated by a CFG,

(a) if and only if it is accepted by a *pda* by empty stack

(b) if and only if it is accepted by a *pda* by final state.

We already know how to go between empty stack and final state.

6.5.1 From CFG's to PDA's

Theorem

For any context free language L, there exists an *pda* M such that L = L(M)

Proof

Let G = (V, T, P, S) be a grammar. There exists a Greibach Normal Form then we can construct *pda* which simulates left most derivations in this grammar.

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

$Q = \{q_0, q_1, q_f\}$ = set of states

Σ = terminals of grammar G

$\Gamma = V \cup \{z\}$ where V is the variables in grammar G

$F = \{q_f\}$ = final state.

The transition function will include

$\delta(q_0, \lambda, z) = \{(q_1, Sz)\}$, so that after the first move of M, the stack contains the start symbol S of the derivation. (The stack symbol z is a marker to allow us to detect the end of the derivation)

In addition, the set of transition rules is such that

(i) $\delta(q_1, \lambda, A) = \{(q, \alpha)\}$ for each $A \rightarrow \alpha$ in P

(ii) $\delta(q, a, a) = \{(q, \lambda)\}$ for each $a \in \Sigma$

Example 6.16

Construct a pda for the context free grammar

G = $(\{s\}, \{a, b\}, S, P)$, where $S \rightarrow aSbb \mid a$

Solution :

Step 1 :

We must construct equivalent context-free grammar with Greibach Normal Form by changing productions to

$S \rightarrow aSA \mid a$

$A \rightarrow bB$

$B \rightarrow b$

Step 2 :

Now we can construct *pda* in the following way

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

$Q = \{q_0, q_1, q_2\}$, q_0 – initial state, q_2 – final state

Σ – terminals of grammar = $\{a, b\}$

Γ – variables of grammar $\cup \{z_0\} = V_G \cup \{z_0\}$

F – $\{q_2\}$ = final state and δ is defined as:

(i) First the start symbol S is put on the stack

$\delta(q_0, \lambda, z_0) = \{(q_1, Sz_0)\}$

(ii) The production $S \rightarrow aSA|a$ will be simulated by *pda* as per $A \rightarrow \alpha$ rule.

$\delta(q_1, \lambda, S) = \{(q_1, aSA), (q_1, a)\}$

(iii) In a similar manner, the other productions are

$\delta(q_1, \lambda, A) = \{(q_1, bB)\}$ for rule $A \rightarrow bB$

$\delta(q_1, \lambda, B) = \{(q_1, b)\}$ for rule $B \rightarrow b$.

(iv) For each $a \in \Sigma$ the corresponding productions are:

$\delta(q_1, a, a) = \{(q_1, \lambda)\}$

$\delta(q_1, b, b) = \{(q_1, \lambda)\}$

(v) Then the end of the derivation is identified by the stack symbol

$\delta(q_1, \lambda, z) = \{(q_2, \lambda)\}$

Example 6.17

Design a *pda* that accepts the language of the grammar.

$S \rightarrow AB$

$A \rightarrow aA|e$

$B \rightarrow aBb|e$ (e= null state) and check for string *aaaabb*

Solution :

We can construct pda

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

$Q = \{q_0, f\}$

$\Sigma = \{a, b\}$ = terminals of grammar

Γ = variables of grammar $\cup \{z_0\} = \{S, A, B, z_0\}$

$F = \{f\}$ and the transition function δ is :

$R_1 : \delta(q_0, \lambda, z_0) = \{(f, Sz_0)\}$

$R_2 : \delta(f, \lambda, S) = \{(f, AB)\}$ for production 1

$R_3 : \delta(f, \lambda, A) = \{(f, aA)\}$ for production 2

$R_4 : \delta(f, \lambda, A) = \{(f, \lambda)\}$ for production 2

$R_5 : \delta(f, \lambda, B) = \{(f, aBb), (f,\lambda)\}$ for production 3

$R_6 : \delta(f, a, a) = \{(f, \lambda)\}$

$R_7 : \delta(f, b, b) = \{(f, \lambda)\}$

$R_8 : \delta(f, \lambda, z_0) = \{(f, \lambda)\}$

Processing of string *aaaabb*

$(q_0, aaaabb, z_0) \vdash (f, aaaabb, Sz_0)$ ·····> by Rule 1

$\vdash (f, aaaabb, ABz_0)$ ·····> by Rule 2

$\vdash (f, aaaabb, aABz_0)$·····> by Rule 3

$\vdash (f, aaabb, ABz_0)$ ·····> by Rule 6

$\vdash (f, aaabb, aABz_0)$ ·····> by Rule 3

$\vdash (f, aabb, ABz_0)$ ·····> by Rule 6

$\vdash (f, aabb, Bz_0)$ ·····> by Rule 4

$\vdash (f, aabb, aBbz_0)$ ·····> by Rule 5

$\vdash (f, abb, Bbz_0)$ ·····> by Rule 6

$\vdash (f, abb, aBbbz_0)$ ·····> by Rule 5

$\vdash (f, bb, Bbbz_0)$ ·····> by Rule 6

$\vdash (f, bb, bbz_0)$ ·····> by Rule 5

$\vdash (f, b, bz_0)$ ·····> by Rule 7

$\vdash (f, \lambda, z_0)$ ·····> by Rule 7

$\vdash (f, \lambda)$ ·····> by Rule 8

Hence the string is accepted.

Example 6.18

Given G = (V, T, P, S) where

V = {S}, T = {*a*, *b*, *c*}

P = {S $\rightarrow$ *a*S*a*, S $\rightarrow$ *b*S*b*, S $\rightarrow$ *c*}

Construct a *pda* and run for string *abcba*?

Solution :

M = (Q, Σ, Γ, δ, q_0, z_0, F) be a *pda*.

Q = {p, q} p = start state q = final state

Σ = {*a*, *b*, *c*} terminals of grammar

Γ = {S} $\cup$ {z_0}

F = {q}

and transition function δ is given below

$R_1 : \delta(p, \lambda, z_0) = \{(q, Sz_0)\}$

The start symbol S is put on the stack.

$R_2 : \delta(q, \lambda, S) = \{(q, aSa)\}$ production 1

$R_3 : \delta(q, \lambda, S) = \{(q, bSb)\}$ production 2

$R_4 : \delta(q, \lambda, S) = \{(q, c)\}$ production 3

$R_5 : \delta(q, a, a) = \{(q, \lambda)\}$

$R_6 : \delta(q, b, b) = \{(q, \lambda)\}$

$R_7 : \delta(q, c, c) = \{(q, \lambda)\}$

$R_8 : \delta(q, \lambda, z_0) = \{(q, \lambda)\}$

Processing of *abcba*

$(p, abcba, z_0) \vdash (q, abcba, Sz_0)$ ·····> by R_1

$\vdash (q, abcba, aSaz_0)$ ·····> by R_2

$\vdash (q, bcba, Saz_0)$ ·····> by R_5

$\vdash (q, bcba, bSbaz_0)$ ·····> by R_3

$\vdash (q, cba, Sbaz_0)$ ·····> by R_6

$\vdash (q, cba, cbaz_0)$ ·····> by R_4

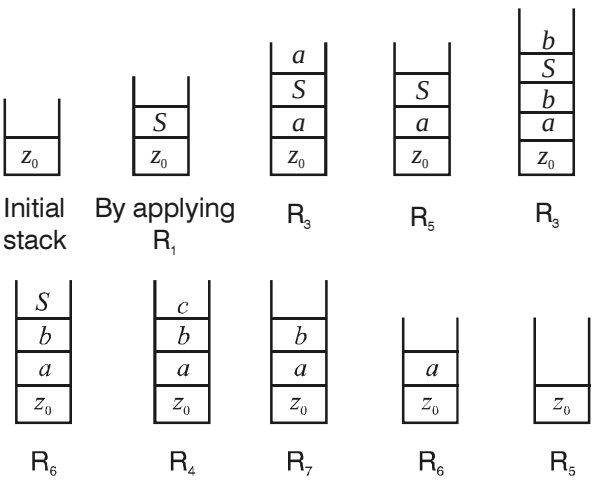
$\vdash (q, ba, baz_0)$ ·····> by R_7

$\vdash (q, a, az_0)$ ·····> by R_6

$\vdash (q, \lambda, z_0)$ ·····> by R_5

$\vdash (q, \lambda)$ ·····> by R_8

Hence string is accepted. The stack variable is as follows.



Example 6.19

Consider the grammar G = (V, T, P, S), where S $\rightarrow$ *a* A the productions are

$A \rightarrow aABC \mid bB \mid a$

$B \rightarrow b,$

$C \rightarrow c$

Find the *pda* and process *aaabc*.

Solution :

(i) Since the grammar is in Greibach Normal Form, we can use the theorem directly.

(ii) The transition rules are

$R_1 : \delta(q_0, \lambda, z) = \{(q_1, Sz)\}$

$R_2 : \delta(q_1, \lambda, z) = \{(q_2, \lambda)\}$

for productions the transition rules are

$R_3 : \delta(q_1, \lambda, S) = \{(q_1, aA)\}$ for production 1

$R_4 : \delta(q_1, \lambda, A) = \{(q_1, aABC), (q_1, bB), (q_1, a)\}$ for production 2

$R_5 : \delta(q_1, \lambda, B) = \{(q_1, b)\}$ for production 3

$R_6 : \delta(q_1, \lambda, C) = \{(q_1, c)\}$ for production 4

$R_7 : \delta(q_1, a, a) = \{(q_1, \lambda)\}$

$R_8 : \delta(q_1, b, b) = \{(q_1, \lambda)\}$

$R_9 : \delta(q_1, c, c) = \{(q_1, \lambda)\}$

Processing string *aaabc*

$(q_0, aaabc, z) \vdash (q_1, aaabc, Sz)$ ·····> by rule R_1

$\vdash (q_1, aaabc, aAz)$ ·····> by rule R_3

$\vdash (q_1, aabc, Az)$ ·····> by rule R_7

$\vdash (q_1, aabc, aABCz)$ ·····> by rule R_4

$\vdash (q_1, abc, ABCz)$ ·····> by rule R_7

$\vdash (q_1, abc, aBCz)$ ·····> by rule R_4

$\vdash (q_1, bc, BCz)$ ·····> by rule R_7

$\vdash (q_1, bc, bCz)$ ·····> by rule R_5

$\vdash (q_1, c, Cz)$ ·····> by rule R_8

$\vdash (q_1, c, cz)$ ·····> by rule R_6

$\vdash (q_1, \lambda, z)$ ·····> by rule R_9

$\vdash (q_1, \lambda)$ ·····> by rule R_2

Example 6.20

Convert the grammar

$S \rightarrow 0S1 \mid A$

$A \rightarrow 1A0 \mid S \mid \varepsilon$

to a PDA that accepts the same langauge 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, $\sum$, Γ, δ, q_0, z_0, ϕ), where

Q = {*p*, *q*}

$\sum$ = T = {0, 1} = terminals of the grammar

G = {V} $\cup$ {z_0} = {S, A, 0, 1, z_0}

The transition function is

(i) $\delta(p, \varepsilon, z_0) = (q, S)$

For each production

(ii) $\delta(q, \varepsilon, S) = (q, 0S1)$

(iii) $\delta(q, \varepsilon, S) = (q, A)$

(iv) $\delta(q, \varepsilon, A) = (q, 1A0)$

(v) $\delta(q, \varepsilon, A) = (q, S)$

(vi) $\delta(q, \varepsilon, S) = (q, \varepsilon)$

For each terminal

(vii) $\delta(q, 0, 0) = (q, \varepsilon)$

(viii) $\delta(q, 1, 1) = (q, \varepsilon)$

ε or λ are same, it pops stack top element

Example 6.21

Construct a PDA for the Grammar

$I \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$E \rightarrow I \mid E * E \mid E + E \mid (E)$

Solution :

PDA P (Q, Σ, Γ, δ, start state, z_0, final state), where

Q = {p, q} $p \rightarrow$ start state

$q \rightarrow$ final state

Σ = Terminals of grammar

= {a, b, 0, 1, (,), +, *}

G = {V} $\cup$ {z_0}

= {I, E, a, b, 0, 1, (,), +, *}

F = q

The transition rules are

(i) $\delta(p, \varepsilon, z_0) = \{(q, E)\}$

For each production

(ii) $\delta(q, \varepsilon, E) = \{(q, I), (q, E*E), (q, E+E), (q, (E))\}$

(iii) $\delta(q, \varepsilon, I) = \{(q, a), (q, b), (q, Ia), (q, Ib), (q, I0), (q, I1)\}$

For each terminal

(iv) $\delta(q, a, a) = (q, \lambda)$

(v) $\delta(q, b, b) = (q, \lambda)$

(vi) $\delta(q, 0, 0) = (q, \lambda)$

(vii) $\delta(q, 1, 1) = (q, \lambda)$

(viii) $\delta(q, (, () = (q, \lambda)$

(ix) $\delta(q,),)) = (q, \lambda)$

(x) $\delta(q, +, +) = (q, \lambda)$

(xi) $\delta(q, *, *) = (q, \lambda)$

6.5.2 From PDA's to CFG's

Theorem

If L is N(M) for some PDA M, then L is CFL.

Proof

1. It has a single final state q_f iff the stack is empty.
2. All transitions must have the form

   $\delta(q_i, a, A) = \{C_1, C_2 ..... C_n\}$, where

   $\delta(q_i, a, A) = (q_j, \lambda)$ $\rightarrow (1)$

   $\delta(q_i, a, A) = (q_j, BC)$ $\rightarrow (2)$

That is, each move either increases or decreases the stack content by a single symbol.

Given $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, \{q_f\})$ satisfies the condition (1) and (2)

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

V - elements of the form $[q, A, p]$, q and p in Q and A in Γ

$T = \Sigma$

S - start symbol

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

P consists of: $u, v \in \Sigma^*$

$A, X \in \Gamma^*$, $q_i, q_j \in Q$

$(q_i, uv, AX) \vdash^* (q_j, v, X)$

implies that $(q_i, A, q_j) \rightarrow u$

Consider $[q_i, A, q_k] \rightarrow a\ [q_j, B, q_l]\ [q_l, C, q_k]$

The corresponding transition for PDA is

$\delta(q_i, a, A) = \{(q_j, B\ C).....\}$

Similarly if $[q_i, A, q_j] \rightarrow a$ then the corresponding transition is

$\delta(q_i, a, A) = \{(q_j, \lambda)\}$

For all sentential forms leading to a terminal string, the argument holds true.

The conclusion is

$(q_0, w, z_0) \vdash^* (q_f, \lambda, \lambda)$ is true iff $(q_0 z_0 q_f) \overset{*}{\Rightarrow} w$

consequently L(M) = L(G)

Example 6.22

Consider the pda with transitions

$\delta(q_0, a, z) = \{(q_0, Az)\}$

$\delta(q_0, a, A) = \{(q_0, A)\}$

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

$\delta(q_1, \lambda, z) = \{(q_2, \lambda)\}$

find the equivalent CFG.

Solution :

Step 1 :

Check the two restrictions given prior in the theorem.

Here q_0 is initial state and q_2 is final state, hence restriction 1 is satisfied.

Restriction 2 is not satisfied, for transistion function 2

(i.e.) $\delta(q_0,a,A) = \{(q_0,A)\}$

Step 2 :

Hence we introduce state q_3 and an intermediate step in which we first remove A from the stack, then replace it in the next move.

Hence new transition rules are

$R_1 : \delta(q_0, a, z) = \{(q_0, Az)\}$

$R_2 : \delta(q_3, \lambda, z) = \{(q_0, Az)\}$

$R_3 : \delta(q_0, a, A) = \{(q_3, \lambda)\}$

$R_4 : \delta(q_0, b, A) = \{(q_1, \lambda)\}$

$R_5 : \delta(q_1, \lambda, z) = \{(q_2, \lambda)\}$

For $\delta(q_0, a, A) = \{(q_3, \lambda)\}$ the corresponding production in CFG is $q_0Aq_3 \rightarrow a$

For $\delta(q_0, b, A) = \{(q_1, \lambda)\}$ the corresponding production in CFG is $q_0Aq_1 \rightarrow b$

For $\delta(q_1, \lambda, z) = \{(q_2, \lambda)\}$ the corresponding production in CFG is $q_1zq_2 \rightarrow \lambda$

For $\delta(q_0, a, z) = \{(q_0, Az)\}$ the corresponding productions are

$(q_0zq_0) \rightarrow a(q_0Aq_0)(q_0zq_0) \mid a(q_0Aq_1)(q_1zq_0) \mid a(q_0Aq_2)(q_2zq_0) \mid a\,(q_0Aq_3)(q_3zq_0)$

$(q_0zq_1) \rightarrow a(q_0Aq_0)(q_0zq_1) \mid a(q_0Aq_1)(q_1zq_1) \mid a\,(q_0Aq_2)\,(q_2zq_1) \mid a\,(q_0Aq_3)\,(q_3zq_1)$

$(q_0zq_2) \rightarrow a(q_0Aq_0)(q_0zq_2) \mid a(q_0Aq_1)\,(q_1zq_2) \mid a\,(q_0Aq_2)(q_2zq_2) \mid a(q_0Aq_3)\,(q_3zq_2)$

$(q_0zq_3) \rightarrow a(q_0Aq_0)(q_0zq_3) \mid a(q_0Aq_1)(q_1zq_3) \mid a(q_0Aq_2)(q_2zq_3) \mid a(q_0Aq_3)(q_3zq_3)$

For $\delta(q_3, \lambda, z) = \{(q_0, Az)\}$ the corresponding productions are

$(q_3zq_0) \rightarrow (q_0Aq_0)(q_0zq_0) \mid (q_0Aq_1)(q_1zq_0) \mid (q_0Aq_2)(q_2zq_0) \mid (q_0Aq_3)(q_3zq_0)$

$(q_3zq_1) \rightarrow (q_0Aq_0)(q_0zq_1) \mid (q_0Aq_1)(q_1zq_1) \mid (q_0Aq_2)(q_2zq_1) \mid (q_0Aq_3)(q_3zq_1)$

$(q_3zq_2) \rightarrow (q_0Aq_0)(q_0zq_2) \mid (q_0Aq_1)(q_1zq_2) \mid (q_0Aq_2)(q_2zq_2) \mid (q_0Aq_3)(q_3zq_2)$

$(q_3zq_3) \rightarrow (q_0Aq_0)(q_0zq_3) \mid (q_0Aq_1)(q_1zq_3) \mid (q_0Aq_2)(q_2zq_3) \mid (q_0Aq_3)(q_3zq_3)$

The start variable is (q_0zq_2).

The string *aab* is accepted by the *pda*, with successive configurations

$(q_0, aab, z) \vdash (q_0, ab, Az)$ ·····> by rule R_1

$\vdash (q_3, b, z)$ ·····> by rule R_3

$\vdash (q_0, b, Az)$ ·····> by rule R_2

$\vdash (q_1, \lambda, z)$ ·····> by rule R_4

$\vdash (q_2, \lambda, \lambda)$ ·····> by rule R_5

The corresponding derivation in CFG is (q_0zq_2) - start state

$\therefore (q_0zq_2) \Rightarrow a(q_0Aq_3)(q_3zq_2)$

$\Rightarrow aa(q_3zq_2)$ (since $q_0Aq_3 \rightarrow a$)

$\Rightarrow aa(q_0Aq_1)\ (q_1zq_2)$

$\Rightarrow aab(q_1zq_2)$ (since $q_0Aq_1 \rightarrow b$)

$\Rightarrow aab$ (since $q_1zq_2 \rightarrow \lambda$)

6.6 DETERMINISTIC PUSHDOWN AUTOMATA AND DETERMINISTIC CONTEXT FREE LANGUAGES

Definition 1

A *pushdown automaton* $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ is said to be *deterministic* if

1. $\delta(q,a,b)$ contains at most one element
2. if $\delta(q,\lambda,b)$ is not empty, then

   $\delta(q, c, b)$ must be empty for

   every $c \in \Sigma$, $q \in Q$, $b \in \Gamma$.

The first condition says that for any given input symbol and any stack top, at most one move can be made.

The second condition says that when a λ-move is possible for some configuration, no input consuming alternative is available.

Definition 2

A language L is said to be *deterministic context free language*, if and only if there exists a *deterministic pushdown automata* (dpda) M such that L=L(M).

Example 6.23

The language L = $\{a^n b^n : n>0\}$ is a deterministic context-free language.

The *pda* M = $(\{q_0, q_1, q_2\}, \{a, b\}, \{0, 1\}, \delta, q_0, z_0, \{q_0\})$ with

$\delta(q_0, a, z_0) = \{(q_1, 1z_0)\}$

$\delta(q_1, a, 1) = \{(q_1, 11)\}$

$\delta(q_1, b, 1) = \{(q_2, \lambda)\}$

$\delta(q_2, b, 1) = \{(q_2, \lambda)\}$

$\delta(q_2, \lambda, z_0) = \{(q_0, \lambda)\}$

accepts given language. It satisfies the definition 1 and is therefore deterministic.

The NPDA for L = $\{a^n b^n \mid n \geq 0\}$ is given as:

M = $\{(q_0, q_1, q_2, q_3\}, \{a,b\}, \{0,1\}, \delta, q_0, z_0, \{q_3\})$

and transitions are:

$\delta(q_0,a,z_0\} = \{(q_1,10)\}$

$\delta(q_0,\lambda,z_0) = \{(q_3,\lambda)\}$

$\delta(q_1,a,1) = \{(q_1, 11)\}$

$\delta(q_1,b,1) = \{(q_2,\lambda)\}$

$\delta(q_2,b,1) = \{(q_2,\lambda)\}$

$\delta(q_2,\lambda,z_0) = \{(q_3,\lambda)\}$

This is not *dpda*, because it violates condition 2 in definition 1.

Similarly look at Example 6.8 where

L = $\{ww^R : w \in \{a,b\}^+\}$ is not deterministic because it violates condition 2 in definition 1

i.e. $\delta(q_0, a, a) = \{(q_0, aa)\}$

and $\delta(q_0, \lambda, a) = \{(q_1, a)\}$

6.7 SOLVED PROBLEMS

1. Construct a PDA that accpets the following language

$L = \{w : n_a(w) = n_b(w)+1\}$

Solution :

Let A be the PDA

$A = \{Q, \Sigma, \Gamma, \delta, q_0, z_0, F\}$ where

$Q = \{q_0, q_1, q_2, q_f\}$ = set of states

$\Sigma = \{a, b\}$ = input alphabet

$\Gamma = \{a, b, z_0\}$ = stack alphabet

z_0 = stack start symbol

$F = \{q_f\}$

The δ is defined as follows:

$R_1 : \delta(q_0, a, z_0) = \{(q_1, az_0)\}$

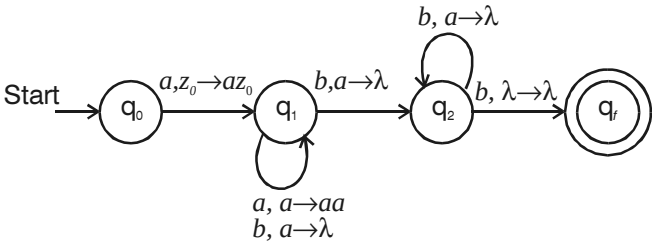
$R_2 : \delta(q_1, a, a) = \{(q_1, aa)\}$

$R_3 : \delta(q_1, b, a) = \{(q_2, \lambda)\}$

$R_4 : \delta(q_2, b, a) = \{(q_2, \lambda)\}$

$R_5 : \delta(q_2, b, \lambda) = \{(q_f, \lambda)\}$

The pda is as follows :



2. Construct a PDA that accepts the following language

$L = \{a^n b^{n+m} c^m : n \geq 0, m \geq 1\}$

Solution :

Let A be a PDA

$A = \{Q, \Sigma, \Gamma, \delta, q_0, z_0, F\}$ where

$Q = \{q_0, q_1, q_2, q_3, q_f\}$ = set of states

$\sum = \{a, b, c\}$ input alphabet

$\Gamma = \{a, b, z_0\}$ stack alphabet

$F = \{q_f\}$

The transition function δ is defined as:

$R_1 : \delta(q_0, a, z_0) = \{(q_1, az_0)\}$

$R_2 : \delta(q_1, a, a) = \{(q_1, aa)\}$

$R_3 : \delta(q_1, b, a) = \{(q_2, \lambda)\}$

$R_4 : \delta(q_2, b, a) = \{(q_2, \lambda)\}$

$R_5 : \delta(q_2, b, z_0) = \{(q_2, bz_0)\}$

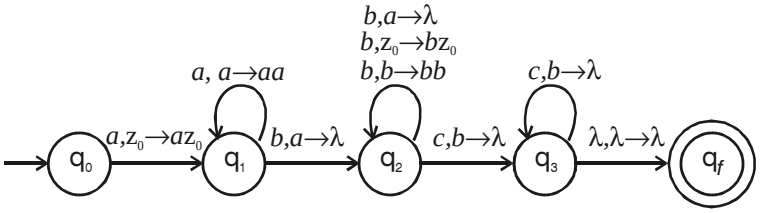
$R_6 : \delta(q_2, b, b) = \{(q_2, bb)\}$

$R_7 : \delta(q_2, c, b) = \{(q_3, \lambda)\}$

$R_8 : \delta(q_3, c, b) = \{(q_3, \lambda)\}$

$R_9 : \delta(q_3, \lambda, \lambda) = \{(q_f, \lambda)\}$

The pda for L is



3. Construct a PDA by accepting $\{a^n b^{2n} : n \geq 1\}$ by final state.

Solution :

Let A be a PDA

$A = \{Q, \sum, \Gamma, \delta, q_0, z_0, F\}$ where

$Q = \{q_0, q_1, q_2, q_3, q_4\}$ = set of states

$\sum = \{a, b\}$ = input alphabet

$\Gamma = \{a, b, z_0\}$ = stack alphabet

$F = \{q_4\}$

The PDA should accept input strings with symbols *a* and *b* where number of *b*'s is twice that of *a*'s.

It is implemented in a way that whenever symbol '*a*' is read in the input string it is pushed on to the stack and when two *b*'s are read in the input string the top most symbol '*a*' is popped or removed from stack. That is we are trying to match two *b*'s with a single '*a*'.

Regarding transition diagram, we can store *a*'s in the stack until the symbol '*b*' is read in the input symbol.

When first '*b*' symbol in the input string is read, there is a change in state, but no change in the stack.

When second '*b*' symbol is read, there is a change in state, and topmost symbol '*b*' in stack is popped out.

The transition function δ is

$\delta(q_0, a, z_0) = \{(q_1, az_0)\}$

$\delta(q_1, a, a) = \{(q_1, aa)\}$

$\delta(q_1, b, a) = \{(q_2, a)\}$

$\delta(q_2, b, a) = \{(q_3, \lambda)\}$

$\delta(q_3, b, a) = \{(q_2, a)\}$

$\delta(q_2, b, a) = \{(q_3, \lambda)\}$

$\delta(q_3, \lambda, z_0) = \{(q_4, z_0)\}$

Consider an input string *aabbbb*, the transitions are

$\delta(q_0, a, z_0) = \{(q_1, az_0)\}$

$\delta(q_1, a, a) = \{(q_1, aa)\}$

$\delta(q_1, b, a) = \{(q_2, a)\}$

$\delta(q_2, b, a) = \{(q_3, \lambda)\}$

$\delta(q_3, b, a) = \{(q_2, a)\}$

$\delta(q_2, b, a) = \{(q_3, \lambda)\}$

$\delta(q_3, \lambda, z_0) = \{(q_4, z_0)\}$

Since it halts in final state, the string is accepted.

4. Construct a PDA accepting $L = \{a^n\, b^m\, a^n : m, n \geq 1\}$ by empty store.

Solution :

The language consists of a set of *a*'s, followed by any number of *b*'s, and then followed by same set of *a*'s.

The number of *a*'s in the set appearing before *b*'s should be equal to the number of *a*'s appearing in the second set after *b*'s.

Steps :

(i) Store all *a*'s in the string until symbol '*b*' is read in input string.

$\delta(q_0, a, z_0) = (q_1, az_0)$

$\delta(q_1, a, a) = (q_1, aa)$

(ii) On seeing '*b*' in input string, there is change in state (to indicate that the symbol is sensed) and there is no change in stack.

$\delta(q_1, b, a) = (q_2, a)$

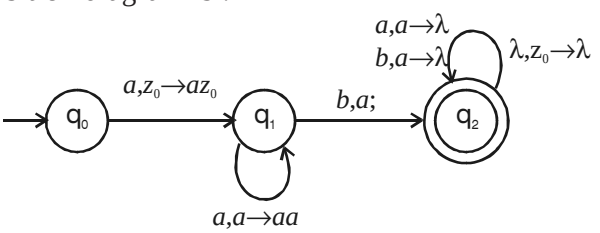
$\delta(q_2, b, a) = (q_2, a)$

(iii) Once all the *b*'s in the input string is exhausted, the remaining *a*'s in the input string is matched with the *a*'s present in the stack.

$\delta(q_2, a, a) = (q_2, \lambda)$

$\delta(q_2, \lambda, z_0) = (q_2, \lambda)$

The transition diagram is :



Consider the input string *aabaa*

$\delta(q_0, a, z_0) = (q_1, az_0)$

$\delta(q_1, a, a) = (q_1, aa)$

$\delta(q_1, b, a) = (q_2, a)$

$\delta(q_2, a, a) = (q_2, \lambda)$

$\delta(q_2, \lambda, z_0) = (q_2, \lambda)$

Hence the string is accepted by empty store.

5. Construct a PDA accepting $L = \{a^n b^{3n} : n \geq 1\}$ by empty store.

Solution :

PDA should accept input strings with symbols '*a*' and 'b' where number of *b*'s is thrice that of *a*'s.

$\delta(q_0, a, z_0) = (q_1, a\ z_0)$

$\delta(q_1, a, a) = (q_1, aa)$

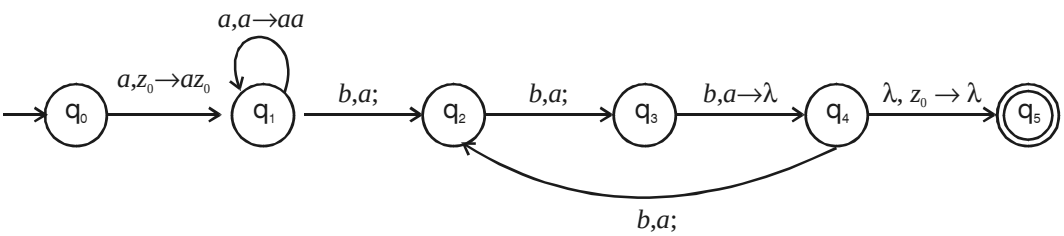
$\delta(q_1, b, a) = (q_2, a)$

$\delta(q_2, b, a) = (q_3, a)$

$\delta(q_3, b, a) = (q_4, \lambda)$

$\delta(q_4, b, a) = (q_2, a)$

$\delta(q_4, \lambda, z_0) = (q_5, \lambda)$



Consider the input string *aabbbbbb*

$\delta(q_0, a, z_0) = (q_1, az_0)$

$\delta(q_1, a, a) = (q_1, aa)$

$\delta(q_1, b, a) = (q_2, a)$

$\delta(q_2, b, a) = (q_3, a)$

$\delta(q_3, b, a) = (q_4, \lambda)$

$\delta(q_4, b, a) = (q_2, a)$

$\delta(q_2, b, a) = (q_3, a)$

$\delta(q_3, b, a) = (q_4, \lambda)$

$\delta(q_4, \lambda, z_0) = (q_5, \lambda)$ it halts in empty store and hence string is accepted.

6. Design a PDA that accepts $L = \{a^n\, b^m\, c^n : n, m \geq 1\}$

Solution :

The language consists of a set of *a*'s followed by any number of *b*'s and then followed by set of *c*'s.

The number of *a*'s in the set appearing before *b*'s should be equal to number of *c*'s appearing in the set after *b*'s.

$\delta(q_0, a, z_0) = (q_1, az_0)$
$\delta(q_1, a, a) = (q_1, aa)$ } push *n* *a*'s on to stack

$\delta(q_1, b, a) = (q_2, a)$
$\delta(q_2, b, a) = (q_2, a)$ } skip occurrence of '*b*', by changing state

$\delta(q_2, c, a) = (q_3, \lambda)$
$\delta(q_3, c, a) = (q_3, \lambda)$ } match occurrence of '*c*' with '*a*'.

$\delta(q_3, \lambda, z_0) = (q_f, z_0)$

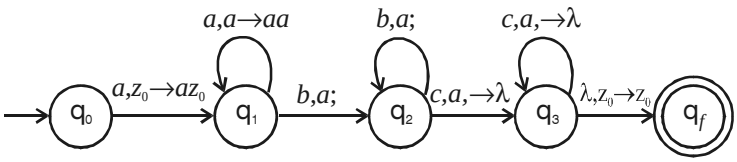
Let A be PDA

$A = \{Q, \Sigma, \Gamma, \delta, q_0, z_0, F\}$

$Q = \{q_0, q_1, q_2, q_3, q_f\}$ $\quad \Gamma = \{a, z_0\}$

$\Sigma = \{a, b, c\}$ $\quad F = \{q_f\}$

Transition diagram is :



7. Design a PDA to accept a language of the form $L = \{a^n\, b^m\, c^m\, d^n \mid m, n \geq 1\}$

Solution :

The transition functions are:

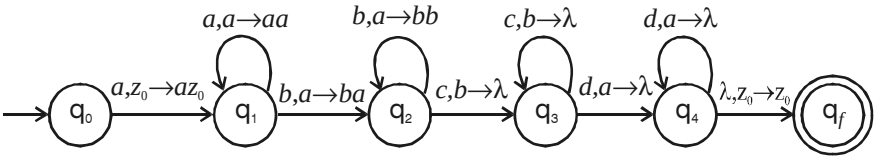
$\delta(q_0, a, z_0) = (q_1, a\, z_0)$
$\delta(q_1, a, a) = (q_1, aa)$ } push *n* *a*'s on to stack

$\delta(q_1, b, a) = (q_2, ba)$
$\delta(q_2, b, b) = (q_2, bb)$ } push *m* *b*'s on to stack

$\delta(q_2, c, b) = (q_3, \lambda)$
$\delta(q_3, c, b) = (q_3, \lambda)$ } match occurrence of '*c*' with occurrence of '*b*'.

$\delta(q_3, d, a) = (q_4, \lambda)$
$\delta(q_4, d, a) = (q_4, \lambda)$ } match occurrence of '*d*' with occurrence of '*a*'.
$\delta(q_4, \lambda, z_0) = (q_f, z_0)$

Transition diagram is



8. Construct a PDA to recognize the language L={$a^n cb^n | n \geq 1$}

Solution :

The transition function δ are:

$\delta(q_0, a, z_0) = (q_1, az_0)$
$\delta(q_1, a, a) = (q_1, aa)$ } push *n* *a*'s on to stack
$\delta(q_1, c, a) = (q_2, \lambda)$ Skip occurrence of '*c*' by changing state
$\delta(q_2, b, a) = (q_2, \lambda)$
$\delta(q_2, \lambda, z_0) = (q_f, z_0)$ } match occurrence of *b*'s with occurrence of *a*'s.

Let A be a PDA

$A = \{Q, \Sigma, \Gamma, \delta, q_0, z_0 F\}$

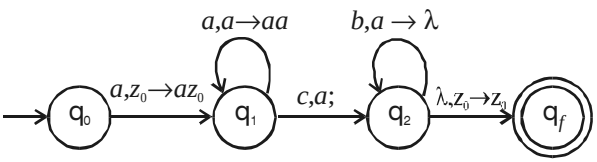
$Q = \{q_0, q_1, q_2, q_f\}$

$\Sigma = \{a, b, c\}$

$\Gamma = \{a, z_0\}$

$F = \{q_f\}$

Transition diagram is



9. Construct a PDA to recognize the language

$L = \{a^m\, b^n\, a^{m+n} \mid m, n \geq 1\}$

Solution :

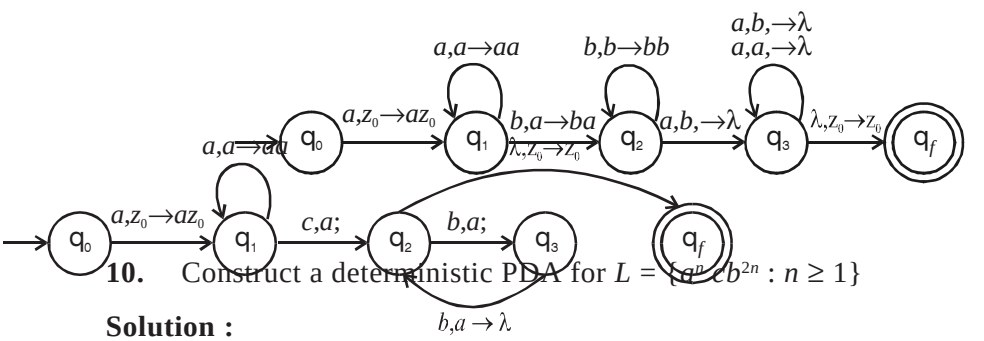
The transition functions are:

$\delta(q_0, a, z_0) = (q_1, a\, z_0)$
$\delta(q_1, a, a) = (q_1, aa)$
} push m a's on to stack

$\delta(q_1, b, a) = (q_2, ba)$
$\delta(q_2, b, b) = (q_2, bb)$
} push n b's on to stack

$\delta(q_2, a, b) = (q_3, \lambda)$
$\delta(q_3, a, b) = (q_3, \lambda)$
$\delta(q_3, a, a) = (q_3, \lambda)$
} match $m+n$ occurrence of a's with m occurrence of a & n occurrence of b.

$\delta(q_3, \lambda, z_0) = (q_f, z_0)$



10. Construct a deterministic PDA for $L = \{a^n\, cb^{2n} : n \geq 1\}$

Solution :

The transition function (δ) are:

$\delta(q_0, a, z_0) = (q_1, az_0)$
$\delta(q_1, a, a) = (q_1, aa)$
} push n a's on to stack

$\delta(q_1, c, a) = (q_2, a)$
} skip occurrence of 'c' by changing state

$\delta(q_2, b, a) = (q_3, a)$
$\delta(q_3, b, a) = (q_2, \lambda)$
} match occurrence of 2 'b' with occurrence of single 'a'.

$\delta(q_2, \lambda, z_0) = (q_f, z_0)$

11. Construct a PDA that accepts L = $\{a^{n+1}\, b^{2n} : n \geq 0\}$

Solution :

The transition function (δ) are :

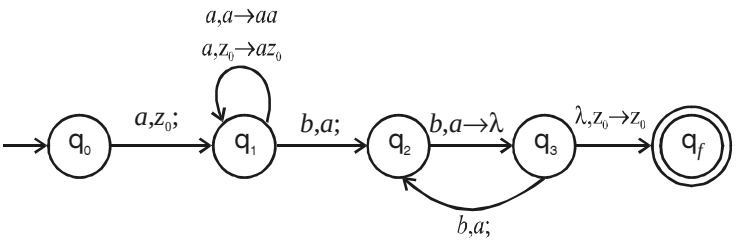
$\delta(q_0, a, z_0) = (q_1, z_0)$
$\delta(q_1, a, z_0) = (q_1, az_0)$
$\delta(q_1, a, a) = (q_1, aa)$

} push *a*'s on to stack with skipping first occurrence of '*a*'

$\delta(q_1, b, a) = (q_2, a)$
$\delta(q_2, b, a) = (q_3, \lambda)$
$\delta(q_3, b, a) = (q_2, a)$

} match occurrence of two '*b*' with occurrence of '*a*'.

$\delta(q_3, \lambda, z_0) = (q_f, z_0)$



12. Construct PDA for $L = \{a^n\, b^m\, c^{m+n} : m, n \geq 0 \;\&\; m \geq n\}$

Solution :

The transition function (δ) are:

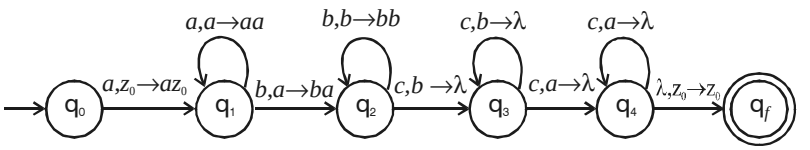
$\delta(q_0, a, z_0) = (q_1, az_0)$
$\delta(q_1, a, a) = (q_1, aa)$
$\delta(q_1, b, a) = (q_2, ba)$
$\delta(q_2, b, b) = (q_2, bb)$

} push *a* & *b* on to stack

$\delta(q_2, c, b) = (q_3, \lambda)$
$\delta(q_3, c, b) = (q_3, \lambda)$
$\delta(q_3, c, a) = (q_4, \lambda)$
$\delta(q_4, c, a) = (q_4, \lambda)$

} match *m*+*n* occurrence of '*c*' with *m*+*n* occurrence of *a*, *b*.

$\delta(q_4, \lambda, z_0) = (q_f, z_0)$



13. Construct a context free grammar G which accepts N(M)

where M = $(\{q_0,q_1\}, \{a,b\}, \{z_0,z\}, \delta, q_0, z_0, \phi)$

where δ is given by

$\delta(q_0,b,z_0) = \{(q_0, zz_0)\}$

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

$\delta(q_0,b,z) = \{(q_0,zz)\}$

$\delta(q_0,a,z) = \{(q_1,z)\}$

$\delta(q_1,b,z) = \{(q_1,\lambda)\}$

$\delta(q_1,a,z_0) = \{(q_0,z_0)\}$ **(April/May 2004)**

Solution :

Let G = $(V_N, \{a,b\}, P, S)$

Step 1 :

V_N consists of S, $[q_0, z_0, q_0]$, $[q_0, z_0, q_1]$, $[q_0, z, q_0]$, $[q_0, z, q_1]$, $[q_1, z_0, q_0]$, $[q_1, z_0, q_1]$, $[q_1, z, q_0]$, $[q_1, z, q_1]$ because

$V_N = \{S\} \cup \{[q, z, q^1] \mid q, q^1 \in Q, z \in \Gamma\}$

Here

$q_0, q_1 \in Q$

$z_0, z \in \Gamma$

Hence we got

$V_N = \{S, [q_0, z_0, q_0],$
$[q_0, z_0, q_1],$
$[q_0, z, q_0],$
$[q_0, z, q_1],$
$[q_1, z_0, q_0],$
$[q_1, z_0, q_1],$
$[q_1, z, q_0],$
$[q_1, z, q_1]\}$

Step 2 :

The productions in P are induced by moves of *pda* as follows:

R_1 : S - productions are given by

$S \rightarrow [q_0, z_0, q]$ for every q in Q

R_2 : Each move erasing a pushdown symbol given by $(q^1, \lambda) \in \delta(q, a, z)$

induces a production

$[q, z, q^1] \rightarrow a$

R_3 : Each move not erasing a pushdown symbol given by $(q_1, z_1, z_2 \ldots\ldots z_m) \in \delta(q, a, z)$ induces many productions of the form

$[q, z, q^1] \rightarrow a\ [q_1, z_1, q_2]\ [q_2, z_2, q_3] \ldots\ldots [q_m, z_m, q^1]$

Where each of the states $q_1, q_2 \ldots\ldots q_m$ can be any state in Q.

Applying step 2 we get

$P_1 : S \rightarrow [q_0, z_0, q_0]$

$P_2 : S \rightarrow [q_0, z_0, q_1]$

because $S \rightarrow [q_0, z_0, q]$ for every q in Q

Here $q = \{q_0, q_1\} \in Q$.

Similarly

$\delta(q_0, b, z_0) = \{(q_0, zz_0)\}$ yields

$P_3 : [q_0, z_0, q_0] \rightarrow b\ [q_0, z, q_0]\ [q_0, z_0, q_0]$

$P_4 : [q_0, z_0, q_0] \rightarrow b\ [q_0, z, q_1]\ [q_1, z_0, q_0]$

$P_5 : [q_0, z_0, q_1] \rightarrow b\ [q_0, z, q_0]\ [q_0, z_0, q_1]$

$P_6 : [q_0, z_0, q_1] \rightarrow b\ [q_0, z, q_1]\ [q_1, z_0, q_1]$

because R_3 says that

$[q, z, q^1] \rightarrow a\ [q_1, z_1, q_2]\ [q_2, z_2, q_3] \ldots\ldots [q_m, z_m, q^1]$

Here $q^1 = \{q_0, q_1\}$

$\delta(q_0, \lambda, z_0) = \{(q_0, \lambda)\}$ gives

$P_7 : [q_0, z_0, q_0] \rightarrow \lambda$ because by R_2

Similarly

$\delta(q_0, b, z) = \{(q_0, zz)\}$ applying R_3 gives

$P_8 : [q_0, z, q_0] \rightarrow b\ [q_0, z, q_0]\ [q_0, z, q_0]$

$P_9 : [q_0, z, q_0] \rightarrow b\ [q_0, z, q_1]\ [q_1, z, q_0]$

$P_{10} : [q_0, z, q_1] \rightarrow b\ [q_0, z, q_0]\ [q_0, z, q_1]$

P_{11}: $[q_0, z, q_1] \rightarrow b\ [q_0, z, q_1]\ [q_1, z, q_1]$

$\delta(q_0, a, z) = \{(q_1, z)\}$ yields using R_3

P_{12} : $[q_0, z, q_0] \rightarrow a\ [q_1, z, q_0]$

P_{13} : $[q_0, z, q_1] \rightarrow a\ [q_1, z, q_1]$

$\delta(q_1, b, z) = \{(q_1, \lambda)\}$ applying R_2 gives

P_{14} : $[q_1, z, q_1] \rightarrow b$

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

P_{15} : $[q_1, z_0, q_0] \rightarrow a\ [q_0, z_0, q_0]$

P_{16} : $[q_1, z_0, q_1] \rightarrow a\ [q_0, z_0, q_1]$

P_1 to P_{16} are the productions of P in the context free grammar G.

14. Let $M=(\{q_0,q_1\}, \{0,1\}, \{X,z_0\}, \delta, q_0, z_0, \phi)$ where δ is given by :

$\delta(q_0, 0, z_0) = \{(q_0, Xz_0)\}$

$\delta(q_0, 0, X) = \{(q_0, XX)\}$

$\delta(q_0, 1, X) = \{(q_1, \varepsilon)\}$

$\delta(q_1, 1, X) = \{(q_1, \varepsilon)\}$

$\delta(q_1, \varepsilon, X) = \{(q_1, \varepsilon)\}$

$\delta(q_1, \varepsilon, z_0) = \{(q_1, \varepsilon)\}$

Construct a CFG G = (V, T, P, S) generating N(M) **(Nov/Dec 2003)**

Solution :

Step 1 :

We define G = (V, T, P, S), where

$V = \{S\} \cup \{[q, z, q^1] \mid q, q^1 \in Q, z \in \Gamma\}$

(i.e.) V is a start symbol or an ordered triple, whose first and third elements are states and the second element is a pushdown symbol.

$\therefore$ V = {S, $[q_0, X, q_0]$, $[q_0, X, q_1]$,

$[q_0, z_0, q_0]$, $[q_0, z_0, q_1]$,

$[q_1, z_0, q_0]$, $[q_1, z_0, q_1]$,

$[q_1, X, q_0]$, $[q_1, X, q_1]$}

Step 2 :

The production in P are induced by moves of PDA as follows :

R_1 : S – productions are given by $S \rightarrow [q_0, z_0, q]$ for every q in Q.

The productions are :

$P_1 : S \rightarrow [q_0, z_0, q_0]$

$P_2 : S \rightarrow [q_0, z_0, q_1]$

Step 3:

R_2 : Each move, not erasing a pushdown symbol, given by:

$(q_1, z_1 z_2 ..... z_m) \in \delta (q, a, z)$ induces many productions of the form

$[q, z, q^1] \rightarrow a[q_1, z_1, q_2] [q_2, z_2, q_3] ..... [q_m, z_m, q^1]$

where each of the states $q_1, q_2 ..... q_m$ can be any state in Q

$\delta(q_0, 0, z_0) = \{(q_0, Xz_0)\}$ yields

$P_3 : [q_0, z_0, q_0] \rightarrow 0[q_0, X, q_0] [q_0, z_0, q_0]$

$P_4 : [q_0, z_0, q_0] \rightarrow 0 [q_0, X, q_1] [q_1, z_0, q_0]$

$P_5 : [q_0, z_0, q_1] \rightarrow 0 [q_0, X, q_0] [q_0, z_0, q_1]$

$P_6 : [q_0, z_0, q_1] \rightarrow 0 [q_0, X, q_1] [q_1, z_0, q_1]$

Similarly :

$\delta(q_0, 0, X) = \{(q_0, XX)\}$ yields

$P_7 : [q_0, X, q_0] = 0 [q_0, X, q_0] [q_0, X, q_0]$

$P_8 : [q_0, X, q_0] = 0 [q_0, X, q_1] [q_1, X, q_0]$

$P_9 : [q_0, X, q_1] = 0 [q_0, X, q_0] [q_0, X, q_1]$

$P_{10} : [q_0, X, q_1] = 0 [q_0, X, q_1] [q_1, X, q_1]$

Step 4 :

R3 : Each move erasing a pushdown symbol given by $(q^1, \lambda) \in \delta (q, a, z)$ induces the production $[q, z, q^1] \rightarrow q$ therefore

$\delta(q_0, 1, X) = \{(q_1, \varepsilon)\}$ gives

$P_{11} : [q_0, X, q_1] \rightarrow 1$

Similarly $\delta(q_1, 1, X) = \{(q_1, \varepsilon)\}$ gives

$P_{12} : [q_1, X, q_1] \rightarrow 1$

Similarly $\delta(q_1, \varepsilon, X) = \{(q_1, \varepsilon)\}$ gives

$P_{13} : [q_1, X, q_1] \rightarrow \varepsilon$

Similarly $\delta(q_1, \varepsilon, z_0) = \{(q_1, \varepsilon)\}$

$P_{14} : [q_1, z_0, q_1] \rightarrow \varepsilon$

P_1 to P_{14} are the productions for the grammar G

It should be noted that there are no productions for the variables $[q_1, X, q_0]$ and $[q_1, z_0, q_0]$. As all 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 terminal string can be derived from $[q_0, X, q_0]$ or $[q_0, z_0, q_0]$ either. Deleting all productions involving one of these four variables on either the right or left, we end up with the following productions.

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

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

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

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

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

$[q_1, X, q_1] \rightarrow \varepsilon$

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

15. Let $P = (\{p, q\}, \{0,1\}, \{X, Z_0\}, \delta, q, Z_0)$

where δ is given by

1. $\delta(q, 1, Z_0) = \{(q, XZ_0)\}$
2. $\delta(q, 1, X) = \{(q, XX)\}$
3. $\delta(q, 0, X) = \{(p, X)\}$
4. $\delta(q, \varepsilon, X) = \{(q, \varepsilon)\}$
5. $\delta(p, 1, X) = \{(p,\varepsilon)\}$
6. $\delta(p, 0, Z_0) = \{(q, Z_0)\}$

to a CFG.

Solution :

We get G = (V, {0, 1}, P, S), where

$$V = \{[p\ X\ p], [p\ X\ q], [pZ_0p], [pZ_0q], S\}$$

and the productions in P are

$S \rightarrow [qZ_0q] \mid [qZ_0p]$

$P_1 : S \rightarrow [qZ_0q]$

$P_2 : S \rightarrow [qZ_0p]$

From rule (1) : $\delta(q, 1, z_0) = (q, Xz_0)$

$P_3 : [qZ_0q] \rightarrow 1\ [qXq][qZ_0q]$

$P_4 : [qZ_0q] \rightarrow 1\ [qXp][pZ_0q]$

$P_5 : [qZ_0p] \rightarrow 1\ [qXq][qZ_0p]$

$P_6 : [qZ_0p] \rightarrow 1\ [qXp][pZ_0p]$

From rule (2) : $\delta(q, 1, X) = (q, XX)$

$P_7 : [qXq] \rightarrow 1\ [qXq][qXq]$

$P_8 : [qXq] \rightarrow 1\ [qXp][pXq]$

$P_9 : [qXp] \rightarrow 1\ [qXq][qXp]$

$P_{10}: [qXp] \rightarrow 1\ [qXp][pXp]$

From rule (3) : $\delta(q, 0, X) = (p, X)$

$P_{11} : [qXq] \rightarrow 0[pXq]$

$P_{12} : [qXp] \rightarrow 0[pXp]$

From rule (4) : $\delta(q, \varepsilon, X) = (q, \varepsilon)$

$P_{13} : [qXq] \rightarrow \in$

From rule (5) : $\delta(p, 1, X) = (p, \varepsilon)$

$P_{14} : [pXp] \rightarrow 1$

From rule (6) : $\delta(p, 0, z_0) = (q, z_0)$

$P_{15} : [pZ_0q] \rightarrow 0[qZ_0q]$

$P_{16} : [pZ_0p] \rightarrow 0[qZ_0p]$

P_1 to P_{16} are the productions for the grammar.