

**Name: Ameer Ayman Ahmed**

**ID: 135058**

Pushdown automata is a finite automaton with extra memory called stack which help the machine to recognize context free languages. In this PDA machine we can determine only symbol input to the state which the machine will move to it and if pushing or popping symbol from the stack. We suppose to design machine that accepts this language {0^𝑛1^2𝑛 |𝑛 ≥ 0 }

We can represent the PDA machine or its transition table by state diagram:

* The nodes represent each state.
* The arcs labelled with the input of the language, the popping symbol from the stack and pushing symbol into the stack that represent the transition for the next state.
* The initial sate with input arc that the machine starts from it.
* The final state or the accept state that indicated with two circles.

This language {0^𝑛1^2𝑛 |𝑛 ≥ 0 } to be accepted must have:

* Any number from zeros must have the doubled number from ones.
* Or it can have no inputs.

The state diagram for this machine as follow:

0, £ 0

£, £ $ 1,0 1 1,1 £

q1 q2 q3 £, $ £

1,0 1

This state diagram implemented in java code. There is a class called Nodes that has linked list each node of the linked list represents one state. Each node of the linked list labeled with string name to define this node.

public class Nodes {

// initializing variable member of the class

//this variable holds the current state of the linked list

private String currentState;

//set function to access the current state variable and store the value by it

public String getCurrentState() {

return currentState;

}

//get function to access the current state variable and retrive the value from the variable

public void setCurrentState(String currentState) {

this.currentState = currentState;

}

// to check if the current state is the accept state or not

public void validation(String state)

{

if("acceptState".equals(state))

System.out.println("the string is accepted");

else

System.out.println("the string is rejected");

}

// class that has the linked list

public void linkedNodes() {

state = new LinkedList();

state.add("q1"); // start state of the list

state.add("q2");

state.add("q3");

state.add("acceptState"); /// accept state

state.add("rejectState");

}

We can represent this state diagram by this transition and each transition consists of

Input pop from push to

Symbol stack stack

0 , £ 0

And this the transition table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| current state | input | pop | push | next state |
| q0 | no | no | $ | Q1 |
| q1 | 0 | no | 0 | 0 Q1  1 Q2 |
| q2 | 1 | 1 | no | Q3 |
| q3 | no  1 | $  0 | no  1 | accept  Q2 |