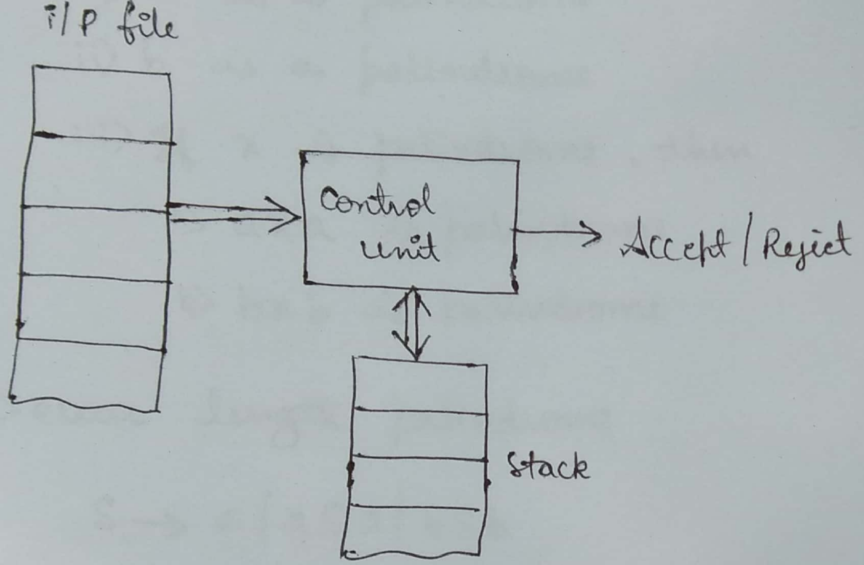
UNIT-4

PUSHDOWN AUTOMATA

Informal description of a PDA (Working of a PDA):

A schematic representation of a PDA as shown below.



- The PDA's Control unit reads input, one symbol at a time. The PDA is allowed to observe the symbol at the top of the stack and base its transition on,
  - i) The current state
  - ii) The symbol on top of stack
  - iii) Current input symbol
- Alternatively, the PDA is allowed to make a "spontaneous transition" using ϵ as its input instead of an input symbol.
- In one transition, the PDA,
  - i) Consumes from the input, the symbol that it uses on the transition.
  - ii) If ϵ is used for the input, then no input symbol is consumed.

ii) Go to a new state, which may or may not be the same as the previous state.

iii) Replaces the symbol on top of the stack by a string. The string could be ϵ or the symbol that appeared on the top of the stack previously or a different symbol or more than one symbol.

a) $\epsilon \Rightarrow$ pop the stack

b) same symbol $\Rightarrow$ no change to the contents of stack.

c) different symbol $\Rightarrow$ pop the symbol on top of stack and push the new symbol (1 pop and 1 push operation)

d) more than one symbol $\Rightarrow$ Pop the symbol on top of stack and push the new symbols. (1 pop & more than 1 push operations).

* The PDA accepts the language either by reaching a final state or emptying its stack.

* Note: Define acceptance by final state and empty stack after this.

Formal definition: A PDA is a 7 tuple $P = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$

where Q - A finite set of internal states.

Σ - A finite set of input alphabets.

Γ - A finite set of stack symbols

q_0 - the start state.

Z - the initial symbol on top of the stack

F - finite set of final states.

δ is a transition function defined as,

$\delta : Q \times \Gamma \times (\Sigma \cup \epsilon) \rightarrow Q \times \Gamma^*$ (For DPDA)

$\delta : Q \times \Gamma \times (\Sigma \cup \epsilon) \rightarrow$ Finite subsets of $Q \times \Gamma^*$ (For NPDA)

Instantaneous description (ID) of a PDA : It is a triple (q, α, w) which describes the execution of a PDA at each instance where q is the current state, α is the contents of the stack and w is the remaining strings to be processed.

If $(q, Z\alpha, aw)$ is the current configuration and if $\delta(q, Z, a) = (P, \beta)$ then the new configuration is,

$(P, \beta\alpha, w)$.

We say that $(P, \beta\alpha, w)$ is derived from $(q, Z\alpha, aw)$ in one step or in one move & is denoted by

$$(q, Z\alpha, aw) \vdash (P, \beta\alpha, w)$$

turn style

Moves: "Move" reflects the idea that, by consuming a symbol from the input and replacing the symbol on top of stack by string, we can go from the current state to a new state. It is denoted by the symbol "$\vdash$" (turn style)

Moves depend on transition rules,

$$\delta(q, Z, a) = \{(P_1, T_1), (P_2, T_2), \ldots (P_m, T_m)\}$$

Each move of a machine is determined by,

i) The current state

ii) The symbol on top of stack

iii) Current input symbol.

and the action consists of 2 parts,

i) Changing the state

ii) Replacing the contents of stack.

Language of a PDA:

i) **Acceptance by final state**: Let $P = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$ be a PDA. Then $L(P)$, the language accepted by a PDA by final state is defined as the set of all strings w for which some sequence of moves causes the PDA to enter the final state

ie $L(P) = \{ w \in \Sigma^* : (q_0, Z, w) \vdash^* (P, V, \epsilon),\ P \in F,\ V \in \Gamma^* \}$

ii) **Acceptance by empty stack**: Let $P = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$ be a PDA. Then $N(P)$, the language accepted by P by empty stack is defined as, the set of all strings w for which some sequence of moves causes the PDA to empty its stack.

ie $N(P) = \{ w \in \Sigma^* : (q_0, Z, w) \vdash^* (q_i, \epsilon, \epsilon) \}$ where $q_i \in Q \}$

PDA grammar: A grammar $G = (V, T, P, S)$ is called a PDA grammar if all productions are of the form,

$A \rightarrow aX$ or $A \rightarrow X$ where $A \in V$, $a \in T$ and $X \in V^*$.

Graphical Notation for PDA:

For all PDA's,

i) States are represented by circles.

ii) Arcs between states corresponds to transition of the PDA. It is of the form "current stack top / i/p symbol / New stack top". This means on reading the input symbol change the current stack top by the new stack top.

iii) An arrow labeled "start" indicates the start state.

iv) One or more states are indicated as final or accepting states. This is represented using concentric / double circles.

Note1: i) To Design PDA by empty stack method.

Step 1: Write the CFG for the given language.

2: Convert it to a PDA grammar.

3: Define the transition rules for state q.

Note 2: To design PDA by final state method.

1: Write the CFG for the given language.

2: Convert it to a PDA grammar.

3: Define the transition rules for state q.

4: Introduce a start state q_0 and final state q_f with,

i) transition from q_0 to q as $X/\epsilon/SX$, where S is the start symbol of the grammar.

ii) transitions from q to q_f as $X/\epsilon/\epsilon$.

Note 3: i) If $A \to aX$ is the production then the transition rule is given by,

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

ii) If $A \to a$ is a production, the transition rule is given by,

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

iii) If $A \to \epsilon$ is a production, the transition rule is given by,

$$\delta(q, A, \epsilon) = (q, \epsilon)$$

iv) If $A \to X$ is a production, the transition rule is given by,

$$\delta(q, A, \epsilon) = (q, X).$$

1) Design a PDA by empty stack to accept the language

$L = \{a^n b^n : n \geq 0\}$

Step 1: Write the CFGs for the given language.

$n=0$	$a^0 b^0$	ϵ
$n=1$	$a^1 b^1$	ab
$n=2$	$a^2 b^2$	$aabb$

$S \rightarrow \epsilon / aSb$

Step 2: Convert it to PDA grammar, we get,

$S \rightarrow \epsilon \mid aSB$

$B \rightarrow b$

Step 3: Define the transition rules for state q.

i) $\delta(q, S, \epsilon) = (q, \epsilon)$

If you are in state q, if top of stack is S, and if no input symbol is read, then remain in the same state q & replace the top of stack by ϵ.

ii) $\delta(q, S, a) = (q, SB)$

If you are in state q, if top of stack is S, and if input symbol read is a, then remain in the same state q and replace the top of the stack by SB.

iii) $\delta(q, B, b) = \delta(q, \epsilon)$

If you are in state q, if top of the stack is B, and input symbol read is b, then remain in the same state q and replace the top of the stack by ϵ

The PDA to accept the language $L = \{a^n b^n : n \geq 0\}$ is given by,

$P = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$ where,

$Q = \{q\}$

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

$\Gamma = \{S, B\}$

$q_0 = q$ is the start state

$Z = S$.

$F = $ __

δ is given by, $\delta(q, S, \epsilon) = (q, \epsilon)$

$\delta(q, S, a) = (q, SB)$

$\delta(q, B, b) = (q, \epsilon)$

Ex:

Trace for accepting $a^3 b^3$

$(q, S, a^3 b^3) \vdash (q, SB, a^2 b^3)$

$\vdash (q, SBB, ab^3)$

$\vdash (q, SBBB, \underset{\uparrow}{\epsilon} b^3)$

$\vdash (q, BBB, b^3)$

$\vdash (q, BB, b^2)$

$\vdash (q, B, b)$

$\vdash (q, \epsilon, \epsilon)$

Since the input of the process is empty & also the contents of the stack is empty the string $a^3 b^3$ is accepted by PDA.

Trace for a^3b^2

$(q, S, a^3b^2) \vdash (q, SB, a^2b^2)$

$\vdash (q, SBB, ab^2)$

$\vdash (q, SBBB, \epsilon b^2)$

$\vdash (q, BBB, b^2)$

$\vdash (q, BB, b)$

$\vdash (q, B, \epsilon)$

Since the input process is empty & contents of stack is not empty a^3b^2 is rejected by PDA.

2) Design the PDA by final state method to accept the language $L = \{a^n b^n : n \geq 0\}$.

Step 1: Write the CFGs for the given language

$n=0$	a^0b^0	ϵ	$S \to \epsilon \mid aSb$
$n=1$	a^1b^1	$a\epsilon b$	
$n=2$	a^2b^2	$aa\epsilon bb$	

Step 2: Converting it into PDA grammar.

$S \to \epsilon \mid aSB$

$B \to b$

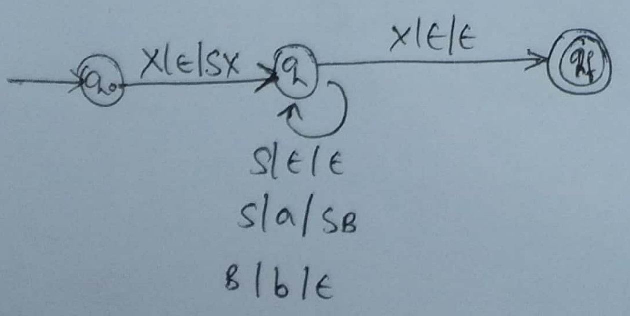
Step 3: Define the transition rules for state q.

i) $\delta(q, S, \epsilon) = (q, \epsilon)$

ii) $\delta(q, S, a) = (q, SB)$

iii) $\delta(q, B, b) = (q, \epsilon)$

* (write it in words).



Step 4: Introduce a start state q_0 & final state q_f with,

i) Transition from q_0 to q as, $X|\epsilon|SX$

ii) Transition from q to q_f as, $X|\epsilon|\epsilon$

← Write transition diagram

The two new transition rules included are,

1) $\delta(q_0, X, \epsilon) = (q, SX)$

If you are in state q_0 & top of the stack is X and if no input is read then go to state q and replace the top of the stack by SX.

2) $\delta(q, X, \epsilon) = (q_f, \epsilon)$

If you are in state q & top of the stack is X and if no input is read then go to state q_f and replace the top of the stack by ϵ.

3) PDA to accept the language by final state method

$P = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$

$Q = \{q_0, q, q_f\}$

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

$q_0 = q_0$

$Z = X$

$\Gamma = \{S, B, X\}$

$F = \{q_f\}$

δ is given by,

1) $\delta(q_0, X, \epsilon) = (q, SX)$

2) $\delta(q, S, \epsilon) = (q, \epsilon)$

3) $\delta(q, S, a) = (q, SB)$

4) $\delta(q, B, b) = (q, \epsilon)$

5) $\delta(q, X, \epsilon) = (q_f, \epsilon)$

Trace for accepting a^3b^3

$(q_0, X, \underset{\uparrow}{\epsilon} a^3b^3) \vdash (q, SX, a^3b^3)$

$\vdash (q, SBX, a^2b^3)$

$\vdash (q, SBBX, ab^3)$

$\vdash (q, SBBBX, \underset{\uparrow}{\epsilon} b^3)$

$\vdash (q, BBBX, b^3)$

$\vdash (q, BBX, b^2)$

$\vdash (q, BX, b)$

$\vdash (q, X, \epsilon)$

$\vdash$ ~~$(q_f, \epsilon, \epsilon)$~~ (q_f, ϵ)

Since the input to be processed is empty & also the contents of the stack is empty and reaches the final state q_f the input is accepted.

Trace for ~~accepting~~ a^3b^2

$(q_0, X, \underset{\uparrow}{\epsilon} a^3b^2) \vdash (q, SX, a^3b^2)$

$\vdash (q, SBX, a^2b^2)$

$\vdash (q, SBBX, ab^2)$

$\vdash (q, SBBBX, \underset{\uparrow}{\epsilon} b^2)$

$\vdash (q, BBBX, b^2)$

$\vdash (q, BBX, b)$

$\vdash (q, BX, \epsilon)$

3) Design a PDA to accept the language $L = \{a^n b^n : n > 0\}$

a) by empty stack method.

Step 1: CFG is, $S \rightarrow aSb \mid ab$

Step 2: PDA grammar

$S \rightarrow aSB \mid aB$

$B \rightarrow b$

Step 3: transition rules:

i) $\delta(q, S, a) = (q, SB)$

ii) $\delta(q, S, a) = (q, B)$

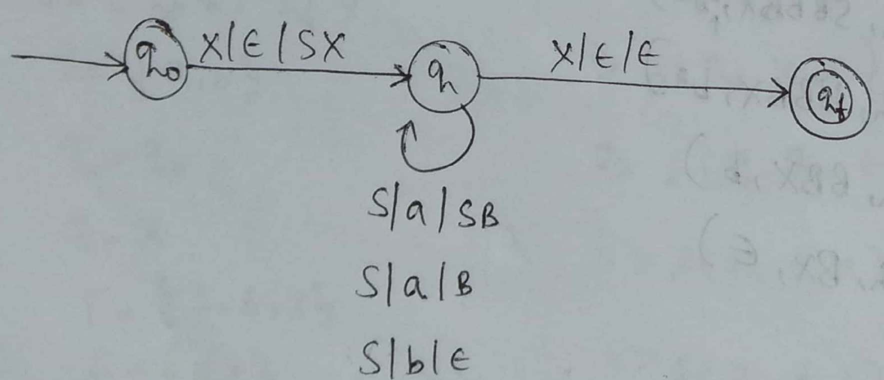
iii) $(q, B, b) = (q, \epsilon)$

b) final state method.

Step 1: -

Step 2: - - same

Step 3: - - -



new rules:

1) $\delta(q_0, X, \epsilon) = (q, SX)$

2) $\delta(q, S, a) = (q, SB)$

3) $\delta(q, S, a) = (q, B)$

4) $\delta(q, B, b) = (q, \epsilon)$

5) $\delta(q, X, \epsilon) = (q_f, \epsilon)$

4) Design a PDA to accept palindromes with even number of characters given $\Sigma = \{0,1\}$.

Note: PDA for $L = \{ww^R : w \in \{0,1\}^*\}$

a) PDA by empty stack method.

Step 1: CFG is, $S \rightarrow 0S0 \mid 1S1 \mid \epsilon$

Step 2: PDA grammar is,

$S \longrightarrow 0SA \mid 1SB \mid \epsilon$

$A \rightarrow 0$

$B \rightarrow 1$

Step 3: Transition rules,

i) $\delta(q, S, 0) = (q, SA)$

ii) $\delta(q, S, 1) = (q, SB)$

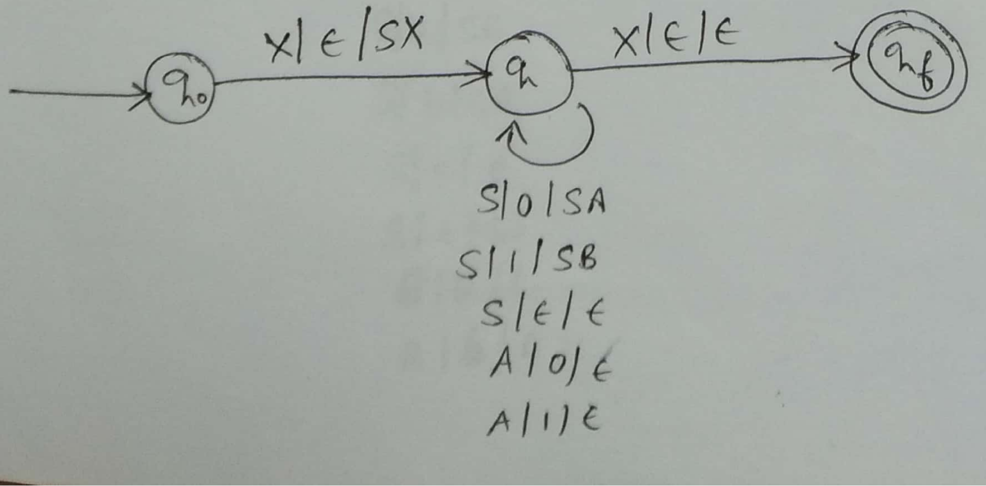
iii) $\delta(q, S, \epsilon) = (q, \epsilon)$

iv) $\delta(q, A, 0) = (q, \epsilon)$

v) $\delta(q, B, 1) = (q, \epsilon)$

b) final state method.

Step 1:
Step 2: Same
Step 3:



i) $\delta(q, X, \epsilon) = (q, SX)$

ii)

iii)

4)

5)

6)

7) $\delta(q, X, \epsilon) = (q, \epsilon)$

5) Design a PDA to accept the language $L = \{wcw^R$ where $w \in \{a,b\}^*\}$ $\Sigma = \{a, b\}$.

a) empty stack

Step 1: $S \rightarrow c \mid aSa \mid bSb$

Step 2: $S \rightarrow c \mid aSA \mid bSB$

$A \rightarrow a$

$B \rightarrow b$

Step 3: i) $\delta(q, S, c) = (q, \epsilon)$

ii) $\delta(q, S, a) = (q, SA)$

iii) $\delta(q, S, b) = (q, SB)$

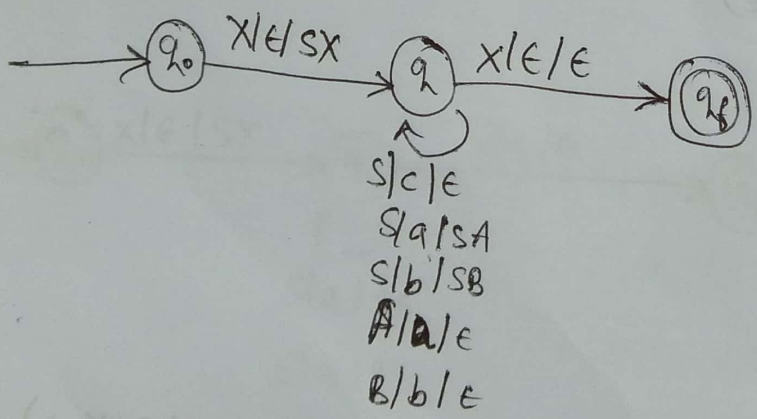
iv) $\delta(q, A, a) = (q, \epsilon)$

v) $\delta(q, B, b) = (q, \epsilon)$

b) final state method

Step 1:
Step 2:
Step 3: } same

Step 4:



6) $L = \{ n_a(w) = n_b(w) \}$

a) empty stack

Step 1: CFG is, $S \to asb \mid bsa \mid \epsilon \mid SS$

Step 2: PDA is,

$S \to aSB \mid bSA \mid \epsilon \mid SS$

$B \to b$

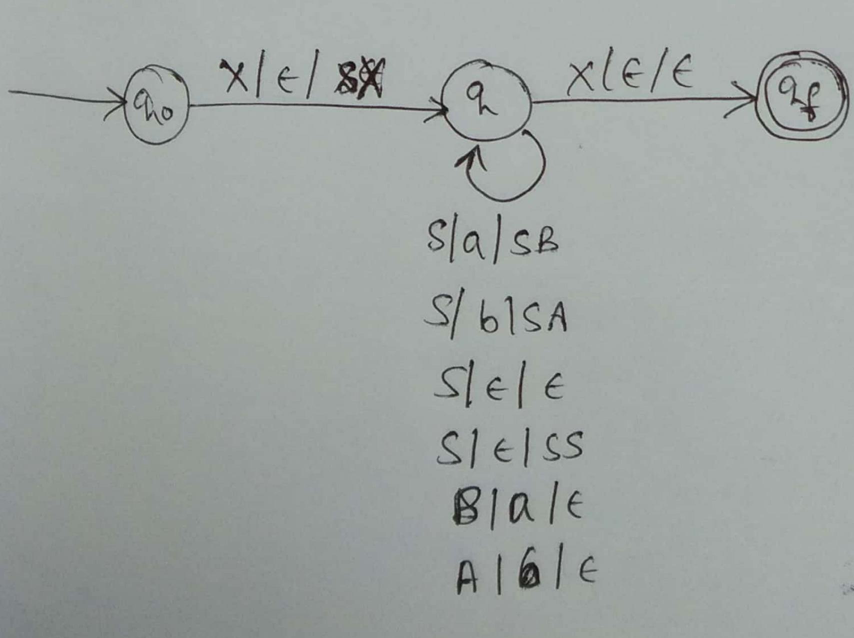
$A \to a$

Step 3: Transition rules.

1) $\delta(q, S, a) = (q, SB)$
2) $\delta(q, S, b) = (q, SA)$
3) $\delta(q, S, \epsilon) = (q, \epsilon)$
4) $\delta(q, S, \epsilon) = (q, SS)$
5) $\delta(q, B, a) = (q, \epsilon)$
6) $\delta(q, A, b) = (q, \epsilon)$

b) final state.

Step 1:
Step 2: - - Same
Step 3:



new transition rules:

7) $\delta(q, X, \epsilon) = (q, SX)$

8) $\delta(q, X, \epsilon) = (q, \epsilon)$.

7) $L = \{a^n b^{2n} : n > 0\}$

a) empty stack

Step1: $S \rightarrow aSbb \mid abb$

Step2: $S \rightarrow aSBB \mid aBB$

$B \rightarrow b$

Step 3: $\delta(q, S, a) = (q, SBB)$

$\delta(q, S, a) = (q, BB)$

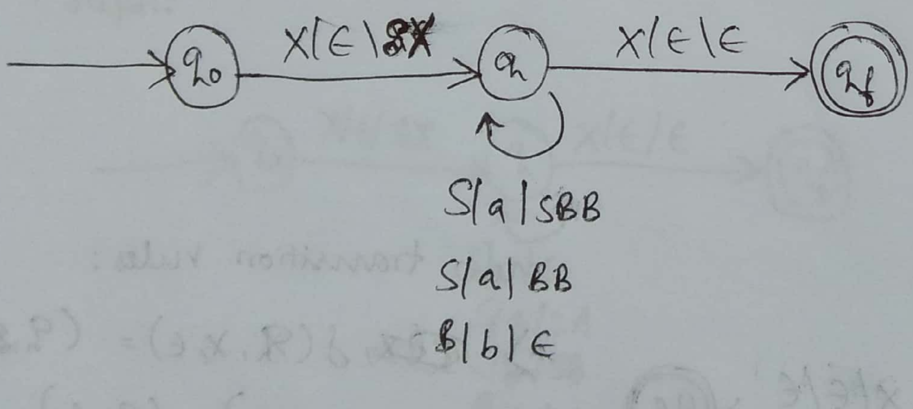
$\delta(q, B, b) = (q, \epsilon)$

b) final State

Step1:

Step 2:

Step 3:



i) $\delta(q, X, \epsilon) = (q, SX)$

ii) $\delta(q, X, \epsilon) = (q, \epsilon)$

8) Design a ~~DFA~~ PDA to accept the language corresponding to the grammar $S \rightarrow AA \mid 0$ (∵ already in PDA format)

$A \rightarrow SS \mid 1$

i) all variables or

ii) terminals followed by variables

a) empty stack

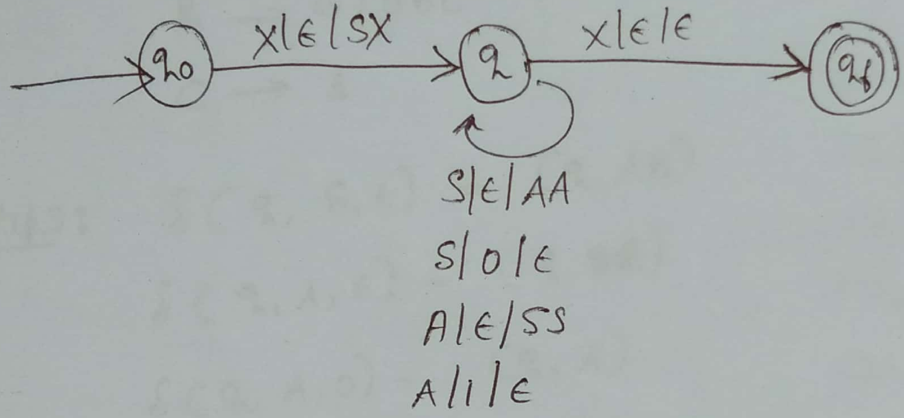
Step3: $\delta(q, S, \epsilon) \rightarrow (q, AA)$

$\delta(q, S, 0) \rightarrow (q, \epsilon)$

$\delta(q, A, \epsilon) \rightarrow (q, SS)$

$\delta(q, A, 1) \rightarrow (q, \epsilon)$

b) final state



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

10) Design a PDA to accept the language $L = \{0^n 1^m : n \geq m, m,n \geq 0\}$

a) empty stack

$L = \{\epsilon, 0, 00, 000, 01, 0011, 000111, 001, 0001$

step1: CFG is,

$S \rightarrow AB$

$A \rightarrow \epsilon / 0A$

$B \rightarrow \epsilon / 0B1$

step2: $S \rightarrow AB$

$A \rightarrow \epsilon / 0A$

$B \rightarrow \epsilon / 0BC$

$C \rightarrow 1$

step3: $\delta(q, S, \epsilon) = (q, AB)$

$\delta(q, A, \epsilon) = (q, \epsilon)$

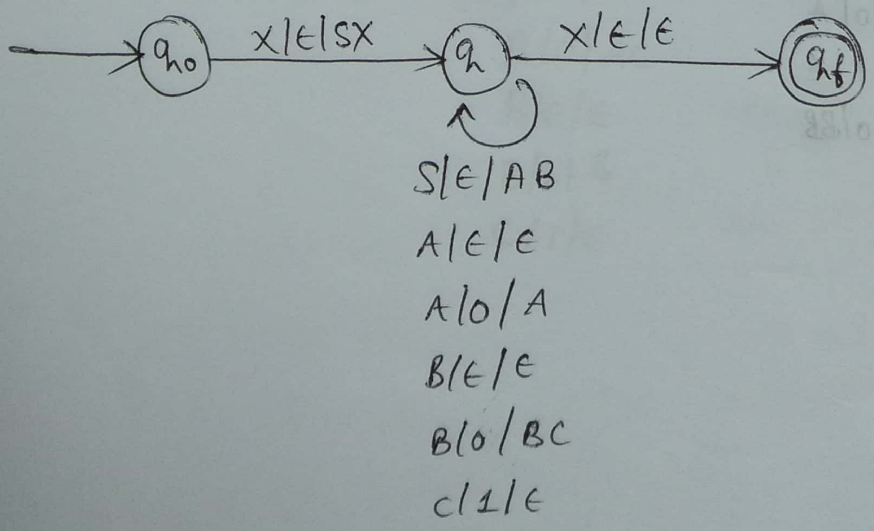
$\delta(q, A, 0) = (q, A)$

$\delta(q, B, \epsilon) = (q, \epsilon)$

$\delta(q, B, 0) = (q, BC)$

$\delta(q, C, 1) = (q, \epsilon)$

b) final state:



11) $L = \{0^n 1^m \quad n \geq m, \; m, n \geq 1\}$ $\quad L = \{01, \dots$

a) empty stack

<u>Step 1:</u> CFG is,

$S \rightarrow AB$

$A \rightarrow 0 / 0A$

$B \rightarrow 1 / 0B1$

<u>Step 2:</u> $S \rightarrow AB$

$A \rightarrow 0/0A$

$B \rightarrow 1/0BB$

<u>Step 3:</u>

i) $\delta(q, S, \epsilon) = (q, AB)$

ii) $\delta(q, A, 0) = (q, \epsilon)$

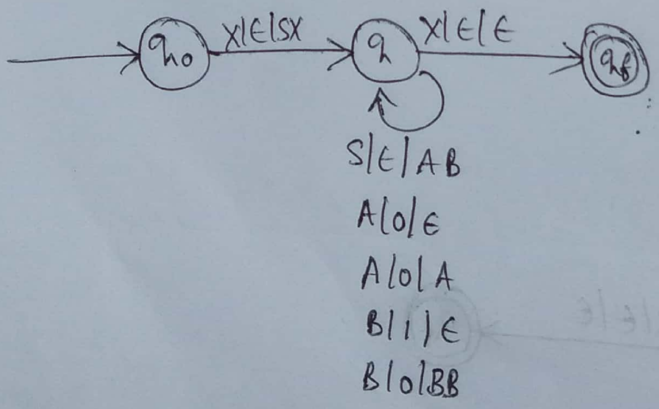
iii) $\delta(q, A, 0) = (q, A)$

iv) $\delta(q, B, 1) = (q, \epsilon)$

v) $\delta(q, B, 0) = (q, BB)$

~~vi) $\delta(q, C, 1) = (q, \epsilon)$~~

b) <u>final state:</u>



1Q) $L = \{0^n 1^m \quad n \le m, \; m, n \ge 0\}$.

a) empty stack: $L = \{\epsilon, 01, 011$

step 1: CFG is,

$$S \to AB$$
$$A \to \epsilon \mid 0A1$$
$$B \to \epsilon \mid 1B$$

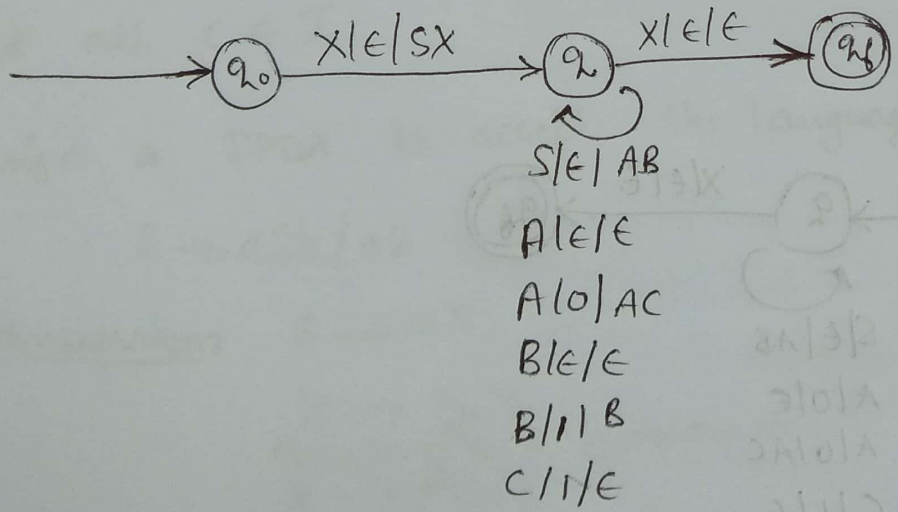
step 2: $S \to AB$

$$A \to \epsilon \mid 0AC$$
$$B \to \epsilon \mid 1B$$
$$C \to 1$$

step 3: $\delta(q, S, \epsilon) = (q, AB)$

$$\delta(q, A, \epsilon) = (q, \epsilon)$$
$$\delta(q, A, 0) = (q, AC)$$
$$\delta(q, B, \epsilon) = (q, \epsilon)$$
$$\delta(q, B, 1) = (q, B)$$
$$\delta(q, C, 1) = (q, \epsilon)$$

b) final state:



13) $L = \{0^n 1^m : n \le m,\ m, n \ge 1\}$

a) empty stack

Step1: CFG is,

$S \to AB$

$A \to 0 / 0A1$

$B \to 1 / 0B1$

Step2: $S \to AB$

$A \to 0 / 0AC$

$B \to 1 / 0BB$

$C \to 1$

Step3: $\delta(q, S, \epsilon) = (q, AB)$

$\delta(q, A, 0) = (q, \epsilon)$

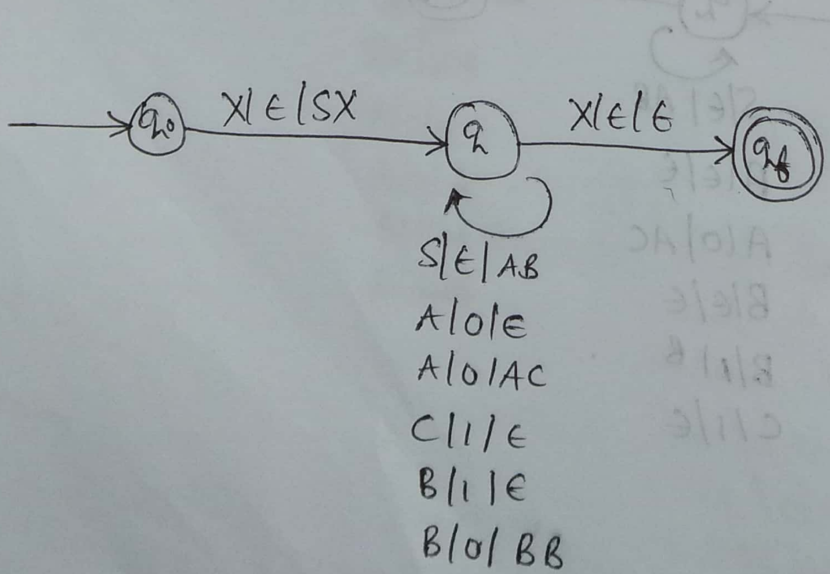
$\delta(q, A, 0) = (q, AC)$

$\delta(q, C, 1) = (q, \epsilon)$

$\delta(q, B, 1) = (q, \epsilon)$

$\delta(q, B, 0) = (q, BB)$

b) final state:



Deterministic PDA (DPDA): A DPDA is a PDA that never has a choice in its moves. is defined as a 7 tuples

$P = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$ where Q - finite set of internal states.

Σ = finite set of input alphabets

Γ = finite set of stack symbols

q_0 = is the start state

Z = initial symbol on top of stack

F = finite set of final states

δ = is a transition function defined as,

$Q \times (\Sigma \cup \epsilon) \times \Gamma \rightarrow Q \times \Gamma^*$

Note: For a DPDA,

i) For any given input symbol on stack top atmost one move can be made. ie $\delta(q, B, a)$ contains atmost one element. Thus $\delta(q, B, a)$ can be undefined also.

ii) When ϵ move is possible for some configuration, no input consuming alternative is available. ie if $\delta(q, B, \epsilon)$ is defined then $\delta(q, B; c)$ must be undefined for all $c \in \Sigma$.

1) Design a DPDA to accept the Language $L = \{a^n b^n : n > 0\}$

$S \rightarrow aSb / ab$

Conversion: $S \rightarrow aX$

$X \rightarrow Sb / b$

q, x, ϵ

q, x, b (q, ϵ)

Rewriting the x production as,

$X \rightarrow aXb / b$

DPDA grammar is:

$S \rightarrow aX$

$X \rightarrow aXB / b$

$B \rightarrow b$

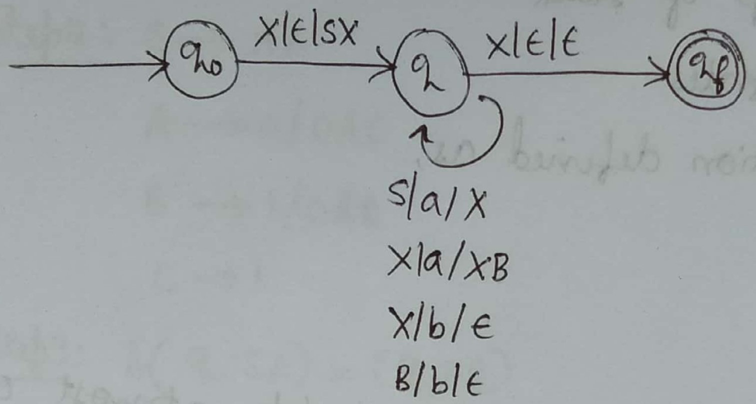
Transition rules: i) $\delta(q, S, a) = (q, X)$

ii) $\delta(q, X, a) = (q, XB)$

iii) $\delta(q, X, b) = (q, \epsilon)$

iv) $\delta(q, B, b) = (q, \epsilon)$

b) Final state method:



DPDA to accept the language by final state method

$P = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$

$Q = \{q_0, q, q_f\}$

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

$\Gamma = \{S, B, X\}$

$q_0 = q_0$

$Z = X$

$F = \{q_f\}$

a) Design a DPDA ~~to~~ accept the language $L = \{wcw^R : w \in \{a,b\}^*\}$

$S \to c \mid aSa \mid bSb$ (For all different symbols we have only one move) So no conversion.

DPDA grammar is:

$S \to c \mid aSA \mid bSB$

$A \to a$

$B \to b$

Transition rules:

a) $\delta(q, S, c) = (q, \epsilon)$

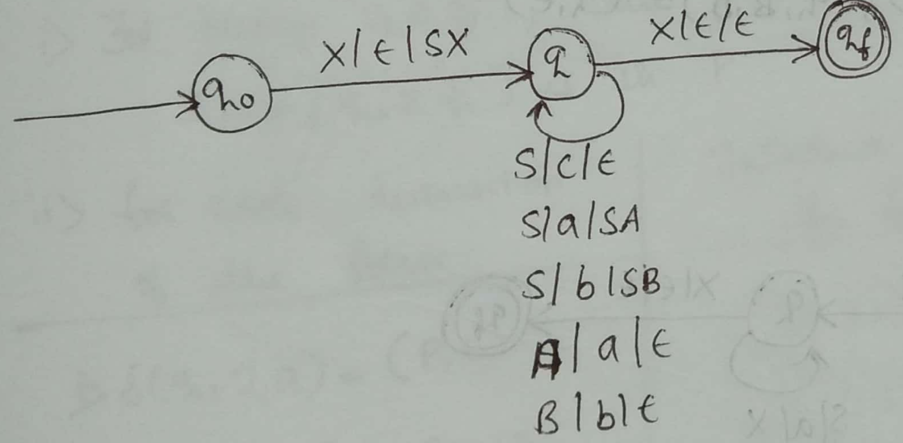
b) $\delta(q, S, a) = (q, SA)$

c) $\delta(q, S, b) = (q, SB)$

d) $\delta(q, A, a) = (q, \epsilon)$

e) $\delta(q, B, b) = (q, \epsilon)$

b) Final State



3> Design a DPDA to accept the language $L = \{a^n b^{2n} : n > 0\}$

$S \to aSbb \mid abb$ $\quad (q, S, a)$ $\quad (q, S, a)$ $\quad$ Conversion is required.

Conversion:

$S \to aX$ $\qquad q, X, \epsilon$

$X \to Sbb \mid bb$ $\qquad q, X, \underline{b}$

Rewriting X production as,

$X \to aXbb \mid bb$

DPDA grammar: $S \to aX$

$X \to aXBB \mid bB$

$B \to b$

Transition rules:

i> $\delta(q, S, a) = (q, X)$

ii> $\delta(q, X, a) = (q, XBB)$

iii> $\delta(q, X, b) = (q, B)$

iv> $\delta(q, B, b) = (q, \epsilon)$

b> Final state: