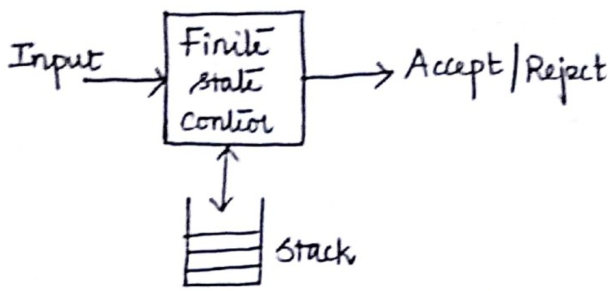
UNIT-III PUSHDOWN AUTOMATA

Pushdown Automata:

⇒ The pushdown automaton is essentially a finite automaton with control of both an input tape and a stack on which it can store a string of stack symbols.

⇒ With the help of a stack, the pushdown automaton can remember an infinite amount of information.



⇒ PDA consists of a finite set of states, a finite set of input symbols and a finite set of pushdown symbols.

⇒ The finite control has control of both the input tape and the pushdown store.

⇒ In one transition of the pushdown automaton,

- The control head reads the input symbol, then goto the new state.
- Replaces the symbol at the top of the stack by any string.

Definition of PDA:

A pushdown automaton consists of seven tuples

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

Where,

Q - A finite non empty set of states

Σ - A finite set of input symbols.

Γ - A finite non empty set of stack symbols.

q_0 - q_0 in Q is the start state

z_0 - Initial start symbol of the stack.

F - $F \subseteq Q$, set of accepting states or final states

δ - Transition function $Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow Q \times \Gamma^*$

Moves: The interpretation of

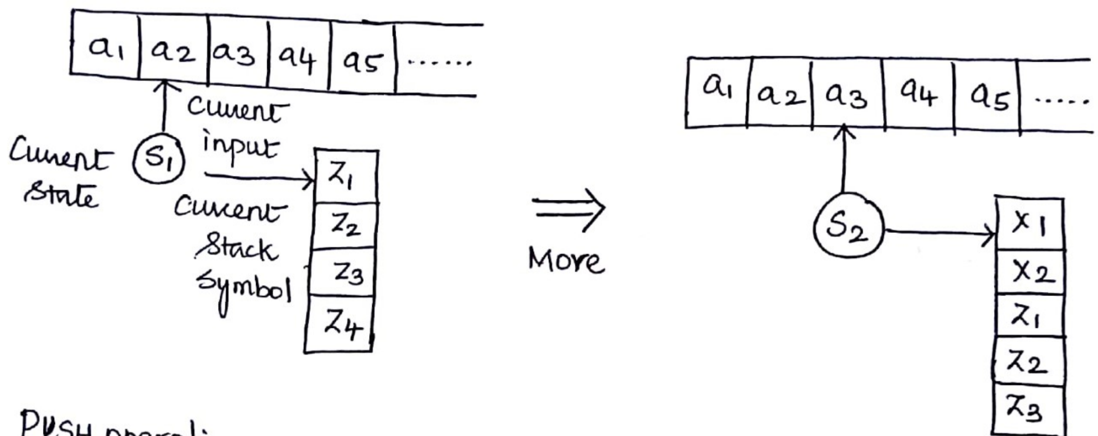
$$\delta(q, a, z) = \{(P_1, \gamma_1), (P_2, \gamma_2) \ldots (P_M, \gamma_M)\}$$

Where q, P_i – states a – input symbol z – Stack symbol

γ_i – a symbol in Γ^*

PDA enters state P_i, replaces the symbol z by the string γ_i and advances the input head one symbol.

Instantaneous Descriptions (ID)



PUSH operation:

Current Stack top

$$\delta(q_0, X, Z_0) = \delta(q_1, XZ_0)$$

push X onto the stack.

Read input on the tape

Change the state from q_0 to q_1

POP operation:

$$\delta(q_0, x, y) = \delta(q_1, \epsilon)$$

Pop the stack.

Read input on tape

Current stack top

Change the state from q_0 to q_1

Problem: PDA Constructions

1) Design a PDA for accepting a language $L = \{a^n b^n \mid n \geq 1\}$

Solution:

Logic: First we will push all a's onto the stack. Then reading every single b each a is popped from the stack.
If we read all b and remove all a's and if we get stack empty then that string will be accepted.

Instantaneous Description:

$\delta(q_0, a, z_0) = \{(q_0, az_0)\}$
$\delta(q_0, a, a) = \{(q_0, aa)\}$ } pushing the elements onto stack

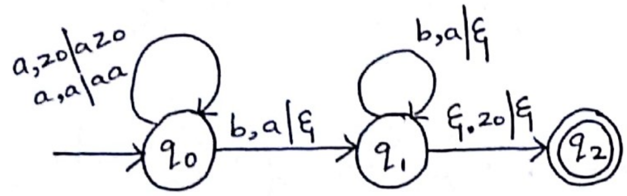
$\delta(q_0, b, a) = \{(q_1, \epsilon)\}$
$\delta(q_1, b, a) = \{(q_1, \epsilon)\}$
$\delta(q_1, \epsilon, z_0) = \{(q_2, \epsilon)\}$ } Popping the elements

PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, z_0, \{q_2\})$

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

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

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



Example: Let $n = 2$, String $w = a^2 b^2 = aabb$

$\delta(q_0, aabb, z_0) \vdash (q_0, aabb, z_0)$

$\vdash (q_0, abb, az_0)$

$\vdash (q_0, bb, aaz_0)$

$\vdash (q_1, b, az_0)$

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

$\vdash (q_2, \epsilon)$ Accept state.

2) Construct a PDA for $L = \{wcw^R \mid w \text{ in } (0+1)^*\}$

Solution:

Logic ⇒ For each move, the PDA writes a symbol on the top of the stack.

⇒ If the tape head reaches the input symbol c, stop pushing onto the stack.

⇒ Compare the stack symbol with the i/p symbol. if it matches pop the stack symbol

⇒ Repeat the process till reaches the final state or empty stack.

Instantaneous Description:

$$\left.\begin{aligned}
&\delta(q_0, 0, Z_0) = \{(q_0, 0Z_0)\}\\
&\delta(q_0, 1, Z_0) = \{(q_0, 1Z_0)\}\\
&\delta(q_0, 0, 0) = \{(q_0, 00)\}\\
&\delta(q_0, 0, 1) = \{(q_0, 01)\}\\
&\delta(q_0, 1, 0) = \{(q_0, 10)\}\\
&\delta(q_0, 1, 1) = \{(q_0, 11)\}
\end{aligned}\right\} \text{PUSH}$$

$$\left.\begin{aligned}
&\delta(q_0, C, 0) = \{(q_1, 0)\}\\
&\delta(q_0, C, 1) = \{(q_1, 1)\}
\end{aligned}\right\} \text{Accept the separator C}$$

$$\left.\begin{aligned}
&\delta(q_1, 0, 0) = \{(q_1, \epsilon)\}\\
&\delta(q_1, 1, 1) = \{(q_1, \epsilon)\}\\
&\delta(q_1, \epsilon, Z_0) = \{(q_2, \epsilon)\}
\end{aligned}\right\} \text{POP}$$

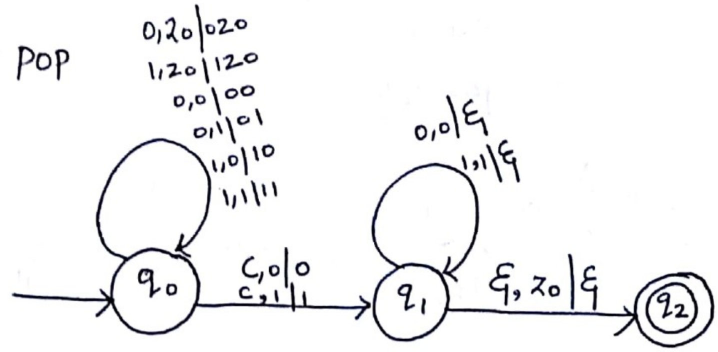
Here

$PDA = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, \{q_2\})$

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

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

$\Gamma = \{0, 1, Z_0\}$



Example:

$\delta(q_0, 100C001, Z_0) \vdash (q_0, 100C001, Z_0)$

$\vdash (q_0, 00C001, 1Z_0)$

$\vdash (q_0, 0C001, 01Z_0)$

$\vdash (q_0, C001, 001Z_0)$

$\vdash (q_1, 001, 001Z_0)$

$\vdash (q_1, 01, 01Z_0)$

$\vdash (q_1, 1, 1Z_0)$

$\vdash (q_1, \epsilon, Z_0)$

$\vdash (q_2, \epsilon)$ Accept State.

3) Construct PDA for the language $L = \{a^n b^{2n} \mid n \geq 1\}$.

Solution:

Logic: $L = \{$ 'n' number of a's followed by '2n' number of b's $\}$

If we read single 'a' push two a's onto the stack.

If we read 'b' then for every single 'b' only one 'a' should get popped from the stack.

Instantaneous Description:

$\delta(q_0, a, Z_0) = \{(q_0, aaZ_0)\}$
$\delta(q_0, a, a) = \{(q_0, aaa)\}$ } PUSH

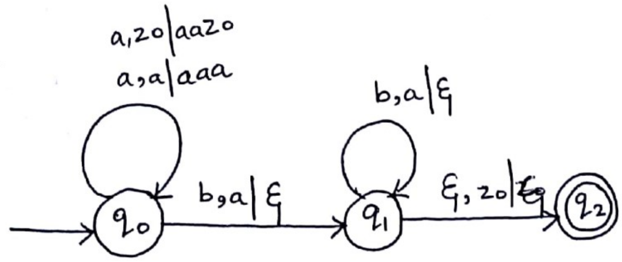
$\delta(q_0, b, a) = \{(q_1, \epsilon)\}$
$\delta(q_1, b, a) = \{(q_1, \epsilon)\}$
$\delta(q_1, \epsilon, Z_0) = \{(q_2, \epsilon)\}$ } POP

PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, \{q_2\})$

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

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

$\Gamma = \{a, Z_0\}$



Example: Let $n=2$. $L = \{a^2 b^4\}$ string $w = aabbbb$

$\delta(q_0, aabbbb, Z_0) \vdash (q_0, aabbbb, Z_0)$

$\vdash (q_0, abbbb, aaZ_0)$

$\vdash (q_0, bbbb, aaaaZ_0)$

$\vdash (q_1, bbb, aaaZ_0)$

$\vdash (q_1, bb, aaZ_0)$

$\vdash (q_1, b, aZ_0)$

$\vdash (q_1, \epsilon, Z_0)$

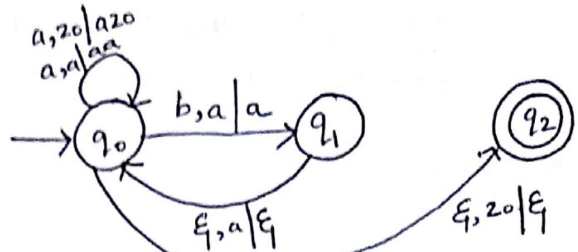
$\vdash (q_2, \epsilon)$ Accept state.

4) Construct the PDA for the language $L = \{a^{2n} b^n \mid n \geq 1\}$. Trace your PDA for the input with $n = 2$.

Solution:

Logic: When we read single 'b', single 'a' popped from the stack. For reading ϵ also single 'b' popped from the stack.

Instantaneous description:



$\delta(q_0, a, Z_0) = \{(q_0, aZ_0)\}$
$\delta(q_0, a, a) = \{(q_0, aa)\}$ — PUSH
$\delta(q_0, b, a) = \{(q_1, a)\}$
$\delta(q_1, \epsilon, a) = \{(q_0, \epsilon)\}$ — POP
$\delta(q_0, \epsilon, Z_0) = \{(q_2, \epsilon)\}$

PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, \{q_2\})$

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

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

$\Gamma = \{a, Z_0\}$

Example: Let $n = 2$ string $w = a^4 b^2 = aaaabb$

$\delta(q_0, aaaabb, Z_0) \vdash (q_0, aaaabb, Z_0)$

$\vdash (q_0, aaabb, aZ_0)$

$\vdash (q_0, aabb, aaZ_0)$

$\vdash (q_0, abb, aaaZ_0)$

$\vdash (q_0, bb, aaaaZ_0)$

$\vdash (q_1, b, aaaZ_0)$

$\vdash (q_0, b, aaZ_0)$

$\vdash (q_1, \epsilon, aZ_0)$

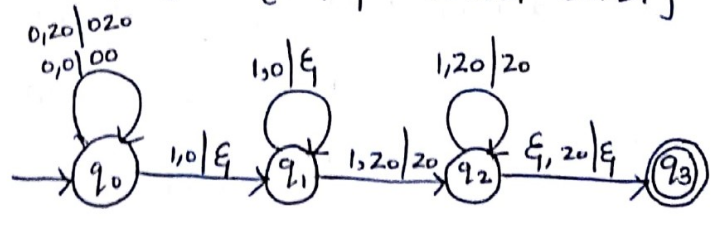
$\vdash (q_0, \epsilon, Z_0)$

$\vdash (q_2, \epsilon)$ Accept state.

5) Construct the DPDA for the language $L = \{0^n 1^m \mid n < m \text{ and } n, m \geq 1\}$

Solution:

ID:



$\delta(q_0, 0, Z_0) = \{(q_0, 0Z_0)\}$

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

$\delta(q_0, 1, 0) = \{(q_1, \epsilon)\}$

$\delta(q_1, 1, 0) = \{(q_1, \epsilon)\}$

$\delta(q_1, 1, Z_0) = \{(q_2, Z_0)\}$

$\delta(q_2, 1, Z_0) = \{(q_2, Z_0)\}$

$\delta(q_2, \epsilon, Z_0) = \{(q_3, \epsilon)\}$

Example: $\delta(q_0, 00111, Z_0) \vdash (q_0, 00111, Z_0)$

$\vdash (q_0, 0111, 0Z_0)$

$\vdash (q_0, 111, 00Z_0)$

$\vdash (q_1, 11, 0Z_0)$

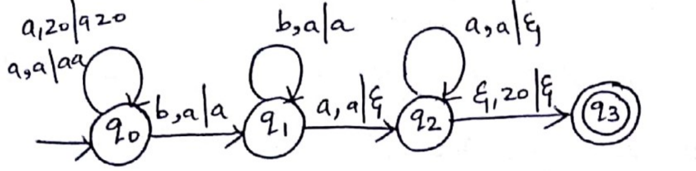
$\vdash (q_1, 1, Z_0)$

$\vdash (q_2, \epsilon, Z_0)$

$\vdash (q_3, \epsilon)$ Accept state

6) Construct the PDA for the language $L=\{a^n b^m a^n \mid m,n \geq 1\}$

Solution:



ID:

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

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

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

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

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

$\delta(q_2, a, a) = \{(q_2, \epsilon)\}$

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

Example: $n=2, m=2$ String $w = a^2 b^2 a^2 = aabbaa$

$\delta(q_0, aabbaa, z_0) \vdash (q_0, aabbaa, z_0)$

$\vdash (q_0, abbaa, az_0)$

$\vdash (q_0, bbaa, aaz_0)$

$\vdash (q_1, baa, aaz_0)$

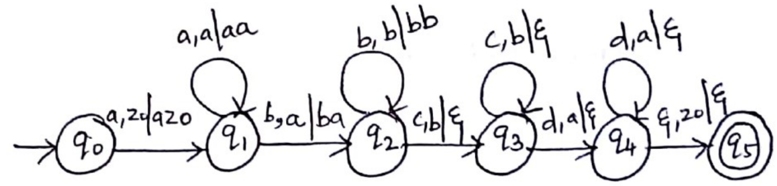
$\vdash (q_1, aa, aaz_0)$

$\vdash (q_2, a, az_0)$

$\vdash (q_2, \epsilon, z_0) \vdash (q_3, \epsilon)$ Accept state.

7) Construct the PDA for the language $L=\{a^n b^m c^m d^n \mid m,n \geq 1\}$

Solution



ID:

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

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

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

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

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

$\delta(q_3, c, b) = \{(q_3, \epsilon)\}$

$\delta(q_3, d, a) = \{(q_4, \epsilon)\}$

$\delta(q_4, d, a) = \{(q_4, \epsilon)\}$

$\delta(q_4, \epsilon, z_0) = \{(q_5, \epsilon)\}$

Example:

$\delta(q_0, aabbccdd, z_0) \vdash (q_0, aabbccdd, z_0)$

$\vdash (q_1, abbccdd, az_0)$

$\vdash (q_1, bbccdd, aaz_0)$

$\vdash (q_2, bccdd, baaz_0)$

$\vdash (q_2, ccdd, bbaaz_0)$

$\vdash (q_3, cdd, baaz_0)$

$\vdash (q_3, dd, aaz_0)$

$\vdash (q_4, d, az_0)$

$\vdash (q_4, \epsilon, z_0)$

$\vdash (q_5, \epsilon)$ Accept state.

Deterministic pushdown automata

A PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ is deterministic if and only if it satisfies the following condition.

(i) $\delta(q, a, x)$ has at most one element

(ii) If $\delta(q, a, x)$ is nonempty for some $a \in \Sigma$ then $\delta(q, \epsilon, x)$ must be empty.

Non-Deterministic pushdown Automata

The non-deterministic pushdown automata is very much similar to NFA. The CFG which accept deterministic PDA accept non-deterministic PDAs as well.

Similarly there are some CFG's which can be accepted only by NDPA and not by DPDA. Thus NDPA is more powerful than DPDA.

Compare NFA and PDA.

	NFA	PDA
1.	NFA stands for non-deterministic finite automata.	PDA stands for pushdown automata.
2.	This model does not have memory to remember input symbols.	This model has stack memory to remember input symbols.
3.	It is always non-deterministic. It has two versions. (i) NFA with ϵ (ii) NFA without ϵ.	It has two Versions. (i) Deterministic PDA (ii) Non-deterministic PDA.

Equivalence : pushdown automata to CFL.

Let, $P = (Q, \Sigma, \Gamma, \delta, q_0, z_0, q_n)$ is a PDA there exists CFG G which is accepted by PDA P. The G can be defined as,

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

Where S is a start symbol, T – Terminals V – Non-terminals

For getting production rules P, we follow the following algorithm.

Algorithm for getting production rules of CFG

1. If q_0 is start state in PDA and q_n is final state of PDA then $[q_0 \; Z \; q_n]$ becomes start state of CFG.
2. The production rule for the ID of the form $\delta(q_i, a, z_0) = (q_{i+1}, Z_1 Z_2)$ can be obtained as,

$$\delta\ (q_i \; z_0 \; q_{i+K}) \rightarrow a\ (q_{i+1} \; Z_1 \; q_m)\ (q_m \; Z_2 \; q_{i+K})$$

Where q_{i+K}, q_m represents the intermediate states, z_0, Z_1, Z_2 are stack symbols and a is input symbol.

3. The production rule for the ID of the form.

$\delta(q_i, a, z_0) = (q_{i+1}, \epsilon)$ can be converted as $(q_i \; z_0 \; q_{i+1}) \rightarrow a$

Problems: PDA to CFG

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

$$\delta(q_0, 0, z_0) = \{(q_0, X z_0)\}$$
$$\delta(q_0, 0, X) = \{(q_0, XX)\}$$
$$\delta(q_0, 1, X) = \{(q_1, \epsilon)\}$$
$$\delta(q_1, 1, X) = \{(q_1, \epsilon)\}$$
$$\delta(q_1, \epsilon, X) = \{(q_1, \epsilon)\}$$
$$\delta(q_1, \epsilon, z_0) = \{(q_1, \epsilon)\}$$

Construct CFG $G = (V, T, P, S)$ generating N(M).

Solution: Given $M = (\{q_0, q_1\}, \{0, 1\}, \{X, Z_0\}, \delta, q_0, Z_0, \phi)$

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

$T = \{0, 1\}$

$V = (S, [q_0, X, q_0], [q_0, X, q_1], [q_1, X, q_0], [q_1, X, q_1]$
$[q_0, Z_0, q_0], [q_0, Z_0, q_1], [q_1, Z_0, q_0], [q_1, Z_0, q_1])$

Start state production S

$S \rightarrow [q_0, Z_0, q_0]$
$S \rightarrow [q_0, Z_0, q_1]$

Now productions for $[q_0, Z_0, q_0]$ and $[q_0, Z_0, q_1]$

(1) $\delta(q_0, 0, Z_0) = \{(q_0, XZ_0)\}$

$[q_0, Z_0, q_0] \rightarrow 0 [q_0, X, q_0] [q_0, Z_0, q_0]$
$[q_0, Z_0, q_0] \rightarrow 0 [q_0, X, q_1] [q_1, Z_0, q_0]$
$[q_0, Z_0, q_1] \rightarrow 0 [q_0, X, q_0] [q_0, Z_0, q_1]$
$[q_0, Z_0, q_1] \rightarrow 0 [q_0, X, q_1] [q_1, Z_0, q_1]$

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

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

(iii) $\delta(q_1, 1, X) = (q_1, \epsilon)$

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

(iv) $\delta(q_1, \epsilon, X) = (q_1, \epsilon)$

$[q_1, X, q_1] \rightarrow \epsilon$

(v) $\delta(q_0, 1, X) = (q_1, \epsilon)$

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

(vi) $\delta(q_1, \epsilon, Z_0) = (q_1, \epsilon)$

$[q_1, Z_0, q_1] \rightarrow \epsilon$

After analysing all the productions

useless production $\Rightarrow [q_0, Z_0, q_0]$ $[q_0, X, q_0]$

Unknown productions $\Rightarrow [q_1, Z_0, q_0]$ $[q_1, X, q_0]$

Deleting all these productions, Final productions are.

$S \rightarrow [q_0, Z_0, q_1]$

$[q_0, Z_0, q_1] \rightarrow 0 [q_0, X, q_1] [q_1, Z_0, q_1]$
$[q_0, X, q_1] \rightarrow 0 [q_0, X, q_1] [q_1, X, q_1]$
$[q_0, X, q_1] \rightarrow 1$
$[q_1, Z_0, q_1] \rightarrow \epsilon$
$[q_1, X, q_1] \rightarrow \epsilon$
$[q_1, X, q_1] \rightarrow 1$

After Renaming

$$\boxed{\begin{array}{l} S \rightarrow A \\ A \rightarrow 0BC \\ B \rightarrow 0BD \mid 1 \\ C \rightarrow \epsilon \\ D \rightarrow 1 \mid \epsilon \end{array}}$$

Scanned by CamScanner

2) Let $M = (\{q_0, q_1\}, \{a, b\}, \{Z, Z_0\}, \delta, q_0, Z_0, \phi)$ where δ is given by

$\delta(q_0, b, Z_0) = \{(q_0, ZZ_0)\}$ $\quad$ $\delta(q_0, a, Z) = \{(q_1, Z)\}$

$\delta(q_0, \epsilon, Z_0) = \{(q_0, \epsilon)\}$ $\quad$ $\delta(q_1, b, Z) = \{(q_1, \epsilon)\}$

$\delta(q_0, b, Z) = \{(q_0, ZZ)\}$ $\quad$ $\delta(q_1, a, Z_0) = \{(q_0, Z_0)\}$

Construct CFG $G = (V, T, P, S)$ generating $N(M)$.

Solution: Given $M = (\{q_0, q_1\}, \{a, b\}, \{Z, Z_0\}, \delta, q_0, Z_0, \phi)$

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

$T = \{a, b\}$

$V = S, [q_0, Z, q_0], [q_0, Z, q_1], [q_1, Z, q_0], [q_1, Z, q_1]$
$[q_0, Z_0, q_0], [q_0, Z_0, q_1], [q_1, Z_0, q_0], [q_1, Z_0, q_1])$

Start state production S

$S \rightarrow [q_0, Z_0, q_0]$

$S \rightarrow [q_0, Z_0, q_1]$

Now productions for $[q_0, Z_0, q_0]$ and $[q_0, Z_0, q_1]$

(i) $\delta(q_0, b, Z_0) = \{(q_0, ZZ_0)\}$

$[q_0, Z_0, q_0] \rightarrow b [q_0, Z, q_0] [q_0, Z_0, q_0]$

$[q_0, Z_0, q_0] \rightarrow b [q_0, Z, q_1] [q_1, Z_0, q_0]$

$[q_0, Z_0, q_1] \rightarrow b [q_0, Z, q_0] [q_0, Z_0, q_1]$

$[q_0, Z_0, q_1] \rightarrow b [q_0, Z, q_1] [q_1, Z_0, q_1]$

(ii) $\delta(q_0, \epsilon, Z_0) = \{(q_0, \epsilon)\}$

$[q_0, Z_0, q_0] \rightarrow \epsilon$

(iii) $\delta(q_0, b, Z) = \{(q_0, ZZ)\}$

$[q_0, Z, q_0] \rightarrow b [q_0, Z, q_0] [q_0, Z, q_0]$

$[q_0, Z, q_0] \rightarrow b [q_0, Z, q_1] [q_1, Z, q_0]$

$[q_0, Z, q_1] \rightarrow b [q_0, Z, q_0] [q_0, Z, q_1]$

$[q_0, Z, q_1] \rightarrow b [q_0, Z, q_1] [q_1, Z, q_1]$

(iv) $\delta(q_0, a, Z) = (q_1, Z)$

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

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

(v) $\delta(q_1, b, Z) = \{(q_1, \epsilon)\}$

$[q_1, Z, q_1] \rightarrow b$

(vi) $\delta(q_1, a, Z_0) = \{(q_0, Z_0)\}$

$[q_1, Z_0, q_0] \rightarrow a [q_0, Z_0, q_0]$

$[q_1, Z_0, q_1] \rightarrow a [q_0, Z_0, q_1]$

After analysing all the productions,

Useless productions $\Rightarrow [q_0, Z, q_0] \; [q_0, Z_0, q_1]$

Unknown productions $\Rightarrow [q_1, Z_0, q_1] \; [q_1, Z, q_0]$

Deleting all these productions.

$S \rightarrow [q_0, Z_0, q_0]$

$[q_0, Z_0, q_0] \rightarrow b [q_0, Z, q_1] [q_1, Z_0, q_0]$

$[q_0, Z, q_1] \rightarrow b [q_0, Z, q_1] [q_1, Z, q_1]$

$[q_0, Z_0, q_0] \rightarrow \epsilon$

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

$[q_1, Z, q_1] \rightarrow b$

$[q_1, Z_0, q_0] \rightarrow a [q_0, Z_0, q_0]$

After Renaming

$$\boxed{\begin{aligned} S &\rightarrow A \\ A &\rightarrow bBC \mid \epsilon \\ B &\rightarrow bBD \mid aD \\ D &\rightarrow b \\ C &\rightarrow aA \end{aligned}}$$

3) Construct a PDA accepting $\{a^n b^m a^n \mid m, n \geq 1\}$ by empty stack. Also construct the corresponding context free grammar accepting the same set.

4) Let $M = (\{P, q\}, \{0, 1\}, \{X, Z_0\}, \delta, q, Z_0)$ where δ is given by

$\delta(q, 1, Z_0) = \{(q, XZ_0)\}$ $\delta(q, \epsilon, Z_0) = \{(q, \epsilon)\}$

$\delta(q, 1, X) = \{(q, XX)\}$ $\delta(P, 1, X) = \{(P, \epsilon)\}$

$\delta(q, 0, X) = \{(P, X)\}$ $\delta(P, 0, Z_0) = \{(q, Z_0)\}$

Construct CFG $G = (V, T, P, S)$ generating $N(M)$.

Equivalence : CFL to pushdown automata

Algorithm:

(1) Convert the CFG to Greibach Normal form.

(2) The δ function is to be developed for the grammar of the form

$A \rightarrow aB$ as $\delta(q_i, a, A) \rightarrow \delta(q_i, B)$

(3) Finally add the rule

$\delta(q_i, \epsilon, Z_0) \rightarrow (q_i, \epsilon)$

Where Z_0 – Stack symbol (Accepting state)

Problem 1: Construct PDA for the following grammar.

$S \rightarrow AB$, $B \rightarrow b$, $A \rightarrow CD$, $C \rightarrow a$, $D \rightarrow a$

Solution:

GNF form:

$S \rightarrow AB$

$\rightarrow CDB$

$\rightarrow aDB$

$A \rightarrow CD$

$\rightarrow aD$

$B \rightarrow b$

$C \rightarrow a$

$D \rightarrow a$

Equivalent PDA is

$\delta(q_1, a, S) \rightarrow (q_1, DB)$

$\delta(q_1, a, A) \rightarrow (q_1, D)$

$\delta(q_1, b, B) \rightarrow (q_1, \epsilon)$

$\delta(q_1, a, C) \rightarrow (q_1, \epsilon)$

$\delta(q_1, a, D) \rightarrow (q_1, \epsilon)$

Example: $\delta(q_1, aab, S) \vdash \delta(q_1, ab, DB)$

$\vdash \delta(q_1, b, B)$

$\vdash \delta(q_1, \epsilon, Z_0)$

$\vdash \delta(q_f, \epsilon)$ Accepting state.

Problem 2: Construct an unrestricted PDA equivalent of the grammar given below

$S \to aAA, \quad A \to aS \mid bS \mid a$

Solution:

The given grammar is already in GNF. Hence the PDA can be-

$\delta(q_1, a, S) \to (q_1, AA)$

$\delta(q_1, a, A) \to (q_1, S)$

$\delta(q_1, b, A) \to (q_1, S)$

$\delta(q_1, a, A) \to (q_1, \epsilon)$

$\delta(q_1, \epsilon, z_0) \to (q_1, \epsilon)$ Accept.

The simulation of abaaaa is,

$\delta(q_1, abaaaa, S) \vdash \delta(q_1, baaaa, AA)$

$\vdash \delta(q_1, aaaa, SA)$

$\vdash \delta(q_1, aaa, AAA)$

$\vdash \delta(q_1, aa, AA)$

$\vdash \delta(q_1, a, A)$

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

$\vdash \delta(q_1, \epsilon)$ Accept.

Problem 3: Consider GNF $G = (\{S, T, C, D\}, \{a, b, c, d\}, S, P)$ where P is.

$S \to cCD \mid dTC \mid \epsilon \qquad C \to aTD \mid c$

$T \to cDC \mid cST \mid a \qquad D \to dC \mid d$

Present a PDA that accepts the language generated by this grammar.

Solution:

Let PDA $M = \{\{q\}, \{c, a, d\}, \{S, T, C, D, c, d, a\}, \delta, q, S, \phi\}$

The production rules δ is given by

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

$\delta(q, \epsilon, C) = \{(q, aTD), (q, c)\}$

$\delta(q, \epsilon, T) = \{(q, cDC), (q, cST), (q, a)\}$

$\delta(q, \epsilon, D) = \{(q, dC), (q, d)\}$

$\left.\begin{array}{l} \delta(q, c, c) = \{(q, \epsilon)\} \\ \delta(q, d, d) = \{(q, \epsilon)\} \\ \delta(q, a, a) = \{(q, \epsilon)\} \end{array}\right\}$ Acceptance by Empty stack

Simulation for String "caadd"

$\delta(q, \epsilon, S) \vdash \delta(q, caadd, S)$

$\vdash \delta(q, aadd, CD)$

$\vdash \delta(q, add, TDD)$

$\vdash \delta(q, dd, DD)$

$\vdash \delta(q, d, D)$

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

Problem 4: Find the PDA equivalent to given CFG with following productions.
$S \to A$, $A \to BC$, $B \to ba$, $C \to aC$

Problem 5: convert the grammar $S \to aSb \mid A$, $A \to bSa \mid S \mid \epsilon$ to a PDA That accepts the same language by empty stack.

Problem 6: convert the grammar $S \to 0S1 / A$, $A \to 1A0 \mid S \mid \epsilon$ into PDA that accepts the same language by empty stack. check whether 0101 belongs to N(M).

Problem 7: construct CFL for the grammar $S \to aSbb \mid a$ and also construct its corresponding PDA.

Sln: $\{L = a^n b^m \mid m > n\} \to$ PDA

Problem 8: construct CFL for the grammar $S \to aSa \mid bSb \mid \epsilon$ and also construct its corresponding PDA.

Sln: $L = \{WW^R / W \text{ is in } (a+b)^*\} \to$ PDA.

Problem 9: construct CFL for the grammar $S \to aSb \mid A$, $A \to bSa \mid S \mid \epsilon$ And also construct its corresponding PDA.

Sln: $L = \{a^n b^n \mid n \geq 1\} \to$ PDA.

Problem 10: convert the grammar $E \to E+E$, $E \to id$ into PDA and trace the string "id+id+id".

Pumping lemma for CFL

Lemma: Let L be any CFL. Then there is a constant n, depending only on L, such that z is in L and $|z| \geq n$, then we can write $z = uvxyz$ such that

(i) $|vxy| \leq n$

(ii) $|vy| \geq 1$ (or) $|vy| \neq \epsilon$

(iii) for all $i \geq 0$, $uv^ixy^iz \in L$.

Problems:

1) Prove that $L = \{a^ib^ic^i \mid i \geq 1\}$ is not context free Language.

Solution:

(i) Let us assume that L is regular / CFL

(ii) Let $w = a^ib^ic^i$ where i is constant

(iii) W can be written as $uvxyz$ where

(a) $|vxy| \leq n$

(b) $|vy| \neq \epsilon$

(c) for all $i \geq 0$, $uv^ixy^iz \in L$

Since $vy \neq \epsilon$, Either $v = ab \mid bc \mid ca$ (or)

$y = ab \mid bc \mid ca$.

If $i=2$, $uv^ixy^iz = uv^2xy^2z$ becomes

Case (i) If $v=ab$ and $y=c$

$$uv^2xy^2z = (ab)^2c^2 \Rightarrow uv^ixy^iz \notin L$$

Case (ii) If $v=a$ and $y=bc$

$$uv^2xy^2z = a^2(bc)^2 \Rightarrow uv^ixy^iz \notin L$$

Hence L is not a CFL.

2) Prove that $L=\{a^i b^i c^j \mid j>i\}$ is not CFL.

Solution:

(i) Let us assume that L is CFL

(ii) Let $w=a^i b^i c^j$, where i,j is a constant

(iii) W can be written as, uvxyz where

(a) $|vxy| \le n$

(b) $vy \ne \epsilon$

(c) for all $i \ge 0$, $uv^ixy^iz \in L$.

Case(i) If v=ab and y=c

$i=2$, $uv^2xy^2z = (ab)^2 c^2 \notin L$

Since, Power of c should be greater than ab

Case(ii) If v=a and y=bc

$uv^2xy^2z = (a)^2 (bc)^2 \notin L$, $j>i$ is not true

Hence L is not a CFL.

3) Prove that $L=\{a^n b^m c^p \mid 0 \le n < m < p\}$ is not CFL.

Solution:

(i) Let $L=\{a^n b^m c^p \mid 0 \le n < m < p\}$

(ii) Let $w= a^n b^{n+1} c^{n+2}$, where n is constant [Since $n<m<p$]

(iii) W can be rewritten as uvxyz where

(a) $|vxy| \le n$

(b) $vy \ne \epsilon$

(c) for all $i \ge 0$, $uv^ixy^iz \in L$.

Case(i) If v=ab, and y=c

$i=2$, $uv^2xy^2z = (ab)^2 c^2$ — ①

Case(ii) If v=a and y=bc

$i=2$, $uv^2xy^2z = a^2 (bc)^2$ — ②

From ① and ② $uv^ixy^iz \notin L$.

Hence L is not a CFL.

4) Prove that $L=\{0^i 1^j 2^i 3^j \mid i \ge 1, j \ge 1\}$ is not CFL.

Solution:

(i) Let us assume that L is CFL

(ii) Let $W= 0^n 1^n 2^n 3^n$ Where n is constant

(iii) Let W can be rewritten as, uvxyz where,

(a) $|vxy| \le n$

(b) $|vy| \ne \epsilon$

(c) for all $i \ge 0$, $uv^ixy^iz \in L$.

Case(i) if v=01 and y=2

$i=2$, $uv^2xy^2z = (01)^2 2^2 3^1$ — ①

Case(ii) if v=12 and y=3

$i=2$, $uv^2xy^2z = (12)^2 (3)^2 0^1$ — ②

From ① and ② $uv^ixy^iz \notin L$

∴ Given L is not a CFL.