Pushdown Automata

Pushdown Automata

- What is PDA?
- Why is the need of PDA?
- Model of PDA
- Mathematical Definition of PDA
- Moves of PDA
- ID of PDA
- Move relation
- Language accepted by PDA
- Representation of PDA

What is PDA?

- It is an automata or machine which is used to recognize a set of strings.
- A PDA is an enhancement of finite automata(FA).
- Finite automata with a stack memory can be viewed as Pushdown automata.

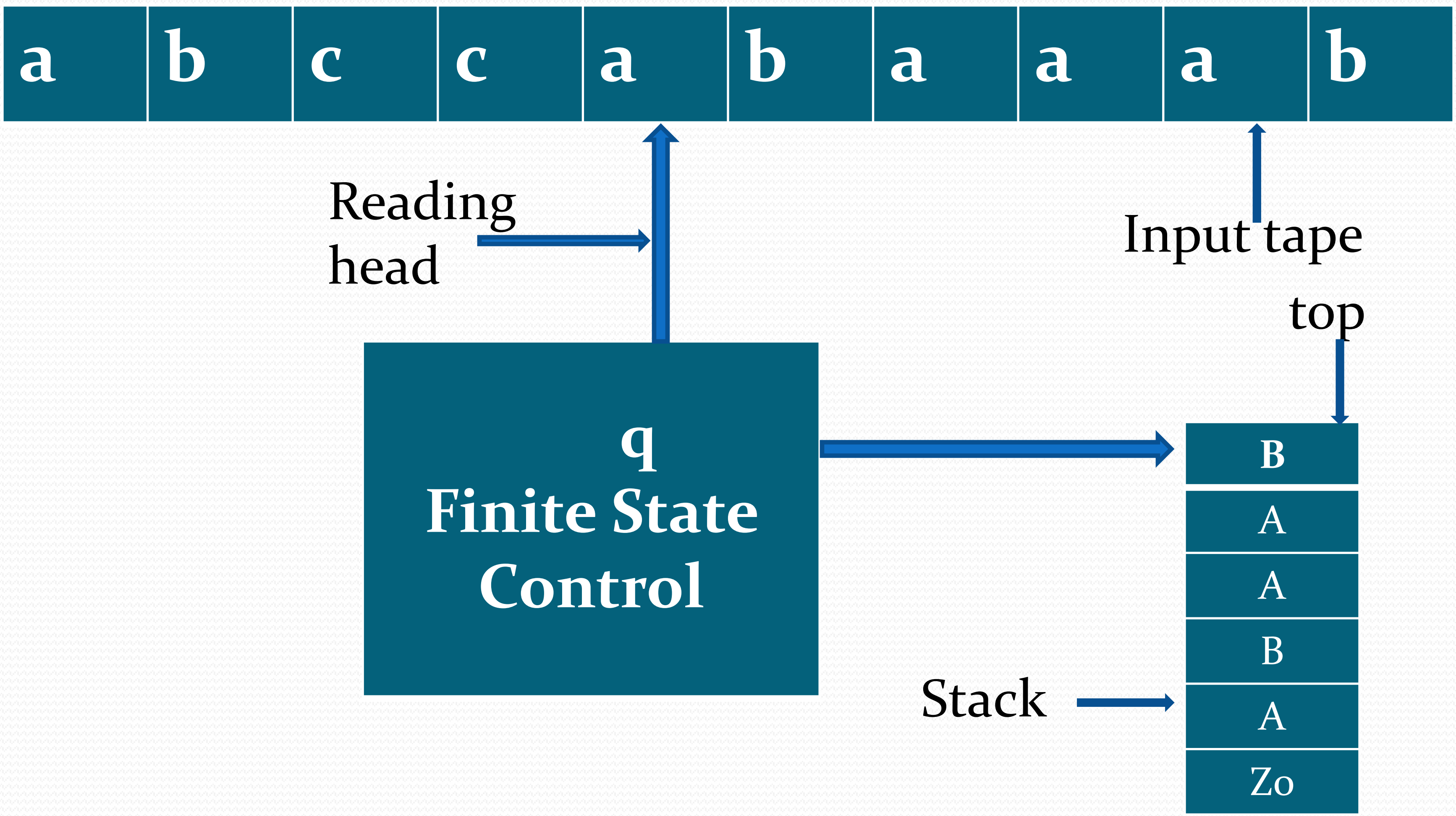
  PDA = FA + Stack

- Addition of stack memory enhances the capability of Pushdown automata as compared to Finite automata.

Why is the need of PDA?

- Since Finite automata can accept only regular languages, therefore FA is incapable for accepting context free language, context sensitive language and recursive language.
- PDA is designed for accepting context free languages.
- Since every regular language is also context free, therefore PDA can accept both type of languages i.e. regular and context free language.

Model of PDA



a b c c a b a a a b
Reading head
Input tape
top
q
Finite State Control
B
A
A
B
A
Zo
Stack

Mathematical Definition of PDA

A Pushdown automata is described by a 7-tuple

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

- Q is the finite set of states,
- Σ is the set of input symbols
- Γ is the set of stack symbols,
- $q_o \in Q$ is the initial state,
- $Z_o \in \Gamma$ is a bottom symbol of stack
- F is the set of final states, and
- δ is a transition function which is defined as following:-
- $\delta: Q\times(\Sigma \cup \{\epsilon\})\times\Gamma \rightarrow$ finite subset of $Q\times\Gamma^*$

Moves of PDA

There are two types of moves in PDA.

Type-1:

$\delta(q, a, Z) = \{(p_1, \alpha_1), (p_2, \alpha_2), \ldots\ldots\ldots. (p_n, \alpha_n)\}$,

where $q, p_i \in Q$, $a \in \Sigma$, $Z \in \Gamma$ $\alpha_i \in \Gamma$ and $1 \leq i \leq n$.

Type-2:

$\delta(q, \epsilon, Z) = \{(p_1, \alpha_1), (p_2, \alpha_2), \ldots\ldots\ldots. (p_n, \alpha_n)\}$,

where $q, p_i \in Q$, $a \in \Sigma$, $Z \in \Gamma$ $\alpha_i \in \Gamma$ and $1 \leq i \leq n$.

Instantaneous Description(ID)

An instantaneous description of PDA is of the following form:-

(q, x, α)

Where $q \in Q$, $x \in \Sigma^*$ and $\alpha \in \Gamma^*$.

Here α represents whole contents with in stack.

And initial ID will be

(q_o, x, Z_o)

q_o → The initial state

x → The input string which we have taken for processing

Z_o → Initial contents with in stack.

Move relation

This relation exist between two consecutives ID's. It is dented by $\vdash_M$.

$$(q, ax, Z\beta) \vdash_M (p, x, \alpha\beta)$$

If $\delta(q, a, Z)$ contains (p, α), where $Z, \beta, \alpha \in \Gamma^*$, a may be ϵ or $a \in \Sigma$, and $p , q \in Q$.

Language accepted by PDA

PDA can accept languages either by final state or empty stack.

Language accepted by final state

It is denoted by L(M). It is defined as following:-

$L(M) = \{ x\ !\ (q_o , x, Z_o) \vdash_M^* (f, \epsilon, \alpha) , \text{ where } f \in F \text{ and } \alpha \in \Gamma^* \}$

Language accepted by final state

It is denoted by N(M). It is defined as following:-

$N(M) = \{ x\ !\ (q_o, x, Z_o) \vdash_M^* (p, \epsilon, \epsilon) , \text{ where } p \in Q \}$

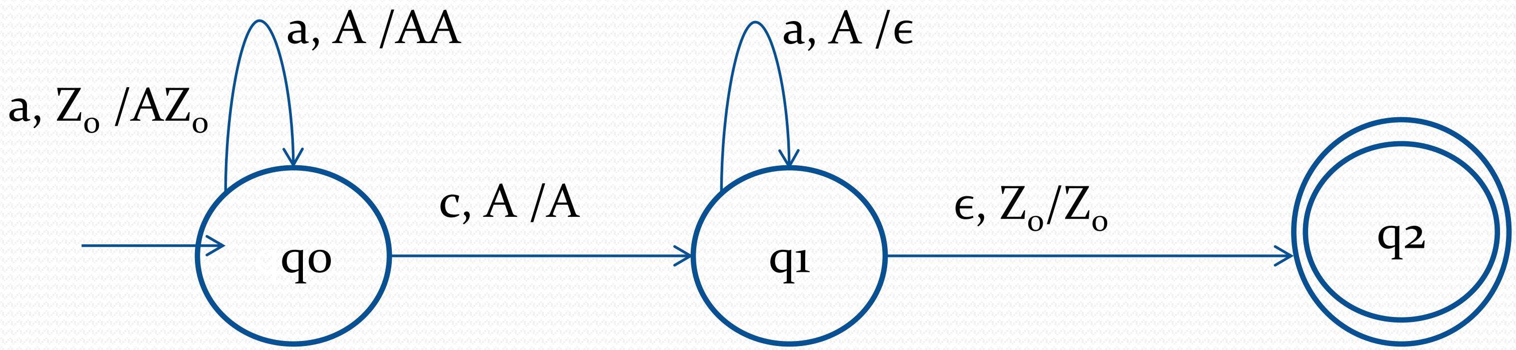
Representation of PDA

We represent PDA by a transition diagram.

Ex. Consider following PDA

$$M=(\{q_0,q_1,q_2\}, \{a, c\}, \{A, Z_0\}, \delta, q_0, Z_0, \{q_2\})$$

Check the acceptability of this string aacaa.



- Edges represent input before / symbol and output after this symbol.

$(q_0, aacaa, Z_0) \vdash (q_0, acaa, AZ_0) \vdash (q_0, caa, AAZ_0) \vdash (q_1, aa, AAZ_0) \vdash (q_1, a, AZ_0) \vdash (q_1, \epsilon, Z_0) \vdash (q_2, \epsilon, Z_0)$

Pushdown Automata

Part-2

Construction of PDA

In this section, we shall see how PDA's can be constructed.

Ex. Construct PDA to accept the language L={ 0^n1^n ! n ≥ 1} by final state.

Solution: First we consider a string of a given language and check how it can accept.

Procedure:

In this language, since n numbers of 0' are followed by n numbers of 1's, therefore, to check equal number of 0 and 1, we have to push a symbol corresponding to 0 and pop that symbol corresponding to 1. Let that symbol is denoted by A.

Push stack symbol A in to the stack as long as scanned input symbol 0. When next scanned input symbol is 1, find top symbol of stack. If top symbol is A, then pop A from stack. When input pointer reaches at the end of string i.e. input string is empty, find top symbol of stack. If top symbol is Z_o, then machine goes to final state. And at this situation, machine accept string.

Ex. L={ 0^n1^n ! n ≥ 1} continue.

Step-1: Let q_o is the initial state and Z_o is the bottom symbol of stack. We will push the stack symbol A into the stack if scanned input symbol 0 appears on the input tape. PDA will stay in this state q_o. The top symbol may be any thing.

The transition rules corresponding to this step are the following:-

$$\delta(q_o, 0, Z_o) = \{(q_o, AZ_o)\}$$
$$\delta(q_o, 0, A) = \{(q_o, AA)\}$$

Step-2: In state q_o, if the next scanned input symbol is 1 and if the top of stack is A, then PDA will pop the top symbol A from the stack and PDA changes its state to q_1.

The transition rule corresponding to this step is the following:-

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

Ex. L={ 0^n1^n ! n ≥ 1} continue.

Step-3: Now PDA is at state q_1. Now the input symbols in input string are 1's only. If current state is q_1, current input symbol is 1 and top symbol is A, then PDA will pop the top symbol A. This action continues till input string becomes empty or top symbol becomes Z_o.

The transition rule corresponding to this step is the following:-

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

Step-4: Now the sate is q_1 and input string is empty(ϵ). If top symbol is Z_o then PDA goes to final state without push or pop. Let the final state is q_2.

The transition rule corresponding to this step is the following:-

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

Ex. L={ $0^n 1^n$! n ≥ 1} continue

Therefore final PDA is

$M = (\{q_0, q_1, q_2\}, \{0, 1\}, \{A, Z_0\}, \delta, q_0, Z_0, \{q_2\})$

δ is defined as following:-

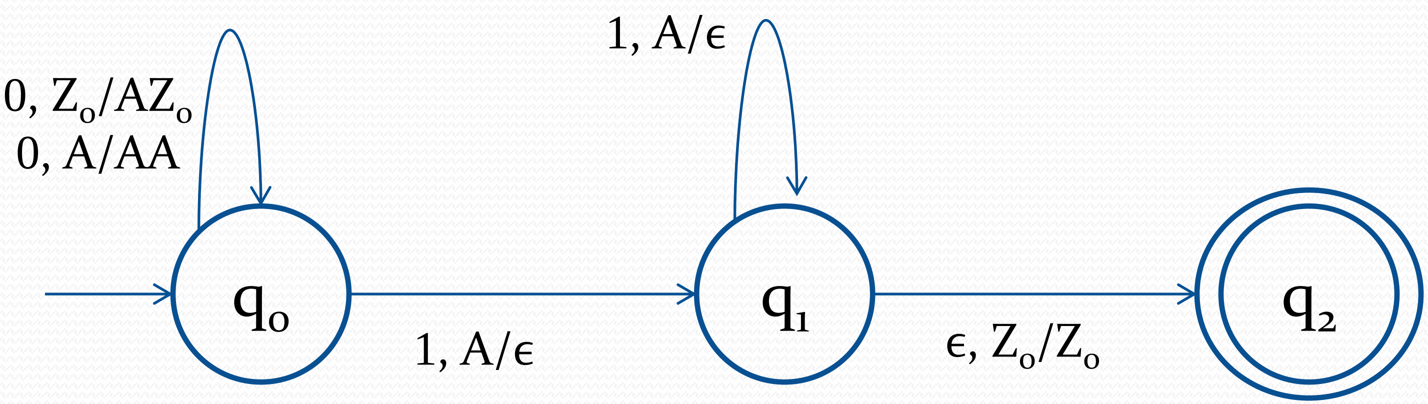
$\delta(q_0, 0, Z_0) = \{(q_0, AZ_0)\}$

$\delta(q_0, 0, A) = \{(q_0, AA)\}$

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

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

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



Processing and Verification of above PDA

Acceptance

Consider string $w = 000111$.

Processing of this string by PDA

$(q_0, 000111, Z_0) \vdash (q_0, 00111, AZ_0) \vdash (q_0, 0111, AAZ_0)$
$\vdash (q_0, 111, AAAZ_0) \vdash (q_1, 11, AAZ_0) \vdash (q_1, 1, AZ_0) \vdash (q_1, \epsilon, Z_0)$
$\vdash (q_2, \epsilon, Z_0)$ (Final configuration)

Rejection

Consider string $w = 00111$.

Processing of this string by PDA

$(q_0, 00111, Z_0) \vdash (q_0, 0111, AZ_0) \vdash (q_0, 111, AAZ_0)$
$\vdash (q_1, 11, AZ_0) \vdash (q_1, 1, Z_0)$ (Non-final configuration)

PDA examples continue

Ex. Construct PDA to accept the language

$L = \{ wcw^R \mid w \in \{a, b\}^* \}$ by final state.

Solution:

In this language, w is any string of a and b. w^R is the reverse string of w.

If w= abb, then string abbcbba ∈ L. Clearly all the strings belong in to L are palindrome.

Some strings belong in to this set are c, aca, bcb, abcba, bacab etc.

Procedure: In this PDA, we push symbol A and B in to the stack corresponding to input symbol a and b in input string. PDA will stay at the q_0. when c appears in input string, it changes its state to other state(Let it be q_1) without push or pop. At q_1 state, it only pop.

- If current input symbol is a and top symbol is A, then pop the top symbol A.
- Similarly, If current input symbol is b and top symbol is B, then pop the top symbol B.
- At last if input string is empty and top symbol is Z_0, then machine goes to final state.

Ex. L={ wcw^R ! $w \in \{a, b\}^*$ } continue

Therefor e the PDA corresponding to above language is constructed as following:-

$M = (\{q_{o,} q_{1,} q_2\}, \{a, b, c\}, \{A, B, Z_o\}, \delta, q_o, Z_{o,} \{q_2\})$

δ is defined as following:-

$\delta(q_o, a, Z_o) = \{(q_o, AZ_o)\}$ $\delta(q_o, b, Z_o) = \{(q_o, BZ_o)\}$

$\delta(q_o, a, A) = \{(q_o, AA)\}$ $\delta(q_o, b, B) = \{(q_o, BB)\}$

$\delta(q_o, a, B) = \{(q_o, AB)\}$ $\delta(q_o, b, A) = \{(q_o, BA)\}$

$\delta(q_o, c, Z_o) = \{(q_1, Z_o)\}$ $\delta(q_o, c, A) = \{(q_1, A)\}$

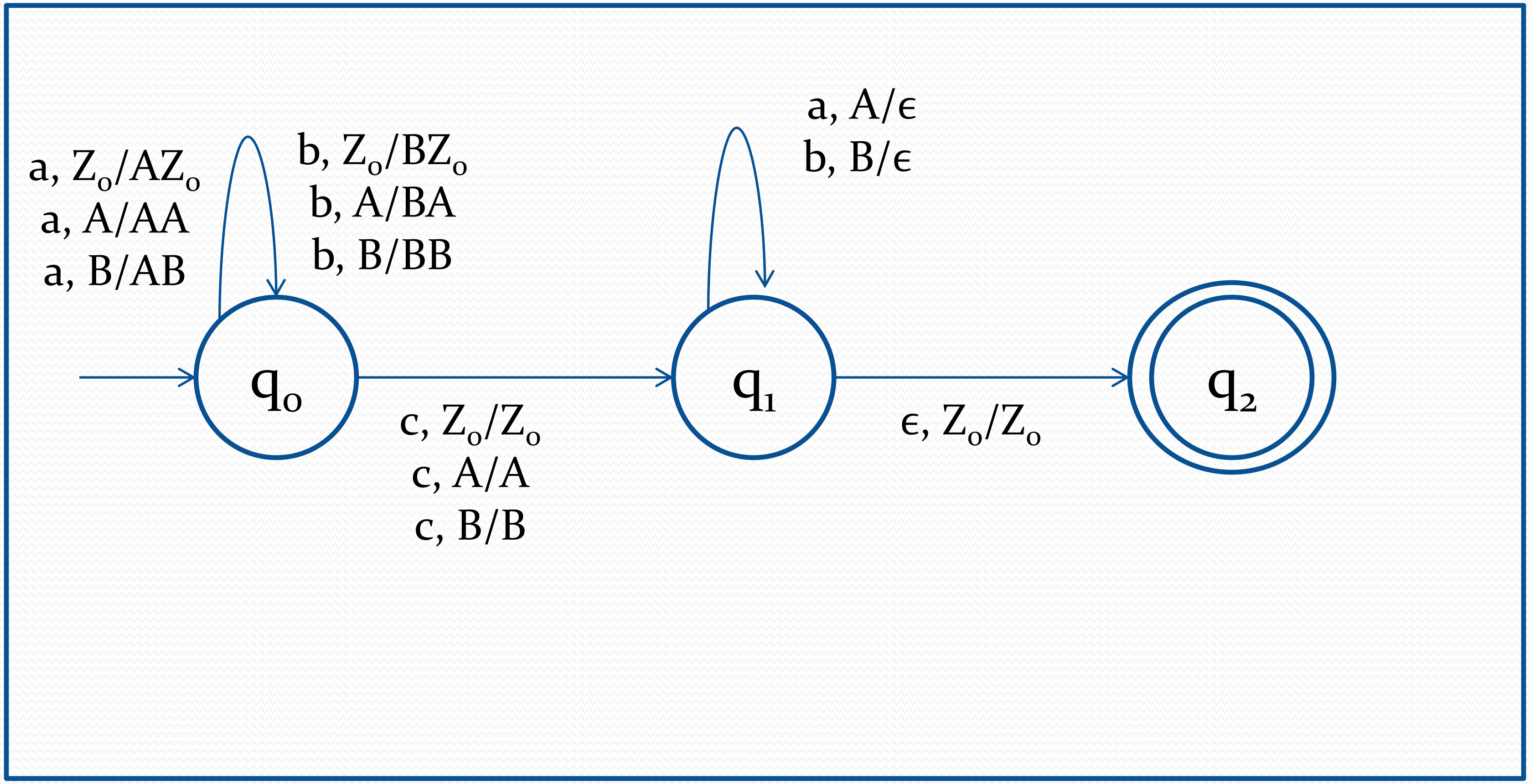
$\delta(q_o, c, B) = \{(q_1, B)\}$

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

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

Ex. L={ wcw^R ! w ∈ {a, b}* } continue

Transition diagram of PDA is the following:-



Processing and Verification of above PDA

<u>Acceptance</u>

Consider string x = abbcbba .

Processing of this string by PDA

$(q_0, abbcbba, Z_0) \vdash (q_0, bbcbba, AZ_0) \vdash (q_0, bcbba, BAZ_0)$
$\vdash (q_0, cbba, BBAZ_0) \vdash (q_1, bba, BBAZ_0) \vdash (q_1, ba, BAZ_0)$
$\vdash (q_1, a, AZ_0) \vdash (q_1, \epsilon, Z_0) \vdash (q_2, \epsilon, Z_0)$ (Final configuration)

<u>Rejection</u>

Consider string x = abbcba .

Processing of this string by PDA

$(q_0, abbcba, Z_0) \vdash (q_0, bbcba, AZ_0) \vdash (q_0, bcba, BAZ_0)$
$\vdash (q_0, cba, BBAZ_0) \vdash (q_1, ba, BBAZ_0) \vdash (q_1, a, BAZ_0)$

(Non-final configuration)

Pushdown Automata

Part-3

PDA examples continue

Ex. Construct PDA to accept the language

$L = \{ ww^R \,!\, w \in \{a, b\}^* \}$ by final state.

Solution:

This question is similar to previous question.

Some strings belong in to this set are ϵ, aa, bb, abba, baab, abbbba, bbaabb etc.

The concept of making PDA of this language is same as previous question. But in this question, to find mid point of string is difficult.

Procedure: In this question, there will be two moves at the same configuration.

When current input symbol is a and top symbol is A or current input symbol is band top symbol is B, then PDA will take one of the following moves:-

1) In the first move, corresponding stack symbol will be pushed(A or B) and state will not change.
2) In the second move, top symbol of stack will be popped and the state will also be changed.

Ex. L={ ww^R ! w ∈ {a, b}* } continue

- Therefor e the PDA corresponding to above language is constructed as following:-

$M = (\{q_o, q_1, q_2\}, \{a, b\}, \{A, B, Z_o\}, \delta, q_o, Z_o, \{q_2\})$

δ is defined as following:-

$\delta(q_o, a, Z_o) = \{(q_o, AZ_o)\}$ $\delta(q_o, b, Z_o) = \{(q_o, BZ_o)\}$

$\delta(q_o, a, A) = \{(q_o, AA), (q_1, \epsilon)\}$ $\delta(q_o, a, B) = \{(q_o, AB)\}$

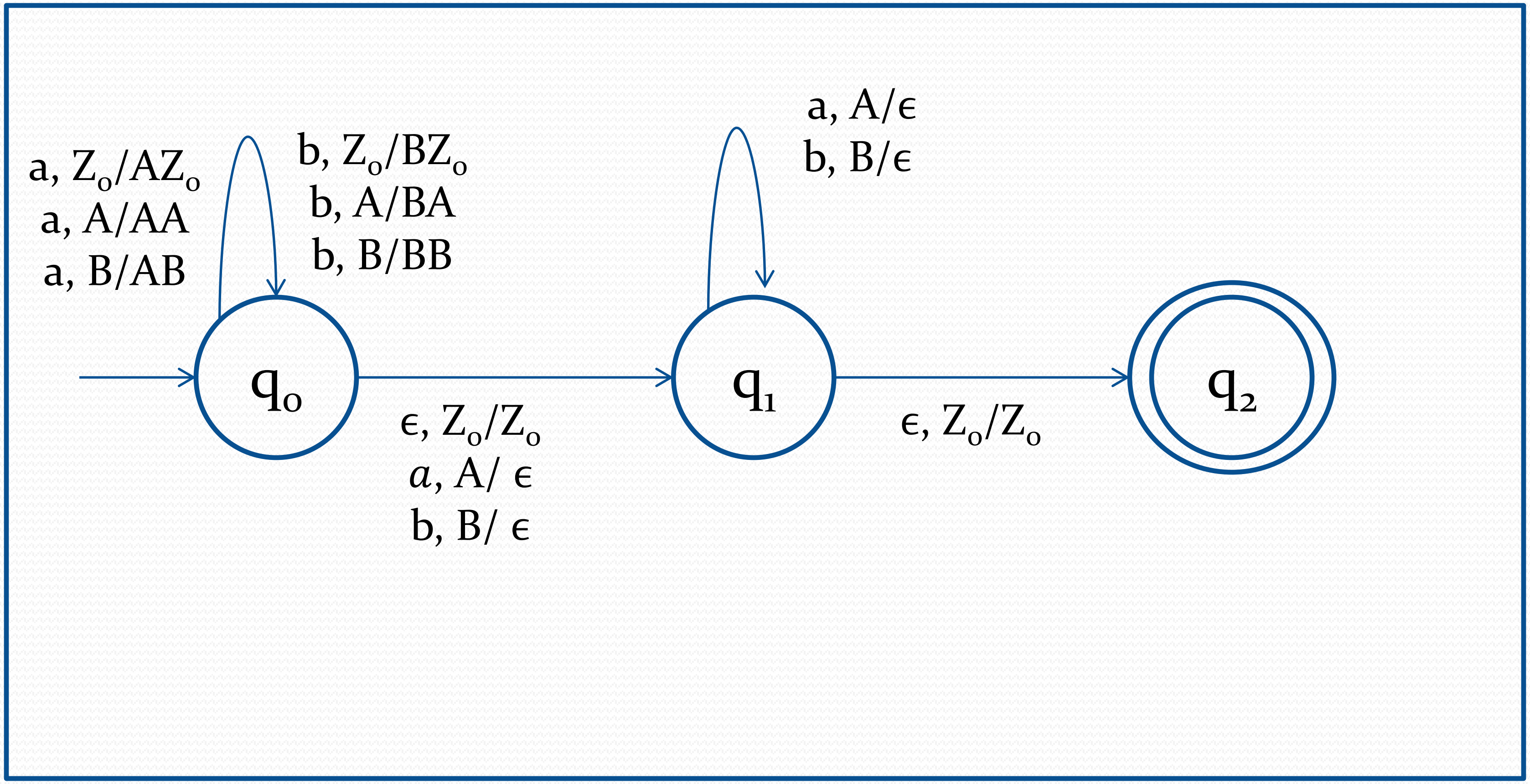
$\delta(q_o, b, B) = \{(q_o, BB), (q_1, \epsilon)\}$ $\delta(q_o, b, A) = \{(q_o, BA)\}$

$\delta(q_o, \epsilon, Z_o) = \{(q_1, Z_o)\}$ $\delta(q_1, a, A) = \{(q_1, \epsilon)\}$

$\delta(q_1, b, B) = \{(q_1, \epsilon)\}$ $\delta(q_1, \epsilon, Z_o) = \{(q_2, Z_o)\}$

Ex. L={ ww^R ! w ∈ {a, b}* } continue

Transition diagram of PDA is the following:-



Processing and Verification of above PDA

Acceptance

Consider string x = abbbba .

Processing of this string by PDA

$(q_0, abbbba, Z_0) \vdash (q_0, bbbba, AZ_0) \vdash (q_0, bbba, BAZ_0)$
$\vdash (q_0, bba, BBAZ_0) \vdash (q_1, ba, BAZ_0) \vdash (q_1, a, AZ_0) \vdash (q_1, \epsilon, Z_0)$
$\vdash (q_2, \epsilon, Z_0)$ (Final configuration)

Rejection

Consider string x = abbba .

Processing of this string by PDA

$(q_0, abbba, Z_0) \vdash (q_0, bbba, AZ_0) \vdash (q_0, bba, BAZ_0) \vdash (q_0, ba, BBAZ_0) \vdash (q_1, a, BAZ_0)$ (Non-final configuration)

Some questions

Construct PDA to accept the following languages:-

1) $L = \{ a^n b^{2n} ! n \geq 1 \}$
2) $L = \{ a^n b^{3n} ! n \geq 1 \}$
3) $L = \{ a^m b^n c^n d^m ! m, n \geq 1 \}$
4) $L = \{ a^n b^m c^n ! m, n \geq 1 \}$
5) $L = \{a^i b^j c^k ! i = j \text{ or } j = k\}$

Ex. Construct PDA to accept the language

$L = \{ a^n b^{2n} ! n \geq 1 \}$ by empty stack.

Solution:

In this question, number of b is two times of number of a. Therefore, the PDA should read two b corresponding to one a.

In this question, when a appears in input string, then push the stack symbol A in to the stack.

When b appears in input string, then machine change its state. When second b appears input string, then we change sate and pop the top symbol of stack.

In this question, PDA will pop a top symbol from stack when bb(i.e two b) appears in the input string.

$L = \{ a^n b^{2n} ! n \geq 1 \}$

Therefor e the PDA corresponding to above language is constructed as following:-

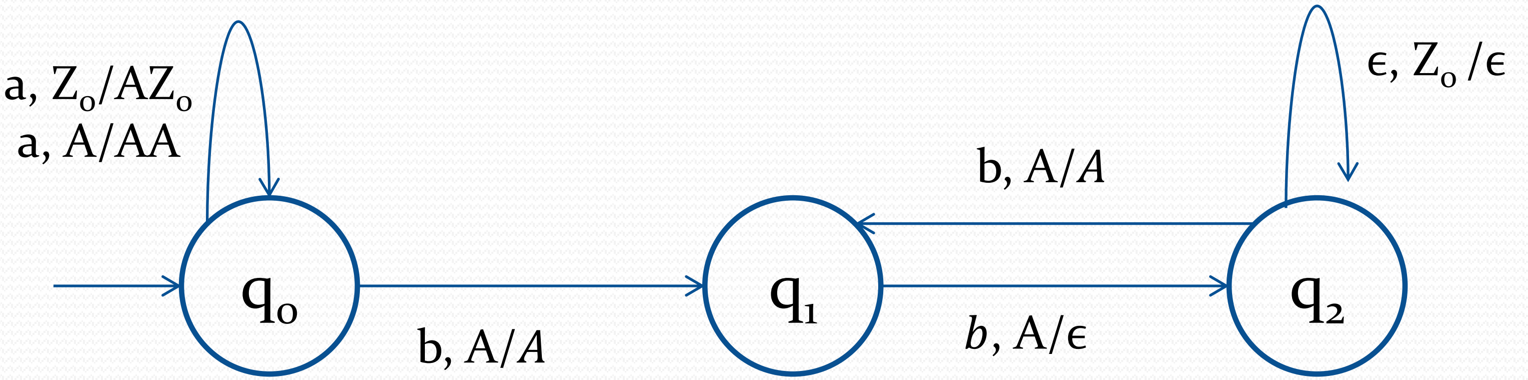
$M = (\{q_o, q_1, q_2\}, \{a, b\}, \{A, Z_o\}, \delta, q_o, Z_o, \phi)$

δ is defined as following:-

$\delta(q_o, a, Z_o) = \{(q_o, AZ_o)\}$ $\qquad$ $\delta(q_o, a, A) = \{(q_o, AA)\}$

$\delta(q_o, b, A) = \{(q_1, A)\}$ $\qquad$ $\delta(q_1, b, A) = \{(q_2, \epsilon)\}$

$\delta(q_2, b, A) = \{(q_1, A)\}$ $\qquad$ $\delta(q_2, b, A) = \{(q_2, \epsilon)\}$



Ex. Construct PDA to accept the language

$L = \{ a^m b^n c^n d^m ! m, n \geq 1 \}$ by empty stack.

Soultion:

Some strings belong in to this set are abcd, abbccd, aabcdd, aaabbccddd etc.

In this language, number of a and d are equal and number of b and c are equal.

Therefore, we have to push stack symbol corresponding to a and pop that stack symbol corresponding to d.

Similarly, we have to push another stack symbol corresponding to b and pop that stack symbol corresponding to c.

To preserve the order of a, b, c and d, machine changes its state when move from a to b, b to c, and c to d.

Therefor e the PDA corresponding to above language is constructed as following:-

$M = (\{q_o, q_1, q_2, q_3\}, \{a, b, c, d\}, \{A, B, Z_o\}, \delta, q_o, Z_o, \phi)$

δ is defined as following:-

$\delta(q_o, a, Z_o) = \{(q_o, AZ_o)\}$

$\delta(q_o, a, A) = \{(q_o, AA)\}$

$\delta(q_o, b, A) = \{(q_1, BA)\}$

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

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

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

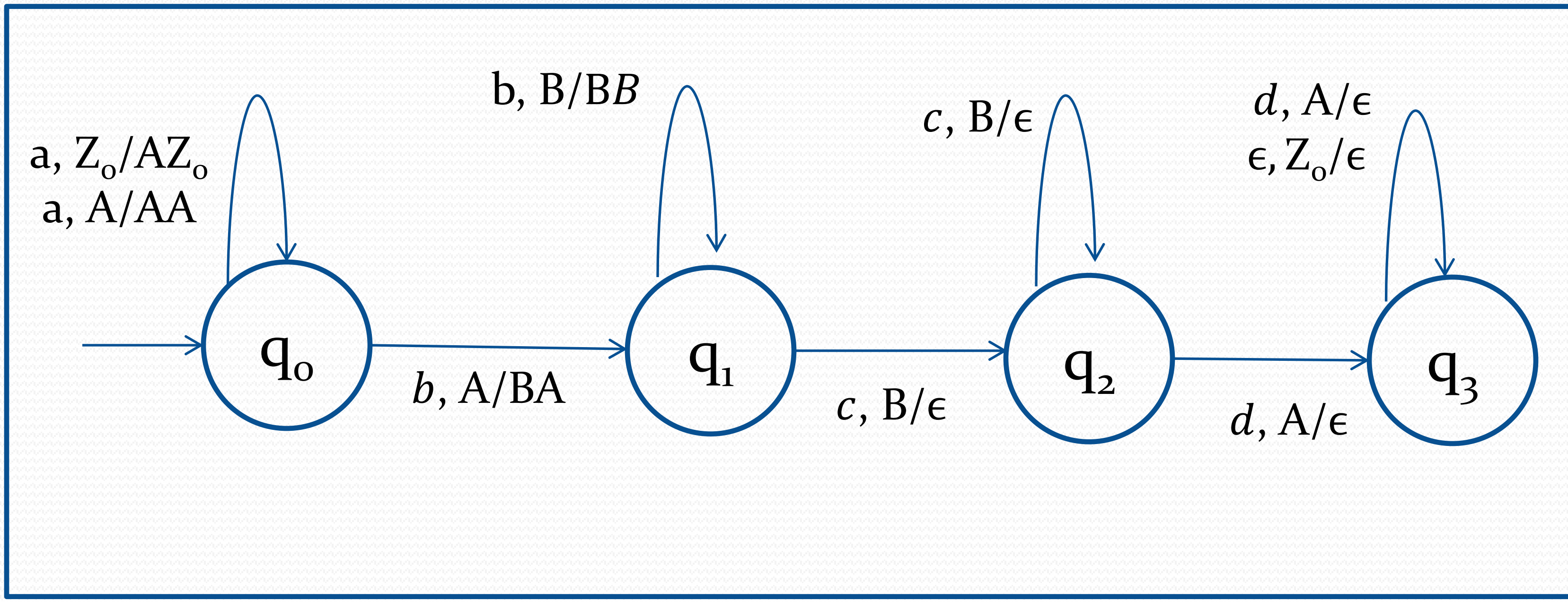
$\delta(q_2, d, A) = \{(q_3, \epsilon)\}$

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

$\delta(q_3, \epsilon, Z_o) = \{(q_3, \epsilon)\}$

$L = \{ a^m b^n c^n d^m \; ! \; m, n \geq 1 \}$

Transition diagram of PDA is the following:-



Processing and Verification of above PDA

Acceptance

Consider string $x = aaabbccddd$.

Processing of this string by PDA

$(q_0, aaabbccddd, Z_0) \vdash (q_0, aabbccddd, AZ_0) \vdash (q_0, abbccddd, AAZ_0)$
$\vdash (q_0, bbccddd, AAAZ_0) \vdash (q_1, bccddd, BAAAZ_0) \vdash (q_1, ccddd, BBAAAZ_0)$
$\vdash (q_2, cddd, BAAAZ_0) \vdash (q_2, ddd, AAAZ_0) \vdash (q_3, dd, AAZ_0) \vdash (q_3, d, AZ_0)$
$\vdash (q_3, \epsilon , Z_0) \vdash (q_3, \epsilon, \epsilon)$ (Final configuration)

Rejection

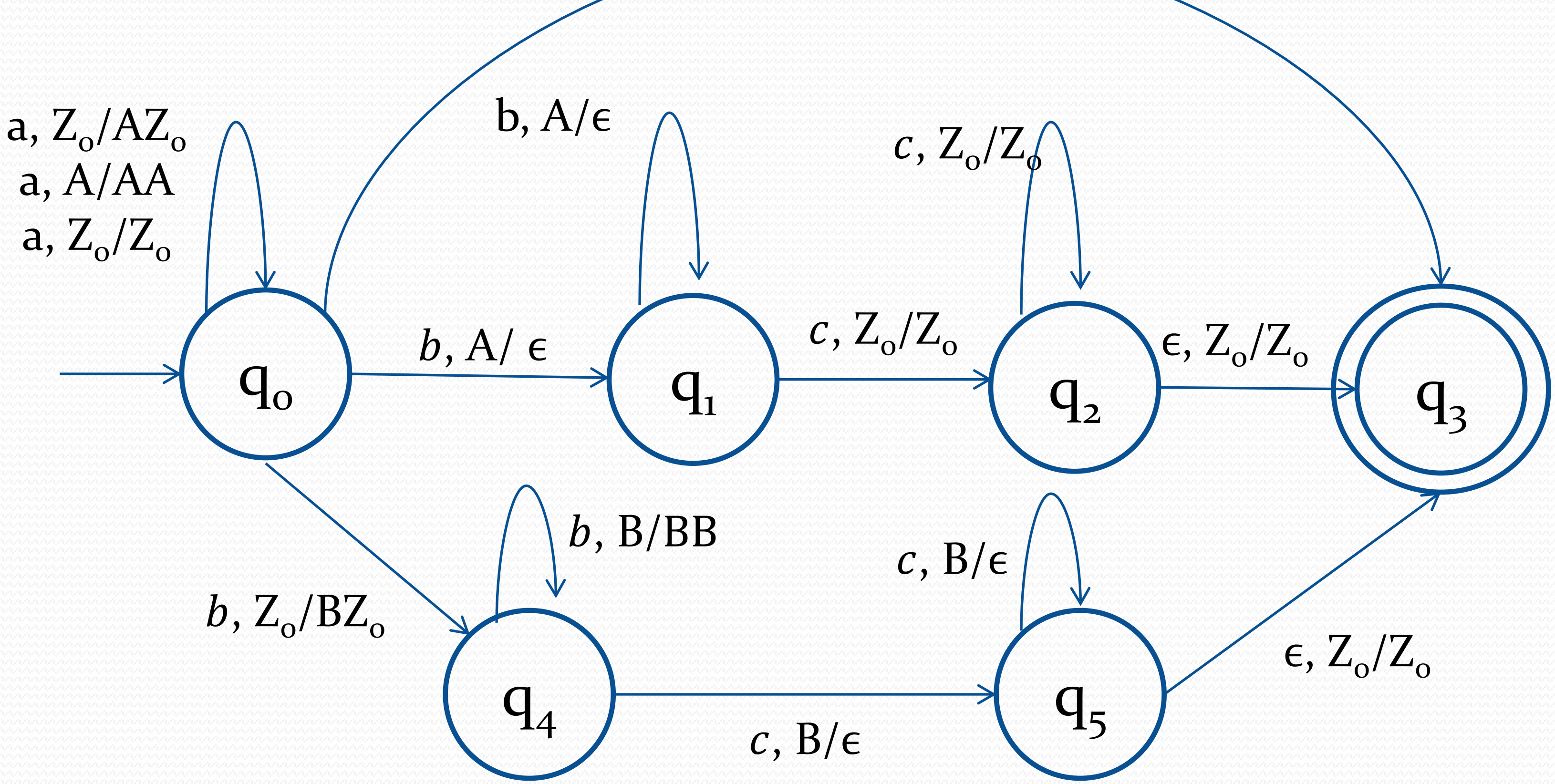
Consider string $x = abbcd$.

Processing of this string by PDA

$(q_0, abbcd, Z_0) \vdash (q_0, bbcd, AZ_0) \vdash (q_1, bcd, BAZ_0) \vdash (q_1, cd, BBAZ_0)$
$\vdash (q_2, d, BAZ_0)$ (Non-final configuration)

Ex. $L = \{a^i b^j c^k \,!\, i = j \text{ or } j = k\}$

The PDA corresponding to this language is the following:-



PDA examples continue

Construct PDA to accept the following languages:-

1) $L = \{ w \mid w \in \{a,b\}^* \text{ and } n_a(w) = n_b(w) \}$
2) $L = \{ w \mid w \in \{a,b\}^* \text{ and } n_a(w) \geq n_b(w) \}$
3) $L = \{ w \mid w \in \{a,b\}^* \text{ and } n_a(w) \neq n_b(w) \}$

Ex. L = { w ! w ∈ {a,b}* and $n_a(w) = n_b(w)$ }

Solution:

Some strings of this set are ϵ, ab, ba, aabb, bbaa, abab, baba etc.

Procedure:

In this question, when first symbol either a or b is current input symbol then push the stack symbol A or B respectively.

For remaining input symbols, we push if same type of symbols occurs as input or at the top. If different type of symbol occurs on input and on stack, then pop from stack.

PDA for this language is

$M = (\{q_o, q_1\}, \{a, b\}, \{A, B, Z_o\}, \delta, q_o, Z_o, \{q_1\})$

PDA examples continue

δ is defined as following:-

$\delta(q_o, a, Z_o) = \{(q_o, AZ_o)\}$

$\delta(q_o, a, A) = \{(q_o, AA)\}$

$\delta(q_0, a, B) = \{(q_0, \epsilon)\}$

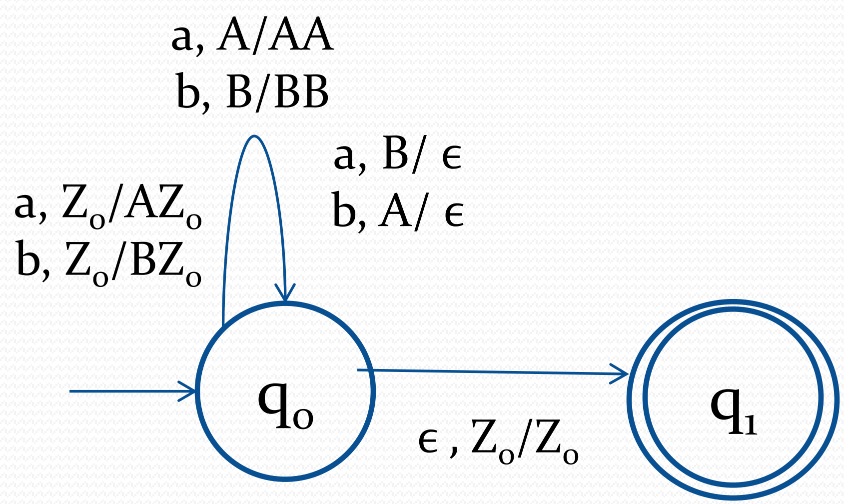
$\delta(q_o, \epsilon, Z_o) = \{(q_1, Z_o)\}$

$\delta(q_o, b, Z_o) = \{(q_o, BZ_o)\}$

$\delta(q_o, b, B) = \{(q_1, BB)\}$

$\delta(q_0, b, A) = \{(q_2, \epsilon)\}$

Transition diagram of PDA for above language is



Processing and Verification of above PDA

Acceptance

Consider string x = aaabbabb.

Processing of this string by PDA

$(q_0, aaabbabb, Z_0) \vdash (q_0, aabbabb, AAZ_0) \vdash (q_0, abbabb, AAZ_0) \vdash (q_0, bbabb, AAAZ_0) \vdash (q_0, babb, AAZ_0) \vdash (q_0, abb, AZ_0) \vdash (q_0, bb, AAZ_0) \vdash (q_0, b, AZ_0) \vdash (q_0, \epsilon, Z_0) \vdash (q_1, \epsilon, Z_0)$ (Final configuration)

Rejection

Consider string x = abbab .

Processing of this string by PDA

$(q_0, abbab, Z_0) \vdash (q_0, bbab, AZ_0) \vdash (q_0, bab, Z_0) \vdash (q_0, ab, BZ_0) \vdash (q_0, b, Z_0) \vdash (q_0, \epsilon, BZ_0)$ (Non-final configuration)

Deterministic Pushdown Automta(DPDA)

A pushdown automata $M=(Q, \Sigma, \Gamma, \delta, q_o, Z_o, F)$ is said to be deterministic PDA if it satisfies the following properties:-

1) $\delta(q, a, Z)$ contains at most one element for any $q \in Q, a \in \Sigma \cup \{\epsilon\}$ and $Z \in \Gamma$.
2) For any $q \in Q$ and $Z \in \Gamma$, if $\delta(q, \epsilon, Z) \neq \phi$ then $\delta(q, a, Z) = \phi$ for every $a \in \Sigma$.

Equivalence of PDA and CFG

Construction of PDA from CFG

Suppose the given context free grammar is $G=(V, \Sigma, S, P)$.

Step-1: Convert Grammar into GNF if it is not.

Step-2: If G is in GNF, the use following procedure:

The PDA equivalent to G is constructed as following:-

$$M= (\{q\}, \Sigma, (V \cup \Sigma), \delta, q, S, \phi)$$

Transition function δ is defined by the following two types of rule:-

1) For each production rule $A \rightarrow \alpha$, make the following rule $(q, \alpha) \in \delta(q, \epsilon, A)$.
2) Make the following types of rule corresponding to each input symbol a.

$\delta(q, a, a) = \{(q, \epsilon)\}$ for every $a \in \Sigma$.

PDA from CFG

Ex. For the following grammar, find an equivalent PDA.

S->aABC, A→aB/a, B→bA/b, C→a

Solution:

Since this grammar is already in Greibach normal form, therefore first step is completed.

Now PDA corresponding to this grammar is constructed as the following:-

M = ({q}, {a, b}, {S,A,B,C,a,b}, δ , q, S, ϕ)

And δ is constructed as following:-

According to first rule

$\delta(q, \epsilon, S) = \{(q, aABC)\}$, $\delta(q, \epsilon, A) = \{(q, aB), (q, a)\}$

$\delta(q, \epsilon, B) = \{(q, bA), (q, b)\}$ $\delta(q, \epsilon, C) = \{(q, a)\}$

According to second rule

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

Check the acceptability of this string aabba by above PDA.

Solution:

$\delta(q, \epsilon, S) = \{(q, aABC)\}$, $\delta(q, \epsilon, A)= \{(q, aB), (q, a)\}$

$\delta(q, \epsilon, B)= \{(q, bA), (q, b)\}$ $\delta(q, \epsilon, C)= \{(q, a)\}$

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

$(q, aabba, S) \vdash (q, aabba, aABC) \vdash (q, abba, ABC) \vdash (q, abba, aBBC) \vdash (q, bba, BBC) \vdash (q, bba, bBC) \vdash (q, ba, BC) \vdash (q, ba, bC) \vdash (q, a, C) \vdash (q, a, a) \vdash (q, \epsilon, \epsilon)$ (Final Configuration)

Therefore, this string is accepted by this PDA.

Ex. Construct PDA equivalent to the following CFG,

$S \rightarrow 0BB$, $B \rightarrow 0S/1S/0$

And check whether 010000 is in N(M) or not.

Construction of CFG from given PDA

Procedure:

Suppose the given PDA is $M = (Q, \Sigma, \Gamma, \delta, q_o, Z_o, \phi)$.

The context free grammar equivalent to this PDA is constructed as following:-

$G = (V, \Sigma, S, P)$, Where

$V = \{S\} \cup \{ [p, Z, q] \,!\, p, q \in Q \text{ and } Z \in \Gamma \}$

And P is defined by following three types of rules:-

1) Add $S \rightarrow [q_0, Z_0, p]$ into P, for every $p \in Q$.
2) If $(q, \epsilon) \in \delta(p, a, Z)$ then add $[p, Z, q] \rightarrow a$ into P for every p, q $\in Q$, $a \in (\Sigma \cup \{\epsilon\})$ and $Z \in \Gamma$.
3) If $(q, A_1A_2A_3 \ldots \ldots An) \in \delta(p, a, Z)$ then add

$$[p, Z, q_1] \rightarrow a[q, A_1, q_2][q_2, A_2, q_3]\ldots\ldots\ldots\ldots[q_n, A_n, q_1]$$

into P for every p, $q_i \in Q$, $a \in (\Sigma \cup \{\epsilon\})$ and Z, $Ai \in \Gamma$.

Where $1 \le i \le n$.

Ex. Construct CFG equivalent to the following PDA:-

$$M = (\{q_o, q_1\}, \{a, b\}, \{A, Z_o\}, \delta, q_o, Z_o, \phi)$$

And δ is defined as following:-

$\delta(q_o, a, Z_o) = \{(q_o, AZ_o)\}$ $\delta(q_o, a, A) = \{(q_o, AA)\}$

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

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

Solution:

The equivalent CFG is constructed as

$$G = (V, \Sigma, S, P), \text{ Where}$$

$V = \{ S, [q_0, Z_0, q_0], [q_0, Z_0, q_1], [q_0, A, q_0], [q_0, A, q_1],$
$[q_1, Z_0, q_0], [q_1, Z_0, q_1], [q_1, A, q_0], [q_1, A, q_1] \}$

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

And P is determined as following:-

Type-1: $S \rightarrow [q_0, Z_0, q_0] / [q_0, Z_0, q_1]$

Type-2: In this type, we consider only the transition rules which pop the symbols.

Consider rule, $\delta(q_1, a, A) = \{(q_1, \epsilon)\}$

The production rule for it will be

$$[q_1, A, q_1] \rightarrow a$$

Similarly, for $\delta(q_1, \epsilon, Z_0) = \{(q_1, \epsilon)\}$

The production will be

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

Type-3: In this type, we consider only the transition rules which push the symbols or no push and no pop operation.

Consider rule, $\delta(q_0, a, Z_0) = \{(q_0, AZ_0)\}$
The production rule for it will be

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

Consider rule, $\delta(q_0, b, A) = \{(q_1, A)\}$
The production rule for it will be

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

Consider rule, $\delta(q_o, a, A) = \{(q_o, AA)\}$

The production rule for it will be

$[q_o, A, q_o] \rightarrow a\,[q_o, A, q_o]\,[q_o, A, q_o]$

$[q_o, A, q_o] \rightarrow a\,[q_o, A, q_1]\,[q_1, A, q_o]$

$[q_o, A, q_1] \rightarrow a\,[q_o, A, q_o]\,[q_o, A, q_1]$

$[q_o, A, q_1] \rightarrow a\,[q_o, A, q_1]\,[q_1, A, q_1]$

Consider rule, $\delta(q_1, b, A) = \{(q_1, A)\}$

The production rule for it will be

$[q_1, A, q_o] \rightarrow b\,[q_1, A, q_o]$

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

Ex. Construct CFG equivalent to the following PDA:-

$M = (\{q_0, q_1\}, \{a, b\}, \{A, Z_0\}, \delta, q_0, Z_0, \phi)$

And δ is defined as following:-

$\delta(q_0, a, Z_0) = \{(q_0, AZ_0)\}$

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

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

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

$\delta(q_1, \epsilon, A) = \{(q_1, \epsilon)\}$

$\delta(q_1, \epsilon, Z_0) = \{(q_1, \epsilon)\}$

Two Stack PDA(2PDA)

Two stack pushdown automata is described by a 7-tuple $M=(Q, \Sigma, \Gamma, \delta, q_o, Z_o, F)$ where,

- Q is the finite set of states,
- Σ is the set of input symbols
- Γ is the set of stack symbols,
- $q_o \in Q$ is the initial state,
- $Z_o \in \Gamma$ is a bottom symbol of stack
- F is the set of final states, and
- δ is a transition function which is defined as following:-
- $\delta: Q\times(\Sigma\cup\{\epsilon\})\times\Gamma\times\Gamma \rightarrow$ finite subset of $Q\times\Gamma^*\times\Gamma^*$

Ex. Construct 2-stack PDA for the following language $L = \{ a^n b^n c^n \mid n \geq 1\}$.

Solution:

In this set, some strings are abc, $a^2b^2c^2$, $a^3b^3c^3$ etc.

Clearly, this set contains all the strings a, b and c, in which number of a, b and c are equal. And the order of a ,b and c are also fixed.

Procedure: In this PDA, we have to push the symbol A into first stack when a appears in input string.

When first b appears then we have to push symbol A into second stack and also change the state. For remaining b, we have to push symbol A into second stack at that state.

When first c appears then we check the top symbols of both stack. If both top symbols are A then we pop the top symbols from both stack. For remaining c, same operation is applied.

When string becomes empty, we check both stack. If top symbols of both stack are Zo, then string will be accepted.

$L = \{ a^n b^n c^n \,!\, n \geq 1\}$

Therefore the 2PDA for this language will be

$M = (\{q_{o,}\ q_1, q2,\}, \{a, b, c\}, \{A, Z_o\ \}, \delta, q_o, Z_{o,}\ \phi)$

And δ is defined as following:-

$\delta(q_o, a, Z_o, Z_o) = \{(q_o, AZ_{o,}\ , Z_o)\}$

$\delta(q_o, a, A, Z_o) = \{(q_o, AA_{,}\ , Z_o)\}$

$\delta(q_o, b, A, Z_o) = \{(q_1, A_{,}\ , AZ_o)\}$

$\delta(q_1, b, A, A) = \{(q_1, A_{,}\ , AA)\}$

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

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

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

$L = \{ a^n b^n c^n ! n \geq 1\}$

Transition diagram this 2PDA is