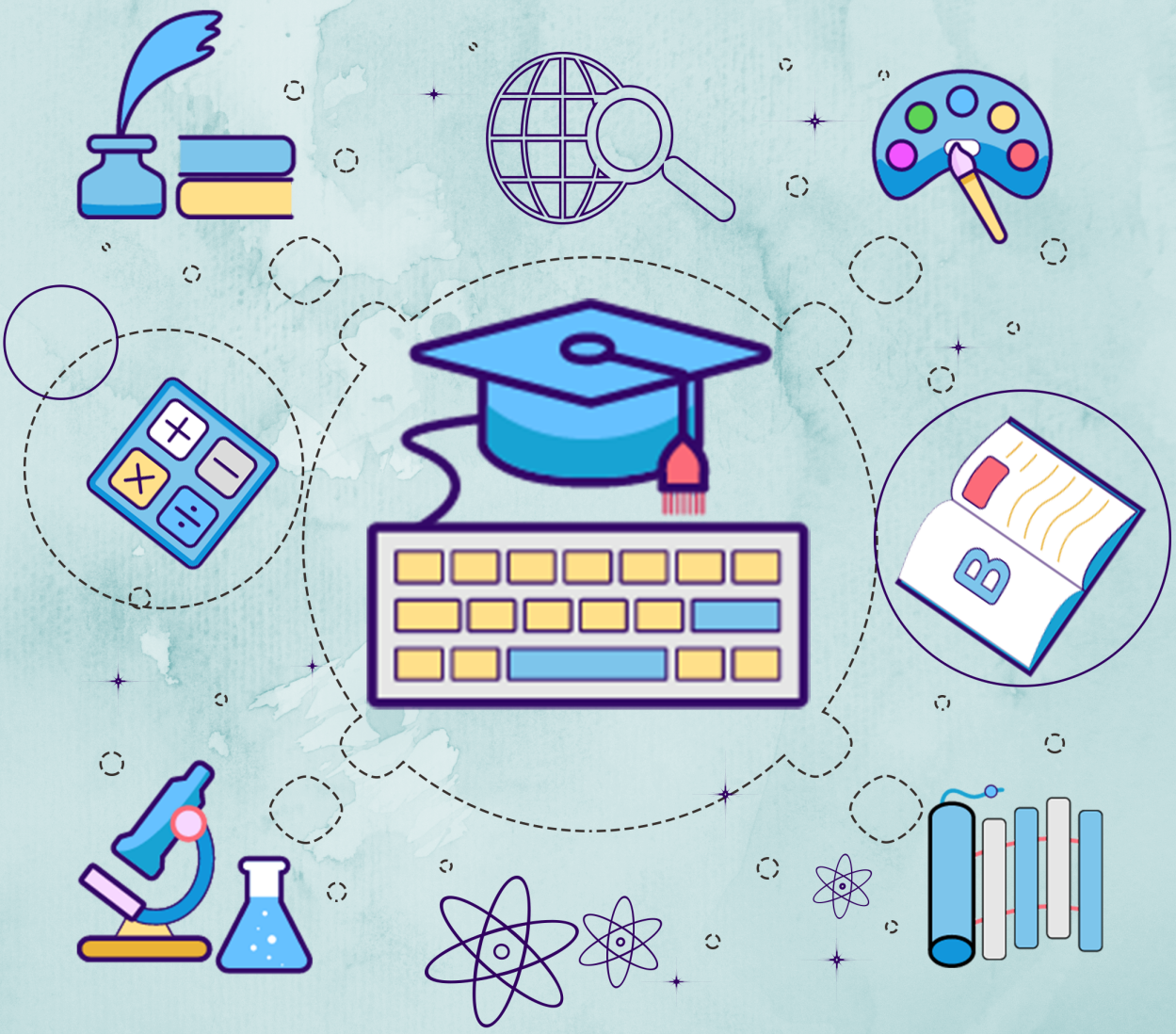
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

KeralaNotes



SYLLABUS | STUDY MATERIALS | TEXTBOOK

PDF | SOLVED QUESTION PAPERS

www.keralanotes.com

KeralaNotes

KTU STUDY MATERIALS

FORMAL LANGUAGES AND AUTOMATA THEORY

CST 301

Module 4

Related Link :

- KTU S5 STUDY MATERIALS
- KTU S5 NOTES
- KTU S5 SYLLABUS
- KTU S5 TEXTBOOK PDF
- KTU S5 PREVIOUS YEAR SOLVED QUESTION PAPER

FORMAL LANGUAGES AND AUTOMATA THEORY

Module 4

Nondeterministic Pushdown Automata (PDA), Deterministic Pushdown Automata (DPDA), Equivalence of PDAs and CFGs (Proof not required), Pumping Lemma for Context-Free Languages (Proof not required), Closure Properties of Context-Free Languages.

Prepared by:

Karishma PK

Assistant professor

Computer Science Department EKCTC

MODULE - IV

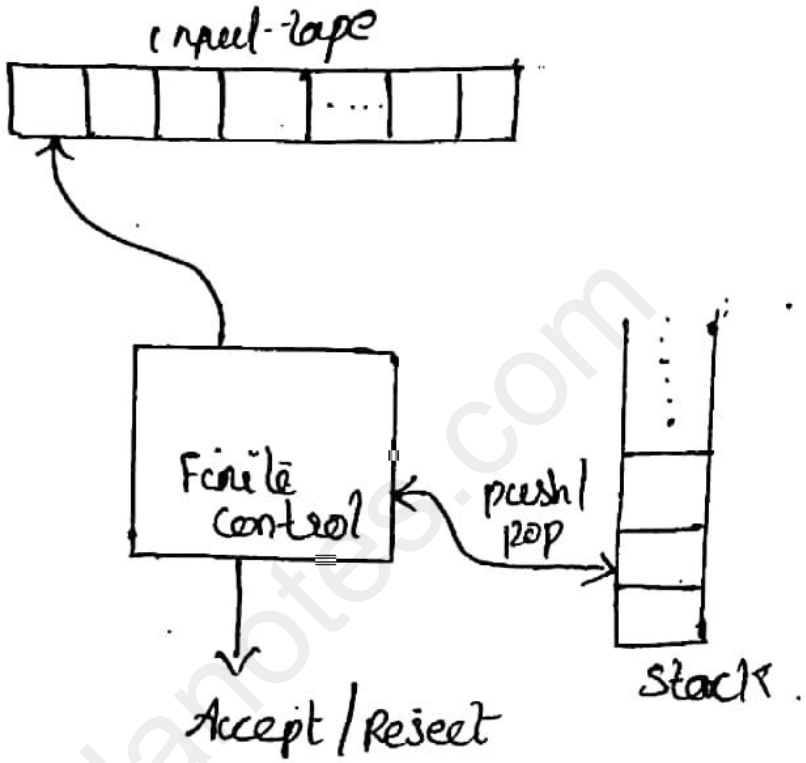
Pushdown Automata (PDA)

- Pushdown Automata is a way to implement a CFG in the same way we design Finite Automata for a Regular Grammar.
- A FA can remember a finite amount of information, but a PDA can remember an infinite amount of information.

  PDA = "Finite State Machine + stack memory"

- The addition of stack is used to provide a last-in first-out memory management capability to PDA
- PDA can store an unbounded amount of information on the stack.
- PDA is more powerful than FSM (Finite state Machine)
- A PDA can push an element on to the top of the stack and pop off an element from the top of the stack.
- To read an element into the stack, the top elements must be popped off and are lost.
- A PDA is more powerful than FA.

- Any language which can be acceptable by FA can also be acceptable by PDA.
- PDA also accepts a class of language which even can't be accepted by FA. Thus PDA is much more superior to FA



PDA components →

Input tape

The input tape is divided in many cells or symbols. The input head is read-only and may only move from left to right one symbol at a time.

Finite control

The finite control has some pointer which points the current symbol which is to be read.

Stack

- The stack is a structure in which we can push and remove the items from one end only.
- It has an infinite size.
- In PDA, the stack is used to store the items temporarily.

Formal definition of DPDA

The PDA can be defined as a collection of 7 components:

Q : Finite set of states

Σ : Finite set of input alphabets.

Γ : A stack symbol which can be pushed and popped from the stack

or

Finite set of stack alphabets

q_0 : Initial state

z_0 : a start symbol which is in Γ
(ie, $z_0 \in \Gamma$)

F : a set of final states

δ : Transition function.

$$Q \times \{\Sigma \cup \varepsilon\} \times \Gamma \longrightarrow Q \times \Gamma^*$$

- Transition function δ takes an argument a tuple.

  $\delta(q, a, X)$, where

  q is in state Q

  a is either an input symbol in Σ or 'a' can be ε.

  X is stack symbol.

  The output of δ is finite set of pairs (p, γ)

  p – new state

  γ – string of stack symbols that replaces X at the top of the stack.

  If $\gamma = \varepsilon$ then the stack is popped

  If $\gamma = X$, then the stack is unchanged

  If $\gamma = YZ$, then X is replaced by Z & Y is pushed on to the stack

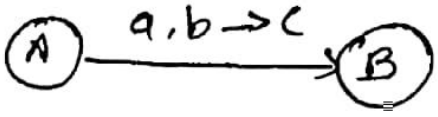
  If $\gamma = Z$, X is replaced by Z from stack top.

Graphical notation of PDA

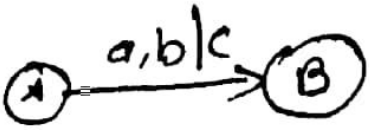
a :→ input symbol (This can also be ε)

b :→ Symbol on top of stack. This symbol is popped (ε means the stack is neither read nor popped)

c :→ This symbol is pushed on to the stack (ε means nothing is pushed)

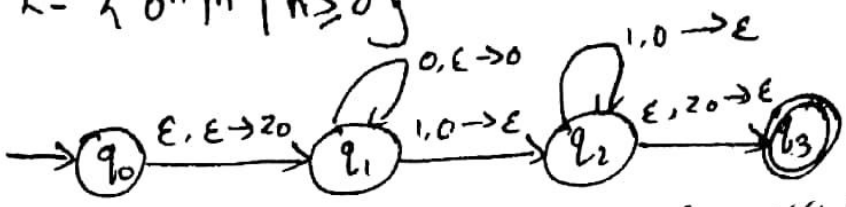


or

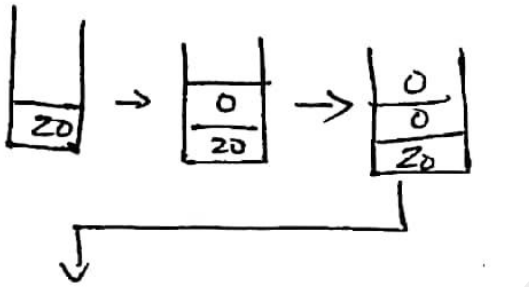


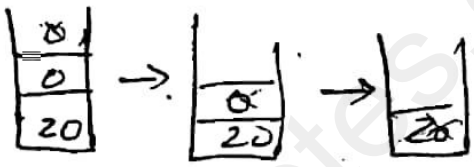
3) Construct a PDA that accepts

$L = \{ 0^n 1^n \mid n \geq 0 \}$



Suppose 001111 input string is given





4) Design a PDA for the language $L = a^n b^n ; n \geq 1$

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

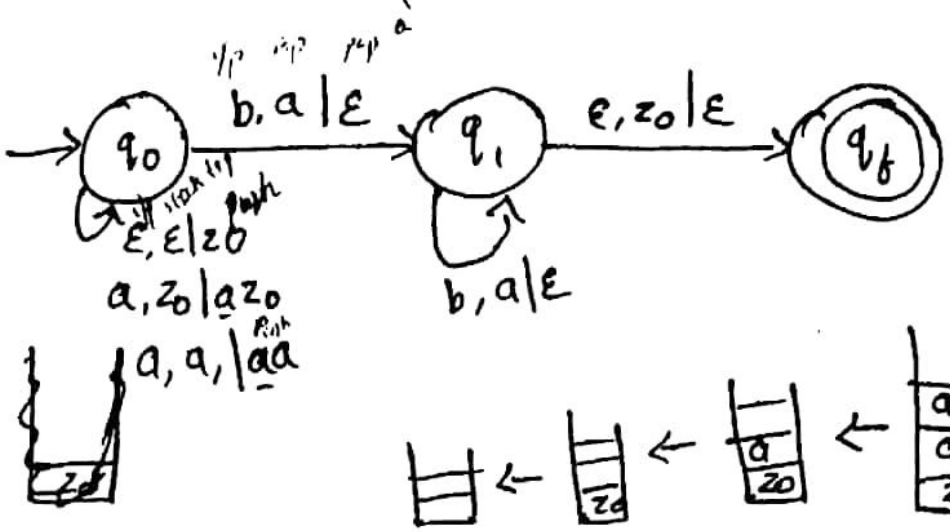
$\Sigma = \{ a, b \}$

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

$F = q_f$

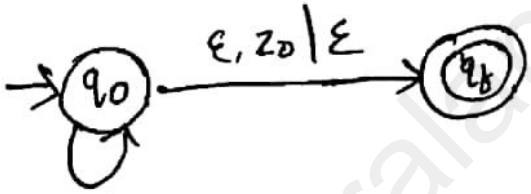
$L = \{ ab, aabb, aaabbb \ldots \}$

Let take string aabb.



2) Design a PDA for the language $L = \{w \in (ab)^* |$ w has equal no: of a's & b's$\}$.

Soln:- Let take the string bbababaa



ε, ε | z0
a, z0 | az0
a, a | aa
b, z0 | bz0
b, b | bb
b, a | ε
a, b | ε

NOTE

Acceptance of PDA is by empty stack or by final state

Equivalence of PDA's & CFG's

Converting CFG to PDA

Assumptions.

- Generating npPDA
  - Accept input by empty stack (no final state)
  - Bottom of stack contains Z_0.

CFG = (V, T, P, S) to PDA = $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

$Q = \{q\}$, $\Sigma = \{T\}$, $\Gamma = \{V\} \cup \{T\} \cup Z_0$,

$F = \phi$.

δ (Transition fn)

Step 1 : Starting symbol

start symbol (push)

$$\delta(q, \epsilon, Z_0) = (q, S, Z_0)$$

Step 2 : For all the variable x, if there is a production $x \rightarrow y \mid z$ then

$$\delta(q, \epsilon, x) = \{(q, y), (q, z)\}$$

Step 3 : For all terminals a

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

i/p stack top — pop it

step 4 3. Final Transition

$$\delta(q, \epsilon, z_0) = (q, \epsilon)$$

pop

5) $E \to E+E \mid E*E \mid a$ to PDA?

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

$Q = \{q\}$.

$\Sigma = \{a, +, *\}$

$\Gamma = \{a, +, *, \epsilon, z_0\}$

$F = \phi$

Transition δ:

1) Read-symbol

$$\delta(q, \epsilon, z_0) = (q, \epsilon, z_0)$$

2) non terminals

$$\delta(q, \epsilon, E) = \{(q, E+E), (q, E*E), (q, a)\}$$

3) Terminals

$$\delta(q, a, a) = (q, \epsilon)$$
$$\delta(q, +, +) = (q, \epsilon)$$
$$\delta(q, *, *) = (q, \epsilon)$$

4) Final Transition

$\delta(q, \epsilon, z_0) = (q, \epsilon)$

suppose take a*a

$\delta(q, a*a, z_0) \Rightarrow (q, a*a, \epsilon, z_0)$
$\rightarrow (q, a*a, \epsilon*\epsilon, z_0)$
$\rightarrow (q, a*a, a*\epsilon, z_0)$
$\rightarrow (q, a*a, *\epsilon, z_0)$
$\rightarrow (q, a, \epsilon, z_0)$
$\rightarrow (q, a, a, z_0)$
$\rightarrow (q, \epsilon, z_0)$
$\rightarrow (q, \epsilon)$

Q) $S \rightarrow aAA$,
$A \rightarrow aS|bS|a$ convert into CFA ?

Soln:- PDA = $(Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

$Q = \{q\}$, $\Sigma = \{a, b\}$, $\Gamma = \{a, b, S, A, z_0\}$, F=

Transition fn

1) start symbol

$\delta(q, \epsilon, z_0) = (q, S, z_0)$

2) Non terminals

$\delta(q, \epsilon, S) = \{(q, aXX)\}$

$\delta(q, \epsilon, X) = \{(q, aS), (q, bS), (q, a)\}$

3) Terminals

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

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

4) Final transition

$\delta(q, \epsilon, Z_0) = (q, \epsilon)$

Q) construct a PDA for the following CFG.

$S \to XX \mid a$, $X \to SX \mid b$?

soln:- 1) Start symbol

$\delta(q, \epsilon, Z_0) = (q, S, Z_0)$

2) non terminals

$\delta(q, \epsilon, S) = \{(q, XX), (q, a)\}$

$\delta(q, \epsilon, X) = \{(q, SX), (q, b)\}$

3) Terminals

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

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

4) Final transition

$\delta(q, \epsilon, Z_0) = (q, \epsilon)$

Converting PDA to CFG

PDA = $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ to CFG (V, T, P, S)

- $T = \{\epsilon\}$, $S - S$
- Variables are represented as 3 element combination $[QTQ]$

  Eg:- if we consider state as q_0, q_1 & stack element as a.

  $V = \{S, [q_0 a q_0], [q_0 a q_1], [q_1 a q_0], [q_1 a q_1]\}$

- $\delta \rightarrow P$

  Two types of transition δ:

  1) which doesn't push anything on to the stack.

  $\delta(q, a, Z_0) = (q, \epsilon)$ (state, i/p, top)

  2) which push elements on to stack

  $\delta(q, a, Z_0) = (q, aZ_0)$ (push)

P : is a combination of [Q Γ Q]

state i/p top

- $\delta[q_0, a, z_0] = (q_0, \epsilon)$ (pop cond^n) δ

$[q_0 z_0 q_0] \rightarrow a$ Pdn rule

- when elements are pushed into stack

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

$[q_0 z_0 \_] \rightarrow a\,[q_0 a \_]\,[\_ z_0 \_]$ (same, same)

Try all the combinations of states.

eg:- $PDA = (\{q_0, q_1\}, \{a, b\}, \{a, z_0\}, \delta, q_0, z_0, \phi)$ (Q, Σ, Γ, F)

where δ is given by

1) $\delta(q_0, a, z_0) = (q_0, a z_0)$
2) $\delta(q_0, a, a) = (q_0, aa)$
3) $\delta(q_0, b, a) = (q_1, \epsilon)$
4) $\delta(q_1, b, a) = (q_1, \epsilon)$
5) $\delta(q_1, \epsilon, z_0) = (q_1, \epsilon)$

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

where $T = \{a, b\}$

$S = S$

$\delta \rightarrow p$

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

$[q_0 z_0 \_] \rightarrow a [q_0 a \_][\_ z_0 \_]$

$[q_0 z_0 q_0] \rightarrow a [q_0 a q_0][q_0 z_0 q_0]$

$[q_0 z_0 q_0] \rightarrow a [q_0 a q_1][q_1 z_0 q_0]$

$[q_0 z_0 q_1] \rightarrow a [q_0 a q_0][q_0 z_0 q_1]$

$[q_0 z_0 q_1] \rightarrow a [q_0 a q_1][q_1 z_0 q_1]$

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

$[q_0 a \_] \rightarrow a [q_0 a][\_ a \_]$

$[q_0 a q_0] \rightarrow a [q_0 a q_0][q_0 a q_0]$

$[q_0 a q_0] \rightarrow a [q_0 a q_1][q_1 a q_0]$

$[q_0 a q_1] \rightarrow a [q_0 a q_0][q_0 a q_1]$

$[q_0 a q_1] \rightarrow a [q_0 a q_1][q_1 a q_1]$

3) $\delta(q_0, b, a) = (q_1, \varepsilon)$

$[q_0 a q_1] \rightarrow b$

4) $\delta(q_1, b, a) = (q_1, \varepsilon)$

$[q_1 a q_1] \rightarrow b$

5) $\delta(q_1, \varepsilon, z_0) = (q_1, \varepsilon)$

$[q_1 z_0 q_1] \rightarrow \varepsilon$

- Find start symbol

$S \rightarrow$ [start state, Bottom of stack, combinations of states]

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

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

→ Problem is presence of unwanted productions. So remove useless productions.

$[q_0\ z_0\ q_0] \rightarrow a\ [q_0\ a\ q_0]\ [q_0\ z_0\ q_0]$

$[q_0\ z_0\ q_0] \rightarrow a\ [q_0\ a\ q_1]\ [q_1\ z_0\ q_0]$

B $[q_0\ z_0\ q_1] \rightarrow a\ [q_0\ a\ q_0]\ [q_0\ z_0\ q_1]$

$[q_0\ z_0\ q_1] \rightarrow a\ [q_0\ a\ q_1]\ [q_1\ z_0\ q_1]$

C $[q_0\ a\ q_0] \rightarrow a\ [q_0\ a\ q_0]\ [q_0\ a\ q_0]$

$[q_0\ a\ q_0] \rightarrow a\ [q_0\ a\ q_1]\ [q_1\ a\ q_0]$

D $[q_0\ a\ q_1] \rightarrow a\ [q_0\ a\ q_0]\ [q_0\ a\ q_1]$

$[q_0\ a\ q_1] \rightarrow a\ [q_0\ a\ q_1]\ [q_1\ a\ q_1]$

$[q_0\ a\ q_1] \rightarrow b$

E $[q_1\ a\ q_1] \rightarrow b$

F $[q_1\ z_0\ q_1] \rightarrow \epsilon$

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

$S \to A|B$

o) $A \to aCA$

$A \to aD[q_1 z_0 q_0]$

$B \to aCB$

$B \to aDF$

$A \to aD[q_1 a q_0]$

$C \to aCC$

$D \to aDE$

$D \to aCD$

$D \to b$

$E \to b$

$F \to \varepsilon$

$S \to A|B$

- $S \to A|B$ useless pdn

$A \to aCA \mid aD[q_1 z_0 q_0] \mid aD[q_1 a q_0]$ useless pdn

$B \to aCB \mid aDF$

$C \to aCC$

$D \to aDE \mid aCD \mid b$

$E \to b$

$F \to \varepsilon$

- $S \to A|B$

$A \to aCA$ –

$B \to aCB \mid aDF$

$C \to aCC$

$D \to aDE \mid aCD \mid b$

$E \to b$

$F \to \varepsilon$

- check production $A \rightarrow aCA$. here there is no terminals. so delete i remove it (because it doesn't end) ie, self loop
- Then remove $S \rightarrow A$ from the production rule
- check $B \rightarrow aCB$, self loop generated so remove it (because no terminal)
- Remove $C \rightarrow aCC$ (self loop generated)

  There is no production rule for C

  ∴ remove $D \rightarrow aCD$. from production rule.

∴ Final production rule is.

$S \rightarrow B$

$B \rightarrow aDF$

$D \rightarrow aDE \mid b$

$E \rightarrow b$

$F \rightarrow \varepsilon$

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

$[q_0 z_0 q_1] \rightarrow a[q_0 a q_1][q_1 z_0 q_1]$

$[q_0 a q_1] \rightarrow a[q_0 a q_1][q_1 a q_1] \mid b$

$[q_1 a q_1] \rightarrow b$

$[q_1 z_0 q_1] \rightarrow \varepsilon$

Pumping Lemma for context free languages

Every infinite language L is context free if and only if for every sufficiently longer string in that language, we can always find 2 substring called V & X, that can be pumped as many times as we want, the resulting strings are in the language.

Pumping lemmas are used to prove that certain languages are non-context free by showing that there exists atleast one sufficiently longer string in the language that doesn't satisfy the above property.

Theorem

If L is a CFL, then L has pumping length P such that any string 's' where $|S| >= P$ may be divided into 5 pieces.

$S = uvxyz$ such that the following conditions must be true

1) $uv^k xy^k z$ is in L, $\forall k >= 0$
2) $|vy| > 0$
3) $|vxy| <= p$

8) Eg:- Prove that $L = \{ a^i \mid i \text{ is prime} \}$ is not a CFL?

Assume that L is context free, then there exist a pumping length p such that, every string 's' in L where $|s| >= p$ can be decided into 5 parts uvxyz such that.

Let $p = 7$, $i = 7$

$L = a^7 = \underbrace{a}_{u}\ \underbrace{aa}_{v}\ \underbrace{aa}_{x}\ \underbrace{a}_{y}\ \underbrace{a}_{z}$

Let $k = 2$, then uv^2xy^2z

$\Rightarrow$ a aaaa aa aa a

$\therefore |L| = 10$, which is not prime, uv^2xy^2z is not in L. So L is not in CFL

Hence proved

9) Prove that $L = \{ a^i b^i c^i \mid i >= 1 \}$ is not a CFL?

Soln:- Let $p = 5$, $i = 3$

Case 1

Here vy contains a, b, c

$L = \underbrace{aa}_{u}\ \underbrace{ab}_{v}\ \underbrace{bb}_{x}\ \underbrace{cc}_{y}\ \underbrace{c}_{z}$

Let $k = 2$, then uv^2xy^2z

aa abab bb cccc c $\Rightarrow$ which is not in L

Case 2

Vy vy contains b & c only

$h = \underbrace{aaa}_{u}\ \underbrace{b}_{v}\ \underbrace{bb}_{x}\ \underbrace{cc}_{y}\ \underbrace{c}_{z}$

Let $K = 2$, then uv^2xy^2z

$\Rightarrow$ aaa bb bb cccc c, which is not in h because no:of a's, b's & c's are not equal.

Case : 3

Vy vy contains a & b only.

$h = \underbrace{a}_{u}\ \underbrace{aa}_{v}\ \underbrace{bb}_{x}\ \underbrace{b}_{y}\ \underbrace{ccc}_{z}$

Let $K = 2$, then uv^2xy^2z

$\Rightarrow$ a aaaa bb bb ccc, which is not in L because no:of a's, b's and c's are not equal.

Hence uv^2xy^2z is not in L

∴ L is not Context Free language.

steps to follow

1) Assume that A is context Free
2) It has to have a pumping length (say p)
3) All the strings longer than p can be pumped
   $|S| \geq p$
4) now find a string (s) in A such that $|s| \geq p$.
5) Divide s in to uvxyz
6) Show that- $uv^ixy^iz \notin A$ for some i
7) Then consider the way that 's' can be divided in to uvxyz
8) show that- none of those can satisfy all the 3 pumping condition at the same time.
9) S cannot be pumped = contradiction.

Closure Properties of CFL

Union

Let L_1 & L_2 are two CFL. Then $L_1 \cup L_2$ is also context-Free.

Eg:- $L_1 = \{a^n b^n, n>0\}$, Grammar G_1
$P: S_1 \rightarrow aA1blab$

$L_2 = \{c^m d^m, m \geq 0\}$, Grammar G_2
$P: S_2 \rightarrow cBd|\epsilon$

cenion of L_1 & L_2

$= L_1 \cup L_2$

$= \{a^n b^n\} \cup \{c^m d^m\}$

$S \rightarrow S_1 | S_2$.

Concatenation

If L_1 & L_2 are CFL, then $L_1 L_2$ also context free.

L_1 & $L_2 \Rightarrow L_1 \cdot L_2$

Eg:- $L_1 = \{a^n b^n, n \geq 0\}$

$L_2 = \{c^m d^m, m \geq 0\}$

$\therefore L_1 \cdot L_2 \Rightarrow \{a^n b^n \cdot c^m d^m\}$

$S \rightarrow S_1 S_2$

Kleene star

If L is a CFL, then L^* is also context free

Eg:- $L_1 = \{a^n b^n, n \geq 0\}$,

$L_1^* = \{a^n b^n\}^*$

$\therefore$ CFL are closed under Kleen closure

Intersection & complementation

If L_1 & L_2 are 2 context free languages, their intersection

$L_1 \cap L_2$ need not be context free.

similarly, complementation of context free language L_1 which is

L', need not be context free.

so CFL are not closed under intersection and complementation.

Types of PDA

Push Down Automata (PDA)

- Nondeterministic Push Down Automata (NPDA)
- Deterministic Push Down Automata (DPDA)

Formal definition of DPDA

Collection of 7 tuples

$M = (Q, \Sigma, \delta, \Gamma, q_0, \Gamma, Z_0)$

Q = Finite set of states

Σ = Finite set of input alphabets

Γ = set of stack alphabets (symbols)

q_0 = Initial state

F = Final states sets.

Z_0 = A start symbol which is in Γ (ie, $Z_0 \in \Gamma$)

δ = Transition function

$$Q \times \{\Sigma \cup \epsilon\} \times \Gamma \rightarrow Q \times \Gamma^*$$

Formal definition of NPDA

Collection of 7 tuples

$M_2 = (Q, \Sigma, \delta, \Gamma, q_0, Z_0, F)$

Q = Finite set of states

Σ = Finite set of input alphabets

Γ = set of stack symbols

q_0 = Initial state

F = Set of final states

z_0 = Start symbol which is in Γ.
($z_0 \in \Gamma$)

δ = Transition function

$$Q \times \{\varepsilon \cup \Sigma\} \times \Gamma \longrightarrow 2^{Q \times \Gamma^*}$$

Acceptance by PDA

Acceptance by final state

- A string w accepted by a PDA, if starting with start state in the finite control, the start pushdown symbol on the top of the stack, by a sequence of moves, string w is read from left to right and the PDA enters a final state.
- Let $P = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$, the set accepted by PDA by final state is defined by.

$$T(P) = \{w \in \Sigma^* \mid (q_0, w, z_0) \xrightarrow{*} (q_i, \alpha)\}$$

for some $q_i \in F$ & $\alpha \in \Gamma^*$

Acceptance by empty stack

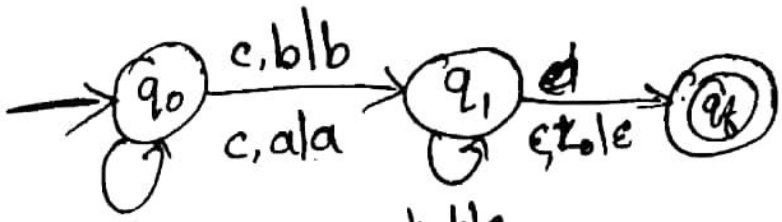
- A string w can be accepted by empty push down stack
- when the last input character is read the PDA accepts the string w or w' if the pushdown stack is empty after reading the right most input character.
- Let $P = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$, The set accepted by null store (empty stack / state) is defined as

$N(P) = \{ w \in \Sigma^* \mid (q_0, w, z_0) \xrightarrow{*} (q_i, \lambda, \lambda)$ for some $q_i \in Q$

5). Design a PDA for the language $\{ wcw^R, w \in (a,b)^* \}$.

Soln:- $L = \{ abbcbba, aca, bcb, \ldots \}$ (wcw — string c — reverse of string)

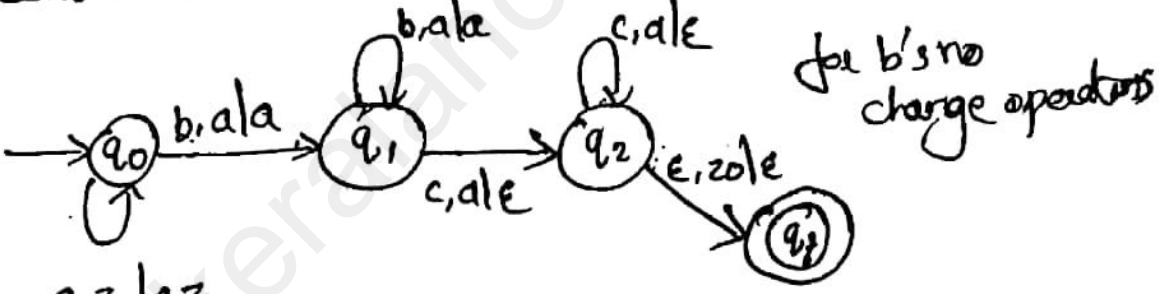
The PDA will go pushing symbol a, b on to the stack till it encounters 'c' on the input.
It reads c but will not push it to the stack.
After that for every symbol appearing next in the input it checks whether it matches with the topmost symbol on the stack.
If there is a match it pops the topmost element of the stack and advances the tape head one position right.



ε, ε|z0
a, z0|az0
a, a|aa
b, z0|bz0
b, b|bb
b, a|ba
a, b|ab

b, b|ε
a, a|ε

Q) Construct a PDA for $L = \{a^n b^m c^n \mid n \geq 1, m \geq 1\}$:



a, z0|az0
a, a|aa

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

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

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

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

$\delta(q_1, c, a) = (q_2, \epsilon)$

$\delta(q_2, c, a) = (q_2, \epsilon)$

$\delta(q_2, \epsilon, z_0) = (q_3, \epsilon)$

(9) Construct a PDA for $L=\{a^n b^{2n} \mid n \geq 1\}$?

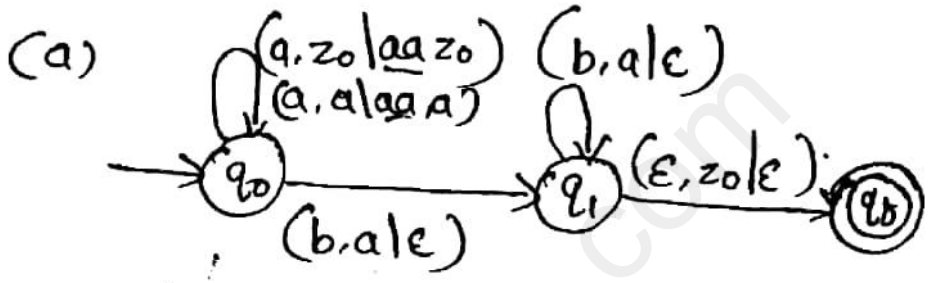
$L = \{aabbbb, abb, \ldots\}$

Soln:- We can solve the question in 2 ways.

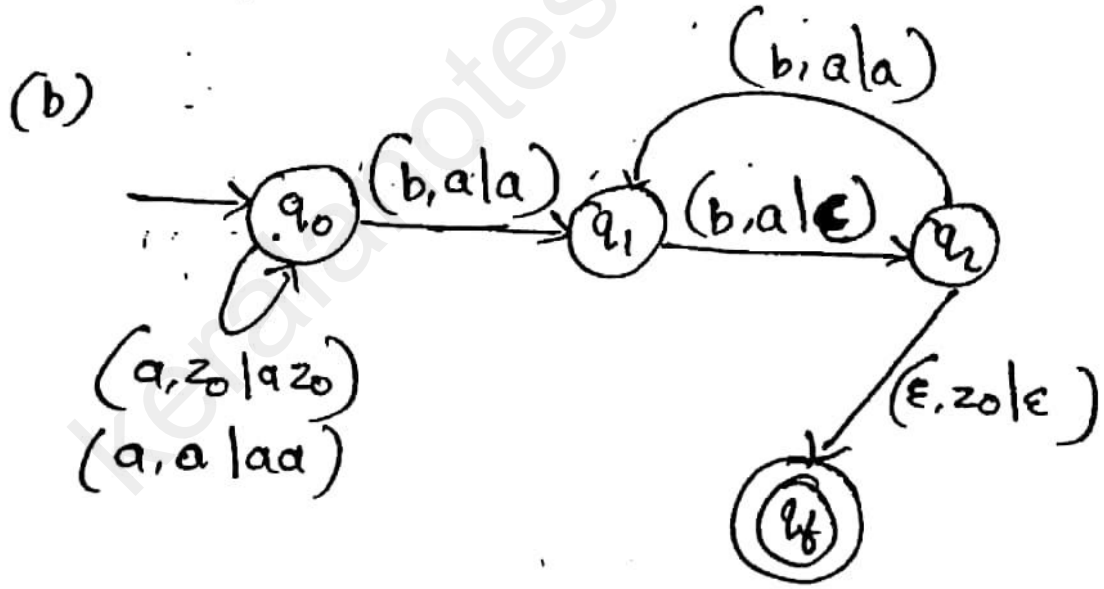
(a) For every a, push 2a's

(b) For every 2 b's pop 1a.

(a)

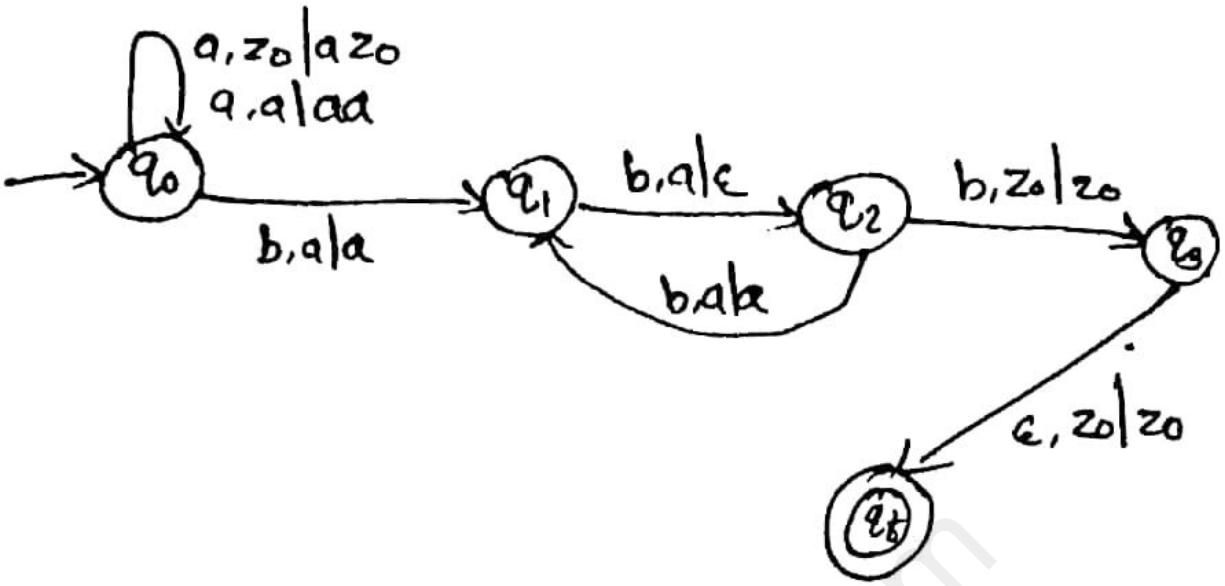


(b)



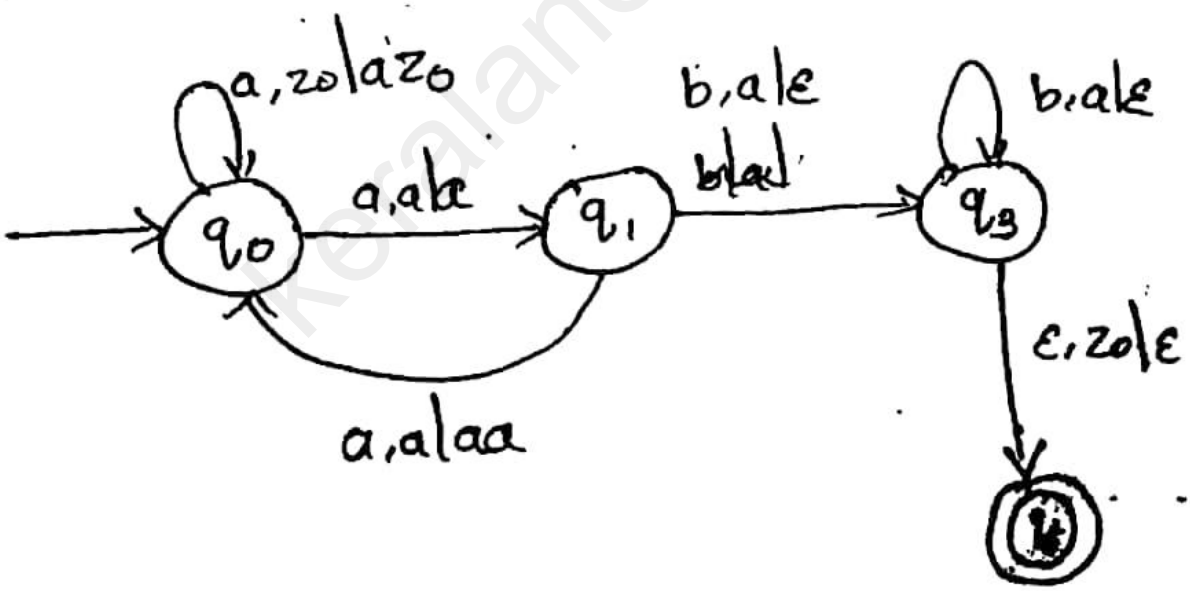
1) Construct a PDA for $L = \{ a^n b^{2n+1} \mid n \geq 1 \}$?

soln:-

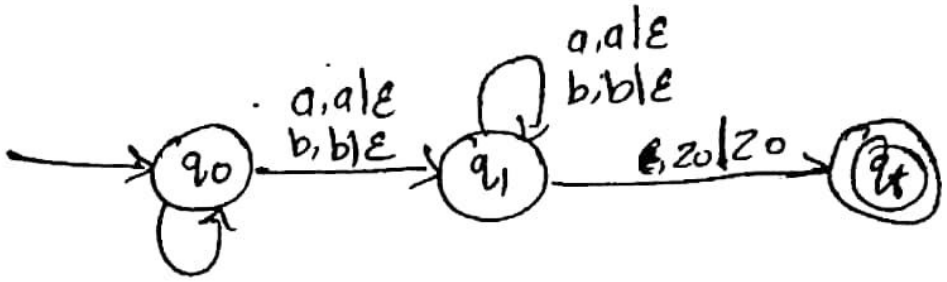


2) Construct a PDA for $L = \{ a^{2n} b^n \mid n \geq 1 \}$?

soln:-



(f) Construct a PDA for accepting the language $L = \{ww^R \mid w \in \{a,b\}^*\}$ — w – string, w^R – Reverse of a string.



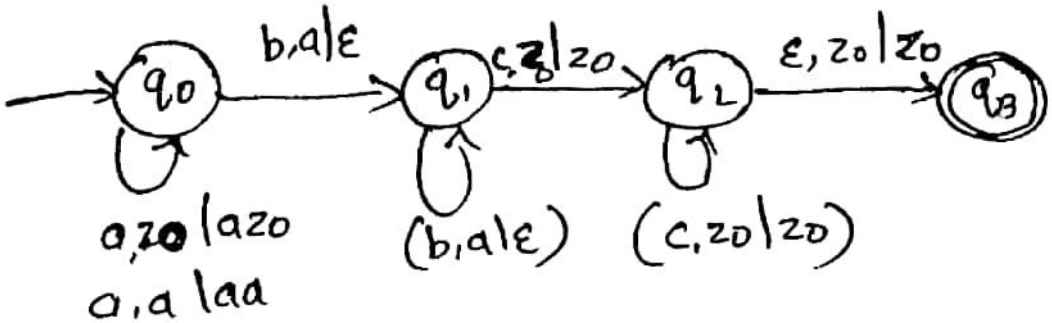
a, z0 | az0
b, z0 | az0
a, a | aa
b, b | bb
a, b | ab
b, a | ba

(g) Construct a PDA that accepts the language $L = \{a^n b^n c \mid n \geq 1\}$. where $\Sigma = \{a,b,c\}$

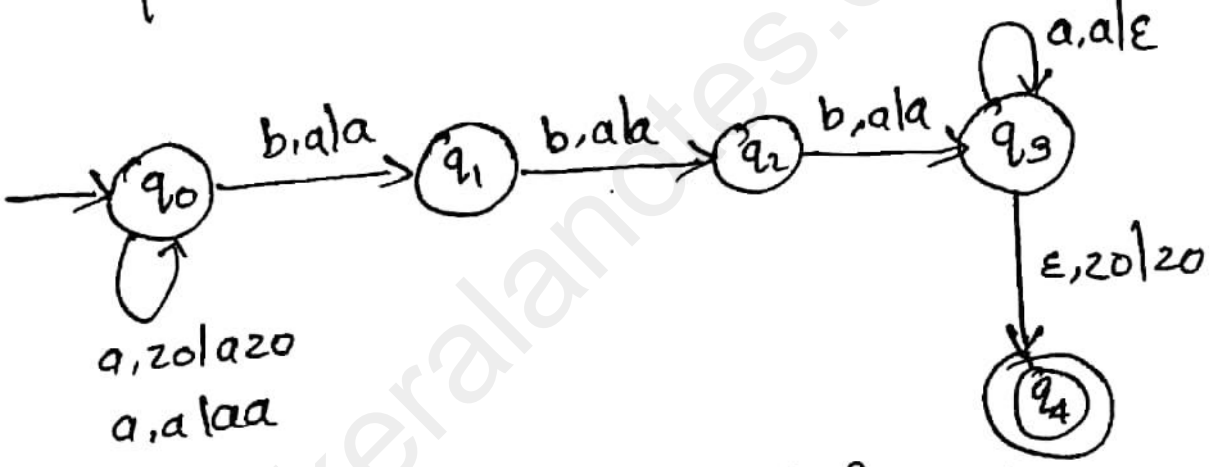
Soln:- $L = \{abc, aabbc, aaabbbc \ldots\}$

(a, z0 | az0)
(b, a | ε)
(a, a | aa)

(4) Construct a PDA that accepts the language $L = \{a^n b^n c^m \mid m, n \geq 1\}$, where $\Sigma = \{a, b, c\}$?



(5) Construct a PDA that accepts the language $L = \{a^n bbb a^n \mid n \geq 1\}$, where $\Sigma = \{a, b\}$?



(6) Construct a PDA that accepts the language $L = \{a^n b^{n+m} a^m \mid m, n \geq 1\}$, where $\Sigma = \{a, b\}$!

soln:- $L = a^n b^n . b^m a^m$. , $L = \{abba, aabbba, abbbaa, \ldots\}$