AUTOMATA FORMAL LANGUAGES AND LOGIC

Lecture notes on Pushdown Automata

Prepared by:
Prof.Divya S J
Assistant Professor

Department of Computer Science & Engineering

PES UNIVERSITY
(Established under Karnataka Act No.16 of 2013)
100-ft Ring Road, BSK III Stage, Bangalore – 560 085

Table of contents

Examples Solved

11	Construct a PDA for $L=\{a^n b^{2n+1} \mid n \geq 1\}$	**11**
12	Construct a PDA for the language $L=\{wcw^R \mid w \in \{a,b\}\}$	**12**

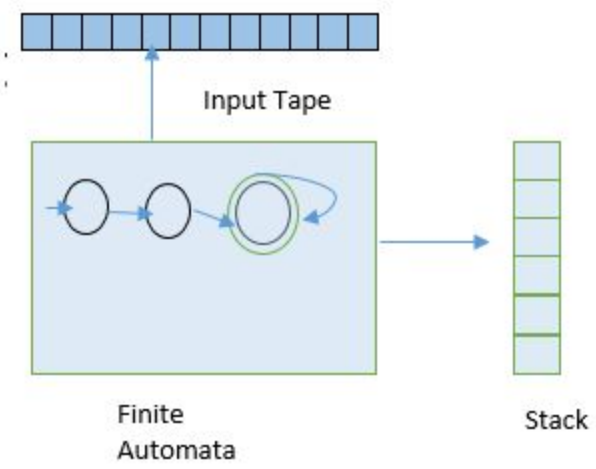
Pushdown Automata

Context Free Languages(CFL) is the next class of languages outside of Regular Languages:
– Means for defining: Context Free Grammar
– Machine for accepting: Pushdown Automata

Finite automata were unable to handle CFLs primarily because they did not have sufficient memory to remember parts of the input.For example such as $a^n b^n$,the automaton needs to count an unlimited number of symbols and remember the count to be able to match it to the count of symbols in later parts of input string.
In order to accept CFL we need a machine which is similar to Finite Automata and it is called a Pushdown Automata.

Definition of PDA
A PDA is defined by the septuple $M=(Q, \Sigma, \delta, q_0, Z_0, F, \Gamma)$ where
Q is a finite set of internal states of the control unit,
Σ is the input alphabet,
Γ is a finite set of symbols called the stack alphabet,
$\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow$ set of finite subsets of $Q \times \Gamma^*$ the transition function,
$q_0 \in Q$ is the initial state of the control unit
$Z_0 \in \Gamma$ is the stack start symbol,
$F \subseteq Q$ is the set of final states.

The argument of δ are the current state of the control unit, the current input symbol, and the current symbol on top of the stack. The result is a set of pairs (q,x) where q is the next state of the control unit and x is a string that is put on top of the stack in place of a single symbol.



A schematic representation of a pushdown automata is shown in the above figure. Each move of the control unit reads a symbol from the input file, while at the same time changing the contents of the stack through the usual stack operations. Each move of the control unit is determined by the current input symbol as well as by the symbol currently on top of the stack. The result of the move is a new state of the control unit and a change in the top of the stack.

Example: – δ (q, a, a) = (p, aa) ,(q, λ)

Meaning: – When in state q,

– Reading in an a from the tape

– With an a popped off the stack

1. The machine will – Go into state p

   – Push the string "aa" onto the stack
2. The machine will stay in state q with the symbol a pops off the stack.

In our notation we assume that the insertion of a string into a stack is done symbol by symbol, starting at the right end of the string.

The only operations that a PDA is allowed to perform are as follows:

1. Start with q_0 as the current state and Z_0 as the only symbol on the stack.
2. Examine the next input symbol reading it from left to right in a single pass.
3. Examine the current symbol at the top of the stack.
4. Based on the input symbol, the current state and the stack symbol, find one or more matching transitions.

5. Execute the transition by:
    a. Consuming the input symbol
    b. Moving to the new state as specified by the transition function.
    c. Popping the symbol on the top of the stack if so specified by the transition function
    d. Pushing the symbol specified in the transition (if any) onto the stack.

When the input symbols are exhausted, if the machine is in a final state and if the stack is empty i.e., if the symbol at the top of the stack is Z_0, then we say the machine has accepted the input string.Otherwise, the machine rejects the input string.

Configuration of a PDA – Gives the current "configuration" of the machine

– (p, x, α) where

- p is the current state
- x is string indicating what remains to be read on the tape
- α is the current contents of the stack.

Move of a PDA: – We can describe a single move of a PDA:

• $(q, x, \alpha) \vdash (p, y, \beta)$

• If:

– $x = ay, \alpha = \gamma X, \beta = YX$

» And

– $\delta (q, x, \gamma)$ includes (p, Y) or

– $\delta (q, \lambda, \gamma)$ includes (p, Y) and x = y

Moves of a PDA – We can write:

• $(q, x, \alpha) \vdash^* (p, y, \beta)$

• If – You can get from one configuration to the other by applying 0 or more moves.

There are two types of PDAs

1. Deterministic Pushdown Automata
2. Non Deterministic Pushdown Automata

Deterministic Pushdown Automata

The transition $\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow Q \times \Gamma^*$

Some examples of DPDA are as follows.

- $L = \{a^n b^n \mid n \geq 1\}$

- $L = \{n_a(w) = n_b(w) \text{ where } w \in \{a,b\}\}$
- $L = \{a^n b^n c^m | n,m \geq 1\}$
- $L = \{a^{m+n} b^m c^n | n,m \geq 1\}$
- $L = \{a^n b^{2n+1} | n \geq 1\}$
- $L = \{a^n b^n c^m d^m | n,m \geq 1\}$
- $L = \{wcw^R | w \in \{a,b\}\}$

Non Deterministic Pushdown Automata

The transition $\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$

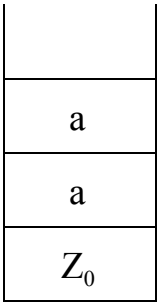
Examples of NPDA are:

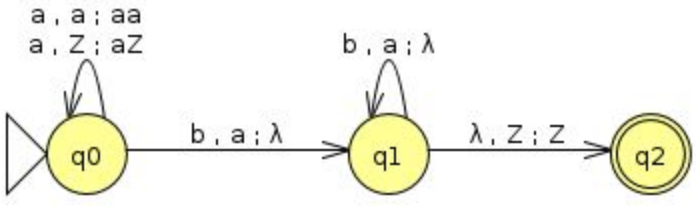
- $L = \{ww^R | w \in \{a,b\}\}$
- $L = \{a^n b^n | n \geq 1\} \cup \{a^n b^{2n} | n \geq 1\}$
- $L = \{a^i b^j c^k d^l | i=k \text{ or } j=l\}$

Construction of PDA

1. Language $L = \{a^n b^n | n \geq 1\}$.

   Solution : Consider the simplest string which belongs to the language L i.e., aabb.





In state q_0, the automaton consumes all the a symbols and counts them (or remembers them) by pushing a onto the stack. It changes state to q_1 upon seeing the first b where it consumes the remaining bs while at the same time popping off a from the stack for every b.Thus the machine matches every b with a corresponding a which was seen before. At the end, when the input is empty and the stack is empty denoted by the transitions (λ ,Z;Z) , it reaches the final state q_2 where the input is accepted.

In terms of transition function of PDA can be written as:

$\delta (q_0,a,Z) = (q_0,aZ)$

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

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

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

$\delta (q_1, \lambda ,Z) = (q_2,Z)$

Configurations of PDA on string aabb

$\delta (q_0,aabb,Z) \vdash \delta (q_0,abb,aZ)$ //push a

$\vdash \delta (q_0,bb,aaZ)$ //push a

$\vdash \delta (q_1,b,aZ)$ //pop a

$\vdash \delta (q_1, \lambda ,Z)$ //pop a

$\vdash \delta (q_2, \lambda ,Z)$ //Accept!

Configurations of PDA on string abab

$\delta (q_0,abab,Z) \vdash \delta (q_0,bab,aZ)$ //push a

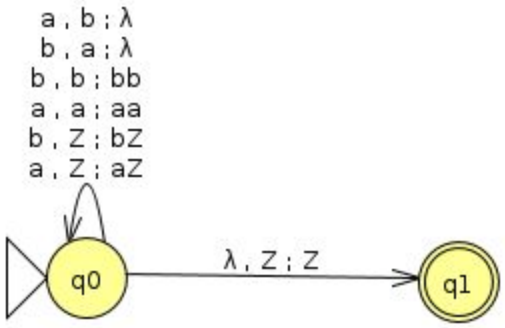
$\vdash \delta (q_1,ab,Z)$ //pop a

Now where to go //Reject!

It is a dead configuration. PDA will halt in state q_1 and it is rejected.

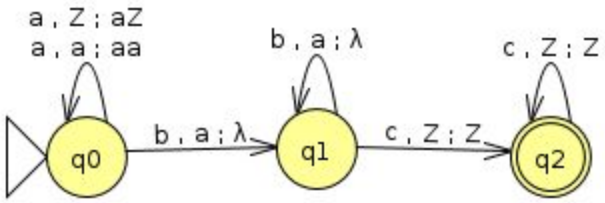
2. $L=(n_a(w)=n_b(w))$ where $w \in \{a,b\}$

   Solution: a's and b's can appear in any order . The language L ={ab,ba,abab,baab,.....}



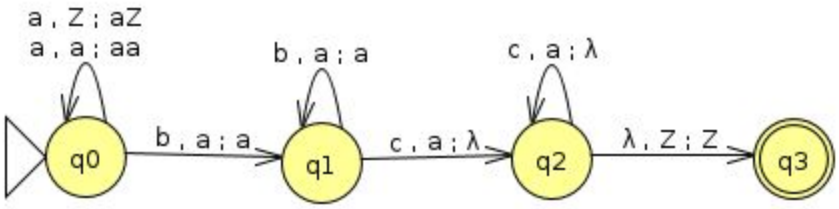
3. $L=\{a^n b^n c^m | n,m \geq 1\}$

   Solution: The language says that a's and b's must be equal,there can be any number of c's later.



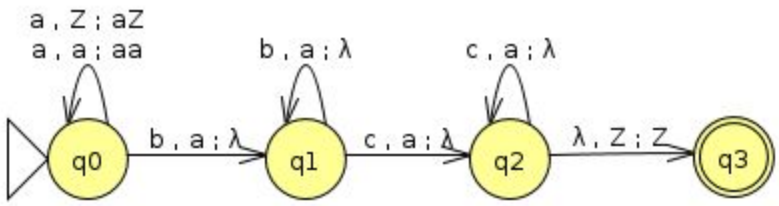
4. $L=\{a^n b^m c^n | n,m \geq 1\}$

   Solution:The language says that the number of a must be equal to the number of c but no restrictions on the number of bs.

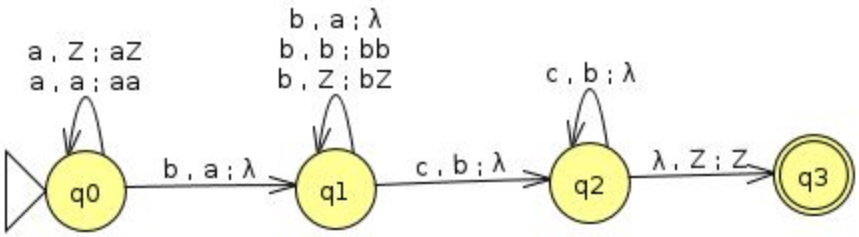


5. $L = \{a^{m+n} b^m c^n | n,m \geq 1\}$

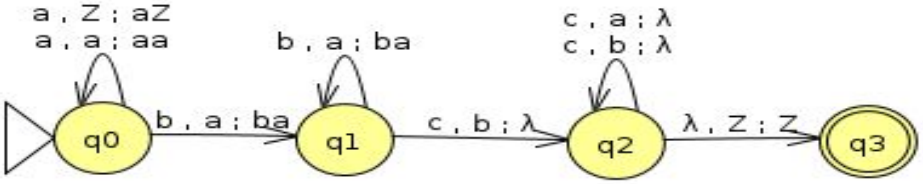
   Solution: a^{m+n} is written as $a^m a^n$, so the language becomes $a^n a^m b^m c^n$.



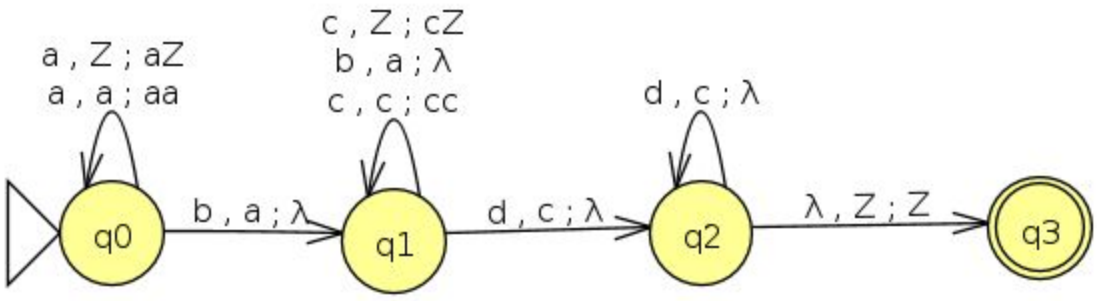
6. $L = \{a^n b^{n+m} c^m \mid n,m \geq 1\}$



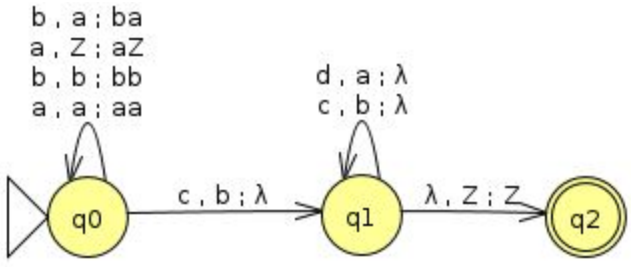
7. $L = \{a^n b^m c^{m+n} \mid n,m \geq 1\}$



8. $L = \{a^n b^n c^m d^m \mid n,m \geq 1\}$



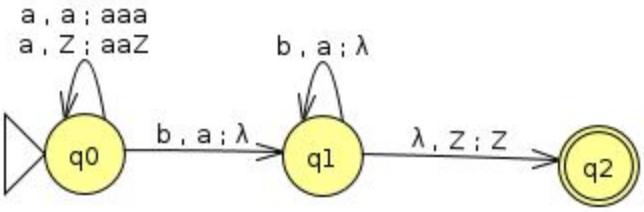
9. $L = \{a^n b^m c^m d^n \mid n,m \geq 1\}$



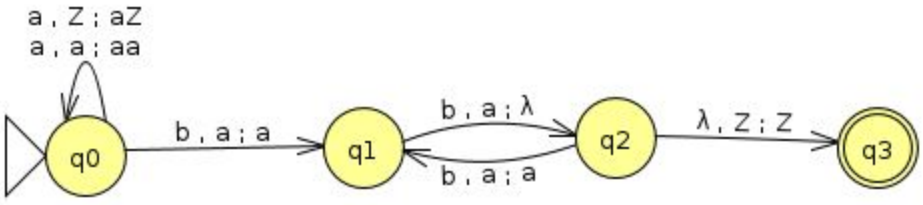
10. $L = \{a^n b^{2n} \mid n \geq 1\}$

Solution: There are two ways of solving this

- Push two a's by encountering one a

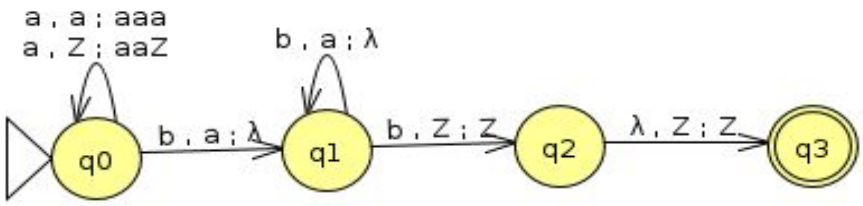


- For every two b's pops off one a



11.L=$\{a^n b^{2n+1} \mid n \geq 1\}$

Solution:



12.L =$\{wcw^R \mid w \in \{a,b\}\}$

Solution: Read symbols i.e., a's and b's off the tape until you reach the 'c'. As you read a's and b's push them on the stack . After reading the c, match the symbol read with the symbol popped off the stack until all symbols are read. If at any point the symbol read does not match the symbol popped, the machine "crashes".